The
BIOS Companion

Phil Croucher

Legal Bit

This book and any included software is sold as is without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Neither the Author, the Publisher nor its dealers or distributors assumes any liability for any alleged or actual damages arising from their use. **Translation**: Although this information has been gathered from original manufacturer's details or practical experience, it is always changing, or scarce, so there could be technical inaccuracies or typographical errors. As a result, changes will be made to the information in this book and included software without reference to anyone, and we don't guarantee that the product suits your purposes. As well, no liability is accepted for loss of data or business or damage to equipment as a result of using the information contained herein - backups are your responsibility!

Copyrights, etc

Sources

Which are gratefully acknowledged:

- Experience.
- Many conversations with technicians.
- Hundreds of motherboard manuals, not all of which were helpful!
- AMI BIOS Tech Ref manual.
- MR BIOS Tech Ref Manual. Thanks to Mike at Microid Research!
- Readers, including Mick O'Donnell, Martyn Smith, Chris Crook, Chris Nicholson, Dart Computers, Pat Tan, John Dallman, Ulf Boehlau, Rick and Tilman at ProData, Adrian Clint of Samsung, Peter Farrow, Kerry and Toni at Award Software, Chuck French at Unicore, Ali Kiafar at ECS/TTX, John Dann at ProData, Jerome Czeikus and Mike Echlin.
- **amibios.txt**, available from Jean-Paul Rodrigue in the University of Montreal, which had useful snippets, especially the explanation of Fast Decode.
- **amisetup**, a shareware program from Robert Muchsel.Copyrights, etc.

Praise For The BIOS Companion

"The computer book of the month is The Bios Companion by Phil Croucher. Long-time readers of this column will recall I have recommended his book before. This tells you everything you ought to know about the BIOS in your system. Post codes, options, upgrades, you name it. Years ago, I called an earlier edition of this invaluable and I see no reason to change my view. Recommended."

Jerry Pournelle, *Byte Magazine*

"You will find more information about your motherboard assembled here than I have ever seen."

Frank Latchford *PCCT*

"Thank! I really appreciated this. I read it and was able to adjust my BIOS settings so that my machine runs about twice as fast. Pretty impressive. Thanks again."

Tony

"This book is worth far more than is charged for it. Very well written. Probably the most-used reference book in my shop. ....a great value as the feature explanations trigger your thinking and allow you to figure out many related BIOS features in some of the newer versions."

Amazon reader

"For those who need or want to fine tune, or simply understand, the basic and advanced features of their PC's BIOS, this book is an invaluable guide. It has a very broad range and covers both fundamental and more advanced topics as well as issues specific to particular bios types (AMI, PHOENIX, etc.) and versions. This is one book you need to have as a PC technician and a valuable resource for trouble shooting and configuring your personal PC even if your not."

Amazon reader

"I found The Bios companion so useful that I "just have" to have all 3 books in the set. The extra Bios Companion is going to a friend who will gain great benefit from it. Yes I definitely want all three books. Thank you very much."

Mike Reinbolt

"I received my package today containing the BIOS Companion book and 2 CD set.... I'm really impressed with what I did receive. I already had about HALF of the information, and to get THAT much, I had to get several books and web pages. GOOD JOB!!

I had more time to go thru the book and think that you should change the word "HALF" to "FOURTH".

I commend you on the great job you did. That's a hell of a lot of work for any major company to do, let alone an individual."

Craig Stubbs

"I thoroughly enjoyed my purchase! The BIOS Companion is worth the cost just for the beep-code section alone. I am new to computers and have found the book and your site to be quite informative."

pcworker

"I thought the BIOS Companion was quite good. Just chock full of the kind of info I had been looking for. First book I've gotten that was worth the more than price I paid."

Tony

"While you are appreciative of my order, I am likewise appreciative of your efforts to make such a reference available. BIOS's are the most mysterious things in the computing world to figure out. I realize the BIOS manufactures have made great effort to provide detailed information in the BIOS help (F1) (ok, so that's a bit of sarcasm). Traditionally, I have had to piece bits of information together that I have found at various locations. Once again thanks."

Brian Presson, System Engineer

"The Bios companion is an absolute must for anyone who builds or configures PC's! It is by far worth the money you pay for it. Phil Croucher has done a superb job! He explains in great detail all of the settings that even most PC technicians have no idea of what they do or effect, and mostly some very helpful suggestions on system settings as well. An Absolute Must have!"

Larry Stark, LPG Computers
Memphis, TN

"I purchased the 2000 edition of the complete The BIOS Companion - PDF from DigiBuy today. Any way you look at it, the information contained is well worth the $15 dollar investment. I must personally thank you for publishing such a wonderful resource for techies such as myself. Thank you again for all of the hard work."

Sincerely, Boyd Stephens

"I spent two hours going through the different sections therein. Everything is there and I can only say, 'AWSOME'."

Robert, San Francisco

"Hi, Phil

The book is absolutely phenomenal !! - Congratulations ! This is exactly the kind of reference many people (including our instructors) need - everything in one place, beautifully organised, crammed full of essential, UNDERSTANDABLE, info."

Alain Hendrikse, South Africa

"Your BIOS guide I had from 1994 was one of those 'never throw it away' items that I knew I would need an update for."

Adrian Clint

Contents

1

THE BIOS

The instructions that turn a PC into a useful machine come in three stages, starting with application programs, which are loaded by an operating system, which in turn is loaded by a bootstrap loader in the BIOS (the *Basic Input/Output System*). There are several in a PC, a good example being the one on the video card that controls the interface between it and the computer. However, we are concerned with the *System BIOS*, which is a collection of assembly language routines that allow programs and the components of a PC to communicate with each other at low level. It therefore works in two directions at once and is active all the time your computer is switched on. In this way, software doesn't have to talk to a device directly, but can call a BIOS routine instead. However, the BIOS is quite an Achilles Heel and can produce many incompatibilities, so these days it is often bypassed by 32-bit software (DOS relied on it totally) - some functions have migrated to the operating system, starting with Power Management (see *ACPI*), but NT and W2K have long been replacing BIOS Code with their own *Hardware Abstraction Layer* (HAL) in the Shadowed ROM area traditionally used by the BIOS after the machine has started.

LinuxBIOS is an Open Source project aimed at replacing it with a little hardware initialization and a compressed Linux kernel that can be booted from a cold start (inside 3 seconds at last count). Linux, once bootstrapped, does not make use of BIOS calls, as it has all the low level hardware drivers itself. In addition, a "trusted BIOS" is being developed that can be included in any system that requires high assurance, such as NetTop. Some access to the Video BIOS is also allowed by some manufacturers.

For the moment, though, the System BIOS will work in conjunction with the *chipset*, which is really what manages access to system resources such as memory, cache and the data buses, and actually is the subject of this book, as all those advanced settings relate to the chipset and not the BIOS as such.

On an IBM-compatible, you will find the BIOS embedded into a ROM on the motherboard, together with hard disk utilities and a CMOS setup program, although this will depend on the manufacturer (the BIOS and CMOS are separate items). The ROM will usually occupy a 64K segment of upper memory at F000 in an ISA system, and a 128K segment starting at E000 with EISA or similar. It's on a chip so it doesn't get damaged if a disk fails, as sometimes used to happen on the Victor 9000/Sirius, which had the BIOS and system on the boot floppy.

Older machines, such as 286s, will have two ROMs, labelled *Odd* and *Even*, or *High* and *Low* (they must be in the right slots), because of the 16-bit bus, but these days there tends to be only one-look for one with a printed label (older 386s sometimes had 4). You can get away with one because BIOS code is often copied into *Shadow RAM* (explained later), and not actually executed from ROM, but from extended memory. In addition, much of the code is redundant once the machine has started, and it gets replaced by the operating system anyway. Some newer machines may actually have two single-chip BIOSes, so if one fails, the back-up kicks in. Well, in theory, anyway - there have been reports of the BIOSes flashing each other out, so later backups have become read-only.

A *Flash ROM* allows you to change BIOS code without replacing chip(s). Flash ROM, or *programmable read-only nonvolatile RAM*, if you want to be posh, is similar to the EEPROM, being a storage medium that doesn't need a continuous power source, but deals with several blocks of memory at once, rather than single bytes, making it slightly faster, but only just. Also, Flash devices can be programmed in situ, whereas EEPROMS need a special device.

Older BIOSes used EPROMs, which require ultra violet light to erase them, so were a more permanent solution. Even older BIOSes used PROMs, which can't be changed at all once programmed. All are *nonvolatile*, which means that they don't need a continuous source of power to keep information in them. Actually, this does include CMOS chips, as the power referred to is mains and not battery power, but the A+ exam might not agree.

BIOS DATA AREA

As well as ROM space, the BIOS takes 256 bytes of low memory as a *BIOS Data Area*, which contains details about the Num Lock state, keyboard buffer, etc. DOS, or whatever, loads higher than this, so it's quite safe. When power is applied, the BDA is created at memory location 0040:0000h. Here is what's in it::

Hex	Dec	Service	Size	Function
00h	0	Int 14h	2 bytes	Base I/O address for serial port 1 (COM 1)
02h	2	Int 14h	2 bytes	Base I/O address for serial port 2 (COM 2)
04h	4	Int 14h	2 bytes	Base I/O address for serial port 3 (COM 3)
06h	6	Int 14h	2 bytes	Base I/O address for serial port 4 (COM 4)
08h	8	Int 17h	2 bytes	Base I/O address for parallel port 1 (LPT 1)
0Ah	10	Int 17h	2 bytes	Base I/O address for parallel port 2 (LPT 2)
0Ch	12	Int 17h	2 bytes	Base I/O address for parallel port 3 (LPT 3)
0Eh	14	POST	2 bytes	Base I/O address for parallel port 4 (LPT 4)
10h	16	Int 11h	2 bytes	Equipment Word Bits 15-14 - parallel ports installed 00b = 1 parallel port 01b = 2 parallel ports 03b = 3 parallel ports Bits 13-12 are reserved Bits 11-9 - serial ports installed 000b = none 001b = 1 serial port 002b = 2 serial ports 003b = 3 serial ports 004b = 4 serial ports Bit 8 is reserved Bit 7-6 - floppy drives installed 0b = 1 floppy drive 1b = 2 floppy drives Bits 5-4 - video mode 00b = EGA or later 01b = color 40x25 10b = color 80x25 11b = monochrome 80x25

Hex	Dec	Service	Size	Function
				Bit 3 is reserved Bit 2 - PS/2 mouse 0b = not installed 1b = installed Bit 1 - math coprocessor 0b = not installed 1b = installed Bit 0 - boot floppy 0b = not installed 1b = installed
12h	18	POST	1 byte	Interrupt flag - Manufacturing test
13h	19	Int 12h	2 bytes	Memory size in Kb
15h	21		2 bytes	Error codes for AT+; Adapter memory size
17h	22	Int 16h	1 byte	Keyboard shift flags 1 Bit 7 - Insert 0b = Insert off 1b = Insert on Bit 6 - CapsLock 0b = CapsLock off 1b - CapsLock on Bit 5 - NumLock 0b = NumLock off 1b = NumLock on Bit 4 - ScrollLock 0b = ScrollLock off 1b = ScrollLock on Bit 3 - Alt key 0b = Alt key is up 1b = Alt key is down Bit 2 - Control key 0b = Control key is up 1b = Control key is down Bit 1 - Left Shift key 0b = Left Shift key is up 1b = Left Shift key is down Bit 0 - Right Shift key 0b = Right Shift key is up 1b = Right Shift key is down
18h	23	Int 16h	1 byte	Keyboard shift flags 2 Bit 7 - Insert key 0b = Insert key is up 1b = Insert key is down Bit 6 - CapsLock 0b = CapsLock is key is up 1b = CapsLock key is down Bit 5 - NumLock key 0b = NumLock key is up 1b = Numlock key is down Bit 4 - ScrollLock key 0b = ScrollLock key is up 1b = ScrollLock key is down Bit 3 - Pause key 0b = pause key is inactive 1b = Pause key is active Bit 2 - SysReg key 0b = SysReg key is up 1b = SysReg key is down

Hex	Dec	Service	Size	Function
				Bit 1 - Left Alt key 0b = Left Alt key is up 1b = Left Alt key is down Bit 0 - Right Alt key 0b = Right Alt key is up 1b = Right Alt key is down
19h	24	Int 09h	1 byte	Alt Numpad work area
1Ah	26	Int 16h	2 bytes	Pointer - next character in keyboard buffer
1Ch	28	Int 16h	2 bytes	Pointer - last character in keyboard buffer
1Eh	60	Int 16h	32 bytes	Keyboard buffer
3Eh	61	Int 13h	1 byte	Floppy disk drive calibration status Bits 7-4 are reserved Bit 3 = floppy drive 3 (PC, XT) Bit 2 = floppy drive 2 (PC, XT) Bit 1 = floppy drive 1 Bit 0 = floppy drive 0 0b not calibrated 1b calibrated
3Fh	62	Int 13h	1 byte	Floppy disk drive motor status Bit 7 - current operation 0b = read or verify operation 1b = write or format operation Bit 6 is not used Bit 5-4 - drive select 00b = Drive 0 01b = Drive 1 10b = Drive 2 (PC, XT) 11b = Drive 4 (PC, XT) Bit 3 - drive 3 motor 0b = motor off 1b = motor on Bit 2 - drive 2 motor 0b = motor off 1b = motor on Bit 1 - drive 0 motor 0b = motor off 1b = motor on
40h	63	Int 13h	1 byte	Floppy disk drive motor time-out
41h	64	Int 13h	1 byte	Floppy disk drive status Bit 7 - drive ready status 0b = drive ready 1b = drive not ready (time out) Bit 6 - seek status 0b = no seek error detected 1b = seek error detected Bit 5 - floppy disk controller test 0b = floppy disk controller passed 1b = floppy disk controller failed

Hex	Dec	Service	Size	Function
				Bit 4-0 error codes 00000b = no errors 00001b = illegal function requested 00010b = address mark not found 00011b = write protect error 00100b = sector not found 00110b = diskette change line active 01000b = DMA overrun 01001b = DMA boundary error 01100b = unknown media type 10000b = CRC error during read
42h	65	Int 13h	1 byte	Hard disk and floppy controller status register 0 Bit 7-6 - the interrupt code 00b = command completed normally 01b = abnormal termination 10b = abnormal termination, ready on, diskette changed 11b = seek command not completed Bit 5 - seek command 0b = seek command not completed 1b = seek command completed Bit 4 - drive fault 0b = no drive fault 1b = drive fault Bit 3 - drive ready 0b = drive ready 1b = drive not ready Bit 2 - head state when interrupt occurred 00b = drive 0 01b = drive 1 10b = drive 2 (PC, XT) 11b = drive 3 (PC, XT) Bit 1-0 indicates drive select 00b = drive 0 01b = drive 1 10b = drive 2 (PC, XT) 11b = drive 3 (PC, XT)
43h	66	Int 13h	1 byte	Floppy drive controller status register 1 Bit 7, 0b = no error 1b = access beyond last cylinder Bit 6, 0b = not used Bit 5, 1b = CRC error during read Bit 4, 1b = DMA overrun Bit 3, 0b = not used Bit 2, 1b = Sector not found or read ID fail Bit 1, 1b = medium write protected Bit 0, 1b = missing address mark
44h	67	Int 13h	1 byte	Floppy drive controller status register 2 Bit 7, 0b = not used Bit 6, 1b = deleted data address mark Bit 5, 1b = CRC error detected Bit 4, 1b = wrong cylinder Bit 3, 1b = condition of equal during verify Bit 2, 1b = sector not found during verify Bit 1, 1b = bad cylinder Bit 0, 1b = address mark not found on read
45h	68	Int 13h	1 byte	Floppy disk controller: cylinder number
46h	69	Int 13h	1 byte	Floppy disk controller: head number
47h	70	Int 13h	1 byte	Floppy disk controller: sector number

Hex	Dec	Service	Size	Function
48h	71		1 byte	Floppy disk controller: number of byte written
49h	72	Int 10h	1 byte	Active video mode setting
4Ah	74	Int 10h	2 bytes	Textcolumns per row for the active video mode
4Ch	76	Int 10h	2 bytes	Size of active video in page bytes
4Eh	78	Int 10h	2 bytes	Offset address of active video page relative to start of video RAM
50h	80	Int 10h	2 bytes	Cursor position for video page 0
52h	82	Int 10h	2 bytes	Cursor position for video page 1
54h	84	Int 10h	2 bytes	Cursor position for video page 2
56h	86	Int 10h	2 bytes	Cursor position for video page 3
58h	88	Int 10h	2 bytes	Cursor position for video page 4
5Ah	90	Int 10h	2 bytes	Cursor position for video page 5
5Ch	92	Int 10h	2 bytes	Cursor position for video page 6
5Eh	94	Int 10h	2 bytes	Cursor position for video page 7
60h	96	Int 10h	2 bytes	Cursor shape
62h	97	Int 10h	1 byte	Active video page
63h	99	Int 10h	2 bytes	I/O port address for the video display adapter
65h	100	Int 10h	1 byte	Video display adapter internal mode register Bit 7, 0b = not used Bit 6, 0b = not used Bit 5 0b = attribute bit background intensity 1b = attribute bit controls blinking Bit 4, 1b = mode 6 graphics operation Bit 3 - video signal 0b = video signal disabled 1b = video signal enabled Bit 2 - color operation 0b = color operation 1b = monochrome operation Bit 1, 1b = mode 4/5 graphics operation Bit 0, 1b = mode 2/3 test operation
66h	101	Int 10h	1 byte	Color palette Bit 7, 0b = not used Bit 6, 0b = not used Bit 5 - mode 5 foreground colors 0b = green/red/yellow 1b = cyan/magenta/white Bit 4 - background color 0b = normal background color 1b = intensified background color Bit 3 - intensified border color (mode 2) and background color (mode 5) Bit 2 - red Bit 1 - green Bit 0 - blue
67h	103		2 bytes	Adapter ROM offset address
69h	106		2 bytes	Adapter ROM segment address

Hex	Dec	Service	Size	Function
6Bh	107		1 byte	Last interrupt (not PC) Bit 7 - IRQ 7 0b = did not occur 01 = did occur Bit 6 - IRQ 6 0b = did not occur 01 = did occur Bit 5 - IRQ 5 0b = did not occur 01 = did occur Bit 4 - IRQ 4 0b = did not occur 01 = did occur Bit 3 - IRQ 3 0b = did not occur 01 = did occur Bit 2 - IRQ 2 0b = did not occur 01 = did occur Bit 1 - IRQ 1 0b = did not occur 01 = did occur Bit 0 - IRQ 0 0b = did not occur 01 = did occur
6Ch	111	Int 1Ah	4 bytes	Counter for Interrupt 1Ah
70c	112	Int 1Ah	1 byte	Timer 24 hour flag
71h	113	Int 16h	1 byte	Keyboard Ctrl-Break flag
72h	115	POST	2 bytes	Soft reset flag
74h	116	Int 13h	1 byte	Status of last hard disk operation 00h = no errors 01h = invalid function requested 02h = address mark not found 04h = sector not found 05h = reset failed 06h = removable media changed 07h = drive parameter activity failed 08h = DMA overrun 09h = DMA boundary overrun 0Ah = bad sector flag detected 0Bh = bad track detected 0Dh = invalid number of sectors on format 0Eh = control data address mark detected 0Fh = DMA arbitration level out of range 10h = uncorrectable ECC or CRC error 11h = ECC corrected data error 20h = general controller failure 40h = seek operation failed 80h = timeout AAh = drive not ready BBh = undefined error occurred CCh = write fault on selected drive E0h = status error or error register is zero FFh = sense operation failed
75h	117	Int 13h	1 byte	Number of hard disk drives

Hex	Dec	Service	Size	Function
76h	118	Int 13h	1 byte	Hard disk control byte Bit 7 0b = enables retries on disk error 1b = disables retries on disk error Bit 6 0b = enables reties on disk error 1b = enables reties on disk error Bit 5, 0b = not used Bit 4, 0b = not used Bit 3 0b = drive has less than 8 heads 1b = drive has more than 8 heads Bit 2, 0b = not used Bit 1, 0b = not used Bit 0, 0b = not used
77h	119	Int 13h	1 byte	Offset address of hard disk I/O port (XT)
78h	120	Int 17h	1 byte	Parallel port 1 timeout
79h	121	Int 17h	1 byte	Parallel port 2 timeout
7Ah	122	Int 17h	1 byte	Parallel port 3 timeout
7Bh	123		1 byte	Parallel port 4 timeout (PC, XT) support for virtual DMA services (VDS) Bit 7, 0b = not used Bit 6, 0b = not used Bit 5 - virtual DMA services 0b = not supported 1b = supported Bit 4, 0b = not used Bit 3 - chaining on interrupt 4Bh 0b = not required 1b = required Bit 2, 0b = not used Bit 1, 0b = not used Bit 0, 0b = not used
7Ch	124	Int 14h	1 byte	Serial port 1 timeout
7Dh	125	Int 14h	1 byte	Serial port 2 timeout
7Eh	126	Int 14h	1 byte	Serial port 3 timeout
7Fh	127	Int 14h	1 byte	Serial port 4 timeout
80h	129	Int 16h	2 bytes	Starting address of keyboard buffer
82h	131	Int 16h	2 bytes	Ending address of keyboard buffer
84h	132	Int 10h	1 byte	Number of video rows (minus 1)
85h	134	Int 10h	2 bytes	Number of scan lines per character
87h	135	Int 10h	1 byte	Video display adapter options Bit 7 - bit 7 of last video mode 0b = clear display buffer setting mode 1b = do not clear the display buffer Bit 6-4 - memory on video adapter 000b = 64Kb 001b = 128Kb 010b = 192Kb 011b = 256Kb 100b = 512Kb 110 = 1024Kb or more Bit 3 - video subsystem 0b = not active 1b = active Bit 2 is reserved

Hex	Dec	Service	Size	Function
				Bit 1 - monitor type 0b = color 1b = monochrome Bit 0 - alphanumeric cursor emulation 0b = disabled 1b = enabled
88h	136	Int 10h	1 byte	Video display adapter switches Bit 7 - state of feature connector line 1 Bit 6 - state of feature connector line 0 Bit 5-4 not used Bit 3-0 - adapter type switch settings 0000b = MDA/color 40x25 0001b = MDA/color 80x25 0010b = MDA/high-resolution 80x25 0011b = MDA/high-res enhanced 0100b = CGA 40x25/monochrome 0101b = CGA 80x25/monochrome 0110b = color 40x25/MDA 0111b = color 80x25/MDA 1000b = high-resolution 80x25/MDA 1001b = high-res enhanced/MDA 1010b = monochrome/CGA 40x25 1011b = monochrome/CGA 80x25
89h	137	Int 10h	1 byte	VGA video flags 1 Bit 7 and 4 - scanline mode 00b = 350-line mode 01b = 400-line mode 10b = 200-line mode Bit 6 - display switch 0b = disabled 1b = enabled Bit 5 is reserved Bit 3 - default palette loading 0b = disabled 1b= enabled Bit 2 - monitor type 0b = color 1b = monochrome Bit 1 - gray scale summing 0b = disabled 1b = enabled Bit 0 - VGA active state 0b = VGA inactive 1b = VGA active
8Ah	138	Int 10h	1 byte	VGA video flags 2
8Bh	139	Int 13h	1 byte	Floppy disk configuration data Bit 7-6 - last data sent to controller 00b = 500 Kbit/sec/sec 01b = 300 Kbit/sec 10b = 250 Kbit/sec 11b = rate not set or 1 Mbit/sec Bit 5-4 - last drive steprate to controller 00b = 8ms 01b = 7ms 10b = 6ms 11b = 5ms Bit 3-2 - data rate, set at start (Bits 7-6) Bit 1-0 not used

Hex	Dec	Service	Size	Function
8Ch	140	Int 13h	1 byte	Hard disk drive controller status
				Bit 7 - controller state
				0b = controller not busy
				1b = controller busy
				Bit 6 indicates drive ready state
				0b = drive selected not ready
				1b = drive selected ready
				Bit 5 - write fault
				0b = write fault did not occur
				1b = write error occurred
				Bit 4 - seek state
				0b = drive selected seeking
				1b = drive selected seek complete
				Bit 3 - data request
				0b = data request is inactive
				1b = data request is active
				Bit 2 - data correction
				0b = data not corrected
				1b = data corrected
				Bit 1 - index pulse state
				0b = index pulse inactive
				1b = index pulse active
				Bit 0 - error
				0b = no error
				1b = error in previous command
8Dh	141	Int 13h	1 byte	Hard disk drive error
				Bit 7 - bad sector
				0b = not used
				1b = bad sector detected
				Bit 6 - ECC error
				0b = not used
				1b = uncorrectable ECC error
				Bit 5 - media state
				0b = not used
				1b = media changed
				Bit 4 - sector state
				0b = not used
				1b = ID or target sector not found
				Bit 3 - media change request state
				0b = not used
				1b = media change requested
				Bit 2 - command state
				0b = not used
				1b = command aborted
				Bit 1 - drive track error
				0b = not used
				1b = track 0 not found
				Bit 0 - address mark
				0b = not used
				1b = address mark not found
8Eh	142	Int 13h	1 byte	Hard disk drive task complete flag
8Fh	143	Int 13h	1 byte	Floppy disk drive information
				Bit 7 not used
				Bit 6 - drive 1 type determination
				0b = not determined
				1b = determined
				Bit 5 - drive 1 multirate status
				0b = no
				1b = yes

Hex	Dec	Service	Size	Function
				Bit 4 - diskette 1 change line detection 0b = no 1b = yes Bit 3 not used Bit 2 - drive 0 type determination 0b = not determined 1b = determined Bit 1 - drive 0 multirate status 0b = no 1b = yes Bit 0 - diskette 0 change line detection 0b = no 1b = yes
90h	144	Int 13h	1 byte	Diskette 0 media state Bit 7-6 - transfer rate 00b = 500 Kbit/sec 01b = 300 Kbit/sec 10b = 250 Kbit/sec 11b = 1 Mbit/sec Bit 5 - double stepping 0b = not required 1b = required Bit 4 - media in floppy drive 0b = unknown media 1b = known media Bit 3 not used Bit 2-0 - last access 000b = 360k media in 360K drive 001b = 360K media in 1.2M drive 010b = 1.2M media in 1.2M drive 011b = known 360K media 360K drive 100b = known 360K media in 1.2M drive 101b = known 1.2M media in 1.2M drive 110b = not used 111b = 720K media in 720K drive or 1.44M media in 1.44M drive
91h	145	Int 13h	1 byte	Diskette 1 media state As for Diskette 0
92h	146	Int 13h	1 byte	Diskette 0 operational starting state Bit 7 - data transfer rate 00b = 500 Kbit/sec 01b = 300 Kbit/sec 10b = 250 Kbit/sec 11b = 1 Mbit/sec Bits 5-3 not used Bit 2 - drive determination 0b = drive type not determined 1b = drive type determined Bit 1 - drive multirate status 0b = drive is not multirate 1b = drive is multirate Bit 0 - change line detection 0b = no change line detection 1b = change line detection
93h	147	Int 13h	1 byte	Diskette 1 operational starting status As for Diskette 0
94h	148	Int 13h	1 byte	Diskette 0 current cylinder
95h	149	Int 13h	1 byte	Diskette 1 current cylinder

Hex	Dec	Service	Size	Function
96h	150	Int 16h	1 byte	Keyboard status flags 3 Bit 7, 1b = reading 2 byte keyboard ID Bit 6, 1b = last code was first ID character Bit 5, 1b = forced Numlock on Bit 4 - 101/102 key keyboard 0b = present 1b = not present Bit 3 - right alt key active 0b = not active 1b = active Bit 2 - right control key active 0b = not active 1b = active Bit 1, 1b = last scancode was E0h Bit 0, 1b = last scancode was E1h
97h	151	Int 16h	1 byte	Keyboard status flags 4 Bit 7, 1b = keyboard transmit error Bit 6, 1b = LED update in progress Bit 5, 1b = re-send code received Bit 4, 1b = acknowledge code received Bit 3, 1b = reserved Bit 2 indicates CapsLock LED state 0b = CapsLock LED off 1b = CapsLock LED on Bit 1 indicates NumLock LED state 0b = NumLock LED off 1b = NumLock LED on Bit 0 indicates ScrollLock LED state 0b = ScrollLock LED off 1b = ScrollLock LED on
98h	155		4 bytes	Segment:Offset address of user wait flag pointer
9Ch	159		4 bytes	User wait count
A0h	160		1 byte	User wait flag Bit 7, 1b = wait time has elapsed Bit 6, 1b not used Bit 0 - wait progress 0b = no wait in progress 1b = wait in progress
A1h	167		7 bytes	Local area network (LAN) bytes
A8h	171		4 bytes	Segment:Offset address of video parameter control block
ACh	239		68 bytes	Reserved
F0h	255		16 bytes	Intra-applications communications area

There are several types of BIOS because so many computers need to be IBM-compatible; they're not allowed to copy each other, for obvious reasons. The BIOS worries about all the differences and presents a standard frontage to the operating system, which in turn provides a standard interface for application programs. PC and motherboard manufacturers used to make their own BIOSes, and many still do, but most are now based on code from third party companies, the most well-known of which are Phoenix, Award, Microid Research and American Megatrends (AMI). However, all is not what it seems! Award Software owns Unicore (aka **esupport.com**, the upgraders), which in turn owns MR, which does the customised stuff. Phoenix also owns Quadtel and has merged with Award.

WHAT HAPPENS WHEN YOU SWITCH ON

The (x86) CPU is programmed to read the address space at FFFF:0000h, the last 16 bytes of memory in the first megabyte, which is just large enough to contain a jump command (JMP) that tells the processor where to find the BIOS code it is looking for (this is the bootstrap process). Next, the *Power On Self Test* (or POST) is run, to ensure the hardware is working (see the listings for each manufacturer to see what is actually done). During the POST, the BIOS will look for a video BIOS between C000:000h and C780:000h, and test its checksum, after which it will allow the video BIOS to initialise itself and retake control afterwards (you will see the manufacturer's logo and various ID strings on the screen). Then the area between C800:000h to DF80:0000h will be searched in 2 K increments, looking for other ROMs. They, too, will be initialised after a checksum test.

The memory area at 0000:0472h contains a flag which will tell the BIOS if a cold or warm boot has occurred (a value of 1234h means it is a warm boot. Being in *little endian format*, where the least significant byte comes first, it will be in memory as 3412). A warm boot means that most of the POST can be skipped. Once the POST is over, the BIOS looks for an operating system in various locations. Traditionally, the order is the first floppy then the first hard drive, but you can change all that in the CMOS, to include CD ROM drives, Zip drives, etc.

If the floppy drive has a bootable disk in it, the BIOS will load sector 1, head 0, cylinder 0 into memory, starting at 0000:7C00h.

HOW OLD IS MY BIOS?

If you want to check how old your BIOS is, the date is on the start-up screen, usually buried in the BIOS ID String, which looks a bit like this (**121291** is the date in this AMI sample):

```
40-0201-BY6379-01101111-121291-UMCAUTO-04
```

If you don't get one, you can also use **debug**. The BIOS lives between F000:0000 and F000:FFFF, with copyright messages typically at F000:E000, F000:C000 and F000:0000.

Type:

```
debug
```

at the DOS prompt. A minus sign will appear. Press *D* followed by an address in memory to see the 128 bytes' worth of the values stored there, for example:

```
-d f000:e000
```

You can also use the *S* command to search for the word "version", although some computers, IBM and Compaq, for example, don't use version numbers. In this case, the date will be near F000:FFE0. Quit **debug** by pressing *q* at the prompt. The AMI WinBIOS has a normal date on the startup screen. Otherwise, as you can see, you don't just get the date - many manufacturers include extras that identify the state of the chipset inside. For example, with the AMI Hi-Flex BIOS, there are two more strings, displayed by pressing **Ins** during bootup, or any other key to create an error condition.

IDENTIFYING YOUR BIOS

Acer ID Strings

In the bottom left hand corner of the screen:

```
ACR89xxx-xxx-950930-R03-B6
```

The first 2 characters after ACR identify the motherboard (see table). The last few are the BIOS revision. The ones before that are the date (e.g. **950930**).

ID	Board	Product	ID	Board	Product
05	X1B	Altos 19000	4B	V55LA-2M	Acros, Power, Aspire
07	M7	Altos 900 & 9000M	5A	X3	Altos 19000 Pro 4
19	V55-2	Acros, Power	62	V65X	AcerAcros PII
1A	M3A	Altos 300	63	V58	Entra
1B	V35	Power	67	V65LA	Acros, Power
22	V50LA-N	Acros, Power	6B	A1G4	Acros
24	M9B	Altos 9000/Pro	6D	V20	AcerPower
25	V55LA	Acros, Power, Aspire	89	M5	Altos 7000P
29	V60N	AcerPower	8F	M3 (SCSI)	Altos 9000
2F	M11A	Altos 900/Pro	8F	M3-EIDE	AcerPower (590)
30	V56LA	Acros, Power, Aspire	99	A1GX, -2	Acros, Power
33	V58LA	Acros, Power, Aspire	9A	V30, -2	Acros, Power
35	V35N	Acros, Power	9C	V12LC, -2X	Acros, Power, Aspire
46	M9N	Altos 920 and 9100			

ALR (Gateway) ID Strings

BIOS ID Begins	Motherboard
SU81010A	E-1400
0AAGT	E-1000
0AAKW	PII
404CL0X0	PII
4D4KL0X0	Dual PII
4J4NB0X1	Pentium
4K4UE0X1	E-1200
4M4PB0X1	PII
4M4SG0X0	PII
4R4CB0XA	Pentium 440BX

AMI ID Strings

The release number is at the top left of the screen for AMI boards. The ID string is at the bottom left. The AMI BIOS and BIOS Plus series (1986-1990) looks like this (for example):

```
DINT-1123-04990-K8
```

Or, in other words:

```
aaaa-bbbb-mmddyy-Kc
```

where:

```
aaaa   BIOS type
bbbb   Customer Number
mmddyy Release date
Kc     Keyboard BIOS version number
```

If the first customer number (in bold above) is **1**, **2**, **8** or a **letter**, it is a non-AMI Taiwanese motherboard. If it is **3**, **4** or **5**, it is from AMI. **50** or **6** means a non-AMI US motherboard and **9** means an evaluation BIOS for a Taiwanese manufacturer. Otherwise, there can be up to three lines (from 1991 onwards) at the bottom left of the screen. The first is displayed automatically, the other two can be seen by pressing the **Insert** key. Aside from version numbers, the 1s and 0s indicate the state of the settings inside. The Hi-Flex BIOS might look like this (from 1991):

41-0102-zz**5123**-00111111-101094-AMIS123-P

Again, check the bold numbers in the third set for the manufacturer.

NON-AMI TAIWANESE BOARDS (1XXX, 8XXX)

Code	Manufacturer	Code	Manufacturer
1003	QDI	1514	Wuu Lin
1045	Vtech/PC Partner	1519	Epox
1101	Sunlogix	1526	Eagle
1102	Soyo	1531	Force
1103	Tidalpower	1540	BCM
1105	Autocomputer	1546	Golden Horse
1106	Dynasty	1549	CT Continental
1107	Dataexpert	1564	Random Technology
1108	Chaplet	1576	Jetta
1109	Fair Friend	1585	Gleem
1111	Paoku	1588	Boser
1112	Aquarius Systems	1593	Advantech
1113	MicroLeader	1594	Trigon
1114	Iwill	1608	Consolidated Marketing
1115	Senior Science	1612	Datavan
1116	Chicony	1617	Honotron
117	A-Trend	1618	Union Genius
1120	Unicorn	1621	New Paradise
1121	First International	1622	RPT Intergroups
1122	MicroStar/NoteStar	1628	Digital Eqpt Intl
1123	Magtron	1630	Iston
1124	Tekram	1647	Lantic
1126	Chuntex	1652	Advanced Semiconductor
1128	Chaintech	1655	Kingston Tech
1130	Pai Jung	1656	Storage System
1131	ECS (Elite Group)	1658	Macrotek
1132	Dkine	1666	Cast Technology
1133	Seritech	1671	Cordial Far East
1135	Acer	1672	Lapro

Code	Manufacturer	Code	Manufacturer
1136	Sun Electronics	1675	Advanced Scientific
1138	Win Win	1685	High Ability
1140	Angine	1691	Gain Technology
1141	Nuseed	1700	DSG Technology
1142	Firich	1707	Chaining Computer
1143	Crete	1708	E-San
1144	Vista	1719	Taiwan Turbo
1146	Taste	1720	Fantas
1147	Integrated Tech Express	1723	NTK
1150	Achitec	1727	Tripod
1151	Accos1	1737	Ay Ruey
1152	Top-Thunder	1739	Jetpro
1154	San Li	1743	Mitac
1156	Technical House	1759	Bek-Tronic
1158	Hi-Com	1762	Ansoon
1159	Twinhead	1770	Acer Incorp.
1161	Monterey Intl	1771	Toyen
1163	Softek	1774	Acer Sertek
1165	Mercury	1776	Joss
1168	Rio Works	1780	Acrosser
1169	MicroStar	1783	Efar
1170	Taiwan Igel	1788	Systex
1171	Shining Yuan	1792	U-board
1172	Giantec	1794	CMT
1175	Applied Component Tech	1796	J & J
1176	Sigma	1800	Syzygia
1177	High Tech Information	1801	Palit
1178	Clevo	1806	Interplanetary Info
1180	Paladin	1807	Expert
1181	Leo Systems (FIC)	1810	Elechands Intl
1182	Alpha-Top	1815	Powertech
1183	Mirle Automation	1820	Ovis
1184	Delta Electronics	1823	Inlog Micro
1188	Quanta	1826	Tercomputer
1190	Chips & Technologies	1827	Anpro
1192	Interlogic Industries/ICP	1828	Axiom
1193	Sercom	1840	New Union KH
1195	GNS	1845	PC Direct/Proware
1196	Universal Scientific	1846	Garnet Intl
1197	Golden Way	1847	Brain Power
1199	Gigabyte	1850	HTR Asia Pacific
1201	New Tech Intl	1853	Veridata
1203	Sunrex	1856	Smart D & M
1204	Bestek	1867	Lutron
1209	Puretek	1868	Soyo
1210	Rise	1879	Aeontech Intl
1211	DFI	1881	Manufacturing Tech
1214	Rever Computer	1888	Seal Intl
1218	Elite Computer	1889	Rock

Code	Manufacturer	Code	Manufacturer
1221	Darter tech	1906	Freedom Data
1222	Domex	1914	Aquarius Systems
1223	BioStar	1917	Source of Computer
1225	Yung Lin	1918	Lanner
1229	Dataworld Intl	1920	Ipex ITG Intl
1234	Leadman Electronics	1924	Join Corp
1235	Formosa Industrial	1926	Kou Sheng
1238	Win Tech	1927	Seahill Tech
1240	Free Computer	1928	Nexcom Intl
1241	Mustek	1929	CAM Enterprise
1242	Amptek	1931	Aaeon Techlogu
1244	Flytech	1932	Kuei Hao
1246	Cosmotech	1933	ASMT
1247	Abit	1934	Silver Bally
1248	Muse	1935	Prodisti
1251	Portwell	1936	Codegen
1252	Sono Computer	1937	Orientech
1256	Lucky Star	1938	Project Info
1258	Four Star	1939	Arbor
1259	GVC	1940	Sun Top
1260	DT Research	1941	Funtech
1262	Arima	1942	Sunflower
1266	Modula	1943	Needs System
1270	Portwell	1945	Norm Advanced
1271	Tidal	1947	Ten Yun
1272	Ultima Electronics	1948	Beneon
1273	UFO Systems	1949	National Advantage
1274	Full Yes	1950	MITS
1275	Jackson Dai Industrial	1951	Macromate
1276	Jetway	1953	Orlycon
1277	Tarng Bow	1954	Chung Yu
1281	EFA	1955	Yamashita
1283	Advance Creative	1957	High Large
1284	Lung Hwa	1958	Young Micro
1286	Askey Computer	1959	Fastfame
1291	TMC	1960	Acqutek
1292	Asustek	1961	Deson Trade
1297	DD&TT	1962	Atra Comms
1298	Trigem	1963	Dimensions Electronics
1299	Trigem	1964	Micron design
1301	Taken	1965	Cantta
1304	Dual Enterprises	1968	Khi Way
1306	Sky Computer Europe	1969	Gemlight
1309	Protronic	1970	MAT
1317	New Comm	1973	Fugutech
1318	Unitron	1974	Green Taiwan
1323	Inventec	1975	Supertone
1343	Holco	1977	AT&T
1346	Snobol	1978	Winco

Code	Manufacturer
1351	Singdak
1353	J Bond
1354	Protech
1355	Argo Systems
1357	Portwell
1367	Coxswain
1371	ADI
1373	SiS
1379	Win Technolgies
1391	Aten Intl
1392	ACC
1393	Plato Technology
1396	Tatung
1398	Spring Circle
1400	Key Win Electronics
1404	Alptech
1421	Well Join
1422	Labway
1425	Lindata
1437	Hsing Tech
1440	Great Electronics
1450	Win-Lan
1451	Ecel Systems
1452	United Hitech
1453	Kai Mei
1461	Hedonic
1462	Arche
1470	Flexus
1471	CP technology
1472	Datacom
1473	PC Chips
1484	Mitac
1490	Great Tek
1491	President Technology
1493	Artdex
1494	Pro Team
1500	Netcon/Foxen Co
1503	Up Right

Code	Manufacturer
1980	Teryang
1981	Nexcom
1982	China Semiconductor
1985	Top Union
1986	DMP
1988	Concierge
1989	Atherton
1990	Expentech
1994	CBR (Japan Cerebro)
1996	Ikon
1998	Chang Tseng
2100	Kapok
2292	Olivetti
6069	Ocean Tech
6081	CSS Labs
6082	Pioneer Computers
6105	Dolch
6132	Technology Power
6165	Genoa
6182	Peaktron
6214	HP
6259	Young Micro
6285	Tyan
6326	Crystal
6328	Alaris
6347	Teknor
6386	Pacific Information
6389	Super Micro
6399	Mylex
6407	Elonex
6423	American Predator
8003	QDI
8005	AVT Industrial
8031	Zida
8045	PC Partner (VTech)
8054	Pine
8078	Weal Union

NON-AMI USA BOARDS (6XXX)

Code	Manufacturer	Code	Manufacturer
105	Dolch	326	Crystal
132	Tech Power Enterprises	386	Pacific Info
156	Genoa	389	Supermicro
259	Young Micro		

ID STRING LINE 1

12_4-7_9-14_16-23_25-30_32-39_41 decodes as follows:

Byte	Description		
1	Processor Type	0 2 3	8086/8 80286 80386, 80486, Pentium
2	Size of BIOS	0 1	64K 128K
4-5	Major Version Number		
6-7	Minor Version Number		
9-14	Reference Number		
16	Halt on Post Error	Set to 1 if On	
17	Initialize CMOS every boot	Set to 1 if On	
18	Block pins 22 & 23 of keyboard controller	Set to 1 if On	
19	Mouse support in BIOS/keyboard controller	Set to 1 if On	
20	Wait for if error found	Set to 1 if On	
21	Display Floppy error during POST	Set to 1 if On	
22	Display Video error during POST	Set to 1 if On	
23	Display Keyboard error during POST	Set to 1 if On	
25-26	BIOS Date	Month (1-12)	
27-28	BIOS Date	Date (1-31)	
29-30	BIOS Date	Year (0-99)	
32-39	Chipset Identification	BIOS Name	
41	Keyboard controller version number		

ID STRING LINE 2

123 5_7-10_12-13_15-16_18-21_23-24_26-27_29-31

Byte	Description
1-2	Pin no for clock switching through keyboard controller
3	High signal on pin switches clock to High(H) or Low (L)
5	Clock switching through chipset registers 0=Off 1=On
7-10	Port address to switch clock high through special port
12-13	Data value to switch clock high through special port
15-16	Mask value to switch clock high through special port
18-21	Port Address to switch clock low through special port
23-24	Data value to switch clock low through special port
26-27	Mask value to switch clock low through special port
29-31	Turbo Sw Input Pin info (Pin no for Turbo Sw Input Pin)

ID STRING LINE 3

1-3 5 7-10 12-13 15-16 18-21 23-24 26-27 29-30 31 33

Byte	Description
1-2	Keyboard Controller Pin no for cache control
3	Keyboard Controller Pin number for cache control
5	High signal is used on the Keyboard Controller pin
7-10	Cache Control through Chipset Registers. 0= control off 1= Control on
12-13	Port Address to enable cache through special port
15-16	Data value to enable cache through special port
18-21	Mask value to enable cache through special port
23-24	Port Address to disable cache through special port
26-27	Data value to disable cache through special port
29-30	Mask value to disable cache through special port
31	Pin number for Resetting 82335 Memory controller.
33	BIOS Modified Flag; Incremented each time BIOS is modified from 1-9 then A-Z and reset to 1. If 0 BIOS has not yet been modified.

INTEL

The AMI version number looks like this when used on Intel motherboards:

```
1.00.XX.??Y
```

where:

```
XX     BIOS version number
??     Intel Motherboard model
Y      Usually 0 or 1
```

1.00.07.DH0 would be BIOS version 7 and a TC430HX (Tucson) motherboard.

AOpen ID Strings

Normally starts with **R** and found in between the model name and the date:

AP58 **R1.00** July.21.1997

Award ID Strings

The date is at the front:

05/31/94-OPTI-596/546/82-2A5UI**M2**00-00

The next bit is the chipset and the next to last the Part Number, of which characters 6 and 7 identify the manufacturer (**M2**). The first 5 letters (of the part number) refer to the chipset (here **2A5UI**) and the last 2 (**00**) are the model number. An *i* suffix means an Intel 12v Flash ROM, and *s* refers to an SST 5v (the difference is where ESCD is stored in upper memory).

BYTE CODE

Byte No	Manufacturer	Code	Manufacturer
1	Version	1	BIOS, pre 4.2
		2	4.5x Elite
		3	PowerBIOS 5.0
		4	Cardware PCMCIA
		5	CAMPliant SCSI
		6	6.0 Medallion
2	Bus Type	1	ISA
		2	PS/2 (Micro Channel)
		3	EISA
		5	EISA/ISA
		7	N/A
		A	ISA/PCI
		B	EISA/PCI
		C	ISA/PM
		D	EISA/PM
		E	PCI/PnP
3	CPU Type	4	486
		5	586
		6	686
		9	New unknown type
		U	Universal
4-5	Chipset	(9K)	See below
6-7	Customer ID	(S2)	See below
8-9	Customer Project	GC	
8	Location	A	USA
		E	End User
		S	Sample
		P	Return

MANUFACTURER ID

Code	Manufacturer	Code	Manufacturer
00	Unknown (Micom + others)	L7	Lanner
99	Beta Unknown	L9	Lucky Tiger
A0	Asustek	LB	Leadtek/Super Micro
A1	Abit (Silicon Star)	M0	Matra
A2	A-Trend	M2	Mycomp (TMC)/Interlogic
A3	ASI (Aquarius)/BCOM	M3	Mitac
A5	Axiom	M4	Micro-Star (Achme)
A7	Arima Taiwan AVT (Concord)	M8	Mustek
A8	Adcom	M9	MLE
AB	Aopen (Acer)	MH	Macrotek/Interlogic
AC	Spica?	MP	Maxtium
AD	Amaquest/Anson	N0	Nexcom
AK	Advantech/Aaeon	N5	NEC
AM	Mirage/Acme/Achme	NM	NMC

Code	Manufacturer
AT	ASK
AX	Achitec
B0	Biostar
B1	Bestkey/BEK-Tronic
B2	Boser
B3	BCM
C0	Matsonic
C1	Clevo
C2	Chicony
C3	ChainTech
C5	Chaplet
C9	Computrend
CF	Flagpoint
CS	Gainward or CSS
CV	California Graphics
D0	Dataexpert
D1	DTK (also Gemlight/Advance)
D2	Digital
D3	Digicom
D4	DFI (Diamond Flower)
D7	Daewoo
D8	Nature Worldwide
DE	Dual Technology
DI	Domex (DTC)
DJ	Darter
DL	Delta
E1	ECS (Elite Group)
E3	EFA
E4	ESP Co
E6	Elonex
E7	Expen Tech
EC	ENPC/EONTronics
EN	ENPC
EO	Evalue
F0	FIC
F1	Flytech
F2	Freetech/Flexus
F3	FYI (Full Yes)
F5	Fugutech
F8	Formosa
F9	Fordlian/Redfox/BravoBaby
FD	DataExpert/Atima/GCT?
FG	Fastfame
FH	Amptron?
FI	FIC
FN	Amptron?
G0	Gigabyte
G1	GIT
G3	Gemlight

Code	Manufacturer
NX	Nexar
O0	Ocean/Octek
P1	PC-Chips
P4	Asus
P6	SBC/Protech
P8	Azza/Proteam
P9	Powertech
PA	Pronix (Epox)
PC	Pine
PF	President
PK	ALD technology
PN	Crusader/Procomp
PR	Super Grace
PS	Palmax (notebooks)
PX	Pionix
Q0	Quanta
Q1	QDI
R0	Rise (Mtech)
R2	Rectron
R3	Datavan
R9	RSAptek
RA	Rioworks
RC	Arstoria
S2	Soyo
S3	Smart D&M
S5	Shuttle (Holco)
S9	Spring Circle
SA	Seanix/Yukon
SC	Sukjung (Auhua)
SE	SMT (Sundance?) Newtech
SH	SYE (Shing Yunn)
SJ	Sowah
SL	Winco
SM	San-Li/Hope Vision
SM	SMT (Superpower)
SN	Soltek
SW	S & D, A-Corp, Zaapa
SX	Super Micro
T0	Twinhead
T1	Taemung/Fentech
T4	Taken
T5	Tyan
T6	Trigem
TB	Totem/Taeli
TG	Tekram
TJ	Totem
TL	Transcend
TP	Commate/Ozzo?
TR	Topstar

Code	Manufacturer	Code	Manufacturer
G5	GVC	TT	T & W
G9	Global Circuit/Supertek/CP	TX	Tsann Kuen
GA	Giantec	TY	Aeton
GE	Zaapa/Globe Legate	U0	Uboard/Teknor
H0	HsinTech/PC Chips	U1	USI
H2	Holco (Shuttle)	U2	AIR (UHC)
H9	HsinTech	U3	Umax
HH	Hightec	U4	Unicom
HJ	Sono	U5	Unico
I3	Iwill	U6	Unitron
I4	Inventa	U9	Warp Speed
I5	Informtech	V3	Vtech (PC Partner)/S Grace
I7	Inlog Microsystem	V5	Vision Top
I9	ICP	V6	Vobis
IA	Infinity	V7	YKM (Dayton Micron)
IC	Inventech (notebooks)	W0	Wintec (Edom)
IE	Itri	W1	WellJoin
J1	Jetway (Jetboard, Acorp)/Jmark	W5	Winco
J2	Jamicon	W7	Winlan
J3	J-Bond	W9	Weal Union
J4	Jetta	XA	ADLink
J6	Joss	X3	A-Corp
K0	Kapok	X5	Arima
K1	Karnei	Y2	Yamashita
KF	Kinpo	Z1	Zida
L1	Lucky Star/Luckstar	Z3	ShenZhen Zeling

CHIPSET ID

Code	Chipset	Code	Chipset
154I5	SiS 85C431/ 85C420/	2A5UN	Opti Viper-M 82C556/557/558
213V1	SARC RC2018	2A5UP	Opti Viper Max
21480	HiNT (Sierra) HMC82C206	2A5X7	UMC 82C890
214D1	As above	2A5X8	UM8886BF/8891BF/8892BF
214I8	SiS 85C471	2A5XA	UMC 890C
21419	E version of above	2A69H	440FX
214L2	VIA VT82C486A	2A69J	440LX/EX
214L6	Venus VT82C486A/VT82C496G	2A69K	440 BX/ZX
214W3	VD88C898	2A69L	Camino 820
214X2	UMC 491	2A69M	Whitney 810
215UM	OPTi 82C546/82C597	2A69N	Banister Mobile c/w C&T 69000
21917	ALD	2A6IL	SiS 5600
219V0	SARC RC2016	2A6IN	SiS 620
2A431	Cyrix MediaGX Cx5510	2A6KL	ALi 1621/1543C
2A431	Cyrix Gxi CX5520 (MediaGX)	2A6KO	ALi M16311535D
2A433	Cyrix GXm Cx5520	2A6LF	Apollo Pro (691/596)
2A434	Cyrix GXm Cx5530	2A6LG	Apollo Pro Plus (692/596)
2A496	Saturn	2A6LI	MVP4 VIA 601/686A
2A498	Saturn II	2A6LJ	VIA 694X/596B/686A

Code	Chipset	Code	Chipset
2A499	Aries	2A6LK	VIA VT8371 (KX-133)
2A4H2	Contaq 82C596-9	2A9KG	ALi M6117/1521/1523
2A4IB	Sis 496/497	2A9GH	Neptune ISA
2A4J6	Winbond W83C491 (Symphony)	2B496	Saturn I EISA
2A4KA	Possibly ALI	2B597	Mercury EISA
2A4KC	ALi 1438/45/31	2B59A	Neptune EISA
2A4KD	ALi 1487/1489	2B59F	430HX EISA
2A4L4	VIA 486A/482/505	2B69D	Orion EISA
2A4L6	VIA 496/406/505	2C460	UNIchip U4800-VLX
2A4O3	EFAR EC802GL/EC100G	2C470	HYF82481
2A4UK	OPTi 802G 822	2C4D2	HiNT SC8006 HMC82C206
2A4X5	UMC 881E/8886B	2C4I7	SiS 461
2A597	Mercury	2C4I8	SiS 471B/E
2A59A	Natome (Neptune) EISA	2C4I9	SiS 85C471B/E/G
2A59B	Mercury	2C4J6	Winbond W83C491 SLC82C491
2A59C	Triton FX (Socket 7)	2C4K9	Ali 14296
2A59F	Triton II HX (430 HX) Socket 7	2C4KC	ALi8 1439/45/31
2A59G	Triton VX (Socket 7)	2C4L2	VIA 82C486A
2A59H	Triton VX (Socket 7) Illegal	2C4L6	VIA VT496G
2A59I	Triton TX (Socket 7)	2C4L8	VIA VT425MV
2A5C7	VIA VT82C570	2C403	EFAR EC802G-B
2A5G7	VLSI VL82C594	2C4S0	AMD Elan 470
2A5GB	VLSI Lynx VL82C541/543	2C4T7	ACC Micro 2048
2A5IA	SiS 501/02/03	2C4UK	Opti 802G
2A5IC	SiS 5501/02/03	2C4X2	UMC UM82C491/493
2A5ID	SiS 5511/12/13	2C4X6	UMC UM498F/496F
2A5IE	SiS 5101-5103	2C4Y1	Samsung KS82C884
2A5IF	SiS 5596/5597	2C917	ALD
2A5IH	SiS 5571	2C9V0	SARC RC2016
2A5II	SiS 5582/5597/5598	3A6LF	Apollo Pro (691/596)
2A5IJ	SiS 5120 Mobile	6A450	STMicroelectronics PC Client ST86
2A5IK	SiS 5591	6A69L	Camino 820 Award
2A5IM	SiS 530	6A69M	810E
2A5KB	ALi 1449/61/51	6A69R	Solano 815E
2A5KE	Ali 1511	6A69S	Intel 850
2A5KF	Ali 1521/23	6A69V	Intel 845
2A5KI	Ali IV+ M1531/1543 (Super TX)	6A6IS	SiS 730
2A5KK	Aladdin V	6A6IT	SiS 635
2A5L5	VIA	6A6IU	SiS 733
2A5L7	VIA VT82C570	6A6KT	ALI MAGiK 1 (M1647/M1535D+)
2A5L9	VIA VT82C570M	6A6LJ	Apollo Pro 133A 694X/686A
2A5LA	Apollo VP1 (VT82C580VP/VxPro)	6A6LK	VIA VT8371 KX 133
2A5LC	Apollo VP2 (AMD 640)	6A6LL	VIA VT8605 chipset (Video Integrated)
2A5LD	VIA VPX (VxPRO+)	6A6LM	VIA VT8363 KT 133
2A5LE	Apollo MVP3	6A6LN	VIA VT8365 (KM-133)/VT8364 (KL-133)
2A5LH	Apollo VP4	6A6LU	VIA Apollo Pro266
2A5R5	Forex FRX58C613/601A	6A6LV	VIA VT8366/VT8233
2A5R6	Forex FRX58C613A/602B/601B	6A6LW	VIA P4X266 (VT8753+VT8233)
2A5T6	ACC Micro 2278/2188 (Auctor)	6A6S2	AMD 751

Code	Chipset	Code	Chipset
2A5UI	Opti 82C822/596/597 596/546/82	6A6S6	AMD 760
2A5UL	Opti 82C822/571/572	JA6LM	VIA VT8363 KT133 Matsonic
2A5UM	Opti 82C822/546/547		

BIOStar ID Strings

For example:

```
TVX0917B
```

TVX is the the board, 0917 the date (in this case Sep 17). B stands for BIOSTAR.

DTK ID Strings

Check the first two digits of the line starting with #. For example:

```
#34062890S
```

refers to revision number 34.

Gateway ID Strings

See ALR.

Intel ID Strings

Recent Intel desktop boards use an Intel/AMI BIOS pattern that looks like this:

```
MV85010A.86A.0011.P05
```

The characters before the first dot indicate what board you have. Here, MV85010A identifies the standard Intel® Desktop Board D850MV. 86A indicates a standard Intel version (86B, 86C, or 86E are also used on some older boards). If the first two sections do not match, then your desktop board was manufactured for a specific OEM. Older Intel desktop boards used a BIOS pattern like this:

```
1.00.12.CS1
```

The characters CS1 identify which desktop board you have, in this case a VS440FX.

Micronics ID Strings

Refer to Phoenix, as Micronics make their own Phoenix upgrades.

MR BIOS ID Strings

Code	Board
ACER300	Acer/ALI M1209
ACER301	Acer/ALI M1209
ACER304	Acer/ALI M1209
ACER305	Acer/ALI M1209
ACER306	Acer/ALI M1209
ACER307	Acer/ALI M1209
ACER308	Acer/ALI M1209-Cyrix 486SLC
ACER309	Acer/ALI M1209-Cyrix 486SLC
ACER30C	Acer/ALI M1209-Cyrix 486SLC
ACER30D	Acer/ALI M1209-Cyrix 486SLC
ACER30E	Acer/ALI M1209-Cyrix 486SLC
ACER30F	Acer/ALI M1209-Cyrix 486SLC
ACER310	Acer/ALI M1217
ACER311	Acer/ALI M1217
ACER314	Acer/ALI M1217
ACER315	Acer/ALI M1217
ACER316	Acer/ALI M1217
ACER317	Acer/ALI M1217
ACER318	Acer/ALI M1217-Cyrix 486SLC
ACER319	Acer/ALI M1217-Cyrix 486SLC
ACER31C	Acer/ALI M1217-Cyrix 486SLC
ACER31D	Acer/ALI M1217-Cyrix 486SLC
ACER31E	Acer/ALI M1217-Cyrix 486SLC
ACER31F	Acer/ALI M1217-Cyrix 486SLC
C&T_300	Chips & Technologies CS8230
C&T_304	Chips & Technologies CS8230
C&T_305	Chips & Technologies CS8230
C&T_308	Chips & Technologies CS8230
C&T_309	Chips & Technologies CS8230
CNTQ400	Contaq 82C591/82C592 WriteBack
CNTQ404	Contaq 82C591/82C592 WriteBack
CNTQ405	Contaq 82C591/82C592 WriteBack
CNTQ406	Contaq 82C591/82C592 WriteBack
CNTQ407	Contaq 82C591/82C592 WriteBack
CNTQ410	Contaq 82C596 WriteBack
CNTQ411	Contaq 82C596 WriteBack
CNTQ412	Contaq 82C596 WriteBack
EFAR400	Efar Microsystems 82EC495 WriteBack
EFAR401	Efar Microsystems 82EC495 WriteBack-82C711 Combo I/O
EFAR402	Efar Microsystems 82EC495 WriteBack-PC87310 Super I/O
EFAR404	Efar Microsystems 82EC495 WriteBack
EFAR405	Efar Microsystems 82EC495 WriteBack
EFAR406	Efar Microsystems 82EC495 WriteBack
EFAR407	Efar Microsystems 82EC495 WriteBack
EFAR408	Efar Microsystems 82EC495 WriteBack-82C711 Combo I/O
EFAR409	Efar Microsystems 82EC495 WriteBack-82C711 Combo I/O
EFAR40A	Efar Microsystems 82EC495 WriteBack-82C711 Combo I/O

Code	Board
EFAR40B	Efar Microsystems 82EC495 WriteBack-82C711 Combo I/O
EFAR40C	Efar Microsystems 82EC495 WriteBack-PC87310 Super I/O
EFAR40D	Efar Microsystems 82EC495 WriteBack-PC87310 Super I/O
EFAR40E	Efar Microsystems 82EC495 WriteBack-PC87310 Super I/O
EFAR40F	Efar Microsystems 82EC495 WriteBack-PC87310 Super I/O
EFAR410	Efar Microsystems 82EC798 WriteBack
EFAR411	Efar Microsystems 82EC798 WriteBack-82C711 Combo I/O
EFAR412	Efar Microsystems 82EC798 WriteBack-PC87310 Super I/O
EFAR414	Efar Microsystems 82EC798 WriteBack
EFAR415	Efar Microsystems 82EC798 WriteBack
EFAR416	Efar Microsystems 82EC798 WriteBack
EFAR417	Efar Microsystems 82EC798 WriteBack
EFAR418	Efar Microsystems 82EC798 WriteBack-82C711 Combo I/O
EFAR419	Efar Microsystems 82EC798 WriteBack-82C711 Combo I/O
EFAR41A	Efar Microsystems 82EC798 WriteBack-82C711 Combo I/O
EFAR41B	Efar Microsystems 82EC798 WriteBack-82C711 Combo I/O
EFAR41C	Efar Microsystems 82EC798 WriteBack-PC87310 Super I/O
EFAR41D	Efar Microsystems 82EC798 WriteBack-PC87310 Super I/O
EFAR41E	Efar Microsystems 82EC798 WriteBack-PC87310 Super I/O
EFAR41F	Efar Microsystems 82EC798 WriteBack-PC87310 Super I/O
EFAR41G	Efar Microsystems 82EC798 WriteBack-Cyrix 486DLC
EFAR41H	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-82C711 Combo I/O
EFAR41J	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-PC87310 Super I/O
EFAR41K	Efar Microsystems 82EC798 WriteBack-Cyrix 486DLC
EFAR41L	Efar Microsystems 82EC798 WriteBack-Cyrix 486DLC
EFAR41M	Efar Microsystems 82EC798 WriteBack-Cyrix 486DLC
EFAR41N	Efar Microsystems 82EC798 WriteBack-Cyrix 486DLC
EFAR41P	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-82C711 Combo I/O
EFAR41Q	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-82C711 Combo I/O
EFAR41R	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-82C711 Combo I/O
EFAR41S	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-82C711 Combo I/O
EFAR41T	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-PC87310 Super I/O
EFAR41U	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-PC87310 Super I/O
EFAR41V	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-PC87310 Super I/O
EFAR41W	Efar Microsystems 82EC798 W/B-Cyrix 486DLC-PC87310 Super I/O
EFAR41X	Efar Microsystems 82EC798 WriteBack-Cyrix 486DLC
ELIT320	Elite Microelectronics Eagle Rev. A1
ELIT324	Elite Microelectronics Eagle Rev. A1
ELIT325	Elite Microelectronics Eagle Rev. A1
ELIT420	Elite Microelectronics Eagle Rev. A1
ELIT424	Elite Microelectronics Eagle Rev. A1
ELIT425	Elite Microelectronics Eagle Rev. A1
ELIT426	Elite Microelectronics Eagle Rev. A1
ELIT427	Elite Microelectronics Eagle Rev. A1
ETEQ301	Eteq Microsystems 82C491/82C493 Bobcat Rev. A
ETEQ303	Eteq Microsystems 82C491/82C492 Cougar Rev. B, C
ETEQ304	Eteq Microsystems 82C491/82C492 Cougar Rev. B, C
ETEQ305	Eteq Microsystems 82C491/82C492 Cougar Rev. B, C
ETEQ311	Eteq Microsystems 82C491/82C493 Bobcat Rev. A

Code	Board
ETEQ314	Eteq Microsystems 82C491/82C493 Bobcat Rev. A
ETEQ315	Eteq Microsystems 82C491/82C493 Bobcat Rev. A
ETEQ321	Eteq Microsystems 82C4901/82C4902 Bengal WriteBack
ETEQ324	Eteq Microsystems 82C4901/82C4902 Bengal WriteBack
ETEQ325	Eteq Microsystems 82C4901/82C4902 Bengal WriteBack
ETEQ421	Eteq Microsystems 82C4901/82C4902 Bengal WriteBack
ETEQ428	Eteq Microsystems 82C4901/82C4902 Bengal WriteBack
ETEQ429	Eteq Microsystems 82C4901/82C4902 Bengal WriteBack
ETEQ401	Eteq Microsystems 82C491/82C493 Bobcat Rev. A
ETEQ403	Eteq Microsystems 82C491/82C492 Cougar Rev. B, C
ETEQ404	Eteq Microsystems 82C491/82C492 Cougar Rev. B, C
ETEQ405	Eteq Microsystems 82C491/82C492 Cougar Rev. B, C
HDK_200	EverTech 286 Hedaka
HDK_210	EverTech 286 Hedaka-built-in EMS
FORX300	Forex 36C300/200 [36C300/46C402] WriteThru
FORX303	Forex 36C300/200 [36C300/46C402] WriteThru
FORX320	Forex 36C311 Single Chip 386SX with Cache
FORX323	Forex 36C311 Single Chip 386SX with Cache
FORX410	Forex 46C411/402 WriteThru
FORX413	Forex 46C411/402 WriteThru
FORX418	Forex 46C411/402 WriteThru
FORX419	Forex 46C411/402 WriteThru
FORX420	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX421	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX422	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX423	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX424	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX425	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX426	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX427	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX428	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FORX429	Forex 46C521 WriteBack Forex 46C421A/422 WriteBack
FTDI400	FTDI 82C3480 WriteBack/WriteThru
FTDI401	FTDI 82C3480 WriteBack/WriteThru with 82C711 Combo I/O
FTDI402	FTDI 82C3480 WriteBack/WriteThru with PC87310 Super I/O
FTDI408	FTDI 82C3480 WriteBack/WriteThru
FTDI409	FTDI 82C3480 WriteBack/WriteThru with 82C711 Combo I/O
FTDI40A	FTDI 82C3480 WriteBack/WriteThru with PC87310 Super I/O
HKT_301	Hong Kong Technology HK3000 (Phoenix 8242 Keyboard Controller)
HKT_302	Hong Kong Technology HK3000 (MR BIOS 8042 Keyboard Controller)
HT12200	Headland Technologies HT12/HT12+
HT12201	Headland Technologies HT12/HT12+
HT12202	Headland Technologies HT12/HT12+
HT12210	Headland Technologies HT12/HT12+ with built-in EMS
HT12211	Headland Technologies HT12/HT12+ with built-in EMS
HT12211	Headland Technologies HT12/HT12+ with built-in EMS
HT22300	Headland Technologies HT22/HT18C
HT22302	Headland Technologies HT22/HT18C
HT22303	Headland Technologies HT22/HT18C

Code	Board
HT2230A	Headland Technologies HT22/HT18C with 82C711 Combo I/O
HT2230B	Headland Technologies HT22/HT18C with PC87310 Super I/O
HT2230C	Headland Technologies HT22/HT18C with 82C711 Combo I/O
HT2230D	Headland Technologies HT22/HT18C with PC87310 Super I/O
HT2230E	Headland Technologies HT22/HT18C with 82C711 Combo I/O
HT2230F	Headland Technologies HT22/HT18C with PC87310 Super I/O
HT32300	Headland Technologies HT320 Shasta
HT32302	Headland Technologies HT320 Shasta
HT32303	Headland Technologies HT320 Shasta
HT3230A	Headland Technologies HT320 Shasta with 82C711 Combo I/O
HT3230B	Headland Technologies HT320 Shasta with PC87310 Super I/O
HT3230C	Headland Technologies HT320 Shasta with 82C711 Combo I/O
HT3230D	Headland Technologies HT320 Shasta with PC87310 Super I/O
HT3230E	Headland Technologies HT320 Shasta with 82C711 Combo I/O
HT3230F	Headland Technologies HT320 Shasta with PC87310 Super I/O
HT34400	Headland Technologies HT340 Shasta
HT34408	Headland Technologies HT340 Shasta
HT34409	Headland Technologies HT340 Shasta
HT3440A	Headland Technologies HT340 Shasta with 82C711 Combo I/O
HT3440B	Headland Technologies HT340 Shasta with PC87310 Super I/O
HT3440C	Headland Technologies HT340 Shasta with 82C711 Combo I/O
HT3440D	Headland Technologies HT340 Shasta with PC87310 Super I/O
HT3440E	Headland Technologies HT340 Shasta with 82C711 Combo I/O
HT3440F	Headland Technologies HT340 Shasta with PC87310 Super I/O
MOSL400	Mosel MS400 Single Chip
MOSL403	Mosel MS400 Single Chip
MOSL404	Mosel MS400 Single Chip
MOSL410	Mosel MS400 Single Chip with 82C711 Combo I/O
MOSL413	Mosel MS400 Single Chip with 82C711 Combo I/O
MOSL415	Mosel MS400 Single Chip with 82C711 Combo I/O
MXIC300	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC302	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC303	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC304	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC305	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC308	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC30A	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC30B	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC30C	Micronix MX83C305/306 (with built-in 8Kb cache)
MXIC30D	Micronix MX83C305/306 (with built-in 8Kb cache)
OPTI306	OPTi 82C381 WriteThru
OPTI308	OPTi 82C381 WriteThru
OPTI309	OPTi 82C381 WriteThru
OPTI315	OPTi 82C281 SxPW Single-Chip Posted-Write
OPTI316	OPTi 82C281 SxPW Single-Chip Posted-Write
OPTI319	OPTi 82C281 SxPW Single-Chip Posted-Write, 82C711 Combo I/O
OPTI31A	OPTi 82C281 SxPW Single-Chip Posted-Write, PC87310 Super I/O
OPTI31K	OPTi 82C281 SxPW Single-Chip Posted-Write
OPTI31L	OPTi 82C281 SxPW Single-Chip Posted-Write

Code	Board
OPTI31M	OPTi 82C281 SxPW Single-Chip Posted-Write, 82C711 Combo I/O
OPTI31N	OPTi 82C281 SxPW Single-Chip Posted-Write, 82C711 Combo I/O
OPTI31P	OPTi 82C281 SxPW Single-Chip Posted-Write, PC87310 Super I/O
OPTI31Q	OPTi 82C281 SxPW Single-Chip Posted-Write, PC87310 Super I/O
OPTI317	OPTi 82C283 SxPI Single-Chip
OPTI318	OPTi 82C283 SxPI Single-Chip
OPTI31B	OPTi 82C283 SxPI Single-Chip with 82C711 Combo I/O
OPTI31C	OPTi 82C283 SxPI Single-Chip with PC87310 Super I/O
OPTI31D	OPTi 82C283 SxPI Single-Chip
OPTI31E	OPTi 82C283 SxPI Single-Chip
OPTI31F	OPTi 82C283 SxPI Single-Chip with 82C711 Combo I/O
OPTI31G	OPTi 82C283 SxPI Single-Chip with 82C711 Combo I/O
OPTI31H	OPTi 82C283 SxPI Single-Chip with PC87310 Super I/O
OPTI31J	OPTi 82C283 SxPI Single-Chip with PC87310 Super I/O
OPTI324	OPTi 82C391 WriteBack Rev. A & Rev. B
OPTI32B	OPTi 82C391 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI32C	OPTi 82C391 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI32E	OPTi 82C391 WriteBack Rev. A & Rev. B
OPTI32F	OPTi 82C391 WriteBack Rev. A & Rev. B
OPTI32G	OPTi 82C391 WriteBack Rev. A & Rev. B
OPTI32H	OPTi 82C391 WriteBack Rev. A & Rev. B
OPTI32J	OPTi 82C391 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI32K	OPTi 82C391 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI32L	OPTi 82C391 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI32M	OPTi 82C391 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI32P	OPTi 82C391 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI32Q	OPTi 82C391 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI32R	OPTi 82C391 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI32S	OPTi 82C391 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI330	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI331	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI332	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI334	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI335	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI336	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI337	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI338	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI339	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI33A	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI33B	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI33C	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI33D	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI33E	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI33F	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI340	OPTi 82C291 SxWB Single-Chip WriteBack
OPTI341	OPTi 82C291 SxWB Single-Chip WriteBack, 82C711 Combo I/O
OPTI342	OPTi 82C291 SxWB Single-Chip WriteBack, PC87310 Super I/O
OPTI344	OPTi 82C291 SxWB Single-Chip WriteBack
OPTI345	OPTi 82C291 SxWB Single-Chip WriteBack

Code	**Board**
OPTI346	OPTi 82C291 SxWB Single-Chip WriteBack
OPTI347	OPTi 82C291 SxWB Single-Chip WriteBack
OPTI348	OPTi 82C291 SxWB Single-Chip WriteBack, 82C711 Combo I/O
OPTI349	OPTi 82C291 SxWB Single-Chip WriteBack, 82C711 Combo I/O
OPTI34A	OPTi 82C291 SxWB Single-Chip WriteBack, 82C711 Combo I/O
OPTI34B	OPTi 82C291 SxWB Single-Chip WriteBack, 82C711 Combo I/O
OPTI34C	OPTi 82C291 SxWB Single-Chip WriteBack, PC87310 Super I/O
OPTI34D	OPTi 82C291 SxWB Single-Chip WriteBack, PC87310 Super I/O
OPTI34E	OPTi 82C291 SxWB Single-Chip WriteBack, PC87310 Super I/O
OPTI34F	OPTi 82C291 SxWB Single-Chip WriteBack, PC87310 Super I/O
OPTI406	OPTi 82C481 WriteThru
OPTI408	OPTi 82C481 WriteThru
OPTI409	OPTi 82C481 WriteThru
OPTI424	OPTi 82C491 WriteBack (original)
OPTI428	OPTi 82C491 WriteBack Rev. A & Rev. B
OPTI42B	OPTi 82C491 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI42C	OPTi 82C491 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI42E	OPTi 82C491 WriteBack Rev. A & Rev. B
OPTI42F	OPTi 82C491 WriteBack Rev. A & Rev. B
OPTI42G	OPTi 82C491 WriteBack Rev. A & Rev. B
OPTI42H	OPTi 82C491 WriteBack Rev. A & Rev. B
OPTI42J	OPTi 82C491 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI42K	OPTi 82C491 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI42L	OPTi 82C491 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI42M	OPTi 82C491 WriteBack Rev. A & Rev. B with 82C711 Combo I/O
OPTI42P	OPTi 82C491 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI42Q	OPTi 82C491 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI42R	OPTi 82C491 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI42S	OPTi 82C491 WriteBack Rev. A & Rev. B with PC87310 Super I/O
OPTI430	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI431	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI432	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI434	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI435	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI436	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI437	OPTi 82C496/497 DxPI Rev. A & Rev. B
OPTI438	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI439	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI43A	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI43B	OPTi 82C496/497 DxPI Rev. A & Rev. B with 82C711 Combo I/O
OPTI43C	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI43D	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI43E	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI43F	OPTi 82C496/497 DxPI Rev. A & Rev. B with PC87310 Super I/O
OPTI450	OPTi 82C498 DxWB WriteBack
OPTI451	OPTi 82C498 DxWB WriteBack with 82C711 Combo I/O
OPTI452	OPTi 82C498 DxWB WriteBack with PC87310 Super I/O
OPTI454	OPTi 82C498 DxWB WriteBack
OPTI455	OPTi 82C498 DxWB WriteBack

Code	Board
OPTI456	OPTi 82C498 DxWB WriteBack
OPTI457	OPTi 82C498 DxWB WriteBack
OPTI458	OPTi 82C498 DxWB WriteBack with 82C711 Combo I/O
OPTI459	OPTi 82C498 DxWB WriteBack with 82C711 Combo I/O
OPTI45A	OPTi 82C498 DxWB WriteBack with 82C711 Combo I/O
OPTI45B	OPTi 82C498 DxWB WriteBack with 82C711 Combo I/O
OPTI45C	OPTi 82C498 DxWB WriteBack with PC87310 Super I/O
OPTI45D	OPTi 82C498 DxWB WriteBack with PC87310 Super I/O
OPTI45E	OPTi 82C498 DxWB WriteBack with PC87310 Super I/O
OPTI45F	OPTi 82C498 DxWB WriteBack with PC87310 Super I/O
OPTI470	OPTi 82C495SxLC
OPTI471	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI472	OPTi 82C495SxLC with PC87310 Super I/O
OPTI474	OPTi 82C495SxLC
OPTI475	OPTi 82C495SxLC
OPTI476	OPTi 82C495SxLC
OPTI477	OPTi 82C495SxLC
OPTI478	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI479	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI47A	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI47B	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI47C	OPTi 82C495SxLC with PC87310 Super I/O
OPTI47D	OPTi 82C495SxLC with PC87310 Super I/O
OPTI47E	OPTi 82C495SxLC with PC87310 Super I/O
OPTI47F	OPTi 82C495SxLC with PC87310 Super I/O
OPTI47G	OPTi 82C495SxLC
OPTI47H	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI47J	OPTi 82C495SxLC with PC87310 Super I/O
OPTI47K	OPTi 82C495SxLC
OPTI47L	OPTi 82C495SxLC
OPTI47M	OPTi 82C495SxLC
OPTI47N	OPTi 82C495SxLC
OPTI47P	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI47Q	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI47R	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI47S	OPTi 82C495SxLC with 82C711 Combo I/O
OPTI47T	OPTi 82C495SxLC with PC87310 Super I/O
OPTI47U	OPTi 82C495SxLC with PC87310 Super I/O
OPTI47V	OPTi 82C495SxLC with PC87310 Super I/O
OPTI47W	OPTi 82C495SxLC with PC87310 Super I/O
OPTI480	OPTi 82C499 DxSC Single Chip
OPTI481	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI482	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI484	OPTi 82C499 DxSC Single Chip
OPTI485	OPTi 82C499 DxSC Single Chip
OPTI486	OPTi 82C499 DxSC Single Chip
OPTI487	OPTi 82C499 DxSC Single Chip
OPTI488	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI489	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O

Code	Board
OPTI48A	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI48B	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI48C	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48D	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48E	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48F	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48G	OPTi 82C499 DxSC Single Chip
OPTI48H	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI48J	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48K	OPTi 82C499 DxSC Single Chip
OPTI48L	OPTi 82C499 DxSC Single Chip
OPTI48M	OPTi 82C499 DxSC Single Chip
OPTI48N	OPTi 82C499 DxSC Single Chip
OPTI48P	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI48Q	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI48R	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI48S	OPTi 82C499 DxSC Single Chip with 82C711 Combo I/O
OPTI48T	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48U	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48V	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48W	OPTi 82C499 DxSC Single Chip with PC87310 Super I/O
OPTI48Z	OPTi 82C499 DxSC Single Chip with PC87311/312 Super I/O
OPTI490	OPTi 82C495 SLC
OPTI491	OPTi 82C495 SLC with 82C711 Combo I/O
OPTI492	OPTi 82C495 SLC with PC87310 Super I/O
OPTI493	OPTi 82C495 SLC
OPTI494	OPTi 82C495 SLC with 82C711 Combo I/O
OPTI495	OPTi 82C495 SLC with PC87310 Super I/O
OPTI496	OPTi 82C495 SLC
OPTI497	OPTi 82C495 SLC with 82C711 Combo I/O
OPTI498	OPTi 82C495 SLC with PC87310 Super I/O
OPTI499	OPTi 82C495 SLC
OPTI49A	OPTi 82C495 SLC with 82C711 Combo I/O
OPTI49B	OPTi 82C495 SLC with PC87310 Super I/O
OPTI4A0	OPTi 82C801 SCWB2 Single Chip WriteBack
OPTI4A1	OPTi 82C801 SCWB2 Single Chip WriteBack, 82C711 Combo I/O
OPTI4A2	OPTi 82C801 SCWB2 Single Chip WriteBack, PC87310 Super I/O
OPTI4A3	OPTi 82C801 SCWB2 Single Chip WriteBack, PC87311 Super I/O
OPTI500	OPTi 586 VHP Pentium Chipset
PKDM301	Chips & Technologies CS82310 PEAKset DM Rev-0
PKDM304	Chips & Technologies CS82310 PEAKset DM Rev-0
PKDM305	Chips & Technologies CS82310 PEAKset DM Rev-0
PKDM311	Chips & Technologies CS82310 PEAKset DM Rev-0-82C711 Combo I/O
PKDM314	Chips & Technologies CS82310 PEAKset DM Rev-0-82C711 Combo I/O
PKDM315	Chips & Technologies CS82310 PEAKset DM Rev-0-82C711 Combo I/O
PKDM321	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM322	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM323	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM324	Chips & Technologies CS82310 PEAKset DM Rev-B1

Code	Board
PKDM325	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM331	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM332	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM333	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM334	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM335	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM420	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM421	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM424	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM425	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM428	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM429	Chips & Technologies CS82310 PEAKset DM Rev-B1
PKDM430	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM431	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM434	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM435	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM438	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
PKDM439	Chips & Technologies CS82310 PEAKset DM Rev-B1-82C711 Combo I/O
SARC302	SARC RC2016A Rev. A3 (standard)
SARC306	SARC RC2016A Rev. A3 with built-in EMS
SARC30A	SARC RC2016A Rev. A3 Cyrix
SARC30E	SARC RC2016A Rev. A3 Cyrix, with built-in EMS
SCAT300	Chips & Technologies 82C236 SCATsx
SCAT304	Chips & Technologies 82C236 SCATsx
SCAT305	Chips & Technologies 82C236 SCATsx
SIS_303	SiS 85C310/320/330 Rabbit Rev. A, B & C
SIS_306	SiS 85C310/320/330 Rabbit Rev. A, B & C
SIS_307	SiS 85C310/320/330 Rabbit Rev. A, B & C
SIS_308	SiS 85C310/320/330 Rabbit Rev. A, B & C
SIS_309	SiS 85C310/320/330 Rabbit Rev. A, B & C
SIS_400	SiS 85C460 & 85C461V Single-Chip
SIS_404	SiS 85C460 & 85C461V Single-Chip
SIS_405	SiS 85C460 & 85C461V Single-Chip
SLGC301	SysLogic 386 non-cache
SLGC302	SysLogic 386 with cache
SLGC304	SysLogic 386 non-cache
SLGC305	SysLogic 386 non-cache
SLGC306	SysLogic 386 with cache
SLGC307	SysLogic 386 with cache
SLGC401	SysLogic 486 no external cache
SLGC404	SysLogic 486 no external cache
SLGC405	SysLogic 486 no external cache
STD_286	Generic 286 (TTL/Discrete Logic)
STD_202	Generic 286 (TTL/Discrete Logic)
STD_203	Generic 286 (TTL/Discrete Logic)
STD_386	Generic 386 (TTL/Discrete Logic)
STD_302	Generic 386 (TTL/Discrete Logic)
STD_303	Generic 386 (TTL/Discrete Logic)
STD_486	Generic 486 (TTL/Discrete Logic)

Code	Board
STD_408	Generic 486 (TTL/Discrete Logic)
STD_409	Generic 486 (TTL/Discrete Logic)
SYML401	Symphony Labs SL82C46x Haydn Rev. 1.1
SYML402	Symphony Labs SL82C46x Haydn Rev. 1.1 with 82C711 Combo I/O
SYML403	Symphony Labs SL82C46x Haydn Rev. 1.1 with PC87310 Super I/O
SYML404	Symphony Labs SL82C46x Haydn Rev. 1.1
SYML405	Symphony Labs SL82C46x Haydn Rev. 1.1
SYML406	Symphony Labs SL82C46x Haydn Rev. 1.1 with 82C711 Combo I/O
SYML407	Symphony Labs SL82C46x Haydn Rev. 1.1 with 82C711 Combo I/O
SYML408	Symphony Labs SL82C46x Haydn Rev. 1.1 with PC87310 Super I/O
SYML409	Symphony Labs SL82C46x Haydn Rev. 1.1 with PC87310 Super I/O
SYML411	Symphony Labs SL82C46x Haydn Rev. 1.2
SYML412	Symphony Labs SL82C46x Haydn Rev. 1.2 with 82C711 Combo I/O
SYML413	Symphony Labs SL82C46x Haydn Rev. 1.2 with PC87310 Super I/O
SYML414	Symphony Labs SL82C46x Haydn Rev. 1.2
SYML415	Symphony Labs SL82C46x Haydn Rev. 1.2
SYML416	Symphony Labs SL82C46x Haydn Rev. 1.2 with 82C711 Combo I/O
SYML417	Symphony Labs SL82C46x Haydn Rev. 1.2 with 82C711 Combo I/O
SYML418	Symphony Labs SL82C46x Haydn Rev. 1.2 with PC87310 Super I/O
SYML419	Symphony Labs SL82C46x Haydn Rev. 1.2 with PC87310 Super I/O
TACT300	Texas Instruments TACT83000 Tiger non-cache
TACT302	Texas Instruments TACT83000 Tiger with Intel 82385 cache
TACT303	Texas Instruments TACT83000 Tiger with Austek cache
TACT30A	Texas Instruments TACT83000 Tiger non-cache
TACT30B	Texas Instruments TACT83000 Tiger non-cache
TACT30C	Texas Instruments TACT83000 Tiger with Austek cache
TACT30D	Texas Instruments TACT83000 Tiger with Austek cache
TACT30E	Texas Instruments TACT83000 Tiger with Intel 82385 cache
TACT30F	Texas Instruments TACT83000 Tiger with Intel 82385 cache
TACT400	Texas Instruments TACT83000 Tiger no external cache
TACT40A	Texas Instruments TACT83000 Tiger no external cache
TACT40B	Texas Instruments TACT83000 Tiger no external cache
UMC_301	UMC 82C48x WriteBack Rev. 0
UMC_302	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_304	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_310	UMC 82C330 Twinstar
UMC_314	UMC 82C330 Twinstar
UMC_315	UMC 82C330 Twinstar
UMC_401	UMC 82C48x WriteBack Rev. 0
UMC_402	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_403	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_404	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_405	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_406	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_407	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_40A	UMC 82C48x WriteBack Rev. B
UMC_40B	UMC 82C48x WriteBack Rev. B
UMC_40C	UMC 82C48x WriteBack Rev. B
UMC_40D	UMC 82C48x WriteBack Rev. B

Code	**Board**
UMC_40E	UMC 82C48x WriteBack Rev. B
UMC_40F	UMC 82C48x WriteBack Rev. B
UMC_40G	UMC 82C48x WriteBack Rev. A & Rev. B
UMC_410	UMC 82C491 Single-Chip
VLSI301	VLSI Technology 386 Topcat-Intel 82340 non-cache
VLSI302	VLSI Technology 386 Topcat-Intel 82340 non-cache with 82C106 IPC
VLSI312	VLSI Technology 386 Topcat-Intel 82340, 82385 cache and 82C106 IPC
VLSI401	VLSI Technology 386 Topcat-Intel 82340
VLSI402	VLSI Technology 386 Topcat-Intel 82340 with 82C106 IPC
VLSI404	VLSI Technology 386 Topcat-Intel 82340 with 82C106 IPC

Packard Bell ID Strings

Normally check the FCC number on the back of the system unit. For example:

```
400-409 = PB400
410-449 = PB410 or 430
450-459 = PB450
```

Phoenix ID Strings

These start with a product family identifier (4A3NT0X in this example):

```
4A3NT0X0.86A.0047.P03.9704071222
```

It decodes to AN430TX (i.e. Anchorage). 4L3TT0X would be LT430TX (Lonetree). The number after the X is the revision. 86 is the BIOS OEM ID (Intel here), and the next letter indicates the type of motherboard:

```
A      Consumer Desktop
B      Corporate Desktop
C      Server Products
```

0047 is the BIOS build number. P is the BIOS release type:

```
P      Production (03 is production release number)
D      Development
A      Alpha
B      Beta
```

9704071222 is the BIOS build date and time (here, 7 April 1997 at 12.22).

Sony Vaio ID Strings

For PCG F1xxx, F2xxx and F3xxx (though not for the F370/F390). These have a Phoenix BIOS that is generally not compatible with Windows 2000, shown by the first three letters of the ID string. It is normally R02, but will be W2K if OK. The last two letters specify the BIOS type, and they have to be K0. In the middle are two digits like 05 or 06, so it will look something like:

```
W2K06K0
```

Tandon ID Strings

The BIOS is not identified, but the version should be at the top right-hand of the screen.

Tyan ID Strings

TYAN motherboards use AMI or Award BIOSes. The version (at bootup) is just below the left hand corner logo (ignore those that do not start with TYN):

```
TYN [motherboard model] Vx.xx MM/DD/YY
```

x.xx is the BIOS version number, and MM/DD/YY is the release date.

Zeos ID Strings

Use a modified version of Phoenix BIOS 1.01, actually writing their own version and required to display the Phoenix rev number as per their agreement.

Using The Registry

Check the BIOSDate, BIOSName, and BIOSVersion string values in HKEY_LOCAL_MACHINE\Enum\Root*PNP0C01\0000, assuming you haven't updated or changed the BIOS since you last ran Windows Setup.

WHAT'S IN MY MACHINE (USING DEBUG)?

Here's how to see what equipment your machine has with **debug**. During boot, the BIOS examines the computer's connectors and sets an equipment-list word, which lives at 410 hex or segment 0000, offset 0410 (hex). Interrupt 11 hex returns the word in the AX register. The bits of the word are listed below, although some early versions of DOS (i.e. pre 4.0) ignore it and use their own methods (see the *CMOS Memory Map* at the end of the *Memory* chapter).

Bit	Description
0	Set if floppies are present
1	Set if maths coprocessor installed
2	Set if pointing device attached (PS/2)
3-2	RAM size (only for original IBM PC, PCjr): 00 = 16K 01 = 32K 10 = 48K 11 = 64K
5-4	Initial video mode: 00 = reserved 01 = 40-column color 10 = 80-column color 11 = 80-column mono
7-6	Number of floppies (if bit 0 set): 00 = 1 drive 01 = 2 drives 10 = 3 drives 11 = 4 drives
8	Reserved
11-9	Serial ports

Bit	Description
12	Game adapter installed
13	Serial printer attached (PCjr) or internal modem installed (PC/XT only)
15-14	Parallel ports

Testing Your Ports

Type:

```
d40:00
```

The first line of the output displayed on the screen, 0040:0000, lists the ports. It might look like this:

```
F8 03 F8 02 E8 03 E8 02
```

The addresses are displayed in reverse order, so the first two (F8 03) read as 03F8, which is COM 1. The next (F8 02) read as 02F8, which is COM 2, etc.

```
BC 03 78 03 78 02 C0 9F
```

BC 03 reads 03BC, which is the parallel port address for LPT 3. 78 03 reads as 0378, address for LPT 1 and 78 02 reads as 0278, the address for LPT 2.

Identifying Your Graphics Card

Type:

```
d C000:0010
```

Type *d* at the prompt to see the next memory location for more information.

Clearing The Master Boot Record

You might want to do this if **fdisk** doesn't work, or there is a stubborn non-DOS partition, or an equally stubborn virus. This will wipe all the data from a hard drive - remember that backups are your responsibility. Type the following at the debug prompt:

```
-F9000 L 200 0 <Enter>
-A <Enter>
-MOV dx, 9000 <Enter>
-MOV ES,DX <Enter>
-XOR BX,BX <Enter>
-MOV CX,0001 <Enter>
-MOV DX,0080 <Enter>
-MOV AX,0301 <Enter>
-INT 13 <Enter>
-INT 20 <Enter>
<Enter>
-U 100 L 12 <Enter>
-G (This executes the routine) <Enter>
-Q (This quits debug) <Enter>
```

Try this as well:

```
-FCS:200 400 0 <Enter>
-RAX <Enter>
-0301 <Enter>
-RBX <Enter>
-0600 <Enter>
-RCX <Enter>
-0001 <Enter>
-RDX <Enter>
-0080 <Enter>
-E 100 CD 13 <Enter>
-P <Enter>
-Q <Enter>
```

TO CLEAR THE CMOS

Enter debug and type:

```
-A <Enter>
-MOV AX,0 <Enter>
-MOV AX,CX <Enter>
-OUT 70,AL <Enter>
-MOV AX,0 <Enter>
-OUT 71,AL <Enter>
-INC CX <Enter>
-CMP CX,100 <Enter>
-JB 103 <Enter>
-INT 20 <Enter>
-<Enter>
-G <Enter>
-Q <Enter>
```

Try this, too:

```
-070 16
-071 16
-Q
```

And this:

```
-o71 2E <enter>
-o71 0 <enter>
-Q <enter>
```

Reboot afterwards.

WHERE CAN I GET A NEW BIOS?

In the early days, it was enough to be "IBM compatible" and you could literally swap BIOS ROMs between motherboards. It's not the case these days, as they are matched to a particular chipset by the motherboard manufacturer and are therefore specific to each other, even though they might work up to about 80% at DOS level. Before spending too much time on this, be aware that it's often easier (and cheaper) just to buy a new motherboard!

For a Flash BIOS (below), aside from the motherboard manufacturer, you may get one from:

```
MR     www.esupport.com/www.biosworld.com
Award  www.esupport.com/www.biosworld.com
AST    www.centercomp.com/ast
AMI    www.megatrends.com
```

For Olivetti (and maybe others relatively less available), try *PC Care* in UK on 44 1992 462882. AMI BIOS and BIOS Plus series (with 16-character ID code) for cached motherboards are customised, and only obtainable from the OEM, except::

- Those with E307 as first 4 characters (aaaa), often replaceable with a standard type.
- Northgate or Motherboard Factory motherboards (except the Northgate slimline), which can take a standard type.
- Those with aaaa = DAMI, DAMX or EDAMI usually for AMI cached boards.

Gateway use Intel motherboards and modify the AMI BIOS, so don't expect an upgrade from Intel to work. A Gateway BIOS has a T suffix. Here are others:

```
H      Vobis
K      NEC
L      Hewlett Packard
Q      AST
R      Packard Bell
```

Otherwise, try **biosworld.com**, **eSupport.com** or **flashbios.org**. You need the proper information when you call; if you already have an AMI BIOS, for example, you will need the reference or part number in the ID string. Be sure that a new BIOS will actually solve a problem that you have - in other words, if your machine ain't broke, don't fix it! New code may add more hassle.

FLASH BIOS UPGRADES

Your motherboard manual should state whether it has a Flash BIOS (most modern ones do), but if you don't have one, or just want to make sure, look under the sticker for these codes on the chip (*xxx* just denotes the capacity):

Code	Type
28Fxxx	Fujitsu or ISSI 12v

Code	Type
29Cxxx	5v
29LVxxx	3v (not often seen)
28Cxxx	EEPROM (like Flash, but needs a special device)
27Cxxx	EPROM, so you need UV to erase and a programmer to rewrite.
PH29EE010	SST flashable ROM chip
29EE011	5v flashable Winbond chip
29C010	5v flashable Amtel chip
Am29F010	AMD 5v
Am28F010 Am28F010A	AMD 12v
AT28C010 AT28MC010 AT29C010	
AT29LC010, AT29MC010 AT49F002T	Atmel 5v
CAT28F010V5 CAT28F010V5I	Catalyst 5v
CAT28F010, CAT28F010I	Catalyst 12v
HN58C1000	Hitachi 5v
HN28F101 HN29C010 HN29C010B HN58C1001 HN58V1001	Hitachi 12v
28F010	Intel 12v
M5M28F101FP M5M28F101P M5M28F101RV M5M28F101VP	Mitsubishi 12v
MX28F1000	MXIC 12v
MSM28F101	OKI 12v
KM29C010	Samsung 5v
DQ28C010 DYM28C010 DQM28C010A	SEEQ 5v
DQ47F010 DQ48F010	SEEQ 12v
M28F010 M28F1001	SGS-Thomson 12v
28EE011 29EE010	SST 5v
PH29EE010	SST Flashable ROM
TMS29F010	Texas Instruments 5v
TMS28F010	Texas Instruments 12v

Code	Type
W29EE011 W29C020C W49F002U	Winbond 5v
W27F010	Winbond 12v
X28C010 X28C010I XM28C010 XM28C010I	XICOR 5v

Anything without a window and a 28, 29, or 49 in the first bit of the part number is most likely to be a standard ROM and not flashable.

All the software you need will fit onto a boot floppy, which should naturally be checked for viruses, and maybe formatted unconditionally to ensure it is properly serviceable (don't use one with errors on it!). Aside from DOS, you will need the upgrade utility and the data file for your motherboard. Both will be obtainable from the web site or BBS of either your motherboard or BIOS manufacturer (try the former first). It will usually be a file with a **.bin** extension, or maybe **.rom**. The disk should have the DOS boot files only, with no memory drivers, but you might want to include an **autoexec.bat** file to automate the process, in case you have to do the job blind, or you want to minimise mistakes (Award BIOSes look for it automatically). You can elaborate on this for the contents (assuming an Award BIOS):

```
@echo off
if exist oldbios.bin goto old
awdflash.exe newbios.bin oldbios.bin /py /sy /cc /cp /cd /r
goto end
:old
awdflash.exe oldbios.bin /py /sn /cc /cp /cd /sb /r
:end
```

newbios.bin is the downloaded image for your motherboard. **oldbios.bin** is the current version of your BIOS that you save when prompted (please do it!). If the process fails, the old one can be reflashed automatically when you reboot with the floppy in the drive (take it out when successful). **/py** answers *Yes* automatically to the question about really wanting to flash the BIOS (default). **/sy** answers *Yes* automatically to the one about saving the previous version (default). **/cc** clears the CMOS after reflashing. **/cp** clears the PnP (ESCD) data, and **/cd** clears the DMI (see below). **/r** reboots.

If something goes wrong, some Award BIOS chips have a small amount code hardwired into them that allows at least a boot from a floppy, although only ISA video cards are supported (because they are initialized early). An AMI BIOS will look for a **.rom** file on the floppy when the **Ctrl** and **Home** keys are held down.

Intel boards have the same arrangement, and the code is activated by moving a Flash Recovery jumper to activate some non-erasable code in the boot block area. Put the jumper in the recovery position, start with a bootable diskette, listen to the speaker and watch the floppy light (there's no video, as the code is small).

When you hear a beep and the light comes on, the recovery code is being reloaded. When the light goes out, switch the machine off, put the jumper back to its normal position and continue. Note that motherboards manufactured by Intel typically use encryption to make sure you flash the correct bios, and it is difficult for upgraders to cope with this.

Asus motherboards have a POST Flash utility, which is obtained by pressing alt + F2 after the memory test - this means that you don't need a floppy, except for the new code file.

The Flash ROM requires a relatively high voltage to burn it, which is usually set with a jumper on the motherboard (it may be marked 12v or 5v). If you don't have one, it will probably be done by the Flash software anyway. The chips concerned can only be flashed for a limited number of times, and not a high one at that.

Take note of the current settings, so you can reinstall them after you have upgraded (maybe use **Prt Sc** for a hard copy) - turn off the *System BIOS Cacheable* option, as well as Shadowing or Power Management. If updating a portable, run it from the AC mains. You may also need to set a jumper or switch on the motherboard to allow the ROM to be written to. Making sure you actually have the right software, and that your machine is not overclocked, boot from the upgrade floppy, and run the utility. The command line will include the name of the utility and the file for the upgrade, typically:

```
bflash p5_aw.14g
```

But see the example **autoexec.bat** file, above, for command line switches. In the above example, **bflash** is the name of the utility (**bflash.exe**) and **p5_aw.14g** is the file with the code for the BIOS; in this case, it's for the P5 motherboard, which has an Award BIOS (aw), revision 14g. *Always save the current BIOS*, when asked, so you can recover later, because the new code may cause different problems. Also, DO NOT TURN THE MACHINE OFF DURING THE UPGRADE, even if there is a recovery procedure-just repeat the process. If the problem persists, reload the BIOS you saved earlier. It's not a good idea to use another manufacturer's software, but, if you have an emergency, it would appear that Award's works with all except Asus boards, and MR's **29C010.exe** is good, too.

Once everything has finished, clear the CMOS, in case the old settings are not compatible with the new code, then check for a successful upgrade with the BIOS identifier on the screen, turn the machine off, reset the jumper, reboot and enter all the previous settings (though you may have to accept the defaults). Reboot again.

Tip: If you get problems after upgrading an AMI BIOS, press F5 in Setup to clear the CMOS.

There's lots of lots of good stuff about Flash BIOSes at **wimsbios.com**.

RECOVERING A CORRUPT BIOS

Note: Do this with care (we accept no responsibility, etc.....)

Note: When doing this, the **/F** parameter (of **awdflash**, at least) is required. This forces the flash to go through. Also, it must be used by itself.

Generally, all you need is a BIOS chip from a similar motherboard - although they are specifically made, very often you can use one where the chipset doesn't vary too much, say, between an FX or HX motherboard (this is a slim chance, though - it's best if the motherboards are identical). The I/O chip should be the same as well, just make sure the floppy works properly, but all you need to do is be able to boot to DOS so you can change the chip when the machine is running (don't use a metal extractor tool). So, remove the corrupt chip, fit the good one so it makes contact, but is loose enough to remove easily, boot the machine with DOS and swap the chips again. By this time, the BIOS will have been shadowed, and running from RAM, so the machine will still work. Reflash the chip.

DMI

DMI (*Desktop Management Interface*) is a system which works with a Flash BIOS to keep a *Management Information Format* database up to date so you can find out what's inside a PC without opening it up, including device settings, so it's for managing system components, hardware or software. Version 2.0 will allow remote network access, although this capability is unofficially available from some vendors with 1.1. DMI can autodetect and record information concerning the computer, such as the CPU type and speed, and memory size-the BIOS stores the information in a 4K block in the Flash ROM, from where DMI can retrieve it. Plug and Play technology allows this to be updated by the operating system, which is better than having you update the whole BIOS every time. Indeed, NT occasionally flashes up a message that it's "updating DMI" as it boots.

Motherboards that can use DMI have a configuration utility that allows you to put other information in, like serial numbers, company addresses, etc.

FACILITIES PROVIDED

The BIOS ROM will include a bootstrap loader, Power On Self Test (POST), hardware initialisation, software interrupts and CMOS Setup routines, possibly with diagnostic or utility software and more.

The Bootstrap Loader

Looks for an operating system, and hands over control to it, if found, on a floppy or a hard drive (Late Phoenix BIOSes will boot from a CD-ROM, and AMI from a Zip drive; Award BIOSes can boot from a CD-ROM, SCSI drive, Zip drive and LS-120 diskette). If an error is encountered before the display is initialized, a series of Nasty Noises will tell you what's wrong (see later). Otherwise, you will see an error message (again, later in the book). A hard reset goes through the whole POST procedure. A soft reset (**ctrl-alt-del**) just runs a subset of POST and initialisation, after calling INT 19 from the BIOS.

The Power On Self Test

The POST verifies that:

- The motherboard is working, and
- The equipment in the machine is in the same condition (i.e. working) as when it was switched off. The testing is an exercising of the components; that is, it checks they are working, but not how well they are working.

Special diagnostic codes are issued during this procedure - refer to *POST Codes*.

CMOS settings

In AT-class computers, hardware setup information is kept in the CMOS RAM so the POST can refer to it. CMOS stands for *Complementary Metal Oxide Semiconductor*, which actually refers to a way of making chips with low power requirements, but has also come to mean the memory area which retains the information, because the clock chip that stored it was made that way (back in 286 days, this may have been the only such chip on a motherboard, so it became known as *the* CMOS chip).

Anyway, the purpose of the CMOS is to remember what equipment the computer has. Some computers have this separately on a disk, e.g. with early NEAT chipsets, Award v2.x or Samsungs, but now it's commonly included in the System BIOS.

Every machine has Standard CMOS settings, but some will have *Advanced CMOS* or *Chipset Features* (the whole point of this), discussed later. You may get a similar system for an embedded VGA BIOS.

Utilities

Many utilities come with the BIOS, particularly diagnostic and low-level format routines for the hard disk. The setup menu may have this heading:

```
HARD DISK UTILITY
```

It allows you to low-level format the drive attached to your computer.

```
DO NOT USE IT TO LOW LEVEL FORMAT AN IDE DISK!
```

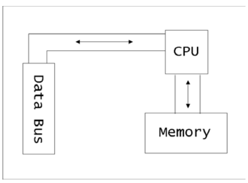
Not that it will, anyway. Sorry for shouting, by the way, but that's quite important, because it will erase the head positioning tracks. You need manufacturer's software to do it properly.

THE MOTHERBOARD

2

Today's PC is the way it is as a result of history, with new functionality built on top of the original design, which was only intended for basic work. To understand how today's machines work, it is necessary to understand how they came about, based on the concept of *backwards compatibility*. In other words, there is a lot of material in this chapter which relates to older equipment whose features have been incorporated in newer designs as time progressed, which must be understood to appreciate how they work.

The motherboard is a large printed circuit board (PCB) to which are fixed the Central Processor, the data bus, memory and various support chips, such as those that control speed and timing (chipset), the keyboard, etc. Printed circuits use copper traces instead of vast amounts of wires, and when there is no space left, another layer is added with more on, so the board gets thicker. In the early days, two-layer boards were common, but six or eight-layer ones are what you get from reputable manufacturers now, because the layers also keep the circuit traces separate; you would have one with them going one way and another at 90 degrees, both reducing crosstalk and making the board sturdier - a good example is the FIC SD-11 for the Athlon, which has six layers. As a result, some of the circuitry resembles twisted pair or coax cabling, which are both well-known methods of reducing radiation and interference in networks, so you might find a signal cable sandwiched between two ground layers, or a multi-layer board could have signals on the outer layers and ground planes in the inner ones, so there is a relatively wide separation.



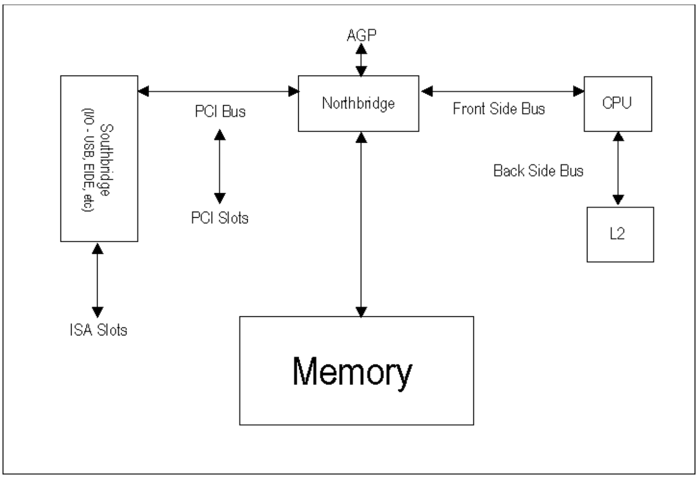
On the left is a simplified picture of a typical PC-compatible motherboard. The Central Processor does all the thinking, and is told what to do by instructions contained in memory, so there is a direct two-way bus connection between them-the bus width (multiplied by its frequency and number of times data is transferred per clock cycle) determines how much data can be read or written in one go.

Extra circuitry in the form of expansion cards is placed into expansion slots on the data bus, so the computer's basic setup can be changed easily (for example, you can connect more disk drives or a screen here). To save you typing in the same old instructions every time, you buy software prepared earlier and copy it over the data bus into memory via the processor (there is a short cut between the data bus and memory, called DMA, which is explained in the *Data Buses* chapter). One problem is that the parts of the computer that do the most work, such as the video card and hard disk, live here, which is at once the slowest and busiest part of the machine - the ISA bus runs at only 8 MHz (think of it as miles per hour), and the PCI bus at 33 MHz. Of course, the PCI bus is also 4 times wider, but the point is that it is still very much slower than the rest of the machine.

Sometimes a math co-processor is fitted to work alongside the main processor, which is specially built to cope with floating point arithmetic (e.g. decimal points). Later CPUs (i.e. some versions of the 486 and upwards) have it integrated, so it's more correctly called a *floating point unit*, or FPU. The main processor has to convert decimals and fractions to whole numbers before calculating on them, and then convert them back again, and the size of the number it can cope with depends on the register width.

A coprocessor won't be used automatically - your software must be aware of it, otherwise you won't get any benefit. If you're only doing addition, multiplication, subtraction and division, you won't find much difference in performance. Oddly enough, a copro in a 286 is *slower* than one in an XT, due to the connections.

Over time, more functionality has been added, and, after the Pentium, our simplified picture now looks something like:



Essentially, the functionality of the chipset is combined on two main components, the North- and South- Bridge chips, which live at each end of the PCI bus. The CPU, Memory and AGP talk to the Northbridge and the Southbridge handles all the I/O, including the ISA bus. NVidia calls them the IGP and MCP, respectively, standing for *Integrated Graphics Processor* and *Media and Communication Processor* (their chipset also uses AMD's Hypertransport technology, which is mentioned below).

The link between the CPU and Northbridge is called *Front Side Bus* (FSB), which is (usually) the same speed as the Memory Bus, and can be varied, as when overclocking. The Backside bus connects the CPU with L2 cache.

But even this is no better, because the essential problems that plagued the original PC still haven't gone away, in that some parts of the machine simply run too slowly. The PCI bus, at 33 MHz, is 10 times slower than even a 333 MHz CPU. If that weren't bad enough, all the I/O, including USB and Firewire, goes through the Southbridge and hence the PCI bus, losing all that speed advantage when subject to arbitration, etc. The latest architecture increases efficiency by handling multiple data streams better. To improve communications, Intel and AMD have removed the PCI bus connection between the North and South Bridges.

Intel's solution is found in the 800 series chipsets, and works like a network system, which is why individual components are called *hubs*. For example, the North Bridge is replaced by the *Memory Controller Hub* (MCH), in the middle, and the South Bridge by the *I/O Controller Hub* (ICH), which now has the PCI bus as a subcomponent. The BIOS becomes a *Firmware Hub* (FWH), which handles the ISA bus, if present. The MCH, unfortunately, is only designed to work with the very expensive Rambus. All are tied together with a 266 Mbps interface.

AMD uses a high speed, high performance, unidirectional point-to-point link between components, called HyperTransport, which can be up to 32 bits wide. It can be likened to something like Firewire just for the motherboard, but faster. In fact, it is about 50 times faster than PCI, 6 times faster than PCI-X and 5 times faster than InfiniBand 4 channels. Asymmetric buses can be used where different up- and downstream requirements exist.

Bits and Bytes

Computers talk in binary language, which means that they count to a base of 2 (we use 10). When electrical signals are sent around the computer, they are either On or Off, which matches this perfectly. A state of On or Off is called a *Binary Digit*, or *Bit* for short, and is represented on paper by a 1 for On or 0 for Off (the same as on power switches for electrical appliances). To place one character on the screen takes eight bits (a byte), so when a machine is spoken of as being eight- or sixteen-bit, it's effectively dealing with one or two letters of the alphabet at the same time - a 32-bit computer can therefore cope with 4 characters in one go. 2 bytes are called a word, 4 bytes (32-bits) are a double word and 16 bytes are a paragraph.

Because it uses multiples of 8, a computer will also count to a base of 16, or hexadecimal, which uses letters as well as numbers, and the order is 0 1 2 3 4 5 6 7 8 9 A B C D E F (numbers run out after 9).

THE CENTRAL PROCESSOR

Using multiple CPUs could only be done with Intel chips at first, except for the Celeron and some older Pentiums, but Abit has made a dual Socket 370 (Celeron PPGA) board that bypasses the SMP limitations inside the chip. However, the AMD Athlon 4 also supports it (with the AMD 760 chipset), though, oddly, some of the PIII range may not. The Pentium Pro natively supports four CPUs, whilst the Pentium and Pentium II support only two (although there is a 9,000 processor Pentium Pro machine around!). *Glueless Multiprocessing* means without special bridges and chipsets. The Athlon MP and Xeon are designed for multiprocessing with the XP and Pentium 4, respectively.

The MP/MPX and the 860/E7500 chipsets (for AMD/Intel) are used for dual processor boards. Two processors require more power, so your power supply must be able to cope as well; at least 400 W is suggested. Dual boards also use ECC memory. Xeon-based machines are better performers, not only because of the chip, but the 860 chipset which uses RAMBUS memory that clocks at (much) higher speeds than DDR. Naturally, such machines are more expensive, especially as motherboard sizes can be non-standard and may need a special case.

You may also only do *Symmetrical Multiprocessing* (SMP), as it's called, with certain operating systems, notably Windows 2000 or NT, OS/2, Sun, SCO, HP, FreeBSD and Linux. Rather than sheer performance, however, there is more likely to be an improvement in multitasking, as a single threaded application can be run on one CPU and the OS can use the other, so the machine will run more smoothly - particularly useful for stuff like video editing. A multi-threaded application, on the other hand, can use both CPUs. Note, however, that you don't get a 100% increase as there is an overhead from the various CPUs talking to each other. You also need larger caches to prevent CPUs going after data in memory, which means you need cache coherency, that is, that the data in each CPU's cache is the same.

Where two Athlons talk to memory at 266MHz, the chipset behaves like a switch between the CPUs and memory - in networking terms, a switch provides a dedicated connection between two devices that wish to talk to each other, thus allowing them both use of the maximum bandwidth. In this case, AMD's Hypertransport system allows independent access to the whole of memory by each CPU and also ensures cache coherency with reduced latency, done by tagging the state of the data in the cache of one Athlon, then allowing access by the other - the data is not duplicated.

Intel's system, by contrast, allows two CPUs to share one channel - two PIIIs sharing a bus with their 840 chipset will have 1.5 Gb/s. AMD can claim 4.2! In fact, their SmartMP technology can use a second processor to double a system's performance.

Intel has a Hyperthreading system, which allows one processor to look like two to the operating system and some applications, so two threads can be executed at once (each thread would slow down, but total throughput increases). *Thread Level Parallelism* allows two threads to share one set of instructions, to improve utilization. It's similar to multiprocessing, but not as powerful, and applications must be multithreaded. Hyperthreading can be done because most operations use less than 35% of the Netburst processor’s capabilities, due to cache misses, which leave the execution unit is waiting while data is copied into the cache.

You may have to disable *Delayed Transactions* when using a single-CPU OS.

Slots and Sockets

With improvement, CPUs have used different sockets, up to Socket 8 for the Pentium Pro, the more recent Socket 370 for the Celeron and Sockets 478 and 423 for the Pentium IV.

Socket 7 has been kept alive by Intel's competitors and improved to *Super Socket 7*, which actually refers to the facilities on the motherboard, and not the capabilities of the CPU it houses. These include AGP, large L2 cache (up to 1 Mb) and customisation for just about everything from speeds to voltages. The Pentium II uses Slot 1 technology, with Slot 2 around somewhere (Intel refers to them as 242- and 330-contact slot connectors, respectively). The reason for the change, according to them, is the increased bandwidth, but just to confuse things further, the Celeron now uses a Socket (370), having previously been able to use Slot 1.

Slot A, for the Athlon, looks like Slot 1 (don't confuse them), but the pinouts are different, partly for copyright reasons and partly because AMD felt it was time to blaze new trails (but maybe Intel not licensing the design was an influence, too). The result is a bus design that is technically superior, but which can use current cooling technology.

The 8088

This was the brains of the original IBM PC (history has a great bearing on what we get up to today, as we will discover), and was manufactured by Intel. No more need be said about it, except that although it was classified as being 16-bit, it spoke to the data bus and memory with 8 bits, which was both to keep the costs down and keep in line with the capabilities of the support chips. The 8086 was 16-bit internally and externally, so was about 20% faster, but was more expensive. The 80186 and 80188 also had about 15 or 20 system components included in the same chip, and became useful for dedicated expansion cards, as well as paving the way for the 80286 (see below). NEC made a clone, called the V22.

Anyway, when the 8088 wanted to send two characters to the screen over the data bus, they had to be sent one at a time, rather than both together, so there was an idle state where nothing was done every time data was sent (even at 4.77 MHz!). In addition, it could only talk to 1 Mb of memory; the width of the address bus determines the amount of memory locations that can be addressed at any time (the address range) and there were 20 physical connections between memory and the Central Processor. Since computers work on the binary system, and therefore count with only two fingers, it's a simple calculation as to how much memory the CPU can talk to at once:

```
220=1048,576K
```

In fact, 8-bits, as used in the original PC can only represent 28, or 256 possible values, and the 16-bit word in the CPU could address 65,536 (or 64K), which still wasn't enough for serious work, so a segment:offset scheme of memory addressing was devised, where two numbers are used for an address to get a bigger total (see *Base Memory*, below, for more about this). The problem was to maintain compatibility with the 16-bit registers in the CPU while using 20 address lines. For the moment, just bear in mind that, although the CPU can see 1 Mb in total, it can only see it 64K at a time, because the offset is limited to 16 bits, and the largest number you can create with them is 65, 535.

The 80286

The 80286 was introduced as a response to clones of the IBM PC. The connections around the motherboard became 16-bit throughout, increasing efficiency-at the same clock speed, the throughput is 4 times more. It also had 24 memory address lines, so it could talk to 16 Mb of physical memory (1 Gb virtual). Having said that, DOS couldn't use it, since the extra had to be addressed in protected mode, using something like Xenix (or OS/2, which was created a little afterwards). DOS can only run in real mode, which is restricted to the 1 Mb that can be seen by the 8088.

Just to emphasise the point-when a 286 (or above) emulates an 8086 to run DOS, it's running in real mode - a Pentium running DOS is just a fast PC!

Protected mode is there to protect programs or running processes from interfering with each other, hence the name. The idea is that programs don't write to the wrong place in memory because a protected mode memory address is not the same as one used in real mode; that is, there is no guarantee to a program that an address used is the same as its real equivalent. A memory segment in real mode, or the first part of a segment:offset address becomes a *selector*, which refers to a *descriptor table*, which is like a table of contents of what's in memory, so you get a **selector:offset** system. The descriptor table's job is to relate sectors to real addresses in memory, so there is one more step to the process of memory addressing in protected mode as there is in real mode. A 286 descriptor can store addresses as large as 16,777,216 bytes (16 Mb). Because the selector pointer is a smaller number than the full segment address, more selectors can fit into the same number of registers, which may go part of the way to explaining how you can see an extra bit of memory above 1 Mb in real mode, to get the High Memory area (see *Memory*).

As an aside, the first three bytes of a selector are used by Windows to check that the selector concerned relates to memory actually owned by the program you are using, and that memory can be written to, otherwise the program is shut down.

One problem was that the 286 went into protected mode easily, but found it difficult to get out again, and needed the chip level equivalent of **ctrl-alt-del** to do so. This used to be done with special codes that were interpreted by the keyboard controller (through an unused pin), but chips were later inserted to watch for these codes and reset the CPU immediately, rather than wait. This "fast decode" of the reset command allowed faster switching between real and protected mode (for 16-bit software), with resultant better performance, although the 286 is still ungainly at running Windows.

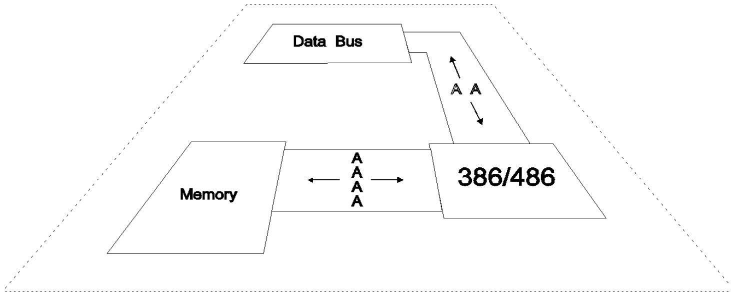
The 286 also began to be cloned, but legally, as Intel had to farm out manufacturing to keep up with the demand.

The 80386

Compaq was first to use the 80386 (the DX, as opposed to the SX-see below), which uses 32 bits between itself and memory, but 16 towards the data bus, which hasn't, until recently, been developed in tandem with the rest of the machine. This is partly to ensure backwards compatibility and partly due to the plumbing arrangements - because of its design (based on the technological knowledge of the time), if the data bus is run too fast, you get electrical noise, or extra voltages (extra 1s), which will look to the computer like extra data.

You also now have a speed problem.....

Compaq also separated the data bus from the processor bus, which means that, although the Central Processor may run at 33 Megahertz or so (think of it as miles per hour), the (ISA) data bus still ran at 8, because of the original design constraints mentioned above. It is at once the busiest and slowest part of the computer, and not only do you therefore have the equivalent of four-lane highways narrowing down to two-lane ones, these days you have to slow right down from anything up to 600 mph (MHz) in the CPU area, through 100 around memory, right down to 8 mph by the time you reach the ISA bus. Even with a PCI bus running at 33 MHz, a 450 MHz CPU is running 14 times faster, so your drive has to be really fast to catch up, as does the design of the chipset - these started become more important when the 386 came into service, as they were able to help bridge the speed gaps by managing data flow more efficiently.



In view of the above, you can see that processor speed alone is no guide to performance, and in some cases may even be irrelevant. A slow hard disk (on the data bus), for instance, will always make any processor wait for its data and waste cycles that could be used for serious work. In fact, as far as NetWare is concerned, a 486/33 is only noticeably better than a 386 when network loading is heavy (the same goes for later chips, I'm sure).

The 386 can run multiple copies of real mode, that is, it can create several 8088s inside itself, called *Virtual Machines*, so that real mode programs, provided they are well behaved, can have some benefits of protected mode. It uses paging to remap memory so these machines are brought to the attention of the CPU when the programs in them need to be run; this is done on a timeslice basis, around 60 times a second, which is how we get multitasking in Windows, or Multiuser DOS (in '95, the slice is every 20 ns). It doesn't sound like much, but programmers tell me you can do a lot inside 60 ns.

Because of paging, these DOS sessions can be anywhere in memory, but, when used, they are made to look as if they are below 1 Mb, in real mode.

Virtual DOS machines can be created in extended memory because real mode programs under DOS (and/or Windows) don't write to real addresses, but selectors, and therefore have their calls redirected to the descriptor table. By changing the relationships in the descriptor table, programs can be moved around memory without them knowing anything about it; all a program needs to do is know how to work with selectors.

The 386 can also switch in and out of protected mode on the fly, or at least in a more elegant way than the 286. To get to the hard disk and other parts of the computer, protected-mode software, such as Windows, has to get DOS to perform real mode services, so the CPU has to switch in and out of protected mode continually (actually, on a 386 or above, the switching is to *virtual 8086 mode* rather than real mode). The goal is, therefore, to use real mode as little as possible and run in protected mode. Windows does this with 32-bit instructions.

The 386 uses pipelining to help streamline memory accesses - they are done independently of each other (at the same time) while other units get on with their jobs, reducing intermediate steps and latency. Prefetching is where data is stored in CPU registers while spare cycles are used to fetch the next. The 386 has a prefetch unit for instructions, that tries to guess which ones you want next (a cache, sort of).

The 386 uses an externally generated clock frequency, and only the rising (positive) edge of it to calculate the output signal and the processor frequency, so the clock must run at twice the speed of the CPU. The bus interface operates with a two clock pulse cycle.

Although the 386 is 32-bit and has certain benefits, like the ability to manipulate memory and switch in and out of protected mode more readily, replacing a 286 with a 386 doesn't automatically give you performance benefits if you're running 16-bit 80286 code (i.e. most programs in DOS and sometimes Windows, which sits on top of it). At the same clock speed, the 286 requires fewer clock cycles to execute many instructions, as well as executing some in the same number as the 386 (74 are faster, 66 the same speed, leaving 50 that actually run better in a 386). This is because the 386 has to emulate a 286 and needs more cycles to do it.

THE 80386SX

The 80386SX is 32-bit internally, but 16-bit externally to both memory and the data bus, so you get bottlenecking, although it wasn't designed with that in mind. It is a cut-down version of the 80386DX, created both to cut costs and give the impression that the 286 is out of date, because at the time other manufacturers could make the 286 under licence. Although it can run 386-specific software, it looks like a 286 to the machine it is in, so existing motherboards could be used, with a little redesigning, as the chips are not pin-compatible. At the same clock speed, a 386SX machine is around 25% slower than the 386DX.

THE 80386SL

A low power chip, laptops, with a cache controller designed for 16-64K, SMI (*System Management Interrupt*) with power management and expanded memory support. It also came with the 82360SL I/O subsystem, the first combination of many functions into one chipset.

The 80486

To non-technical people, the 80486 is a fast 80386 (DX) with an on-board maths coprocessor and 8K of cache memory. It's not really newer technology as such (although it is second-generation), but better use is made of its facilities. For example, it takes fewer instruction cycles to do the same job (2 rather than 4.5 on the 386), and is optimised to keep as many operations inside the chip as possible. The 386 prefetch unit was replaced by 8K of SRAM cache, and pipelining was replaced by burst mode, which works on the theory that most of the time getting data concerns getting its address; you don't need it again once you're there.

Burst allows a device to send lots of data in a short time without interruption. Pipelining on the 386 requires 2 clocks per transfer; only one is needed with 486 Burst Mode. Memory parity checks also take their own path at the same time as the data they relate to. The 486 has an on-board clock, and both edges of the square wave signal are used to calculate the clock signal, so the motherboard runs at the same speed as the CPU. In addition, the bus system uses a single pulse cycle.

The cache in the CPU (known as Level 1, or L1) is the fastest in the machine, as it runs at the same speed, and has no delays. It updates main memory only when the CPU hands over control to another device (e.g. a bus master), and so needs to know what changes there are. Generally speaking, at the same clock speed, a 486 delivers between 2-3 times the performance of a 386.

THE 80486SX

The 486SX is as above, but with the maths co-processor facility disabled, therefore (generally speaking) you should find no significant difference between it and a 386; a 386/40 is broadly equivalent to a 486/25.

THE 80486SL ENHANCED

Again, for notebooks, like the 386SL, but also having a Suspend/Resume feature.

Clock Doubling

The DX/2 chip runs at double the speed of the original, but only inside itself; for example, the bus will still be running at "normal" speed. Unfortunately, high speed motherboards are more expensive for technical reasons. Actual performance depends on how many accesses are satisfied from the chip's cache, which is how (in case you were wondering) the CPU is kept busy, rather than waiting for the rest of the machine. If the CPU has to go outside the cache, effective speed is the same as the motherboard or, more properly, the relevant bus (memory or data), so best performance is obtained when all the CPU's needs are satisfied from inside itself. The DX/4 has a larger cache (16K) to cope with the higher speed.

Sadly, a cached DX/2 system wastes twice as many useable cycles as a normal one does! An Overdrive Chip and a DX/2 are more or less the same thing, but the former can be fitted by the end-user (i.e. you), and the latter is intended for manufacturers. The DX/4 is actually clock tripled (the 4 is to do with the 486 number; not the speed), but can be clock doubled with appropriate switching on the motherboard, so you could use a 50 MHz board and get better performance from the various buses.

Overclocking

This is the practice of making certain parts of your machine run faster than their rated speed, particularly CPUs, but it really started way back with the first AT, when people used faster clock crystals. It is based on the premise that the items concerned come from the same production run and only get segregated on testing - in other words, some CPUs will be made to run at 200 MHz, but others will fail and be reclassified for 166 MHz (although I don't leave out the hand of the marketing department somewhere), and advantage is taken of manufacturer's tolerance ranges. The main problems are overheating (don't forget the voltage regulators) and bus timing signals, especially AGP and PCI, and, to be sure, the failures may be in subtle areas which your software will never touch, but, to my way of thinking, Intel and the other companies have far more money and facilities for testing than I have, and my Aviation background makes me uncomfortable test flying strange equipment, so the recommendation is to be very careful. In any case, non-Intel processors tend to be overclocked already, and SCSI buses are self clocking anyway.

Having said all that, if data safety is not a problem, it's true that Pentium II/Celeron CPUs are better at it than others - some people report an increase of up to 25% in speed for over 8 months without any troubles. Even though Intel CPUs have a projected life of 10 years, they will realistically be out of date well before that, so any life-shortening tricks like overclocking will not matter. Try **www.aceshardware.com/articles/how-to/overclockcrazy.shtml** for a good article on this.

The Pentium

Essentially two 486s in parallel (SX and DX), so more instructions are processed at the same time; typically two at once, assuming software can take advantage of it, and get the timing of the binary code just right. It has separate 8K caches, for instructions and data, split into banks which can be accessed alternately. It has a 64-bit external bus, but is 32-bit internally. Also, the data bus is not necessarily as large as the address bus. The core speed (in the chip; not the core voltage, for MMX) will be more than the external, or front side bus, speed, so a 90 MHz CPU has the bus running at 60 MHz. The multiplication is set by two external pins, BF0 and BF1, so you can run a 100 MHz Pentium at 1.5 rather than 2, and with an external speed of 66 MHz, as opposed to 50. The PCI bus can be switched also). 60 and 66 MHz versions are 5 volt-the remainder approx 3.3v. 3.52 Volts is known as the VRE spec, also used by Cyrix. Three codes indicate the voltage an earlier Pentium has been tested at:

Code	Voltage	Allowed Range
V	Standard	3.135-3.465v (3.3v)
VR	Voltage Regulated	3.3-3.465v
VRE	Voltage Regulated Extension	3.45-3.6v (3.52v)

VR processors won't run if the voltage is below 3.3v, and VRE processors need a higher voltage to run at all, so these codes stem from quality control. VRE became a standard because the higher voltage allows a chip to be run faster. On a newer Pentium, voltage information will be on the bottom, after the s-spec marking, which is a 3-digit number following SX, SK, SU, SY, or SZ, which includes such things as stepping, or version numbers, together with other characteristics. For voltages, there will be a slash mark followed by three letters, such as SK110/ABC. It all decodes as follows:

MARKINGS

Spec	SX???, SY???, SK???, Q0???
Vcc (A	S=STD V=VRE (3.52, or 3.135-3.6v)
Timings(B)	S=STD
Timings(C)	S=STD M=Min valid MD timing
DP Support	S=STD U=Uniprocessor and multiprocessing
I75	For 75MHz
iPP	For 75/90/100/120/133MHz

In other words, the first letter after the slash indicates voltage class, the second the timing specification and the last the dual processor capability. The best processor (for overclocking anyway) is one with SSS after the slash. The worst? VMU. iPP just means you have a P54C.

PENTIUM PRO

This is a Socket 8 RISC chip with a 486 hardware emulator on it. Several techniques are used by it to produce more performance than its predecessors; speed is achieved by dividing processing into more stages, and more work is done within each clock cycle; three instructions can be decoded in each one, as opposed to two for the Pentium. In addition, instruction decoding and execution are decoupled, which means that instructions can still be executed if one pipeline stops (such as when one is waiting for data from memory; the Pentium would stop all processing at this point). Instructions are sometimes therefore executed out of order, that is, not necessarily as written down in the program, but rather when information is available, although they won't be that much out of sequence; just enough to make things run smoother.

It has an 8K cache for programs and data, but has the processor and a 256K L2 cache in the same package, able to cache up to 64 Gb. The cache runs at full processor speed. The chip is optimised for 32-bit code, so will run 16-bit code no faster than a Pentium. Good for multiprocessor work.

PENTIUM II

An MMX-enhanced Pentium Pro using Slot 1 technology with no L2 cache on board, but included on the daughtercard inside the Slot 1 cartridge, running at half the processor speed on its own bus. The II can be slower than the Pro for certain applications, as the Pro's FPU is better and the L2 cache is on board the die (actually at the back, hence backside cache). It can also only cache up to 512 Mb of RAM, but it also has twice as much L1 and L2 cache. Up to 333 MHz, the P II only runs on a 66 MHz bus (even if you switch a BX chipset motherboard to 100 MHz, the chip can be autodetected and the bus speed reduced automatically. To get around this, see Dr Pabst's Hardware Page at **www.tomshardware.com**). Later versions, running above 350 MHz, can use a 100 MHz bus. The L2 cache on the 333s and above only use 2 chips instead of 4, which your BIOS needs to be aware of to get the maximum benefit.

The Pentium II Xeon (for Slot 2) has an L2 cache at full processor speed. The cartridge was used to house the cache, but now it is on-die again, due to the movement toward sockets.

PENTIUM III

Aside from higher clock speeds, the only essential difference between this chip and the PII is SSE, or Streaming SIMD Extensions. SIMD stands for Single Instruction, Multiple Data. Streaming concerns the transfer of long streams of data to and from memory, very useful for databases. Also included are a few extensions to MMX to speed up video processing, particularly 3D and lighting calculations, assuming your software can use the instructions.

PIIIs are made for specific bus speeds - those made with the Katmai (.25 micron) manufacturing process have a larger (512K) 2-way set associative L2 cache running at half the processor speed, i.e. the same as the P II. Not until the Coppermine version did the faster, full speed, L2 cache come along. In fact, there are two versions of the PIII - one with 256K L1 cache and one with 512K.

Coppermine is the .18 micron process (look for the *E* suffix, for *Enhanced*, up to 600 MHz - the 650 MHz ones all use it - it means Advanced Transfer Cache and Advanced Buffering

Support) with 256K of 8-way set associative L2 cache running at full processor speed, because it is now on the die. A *B* suffix indicates the ability to run with the 133 MHz *FSB* (533, 600 MHz). 677 & 733 MHz processors don't have suffixes because they are the only chips to run at that speed, so there is no confusion. All used Slot 1 initially, but migrated to Socket 370 (the flip-chip module), as used by the Celeron, which means they can be made on the same production line. This makes it a similar size chip to the Pentium Pro. This sort of design is possible because cache chips are not needed locally or on a daughtercard, as the process is down to .18 microns, leaving more room on-chip. This also means a cooler chip, less heating and more overclocking! One last note - later (Coppermine) versions use 1.65v, as opposed to 2v.

The Pentium III-M (for mobile) uses a .13 micron manufacturing process, which allows an increase in the L2 cache to 512K.

PENTIUM IV

Something different from the P6 derivatives, this one uses NetBurst micro-architecture, and SSE2 (Streaming SIMD Extensions 2), but will need special software to use them properly, so don't expect immediate performance gains, even though it uses clock rates over 1.8 Ghz with a 400 MHz Front Side Bus, which looks like three times faster than the Pentium III's (133 MHz) and twice as fast as the Athlon's, just to put it in perspective (it also has 55 million transistors, compared to ten), but it's really a 100 MHz FSB working four times as hard, using two overlapping strobe signals to transfer data on the rising and falling edges of each pulse (AMD use a similar trick with the Athlon). The FPU was scaled down to make room for so many transistors, and because it runs so fast, it needs more power, which generates heat, and the case and power supply are different to cope with this (the heat sink mounts directly to the case). There is a new ATX 12V connector, and a 350+ watt power supply is recommended.

A lot of the P IV's facilities are there for future development.

Although Intel's idea is to use RDRAM with it, through its 850 chip set, other chipsets support different memory, with less latency, cost and bandwidth. RIMMs must be fitted in pairs, as the system is dual-channel, with empty slots filled with Continuity RIMMs using a pass-through arrangement. The original socket had 423 pins, superseded by 478.

It improves on the P6's out-of-order core technique, and can have up to 126 instructions in progress, 3 times more than the PIII. Although L1 cache reduces to 8K, it is more efficient. Given that the maximum clock speed of a processor is governed by the length of its pipeline (each stage needs a certain amount of time), the P IV's is 20 stages long, reducing the number of gates per stage and allowing them to complete quicker. This also helps with latency, as does the *Rapid Execution Engine*, concerning the ALUs, which are clock-doubled so their operations complete in half a clock cycle.

The Prescott series (which can clock up to 2.8, 3 3.2 and 3.4 GHz) are distinguished from the Northwood series with an E designation. They are made with a 90 nm process and have a longer 31-stage pipeline.

Centrino

Centrino is actually a collection of components, starting with the Pentium-M processor (which is actually in combination with the Pentium III-M, so you get more performance per clock cycle on top of the faster 400 MHz FSB), 855 chipset and ending with a Wireless 2100 network connection.

Celeron

A cut-down version of the PII aimed at the low-cost market, initially supplied without an L2 cache, which prompted the unofficial name of DeCeleron. It was subsequently reissued with 128K of L2 cache running at processor speed, resulting in a chip that has gained some respect, especially as it rivals the PII in many areas. It started off using Slot 1, but now uses Socket 370, with the proviso that, from 533 MHz, Coppermine (.18 micron) technology was used and won't necessarily fit your socket, as some of the pinouts were changed. Converters are available, though, including those to allow Socket 370 chips to use Slot 1. Although the chip is as fast, if not faster, than PIIs or even PIIIs, its front side bus only runs at 66 MHz (although 100 MHz ones compete with the Duron, if they use the 440ZX chipset). Also, be aware that the 400 and 433 MHz versions use fixed clock multipliers of 6 and 6.5, which means 600 and 650 if you try to use an FSB of 100 MHz.

It would appear that the 1.7 MHz Celeron has reverted to 128K of L2 cache, so will be outperformed by the 1.3 or 1.4 GHz versions, which have 256K, even though the FSB runs at only 100 MHz (the 1.7's FSB runs at 400 MHz). Try an older PIV or Athlon instead.

Cyrix Instead

The 6x86 is a Pentium-type chip with Pentium Pro characteristics, as it can execute faster instructions out of sequence, amongst other things. It is also made by IBM under licence (see below). They use a P-Rating to determine performance relative to the Pentium, so a 6x86-166 is equivalent to a Pentium 166, even when it runs at 133 MHz. The 233MHz version of the 6x86MX uses a 75 MHz bus, for which you should use a Cyrix-specific chipset, since no Intel chipset runs at that speed. Well, officially, anyway. These would include SiS, ALI and VIA, which all work with Intels, of course. The MediaGX is based on the 5x86 and includes a graphics controller, DRAM controller and PCI bus interface. There are lots of functions on Cyrix chips that need BIOS support, and there are lots of separate utilities that will turn them on (such as the L1 cache under NT) if the support isn't there.

VIA, the chipset manufacturers, purchased Cyrix and produced the VIA Cyrix III processor, running at speeds between 533MHz and 667MHz. It has a very low power consumption rate, at 10 watts and is compatible with Socket 370.

The Cyrix 4 is completely new, although there a few common links between it and the III. It has 2 SSE units, to support Streaming SIMD extensions, and has been designed for high clock speeds - there are 17 stages in its pipeline to help cope with this. It supports Out Of Order Execution, but only with MMX.

486

A DX/4 with iDX4 pinout has "DX4 P/O" on the second row of the lower CPU label. One with M7 pinout does not have this indication and the lower CPU label has only two rows. The line might look like this:

```
Cx 486DX2 V 66 G P
```

where:

Cx	Manufacturer
486DX2	Chip type
V	Voltage V=3.3-4v. Blank=5v
66	Speed range
G	Package. G=168 Ceramic
P	Commercial. 0-85.C at 5v. 0-70.C at 3.3-4v

586

If not labelled, (028) = STD, (16) = VRE . If the label is6x86L and 2.8V, use P55C settings.

6X86

Should be labelled with core voltage. VCC spec (028) = 3.4-3.7v, (16) = 3.15-3.45v.

IBM

IBM has a specific licence to make chips produced by Intel, so they can use the official masks (photographic blueprints) that others cannot, as well as adding more features. The Blue Lightning was a 32-bit chip similar to the 486DX.

AMD

For the K5, use the same settings as the equivalent Pentium. After the *P-rating* on the face of the chip are three letters. The first is the package type, the second is the voltage and the third the case temperature. Voltages are in the first table and temperatures in degs C in the second:

K5 Letter	Voltage (Core/I/O)
B	3.45-3.6 (3.5)
C	3.3-3.465 (3.3)
F	3.135-3.465 (3.3)
G	x/y
H	2.86-3/3.3-3.465 (2.9/3.3)
J	2.57-2.84/3.3-3.465 (2.7/3.3)
K	2.38-2.63/3.3-3.465 (2.5/3.3)

Letter	^oC	Letter	^oC
Q	60	X	65
R	70	Y	75
W	55	Z	85

The K6 has a 64K cache, as opposed to 32K on Intel chips, and uses a RISC core with two decoder units to translate x86 commands that are parallel-processed 6 at a time. The K6-2 and 3 (Socket 7) are better versions of the same and easily give the Pentium II a run for its money. A feature called *Write Allocation* lessens the impact of a L1 cache miss and increases performance by about 5%, assuming your chipset behaves properly with it. Write allocation is a form of prefetching data from main memory, based on its locality. That is, when data is fetched, everything around it is grabbed as well and only needs to be allocated properly - the larger the buffers, the more the chances that the data you want is actually there.

The Athlon, or K7, was influenced by the DEC Alpha and started off with Slot A technology, intended for OEMs. Now it uses Socket A. It has three sets of three processing units that can work without waiting for the others to finish what they were doing. Three are for floating point or MMX calculations, because 3D uses x,y and z coordinates. Another three are for integer calculations and the third three take some load off the other two by calculating addresses for them. It is therefore theoretically possible to get 9 operations per clock cycle.

The 128K of L1 cache and 512K of half-speed off-chip L2 cache (on earlier models) has its own special bus. Later models have 256K of L2 cache on-chip, so expect to see a total of 384K. The L1 is four times larger than that on the PIII, and the Athlon can decode any three x86 instructions at a time, whereas the PIII can only do this if two of the three are simple and relate to a single internal operation. It can also send up to nine internal instructions per clock cycle compared to the PIII's five.

The Athlon 4 (formerly known as the Palomino), or XP, was originally meant for notebooks, against the mobile Pentium III and Pentium IV, but can also be used in dual-CPU machines. It succeeds the successful Thunderbird. Its 256K L2 cache has a prefetch system that is said to improve application performance by 5-15%. There are also enhancements to 3DNow and power saving, and it is made of a fibreglass substrate rather than ceramic packaging.

The system bus (between CPU and system logic) also runs at 200 MHz, being developed from the EV6 bus used with the DEC Alpha, so a new chipset is required, supplied by AMD initially (this is really a 100 MHz FSB doing twice the work). A side benefit is that multiprocessor chipsets for the Alpha 21264 will also support the Athlon. Slot A looks like Slot 1, but the pinouts are different, partly for copyright reasons and partly because AMD felt it was time for change. The result is a bus design that is technically superior. Athlon power consumption is relatively high. The Athlon MP, basically identical, supports a snoop bus, which allows each CPU in a multiprocessor system to see what the other has in its cache.

The Duron uses Socket A as well (it's actually based on the Athlon core, varied for the target market), and a 200 MHz Front Side Bus, against the Celeron's 66 MHz, so it's a good upgrade path to the Athlon because you don't have to change your motherboard. Its 192K of cache (128K L1 and 64K L2) certainly helps, but, when included in some cheap machines, is badly let down by support chips. The performance of the 900 MHz version is very close to the 1 GHz Pentium III and Athlon CPUs.

Above about 1.2 GHz, the Athlon uses DDR SDRAM, which doubles the FSB speed to 266 MHz, and increases cost. The Duron can also use it, but officially stays with SDRAM as it is meant to be lower end.

IDT

This company makes, or made, the WinChip (the C6), which was designed to run Windows business applications. The 200-speed version performs about 18% faster than the Intel 200 MMX and is approximately 25% cheaper. It is single-voltage, so you can get MMX on older motherboards, has a larger internal cache and disables the L2 cache when fitted, on the basis that a multitasking operating system tends not to benefit from it anyway. The company is now owned by VIA, the chipset manufacturer.

Transmeta

This company produces the *Crusoe*, designed to boost battery life in portables. It is a VLIW processor and translates x86 commands into its own language, so it's an emulator, in effect - the work is done in software, so it needs less transistors, and therefore less power.

MMX

This is an extension to x86 code to allow the better handling of the repetitive instructions typically found with multimedia applications, allowing parallel processing of many data items with only one instruction, or as many 8-bit instructions as will fit into a 64-bit register, so video, at least, will be smoother and faster. For example, normal Pentiums only process 1 pixel per clock cycle, where the 64-bit MMX registers will be able to handle 8, although a 32 K cache also has something to do with it. MMX also performs many of the functions of sound, video or modem cards. The MMX processor's core runs between 2.0-3.5 volts, but the output uses 3.1-3.6v (3.3), so motherboards need 2 voltage regulators. Talking of which, see the chart at the end of the chapter for chip voltages and other settings. MMX uses Socket 7 and above. Intel chips have 2 MMX pipelines, whereas the AMD K6 and Cyrix 6x86 have only one, but their MMX registers are in a dedicated unit, so they only need one cycle to switch to MMX. On Intel chips, they are integrated into the FPU so you can't do maths and MMX instructions at the same time, and over 50 instructions are required to change from one function to the other. So, if you're using 3D video, for example, the MMX instructions produce the speed, but much of the advantage is lost after the coordinates are calculated by the FPU and the registers have to be changed over.

Summing up

In principle, the faster the CPU the better, but only if your applications do chip-centred work rather than writing to disk. For a database, which accesses the storage a lot, spend the money on a faster hard disk. Since the PCI bus runs at 33MHz (actually a proportion of the front side bus speed which, coincidentally, is often the same as the memory bus speed), the bottleneck is the disk I/O, running at much the same speed on them all. This is especially true if you use *Programmed I/O* (PIO), where the CPU scrutinises every bit to and from the hard drive (although *Multi-sector I/O* or EIDE will improve things).

As the Pentium 90's motherboard runs faster (60 MHz), I/O can proceed much faster (although a more sophisticated chipset helps). With 16 Mb of RAM, on the other hand, performance will be almost double anyway, because the need to go to the hard disk is so much reduced, and the processor can make a better contribution to performance. The biggest jump is from a DX2/66 to a DX/4, with the curve flattening out progressively up to the Pentium 90. There is also not a lot of difference between a 166-200 Mhz Pentium, the 200-

233 MHz MMX and 266-300 MHz Pentium II, unless you speed up the I/O systems. Intel's competitors do relatively poorly with the MMX and FPU side of things, so maybe combine them with a good quality graphics accelerator to narrow the gap for 3D, though this won't help with image editing.

In short, more memory won't improve boot up speed, or increase the clock speed. What it will do, however, is reduce the need to go to the swap file, and make switching between tasks quicker, so your performance improvements actually come after the machine has started.

CHIP REFERENCE CHART

Below are speeds of processors against the system clock. The PCI bus can run at up to 33MHz (officially), which used to be switchable on the motherboard to match the PCI bus to the CPU speed; for example, 33 MHz does not divide smoothly into the Pentium 120's FSB speed of 60, so you're automatically introducing synchronisation problems. Unofficially, the PCI bus can be run higher if the motherboard designer allows you to and your PCI cards can handle it. Although the PCI bus is not linked directly to the CPU, and can catch up here and there, switching properly does make a difference. Notice the motherboard speed of the P 150-slower than that of the 133!

Voltages are also included, but there might be slight differences from one motherboard manufacturer to another. For example, Asustek list the Cyrix/IBM 6x86MX as 2.9/3.3, where others might use 2.8/3.3. The difference will not do any harm other than a slight change in temperature or stability. Also, 3.3 or 3.5 volts for single plane processors refers to the STD or VRE settings, respectively. The higher voltage allows a cleaner detection between 0 and 1, hence more reliability at higher speeds, such as when overclocking.

Socket 7 is backwards compatible with Socket 5, but doesn't have enough bandwidth for the high end, hence Socket 8, for the Pentium Pro, with an extra (faster) 64-bit bus to talk directly to the L2 cache. Slot 1, for the Pentium II, is electrically identical to Socket 8, but an entirely different shape-it's actually a daughtercard inside a cartridge. Slot 2 is a larger version of Slot 1 that is meant for high-end machines. Luckily, later chips, like the Pentium Pro, support VID (Voltage ID) so it can be automatically regulated. I/O processes only take up about 10% of the power used by the CPU, so voltages for this will likely work within a range of settings.

Intel SLE 486DX/DX2/DX4/OPD CPUs marked with & E XXXX support green functions. P24Ds marked with & E W XXXX support writeback mode as well. The P24T-63/83 are Overdrive CPUs, and the board should be set to 5v. AMD normal CPUs are marked NV8T - the enhanced ones (with w/b cache) are marked SV8B.

Maker	Processor	Socket	Voltage	FSB	Clock X	PCI Bus
Intel	486DX (P24)	LIF/3	5	As CPU	1	As CPU
Intel	486DX2/50 (P24)	LIF/3	5	25	2	25
Intel	486DX2/66 (P24)	LIF/3	5/3.3	33	2	33
Intel	486SX (P23/P24)	LIF	5	As CPU	1	As CPU
Intel	486 SL-Enhanced		5	As CPU	1	As CPU

Maker	Processor	Socket	Voltage	FSB	Clock X	PCI Bus
Intel	486DX4/75	3	3.3	25	3	25
Intel	486DX4/100P24C	3	3.45	33	3	33
Intel	P24D		5			
AMD	486DX2/80	3	3.45	40	2	
AMD	486DX4/100	3	3.45	33	3	
AMD	486DX4/120	3	3.45	40	3	
AMD	486DX4/133	3	3.45	33	4	33
AMD	486SX	3	5	As CPU	1	As CPU
AMD	486DX	3	5	As CPU	1	As CPU
Cyrix/IBM	486DX	3	5	As CPU	1	As CPU
Cyrix/IBM	486DX2-V50	3	3.3	25	2	
Cyrix/IBM	486DX2-V66	3	3.6	33	2	
Cyrix/IBM	486DX2-V80	3	4	40	2	
Cyrix/IBM	486DX4-100	3	3.45	33	3	33
Cyrix/IBM	5x86-100	3	3.45	33	3	
Cyrix/IBM	5x86-120	3	3.45	40	3	
Cyrix/IBM	5x86-133	3	3.45	33	4	
Evergree	486 upgrade	1,2,3,6	5	33	4	33
Kingston	Turbo 133	1,2,3,6	5	33	4	33
Intel	P 60	4	5	60	1	30
Intel	P 66	4	5	66	1	33
Intel	Pent OD P5T	4	5	60/66	2	
Intel	Pent OD P54CTB	5/7	3.52	50/60/66	2.5	
Intel	P 54C-75	5/7	3.52	50	1.5	25
Intel	P 54C-90	5/7	3.52	60	1.5	30
Intel	P 54C-100	5/7	3.52	66	1.5	33
Intel	P 54C-100	5/7	3.52	50	2	25
Intel	P 54C-120	5/7	3.52	60	2	30
Intel	P 54C-133	5/7	3.52	66	2	33
Intel	P 54C-150	5/7	3.52	60	2.5	30
Intel	P 54C-166	7	3.52	66	2.5	33
Intel	P 54C-200	7	3.52	66	3	33
Intel	P54C-233	5/7	3.52	66	3.5	33
AMD	K5-PR75	5/7	3.52	50	1.5	25
AMD	K5-PR90	5/7	3.52	60	1.5	30
AMD	K5-PR100	5/7	3.52	66	1.5	33
AMD	K5-PR120	5/7	3.52	60	2	30
AMD	K5-PR133	5/7	3.52	66	2	33
AMD	K5-PR150	5/7	3.52	60	2.5	30
AMD	K5 PR166	5/7	3.52	66	2.5	33
Cyrix/IBM	6x86 P120+ (100)	5/7	3.52	50	2	25

Maker	Processor	Socket	Voltage	FSB	Clock X	PCI Bus
Cyrix/IBM	6x86 P133+ (110)	5/7	3.52	55	2	
Cyrix/IBM	6x86 P150+ (120)	5/7	3.52	60	2	30
Cyrix/IBM	6x86 P166+ (133)	5/7	3.52	66	2	33
Cyrix/IBM	6x86 P200+ (150)	7	3.52	75	2	37.5
Intel	P55C-166 MMX	7	2.8/3.3	66	2.5	33
Intel	P55C-200 MMX	7	2.8/3.3	66	3	33
Intel	P55C-233 MMX	7	2.8/3.3	66	3.5	33
AMD	K6 166	7	2.9/3.3	66	2.5	33
AMD	K6 200	7	2.9/3.3	66	3	33
AMD	K6 233	7	2.1/3.3	66	3.5	33
AMD	K6 266	7	2.2/3.3	66	4	33
AMD	K6 300	7	2.1/3.3	66	4.5	33
AMD	K6 PR233 (.35m)	7	3.2/3.3	66	3.5	33
AMD	K6 3D	7	2.2/3.3			
Cyrix/IBM	6x86MX PR150	7	2.9/3.3	60	2	30
Cyrix/IBM	6x86MX PR166	7	2.9/3.3	60	2.5	30
Cyrix/IBM	6x86MX PR200	7	2.9/3.3	66	2.5	33
Cyrix/IBM	6x86MX PR233	7	2.9/3.2	66	3	33
Cyrix/IBM	6x86MX PR266	7	2.9/3.2	66	3.5	37.5
IDT	C6	7	3.3			
Intel	Pro 150	8	3.1	60	2.5	30
Intel	Pro 180	8	3.3	60	3	30
Intel	Pro 200	8	3.3	66	3	33
Intel	Pentium II 233	Slot 1		66	3.5	33
Intel	Pentium II 266	Slot 1		66	4	33
Intel	Pentium II 300	Slot 1		66	4.5	33
Intel	Pentium II 333	Slot 1		66	5	33
Intel	Pentium II 350	Slot 1		100	3.5	33
Intel	Pentium II 400	Slot 1		100	4	33
Intel	Pentium III Katmai	Slot 1	2ish*	100		
Intel	P III 500/550	370	1.65ish**	133		

*1.93-2.07v **1.52-1.69v

A board with the same jumper settings for the 1.5x and 3.5x clock multipliers has the 233 MHz chip wired internally to redefine the original 1.5 setting. And if you ever wondered, overleaf are the specs for the sockets:

Socket	Description
LIF	486 boards, no lever
ZIF 1	486 boards, with 168 or 169 pins
ZIF 2	486 boards, 238 pins
ZIF 3	486 boards, 237 pins, most common
ZIF 4	Pentium P5 (60/66)
ZIF 5	Pentium Classic (P54C), single voltage, up to 166 MHz
ZIF 6	486 boards, 235 pins
ZIF 7	As for ZIF 5, plus 1 pin for Overdrive P55CT, over 166 MHz
ZIF 8	Pentium Pro
Slot 1	Pentium II/Some Celerons
Slot II	Pentium III
Slot A	AMD Athlon
Socket 370	Pentium III/Some Celerons
Socket 423	Pentium 4
Socket 478	Pentium 4, with extra power and ground pins, about half size of 423

MEMORY

3

The memory contains the instructions that tell the Central Processor what to do, as well as the data created by its activities. Since the computer works with bits that are either on or off, memory chips work by keeping electronic switches in one state or the other for as long as they are required. Where these states can be changed at will or, more properly, the operating system is able to reach every part of memory, it is called *Random Access Memory*, or RAM, which comes from when magnetic tapes were used for storage, and information could only be accessed sequentially; that is, not at random. A ROM, on the other hand, has its electronic switches permanently on or off, so they can't be changed, hence *Read Only Memory*. ROMs are *non-volatile*, meaning that data inside isn't lost when the power is off. System memory, described below, is volatile, so RAM disks are vulnerable.

STATIC RAM

Static RAM (SRAM) is the fastest available, with a typical access time of 20 nanoseconds (the lower the number, the faster the chip can be accessed). It is expensive, however, and can only store a quarter of the data that Dynamic RAM (or DRAM) is able to, as it uses either four transistors and two resistors, or six transistors to store a bit against DRAM's one, although it does retain it for as long as the chip is powered (the transistors are connected so that only one is either in or out at any time; whichever one is in stands for a 1 bit). Synchronous SRAM allows a faster data stream to pass through it, because it uses its own clock, which is needed for cacheing on fast Pentiums. Because of its expense, SRAM is used in caches in the CPU and between it and system memory, which is composed of Dynamic RAM.

DYNAMIC RAM

DRAM uses internal capacitors to store data (actually one transistor and a capacitor for each bit of memory), with a MOSFET transistor charging or discharging the capacitor to create your 1s and 0s in a write operation, or just to sense the charge, which is a read. The capacitors lose their charge over time, so they need constant refreshing to retain information, otherwise 1s will turn into 0s. The result is that, between every memory access, an electrical charge refreshes the capacitors to keep data in a fit state, which cannot be reached during that time (like changing the batteries millions of times a second). Normal bus operation is a 2-clock cycle external bus access; the first is called T1, and the second T2. Address and control signals are set up in the former, and the operation completed at the end of the latter.

Burst bus operation executes 4 consecutive external bus cycles. The first is the same setup and completion done in T1 and T2, and the next three operate without the setup cycle, by defining the addresses that follow the first. As the first takes the longest, burst timings look like 2-1-1-1 or similar. Memory addresses are found by a combination of row and column inside memory chips, with two strobe signals, *Row Address Strobe* (RAS) and *Column Address Strobe* (CAS), normally in that order. Fast Page Mode memory, for example, toggles CAS on and off as addresses change, that is, as columns are accessed within the row (further described under Wait States, below). FPM makes 60 ns RAM look like 40 ns, allowing you a 25 MHz CPU. Quick explanation:

Under normal circumstances, a 33MHz CPU takes about 30 ns per cycle:

Clock Speed	Cycle Time
1 MHz	1000 ns
5	200
8	125
12	83
16	63
20	50
25	40
33	30
40	25

At that speed, memory chips must operate at something like 20 nanoseconds to keep up, assuming the CPU needs only 1 clock cycle per 1 from the memory bus; 1 internal cycle for each external one, in other words. Intel processors mostly use 2 for 1, so the 33 MHz CPU is actually ready to use memory every 60 ns, but you need a little more for overheads, such as data assembly and the like, so there's no point in using anything faster anyway. With Static Column memory, CAS may be left low (or active) with only the addresses changing, assuming the addresses are valid throughout the cycle, so cycle time is shorter.

The *cycle time* is what it takes to read from and write to a memory cell, and it consists of two stages; *precharge* and *access*. Precharge is where the capacitor in the memory cell is able to recover from a previous access and stabilise. Access is where a data bit is actually moved between memory and the bus or the CPU. Total access time therefore includes the finding of data, data flow and recharge, and parts of it can be eliminated or overlapped to improve performance, as with SDRAM. The combination of *Precharge* and *Access=Cycle Time*, which is what you should use to calculate wait states from (see below). Refresh is performed with the 8253/8254 timer and DMA controller circuit (Ch 0).

Refreshes can be made to happen so that the CPU doesn't notice (i.e. Concurrent or Hidden), which is helped by being able to use its on-board cache and not needing to use memory so often anyway-turn this off first if you get problems. In addition, you can tinker with the Row Access Strobe, or have Column Access Strobe before RAS, as described in the *Advanced Chipset Setup* (see the *Memory* section). The fastest DRAM commonly available is rated at 60 nanoseconds (a nanosecond is a billionth of a second). Although SDRAM is rated at 10ns, it is not used at that speed - typically, between 20-50 ns is more like it, since the smaller figure only refers to reads from sequential locations in bursts - the larger one is for the initial data fetch. With a CPU clock cycle at 500 MHz taking, say, 2ns, you will get at least 5 CPU clock cycles between each SDRAM cycle, hence the need for special tricks. As memory chips need alternate refresh cycles, under normal circumstances data will actually be obtained every 120 ns, giving you an *effective speed* of around 8 MHz for the *whole computer*, regardless of CPU speed, assuming no action is taken to compensate, which is a sobering thought when you're streaming audio through an ISA sound card.

One way of matching components with different speeds is to use wait states.

WAIT STATES

More relevant on older machines, these indicate how many ticks of the system clock the CPU has to wait for other parts of the computer, typically for memory-it will generally be 0 or 1, but can be up to 3 if you're using slower memory chips. They are needed because there is no "data valid" signal from memory, so the system waits a bit to ensure it's OK. Ways of avoiding wait states include:

- **Page-mode memory**. This uses cut-down address cycles to retrieve information from one general area, based on the fact that the second access to a memory location on the same page takes around half the time as the first; addresses are normally in two halves, with high bits (for row) and low bits (for column) being multiplexed onto one set of address pins. The page address of data is noted, and if the next data is in the same area, a second address cycle is eliminated as a whole row of memory cells can be read in one go; that is, once a row access has been made, you can get to subsequent column addresses in that row in the time available (you should therefore increase row access time for best performance). Otherwise, data is retrieved normally, taking twice as long. *Fast* Page Mode is a quicker version of the above; the DRAMs concerned have a faster CAS (Column Access Strobe) access speed, and can anticipate access to the next column while the previous column is deactivating, and the data output buffer is turned off, assuming the data you need is in that location. Memory capable of running in page mode is different from the normal bit-by-bit type, and the two don't mix. It's unlikely that low capacity SIMMs are so capable. With banks of page mode DRAM in multiples of 2, you can combine it with ...
- **Interleaved memory**, which divides memory into two or four portions that process data alternately, so you can address multiple segments in parallel (rather like disk striping); that is, the CPU sends information to one section while another goes through a refresh cycle; a typical installation will have odd addresses on one side and even on the other (you can have *word* or *block* interleave). If memory accesses are sequential, the precharge of one will overlap the access time of the other. To put interleaved memory to best use, fill every socket you've got (that is, four 64 Mb SIMMs are better than two 128 Mb ones, and if you only have three sockets, it won't work if you only fill two). The SIMM types must be the same.

- A **processor RAM cache**, which is a (Level 2) bridge between the CPU and slower main memory. It consists of anywhere between 32-512K of (fast) Static RAM chips and is designed to retain the most frequently accessed code and data from main memory. It can make 1 wait state RAM look like that with 0 wait states, without physical adjustments, assuming the data the CPU wants is in the cache when required (a *cache hit*). To minimise the penalty of a cache miss, cache and memory accesses are often made in parallel, with one being terminated when not required.

  How much L2 cache you need really depends on the amount of system memory; according to Dell, jumping from 128K to 256K only increases the hit rate by around 5%, and Viglen think you only need more than 256K with over 32 Mb RAM. L2 cache is not as important if you use Fast Page Mode DRAM, but once you start clock doubling, and increasing memory writes, the need for a writeback cache becomes more apparent. Several Intel chipset designs, such as the HX (for Socket 7) may need additional Tag RAM to cache more than 64 Mb (i.e. more than 8-bit). Pentium Pro/II boards aren't restricted this way, as the cache is with the processor, originally on the same circuit board, but now inside the core. This helps with quality control, since the motherboard doesn't have to cope with too many variations of cache chip, and can have loser manufacturing tolerances.

  A cache should be fast and capable of holding the contents of several different parts of main memory. Software plays a part as well, since cache operation is based on the assumption that programs access memory where they have done so already, or are likely to next, maybe through looping (where code is reused) or organising what's wanted to be next to other relevant parts. In other words, it works on the principle that code is sequential, and only a small proportion of it is used anyway. In fact, as cache is used for 80-90% of CPU memory accesses, and DRAM only 1-4% of the time, less errors result (actually a lower *Soft Error Rate*), hence the reduced need for parity. A side effect is that DRAM speed is not so critical.

  Asynchronous SRAM is the cheapest solution, which needs wait states. A basic design will look up an address for the CPU and return the data inside one clock cycle, or 20 ns at 50 MHz, with an extra cycle at the start for the tag lookup. As the round trip from the CPU to cache and back again takes up a certain amount of time, there's less available to retrieve data, which total gets smaller as the motherboard speed is increased.

  Synchronous SRAM chips have their own internal clock and use a buffer to keep the whole 2 or 3 cycle routine inside one. The address for data required by the CPU is stored, and while that for the next is coming in to the buffer, the data for the previous set is read by the CPU. It can also use burst timing to send data without decoding addresses or, rather, sending addresses only once for a given data stream.

  Pipeline SRAM also uses buffers, but for data reads from memory locations, so the complete distance doesn't have to be travelled, so it's a bit like a cache within a cache. *Pipeline Burst SRAM* will deliver 4 words (blocks of data) over four consecutive cycles, at bus speeds over 75 MHz. Up till 66 MHz, it delivers about the same performance as synchronous, but is cheaper to make. Data is read in packets, with only the first step slower than the other three, as it has to get the address as well. You will see these settings described as 2-1-1-1 or similar, where lower numbers mean faster access.

In practice, it would appear that performance between synchronous and pipeline cache is similar. Asynchronous is not often found on fast motherboards, anyway, but should be about the same at or below 50 MHz (it's slowest and cheapest). Note that L2 cache can be unreliable, so be prepared to disable it in the interests of reliability, particularly with NT, but that defeats the object somewhat.

For maximum efficiency, or minimum access time, a cache may be subdivided into smaller blocks that can be separately loaded, so the chances of a different part of memory being requested and the time needed to replace a wrong section are minimised. There are three mapping schemes that assist with this:

- **Fully Associative**, where the whole address is kept with each block of data in the cache (in tag RAM), needed because it is assumed there is no relationship between the blocks. This can be inefficient, as an address comparison needs to be made with every entry each time the CPU presents the address for its next instruction. Associative mapping relates specific cache cells to specific main memory cells based on the low order bits of the main memory address.
- **Direct Mapped**, also known as 1-way associative, where a block of memory is mapped to one place in the cache, so only one address comparison is needed to see if the data required is there. Although simple, the cache controller must go to main memory more frequently if program code needs to jump between locations with the same index, which seems pointless, as alternate references to the same cache cell mean cache misses for other processes. In other words, memory cells mapped to the same location in cache will kick each other out. The "index" comes from the lower order addresses presented by the CPU.
- **Set Associative**, a compromise between the above two. Here, an index can select several entries, so in a 2 Way Set Associative cache, 2 entries can have the same index, so two comparisons are needed to see if the data required is in the cache. Also, the tag field is correspondingly wider and needs larger SRAMs to store address information.

  As there are two locations for each index, the cache controller has to decide which one to update or overwrite. The most common methods used to make these decisions are Random Replacement, First In First Out (FIFO) and Least Recently Used (LRU). The latter is the most efficient. If the cache size is large enough (say, 64K), performance improvements from this over direct-mapping may not be much. Having said that, 2-way set can be better than doubling the size of a direct-mapped cache, even though it is more complex. The higher the set-associativity, the longer it takes for the cache controller to find out whether or not the requested data is in the cache. 2-way set-associative cache allegedly equals the performance of a direct-mapped one twice its size. To find the equivalency, multipy the associativity by the size-a 256K cache with an eight-way associativity comes out as 2 Mb, whereas 512K with 2-way is 1 Mb.

NT can figure out the size of any set-associative L2 cache, using its Hardware Abstraction Layer. If not (you may have a direct-mapped cache) it assumes 256K. To change it, go to **HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management\SecondLevelDataCache**. Open a DWORD editor window, change from Hex to Decimal, then insert your L2 cache size in Kb.

A *Write Thru Cache* means that every write access is saved to the cache and passed on to memory at the same time, so although cache and memory contents are identical, it is slower, as the CPU has to wait for DRAMs. Buffers can be used to provide a variation on this, where data is written into a temporary buffer to release the CPU quickly before main memory is updated (see *Posted Write Enable*).

A *Write Back Cache*, on the other hand, exists where changed data is temporarily stored in the cache and written to memory when the system is quiet, or when absolutely necessary. This will give better performance when main memory is slower than the cache, or when several writes are made in a very short space of time, but is more expensive, and the L2 cache can only handle half the amount of RAM. "Dirty words" are the differences between cache and main memory contents, and are kept track of with dirty bits. Some motherboards don't have the required SRAM for the dirty bit, but it's still faster than Write Thru.

Write Back becomes more important with clock doubling, where more memory writes are created in the course of a CPU's work, but not all motherboards support it. Early Write cache exists where the address and data are both known and sent simultaneously to SRAM. A new address can be used once every clock. Late Write is where data follows the address by 1 clock cycle, so a new address can be written to every 2nd clock.

DOS-based software is happy with a 64K external cache because 64K is the largest chunk of memory that can be addressed, which is also true for Windows (3.x anyway) because it runs on top of DOS. You may need something like Windows '95, OS/2, Windows NT, Multiuser DOS 7, REAL/32 or NetWare 3/4 to get much out of a larger L2 cache. DOS has hit-rates of around 96% while multi-tasking operating systems tend to achieve 70% or so, because of the way that they jump around memory, so a cache can slow things down against a cache-less motherboard with efficient memory management. With multi-tasking, interleaving can often get more performance than a cache (e.g. check out Headland/ICL and OPTi chipsets). Not only that, cache management often delays memory access by 1 to 2 clock cycles.

- **Refresh Bypass**, used by AMI on their 486-based motherboards.
- **Synchronous DRAM**, whose timing is linked directly to the PC's system clock, so you don't need wait states (see below).

EDO (*Extended Data Output*) is an advanced version of fast page mode (often called *Hyper Page Mode*, but see below), which can be up to 30% better and only cost 5% more. *Single-cycle EDO* carries out a complete memory transaction in 1 clock cycle by overlapping stages that otherwise would take place separately; for example, precharging can start while a word is still being read, and sequential RAM accesses inside the same page take 2 clock cycles instead of 3, once the page has been selected, because the data output buffer is kept open rather than being turned off, as it would be with Fast Page Mode Memory (see *Wait States*, above). It is assumed that if one address is needed, others nearby will be, too, so the previous one is held open for a short while. In other words, output is not turned off when CAS goes high (i.e. turned off, or has stopped allowing addresses to be moved to the device). In fact, data can still be output after CAS has gone high, then low again (and another cycle has therefore started), hence the name, *Extended Data Out*. Data remains available until that from the next access begins to appear - a memory address can hold data for multiple reads. This means you can begin precharging CAS whilst waiting for data. The end result is that cycle time is cut by around 20% and data is available longer. The really neat thing is that CAS can go high before data appears (well, maybe not to you and me, but it is to a motherboard designer). EDO is only faster with memory reads, though; writes take place at the same speed as Fast Page Mode. In any case, it only works if your cache controller supports pipeline burst transfer. When it does, it effectively reduces 60 ns RAM to 25 ns, giving you a 40 MHz CPU, without wait states.

The combination of DRAM plus an external latch between it and the CPU (or other bus mastering device), would look like EDO DRAM because the external latch can hold the data valid while the DRAM CAS goes high and the address is changed. It is simpler and more convenient to have the latch inside the DRAM, hence EDO. As it replaces a Level 2 cache and doesn't need a separate controller, space on the motherboard is saved, which is good for notebooks. It also saves on battery power. In short, EDO gives increased bandwidth due to shortening of the page mode cycle (and 3-2-2-2 bursts rather than 7-4-4-4)-an entire block of memory can be copied to its internal cache and a new block collected while the CPU is accessing it. It appears to be able to run (unofficially) above 66 MHz. Don't get 70 ns EDO, as it will be difficult to upgrade the CPU.

BEDO, or *Burst Extended Data Out*, is as above, but has a pipeline stage and a 2-bit burst counter that can read and write large streams of data in 4-cycle bursts for increased performance, based on the addresses being dealt with in the first cycle. The pipelining system can save 3 cycles over EDO. It is designed to achieve 0 wait state performance at 66 MHz and upwards, as it brings your 60 ns RAM down to 15 ns (again, see chart above). The relevant speeds for Fast Page Mode and EDO are 25 and 40, respectively, and the increase in performance 100% and 40%.

Enhanced DRAM (**EDRAM**) replaces standard DRAM and the L2 cache on the motherboard, typically combining 15ns SRAM inside 35ns DRAM. Since the SRAM can take a whole 256 byte page of memory at once, it gives an effective 15ns access speed when you get a hit (35ns otherwise), so system performance is increased by around 40%. The L2 cache is replaced with an ASIC chip to sort out chipset/memory requirements (an ASIC chip is one specially made for the purpose). EDRAM has a separate write path that accepts and completes requests without affecting the rest of the chip. NEC is producing **RDRAM** which, they say, gives 2 ns access speed. It interconnects with a system called **RAMBUS**, which is a narrow (16-bit), but ultra high speed, local memory bus, made with CMOS technology. It also uses a packet technique for data transfer, rather than coping with individual bytes. BIOS support is needed in the chipset for this to work as system memory.

Although its data transfer rate is twice that of SDRAM (see below), it suffers from latency problems, which reduces the performance edge, and is more expensive. RDRAM has its own communications bus with a separate controller that mediates between it and the CPU, using a relatively narrow serial connection 16 bits wide with separate lines for rwo and column signals. It runs at 400 MHz and uses both sides of the clock cycle. As the signal lines are separate from the data lines, you can be reading or writing at the same time as preparing for a second or even a third operation. The memory itself uses 184-pin RIMMs, which are similar to DIMMs but with a heat sink, required because the chips are more tightly packed together, even though they require less power and generate less heat. The 820 chipset supports only 2 RAMBUS modules from mixed suppliers. It works better with the Pentium IV, but RAMBUS stuff has now almost disappeared anyway. The various modules available are PC800 and PC1066, which refer to their speeds of operation, so they are capable of delivering 1600 and 2132 Mbytes/sec. Being 16-bit, they must be used in pairs. Used in dual channels, the memory system can just about keep up with the P4.

PC4200 modules run at 1066MHz and are 32-bit, meaning that you only need one at a time (it actually consists of 2 16-bit RIMMs back-to-back, although they have 232 pins). Do not confuse 800Mhz 32-bit RIMMs with DDR400, which are both called PC3200.

Virtual Channel RAM (**VCRAM**) is a development of SDRAM (see below), also from NEC, using standard DIMM sockets. Because of its low latency and speed (133 MHz), it is a very good choice, and is supported by VIA's Apollo Pro Plus chipset.

WRAM (*Windows RAM*), created by Samsung, is dual ported, like VRAM, but costs about 20% less and is 50% faster with around 25% more bandwidth (dual porting means reading and writing takes place at the same time). It runs at 50 MHz and can transfer blocks and support text and pattern fills. In other words, some graphics functions are built in, so look for these on graphics cards. VRAM, by the way, is used on graphics cards that need to achieve high refresh rates; DRAM must use the same port as it does for data to do this, where VRAM uses one port to refresh the display and the other to change the data. Otherwise, it is generally the same speed as DRAM. **SGRAM**, or *Synchronised Graphics RAM*, is single ported, using dual banks where 2 pages can be opened at once. It has a block write system that is useful for 3D as it allows fast memory clearing.

Synchronous DRAM (**SDRAM**) was originally a lower cost alternative to VRAM, using a 168-pin DIMM. It is synchronised to the system clock (that is, the external CPU frequency), taking memory access away from the CPU's control; internal registers in the chips accept a request, and let the CPU do something else while the data requested is assembled for the next time it talks to the memory, as the memory knows when the next cycle is due because of the synchronisation. In other words, SDRAM works like standard DRAM, but includes interleaving, synchronisation and burst mode, so wait states are virtually eliminated (SDRAM DIMMs also contain two cell banks which are automatically interleaved). It's not actually faster than DRAM, just more efficient. In fact, the main specification for SDRAM when shopping is its access speed, which originally was 12 or 10 ns, latterly becoming 8 ns, which relates to the PC 100 standard. PC 133 uses 7.5 ns.

Data bursts are twice as fast as with EDO (above), but this is slightly offset by the organisation required. The peak bandwidth of 133 SDRAM is about 33% higher than that of

100. *Registered DIMMs* are meant for mission critical systems where the data must arrive in the proper format. They contain a register, or buffer chip, between the memory controller and chips on the DIMM to delay everything by one clock cycle to ensure everything is there. Buffered memory, which is nearly the same, redistributes the addresses and reduces the load on the memory clock, so you can have more chips - indeed, the buffer is there to handle the large electrical loads that are caused by having large amounts of memory. In the trade, Major on Third chips are those from a major manufacturer used on third party motherboards. Major on Major means they are used on their own.

SLDRAM uses an even higher bus speed and a packet system. However, with a CPU running at 4 or 5 times the memory speed, even SDRAM is finding it hard to keep up, although **DDR** (*Double Data Rate*) SDRAM doubles the memory speed by using the rising and falling edges of the clock pulse, and has less latency than RAMBUS, giving it a slight edge, if only in cost. It also uses a lower voltage and 184-pin DIMMs. Because of the timing difficulties, different chipsets treat DDR in different ways, so be careful when changing motherboards - probably about 40% of DIMMs will work for any given board. It is specified as PC1600 or PC2100 (that is, the peak data rate in Mbytes/sec) for 100 MHz and 133 MHz, respectively. That is, DDR SDRAM is named after its bandwidth or *effective speed*.

Common DDR modules are PC1600, PC2100 and PC2700, corresponding to 200, 266 and 333 MHz, respectively, which is why they are also known as DDR200, 266 and 333. DDR400 (PC3200) may or may not be compatible. PC2700 uses a six-layer circuit board to reduce crosstalk. Hitachi have developed a way of replacing the capacitor in DRAM with a transistor attached to the MOSFET, where a 1 or 0 is represented by the presence (or not) of electrons between its insulating layers. This means low power requirements, hence less heat, and speed.

In 2003 Intel launched a new motherboard architecture with dual-channel memory (previous stuff used single-channel), in order not to restrict the processor, where increasing speed alone was not an option. Data is transferred 128 bits at a time through two channels, through a memory controller (part of the chipset), which manages all the transfers.

Most dual channel boards will have four DIMM sockets, two each for channels A and B. You must use identically paired chips in each, of whatever capacity.

Type	Name	True Speed	Effective Speed	Bandwidth
SDRAM	PC 66	66		550
	PC 100	100		800
	PC 133	133		1050
	PC 150	150		1200
	PC 166	166		1350
DDR	PC 1600/DDR 200	100	200	1600
	PC 2100/DDR 266	133	266	2100
	PC 2700/DDR 333	166	333	2700

Type	Name	True Speed	Effective Speed	Bandwidth
	PC 3200/DDR 400	200	400	3200
	PC 3500/DDR 433	217	433	3500
	PC 3700/DDR 466	233	466	3700
	PC 4000/DDR 500	250	500	4000
	PC 4200/DDR 533	266	533	4200
	PC 4400/DDR 550	275	550	4400
	PC 4500/DDR 566	283	566	4500
	PC 4800/DDR 600	300	600	4800

SHADOW RAM

ROMs are used by components that need their own instructions to work properly, such as a video card or cacheing disk controller; the alternative is loading the instructions from disk every time they are needed. ROMs are 8-bit devices, so only one byte is accessed at a time; also, they typically run between 150-400 ns, so using them will be s-l-o-w relative to 32-bit memory at 60-80 ns, which is also capable of making four accesses at once (your effective hard disk interleave will drop if data is not picked up in time).

Shadow RAM is the process of copying the contents of a ROM directly into extended memory which is given the same address as the ROM, from where it will run much faster. The original ROM is then disabled, and the new location write protected. You may need to disable shadow RAM when installing Multiuser DOS. It is also now used by NT (and above) for the HAL.

If your applications execute ROM routines often enough, enabling Shadow RAM will increase performance by around 8 or 9%, assuming a program spends about 10% of its time using ROM instructions, but theoretically as high as 300%. The drawback is that the RAM set aside for shadowing cannot be used for anything else, and you will lose a corresponding amount of extended memory; this is why there is a shortfall in the memory count when you start your machine if shadowing is enabled. The remainder of Upper Memory, though, can usually be remapped to the end of extended memory and used there. However, with Windows, including 3.x, or other operating systems that take over some BIOS functions directly, like NT, it is arguable as to whether any performance increase is actually noticeable, as the old slow routines are not used anyway.

With some VGA cards, with video shadow disabled, you might get DMA errors, because of timing as code is fetched from the VGA BIOS, when the CPU cannot accept DMA requests. Some programs don't make use of the video ROM, preferring to directly address the card's registers, so you could use the extended memory for something else. You may also get better results from increasing the bus clock speed.

If your machine hangs during the startup sequence for no apparent reason, check that you haven't shadowed an area of upper memory containing a ROM that doesn't like it-particularly one on a hard disk controller, or that you haven't got two in the same 128K segment. NetWare doesn't really benefit from Shadow RAM, certainly for the video, and can make better use of the memory.

Flash ROM is now quicker than DRAM, so with a Flash BIOS you may find Shadow RAM is not required.

RANDOM ACCESS MEMORY

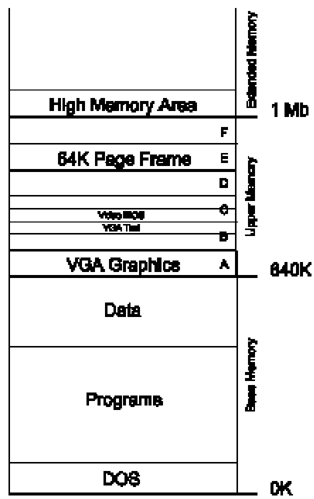
There are 6 types of Random Access Memory a program can use. Be aware that the A+ exam calls the whole of the 1Mb that the 8088 can see Conventional Memory, and what's left in the 1st 640K after DOS, etc has loaded, Base Memory.

Base (or Conventional) Memory

The first 640K, which usually contains DOS (when used), device drivers, TSRs and any programs, plus their data, so the less room DOS takes up, the more there is for the rest.

Different operating systems are better or worse in this respect. Under normal circumstances, you can expect the first 90K or so to consist of:

- An *Interrupt Vector Table*, which is 1K in size, including the name and address of the program providing the interrupt service. Interrupt vectors point to any of 255 routines in the BIOS or DOS that programs can use to perform low level hardware access. The interrupt vector table is an index of them. DOS uses io.sys and msdos.sys for the BIOS and DOS, respectively. This also includes user-defined hard disk data (Type 47). During the POST, the BIOS checks the CMOS for an I/O port, which is assigned a hardware address by the CPU, to which the vector table points when moving instructions back and forth between the device and software.
- *ROM BIOS tables*, which are used by system ROMs to keep track of what's going on. This will include I/O addresses.
- *DOS itself*, including the resident portion of command.com, plus any associated data files it needs (e.g. buffers, etc).



Note: Sometimes, on the A+ exam (depending on which book you read), the whole of the first 1 Mb of memory is referred to as Conventional Memory, and the remainder of the 640K after DOS, etc, has been loaded as Base Memory.

DOS was written to run applications inside the bottom 640K block simply because the designers of the original IBM PC decided to - memory then was expensive, and most CP/M machines only used 64K anyway (the PC with 128K was $10,000!). Other machines of the same era used more; the Sirius allowed 896K for programs.

Contrary to popular belief, Windows (3.x, certainly) uses memory below 1Mb, for administration purposes; although it pools all memory above and below 1 Mb (and calls it the *Global Heap*), certain essential Windows structures must live below 1 Mb, such as the *Task DataBase* (TDB) which is necessary for starting new tasks. Every Windows application needs 512 bytes of memory below 1Mb to load, but some will take much more, even all that's available, thus preventing others from loading, which is one source of "Out Of Memory" messages. There are programs that will purposely fragment base memory so it can't be hogged by any one application.

Rather than starting at 0 and counting upwards, memory addressing on the PC uses a two-step **segment:offset** addressing scheme. The *segment* specifies a 16-byte paragraph, or segment, of RAM; the *offset* identifies a specific byte within it. The reason for using two numbers for an address is that using 16 bits by themselves will only give you 65536 bytes as the longest number you can write.

The CPU finds a particular byte in memory by using two registers. One contains the starting segment value and the other the offset, the maximum that can be stored in each one being 65,536 (FFFF in hex), as we said. The CPU calculates a physical address by taking the contents of the segment register, shifting it one character (bit) to the left, and adding the two together (see High Memory, below). To get a decimal number, multiply the segment by 16 and add the offset to the total.

Sometimes you'll see both values separated by a colon, as with FFFF:000F, meaning the sixteenth byte in memory segment FFFF; this can also be represented as the effective address 0FFFFFh. When referring only to 16-byte paragraph ranges, the offset value is often left out.

The 1024K of DOS memory is divided into 16 parts of 64K each. Conventional memory contains the ten from 0000h to 9FFFh (bytes 0 to 655,167), and Upper memory (below) contains the six ranging from A000h to FFFFh.

Upper Memory

The next 384K is reserved for private use by the computer, so that expansion cards with their own memory or ROMs can operate safely there without interfering with programs in base memory, and *vice versa*. Typical examples include Network Interface Cards.

There is no memory in it - the space is simply reserved. This is why the memory count on older machines with only 1 Mb was 640 + 384K of *extended memory* (see below); the 384K was remapped above 1 Mb so it could be used. When upper memory blocks are needed, as when using **emm386.exe**, that memory is remapped back again, so you lose a bit of extended memory. This area is split into regions, A-F, which in turn are split into areas numbered from 0000 to FFFF hexadecimally (64K each). With the right software, this area can be converted into Upper Memory Blocks for use by TSRs (memory-resident programs) to make more room downstairs. The amount of upper memory available varies between computers, and depends on the amount of space taken up by the System BIOS and whether you have a separate VGA BIOS (on board video sometimes has its BIOS integrated in the system BIOS). It also depends on the number of add-in cards you have, e.g. disk controllers, that normally take up around 16K.

Some chipsets (such as Chips & Technologies) will always reserve this 384K for shadowing, so it will not appear in the initial memory count on power-up, the system configuration screen, or when using **mem** (if you've ever wondered why you're missing 384K, this is the reason). Other chipsets have a *Memory Relocation* option which will re-address it above 1 Mb as extended memory.

Occasionally, some ROM space is not needed once the machine has booted, and you might be able to use it. A good example is the first 32K of the System BIOS, at F000 in ISA machines. It's only used in the initial stages of booting up, that is, before DOS gets to set up device drivers, so this area is often useable.

Note: Many proprietary machines, such as Compaq or NEC, and particularly portables, have different arrangements; VGA ROMs sometimes turn up at E000!

If you have Plug and Play, you will lose another 4K for ESCD (*Extended System Configuration Data*), which is part of the specification and largely a superset of Extended ISA (EISA) that stores information on PnP or non-PnP EISA, ISA or PCI cards, so the operating system can reserve specific configurations, which is its primary purpose, that is, to lock them down for individual PnP adapters. ESCD occupies part of Upper Memory (from E000-EDFF), not available to memory managers. PC Cards, incidentally, like to use 4K at D000.

Extended Memory

Memory above 1 Mb is known as extended memory, and is not normally useable under DOS, except to provide RAM disks or caches, because DOS runs in real mode, and can't access extended memory in protected mode; you need something like OS/2 for that. However, some programs, such as AutoCAD (and Windows!), are able to switch the CPU from one to the other by themselves, and some can use DPMI, the *DOS Protected Mode Interface* (DPMI is a method of allowing programs to run in protected mode, as is VCPI, another system promoted by Phar Lap Software. **win.com** starts a DPMI host, to run the rest of Windows).

The difference between the two:

- VCPI provides an interface between DOS Extenders and Expanded Memory Managers so they can run smoothly together by allowing them both access to extended memory with the same interrupt as that used for expanded memory (see below). It was originally designed for 386 systems and above, and doesn't support multitasking (or windowed DOS displays in Windows), hence.....
- DPMI allows multitasking under similar circumstances as VCPI, but also works on a 286. It was designed by Microsoft, with the object of supporting Windows and controlling DOS software using 32-bit addressing in protected mode on any CPU.

Although extended memory first appeared on the 286, and some software took advantage of it, the 286 was used mostly as a fast XT, because DOS wasn't rewritten (history again). It wasn't until the 386, with its memory paging capability, that extended memory came to be used properly.

High Memory

The first 64K (less 16 bytes) of extended memory, which is useable only by 286, 386 or 486 based computers that have more than 1Mb of memory. It's a quirk in the chip design (or a bug!) that can be exploited by playing with certain I/O addresses to use that portion of extended memory as if it were below 1 Mb, leaving yet more available for programs in base memory. In other words, it is extended memory that can be accessed in real mode. It is activated with **himem.sys** (MS-DOS/ Novell DOS) or **hidos.sys** (DR DOS).

HMA access is possible because of the segment:offset addressing scheme of the PC, which can actually count to just under 64K more than 1 Mb, but the 20 address lines still restrict you. If you remember, memory addresses on a PC are 20 bits long, and are calculated by shifting the contents of a 16-bit register (a paragraph) one character to the left, and adding it to a 16-bit offset. For example, address 1234:5678 is interpreted like this:

```
1234          Segment Register
 5678         Address Register
179B8         20-bit address
```

Shifting 1 to the left is the same as adding a zero to the right, thus multiplying by 16 to get the total byte count (like you do with decimals).

Address references near the last memory address in Upper Memory (FFFF:000F, or the sixteenth byte in segment FFFF) generate a "carry bit" when the 16-bit offset value (0FFFFh) is added to the 20-bit shifted segment value (FFFF0h):

```
FFFF          segment (FFFF0, or 1Mb-16bytes)
FFFF          offset (64K)
10FFEF
```

The 8088, with only 20 address lines, cannot handle the address carry bit (1), so the processor simply wraps around to address 0000:0000 after FFFF:000F; in other words, the upper 4 bits are discarded (the number 1 above).

On a 286 or later, there is a 21st memory address which can be operated by software (see below), which gives you a carry bit. If the system activates this bit while in 8088 (real) mode, the wraparound doesn't happen, and the high memory area becomes available, as the 1 isn't discarded. The reason for the HMA's size restriction is simply that it's impossible to create an address more than 64K above 1 Mb using standard real mode segments and offsets. Remember that segments in real mode become selectors in protected mode and don't follow the same rules; they can address more than 1 Mb.

GATE A20

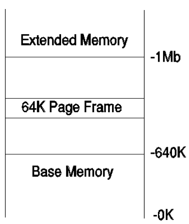
The 8088 in the original PC would wrap around to lowest memory when it got to 1 Mb; the 286 would do it at 16 Mb. On some machines, an AND Gate was installed on CPU address line 20 (the 21st address line) that could switch to allow either wraparound, or access to the 16 Mb address space, so the 286 could properly emulate the 8088 in real mode. A spare pin on the keyboard controller was used to control the gate, either through the BIOS or with software that knew about it. Windows enters and leaves protected mode through the BIOS, so Gate A20 needs to be continually enabled and disabled, at the same time as the command to reset the CPU into the required mode is sent. Programs in the HMA must be well behaved enough to disable the A20 line when they are not in use and enable it when they are. Only one program at a time can control A20, so only one can run in the HMA, which should do so as efficiently as possible. DOS Extenders were one way of using this under DOS until something like OS/2 came along. Many were incorporated into applications, such as Lotus 123, v3 or AutoCAD. They typically intercept interrupts, save the processor state, switch the CPU into real mode, reissue the interrupt, switch back to protected mode, restore the CPU state and resume program execution. All very long-winded.

XMS

As there was originally no operating system to take advantage of extended memory, developers accessed it in their own way, often at the same time. Lotus, Intel and Microsoft, together with AST, came up with an eXtended Memory Specification that allowed real-mode programs to get to extended memory without interfering with each other. The software that provides XMS facilities in DOS is **himem.sys**.

Expanded Memory

This is the most confusing one of all, because it sounds so much like expansion memory, which was what extended memory was sometimes called! Also, it operates totally outside the address space of the CPU. Once the PC was in the market, it wasn't long before 640K wasn't enough, particularly for people using Lotus, the top-selling application of the time, and the reason why many people bought PCs in the first place. They were creating large spreadsheets and not having enough memory to load them, especially when version 2 needed 60K more memory than the original. It wasn't entirely their fault; Lotus itself in its early days was very inefficient in its use of memory.



Users got onto Lotus, Intel and Microsoft for a workaround, and they came up with LIM memory (from the initials), also known as Expanded. It's physical bank-switching, where several extra banks of memory can be allocated to a program, but only one will be in the address space of the CPU at any time, as that bank is switched, or paged, in as required. In other words, the program code stays in the physical cells, but their electronic address is changed, either by software or circuitry.

You added a memory card to your PC that divided its memory into pages of 16K, up to 8 Mb. Four of those (contiguous) 16K pages were allocated space in upper memory, added to base memory and used to access the card. Software was used to map pages back and forward between the card and upper memory. In effect, LIM (4.0) directly swaps the contents of any 16K block of expanded memory with a similar one inside upper memory; in fact, no swapping takes place, but the pages have their address changed to look as if it does; bank switching. Once the page frame is mapped to a page on the card, the data in that page can be seen by the CPU (imagine software using a torch through the page frame, and seeing the memory where the light falls). Points to note about LIM:

- It's normally for data (not program code).
- Programs need to be specially written to use it.

There are two LIM standards, 3.2 and 4, the latter incorporating standards from E(nhanced)EMS, which came from AST. Although, in theory, LIM 4 doesn't need a page frame, the programs you run may well expect to see one. In addition, there could be up to 64 pages, so you could bank switch up to a megabyte at a time, effectively doubling the address space of the CPU, and enabling program code to be run, so you could multitask for the first time (check out desqview). This was called large-frame EMS, but it still used only four pages in upper memory; the idea was to remove most of the memory on the motherboard. The memory card backfilled conventional memory and used the extra pages for banking.

On an 8086 or 286-based machine, expanded memory is usually provided by circuitry on an expansion card, but there are some (not always successful) software solutions. 386 (and 486) -based machines have memory management built in to the CPU, so all that's needed is the relevant software to emulate LIM (**emm386.exe** or similar). At first this idea used the hard disk for the pages (on 286s), but later they were moved to extended memory; the extended memory is made to look like expanded memory to those programs that require it, helped by protected mode and the paging capabilities of the 386 and above.

When manually selecting a page frame, you will need 64K of contiguous upper, or non-banked, memory (that is, it needs to be all together in one place). Various programs (such as msd, which comes with Windows, or DOS 6) will inspect upper memory and tell you how it's being used, and help you place the page frame properly. Try placing it directly next to a ROM, and not in the middle of a clear area, so what's left is as contiguous as possible for other programs. A good place is just under the system ROM, at E000, or above the video ROM, at C800 (its position in the diagram above is for illustration purposes only).

Virtual Memory

"Virtual" in the computer industry is a word meaning that something is other than what it appears to be. In view of that, Virtual Memory isn't memory at all, but hard disk space made to look like it - the opposite of a RAM disk, in fact. Windows uses virtual memory for swap files when physical memory runs out (you can only use virtual memory with 286s and above, because you need protected mode). Like disk cacheing, VM was used on mainframes for some time before migrating to the PC; VMS, the OS used on DEC VAXes, actually stands for Virtual Memory System. There is a speed penalty, of course, as you have to access the hard disk to use it, but Virtual Memory is a good stopgap when you're running short.

Shared Memory

This is where VGA and System memory share the same chips, and needs a BIOS to suit (and a little more RAM!). It comes under the name of *Unified Memory Architecture* (UMA) and uses three buses, two of which share memory address, data and control (CAS, RAS, WE). The third arbitrates between them all. There will be a buffer for the screen display, and you often have to set this in the BIOS. Typically, the graphics controller has to wait its turn behind the CPU, PCI or ISA master. Shared memory lives either at the top of overall system memory or the top of the first bank of DRAM. A scramble table translates between the CPU host address and memory Row and Column address.

Timing is important. The graphics controller must be able to get to as much data as possible in the short time it has access to its memory, often done while the CPU is accessing L2 cache.

The RAM on a video card is called the frame buffer, which holds a complete frame and defines the colour of each pixel. It follows that the greater the frame buffer (or the more memory there is on your card) the greater the resolution and/or colour depth you get.

How much video memory you need depends on what resolution you are trying to run, plus the colour depth and refresh rate. At 60 MHz refresh rates at 800 x 600, the controller is drawing dots on the screen at 40 MHz to keep up. For 256 colours, one byte is needed for each one. With 24-bit colour at 72 KHz, 103, 680, 000 bytes are being written to the screen every second, without you making any changes! 24-bit colour uses 3 bytes per dot, 16-bit 2, and 1 colours only .5. For a particular resolution, multiply horizontal pixels by vertical; 1024 x 768 = 786,43, for example. 256 colours needs 1 byte per dot, so in this case you need 768 K of RAM. 800 x 600 needs 469 K and 300 is needed for 640 x 480.

CMOS MEMORY MAP

In the average 128-byte ISA CMOS, 16 bytes (00-0Fh) concern the RTC, 32 bytes (10-2Fh) ISA configuration data, 16 bytes (30-3Fh) BIOS-specific data and 64 bytes (40h-7Fh) ESCD.

Hex	Dec	Size	Function
00h	0	1 byte	RTC seconds. Contains seconds value of current time
01h	1	1 byte	RTC seconds alarm. Contains seconds value for RTC alarm
02h	2	1 byte	RTC minutes. Contains minutes value of current time
03h	3	1 byte	RTC minutes alarm. Contains minutes value for RTC alarm
04h	4	1 byte	RTC hours. Contains hours value of current time
05h	5	1 byte	RTC hours alarm. Contains hours value for RTC alarm
06h	6	1 byte	RTC day of week. Contains current day of the week
07h	7	1 byte	RTC date day. Contains day value of current date
08h	8	1 byte	RTC date month. Contains month value of current date
09h	9	1 byte	RTC date year. Contains year value of current date
0Ah	10	1 byte	Status Register A Bit 7=Update in progress (0=Date/Time can be read, 1=Time update in progress) Bits 6-4 = Time frequency divider (010 = 32.768KHz Bits 3-0 = Rate selection frequency (0110 = 1.024KHz square wave frequency)
0Bh	11	1 byte	Status Register B Bit 7 = Clock update cycle (0 = Update normally, 1 = Abort update in progress) Bit 6 = Periodic interrupt (0 = Disable interrupt (default), 1 = Enable interrupt) Bit 5 = Alarm interrupt (0 = Disable interrupt (default), 1 = Enable interrupt) Bit 4 = Update ended interrupt (0 = Disable interrupt (default), 1 = Enable interrupt) Bit 3 = Register A square wave frequency (0 = Disable (default), 1 = Enable square wave) Bit 2 = 24 hour clock (0 = 24 hour mode (default), 1 = 12 hour mode) Bit 1 = Daylight savings time (0 = Disable (default), 1 = Enable)

Hex	Dec	Size	Function
0Ch	12	1 byte	Status Register C - Read only flags indicating system status conditions Bit 7 = IRQF flag Bit 6 = PF flag Bit 5 = AF flag Bit 4 UF flag Bits 3-0 = Reserved
0Dh	13	1 byte	Status Register D - Valid CMOS RAM flag on bit 7 (battery condition flag) Bit 7 = Valid CMOS RAM flag (0 = CMOS battery dead, 1 = CMOS battery power good) Bit 6-0 = Reserved
0Eh	14	1 byte	Diagnostic Status Bit 7 = RTC power status (0 = CMOS has not lost power, 1 = CMOS has lost power) Bit 6 = CMOS checksum status (0 = Checksum is good, 1 = Checksum is bad) Bit 5 = POST config info status (0 = information is valid, 1 = information in invalid) Bit 4 = Mem compare during POST (0 = memory equals configuration, 1 = memory not equal) Bit 3 = Fixed disk/adapter initialization (0 = Initialization good, 1 = Initialization bad) Bit 2 = CMOS time status indicator (0 = Time is valid, 1 = Time is invalid) Bit 1-0 = Reserved
0Fh	15	1 byte	CMOS Shutdown Status 00h = Power on or soft reset 01h = Memory size pass 02h = Memory test pass 03h = Memory test fail 04h = POST complete; boot system 05h = JMP double word pointer with EOI 06h = Protected mode tests pass 07h = protected mode tests fail 08h = Memory size fail 09h = Int 15h block move 0Ah = JMP double word pointer without EOI 0Bh = Used by 80386
10h	16	1 byte	Floppy Disk Drive Types Bits 7-4 = Drive 0 type Bits 3-0 = Drive 1 type 0000 = None 0001 = 360KB 0010 = 1.2MB 0011 = 720KB 0100 = 1.44MB
11h	17	1 byte	System Configuration Settings Bit 7 = Mouse support disable/enable Bit 6 = Memory test above 1MB disable/enable Bit 5 = Memory test tick sound disable/enable Bit 4 = Memory parity error check disable/enable Bit 3 = Setup utility trigger display disable/enable Bit 2 = Hard disk type 47 RAM area (0:300h or upper 1KB of DOS area) Bit 1 = Wait for <F1> if any error message disable/enable Bit 0 = System boot up with Numlock (off/on)

Hex	Dec	Size	Function
12h	18	1 byte	Hard Disk Types Bits 7-4 = Hard disk 0 type Bits 3-0 = Hard disk 1 type 0000 = No drive installed 0001 = Type 1 installed 1110 = Type 14 installed 1111 = Type 16-47 (defined later in 19h)
13h	19	1 byte	Typematic Parameters Bit 7 = typematic rate programming disable/enabled Bit 6-5 = typematic rate delay Bit 4-2 = Typematic rate
14h	20	1 byte	Installed Equipment Bits 7-6 = floppy disks (00 = 1 floppy disk, 01 = 2 floppy disks) Bits 5-4 = Primary display (00=Use adapter BIOS, 01=CGA 40, 10=CGA 80, 11=MDA) Bit 3 = Display adapter installed/not installed Bit 2 = Keyboard installed/not installed Bit 1 = math coprocessor installed/not installed Bit 0 = Always set to 1
15h	21	1 byte	Base Memory Low Order Byte - Least significant byte
16h	22	1 byte	Base Memory High Order Byte - Most significant byte
17h	23	1 byte	Extended Memory Low Order Byte - Least significant byte
18h	24	1 byte	Extended Memory High Order Byte - Most significant byte
19h	25	1 byte	HD 0 Extended Type - (10h to 2Eh = Type 16 to 46 respectively)
1Ah	26	1 byte	HD 1 Extended Type - (10h to 2Eh = Type 16 to 46 respectively)
1Bh	27	1 byte	User Defined Drive C: - Number of cylinders least significant byte
1Ch	28	1 byte	User Defined Drive C: - Number of cylinders most significant byte
1Dh	29	1 byte	User Defined Drive C: - Number of heads
1Eh	30	1 byte	User Defined Drive C: - Write precomp cylinder least significant byte
1Fh	31	1 byte	User Defined Drive C: - Write precomp cylinder most significant byte
20h	32	1 byte	User Defined Drive C: - Control byte
21h	33	1 byte	User Defined Drive C: - Landing zone least significant byte
22h	34	1 byte	User Defined Drive C: - Landing zone most significant byte
23h	35	1 byte	User Defined Drive C: - Number of sectors
24h	36	1 byte	User Defined Drive D: - Number of cylinders least significant byte
25h	37	1 byte	User defined Drive D: - Number of cylinders most significant byte
26h	38	1 byte	User Defined Drive D: - Number of heads
27h	39	1 byte	User Defined Drive D: - Write precomp cylinder least significant byte
28h	40	1 byte	User Defined Drive D: - Write precomp cylinder most significant byte
29h	41	1 byte	User Defined Drive D: - Control byte
2Ah	42	1 byte	User Defined Drive D: - Landing zone least significant byte
2Bh	43	1 byte	User Defined Drive D: - Landing zone most significant byte
2Ch	44	1 byte	User Defined Drive D: - Number of sectors
2Dh	45	1 byte	System Operational Flags Bit 7 = Weitek processor present/absent Bit 6 = Floppy drive seek at boot enable/disable Bit 5 = System boot sequence Bit 4 = System boot CPU speed high/low Bit 3 = External cache enable/disable Bit 2 = Internal cache enable/disable Bit 1 = Fast gate A20 operation enable/disable Bit 0 = Turbo switch function enable/disable
2Eh	46	1 byte	CMOS Checksum High Order Byte - Most significant byte
2Fh	47	1 byte	CMOS Checksum Low Order Byte - Least significant byte

Hex	Dec	Size	Function
30h	48	1 byte	Actual Extended Memory Low Order Byte - Least significant byte
31h	49	1 byte	Actual Extended Memory High Order Byte - Most significant byte
32h	50	1 byte	Century Date BCD - Value for century of current date
33h	51	1 byte	POST Information Flags Bit 7 = BIOS length (64KB/128KB) Bit 6-1 = reserved Bit 0 = POST cache test passed/failed
34h	52	1 byte	BIOS and Shadow Option Flags Bit 7 = Boot sector virus protection disabled/enabled Bit 6 = Password checking option disabled/enabled Bit 5 = Adapter ROM shadow C800h (16KB) disabled/enabled Bit 4 = Adapter ROM shadow CC00h (16KB) disabled/enabled Bit 3 = Adapter ROM shadow D000h (16KB) disabled/enabled Bit 2 = Adapter ROM shadow D400h (16KB) disabled/enabled Bit 1 = Adapter ROM shadow D800h (16KB) disabled/enabled Bit 0 = Adapter ROM shadow DC00h (16KB) disabled/enabled
35h	53	1 byte	BIOS and Shadow Option Flags Bit 7 = Adapter ROM shadow E000h (16KB) disabled/enabled Bit 6 = Adapter ROM shadow E400h (16KB) disabled/enabled Bit 5 = Adapter ROM shadow E800h (16KB) disabled/enabled Bit 4 = Adapter ROM shadow EC00h (16KB) disabled/enabled Bit 3 = System ROM shadow F000h (16KB) disabled/enabled Bit 2 = Video ROM shadow C000h (16KB) disabled/enabled Bit 1 = Video ROM shadow C400h (16KB) disabled/enabled Bit 0 = Numeric processor test disabled/enabled
36h	54	1 byte	Chipset Specific Information
37h	55	1 byte	Password Seed and Color Option Bit 7-4 = Password seed (do not change) Bit 3-0 = Setup screen color palette 07h = White on black 70h = Black on white 17h = White on blue 20h = Black on green 30h = Black on turquoise 47h = White on red 57h = White on magenta 60h = Black on brown
38h-3Dh	56-61	6 bytes	Encrypted Password - (do not change)
3Eh	62	1 byte	Extended CMOS Checksum - Most significant byte
3Fh	63	1 byte	Extended CMOS Checksum - Least significant byte
40h	64	1 byte	Model Number Byte
41h	65	1 byte	1st Serial Number Byte
42h	66	1 byte	2nd Serial Number Byte
43h	67	1 byte	3rd Serial Number Byte
44h	68	1 byte	4th Serial Number Byte
45h	69	1 byte	5th Serial Number Byte
46h	70	1 byte	6th Serial Number Byte
47h	71	1 byte	CRC Byte
48h	72	1 byte	Century Byte
49h	73	1 byte	Date Alarm
4Ah	74	1 byte	Extended Control Register 4A
4Bh	75	1 byte	Extended Control register 4B
4Ch	76	1 byte	Reserved
4Dh	77	1 byte	Reserved

Hex	Dec	Size	Function
4Eh	78	1 byte	Real Time Clock - Address 2
4Fh	79	1 byte	Real Time Clock - Address 3
50h	80	1 byte	Extended RAM Address - Least significant byte
51h	81	1 byte	Extended RAM Address - Most significant byte
52h	82	1 byte	Reserved
53h	83	1 byte	Extended RAM Data Port
54h	84	1 byte	Reserved
55h	85	1 byte	Reserved
56h	86	1 byte	Reserved
57h	87	1 byte	Reserved
58h	88	1 byte	Reserved
59h	89	1 byte	Reserved
5Ah	90	1 byte	Reserved
5Bh	91	1 byte	Reserved
5Ch	92	1 byte	Reserved
5Dh	93	1 byte	Reserved

NUMBERS ON CHIPS

The speed is indicated by the last number of the ID, typically after a hyphen, like -70, which means 70 nanoseconds. There may or may not be a leading zero.

Numbering on a chip is split into two, although it never looks like that. The first part indicates complexity, and the second the data path size, or how many bits can be read or written at the same time. To find capacity, multiply the first part by the second, divide by 8 and throw away the remainder:

- 4 banks of 256, meaning 1 Mb
- 1 bank of 1 Mb, meaning 1 Mb
- 4 banks of 1 Mb, meaning 4 Mb

You might see a date looking like this:

```
8609 = 9th month of 86
```

Numbers on FPM chips tend to end with 00, whereas EDO chips are more variable. You may also see a letter, such as A or B, etc., which is how chips are graded for performance (not always). Yet other letters may signify the packaging, like S for SOJ, the most common type.

SIMMs

SIMM stands for *Snap-In Memory Module* (or *Single In-line*). It is a small circuit board a few inches long on which are soldered some memory chips, vertically or horizontally. A 256K chip on a SIMM has connections on all sides. If there are nine on each side, it is parity memory. Nine of these on a SIMM makes a 256 K SIMM with parity. A 1 Mb chip has 10 on each side, in two groups of 5, or 13 on each side. A 4 Mb chip is mostly about 20% wider than a 1 Mb, also with 10 leads in two groups of 5, or 14 on each side. The latter is slightly taller.

SIMMs can identified with chip ID (see above) and placement, e.g. whether horizontal, vertical, on both sides, etc., and resistors, which are often used to tie the presence detect pins, 67-70, to ground. If you really want to show off, you can ID 72-pin SIMMs by checking the resistance of those pins against 72, which is ground (if the notch is on the left, 72 is the one on the far right). For example, this table refers to IBM products:

70	69	68	67	Size speed and part no
I	I	I	I	Not valid
I	I	I	C	1 Mb 120 ns
I	I	C	I	2 Mb 120 ns
I	I	C	C	2 Mb 70 ns 92F0102
I	C	I	I	8 Mb 70 ns 64F3606
I	C	I	C	Reserved
I	C	C	I	2 Mb 80 ns 92F0103
I	C	C	C	8 Mb 80 ns 64F 3607
C	I	I	I	Reserved
C	I	I	C	1 Mb 85 ns 90X8624
C	I	C	I	2 Mb 85 ns 92F0104
C	I	C	C	4 Mb 70 ns 9F0105
C	C	I	I	4 Mb 85 ns 79F1002
C	C	I	C	1 Mb 100 ns 8 Mb 80 ns 79F1004
C	C	C	I	2 Mb 100 ns
C	C	C	C	4 Mb 80 ns 92F33372 Mb 85 ns 79F1003

30 PIN

There are two types, so-called *3-chip* or *9-chip*. You could include 2-chip or 8-chip if you ignore the parity bit. In theory, software can't tell the difference, but Windows has been known to work better on the 9-chip variety; there are cost and refresh timing differences between the two, and some motherboards work with one but not the other. It's probably also due to the chips on a 9-chip being identical, and those on a 3-chip having a diferent parity chip.

72 PIN

These come as a longer circuit board with fine edge connectors and a notch in between. Some manufacturers, such as IBM, move the notch so the SIMM will only fit into one machine, or rather that their machine will only take one type of SIMM (guess whose?). They are 32 bits wide (or 36 with parity). The 4 extra bits in a 36-bit SIMM can be used for ECC instead, where single-bit errors will be corrected and not halt the machine, unlike parity which will merely report the error and halt it. Multiple-bit errors are reported with a halt.

SIMMs have address lines and a select line-a chip will respond when its select line is active. Motherboards that only accept single-sided SIMMs have only one select line, so will not read the two lines on a double-sided SIMM.

1Mb, 4Mb, 16Mb and 64 Mb SIMMs are generally single-sided, and 2Mb, 8Mb, and 32Mb SIMMs double-sided. They all load the chipset equally, as they use 4 x chips, except for one version of the 64 Mb, which uses 4 x 16 Mb ones, although the others are becoming available. It is not recommended to use the conventional 16 Mb SIMM (4 x 16) with the Triton II, and only use 2 SIMMs maximum with the Natoma. Note that electrically single-sided SIMMs may look double sided; they just have chips on both sides. Motherboards use these in different ways; some may treat a double-sided SIMM as two singles, and some may take two double sided or four single sided. You can't use a double sided as a 64-bit chip in a Pentium based machine; they can still only be accessed 32 bits at a time.

There are two types of 36-bit SIMMs; those with logic parity, and those with true parity. A logic parity chip is programmed to answer yes if the computer checks for parity. If you use one in a machine that does more than just query for parity, it will complain loudly (e.g. Gateways), as it adds extra loading to the memory bus and the parity bit is computed later, so it also runs slower. Non-parity chips can be used in machines that either don't use parity (Macs) or allow you to turn off parity checking in the BIOS.

DIMMs

These are 64/72 bit modules, so you only need one for Pentiums. They use one set of contacts and chips for each side of the circuit board, have 168 pins and run at 3.3 and 5.0V, depending on the machine. They are 5¼ inches wide and range from 1-1½ in height. The notch on the non-buffered type is further in towards the centre. If pins 79 and 163 have traces, the DIMM is probably 4-clock. Otherwise, it is 2-clock.

Buffering refers to extra (non-memory) chips that store details about the DIMM that the BIOS can read, if the system supports it. SO DIMMs tend to be unbuffered and non-parity.

The terms CL2 and CL3 refer to the speed that memory can be refreshed at. CL2 means that data will be shoved out on the second rising clock edge after a read, CL3 means it happens on the third rising clock edge. The lower the figure, the better, but the chips are slightly more expensive. Don't forget to tell the BIOS what chips you are using so full advantage can be taken of them, but be aware that some of the settings have no effect anyway, due to the plumbing arrangements (only chips with corresponding pipeline stages can be programmed).

SIPPs

Single Inline Pin-leaded Package. Used by many older machines. They look like SIMMs, but they have pins instead of edge connectors (these can be unsoldered if you're desperate for SIMMS!).

NOTES

4

BUS TYPES

A bus is a shared connection between devices, of which the PC has several; for example, the Frontside Bus connects the CPU to its support chips, the memory bus connects it to memory, and the expansion bus (e.g. PCI) is an extension of the Central Processor, so when adding cards to it, you are extending the capabilities of the CPU itself. Each bus is made up in turn of an address bus and a data bus; the latter transfers data to a memory address located by the former; they are not necessarily the same size, but often are. CPU signals on them have an A or a D before the number, like A31, or D31, for Address and Data, respectively.

The I/O bus is what concerns us here, and the relevance of it with regard to the BIOS is that older cards are less able to cope with modern buses running at higher speeds than the original design of 8 or so MHz for the ISA bus, when fitted. Also, when the bus is accessed, the whole computer slows down to the bus speed, so it's often worth altering the speed of the bus or the wait states between it and the CPU to speed things up.

Note: The DMA clock is coupled to the bus clock, and can be damaged if run too fast. If you have problems with floppies, look here for a possible cause.

ISA

The 8-bit version came on the original PC, and the AT used an extension to make it 16-bit, so there is backwards compatibility - some people call the latter version the AT Bus to make the distinction. It has a maximum data transfer rate of about 8 megabits per second on an AT, which is actually well above the capability of its disk drives, or most network and video cards. The average data throughput is around a quarter of that, but you can increase performance a little(between 1-3&) by disabling the refresh line, if your BIOS allows (C&T or Opti), which is a line used by cards that have memory - you don't need it if you don't have them.

Its design makes it difficult to mix 8- and 16-bit RAM or ROM in the same 128K block of upper memory; an 8-bit VGA card could force all other cards in the same (C000-DFFF) range to use 8 bits as well, which was a common source of odd crashes where 16-bit network cards were involved.

Data movement between the ISA bus and memory is done 16 bits at a time with a block I/O instruction, which, even on a 486, involves a slow microcode loop, so the CPU will not use the bus at its maximum rate. With bus mastering, the controller itself takes over the bus, and blocks can be transferred 32 bits at a time, if the BIOS can cope (see IDE 32-bit Transfer). Bus masters can also transfer data between devices on the bus, rather than just to memory, like the DMA system. ISA only allows one bus master board, but the gains are not brilliant, and you can only access the first 16 Mb of RAM this way.

EISA

Extended Industry Standard Architecture is an evolution of ISA and (theoretically, anyway) backwards compatible with it, including the speed (8.33 MHz), so the increased data throughput is mainly due to the bus doubling in size-but you must use EISA expansion cards. It has its own DMA arrangements, which can use the complete address space, and supports bus masters. Although EISA can handle up to 33 MB/s (PCI can deliver 132), the peak is 20 MB/s (40 for PCI), so for random access applications, there is not a significant difference between them. One advantage of EISA (and Micro Channel) is the relative ease of setting up expansion cards-plug them in and run configuration software which will automatically detect their settings.

MICRO CHANNEL ARCHITECTURE

A proprietary standard from IBM to take over from ISA, and therefore incompatible with anything else. It comes in two versions, 16- and 32-bit and, in practical terms, is capable of transferring around 20 mbps. It runs at 10 MHz, and is technically well designed, supporting bus mastering.

LOCAL BUS

The local bus is one more directly suited to the CPU, being next door with access to the processor bus (hence local) and memory, with the same bandwidth and running at the same speed, so the bottleneck is less (ISA was local in the early days). Data is therefore moved at processor speeds. The original intention was to deal with graphics only, but other functions got added. Faster processing results from the proximity to the CPU and reduced competition between cards on the expansion bus.

There are two varieties, VL-Bus and PCI:

VL-BUS

Otherwise known as VESA Local Bus, this is a 32-bit version more or less tied to the 486 which allows bus mastering, using two cycles to transfer a 32-bit word, peaking at 66 Mb/sec. It also supports burst mode, where a single address cycle precedes four data cycles, meaning that 4 32-bit words can move in only 5 cycles, as opposed to 8, giving 105 Mb/sec at 33 MHz. Up to 33 MHz, write accesses require no wait states, and read accesses require one.

Motherboards will have a switch marked <=33 or >33, which halves the VESA bus speed when switched to > (greater than) 33 MHz. The speed is mainly obtained by allowing VL-Bus cards first choice at intercepting CPU cycles. It's not designed to cope with more than a certain number of cards at particular speeds; e.g. 3 at 33, 2 at 40 and only 1 at 50 MHz, and even that often needs a wait state. VL-Bus 2 is 64-bit, yielding 320 Mb/sec at 50 MHz.

There are two types of slot; *Master* or *Slave*. Master boards, such as SCSI controllers, have their own CPUs which can do their own thing; slaves (i.e. video boards) don't. A slave board will work in a master slot, but not *vice versa*. It is accomplished with an additional slot behind the ISA connector (actually the one now used for PCI, but the other way round). Opti did a similar thing for EISA motherboards. The bus is obsolete, but has resurfaced as AGP.

PCI

A mezzannine bus (meaning divorced from the CPU) with some independence and the ability to cope with more devices, so it's more suited to cross-platform work (it's used on the Mac as well). It is time multiplexed, meaning that address and data (AD) lines share the same connections. It has its own burst mode that allows 1 address cycle to be followed by as many data cycles as system overheads allow. At nearly 1 word per cycle, the potential is 264 Mb/sec. It can operate up to 33 MHz, or 66 MHz with PCI 2.1, and can transfer data at 32 bits per clock cycle so you can get up to 132 Mbyte/sec (264 with 2.1). Being asynchronous, it can run at one speed (33, or 66 MHz) without worrying about coordination with the CPU, but matching them is still a good idea. PCI 2.2 compatibility concerns hardware only - it does not impact the BIOS.

Each PCI card can perform up to 8 functions, and you can have more than one busmastering card on the bus. It should be noted, though, that many functions are not available on PCI cards, but are designed into motherboards instead, which is why PCI multi-I/O cards don't exist. Basic PCI bus transactions are controlled with the following signals:

- **FRAME** - Driven by the master to indicate the beginning and end of a transaction.
- **IRDY** - Driven by the master to force (add) wait states to a cycle.
- **TRDY** - Driven by the target to force wait states.
- **STOP** - Driven by the target to initiate retry cycles or disconnect sequences.
- **C/BE3..0** - These determine, during the address phase, the type of bus transaction with a bus command, and during the data phase, which bytes will be transferred.

PCI is part of the Plug and Play standard, assuming your operating system and BIOS agree, so is auto configuring. There is more in *Plug and Play/PCI*. The PCI chipset handles transactions between cards and the rest of the system, and allows other buses to be bridged to it (typically an ISA bus to allow older cards to be used). Not all of them are equal, though; certain features, such as *byte merging*, may be absent. It has its own internal interrupt system, which can be mapped to IRQs if required. The connector may vary according to the voltage the card uses (3.3 or 5v; some cards can cope with both).

PCI Express is a serial bus, with only four connectors and increased bandwidth, expected to supersede PCI and AGP.

PCMCIA

A 16-bit, 8 MHz *PC Memory Card International Association* standard originally intended (in 1990) for credit-card size flash memory additions to portable computers, as a replacement for floppies, but types 2 and 3 cover modems and hard disks, etc, each getting thicker in turn. The cards are now called PC-Cards, and the current standard is 2.1. Most of version 5's standards have been implemented, but many haven't, so it's still not officially in force. It supports 32-bit bus mastering, multiple voltage (5/3.3) and DMA support, amongst others.

PC Cards usually need 4K in upper memory to initialise, which is not used afterwards. D000-D1FF seems to be popular. An enabler program is often supplied, which is better than using the Card and Socket Service software that is supposed to provide compatibility, but is very cumbersome, consisting of up to 6 device drivers that take up nearly 60K of memory (Windows '95 has it built in). The components of a PC Card system consist of:

- *Host Bus Adapter*. The interface between a bus and the sockets for the cards.
- *Sockets*, type I, II, III and IV, each thicker in turn. A mechanical key stops 3.3v cards being inserted into 5v sockets. Type IV are unofficial Toshiba hard disks.
- *Cards*. These are credit-card size and have 68-pin connectors.
- *Software*:
    - *Socket Services* tell your PC how to talk to its slots or provide an interface between the BIOS and PCMCIA host chips, such as the Intel 82365SL PCIC and the DataBook TCIC-2/N (written for a specific controller). It might configure the socket for an I/O or memory interface and control socket power voltages.
    - *Card Services* tell the operating system or other software how to talk to the card that's in it, or provide an interface between the card and the socket.

There may be a *Resource Initialisation Utility* that checks on I/O ports, IRQs and memory addressing and report to Card Services, as well as software to help Windows (3.x) to recognise cards after it has started, since it assumes a card is not present if it is not seen at start up. A *Card Installation Utility* detects the insertion and removal of PC Cards and automatically determines the card type so the socket can be configured properly. This is where the beeps come from.

The main suppliers of software are *Phoenix*, *Award*, *Databook* and *SystemSoft*. CardSoft comes from the latter, and is a variation for PCI-capable devices, so bus mastering can take place at 33 MHz to cope with 100 Mbps Ethernet, or later versions of SCSI. It uses the same protocol as PCI, and is 32-bit. Client drivers work with the software above, and tend to like their own cards; their purpose is to cover the card's resource requirements, as there are no switches to set IRQs, etc with. *Generic enablers* cover a variety of products. *Point enablers* are specific; they don't need C&SS, but neither do they support hot swapping, and other facilities. Sometimes, you can only run one point enabler at a time.

USB

The *Universal Serial Bus* is a standard replacement for the antiquated connectors on the back of the average PC; computers will likely come with two USB ports as standard, but they can be added with an expansion card. It actually behaves more like a network, since one host (e.g. a PC) can support up to 127 devices, daisychained to each other, or connected in a star topology from a hub, but this depends on the bandwidth you need. Each device can only access up to about 112 Mbps, at varying speeds to stop any one hogging the bandwidth, so Firewire (below), or USB 2.0 are better choices for higher throughput. A hub will have one input connector, from the host or an upstream device, and multiple downstream ones. Otherwise, each device has an upstream and downstream connection.

The maximum distance from one device to another is 5m, and the last device must be terminated. There are three types of device:

- Low power, bus powered (100 mA).
- High power, bus powered (500 mA).
- Self powered, but may use bus power in power save mode.

The bus complies with Plug and Play, so devices are hot-swappable, as they register automatically with the host when connected. More technically, USB is an external 4-wire serial bus with two 90 ohm twisted pairs in a token-based star network. Two lines carry signals based on *Differential Manchester NRZI*, one being for ground, and the other +5v. Zero/half amplitude pulses are used for control. Transmission speed is either 12Mbps with shielded wire or 1.5Mbps for unshielded. Data packets are up to 1023 bits in size, with an 8 bit synch pattern at the start of each frame.

A 1000 msec frame is used, whose usage is allocated by the USB controller based on information provided by devices when logging in, which ensures that they all get bandwidth, and frequently. The controller sends data packets to the USB, from where the targeted device responds. A packet can either contain data or device control signals; the latter go one way only. When the transaction is complete, the next one in the transfer queue is executed. If more than one millisecond is needed, an extra transaction request is placed in the transfer queue for another time frame.

There is backward compatibility with ISA BIOSes. The USB software is too much for an EPROM, so some space in the BIOS is used as well, because access to it is needed anyway (during POST, etc) for USB devices. DOS, OS/2, BeOS, Linux, Win 95 2.1 and 2.5, 98 and 2000 all support USB. NT doesn't, although Iomega have drivers for their equipment. Low end USB chipsets have problems switching device speeds and have signal synchronisation problems. Cheap cables don't help.

USB 2.0 is set to increase the data throughout to about 480 M/bits per second, with an isochronous rate of 24 Mbps. USB 1.1 devices will work in a USB 2.0 socket, but there will be no performance increase. You will need better cables, too.

FIREWIRE

A similar idea to USB, but faster, originally developed by Apple, and now called IEEE 1394, or even HPSB (*High performance Serial Bus*). Sony calls it iLink. It clocks in at a minimum speed of 100 Mbps, going up to somewhere near 400. Because it also guarantees bandwidth, isochronous data, that is, needing consistency to be effective, like digital video, can be transferred properly.

There are two more connections than USB, and it only supports up to 63 devices of varying speeds on the bus. It is also complex and expensive, and could be an alternative to SCSI for hard disks, etc. were it not for USB 2.0.

EXPANSION CARDS

5

Modern motherboards have the basic peripherals built in. The usual suspects are 2 IDE channels, a floppy, 2 serial ports, a parallel port, IR, PS/2 mouse and USB. SCSI, Video and network interfaces only tend to come built in from major manufacturers. Although the connectors are separate, the circuitry is in a *Super I/O chip*, previously found on multi I/O cards. With any luck, you can disable built-in peripherals, but not always, which is useful when you can't upgrade them and want to add something else. Also be aware that IDE channels, particularly secondary ones, may actually be on the ISA bus, as opposed to the PCI.

Expansion cards communicate with the rest of the computer in four ways; Direct Memory Access (DMA), Base Memory Address, I/O address and Interrupt Setting (IRQ).

DIRECT MEMORY ACCESS (DMA)

With this, high speed devices on the expansion bus can place data directly into memory over reserved DMA channels without having to involve the CPU for more than a minimum time, that is, enough for it to write the destination RAM address in the DMA controller, along with the number of bytes to be transferred, so it can get on with something else. Third-party DMA involves the DMA controller as an intermediary between the source and destination, whereas, with First-party DMA, the peripheral doing the transfer does it directly (e.g. bus-mastering).

The DMA controller chip will be programmed by whatever software you're running, and is prone to burning out if run too fast (it's linked to bus speed, adjusted through your *Advanced Chipset Setup*). Typically, a hard drive controller might notify the DMA controller (over its request line) that it wants to move data to memory, whereupon the DMA controller will allocate a priority for that request according to its inbuilt logic and pass it on to the CPU.

If the CPU accepts the request, the DMA controller is given control of the bus (the ALE, or *Address Latch Enable* signal helps here) so it can send a start signal to the hard disk controller.

The DMA Controller (8237A or equivalent) activates two lines at once; one to read and one to write. As the write line is open, data, when read, is moved directly to its destination. When DMA transfers are under way, the CPU executes programs, and the DMA Controller moves data, so it's primitive multitasking. DRQ lines, in case you're wondering, are used by the DMA controller to receive requests.

You can transfer one byte per request, or a block. DMA Controllers need to know where the data to be moved is, where it has to go, and how much there is. PCs and XTs use one DMA chip, and the standard setup is:

Channel	**Device**
0	Refresh (System Memory)
1	Available
2	Floppy controller
3	Hard Disk

ATs use 2 8237As to provide 8 channels, 0-7. Channel 4 joins the two controllers, so is unavailable. 0-3 are eight-bit (64K at a time), and 5-7 are 16-bit (128K); the controller for the former is known as DMA 1, and the one for the latter as DMA 2. Floppies use channel 2. Don't count on channel 0, either, as it may be used for memory refresh (there's no harm in trying, though). PS/2s use 5 for hard disk transfers and XTs use 3. Newer machines use an APIC. If two devices try to use a channel at the same time, one or both will not work, though the channel can often be shared if only one uses it. Channels available in AT compatibles are listed below:

Channel	**Device**	**Notes**
0	Memory Refresh	16-bit
1	Available	8-bit
2	Floppy	
3	Available	8-bit
4	DMA controller 1	
5	Available	16-bit
6	Available	16-bit
7	Available	16-bit

DMA transfers must take place within a 64K segment, and in the first 16 Mb, so memory problems can arise when remapping takes place and data is therefore moved around all over the place, particularly in extended memory. This is especially noticeable with ISA systems (you can use more than 16 Mb, provided it's not used or controlled by the operating system).

A program's request for memory access will be redirected by the CPU, but if it's not involved with the transfer (as with DMA), the DMA controller won't know the new location. Memory managers trap the calls so they can be redirected properly; data is redirected to a buffer owned by the memory manager inside the proper address range. Sometimes you can adjust the DMA buffer size (use d= with emm386.exe), but some systems don't use it, particularly Multiuser DOS (because there's no way of using interrupts to see if DMA transfers have finished, so the controller has to be polled, which is one more thing for the CPU to do when serious multitasking is taking place).

When the AT was made, DMA for hard disk transfers was given up in favour of Programmed I/O (PIO), where the CPU oversees the whole job by letting the BIOS tell the controller what it wants through I/O addresses, and letting the controller and CPU talk amongst themselves - that is, a disk (or network) controller places a block of data into a transfer location in low memory, from where it is moved by the CPU to its destination. The reason is that the DMA controller had to run at 4.77 MHz for compatibility reasons and was too slow on later machines, and with DOS/Windows, the CPU has to wait for the transfer to finish anyway, so PIO isn't as performance-draining as it sounds.

Now quicker buses exist, DMA is again used in the shape of *Fast MultiWord DMA*, which transfers multiple sets of data with only one set of overhead commands, for high performance, but PIO (especially with ATA) is still fast enough to give it a run for its money. MultiWord DMA is used in EISA, VLB, and PCI systems, being capable of the very fast transfer rates, utilizing cycle times of 480ns or faster. Once the entire data transfer is complete, the drive issues an interrupt to tell the CPU the data is where it belongs.

The original ATA interface is based on TTL bus interface technology, which in turn uses the old ISA bus protocol, which is asynchronous, where data and command signals are sent along a signal strobe, but are not interconnected. In fact, only one can be sent at a time, meaning a data request must be completed before a command or other type of signal can be sent along the same strobe.

- **ATA-2** was synchronous, giving faster PIO and DMA modes, where the drive controls the strobe and synchronizes the data and command signals with the rising edge of each pulse, which is regarded as a signal separator. Each pulse can carry a data or command signal, so they can be interspersed along the strobe. Increasing the strobe rate increases performance, but also EMI, which can cause data corruption and transfer errors. ATA-2 also introduced ATAPI (ATA Packet Interface), for devices like CD-ROMs that use the ordinary ATA (IDE) port. EIDE (*Enhanced IDE*) is WD's version based on them both, and Fast ATA is Seagate and Quantum's answer, based on ATA-2 only.
- **ATA-4** includes *Ultra ATA* which, in trying to avoid EMI, uses both rising and falling edges of the strobe as signal separators, so twice as much data is transferred at the same strobe rate in the same period. It was designed by Quantum, in association with Intel, to better match the Pentium processor, and to take over from PIO Mode 5, which was abandoned because of electrical noise. While ATA-2 and -3 can burst up to 16.6 Mbytes/sec, Ultra ATA gives up to 33.3 Mbytes/sec. ATA-4 also adds Ultra DMA Mode 2 (33.3 Mbytes/sec) to the previous PIO modes 0-4 and traditional DMA modes 0-2.
- **ATA-5** includes *Ultra ATA/66* which doubles the Ultra ATA burst transfer rate by reducing setup times and increasing the strobe rate, which again increases EMI to a point where a special cable is needed, which adds 40 ground lines between each of the original 40 ground and signal lines, so the connector stays the same, except that pin 34 is knocked out to allow for cable selection of Master and Slave (it's colour coded, too - the blue connector goes to the motherboard, the grey to the slave and the black to the master device on the channel). ATA-5 adds Ultra DMA modes 3 (44.4 Mbytes/sec) and 4 (66.6 Mbytes/sec) to the previous PIO modes 0-4, DMA modes 0-2, and Ultra DMA mode 2. ATA/100 can burst up to 100 Mbps.
- **UDMA 5** is equivalent to ATA/100, and UDMA 6 to ATA/133.

Having said all that, *Bus Master DMA* is available for IDE, which helps with multimedia under a multithreaded operating system. Traditional DMA still uses the CPU, even if only for setting up data transfers in the first place. A Bus Master DMA device can do its own setup and transfer, even between devices on the same bus, leaving the CPU (and the motherboard DMA controller) out of it (it doesn't improve IDE throughput, however).

Many BIOSes support the following DMA transfer modes:

- *Single Transfer Mode*, where only one transfer is made per cycle; the bus is released when the transfer is complete.
- *Block Transfer Mode*, where multiple sequential transfers are generated per cycle. A DMA device using ISA compatible timing should not be programmed for this, as it can lock out other devices (including refresh) if the transfer count is programmed to a large number. Block mode can effectively be used with Type A, B or Burst DMA timing since the channel can be interrupted while other devices use the bus.
- *Demand Transfer Mode*, as above, but used for peripherals with limited buffering capacity, where a group of transfers can be initiated and continued until the buffer is empty. DREQ can then be issued again by the peripheral. A DMA device using ISA compatible timing should not be programmed for this unless it releases the bus periodically to allow other devices to use it. It is possible to lock out other devices (including refresh) if the transfer count is programmed to a large number. Demand mode can effectively be used with Type "A," Type "B," or Burst DMA timing since the channel can be interrupted while other devices use the bus.
- *Cascade Mode* is used to connect more than one DMA controller together, for simple system expansion, through DMA Channel 4. As it is always programmed to cascade mode, it cannot be used for internal operations. Also, a 16 bit ISA bus master must use a DMA channel in Cascade Mode for bus arbitration.

You may come across these types of DMA transfer:

- *Read transfers*, from memory to a peripheral.
- *Write transfers*, from peripherals to memory.
- *Memory-Memory Transfer*. What it says.
- *Verify transfers*. Pseudo transfers, for diagnostics, where memory and I/O control lines remain inactive, so everything happens, except the command signal. Verify transfers can only happen in ISA timing mode.

BASE MEMORY ADDRESS

Cards often contain small amounts of memory as buffers for temporary data storage when the computer is busy. The Base Memory Address indicates the starting point of a range of memory used by any card. Here is what may be used already:

```
A0000-AFFFF    EGA/VGA video memory (buffer)
B0000-B7FFF    Mono video memory (buffers)
B8000-BFFFF    RGB (CGA) and mono video
C0000-C7FFF    EGA/VGA BIOS ROM (EGA to C3FFF)
C8000-CFFFF    XT hard disk BIOS ROM (can vary)
D0000-DFFFF    LIM area (varies)
E0000-EFFFF    Some EISA BIOS/ESCD/32-bit BIOS
F0000-FFFFF    System BIOS-1st page available?
```

What address in Upper Memory to use for your card (that is, the *Lowest Free Address*) initially depends on the video card, e.g.

Video type	LFA
Hercules	C000
EGA	C400
VGA	C800

As an example, the video ROM typically occupies the area C000-C7FF, so the Lowest Free Address for another card is C800. However, C800 is also a good choice for (16K) hard disk controller ROMs in ISA or EISA machines, so if you have a VGA card as well, you wouldn't normally expect to use anything lower than CC00. Using a base address of D0000 as an example, here are the ranges of memory occupied by a ROM or adapter RAM buffer:

ROM size	Range used
8 K	D0000-D1FFF
16 K	D0000-D3FFF
32 K	D0000-D7FFF

BASE I/O ADDRESS

I/O addresses (I/O = Input/Output) act as "mailboxes", where messages or data can be passed between programs and components, typically responses to IN or OUT instructions from the CPU; they are 1-byte wide openings in memory, also expressed in hexadecimal. On a 386, there are 65,536, mostly never used, because the ISA bus, which only implements 1024 of them, usually only decodes the lower 10 bits, thus using 0-3FF. To get more addresses, some boards, such as 8514/A compatible graphics ones, decode the upper 6 bits as well. When they use 2E8 and 2EA, you will get problems with COM 4, as it uses the former. Watch out for 3C0-3DA as well.

The bottom 256 I/O addresses (000-0FF) relate to the system board, so your cards will only be able to use between 100-3FF. Hybrid motherboards (e.g. with EISA/PCI/VESA as well) will support up to address FFFFFFFF, and the ISA part may get confused if you use a card with an address higher than 3FF.

The Base I/O Address is the first of a *range* of addresses rather than a single one; for example, most network adapters use a range of 20h, so 360h really means 360h-37Fh (in which case watch for LPT 1, whose base is 378) - if you suddenly lose your printer when you plug in a network card, this is the reason. Additionally, COM 1 reserves a range of addresses from 3F8h to 3FFh, which are used for various tasks, like setting up speed, parity, etc. The I/O address table is 00-FFFFh.

You can still get a conflict even when addresses appear to be different, because the cards may think in hexadecimal, when their drivers don't! They may resolve them in binary format, and from right to left (we read hex from left to right). Sound cards suffer from this in particular. Don't forget that most I/O cards only decode the lower 10 address lines, and few use all 16, which is why some video cards get confused with COM 4; as far as the lower 10 address lines are concerned, they're the same!

For example, 220h (standard Sound Blaster) converts to 10 0010 0000 in binary. If you have a card at 2A20, the first 10 digits are the same as 220 (10 1010 0010 0000-right to left, remember), so it won't work. The same goes for:

Hex	Binary
220	10 0010 0000
0A20	1010 0010 0000
0E20	1110 0010 0000
1A20	1 1010 0010 0000
1E20	1 1110 0010 0000
2A20	10 1010 0010 0000
2E20	10 1110 0010 0000
3A20	11 1010 0010 0000

See also *Extended I/O Decode*. The Windows calculator can be used in binary mode to check this. Addresses can vary, especially COM 3 and COM 4, but "standard" ones are used by convention. Here's a list of the usual ones:

```
000-01F        DMA controller 1
020-03F        Interrupt controller 1
040-05F        System timers
060-063        8042 (keyboard Controller)/PPI (XT)
070-07F        Real Time Clock (AT)
080-09F        DMA page registers
0A0-0BF        NMI (in XT to 0AF); PIC 2 (AT & PS/2)
0C0-0DF        DMA controllers (AT & PS/2)
0E0-0EF        Real-time clock (PS/2 30)
0F0-0FF        Maths coprocessor
170-177        2nd IDE/EIDE Controller
1F0-1F8        1st (AT) Hard disk controller
200-20F        Game port
210-217        XT Expansion Unit
220-22F        NetWare Key Card (old)
230-23F        Bus mouse/Soundblaster CD
```

258-25F	Intel Above Board
270-277	LPT3
278-27F	LPT 2
280-28F	LCD display on Wyse 2108 PC
2E0-2EF	GPIB adapter 0
E8-2EF	COM 4
2F8-2FF	COM 2
300-30F	Most cards' default setting/MIDI output
320-32F	Hard disk controller (XT)
330-333	Adaptec 154x
350-	WD 7000 FASST
378-37F	LPT 1
3A0-	MDA
3B0-3BF	Mono display/printer adapter
3BC-3BF	LPT
3C0-3CF	EGA/VGA adapter
3D0-3DF	CGA/EGA/VGA adapter
3E8-3EF	COM 3
3F0-3F5	Floppy drive controller
3F6-3F7	Fised Disk Controller
3F8-3FF	COM 1

INTERRUPT SETTING

If any part of the computer needs attention, it will have to interrupt the CPU, which is more efficient than having the CPU poll each device in turn, and wasting cycles when the device(s) are quite happy to be left alone, thank you very much. On a PC, a hardware interrupt, or IRQ, is a convenient way of calling subroutines from DOS or the BIOS, which are unfortunately also called interrupts! In other words, the BIOS (and DOS) contains code which is allocated an *interrupt number* according to the service provided, which can be used by hardware or software. There are 256, because they must fit into 1024 bytes. Interrupt Vectors are loaded at boot time to create pointers to the appropriate handlers, in a table loaded into base memory, from 0000:0000 to 0000:03FCh. This is so programs can use facilities whose actual address is unknown, and devices can be used regardless of where the software that drives them is located in memory. An interrupt vector is a 4-byte value of the form offset:segment, which represents the address of a routine to be called when the CPU receives an interrupt. Although the vector table is initialized by the start up ROM, changes are made to its contents as ROM extensions and system files are loaded, which allows expansion of operating system services.

Hardware interrupts (described more fully below), or IRQs, are translated into software interrupts, and they should naturally not be called by software. For example, IRQ 1 is used by the keyboard, which is translated to INT 09h. In fact, IRQs 0-7 relate to 08h-0Fh, and 8-15 (on ATs and above) to 70h-77h.

Each IRQ has a different priority, and each device must use a unique one. Classic symptoms of (hardware) interrupt conflicts include colour screens turning black and white, machines hanging up when programs load, and mouse problems. Ther are six types of interrupt:

- *Internal*, called directly by the CPU (00-07h), but also by INTs.
- *External*, generated by hardware other than the CPU, of which there are two variations; NMI (*Non-Maskable Interrupt*), which informs the CPU of catastrophic events, like memory parity errors or power failure, and IRQ, or Interrupt ReQuest, which is used by a device to grab the CPU's attention. IRQs are maskable, which means they can be turned off, or ignored by the CPU. NMIs need immediate attention and cannot be turned off, or worked around. XTs have eight IRQ levels; ATs and PS/2s have two sets of eight. A device will send an Interrupt Request (IRQ) to the 8259 PIC, which allocates priorities and passes interrupts on for translation one at a time, as the CPU only has one interrupt line. Hardware interrupts can be edge triggered, by a sudden change in voltage, or level triggered, by a small change in voltage (which means they can be shared). ISA buses are edge triggered; EISA can be level triggered.
- *Software*, initiated from the BIOS by INT and INTO instructions, and not the same as the above. An example is INT 13, used by Windows 32-bit Disk Access, which is an access point inside the BIOS used for disk-related requests. An operating system will hook into that point and run the code sitting there, rather than run its own; 32-bit disk access, of course, does run its own, hence the speed. These can be shared, otherwise the PC wouldn't run as fast. The clock tick, for instance, at 1Ch, is passed from program to program in turn, known as being *chainable*.
- DOS Interrupts, available when DOS is running.
- ROM Basic Interrupts - available when Basic is running.
- General Interrupts - for use by other programs.

Whereas an interrupt handles asynchronous external events, an *exception* handles instruction faults - software interrupts are treated as exceptions. The lower the IRQ level, the higher the priority the associated device is given, but where a system has a dual interrupt controller (e.g. ATs, PS/2s, 386 and 486 machines) IRQ levels 8 to 15 have priority over levels 3 to 7, because the second controller's single output line is wired to IRQ 2 on the first chip. This makes IRQ 2 more complex to service and should therefore be avoided. If you're using an EISA or Micro Channel machine, you may come across *arbitration levels*, which are similar.

Many cards use IRQ 5 as a default (it's usually used for LPT 2:). As printing isn't interrupt-driven (in DOS, at least), you may be able to use IRQ 7, provided nothing strange is hanging off the parallel port (like a tape streamer). Also, your VGA card may not need IRQ 9, and if you use SCSI you can reclaim IRQs 14 & 15 from the IDE controllers.

Boards with 8-bit edge connectors are limited to IRQ 3-7 or 9 (in ATs) only. With PCI machines, IRQs are allocated to ISA, Plug and Play and PCI cards in that order. The BIOS will automatically allocate an IRQ to a PCI card that requires one, mapping it to a PCI INT#. Leave all PCI INT assignments on A. PCI slot 1 automatically starts with A, 2 starts with B, 3 with C and so on. More in PCI Slot Configuration.

This table shows IRQ lines assigned (in the AT), in order of priority:

0	System timer
1	Keyboard Controller
2	Slave (from IRQ 9 - leave alone!)
8	Real-time clock
9	Redirected to IRQ 2
10	USB or general use
11	SCSI cards, Windows Sound
12	PS/2 Mouse
13	Maths coprocessor
14	Hard disk controller/Primary IDE
15	Secondary IDE
3	COM 2/COM 4
4	COM 1/COM 3
5	LPT 2
6	Floppy controller
7	LPT 1

The Interrupt Vector Table starts at 0000:0000h and ends at 0000:03FCh. A vector is a 4-byte value in offset:segment format, representing the address of a routine to be called when the CPU receives an interrupt. The table is first initialised by the start up ROM but changes are made to its contents as the first ROM Extensions and the operating system files are loaded. Updating the Interrupt Vector Table allows operating system services to be easily expanded.

NOTES

6 PERFORMANCE

Although computers may have basic similarities, that is, they all look the same on the supermarket shelf, performance will differ markedly between them, just the same as it does with cars - it's all too easy to put a big engine in (or a fast processor) and forget to improve the brakes and suspension, so you can't hold the road properly. Aside from that, you will never get a PC set up properly from the shop because there simply isn't enough incentive in terms of time or money for the builders to do so. They will just choose the safe settings to suit the widest variety of circumstances and leave you to it, which is where this book comes in. As an example, the default for some BIOSes is to have both internal and external CPU caches off, which is the slowest option!

The PC contains several processes running at the same time, often at different speeds, so a fair amount of co-ordination is required to ensure that they don't work against each other. Most performance problems arise from bottlenecks between components that are not necessarily the best for a job, but a result of compromise between price and performance. Usually, price wins out and you have to work around the problems this creates. The trick to getting the most out of any machine is to make sure that each part is giving of its best, then eliminate bottlenecks between them. You can get a bottleneck simply by having an old piece of equipment that is not designed to work at modern high speeds (a computer is only as fast as its slowest component), but you might also have badly written software).

Setting up the BIOS for the best performance (or rate of data transfer around the machine, at least) involves quite a bit of tedious trial and error, rebooting your system time and again to check the results. For this reason, you want a quick and easily used diagnostic program (e.g. the Core hard disk performance test, the Quake 1.06 benchmark, Sisoft Sandra's memory benchmark or MadOnion's 3DMark2000, for games) with which to check your hard disk data transfer rate, or whatever. It doesn't really matter about the figures; they will only be used for comparison purposes. In fact, increases in performance will often not be indicated by them, but by your own judgments. www.sisoftware.demon.co.uk/sandra is where you can get Sandra from. Another one for VIA chipsets, from VIA Hardware, is wpcredit.

Performance between motherboards can be affected by the chipset, or who makes the support chips for the CPU; so much so that a 200 MHz Pentium with a slow chipset can be seriously outperformed by a 133 MHz one supported properly (the same applies to later machines). The *Advanced Chipset Setup* helps you to tweak the settings provided if required. You want to concentrate on the following areas:

- *Burst Mode* - used on 486s and above, where a single address cycle precedes four data cycles; 4 32-bit words can move in only 5 cycles, not 8. You need long bursts with low wait states; 1 wait state during a burst loses half the bandwidth.
- *Optimising Memory Cycles* - for example, Concurrent Refresh allows the CPU to read cache memory during a RAM refresh cycle - however, this should be the first to be turned off if you get a problem. You can also control SDRAM Precharge Time,

RAS to CAS Delay and Latency Times (see below - CAS latency is the most important, as page hits occur about 50% of the time, together with cycle length, when it comes to memory. Lower figures almost always mean better performance).

- *Latency*, which affects the PCI bus as well as memory. In other words, how long it may be tied up before being released to either another card or the ISA bus - latency is the delay between a signal being initiated and arriving at its destination. A short latency time means the bus is given up more quickly, which is good for speed but not when you're mastering CDs, where you want long data streams with as few interruptions as possible. Using higher numbers with any form of latency allows you to run faster, but 32-64 seem to be best for most PCs. CAS latency is most the important memory parameter - it is the time allowed between CAS refreshes, and the shorter the better, for performance. SDRAM has a CL rating of 2 or 3, while you might see 2, 2.5 or 3 with DDR. Use Sandra 2002 to check the capabilities of your current chips before you buy new ones, as they must match. If they do, or you only have on chip, using By SPD is best.
- *Interleaving* - allows memory access while refreshing other blocks. Enabling 4 bank interleaving seems to work very well.
- *I/O recovery time* - that is, the timing parameters of your main board and its relation to cards on the ISA bus (use No, Disabled or the lowest settings for best performance!). Preferred to increasing bus speeds.
- *Shadow RAM* - ROM contents are transferred to main memory, which is given the same electronic address as the original ROM, and run much faster. Not much good with NetWare or NT, and possibly '95 & '98, as they use their own drivers.
- *Timings*. Given that you might have four different bus speeds in the machine (that is, 166, 133, 66 or 33 MHz, not to mention the CPU, at more than 20 times that), it is best to ensure that they are all based on the same multiple (usually 33 MHz, luckily) so the clock generator doesn't have to pay them so much attention. There's not much you can do with the PCI or AGP buses, but the FSB and memory buses should be at the same speeds if possible. Note that memory speed has little effect until it becomes a bottleneck. Look for *CPU Host* or *PCI Clock*, but be aware that, if the speed of anything is increased, you might also need to tweak voltages, etc.

Take a note of all the settings in your Advanced Chipset Setup (you can use **Prt Scrn**), and vary them one at a time, taking a note of the test results each time. You will probably find, perversely, that relatively high wait states and low bus speeds will actually result in better performance because the components are better matched. For example, a 60 MHz bus with a 120 MHz Pentium will run with zero wait states, whereas the 100 MHz version may need one. Just remember that the faster you go, the less stability you have, or, in other words, you can have speed or stability, but not both. Changing DMA settings often affects reliability rather than performance. Phoenix recommends that the first place to start if you have a problem is to turn off any Hidden or Concurrent Refresh options.

Operating systems that supply their own 32-bit drivers) will often override some of these settings, especially when it comes to hard disk operation (PIO, Block Mode).

7

OPEN SESAME

The ways of getting into a BIOS are many and varied; if your PC doesn't need a setup disk, you could try any one of the following, in no particular order (of course, whether they work or not often depends on which keyboard driver you have loaded). HP, Dell, Gateway often have a splash screen that hides POST messages, which can be bypassed with the **Esc** or **Tab** keys. Some motherboards have a jumper that disables BIOS access, that should be enabled first.

- Press **del** during boot (AMI, Award).
- Press **Esc** during boot - Toshiba, DTK, Tandon after power on.
- Press **F1** during boot (Toshiba; some Phoenix; Late PS/1 Value Point and 330s, Gateway, HP, IBM Aptiva).
- Press **F2** during boot (NEC, newer Phoenix, ALR, sharp laptops).
- Press **F10** when square in top RH corner of screen (Compaq).
- Press **Ins** during boot-IBM PS/2 with reference partition.
- Press **reset** twice - some Dells.
- **Ctrl Alt Enter** - Dell.
- **Ctrl Alt ?** - some PS/2s, such as 75 and 90.
- **Ctrl-Esc**
- **Ctrl Ins** - some PS/2s when pointer at top right of screen.
- **Ctrl Alt Esc** - AST Advantage, Award, Tandon, older Phoenix, Acer, ALR.
- **Ctrl Alt +**
- **Ctrl Alt S** - older Phoenix.
- **Ctrl Alt Ins** (Zenith, Phoenix, PS/2 after Ctrl Alt Del)
- **Ctrl S** (Phoenix).
- **Ctrl Shift Esc** - Tandon 386.
- **Shift Ctrl Alt** + Num Pad **del** - Olivetti PC Pro.
- Setup disk - Old Compaqs, Epson (Gemini), IBM, IBM PS/2, Toshiba, old 286s.
- **Fn** + **F2**. AST Ascentia 950N
- Both mouse buttons - older IBMs

SETUP PROGRAMS

Compaq

In a partition on the hard disk.

Epson

Try **www.epson.com/connects/ftp.shtml**

GRiD

Originally made laptops, but were bought by Tandy, and later AST, so try **support.tandy.com/grid.htm** or **www.ast.com/americas/files.htm**.

NEC

Try **support.neccsdeast.com/ftp/pmate_2.asp**

Panasonic

Try **www.panasonic.com/host/support**

Samsung

Try **www.sosimple.com/service/bbs.htm**

SOFTMENU SETUP

8

This is a kind of introductory screen for setting up parameters that are usually locked. There may be fields such as Brand Name, Frequency, Cache Size, etc., which just tell you about the CPU and you can't change anyway (there will also be dire warnings about using the wrong settings).

Brand Name

The CPU model name, for example: Intel Pentium (R) 4.

Frequency

Processor Speed.

Cache Size

The L2 cache size of your CPU.

CPU Operating Speed

The speed according to the type and speed of your CPU. You can also select the [User Define] option to enter the manual option.

User Define: Warning

The wrong settings of the multiplier and external clock may cause CPU damage. Setting the working frequecy higher than the PCI chipset or processor specs may cause abnormal memory module functioning, system hangs, hard disk drive data loss and abnormal functioning of add-on cards. Non-specification settings for your CPU should be used for engineering testing, not for normal applications.

There will be no guarantee for the settings beyond specification, any damage of any component on this motherboard or peripherals result therein is not our responsibility.

Ext. Clock (CPU/AGP/PCI)

This selects the external clock frequency.

Multiplier Factor

This item selects the multiplier factors for your CPU if it is not locked.

Estimated new CPU clock

Displays the frequency sum up from the previous items [Ect. Clock] and [Multiplier Factor].

DRAM Ratio (CPU:DRAM)

Determines the frequency ratio between CPU and DRAM.

AGP Ratio (CPU:AGP:PCI)

This item determines the ratio between the CPU, AGP and PCI.

Fixed AGP/PCI Frequency

This item determines the AGP/PCI bus frequency. It allows you to keep your AGP/PCI clock at a frequency that improves system stability.

CPU Power Supply

This option allows you to switch between CPUdefault and user-defined voltages. Leave this setting to default unless the current CPU type and voltage settings cannot be detected or is not correct. The option "User Define" enables you to select the Core Voltage manually.

CPU Core Voltage

Selects the CPU core voltage. Warning: A wrong voltage setting may cause the system to become unstable or even damage the CPU. Please leave it to default settings unless you are fully aware of its consequences.

DDR SDRAM Voltage

Selects the voltage for DRAM slot.

AGP Voltage

Selects the voltage for the AGP slot.

9

STANDARD CMOS SETUP

This deals with the basic information, such as time of day, what disk drives and memory you have, etc. It is mostly self-explanatory, and will be found in every AT-class machine. Memory settings are usually dealt with automatically.

SETTINGS

Date and Time

Speak for themselves, really, except the timekeeping won't be wonderful, due to variations in voltages, etc.

Daylight Saving

American for automatically adding an hour during Summer, at 0200 on the first Sunday in April; the clock chip is hardwired for it and activated by this setting. It resets to Standard Time on the first Sunday in October. Only relevant for North America, and Windows '95 does this by itself anyway.

Hard Disk (C and D)

Several types of hard disk are catered for here (from *Not Installed* to as many as 125). Choose a drive size equal to or lower than the one you propose to fit. User-defined fields are provided for anything strange you may want to fit, in which case you need to specify the following for each drive:

- **Cyln** - number of cylinders.
- **Head** - number of heads on the drive.
- **WPcom** - means *Write Precompensation*. Sectors get smaller towards the centre of the drive, but they still have to hold 512 bytes, so WP circuitry compensates by boosting the write current for sectors on the inner tracks. The setting here is for the cylinder it starts from. Not needed for most modern drives, but some manufacturers (e.g. Conner) specify 0. Be careful with this; what they really mean to say is "disabled", so set 65535 or 1 more than the last cylinder. Setting 0 may mean that WPC actually starts at 0 and confuses the drive.
- **LZone** - the landing zone of the heads, which is where they will go when the system is shut down or they are deliberately parked. Not needed if your drive is autoparking (most are).
- **Sectors per Track** - Usually 17 (MFM) or 26 (RLL), but ESDI, SCSI or IDE may vary.

- Capacity - the formatted capacity of the drive based on the formula below (the calculation is automatically made):

$$\frac{\texttt{Hds x Cyls x Secs/track x 512 bytes (per cyl)}}{\texttt{1048,576}}$$

- **Mode type**. That is, the PIO Mode (0, 1, 2, 3, 4), and only applies to IDE drives. Usually Auto does the trick, and allows you to change drives without entering setup, but if the drive responds incorrectly, you may have to set it manually. This may also be a size selection (with a different CMOS setting for each):
  - **Normal**, through the BIOS, with only one translation step in the drive (so is invisible) and a maximum drive size of 528 Mb, derived from 1024 cylinders, 16 heads and 63 sectors per track (see Large, below, for an explanation). Use if your drive is below 528 Mb, or your OS has a problem with translation.
  - **Large**, using CHS translation for drives over 1024 cylinders, but without LBA (see below). The number of cylinders is divided by 2 and the heads multiplied by 2 automatically, with the calculation reversed by INT 13, so one translation is used between the drive and BIOS, and another between the BIOS and the rest of the machine, but not at the same time, which is the real trick. This is sometimes called Extended CHS, and is often best for performance, if not for compatibility.

CHS stands for *Cylinders, Heads, Sectors-per-track*. As Intel-based PC's use 16-bit registers, processes must use them for compatibility. In case you're interested:

- DX uses 8 bits for head number and 8 for the drive.
- CX uses 10 bits for cylinder number, 6 for the sector.

It's well known that there is a limit to the size of hard drive you can put in a machine. The normal ATA interface only allows up to 528 Mb because of a combination of the field sizes used by INT 13 and ATA (see above), even though ATA by itself can cope with up to 136.9 Gb (see below).

The parameters are limited to the smallest field size:

	INT13	ATA	Limit
Max secs/track	63	255	63
Heads	255	16	16
Cylinders	1024	65536	1024
Max capacity	8.4 Gb	136.9 Gb	528 Mb

With INT 13, the largest 10-bit number you can use (see above) is 1024 (0-1023), which is where the limit on cylinder numbers comes from, and the largest 6-bit number is 63 (1-63), allowing 63 sectors per track, but as the DX register with 8 bits actually allows up to 256 heads (0-255), you can use translation for drives up to 8 Gb and still remain compatible. Although you could use the same logic to support up to 255 drives as well (8 bits for the drive number in DX), the Interrupt Vector Table only has pointers to two I/O addresses (104h and 118h) in the BIOS Data Area, where such data is stored as the machine boots.

In addition, the WD 1003 controller, on which INT 13 is based, only allowed 4 bits for the head number and one for the drive (SCSI bypasses all this by setting the drive type as Not Installed, and including its own ROM on the controller). With translation, you end up with two levels of CHS-one for INT 13H and one for the device. The device CHS stops at 16 heads, hence 528 Mb. The cylinder problem is catered for by clever programming, or translation of parameters, fooling the PC into thinking it has the right apparent size of drive, when it hasn't. A controller will have a Translator ROM on board to do this. When it comes to translation, later Phoenix, AMI, Award and MR BIOSes are based on the Microsoft/IBM specification, which is the standard. Others may use the WD EIDE system, which could mean problems when moving drives between machines.

Operating Systems still have to check the drive types using INT 13 when they start, however much they may bypass them with their own code later, so everything you need to get things running in the first place should be inside the first 1024 cylinders (especially with Linux). Extended INT13 and LBA (below) are solutions to this. In fact, the maximum capacity of your drive may be determined by your operating system; early versions of DOS (2.0-3.2) only supported up to 32 Mb in one volume on a physical drive. With v 3.3, you could have a 32 Mb primary partition and an extended partition, inside which you could put several volumes, up to 32 Mb in size (you can have a maximum of 23, because that's how many letters of the alphabet are left once A, B and C are used up). Although present versions are better, until recently, DOS and/or the BIOS and the IDE interface could still only cope with 1024 cylinders and 528 Mb, as described above, although you can have more than two drives (post DOS 5). DOS (and hence Windows) cannot handle a translated drive geometry with 256 heads. DOS 6.22 is limited to 8.4 Gb, and although Windows can handle more than this, your BIOS may not, due to LBA translation methods (see below) - very few written before 1998 can do so. Drives over 8.4 Gb are supposed to report in with a geometry of 16282 x 16 x 63. There is a workaround for this that uses system memory to keep drive information as well as the normal registers, but this will still limit you to 137.4 Gb. You can't access more than 2.1 Gb with FAT 16 anyway, unless you're using NT, which can format FAT 16 drives up to 4 Gb because it uses 64K clusters.

- **LBA**, where CHS is internally translated into sequentially numbered blocks, a system stolen from SCSI. It allows drives larger than 528 Mb to be used (8.4 Gb), but only in conjunction with CHS and has nothing to do with performance. In fact, it can make things slower, as it only reduces CPU overhead in operating systems that use LBA themselves (more CPU cycles are used). Even then, they must still boot with CHS and not use sectors beyond those allowed by it, so the drive size is the same in either case.

It must be supported by the drive and the BIOS, and the BIOS in turn must support the INT 13 extensions, as must any operating system or application to get the best effect; for example, with Phoenix BIOS 4.03, if LBA is enabled with an appropriate drive, LBA will be used on all accesses to the drive. With 4.05, LBA will only be used if the INT 13 extensions are invoked, which saves an extra translation step by the BIOS.

LBA can therefore be enabled, but not necessarily used. Windows '95 supports INT 13, but LBA calls will only be made if '95 's fdisk has been used and a new partition type (0E or 0F) created. You may lose data if LBA is altered after the drive has been partitioned with it (or not), but it depends on the BIOS. Phoenix is OK in this respect. A Phoenix BIOS converts between the device CHS and INT 13, with LBA in the middle. Others use their own methods, and 32-bit drivers, such as those used in Windows, must be able to cope with all the variations, especially when they have to provide backwards compatibility for older drives, since most people insist on using their previous drive when they add a new one.

As there so many variations, it is possible that LBA mode may be slower with your particular BIOS, in which case use the Large setting instead. Also, be aware that logical block 100 won't necessarily be in the same place on the same drive between different machines.

Large and *LBA* may not be supported by Unix, as it can already handle big drives. Also, if your OS replaces INT 13, the drive may not be accessed properly.

Here some common error codes for Int 13h:

Code	Message
00h	No error
01h	Invalid command
02h	Address mark not found
03h	Disk write protected (floppy)
04h	Request sector not found
05h	Reset failed (hard disk)
06h	Floppy disk removed/Disk changeline (floppy)
07h	Bad parameter table (hard disk)/Initialization failed
08h	DMA overrun (floppy)
09h	DMA crossed 64K boundary
0Ah	Bas sector flag (hard disk)
0Bh	Bad track flag (hard disk)
0Ch	Media type not found (floppy)
0Dh	Invalid number of sectors on format (hard disk)
0Eh	Control data address mark detected (hard disk)
0Fh	DMA arbitration level out of range (hard error - retry failed)
10h	Uncorrectable CRC or ECC data error (hard error - retry failed)
11h	ECC corrected data error (soft error - retried OK) (hard disk)

Code	Message
20h	Controller failure
40h	Seek failure
80h	Disk timout (failed to respond)
AAh	Drive not ready (hard disk)
BBh	Undefined error (hard disk)
CCh	Write fault (hard disk)
E0h	Statur register error (hard disk)
FFh	Sense operation failed (hard disk)

ESDI drives should be set to type 1, and SCSI to 0, or *Not Installed*, but some SCSI controllers, such as the Mylex DCE 376, require drive type 1.

Many new BIOSes can set all the above automatically by fetching the ID string from the (IDE) drive (with Hard Disk Autodetect on the main setup screen), so you would only set them manually if you are using a drive partitioned to something other than the standard. Some PCI boards can use up to four drives (2 each for PCI and ISA). Drive letters will be assigned to primary partitions first, so logical drive names in extended partitions could be all over the place. Some older AMI (pre 4-6-90) and Award BIOSes have compatibility problems with IDE and SCSI drives. AMI BIOSes dated 7-25-94 and later and support translation, as do some versions of Award 4.0G, which implies various versions of the same BIOS! If yours is earlier than 12/13/1994, the address translation table is faulty, so for drives with more than 1024 cylinders, you must use *LBA* rather than *Large*. MR have supported it since early 1990. Only BIOSes conforming to the IBM/Microsoft/Phoenix standards allow access to disks larger than 8GB.

Two devices on the same channel should be configured as *Master* or *Slave* in relation to each other, and a device on its own should be a Master (some CD-ROMs come out of the box as Slaves). The hard drive should be the Master if it coexists with a CD-ROM on the same channel. Note that with a master and slave on the same channel, only one device can be active at the same time - putting an HD and CD-ROM as two masters on two channels will improve performance, but if you set the detection to Auto, bootup will be slower as the BIOS will look for Slaves that aren't there. A slower hard drive will reduce the performance of a faster on on the same channel if they are both accessed at the same time, otherwise there is a negligible effect on performance. 2 master hard drives on different channels will only waste an interrupt and make the CPU work harder to cover them both. The configuration is usually done with jumpers or switches on the device itself but, increasingly, Cable Selection (CS) is used, where both are Masters, and the difference is resolved by the way the cable is made.

It's best not to have EIDE CD-ROMs on IDE channels by themselves, (say, in a SCSI system) as 32-bit addressing may only be turned on with a suitable hard drive as well. 24x CD ROMs cannot reach full speed in 16-bit mode. See also *IDE Translation Mode*.

Primary Master/Primary Slave, etc.

As above, for the primary and secondary EIDE channels.

Floppy Disks

Again, these speak for themselves. 360K drives can be automatically detected, but the BIOS can only tell whether others have 80 tracks or not, so you will get the default of 1.2 Mb. Sometimes you have to put the 360K drive as B: if used with another (on Vanilla PCs). With MR, you can also set the step rate, or track to track speed of the recording heads.

- *Fast* gives you improved performance on modern equipment.
- *Slow* gives you backwards compatibility with anything older.

2.88 Mb drives need an i82077 or NSC8744 controller. The capacity can be used to increase performance of QIC80 or Travan tape drives on the floppy cable. They are known as Extra Density drives. Microsoft has another format storing 1.7 Mb on a floppy, called Distribution Media Format, or DMF. Neither are supported by DOS.

Removable Device (Legacy Floppy)

You won't necessarily just have the choice of a floppy, but also an LS-120, Zip, etc.

Keyboard Installed

Disables keyboard checking and is for fileservers, which don't need keyboards once they're up and running, mainly to discourage people from interfering with them.

Video Display

Mostly autodetects, since all screens except Mono can identify themselves to the system. With two monitors, you can assign the primary one from here.

Halt on

When the computer will stop if an error is detected on startup. Choices are:

Message	Effect
All errors	Every time a non-fatal error is detected
No errors	System will not stop at all.
All but keyboard	System will not stop for a keyboard error.
All but diskette	System will not stop for a disk error.
All but Disk/Key	System will not stop for keyboard or disk errors

Disks and keyboards are excepted because the machine may be a server and not have them anyway. There may be one that stops the system when it runs out of specification.

Floppy 3 Mode Support

This is for the Japanese standard floppy, which gets 1.2 Mb onto a 3.5" diskette. Normally disable, unless you have one installed.

Full Screen Logo Show

When disabled, you see all the POST messages. Otherwise, you can show your company logo if you have programmed it in (the default is the AMI or Award logo).

Full Screen Logo

As above, for the Phoenix BIOS. The default could be the chipset screen

Boot Sequence

Fairly self-explanatory (you can set the sequence), but it's worth noting that some motherboards, like the Abit BE6 or BP6, have an extra onboard IDE controller, which gives you a third or fourth port under the EXT option below, which replaces the usual SCSI option, which has also been moved.

Boot Other Device

Like the above, but this setting wants to know what device to try after the first three choices have been attempted. In other words, any not specifically mentioned in your list will be tried if this is enabled - disabling this will make the system choose only from those specifically mentioned.

Try Other Boot Device

See above - this one is from the AMI BIOS.

Boot Sequence EXT means

EXT means *Extra*. This is only valid if the *Boot Sequence* or *Boot Other Device* functions above have been set to EXT. It allows you to specify booting from an IDE hard disk connected to the third or fourth IDE ports found on some motherboards, or a SCSI hard disk.

First Boot Device

Choose the one you want to boot from first. You can sometimes do this without entering the (AMI) BIOS Setup by pressing F11 on bootup. If you don't have a device in the internal list, it will not show up as a valid choice. BBS stands for BIOS Boot Specification, which is something devices have to comply with to boot from a BIOS. An ARMD device is a removeable device that can function as a floppy or hard drive.

Second Boot Device

See above. Choose the one you want to boot from second.

Third Boot Device

See above. Choose the one you want to boot from third.

IDE Hard Drive

Which drive fitted in the machine will be used to boot from. Press Enter to see the choices.

ATAPI CD ROM

Which one fitted in the machine will be used in the boot sequence. Press Enter to see the choices.

Quick Boot

Enabled allows the system to boot within 5 seconds, but it skips just about everything.

Other Boot Device Select

You might want to boot from SCSI or an INT 18 device (the network) instead of more traditional methods, like the hard drive or floppy.

ADVANCED CMOS SETUP

10

This deals with the basic information, such as time of day, what disk drives and memory you have, etc. It is mostly self-explanatory, and will be found in every AT-class machine, although memory settings are usually dealt with automatically.

SETTINGS

These allow you to tinker more deeply, particularly with the *Password* setting, which is often responsible for locking people out of their own computer.

Typematic Rate Programming

Concerns keyboard sensitivity, or the rate at which keystrokes are repeated, and subsequently the speed of the cursor.

- The *Typematic Rate Delay* is the point at which characters are repeated when the key is continually pressed. Default is usually 250milliseconds, or approx .25 secs.
- The *Typematic Rate* is how many are generated per second (DOS - max 30).

The **alt**, **shift**, **ctrl**, **numlock**, **caps lock** and **scroll lock** keys are excluded.

Above 1 Mb Memory Test

Invokes tests on extended memory, but usually disabled to save time during startup (unless you've got a slow-to-boot hard drive), but the drawback is that only the first 1Mb of memory is tested-the rest is just cleared (himem.sys does it better anyway). Inoperative address lines are also detected.

Memory Parity Error Check

Tests for errors when data is read into memory. If disabled, only the first Mb is checked. If a parity error occurs, you get an error message:

```
Parity Error
System Halted
Have A Nice Day
```

(only joking!) A lot of people find they get many more of these immediately after upgrading from Windows 3.x. They are usually caused by bad memory chips, but they could also be mismatched (in which case change the wait states), or the wrong ones for the motherboard.

Parity is a very basic check of information integrity, where each byte of data actually requires nine bits; the ninth is the parity bit, used for error checking (it was introduced in the early 80s because of doubts about the reliability of memory chips, but the problem was actually found to be emissions from the plastic packaging!). In fact, as cache is used for 80-90% of CPU memory accesses, and DRAM only 1-4% of the time, less errors now result (actually a lower Soft Error Rate), so the need for parity checking is reduced, but '95 uses much more 32-bit code. In Windows 3.x, 32-bit code lives at the low end of physical memory, inside the first 4 Mb, hence the increase in detection of parity errors on upgrading-very likely the memory with a problem has never been exercised properly.

Some memory checking programs use read/write cycles where Windows would use execute cycles, which are more vulnerable to parity errors, so memory would have to be extremely bad for memory checkers to actually find a problem. As it happens, parity is not checked during reads anyway. Other machines, on the other hand, like the Mac, use only 8-bit RAM, and you can use it in motherboards with this option disabled (they are cheaper, after all). The Intel Triton chipset doesn't use parity.

A similar system is ECC (*Error Correction Code*), which corrects memory errors of one bit, for which you need DIMMs with an extra 8 bits of bandwidth (they have an x72 designation, as opposed to x64). It works with the memory controller to add bits to each bit sent to memory which are decoded to ensure that data is valid, and used to duplicate information should it be necessary. Multi-bit errors are detected but not corrected. Unlike parity, there is only a penalty cycle when a 1-bit error is detected, so there is no performance hit during normal operations.

Memory Priming

Found with the MR BIOS and similar to the above. The Full Test works at 1 Mb per second, and Quick Scan at 8, but the latter only primes memory by writing zeros to it. Skip Test means what it says.

Memory Test Tick Sound

Enable if you want to hear memory being tested.

Hit <Del> Message Display

Suppresses the instruction to hit **Del** to enter the setup routine during startup. You can still hit Del to get into it, but the message won't be there (helps keep ignoramuses out!).

Hard Disk Type 47 Data Area

Sometimes called an *Extended BIOS RAM Area*, or *Extended Data Segment Area*. Hard disk parameters (for the Standard Setup) are normally kept in the BIOS ROM, but you can also specify your own parameters for those not already catered for. As the ROM can't be changed, these extra Type 47 details are kept in a small area of reserved memory, normally in an unused area of interrupt vector address space in lower system RAM (at 0:300), or a 1Kb area at the top of base memory, using up DOS address space, in which case you go down to 639K. For Multiuser DOS, select :300 to prevent fragmentation of memory in the TPA, or if you find difficulties booting from the hard disk, especially SCSI. On the other hand, some network operating systems may object to :300 (ROM address :300 is *not* the same as I/O address 300!).

This is sometimes ignored if Shadow RAM or *PS/2 Mouse Support* is enabled because the memory it needs is already being used.

Scratch RAM Option

See *Hard Disk Type 47 Data Area*.

Wait For <F1> If Any Error

Stops the computer until the **F1** key is pressed when a non-fatal error is encountered during POST tests. In other words, if disabled, the system does not halt after this message.

System Boot Up <Num Lock>

Allows you to specify in what mode the calculator pad on the keyboard wakes up in. If you have a 102-key keyboard, and therefore have a separate cursor-control pad, you should keep this On (usually the default) to get numbers out of the keypad. With the 84-key version, you have the choice. If set to Off, both sets of arrow keys can be used.

Boot Up NumLock Status

See *System Boot Up <Num Lock>*, above.

Numeric co-processor

Whether you have one present or not (a 486SX doesn't).

Weitek Processor

Used to tell the computer if a Weitek maths co-processor (3167/4167) is present. The Weitek, beloved of scientists, and having 2-3 times the performance of Intel's version, uses memory address space which must be remapped, which is why the computer needs to know about it. Note that the Weitek processor needs to be the same speed as the CPU.

System Boot Up Sequence

Specifies in which order drives are searched for an operating system, assuming you haven't disabled the floppy drive search (above), in which case this setting will have no effect. The fastest (and least virus inducing) method is C:, A:, but if you have the MR BIOS, there may be other choices:

- *Auto Search*, which searches all floppies before defaulting to drive C:, useful if you have a 5.25" boot disk and a 3.5" first drive.
- *Network 1st* lets you use a Boot ROM, whether your C: drive is bootable or not.
- *Screen Prompt*. You can choose from a short menu.

With *Multiboot*, from Phoenix, the BIOS will identify all boot devices and prioritise them according to your choice (v4.0 of the Phoenix BIOS, and later AMI BIOSes will boot from a Zip drive, while Award's Elite BIOS supports CD-ROMs, SCSI, LS-120 and Zip drives). Multiboot is only relevant to Plug and Play, and devices that the BIOS is aware of. Your only adjustment is the boot priority. Only certain systems, such as NT, have bootable CD ROMs.

Boot Up Sequence

See *System Boot Up Sequence*.

Boot Sequence

As for Boot Up Sequence, with a menu (Award Software).

Permit Boot from...

Stops the system seeking a boot sector on A: or C: (MR BIOS), for speed.

Drive C: Assignment

Whether to boot from a primary IDE drive or the first bootable SCSI drive, if you have both.

Floppy Drive Seek At Boot

Allows you to stop the computer checking if floppy drives are available for reading or writing when it starts, saving time on startup and possible wear and tear on the drive heads when they are initialised (the drive is activated, the access light comes on and the head is moved back and forth once). It's also good for security as it stops people booting up with their own disks and giving you viruses, though it apparently doesn't stop the disk being used once the machine has started, or even when it starts if you have it listed as a possible boot source, so you may need to go to the peripherals section to completely disable it.

Boot Up Floppy Seek

See *Floppy Drive Seek At Boot*. This comes from Award, and looks for a 360K drive. Later versions determine whether the drive is 40 or 80 track. As the only drive to have 40 tracks is a 360K, and the BIOS can't tell the capacity of the others anyway (it can only determine track size), disable this in the interests of speed and security, and make the machine use the CMOS settings instead, or if you don't have a 360K drive.

Boot E000 Adapters

Works with *Drive C: Assignment* to allow boot from a ROM at E000 (usually SCSI).

HDD Sequence SCSI/IDE First

Normally the IDE drive would be the boot disk where SCSI is also in a system, but this option allows you to set the SCSI drive as the boot device instead.

Quick Power On Self Test

Skips retesting a second, third or fourth time.

Swap Floppy Drive

Changes floppy assignments, so the 1st and 2nd drives can exchange drive letters (Award BIOS). Useful if your system diskette is the wrong type for your first drive, such as with a combination of 1.4 and 1.2 Mb drives, but few people have the latter these days anyway. It is useful, however, if you want to tidy up the cables in your machine and cut off the first connector on the floppy cable. Because there is no twist in the second connector, the floppy will report itself as the B: drive, and you can change the letter assignment here.

Floppy Disk Access Control

Allows reads from the floppy (*Read Only*), but not writes, for security. *R/W* allows reads and writes.

Legacy Diskette A:

The type of diskette drive used as the first drive.

Legacy Diskette B:

The type of diskette drive used as the second drive.

System Boot Up CPU Speed

Sets the computer's operating speed during the POST, *High* or *Low*. *Low* = ½ speed and should be set for 40 MHz CPUs or if you get problems booting. Bus timing is based on the CPU clock at boot time, and may be set low if your CPU speed is high.

Boot Up System Speed

Similar to the above - *High* selects the default speed, *Low* the speed of the AT bus, for older peripherals. This apparently only affects the machine during startup anyway.

Cold Boot Delay

Gives slow devices more time to get their act together - older IDE drives won't work if they're accessed too early, and newer ones have problems with fast motherboards as well. Many SCSI drives have a problem, too, because they may get a separate spin up signal. Usually disabled by selecting *None*. (MR BIOS). The 0 (zero) setting gives faster booting.

System Warmup Delay

As above, between 0-30 seconds.

Delay IDE Initial (sec)

As above.

External Cache Memory

Sometimes called *Internal Cache Memory* on 386 boards (as 386s don't have internal cache), this refers to the Level 2 static RAM on the motherboard used as a cache between the CPU and main memory, anywhere between 64-256K. Usually, you will want this *Enabled*, or *Present*, but disabling sometimes helps problem ROMs or interface cards to work. Don't enable this if you don't have cache memory, or when you see the

Cache memory bad, do not enable

error message. There are two types of cache, write-back or write-through, and there are cost/performance tradeoffs with each; write-back is a better choice for performance. Talking of management, often you get better performance by using 1 bank of DRAM with only one bank of cache RAM, e.g. 128K with 4 Mb. This seems to provide better balance.

Internal Cache Memory

Refers to the 8K (or 1K for Cyrix) of cache memory on 486 chips. This should be *Enabled* for best performance. Known as *CPU Internal Cache* with Award.

Fast Gate A20 Option

Or *Turbo Switch Function*, determines how Gate A20 accesses memory above 1 Mb, which is usually handled through the keyboard controller chip (8042 or 8742). Now used for legacy software. The 8088 in the original PC would wrap around to lowest memory when it got to 1 Mb, but the problem was that some software addressed low memory by addressing high memory (Wordstar 3.3 would complain loudly if you had too much available!).

For these older programs, an AND Gate was installed on CPU address line 20 that could switch to allow either wraparound to 1 Mb or access to the 16 Mb address space on the 286 by forcing A20 to zero. A convenient TTL signal from a spare pin on the keyboard controller was used to control the gate, either through the BIOS or with software that knew about it.

The keyboard controller is actually a computer in its own right; at least there is a PROM and a microcomputer in it (hence keyboard BIOS), and it had some spare programming space for code that was left out of the 286.

Programs such as Windows and OS/2 enter and leave protected mode through the BIOS, so Gate A20 needs to be continually enabled and disabled, at the same time as another command to reset the CPU into the right mode is sent.

Enabling this gives the best Windows performance, as a faster method of switching is used in place of using the (slower) keyboard controller, using I/O ports, to optimise the sending of the two commands required; the Fast Gate A20 sequence is generated by writing D1h to port 64h, and data 02h to port 60h. The fast CPU warm reset is generated when a port 64h write cycle with data FEh is decoded (see Gate A20 Emulation). Some BIOSes use Port 92. You will notice very little difference if all your programs operate inside conventional memory (that is, under DOS). However, this may cause Multiuser DOS not to boot. If you get keyboard errors, enable this, as the switching is probably going too fast.

One problem can occur with this option in AMI BIOSes dated 2/2/91 and later; it doesn't always work with the DOS 5.00 version of himem.sys. If you get an error message, disable this. If the error persists, there is a physical problem with the Gate A20 logic, part of which is in the keyboard BIOS chip, in which case try changing this chip. Some machines can take up to 20 minutes to boot when this is enabled.

This is nothing to do with the Turbo switch on the front of the computer (see below); the alternative heading could be *Turbo Switching Function*.

Gate A20 Option

See above. Some modern BIOSes suggest leaving this at the Normal setting, as it is provided for compatibility with older 286 software.

Low A20# Select

You can choose whether the Low A20# signal is generated by the chipset or keyboard controller.

Turbo Switch Function

As above, but could also enable or disable the system Turbo Switch; that is, if this is disabled (*No*), computer speed is controlled through setup or the keyboard. On some machines the 486 internal cache is switched on or off as a means of speed control; on others the CPU clock is altered as well. Others extend the refresh duration of DRAM. With power saving systems, you can set the turbo pin to place the system into Suspend mode instead of changing the speed, so the other choice will be *Break Key*. Sometimes known as *Set Turbo Pin Function*.

Gate A20 Emulation

As for *Fast Gate A20 Option*, but you get the choice of *Keyboard Controller* (if disabled) or *Chipset*, which is faster. This is for programs that use BIOS calls or I/O ports 60/64H for A20 operations, where the chipset will intercept those commands and emulate the keyboard controller to allow the generation of the relevant signals (see above). The sequence is to write D1h to port 64h, followed by an I/O write to 60h with 00h. A fast reset is an I/O write to 64h with 1111XXX0b.

Fast means that the A20 gate is controlled by I/O port 92H where programs use BIOS calls. Both means Gate A20 is controlled by the keyboard controller and chipset where programs use I/O port 60/64H.

Gateway A20 Option

See *Gate A20 Emulation*.

Fast Reset Emulation

Enhances the speed of switching into and out of protected mode by delaying certain signals (INIT or CPURST) by a certain time and holding them for 25 CPUCLK. Switching from Protected to Real Mode requires a "reboot" at chip level, and this setting allows the BIOS to re-boot your system without having to re-initialize all of the hardware. In fact, a pulse is used to take the CPU out of protected mode, which is left set on a fast CPU reset, so is detectable by software (in a bootup, a bit is looked for which indicates whether this is a "boot-start" or a return to 8088. If the latter, the contents of the registers are kept). This setting helps solve problems caused by switching in and out of protected mode too fast.

See above and *Fast Reset Latency* (below).

Fast Reset Latency

The time in microseconds for software reset, between real and protected modes. The lower the figure, the better the performance, but this may affect reliability.

Keyboard Emulation

Enabling this allows the chipset to generate the signal normally provided by the keyboard controller, that is, Gate A20 and software reset emulation for an external keyboard controller are enabled. It also enables Fast Reset Emulation, above. See also *Gate A20 Emulation*, above, whose setting should match this one.

KBC Input Clock

The frequency for the keyboard controller input clock.

Keyboard Controller Clock

Either a fixed speed of 7.16 MHz or a fraction of PCICLKI, the timing signal of the PCI bus.

Video ROM Shadow C000, 32K

Allows you to shadow (or electronically move) the contents of the Video ROM at the specified address, e.g. C000, into extended memory for better performance. The extended memory is then given the same address so the code thinks it's where it should be, and then write-protected (if you're programming or debugging you can sometimes set shadowed areas as Read/Write).

ROM instructions are 8-bit, and s-l-o-w-that is, accessed one bit at a time. Shadowing copies the contents of the ROM into 32-bit (or 16-bit on a 286 or 386SX) memory, disables the ROM and makes that memory look as if it's in the original location, so the code is executed faster. However, you will lose a corresponding amount of extended memory. If your video card has 16K of ROM, shadow at C400 only. If it has 32K (most do), you should include C000 as well. If you have more than that, ensure you include C800 or you might get instability when only part of the code is shadowed, or if you upgrade the BIOS on the card.

Windows NT and (presumably) 95/98 derive no benefit from shadowing, so disabling this makes more RAM available. However, if you use a lot of older DOS games, you may well see a difference, though increasing the bus clock speed may be better. On the other hand, today's video cards use Flash ROM, which is faster, and may not need this setting-sometimes, disabling this with such cards can increase graphics performance, because the Video BIOS does not handle acceleration tasks - this is done by the driver, which may well bypass the BIOS anyway. Note that the 3D part of a video card does not require a BIOS, but uses that on the 2D section.

Shadowed ROMs can also be cached in their new locations through the Advanced Chipset Setup, although this is not always adviseable (see below). Some video cards can't be shadowed because they use an EEPROM (or flash ROM) to store configuration data, and you won't be able to change the contents if this is enabled. Never mind! If you've got a large cache this setting may not be needed anyway. C000 cacheing has one drawback, in that it's done in the 486 internal cache, which cannot be write-protected. Whenever a diagnostic test is done, the program sees there is a BIOS present, but has no knowledge of the cacheing, so it will treat the code as being a non-write-protected BIOS, which is regarded as an error condition. If you get failures in this area, disable this option. The same applies to later CPUs, which use the L2 cache for this. It's a waste of cache bandwidth, anyway, since modern OSes don't use the System BIOS, and the video signals require much more than the cache can provide.

Video BIOS Shadow

See *Video ROM Shadow C000, 32K*, above.

Fast Video BIOS

See *Video ROM Shadow C000, 32K*, above. This one is from Dell.

Adapter ROM Shadow C800, 16K

Together with others, this functions in the same way as Video ROM Shadow, above, but refers to 16K blocks of Upper Memory which cover ROMs on adapter cards, such as hard disk controllers. To use this item effectively, you need to know what memory addresses your expansion cards use (but you could enable them all if you don't know). However, some ROMs don't like being shadowed, particularly those on hard disk controllers, so the best you can do is experiment. Using this reduces available extended memory. Windows NT and (presumably) 95/98 derive no benefit from shadowing, and more RAM is available.

System ROM Shadow

Allows the 64K block of upper memory containing the system BIOS (starting at F000) to be shadowed for better performance, but only when using DOS or another single-user operating system. Disable for Linux, Unix, Xenix or similar, as they have their own arrangements. Windows NT ans (presumably) 95/98 do not use the BIOS (except during startup), so there is no benefit from shadowing, and more RAM is available.

Shadowing Address Ranges (xxxxx-xxxxx Shadow)

See System ROM Shadow, above. Be aware, though, that if using an add-on card that uses an area for I/O, shadowing might stop it working if memory R/W requests are not passed to the ISA bus.

C8000-CFFFF Shadow/D0000-DFFFF Shadow

See *System ROM Shadow*.

C8000-CFFFF Shadow/E0000-EFFFF Shadow

See *System ROM Shadow*.

CPU Internal Core Speed

When you select the speed your CPU should be at, the correct host bus speed and bus frequency multiplier will automatically be selected. However, if you choose the Manual setting, as when overclocking, you will also see:

CPU Host Bust Frequency

Whatever you want the bus speed to be.

CPU Core: Bus Freq. Multiple

Whatever you want the CPU multiplier to be

CPU Core Voltage

If you choose the Default setting, it will be set automatically.

CPU Clock Failed Reset

If you enable this, and your system crashes three times because your overclocking is too much, your CPU speed will automatically be reset to twice the bus speed.

Password Checking Option

You can use a password during the computer's startup sequence. Options are:

- *Always*, which means every time the system is started.
- *Setup*, which only protects the BIOS routine from being tampered with, or
- *Disabled* (sometimes not available).

You can still boot from a floppy and alter things with diagnostic programs, though.

The original AMI BIOS did not encrypt the password, so any utility capable of reading the CMOS should be able to edit it. The AMI WinBIOS uses a simple substitution system. Award saves them in an encrypted form in two bytes, which means there are only 65536 different possibilities, as opposed to the 2E+11 you could have with a password 8 characters long (that's if you only use big characters). Thus, at least in Award's case, the same encrypted value can represent many passwords, so you can use a password with the same value as the original.

Normally, you get three attempts, after which the system will have to be rebooted, or you might get locked out completely. The default is usually the manufacturer's initials (try **ami**), or **biostar**, but try also:

Manufacturer	Password
Advanced Integration	Advance
AMI	AMI, BIOS, PASSWORD, HEWITT RAND, AMI?SW, AMI_SW, LKWPETER, A.M.I., CONDO, 589589, aammii, AMI!SW, AMI.KEY, ami.kez, AMI~, ami°, amiami, amidecod, AMIPSWD, amipswd, AMISETUP, bios310, BIOSPASS, CMOSPWD, helgaßs, HEWITT RAND, KILLCMOS
Amptron	Polrty
AST	SnuFG5
Award	01322222, 589589, 589721, 595595, 598598, AKAKALXZ, ALFAROME, aLLy, aLLY, ALLY, aPAf, _award, AWARD_SW, AWARD?SW, AWARD SW, AWARD PW, AWARD?PW, AWKWARD, awkward, BIOSTAR, biostar, biosstar, CONCAT, CONDO, Condo, d8on, djonet, HLT, J64, J256, J262, j332, j322, KDD, ZAAADA, ALLy, Lkwpeter, LKWPETER, PINT, pint, SER, SKY_FOX, SWITCHES?SW, SYXZ, syxz, TTPTHA, ZBAAACA, ZJAAADC or Shift + SYXZ (syxz) for Award (before 19 Dec 96), ?award, 1EAAh, 256256, admin, alfarome, award.sw, award_?, award_ps, AWARD_PW, BIOS, bios*, condo, efmukl, g6PJ, h6BB, HELGA-S, HEWITT RAND, j09F, lkw peter, PASSWORD, setup, Sxyz, SZYX, SWITCHES_SW, t0ch20x, t0ch88, ttptha, TzqF, wodj, ZAAADA, zbaaaca, zjaaadc
BIOSTar	Biostar, Q54arwms
Compaq	Compaq
Concord	last
CTX International	CTX_123
Cybermax	Congress
Daewoo	Daewuu
Daytek	Daytec
DEC	komprie
Dell	Dell
Enox	xo11nE

Manufacturer	Password
Epox	Central
Freetech	Posterie
HP Vectra	hewlpack
IBM	IBM, MBIUO, sertafu, merlin. For the Aptiva, press both mouse buttons during bootup until the first beep. Thinkpads lock the hard drive as well when the supervisor password is set, even in a new laptop. There are jumpers on the motherboard that should be used to reset the system. Try entering the CONFIG or SYSTEM INFORMATICS menu (or others) and pressing Ctrl-D for a backdoor.
Iwill	Iwill
Jetway	spooml
Joss Technology	57gbz6, technolgi
Machspeed	sp99dd
Magic-Pro	prost
Megastar	star
Micron	sldkj754, xyzall
Micronics	dn_04rjc
Miscellaneous	CMOS, cmos, setup, SETUP, Syxz, Wodj. For many old machines, hold down a key, like ins, del or F1, start the computer and release the key when the memory test is complete. You could also try flooding the keyboard buffer to crash the password routine - just wait for the password prompt, then keep pressing esc. Another trick would be to save the BIOS from a similar machine with the same BIOS and flashing the one with the unknown password (some BIOSes allow boot from a floppy without any password)
M Technology	mMmM
Nimble	xdfk9874t3
Packard Bell	bell9
Phoenix	phoenix, PHOENIX, CMOS, BIOS. Also, try removing hard drive cable during boot - this can reset the password.
QDI	QDI
Quantex	teX1, xljlbj
Research	Col2ogro2
Shuttle	Spacve
Siemens Nixdorf	SKY_FOX
SpeedEasy	lesarot1
SuperMicro	ksdjfg934t
Tiny	Tiny
TMC	BIGO
Toshiba	24Banc81, Toshiba, toshy99. Otherwise, hold left shift down during boot to skip the password. A parallel loopback could work: 1-5-10, 2-11, 3-17, 4-12, 6-16, 7-13, 8-14, 9-15 (and 18-25 optional), but it has been known to remove them.
Vextrec Technology	Vextrex
Vobis	Merlin
Zenith	Zenith, 3098z
Zeos	zeosx

If the above don't work, or you forget your own password, you must discharge the CMOS, or at least clear the CMOS data area. One way to do this is simply to wait for five years until the battery discharges (ten if you've got a Dallas clock chip)! You could also remove the CMOS chip or the battery and just hang on for twenty minutes or so, unless there is a capacitor in line, in which case try 24 hours (look for the chips mentioned below, under Clearing Chips). There are password utilities at **www.dewassoc.com/support/bios/bios_password.htm**.

Note: Since 19 Dec 96, Award Software has not used a default password, leaving it for OEMs. Discharging the battery will not clear the OEM password.

Note: When CMOS RAM loses power, a bit is set which indicates this to the BIOS during the POST test. As a result, you will normally get slightly more aggressive default values.

If your battery is soldered in, you could discharge it enough so the CMOS loses power, but make sure it is rechargeable. To discharge it, connect a small resistor (say 39 ohms, or a 6v lantern lamp) across the battery and leave it for about half an hour. Some motherboards use a jumper for discharging the CMOS; it may be marked CMOS DRAIN. Sometimes, you can connect P15 of the keyboard controller (pin 32, usually) to GND and switch the machine on to make the POST run and delete the password after one diagnostic test. Then reboot.

Very much a last resort is to get a multimeter and set it to a low resistance check (i.e. 4 ohms), place one probe on pin 1 of the chip concerned, and draw the other over the others, which will shock out the chip and scramble its brains. This is not for the faint hearted, and only for the desperate-use other methods first! We assume no responsibility for damage!

The minimum standby voltage for the 146818 is 2.7v, but settings can remain even down to around 2.2v. Usually, the clock will stop first, as the oscillator needs a higher voltage. 3v across a CMOS is common with 3.6v nicad & lithium batteries, as the silicon diodes often used in the battery changeover circuit have a voltage drop of 0.6v (3.6v-.6v = 3v). If your CMOS settings get lost when you switch off and the battery is OK, the problem may be in the changeover circuit-the 146818 can be sensitive to small spikes caused by it at power down.

CLEARING CHIPS

The CMOS can mostly be cleared by shorting together appropriate pins with something like a bent paperclip (with the power off!). You could try a debug script if you are able to boot:

```
A:\DEBUG
- o 70 2E
- o 71 FF
- q
```

The CMOS RAM is often incorporated into larger chips:

- **P82C206** (Square). Also has 2 DMA controllers, 2 Interrupt controllers, a Timer, and RTC (Real-Time Clock). It's usually marked CHIPS, because it's made by Chips and Technologies. Clear by shorting together pins 12 and 32 on the bottom edge or pins 74 and 75 on the upper left corner.

- **F82C206** (Rectangular). Usually marked OPTi (the manufacturer). Has 2 DMA Controllers, 2 Interrupt Controllers, Timer, and Real Time Clock. Clear by shorting pins 3 and 26 on the bottom edge (third pin in from left and 5th pin from right).
- **Dallas DS1287**, DS1287A, Benchmarq bp3287MT, bq3287AMT. The DS1287 and DS1287A (and compatible Benchmarq bp3287MT and bq3287AMT chips) have a built-in battery, which should last up to 10 years. Clear the 1287A and 3287AMT chips by shorting pins 12 and 21-you cannot clear the 1287 (and 3287MT), so replace them (with a 1287A!). Although these are 24-pin chips, the Dallas chips may be missing 5, which are unused anyway.
- **Motorola MC146818AP** or compatible. Rectangular 24-pin DIP chip, found on older machines. Compatibles are made by several manufacturers including Hitachi (HD146818AP) and Samsung (KS82C6818A), but the number on the chip should have 6818 in it somewhere. Although pin-compatible with the 1287/1287A, there is no built-in battery, which means it can be cleared by just removing it from the socket, but also short pins 12 and 24.
- **Dallas DS12885S** or Benchmarq bq3258S. Clear by shorting pins 12 and 20, on diagonally opposite corners; lower right and upper left (try also pins 12 and 24).

For reference, the bytes in the CMOS of an AT with an ISA bus are arranged thus:

```
00          Real Time Clock
10-2F       ISA Configuration Data
30-3F       BIOS-specific information
40-7F       Ext CMOS RAM/Advanced Chipset info
```

The AMI password is in 37h-3Fh, where the (encrypted) password is at 38h-3Fh. If byte 0Dh is set to 0, the BIOS will think the battery is dead and treat what's in the CMOS as invalid. One other point, if you have a foreign keyboard (that is, outside the United States)-the computer expects to see a USA keyboard until your keyboard driver is loaded, so DON'T use anything in your password that is not in the USA keyboard!

Security Option

As for *Password Checking Option*, with two choices:

- *System*, where the machine will not boot and access to setup is denied
- *Setup*, where access to setup is denied

This can be disabled by selecting *Supervisor/User Password Setting* and pressing **Enter**.

Supervisor/User Password

Gives two levels of security; *Supervisor* has higher priority. Disable as above.

Network Password Checking

When *Enabled*, you are prompted for a password when connecting to a network. If disabled, password checking is left to the network. Best disabled.

Boot Sector Virus Protection

All it does is warn you when attempts are made to write to your boot sector or partition table, so it can be annoying when you see the error message every few seconds or so. Actually, it's useless for those drives with their own BIOS in the controller (ESDI/SCSI). Disable when installing software. Only for operating systems such as DOS that do not trap INT 13.

CIH Buster Protection

Protects against viruses that try to destroy the BIOS. See above.

Anti-Virus Protection

See above.

Virus Warning

See *Boot Sector Virus Protection* (Award).

ChipAway Virus On Guard

See above. Guards against boot viruses early in the boot cycle, before they load.

Small Logo (EPA) Show

You can turn off the EPA logo during boot up.

Report no FDD for Win 95

Set to *Yes* if using Windows 95/98 without a floppy to release IRQ6 (this is required to pass Windows 95/98's SCT test and get the logo). Also disable the Onboard FDC Controller in the *Integrated Peripherals* screen.

ATA 66/100 IDE Cable MSG

Stops or displays a message about the ATA cable on boot up - it appears with a 40 pin cable.

Turbo Frequency

Boosts your CPU external speed by mildly overclocking it by (2-5%). Used for testing.

11

ADVANCED CHIPSET SETUP

What you can do here depends on what the motherboard manufacturer decides to supply you with when you want to program the chipset registers-it is not information used by the BIOS, but by the chipset. All the BIOS manufacturer has done is provide a screen so you can make your changes, if the motherboard designer allows you to use them. Remember that the items in this area are actually provided for debugging purposes or to provide some level of tolerance for older expansion cards and slow memory chips; you alter the settings to help the machine cope with them. What one motherboard doesn't like is not necessarily wrong on another, so experiment!

There is a program called **amisetup**, written by Robert Muchsel, which interrogates your chipset settings at a very deep level, often allowing you to tweak settings not displayed. The shareware version can be got from **ftp://194.163.64.1/pub/sanisoft/amisetup.zip**. There's another one for other BIOSes, called **ctchip-something**, available from **www.sysdoc.pair.com**, but it doesn't work on all of them. Highly recommended is **TweakBIOS**, which actually programs the chipset and PCI bridges. It is available from **www.miro.pair.com/tweakbios/**.

Otherwise, you may find two or three sets of default settings, for convenience if you don't want to do too much tinkering. Power-On (or Setup) Defaults gives you the optimum (best case) settings for regular use, and BIOS Defaults are more conservative, being minimised for troubleshooting (that is, CPU in slow speed, no cache, etc). High Performance defaults, if you have them, may produce some instability, so only set them if you have a high end system with quality components. You will need to clear the CMOS if you get a problem.

For older AMI BIOSes (pre-1991), you can set the default values by holding down the Ins key and turning on the computer. An XCMOS Checksum Error will be generated. This can be corrected by entering XCMOS Setup, writing CMOS registers and exiting, and rebooting.

For newer versions, enter CMOS Setup and select:

```
LOAD DEFAULT VALUES
```

from the menu.

Note: If your machine hangs after changing anything, hold down the **Ins** key whilst switching the machine on, or the **Esc** key after rebooting - you can then load the default settings of your choice. Unfortunately, this takes you right back to the start, so take notes as you go along!

If you have a green BIOS, you might have *Auto Keyboard Lockout* set, in which case you need to press **Ctrl-Alt-Bksp**. The three keyboard lights will flash on and off and you will be prompted to enter the CMOS password. Instructions for discharging the CMOS are in the *Advanced CMOS Setup* section.

Note: The names of some memory timing fields have been adopted from fast page and EDO and may have nothing to do with current technology.

Automatic Configuration

When this is Enabled, the BIOS sets its own values for some items, such as the Bus Clock Speed, Fast Cache Write Hit, Fast Cache Read Hit, Fast Page Mode DRAM, DRAM Wait State, DMA CAS Timing Delay, Keyboard Clock, etc (the items will vary between motherboards). The important thing to note is that your own settings will be ignored, so disable this one if you want to play, or have to change any of the above settings to accommodate a particular card, such as a Bus Logic BT-445S on a 50 MHz 486 system.

REFRESH

Memory (see below) is addressed by row and column, with two strobe signals, *Row Address Strobe* (RAS) and *Column Address Strobe* (CAS). Normally, when a DRAM controller refreshes DRAM, CAS is asserted before RAS, which needs a CPU cycle for each event (known as *cycle steal*), but some techniques allow a RAS signal to be kept active whilst a series of CAS signals can be sent, or delaying a cycle from the CPU (*cycle stretch*).

The charge in a DRAM cell can go up or down, because it is surrounded by electrically active conductors and other cells, which leak their charges. DRAM refreshes correct for this by reading the charge, deciding on its value (0 or 1) and restoring the bit to a full 0 or 1, if the charge level is above or below a certain threshold. In short, the data is read into the sense amplifiers and moved back into the cells without being output. However, there is even a time limit for the amplifiers.

Most DRAM can maintain an accurate charge for 16-128 milliseconds between refreshes, but data loss can result if it is too slow. Every time an address is read, the whole row is refreshed when the access is completed. As long as the cell hasn't leaked so much that it changes state, it begins from scratch after each refresh. Refreshes are staggered to spread out current surges, and to prevent the stalling of data requests if all rows were done at the same time, as the driver can only supply so much current. The most economical way is to divide the rows (say 4096) into the maximum interval (64 Msec is the JEDEC standard) and refresh alternately:

```
64000 µsec/4096 = 15.6
```

15.6 is normally adequate, but with SDRAM density at 1 Gb per DIMM, more address lines must be served, so the interval must be shortened (chips above 256 Mbit have 8192 rows, so the interval needs to be 7.8 msec). However, as mentioned above, a charge can usually be maintained for longer, so you might find better performance by increasing the refresh interval. In PCs, the DRAM voltage can be nearly 6 volts because of reflections and ringing driving the normal +5 up, which can make the memory run hotter.

A *burst refresh* consists of a series of refresh cycles one after the other until all rows have been accessed. A distributed refresh is most common, occurring every 15.6 ns when DRQ0 is called by the OUT1 timer. The controller allows the current cycle to be completed and holds all the instructions while a refresh is performed. A *RAS Only refresh* occurs when a row address is put on the address line, and RAS is dropped, whereupon that row is refreshed.

CAS-before-RAS (CBR) is for powersaving. CAS is dropped first, then RAS, with one refresh cycle being performed each time RAS falls. The powersaving occurs because an internal counter is used, not an external address, and the address buffers are powered down.

If using a Cyrix chip, you may need to increase the refresh interval or enable Hidden Refresh (below) if your BIOS has no special handling facilities.

Without EMS, cacheing controllers or laser direct printing cards on the expansion bus, disabling refresh for the bus can improve throughput by 1-3%.

SDRAM PH limit

As mentioned above, there is a time limit for a page to be open while data is in the sense amplifiers. Here, you can set the page hit limit (PH-limit), or the number of page hits allowed before the page must be closed to deal with a non-page-hit request, otherwise they could sit there forever waiting to be serviced. Higher numbers give priority to more page hit requests, for better performance. Lower numbers tend to give you interruptions.

SDRAM Page Hit Limit

See above.

SDRAM Idle Limit

Sets the number of idle cycles the SDRAM bank has to wait before recharging. The effect is to force refreshes during idle cycles so that read/write commands can be satisfied quicker. You can force refreshing before anything you may have already set in a Refresh Interval setting (see below), but not delay it.

Using 0 cycles (the default is 8) means that refreshing will take place as soon as no valid requests are sent to the memory controller, which may increase efficiency, but will likely make refreshes happen too often and cause data to stall. Although this looks like Hidden Refresh, as refreshing is done during idle cycles, data requests coming after the bank starts refreshing will have to wait till the bank is completely refreshed and activated before they can be satisfied, although there is less chance of losing it due to inadequate charges.

For best performance, disable this to delay refreshing for as long as possible, unless you have already set a long Refresh Interval and want to boost reliability and make best use of idle cycles for refresh.

SDRAM Idle Cycle Limit

See above.

Hidden Refresh

Normally, a refresh takes up a CPU cycle. When enabled, the DRAM controller seeks the most opportune moment for a refresh, regardless of CPU cycles.

When CAS is low, RAS is made high, then low. Since CAS is low before RAS, you get a CBR refresh. The "hidden" part comes from the fact that data out stays on the line while refresh is being carried out, otherwise this is the same as CBR. If CAS is hidden, you can eliminate a CPU cycle whilst maintaining the cache status if the system starts power saving.

Best system performance is naturally obtained with this enabled, as no HOLD cycles will be asserted to the CPU, but expect to disable it if you are using 4Mb DRAMs (or certain SIMMs), or you get problems. Most of the effects of this are masked if you have a cache.

Hidden Refresh Control

See *Hidden Refresh*.

DRAM Refresh Mode

See *Hidden Refresh*.

AT Style Refresh

This happens when the refresh cycle starts with a process called Hold Arbitration, and proceeds when the CPU releases control of the memory, but since it holds the CPU up is now out of date. Disable.

Concurrent Refresh

If enabled, the CPU can read cache memory during a DRAM refresh cycle or, in other words, the CPU and refresh system have access to memory at the same time. Otherwise it is idle until refresh is complete, which is slower. Enable for Multiuser DOS on an Intel Express.

Decoupled Refresh Option

This is often called *Hidden Refresh*. Normally, motherboard DRAM and that on the data bus is refreshed separately, that is, the CPU sends refresh signals to both system RAM and the ISA bus; the latter takes longer because it's running slower. If enabled, the bus controller will perform arbitration between the CPU, DMA and memory refresh cycles on the bus, carrying them out in the background (i.e. hidden) so as not to hold the CPU up, and the DRAM controller will sort things out between the CPU and motherboard DRAM, thus the ISA bus refresh finishes while the CPU gets on with another instruction.

The problem is that some expansion cards (particularly video) need to have the CPU handle the first bus refresh cycle. Disable this if you get random characters or snowy pictures during high resolution graphics modes (you may need to disable Memory Relocation as well), albeit with the loss of a little performance. This is especially true with S3 801 boards (such as the SPEA V7 Mirage) coupled with Adaptec C cards and Bs fitted with enhanced ROMs for drives greater than 1 Gb.

Burst Refresh

Reduces overheads by performing several refresh cycles during a single Hold sequence.

Refresh When CPU Hold

Causes the CPU to pause whilst refreshing takes place. Slower.

DRAM Burst of 4 Refresh

Allows refreshes to occur in sets of four, at a quarter the frequency of normal, or in bursts occurring at quarter cycles. Enabling increases performance.

Fast DRAM Refresh

Two refresh modes are available here, Normal, and Hidden. CAS takes place before RAS in both but, in the latter, a cycle can be eliminated by hiding CAS refresh, which is faster and more efficient. It also allows the CPU to maintain the cache status even in Suspend mode.

Divide for Refresh

As above, but you will have the choice of 1/1 or 1/4. 1/4 is best for performance.

Hi-speed Refresh

Affects system performance, except with some types of DRAM which cannot support it, in which case disable (especially for a 33MHz CPU). Slow Refresh (below) is preferred, since it gives longer between refresh cycles.

Slow Refresh

Enabled, makes refresh periods happen less often (typically 4 times slower than normal, at 64 rather than 16 ns, which is AT-compatible), so there is less conflict between refreshes and the CPU's activities, thus increasing performance (in other words, there is a longer time between refresh cycles, as modern memory chips can retain their contents better). You might use it if you were getting corruption because your DRAMs aren't fast enough. The timing is measured in microseconds. Slow Refresh also saves power, which is useful on laptops. Not all DRAMs support this, so don't be surprised if you get parity errors! It requires proper DRAMs, and use 125ns if you get the option.

If you want to set a long refresh period, but are worried about stability, you may be able to force refreshes during idle cycles with *SDRAM Cycle Limit*, above, ifyour motherboard has it.

Slow Refresh Enable

See above.

DRAM Slow Refresh

See above. A 16-bit ISA bus master may activate a refresh request when it has bus ownership. This specifies the timing of the master's signal.

Refresh Interval (15.6 μsec)

See above.

Refresh Mode Select

See above.

Staggered Refresh

Where memory banks are refreshed one after the other, this helps to limit the current and help stop interference, or noise, between banks. The RAS of odd banks will go active 1T after even banks.

DRAM Refresh Period

As for *Slow Memory Refresh Divider*, sets the time, in microseconds, between DRAM refresh cycles. The longer the interval, the better the performance because the CPU will not be interrupted as often, assuming your DRAM is capable. If you lose data, knock this figure down a bit. Choices are:

- *15us*. 15 microseconds (default)
- *30us*. 30 microseconds
- *60us*. 60 microseconds
- *120us*. 120 microseconds

Refresh RAS active time

The time needed for the Row Address Strobe when DRAM is being refreshed, in T states. The lower the figure, the better the performance. Choices are:

- *6T*. Six CPU cycles (default).
- *5T*. Five CPU cycles.

Slow Memory Refresh Divider

Normally, in the AT, DRAM is refreshed every 16 ns. A higher setting, say 64 ns, will give best performance. Sometimes 4 sets 60 ns.

Refresh Value

Sets the refresh value for System RAM by programming the refresh timer (many shareware programs do this as well).

Refresh RAS# Assertion

The number of clock ticks for which RAS# is asserted for refresh cycles - the type of refresh clock delay. The lower the better for performance.

DRAM RAS Only Refresh

An older alternative to CBR. Leave disabled unless needed for older DRAMs.

DRAM Refresh Queue

Enabled, permits queueing of DRAM refresh requests so DRAM can refresh at the best time in burst mode, with the last request taking priority. Otherwise, all refreshes take priority as normal. Most DRAMs can support this.

DRAM Refresh Method

Specifies the timing pulse width where the Row Address Strobe (RAS) will be on the falling edge and followed by the Column Address Strobe (CAS). You get the choice of RAS Only or CAS before RAS. A RAS Only refresh occurs when a row address is put on the address line, and RAS is dropped, whereupon that row is refreshed.

CAS-before-RAS (CBR) is for powersaving. CAS is dropped first, then RAS, with one refresh cycle being performed each time RAS falls. The powersaving occurs because an internal counter is used, not an external address, and the address buffers are powered down.

DRAM Refresh Rate

Use 15.6 for SDRAM and EDO/FPM, and 31.2 for EDO/FPM only.

DRAM Refresh Stagger By

The number of clock ticks (0-7) between refreshing rows in the memory array. Zero does all at once.

DRAM Read Burst (EDO/FPM)

The lower the timing for reads from EDO or FPM memory, the faster memory is accessed, at the expense of stability and preservation of data.

Refresh Cycle Time (187.2 us)

The default of 187.2 us is safest against data loss.

PLT Enable

The ALi M1647 memory controller can close all pages if the Page Life-Time counter expires, by relying on bus cycles to determine page expiration. Page Life-Time (or Enhance Page Mode Time) is the equivalent of the AMD 761's Page Hit Limit (PH Limit), which limits the number of consecutive page hits and forces a page to be closed before it expires.

This timer only works after the bus is idle since each read/write command resets the counter, so, as long as consecutive R/W commands are issued, the page stays open until a miss occurs.

DATA BUS

To avoid confusion, a private message is sent along the data bus for 16-bit cards, before data is sent. The high part of the target address is sent out first, so 16-bit cards are alerted as to where instructions are headed. As these are sent out over the extra 4 address lines on the extended bus (20-23), the only information the cards really get is which of the 16 possible megabytes is the destination, so 3 of the original 8-bit lines are duplicated (17-19), narrowing it down to the nearest 128K.

Once a card decides the message is for itself, it places a signal on memcs16, a line on the extended bus, which triggers a 16-bit signal transfer (without the signal, the message is sent as 8-bit). When the CPU sees memcs16, it assumes the current access will be to a 16-bit device, and begins to assemble data so any mismatches are transparent to the CPU and adapter card. The trouble is that there's no specification governing the amount of time between the advance notice and the actual transfer, and some cards don't request 16-bit transfers quickly enough, so it gets its data as 8-bit, hence confusion, and the need for wait states. VGA cards can switch into 8-bit mode automatically, but many others cannot. I/O operations on the bus generally have an extra wait state compared to memory.

AT Cycle Wait State

The number of wait states inserted before an operation is performed on the AT bus, to lengthen the I/O cycle for expansion cards with a tight tolerance on speed, such as high-end graphics cards, or you might be overclocking and the ISA bus is tied to the PCI bus speed and you can't change it. The higher the delay in bus timing, the slower your system will run; 1 wait state can half the bus speed, and you will also need to set a higher DMA wait state. I/O on the bus tends to have an extra wait state as compared to memory operations, which is why memory-mapped cards can work faster.

Extra AT Cycle Wait State

See above. Inserts 1 wait state in the standard AT bus cycle.

16-bit Memory, I/O Wait State

The number of wait states inserted before 16-bit memory and I/O operations. You can often set this to the smallest value, since the device itself will activate the I/O-CHRDY signal, which allows it to extend the bus cycle by itself if required. If the bus is running faster than 8 MHz, 2 is generally safest. Try between 1-2 when running the bus slower.

8-bit Memory, I/O Wait State

If you get bus timing problems, this setting will insert wait states when accessing devices on the bus. You can often set this to the smallest value, since the device itself will activate the I/O-CHRDY signal, allowing it to extend the bus cycle by itself if required. If the bus is running faster than 8 MHz, 1 is generally safest. Try 0 when running the bus slower.

Command Delay

The length of the address phase of 8- or 16-bit bus cycles (data phases are controlled elsewhere), expressed in wait states, typically 0-3.

AT Bus I/O Command Delay

See *AT Bus 16-bit I/O Recovery Time* (below). Refers to a delay before starting an operation.

AT Bus 16 Bit Command Delay

Specifies the length of the address phase of 16 Bit AT Bus Cycles (data phases are controlled elsewhere - see *AT Bus n Bit Wait States*, below). The typical delay will vary from 1-4 cycles (0-3 wait states), but the 82C211 to which this refers defaults to 2 normally and this may be ignored. Leave alone normally.

AT Bus Address Hold Time

See *AT Bus 16-bit Command Delay* (above).

AT Bus n Bit Wait States

Specifies the duration (in wait states) of the data phase of I/O operations on the AT bus (see AT Bus 16 Bit Command Delay, above for address phases). 16 bit values vary between 0-3 wait states and 8 bit values from 2-5, though this may vary. Again, normally, leave this alone.

16-bit I/O Recovery Time

The length of an additional delay inserted after 16-bit operations, for older ISA cards; in other words, the system allows more time for devices to respond before assuming a malfunction and stopping requests for I/O. There is usually an automatic minimum delay of four SYSCLKs between back-to-back I/O cycles to the ISA bus, so these are extra. SYSCLKs are complete machine clock cycles; get best performance with the lowest figure. On PCI systems, bus clock cycles are added between PCI-originated I/O to the ISA bus.

8-bit I/O Recovery Time

As for *16-bit I/O Recovery Time.*

ISA I/O Recovery

As for *16-bit I/O Recovery Time.*

ISA I/O wait state

Adds wait states to the bus so expansion cards can cope with higher speeds better. Normal is compatible with standard AT timing, and wait states are on top of that.

ISA memory wait state

Adds wait states to the bus so memory on expansion cards can cope with higher speeds better. *Normal* is compatible with standard AT timing, and wait states are in addition to that.

ISA write insert w/s

If your ISA card doesn't like write cycles on the bus, extend the timing here.

W/S in 32-bit ISA

Selects the 32-bit ISA cycle wait state. Lower numbers mean better performance.

16 Bit ISA I/O Command WS

The number of wait states between back-to-back input and output to 16-bit ISA devices, which will be slower than the main system - if a device doesn't respond quickly enough, the system may think it has malfunctioned and stop its request for I/O. Increase the delays to allow the devices to catch up.

16 Bit ISA Mem Command WS

The wait states between back-to-back memory reads or writes to memory on 16-bit ISA devices, which will be slower than system memory and may need some allowance.

AT Bus Clock Source

The AT bus clock is an output clock for the I/O channel. This allows you to change the access speed of the (ISA) bus, which should be between 6-8.33 MHz to be compatible with AT specifications (not that any were officially issued), so if your motherboard or PCI bus is running at 33 MHz, divide this by 4 (CLKIN/4, or PCI/4) for memory rated at 70 ns. Choosing Autosync sets this item based on the CPU clock speed. Only valid when Auto Config is disabled. A 16-bit card run too fast may revert to 8-bit mode. Other cards may inject wait states. Values derived from CLKIN are synchronous - the 7.159 MHz option, if you have one, is asynchronous.

The "clock" is responsible for the speed at which numbers are crunched and instructions executed, based on an electrical signal that switches constantly between high and low voltage several million times a second. In fact, there are several clocks, all aligning themselves with a clock chip that generates the appropriate signals and feeds them through a variety of feedback loops and phase shifts to get the right frequencies. Generally, the clock chip is initialized by hardware and doesn't need to be programmed by the BIOS, except on jumperless models that set the CPU speed by software, which makes the clock chip quite important when it comes to upgrading BIOSes, especially when it is the only different component over a range of otherwise identical motherboards - the problem is that they all need to be programmed in different ways.

The *System Clock*, or CLKIN, is the frequency used by the processor; on 286s and 386s, it's half the speed of the main crystal on the motherboard (the CPU divides it by two), which is often called CLK2IN. 486 processors run at the same speed as the main crystal, because they use both edges of the timing signal, which is a square wave. A clock generator chip (82284 or similar) is used to synchronise timing signals around the computer, and the data bus would be run at a slower speed synchronously with the CPU, e.g. CLKIN/4 for an ISA bus with a 33 MHz CPU, giving in the "standard" 8 MHz or so, although it was never properly established.

ATCLK is a separate clock for the bus, when it's run asynchronously, or not derived from CLK2IN. There is also a 14.31818 MHz crystal which was used for all system timing on XTs. Now it's generally used for the colour frequency of the video controller (6845), although some chipsets (i.e. the BX) still use it for timing.

AT Clock

See *AT Bus Clock Source* (above).

AT Bus Clock

The speed of memory access (not ISA bus speed, as above), set to various fractions of PCI clock speed (default PCI/3, or 11MHz, which allows about 90 ns for each one). This comes from the Opti Viper chipset - most others use wait states. In some, this refers to generating the ISA bus clock speed from PCICLK, and setting the AT bus speed in terms of CPU speed or 7.16 MHz.

AT Clock Option

Whether the AT bus clock is synchronised with the CPU clock or is asynchronous. See above.

ATCLK Stretch

Stops the I/O bus clock when there is no activity on the bus. ATCLK is used if the bus is asynchronous.

Synchronous AT Clock

Measured as a fraction of CLK, the CPU timing signal.

ISA Bus Speed

As for *ATCLK Stretch*, but for PCI Pentiums. What speeds you get for the compatible and enhanced selections depends on the CPU speed:

CPU Speed	Compatible	Enhanced
60	7.5	10
66	8.25	16

Bus Clock Selection

As for *ATCLK Stretch*.

Bus Mode

You can set the bus to run synchronously or asynchronously with the CPU. When synchronous, the bus will run at a speed in sympathy with the CPU clock, e.g. 33 MHz=CLKIN/4.

Fast AT Cycle

Similar to *Bus Mode*, affecting wait states. May speed up transfer rates if enabled by shortening AT bus cycles by one ATCLK signal.

ISA IRQ

To let PCI cards know which IRQs are in use by ISA cards so the PnP system doesn't use them.

Master Mode Byte Swap

For bus mastering cards, such as SCSI controllers and fast network cards, affecting transfers from the bus master to 8-bit peripherals; Low, then High and back. Normally disabled.

DMA clock source

DMA controllers allow certain peripherals to access memory directly (hence *Direct Memory Access*). Usually, only the floppy controller uses it, but tape streamers, network cards and SCSI adapters might, amongst others. This setting selects the source for the DMA clock, which runs at ½ the bus clock speed (e.g. ATCLK/2, or SYSCLK/2). Maximum is usually 5 MHz.

DMA Clock

As above - sets DMA speed at equal to or ½ the speed of SYSCLK.

DMA Wait States

Affects the number of wait states inserted before DMA commands are executed. Often separate for 8 and 16-bit transfers (as 8 is used for floppy transfers, adjusting the 16-bit variety doesn't affect them). In general, slower cards may require more wait states. DMA settings often affect reliability rather than performance. For low CPU speeds (<= 25 MHz, this should be 0; otherwise set to 1).

DMA Command Width

You can compress the "normal" DMA transfer cycle of 4 clocks to 3 with this setting.

MEMR# Signal

Concerning DMA transfers, you can set the MEMORY READ control signal to start one clock cycle earlier than normal with this setting. Affects reliability.

MEMW# Signal

As above, but for the MEMORY WRITE signal.

DMA Address/Data Hold Time

"During the DMA/Master cycle, address and data from the X or S-buses are latched and held to local bus-DRAM/CACHE RAM operation". I haven't a clue what that means, but the X-bus is the peripheral bus where the support chips are located (e.g. 82C206 or equivalent), and the S-bus is the expansion bus. Perhaps it means that when DMA mode is operative, data in the local bus, cache or DRAM is held where it is. Latch is techie-speak for "read".

DMA MEMR Assertion Delay

Whether the signal to write to memory is delayed by a cycle from the signal to read the I/O port during DMA operations. This affects reliability and should normally be left alone.

I/O Recovery Time Delay

The AT Bus uses wait states to increase the width of an AT BUS cycle, for slower-reacting expansion cards, and this refers to the delay before starting Input/Output cycles. The lower the value, the better the performance, but you might have to change DMA settings as well.

I/O Recovery Select

As for *I/O Recovery Time Delay*.

AT Bus Precharge Wait State

Set to 0 for best performance, but you may need 1 for some items, like AHA 1542Bs, at high speeds.

I/O Cmd Recovery Control

If enabled, a minimum of 7 bus clocks will be inserted between any 2 back-to-back I/O commands. This helps with problematic expansion cards and can affect ROM wait states, DMA and bus timing. Disable this, or set to Normal or the lowest figure available for best performance. Also known as *Timing Parameter Selection*.

Single ALE Enable

ALE stands for *Address Latch Enable*, an ISA bus signal used by 808x processors when moving data inside the memory map; it is used by DMA controllers to tell the CPU it can move data along the data bus, or that a valid address is posted. Conversely, they can stop this signal and make the CPU wait while data is moved by the controller, so set to *No* for normal use.

When the CPU wants data, it places the addresses it wants to look at on the bus, followed by a control signal to let the memory controller know the address is there, which then latches the address, decodes it and puts what the CPU wants on the bus, where it can be latched in turn by the CPU (latch means read).

If this is enabled, single instead of multiple ALEs will be activated during data bus access cycles. Yes is compatible with AT bus specifications, giving less performance, as multiple ALE signals during a single bus cycle effectively increase the bus speed, if the hardware can handle it. This sometimes appears in older BIOSes as Quick Mode, and you might see Extended ALE instead of Multiple. May slow the video if enabled, or you might get missing characters on screen.

ALE During Bus Conversion

Selects single or multiple ALE signals during bus conversion cycles. Depends on system speed.

E0000 ROM belongs to AT BUS

Officially, the E000 area of upper memory is reserved for System BIOS code, together with F000, but many machines don't use it, so E000 can often be used for other purposes (note, however, that this 64K is needed to run protected mode software, such as Windows, OS/2, or Multiuser DOS, which loads Advanced BIOS code into it). This will only tend to appear on older machines, as PCI needs it too. It determines whether access to the E area of upper memory is directed to the system board, or to the AT bus. Set Yes if you want to use it for anything like a page frame or a Boot ROM), or if you're using Multiuser DOS and want the maximum TPA to be available. Can also turn up as E000 ROM Addressable.

Internal MUX Clock Source

Mux means Multiplex. Controls the frequency of polling the IRQ, DRQ and IOCHCK# signals. Sometimes this has an AUTO setting which sets the frequency according to CPU speed, but usually SCLK/1 is recommended. I don't think it refers to Memory, Upper and XMS in some operating systems, like Novell DOS 7.

Fast Decode Enable

According to one motherboard manual, DRAM access is speeded up if this is enabled, and it's possibly ignored if internal/external cache is present. Otherwise, it enables a chipset initiated reset of the CPU when the keyboard controller is instructed so to do, speeding up transitions from protected to real mode on 80286 CPUs and above. See also Fast Gate A20 Option, and *Fast Reset Emulation*.

Fast CPU Reset

See *Fast Reset Emulation*.

Extended I/O Decode

In (8-bit) ISA systems, ten address lines are normally used for I/O address decoding, that is, in ports 000-03FF. If your motherboard uses more, enable this for better performance to get

0000-FFFF. Some cards can use the same lower 10 bits by accident, in which case enable this. Otherwise, leave it (more in Base I/O Address in Expansion Cards).

Local Bus Ready

Selects the timing the system will use to exchange data with a VL-bus device after it has signalled that it is ready. The choices are:

- *Synchronize*. Synchronize and pass to VESA slot in the next clock (default).
- *Transparent*. Enable the exchange immediately, i.e. pass the LRDY# signal directly from VESA slot via chipset to CPU.

Local Bus Ready Delay 1 Wait

Mostly disable this in systems running at 33 MHz or below, but some VL-bus devices may need 1 wait state anyway. You may need to enable this (i.e. insert 1 wait state) for 50 MHz.

Local Bus Latch Timing

Specifies the time period in the AT machine cycle when the VL-bus is latched (read), so data can be transferred reliably, that is, to hold data stable during transactions with the local bus, the local bus will be latched after a read command and before the end of the AT cycle. This determines how long the system will wait to latch the bus after the read command has gone inactive. Use T2 (2 clocks) for 25/33 MHz, or T3 (3 clocks) for 40/50 MHz. T2 is earlier in the cycle than T3.

Latch Local Bus

See *Local Bus Latch Timing*.

ADS Delay

Concerns the local bus. If enabled, it affects performance; the default is disabled, or *no delay*. ADS# is a bus control signal, or an Address Status strobe driven by the CPU to show the start of a CPU bus cycle, indicating that a valid command and address is stable on the bus. When enabled, more time is allocated for ADS; you only need this with a faster processor.

IDE Multi Block Mode

This setting may only be relevant under DOS or Win 3.x, as 95/98 and NT have their own drivers (and NT before SP2 doesn't like it anyway, so turn it off or you might get corruption). It enables suitably configured IDE hard drives to transfer multiple sectors per interrupt, as opposed to one (there may be an option to specify the number of sectors), using the ATA Read Multiple and Write Multiple commands. For example, setting 16 saves 1920 (2048-128) interrupts-this is to avoid situations where the CPU can take some time to reply to an interrupt. There are several modes, often dependent on the size of your hard disk cache, because if there isn't one, data cannot be queued properly. The first three, 0-2, are from the old ATA standard. The others (3 and 4) are ATA-2 specific and use the IORDY line to slow the interface down if necessary. Interfaces without proper IORDY support may cause data corruption, so don't expect to mix two drives with different modes on the same channel, although performance reduction only appears to happen if both drives are accessed at the same time. If you must mix, and you get problems, force each drive to its proper mode.

- *Mode 0*. Standard Mode; conforms to original PC standard, compatible with all drives. Single sectors transferred with interrupts.
- *Mode 1*. Polls the drive to see if it's ready to transfer data (no interrupts).
- *Mode 2*. Groups of sectors are transferred in a single burst.
- *Mode 3*. Uses 32-bit instructions, up to 11.1 Mb/sec.
- *Mode 4*. Up to 16.7 Mb/sec. Two versions; the second supports 32-bit transfer, possibly to cope with 32-bit disk access.
- *Mode 5*. Up to 20 Mb/sec, but abandoned for *Ultra DMA*, due to electrical noise.

This setting only concerns transactions between the CPU and IDE controller - UDMA or Ultra ATA are not the same thing and concern themselves with the IDE controller and the device. It can mess up comms software when up- or downloading, because multi block transfers cannot be interrupted, and you may lose characters. For example, you need to run telix with the D option (e.g. drop DTR when writing to disk), or use buffered UARTs for terminals with Multiuser DOS. Consider also disabling Smartdrive.

The T I Chipset has problems with this as well, due to its plumbing arrangements; it gets its timing from the PCI clock, with a minimum (fastest) cycle of 5 clocks, so the maximum transfer rates achievable are:

PCI Clock (MHz)	Transfer Rate (Mb/sec)
25	10
30	12
33	13.3

There is also a reliability problem, and you may get data corruption if you try to get more than 11 Mb/sec or so with Mode 4 (Microsoft also suggest that this should be disabled for Windows NT before SP2- see article Q152/3/07.asp), so the MR BIOS doesn't select rates beyond that automatically. If you can set block sizes, the FAT system seems to like them the same as the cluster size, and as what's best for the drive is not necessarily best for the system as a whole, check this with a high level benchmark, that is, at application level. Quantum have a document called ATA Signal Integrity Issues that explains more.

It's best not to have EIDE CD-ROMs on IDE channels by themselves, (say, in a SCSI system) as 32-bit addressing may only be turned on with a suitable hard drive as well. 24x CD ROMs cannot reach full speed in 16-bit mode.

IDE Block Mode Transfer

As for *IDE Multi Block Mode*.

Multi-Sector Transfers

As for *IDE Multi Block Mode*, allowing you the choice of 2, 4, 8 or 16 sectors. An auto setting queries the drive and allows it to set itself.

IDE Multiple Sector Mode

If *IDE Multi Block Mode* (or similar) is enabled, this sets the number of sectors per burst. Setting 64 gives the largest size your drive supports. Watch this with comms; when multiple sectors are being transferred, they can't be interrupted, so you may lose characters if you don't have buffered UARTS. See *IDE Multi Block Mode* above.

Multiple Sector Setting

As for *IDE Multi Block Mode*. The number of sectors transferred per interrupt. If disabled, an interrupt will be generated for each sector transferred. You get a choice of 4, 8 or AUTO.

IDE (HDD) Block Mode

Makes multi-sector transfers, as opposed to single-sector transfers, or reads and writes using large blocks of data rather than single bytes. It affects the number of sectors transferred per interrupt. Only appears in BIOSes dated approximately 08/08/93 or later. This can also be called block transfer, multiple commands or multiple sector read/write. The automatic setting will sort out the optimum rates.

IDE 32-bit Transfer

Many local bus interfaces can combine two 16-bit words into a 32-bit doubleword when reading data to and from the disk, since the IDE channel itself is only 16-bit. This is particularly useful with bus mastering, and is often called 32-bit access, though it's really 32-bit host bus transfers. Either way, more efficient use is made of the bus and CPU, so this may or may not make much difference if you don't actually have a bottleneck. This is not the same as Windows' 32-bit features, which are also misnamed as they just work in protected mode.

Like *Block Mode*, this setting only concerns transactions between the CPU and the IDE controller - UDMA or Ultra ATA are not the same thing and concern themselves with the IDE controller and the device. If disabled, 16-bit data transfers are used, so performance will be less. If enabled, hard disk data is read twice before request signals are sent to the CPU. This setting can only be enabled if IDE Prefetch Mode is also enabled (below). As far as AMI are concerned, the WinBIOS will initialise the hard disk firmware for 32-bit I/O, assuming your hard disk is capable-it refers to the new release of high performance Mode 4 drives. Microsoft suggest that this should be disabled for Windows NT - see article Q152/3/07.asp.

CPU ADS# Delay 1T or Not

With a CPU clock at 50Mhz, choose Delay 1T. Otherwise, disable. Probably only for BIOSes that support PS/2 mice.

Fast Programmed I/O Mode

Controls the speed at which Programmed I/O (PIO) transfers occur on the PCI IDE interface. If disabled, Mode 0 (e.g. unoptimised) is used, so only use this if a device cannot function with advanced timings.

IDE Primary Master PIO

Enables PIO mode (as opposed to DMA) where all data is passed through the CPU, which is inefficient, but at least maintains cache coherency and allows the operating system to move buffers around without problems. Phoenix have recommended using fast IDE timing and Block Mode instead of PIO Mode 3.

IDE Primary/Secondary Master/Slave PIO

You can set a PIO mode (see above) for each of the four IDE devices your system supports. Auto is usually best, especially if you change drives a lot.

IDE Primary/Secondary Master/Slave UDMA

See above.

Channel 0 DMA Type F

What DMA channel the first drive (0) in the system uses when set to F (see *IDE DMA Transfer Mode*). Choices are Disabled (no drive), 0, 1, 2, or 3.

Channel 1 DMA Type F

As for Channel 0 DMA Type F, but for the second drive.

IDE DMA Transfer Mode

The default is *Disabled* (=PIO), but you have the choice of:

- Type B (for EISA)
- F or Standard (PCI) as well (EIDE supports B/F, for 8.53-13.33 Mb/sec)

Type F is an 8.33 MHz EISA-style PCI DMA (normal is 5 MHz) for PCI/ISA, which replaces EISA type C, although A and B transfers are supported. C is a burst mode that needs special controller logic. However, with F, you cannot DMA into ISA memory, only PCI, and neither does Type F apply to PCI bus mastering. The Standard setting is the same as Disabled, but you can set the number of sectors per burst (see below). Type F is fastest, but there may be conflicts with multimedia. Use Standard or Disabled IDE CD ROMs.

Large Disk DOS Compatibility

For drives greater than 528 Mb not using LBA. This and LBA are not supported by all operating systems (e.g. UNIX R3.2.4).

IDE LBA Translations

See *IDE Translation Mode.*

LBA Mode Control

See *IDE Translation Mode.* Turns LBA on or off.

IDE Prefetch Mode

Enables prefetching for IDE drives that support it for the onboard IDE connectors. If you are getting drive errors, change the setting to omit the drive interface where the errors occur, or, if you install a primary and/or secondary add-in IDE interface, set this to Disabled if the interface does not support prefetching. Does not appear when Internal PCI/IDE is disabled.

ISA IRQ 9,10,11

These may be used by the PCI bus if they are available, so set them as *Used* if you want to reserve them. Some VGA cards like 9, but many don't, so you might save an interrupt.

IDE Translation Mode

For large IDE drives. Disable for smaller drives below 528 Mb. Choices are:

- *Standard CHS*. (Cylinders, Heads, Sector) - limit is 528 Mb
- *LBA*. Logical Block Addressing; both BIOS and drive must support it. CHS addresses are used to create a 28-bit Logical Block Address rather than being mapped separately; in short, LBA sequentially assigns unique numbers to sectors, which are not necessarily in the same place if the drive is used on another machine.
- *Extended CHS*. Similar to LBA, but not quite. Also known as Large. Can better performance of LBA.

Different systems cope with the above in different ways; Unix does its own thing, OS/2 2.1 can support them all, as can DOS and Windows, but if you're running Windows' 32-bit Disk Access, select Standard CHS, unless you have a version of wdcddrv.386 that supports advanced geometries. OS/2 2.0 and Netware cannot support LBA. If set to Auto Detect, the BIOS will detect what the drive is capable of, not what it is formatted with. Your hard drive may require different input to the CMOS for each method. See also Hard Disk (C and D).

Onboard CMD IDE Mode 3

Found where CMD Enhanced IDE chipsets are built in to the motherboard. The code is kept in a ROM at E800, and this setting allows access to it. Enable for best performance, as the code will still be used to optimise hard disk useage, with 32-bit I/O, even if it is not compatible with Mode 3.

Note: There are problems with many PCI motherboards and CMD controllers, especially with true 32-bit operating systems, where subtle changes are made to your files; that is, bytes are randomly changed once in a while. The problems also appear with Windows for Workgroups in 32-bit mode during floppy backup/restore.

More information from **http://tcp.ca/Nov95/PCIController.html**.

Enhanced ISA Timing

Gives higher bus speeds, set by manufacturer.

Back To Back I/O Delay

Inserts a slight pause (say 3 ATCLK signals) in between 2 processes talking to the same I/O port.

DMA FLOW THRU Mode

Enable this if you enable write buffers to avoid inconsistencies; this makes the DMA wait until all write buffers are empty. You won't increase performance by increasing the DMA clock by itself but, since it's often linked to the bus clock, will increase in sympathy with it. Generally, only floppies use DMA anyway, but some tape streamers and sound cards do.

Extended DMA Registers

DMA normally takes place within the first 16 Mb of address space on an AT. This setting allows you to use the whole 4 Gb of address space of a 32-bit processor.

Hold PD Bus

Sets the timeout function of the processor data bus, presumably before it assumes a malfunction. The default is 1-2T.

DMA Channel Select

Helps you change IRQ and DMA channels of a built-in SCSI controller.

Concurrent Mode

Allows DMA access for floppies and tapes, as QIC and others commonly share controllers with floppy disks. However, many computers will not support this.

Fast Programmed I/O Modes

Controls the speed at which PIO transfers occur over the PCI IDE interface:

- *Disabled*. Mode 0
- *Autodetect*. Rated maximum of the drive

Only set disabled if a drive incorrectly reports its capabilities. Do not use mixed mode drives on the same channel; at least, don't let the BIOS on a board with a Triton chipset make its own decision, as it seems unable to handle two drives with separate EIDE rates; they share a common timing register. The MR BIOS can handle this better than most.

Local Device Syn. Mode

Concerns Synchronous and Bypass mode for the CPU's signal to terminate Local Bus cycles. Bypass mode, or transparent mode, gives better performance, but is limited to 33 MHz or below because it is not compatible with VL bus cards.

Data Transfer

You have the following choices:

- *PIO*. Polling mode; the CPU controls everything and fetches each byte from the controller through I/O addresses.
- *DMA*. Transfer is done by DMA, which is faster when multitasking, as the CPU can get on with something else whilst data is being transferred. With ISA, this only works below 16 Mb.

Don't switch on DMA mode with a PIO device installed.

DMA Frequency Select

Sets the frequency at which DMA (Direct Memory Access) data transfers take place as a function of the system clock. Choices are:

- *SYSCLK/1*. Enable one full system clock cycle
- *SYSCLK/2*. Enable one-half system clock cycle (default)

Hard Disk Pre-Delay

POST procedures are quite fast these days. This setting delays the BIOS's attempts to initialise the first IDE drive in the system, so slower devices can have a chance to get their act together; some drives may hang if they are accessed too soon. Set this in conjunction with Initialisation Timeout (below). See also Cold Boot Delay.

Initialisation Timeout

The number of seconds the BIOS will wait to see if an IDE drive is there before proceeding. Works with Hard Disk Pre-Delay. If your drive doesn't respond within the specified period, the system will not recognize it.

CACHEING

Disabling cacheing often cures obscure memory problems; it may be because non-32-bit address cycles are redirected to the AT Bus. Certainly, with cacheing enabled, only 32-bit cycles are affected, but Hidden Refresh is often automatic as well. Also, Shadow RAM is cached here. Be aware that some chipsets do more than just disable the cache when you select Disable. Cache SRAM can be tested in the same way as DRAM, except for Tag RAM, which cannot be written to directly, so there is a special access channel for testing. Data is written, read and checked for consistency. If this can be done in a certain time, say by the end of T2, it is likely to be Burst SRAM. SRAM chips share a common data bus with other memory processor devices which need to control the bus at some time or other. If you minimise the cycle times for each, you get the maximum performance. Bus contention occurs when 2 devices are trying to use the bus at the same time. Any settings with regard to this therefore affect reliability. Certain cycles are non-cacheable anyway, such as I/O cycles, interrupt acknowledge cycles, halt/shutdown cycles and some memory areas.

Cacheable cycles come in four varieties:

- *Read Hit* means the system reads the data from the cache, therefore not needing to go to system memory.
- *Read Miss* means the data is not in the cache, so it goes to system memory and will copy the same data to the cache.
- *Write Hit* means the system writes the data to cache and main memory.
- *Write Miss* means the system only writes the data to system memory.

A non-cacheable location is not updated on a read miss, so when a shadow RAM location is changed to it from cacheable, the memory cache must be flushed to guarantee that the memory has been purged. Some chipsets cannot cache more than a certain amount of system memory, but your operating system will determine whether or not you get a performance hit if you have more than that on board. Windows, for example, uses memory from the top downwards, so will always be using the non-cacheable area. Linux uses it from the bottom up, so will only slow down once you enter the critical area.

Cache RAM (SRAM) Types

Here you can tell the machine what L2 RAM it has to deal with, *Pipeline*, *Burst* or *Synchronous*. They are fully described in the *Memory* chapter.

Pipeline Cache Timing

Two choices, *Faster* and *Fastest*, to suit the speed of your memory. Select the former for a one-bank L2 cache, and the latter for two banks.

Cache Timing

As above.

F000 Shadow Cacheable

When enabled, accesses to the System BIOS between F0000H-FFFFFH are cached, if the cache controller is enabled.

Fast Cache Read/Write

Usually used if you have two banks of external SRAM cache chips, that is, 64 or 256K. It's similar to Page Mode for DRAM.

Flush 486 cache every cycle

Enabled, flushes the internal 8K cache of the 486 every cycle, which seems to defeat the object somewhat. Disable this.

Read/Write Leadoff

Before data can be accessed, the core logic must issue the memory address signal, the column address strobe (CAS) signal and the row address strobe (RAS) signal to the DRAM. However, these signals are not issued at the same time-the time difference between them is called the lead-off time, and often equates to the timing of the first cycle in a burst. It varies for read and write actions, depending on the DRAM-some may require longer delays.

Async SRAM Read WS

Choose the timing combination for your motherboard and memory against read cycles.

Async SRAM Write WS

Choose the timing combination for your motherboard and memory against write cycles.

Async SRAM Leadoff Time

Sets the number of CPU clock cycles your asynchronous SRAM needs before each read from or write to the cache. See also *Read/Write Leadoff.*

Sync SRAM Leadoff Time

Sets the number of CPU clock cycles your asynchronous SRAM needs before each read from or write to the cache. See also *Read/Write Leadoff.*

Async SRAM Burst Time

Sets the timing for burst mode cache operations. The fewer the faster.

Cache Burst Read Cycle Time

See *Cache Read Hit Burst*, below. Automatically set to 2T if only one bank of Level 2 cache is available, that is, the whole cycle takes place inside 2 T-states.

Cache Read Burst

This covers how data is read from the cache, depending on the cache size and speed of its memory. In this particular case, the default may be best.

Cache Write Burst

Similar to above, but for writes to the cache.

Cache Read Wait State

Sets the number of wait states to be added on reads from cache memory, just in case you're using slow cache chips, or you wish to preserve data integrity. This affects the cache output enable signals, specifically CROEA# and CROEB#. They are active for 2 CPU clocks at 0 wait states, or 3 at 1, which should be used for 40 MHz 486s (you can use 0 wait states at 33 MHz). Some VL bus devices need 1 wait state on 50 MHz systems. Whatever you set here is automatically adjusted anyway during L2 write-back-to-DRAM cycles for synchronisation purposes with the DRAM controller.

Cache Write Wait State

Similar to the above, but for writes. May be selected by the board designer.

Cache Read Hit Burst

Burst Mode is a 486 function for optimising memory fetcheswhen going off-chip, which works by reading groups of four double-words in quick succession, hence burst. The first cycle deals with the start address as well as its data, so it takes the longest (the other three addresses are deduced). Once the transfer has been started, 4 32-bit words could therefore move in only 5 cycles, as opposed to 8, by interleaving the address and data cycles after the first one. For this, you need fast RAM capable of Page Mode.

These SRAM timing numbers are the pattern of cycles the CPU uses to read data from the L2 cache, by determining the number of cycle times to be inserted when the CPU reads data from the external (Level 2) cache, when it can't catch up with the CPU (you may see similar figures allocated to L1 cache, on chip). The Secondary Cache Read Hit can be set to 2-1-1-1, 3-1-1-1, 2-2-2-2 or 3-2-2-2 (3-1-1-1 means the first 32-bit word needs three clock cycles and the remainder need one, giving a total of 6 clock cycles for the operation). Performance is affected most by the first value; the lower the better; 2-1-1-1 is fastest. You can alter it with the Cache Read Hit 1st Cycle WS setting. This will have no effect if all the code executes inside the chip.

For example, on older machines, the setting for 33 MHz may need to be changed to 3-2-2-2 if you only have 128K, or with Asynchronous SRAM. If you are allowed to change them, the following may be useful as a starting point (1 bank cache/2 banks cache):

Item	20 MHz	25 MHz	33 MHz	50 MHz
SRAM Read Burst	3222/2111	3222/2111	3222/3111	3222
SRAM Write Wait States	0W	0W	1/0W	1W
DRAM Write Wait States	0W	0W	1W	1W
DRAM Read Wait States	1W	2W	2W,	3W
RAS# to CAS# Delay	1 Sysclk	1 Sysclk	1 Sysclk	2 Sysclk

Pentiums can perform Burst Writes as well as Burst Reads, so you might have a separate selection for these. 4-1-1-1 is usually recommended.

SRAM Read Timing

Similar to *Cache Read Hit Burst*, above. Relates the number of cycles taken for the SRAM address signal to the number allocated for the actual read. 2-1-1-1 is the default.

SRAM WriteTiming

Sets timing, in CPU wait states, for writes to external cache. 0 WS is the default.

Cache Address Hold Time

The number of cycles it takes to change the CAS address after CAS has been initiated (asserted) aimed at a target address (location) in DRAM.

Burst SRAM Burst Cycle

This sets the precise timing of the burst mode read and write cycles to and from the external cache. Choices are:

- *4-1-1-1*. Slower.
- *3-1-1-1*. Fastest (Default).

Cache Mapping

Direct mapping is where data is loaded in one block. N-way is divided into n-banks (2-way, 4-way, etc). Further explained in the Memory chapter.

Data Pipeline

With reference to cache mapping above, after accessing DRAM for the first time, the data is stored in a pipeline. Enabling this is best for performance.

Cache Wait State

0 for best performance, but 1 may be required for VL bus devices at higher speeds. SRAM used for cacheing has a minimum access time requirement, otherwise you will get malfunctions. The trick is to use the least number of wait states that don't cause failures.

Cache Read Burst Mode

An Award setting, for 486s. See *Cache Wait State*, above.

Cache Write Burst Mode

An Award setting. See Cache Wait State, but delete Read and insert Write.

Cache Read Cycle

As for *Cache Wait State*.

SRAM Back-to-Back

Reduces the latency between 32-bit data transfers, so it is transferred in 64-bit bursts. Back-to-Back means that address reads can alternate with page hits, which measn that data transfer happens at half-speed, but also that there are very low latencies, zero in some cases.

SRAM Type

Which type, Async or Synchronous, is installed.

CPU Internal Cache/External Cache

Enables or disables L1 and L2 caches.

CPU Cycle Cache Hit WS.

- *Normal*. Refresh with normal CPU cycles
- *Fast. Refresh* Refresh without CPU cycles for CAS

The second option saves a CPU cycle; see also *Hidden Refresh*.

Cache Write (Hit) Wait State

Sets the wait states to be added on writes to cache memory. 1 should be used for 40 MHz systems, and you can use 0 at 33 MHz. Some VL bus devices need 1 on 50 MHz systems.

Fast Cache Read Hit

Should be enabled with 64 or 256K of cache memory; otherwise it should be disabled.

Fast Cache Write Hit

See *Fast Cache Read Hit*.

Cache Tag Hit Wait States

This is similar to Cache Read Wait States, in that it allows you to set the number of wait states, 0 or 1, used to test for a cache tag hit.

Tag Compare Wait States

The tag sample point can be in the first T2 cycle (0 wait states) or the second (1 wait state). For the former, you need 12 ns SRAM or faster.

Cache Scheme

Concerns the L2 cache on the motherboard, between the CPU and memory, and whether it is to be *Write Back* (WB) or *Write Thru* (WT). The latter means that memory is updated with cache data every time the CPU issues a write cycle. Write Back causes main memory updates only under certain conditions, such as read requests to memory locations with contents currently in the cache. This allows the CPU to operate with fewer interruptions, increasing efficiency, but is not as safe in the event of power loss.

HITMJ Timing

For a write-back L1 cache, you can select the HITM# signal as inactive to the timing relating to IOCHRDY inactive. The choices are 2, 3, 4 or 6T. With only write-through, this cannot be used. 1t is equal to 1 CPU clock.

Internal Cache WB/WT

See *Cache Scheme*.

External Cache WB/WT

See *Cache Scheme*.

CPU Level 1 Cache

Enables or disables the internal CPU cache, maybe for stability reasons, game performance, manipulating really large files or troubleshooting when overclocking, but it's not a good idea to leave it off permanently.

CPU Level 2 Cache

See above.

CPU Level 2 Cache ECC Checking

This setting enables or disables ECC checking by the L2 cache, to detect and correct single-bit errors in data stored there. It's mainly for file servers, where errors would be spread round the network. ECC (*Error Correction Code*) needs DIMMs with an extra 8 bits of bandwidth (they have an x72 designation, as opposed to x64). It works with the memory controller to add bits to each bit which are decoded to ensure that data is valid, and used to duplicate information should it be necessary. Multi-bit errors are detected but not corrected. Although similar to parity, there is only a penalty cycle when a 1-bit error is detected, so there is no performance hit during normal operations. You can use ECC chips in a non-ECC board, but you won't get the benefits. This may be useful when overclocking causes errors.

CPU L2 cache ECC Checking

See above.

Cache Write Back

See *Cache Scheme*.

L2 Cache Write Policy

See also *Cache Scheme*, above. Depending on the SRAM, for this setting, in addition to the Write-Back and Write-Through options, the L2 cache also offers *Adaptive WB1* and *Adaptive WB2*, which try to reduce their disadvantages.

L1 Cache Write Policy

As for *Cache Scheme*, for L1 (internal) cache on the CPU.

L1 Cache Policy

See above.

L1 Cache Update Mode

See *Cache Scheme*.

L2 Cache Write Policy

Similar to above, but you might also see *Adaptive WB1* and *Adaptive WB2*, which try to reduce the disadvantages of write-back and write-thru caches.

L2 Cache Enable

When disabled, cache addresses are regarded as misses, so the CPU talks directly to main memory; the effect is the same as not having it, as the cache is not actually turned off (you just can't read from it). If it does become enabled, you can get coherent data immediately, as it is still being updated. Since the cache is now on theprocessor core, this should be enabled.

L2 Cache Zero Wait State

If you have a slower cache, disable this to have one wait state when accessing the external cache controller. When enabled, the chipset will not wait.

L2 Cache Cacheable Size

The size of the system memory the L2 cache has to cope with, for motherboards that can take it. Up to 64 Mb or 512 Mb on HX motherboards, and must be set at least as high as the memory you have - select 512 MB only if your system RAM is greater than 64 MB. Chips with an integrated L2 cache (i.e. Pentium Pro, PII, etc) will not use this.

L2 Cache ECC Checking

This is not supported by any chipset or CPU with L2 cache in the core, so is redundant.

L2 Cache Cacheable DRAM Size

See above.

L2 Cache Latency

In theory, the lower the value, the faster the performance, at the expense of stability, until it is set too low, whereupon the cache will not work at all and neither will the system-the best way to find out the optimum value is to test. Performance gains are reported to be small, but high values here help with overclocking, which is probably why it was included. The default setting, for the Celeron anyway, is 5.

Cache Over 64 Mb of DRAM

See above.

Linear Mode SRAM Support

Enable for an IBM/Cyrix CPU and linear mode SRAM, to get slightly better performance. Disable for Intel CPUs, as they only support Toggle Mode.

M1 Linear Burst Mode

See above. Enable for a Cyrix M1.

Cache Write Cycle

Affects the data hold time for writes to DRAM.

Posted Write Enable

A Posted Write Cache has "write buffers" that buffer data and write when things are quiet or, rather, when they don't interfere with reads. It's somewhere in between a write thru and write back cache. With write back, if the CPU writes a single byte to memory, and that address is in the L1 cache, the cache line with the newly written data is marked 'dirty' to indicate there is a difference between it and main memory. When the dirty cache line needs to be overwritten with newer information, the cache management routine uploads the new line (16 bytes) from lower memory, from which it cannot tell the new data, so it first writes all 16 bytes to memory, which can use as many as 18 clocks (6-4-4-4). Once the dirty line is written, the upload of the new line can begin. A good posted write system can accept the CPU write operation in a single clock, write the data to main memory when the bus is otherwise not in use, and never have to suffer the 18 clock penalty. Write Back cache is therefore best when most or all of a line is made dirty and writes occur to addresses inside the cache system, which is not usual with multitasking and large active memory windows. Posted Write Buffers are typically used between PCI bus and IDE interface by decoupling the wait states effect from the slower IDE side, but also between the CPU and PCI bus. Read-ahead buffers eliminate idle cycles.

Posted Write Framebuffer

Good for video performance, especially for the Matrox G200, so disable only if you have instability.

Posted I/O Write

Disable if using Multiuser DOS on an Intel Express.

Tag Ram Includes Dirty

Enabling this tells the system that the SRAM needed for the machine to remember that the Level 2 cache and main memory contents are different is actually present on the motherboard (not often the case). If you can enable this, you will get about 10% extra performance, because unnecessary line replacement cycles can be eliminated (e.g. when flushing the old data then replace it with the new).

Tag RAM is used as a directory, or table of contents, between main memory and cache RAM, storing the addresses of whatever data is in cache memory, so it is slightly faster as it needs to be accessed first. The CPU checks Tag RAM for the address of any data it requires, which is how it knows it has to go to main memory if it's not there.

On top of whether the chipset can support it, it is actually the amount of tag RAM (or rather its width) that determines how much system memory is cacheable, since it can only store a certain amount of addresses. If the tag ram has a -8 or a -10 stamped on it, it is capable of caheing 128 or 156 Mb of system RAM (stands for 8 or 10-bit).

Some cache controllers support two methods of determining the state of data in the cache. One separates the tag signal from the alter (or dirty) signal, which imposes a minimal performance decrease, since the system must assume that some cache lines have been altered. When the dirty and tag bits are combined, the system performs more efficiently, but less cache will be available (default).

Tag/Dirty Implement

One way of checking the state of data in the cache separates the tag from the dirty signal, while the other combines them into a single 8- or 9-bit signal.

- *Combine.* Tag and Dirty combined in one 8- or 9-bit signal, depending on whether 7 or 8 bits are selected in *Tag RAM Size* (default)
- *Separate.* Tag and Dirty signals are separate

Alt Bit Tag RAM

Choices are 7+1 or 8+0. 7+1 is recommended. The Alt Bit means Alter Bit, or dirty bit, which indicates the particular line in L2 cache that contains modified data, so it keeps a note of the state of data in the cache. If you have selected Write Back for the external (L2) cache, 7+1 bits (the default) provides better error detection. With 8+0 Bits, the Alt bit is always assumed active.

Tag Option

If you have WB (Write Back) for L2 cache, 7 + 1 provides better error detection. It means 7-bit tag cache RAM with one dirty bit. The alternative is an 8-bit tag.

Tag RAM Size

Set the specifications here, whether 7 or 8 bits. See above for definitions. 8-bits means 128 Mb of system RAM can be cached.

Non-cacheable Block-1 Size

Depending on the chipset, this concerns memory regions (including ROMs) not within the 32-bit memory space, e.g. those on 16-bit expansion cards on the expansion bus (video cards, cacheing disk controllers, etc) that should not be cached because RAM on them is updated by the card itself, and the main board cache controller can't tell if the contents change. These devices communicate as if they were DRAM memory (that is, they are memory-mapped), which means they need to react in real time and would be seriously affected by cacheing. You would also use this to lock out any ROMs you can't otherwise disable cacheing for; certain cacheing IDE controllers use a space at the top end of base memory for hard disk details, and therefore cause timing problems if the information is cached; symptoms include consistent bad sectors when formatting floppies, or a scrambled hard disk.

Also, video cards sometimes use a 1 Mb area in the 16 Mb address space of the ISA bus so they don't have to bank switch through the usual 64K page (early Video Blaster cards are notable for this requirement; they won't work in a machine with more than 15 Mb RAM).

You might get a choice of System Bus or Local DRAM. The former produces a hole in Local DRAM. NCB areas can be separate, contiguous or overlapped. With Asustek cache controllers, include the video buffer at A000-BFFF. This setting is closely linked to the next.

Note: Some chipsets (e.g. SiS) use this to define non-cacheable regions only in local DRAM; with them, memory on PCI or VESA add-ons is always non-cacheable. Where memory space is occupied by both local DRAM and an add-on card, the local DRAM will take priority (as does VESA over PCI), so disable this to allow access or give priority to the card.

Non-cacheable Block-1 Base

The base address of the above block must be a multiple number of its size; e.g. if 512K was selected above, the starting address should be a multiple of 512K. In other words, if the previous option has a number other than Disable, this option will increment by that number.

Non-cacheable Block-2 Size

Can be 64K-16 Mb; otherwise, as above.

Non-cacheable Block-2 Base

See *Non-cacheable Block-2 Size*.

Memory above 16 Mb Cacheable

See *Cacheable RAM Address Range*.

Cacheable RAM Address Range

Memory is cached only up to the 16 or 32 Mb boundaries to reduce the bits that need to be saved. The lower the setting here, the better, according to your main memory; that is, if you have 4 Mb, set 4 Mb. This memory is cached into SRAM.

XXXX Memory Cacheable

Some shadowed memory segments (e.g. starting at address C800) can be cached (or not). However, cacheing certain code (video or ROM BIOS) is sometimes inefficient because it is constantly updated, and you may get "cache thrash", where data feeds on itself in a circular fashion as new data constantly replaces the old. Also, certain programs that depend on timing loops could run too fast. Where you can select Associativity, you can improve on the normal direct mapped cache, where alternating references are made to main memory cells that map to the same cache cell, and all attempts to use the cache therefore result in misses. Associativity concerns the amount of blocks that the cache memory is split into. For example, a 4-Way Set Associative cache is in four blocks, and is used as four locations in which different parts of main memory are cached at the same time; a lot to keep track of. Its performance yield is not normally enough over a 2 Way Set to justify its use. Direct mapping is known as 1-way Associativity. Non-cacheable regions set elsewhere (above) override this.

C000 Shadow Cacheable

See *XXXX Memory Cacheable*.

Video BIOS Area cacheable

See also *XXXX Memory Cacheable*. Only valid when Video BIOS Shadow is enabled, in which case the shadowed BIOS code will be cacheable. Be prepared to say No for an accelerator card which does its own thing, as the CPU needs to be kept informed of its activities, and if you have write-back cacheing enabled, your video won't be updated properly because the data will not reach the video board until the cache line it's in needs flushing. See also XXXX Memory Cacheable, above.

Cacheing RAM that is already shadowed is not often a good idea, as the data often ends up in the internal cache of the CPU. Disable for safety, though it might work.

Video BIOS cacheable

See above.

Video Buffer Cacheable

When enabled, the video BIOS (C0000h-C7FFFh) is cached.

System video cacheable

See above.

System BIOS Cacheable

Enables or disables the caching of the system BIOS ROM at F0000h-FFFFFh inside the L2 cache, which not only has the potential for trouble if a program writes to this area, but is a bit of a waste because operating systems such as Windows, etc. do not access the system BIOS much anyway - after booting, all parameters are loaded into memory.

A performance drop has been noticed under W2k Pro on a dual processor machine with this disabled, together with a very long shutdown time.

Video BIOS Cacheable

As above, but enables or disables caching of the video BIOS ROM at C0000h-C7FFFh, also inside the L2 cache. Disable for the same reasons as above, unless you're using one of those old video cards with 1 Mb of memory.

Video RAM Cacheable

Cache technology (in L2) for the contents of video RAM (used by the graphics adapter) at A0000h-AFFFFh, not the same as cacheing the video BIOS instructions that are already shadowed (see Video BIOS Area cacheable above). Leave on the default of Disabled if your display card does not support it, otherwise your system may not boot (and programs writing into this memory area will crash the sytem). It also reduces performance, as high-bandwidth video RAM contents are transferred to L2 over the AGP/PCI bus, and back when needed, so its moving twice in a slower environment than its natural habitat. That is, although the L2 cache is faster than system memory, the graphics chip can only access the data there though the AGP (or PCI) bottleneck.

VESA L2 Cache Write

Sets the timing of writes from the VESA bus to the external cache. Using a long cycle gives you greater system stability, but you lose some performance.

- *Normal*. VESA to cache writes handled normally (Default)
- *Long*. Longer timing used in VESA to cache writes

Shadow RAM cacheable

Again, not often a good idea, as the data often ends up in the internal cache of the CPU. Disable for safety, though it might work.

SRAM Speed Option

The speed of standard SRAM cache during normal read. Similar to DRAM Speed.

Cache Early Rising

This allows you to select the fast write-pulse rising edge technique of writing to the external cache over the normal timing, which is faster. Use this to cope with older DRAMs.

- *Enable*. Write pulse on the rising edge (Default)
- *Disabled*. Normal write pulse to the cache

L2 Cache Tag Bits

Cache tag bits report the status of data in the cache. This selects the number of bits used.

- *8 Bits*. Eight tag bits (Default)
- *7 Bits*. Seven tag bits

SRAM Burst R/W Cycle

The speed of the SRAM burst read/write cycles. The lower figure is fastest.

SYNC SRAM Support

If synchronous cache memory is installed, this setting allows you to specify whether it is the standard synchronous or less expensive pipelined SRAM.

L2 (WB) Tag Bit Length

See *L2 Cache Tag Bits*. For 8-bit, *Enhanced Memory Write* must be disabled.

DIRTY PIN SELECTION

When *Combine* is selected above, this chooses which pin the dirty data is tied to.

- *I/O*. Bi-directional input/output (default)
- *IN*. Input only

Shortened 1/2 CLK2 of L2 cache

Working on this.

VESA L2 Cache Read

See *VESA L2 Cache Write*.

1MB Cache Memory

Informs the system that a larger than usual L2 cache is present.

Cache Memory Data Buffer

Activate half T state earlier when a cache hit is made during a read cycle. Enable if your system runs faster than 33 MHz.

Cache Cycle Check

L2 cache checkpoint for hit or miss.

Pipeline Burst Cache NA#

With pipeline burst cache in the L2 cache, or L2 cache is disabled, enabling this may improve performance. *NA#* means assertion next address.

Cache Read Pipeline

Disable for stability, enable for performance (FIC board, VIA MVP3 chipset).

MEMORY

RAM is organised into rows and columns, and is accessed by electrical signals called strobes, which are sent along rows to the columns; when data is needed, the CPU activates the RAS (Row Access Strobe) line to specify the row where data is to be found (high bits), then, after a short time, the CAS, or Column Access Strobe, to specify the column (low bits). Predictably, the time between the two is called the RAS to CAS Delay, which can be two or three cycles long. During that time, a row's worth of data is selected and moved towards sense amplifiers over data lines, where it is latched, or fixed in place, with an internal timing signal.

Then the read command is issued with the address of the column that contains the first word, after which there is another delay, called CAS Latency while the data heads towards the output pin. This can be another 2 or 3 cycles long. Another word is pumped out for every subsequent cycle until the transfer is complete (i.e. burst transfers). Around half of any data requests are found in the same row (called a page hit), which means that you don't need to use all the signals again if it is in page - in other words, you only need to change the column address with CAS, which all of a sudden assumes some importance. If the data is in a different row, the whole bank needs to be closed and reactivated.

After all that, the data is put back where it came from, using up more cycles, and you might get even more delays if the contents need refreshing. The Bank Cycle Time refers to the clock cycles needed after a bank activate command before a precharge can occur or, inother words, the minimum time it must stay open. For most high-end SDRAM, this is about 50-50 ns, which explains one limitation on memory speed, since, if you increase the clock speed, you must increase the cycle speed as well.

The reason why a minimum time is allowed is because the signals are quite weak and need to be amplified, and cells need to be recharged. Data goes to the sense amplifiers over bitlines, of which there are two - one for data and one for the reference. The difference in voltage between them is sensed and amplified, and restored in the cell it came from. Data is lost if a precharge occurs before restoration takes place. The precharge wipes the bitlines and closes the bank. The combination of RAS and CAS therefore specifies a particular location in a particular RAM chip, where they intersect. Unfortunately, as is seen above, a lot of time is used transferring these values rather than data, and it follows that best performance will be obtained by shortening the latency (and precharging) times as much as possible, always bearing in mind that there is a minimum below which you cannot go because you will start to lose data. This, of course, is the same effect as overclocking, so it also follows that the better the material you have, the more successful you will be. In short, cheap chips won't cut it.

Rather than have separate pins providing power and data for CAS and RAS, each pin does double duty, serving rows or columns according to which pin is being asserted (that is, receiving current). With page mode, any column of DRAMs in a row (page) can be accessed any number of times within a short period; since the row is already specified, only the CAS needs to be applied on subsequent memory accesses, making things quicker.

RAS and RAS-to-CAS are usually set to 2 or 3 with SDRAM Cycle Length, although you may be able to set them independently, and preferably in the reverse order to the above. Numbers on the chip looking like 3-2-2 refer to CAS, RAS-to-CAS and RAS, respectively.

Anyhow, with PC100 SDRAM, the first transfer takes about 50 ns, and the remaining three inside one cycle, assuming burst mode is active and they are in the same column.

RAS and CAS are measured in nanoseconds; the lower the value, the faster the RAM can be accessed, so the *T state delay* is similar to wait states. The RAS access time is actually the speed rating marked on the chip; CAS access time is around 50% less. Generally, choose the same speed for reading and writing, with as few wait states as possible. To get the maximum theoretical speed of any memory, divide 1000 by the access time, thus 1000/7=143 MHz.

Burst cycles work the same way as they do for SRAM, consisting of four figures, with the first being larger because that's where the address is read; the remaining figures indicate the clock cycles for the reading of data. They might look like this on the screen:

```
x222/x333
```

In a typical BIOS setting, the first set would be for EDO and the second for Fast Page Mode RAM. The 430 HX chipset can use lower figures than the VX. The idea is to keep the figures as low as possible, consistent with your machine working properly. Note that EDO is only faster when being read from; writes take place at the same speed as FPM RAM.

Read requests that fall in the same row are known as page hits, which are good for the machine because some of the command signals can be eliminated. In fact, page hits occur about 50% of the time, so CAS Latency is the signal to concentrate on. If a page hit does not happen, though, the data must be moved back to where it came from and the bank closed so a new search can start, in which case you must also look at RAS to CAS Delay, which is often accessible under *Bank X/Y DRAM Timing*, below.

Note that many settings below, while referring to modern memory such as DDR, SDRAM, etc., are really hangovers from EDO and FPM, which are not supported anyway because they require higher voltages than newer power supplies can provide. The point about this is that faster settings may be suggested than should be, and adjusting one setting may change several parameters in the background.

If you get blue screens of death, it's certainly worth checking that you're not running the memory too aggessively. Some VIA-based motherboards with 4 DIMM slots have slots 1 & 2 and 3 & 4 sharing a voltage regulator, so if you have only two DIMMs, put them in 1 & 3 to decrease their load and give them the best chance (raising the voltage slightly always helps a marginal component).

Bank X/Y DRAM Timing

An older name for *RAS to CAS Delay*. It is actually a mixture of several settings, including bank interleaving. The selections are SDRAM 8-10ns, Normal, Medium, Fast and Turbo. However, only Normal and Turbo seem to make a difference, with VIA chipsets, anyway, in which the former enables 4-way bank interleaving and the latter reduces RAS to CAS delay down to 2T.

SDRAM SRAS Precharge Delay: tRP

The number of cycles needed to move data back to where it came from to close the bank or page before the next bank activate command can be issued. In short, how long it takes to switch between memory rows.

SDRAM Addr A Clk Out Drv

Believed to have something to do with the drive strength of the output clock on a memory bank. Set high for stability.

SDRAM Addr B Clk Out Drv

See above.

SDRAM CAS/RAS/WE CKE Drv

Believed to have something to do with drive strength. Set high for stability.

SDRAM DQM Drv

Believed to have something to do with drive strength. Set high for stability.

SDRAM TRC

Bank cycle timing, or the minimum cycles between consecutive activations of the same bank. The shorter the better for performance, at the possible expense of stability.

SDRAM TRP SRAS Precharge

The delay from the precharge command to the bank activate command. The shorter the better for performance, at the possible expense of stability.

SDRAM TRAS Timing

The minimum bank active time.

SDRAM Trrd Timing Value

See Act Bank A to B CMD Delay.

SDRAM CAS Latency

Controls the time delay (in CLKs) before SDRAM starts a read command after receiving it. Because reading data in a row is twice as fast, reducing this can help quite a bit (at the expense of stability), but the higher it is, the faster you can run the machine, if the memory is capable.

CAS Latency is actually column access time (tCAC) divided by clock cycle time (tCLK), rounded to the next higher whole number. An access time of 20 nanoseconds with a cycle time of 10 nanoseconds (for a 100 Mhz system bus - clock cycle time is the inverse of the bus speed) would therefore produce a CAS Latency of 2.

When the CPU needs data, it sends a signal saying what memory bank and row it is interested in, followed by the column it needs, after a specific period of time, which is variable according to the quality of the memory chips (the gap is called the column access time). Then the data moves to the output line, from where it is transferred with the next clock tick. One clock tick takes 10 seconds on a 100 MHz bus, including the time required for output (tAC) of 6 ns, plus 2 ns to stabilise and 2 for transfer time, as described above.

SDRAM modules are usually defined by three numbers, such as 2-2-2 or 3-2-2, which mean different things according to the bus speed. The first number refers to CAS Latency, the second to tRP (switching between banks), and the third to tRCD (between RAS and CAS access). The difference between a CAS Latency of 3 or 2 is probably not noticeable when using normal office applications.

SDRAM TRCD

The time between a bank activate command to accepting a read or write command. The shorter the better for performance, at the risk of stability.

SDRAM Trcd Timing Value

See above.

Super Bypass Mode

This allows the *Memory Request Organiser* (MRO) to skip pipeline stages when transferring data, but only if one processor is installed (some dual boards may allow this with one CPU), running at more than 4 times its bus speed.

Super Bypass Wait State

When enabled, this forces a delay for Super Bypass requests (see above), to improve stability. Officially, this seems to be 1 for 133 MHz (none for 100 MHz), but try 0 first.

Write Data In to Read Delay

Controls the Twtr parameter - the minimum cycles that must occur between the last valid write and the next read to the same internal bank of the DDR device. It only applies to reads that follow writes. Lower figures give better performance at the expense of stability.

Write Recovery Time

Twtr again - but, this time, the delay between a write and subsequent precharge to the same internal bank, or when it can start the precharge. Lower figures give better performance at the expense of stability.

DRAM Read/Write Timing

See below.

RAS# To CAS# Delay

Adds a delay between the assertion of RAS# and CAS#. In other words, this allows you to set the time it takes to move between RAS and CAS, or insert a timing delay between them. Reads, writes or refreshes will therefore take slightly longer, but you get more reliability.

Memory Read Wait State

You can use slower DRAMs by inserting wait states (e.g. use 1 wait state for chips rated at 80ns at 33 MHz). This setting concerns the number of wait states inserted between DRAM write operations.

Memory Write Wait State

As for *Memory Read Wait State* (above).

DRAM Read Wait State

As for *Memory Read Wait State* (above).

Add Extra Wait for RAS#

Same as below.

Add Extra Wait for CAS#

Same as below.

DRAM (Read/Write) Wait States

For older machines, this sets the cycles the CPU should be idle for whilst memory is being refreshed, such as 1 W/S for 80 nanosecond DRAMs (for 40 MHz machines, 2 is suggested). This won't affect performance with internal or external cache memory. A rule of thumb is:

```
Wait States = ns + 10 x Clock Speed
1000 -2
```

So:

```
.97 = 80 + 10 x 33
       1000 -2
```

gives you (almost) 1 wait state for 80 ns RAM at 33 MHz. For clock-doubled CPUs, you should use the motherboard speed. The chart below should be a useful starting point:

CPU	Write	Read	Speed (ns)
386DX-25/33/40	1	2	80
	0	1	70
	0	0	60
485-20/25	0	2	80
	0	1	70
	0	0	60
486DX-33/DX2-50	1	2	80
	0	1	70
	0	0	60
486DX-50/DX2-66	1	3	80
	0	2	70
	0	1	60

Actually, wait states are *additional* to those built in by the manufacturer. 0 wait states probably means 6, so 1 would mean you get 7. Each wait state adds about 30ns to the RAM access cycle here. Theoretically, 9-chip 30-pin SIMMs are faster, because it can be marginally longer getting data from the 4-bit chips on the 3-chip variety. Windows has been known to work with less GPFs with 9-chip SIMMs. Certainly, never mix in the same bank.

DRAM Burst Write Mode

Enabled is best for performance.

DRAM Read Burst Timing

Of burst data transfers to and from DRAM. Similar to Cache Read Hit Burst. With EDO, select *x222* for best performance.

DRAM Write Burst Timing

See *FP Mode DRAM Read WS*.

DRAM Timing Option

See *DRAM Speed*.

EDO:SPM Read Burst Timing

Adjusts the read wait state for EDO and SPM (Standard Page Mode) DRAM. Every time the CPU reads an L2 cache miss, it reads four continuous memory cycles on four continues addresses from the EDO and SPM cache, so it has four settings to adjust.

FP Mode DRAM Read WS

This configures the exact timing of the read cycle from Fast Page (FP) mode memory. The timing consists of an address cycle, where the location of the read to take place is indicated, and three data cycles, where the data is actually read. The shorter each phase (or cycle) is, the better the performance, but you will lose data if you don't allow enough time for each cycle. Choices are:

- 7-3-3-3
- 7-2-2-2
- 6-3-3-3
- 6-2-2-2 (default)

Try the lowest figures first till your machine is running successfully.

DRAM Timing

The speed of the RAM in your system. With Award, the choices are 60 or 70 ns. What you set here affects the settings for Auto Configuration.

DRAM Timing Selectable

This item sets the optimal timings for the following four items, depending on the memory module. The default *By SPD* configures them by reading the contents in the SPD (Serial Presence Detect) device. The EEPROM on the memory module stores critical parameter information, such as memory type, size, speed, voltage interface, and module banks.

DRAM Post Write

An Award setting. Still working on it, but see *Posted Write Enable*.

DRAM Read/FPM

Sets the timing for burst mode reads according to your type of memory, EDO or Fast Page Mode. With EDO, select x222 for best performance.

Fast DRAM

The system expects memory to run at the fastest speed-if you have mixed speed SIMMs, you might experience data loss. Disable this to use slower timing for all access to DRAM.

DRAM Write Burst (B/E/P)

See *DRAM Read Burst (B/E/P)*, below.

DRAM Read Burst (B/E/P)

The timing for burst mode reads from DRAM, depending on the type on a per-row basis (Burst/EDO/Page) The lower the timing numbers, the faster the system addresses memory, so select higher numbers for slower memory. With EDO, select x222 for best performance.

DRAM Write Page Mode

Enabled, RAS is not generated during a page hit in page mode, so a cycle is eliminated and makes things faster as more data is written at once.

DRAM Last Write to CAS#

Sets how much time (or how many cycles) elapse between the last data signalled to when CAS# is asserted. This is used as setup time for the CAS signal. Choices are 2 (default), 3 or 4.

DRAM Code Read Page Mode

Affects access speeds when program code is being executed, based on its sequential character, so enabling page mode here will be more efficient, to allow the CPU to access DRAM more efficiently during read cycles. If code is not sequential, you may be better off without this.

MD Driving Strength

Related to *DRAM Read Latch Delay*, and concerns the signal strength of the memory data (MD) line, with higher values giving stronger signals to cope with heavy DRAM loading, or to increase stability with overclocking.

DRAM Read Latch Delay

Provides a small delay before data is read from a module, to allow for those with strange timing requirements, or for varying DRAM loadings, where one single sided DIMM provides the lowest. Normally, disable unless you experience odd crashes.

Delay DRAM Read Latch

Similar to *DRAM Read Latch Delay* (above). *Auto* lets the BIOS decide for itself, but you might need your own delay if you have lots of double sided DIMMs producing a heavy loading. Longer delays decrease performance so use the lowest value that works. *No delay* is fastest.

Page Code Read

See *DRAM Write Page Mode.*

Page Hit Control

For testing the controller.

DRAM Precharge Wait State

Use 0 for 60-70 ns and 1 for 70 ns DRAM.

DRAM Wait State

Same as above.

DRAM RAS# Precharge Time

See also *FP DRAM CAS Prec.* Timing. The CPU clocks allocated for the RAS# signal to accumulates its charge before DRAM is refreshed. If this time is too short, you may lose data.

DRAM Speed

Set CPU speed instead of tinkering with RAS/CAS timings (these are for 100ns chips; push it a bit with faster ones). There may also be a *Normal* setting, which seems to be automatic.

- *Fastest*. 25 MHz (25/33 with Award)
- *Faster*. 33 MHz (40/50 with Award)
- *Slower*. 40 MHz
- *Slowest*. 50 MHz

Here's a comparison chart that may give you a good start:

CPU	DRAM Speed	Write CAS Width	Cache Write	Cache Read	BUSCLK
486SX-20	Fastest	1T	2T	1T	1/5
486SX-25	Fastest	1T	2T	1T	1/3
486DX2-50	Fastest	1T	2T	1T	1/3
486DX-33	Faster	1T	3T	2T	1/4
486DX2-66	Faster	1T	3T	2T	1/4
486DX-50	Slowest	2T	3T	2T	1/6

Notice that the higher the chip speed is, the more the wait states. Turbo mode reduces CAS access time by 1 clock tick.

DRAM Timing Control

See above. Selections are *Fast*, *Fastest*, *Normal* (default) and *Slow*.

DRAM to PCI RSLP

When enabled, the chipset allows the prefetching of two lines of data from memory to the PCI bus.

FP DRAM CAS Prec. Timing

The number of CPU clock cycles for CAS to accumulate its charge before FP DRAM is allowed to recharge. The lower figure is best for performance, but if you don't allow enough time, you could lose data.

FP DRAM RAS Prec. Timing

See *FP DRAM CAS Prec. Timing*.

CAS Low Time for Write/Read

The clock cycles CAS is pulled low for memory operations, based on memory timing.

DRAM CAS# Hold Time

Sets the number of cycles between when RAS# is signalled and CAS# is asserted. Choices are 4, 5, 6 (default) and 7.

CAS Address Hold Time

Sets how long it will take to change the CAS address after CAS has been initiated (asserted) and aimed at a target address (location) in DRAM. Choices are 1 or 2 (default) cycles.

Read CAS# Pulse Width

How long the CAS remains asserted for a DRAM read cycle. Choices are 2, 3 (default), 4 or 5 cycles. The same effect as wait states.

Write CAS# Pulse Width

How long the CAS remains asserted for a DRAM write cycle. Choices are 2 (default), 3, 4 or 5 cycles. The same effect as wait states.

CAS Read Pulse Width in Clks

Essentially the same as *DRAM Read Wait States*, except that the value is 1or 2 more than the number of Waits. The fewer the better.

DRAM RAS# Pulse Width

The number of CPU cycles allotted for RAS pulse refresh.

DRAM RAS Precharge Time

Controls the memory timing by setting the number of cycles the RAS needs to accumulate its charge before SDRAM refreshes. Reducing this too low affects the ability to retain data.

Write Pipeline

Enable when PBSRAMs are installed.

RAMW# Assertion Timing

RAMW is an output signal to enable local memory writes. The difference between Normal or Faster is one timer tick.

EDO CAS Pulse Width

The number of CPU cycles the CAS signal pulses during EDO DRAM reads and writes, when memory is not interleaved.

EDO CAS Precharge Time

See *FPDRAM CAS Prec. Time.*

EDO RAS Precharge Time

The number of CPU clock cycles for RAS to accumulate a charge before EDO DRAM can recharge. The lower figure is best for performance, but if you don't allow enough time, you could lose data.

EDO RAS# to CAS# Delay

Enabled, adds a delay between the assertion of RAS# and CAS# strobes (slower but more stable). *Disabled* gives better performance.

EDO RAS# Wait State

Inserts one additional wait state before RAS# is asserted for row misses, allowing one extra (MAX 13:0) clock of MA setup time to RAS# assertion. Only applies to EDO memory.

EDO MDLE Timing

Memory Data Read Latch Enable timing when EDO is read. Sets the CPUCLK signal delay from the CAS pulse. 1 is fastest, but 2 is more stable.

EDO BRDY# Timing

When the Burst Ready Active signal is low, the presented data is valid during a burst cycle. 1 is fastest, 2 is more stable.

EDO RAMW# Power Setting

RAMW# is an active low output signal that enables local DRAM writes. This setting lets you enable RAMW# power-saving mode when an EDO bank is being accessed.

EDO DRAM Read Burst

The timing you set here depends on the type of DRAM you have in each row. Use slower rates (bigger numbers) for slower DRAM.

EDO DRAM Write Burst

The timing you set here depends on the type of DRAM you have in each row. Use slower rates (bigger numbers) for slower DRAM.

EDO Read Wait State

Use this only if your system has EDO (Extended Data Out) DRAM, to configure the exact timing of the read cycle. The timing is composed of an address cycle, for the location of the read, and three cycles where the data is actually read. The shorter each phase (or cycle) is, the faster the system is operating, but if not enough time is allowed for each cycle, data will be lost. Choices are *7-2-2-2* (default) and *6-2-2-2*.

EDO read WS

See above.

DRAM CAS Timing Delay

Sets *No CAS delay* (default) or *1 T state delay*. Use this only if you're using slow DRAMs. It's often ignored anyway if cache is enabled.

EDO Back-to-Back Timing

The number of timer ticks needed for back-to-back accesses, depending on your memory. (SiS). Back-to-Back means that address reads can alternate with page hits, which means that data transfer happens at half-speed, but also with very low latencies, zero in some cases.

DRAM RAS# Active

Controls whether RAS# is actually activated after CAS; Deassert means not, which increases performance by saving a CPU cycle. The latter makes each DRAM cycle a Row miss.

- *Assert*. Wwill be asserted after every DRAM cycle
- *Deassert*. Will be deasserted after every DRAM cycle

DRAM R/W Burst Timing

Allows DRAM read and write bursts to have their timings coordinated. These are generated by the CPU in four parts, the first providing the location, and the remainder the data. The lower the timing numbers, the faster memory is addressed.

- *X444/X444*. Read and write DRAM timings are X-4-4-4
- *X444/X333*. Read timing = X-4-4-4, write timing = X-3-3-3
- *X333/X333*. Rread and write DRAM timings are X-3-3-3

Try the lowest figures first, until your machine is running successfully.

Fast EDO Path Select

When enabled, a fast path is selected for CPU-to-DRAM read cycles for the leadoff, assuming you have EDO RAM. "It causes a 1-HCLK pull-in for all read leadoff latencies" (that is, page hits, page and row misses). Enabled is best. Possibly the same as Fast EDO Leadoff. See also *Read/Write Leadoff*.

RAS Precharge Time

The Row Access Strobe is used to refresh or write to DRAM. The precharge time is the time taken for internal recovery of the chip before the next access, or when the system gets up enough power to do the refresh, about the same as the RAM access time, so use that as an estimate to start off with. If there is not enough time, you won't get a proper refresh, and you may lose data.

This determines the number of CPU clocks for RAS to accumulate a charge before DRAM is refreshed. If you have a 33 MHz CPU or higher, set this to 4, but try a lower number if your CPU is slower (e.g. 2 for 25 MHz, so as not to waste time), reducing idle time, unless your DRAMs can't operate with a lower figure anyway. Often ignored if cache is enabled.

RAS Precharge Period

See above.

RAS Precharge @Access End

Enabled, RAS# remains asserted at the end of access ownership. Otherwise, it is deasserted.

RAS Precharge In CLKS

An Award Setting. Sets the length of time required to build up enough charge to refresh RAS memory. Choices are 3, 4, 5 or 6. Lower figures are best for performance.

CAS Precharge In CLKS

An Award Setting. As above, but for CAS.

CAS# Precharge Time

How long (in CPU clocks) the CAS# signal is allowed to accumulate its charge before refresh. If this is too short, you may lose data.

CAS# width to PCI master write

The pulse width of CAS# when the PCI master writes to DRAM. Lower figures are best for performance.

RAS Active Time

Controls the maximum time that DRAMs are kept activated by increasing the Row Access Strobe (RAS) cycle, meaning that a row can be kept open for more than one access, allowing more column access in that time. The higher the figure, the better the performance.

Row Address Hold In CLKS

An Award setting, for the length of time in CPU cycles to complete a RAS refresh. A CLK is a single CPU clock tick, so the more you use here, the slower your machine will perform.

RAS Pulse Width In CLKS

The length of the RAS pulse refresh. Choices are between 4-6 CLKs, and the higher the number, the slower your machine will be.

RAS Pulse Width Refresh

The number of CPU cycles allotted.

CAS Pulse Width

The duration of a CAS signal pulse in timer clicks.

CAS Read Width In CLKS

An Award Setting. Sets the number of CPU cycles required to read from DRAM using Column Address Sequence (CAS) logic. Choices are 2 or 3.

CAS Write Width In CLKS

Award Setting. As above, for write cycles.

RAS to CAS delay time

The amount of time after which a CAS# will be succeeded by a RAS# signal, or the time delay between Row Address Strobe and Column Address Strobe, to allow for the transition. Performance is best with lower figures at the expense of stability.

RAS(#) To CAS(#) Delay

As for *RAS to CAS delay time*. When DRAM is refreshed, rows and columns are addressed separately. This allows you to set the time to move between RAS and CAS, or insert a timing delay between them, in CPU cycles. The shorter the better for performance.

- *2T.* Two cycles
- *4T.* Four cycles (Default)
- *6T.* Six cycles

RAS to CAS Delay Timing

See above.

RAS#-to-CAS# Address Delay

Inserts a timing delay from the time RAS# is asserted to when Column Address is asserted.

DRAM write push to CAS delay

The number of cycles needed by DRAM to force the CAS to slow down (delay) to match DRAM timing specifications.

CAS Before RAS

A technique for reducing refresh cycles, to help the CPU and power consumption. CAS is dropped first, then RAS, with one refresh performed each time RAS falls. Powersaving is from an internal counter, not an external address, so the address buffers are powered down.

Late RAS Mode

Controls the generation of an earlier RAS signal during memory accesses, extending the length of the RAS signal for slower Tag RAM. It could also mean *RAS after CAS* (see below).

RAS Timeout Feature

For DRAMs that need a 10 microsecond maximum RAS-active time. If timeout is enabled, RAS is not allowed to remain low for longer than about 9.5 microseconds. Otherwise, it is limited to a maximum of about 15 microseconds. This affects reliability - *Disabled* is default.

RAS Timeout

See above.

Turbo Read Leadoff

Sometimes needed for faster memory, and disabled by default. When Enabled, the BIOS skips the first input register in the DRAM when reading data, speeding up the read timings. In other words, it shortens the leadoff cycles and optimizes performance in cacheless, 50-60 MHz, or 1-bank EDO systems, but it is known to speed up those with a 512K Level 2 Cache and 2 banks of EDO (2X16, 2X32 Mb SIMMs), especially when copying data, such as when backing up a hard drive. However, after a few hours of use, errors start in applications and when loading data from the hard drive, especially when switching between applications. Suggest enable this for games, but disable otherwise. See also *Read/Write Leadoff*.

CAS Width in Read Cycle

Determines the number of wait states when the CPU reads data into the local DRAM, in T states. The lower the figure, the better the performance.

Read-Around-Write

As data can only be transmitted in one direction at a time along the memory path, write commands interrupt reads in progress. Although they are a relatively small part of the total amount of transactions, their effect is disproportionate, so writes can be held in a buffer and transmitted as a burst to minimise their transmission time. This effect is also a sort of mini-cache, in that the processor can execute read commands out of order if there is independence between them and other write commands. In other words, if a memory read is addressed to a location whose latest write is in a buffer before being written to memory, the read is satisfied from the buffer instead of memory, as the information will be more up to date.

This is very useful for multi-processor systems using the AMD 762 NorthBridge, as several CPUs could snoop or share data without accessing main memory.

DRAM Read-Around-Write

See above.

OMC Read Around Write

Similar to the above, enabling the memory controller on an Orion chipset to let read operations bypass writes as long as their memory addresses don't match. In other words, priority is given to reads, except when they have the same address as a write, in which case the write is done first so the read gets the most up to date information. Found on a Pentium Pro. Enabled increases performance slightly at the expense of some stability.

DRAM Write CAS Pulse Width

See *DRAM Head Off Timing*.

DRAM Head Off Timing

7/5 or 8/6. See *DRAM Leadoff Timing*.

Interleave Mode

Controls how memory interleaving takes place, or how DRAM access is speeded up because succeeding memory accesses go to different DRAM banks, and take place while another is being refreshed (2- or 4-way interleave). Not always possible.

Bank Interleaving

When one bank of SDRAM is open, the memory controller can activate another bank. In other words, if it knows the next data needed is in a different one, it can issue read commands to the next location without ruining the first burst, so you can hop from one bank to another with only one penalty cycle (i.e. bank-to-bank latency) between four word bursts. As well, precharging and closing can run in the background.

For an application dependent on the CPU cache, this may actually cause a performance hit if a wrong bank is open and must be closed before the next access.

Extended Read Around Write

When Enabled, reads can bypass writes within the 82450GX memory interface component(s), provided their addresses do not match.

F000 UMB User Info

Found with MR, lets you know what's going on in the F000-FFFF range usually occupied by System ROM. The first 32K can often be used for UMBs as it is only used on startup.

```
BIOS   FC14-FFFF
UTILS  FBAA-FC13
POST   F787-FBA9
SETUP  F1C0-F786
AVAIL  F000-FBA9
```

The above is information fed to your memory manager so it can make the best use of what's available. You can't reassign the BIOS area, and you should leave the UTILS section alone, because various hot key and cache functions are kept there. POST and SETUP only contain power up and boot code.

Fast Page Mode DRAM

Should be enabled with DRAM capable of Fast Page Mode on your motherboard (not 256K SIMMs). Page Mode speeds up memory accesses when they occur in the same area; the page address of data is noted, and if the next data is in the same area, page mode is invoked to reduce the access time to about half (that is, the row and column need not be specified again, so the RAS or CAS lines don't need to be reset). Otherwise data is retrieved normally from another page. Fast page mode is a quicker version of the same thing. This technique is not necessarily the best for the PC; you may be better off adjusting the RAS values and extending the signal's length so that a row can be kept open for as long as possible.

Fast R-W Turn Around

Reduces the delay betweeen the CPU's first read from RAM and subsequent write - in other words, it reduces the switch time, or the wait states after a read until a write command is issued. Enabling increases performance at the risk of stability.

R/W Turnaround

See above.

Highway Read

If no operation is scheduled (NOP), the memory command bus is idle-parked. If disabled, it will be parked on CAS READ, which means zero latency on the next read.

DDR Read Path Short Latency Mode

Specifies the time when a read command can be issued during an ongoing burst.

Enhanced Page Mode

Enable or Disable, according to your memory.

Enhanced Memory Write

Affects the Memory Write and Invalidate command on the PCI bus. Disable if the cache size is 512 Kb and the tag address is 8 bits.

Page Mode Read WS

The cycle time combination.

Pipelined CAS

When enabled, the DRAM controller will not provide time between two successive CAS cycles. Otherwise, one Host Bus clock between successive CAS cycles will be provided (default). The former is best for performance.

*00 Write Protect

Normally, when a ROM is shadowed, the original ROM is disabled and the RAM area where its contents goes is write protected. You can disable this for special reasons, such as debugging ROM code, but very little else. Normally, leave enabled.

Parity Checking Method

You can check parity for every double word, or only the last double word during cache line fill. The Triton chipset does not support parity.

Parity Check

Enabled on a Phoenix BIOS, an NMI interrupt is produced with a parity error.

Memory Parity Check

Enable if you want to use parity, though your DRAM must support it.

Base Memory Size

You might want to disable on-board RAM (i.e. base memory) between 80000-9FFFF (512K-640K), so you can give 128KB of contiguous address space to cards that need it (it is not normally available in upper memory). Normally set at 640, but set 512K for such a card.

Memory Parity/ECC Check

To enable memory checking when ECC or parity-equipped RAM is installed, as appropriate.

F/E Segment Shadow RAM

How the E/F segments of Upper Memory are used (in cacheing). Choices are:

- *Disabled*. (E segment default)
- *Enabled*. (F segment default)
- *Cached*. L2 cache
- *Into-486*. L1 cache

Disable Shadow Memory Size

Sets a shadow memory size for *Disable Shadow Memory Base*, below. It doesn't disable anything.

Disable Shadow Memory Base

Alters the location of non-shadowed memory, e.g. if using a SCSI host adapter, set this to the address of the adapter and the size to 16K (see below).

Memory Remapping (or Relocation/Rollover)

The memory between A000-FFFF (that is, the 384K of upper memory normally for ROMs, etc) can be remapped above the 1 Mb boundary for use as extended memory-this is sometimes not available with more than 1 Mb installed. Thus, your memory will run from 0-640K and 1-1.384Mb if you have 1 Mb. You usually have the choice of moving 256K (areas A, B, D and E) or 384K (Areas A-F), if no ROMs are shadowed. Relocated memory blocks must not be used for Shadow RAM, so relocating the full 384K means no Video or System BIOS Shadow! What you get from this depends on the total memory you have, and whether you use DOS or Windows. Use mostly when memory is tight. More precise control may be obtained from a memory manager.

384 KB Memory Relocation

See *Memory Remapping*. Can solve problems if you have more than 16 Mb.

256 KB Remap Function

See *Memory Remapping*.

RAM Wait State

Allows an additional T-state (2 PROCCLK cycles) to be inserted on local memory accesses during CAS active interval, extending the width of the CAS pulse, and slowing the machine.

Memory Reporting

You get the choice of *Standard* or *Windows NT*, for getting around the limitations imposed by the ISA bus on the amount of memory the CPU can address. The 16-bit ISA bus has 24 address lines, which means it can theoretically see only 16Mb.

Global EMS Memory

Whether expanded memory is used or present. If disabled, this is ignored:

- EMS I/O port access. Enable if using EMS.
- EMS Page Registers. Accessed through 3 I/O ports at: EMS 0 (208, 209, 20Ah), the default, or EMS 1 (218, 219, 21A)

DRAM Relocate (2, 4 & 8 M)

Remaps 256K of upper memory to the top of DRAM size. Only applies when the D and E segments are not shadowed, and with 2, 4 or 8 Mb of on-board memory.

Extended Memory Boundary

Where extended memory ends, and expanded memory begins.

Shared Memory Size of VGA

System memory to be allocated to VGA in a shared memory system (see Memory).

Shared Memory Enable

Enable or Disable.

Cycle Check Point

This allows you to select how much time is allocated for checking memory read/write cycles. In effect, each selection sets a predetermined wait state for decoding cycle commands.

- *Fast.* 0, 1 waits (Default)
- *Fastest.* 0, 0 waits
- *Normal.* 1, 2 waits
- *Slow.* -, 3 waits

VGA Shared Memory Size

The size of system memory allocated to video memory, 512K-4Mb.

Cycle Early Start

Allows read/write cycles to start half a clock cycle early, assuming addresses and other control signals are stable. Enabling this may eliminate a wait state.

MA Timing Setting

MA = *Memory Access*. Disable with EDO RAM. Also set *CAS Pulse Width* and *precharge* to 1T.

MA Additional Wait State

Enabled, inserts an extra wait state before the assertion of the first MA (Memory Address) and CAS#/RAS# during DRAM read or write leadoff cycles. This affects page hits, row and page misses. In English, inserts an additional wait state before the beginning of a memory read. Always use the default unless you are getting memory addressing errors. See also *Read/Write Leadoff*.

EDO CAS# MA Wait State

Similar to above. It puts in an additonal wait state before the assertion of the first CAS# for page hit cycles, allowing it an extra clock of memory address (MA) setup time for the leadoff. This is only for EDO memory and only needs changing for memory addressing errors.

DRAM Fast Leadoff

Select Enabled to shorten the leadoff cycles and optimize performance - the system will reduce the number of clocks allowed before reads and writes to DRAM are performed. See also *Read/Write Leadoff*.

Reduce DRAM Leadoff Cycle

Enabling this optimises DRAM performance by shortening the time before memory operations, assuming the DRAM supports it.

DRAM Read Pipeline

Disable for stability, enable for performance. AOpen, VIA MVP3 chipset.

Read Pipeline

Pipelining improves system performance. Enable this when you have PBSRAMs installed.

DRAM R/W Leadoff Timing

Sets the CPU clocks before reads and writes to DRAM are performed, or the combination of CPU clocks your DRAM requires before each read from or write to the memory. Similar to cache burst timings, but reads 7-3-3-3 or similar for 50 MHz. The higher the first figure, the less the performance. EDO RAM uses one less wait state. The 430 HX chipset can use lower figures than the VX.

- *8/7*. 8 clocks leadoff for reads and 7 for writes.
- *7/5*. 7 clocks leadoff for reads and 5 for writes.

See also *Read/Write Leadoff*.

DRAM Leadoff Timing

See *DRAM R/W Leadoff Timing*. This is the AMI version and the settings are:

```
8-6-3  7-5-3  8-6-4  7-5-4
```

The Award one selects the combination of CPU clocks your DRAM requires before each read from or write to the memory. Changing the value from that set by the board designer may cause memory errors. See also Read/Write Leadoff.

MA Drive Capacity

Or *Memory Address Drive Strength*. Sets current draw of multiplexed DRAM chips. The smaller the number, the less power consumption, and therefore heat, but if set too low you need an extra wait state-too high and you get ringing and reflections, and errors (in PCs, the DRAM voltage can be nearly 6 volts because ringing and reflections can drive the +5 up, making the memory run hotter). Drive capacity of modern chipsets is limited because of the lack of memory buffer, to improve performance, so the DRAM chip count becomes important. If your SIMMs have a high loading, (that is, you have over 64 memory chips), select 16ma/16ma. The more chips, the higher the figure. The BIOS cannot count them for you.

Memory Address Drive Strength

See above.

Mem. Dr.Str. (MA/RAS)

As above - controls the strength of the output buffers driving the MA and BA1 pins (first value) and SRASx#, SCASx#, MWEx# and CKEx# (second value).

DRAM Speed Selection

Set the access speed of the memory in your system.

EDO Speed Selection

See above.

Fast EDO Leadoff

Select *Enabled* only for EDO in systems with either a synchronous cache or which are cacheless. It causes a 1-HCLK pull-in for all read leadoff latencies for EDO memory (that is, page hits, page and row misses). Disable for FPM or SDRAM. Possibly the same as Fast EDO Path Select. See also Read/Write Leadoff.

Speculative Leadoff

A read request from the CPU to the DRAM controller includes the memory address. When Enabled, Speculative Leadoff lets the controller pass the read command to memory slightly before it has fully decoded the address, thus speeding up the read process and reducing latencies, including the cache, DRAM and PCI. Disabled is the default. The "speculative" bit arises from the chipset's ability to process what might be needed in the future, or speculate on a DRAM read address, so as to keep the pipeline full. See also Read/Write Leadoff.

DRAM Speculative leadoff

See above.

SDRAM Speculative Read

As above.

DRAM Speculative Read

See above.

SDRAM Wait State Control

Inserts a wait state into the memory addres data cycle.

SDRAM WR Retire Rate

The timing for data transfers from the write buffer to memory.

USWC Write Posting

USWC stands for *Uncacheable Speculative Write Combination*. It may improve performance for some Pentium Pro systems using graphic cards with linear frame buffers (i.e. all new ones), but don't hold your breath. By combining smaller writes (bytes and 16-bit words) into 64-bit writes, you need fewer transactions to move data, but you might also get corruption or crashes. The separate settings for ISA and PCI apparently affect different memory regions. The older your chipset, the more chance there is of extra performance. See also *PCI Burst Write Combine* and *Video Cache Memory*.

This can cause video problems and/or intermittent crashes on many systems, including a conflict with sound cards on NT systems. Use the default NT sound driver and put the sound card on DMA channel 3, 16 bit DMA on 7; and set the BIOS DMA Type F Buffer to the floppy DMA channel.

USWC Write Post

See above. Enable for wite-back mode when video memory cache is set for USWC mode.

Video Memory Cache Mode

Video memory is not normally cached because the L2 cache would get filled up (there's a lot of data). In addition, 3D operations need to use the FPU (maths copro), which can then only be used by the CPU on every alternate cycle, since the other one is used for writing to graphics memory. The Pentium II has write combine buffers that allow single bit writes to be combined and sent in burst mode (the data has high locality), improving the CPU's graphics performance.

AMD, on the other hand, added two write combine buffers to the K6 II and K6 III (with the CXT revision), and proper addressing of them can boost graphics performance by an extra 30%. The Athlon has four 64-bit buffers that can be placed over the local frame buffers for data and hardware acceleration, that can do out of order writes in ascending and descending order, so 3D FPU operations can be almost doubled.

Choices on ASUS boards are UC (UnCacheable) or USWC (Uncacheable Speculative Write Combine). The latter gives better performance, but it needs support from the graphics adapter and drivers.

CPU Burst Write Assembly

The (Orion) chipset maintains four posted write buffers. Posted writes are write operations held until it is convenient to execute them-under normal circumstances, the buffers hold data destined for memory, but here you can use them to collect data for the PCI bus as well. When this is enabled, the chipset can assemble long PCI bursts, or sequential writes without wasting cycles posting addresses between words, which is best for performance. The default is Disabled.

OPB Burst Write Assembly

Similar to the above, found on a Pentium Pro machine. It relates to USWC (see below), which affects video cards. OPB may stand for Orion Post Buffers. Then again, it may not.

SDRAM Leadoff Command

Allows you to adjust the time before data in SDRAM can be accessed - it usually affects the first data element, which will contain the address of the data affected. The lower the number, the faster the performance at the expense of stability.

SDRAM (CAS Lat/RAS-to-CAS)

You can select a combination of CAS latency and RAS-to-CAS delay in HCLKs of 2/2 or 3/3. This sets up the SDRAM CAS latency time or RAS to CAS Delay. You will only see this if you have SDRAM installed. Usually set by the system board designer, depending on the DRAM installed. Do not change this unless you change the DRAM or the CPU, or you have instability problems.

SDRAM RAS to CAS Delay

You can insert a delay between the RAS (Row Address Strobe) and CAS (Column Address Strobe) signals when SDRAM is written to, read from or refreshed - in other words, this determines how quickly memnory is accessed. The lower the number, the faster the performance at the expense of stability.

SDRAM RAS Precharge Time

Controls the memory timing by setting the number of cycles the RAS needs to accumulate its charge before SDRAM refreshes. Reducing this too low affects the ability to retain data.

SDRAM Precharge Control

See above. If disabled, CPU cycles to SDRAM will result in an *All Banks Precharge* command on the SDRAM interface. *Enabled* is best for performance at the expense of stability.

SDRAM Page Closing Policy

Also known as *SDRAM Precharge Control* (above). It determines whether the processor or SDRAM controls precharging. The *All Banks* setting improves stability but reduces performance. With *One Bank*, precharging is left to SDRAM, reducing the number of times it is precharged, since multiple CPU cycles to SDRAM can occur before refresh is needed.

SDRAM CAS Latency Time

Optimises the speed at which data is accessed in a column by defining CAS latency time in 66 or 100 MHz clocks, depending on the memory bus speed - it controls the time delay (in CLKs) before SDRAM starts a read command after receiving it. Because reading data in a row is twice as fast, reducing this number can help quite a bit at the expense of stability, but the higher it is, the faster you can run the machine, if the memory is capable.

SDRAM RAS Latency Time

See above.

SDRAM Cycle Length

Similar to SDRAM CAS Latency Time, setting the number of CPU cycles between refreshes, or the time before a read command is actioned after being received (it also sets the number of clocks to complete the first part of a burst transfer). The Column Address Strobe dictates how many clocks the memory waits before sending data to its next destination. All registers should be full, or errors will result, which means a longer wait and slower operation. In other words, the shorter the cycle length, the faster the machine runs, at the expense of stability and data, although increasing this may help with overclocking, as it allows memory to run faster.

Linked with CAS are two other settings, *RAS* and *RAS-to-CAS*, usually set to 2 or 3 here, although you may be able to set them independently, and preferably in the reverse order to the above. Numbers on the chip looking like 3-2-2 refer to CAS, RAS-to-CAS and RAS, respectively. Running chips at higher than rated speeds will mean dropping a CAS/RAS level.

SDRAM Cycle Time Tras/Trc

Similar to the above, Tras is Row Active Time, or how long a row stays open for data transfers (also known as Minimum RAS Pulse Width). Trc is Row Cycle Time, or how long the row-stays open for refresh purposes. The fastest setting is 5/6, but 6/8 is more stable. As always, try the fastest first, and revert to the other if your system becomes unstable.

DRAM Cyle Time

The wrong name for *CAS Delay*, or *CAS Latency*.

DRAM Read Latch Delay

If the memory clock frequency is increased with the VIA chipset, the cycle time gets shorter, so the data valid window (tDV), that is, time in which the chipset can receive data from DRAM, would come earlier and may expire before data from the DIMMs actually arrives, so this setting delays or moves it further back on the cycle.

Bank cycle time tRC (SDRAM active to precharge time), tRAS

The clock cycles needed after a bank active command, before precharging takes place. The minimum time a page must be open before it can be closed again is specified by tRC (bank cycle time), which is the sum of tRAS (time needed to develop a full charge and restore data in memory cells) and tRP (RAS precharge time), assuming precharging has a latency of 2 or 3 cycles. tRP in the i815 chipset is fixed at 2T. VIA chipsets allow 2 and 3 cycles.

For 100 MHz memory, set 5/7; for 133 MHz, try 5/8 or 6/8.

tRCD is the RAS-To-CAS Delay, or the minimum time between a bank activate command and a read. tDPL is the Data Phase Latency, or turnaround between the last Write Data Phase and a precharge command. Also known as tWR or write-to-read interval.

SDRAM Bank Interleave

Supports interleaving SDRAM banks, for better performance. Use 2- or 4-bank interleave for 64 Mb SDRAM. Otherwise disable, especially for 16 Mb DIMMs. See also....

DRAM Interleave Time

Sets the additional delay between accesses when SDRAM Bank Interleave, above has been enabled. The shorter the better, but watch for memory errors.

Force 4-Way Interleave

Enable for best performance, but you must have over 4 banks for it to work. Banks do not equal the number of DIMMs, as one DIMM can have many banks. Normally, 2-bank DIMMs use 16Mbit chips and are less than 32 Mb. 4-bank DIMMs usually use 64Mbit chips, with a density up to 256Mbit per chip. All DIMMs over 64 Mb are 4-banked.

CAS Latency

Optimises the speed at which data is accessed in a column by defining the time delay (in CLKs) before SDRAM starts a read command after receiving it.

Row Precharge Time

The number of cycles RAS for SDRAM takes to precharge. If too short, you may lose data, but you only have two choices anyway.

Configure SDRAM Timing By

From the AMI BIOS, this is similar to the above. Setting to SPD allows CAS Latency, Row Precharge Time, RAS Pulse Width, RAS to CAS Delay and Bank Interleave (see below) to be automatically determined. Otherwise, you can do this manually with the *User* setting.

SDRAM Configuration

Either Disabled or By SPD. SPD (*Serial Presence Detect*) refers to a little EEPROM on the DIMM that holds data relating to its performance, which is checked during startup to match timings, required for the PC100 standard as things are a little tight at that speed. In other words, it talks to the BIOS to coordinate memory timings between main memory and L2 cache as, although the two systems may be running at the same frequency, there may still be a mismatch. Do not accept its findings as gospel - the EEPROM is not write protected and can be overwritten with false specifications. In addition, if the manfacturer is unrecognised, you will get the slowest settings anyway, and, very often, when the manufacturer is recognised, good parameters are assumed without checking. See also.....

RAS Pulse Width

The clock cycles allotted for RAS. Again, only two choices, but the lower the number that works, the better the performance.

SDRAM Frequency

HCLK means the same as the Host Clock, HCLK +33 means the Host Clock plus 33 MHz, HCLK -33 is self-explanatory (the last two depend on what CPU is present - for example, you will only see the -33 setting if your FSB is running at 133 MHz). SPD means the details will be read from the SPD device on the DIMM.

Burst Length

Bursting, where memory is concerned, allows DRAM to predict for itself the address of the next memory location after the first has been found. However, the burst length must be determined first (the larger the better for performance), which consists of the data plus the starting address. This allows the internal counter to generate the next location properly.

SDRAMIT Command

Controls the *SDRAM command rate* (see below). *Enabled* allows the SDRAM signal controller to run at 1T (that is 1 clock cycle). Otherwise, it runs slower, at 2T.

SDRAM WR Retire Rate

The number of clocks required to assert the SDRAM Write Retire Rate.

Special DRAM WR Mode

Enables a special inquiry filter for bus master attempts to write to DRAM; the system checks the address of the write cycle to see if it was previously detected in the preceding cycle, and if it was the transaction will pass directly to system memory without the overhead of an extra inquiry cycle. Enabling is therefore best for performance.

DRAM Clock

Allows DRAM to work concurrently with host bust clock. *Disabled* aligns with AGP Clock.

SDRAM Burst X-1-1-1-1-1-1-1

Allows burst mode. *Enabled* is best for performance.

DRAM Command Rate

Used according to the type of DDR memory you have. Two cycles is the standard latency, that is, the bank activate command is latched onto DRAM on the second clock after the chip select signal (CS).

For registered DIMMs (that is, with a register or buffer chip between the memory controller and chips on the DIMM to redistribute the addresses and reduce the load on the memory clock), this early issuing of the command saves the register having to wait for the next clock before addressing the chips. For unbuffered (non-registered) DIMMs, you can reduce the command latency to 1 cycle, meaning the next rising edge of the clock signal, which increases the memory bandwidth.

DRAM Act to PreChrg CMD

This affects the memory row timing, specifically the time from the active command to the precharge command on the same bank. The shorter the better, but watch for memory errors. See also DRAM PreChrg to Act CMD, below.

DRAM PreChrg to Act CMD

In league with the above, this controls the time taken for precharge to complete and make the memory row available. The shorter the better, but watch for memory errors.

Sustained 3T Write

Affects PBSRAM. Enables or disables direct map write back/write through the L2 cache, or enables sustained three-cycle write access for PBSRAM access at 66 or 75 MHz. Enabled is best for performance.

2 Bank PBSRAM

Sets the burst cycle for PBSRAM. 3-1-1-1 timing is available for read and write transactions at 66 or 75 MHz (VP2).

Turn-Around Insertion

When enabled, the chipset inserts one extra clock to the turn-around of back-to-back DRAM cycles. More technically, the extra clock is added to the MD signals after asserting the MWE signal before enabling the MD buffers, whatever that means. Disabled is the default, and best for performance. May need to be on for EDO. Back-to-Back means that address reads can alternate with page hits, which measn that data transfer happens at half-speed, but also that there are very low latencies, zero in some cases.

Turn-Around Insertion Delay

See *Turn-Around Insertion* (above).

DRAM ECC/PARITY Select

Allows you to select between two methods of DRAM error checking, ECC and Parity (default). ECC memory can correct single-bit errors, but only detect multi-bit errors. It works by adding some redundancy to data bits to enable later duplication of the information if required, typically used in servers.

Single Bit Error Report

When a single-bit error is detected, the offending DRAM row ID is latched, and the value held until the error status flag is explicitly cleared by software. If ECC (Error Correcting Code) is active, this will correct the error, but inform you that one has occurred. If ECC is used, enable.

ECC Checking/Generation

Enable with ECC SIMMs in all rows.

Memory Parity/ECC Check

Choose between methods of memory error checking. *Auto*, *Enabled* and *Disabled*.

Memory Parity SERR# (NMI)

The default of *Disabled* will not show memory errors. If you have parity chips, you can select Parity or ECC to correct 1 bit errors.

OMC Mem Address Permuting

Enable to allow the Orion Memory Controller to permute memory addresses to get alternate row selection bits. May hang the machine.

OMC DRAM Page Mode

Affects the Orion Memory Controller on a Pentium Pro motherboard. See *DRAM Page Mode Operation* (below).

DRAM Page Mode Operation

Page mode allows faster timing on consecutive memory accesses within a single DRAM page. Mostly, page mode is invoked automatically if the DRAM supports it.

CPU to DRAM Page Mode

Determines whether a DRAM memory page is held open after a memory access, as those to open pages can be between 30-40% faster than to closed ones, because they don't need precharging. Enabling this keeps all pages open. Disabling only opens them during burst operations, etc., when subsequent accesses will be to the same page - otherwise, DRAM pages are closed after accesses.

Fast Command

Controls the CPU' sinternal timing - *Enabled* allows it to handle instructions at a higher speed.

Fast Strings

Possibly related to 4-way memory interleaving. Enabled is best for performance.

Fast MA to RAS# Delay

Selects *DRAM Row Miss Timings*, which are independent of DLT timing adjustment. Don't change unless you change DRAM or the CPU. MA means *Memory Access*. *Low* is best for performance.

Fast RAS to CAS Delay

Determines the timing of transition from RAS to CAS. The lower the better for performance.

DRAM Quick Read Mode

For 386s only. Set to Normal.

Bank 0/1 DRAM Type

You can't change this, but it tells you whether you have FPM or EDO memory installed.

386 DRAM Quick Write Mode

As above.

DRAM Page Idle Timer

The time in HCLKs the DRAM controller waits to close a DRAM page after the CPU becomes idle. The shorter the better for performance.

DRAM Page Open Policy

When disabled, the page open register is cleared and the corresponding memory page closed. Otherwise, the page remains open, even with no requests to service.

DRAM Enhanced Paging

When enabled, the chipset keeps the page open until a page/row miss occurs. Otherwise it uses additional information to keep the page open when the host bus is active or the PCI interface owns the bus (when the host may be "Right Back").

DRAM Posted Write Buffer

When the chipset's internal buffer for DRAM writes is enabled, CPU write cycles to DRAM are posted to it so the CPU can start another write cycle before DRAM finishes its own cycle.

DRAM Data Integrity Mode

Select whether you want ECC or Non-ECC (parity) error checking. ECC (Error Correction Code) corrects memory errors of one bit, for which you need DIMMs with an extra 8 bits of bandwidth (they have an x72 designation, as opposed to x64). It works with the memory controller to add bits to each bit sent to memory which are decoded to ensure that data is valid, and used to duplicate information should it be necessary. Multi-bit errors are detected but not corrected. Unlike parity, there is only a penalty cycle when a 1-bit error is detected, so there is no performance hit during normal operations. See also….

SDRAM ECC Setting

An extension of the above - it adds memory scrubbing. Use *Disabled* if you are not using ECC, or you just want performance without error checking. *Check Only* reports errors, without correcting any. *Correct Errors* corrects single-bit errors, which takes an extra clock cycle. *Correct+Scrub* provides maximum reliability - it puts the corrected item back into memory.

LD-Off DRAM RD/WR cycles

Delay 1T gives you slower performance as it introduces a wait state.

CPU-DRAM back-to-back transaction

Back-to-back means that address reads alternate with page hits, so data transfer effectively happens at only ½ clock speed. However, it also means low latencies, zero in some cases.

PCI-to-DRAM Prefetch

Allows prefetching of large parts of memory, assuming coherent data, so the contents can be accessed with very low latency, thus boosting performance, particularly for sound and Firewire cards.

Bank n DRAM Type

Indicates whether DRAM in the corresponding bank (n) is treated as FPM or EDO (EDO can hold the output from the last read on the output pins while the next data transfer is set up). FPM works with anything, but the EDO setting may cause a malfunction if FPM is actually used, although it will improve performance slightly.

RDRAM Pool B State

The i850 chipset uses dual channel Rambus technology, in which the second channel is not used. To save power and reduce the risk of overheating you can choose Nap or Standby mode. With the latter, the RIMMs remain powered up and ready to function after initial latencies. With the former, RIMMs go into power-saving mode, which increases latencies if data in them is requested.

DRAM Background Cycles

Sets the DRAM background cycles, or the amount of work done behind the scenes. *Delay 1T* gives you slower performance as it introduces a wait state.

Leadoff DRAM R/W Command

1T sends the read/write commands one clock cycle behind the memory address. Otherwise they are sent at the same time.

Leadoff DRAM B/G Command

1T sends the background commands one clock cycle behind the memory address. Otherwise they are sent at the same time.

DRAM Addr/Cmd Rate

Sets the lead off DRAM read and write cycles. 1T sends the commands one clock cycle behind the memory address. Otherwise they are sent at the same time.

Auto Detect DIMM/PCI Clk

Enabling allows the system to detect and close clock signals to empty DIMM/PCI slots to reduce EMI.

DRAM Idle Timer

The clocks the DRAM controller will remain in the IDLE state before precharging all pages.

EMS Enable

Found on some 80286 or 80386 motherboards, often using the C&T NEAT Chipset. It enables Expanded Memory through the BIOS. Best done with supplied software.

MISCELLANEOUS

Turbo Mode

Used for testing on newer boards, this boosts the external clock frequency by about 2.5% to verify stability. On older machines, this turned the L2 cache on or off, or switched the motherboard into a higher speed by allowing Gate A20 to be used.

Prefetch Caching

Enables PCI slave prefetch caching to increase performance.

CPU Low Speed Clock

Or *Low Speed CPU Clock Select* selects the speed you want to use as the slow speed when you select *Turbo Off* on the front of your computer, or via your keyboard. This will be CLKIN (CPU speed) divided by 1, 2, 3 or 4.

Co-processor Ready# Delay

Enabling this with a non-compatible processor delays the ready signal by 1 T state, giving you a wider tolerance range, but less performance.

Co-processor Wait States

Number of wait states for the ready signal from NPU to CPU for similar reasons to *Co-processor Ready# Delay*, above.

C000 32K Early Shadow

Shadows the video BIOS before it initialises, assuming your VGA card agrees. As it happens before the POST you get reduced POST time and faster booting.

Check ELBA# Pin

Sets the time when the *External Local Bus Access#* pin is checked, during T1 or T2. Should mostly be set to T2, that is, later in the cycle for better reliability, but it can depend on other settings. The pin is active during local bus access cycles, so the CPU can communicate with devices on it without disturbing some support chips. This can hang the machine-DO NOT CHANGE IT IF YOUR MACHINE IS WORKING!

Appian Controller

An advanced IDE controller. You also need special software to activate it.

Mouse Support Option

Used to support a PS/2 type mouse on the keyboard port. Takes up 1K of base memory for an Extended BIOS Data Area, so you only get 639K.

IRQ XX Used By ISA

Here you can specify whether a particular IRQ is used by a Legacy card (that is, non PnP ISA) or available for the PCI bus. No/ICU means you are using the ISA Configuration Utility (ICU) and this IRQ is not required for Legacy purposes. Yes means the IRQ is required, and you are not using the ICU. In addition, you can use this setting to reserve an IRQ if using an NMI (not for W2K or XP).

IRQ 12 used by ISA or PS/2 Mouse

If you're not using a PS/2 mouse, you can use its IRQ for the ISA bus.

PS/2 Mouse Function Control

As above. *Enabled* allows the system to allocate IRQ 12 automatically.

CPU Address Pipelining

An Award Setting found on Pentiums, where the chipset signals the CPU for a new memory address before the current cycle is complete. Can be enabled if required by a multithreaded operating system.

CPU Drive Strength

Varies the signal strength of data transfer from the chipset to the CPU, with higher values representing stronger signals, so can be used for stability (not performance) when overclocking, at the expense of extra EMI and heat.

Keyboard Reset Control

If enabled, CPU operations will be halted before the System Reset signal is actually sent. Put more technically, HALT is executed before SYSC generates CPU reset from **Ctrl-Alt-Del**.

Keyboard Clock Select

As with bus speed, this should end up as standard, in this case 7.25 MHz, so for a 40 MHz CPU, you want CPUCLK/5. You can often decouple the keyboard clock from the bus clock, so you can run one faster than the other. Some motherboards give you an option of running at 9.25 MHz, but this is not often a good idea. The keyboard controller is actually a computer in its own right; at least, it has a microprocessor, and its own BIOS inside.

Novell Keyboard Management

Normally set to *No*, but if you find the keyboard sluggish when using a Novell product, set it for the smallest number between 1-30 that gives you best performance.

Middle BIOS

Sets the System BIOS to appear at E000. It's only for old software, so disable.

Delay Internal ADSJ Feature

ADS# is a bus control signal, or an Address Status strobe driven by the CPU to indicate the start of a CPU bus cycle, showing that a valid command and address is stable on the bus. The J is a substitute for #, which stands for signal. See Synch ADS below. Enable at 50 Mhz for best compatibility for VL bus cards, but performance will be reduced.

Synch ADS

If *Disabled*, can improve the performance on low speed machines (e.g. 25 MHz). Enable for 50 MHz 486s and 386/40s. Disable Auto Setup to use this.

Internal ADS Delay

Enabled, allows an additional span of time for the Address Data Status. Only use this if you have a fast processor.

NMI Handling

DO NOT DISABLE THIS! (sorry for shouting). It's for engineering *only*. Your machine will hang without the right equipment on to the board and you will need to discharge the CMOS. NMI stands for *Non Maskable Interrupt*, which is one that can't be worked around.

Power-On Delay

Specifies a short delay when power is turned on so the PSU can stabilise.

Software I/O Delay

Can be 0-255 units. Each increment adds a fixed delay based on CPU speed. Set to 10, 12, 14, 18 or higher for 16, 20, 25 or 33 MHz systems, respectively.

Sampling Activity Time

Selects the delay time when the chipset monitors and samples SMI (System Management Interrupt). You get a choice of *No Delay* or *Delay 1T*.

GAT Mode

Also known as *Guaranteed Access Timing Mode* on Acer motherboards. This setting guarantees the 2.1us CHRDY timeout spec from EISA/ISA buses, to allow their adapters the maximum time to respond to bus signals. Disabled takes advantage of PCI reponse time - an ISA bus master is granted the ISA bus and the SIO chip arbitrates.

Guaranteed Access Time

See above.

SIO GAT Mode

Found on a Pentium Pro , similar to the above. Disabling can improve performance slightly.

NA# Enable

Allows pipelining, where the chipset signals the CPU for a new memory address before data transfers for the current cycle are complete, giving faster performance.

Chipset NA# Asserted

Allows you to choose between two methods of asserting the NA# signal during CPU line fills (maybe). NA# stands for *Assertion Next Address*. *Enabled* helps performance, as it permits pipelining, where the chipset signals the CPU for a new memory address before the current cycle is complete.

LGNT# Synchronous to LCLK

When a VL bus is prepared to give a VL Bus Master access to it, it returns the LGNT# signal active, which acknowledges a request for control of the VL Bus; by default, the bus issues LGNT# as soon as the current bus master finishes with it. When this is enabled, the VL bus will also synchronize its response with the LCLK, the VL bus clock. Concerns reliability - normally, disable.

LOCAL ready syn mode

Whether the VESA Ready signal is synchronized by the CPU clock's ready signal, or bypassed.

- *SYN*. VESA ready synchronized by the CPU (default).
- *BYPASS*. Synchronization bypassed.

Local Ready Delay Setting

Set the Local Ready Signal to *No Delay*, *1T*, *2T* or *3T*.

Cyrix A20M Pin

Cyrix chips need special BIOS handling, if only because their 386 version has a cache (Intel's doesn't), and it may have trouble keeping the cache contents up to date if any part of the PC is allowed to operate by itself, in this case, the keyboard controller toggling the A20 gate. The A20M signal can be raised separately by the BIOS to tell the CPU the current state of the A20 gate. This also allows the CPU's internal cache to cache the first 64K of each Mb in real mode (the gate is always open in protected mode), and is fastest.

Cyrix Pin Enabled

As above, but refers to DMA and the FLUSH pin on the CPU, which invalidates the cache after any DMA, so the contents are updated from main memory, for consistency. If you can't set the FLUSH pin, increase the refresh interval and use Hidden Refresh.

Chipset Special Features

When disabled, the (TII or HX) chipset behaves as if it were the earlier Intel 82430FX.

Host Bus Slave Device

This allows you to use an Intel 486 Host Bus Slave (e.g. a graphics device).

Cyrix LSSR bit

Or LSSER. LSSR stands for *Load Store Serialize Enable* (Reorder Disable). It was bit 7 of PCR0 in the 5x86 (index 0x20), but does not apply to the 6x86 or the 6x86MX, as they have no PCR0 or index 0x20.

Host Bus LDEV

When enabled, the chipset monitors the LDEV (local device) signal on the host bus for attempts to access memory and I/O ranges out of the its range.

Polling Clock Setting

The rate the system polls all sub-systems (buses, memory, etc.) for service requests.

```
14.318 MHz
CLK2 (Default)
CLK2/2
CLK2/3
CLK2/4
28.636 MHz
```

Assert LDEV0# for VL

Enabled, allows a VLB slave device to talk to the chipset on a VL/PCI-based machine when there is no VL master present.

Signal LDEV# Sample Time

Choose T2, T3, T4 or T5.

Host Bus LRDY

When this is enabled, the chipset will monitor the LRDY (local ready) signal on the host bus, returning RDY to the CPU.

Memory Hole At 512-640K

When enabled, certain space in memory is reserved for ISA cards to improve performance - once reserved it cannot be cached, as it is mapped to the AT bus. Allegedly for OS/2.

LBD# Sample Point

Allows you to select the cycle check point, which is the point where memory decoding and cache hit/miss checking takes place. Doing it at the end of T3 rather than T2 gives you more time for checking, for greater stability.

486 Streaming

As well as burst mode, the 486 (and compatibles) supports a streaming mode where larger amounts of data are moved to/from memory per cycle. Enabling improves performance.

CHRDY for ISA Master

When enabled, this allows an ISA bus master device to assert CHRDY (*Channel Ready*), giving it immediate access to DRAM. The default is enabled.

NA (NAD) Disable for External Cache

Controls whether the chipset *Next Address* pin will be enabled, for early posting of the next address when making back-to-back accesses to L2 cache. *Enabled* is best for performance.

Set Mouse Lock

You can lock the PS/2 Mouse as a security precaution.

ATA-Disc

This only appears (in the MR BIOS) if you have an ATA device (actually up to eight). The fields are mostly filled automatically on selection, and should only be changed if you know the settings (transfer rates) are not correct.

P6 Microcode Updated

This allows you to load new microcode into the CPU (Pentium Pro/II) through the BIOS to correct minor errors, so disable for normal use.

Disconnect Selection

Turns the SCSI Disconnect function on or off. On is best for performance, as the SCSI device can disconnect and allow the CPU to get on with something else, although your operating system must be able to support this.

ChipAwayVirus

Helps the BIOS with a special virus detector card that checks the boot sector.

OS Select For DRAM >64MB

Use the *OS/2* setting with older versions (pre Warp 3.0) or NT and maybe Linux, when you have more than 64 Mb, since a specific BIOS switch is required to access memory above 64 Mb (the maximum reportable size is 64 Mb, due to the size of the register used (AX)). OS/2 and NT can get this reported as 16 Mb and convert it internally. Otherwise, use *Non-OS/2*.

OS Support for more than 64 Mb

See *OS Select For DRAM >64MB* (above).

OS/2 Compatible Mode

See *OS Select For DRAM >64MB* (above).

Boot to OS/2, DRAM 64 Mb or Above

See *OS Select For DRAM >64MB* (above).

Verifying DMI Status

To do with the Intel-Microsoft Desktop Management Interface, which is for remote sensing of computer configurations over a network.

MPS 1.1 Mode

The version of the multiprocessor specification.

POST Testing

Found on AST machines, determines whether POST testing will be normal, or in-depth. *Normal* just checks the memory.

MPS Version Control For OS

This specifies the version of the Multiprocessor Specification (MPS) to be used. Version 1.4 has extended bus definitions for multiple PCI bus configurations and future expandability, together with allowing a secondary PCI bus to work without a bridge - use 1.4 for NT, and possibly Linux. Leave as 1.1 for older Operating Systems, and for W2K on the Abit BP6.

Use Multiprocessor Specification

See above.

BIOS Update

Leave disabled unless actually updating the BIOS.

In Order Queue Depth

Determines the length of the queue of instructions to be processed in sequence (the bus command queue), as the Pentium Pro (or above) can execute out-of-order for smoother processing. Can be set to 1 or 8, meaning you can track up to 8 pipelined bus transactions (you might see 1 and 4, 1 and 8 or 1 and 12, depending on the chipset). The first option of *1* disables queuing- commands will only be issued after the processor has finished with the previous one, meaning the maximum amount of latency, which varies from 4 clock cycles for a 4-stage pipeline to 12 for 12-stager (this will help with overclocking). Otherwise, latency can be masked by queuing outstanding commands, which will boost performance somewhat.

Large Disk Access Mode

Choices are *DOS*, or *Other*. This was on a Packard Bell with A Phoenix BIOS.

Assign IRQ for VGA

If enabled, the BIOS will assign an IRQ for the VGA card, as most modern cards do. It's for the 3D features of a bus mastering card, but it may allow an AGP card to share an IRQ with the PCI 1 slot. Disabling releases the IRQ for another device, or reserves it for PCI 1.

Assign IRQ for USB

Enables or disables IRQ allocation for USB.

Monitor Mode

Interlaced or *Non-Interlaced*, according to whether the video system should output a full screen in sequence (NI) or lines in alternate passes (Interlaced). Cheap monitors won't support full interlace at higher resolutions.

Speed Model

For BIOSes that autodetect the CPU. Speedeasy does it for you. Jumper emulation is for the settings as taken from the manual, in terms of bus clock, multiplier, voltage and CPU speed.

Language

Sets the language on BIOS setup screens and error messages. Has no affect on the language used by the Operating System or applications.

Audio DMA

Selects a DMA Channel for motherboard sound systems.

S.M.A.R.T. for Hard Disks

Self-Monitoring Analysis & Reporting Technology, a feature of EIDE. Allegedly allows a drive to monitor itself and report to the host (through management software) when it thinks it will fail, so network managers have time to order spares. In fact, the management software sits between the BIOS and the hard drive and allows the BIOS to look at the data and decide whether or not to give you warning messages. This has nothing to do with performance, but convenience. Unfortunately, although Win 95 OSR2 and OS/2 (Merlin) are S.M.A.R.T aware, many failures cannot be sensed in advance. Since this system allows the monitoring of hard drives over a network, you will get extra packets not necessarily controlled by the operating system - if you get mysterious reboots and crashes, disable this.

Spread Spectrum Modulated

This setting is for electromagnetic compatibility (EMC) purposes, based on the idea that harmonic waves generated by bus activity may interfere with the signals that generated them in the first place. Otherwise, electrical components running at very high frequencies will interfere with others nearby, hence the FCC rules. This setting gets around the FCC by reducing EMI radiations with slightly staggered normally synchronous clocks, the idea being to lower the peak levels at multiples of the clock frequency by sending a wider, weaker pulse - in other words, the pulse spikes are reduced to flatter curves. It may also stop the sending of clock signals to unused memory sockets (see Auto Detect DIMM/PCI Clk, below). However, some high performance peripheral devices might stop working reliably because of timing problems. This means that, although the energy is the same, the FCC detection instruments only see about a quarter of what they should, since the energy is spread over a wider bandwidth than they can cope with. It is therefore possible that your PC is emitting much more EMI than you expect. The frequencies are varied slightly so that no particular one is used for more than a moment. Older boards either centered around the nominal value or were set with the nominal frequency as the maximum (low modulation). Most current ones use the centered method.

The settings could be *1.5% Down*, *0.6% Down*, *1.5% Center* or *Disabled* (the percentage is the amount of jitter, or variation performed on the clock frequency). Center means centered on the nominal frequency. Shuttle recommends 1.5% Down for the HOT631, but others allow enabling or disabling. The latter may be worth trying if your PC crashes intermittently, as there may be interference with clock multiplying CPUs that phase lock the multiplied CPU clock to the bus clock-if the frequency spread exceeds the lock range, the CPU could malfunction - even a .5% modulation up or down with today's frequencies can vary the bus speed by as much as 10 MHz inside one modulation cycle (.25% at 1 GHz means a change of 25 MHz) . In other words, disable when overclocking, because this setting may change the bus speed. In addition, the FSB setting could be cancelled out due to a pin address overlap on the clock generator chip.

The *Smart Clock* option turns off the AGP, PCI and SDRAM clock signals when not in use instead of modulating the frequency of the pulses over time, so EMI can be reduced without compromising stability. It also helps reduce power consumption.

FSB Spread Spectrum

See above.

Clock Spread Spectrum

See above.

Auto Detect DIMM/PCI Clk

This is similar to the Smart Clock option mentioned above. If there are no cards in the DIMM or PCI slots controlled by it, the clock signals are turned off, together with those for slots with no activity, to reduce EMI. This also reduces power consumption because only components that are running will use it.

Boot Speed

Turbo is actually the normal setting. *De-Turbo* turns off the CPU cache and increases memory refresh cycles, without slowing down the CPU or altering bus clocks and clock multipliers, unlike older versions which will reduce the ISA bus speed to about 8 MHz.

Physical Drive

Allows logical hard drives to be interchanged, but not with operating systems such as Unix that bypass the BIOS. Dropped in 1995 in Phoenix BIOS v4.05.

NCR SCSI at AD17 Present in

Specifies the slot in which a PCI NCR 53C810 SCSI card at AD17 is inserted. The options are Slot 1, Slot2, Slot 3,and Slot 4. You won't see this if the card isn't there.

CPU Warning Temperature

Sets the upper and lower thresholds of the CPU warning temperature, either side of which the system will behave as specified by you.

PCI Secondary IDE INT# Line

See *PCI Primary IDE INT# Line* (above).

Power-Supply Type

AT or *ATX*. It seems a bit late to set this after the machine has started, but it really concerns enabling soft-off options, etc.

CPU Core Voltage

Sets the voltage of the installed CPU. Use Auto normally, but you can override the settings to suit different circumstances.

Quick Frame Generation

When the PCI-VL bus bridge is acting as a PCI Master and receiving data from the CPU, a fast CPU-To-PCI buffer is enabled if this is also enabled, which allows the CPU to complete a write even though the data has not been delivered to the PCI bus, reducing the CPU cycles involved and speeding overall processing.

PCI Primary IDE INT# Line

Assigns an interrupt line to an add-on PCI primary IDE controller.

IN0-IN6(V)

The current voltage of up to 7 voltage input lines, if you have a monitoring system.

Current CPU Temperature

Indicates only if you have a monitoring system, but the CPU is not working hard anyway while you are in the BIOS setup.

Current System Temperature

Indicates current main board temperature if you have a monitoring system.

Current CPUFAN1 Speed

The mainboard can detect the rotation speed of two fans, for the CPU cooler and the system. This indicates the CPU cooling fan's rotation speed.

Current CPUFAN1/2/3 Speed

See above, for up to three fans, if you have a monitoring system.

Vcore/Vio/+5V/+12V/-5V/-12V

Detects the output of the voltage regulators and power supply.

MB Temperature, CPU Temperature, POWER Temperature

The hardware monitor will automatically detect these temperatures within the selected range.

CPU Fan Speed, POWER Fan Speed, CHASSIS Fan Speed

The hardware monitor will automatically detect these fan speeds in the selected RPM range.

VCORE Voltage, +3.3V Voltage, +5 Voltage, +12 Voltage

The hardware monitor automatically detects these voltages through onboard regulators.

Turbo External Clock

Disable for AMD CPUs.

Starting Point of Paging

The clocks required for starting of page miss cycles, or controls start timing of memory paging operations.

Processor Number Feature

For Pentium IIIs - you might not even see it if you don't have one. It allows you to control whether the Pentium III's serial number can be read by external programs.

Linear Merge

When enabled, only consecutive linear addresses can be merged.

DREQ6 PIN as

Invokes a software suspend routine by toggling the DREQ6 signal. Select Suspend SW only if your board has such a feature.

Flash BIOS Protection

Protects the BIOS from accidental corruption by unauthorized users or computer viruses. To update the BIOS, you must disable this, otherwise it should be enabled.

BIOS Protection

See above.

Drive NA before BRDY

When enabled, the NA signal is driven for one clock before the last BRDY# of every cycle for read/write hit cycles, generating ADS# in the next cycle after BRDY#, and eliminating a dead cycle. Enabled is best for performance.

591 Version A Function

You can enable or disable this. It probably refers to a special function in the SiS 591 chipset, but I haven't been able to find out what it is. It was found on a very old 386/486 motherboard, so is unlikely to be relevant anyway.

Hardware Reset Protect

When enabled, the hardware reset button will not function, preventing accidental resets (good for file servers, etc).

MWB Write Buffer Timeout Flush

The *Master Write Buffer* has a valid window that can be preset to a number of memory cycles, after which it will be flushed. Disabling forced flushing can increase performance but may corrupt data.

IOQ (4 level)

Apollo chipsets have a four stage pipeline (four buffers) for fast memory reads to CPU, called the In Order Queue or IOQ. Using all four buffers handles a full data burst, increasing performance - up to 5% for 3D applications and over 10% for office applications

Chassis Intrusion Detection

Alerts you when the computer case is interfered with. Clear the message with Reset, and it will revert to Enabled later.

CPU FSB Clock

Selects the CPU's Front Side Bus clock frequency.

PCI IDE Busmaster

This increases IDE performance under DOS if you use programs like Norton Ghost or Partition Magic. It has no effect in Windows, which has its own IDE busmastering software.

CPU FSB/PCI Overclocking

Sets the combination of CPU Front Side Bus and PCI frequency. H/W follows the hardware configuration. Depending on the speed of your CPU's FSB, you can alter the speed within a small range - at 100 MHz, it is 100-120 MHz. At 133 MHz, try 100-131 or 133-164 MHz.

S2K I/O Compensation

Concerns the signal strength on the S2K bus, which is a point-to-point bus from the memory controller to the CPU (Athlon), licensed from Alpha. It uses its own protocol to deliver an effective 200 MHz data transfer rate. Increase voltage for more stability when overclocking.

S2K Bus Driving Strength

The default (*Auto*) allows the chipset to dynamically adjust the Athlon bus strength or use preset values. However, the Manual setting allows you to increase it for stability when overclocking, at the risk of damaging the processor.

S2K STROBE N CONTROL

This appears when you select *Manual*, above. It determines the N transistor drive strength, which is represented by Hex values from 0-F (0 to 15 in decimal). The higher the drive strength, the greater the compensation for the motherboard's impedance on the S2K bus.

S2K STROBE P CONTROL

This appears when you select Manual, above. It determines the P transistor drive strength, which is represented by Hex values from 0-F (0 to 15 in decimal). The higher the drive strength, the greater the compensation for the motherboard's impedance on the S2K bus.

CPU/DRAM CLK Synch CTL

Sets the CPU/DRAM synchronisation. Synchronous is best for performance.

CPU Ratio/Vcore (V)

Two items that adjust the CPU clock multiplier and core voltage, for overclocking.

CS[5.0]# Hold Time CTL

Sets the CS hold time.

DQS/CSB Hold Time CTL

Sets the DQS/CSB hold time.

CKE Hold Time CTL

Sets the CKE hold time.

DRAM Ratio H/W Strap

This gets around the CPU-to-DRAM ratio limitation in Intel i845 chipsets (normally, with a 400MHz FSB, you are limited to 1:1 or 3:4. For 533 MHz, it's 1:1 or 4:5). It controls the external hardware reset strap assigned to the MCH, and fools the chipset into thinking a 400 or 533 MHz FSB is present (Low or High, respectively). The default is *By CPU.*

DOS Flat Mode

For using the DOS method of memory addressing, where every memory address is a real, for better stability (see the Memory chapter). Windows uses extended memory this way automatically, so a setting like this in the BIOS would be for when you are using software that needs it to run. Using this, therefore, memory addresses consist of one piece, rather than the segment and offset. The result is safer addressing, and the possibility of creating and running larger programs.

Act Bank A to B CMD Delay

For DDR, this sets the delay between Active commands (tRRD). The shorter the interval, the better the performance, at the expense of stability. In a notebook, you maybe want to increase the numbers, to reduce current surges (and heat), but 2 should be OK in a desktop.

DRAM Driver Slew

For DDR. Enable for a small performance increase.

DDR RAM CAS Latency

Set at the factory when a machine comes with memory installed. The lower the figure, the better the performance at the possible expense of stability.

DDR Voltage

Adjusts the DDR voltage to increase the DDR rate. Naturally, the recommendation is not to use it long term.

FPU OPCODE Compatible Mode

The (x87) FPU stores the opcode of the last non-control instruction (the fopcode or FOP code) in an 11-bit register, to provide state information for exception handlers. The first 5 bits of the first opcode byte are the same for all opcodes, so only the last 3 bits of the first opcode byte are stored. The second opcode byte provides the remaining 8-bits of data. When enabled, the processor reverts to an old system (FOP code compatibility mode) and stores the last non-transparent floating point instruction in the 11-bit FOP register. When disabled, the processor stores only the FOP of the last non-transparent floating point instruction that had an unmasked exception. This is quite an esoteric thing and should normally be disabled, for performance. Intel recommends it should only be enabled if your (older) software uses the fopcode to analyze performance, or to restart after an exception has been handled.

Game Accelerator

Found on Abit motherboards. *F1* is fastest.

N/B Strap CPU As

See above.

Athlon 4 SSED Instruction

There is a status bit to tell software that the Athlon XP/MP supports the full SSE instruction set, but it can cause compatibility problems with BeOS and some graphics cards. This is a toggle - when enabled, the processor is recognised as being SSE compatible. You would only disable it if you are getting problems.

CPU Hyper-Threading

Hyperthreading allows one processor to execute multiple threads concurrently, making the system look like a multiprocessor system (in which case enable APIC Function as well). It is only supported by the Pentium 4 (faster than 3.06GHz) and the Xeon. In addition, you will need a suitable chipset and BIOS, and an operating system, such as XP or Linux 2.4.x.

ISA Enable Bit

Allows the system controller to perform ISA aliasing to prevent conflicts between ISA devices. *Enabled* forces it to do so with addres bits 15:10, meaning that 16-bit devices hae only a maximum contiguous I/O space of 256 bytes. When disabled, all 16 address lines can be used for I/O address space decoding, so 16-bit devices gain access to the full 64 Kb of I/O space. As with anything with the ISA bus, disable for optimal AGP and PCI performance.

Master Priority Rotation

Controls the priority of the processor's accesses to the PCI bus. *1 PCI* means it will will always be granted access after the current PCI master completes its transaction, irrespective of how many others are in the queue. *2 PCI* means access is granted after the second bus master has finished, and so on. The lower the figure, the better the processor's performance, but higher figures improve PCI bus performance.

P2C/C2P Concurrency

This allows PCI-to-CPU and CPU-to-PCI traffic to occur simultaneously, so the CPU does not get locked up during PCI transfers. It also allows PCI traffic to the processor to occur without delay, even when it is writing to the PCI bus (this may prevent performance issues with some PCI cards). Enabled is best for performance.

Memory Termination

This is like terminating the (high speed) memory bus in a similar way to SCSI (it uses things like termination voltage regulators), that is, it stops stray signals bouncing around all over the place. As such, it may be a fix for ghost memory, where the board thinks it has more memory installed than is actually there (2 modules are shown, where you only have one, for example).

ISA 14.318MHz Clock

When the ISA bus is accessed, the whole computer slows down to the "usual" speed of 8 MHz for 16-bit cards ("usual", because 8 MHz was never established as a standard - it's 4.77 MHz for 8-bit cards). This settings allows you to overclock the ISA bus (if you have one) to 14.318 MHz (the reference clock speed), but the cards may not like it. In fact, this has been tried from the early days and was never really successful anyway, so use it with caution.

VGA BIOS

12

Many manufacturers, such as AOpen, are beginning to allow access to the Video BIOS as well as the System BIOS, so you can use various display devices and extract maximum performance from the more complex chipsets, even down to adjusting the speed of the fan on the video card. In AOpen's case, at least, you get to them by pressing the **Insert** key as the machine starts. Otherwise, you may find these settings available through software supplied with your card. When optimizing video, ensure that AGP is given the maximum transfer rate, and that *AGP Sideband* and *AGP Fast Write* support is turned on, although these sometimes cause their own problems (see below). Some settings described here may also be found in the BIOS Setup.

AGP

Standing for *Accelerated Graphics Port,* this is a system based on PCI and the old VESA local bus, which started life in Pentium II machines with the Intel 440 LX chipset and above (other chipsets support AGP with Socket 7). The idea was allow graphic instructions to be controlled by the CPU and bypass the PCI bus, at 66 MHz, and reduce costs; 3-D data would move to system memory, making room in the graphics controller for other functions, so, in effect, the graphics system acquires its own bus and the AGP card becomes just an interface for the monitor, as the memory on it can be bypassed. However, memory on video cards became now faster, and very plentiful, up to 128 Mb in cases, and manufacturers tend to ignore Intel's original intentions because, as usual, many proposed features have not actually been implemented. Some cards use the system memory as a giant cache, where textures are pulled across the AGP bus once and subsequently accessed from the memory. The original voltage (for 1x and 2x, under AGP 1) was 3.3v, for the slot and signalling protocol, which reduced to 1.5v with AGP2 (which covers AGP 4x). AGP 8x (covered by AGP 3), however, uses a 1.5v slot and *.8v* for the signalling protocol. AGP Pro requires four times the electrical power. *Only remove the label over the socket if you are using an AGP Pro card.*

An AGP motherboard will generally have a 1.5v slot and will support 1.5v and .8v signalling, so you can use a 4x card if you want (but you won't get AGP 8x service). An 8x card will also support both, so you can use it in a 4x motherboard, though, as before, you won't get AGP 8. In short, don't mix AGP 1x and 2x with 8x - the voltages are different..

When both sides of the timing signal are used (known as X2), you can move twice as much data and achieve an effective 133 MHz clock speed, allowing up to 533 Mb/sec, which is four times what PCI is allegedly capable of, which is probably why it may be unstable on LX motherboards. There is also no arbitration to slow things down. Peak AGP 2x bandwidth is the same as that of 66 MHz SDRAM. Since the CPU will need some of this, you need higher memory bandwidth and higher speeds to give AGP the headroom it needs. Aside from (at least) the 440 LX chipset, you also need at least DirectX 5.0, Windows '95 OSR 2.1 and vgartd.vxd, an Intel driver, not forgetting SDRAM for the bandwidth.. *AGP Texturing* allows textures to be executed directly from AGP system memory. Local Texturing is the old way, where textures are copied to the local frame buffer memory and then used.

AGP Pipelining is a data transfer method that queues multiple requests at a time by using the "PIPE#" signal of the AGP protocol, and which may be more stable than Sideband Addressing (see below), which uses SBA signals to request and receive data at the same time. AGP Frame Mode uses "FRAME#".

PCI Express is expected to supersede AGP (and PCI).

AGP Aperture Size (64 Mb)

The AGP memory aperture is the range of PCI memory address space used for DMA by an AGP card for 3D support, in which host cycles are forwarded to the card without translation, giving extra speed (it is used like a swapfile). It is where memory-mapped graphics data structures reside and is the amount of memory the GART (Graphics Address Remapping Table) can see, which makes the processor on the video card see the card memory and is that specified here as one continuous block. This also determines the maximum amount of system RAM allocated to the graphics card for texture storage, so is a combination of card and system memory used as a total (this was done because video memory is expensive, and could be bypassed, but now video cards have lots of memory, it is less important). However, the memory isn't actually in one block, except by coincidence-it is assembled from 4K memory pages scattered around the memory map. Note that this is just memory allocated - AGP and system memory (in that order) is only used as a last resort, when 3D runs out of local memory, so whatever you set here doesn't automatically take away your system RAM - it's just a limitation on future use. There is no universally correct setting, but you could try doubling your AGP memory size, and adding 12 Mb for virtual addressing (the doubled amount is for write combining, as AGP memory is uncached). Alternatively, half the video memory size and divide it into system memory, to account for modern cards with lots of RAM (cards with more RAM need less of an aperture to work with - that is, this will shrink as memory on your video card increases). If you specify too little here, you will get paging to hard disk. On the other hand, you may get errors if you specify too much - some motherboards run more stably when this is set to the amount of RAM in your video card.

The 64 Mb default is usually OK, but if you don't have much video memory, use up to half system memory, bearing in mind that more than twice whatever the texture storage space required is actually used (check the manufacturer, but most cards need 16 Mb anyway). This setting is not performance related, and neither does it affect 3DFX cards, as they do not support AGP texturing. However, it does affect the registry - AGPSize in HKEY_CURRENT_CONFIG\Display\Settings cannot be more than set here.

Some applications alter this for their own purposes anyway, and override this.

Graphic Win Size

This sets the size of the aperture (see above) and the Graphics Address Relocation Table, or GART, which is a table that translates AGP memory addresses into actual ones, which are often fragmented, allowing the graphics card to see the memory available to it as a contiguous range. Again, see above.

AGP 2X Mode

Allows your AGP VGA card to switch to 133 MHz transfer mode, if supported, where both the rising and falling edges of the signal are used to transfer data, at two transfers per clock. Otherwise the card operates in 1X mode (66 MHz). 2X may be unstable on LX boards.

AGP 4x Mode

Disable if you have 1x or 2x cards, otherwise they won't work properly. AGP 4X sends four data transfers per clock, and has a lower voltage.

AGP 8x Mode

Disable if you do not have an 8x card. 4X is the default.

4X AGP Support

Disable if you do not have a 4x card. 2X is the default.

AGP 4x Drive Strength

See also AGP Driving Control, below. You can set whether the drive control is manual or done by the chipset (auto). The choices in manual mode are made under *AGP Drive Strength P or N Ctrl*, below.

AGP Drive Strength P or N Ctrl

This setting appears when manual is selected under AGP 4x Drive Strength, above. The figures are in hex, and higher values represent stronger signals, which is what you need when overclocking, so this is only indirectly related to performance. Both ranges are from 0-Fh, but *P* covers the first part, *N* the second, of the values concerned, so the actual range is 00-FFh (that is, 0-255). C4 is a typical default.

AGP Driving Control

Allows overriding of the automatic settings for more precise control when overclocking. Choices are manual and auto. Used in conjunction with......

AGP Driving Value

See also *AGP Drive Strength P or N Ctrl*, above. This only works if *AGP Driving Control* (above) is set to *Manual*. It concerns the signal strength of the AGP bus, with higher values representing stronger signals, making it useful for stability when overclocking, so it only affects performance indirectly (be careful about your card). The range of 00-FF means 0-255 in decimal values, with a typical default being DA (218).

AGP Comp. Driving

Used by AMI BIOSes to adjust the AGP driving force. Selecting Manual gives you another setting, Manual AGP Comp. Driving, below.Auto is best for safety.

Manual AGP Comp. Driving

See above.

AGPCLK/CPU FSB CLK

The relative speeds against each other of AGP and CPU bus clocks. *1/1* means that the AGP is running at the same speed (for older processors). 1/2 means that the FSB is at 133 MHz, so AGP can use its standard speed of 66 MHz, but you can stretch it to 75, or even 83 in some cases. For best performance, the AGP figure should be easily divisible into the higher speed.

AGP Fast Write Transaction

This allows data to be sent directly from the core logic (i.e. chipset) to the AGP master (graphics chip) instead of keeping a copy in system memory and making the AGP master fetch it. In other words, it creates a short cut between the AGP chipset and graphics memory, and leaves system memory out of the loop. On some BIOSes, it enables or disables cacheing of the video memory. *Enabled* is best for performance, but probably only works with 4x cards.

AGP Fast Write

See above (AMI BIOS).

AGPCLK/CPUCLK

See below.

AGP Transfer Mode

Seems to override the automatic selection of 1x, 2x or 4x.

AGP Master 1 WS Read

Implements a single delay when reading from the AGP Bus. Normally, two wait states are used, allowing for greater stability, but check with your motherboard manufacturer to see if they have already implemented a Master latency of zero, in which case the lowest reading here of 1 will reduce performance.

AGP Master 1 WS Write

As above, but for writes.

AGP Sideband Support

Enable or disable. AGP Sideband Addressing is a transfer mechanism allowing the card to send and receive at the same time, by using a second bus for addresses and commands to the graphics processor, so the data can flow as fast as it can over the AGP bus. It may decrease stability, though, and cause crashes on some cards, due to the design of some motherboards resulting in glitches on strobes (in fact, you need support on the motherboard, graphics card and drivers). Try using *Pipeline Transfer* instead - performance will probably be the same.

AGP Read Synchronisation

Enabled is best for performance.

Core Chip Clock Adjust

Varies the speed of the main CPU on the video card. The higher the speed, the better for 3D.

AGP ISA Aliasing

Most ISA cards only decode the lower 10 address lines, and few use all 16, which is why some video cards get confused with COM 4; as far as the lower 10 address lines are concerned, they're the same! For example, 220h (standard Sound Blaster) converts to 10 0010 0000 in binary. If you have a card at 2A20, the first 10 digits are the same as 220 (10 1010 0010 0000- right to left, remember), so it won't work. ISA devices could later extend their I/O space by using 63 16-bit aliases for the few 10-bit I/O locations, so the total number of I/O locations went up from 768 to 49,152. This also reduced the chances of two ISA cards conflicting (hence the name ISA Aliasing). However, AGP and PCI devices use 16-bit addressing, so they are limited to 256 bytes of contiguous I/O space - more than that,and they creep into ISA allocations, so there may be conflicts between AGP and ISA cards. When enabled, only the first 10-bits are used for decoding. When disabled, all 16 address lines can be used, giving 16-bit devices access to 64K of I/O space. Only enable it if you have ISA cards in the first place, and then only if they are causing conflicts with AGP.

AGP Always Compensate

Whether the AGP controller can dynamically adjust the AGP driving strength or use preset drive strengths - it is automatically set to do this at regular intervals (when enabled), although it can also be disabled or bypassed (this feature does not allow manual configuration). Disabling means the drive strength is only set once, at boot-up. Enabled is recommended.

DBI Output for AGP Trans

DBI stands for *Dynamic Bus Inversion*, and it is specific to AGP 3.0. When enabled (recommended), it is supposed to help reduce power consumption and signal noise, assisting with stability (rather than performance). The AGP bus has two sets of data lines, for a total of 32. If a lot of them switch to the same polarity at the same time, there will be a lot of electrical noise. DBI uses two extra lines, one for each set, to allow only 8 polarity reversals per set (if there are more than 8 lines switched, the DBI line is switched instead).

No Mask of SBA FE

SBA is the *Sideband Address*, and this controls the signal used to calibrate it. If it is masked, the calibration signal is not initiated, and the graphics card will not hang. Otherwise, SBA calibration is done just after the AGP calibration cycle.

Init Display First

Which VGA card, that is, PCI or AGP you want to be initialised first, or which is connected to the primary monitor, for multi-monitor systems (you can use 2 of each, but you've probably got only one AGP card anyway). Whatever combination you have, the PCI is treated as the default, which is probably the opposite of what you need, so you can change it here.

Init AGP Display First

See above-this makes the AGP display the primary one.

Video Memory Clock Adjust

As above, for the memory on the card.

Chip Voltage Adjust

Adjusts voltage to support increased speeds. All voltages listed are within the safety margins.

V-Ref & Memory Voltage Adjust

4 levels of Memory & V-Ref Voltage adjustment, to help stability at different speeds.

Boot Up Display Select

Choose your display device here.

Fan Speed

Adjust it here.

GPU Temperature

Monitors the temperature of the GPU (nVidia Geoforce 2 GTS).

TV-Out Format

PAL, NTSC or whatever.

Post Up Delay

The delay time for the on-screen message when the machine starts.

Post Up Prompt

Whether the prompt for access to these facilities is displayed or not.

ISA Linear Frame Buffer

For an ISA card that features a linear frame buffer (e.g. a second video card for ACAD).

Residence of VGA Card

Whether on PCI or VL Bus.

ISA LFB Size

LFB = *Linear Frame Buffer*. This creates a hole in the system memory map when there is more than 16Mb of RAM, so accesses made to addresses within the hole are directed to the ISA Bus instead of Main Memory. Leave Disabled, unless you're using an ISA card with a linear frame buffer that must be accessed by the CPU, and you aren't using Plug and Play's Configuration Manager or ISA Configuration Utility. If you choose 1 Mb, the *ISA LFB Base* Address field will appear (see below).

ISA LFB BASE ADDRESS

The starting address for the ISA memory hole if 1 Mb has been set for *ISA LFB Size* (above).

ISA VGA Frame Buffer Size

This is to help you use a VGA frame buffer and 16 Mb of RAM at the same time; the system will allow access to the graphics card through a hole in its own memory map; accesses to addresses within this hole will be directed to the ISA bus instead of main memory. Should be set to Disabled, unless you are using an ISA card with more than 64K of memory that needs to be accessed by the CPU, and you are not using the Plug and Play utilities. If you have less than 8 Mb memory, or use MS-DOS, this will be ignored.

VGA Memory Clock (MHz)

The speed of the VGA memory clock.

VGA Frame Buffer

When enabled, a fixed VGA frame buffer from A000h-BFFFh and a CPU-To-PCI write buffer are implemented.

Video Palette Snoop

A PCI video card needs to know if an ISA one is present, especially if it's not standard. This allows multiple VGA cards to be used on multiple buses to handle data from the CPU on each set of palette registers on every video device (Bit 5 of the command register in the PCI device configuration space is the VGA palette snoop). VGA snooping is used by multimedia video devices (e.g. MPEG or video capture boards) to look ahead at the video controller (VGA device) and see what colour palette is currently in use when in 256-colour mode, that is, what 256 colours out of the thousands available are in the VGA memory. This setting controls how a PCI graphics card can snoop write cycles to an ISA video card's colour palette registers. Only set to *Disabled* if:

- An ISA card connects to a PCI graphics card through a VESA connector
- The ISA card connects to a colour monitor, and
- The ISA card uses the RAMDAC on the PCI card, and
- Palette Snooping (RAMDAC shadowing) not operative on PCI card

Palette Snooping

Enable when using a Multimedia (MPEG) video card, so the address space of the PCI VGA palette can be snooped for colour information from the video processor and overlay. In other words, an ISA video card is able to synchronise its colour palette with one on the PCI bus. More in *PCI/VGA Palette Snoop*, below.

Graphic Window WR Combine

The book says it this determines whether the graphics window base address is valid or not. It's more likely to be the ability to combine writes to video memory to increase performance.

PCI/VGA Palette Snoop

Having an MPEG board or a graphics accelerator attached to the feature connector of your video card alters the VGA palette somewhat. Enable this if you have ISA MPEG connections through the PCI VGA feature connector, so you can adjust the PCI/VGA palettes, and solve situations where the colours in Windows are wrong. For example, you may get a black and white display while booting. In the Award BIOS, this tells the PCI VGA card to keep silent (and prevent conflict) when the palette register is updated (i.e. it accepts data without responding). Useful only when two display cards use the same palette address when both are plugged into the PCI bus (such as MPEG or Video capture). In such cases, PCI VGA keeps quiet while the MPEG or capture functions normally. However, you should only need this in exceptional circumstances, like if you have a very old PCI 2D accelerator card, so disable for ordinary systems. (Award BIOS). See also Video Palette Snoop (below).

VGA Palette Snoop

See above.

PCI/VGA Snooping

Enabled, looks for a VGA card on the ISA/VLB bus. Disabled looks on the PCI bus.

VGA DAC Snooping

When enabled, the integrated controller does not claim colour palette accesses to an add-in video card. When off, palette accesses are not forwarded to it.

Snoop Filter

Saves the need for multiple enquiries to the same line if it was inquired previously. When enabled, cache snoop filters ensure data integrity (cache coherency) while reducing the snoop frequency to a minimum. Bus snooping is a technique for checking if cached memory locations have been changed through DMA or another processor; it compares the address being written to by a DMA device with the cache Tag RAM. If a match occurs, the location is marked. If the CPU tries to read that location later it must get the data from main memory, which contains what has been written by DMA. In other words, bus snooping invalidates cached locations modified by anything other than the CPU, to prevent old data being read. Bus snooping must access L1 and L2 caches, using the processor bus in the case of the former. Nine bus clocks are used to perform the snoop, so it involves a loss of performance, particularly as the CPU cycle is delayed if the snoop starts just before a CPU memory access cycle. For these reasons, it is pipelined in the HX chipset.

PCI VGA Buffering

Enabled is best for performance.

Initial Display

This doesn't matter if you have only one display card, but when you have two, you can specify whether your primary monitor is attached to a PCI or VGA card.

Search for MDA Resources

Tells the machine to look for a mono video card if one is fitted, otherwise the memory address space will be used for something else.

Primary VGA BIOS

Allows you to select the primary graphics card.

Video Shadow Before Video Init

See above.

Turbo VGA (0 WS at A/B)

When enabled, the VGA memory range of A0000-B0000 uses a special set of performance figures, more relevant for games, that is, it has little or no effect in video modes beyond standard VGA, those most commonly used for high resolution, high color displays associated with Windows, OS/2, UNIX, etc. Same as *VGA Performance Mode*.

POWER MANAGEMENT

13

This is for Green PCs, or those complying with the EPA Energy Star programs; the intention is to save unnecessary power usage if the system becomes inactive - even though the average PC uses about the same energy as 2 light bulbs in a day, it's still a lot if the grid is overstrained already, as in California, and costs money, too (actually about 90 bucks a year per PC). Unfortunately, companies that rely on remote maintenance cannot afford to have machines completely turned off to save power. They need facilities such as *Wake On LAN* (WOL) to receive signals over a network and wake the machine up (the same sort of thing can be done with serial ports, etc. so that modems can be used, or IRQs).

If you need the computer to switch itself on at a particular time or day, remember that, if you have an ATX power supply, that an external timer switch will be no good at all, because the switch on the front of the machine is only connected to the motherboard and not the PSU. When the timer applies the power, the front switch will still need to be pressed to activate the machine - with AT-style PSUs, you could simply leave the power switch open. In addition, using a time switch to turn off will not be good for Windows, as it has to be shut down properly. A better way is to leave the machine on and find the setting in the BIOS that will do this instead. Leaving the date at zero will ensure that the setting applies every day. There are shutdown utilities at **www.nonags.com**.

The solution to using too much power is to have various parts of the machine go to sleep as they become unnecessary - power is reduced automatically to the devices and restored as quickly as possible when activity is detected (in theory, anyway). This is usually done with idle timing and event monitoring techniques. A point to watch, especially with ACPI (see below), is that a device that goes into too deep a sleep may be deemed by the operating system to have been removed. A *Power Management Unit* (PMU) monitors interrupt signals through an interrupt events detector. If it hears nothing for a while, the system is put gradually and progressively to sleep, in that the longer the time inactive, the more parts of the system will close down. However, setting all this up in the BIOS only goes so far - you should do it in your operating system as well (not NT) - certainly, ensure that 95/98's compatibility with APM 1.0 is enabled through Control Panel.

The 5 choices available range from simple "dozing" to complete shutdown:

- *Dozing* slows the CPU down only, to around half speed.
- *Standby* shuts down HD and video, or CPU and SCLK.
- *Suspend* shuts down all devices except the CPU.
- *Inactive* stops the CPU, slows the SCLK and powers down the L2 cache.
- *HDD Power Down* just shuts down the hard disk (not SCSI).

As with anything, there are industry standards. For energy saving, these include:

- **APM**, or *Advanced Power Management*, devised by Intel/Microsoft. This must be active if you want to keep the time and date when the system is suspended, with power.exe for DOS (try power.drv for Windows) that coordinates BIOS, DOS and program activity. APM is responsible for shutting the system down on quitting the operating system, typically Windows 9x, and other useful tricks. You may need to force version 1.0 in Windows to make it work, but 1.1 allows more control and reports more accurately. It does not, however, allow you to control devices independently, for which see ACPI, below.
- **ATA**, or *AT Attachments Specification*, for IDE drives. Some ATA compliant devices provide Spindown facilities.
- **DPMS**, or *Display Power Management Signalling*. Monitors and cards conforming to this are meant to be matched, as signals are sent between them to put the CRT into various low power states, which need instructions from the BIOS. There are recognised power management states: Run, Standby, Suspend and Off. Suspend is slower to return to the Run state than Standby, which is regarded as being temporarily idle. Disable Standby and Suspend if you don't want PM.
- **ACPI**, or *Advanced Configuration and Power Interface*, hashed out mainly by Intel, Microsoft and Toshiba, which allows desktop PCs to have instant on, and be better for voicemail and household device control, as peripherals can be turned off as well as the main system unit - in other words, individual devices can be switched through motherboard control as required. This system therefore controls system resources as well as power (in fact, PnP is part of it). Only ACP BIOSes later than Jan 1 1999 are guaranteed to work with Windows 2000. S3 mode, or *Suspend-to-RAM*, maintains memory contents even though everything has stopped, using only 5 watts/hour. ACPI routes PCI and AGP devices through 1 IRQ, usually 9 or 11.

ACPI is a substitute for APM, which uses the BIOS (ACPI uses the operating system to manage power). The APM utility will keep the system time updated when the machine is in suspend mode. With Windows 98 and above, this is installed automatically and altered through Control Panel (the Advanced settings).

Some BIOSes have their own maximum and minimum settings for the times allocated, but you may have a "User Defined" option for your own. More options may be available for SL (low power) CPUs. SM Out, by the way, means the *System Management Output* control pin.

Smart Battery System

Circuitry added to a battery to allow better power management, battery life and information, such as time remaining. The battery talks to the system and tells it what services are required (some charging systems depend on battery heat as an indication of charge status). All this has been formalised into the SBS system, which actually stems from five documents containing the specifications for the battery itself, host system hardware, BIOS and charging. The SMBus is a separate bus for direct communication between the host and the battery. The Smart Charger allows a battery to control its own charge, while a Smart Battery Selector is used in multiple systems to determine which one is in use, which is charging, etc.

PM Control by APM

Or Power Management Control by Advanced Power Management. Switches APM on or off; choices are *Yes* or *No*. If Yes, combine DOS and Windows utilities for Green Mode (only with S-series CPUs). When enabled, an Advanced Power Management device will be activated to enhance the maximum Power Saving mode and stop the CPU internal clock. In other words, the BIOS will wait for a prompt from APM before going into any power management mode. If disabled, the BIOS will ignore APM. You need DOS and Windows utilities as well.

Power Management/APM

See above.

Power Management

Selects the type or degree of power saving for Doze, Standby and Suspend modes.

- *Max Saving*. Pre-defined settings at Max values, for SL CPUs only
- *User Define*. You can set each mode individually
- *Min Saving*. Predefined settings at Minimum values
- *Disabled*. Global Power Management will be disabled.
- *Power Up By Alarm*. Set the alarm that returns the system to Full On state

ACPI Function

Indicates whether your operating system is ACPI aware. Select *Yes* or *No*.

ACPI Standby State

How power saving mode is entered through ACPI. Choices are *S1/POS* or *S3/STR*. S1 is a low power state in which no system context is lost. In S3, power is only supplied to essential components (like main memory or wake-capable devices). S5 is Soft Off. All system context is saved to main memory, which is used to restore the PC to its former state on wake-up.

ACPI Suspend Type

See above. Select S1 or S3. Set the latter for *Suspend To RAM*.

USB Wakeup from S3

Allows USB activity to wake the system up from S3.

ACPI Suspend-to-RAM

Your +5vSB of the power supply must provide more than 720ma to support this.

Call VGA at S3 Resuming

This is related to the above. Setting Enabled makes the BIOS call the Video BIOS to wake up the VGA card when coming back from an S3 state, which lengthens the resume time, but you will need an AGP driver to initialise it if you disable. If the driver does not support initialisation, the display may work badly or not at all.

Power/Sleep LED

How the Power LED on the front of the case is used to indicate the sleep state. When Single LED is selected, it blinks without changing colour. With Dual LED, it changes colour.

PM Events

A *Power Management* (PM) *Event* awakens the system from, or resets activity timers for Suspend Mode. You can disable monitoring of some common I/O events and interrupt requests so they do not wake up the system - the default is keyboard activity. When *On*, or named, as for LPT and COM ports, activity from a listed peripheral device or IRQ wakes up the system.

GP 105 Power Up Control

When enabled, a signal from General Purpose Input 05 returns the system to Full On state.

IDE Standby Power Down Mode

Also known as Hard Disk Timeout, or HDD Power Down (Award), allows automatic power down of IDE drives after a specified period of inactivity, but some don't like it (notebook drives are OK). 15 minutes is a suggested minimum, to avoid undue wear and tear on the drive. Probably doesn't affect SCSI drives.

HDD Power Down

See above.

HDD Standby Timer

The hard disk powers down after a selected period of inactivity.

Standby Mode Control

Sets standby clock speed to fractions of CPU speed, and enables/disables the video.

IDE Spindown

As for *Standby Mode Control*, from MR BIOS.

Doze Timer/System Doze

Certain parts of the machine are monitored, i.e. hard disk, keyboard, mouse, serial and parallel ports, interrupts and the like, and if they are inactive for a length of time determined here, the computer dozes off for a short while; that is, it reduces activity and use of power until any of the above items become active again. Gives 80% sleep, 20% work.

Power-down mode timers

From MR, sets a timeout before power saving is entered. Standby slows down the CPU and video clocks. Suspend turns them off. It is used when the computer is thought to be temporarily idle. Power reduction measures include the monitor partially powering down, or the CPU speed slowing to 8 MHz. Gives 92% sleep, 8% work (like me).

Global Standby Timer

After the selected period, the system enters Standby mode.

Green Timer

Either Disable, or establish between 10 secs-3 hours.

Suspend Timer

Comes into force after the system has been idle for some time, say an hour, when the computer thinks it's unattended. The CPU can be stopped, and the monitor disabled to the extent of needing to warm it up. There may be a CRT OFF mode, which will need the on/off switch to get the monitor working again. You may also see an 8X Mode for factory testing and demonstrations; all it does is make everything operate 8 times faster. 99% sleep, 1% work (no, this is more like me). May support a Suspend switch on the motherboard.

Global Suspend Timer

After the selected period, the system enters Suspend mode.

Sleep Clock

Select *Stop Clock* or *Slow Clock* during Sleep Mode.

Sleep Timer

After the selected period of inactivity, all devices except the hard disk and CPU shut off.

Suspend Mode Option

Select the type of Suspend Mode:

- *POS*. Power-On Suspend (CPU and core system remain powered on in a very low-power mode).
- *Auto*. After the selected period of inactivity, the system automatically enters STD mode. Otherwise it enters STR mode (see below).
- *STD*. Save To Disk
- *STR*. Suspend To RAM

Suspend Mode

Sets the time period before the system goes into suspend mode.

Suspend Mode Switch

Controls a hardware switch that puts the computer into Suspend Mode.

Suspend Option

Lets you select a method of global system suspend. *Static Suspend*, sometimes called *Power-on Suspend* (POS), leaves the CPU powered on, but stops its clock. *0v Suspend,* sometimes called *Save To Disk* (STD) Suspend, saves the state of the system to disk then powers it off.

Auto Keyboard Lockout

If the keyboard powers down, use **Ctrl-Alt-Bksp** and wait for the keyboard lights to go on and off, then enter the CMOS password.

CPU Clock (System Slow Down)

After the specified time interval, the CPU will be slowed down to 8 MHz.

Event Monitoring

As for *Individual IRQ Wake Up Events* (System IRQ Monitor Events), from MR (see below).

- *Local*. Monitoring checks only the keyboard, PS/2 mouse and two serial port interrupts.
- *Global*. Monitoring checks all interrupts.

Monitor Power/Display Power Down

You must have a green power supply for this. After the specified time interval, the monitor power will be turned off. Monitors with the circuitry to cope with this can be a pain if it goes wrong and keeps powering down anyway.

Monitor Event in Full On Mode

In On Mode, the Standby Timer (see Standby Timer Select) starts counting if no activity is taking place and the programmable time-out period has expired. Devices checked under this category are included in the list of devices the system monitors during the PM timers countdown. Otherwise their activity doesn't affect it.

Individual IRQ Wake Up Events (System IRQ Monitor Events)

IRQs are monitored as an indirect method of watching the CPU, since it cannot be checked directly. The system can be woken up or sent to sleep if one is generated, or not, typically by a mouse (see Expansion Cards for a full list of IRQs).

IRQ 1(-15) Monitor

As for *Event Monitoring*.

IRQ8 Break Suspend

IRQ8 refers to the system clock. Here, you can enable or disable monitoring so it doesn't wake the system from Suspend mode.

IRQ8 Break [Event From] Suspend

See above.

IRQ8 Clock Event

See above.

DRQ 0 (-7) Monitor

As IRQ 1(-15) Monitor, but for DMA input monitoring. See Expansion Cards for a full list of DMA Channels.

System Events I/O Port Settings

Wakes the system up if one of these is accessed.

Keyboard IO Port Monitor

Allows ports 60 and 64h to be monitored for system activity (or not).

Floppy IO Port Monitor

As for *Keyboard IO Port Monitor*, but for port 3F5h.

Hard Disk IO Port Monitor

As for *Keyboard IO Port Monitor*, but for ports 1F0h-3F6h.

Video Port IO Monitor

As Keyboard IO Port Monitor, but for video ports.

VGA Adapter Type

If you set this to *Green*, and if your video card supports Green features, Vertical and Horizontal scanning will also be stopped when the screen is blanked.

Video Memory Monitor

As Keyboard IO Port Monitor, but for A000-BFFF areas of upper memory.

Low CPU Clock Speed

What speed to use when at slow speed.

Power Management Control

Enabled, turns power management on.

Power Management RAM Select

Where the 32K required for power management is, in Upper Memory (def E000).

O.S

So you can use Non-S and AMD/Cyrix chips to shut down the monitor. Select All O.S. for non-DOS systems, or select the IRQ (e.g. DOS ONLY15).

Factory Test Mode

Do not enable this (if you see it).

APM BIOS

Turns Automatic Power Management On or Off. Use with care, as some motherboards can't maintain the time of day in some power saving modes. However, it can save 25-40 kilowatt hours a month if your PC is left on all the time. Best left off otherwise, as it can be a pain.

APM BIOS Data Area

Where to keep data relating to Power Management, F000 or DOS 1 K.

ACPI I/O Device Node

Enables or disables ACPI device node reporting from the BIOS to the Operating System.

Device Power Management

Has the following headings:

- *Display Type Support*. Set to Green PC if you have an EPA compatible monitor. Otherwise set Standard.
- *Video Off in Suspend Mode.* Permits the BIOS to power down the video display when the computer is in suspend mode.
- *IDD HDD Off in Suspend Mode.* As above, for hard drive.
- *Ser Prt Off in Suspend Mode.* As above, for serial ports.
- *Par Prt Off in Suspend Mode.* As above, for the parallel port.
- *Prog I/O Off in Suspend Mode.* As above, for Prog I/O.

System Power Management

Has the following headings:

- *System Cache Off in Suspend Mode*
- *Slow Refresh in Suspend Mode.* Refreshes DRAM every 45, not 16 ns.

Auto Clock Control

Without APM, or if it isn't enabled, the BIOS will manage the CPU clock in the same way.

Power Button Override

When this is enabled, you must press the power button for over 4 seconds before the machine will turn off. *Disabled*, it powers off immediately. It needs an ATX power supply.

PWR Button < 4 Secs

When set to [Soft Off], the ATX switch can be used as a normal system power-off button when pressed for less than 4 seconds. [Suspend] gives the button a dual function, where pressing for less than 4 seconds puts the machine into sleep mode. In any case, pressing the button for more than 4 seconds turns the machine off.

Power Down and Resume Events

You can disable monitoring of some common I/O events and interrupt requests so they do not wake the system up from Suspend Mode, or reset the activity timers. Select On if you want an IRQ, when accessed, to reload the original count of the global timer, which is the hardware timer that counts down to Doze, Standby and Suspend modes. Selected IRQs also cause the system to wake up from a global Doze, Standby and Suspend mode when accessed. If a Doze timeout is set, the system enters Doze mode when it expires. Then the timer reloads with the standby timeout, if one is set, otherwise it uses the suspend timeout, if one is set. If not, the timer turns off. The effect is similar for Standby and Suspend timeouts. If more than one global timeout is set, the timeouts run one after the other.

DMA Request

Enabled, permits local bus DMA requests.

Reload Global Timer Events

When enabled, an event occurring on each listed device restarts the the global timer for Standby mode.

NON-SMI CPU Support

Selects an IRQ to replace SMI events when the CPU doesn't support SMI.

Video Off In Suspend

Turns off video when entering suspend mode.

Throttle Duty Cycle

The percentage by which CPU speed is cut back when it gets hot, or for power saving.

Soft-off by PWR-BTTN

Instant-Off allows the system to switch off immediately the power button is pressed. Otherwise, it will only do so after you press it for more than 4 seconds. Below this, the switch acts as a suspend button, leaving a small amount of power on the system so that power can be restored not only by the power switch but also by ring detection-your PC is therefore potentially subject to voltage surges on the power line 24 hours a day, whereas a conventional power switch physically disconnects the PC. This option may also leave power on the parallel ports and prevent printers from entering their own power saving modes.

Power Button Function

See above. When set to On/Off, the button works normally. For *Suspend*, the machine goes into suspend mode when pressed for a short time, or off when pressed for over 4 seconds.

Switch Function

Select the operation of the power button, when pressed:

- *Deturbo*. System slows - press a key to return to full power
- *Break*. System enters Suspend Mode - press a key to return to full power
- *Break/Wake*. System enters Suspend Mode - press the power button to return to full power

System Monitor Events

The following are monitored for inactivity:

- *Video ROM Access C000h, 32K*. Allows LB access to Video ROM C000.
- *Video RAM Access A000-C7FF*. Permits local bus access to this area.
- *Video Access A000-C7FF*. Combines the previous two options.
- *Local Bus Device Access*. Enabled, permits local bus device access.
- *Local Bus Master Access*. Enabled, permits bus master device access.
- *Local Bus Access*. Combines previous two options.

Video Off Option

When to turn the video off for power management. Choices are:

- *Always On*. Screen is never turned off
- *Suspend -> Off*. Screen off when system in Suspend mode
- *Susp, Stby -> Off*. Screen off when system in Standby or Suspend mode
- *All modes -> Off*. As above (so why have it?)

Video Off After

See also *Video Off Option*. Turns the video off after a system event:

- *N/A*. Never turn screen off
- *Suspend*. Off when system in Suspend Mode
- *Standby*. Off when system in Standby Mode
- *Doze*. Off when system in Doze Mode

Video Off Method

How the video will be switched off. Choose:

- *DPMS*, if your VGA card and monitor support it (BIOS turns it off).
- *Blank Screen*. The screen will only be blanked when video is disabled. Uses more power than V/H Sync + Blank. Use when there is no power management. The screen saver will not display.
- *V/H Sync + Blank*. As well as Blank Screen, the Vertical and Horizontal Sync signals are turned off, but if your card is not compatible, use Blank Screen only. Green monitors detect V/H-Sync signals to turn off the electron gun - if not, the gun is turned off.

Resume By Ring

Powers the system on when the Ring Indicator signal is received in UART 1 or 2 from an external modem. Needs ATX power supply and IRQ8 Clock Event enabled.

Resume By LAN/Ring

Allows the system to wake up in response to a Ring Indicator signal from an external modem through UART 1 or 2, or a wake-up signal through the network card from a server. Resume By Ring needs IRQ8 Clock Event to be enabled. Wake on LAN gives you the ability to remotely boot a PC from across a network even if it has been powered down.

RTC Alarm Resume

Set the date and time at which the Real Time Clock awakens the system from Suspend mode.

Wake up on PME

Allows the system to wake up from power saving mode through a Power Management Event.

Wake up on Ring/LAN

As above. Naturally, you need either a modem or a network card for this to work properly. And an ATX power supply.

Ring Power Up Act

Powers the system on when the Ring Indicator signal is received in UART 1 or 2 from an external modem. Needs an ATX power supply.

Resume By Alarm

Uses an RTC alarm to generate a work event or, in other words, an alarm from the Real Time Clock can wake the system up from sleeping. Needs an ATX power supply and IRQ8 Clock Event to be enabled. With the AMI BIOS, allows booting up on a specified time or date from the Soft Off (S5) state.

Resume by USB From S3

When Enabled, this allows you to use a USB device to wake up a system in the S3 (STR - Suspend To RAM) state. This item can be configured only if ACPI Suspend Type is set to [S3(STR)].

WakeUp by OnChip Lan

When Enabled, you can remotely wake up a PC in a Soft-Off condition through a LAN card that supports the wake up function.

Alarm Date/Hour/Minute/Second

Appears when the above is enabled. You can set the times and dates here for the system to resume or boot up. When the settings are changed, you must reboot and exit the operating system for them to take effect.

Keyboard Resume

When disabled, keyboard activity does not wake the system up from Suspend mode.

Thermal Duty Cycle

Slows down the CPU by the specifications listed here when it overheats.

CPU Warning Temperature

Sets an alarm when the CPU reaches a specified temperature.

CPU Critical Temperature

See above. Specifies a thermal limit, after which a warning is given.

Fan Failure Control

What happens if the CPU fan fails.

Automatic Power Up

For unattended or automatic power up, such as Everyday, or By Date.

After AC Power Loss

Whether the system will reboot after a power loss or an interrupt. *Power Off* leaves it off. *Power On* makes it reboot. Last State restores it back to where it was before the event occurred.

AC PWR Loss Restart

Whether or not the system should reboot after the power has been interrupted Enabled reboots, while Previous State means what it was doing before it was so rudely interrupted. Disabled means it stays off.

Power On By PS/2 Keyboard

This allows you to use specific keys to turn the system on. The ATX power supply provide at least 1 amp on the +5VSB lead. Only one or two keys are involved, so you can't use it as a password system.

Power On By PS/2 Mouse

You can use the PS/2 mouse to power up the system, for which you need an ATX power supply that can provide at least 1 amp on the +5VSB lead.

Instant On Support

Enable to allow the computer to go to full power on mode when leaving a power-conserving state. Only available if supported by the hardware. The AMI BIOS uses the RTC Alarm function to wake the computer at a prespecified time.

ZZ Active in Suspend

When enabled, the ZZ signal (whatever that is) is active during Suspend mode.

- *Version 1*. Cache controller into sleep mode when system is in Suspend mode.
- *Version 2*. When enabled, PB SRAM (cache) consumes power in PM mode.

Advanced OS Power

Allows the operating system to control power management, but may need to be turned off during some installations to stop the floppy shutting down in the middle.

BIOS PM on AC

For portables, whether power management is active when running on external (AC) power. *On* enables power management at all times. *Off* turns it off except when using batteries.

BIOS PM Timers

After a specified period of inactivity for a selected subsystem, it enters standby mode.

COM Port Activity

The PM system cannot directly monitor CPU activity, but must deduce it by monitoring external activities which require it, in this case, the serial port.

VGA Activity

Determines whether VGA activity is monitored for low power mode.

VGA Active Monitor

When enabled, any video activity restarts the Global Timer for Standby Mode.

Video Timeout

Sets the timeout for automatic video blanking.

LPT Port Activity

Whether parallel port activity is monitored for initiation of low power mode.

CPU Fan Off In Suspend

When this is enabled, the CPU fan is shut down when the CPU is put into suspend mode. As with power supplies, frequent starting and stopping of the fan may cause more wear than just letting it run.

CPU Fan on Temp High

Switches the fan on at a predetermined CPU temperature.

Doze Mode

After a period of system inactivity, the CPU slows down whilst everything else is at full speed.

Doze Timer

As above.

Doze Timer Select

Selects the timeout period (i.e. of system inactivity) after which the system enters Doze Mode.

Doze Mode Control

Sets the Doze Mode clock speed to various fractions of normal CPU speed and permits the VGA to be enabled or disabled. DOS time may be incorrect.

Doze Speed (div by)

Selects a divisor of full CPU speed to reduce the CPU to during Doze Mode.

Standby Speed (div by)

Selects a divisor of full CPU speed to reduce the CPU to during Standby Mode.

Wake/Power Up On Ext. Modem

If the modem receives a call while the system is in Soft-Off mode, this setting allows the machine to wake up or even power up when the modem tells it to (actually when it receives an initialisation string, obtained when the modem is turned off, then on again). Naturally, nothing will happen until the software is up and running, and the first try will produce no results, so you need to build in a time delay for communications.

Standby Mode Control

See *Doze Mode Control*.

Standby Timer Select

Selects the timeout period (i.e. of system inactivity) after which the system enters Standby.

Standby Timers

After the selected period of inactivity for each subsystem (video, hard drive, peripherals), it enters Standby Mode.

FDD/COM/LPT Port

Reloads the global timer when there is a FDD/COM/LPT event.

FDD Detection

Floppy drive activity wakes up the system or resets the inactivity timer.

HDD detection

As above, for hard disks.

Video Detection

When enabled, video activity wakes up the system or resets the inactivity timer.

IRQn Detection

As above, for IRQs.

LREQ Detection

Enabled, activity on the LREQ signal line wakes up the system or resets the inactivity timer.

Wake on Ring

Allows a computer to be brought up from low power mode when telephone rings detected

Wake Up Events

You can turn On or Off monitoring of commonly used interrupt requests so they do not waken the system from, or reset activity timers for, Doze and Standby modes. the default is keyboard activity.

Wake Up Event in Inactive Mode Enable

See above.

WakeUp Event In Inactive Mode

Which interrupts (IRQs) will wake the system up from power saving. It may not work properly with PnP Operating Systems that move IRQs between devices without warning.

Inactive Mode Control

Sets the Inactive Mode clock speed to fractions of normal CPU speed or turned off entirely - it also permits the VGA Display to be enabled or disabled. If 0 clock Speed (STOP CLK) is selected, the CPU cannot monitor external activities and therefore cannot automatically bring the computer back to normal based on actions such as keystroke entries.

Watch Dog Timer

A hardware timer that generates either an NMI or a reset when the software that it monitors does not respond as expected when it is polled. See also *WDT fields*, below.

WDT Active Time

The watch dog timer period.

WDT Configuration Port

The I/O port for the watch dog timer.

WDT Time Out Active For

The watch dog timer response.

Boot from LAN first

Allows booting from a LAN boot image before attempting it from a local device.

CRT Power Down

Allows the CRT to power down when the system is in Green Mode.

CRT Sleep

The manner in which the CRT is blanked.

GPI05 Power Up Control

When enabled, a signal from General Purpose Input 05 returns the system to Full On state.

Day of Month Alarm

Select a date in the month, but use 0 if you want a weekly alarm.

Month Alarm

Select a month by number (1-12) or NA if you want the alarm for all of them.

Week Alarm

Turn the alarm on and off on specific days.

Hot Key Power Off

Enable to use the hot key for soft power off, if your system has one.

LDEV Detection

Detects activity on the LDEV signal line to wake up the system or reset the inactivity timer.

Shutdown Temperature

Selects lower and upper limits for system shutdown temperature, if your computer has environmental monitoring. If the temperature extends beyond either limit, it shuts down.

DRQ Detection

When enabled, activity on a DRQ line wakes the system up or resets the inactivity timer.

Modem Use IRQ

The IRQ line assigned to the modem, on which any activity awakens the system.

Power Up On PCI Card

The system can wake up from a signal coming through a (PCI) network card or modem, for which the ATX power supply must be able to produce 1A on the +5VSB lead.

Suspend To RAM

Part of ACPI 1.0, which drops the power consumption to the lowest possible level and allows the quickest resumption, as the system context is kept in memory. The current of the 5VSB line must be more than .75a, and ACPIU should be enabled, with *ACPI Suspend Type* set to *S3*.

Primary INTR

Acts like a master switch for the interrupts under it - when *On*, they can be configured as resets for the power saving timeouts. *Primary* refers to timeouts using the primary timer (i.e. power saving modes). *Secondary* refers to background maintenance tasks.

Inactive Timer Select

The period of inactivity after which the system becomes inactive (longer than for Standby).

Display Activity/IRQ3/IRQ4……

Whether the BIOS monitors the activity of the selected peripheral. When set to *Monitor*, it will either wake up the system or stop it going into power saving.

Clock Throttle

Determines the processor clock speed when it is in *Suspend To RAM* (STR) mode (normally, it has no effect). The options are in elements of processor power consumption (not clock speed, which is actually based on the selection). 50% therefore means that its clock speed allows it to use 50% less power - the lower the figure, the longer it will take for the processor to get back up to speed. In other words, the duty cycle of the STPCLK# signal

K7 CLK_CTL Select

This is AMD-specific. In power saving mode, bus speed remains constant, but routes through a clock divider to reduce the CPU's internal speed to 1/64 for Palominos and older, or 1/8 for newer ones, so a 2.0 GHz Palomino runs at 31.25MHz (an Athlon with a Thoroughbred core runs at 250MHz). This means the power used is in the same proportion, but older cores take longer to get back to full speed, and they can sometimes overshoot the nominal clock speed to cause a reduction in the bus's I/O drive strength. Although auto-compensation tries to correct this, there isn't enough time, so the drive strengths cannot be obtained before the processor reconnects to the system bus, causing it to fail, and hang the system. This allows the BIOS to force the CLK_CTL register to reduce the run-up time. Use *Optimal* for a Palomino or older Athlon, or a Thoroughbred-B or above. Otherwise, use the default.

PLUG AND PLAY/PCI

14

A system for making the use of expansion cards easier (yes, really!). In this context, ISA cards not compatible with PnP are known as *Legacy Cards*, and are switched as normal to make them fit in ("legacy" describes something that's out of date but is tolerated in modern equipment). You will also have to reserve the IRQ or DMA settings they use in the BIOS, otherwise they might not be found later. Have as few as possible, as accesses to them are slow.

With *Concurrent PCI*, the *Multi Transaction Timer* allows multiple transfers in one PCI request, by reducing re-arbitration when several PCI processes can take place at once; with more than one CPU and PCI bus, both PCI buses can be accessed simultaneously. Passive Release allows the PCI bus to continue working when receiving data from ISA devices, which would normally hog the bus; in other words, it helps with latencies. *Delayed Transaction* allows PCI bus masters to work by delaying transmissions to ISA cards, which may need disabling if using a single-CPU OS with dual processors. Write merging combines byte, word and Dword cycles into a single write to memory. The idea is that plug and play cards get interrogated by the system they are plugged into, and their requirements checked against those of the cards already in there. The BIOS will feed the data as required to the Operating System, typically Windows 9x. Inside the BIOS, the POST is enhanced with automatic resource allocation, with reference to the ESCD.

Here you will be able to assign IRQs, etc to PCI slots and map PCI INT#s to them. Although Windows or a PnP BIOS can do a lot by themselves, you really need the lot, e.g. a Plug and Play BIOS, with compatible devices and an Operating System for the best performance. Operating Systems that natively support PnP are Windows 95/98, 2000, XP and OS/2. Linux can also handle it with its own software, as can Windows NT with a module on the installation CD, but it's not supported by Microsoft. Note that these systems do not require PnP hardware - devices won't be configured without the right system, but you just have to do it manually, like with non-PnP stuff. Be aware that not all PCI (2.0) cards are PnP, and that although PC (PCMCIA) cards are "Plug and Play", they are not considered here. Also, anything using PCI address ranges will not be seen by the BIOS on boot-up, which doesn't mean that it isn't working.

PnP itself was originally devised by Compaq, Intel and Phoenix. The system checks what resources are needed, then it coordinates IRQs, DMA and I/O Ports, finally telling the operating system what it has done. To do this, however, it needs help from the expansion card, which must be able to deactivate itself from normal control signals to avoid conflicts. As well, each board has registers that are accessed through standard I/O port addresses, so the BIOS and operating system can configure it. The ports are *Address*, *Write Data* and *Read Data*.

The Address port is like a pointer that expands the control registers available without taking more resources (PnP defines eight card control registers and two large ranges, one of 24 registers for future expansion and another of 16 for the board maker's own purposes). The Address port allows the Write Data port to choose which devices are active and the resources used by them. Some boards, such as video adapters, start as active because they are needed when booting. Others, like sound cards, come up inactive and wait to be configured by the operating system. PnP boards have circuitry that handles the configuration process, constantly monitoring bus signals.

Every PnP device starts up in the *Wait for Key* state, waiting for the Initialisation key, which is a 32-step process between the host system and each board. After successful initialisation, the expansion board shifts itself into Sleep. Your chipset settings may allow you to choose of two methods of operation (with the Plug and Play OS setting):

- All PnP devices are configured and activated.
- All PnP ISA cards are isolated and checked, but only those needed for booting are activated. The ISA system cannot produce specific information about a card, so the BIOS has to isolate each one and give it a temporary handle so its requirements can be read. Resources can be allocated once all cards have been dealt with (recommended for Windows '95, as it can use the Registry and its own procedures to use the same information every time you boot). This leads to....

ESCD

Extended System Configuration Data is a system which is part of PnP (actually a superset of EISA), that can store data on PnP or non-PnP EISA, ISA or PCI cards to perform the same function as the Windows '95 Registry above, that is, provide consistency between sessions by reserving specific configurations for individual cards. Without ESCD, each boot sequence is a new adventure for the system. It occupies part of Upper Memory (E000-EDFF), which is not available to memory managers. The default length is 4K, and problems have been reported with EMS buffer addressing when this area has been used.

PCI IDENTIFICATION

Company Name	Dec ID	Hex ID
2WIRE	5483	0x156B
3A	4844	0x12EC
3COM	4279	0x10B7
3CX	5351	0x14E7
3Com	4793	0x12B9
3DFX INTERACTIVE	4634	0x121A
3PARDATA	5520	0x1590
3WARE	5057	0x13C1
A-MAX TECHNOLOGY	5534	0x159E
A-TREND	5475	0x1563

Company Name	Dec ID	Hex ID
ABB AUTOMATION PRODUCTS	5317	0x14C5
ABB ROBOTICS PRODUCTS	5086	0x13DE
ABIT	5243	0x147B
ABOCOM SYSTEMS	5073	0x13D1
ACARD TECHNOLOGY	4497	0x1191
ACCTON TECHNOLOGY	4371	0x1113
ACCUSYS	5334	0x14D6
ACER LABS	4281	0x10B9
ACKSYS	5416	0x1528
ACQIRIS	5356	0x14EC
ACQIS TECHNOLOGY	5424	0x1530
ACTEL	4522	0x11AA
ADAPTEC	36868	0x9004
ADDI-DATA GMBH	5560	0x15B8
ADDONICS	5139	0x1413
ADLINK TECHNOLOGY	5194	0x144A
ADMTEK INC	4887	0x1317
ADTEK SYSTEM SCIENCE CO LTD	4972	0x136C
ADVANCED MICRO DEVICES	4130	0x1022
ADVANCED SYSTEM PRODUCTS	4301	0x10CD
ADVANCED TECHNOLOGY LABORATORIES	4487	0x1187
AETHRA S.R.L.	5023	0x139F
AG COMMUNICATIONS	5369	0x14F9
AG ELECTRONICS LTD	5579	0x15CB
AGERE INC.	5606	0x15E6
AGFA CORPORATION	4611	0x1203
AGIE SA	5185	0x1441
AGILENT TECHNOLOGIES	5564	0x15BC
AIM GMBH	5191	0x1447
AIRONET WIRELESS COMMUNICATIONS	5305	0x14B9
ALACRITECH INC	5018	0x139A
ALACRON	4246	0x1096
ALADDIN KNOWLEDGE SYSTEMS	16748	0x416C
ALCATEL	4196	0x1064
ALFA INC	5486	0x156E
ALLEN- BRADLEY COMPANY	4768	0x12A0
ALLIED DATA TECHNOLOGIES	5515	0x158B
ALLIED TELESYN INTERNATIONAL	4697	0x1259
ALOKA CO. LTD	5128	0x1408
ALPHA PROCESSOR INC	5337	0x14D9
ALPHA-TOP CORP	5485	0x156D

Company Name	Dec ID	Hex ID
ALTEON WEBSYSTEMS INC	4782	0x12AE
ALTERA CORPORATION	4466	0x1172
AMBICOM INC	5013	0x1395
AMBIENT TECHNOLOGIES INC	6163	0x1813
AMBIT MICROSYSTEMS CORP.	5224	0x1468
AMDAHL CORPORATION	4614	0x1206
AMERICAN MEGATRENDS	4126	0x101E
AMERICAN MICROSYSTEMS INC	5417	0x1529
AMERSHAM PHARMACIA BIOTECH	5550	0x15AE
AMO GMBH	4775	0x12A7
AMP	4152	0x1038
AMPLICON LIVELINE LTD	5340	0x14DC
AMTELCO	5347	0x14E3
ANALOG DEVICES	4564	0x11D4
ANCHOR CHIPS INC.	4798	0x12BE
ANDOR TECHNOLOGY LTD	5274	0x149A
ANNABOOKS	4428	0x114C
ANTAL ELECTRONIC	5436	0x153C
AOPEN INC.	41120	0xA0A0
APEX INC	5081	0x13D9
APPIAN/ETMA	4247	0x1097
APPLE COMPUTER INC.	4203	0x106B
APPLICOM INTERNATIONAL	5001	0x1389
APPLIED COMPUTING SYSTEMS INC.	5595	0x15DB
APPLIED INTEGRATION CORPORATION	5342	0x14DE
ARALION INC.	5432	0x1538
ARCHTEK TELECOM CORP.	5374	0x14FE
ARDENT TECHNOLOGIES INC	5478	0x1566
ARK RESEARCH CORP.	4939	0x134B
ARM Ltd	5045	0x13B5
ARN	5521	0x1591
ARRAY MICROSYSTEMS	4796	0x12BC
ARTESYN COMMUNICATIONS PRODUCTS INC	4643	0x1223
ARTX INC	5120	0x1400
ASCEND COMMUNICATIONS.	4359	0x1107
ASTRODESIGN	4543	0x11BF
ASUSTEK COMPUTER.	4163	0x1043
ATELIER INFORMATIQUES et ELECTRONIQUE ETUDES S.A.	5433	0x1539
ATI TECHNOLOGIES INC	4098	0x1002
ATLANTEK MICROSYSTEMS PTY LTD	5513	0x1589

Company Name	Dec ID	Hex ID
ATMEL-DREAM	5176	0x1438
AUDIOCODES INC	5368	0x14F8
AURAVISION	4561	0x11D1
AUREAL INC.	4843	0x12EB
AURORA TECHNOLOGIES.	4700	0x125C
AUSPEX SYSTEMS INC.	4290	0x10C2
AUTOMATED WAGERING INTERNATIONAL	5640	0x1608
AVAL NAGASAKI CORPORATION	4708	0x1264
AVANCE LOGIC INC	16389	0x4005
AVID TECHNOLOGY INC	4527	0x11AF
AVLAB TECHNOLOGY INC	5339	0x14DB
AVM AUDIOVISUELLES MKTG & COMPUTER SYSTEM GMBH	4676	0x1244
AVTEC SYSTEMS	5482	0x156A
AYDIN CORP	5115	0x13FB
Aculab PLC	4825	0x12D9
Adaptec/Cogent Data Technologies Inc	4361	0x1109
Advanet Inc	4879	0x130F
Aims Lab	4813	0x12CD
Analogic Corp	4822	0x12D6
B-TREE SYSTEMS INC	5616	0x15F0
B2C2	5072	0x13D0
BALDOR ELECTRIC COMPANY	5215	0x145F
BALTIMORE	5427	0x1533
BANCTEC	5623	0x15F7
BANKSOFT CANADA LTD	5377	0x1501
BARR SYSTEMS INC.	4531	0x11B3
BASIS COMMUNICATIONS CORP	5343	0x14DF
BASLER GMBH	5006	0x138E
BECKHOFF GMBH	5612	0x15EC
BEHAVIOR TECH COMPUTER CORP	5392	0x1510
BELL CORPORATION	5409	0x1521
BIOSTAR MICROTECH INT'L CORP	5477	0x1565
BITBOYS OY	5578	0x15CA
BLUE CHIP TECHNOLOGY LTD	5063	0x13C7
BLUE WAVE SYSTEMS	4465	0x1171
BLUESTEEL NETWORKS INC	5547	0x15AB
BOEING-SUNNYVALE	4981	0x1375
BOPS INC	5523	0x1593
BRAIN BOXES LIMITED	4954	0x135A
BRAINS CO. LTD	4993	0x1381

Company Name	Dec ID	Hex ID
BREA TECHNOLOGIES INC	2697	0x0A89
BROADCOM CORPORATION	5348	0x14E4
BROADLOGIC	5363	0x14F3
BROOKTREE CORPORATION	4254	0x109E
BST COMMUNICATION TECHNOLOGY LTD	5296	0x14B0
BUG.	4509	0x119D
BULL HN INFORMATION SYSTEMS	4511	0x119F
BVM LIMITED	5568	0x15C0
Billionton Systems Inc./Cadmus Micro Inc.	5323	0x14CB
Brooktrout Technology Inc	4836	0x12E4
C-CUBE MICROSYSTEMS	4671	0x123F
C-MEDIA ELECTRONICS INC	5110	0x13F6
C-PORT CORPORATION	5390	0x150E
CACHEFLOW INC	5600	0x15E0
CALCULEX INC	5092	0x13E4
CANON RESEACH CENTRE FRANCE	5360	0x14F0
CAPITAL EQUIPMENT CORP	4860	0x12FC
CARDIO CONTROL N.V.	5309	0x14BD
CARRY COMPUTER ENG. CO LTD	5359	0x14EF
CATALYST ENTERPRISES INC	5538	0x15A2
CATAPULT COMMUNICATIONS	52428	0xCCCC
CCI/TRIAD	5556	0x15B4
CEMAX-ICON INC	5468	0x155C
CENTILLIUM TECHNOLOGY CORP	5393	0x1511
CENTRAL SYSTEM RESEARCH CO LTD	5636	0x1604
CENTURY SYSTEMS.	4668	0x123C
CHAINTECH COMPUTER CO. LTD	9999	0x270F
CHAMELEON SYSTEMS INC	5382	0x1506
CHAPLET SYSTEM INC	5408	0x1520
CHICONY ELECTRONICS CO LTD	5459	0x1553
CHORI JOHO SYSTEM CO. LTD	4940	0x134C
CHRYON CORP.	5425	0x1531
CHRYSALIS-ITS	51966	0xCAFE
CIMETRICS INC	5557	0x15B5
CIPHER SYSTEMS INC	5014	0x1396
CIRTECH (UK) LTD	5331	0x14D3
CIS TECHNOLOGY INC	5174	0x1436
CISCO SYSTEMS INC	4407	0x1137
CLARION CO. LTD	5016	0x1398
CLEVELAND MOTION CONTROLS	5225	0x1469
CLEVO/KAPOK COMPUTER	5464	0x1558

Company Name	Dec ID	Hex ID
CMD TECHNOLOGY INC	4245	0x1095
COGNEX INC.	4855	0x12F7
COGNEX MODULAR VISION SYSTEMS DIV.- ACUMEN INC.	4791	0x12B7
COLOGNE CHIP DESIGNS GMBH	5015	0x1397
COMBOX LTD	5403	0x151B
COMPAL ELECTRONICS INC	5312	0x14C0
COMPAQ COMPUTER CORP.	3601	0x0E11
COMPUMASTER SRL	5536	0x15A0
COMPUTER HI-TECH CO LTD	5329	0x14D1
COMPUTEX CO LTD	5451	0x154B
COMPUTONE CORPORATION	36366	0x8E0E
COMVERSE NETWORKS SYSTEM & Ulticom.	4820	0x12D4
CONCURRENT TECHNOLOGIES	4703	0x125F
CONDOR ENGINEERING INC	5062	0x13C6
CONEXANT	5361	0x14F1
CONTEC CO. LTD	4641	0x1221
CONTEMPORARY CONTROLS	5489	0x1571
CONTROLNET INC	4995	0x1383
CORECO INC	4588	0x11EC
COROLLARY	4492	0x118C
COYOTE TECHNOLOGIES LLC	5366	0x14F6
CREAMWARE GMBH	5301	0x14B5
CREATIVE ELECTRONIC SYSTEMS SA	4342	0x10F6
CREATIVE LABS	4354	0x1102
CREATIVE LABS. MALVERN	4724	0x1274
CREST MICROSYSTEM INC.	4417	0x1141
CRYPTEK	5212	0x145C
CRYSTAL GROUP INC	5024	0x13A0
CTI PET Systems	5294	0x14AE
CYBERFIRM INC.	5594	0x15DA
CYBERNETICS TECHNOLOGY CO LTD	5592	0x15D8
CYCLONE MICROSYSTEMS.	4412	0x113C
CYTEC CORPORATION	5506	0x1582
Chase Research	4832	0x12E0
Colorgraphic Communications Corp	4875	0x130B
Computer Boards	4871	0x1307
Connect Tech Inc	4804	0x12C4
D-LINK SYSTEM INC	4486	0x1186
DAEWOO TELECOM LTD	4208	0x1070
DAINIPPON SCREEN MFG. CO. LTD	4550	0x11C6

Company Name	Dec ID	Hex ID
DALLAS SEMICONDUCTOR	5098	0x13EA
DATA RACE INC	5318	0x14C6
DATACUBE	4375	0x1117
DATAKINETICS LTD	5357	0x14ED
DATALEX COMMUNCATIONS	5431	0x1537
DCM DATA SYSTEMS	5444	0x1544
DDK ELECTRONICS INC	5480	0x1568
DECISION COMPUTER INTERNATIONAL CO. 26214	0x6666	
DELL COMPUTER CORPORATION	4136	0x1028
DELTA ELECTRONICS INC	5529	0x1599
DELTA NETWORKS INC	16435	0x4033
DFI INC.	5565	0x15BD
DIAGNOSTIC INSTRUMENTS INC	5618	0x15F2
DIATREND CORPORATION	5240	0x1478
DIGALOG SYSTEMS INC	5514	0x158A
DIGI INTERNATIONAL	4431	0x114F
DIGIGRAM	4969	0x1369
DIGITAL AUDIO LABS INC	5404	0x151C
DIGITAL RECEIVER TECHNOLOGY INC	44062	0xAC1E
DIGITMEDIA CORP.	5619	0x15F3
DISTRIBUTED PROCESSING TECHNOLOGY	4164	0x1044
DITECT COOP	5519	0x158F
DIVA SYSTEMS CORP.	5525	0x1595
DIVERSIFIED TECHNOLOGY	4200	0x1068
DLoG GMBH	5046	0x13B6
DOLPHIN INTERCONNECT SOLUTIONS AS	4552	0x11C8
DOME IMAGING SYSTEMS INC	4590	0x11EE
DOUG CARSON & ASSOCIATES	5236	0x1474
DREAMTECH CO LTD	5581	0x15CD
DRSEARCH GMBH	5611	0x15EB
DSP RESEARCH INC	5130	0x140A
DTK COMPUTER	5314	0x14C2
DUAL TECHNOLOGY CORPORATION	5497	0x1579
DY4 Systems Inc	54484	0xD4D4
DYNACHIP CORPORATION	4989	0x137D
DYNARC INC	5216	0x1460
Datum Inc. Bancomm-Timing Division	4834	0x12E2
Dialogic Corp	4807	0x12C7
E-TECH INC	5087	0x13DF
EAGLE TECHNOLOGY	59905	0xEA01
EASTMAN KODAK	4530	0x11B2

Company Name	Dec ID	Hex ID
ECHELON CORPORATION	5426	0x1532
ECHOSTAR DATA NETWORKS	5022	0x139E
ECHOTEK CORPORATION	5399	0x1517
EDEC CO LTD	5160	0x1428
EFFICIENT NETWORKS	4378	0x111A
EICON TECHNOLOGY CORPORATION	4403	0x1133
EKF ELEKTRONIK GMBH	58559	0xE4BF
ELECTRONIC EQUIPMENT PRODUCTION & DISTRIBUTION	4983	0x1377
ELECTRONICS FOR IMAGING	4462	0x116E
ELITEGROUP COMPUTER SYS	4121	0x1019
ELSA AG	4168	0x1048
ELTEC ELEKTRONIK GMBH	5171	0x1433
EMC CORPORATION	4384	0x1120
EMTEC CO. LTD	5273	0x1499
EMULEX CORPORATION	4319	0x10DF
ENE TECHNOLOGY INC	5412	0x1524
ENGINEERING DESIGN TEAM.	4669	0x123D
ENNOVATE NETWORKS INC	5298	0x14B2
ENTRIDIA CORPORATION	5590	0x15D6
EPIGRAM INC	65242	0xFEDA
ERICSSON AXE R & D	5328	0x14D0
ERMA-ELECTRONIC GMBH	5253	0x1485
ESD Electronic System Design GmbH	4862	0x12FE
ESSENTIAL COMMUNICATIONS	4623	0x120F
ETRI	4184	0x1058
EUROPOP AG	5638	0x1606
EUROSOFT (UK) LTD	5500	0x157C
EVANS & SUTHERLAND	4317	0x10DD
EVERGREEN TECHNOLOGIES INC	5429	0x1535
EVSX	5572	0x15C4
EXAR CORP.	5032	0x13A8
EXCEL SWITCHING CORP	5145	0x1419
EXTREME PACKET DEVICE INC	5622	0x15F6
Equator Technologies	4821	0x12D5
FAIRCHILD SEMICONDUCTOR	5492	0x1574
FANUC LTD	5150	0x141E
FARADAY TECHNOLOGY CORP	5531	0x159B
FAST CORPORATION	5219	0x1463
FAST MULTIMEDIA AG	4350	0x10FE
FAST SEARCH & TRANSFER ASA	64087	0xFA57

Company Name	Dec ID	Hex ID
FASTPOINT TECHNOLOGIES INC.	5631	0x15FF
FCI ELECTRONICS	4376	0x1118
FEATRON TECHNOLOGIES CORPORATION	5288	0x14A8
FIC (FIRST INTERNATIONAL COMPUTER INC)	5586	0x15D2
FILANET CORPORATION	5437	0x153D
FIRST INTERNATIONAL COMPUTER INC	5385	0x1509
FLYTECH TECHNOLOGY CO LTD	5419	0x152B
FOLSOM RESEARCH INC	5526	0x1596
FORCE COMPUTERS GMBH	4422	0x1146
FORD MICROELECTRONICS INC	5106	0x13F2
FORE SYSTEMS INC	4391	0x1127
FORVUS RESEARCH INC	5386	0x150A
FOUNTAIN TECHNOLOGIES.	4169	0x1049
FOXCONN INTERNATIONAL INC	4187	0x105B
FUJI XEROX CO LTD	4405	0x1135
FUJIFILM	4735	0x127F
FUJITSU COMPUTER PRODUCTS OF AMERICA	5405	0x151D
FUJITSU LIMITED	4303	0x10CF
FUJITSU MICROELECTRONIC	4298	0x10CA
FUJITSU MICROELECTRONICS LTD.	4510	0x119E
FUNDAMENTAL SOFTWARE INC	5124	0x1404
FUTUREPLUS SYSTEMS CORP.	4305	0x10D1
ForteMedia	4889	0x1319
Fujifilm Microdevices	4799	0x12BF
G2 NETWORKS.	4749	0x128D
GALEA NETWORK SECURITY	5535	0x159F
GALILEO TECHNOLOGY LTD.	4523	0x11AB
GARNETS SYSTEM CO LTD	5353	0x14E9
GATEWAY 2000	4219	0x107B
GE VINGMED ULTRASOUND AS	4819	0x12D3
GEMFLEX NETWORKS	5501	0x157D
GENERAL INSTRUMENT	5530	0x159A
GENRAD INC.	5582	0x15CE
GENROCO INC	21845	0x5555
GEOCAST NETWORK SYSTEMS INC	5537	0x15A1
GESPAC	4880	0x1310
GESYTEC GMBH	5461	0x1555
GET ENGINEERING CORP.	5607	0x15E7
GIGA-BYTE TECHNOLOGY	5208	0x1458
GIGAPIXEL CORP	37274	0x919A
GLOBESPAN SEMICONDUCTOR INC.	5308	0x14BC

Company Name	Dec ID	Hex ID
GLOBETEK INC	5402	0x151A
GN NETTEST TELECOM DIV.	5221	0x1465
GRANITE MICROSYSTEMS	5528	0x1598
GRAPHICS MICROSYSTEMS INC	5076	0x13D4
GRAPHIN CO. LTD	5190	0x1446
GROWTH NETWORKS	18755	0x4943
GUILLEMOT CORPORATION	5295	0x14AF
GUZIK TECHNICAL ENTERPRISES	4691	0x1253
GVC CORPORATION	5088	0x13E0
GVC/BCM ADVANCED RESEARCH	5284	0x14A4
HAMAMATSU PHOTONICS K.K.	4513	0x11A1
HERMES ELECTRONICS COMPANY	4394	0x112A
HEWLETT PACKARD	41561	0xA259
HIGH TECH COMPUTER CORP (HTC)	5567	0x15BF
HILSCHER GMBH	5583	0x15CF
HINT CORP	13192	0x3388
HIRAKAWA HEWTECH CORP	5335	0x14D7
HITACHI COMPUTER PRODUCTS	4128	0x1020
HITACHI INFORMATION TECHNOLOGY CO LTD	5000	0x1388
HITACHI SEMICONDUCTOR & DEVICES SALES CO LTD	5516	0x158C
HITACHI ULSI SYSTEMS CO LTD	4688	0x1250
HITACHI ZOSEN CORPORATION	4967	0x1367
HITACHI	4180	0x1054
HITT	5496	0x1578
HIVERTEC INC.	5289	0x14A9
HOLTEK SEMICONDUCTOR INC	4803	0x12C3
HONDA CONNECTORS/MHOTRONICS INC	5384	0x1508
HONEYWELL IAC	4268	0x10AC
HOPF ELEKTRONIK GMBH	5336	0x14D8
HOTRAIL INC.	5580	0x15CC
HTEC LTD	5383	0x1507
I-BUS	4217	0x1079
I-DATA INTERNATIONAL A-S	4959	0x135F
I-O DATA DEVICE.	4348	0x10FC
IBM	4116	0x1014
ICOMPRESION INC.	17476	0x4444
ICP-VORTEX COMPUTERSYSTEM GMBH	4377	0x1119
ICS ADVENT	5397	0x1515
IKON CORPORATION	4565	0x11D5
IMAGING TECHNOLOGY	4399	0x112F

Company Name	Dec ID	Hex ID
IMC NETWORKS	5075	0x13D3
IMODL INC.	5341	0x14DD
IMPACCT TECHNOLOGY CORP	5562	0x15BA
IMPACT TECHNOLOGIES	5413	0x1525
IN WIN DEVELOPMENT INC.	5614	0x15EE
INET TECHNOLOGIES INC	5507	0x1583
INFIMED	4800	0x12C0
INFINEON TECHNOLOGIES AG	5585	0x15D1
INFINILINK CORP.	5599	0x15DF
INFOLIBRIA	5346	0x14E2
INFOTRONIC AMERICA INC	4191	0x105F
INITIO CORPORATION	4353	0x1101
INNOMEDIA INC	5466	0x155A
INNOMEDIALOGIC INC.	5259	0x148B
INNOSYS	4521	0x11A9
INOVA COMPUTERS GMBH & CO KG	5286	0x14A6
INTEC GMBH	5391	0x150F
INTEGRATED DEVICE TECH	4381	0x111D
INTEGRATED TECHNOLOGY EXPRESS.	4739	0x1283
INTEGRATED TELECOM EXPRESS INC	5233	0x1471
INTEL CORP.	32902	0x8086
INTELLIGENT PARADIGM INC	5615	0x15EF
INTERACTIVE CIRCUITS & SYSTEMS LTD	5220	0x1464
INTERCOM INC.	4562	0x11D2
INTERCONNECT SYSTEMS SOLUTIONS	5449	0x1549
INTERNIX INC.	5306	0x14BA
INTERPHASE CORPORATION	4222	0x107E
INTERSIL CORP	4704	0x1260
INTRASERVER TECHNOLOGY INC	5097	0x13E9
INVENTEC CORPORATION	4464	0x1170
INVERTEX	5345	0x14E1
IOI TECHNOLOGY CORP.	5446	0x1546
IOMEGA CORPORATION	5066	0x13CA
ISS	5414	0x1526
ISYTEC-Integrierte Systemtechnik Gmbh	5250	0x1482
ITA INGENIEURBURO FUR TESTAUFGABEN GMBH	5381	0x1505
ITALTEL	5539	0x15A3
ITT AEROSPACE/COMMUNICATIONS DIVISION	5168	0x1430
IWASAKI INFORMATION SYSTEMS CO LTD	5316	0x14C4
IWATSU ELECTRIC CO LTD	4988	0x137C
IWILL CORPORATION	5588	0x15D4

Company Name	Dec ID	Hex ID
Integrated Computing Engines	4810	0x12CA
J P AXZAM CORPORATION	5626	0x15FA
JANZ COMPUTER AG	5059	0x13C3
JAPAN COMPUTER INDUSTRY INC.	5373	0x14FD
JAPAN ELECRONICS IND. INC	5498	0x157A
JAYCOR NETWORKS INC.	4674	0x1242
JET PROPULSION LABORATORY	5448	0x1548
JOYTECH COMPUTER CO. LTD.	5270	0x1496
JUNGSOFT	5479	0x1567
Jaton Corp	6931	0x1B13
Juniper Networks Inc.	4868	0x1304
K.I. TECHNOLOGY CO LTD	5078	0x13D6
KAISER ELECTRONICS	5380	0x1504
KAWASAKI HEAVY INDUSTRIES LTD	5025	0x13A1
KAWASAKI LSI USA INC	5379	0x1503
KAWASAKI STEEL CORPORATION	4971	0x136B
KINGMAX TECHNOLOGY INC	5162	0x142A
KINPO ELECTRONICS INC	5630	0x15FE
KNOWLEDGE TECHNOLOGY LAB.	4761	0x1299
KOGA ELECTRONICS CO	5624	0x15F8
KOLTER ELECTRONIC	4097	0x1001
KONICA CORPORATION	5511	0x1587
KYE SYSTEMS CORPORATION	5257	0x1489
KYOPAL CO LTD	5388	0x150C
KYUSHU ELECTRONICS SYSTEMS INC	5144	0x1418
L3 COMMUNICATIONS	5310	0x14BE
LABWAY COPORATION	5251	0x1483
LANCAST INC	5510	0x1586
LANTECH COMPUTER COMPANY	5376	0x1500
LARA TECHNOLOGY INC	5518	0x158E
LATTICE-VANTIS	5491	0x1573
LAVA COMPUTER MFG INC	5127	0x1407
LAVA SEMICONDUCTOR MANUFACTURING INC.	5639	0x1607
LECROY CORPORATION	5488	0x1570
LECTRON CO LTD	5279	0x149F
LEVEL ONE COMMUNICATIONS	5012	0x1394
LEVEL ONE COMMUNICATIONS	4872	0x1308
LIGHTWELL CO LTD-ZAX DIVISION	5183	0x143F
LITE-ON COMMUNICATIONS INC	4525	0x11AD
LITRONIC INC	5596	0x15DC
LOCKHEED MARTIN-Electronics & Communications	4560	0x11D0

Company Name	Dec ID	Hex ID
LOGIC PLUS PLUS INC	5205	0x1455
LOGICAL CO LTD	5189	0x1445
LOGITEC CORP.	25609	0x6409
LOGITRON	5509	0x1585
LORONIX INFORMATION SYSTEMS INC	5195	0x144B
LP ELEKTRONIK GMBH	5470	0x155E
LSI LOGIC CORPORATION	4138	0x102A
LSI SYSTEMS	4554	0x11CA
LUCENT TECHNOLOGIES	4771	0x12A3
M-SYSTEMS FLASH DISK PIONEERS LTD	5487	0x156F
MAC SYSTEM CO LTD	5469	0x155D
MACRAIGOR SYSTEMS LLC	5420	0x152C
MACROLINK INC	5613	0x15ED
MADGE NETWORKS	4278	0x10B6
MAESTRO DIGITAL COMMUNICATIONS	5561	0x15B9
MAGMA	4553	0x11C9
MAINPINE LIMITED	5410	0x1522
MAKER COMMUNICATIONS	5267	0x1493
MALLEABLE TECHNOLOGIES INC	5598	0x15DE
MAPLETREE NETWORKS INC.	5278	0x149E
MARCONI COMMUNICATIONS LTD	4658	0x1232
MARK OF THE UNICORN INC	4986	0x137A
MASPRO KENKOH CORP	5358	0x14EE
MATRIX CORP.	5406	0x151E
MATROX GRAPHICS.	4139	0x102B
MATSUSHITA ELECTIC INDUSTRIAL CO LTD	4489	0x1189
MATSUSHITA ELECTRIC WORKS LTD	5133	0x140D
MATSUSHITA-KOTOBUKI ELECTRONICS	4705	0x1261
MAVERICK NETWORKS	5283	0x14A3
MAX TECHNOLOGIES INC.	5450	0x154A
MAZET GMBH	4742	0x1286
MEDIA 100	4374	0x1116
MEDIAQ INC.	19793	0x4D51
MEDIASTAR CO. LTD	5463	0x1557
MEDIATEK CORP.	5315	0x14C3
MEIDENSHA CORPORATION	4256	0x10A0
MEILHAUS ELECTRONIC GmbH	5122	0x1402
MEINBERG FUNKUHREN	4960	0x1360
MELCO INC	4436	0x1154
MELEC INC	5422	0x152E
MELLANOX TECHNOLOGY	5555	0x15B3

Company Name	Dec ID	Hex ID
MEMEC DESIGN SERVICES	5527	0x1597
MENTOR GRAPHICS CORP.	5291	0x14AB
MERCURY COMPUTER SYSTEMS	4404	0x1134
METHEUS CORPORATION	5068	0x13CC
MICRO COMPUTER SYSTEMS INC	4271	0x10AF
MICRO INDUSTRIES CORPORATION	4325	0x10E5
MICRO SCIENCE INC	5117	0x13FD
MICRO-STAR INTERNATIONAL CO LTD	5218	0x1462
MICRON TECHNOLOGY INC	4932	0x1344
MICROTECHNICA CO LTD	19796	0x4D54
MILLENNIUM ENGINEERING INC	5282	0x14A2
MINDSHARE.	4506	0x119A
MINTON OPTIC INDUSTRY CO LTD	5164	0x142C
MIPS DENMARK	5439	0x153F
MITAC	4209	0x1071
MITEL CORP	4402	0x1132
MITSUBISHI ELECTRIC AMERICA	4199	0x1067
MITSUBISHI ELECTRIC CORP.	4282	0x10BA
MITSUBISHI ELECTRIC LOGISTICS SUPPORT CO	5378	0x1502
MITUTOYO CORPORATION	5447	0x1547
MOBILITY ELECTRONICS	5362	0x14F2
MODULAR TECHNOLOY HOLDINGS LTD	5319	0x14C7
MOLEX INCORPORATED	4306	0x10D2
MOMENTUM DATA SYSTEMS	4406	0x1136
MORETON BAY	5546	0x15AA
MOSAID TECHNOLOGIES INC.	5554	0x15B2
MOTION ENGINEERING.	49406	0xC0FE
MOTOROLA	49374	0xC0DE
MOXA TECHNOLOGIES CO LTD	5011	0x1393
MUSIC SEMICONDUCTORS	5411	0x1523
MYCOM INC	5203	0x1453
MYLEX CORPORATION	4201	0x1069
MYRICOM INC.	5313	0x14C1
MYSON TECHNOLOGY INC	5398	0x1516
Micron Electronics.	4162	0x1042
Mitan Corporation	4806	0x12C6
Mitsubishi Electric MicroComputer	4874	0x130A
N-CUBED.NET	5629	0x15FD
NAKAYO TELECOMMUNICATIONS INC	5324	0x14CC
NATIONAL AEROSPACE LABORATORIES	5338	0x14DA
NATIONAL DATACOMM CORP.	5608	0x15E8

Company Name	Dec ID	Hex ID
NATIONAL SEMICONDUCTOR CORPORATION	4107	0x100B
NATURAL MICROSYSTEMS	4790	0x12B6
NCIPHER CORP. LTD	256	0x0100
NCR	4122	0x101A
NCS COMPUTER ITALIA SRL	4753	0x1291
NDS TECHNOLOGIES ISRAEL LTD	5587	0x15D3
NEC CORPORATION	4147	0x1033
NEOMAGIC CORPORATION	4296	0x10C8
NEST INC	5091	0x13E3
NET INSIGHT	5239	0x1477
NETACCESS	4558	0x11CE
NETBOOST CORPORATION	5084	0x13DC
NETGAME LTD	5524	0x1594
NETGEAR	4997	0x1385
NETWORK APPLIANCE CORPORATION	4725	0x1275
NETWORTH TECHNOLOGIES INC	5603	0x15E3
NEW WAVE PDG	4575	0x11DF
NEWER TECHNOLOGY INC	5570	0x15C2
NEWTEK INC	5277	0x149D
NEXTCOM K.K.	5297	0x14B1
NIHON UNISYS	5247	0x147F
NINGBO HARRISON ELECTRONICS CO LTD	5533	0x159D
NISSIN INC CO	5175	0x1437
NITSUKO CORPORATION	5333	0x14D5
NKK CORPORATION	4341	0x10F5
NOKIA TELECOMMUNICATIONS OY	5048	0x13B8
NOKIA WIRELESS BUSINESS COMMUNICATIONS	5635	0x1603
NORTEL NETWORKS	4716	0x126C
NORTEL NETWORKS-BWA DIVISION	5034	0x13AA
NORTH ATLANTIC INSTRUMENTS	5548	0x15AC
NORTHROP GRUMMAN-CANADA LTD	5632	0x1600
NOVAWEB TECHNOLOGIES INC	5292	0x14AC
NOVELL	4570	0x11DA
NTT ADVANCED TECHNOLOGY CORP.	5113	0x13F9
NUMBER 9 VISUAL TECHNOLOGY	4189	0x105D
NVIDIA CORPORATION	4318	0x10DE
O2MICRO.	4631	0x1217
OCE'-TECHNOLOGIES B.V.	5105	0x13F1
OCE' PRINTING SYSTEMS GmbH	5126	0x1406
OCEAN MANUFACTURING LTD	4195	0x1063
OCTAVE COMMUNICATIONS IND.	5200	0x1450

Company Name	Dec ID	Hex ID
ODIN TELESYSTEMS INC	5321	0x14C9
OKI ELECTRIC INDUSTRY CO. LTD.	4129	0x1021
OLICOM	4237	0x108D
OLYMPUS OPTICAL CO. LTD.	4720	0x1270
OMNI MEDIA TECHNOLOGY INC.	38553	0x9699
OMRON CORPORATION	4299	0x10CB
ONO SOKKI	5434	0x153A
OPEN NETWORK CO LTD	5456	0x1550
OPTI INC.	4165	0x1045
OPTIBASE LTD	4693	0x1255
OPTO 22	5258	0x148A
OSI PLUS CORPORATION	5262	0x148E
OSITECH COMMUNICATIONS INC	5026	0x13A2
OTIS ELEVATOR COMPANY	5490	0x1572
OVISLINK CORP.	5276	0x149C
OXFORD SEMICONDUCTOR LTD	5141	0x1415
PACIFIC DIGITAL CORP.	5609	0x15E9
PACKARD BELL NEC	4250	0x109A
PAIRGAIN TECHNOLOGIES	5637	0x1605
PALIT MICROSYSTEMS INC	5481	0x1569
PAN INTERNATIONAL INDUSTRIAL CORP	5453	0x154D
PANACOM TECHNOLOGY CORP	5332	0x14D4
PARADYNE CORP.	51	0x0033
PATAPSCO DESIGNS INC	5007	0x138F
PC-TEL INC	4941	0x134D
PE LOGIC CORP.	5322	0x14CA
PENTA MEDIA CO. LTD	5576	0x15C8
PENTEK	4848	0x12F0
PEP MODULAR COMPUTERS GMBH	5400	0x1518
PERFORMANCE TECHNOLOGIES.	4628	0x1214
PERICOM SEMICONDUCTOR	4824	0x12D8
PERLE SYSTEMS LIMITED	5471	0x155F
PFU LIMITED	4449	0x1161
PHILIPS-CRYPTO	5423	0x152F
PHILIPS BUSINESS ELECTRONICS B.V.	5300	0x14B4
PHILIPS SEMICONDUCTORS	4401	0x1131
PHOBOS CORPORATION	5080	0x13D8
PHOENIX TECHNOLOGIES LTD	4963	0x1363
PHOTRON LTD.	4444	0x115C
PIXELFUSION LTD	5349	0x14E5
PIXSTREAM INC	5165	0x142D

Company Name	Dec ID	Hex ID
PLANEX COMMUNICATIONS INC	5354	0x14EA
PLANT EQUIPMENT.	5263	0x148F
PLATYPUS TECHNOLOGY PTY LTD	4491	0x118B
PLD APPLICATIONS	5462	0x1556
PLX TECHNOLOGY.	4277	0x10B5
PMC-SIERRA INC	4600	0x11F8
POINT MULTIMEDIA SYSTEMS	5517	0x158D
PORTWELL INC	5563	0x15BB
POWER MICRO RESEARCH	5621	0x15F5
PPT VISION	4987	0x137B
PRIMEX AEROSPACE CO.	5504	0x1580
PRISA NETWORKS	4925	0x133D
PROCOMP INFORMATICS LTD	5573	0x15C5
PROLINK MICROSYSTEMS CORP.	5460	0x1554
PROMAX SYSTEMS INC	4930	0x1342
PROMISE TECHNOLOGY.	4186	0x105A
PROSYS-TEC INC.	5634	0x1602
PROTAC INTERNATIONAL CORP	5467	0x155B
PROVIDEO MULTIMEDIA CO LTD	5440	0x1540
PROXIM INC	5303	0x14B7
PSION DACOM PLC	5152	0x1420
PURUP-EskoFot A/S	4630	0x1216
PX INSTRUMENTS TECHNOLOGY LTD	5503	0x157F
Packet Engines Inc.	4888	0x1318
QLOGIC	4215	0x1077
QUADRICS SUPERCOMPUTERS WORLD	5372	0x14FC
QUANTA COMPUTER INC	5421	0x152D
QUANTEL	5569	0x15C1
QUANTUM 3D INC	5020	0x139C
QUANTUM DATA CORP.	5302	0x14B6
QUANTUM DESIGNS (H.K.) INC.	13329	0x3411
QUANTUM EFFECT DESIGN	4258	0x10A2
QUATECH INC	4956	0x135C
QUICKLOGIC CORPORATION	4579	0x11E3
QUICKNET TECHNOLOGIES INC	5602	0x15E2
QUICKTURN DESIGN SYSTEMS	5418	0x152A
RACAL AIRTECH LTD	5458	0x1552
RADIOLAN	5163	0x142B
RAMIX INC	5131	0x140B
RASCOM INC	5028	0x13A4
RATOC SYSTEMS INC	4501	0x1195

Company Name	Dec ID	Hex ID
RAYCER INC	5352	0x14E8
RAYCHEM	5395	0x1513
REAL 3D	61	0x003D
REALTEK SEMICONDUCTOR CORP.	4332	0x10EC
RENDITION	4451	0x1163
RICOH CO LTD	4480	0x1180
RIOS SYSTEMS CO LTD	5017	0x1399
ROAD CORPORATION	5428	0x1534
ROCKWELL-COLLINS	5591	0x15D7
ROHM LSI SYSTEMS	4315	0x10DB
ROSUN TECHNOLOGIES INC	5394	0x1512
RUBY TECH CORP.	5228	0x146C
RadiSys Corp.	4913	0x1331
Rainbow Technologies	4830	0x12DE
Real Vision	4842	0x12EA
Reliance Computer	4454	0x1166
S S TECHNOLOGIES	20790	0x5136
S3 INC.	21299	0x5333
SALIX TECHNOLOGIES INC	4901	0x1325
SAMSUNG ELECTRONICS CO LTD	5197	0x144D
SANDISK CORP.	5559	0x15B7
SANRITZ AUTOMATION CO LTC	4992	0x1380
SANTA CRUZ OPERATION	4369	0x1111
SANYO ELECTRIC CO-Information Systems Division	4414	0x113E
SBS TECHNOLOGIES	4683	0x124B
SBS Technologies Inc	4831	0x12DF
SCIEMETRIC INSTRUMENTS INC	5641	0x1609
SCITEX CORPORATION	4526	0x11AE
SCM MICROSYSTEMS	4927	0x133F
SEALEVEL SYSTEMS INC	4958	0x135E
SEANIX TECHNOLOGY INC	19617	0x4CA1
SEH COMPUTERTECHNIK GMBH	5505	0x1581
SEIKO EPSON CORPORATION	5355	0x14EB
SEIKO INSTRUMENTS INC	5275	0x149B
SEMTECH CORPORATION	5307	0x14BB
SEQUENT COMPUTER SYSTEMS	4205	0x106D
SEROME TECHNOLOGY INC	5577	0x15C9
SERVOTEST LTD	5454	0x154E
SHANGHAI COMMUNICATIONS TECHNOLOGIES CENTER	5544	0x15A8
SHAREWAVE INC	5055	0x13BF

Company Name	Dec ID	Hex ID
SHARK MULTIMEDIA INC	5074	0x13D2
SHARP CORPORATION	5053	0x13BD
SHINING TECHNOLOGY INC	5350	0x14E6
SHUTTLE COMPUTER	4759	0x1297
SI LOGIC LTD	5465	0x1559
SICAN GMBH	4652	0x122C
SIEMENS MEDICAL SYSTEMS	5033	0x13A9
SIEMENS PC SYSTEME GMBH	4362	0x110A
SIGMA DESIGNS	4357	0x1105
SIGMATEL INC.	5597	0x15DD
SIIG Inc	4895	0x131F
SILICON GRAPHICS	4265	0x10A9
SILICON INTEGRATED SYSTEMS	4153	0x1039
SILICON LABORATORIES	5443	0x1543
SILICON MAGIC CORP.	34952	0x8888
SILICON MOTION.	4719	0x126F
SITERA	5002	0x138A
SKYWARE CORPORATION	4968	0x1368
SMA REGELSYSTEME GMBH	5271	0x1497
SMART ELECTRONIC DEVELOPMENT GMBH	5457	0x1551
SOFTING GMBH	5280	0x14A0
SOLA ELECTRONICS	5566	0x15BE
SOLECTRON	5415	0x1527
SOLIDUM SYSTEMS CORP	5512	0x1588
SOLITON SYSTEMS K.K.	4961	0x1361
SONY CORPORATION	4173	0x104D
SOPAC LTD	5365	0x14F5
SOURCE TECHNOLOGY INC	5553	0x15B1
SP3D CHIP DESIGN GMBH	5201	0x1451
SPECIALIX INTERNATIONAL LTD	4555	0x11CB
SPIDER COMMUNICATIONS INC.	5311	0x14BF
SPLASH TECHNOLOGY.	4717	0x126D
SSE TELECOM INC	5543	0x15A7
STAR MULTIMEDIA CORP.	5499	0x157B
STELLAR SEMICONDUCTOR INC	4996	0x1384
STRATABEAM TECHNOLOGY	5455	0x154F
STRATUS COMPUTER SYSTEMS	5532	0x159C
STUDIO AUDIO & VIDEO LTD	5071	0x13CF
SUMITOMO METAL INDUSTRIES	4718	0x126E
SUNDANCE TECHNOLOGY INC	5104	0x13F0
SUNLIGHT ULTRASOUND TECHNOLOGIES LTD	5542	0x15A6

Company Name	Dec ID	Hex ID
SUPER MICRO COMPUTER INC	5593	0x15D9
SYBA TECH LIMITED	5522	0x1592
SYMBIOS LOGIC INC/LSI Logic	4096	0x1000
SYMBOL TECHNOLOGIES	5474	0x1562
SYNOPSYS/LOGIC MODELING GROUP	4159	0x103F
SYSKONNECT	4424	0x1148
SYSTEMBASE CO LTD	5281	0x14A1
SYSTRAN CORP	4999	0x1387
SeaChange International	4902	0x1326
Sebring Systems	4839	0x12E7
Spectrum Signal Processing	4859	0x12FB
Standard Microsystems Corp.	4181	0x1055
T.SQWARE	5039	0x13AF
TACHYON.	5229	0x146D
TAIWAN MYCOMP CO LTD	5571	0x15C3
TAMURA CORPORATION	5041	0x13B1
TATENO DENNOU.	4751	0x128F
TATEYAMA SYSTEM LABORATORY CO LTD	5575	0x15C7
TATUNG CO.	5589	0x15D5
TC LABS PTY LTD.	5264	0x1490
TECH-SOURCE	4647	0x1227
TECHNICAL UNIVERSITY OF BUDAPEST	5574	0x15C6
TECHNOTREND SYSTEMTECHNIK GMBH	5058	0x13C2
TECHSAN ELECTRONICS CO LTD	5628	0x15FC
TECHSOFT TECHNOLOGY CO LTD	5304	0x14B8
TECHWELL INC	5438	0x153E
TEK MICROSYSTEMS INC.	5327	0x14CF
TEKNOR INDUSTRIAL COMPUTERS INC	4185	0x1059
TEKRAM TECHNOLOGY CO.LTD.	4321	0x10E1
TEKTRONIX	4712	0x1268
TELEFON AKTIEBOLAGET LM Ericsson	5401	0x1519
TELES AG	5031	0x13A7
TELESOFT DESIGN LTD	5093	0x13E5
TELOSITY INC.	5441	0x1541
TEMPORAL RESEARCH LTD	8193	0x2001
TENTA TECHNOLOGY	5633	0x1601
TERADYNE INC.	4886	0x1316
TERALOGIC INC	21580	0x544C
TERAYON COMMUNICATIONS SYSTEMS	5472	0x1560
TERRATEC ELECTRONIC GMBH	5435	0x153B
TEXAS INSTRUMENTS	4172	0x104C

Company Name	Dec ID	Hex ID
TEXAS MEMORY SYSTEMS INC	5558	0x15B6
TFL LAN INC	5396	0x1514
TIME SPACE RADIO AB	5293	0x14AD
TIMES N SYSTEMS INC	5617	0x15F1
TITAN ELECTRONICS INC	5330	0x14D2
TOKAI COMMUNICATIONS INDUSTRY CO. LTD	5269	0x1495
TOKIMEC INC	5003	0x138B
TOKYO DENSHI SEKEI K.K.	5610	0x15EA
TOKYO ELECTRONIC INDUSTRY CO LTD	5364	0x14F4
TOPIC SEMICONDUCTOR CORP	5407	0x151F
TOSHIBA AMERICA INFO SYSTEMS	4473	0x1179
TOSHIBA AMERICA	4143	0x102F
TOSHIBA ENGINEERING CORPORATION	5079	0x13D7
TOSHIBA PERSONAL COMPUTER SYSTEM CORP.	4752	0x1290
TOSHIBA TEC CORPORATION	4569	0x11D9
TOYOTA MACS INC	5541	0x15A5
TRANSAS MARINE (UK) LTD	5371	0x14FB
TRANSITION NETWORKS	5502	0x157E
TRANSMETA CORPORATION	4729	0x1279
TRANSTECH DSP LTD	4728	0x1278
TRANSWITCH CORPORATION	4747	0x128B
TRIDENT MICROSYSTEMS	4131	0x1023
TRIGEM COMPUTER INC.	4255	0x109F
TRITECH MICROELECTRONICS INC	4754	0x1292
TROIKA NETWORKS INC	5108	0x13F4
TUNDRA SEMICONDUCTOR CORP	4323	0x10E3
TURBOCOMM TECH. INC.	5320	0x14C8
TWINHEAD INTERNATIONAL CORP	5375	0x14FF
TYAN COMPUTER	4337	0x10F1
True Time Inc.	4826	0x12DA
UNEX TECHNOLOGY CORP	5161	0x1429
UNISYS CORPORATION	4120	0x1018
UNIVERSAL SCIENTIFIC IND	5325	0x14CD
UNIWILL COMPUTER CORP	5508	0x1584
V3 SEMICONDUCTOR INC.	4528	0x11B0
VALLEY TECHNOLOGIES INC	5605	0x15E5
VALUESOFT	5620	0x15F4
VARIAN AUSTRIALIA PTY LTD	51792	0xCA50
VELA RESEARCH LP	4733	0x127D
VIA TECHNOLOGIES.	4358	0x1106
VICTOR COMPANY OF JAPAN	4766	0x129E

Company Name	Dec ID	Hex ID
VIDAC ELECTRONICS GMBH	5484	0x156C
VIDEO LOGIC LTD	4112	0x1010
VIEWCAST COM	5494	0x1576
VIEWGRAPHICS INC	5473	0x1561
VIRATA LTD	4635	0x121B
VISIONTEK	5445	0x1545
VISUAL TECHNOLOGY INC.	5452	0x154C
VIVID TECHNOLOGY INC	5442	0x1542
VLSI TECHNOLOGY INC	4100	0x1004
VMETRO.	4762	0x129A
VMWARE	5549	0x15AD
VOICE TECHNOLOGIES GROUP INC	5601	0x15E1
VOLTAIRE ADVANCED DATA SECURITY LTD	5493	0x1575
VSN SYSTEMEN BV	5604	0x15E4
WARPSPPED INC	5389	0x150D
WAVETEK WANDEL & GOLTERMANN	5370	0x14FA
WELLBEAN CO INC	5044	0x13B4
WHISTLE COMMUNICATIONS	5326	0x14CE
WILLIAMS ELECTRONICS GAMES.	5230	0x146E
WINBOND ELECTRONICS CORP	4176	0x1050
WOLF TECHNOLOGY INC	5367	0x14F7
WORKBIT CORPORATION	4421	0x1145
X-NET OY	5540	0x15A4
XILINX.	4334	0x10EE
XIONICS DOCUMENT TECHNOLOGIES INC.	5285	0x14A5
XIOTECH CORPORATION	4777	0x12A9
XIRCOM	4445	0x115D
XPEED INC.	5299	0x14B3
XSTREAMS PLC/ EPL LIMITED	5021	0x139D
YAMAHA CORPORATION	4211	0x1073
YAMAKATSU ELECTRONICS INDUSTRY CO LTD	5476	0x1564
YAMASHITA SYSTEMS CORP	5387	0x150B
YASKAWA ELECTRIC CO. 4883	0x1313	
YOKOGAWA ELECTRIC CORPORATION	4737	0x1281
YUAN YUAN ENTERPRISE CO. LTD.	4779	0x12AB
ZAPEX TECHNOLOGIES INC	5235	0x1473
ZENITH ELECTRONICS CORPORATION	5625	0x15F9
ZIATECH CORPORATION	4408	0x1138
ZILOG INC.	5627	0x15FB
ZOLTRIX INTERNATIONAL LIMITED	5552	0x15B0
ZOOM TELEPHONICS INC	5147	0x141B

PCI SLOT CONFIGURATION

Although an unlimited number of PCI slots is allowed, in practice 4 is the maximum, due to the capabilities of the host controller, which connects the bus to the CPU and DRAM, so bridge devices are used to connect more buses downstream from the first, known as the root, up to 255 (this is how 6 PCI slots can be obtained). However, these extra buses don't have to be PCI; they can be EISA or ISA as well. x86 chips generate two interrupt acknowledge cycles per interrupt; both are converted to one for PCI. As the PCI interrupt system finds it difficult to cope with expansion cards requiring IRQs for each device on them, I/O devices tend to be on the motherboard.

PCI cards and slots use an internal interrupt system, with each slot able to activate up to 4, labelled either INT#A-INT#D, or INT#1-INT#4, but they can sometimes be assigned to cards instead - if you get a problem, it often helps just to change the slot. INTs #A or #1 are always reserved for the Master function of the device concerned, and the remainder for multifunction cards. These are nothing to do with IRQs, although they can be mapped (that is, steered) to them if the card concerned needs it. Typically IRQs 9 and 10 are reserved for this, but any available can be used. There are various ways of implementing this, so don't expect consistency! AGP cards use only INT A and B, and share with PCI Slot #1. PCI Slots 4 & 5 also share, so try not to mix them, or at least only put in cards that can share IRQs.

Four registers control the routing of PCI Interrupts to IRQs, two or more of which can be steered into the same IRQ signal, each of which must be set to level sensitive (see *Edge/Level Select*) so they can be shared. The IRQs affected are IRQs 3-7, 9-12, and 14-15.

ISA cards cannot share IRQs because they are Edge triggered and rely on a single voltage, but PCI cards use Level triggering, which uses different voltage levels. Also, an ISA IRQ is available to every slot, so once the card is set up it can be used in any one. On a PCI PC, the 16 standard IRQs can be set individually for PCI or ISA, but not both-PCI or IRQ Steering is another name for sharing IRQs between PCI devices which is supported by Windows 95 OSR2 and 98, and gives them the ability to reprogram PCI IRQs when mixed with non-PnP ISA devices. However, it is not enabled in OSR2 (Error Code 29 in Device Manager, for the PCI bus under System Devices-just check the box for IRQ steering under Properties. Check also Get IRQ table from PCI BIOS 2.1 Call), which means that the BIOS does all the work, as it would for previous versions. In practice, OSR2 and 98 will accept what the BIOS has already decreed, even though it can change them if it wants to.

In a real world situation, it is common for Windows to share an IRQ between the sound and VGA cards. In the BIOS, you can manually assign IRQ5 for a sound card in whatever slot, which is where most games like to see it, and you may get better stability. In the BIOS setup (the *PCI/PnP Configuration* section), you may see each slot listed with these subheadings:

```
Slot 1
  Latency Timer
  Using IRQ
  Trigger Method
```

A PCI Master can burst as long as the target can send or receive data, and no other device requests the bus. PCI specifies two ways of disconnecting a Master during a long burst cycle so others can get a look in; Master Latency Timer and Target Initiated Termination.

High Priority PCI Mode

Gives a higher priority to the first PCI slot for performance, such as when a Firewire (IEEE 1394) card is installed.

PCI Master Read Caching

Enable for the Thunderbird and disable for the Duron. It concerns the ability of the controller to read data ahead of the PCI Master (it's placed into a buffer). Being cacheable,a snoop to the CPU must be forced.

Resources Controlled By

Whether you let the BIOS assign resources (Auto), or do it yourself (Manual).

If you have problems with Auto, Manual reveals the IRQ and DMA fields so you can assign them to either Legacy ISA or PCI/ISA PnP devices.

Force Updating ESCD

If enabled, the ESCD area in Upper Memory (for PnP information concerning IRQ, DMA, I/O and memory) will be updated once, then this setting will be disabled automatically for the next boot. Use if you have a new card and the subsequent reconfiguration causes a serious conflict of resources (the OS may not boot as a result). The BIOS will reallocate everything.

Clear NVRAM

See above.

430HX Global Features

Enable or disable special features. Enabled is best for performance.

APIC Function

APIC stands for *Advanced Programmable Interrupt Controller*, which is a chip (or a collection of them) that provides symmetric multiprocessing capability on Pentium systems. It's a new set of devices to perform an old job, although the usual 8259 PIC, when fitted, still collects interrupt signals and feeds them to a local APIC, which is actually on the processor die (since the P54C), with up to 8 I/O APICs and APIC buses to each device somewhere else in the system, which is handy because APICs are specific to processors. The I/O APICs collect interrupt requests and send them to the local APICs, thus performing the same function as the 8259 (they can have up to 64 inputs, but 24 is more typical in Intel machines).

This setting will be available on multiprocessor boards, since APICs are only supported by NT, 2000 and XP, and probably Linux, or any system that does not need to support DOS device drivers. Disabling this forces the APIC to behave like an 8259 and will give best behaviour when DOS is involved, as with Windows 98. Otherwise, enable if your system is properly 32-bit, to give a vastly expanded range of IRQs. Having an APIC complies with PC 2001 design specifications.

IO APIC

See above.

APIC Mode

See above.

Interrupt Mode [APIC]

See above.

Latency Timer (PCI Clocks)

Controls how long an agent can hold the PCI bus when another has requested it, so it guarantees a PCI card access within a specified number of clocks. Each PCI slot has a certain number of clock cycles for uninterrupted access to the system bus or CPU.

Since the PCI bus runs faster than ISA, the PCI bus must be slowed during interactions with it, so here you can define how long the PCI bus will delay for a transaction between the given PCI slot and the ISA bus. This number is dependent on the PCI master device in use and varies from 0 to 255. However, when ISA cards are present, PCI latency cannot be increased beyond 64 cycles.

AMI defaults to 66, but 40 clocks is a good place to start at 33MHz (Phoenix). The shorter the value, the more rapid access to the bus a device gets, with better response times, but the lower becomes the effective bandwidth and hence data throughput. Normally, leave this alone, but you could set it to a lower value if you have latency sensitive cards (e.g. audio cards and/or network cards with small buffers). Increase slightly if I/O sensitive applications are being run.

Boot Magazine suggested a performance increase of 15% on doubling latency from 64 to 128 cycles. Going from 32 to 64 gives a noticeable increase, too.

PCI Latency Timer

As above. The default of 32 PCI Clock (80 sometimes) mostly gives maximum performance.

Reset Configuration Data

The Extended System Configuration Data (ESCD) is a small amount of memory that contains information about non-PnP devices and how the system was set up last time it was booted. Normally, *Disabled* retains this stuff in the CMOS. Selecting *Yes* causes the system to clear itself and automatically configure all PnP devices at boot up. You would do this to reset ESCD when you exit setup after installing a new card and you cannot boot, or you want to let the system sort out resources if a new device is not seen.

Slot PIRQ

A PIRQ (PCI IRQ) is signalled to and handled by the PCI bus. Not the same as a normal IRQ.

Using IRQ

Affected by the Trigger method. IRQs can be Level or Edge triggered (see Expansion Cards). Most PCI cards use the former, and ISA the latter. If you select Edge for the slot concerned, you may also need to set jumpers on the motherboard.

Slot 1/5, 2, 3, 4 IRQ

Here, the fields automatically assign the IRQ for each PCI slot. The Auto setting uses auto-routing to determine IRQ assignments.

Host-to-PCI Bridge Retry

Enabled, the peripherals controller (PIIX4) retries, without initiating a delayed transaction, CPU-initiated non-LOCK# PCI cycles. No delayed transactions to the controller may be currently pending and Passive Release must be active, with Delayed Transaction enabled.

PCI Delayed Transaction

Uses a 32-bit posted write buffer to cope with the very much slower ISA bus, and allow the PCI bus to get on with something else while it's waiting for an ISA device to finish what it's doing - a PCI device to write to the buffer (in the chipset) when the system bus is being used by an ISA device, so the contents can be written to it later. Enabled supports PCI 2.1, and is best for performance, but may need disabling if using a single-CPU OS with dual processors. If you haven't got an ISA bus, you shouldn't need it, but sometimes items (such as embedded IDE connections) are on the ISA bus anyway - you just don't see the slots.

Delayed Transaction/PCI 2.1 support/passive release

As above, this allows a PCI device to write to a 32 bit buffer in the chipset when the system bus is being used by an ISA device, so the contents can be written to it later using passive release. When disabled, the PCI device has to wait. This is only relevant with ISA cards present. PCI 2.2 concerns hardware only - it does not impact the BIOS.

PCI Dynamic Bursting

When enabled, every write transaction goes to the write buffer, and sent when there are enough to justify a single burst.

DMA Channel 0/1/3/5/6/7

Whether the AMI BIOS should remove a DMA from those available to BIOS-configurable devices (what is in the pool is in ESCD NVRAM). Manually, assign it to ISA/EISA.

IRQ 3/5/7/9/10/11/14/15

As above, but for IRQs. Onboard stuff is configured by the BIOS anyway and configured as PCI/PnP. If all are set to ISA/EISA, and 14/15 go to the onboard IDE, 9 is still available.

PCI Slot x INTx

Assigns IRQs to PCI INT#s in slot x (or whatever). See Slot X using INT# (below).

PCI Slot 1 IRQ, PCI Slot 2 IRQ

Assigns IRQs to PCI Slots.

Slot x INT# Map To

See Slot X using INT# (below).

Slot X Using INT#

Selects an INT# channel for a PCI Slot, and there are four (A, B, C & D) for each one, that is, each PCI bus slot supports interrupts A, B, C and D. #A is allocated automatically, and you would only use #B, #C, etc if the card needs to use more than one (PCI) interrupt service. For example, select up to #D if your card needs four; a typical situation would be an IDE card with two channels, each requiring an IRQ. However, using Auto is simplest. Most graphics cards don't need this.

Edge/Level Select

Programs PCI interrupts to single-edge or logic level. Select Edge for PCI IDE. IRQ 14 is used for Primary and 15 for Secondary. Some motherboards provide a particular slot for edge-triggered cards. As the interrupts are level sensitive and can be shared, two or more PCI interrupts can be steered into the same IRQ signal.

PCI Device, Slot 1/2/3

Enables I/O and memory cycle decoding.

Enable Device

Enable PCI device as a slave.

Xth Available IRQ

Selects (or maps) an IRQ for one of the available PCI INT#s above. There are ten selections (3, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15). 1st available IRQ (below) means the BIOS will assign this IRQ to the first PCI slots (order is 1, 2, 3, 4). NA means it is assigned to the ISA bus and is therefore not available to a PCI slot.

1st-6th Available IRQ

See *Xth Available IRQ*.

IRQ Assigned To

Specifies the type of device using the interrupt; Legacy ISA, which needs a specific interrupt, or PCI/ISA PnP, which complies with the Plug and Play standard, and will be set up automatically.

PCI IRQ Activated by

The method by which the PCI bus recognises an IRQ request; Level or Edge (see Expansion Cards). Use the default unless advised otherwise, or if you have a PCI device which only recognizes one of them. Affects reliability, not performance.

PIRQ_0 Use IRQ No. ~ PIRQ_3 Use IRQ No.

Here you can set the IRQ for a particular device on the AGP or PCI bus, particularly useful when transferring equipment from one computer to another; and you don't want to go through redetection. The AGP and PCI slot #1 share the same IRQs, as do PCI slot #4 and #5. USB uses PIRQ_4.

	#1	#2	#3	#4
PIRQ_0	INT A	INT D	INT C	INT B
PIRQ_1	INT B	INT A	INT D	INT C
PIRQ_2	INT C	INT B	INT A	INT D
PIRQ_3	INT D	INT C	INT B	INT A

Check the device's slot, then the table above to determine its primary PIRQ. In slot 2, for example, it is PIRQ_1. The assign the IRQ for that slot by assigning it to the appropriate PIRQ in this section.

IDE Speed

Fast or *Slow*; but it is not known whether this concerns PIO modes or not. Pheonix says that most modern drives will run in Fast mode.

DMA Assigned To

Similar to IRQ Assigned To, for DMA channels.

DMA n Assigned To

As above - you can assign DMA channels as Legacy or PCI/ISA PnP.

1st/2nd Fast DMA Channel

Select up to 2 DMA channels for Type F DMA, if supported by the peripheral using them.

IDE Prefetch Buffers

This is disk data caching at the IDE controller level, and works with PIO and DMA, on PCI, ISA or VLB computers. Using them with early versions of the Saturn chipset may result in data corruption when two devices are accessed at the same time. There may also be problems with Partition Magic. See also……

PCI IDE Prefetch Buffers

Disables prefetch buffers in the PCI IDE controller. You may need this with an operating system (like NT) that doesn't use the BIOS to access the hard disk and doesn't disable interrupts when completing a programmed I/O operation.

Disabling also prevents errors with faulty PCI-IDE interface chips that can corrupt data on the hard disk (with true 32-bit operating systems), like a PC-Tech RZ1000 or a CMD PCIO 640, but disabling is done automatically with later boards.

Configuration Mode

Sets the method by which information about legacy cards is conveyed to the system:

- *Use ICU.* The BIOS depends on information provided by Plug and Play software (e.g. Configuration Manager or ISA Configuration Utility). Only set this if you have the utilities. If you select this, you will see….
    - *Boot to PnP Operating System.* When enabled, the BIOS will activate only those Plug and Play cards necessary to boot the system, and hand over to an operating system that can manage Plug and Play cards for the rest. Otherwise, the remaining Plug and Play cards will not be configured, but Legacy cards will operate fine.
- *Use Setup Utility.* The BIOS depends on information provided by you as follows. Don't use the above utilities.
    - *ISA Shared Memory Size.* Specifies a range of memory addresses that will be directed to the ISA bus rather than onboard memory. Enable only for a Legacy card that requires non-ROM memory space (such as a LAN card with onboard memory buffers). Normally, the BIOS will scan C8000-DFFFFh for any BIOSes, note their location and size, then autoconfigure the PCI and PnP expansion cards, shadowing the area above E0000h (other than video) until it is full. Next, the BIOS will assign additional PCI and Plug and Play cards to the area between C8000h and DFFFFh. If a Legacy ISA card has non-BIOS memory requirements, Autoconfigure could write into an area needed by the card, so this setting tells Autoconfigure that the block of memory is reserved, and should not be shadowed. If you set this, you will get this:
        - *ISA Shared Memory Base Address.* If you select 96 KB, this can only be set to C8000h; If the 80 KB setting is selected, the address can only be set to C8000h or CC000h, and so on. With 64K, you can only choose D000 or below.
- I*RQ 3-IRQ 15*. The IRQs in use by ISA Legacy cards. If not used, set to Available. Otherwise, set Used by ISA Card, which means that nothing else can use it.

PCI IDE 2nd Channel

Use if your second IDE channel is PCI based, but disable if you're not using the 2nd channel, or you will lose IRQ 15 on the ISA slots.

PCI Slot IDE 2nd Channel

Enable if your secondary IDE controller is in a slot as opposed to being on the motherboard.

PCI timeout

When disabled, the PCI cycle is disconnected if the first data access is not completed inside 16 PCI clocks. Otherwise, it remains connected.

PCI to L2 Write Buffer

The chipset maintains its own internal buffer for PCI-external cache writes. When enabled, write cycles intended for the external (L2) cache are posted to the buffer instead so devices can complete cycles without waiting for others.

Primary IDE INT#, Secondary IDE INT#

Each PCI peripheral can activate up to four interrupts, A, B, C and D, with A being the default. The others are used when more than one interrupt is required. This assigns 2 INT channels for primary and secondary channels, if supported. This screen is not displayed if ISA is selected:

- *ISA*. Assigns no IRQs to PCI slots. Use for PCI IDE cards that connect IRQs 14 and 15 directly from an ISA slot using a table from a legacy paddleboard.

Primary & Secondary IDE INT#

See above.

Primary 32 Bit Transfers Mode

Enable/Disable 32-bit transfers for the Primary IDE interface.

Secondary 32 Bit Transfers Mode

See above.

PCI IDE IRQ Map to

Used for assigning IRQs 14 (Primary) and 15 (Secondary) to particular slots and INT#s, so is mostly for when you don't have IDE on the system board, but use a card in a slot. You can define the IRQ routing to make them work properly and configure your system to the type of IDE disk controller (an ISA device is assumed; the ISA setting does not assign IRQs). Here, you specify the PCI slot and interrupt (A, B, C or D) associated with connected hard drives (not partitions). Since each IDE controller (primary or secondary) supports two drives, you can select the PCI INT# (not IRQ) for each. Map an IRQ to each with two channels.

PCI-Slot X

If the IDE is not detected, you can manually select the slot.

PCI Bus Parking

Sort of bus mastering; a device parking on the PCI Bus has full control of it for a short time. Improves performance when that device (say a PCI NIC) is being used, but excludes others.

Primary Frame Buffer

The size of the PCI frame buffer selected here should not impinge on local memory.

IDE Burst Mode

When enabled, this reduces latency between each drive read/write cycle, but may cause instability if your IDE cannot support it, so disable if you are getting disk errors. It does not appear when the Internal PCI/IDE field is Disabled.

IDE Data Port Post Write

Speeds up processing of drive reads and writes, but may cause instability if your IDE cannot support it, so disable if you are getting disk errors.

IDE Buffer for DOS & Win

For IDE read ahead and posted write buffers, to increase throughput to and from IDE devices by buffering reads and writes. Slower devices could end up slower, though (Award).

IDE Master (Slave) PIO Mode

Changes IDE data transfer speed; Mode 0-4, or Auto. PIO means Programmed Input/Output. Rather than have the BIOS issue commands to effect transfers to or from the disk drive, PIO allows the BIOS to tell the controller what it wants, and then lets the controller and the CPU perform the complete task by themselves. Modes 1-4 are available.

Host Clock/PCI Clock

Determines the speed of the PCI bus relative to the CPU internal clock, which is assumed to have the value of 1.

HCLK PCICLK

Similar to above. Host CLK vs PCI CLK divider; AUTO, 1-1, 1-1.5.

ISA Bus Clock

See below.

ISA Clock

See below.

ISA Bus Clock Option

See below.

ISA Bus Clock Frequency

Allows you to set the speed of the ISA bus in fractions of the PCI bus speed, so if the PCI bus is operating at its theoretical maximum, 33 MHz, PCICLK/3 would yield an ISA speed of 11 Mhz. Avoid the asynchronous speed of 7.159 because of its overheads. Remember the PCI clock runs at half the speed of the front side bus. Speeding up the ISA bus only seems to affect video cards.

- *7.159 MHz* (default)
- *PCICLK/4*. A quarter speed of the PCI bus
- *PCICLK/3*. One third speed of the PCI bus

PCI Write-byte-Merge

When enabled, this allows data sent from the CPU to the PCI bus to be held in a buffer. The chipset will then write the data to the PCI bus when appropriate.

PCI-ISA BCLK Divider

PCI Bus CLK vs ISA Bus CLK divider; the difference between the PCI and the ISA bus: Assuming 33 MHz, you have:

- *AUTO*
- *PCICLK1/3*. 11 MHz
- *PCICLK1/2*. 16 and a bit
- *PCICLK1/4*. 8 ish

PCI Write Burst

When enabled, consecutive PCI write cycles become burst cycles on the PCI bus.

PCI Write Burst WS

The number of cycles allotted for a PCI master burst write.

CPU-to-PCI Read Buffer

When enabled, up to four Dwords can be read from the PCI bus without interrupting the CPU. When disabled, a write buffer is not used and the CPU read cycle will not be completed until the bus signals its readiness to receive the data. The former is best for performance.

PCI-Auto

If the IDE is detected by the BIOS in a PCI slot, then the appropriate INT# channel will be assigned to IRQ 14.

CPU-To-PCI Write Buffer

See *CPU-to-PCI Read Buffer*.

PCI-to-CPU Write Buffer

See *CPU-to-PCI Read Buffer*.

PCI Write Buffer

As for *CPU-to-PCI Read Buffer*, but you can choose 2, 4 or 8 deep (Phoenix).

PCI-To-CPU Write Posting

When enabled, writes from the PCI bus to the CPU are buffered, so the bus can continue writing while the CPU gets on with something else. Otherwise, the bus must wait until the CPU is free before starting another write cycle.

CPU-to-PCI Read-Line

When On, more time will be allocated for data setup with faster CPUs. This may only be required if you add an Intel OverDrive processor to your system.

L2 to PCI Read Buffer

There is an internal buffer for L2-to-PCI writes. When enabled, L2 write cycles to the PCI bus are posted to the buffer, so devices can complete their cycles without waiting for others.

CPU-to-PCI Read-Burst

When enabled, the PCI bus will interpret CPU read cycles as the PCI burst protocol, meaning that back-to-back sequential CPU memory read cycles addressed to the PCI will be translated into fast PCI burst memory cycles. Performance is improved, but some non-standard PCI adapters (e.g. VGA) may have problems.

Byte Merging

Sometimes called PCI Dynamic Bursting, this exists where multiple writes to non-contiguous memory addresses are merged into one PCI-to-memory operation by the host controller, letting devices sort out the ones they want, which increases bus throughput and hence performance for devices that support it-not all PCI video cards do, so disable this if you get bad graphics (this one is intended to improve video performance - it concerns frame buffer cycles). When enabled, the controller checks the CPU Byte Enable signals (8 of them) to see if data from the PCI bus can be merged. Then, 8- or 16-bit data sent from the CPU to the PCI bus is held in a buffer where it is accumulated, or merged, into 32-bit data for faster performance, and written to the PCI bus when appropriate. Since this was originally intended for video, you may get problems with other peripherals, such as network cards (particularly 3Com) or even operating systems (98 is often OK, where 2000 isn't).

PCI Pipeline and Pipelining combine PCI or CPU pipelining with byte merging. See also Byte Merge Support (next) and CPU-PCI Byte Merge.

Byte Merge Support

In this case, enabling means that CPU-PCI writes are buffered (Award). In other words, 8- or 16-bit data moving between the CPU and PCI bus is accumulated, or merged, into 32-bit chunks and held in a buffer, being written to the PCI bus when time permits. As with Byte Merging, above, since this was originally intended for video, you may get problems with other peripherals, such as network cards (particularly 3Com) or even operating systems (98 is often OK, where 2000 isn't).

CPU to PCI Byte Merge

Consecutive 8- or 16-bit writes in the same double-word address en route from the CPU to the PCI bus are held in a posted write buffer, from where they are sent as a single double-word, giving faster video performance. Byte merging is performed in the compatible VGA range only (0A0000-0BFFFFh). Enabled is best.

Word Merge

Controls the word-merge feature for frame buffer cycles. When enabled, the controller checks the eight CPU Byte Enable signals to see if data words read from the PCI bus by the CPU can be merged.

PCI to DRAM Buffer

Improves PCI to DRAM performance by allowing data to be stored if a destination is busy-buffers are needed because the PCI bus is divorced from the CPU. If enabled, two buffers, capable of holding 4 Dwords each, store data written from the PCI bus to memory. Disabled, PCI writes to DRAM are limited to a single transfer.

Latency for CPU to PCI write

The delay time before the CPU writes data to the PCI bus.

PCI Cycle Cache Hit WS

Similar to *Latency for CPU to PCI Write*.

- *Normal*. Cache refresh during normal PCI cycles.
- *Fast*. Cache refresh without PCI cycle for CAS (the CPU works less with better performance).

Use Default Latency Timer Value

Whether or not the default value for the Latency Timer will be loaded, or the succeeding value will be used. If *Yes* is selected (default), you don't need *Latency Timer Value* (below).

Latency Timer Value

The maximum number of PCI bus clocks that the master may burst, or the time the bus master will occupy the PCI bus. Longer latency time gives more of a chance. See also Latency Timer (PCI Clocks).

PCI Master Latency

If your PCI Master cards control the bus for too long, there is less time for the CPU to control it. A longer latency time gives the CPU more of a chance. Don't use zero.

Latency from ADS# status

This allows you to configure how long the CPU waits for the Address Data Status (ADS) signal; it determines the CPU to PCI Post write speed.

When set to *3T*, this is 5T for each double word. With 2T (default), it is 4T per double word. For a Qword PCI memory write, the rate is 7T (2T) or 8T (3T).

The default should be fine, but if you add a faster CPU to your system, you may need to increase it. The choices are:

- *3T*. Three CPU clocks
- *2T*. Two CPU clocks (Default)

Max burstable range

The maximum bursting length for each FRAME# asserting. In other words, the size of the data blocks transferred to the PCI bus in burst mode. May also set the size of the maximum range of contiguous memory addressed by a burst from the PCI bus, a half or one K. Keep at a half, as larger values have been rumoured to cause some data loss.

CPU Host/PCI Clock

Default uses actual CPU and PCI bus clock values.

CPU to PCI burst memory write

If enabled, back-to-back sequential CPU memory write cycles to PCI are translated to PCI burst memory write cycles. Otherwise, each single write to PCI will have an associated FRAME# sequence. Enabled is best for performance, but some non-standard PCI cards (e.g. VGA) may have problems.

CPU-To-PCI Burst Mem. WR.

As above - in English, it allows the chipset to assemble long PCI bursts from data in buffers.

CPU to PCI Bursting

Enables or disables PCI burst cycles for CPU-PCI write cycles where back-to-back sequential CPU memory writes are sent out on the PCI bus as a burst cycle, which may help improve video performance significantly.

CPU to PCI post memory write

When enabled, up to four words can be written to the buffer for queuing to the PCI when it is ready to receive. When disabled, the CPU can only write to the PCI bus directly and has to wait for it (e.g. write completion is not complete until the PCI transaction completes). *Enabling* reduces CPU idle cycles and is best for performance.

CPU to PCI Write Buffer

As above. Buffers are needed because the PCI bus is divorced from the CPU; they improve overall system performance by allowing the processor (or bus master) to do what it needs without writing data to its final destination; the data is temporarily stored in fast buffers.

CPU to PCI Buffer

Allows buffers to be used between the CPU and PCI bus for faster performance. Otherwise, the CPU must wait until the write is complete before another cycle.

PCI to ISA Write Buffer

When enabled, the system will temporarily write data to a buffer so the CPU is not interrupted. When disabled, the memory write cycle for the PCI bus will be direct to the slower ISA bus. The former is best for performance.

DMA Line Buffer

Allows DMA data to be stored in a buffer so PCI operations are not interrupted. Disabled means that the line buffer for DMA is in single transaction mode. Enabled allows it to operate in an 8-byte transaction mode for greater efficiency.

ISA Master Line Buffer

ISA master buffers are designed to isolate slower ISA I/O operations from the PCI bus for better performance. Disabled means the buffer for ISA master transaction is in single mode. Enabled means it is in 8-byte modereasing the ISA master's performance. See also *ISA Line Buffer*, below.

SIO Master Line Buffer

As above, found on Pentium Pro machines.

ISA Line Buffer

The PCI-to ISA bridge has aan 8-byte bidirectional line buffer for ISA or DMA bus master memory reads from or writes to the PCI bus. When this is enabled, an ISA or DMA bus master can prefetch two doublewords to the line buffer for a read cycle.

CPU/PCI Post Write Delay

The delay time before the CPU writes data into the PCI bus. Use the lowest possible value.

Post Write Buffer

Enables posted writing from the L1 cache, which means that, within limits, writes of altered data from cache can be held until they will not interfere with reads. When disabled, the CPU may be stalled because data required to complete the current instruction cannot be read until a write is completed.

SIO PCI Post Write Buffer

To do with buffering data between the CPU and an Orion Memory Controller.

Post Write CAS Active

Pulse width of CAS# after the PCI master writes to DRAM.

PCI master accesses shadow RAM

Enables the shadowing of a ROM on a PCI master for better performance.

Enable Master

Enables the selected device as a PCI bus master and checks whether it is capable.

AT bus clock frequency

Access speed for the AT bus in a PCI system, actually used for memory access instead of wait states. Choose whatever divisor gives you a speed of 6-8.33 MHz, for 70 ns memory, depending on the speed of the PCI bus (e.g. PCI/4 at 33 MHz).

Base I/O Address

The base of the I/O address range from which the PCI device resource requests are satisfied.

Base Memory Address

The base of the 32-bit memory address range from which PCI device resource requests are satisfied.

Parity

Allows parity checking of PCI devices.

Memory Hole

Enables a memory hole at either 512K-640K or 15M-16M to support adapters that require linear frame buffer memory space (such as early Video Blasters)- once reserved it cannot be cached. Allegedly for OS/2 only. Disable this, as most cards that require it are obsolete, but especially if your extended memory appears to be limited for any reason. However, it does have some uses - enabling it will force Windows to reallocate resources and maybe solve other problems elsewhere, particularly stuttering sound with some sound cards, like the Diamond MX100 or the Soundblaster.

This may also have something to do with Write Allocation, which uses a Write Handling Control Register (WHCR) that starts with the WAE15M (write allocate enable 15-16 Mb) to reserve write allocation for memory mapped I/O adapters that can only use addresses between 15 and 16 Mb. This memory hole can be disabled to free up additional resources.

Memory Map Hole; Memory Map Hole Start/End Address

See ISA VGA Frame Buffer Size, above. Where the hole starts depends on ISA LFB Size. Sometimes this is for information only. If you can change it, base address should be 16Mb, less buffer size. Only one memory hole is allowed with the Triton chipset - once reserved it cannot be cached. This is allegedly for OS/2 only.

Memory Hole Size

Enables a memory hole in DRAM space. CPU cycles matching an enabled hole are passed on to PCI. Options include 1 Mb, 2 Mb, 4 Mb, 8 Mb, Disabled, which are amounts below 1 Mb assigned to the AT Bus, and reserved for ISA cards - once reserved it cannot be cached. Allegedly for OS/2 only. Disable if extended memory appears to be limited for any reason.

Memory Hole Start Address

To improve performance, certain parts of system memory may be reserved for ISA cards which must be mapped into the memory space below 16 Mb for DMA reasons (check the documents). The chipset can then access any code or data directly from the ISA bus. The selections are from 1-15 with each number in Mb. This is irrelevant if the memory hole is disabled (see above). Areas reserved in this way cannot be cached. Allegedly for OS/2 only.

Memory Hole at 15M Addr.

See above.

Memory Hole at 15M-16M

See *Memory Hole Start Address*, but the area above 15 Mb (F00000 to FFFFFF) becomes unavailable to the system and allocated to the ISA bus (since ISA cards can only address 24 bits of memory, the top of the hole must be at 16mb or below, and since some operating systems, like OS/2, have problems working around the hole, it should be put as high as possible). Sometimes this is reserved for expanded PCI commands - once reserved it cannot be cached. Allegedly for OS/2 only. Disable if your extended memory appears to be limited for any reason.

Local Memory 15-16M

To increase performance, you can map slower device memory (e.g. on the ISA bus) into much faster local bus memory. The device memory is then not used, as the start point transferred to system memory. The default is enabled.

15-16M Memory Location

The area in the memory map for ISA option ROMs. Choices are *Local* (default) or *Non-local*.

Multimedia Mode

Enables or disables palette snooping (see below) for multimedia cards.

E8000 32K Accessible

The 64K E area of upper memory is used for BIOS purposes on PS/2s, 32 bit operating systems and Plug and Play. This setting allows the second 32K page to be used for other purposes when not needed, in the same way that the first 32K page of the F range is useable after boot up has finished.

P5 Piped Address

Default is *Disabled*.

PCI Arbiter Mode

Devices gain access to the PCI bus through arbitration (similar to interrupts). There are two modes, 1 (the default) and 2. The idea is to minimize the time to gain control of the bus and move data. Generally, Mode 1 should be sufficient, but try mode 2 if you get problems.

PCI Arbitration Rotate Priority

Typically, access is given to the PCI bus on a first-come-first-served basis. When priority is rotated, once a device gains control of the bus it is assigned the lowest priority and all others moved up one in the queue. When this is enabled, PCI masters arbitrate for bus ownership using rotate priority. Otherwise, fixed priority is used.

Stop CPU When Flush Assert

See *Stop CPU when PC Flush*.

Stop CPU when PC Flush

When enabled, the CPU will be stopped when the PCI bus is being flushed of data. Disabling (the default) allows the CPU to continue processing, giving greater efficiency.

Stop CPU at PCI Master

When enabled, the CPU will be stopped when the PCI bus master is operating on the bus. Disabling (the default) allows the CPU to carry on, giving greater efficiency.

Preempt PCI Master Option

Enabling allows PCI bus operations to be pre-empted by certain activities, such as DRAM refresh. Otherwise, everything takes place concurrently.

I/O Cycle Recovery

When enabled, the PCI bus will be allowed a recovery period for back-to-back I/O, which is like adding wait states, so disable (default) for best performance.

I/O Recovery Period

Sets the length of time of the recovery cycle used above. The range is from 0-1.75 microseconds in 0.25 microsecond intervals.

Action When W_Buffer Full

Sets the behaviour of the system when the write buffer is full. By default, the system will immediately retry, rather than wait for it to be emptied.

CPU Pipelined Function

This allows the system controller to signal the CPU for a new memory address, before all data transfers for the current cycle are complete, resulting in increased throughput.

Pipelined Function

See above.

Fast Back-to-Back

When enabled, the PCI bus will interpret CPU read cycles as the PCI burst protocol, meaning back-to-back sequential CPU memory read cycles will be translated into the fast PCI burst memory cycles. Also, consecutive write cycles targeted to the same slave become fast back-to-back. The default is enabled.

CPU-to-PCI Fast Back to Back

As above, on the Phoenix BIOS. Disabled is recommended unless your expansion cards support it.

PCI Fast Back to Back Wr

When enabled, the PCI bus interprets CPU read cycles as the PCI burst protocol, so back-to-back sequential CPU memory read cycles addressed to the PCI bus will be translated into fast PCI burst memory cycles.

Primary Frame Buffer

When enabled, this allows the system to use unreserved memory as a primary frame buffer. Unlike the VGA frame buffer, this would reduce overall available RAM for applications.

M1445RDYJ to CPURDYJ

Whether the PCI Ready signal is to be synchronized by the CPU clock's ready signal or bypassed.

Delay ISA/LDEVJ check in CLK2

See also LDEVJ Check Point Delay, above. For choosing when the chipset samples whether the current CPU cycle is ISA or VL Bus. Settings are in terms of Standard + CLK2 periods.

VESA Master Cycle ADSJ

Allows you to increase the length of time the VESA Master has to decode bus commands. Choices are Normal (default and fastest) and Long - increasing the delay increases stability. On the Phoenix BIOS, when the VESA Master Speed is less than or equal to 33 MHz, you can set Non-Delay ADSJ. Above that, you can use Delay ADSJ if you get a problem with VESA Master cards running too fast.

CPU Dynamic-Fast-Cycle

Gives you faster access to the ISA bus. When the CPU issues a bus cycle, the PCI bus examines the command to see if a PCI agent claims it. If not, an ISA bus cycle is initiated. The Dynamic-Fast-Access then allows for faster access to the ISA bus by decreasing the latency (or delay) between the original CPU command and the beginning of the ISA cycle.

LDEVJ Check Point Delay

The time allocated for checking bus cycle commands, which must be decoded to see whether a Local Bus Device Access Signal (LDEVJ) is being sent, or an ISA device is being addressed or, in other words, when the chipset checks if the current CPU cycle relates to the VL or ISA bus. Increasing the delay increases stability, especially for VESA, while very slightly degrading the performance of ISA. Settings are in terms of the feedback clock rate (FBCLK2) used in the cache/memory control interface.

- *1 FBCLK2*. One clock
- *2 FBCLK2*. Two clocks (Default)
- *3 FBCLK2*. Three clocks

Master IOCHRDY

Enabled, allows the system to monitor for a VESA master request to generate an I/O channel ready (IOCHRDY) signal.

CPU Memory sample point

This allows you to select the cycle check point, which is where memory decoding and cache hit/miss checking takes place. Each selection indicates the check takes place at the end of a CPU cycle, with one wait state indicating more time for checking to take place than with zero wait states. A longer check time allows for greater stability at the expense of some speed.

Memory Sample Point

Concerns when the chipset checks if the current CPU cycle is at the memory cycle. 0 wait states means at the first T2 rising edge, 1 wait state means at the second. The former is the best for performance.

PCI to CPU Write Pending

Sets the behaviour of the system when the write buffer is full. By default, the system will immediately retry, but you can set it to wait for the buffer to be emptied before retrying, which is slower.

LDEV# Check point

The VESA local device (LDEV#) check point is where the VL-bus device decodes the bus commands and error checks, within the bus cycle itself.

- *0*. Bus cycle point T1 (Default and fastest)
- *1*. During the first T2
- *2*. During second T2
- *3*. During third T2

The slower the motherboard, the lower the number you can use here. Your VL-bus card must be fast enough to produce an LDEV# signal.

Local Memory Detect Point

Selects the cycle check point, or where memory decoding and cache hit/miss checking takes place. More wait states gives greater stability.

Local memory check point

Selects between two techniques for decoding and error checking local bus writes to DRAM during a memory cycle.

- *Slow*. Extra wait state; better checking (default)
- *Fast*. No extra wait state used

Delay for SCSI/HDD (Secs)

The length of time in seconds the BIOS will wait for the SCSI hard disk to be ready for operation. If the hard drive is not ready, the PCI SCSI BIOS might not detect the hard drive correctly. The range is from 0-60 seconds.

FRAMEJ generation

When the PCI-VL bus bridge is acting as a PCI Master and receiving data from the CPU, this enables a fast CPU-to-PCI buffer that allows the CPU to complete a write, before the data is delivered, reducing the CPU cycles involved and speeding overall processing. The chipset will generate two types of FRAME# signal:

- *Normal*. Buffering not employed (Default for compatibility)
- *Fast*. Buffer used for CPU-to-PCI writes

Busmaster IDE on PCI

Reduces CPU and PCI overhead. As the CPU-PCI bridge generates several wait states per bus command, the busmaster gives greater bandwidth by only reading 1 memory cycle (PIO=2).

VGA Type

The BIOS uses this information to determine which bus to use when the video BIOS is being shadowed. Choices are *Standard* (default), *PCI*, *ISA/VESA*.

PCI Mstr Timing Mode

This system supports two timing modes, 0 (default) and 1.

PCI Arbit. Rotate Priority

See *PCI Arbitration Rotate Priority*.

I/O Cycle Post-Write

When Enabled (default), data being written during an I/O cycle will be buffered for faster performance. Posted Write Buffers are used when write-thru cacheing is enabled, to reduce the time the CPU has to wait. Intel CPUs have 4 internal posted write buffers.

PCI Post-Write Fast

As in the above I/O Cycle Post-Write, enabling this will allow the system to use a fast memory buffer for writes to the PCI bus.

CPU Mstr Post-WR Buffer

When the CPU operates as a bus master for either memory access or I/O, this controls its use of a high speed posted write buffer. NA, 1, 2 and 4 (default).

Graphic Posted Write Buffer

When enabled, CPU writes to graphics memory are posted to the chipset's internal buffer so the CPU can start another write cycle before the graphics memory finishes.

PCI Mstr Post-WR Buffer

As above, for PCI devices.

CPU Mstr Post-WR Burst Mode

When the CPU operates as a bus master for either memory access or I/O, this allows it to use burst mode for posted writes to a buffer.

PCI Mstr Burst Mode

As above, for PCI devices.

CPU Mstr Fast Interface

Enables or disables a fast back-to-back interface when the CPU operates as a bus master. Enabled, consecutive reads/writes are interpreted as the CPU high-performance burst mode.

PCI Mstr Fast Interface

As above, for PCI devices.

CAS Delay in Posted-WR

Select the number of CPU cycles for CAS to remain active after a posted write. The fewer, the faster.

CPU Mstr DEVSEL# Time-out

When the CPU initiates a master cycle using an address (target) which has not been mapped to PCI/VESA or ISA space, the system will monitor the DEVSEL (device select) pin to see if any device claims the cycle. Here, you can determine how long the system will wait before timing-out. Choices are 3 PCICLK, 4 PCICLK, 5 PCICLK and 6 PCICLK (default).

PCI Mstr DEVSEL# Time-out

As above, for PCI devices.

IRQ Line

If a device requires an IRQ service into the given PCI slot, use this to inform the PCI bus which IRQ it should initiate. Choices range from IRQ 3-15.

Arbiter timer timeout (PC CLK) 2 x 32

Working on this.

Fast Back-to-Back Cycle

When enabled, the PCI bus will interpret CPU read or write cycles as PCI burst protocol, meaning that back-to-back sequential (e.g. fast) CPU memory read/write cycles addressed to the PCI will be translated into fast PCI burst memory cycles.

State Machines

The chipset uses four state machines to manage specific CPU and/or PCI operations, which can be thought of as highly optimized process centres for specific operations. Generally, each operation involves a master device and the bus it wishes to employ. The state machines are:

- CPU master to CPU bus (CC)
- CPU master to PCI bus (CP)
- PCI master to PCI bus (PP)
- PCI master to CPU bus (PC)

Each have the following settings:

- *Address 0 WS*. The time the system will delay while the transaction address is decoded. Enabled=no delay (fastest).
- *Data Write 0 WS*. The time the system will delay while data is being written to the target address. Enabled=no delay (fastest).
- *Data Read 0 WS*. The time the system will delay while data is being read from the target address. Enabled=no delay (fastest).

On Board PCI/SCSI BIOS

You would enable this if your system motherboard had a built-in SCSI controller attached to the PCI bus, and you wanted to boot from it.

PCI I/O Start Address

Allows you to make additional room for older ISA devices by defining I/O start addresses for the PCI devices, thus overriding the PCI controller.

PCI Memory Start Address

For devices with their own memory which use part of the CPU's memory address space. You can determine the starting point in memory where PCI device memory will be mapped.

VGA 128k Range Attribute

This allows the chipset to apply features like CPU-TO-PCI Byte Merge, CPU-TO-PCI Prefetch to be applied to VGA memory range A0000H-BFFFFH.

- *Enabled*. VGA receives CPU-TO-PCI functions
- *Disabled*. Retain standard VGA interface

CPU-PCI Burst Memory Write

Enabling is best for performance.

CPU-PCI Post Memory Write

Enabling is best for performance.

Posted PCI Memory Writes

When this is enabled, writes from the PCI bus to memory are posted as an intermediate step. If the CPU and PCI-To-DRAM posted write buffer is enabled, the data is interleaved with CPU write data and posted a second time before being written to memory.

CPU-To-PCI Write Posting

Posting refers to the use of buffers between the CPU and PCI bus, or maybe the PCI bus and IDE interface (depends on the manufacturer) to help match their relative speeds - they are called Posted Write Buffers. The idea is that the PCI bus can retrieve data in its own good time without holding up the CPU. In this particular case, they belong to the Orion chipset. When this setting is enabled, writes from the CPU to the PCI bus will be buffered without interfering with reads into the CPU cache. When disabled (default), the CPU is forced to wait until the write is completed before starting another write cycle. Sometimes this cannot be used with certain video cards at cerftain CPU speeds (just try and see). Not the same as PCI Posted Write Enable, which seems to buffer data between buses.

CPU To PCI Write Buffers

See *CPU-To-PCI Write Posting* (above).

OPB P6 to PCI Write Posting

As above, but found on Pentium Pro machines.

OPB PCI to P6 Write Posting

As above, but in reverse.

CPU-To-PCI IDE Posting

Enabled, IDE accesses are buffered in the CPU-PCI buffers, which is best for performance, as cycles are optimised. Otherwise, CPU-PCI IDE postings are treated as normal I/O writes.

CPU Read Multiple Prefetch

A prefetch occurs during a process such as reading from the PCI or memory, when the chipset peeks at the next instruction and begins the next read. The Orion chipset has four read lines, and a multiple prefetch means the chipset can initiate more than one during a process. Default is *Disabled* (slowest).

CPU Line Read Multiple

A line read means the CPU is reading a full cache line, which means 32 bytes (8 DWORDS) of data. Because the line is full, the system knows exactly how much data it will be reading and doesn't need to wait for an end-of-data signal, so blocks of data can be read without pausing every 4 cycles to specify a new address. When this is enabled, the system can read more than one full cache line at a time, so is best for performance. The default is Disabled.

OPB P6 Line Read

As above, but on Pentium Pro machines, possibly with the Orion Chipset.

CPU Line Read Prefetch

See also *CPU Line Read Multiple* and *CPU Read Multiple Prefetch* (above). When enabled, the system is allowed to peek at the next instruction and initiate the next read. Prefetching is used by 80x8x CPUs to read instructions from relatively slow DRAM and store them in fast CPU registers during the execution of previous ones, using unused cycles.

OPB Line Read Prefetch

As above, but found on Pentium Pros, possibly with the Orion chipset.

CPU Line Read

Enables or Disables full CPU line reads. See *CPU Line Read Multiple*, above.

CPU Read Multiple Prefetch

See above. Where a chipset has more than one read line, a multiple prefetch means it can initiate more than one prefetch during a process.

DRAM Read Prefetch Buffer

This controls memory access latency. For every memory access request, a preprogrammed number of local bus clock signals is counted down. If the number of filled posted write buffer slots is at or above a predetermined figure when the count reaches zero, the memory request priority is raised.

Read Prefetch Memory RD

When enabled, the system can prefetch the next read instruction and initiate the next process, which is best for performance

VGA Performance Mode

If enabled, the VGA memory range of A0000-B0000 will use a special set of performance features. This has little or no effect using video modes beyond those commonly used for Windows, OS/2, UNIX, etc, but this range is heavily used by games. Same as *Turbo VGA*.

Snoop Ahead

This is only applicable if the cache is enabled. When enabled, PCI bus masters can monitor the VGA palette registers for direct writes and translate them into PCI burst protocol for greater speed, to enhance the performance of multimedia video.

DMA Line Buffer Mode

Allows DMA data to be stored in a buffer so as not to interrupt the PCI bus. Standard equals single transaction mode. Enhanced means 8-byte transactions.

Master Arbitration Protocol

How the PCI bus determines which bus master device gains access to it.

Host-to-PCI Wait State

1, 0 or Auto.

PCI Prefetch

When enabled, the system controller prefetches 8 quadwords (1 cache line) of data when a PCI device reads from main memory (being prefetch, it is assumed that next cache line of data is required). This is best for performance.

PCI Parity Check

Enables/disables PCI Parity checking. The latter is default and slower due to extra overhead.

PCI Memory Burst Write

When enabled, CPU write cycles are interpreted as the PCI burst protocol (by the PCI bus), meaning that back-to-back sequential CPU memory write cycles addressed to PCI will be translated into (fast) PCI burst memory write cycles. This directly improves video performance when consecutive writes are initiated to a linear graphics frame buffer.

PCI Clock Frequency

Set the clock rate for the PCI bus, which can operate between 0-33 MHz, relative to the CPU, e.g. CPUCLK/2, or half the CPU speed.

- *CPUCLK/1.5*. CPU speed/1.5 (Default)
- *CPUCLK/3*. CPU speed/3
- *14 MHz*. 14 MHz
- *CPUCLK/2*. CPU speed/2

8 Bit I/O Recovery Time

The length of time, measured in CPU clocks, inserted as delay s between PCI originated input/output requests to the ISA bus, needed because PCI is faster, and needs to be slowed down. Clock cycles are added to a minimum delay (usually 5). Choices are from 1 to 7 or 8 CPU clocks. 1 is the default.

16 Bit I/O Recovery Time

As above, for 16 bit I/O. Choices between 1 to 4 CPU clocks.

8/16 Bit I/O Recovery Time

A combination of the above two.

IO Recovery (BCLK)

As for *I/O Recovery Time*.

I/O Recovery Time

As for *I/O Recovery Time Delay*, but concerns refreshing between cycles, so the lower the number the better. Set to Enhanced with Multiuser DOS on an Intel Express. If you get two numbers, the first is for 8-bit cycles, and the second 16-bit. In other words, this is a programmed delay which allows the PCI bus to exchange data with the slower ISA bus without data errors. Settings are in fractions of the PCI BCLK

- *2 BCLK*. Two BCLKS (default)
- *4 BCLK*. Four BCLKS
- *8 BCLK*. Eight BCLKS
- *12 BCLK*. Twelve BCLKS

PCI Mem Line Read

Enabled, PCI Memory Line Reads fetch full cache lines. Otherwise, partial reads are done.

PCI Mem Line Read Prefetch

Enabled, PCI Memory Line Read commands fetch a full cache line and a prefetch of up to 3 more (not crossing 4K boundaries). Irrelevant if PCI Mem Line Read (above) is disabled.

PCI Concurrency

Enabled (default) means that more than one PCI device can be active at a time (Award). With Intel Chipsets, it allocates memory bus cycles to a PCI controller while an ISA operation, such as bus mastered DMA, is taking place, which normally requires constant attention. This involves turning on additional read and write buffering in the chipset. The PCI bus can also obtain access cycles for small data transfers without the delays caused by renegotiating bus access for each part of the transfer, so is meant to improve performance and consistency.

In some Award BIOSes this also controls a Determinancy Latency Bit that stops some CDROMs from being detected or used by Win 95 pre-OSR2. If it occurs, disable.

Concurrent PCI/Host

Allows other PCI devices to work concurrently with the host PCI IDE channel. If disabled, the CPU bus will be occupied during the entire operating period.

Peer Concurrency

Whether or not the CPU can run DRAM/L2 cycles when non-PHLD PCI master devices are targeting the peer device. That is, whether the CPU can use cache or system memory when something else is going on, or talk to the busmaster controller and the card at the same time. This speeds things by allowing several PCI devices to operate at the same time, or as near to it as possible. Enabled is best for performance, but some cards might not like it.

PCI Bursting

Consecutive writes from the CPU are regarded as a PCI Burst cycle, so this allows multiple data bytes to cross the PCI bus in one go. When enabled (default), one address cycle is combined with several data cycles before being sent across the PCI bus, and the receiving agent increments the addresses itself (when disabled, data moves across the PCI bus in a single cycle/data cycle pair). All other users of the PCI bus and destination devices, such as memory, are locked out during the transfer. If a write transaction is a burst, the information goes into the write buffer and burst transfers are later performed on the PCI bus. If not, PCI write occurs immediately, after a write buffer flush. (VP2). You may need to change this for slower PCI Video cards.

PCI (IDE) Bursting

As above, but this enables burst mode access to video memory over the PCI bus. The CPU provides the first address, and consecutive data is transferred at one word per clock, if the device agrees.

PCI Dynamic Bursting

Otherwise known as *Byte Merge*, combines several writes into one 32-bit block of data (i.e. 4 words) with a special packaging protocol which is then transferred with a single command. The bytes concerned must be coherent data, that is, possess a high locality.

PCI Burst Write Combine

This is meant to speed up video processing by up to about 15%, as many writes to video memory are with individual pixels, which don't ordinarily fill up a 32-byte cache line, for which the architecture is optimised - when enabled, internal processor buffers combine smaller or partial writes into burstable writes for a specific memory area, so only one transfer is used. As Pentium Pro, Celeron, Pentium II and III CPUs have a 32-byte buffer, in 8-bit color mode, 32 write operations can be sent at once. The chipset may also assemble large PCI bursts from data stored in burst buffers if the bus is not available. Before SP6, NT did not turn this on for the Athlon.

Burst Write Combine

See above.

PCI Preempt Timer

Sets the time (in LCLK ticks) before one PCI master preempts another when a service request is pending.

- *Disabled*. No preemption (default).
- *260 LCLKs*. Preempt after 260 LCLKs
- *132 LCLKs*. Preempt after 132 LCLKs
- *68 LCLKs*. Preempt after 68 LCLKs
- *36 LCLKs*. Preempt after 36 LCLKs
- *20 LCLKs*. Preempt after 20 LCLKs
- *12 LCLKs*. Preempt after 12 LCLKs
- *5 LCLKs*. Preempt after 5 LCLKs

Keyboard Controller Clock

Sets the speed of the keyboard controller (PCICLKI = PCI bus speed).

- *7.16 MHz*. Default
- *PCICLKI/2*. 1/2 PCICLKI
- *PCICLKI/3*. 1/3 PCICLKI
- *PCICLKI/4*. 1/4 PCICLKI

Burst Copy-Back Option

If a cache miss occurs with this enabled, the chipset will initiate a second, burst cache line fill from main memory to the cache, to maintain the status of the cache.

PCI Streaming

Data is typically moved to and from memory and between devices in chunks of limited size, because the CPU is involved. On the PCI bus, however, data can be streamed, that is, much larger chunks can be moved without the CPU being bothered. Enable for best performance.

PCI-To-DRAM Pipeline

For DRAM optimisation. If enabled, full PCI-DRAM write pipelining is used, where buffers in the chipset store data written from the PCI bus to memory. Otherwise, PCI writes to DRAM are limited to one transfer per write cycle.

IBC DEVSEL# Decoding

Sets the decoding used by the ISA Bridge Controller (IBC) to determine which device to select. The longer the decoding cycle, the better chance it has to correctly decode commands. Choices are *Fast*, *Medium* and *Slow* (default). Fast is less stable and may trash a hard disk.

CPU Pipeline Function

Allows the system controller to signal the CPU for a new memory address, even before data transfers for the current cycle are complete, for increased throughput. *Enabled* means pipelining is active.

PCI Dynamic Decoding

When enabled, this setting allows the system to remember the PCI command which has just been requested. If subsequent commands fall within the same address space, the cycle will be automatically interpreted as a PCI command.

CPU to PCI POST/BURST

Data from the CPU to the PCI bus can be posted (i.e. buffered by the controller) and/or burst. This sets the methods:

- *POST/CON.BURST*. Posting and bursting supported (default)
- *POST/Agg BURST*. Posting and aggressive bursting
- *NONE/NONE*. Neither supported
- *POST/NONE*. Posting but not bursting supported

PCI Pre-Snoop

Pre-snooping is a technique by which a PCI master can continue to burst to the local memory until a 4K page boundary is reached rather than just a line boundary. Enabled is best for performance. If disabled, one line (four words) is transferred in a burst operation and another address must be passed at the start of the next burst.

Secondary CTRL Drives Present

Allows you to manually set the number of drives on your secondary channel.

PCI Read Burst WS

The number of cycles alloted for a PCI master burst read.

PCI Master Cycle

Where the chipset checks for the PCI Master Cycle in local memory. Fast means in the address phase, which is earlier, and Slow the first data phase.

Master Retry Timer

Sets how long the CPU master will attempt a PCI cycle before the cycle is unmasked (terminated). The choices are in PCICLKs. Values are 10 (default), 18, 34 or 66 PCICLKs.

CPU/PCI Write Phase

Determines the turnaround (or number of clock signals) between the address and data phases of the CPU master to PCI slave writes. Choices are *1 LCLK* (default) or *0 LCLK*.

PCI CLK

Whether the PCI clock is tightly synchronized with the CPU clock, or is asynchronous. If your CPU, motherboard and PCI bus are running at multiple speeds of each other, e.g. Pentium 120, 60 MHz m/b and 30 MHz PCI bus, choose synchronise.

IRQ 15 Routing Selection

MISA=Multiplexed ISA for asynchronously interrupting the CPU. IRQ 15 is usually used for Secondary IDE channels or CD-ROMs.

CPU cycle cache hit sam point

Working on this.

PCI cycle cache hit sam point

Working on this.

Plug and Play OS

Here, on the face of it, you specify whether you have one or not, but things aren't as easy as that! Firstly, this only affects ISA PnP cards-PCI cards are initialised anyway. No means the BIOS will allocate interrupt settings. Yes means that they may be reassigned by the operating system, or that the BIOS will only initialise PnP PCI boot devices and leave the rest to Windows, or whatever. However, when IRQ sharing this way, software emulation only works if there are no conflicts in the BIOS anyway, which is a good reason for turning it off (and Windows has a tendency to make everything use IRQ 11, or whatever, and leave others empty). However, Asus have their own hardware IRQ distributor which requires this to be turned on for successful operation. Windows 2000 should have this disabled, because of ACPI, but it will work if you enable it, and you disable APM (on an ACPI-capable motherboard, disable Power Management). Linux should also have this disabled, as it uses isapnptools to do its own thing-if you run it after the BIOS has configured your cards, it will fail, leaving any the BIOS cannot initialise (like AWE 32/64, SB16, etc) unusable.

In short, *Enabled* tells the BIOS to only worry about the basics (video, etc). When *Disabled*, it will initialize all the PnP hardware it can find. You should therefore leave this at *No*, especially if you change operating systems a lot, unless you are happy to let the operating system do all the work. Disabling this may mean a more stable system if you get spontaneous crashes.

PnP OS

See above.

PCI Passive Release

This concerns the PIIX4 (PCI-ISA bridge), and the latency of ISA bus masters. When enabled, ISA cards cannot stop the PCI bus using DMA mode. Put more officially, CPU-PCI bus accesses are allowed during passive release, otherwise the arbiter only accepts another PCI master access to local DRAM. If you have a problem with an ISA card, set it to the opposite of the current setting.

Delayed Transaction

PCI 2.1 is tight on target and master latency, and PCI cycles to and from ISA generally take longer to perform because the ISA bus is slower. When enabled, the chipset provides a programmable delayed completion mechanism (i.e. 32-bit posted write buffers), where the PCI bus is freed during CPU access to 8-bit ISA cards, which otherwise would consume about 50-60 PCI clocks. Disable for bus mastering PCI cards that cannot use the PCI bus, or some ISA cards that are not PCI 2.1 compliant (PCI 2.2 concerns hardware, not the BIOS).

PCI 2.1 Compliance

See Delayed Transaction (above) - this is another name for it. You can enable or disable the PIIX3 register Delayed Transaction and Passive Release. When enabled, the PIIX3 controls USB operation to ensure the system complies with PCI 2.1 (PCI 2.2 concerns hardware- it does not impact the BIOS).

Chipset Global Features

Applies bus masteringto all PCI slots, assuming all cards are compatible.

FDD IRQ Can Be Free

Allows it to be used by the PnP system.

Multi Transaction Timer

Allows PCI cards to hold their request lines high and receive PCI bursts without rearbitration delays and without locking others out of the bus (the Multi Transaction Timer controls the minimum burst size). May improve data transfer for devices needing uninterrupted high data transfer rates (anything to do with video), but may also cause problems.

Multi-function INTB#

Enables or disables multi-function PCI cards using INTA# and INTB#.

Shared VGA Memory Speed

The memory speed of DRAM allocated for video memory.

PCI Master 0 WS Write

Increases write cycle speed when enabled - that is, writes to PCI bus have zero wait states.

PCI Master 1 WS Write

Writes to PCI bus are executed with an extra wait state. Normally disabled.

PCI Master 1 WS Read

Reads to the PCI bus are executed with an extra wait state. Normally disabled

PCI Delay Transaction

When enabled, the CPU can access the PCI bus during Passive Release (when Passive Release is enabled, the bus can operate by itself when the ISA bus is accessed). If disabled, only PCI bus mastering devices can access the PCI bus. AMI uses a 32-bit posted write buffer.

PCI Master Read Prefetch

Enabled, allows the system to prefetch the next read and initiate the next process, so enabled is best for performance.

PCI#2 Access #1 Retry

Enables PCI #2 Access in #1 attempts. When the CPU to PCI Write Buffer is enabled (normal), writes to the PCI bus are written to it instead, which frees the CPU. The writes take place at the next bus cycle. If they fail, and this is enabled, attempts will continually be made until success is achieved, with an obvious tax on performance if you have a slow PCI device. Disabling forces the buffer to flush its contents and register the transaction as failed, making the CPU do the write again to the buffer.

Master Priority Rotation

Controls CPU access to the PCI bus, similar to delay states. It balances the needs of the CPU and devices on the bus. 1 PCI grants access immediately after the current PCI bus master transaction, ignoring other bus masters in the queue, but this means poorer performance for bus devices. With 2 PCI, the CPU has to wait till after the current and next PCI transaction, so 3 PCI (the maximum) allows access after the current transaction and the following two.

PCI Arbitration Mode

Determines the order in which PCI Bus Masters get control of the PCI Bus, i.e. First Come, First Served (FCFS), or Rotated, which invokes scheduling of priorities of attached devices (when priority is rotated, once a device gains control of the bus it is assigned the lowest priority and every other device is moved up one in the priority queue). Affects reliability rather than performance.

PCI Bus Clock

Determines whether the PCI bus clock is tied to the system clock or is independent, which may introduce delays because an asynchronous bus may sometimes force the CPU to wait when the PCI cycle starts late in a CPU cycle. Performance may, however, be a bit more consistent with Synchronous.

PCI IDE Bursting

Enables burst mode memory access to video memory via the PCI bus.

PCICLK-to-ISA SYSCLK Divisor

Defines the ISA (AT) Clock speed as a fraction of PCI bus speed..

Used By Legacy Device

Reserves IRQs (0-15) from the pool of those available to PnP devices. Including them means they can be assigned. Non-PnP (Legacy) devices should be excluded.

Use MultiProcessor Specification

For motherboards with lots of PCI slots, Specification 1.4 allows extended bus definition. It is needed to allow a secondary PCI bus to work without a bridge.

Write Allocate

Write allocation is a feature of the K6 or 6x86 which is like prefetching data from main memory, based on its locality. That is, when data is fetched, everything around it is grabbed as well and only needs to be allocated properly - the larger the buffers, the more the chances that the data you want is actually there. The Write Allocate enablement bits are in different Model Specific Registers (MSRs) on the two CPUs mentioned above, so the BIOS cannot set the bits if the wrong CPU is selected. You can also do this with shareware programs (enwa.exe or msr.zip for NT). The *Write Handling Control Register* (WHCR) starts with the WAE15M (write allocate enable 15-16 Mb) which reserves write allocation for memory mapped I/O adapters that can only use addresses between 15 and 16 Mb. This memory hole can be disabled to free up additional resources.

Extended CPU-PIIX4 PHLDA#

Adds one clock signal to the time the PHLDA# is active during the address phase at the beginning of a PCI read/write transaction, and following the address phase of a CPU LOCK cycle. You also need to enable Passive Release and Delayed Transaction.

Used MEM length

The area used by peripherals requiring high memory (could be upper memory). Choices are between 8, 16, 32 or 64K. Does not appear if no base address (below) is specified.

Used Mem Base Addr

The base address for memory specified above.

Close Empty DIMM/PCI Clk

Stops the clock in an empty DIMM or PCI slot to reduce EMI.

FWH (Firmware Hub) Protection

The BIOS is kept inside the hub so that viruses such as CIH cannot get to it. See also Flash Write Protect. This is set in conjunction with a jumper on the motherboard.

Ultra DMA 66 IDE Controller

Enable or disable the onboard UltraDMA 66 controller.

USB Function For DOS

Enables Passive Release on the PCI bus when not using Windows.

Flash Write Protect

This prevents interference with the BIOS by viruses such as CIH. You can still update DMI with the right setting here. Disable if you want to upgrade the BIOS.

FPU OPCODE Compatible Mode

If enabled, the Pentium IV's FPU unit runs in compatibility mode, or through software emulation, which is the slowest option. Disabled is best for performance.

CPU fast String

Specifies DMA-like capabilities for the Pentium IV, which can work directly on a string in a cache-line using cache-line mode. Enabled is best for performance.

PCI Master Read Caching

From the ASUS A7V - you are supposed to enable this for the Thunderbird and disable it for the Duron, due to the latter's L2 cache being half the size of the former's, resulting in possible cache overflow and lower performance, but it's really down to the controller's ability to read data ahead of the PCI Master from memory into a buffer, especially during bursts. If the memory is cacheable, a snoop must be forced to the CPU.

SDRAM Closing Policy

The i815 chipset can keep up to four memory pages open in separate banks. This is similar to bank interleaving with VIA chipsets, except that, if a page hit occurs, you can close all open pages or only the page(s) in which a miss occurred. If you close one page, all the others are open, so you can do a bank-to-bank access with an additional latency of only 1 cycle. If a page miss occurs, however, the data locality is interrupted in such a way that the next access results in a page miss as well, which means it has to be closed (precharged) and opened again, giving the full amount of penalty cycles. If you close all banks, a consecutive access will be the same as an access from idle, since banks will not have to be closed before an activate command can start a new sequence.

PCI/DIMM Clk Auto Detect

Normally, the clock signal from PCI and memory is evenly distributed between all the slots. Unfortunately, its driving strength can be weakened because of each slot's own traces, which have their own inductance, impedience and capacitance values. Stray signals also cause increased EMF and EMI. This setting allows the clock signal to be routed only to where a device is detected, which makes it stronger, with less EMI.

High Priority PCI Mode

Gives a higher priority to PCI slot No 1. Can boost performance for Firewire cards, etc.

LDT Setting

This controls the speed of the HyperTransport bus, which, on Athlon 64 motherboards, at least, is the link between the processor and the Northbridge. Its base frequency is 400 MHz (actually, 200 in both directions at the same time), which is raised by factors of 1, 2, 3, or 4, so you can get speeds up to 1600 MHz (though not all boards, particularly many nForce3-based ones, will cope with more than 1200 MHz). LDT stands for *Lightning Data Transport*, the original name for HyperTransport.

15 PERIPHERAL SETUP

This section mainly concerns all-in-one motherboards; on board equipment is often not as good as other products, so you may want to disable some of them - IDE, for example, has been known to operate through the ISA interface rather than the PCI.

Programming Option

Auto-BIOS detects and sets up cards and ports automatically. On board I/O is dealt with last.

Configuration Mode

Whether onboard peripherals are configured automatically or manually. Use *Auto* if you think PnP will work, but Manual is usually best, in which case use Auto first, then Manual.

TxD, RxD Active

The setting of the TxD and RxD signals.

Use IR Pins

Concerns the setting of the TxD and RxD signals.

On Chip Serial ATA Mode

This item determines the mode for on-chip Serial ATA, such as IDE or RAID (this is only available when On Chip Serial ATA is set as EnhancedMode).

OnChip Serial ATA

Determines the function for on chip Serial ATA. *Disabled* turns it off, *Auto* allows it to be arranged by BIOS automatically, *Combined Mode* combines Parallel and Serial ATA (up to 4 drives) and *Enhanced Mode* enables both (up to 6). *SATA Only* means it is in legacy mode.

SATA RAID ROM

This allows you to use the boot ROM of on-chip Serial ATA RAID to boot up the system.

On Chip IDE Buffer (DOS/Windows)

See *IDE Buffer for DOS & Win*.

On Chip IDE Mode

Selects PIO Mode for your drive.

IDE 0 Master/Slave Mode, IDE 1 Master/Slave Mode

Sets independent timing for IDE devices on both channels, to stop the slowest interfering with the faster.

On Chip Local Bus IDE

Disable if you add another.

On-Chip Primary PCI IDE

Enables or disables onboard PCI IDE.

On-Chip Secondary PCI IDE

Enables or disables onboard PCI IDE. If you install an extra interface as second channel, see also below.

On-Chip Video Window Size

Selects the size of window used for the graphics display cache, 32 or 64 Mb.

On Chip PCI Device

Click **Enter** to enter the submenu:

ON CHIP USB CONTROLLER

Enables or disables the USB controller.

USB 2.0 CONTROLLER

Enables or disables the USB 2.0 controller.

USB KEYBOARD SUPPORT VIA

Select [BIOS] for using the USB keyboard in DOS environment, or [OS] in OS environment.

USB MOUSE SUPPORT VIA

Allows you to select [BIOS] for using the USB mouse in DOS environment, or [OS] in OS environment.

ONCHIP AUDIO CONTROLLER

Enables or disables the audio controller.

ONCHIP LAN CONTROLLER

Enables or disables the LAN controller.

LAN BOOT ROM

Allows you to use the boot ROM (instead of a disk drive) to boot up the system and access the local area network directly.

2nd Channel IDE

If you install an extra IDE interface as the second channel, disable this to avoid a conflict.

IDE Second Channel Control

See above.

PCI IDE Card Present

Use if secondary IDE card installed.

Onboard Floppy Drive

Disable if you want to use a floppy controller on an expansion card.

Onboard FDC Controller

See above.

Onboard FDC Swap A: B:

For swapping drive assignments through the onboard floppy controller.

Onboard IDE

Enable/Disable. This often goes through the ISA interface.

FDC Function

Enables or disables the floppy controller. Auto lets the system do it.

Onboard Serial Port 1

(or 2). Sets IRQs and I/O addresses.

Onboard UART 1/2

See above.

Onboard UART 1/2 Mode

Modes selected apply to relevant serial port.

Onboard ATA Device First

Controls the priority of the detect sequence of the onboard RAID and PCI slot option. Yes commands the onboard RAID controller to initiate first.

UART Mode Select

Defines what COM 2 does, whether normal or IR.

Internal PCI/IDE

Enable or disable either channel on your motherboard.

UART 1/2 Duplex Mode

Appears in infrared port mode. Select the value required.

UR2 Mode

See above.

Serial Port 2 Mode

See above.

First Serial Port

(or 2). Sets IRQs and I/O addresses.

Parallel Port Address

What I/O address is used.

IRQ Active State

Whether parallel/serial IRQs are active high or low.

SuperIO Device

Click **Enter** for the submenu:

ONBOARD FDC CONTROLLER

Enables or disables the onboard FDController.

Onboard Parallel Port

Enable/Disable - match the logical LPT port address and interrupt.

Onboard IDE Controller

Select the interface you want, or don't want.

Onboard PCI SCSI Chip

Enable/Disable.

Onboard Audio Chip

Enable or disable.

Onboard PCI Device

Click **Enter** for the submenu:

IEEE 1394 CONTROLLER

Enable or disable.

SERIAL ATA CONTROLLER

Enables or disables the Serial ATA controller of Silican Image PCI chip.

ECP DMA Select

Available only if you select ECP or ECP+EPP above. Channels 1 or 3 (default) are available.

ECP Mode DMA

As above.

Parallel Port EPP Type

Sets one of two versions of EPP, 1.7 and 1.9. Try the latter first, but be prepared to use the former if you get problems. See also LPT Extended Mode.

Port Mode

Set the parallel port to EPP, Normal, Bi-Dir or EPP.

Parallel Port Mode

Sets one of two versions of EPP, 1.7 and 1.9. 1.7 was the original, developed by Intel, Xircom and Zenith, but 1.9 was added later by IEEE 1284, which resolved a problem caused by long cables when 1.7 does not check for some device acknowledgements and relies on a 125 ns timeout instead. A 1.7 device would normally be backwards compatible, but some cannot cope with a reduced timeout, hence the choice.Try 1.9 first, but be prepared to use the former if you get problems. See also LPT Extended Mode.

Enable/Disable.UART 2 Mode.

The operating mode for the second serial port, as this is the one most needing to be flexible:

- *Normal*. RS232
- *Standard*. RS232
- *IrDA 1.0*. IR port to 1.0 specs
- *IrDA SIR*. IrDA-compliant serial IR port
- *IrDA MIR*. 1 Mb/sec IR port
- *IrDA FIR*. Fast IR standard
- *FIR*. Fast IT standard
- *MIR 0.57*. M 0.57 Mb/sec IR port
- *MIR 1.15*. M 1.15 Mb/sec IR port
- *Sharp IR*. 4 Mb/sec data transmission
- *HPSIR*. IrDA-compliant serial IR port, up to 115K bps
- *AskIR*. Amplitude Shift Keyed IR port, up to 19.2K bps

Parallel Port

Sets the Base I/O address of the second parallel port. Auto lets the AMIBIOS do it.

IRQ

When the above is set to Auto, this will show Auto as well. Otherwise, you can choose it here.

LPT Extended Mode

Parallel ports come in the following variations:

- Standard Parallel Port (SPP)
- Enhanced Parallel Port (EPP)
- Extended Capability Port (ECP)
- EPP + ECP

The SPP is unidirectional, as it was designed for printers, and only 5 of its wires are for input; bidirectional communications actually use printer status signals. SPP does not need interrupts, so they can be used elsewhere. EPP and ECP have more wires for input, so are bidirectional and do need interrupts. ECP defines register formats, allows RLL compression, is fast (over 1 Mb/sec) and buffered, and allows better communication between the device concerned and the PC - it's good for block transfers, and you can expect it to use DMA 3.

EPP allows devices to be connected in a chain, so you could rig up a small network of two machines connected through their parallel ports. In addition, the port switches on a request strobe when it wants to read or write. The attached device reads the data, and acknowledges with another strobe. The port then negates it when the operation is done. The essential difference between versions 1.7 and 1.9 is the last bit - 1.7 ports don't check that the device has negated acknowledge, but waits for 125 ns, then assumes it as fact (which is why cables should not be too long). 1.9 ports specifically wait for it, so you can use long cables again.

Printers and scanners work best with ECP. Try EPP with Zip drives. ECP was developed by HP and Microsoft in 1997, in advance of the IEEE specification for advanced parallel ports, so EPP is more compatible. Both have around the same performance, but ECP can run faster than the maximum data transfer rate. ECP+EPP (default) allows normal speed in two-way mode. SPP may be helpful if you have printing problems with Windows '95.

EPP Version

See above.

EPP Mode Select

See above.

DMA Channel

Appears only when Port Mode (above) is set to ECP mode, as it requires a DMA. When set to Auto, this will show the same.

WAVE2 DMA Select

The DMA Channel for the WAVE 2 device.

WAVE2 IRQ Select

The interrupt for the WAVE 2 device.

Floppy DMA Burst Mode

Enabled is best for performance.

Serial Port 1 MIDI

Configures serial port 1 as a MIDI interface. Or not. MIDI is a specification from the music industry for controlling devices that emit music, which is probably why it stands for Musical Instrument Digital Interface. And works.

USB Controller

Enabled or not. *Disabled* will free up an IRQ, but Windows 98 or Windows 95 B/C will require it, otherwise you will get instability (if you are using a USB device). You can share the IRQ, though.

USB Function
As above.

Assign IRQ For USB
As above.

USB Keyboard Support
Enables or disables support for a USB keyboard.

USB Keyboard Support Via
Whether the USB keyboard is supported via the operating system or the BIOS. Set the latter if you use DOS and don't have a driver.

USB Latency Time (PCI CLK)
The minimum time, in PCI clock cycles, the USB controller can own the PCI bus.

USB Legacy Support
Set to *All Device* if you need to use USB devices without drivers or with systems that don't support it, such as DOS. Set to *No Mice* if you want to use devices other than the USB mouse.

Infrared Duplex
Whether communications are Disabled, Half-Duplex or Full-Duplex or Simplex or Duplex-Simplex means one-way only in either direction, Duplex means both ways at the same time.

Infra Red Duplex Type
See above.

IR Function Duplex
See above.

IR Duplex Mode
See above.

Duplex Select
See above.

IR Pin Select
Use *IRRX/IRTX* with a module on motherboard pins. Use *SIB/SOUTB* through COM 2.

UART2 Use Infrared
Allocates the onboard infrared feature to the second serial UART. The default is Disabled, which allows it to be used for COM2.

Onboard VGA Memory Size (iMb)
For allocating total VGA memory from shared memory. Choices are 1, 2 or 4 Mb.

Onboard VGA Memory Clock

Onboard Video speed. Normal is 50 MHz, Fast is 60 and Fastest is 66. Decrease this to match the monitor's frequency rate if your screen is unreadable.

IRRX Mode Select

You will only see this if IrDA Mode 1.1 (Fast IR) is selected for UART2 mode. It depends on the type of transceiver module - one has a mode pin (IRMODE) and the other has a second receive data channel (IRRX3) - check your documentation.

NCR SCSI BIOS

Enables or disables the onboard NCR SCSI BIOS.

Write Buffer Level

Select between 4 or 8 level write buffers for the PCI bridge.

Offboard PCI IDE Card

Whether an offboard PCI IDE controller is used, but you must also specify the slot, because it will not have a built-in configuration EPROM as required by PCI specification. The onboard IDE controller on the motherboard is automatically disabled. The settings are *Disabled*, *Auto*, *Slot1*, *Slot2*, *Slot3*, or *Slot4*.

Audio DMA Select

Selects a DMA Channel for the audio.

Audio I/O Base Address

Selects a base I/O address for the audio.

Audio IRQ Select

Selects an IRQ for the audio.

USB Keyboard Support

Through the BIOS or Operating System.

Onboard IR Function

Enabled or Disabled.

Onboard Game Port

Specify the Base I/O address.

Onboard MIDI Port

Set the Base I/O address.

Onboard RAID

Enable or Disable. Only appears if you have a RAID controller on board.

MIDI IRQ Select

Selects the IRQ line for the onboard MIDI port.

Joystick Function

For onboard game ports.

MPU-401 Configuration

Configures the MPU-401 interface.

MPU-401 I/O Base Address

Selects a base address for the MPU-401 interface.

Serial Port 1/2 Interrupt

Select between the default PC AT interrupt or none.

PWRON After PWR-Fail

When *Off*, the system remains on when the PSU comes back on again. Otherwise, it will either power up or go to the former status (*Former-Sts*).

Keyboard Power On Function

How and whether you can power up the system from the PS/2 keyboard. *Specific Key* disables the power button, and requires a password.

SPECIFIC KEY FOR POWER ON

Appears when *Specific Key* is selected above. Type the password here.

Mouse Power On Function

As for *Keyboard Power On Function*, above, but for the PS/2 mouse. You need to double click on the selected button for it to work.

COMn

Usually controls the configuration of one or two serial (COM) ports on the motherboard.

AC97 Audio

Auto allows the main board to detect it automatically, in which case this will be enabled.

MC97 Modem

As above, but for modems.

Port 64/60 Emulation

Affects the USB port 64/60 emulation function. When enabled, you can use special key sequences on the USB keyboard.

Onboard Card Reader Type

Enables or disables two types of external memory device - memory stick and secure digital.

PCI Card Support for SMBus

Enables or disables the SMBus function on PCI slots for PCI 2.3 specification.

Disable Unused PCI Clock

This disables the clock of PCI slot that is not is use. When set to Yes, the system automatically detects the unused DIMM and PCI slots, and stops sending clock signals to them. When set to No, the system always sends clock signals. Set this to No if there are adapters that cannot be automatically detected by the system and will cause malfunctions.

Speech POST Reporter

The motherboard (an ASUS) can give out vocal POST messages when this is switched on.

SYSTEM MONITOR SETUP

Delay Prior To Thermal

For 0.13µ Pentium 4s with a 512KB L2 cache. These have a thermal sensor at the hottest part of the di, near the integer ALU units. When the processor reaches its maximum safe temperature, a PROCHOT# (*Processor Hot*) signal activates the TCC (Thermal Control Circuit), which inserts null cycles, typically 50-70% of the total, so the CPU "rests" for that proportion of the time. As temperature falls, the null cycles reduce. The THERMTRIP# signal is asserted at 135°C, which will shut down the CPU by removing its core voltage within ½ second. This setting controls the Thermal Monitor's automatic mode, allowing you to set when it activates after the system boots - the default is 16 Minutes, because the system is under a heavy load before then, and the temperature is artificially high in the core. It also allows the watchdog timer to generate a System Management Interrupt (SMI), allowing the BIOS to enable the Thermal Monitor under non-ACPI operating systems.

FAN Fail Alarm Selectable

This selects the fan that will be monitored for malfunction.

Shutdown when FAN Fail

When Enabled, the system wull be shut down if the CPU fan is not running.

CPU FanEQ Speed Control

This allows you to control the CPU fan speed down to a specific percentage, if the temperature limit set in Active Temperature is not exceeded, otherwise the CPU fan speed will be 100%.

Active Temperature

Sets the temperature limit that would activate the function of *CPU FanEQ Speed Control*

CPU Shutdown Temperature

Sets the temperature that would shutdown the system automatically to prevent overheating.

All Voltages, Fans Speed and Thermal Monitoring

These (unchangeable) items list the current status of the CPU and environment temperatures, fan speeds, and system power voltage. These features occupy addresses 294H to 297H (so be careful if you have a network adapter, sound card or other add-in cards that might use them).

Fan Speed

The speed of the fan connected to the headers listed here. The value assumes 2 pulses per revolution and should therefore be used as a relative figure.

Voltage Values

Shows current values on the motherboard. +3.3v, +5v, +12v, -12v and -5v come from the ATX power supply. VTT (+1.5) is GTL Termination Voltage from the on-board regulator and VCCVID (CPU) is the CPU core voltage from the on-board switching power supply.

VCCVID(CPU) Voltage, VTT(+1.5V) Voltage

The current value of significant voltages on the motherboard. *VTT* is the GT Termination voltage from the regulator. *VCCVID* is the CPU core voltage from the power supply.

I/O Plane Voltage

When the CPU Power Plane is set to Dual Voltage, you can choose the I/O or external voltage. Otherwise, this setting will not be present.

Core Plane Voltage

When the CPU Power Plane is set to Dual Voltage, you can choose the Core voltage. Otherwise, this setting will not be present.

Plane Voltage

When the CPU Power Plane is set to Single Voltage, you can choose the voltage, which should be correct for your CPU. Otherwise, this setting will not be present.

LCD&CRT

Select the combinations of display you want to use, either or both.

NOTES

16 NASTY NOISES

These are for errors that occur before the screen is initialised, or troubleshooting without a monitor. A text message is often sent to mono and CGA CRTs; EGA/VGA cards may not yet be initialised. Beep codes occur *after* the fact.
Machines with thermal monitoring broadcast a siren-like sound if the temperature of either the system or CPU meets or exceeds that set in the CMOS.

ALR

See Phoenix POST Codes.

AMBRA

See Phoenix POST Codes.

AMI

Beeps	What they mean	What to do
1	The memory refresh circuitry is faulty	Reseat/replace memory
2	Parity Errors in first 64K (detection may be defective)	Reseat/replace memory
3	Failure in the first 64K (could be address line error)	Reseat/replace memory
4	System Timer fail; Timer #1 not working properly (#2 is non-fatal)	Repair motherboard
5	CPU has generated an undetectable error	Repair motherboard
6	8042-Gate A20 Failure. CPU cannot go into protected mode	Keyboard/ controller
7	The CPU has generated an exception error	Repair motherboard
8	Video adapter missing or faulty memory (non-fatal)	Replace memory/ card
9	ROM checksum does not match that in the BIOS	Reseat/replace BIOS
10	Shutdown register for Interrupt #2 (can't retrieve CMOS in POST)	Repair motherboard
11	L2 cache memory has failed and been disabled	
2 short	POST failed; failure of a hardware testing procedure	
1 L 2 S	Video BIOS ROM failure (checksum) or adapter horizontal retrace	
1 L 3 S	Video DAC, monitor detection or video RAM has failed	
1 L 3 S	Conventional/extended memory test failure (older BIOSes)	
1 L 8 S	Display test and vertical and horizontal retrace test failed	
1 long	POST passed	

AST

Long	Short	Problem
0	1	Failed POST 1; Low level processor verification test
0	2	Failed POST 2; Clears keyboard controller buffers
0	3	Failed POST 3; Keyboard controller reset
0	4	Failed POST 4; Low level keyboard controller interface test.
0	5	Failed POST 5; Reading data from keyboard controller
0	6	Failed POST 6; System board support chip initialisation
0	7	Failed POST 7; Processor register r/w verify test
0	8	Failed POST 8; CMOS timer initialisation
0	9	Failed POST 9; ROM BIOS Checksum test
0	10	POST 10; Initialise primary video (never fails)
0	11	Failed POST 11; 8254 timer channel 0 test
0	12	Failed POST 12; 8254 timer channel 1 test
0	13	Failed POST 13; 8254 timer channel 2 test
0	14	Failed POST 14; CMOS power on and time test
0	15	Failed POST 15; CMOS shutdown byte test
1	0	Failed POST 16; DMA channel 0 test
1	1	Failed POST 17; DMA channel 1 test
1	2	Failed POST 18; DMA page register test
1	3	Failed POST 19; Keyboard controller interface test
1	4	Failed POST 20; Memory refresh toggle test
1	5	Failed POST 21; First 64K memory test
1	6	Failed POST 22; Setup interrupt vector table
1	7	Failed POST 23; Video initialisation
1	8	Failed POST 24; Video memory test

Advantage/Bravo/Manhattan/Ascentia/Premium/Premmia

Short	Long	Short	Replaceable Unit
3	1	X	System board
3	2	X	System board
3	3	X	System board
3	4	X	System board
3	5	X	SIMM memory
3	6	X	Integrated VGA or video board

Advantage/Bravo

Beeps	Replaceable Unit
1	System board
2	SIMM memory; System board
3	SIMM memory; System board
4	SIMM memory; System board
5	Processor; System board
6	Keyboard controller; System board
7	Processor; System board
8	Video adapter; Video RAM; System board
9	BIOS; System board
10	System board
11	External cache; System board

Manhattan

Beeps	Error Type	Replaceable Unit
1	Memory Refresh	DIMMs
2	Parity	DIMMs
3	Base 64KB Memory	DIMMs
4	Timer Not Operational	Processor board
5	Processor	Microprocessor or processor board
6	Gate A20	Keyboard or system board
7	Processor Interrupt	Microprocessor or processor board
8	Video Memory	Add-in video/system board (not fatal)
9	ROM Checksum	System board
10	CMOS Register	System board
11	Cache Memory Bad	Processor or processor board
2-2-3		System Board
3-1-1		SIMMs; Processor board
3-1-3		System board
3-4-1		SIMMs; Processor board
3-4-3		SIMMs, Processor board
2-1-2-3		Flash BIOS; System board
2-2-3-1		System board; Processor board

Ascentia J

Beeps	Replaceable Unit
2-2-3	System Board
3-1-1	SIMMs; Processor board

Beeps	Replaceable Unit
3-1-3	System board
3-4-1	SIMMs; Processor board
3-4-3	SIMMs, Processor board
2-1-2-3	Flash BIOS; System board
2-2-3-1	System board; Processor board

Ascentia 810/800/Explorer/Bravo

Short	Long	Short	Replaceable Unit
1	1	X	Processor board
1	2	X	System board
1	3	X	Processor board memory
1	4	X	Processor board
2	X	X	Processor board memory
3	1	X	System board
3	2	X	System board
3	3	X	Video (Processor board, LCD)
3	4	X	Video (Processor board, LCD)
4	2	X	Processor board
4	3	X	Processor board
4	4	1	Serial port/System board
4	4	2	Parallel port/System board
4	4	3	Processor board

BIOS Update Beep Codes

Long	Short	Description
2	0	Update Successful
2	2	CMOS Checksum failure; try again, be prepared to replace system board
2	3	Floppy disk adapter. Reinsert the disk
2	4	Disk belongs to another machine
2	5	Not a BIOS update disk
2	7	Flash programming error
2	8	Flash programming error
2	9	Flash programming error
2	10	Flash programming error
2	11	Flash programming error
2	12	Flash programming error
2	13	Flash programming error
2	14	Flash programming error

AST Enhanced

Short	Long	Short	Processor failure
3	1	X	Flash Loader failure (BIOS)
3	2	X	System Board component failure
3	3	X	System Board component failure
3	4	X	Memory failure
3	5	X	Video failure
0	6	X	Flash BIOS update error. Not early POST failure
	2	Any	Used by AST for low level diagnostics

Early Premium 286

Short	Long	Meaning
1	2	Video Error
1	3	Keyboard Error
2	0	Any Fatal Error
1	0	No errors during POST

Early POST Beep Codes

Beeps	Meaning
1	System Board
2	SIMM Memory; System Board
3	SIMM Memory; System Board
4	SIMM Memory; System Board
5	Processor; System Board
6	Keyboard Controller; System Board
7	Processor; System Board
8	Video Adapter; Video RAM; System Board
9	BIOS; System Board
10	System Board
11	External cache; System Board

AST Phoenix

Beeps	Meaning
1-1-3	CMOS read/write error. Fatal
1-1-4	ROM BIOS Checksum failure. Fatal
1-2-1	Programmable interval timer failure. Fatal
1-2-2	DMA Initialisation failure. Fatal
1-2-3	DMA Page Register r/w failure. Fatal
1-3-1	RAM refresh verification error. Fatal
1-3-3	First 64K RAM chip or data or data line failure multibit. Fatal
1-3-4	First 64K RAM odd/even logic failure. Fatal
1-4-1	Address line failure first 64K RAM. Fatal
1-4-2	Parity failure first 64K RAM. Fatal
2-1-1	First 64K RAM failure bit 0. Fatal
2-1-2	First 64K RAM failure bit 1. Fatal
2-1-3	First 64K RAM failure bit 2. Fatal
2-1-4	First 64K RAM failure bit 3. Fatal
2-2-1	First 64K RAM failure bit 4. Fatal
2-2-2	First 64K RAM failure bit 5. Fatal
2-2-3	First 64K RAM failure bit 6. Fatal
2-2-4	First 64K RAM failure bit 7. Fatal
2-3-1	First 64K RAM failure bit 8. Fatal
2-3-2	First 64K RAM failure bit 9. Fatal
2-3-3	First 64K RAM failure bit A. Fatal
2-3-4	First 64K RAM failure bit B. Fatal
2-4-1	First 64K RAM failure bit C. Fatal
2-4-2	First 64K RAM failure bit D. Fatal
2-4-3	First 64K RAM failure bit E. Fatal
2-4-4	First 64K RAM failure bit F. Fatal
3-1-1	Slave DMA register failure. Fatal
3-1-2	Master DMA register failure. Fatal
3-1-3	Slave interrupt mask register failure. Fatal
3-1-4	Slave interrupt mask failure. Fatal
3-2-4	Keyboard controller test failure. Fatal
3-3-4	Screen memory test failure. Fatal
3-4-1	Screen initialisation failure. Fatal
3-4-2	Screen retrace test failure. Fatal
3-4-3	Search for video ROM failure
4-2-1	No timer tick. Non-fatal
4-2-3	Gate A20 failure. Non-fatal
4-2-4	Unexpected interrupt in protected mode. Non-fatal

AWARD

v4.51

Beeps	Meaning
1 long 3 short	Video error
1 log repeated	DRAM error

XT 8086/88 v3.0

Beeps	Meaning
1 long, 2 short	Video error
2 short with PRESS F1 TO CONTINUE	Any non-fatal error
1 short	No error during POST

286/386 v3.03

Beeps	Meaning
1 long, 2 short	Video error
2 short with PRESS F1 TO CONTINUE	Any non-fatal error
1 short	No error during POST
1 long, 3 short, with system halt.	Keyboard controller error

EGA BIOS v1.6

Beeps	Meaning
1 long, 2 short	Video error
1 long, 3 short	EGA memory error

COMPAQ

General

Message	Beeps	What they mean
163 Time and date not set	2 Short	Invalid time or date
	2 V Short	Power-on successful
RESUME F1 key	None	Any failure
	3 Long	Processor Self-test
	2 Long	Memory map failure
101-I/O ROM error	1 L 1 Short	Option ROM checksum
101-ROM error	1 L 1 Short	System ROM checksum
102-System Board Failure	None	DMA or timers
102-System or Memory Board Failure	None	High-order addresses
162-System Options Error	2 Short	No floppies/mismatched types
162-System Options Not Set (Run SETUP)	2 Short	System SETUP
163-Time and Date Not Set	2 Short	Invalid time or date in CMOS
164-Memory Size Error	2 Short	Memory size discrepancy
170-Expansion Device not Responding	1 Short	Expansion device not responding
172-EISA Configuration Memory Corrupt	1 Short	CMOS Corrupt
173-PCI Slot ID Mismatch	1 Short	CMOS not Updated
174-ISA/PCI Configuration Slot Mismatch	1 Short	Plug & Play board not found
175-ISA/PCI Configuration Slot Mismatch	1 Short	CMOS not updated (Plug & Play)
176-Slot with No Readable ID (Run SETUP)	1 Short	CMOS not updated (Plug & Play)
177-SETUP Not Complete (Run SETUP)	1 Short	EISA Configuration not complete
178-Processor SETUP Invalid (Run setup)	None	Processor SETUP invalid
201-Memory Error	None	RAM failure
203-Memory Error	None	RAM failure
205-Cache Memory Failure	None	Cache Memory Error
206-Secondary Cache Controller Failure	None	Cache Memory Controller Failure
301-Keyboard Error	None	Keyboard failure
301-Keyboard Error or Test Fixture Installed	None	Keyboard test fixture
303-Keyboard Controller Error	None	Keyboard controller
304-Keyboard or System Unit Error	None	Keyboard interface
401-Printer Error	None	Printer controller
401-Port 1 Address Conflict	2 Short	Ext/Int Port assignments to Port 1
402-Monochrome Adapter Failure	1 L 2 short	Monochrome display controller
501-Display Adapter Failure	1 L 2 short	Video display controller
601-Diskette Drive Controller Error	None	Diskette drive controller
602-Diskette Drive Boot Record Error	None	Diskette media not bootable
605-Diskette Drive Type Error	2 Short	Wrong drive type used in setup

Message	**Beeps**	**What they mean**
607-Diskette Drive Controller Error	2 Short	Configuration error
611-Primary Diskette Drive Conflict	2 Short	Configuration error
612-Secondary Diskette Drive Conflict	2 Short	Configuration error
702-A-Coprocessor Detection Error	2 Short	Add copro or configuration error
703-Coprocessor Detection Error	2 Short	Add copro or configuration error
1125-Internal Serial Port Failure	2 Short	Defective internal serial port
1150-xx Comm Port Setup Error	2 Short	Setup not correct (run SETUP)
1151-COM1 Address Conflict	2 Short	Ext/int port assignments to COM1
1152-COM2 Address Conflict	2 Short	Ext/int port assignments to COM2
1153-COM3 Address Conflict	2 Short	Ext/int port assignments to COM3
1153-COM 4 Address Conflict	2 Short	Ext/int port assignments to COM4
1154-Port 4 Address Conflict	2 Short	Incorrect COM 4 assignment
1600-32-Bit System Manager Board	2 Short	Configuration mismatch
1730-HD 0 Does Not Support DMA	2 Short	Configuration mismatch
1731-HD 1 Does Not Support DMA	2 Short	Configuration mismatch
1740-HD 0 Failed Set Block Command	2 Short	Configuration mismatch
1741-HD 1 Failed Set Block Command	None	Wrong drive type
1750-Hard Drive 0 Failed Identify	None	Wrong drive type
1751-Hard Drive 0 Failed Identify	None	Wrong drive type
1760-HD 0 Does Not Support Block Mode	2 Short	Configuration mismatch
1761-HD 1 Does Not Support Block Mode	2 Short	Configuration mismatch
1771-Primary Drive Port Address Conflict	2 Short	Int & ext HD controllers on primary address
1772-Secondary Disk Port Address Conflict		Int & external HD controllers on sec address
1780-Hard Drive 0 Failure	None	Hard drive/format error
1781-Hard Drive 1 Failure	None	Hard drive/format error
1782-Hard Drive Controller Failure	None	Hard drive controller
1790-Hard Drive 0 Error	None	Wrong drive type used in SETUP
1791-Hard Drive 1 Error	None	Wrong drive type used in SETUP
1792-Secondary Drive Controller Error	None	Hard drive error or wrong drive type
1793-Secondary Controller/Drive Failure	None	Hard drive error or wrong drive type
XX000Y ZZ Parity Check 2	None	RAM parity: Addr (XX), byte (Y) data bit (ZZ)
Hard Drive Parameter Table or BIOS Error	3 Long	Configuration or hardware failure
IOCHECK Active, Slot X	None	Defective board in slot x
Bus Master Timeout Slot X	None	Defective board in slot x
Audible	1 Short	Power-On successful
Audible	2 Short	Power-On successful
(RESUME F1 KEY)	None	As indicated to continue

Contura 400 Family

Message on Screen	Beeps	What They Mean
101 System ROM Error	1 L 1 S	System ROM Checksum
101 I/0 ROM Error	None	Option ROM Checksum
102 System Board Failure	None	DMA, timers, or unsupported processor
162 System Options Error	2 Short	No diskette drive or drive mismatch
162 System Options Not Set	2 Short	Configuration incorrect
163 Time & date Not Set	2 Short	Invalid time or date in CMOS
164 Memory Increase Detected	2 Short	CMOS incorrect
164 Memory Decrease Detected	2 Short	CMOS incorrect
168 CMOS Checksum invalid		
201 Memory Error	None	RAM failure
203 Memory Address Error	None	RAM failure
205 Memory Error	None	Cache memory error
207 Invalid Memory Module	None	Memory module installed incorrectly
209 NCA RAM Error	None	RAM Failure Error
211 Memory Failure	None	RAM Failure
301 Keyboard Error	None	Keyboard Failure
303 Keyboard Controller Error	None	System board keyboard controller
304 Keyboard or System Unit Error	None	Keyboard or System Unit Error
401 Printer Error	None	Printer controller
402 Monochrome Adapter Failure	1 L 2 S	Monochrome display controller.
501 Display Adapter Failure	1 L 2 S	Video display controller
601 Diskette Controller Error	None	Diskette controller circuitry
602 Diskette Boot	None	Diskette in drive A not
605 Diskette Drive Error	2 Short	Mismatch in drive type
702 Coprocessor Detection Error	None	Coprocessor upgrade detection error
702A Coprocessor Detection Error	2 Short	Coprocessor upgrade detection error
703 A Coprocessor detected by POST	2 Short	Coprocessor or CMOS Error
1125 Internal Serial Port Failure	2 Short	Defective internal serial port
1780 Disk 0 failure	None	Hard drive/format error
1782 Disk Controller	None	Hard drive circuitry error
1790 Disk 0 Failure	None	Hard drive error or wrong drive type
Audible	1 Short	Poweron successful

DELL (PHOENIX)

A 1-2 code may mean a bootable add-in card is installed without a boot device attached.

Beeps	Meaning
1-1-2	Microprocessor register failure
1-1-3	Non-volatile RAM
1-1-4	ROM BIOS Checksum failure
1-2-1	Programmable interval timer
1-2-2	DMA Initialisation failure
1-2-3	DMA Page Register r/w failure
1-3	Video memory test failure
1-3-1/2-4-4	SIMMs not properly identified or used
3-1-1	Slave DMA register failure
3-1-2	Master DMA register failure
3-1-3	Master interrupt mask register failure
3-1-4	Slave interrupt mask register failure
3-2-2	Interrupt vector loading failure
3-2-4	Keyboard controller test failure
3-3-1	Non-volatile RAM power loss
3-3-2	Non-volatile RAM configuration
3-3-4	Video memory test failure
3-4-1	Screen initialisation failure
3-4-2	Screen retrace failure
3-4-3	Search for video ROM failure
4-2-1	No time tick
4-2-2	Shutdown failure
4-2-3	Gate A20 failure
4-2-4	Unexpected interrupt in protected mode
4-3-1	Memory failure above address
4-3-3	Timer chip counter 2 failure
4-3-4	Time-of-day clock stopped
4-4-1	Serial port test failure
4-4-2	Parallel port test failure
4-4-3/4-4-4	Maths coprocessor test failure/Cache test failure

IBM

Beeps	Meaning
1-1-3	CMOS Read/Write Error
1-1-4	ROM BIOS Check Error
1-2-X	DMA Error
1-3-X	Memory Module
1-4-4	Keyboard
1-4-X	Error in first 64K RAM
2-1-1	Run Setup
2-1-2	Run Setup
2-1-X	1st 64K RAM failed
2-2-2	Video Adapter
2-2-X	1st 64K RAM failed
2-3-X	Memory Module
2-4-X	Run Setup
3-1-X	DMA Register failed
3-2-4	Keyboard controller failed
3-3-4	Screen initialisation failed
3-4-1	Screen retrace test detected an error
3-4-2	POST searching for video ROM
4	Video adapter
All others	System board
1 long, 1 Short	Base 640K or Shadow RAM error
1 Long, 2-3 short	Video adapter
3 Short	System Board Memory
Continuous	System Board
Repeating Short	Keyboard stuck
None	System Board

AT

Beeps	Meaning
1 short	Normal POST, OK
2 short	POST error-check messages on display
Continuous	Power supply, system board
Repeating short beeps	Power supply, system board
1 long, 1 short	System board
1 long, 2 short	Display adapter (MDA, CGA)
1 long, 3 short	EGA adapter
3 long	3270 keyboard card

MR BIOS

More under *POST Codes*.

Long	Short	Problem
0	1	Failed POST 1; Low level processor verification test.
0	2	Failed POST 2; Clears keyboard controller buffers.
0	3	Failed POST 3; Keyboard controller reset.
0	4	Failed POST 4; Low level keyboard controller i/f test.
0	5	Failed POST 5; Reading data from keyboard controller.
0	6	Failed POST 6; System board support chip initialisation.
0	9	Failed POST 9; ROM BIOS Checksum test.
0	13	Failed POST 13; 8254 timer channel 2 test.
0	15	Failed POST 15; CMOS shutdown byte test.
1	0	Failed POST 16; DMA channel 0 test.
1	1	Failed POST 17; DMA channel 1 test.
1	2	Failed POST 18; DMA page register test.
1	5	Failed POST 21; First 64K memory test.
1	6	Failed POST 22; Setup interrupt vector table.
1	7	Failed POST 23; Video initialisation.
1	8	Failed POST 24; Video memory test.

MYLEX/EUROSOFT

Beep	Meaning	386 Codes
1	Always present. (e.g. start)	1L
2	Video Adapter (missing?)	2L
3	Keyboard controller	1L-1S-1L
4	Keyboard	1L-2S-1L
5	8259 PIC 1	1L-3S-1L
6	8259 PIC 2	1L-4S-1L
7	DMA page register	1L-5S-1L
8	RAM Refresh	1L-6S-1L
9	RAM data test	1L-7S-1L
10	RAM parity	1L-8S-1L
11	8237 DMA controller 1	1L-9S-1L
12	CMOS RAM	1L-10S-1L
13	8237 DMA controller 2	1L-11S-1L
14	CMOS battery	1L-12S-1L
15	CMOS RAM checksum	1L-13S-1L
16	BIOS ROM checksum	1L-14S-1L
	Multiple errors	1L +

NEC

Prospeed 486SX/C

Diag Port Output	Beeps	Description
EB_PDIF EQU0400h		Pointing device interface failure (mouse)
E_REGSEQU 01h		80486 register test in progress
E_CRAMEQU 02h	1-1-3	CMOS write/read test in progress or failure
E_BROMEQU 03h	1-1-4	BIOS ROM checksum in progress or failure
E_TIMREQU 04h	1-2-1	Programmable Interval Timer test in progress or failure
E_DMAIEQU 05h	1-2-2	DMA initialization in progress or failure
E_PAGEEQU 06h	1-2-3	DMA page register write/read test in progress or failure
E_REFREQU 08h	1-3-1	RAM refresh verification in progress or failure
E_RAM0EQU 09h		1st 64 K RAM test in progress
E_MMULEQU 0Ah	1-3-3	1st 64 K RAM chip or data line failure
E_MOELEQU 0Bh	1-3-4	1st 64 K RAM odd/even logic failure
E_MADDEQU 0Ch	1-4-1	1st 64 K RAM address line failure
E_RAMPEQU 0Dh	1-4-2	1st 64 K RAM parity test in progress or failure
E_FSTIMREQU 0Eh	1-4-3	Fail-safe timer test in progress
E_NMIIOEQU 0Fh	1-4-4	Software NMI port test in progress
E_MBITEQU 10h	2-1-1	1st 64 K RAM chip or data line failure - bit 0
E_MBITEQU 11h	2-1-2	1st 64 K RAM chip or data line failure - bit 1
E_MBITEQU 12h	2-1-3	1st 64 K RAM chip or data line failure - bit 2
E_MBITEQU 13h	2-1-4	1st 64 K RAM chip or data line failure - bit 3
E_MBITEQU 14h	2-2-1	1st 64 K RAM chip or data line failure - bit 4
E_MBITEQU 15h	2-2-2	1st 64 K RAM chip or data line failure - bit 5
E_MBITEQU 16h	2-2-3	1st 64 K RAM chip or data line failure - bit 6
E_MBITEQU 17h	2-2-4	1st 64 K RAM chip or data line failure - bit 7
E_MBITEQU 18h	2-3-1	1st 64 K RAM chip or data line failure - bit 8
E_MBITEQU 19h	2-3-2	1st 64 K RAM chip or data line failure - bit 9
E_MBITEQU 1Ah	2-3-3	1st 64 K RAM chip or data line failure - bit A
E_MBITEQU 1Bh	2-3-4	1st 64 K RAM chip or data line failure - bit B
E_MBITEQU 1Ch	2-4-1	1st 64 K RAM chip or data line failure - bit C
E_MBITEQU 1Dh	2-4-2	1st 64 K RAM chip or data line failure - bit D
E_MBITEQU 1Eh	2-4-3	1st 64 K RAM chip or data line failure - bit E
E_MBITEQU 1Fh	2-4-4	1st 64 K RAM chip or data line failure - bit F
E_DMASEQU 20h	3-1-1	Slave DMA register test in progress or failure
E_DMAMEQU 21h	3-1-2	Master DMA register test in progress or failure
E_PIC0EQU 22h	3-1-3	Master interrupt mask register test in progress or failure
E_PIC1EQU 23h	3-1-4	Slave interrupt mask register test in progress or failure
E_IVLDEQU 25h		Interrupt vector loading in progress

Diag Port Output	Beeps	Description
E_KEYCEQU 27h	3-2-4	Keyboard controller test in progress or failure
E_CVEREQU 28h		CMOS power-fail and checksum checks in progress
E_CCONEQU 29h		CMOS config info validation in progress
E_CRTMEQU 2Bh	3-3-4	Screen memory test in progress or failure
E_CRTIEQU 2Ch	3-4-1	Screen initialization in progress or failure
E_CRTREQU 2Dh	3-4-2	Screen retrace tests in progress or failure
E_VROMEQU 2Eh		Search for video ROM in progress
E_OKEQU 30h		Screen believed operable w/video ROM
E_OKEQU 31h		Monochromatic screen believed operable
E_OKEQU 32h		40-column color screen believed operable
E_OKEQU 33h		80-column color screen believed operable

LOOPED ON POST

Diag Port Output	Beeps	Description
E_TIMRIEQU 34h	4-2-1	Timer tick interrupt test in progress or failure
E_SHUTDEQU 35h	4-2-2	Shutdown test in progress or failure
E_GTA20EQU 36h	4-2-3	Gate A20 failure
E_PMUXIEQU 37h	4-2-4	Unexpected interrupt in protected mode
E_BAMEMEQU 38h	4-3-1	RAM test in progress or failure above address 0FFFFh
E_TIMR2EQU 3Ah	4-3-3	Interval timer channel 2 test in progress or failure
E_TODCKEQU 3Bh	4-3-4	Time-of-Day clock test in progress or failure
E_SRLPTEQU 3Ch	4-4-1	Serial port test in progress or failure
E_PRLPTEQU 3Dh	4-4-2	Parallel port test in progress or failure
E_80287EQU 3Eh	4-4-3	Math coprocessor test in progress or failurePackard Bell

PACKARD BELL

See *Phoenix*.

PHOENIX

Refer to *POST Codes*.

QUADTEL

Beeps	**Meaning**
1	POST OK
2	Configuration Error; CMOS has changed.
1 long, 2 short	Video or adapter RAM
1 long, 3 short	Faulty expansion card.

TANDON

Slimline 286, 386SX and 486; 486 EISA

Beeps	**Meaning**
L-S-L-S	8254 counter timer.
S-L-S	RAM Refresh
L-L-L	System RAM
S-S-S	BIOS ROM Checksum
L-L	Distinct lack of video adapter
L-L-L-L	Video Adapter Failure

Error Messages & Codes

17

A message saying: *Abnormal System Hardware, Press F1 to enter Setup or any key to continue* means that there is no fan connected to the 3-pin header on the motherboard - even if a 4-pin fan is connected directly to the power supply.

Only the more modern manufacturers have been included here, since even IBM now use third party BIOSes - please contact the author if you need anything more obscure.

AMI

Message	Fault	Action
CH-2 Timer Error	Non fatal. Could be peripheral.	
INTR #1 Error	Interrupt Ch 1 failed POST	Check IRQs 0-7.
INTR #2 Error	As above, for Interrupt Ch 2	Check IRQs 8-15
CMOS Battery Low		Replace battery
CMOS Checksum Failure	A checksum is generated when CMOS values are saved for error checking on subsequent startups. This will appear if the checksum is different	Run Setup again
CMOS Memory Size Mismatch	More memory in, or some doesn't work	Run Setup
CMOS System Options Not Set	CMOS values are corrupt or non-existent	Run Setup
CMOS Display Type Mismatch	The display in the CMOS does not match what is actually found by the POST	Run Setup
CMOS Memory Size Mismatch	The memory in the BIOS does not match that actually found on the motherboard	Run Setup again
Display Switch Not Proper	Some motherboards have a switch or jumper setting which is changed if a monochrome or colour monitor is fitted	Reset the switch
Keyboard is locked ... Unlock it		Unlock keyboard
Keyboard Error	There is a timing problem with the keyboard	Check keyboard BIOS compatible, or set to Not Installed, to skip keyboard test.
K/B Interface Error	Error with keyboard connector	
FDD Controller Failure	The BIOS cannot communicate with the floppy controller	It may be disabled, or cable may be loose
HDD Controller Failure	As above, but for hard disks	

Message	Fault	Action
C: Drive Error	There is no response from hard disk drive C	Hard disk type may be set incorrectly, not formatted, or not properly connected (check voltage)
D: Drive Error	As above	
C: Drive Failure	As above but more serious	
D: Drive Failure	As above	
CMOS Time & Date Not Set		Run the Setup program.
Cache Memory Bad, Do Not Enable Cache!	Speaks for itself	You may need new cache memory. Try reseating first.
8042 Gate-A20 Error	The gate-A20 portion of the keyboard controller has failed	Replace keyboard chip (8042)
Address Line Short!	There is an error in the memory address decoding circuitry	Try rebooting, it might go away!
DMA #1 Error	Error in the first DMA channel	Could be a peripheral
DMA #2 Error	Error in second DMA channel	Could be a peripheral
DMA Error	An error within the DMA controller on the motherboard.	
No ROM Basic	There is nothing to boot from; may be no bootable sector on the boot up disk (A or C). The original IBM PC ran Basic from a ROM at this point (it was in a ROM next to the BIOS), but modern machines don't have it, hence this message	Check you haven't disabled booting from the A: drive, or that you've got A:, C: as the boot sequence, ot you have an active partition
Diskette Boot Failure	Ddiskette in drive A: is corrupt.	
Invalid Boot Diskette	As above, but the disk is readable	
On Board Parity Error	There is a parity error with memory on the motherboard at address XXXX (hex). On board means the memory is not on an expansion card	Possibly correctable with software from motherboard manufacturer
Off Board Parity Error	There is a parity error with memory installed in an expansion slot at address XXXX (hex)	Possibly correctable with software from the board manufacturer.Ttry reseating SIMMs
Parity Error ????	A parity error with memory somewhere in the system, but God knows where. Possibly correctable with software from the motherboard manufacturer	
Memory Parity Error at XXXX	Memory failed, displayed as XXXX. If not, as ????	
I/O Card Parity Error at XXXX	An expansion card failed. If the address can be determined, it is displayed as XXXX, otherwise ????	

Message	Fault	Action
DMA Bus Time-out	A device has driven the bus signal for more than 7.8 microseconds	
Memory mismatch, run Setup		Try disabling Memory Relocation
EISA CMOS Checksum Failure	The checksum for EISA CMOS is bad, or the battery	
EISA CMOS inoperational	Read/Write error in ext CMOS RAM	The battery may be bad
Expansion Board not ready Slot X, Y, Z.	Ccannot find expansion board in whatever slot is indicated	Check board is in correct slot and seated
Fail-Safe Timer NMI Inoperational	Devices that depend on the fail-safe NMI timer are working	
ID information mismatch Slot X, Y, Z	The ID of the EISA Expansion Board in whatever slot is indicated does not match the ID in EISA CMOS RAM	
Invalid Configuration for Slot X, Y, Z	The configuration information for EISA Expansion Board X, Y or Z is not correct.	Run the ECU
Software Port NMI Inoperational	Ssoftware port NMI not working	
BUS Timeout NMI atSlot n	There was a bus timeout NMI at whatever slot is indicated	
(E)nable (D)isable Expansion Board?		Type E to enable the board that had an NMI, or D to disable it.
Expansion Board disabled at Slot n	The board at whatever slot is indicated has been disabled	
Expansion Board NMI at Slot n.	An expansion board NMI was generated from whatever slot is indicated	
Fail-Safe Timer NMI	A fail-safe timer NMI has been generated	
Software Port NMI	A software port NMI has been generated	

AST

See *AMI*.

AWARD

v4.5x

Code	Meaning
6	Cache/controller.
10	More than 1 IDE interface.
40	IDE floppy controller.
80	IDE controller.

XT 8086/88 v3.0

Code	Meaning
201	Memory test failed
301	Keyboard error
601	Diskette power on diagnostic test failed
1801	I/O expansion unit failed power on diagnostic
Parity Check 1	Parity error in system board memory. Fatal
Parity Check 2	Parity error in expansion unit memory. Fatal.

286/386 v 3.03

Msg	Meaning
Refresh Timing Error	The refresh clock is not operating as expected.
Keyboard Error/No Keyboard	Either a keyboard problem, or the keyboard is not attached.
Equipment Configuration Error	The system configuration determined by POST is different from what was defined using SETUP.
Memory Size Error	The amount of memory found by POST is different than the amount defined using SETUP.
Real Time Clock Error	The real time clock is not operating as expected.
Error initialising Hard Drive	Reset of fixed disk failed.
Error Initialising HD Controller	Fixed disk controller fails internal diagnostic.
Floppy Disk Cntrlr Error Or No Cntrlr Present	The floppy disk controller is failing self test, or it is not present.
CMOS RAM Error	CMOS invalid (battery). Run SETU.
Press A key To Reboot	Call made to ROM BASIC; not in Award BIOS.
Memory Address Error At XXXX	Memory errors. Values are as close as possible.
Disk Boot Failure, Insert System Disk & Press Enter	The system is unable to load the system from the boot disk.
Parity Error In Segment XXXX	This fatal error occurs during POST memory test.
Memory Verify Error	POST error. AA is # of MBytes AA:SSSS:FFFF boundary; SSSS=segment FFFF=offset.
IO Parity Error-System Halted	These occur after POST has finished.

ISA/EISA v4.5

Message	Meaning
CMOS BATTERY FAILED	CMOS battery is no longer functional. It should be replaced.
CMOS CHECKSUM ERROR	CMOS Checksum is incorrect. This can indicate that CMOS has become corrupt. This error may have been caused by a weak battery. Check the battery and replace if necessary.
DISK BOOT FAILURE, INSERT SYSTEM DISK AND PRESS ENTER	No boot device was found. Either a boot drive was not detected or the drive has no proper system files. Insert a system disk into Drive A: and press Enter. If you assumed the system would boot from the hard drive make sure the controller is inserted correctly and cables are attached. Also ensure the disk is formatted as a boot device. Then reboot.
DISKETTE DRIVES OR TYPES MISMATCH ERROR-RUN SETUP	Type of diskette drive installed in the system is different from the CMOS definition. Run Setup to reconfigure the drive type correctly.
DISPLAY SWITCH IS SET INCORRECTLY	The Display switch on motherboard is set to a different setting than indicated in Setup. Determine which setting is correct, and then either turn off the system and change the jumper, or enter Setup and change the VIDEO selection.
DISPLAY TYPE CHANGED SINCE LAST BOOT	Since last powering off, the display adapter has been changed. Reconfigure the system.
EISA Configuration Checksum Error	Run the EISA Configuration Utility. The EISA non-volatile RAM checksum is incorrect or cannot correctly read the EISA slot. Either the EISA NVR has become corrupt or the slot has been configured incorrectly. Make sure the card is installed firmly in the slot. When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.
EISA Configuration Is Not Complete	Run EISA Configuration. The slot configuration information stored in the EISA non-volatile memory is incomplete. When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.
ERROR ENCOUNTERED INITIALIZING HARD DRIVE	Hard drive cannot be initialized. Be sure the adapter is installed correctly and all cables are correctly and firmly attached. Also make sure the correct hard drive type is selected in Setup.
ERROR INITIALIZING HARD DISK CONTROLLER	Cannot initialize controller. Make sure cord is correctly installed. Be sure the correct hard drive type is in Setup. Also check to see if any jumper needs to be set correctly on the hard drive.
FLOPPY DISK CNTRLR ERROR OR NO CNTRLR PRESENT	Cannot find or initialize floppy controller. If there are no floppy drives installed, be sure the Diskette Drive selection in Setup is set to NONE.
Invalid EISA Configuration	Run the EISA Configuration Utility. TheNVR containing EISA configuration information was programmed incorrectly or has become corrupt. Re-run EISA configuration utility to correctly program the memory. When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.
KEYBOARD ERROR OR NO KEYBOARD PRESENT	Cannot initialize the keyboard. Make sure it is attached correctly and no keys are being pressed during the boot. If you are purposely configuring the system without a keyboard, set the error halt condition in Setup to HALT ON ALL, BUT KEYBOARD so the BIOS will ignore the missing keyboard and continue the boot.
Memory Address Error at ...	Indicates a memory address error at a specific location. You can use this location along with the memory map for your system to find and replace the bad memory chips.

Message	Meaning
Memory parity Error at ...	Indicates a memory parity error at a specific location. You can use this location along with the memory map for your system to find and replace the bad memory chips.
MEMORY SIZE CHANGED SINCE LAST BOOT	Memory has been added or removed since the last boot. In EISA mode use Configuration Utility to reconfigure the memory configuration. In ISA mode enter Setup and enter the new memory size in the memory fields.
Memory Verify Error at ...	Aan error verifying a value already written to memory. Use the location along with your system's memory map to locate the bad chip.
OFFENDING ADDRESS NOT FOUND	This message is used in conjunction with the I/O CHANNEL CHECK and RAM PARITY ERROR messages when the segment that has caused the problem cannot be isolated.
OFFENDING SEGMENT:	This message is used in conjunction with the I/O CHANNEL CHECK and RAM PARITY ERROR messages when the segment that has caused the problem has been isolated.
PRESS A KEY TO REBOOT	This will be displayed at the bottom screen when an error occurs that requires you to reboot. Press any key and the system will reboot.
PRESS F1 TO DISABLE NMI, F2 TO REBOOT	When BIOS detects a Non-maskable Interrupt condition during boot, this will allow you to disable the NMI and continue to boot, or you can reboot the system with the NMI enabled.
RAM PARITY ERROR-CHECKING FOR SEGMENT ...	A parity error in Random Access Memory.
Should Be Empty But EISA Board Found	When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility. A valid board ID was found in a slot that was configured as having no board ID. When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.
Should Have EISA Board But Not Found	Run EISA Config. The board is not responding to the ID request, or no board ID is in the slot. The system will boot in ISA mode, so you can run EISA Config.
Slot Not Empty	A slot designated empty by the Configuration Utility contains a board. When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.
SYSTEM HALTED, (CTRL-ALT-DEL) TO REBOOT ...	Present boot attempt has been aborted and system must be rebooted. Use CTRL-ALT- DEL.
Wrong Board In Slot	Run EISA Configuration Utility. The board ID does not match the ID stored in the EISA non-volatile memory. When this error appears, the system will boot in ISA mode, which allows you to run the EISA Configuration Utility.

HP VECTRA

Code	Meaning
000f	Microprocessor error
001x	BIOS ROM error
008x	Video ROM error
009x-bx	Option ROM error while testing address range c800-dfff
00cx-dx	Option ROM error while testing address range e000-efff
011x	RTC error while testing the CMOS register

Code	Meaning
0120	RTC error
0130	RTC/System configuration error
0240	CMOS memory/system configuration error
0241	CMOS memory error
0250	Invalid configuration
0280	CMOS memory error
02c0-c1	EEPROM error
02d0	Serial # not present
030x-3x	Keyboard/Mouse controller error
034x-5x	Keyboard test failure
03e0-4	Keyboard/Mouse controller error
03e5-b	Mouse test failure
03ec	Keyboard/Mouse controller error
0401	Protected Mode failure
050x	Serial Port error
0506	Datacomm conflict
0510-20	Serial Port error
0543-5	Parallel Port error
0546	Datacomm conflict
06xx	Keyboard key stuck
07xx	Processor speed error
0800	Boot ROM conflict
0801	Boot ROM not found
081x	Integrated Ethernet Interface errors
0900	Fan error
110x-01	Timer error
20xa	Memory mismatch
21xx/22xx	DMA error
30xx	HP-HIL error
4xxx	RAM error
5xxx	As above
61xx	Memory address line error
62xx	RAM parity error/memory controller error
630x	RAM test error
6400	As above
6500	BIOS ROM shadow error
6510	Video BIOS shadowing error/system ROM error
6520	Option ROM shadowing error
65a0-f0	Shadow error probably caused by system board memory
66xx	Shadow error probably caused by memory on accessory board
7xxx	Interrupt error
8003	Bad drive configuration
8004	CMOS Drive/System Configuration error
8005-6	Bad drive configuration
8007	CMOS Drive/System Configuration error
8048-a	Hard disk drive identity error
8050	Hard disk drive controller conflict
84xx	Bad boot sector
8x0d	Controller Busy/Controller Error

Code	Meaning
8x0e	Hard disk error
8x0f	Hard disk drive mismatch
8x10	Controller Busy/Controller Error
8x11	Hard disk drive control error
8x12	Controller Busy/Controller Error
8x13	Hard disk drive control error
8x20-1	Controller Busy/Controller Error
8x28	Hard disk drive splitting error
8x30	Hard disk drive control error
8x38	Controller Busy/Controller Error
8x39-b	Hard disk drive control error
8x3c	Controller Busy/Controller Error
8x40	As above
8x41-4	Hard disk drive control error
8x45	Controller Busy/Controller Error
8x49	Hard disk drive control error
8x4b	As above
9xxx	Flexible disk drive error
9x0a	Flexible disk drive conflict
9x10	As above
A00x	Numeric coprocessor error
B300	Cache controller error
B320	Memory cache module error
Cxxx	Extended RAM error (for HP-HIL PCs)
Exxx	Bus memory error

OLIVETTI

M24 Memory Errors

Code	Meaning
XXX	Last bank tested
CC	RAM configuration number 01 128K on m'bd 02 256K on m'bd 03 384K (256+128 exp) 04 512K (256+256 exp) 05 640K (256+384 exp) 06 640K (512 on bank 0 + 128K on bank 1)
Y	128K bank failure number (000=segment, ZZZZ=Offset) 1 Bank 0 on m'bd 2 Bank 1 on m'bd 3 Bank 1 on expansion 4 Bank 1 on expansion 5 Bank 2 on expansion
WWWW	Data Written (good byte)
RRRR	Data Read (bad byte)

PHOENIX

Message	Fault	Action
Diskette Drive x Error	Drive x present, fails POST	Check cabling and Setup.
Diskette drive reset failed		Check adapter.
Diskette read failure-strike F1	Diskette not formatted/defective.	Replace and retry.
Display adapter failed.	Colour/mono switch is wrong, or primary video adapter failed.	Check switch or adapter.
Errors found; incorrect config	The size of base or extended memory does not agree with the CMOS.	Run Setup.
Extended RAM failed	Extended memory not working or configured properly.	Restore values, see dealer.
Failing Bits:nnnn	number nnnn is a map of the bits at the RAM address (System, Extended or Shadow Memory) which failed testing. Each 1 represents a failed bit.	Contact your dealer.
Fixed Disk configuration error.	Configuration is not supported.	
Fixed Disk drive failure.		Reboot, or replace HD.
Fixed Disk read failure.		Reboot, replace HD.
Gate A20 function.	8042 not accepting commands.	Check system board.
Keyboard error nn	nn represents the scan code for a stuck key.	
Keyboard clock line failure.	Keyboard or cable is defective.	Check connections.
Keyboard data line failure.	Keyboard controller has failed.	
Memory failure at xxxx	Memory chip circuitry has failed.	Turn PC off, then on again.
No boot device available	Either floppy A or fixed disk is defective.	
No boot sector on fixed disk	Drive C: is not formatted or otherwise bootable.	
No timer tick interrupt.	Timer chip has failed.	Turn PC off, on again.
Option ROM checksum failure.	Expansion card has bad ROM.	Reboot, or replace card.
Parity Check 1	Parity error in the system bus.	
Parity Check 2	Parity error in the I/O bus.	
Pointer device failure.	Mouse failed.	Reboot, check cable.
Real Time Clock Error	RTC failed BIOS test.	May require board repair.
Shadow RAM failed:nnnn	Shadow RAM failed at offset nnnn of 64K block at which the error was detected.	
Shutdown failure.	Kbd controller/logic failed.	Check keyboard controller.
System Cache Error disabled	RAM cache failed the BIOS test, and has been disabled.	Contact your dealer.

Message	Fault	Action
System RAM failed	Shadow RAM failed at offset nnnn of 64K block with error.	
System Timer Error		Repair motherboard.
Timer 2 failure	Timer chip failed	Turn PC off, then on. Otherwise, see dealer.
Timer or interrupt controller.	Either timer chip or interrupt controller is defective.	Check timer chip.
Timer interrupt did not occur.	Either timer chip or interrupt controller is defective.	Check timer chip.
Unexpected interrupt	Hardware interrupt or NMI occurred.	timer chip or int controller.

POST CODES

18

During the POST on AT-compatibles and above, special signals are sent to I/O port 80H at the beginning of each test (XT-class machines don't issue POST codes, although some with compatible BIOSes do). Some computers may use a different port, such as 84 for the Compaq, or 378 (LPT1) for Olivettis. IBM PS/2s use 90 or 190 (20-286), whilst some EISA (Award) machines send them to 300H as well. Try 680 for Micro Channel. Those at 50h are chipset or custom platform specific, and you might find a few go to the parallel port (AT&T, NCR).

POST Diagnostic cards can display these POST codes, so you can check your PC's progress as it starts and hopefully diagnose errors when the POST stops, though a failure at any given location does not necessarily mean that part has the problem; it's meant to be a guidepost for further troubleshooting. In this chapter, some general instructions are given for a typical POST card, which were provided by Xetal Systems, together with some of the more obscure POST codes. Having obtained a code, identify the manufacturer of the chipset on the motherboard, then refer to the tables that follow. The POST checks at three levels, *Early*, *Late* and *System Initialisation*. Early POST failures are generally fatal and will produce a beep code, because the video will not be active; in fact, the last diagnostic during Early POST is usually on the video, so that Late failures can actually be seen. System Initialisation involves loading configuration from the CMOS, and failures will generate a text message. Consistent failures at that point indicate a bad battery backup. During the POST, all the devices on the machine are counted and given parameters within which to opperate. The responses are gathered together and compared against a previously calculated checksum as a further check of readiness.

Be aware that not all PCI POST cards allow all codes through the bridge.

Shutdown or Reset Commands

The Reset command stops the current operation and begins fetching instructions from the BIOS, as if the power has just been switched on. The Shutdown command, on the other hand, just forces the CPU to leave protected mode for real mode, so the system behaves differently after each one. Before issuing the shutdown command, the BIOS sets a value into the shutdown byte in the CMOS, which is checked after a reset, so the BIOS can branch to the relevant code and continue where it left off.

One of the problems with shutdown handling is that the POST must do some handling before anything else, immediately after power-on or system reset. The path between the CPU and the BIOS ROM, as well as basic control signals, has to be working before the POST gets to its first diagnostic test (usually the CPU register test), so some of the circuitry that the CPU test is supposed to check will be checked by the shutdown handling instead, and you will get no POST indication if a critical failure occurs.

Manufacturing Loop Jumper

The phrase *Check for Manufacturing Jumper* refers to one on the motherboard that makes the POST run in a continuous loop, so you can burn in a system, or use repetitive cycling to monitor a failing area with an oscilloscope or logic analyzer. It usually forces a reset, so the POST has to start from the beginning every time. Compaq used the shorted jumper to make the POST to jump to another ROM at E000 just after power-on, which could have diagnostic code in it. IBM and NCR used a germanium or silicon diode to short together the keyboard connector pins 1 (cathode, bar) and 2 (5-pin DIN) or 1 (anode, arrow) and 5 (6-pin mini-DIN), so the POST checks the keyboard controller to see if the jumper is there.

WHAT IS A POST DIAGNOSTIC CARD?

Note: Under no circumstances shall the publisher, author, any manufacturer of POST diagnostic cards, or their agents or distributors be held liable in any way for damages, including lost profits, lost savings, or other incidental or consequential damages arising out of the use of, or inability to use, any product designed to make POST diagnostic codes visible on your system.

A POST card is an operating system independent expansion card for use with any x86-based computer with a suitable expansion bus (although older cards are usually 8-bit, XT class machines do not generally issue POST codes). There may be conversion products for purely Micro Channel and PCI systems, depending on the manufacturer, but PCI versions suffer from some shortcomings. When you install a PCI and an ISA POST card into a system, the ISA card will report all the faults. The PCI one, on the other hand, reports only some or the later codes, that is, those issued after ISA initialisation. In some cases, they do not provide any codes at all (the culprits are actually the bridge circuits in the various motherboard chipsets rather than the design of the card, and is one reason why some motherboard vendors suggest that their boards be tested with an ISA video card, especially with corrupted CMOS settings or when flashing the BIOS).

In bus master mode, PCI slots can be addressed individually by North Bridge logic, so the North Bridge acts like a filter, allowing information only to certain PCI slots. The same applies to IRQ lines in PCI designs; the IRQ lines are not bussed or are only bussed between selected PCI slots. The backwards compatibility to the legacy PC architecture is established through a routing table (a lookup table) via the BIOS, chipset and PCI Steering logic. This table differs for each particular design and/or chipset. According to the PCI standard, the 4 PCI interrupts can hardwired in various ways. The preferred/ideal way would be to have the 4 Interrupt lines from every PCI slot go to an Interrupt router.

Also allowed by the PCI bus specification is a more low end approach where the IRQ lines are bussed (all INTA# pins on every slot are bussed to an INTA# input on the Interrupt Router etc.). The resulting 4 lines would be routed to the classical IRQs #9 through #12.

Unfortunately, the ISA slots, together with the BIOS that generates the POST codes, connect to the South Bridge in the chipset, so a defective motherboard or system with corrupted CMOS settings may be unable to send information to the PCI bus anyway. In addition, with no ISA bus there is no universal diagnostic port. For example, some high end motherboards by Epox and ASUS come with built in POST code displays or have a proprietary ports which need a special display unit.

Obtaining Information About Your Computer

At least the BIOS ROM's manufacturer and firmware revision number should be known, so you can check the codes in the following pages (see the front of the book for BIOS IDs). The manufacturing port or POST port should also be known.

ACER

Based on Award BIOS 3.03, but not exactly the same. Port 80h.

Code	Meaning	Code	Meaning
04	Start	4C	Shutdown 3
08	Shutdown	50	Shutdown 2
0C	Test BIOS ROM checksum	54	Shutdown 7
10	Test CMOS shutdown byte	55	Shutdown 6
14	Test DMA controller	5C	Test keyboard and auxiliarv I/O
18	Initialise system timer	60	Set up BIOS interrupt routines
1C	Test memory refresh	64	Test real time clock
1E	Determine memory type	68	Test diskette
20	Test 128K memory	6C	Test hard disk
24	Test 8042 keyboard controller	70	Test parallel port
28	Test CPU descriptor instruction	74	Test serial port
2C	Set up and test 8259 controller	78	Set time of day
30	Set up memory interrupts	7C	Scan for and invoke option ROMs
34	Set interrupt vectors and routines	80	Determine presence of math copro
38	Test CMOS RAM	84	initialize keyboard
3C	Determine memory size	88	Initialise svstem 1
XX	Shut down 8 (system halt C0h + checkpoint)	8C	Initialize system 2
40	Shutdown 1	90	Invoke INT 19 to boot OS
44	Initialise Video BIOS ROM	94	Shutdown 5
45	Set up and test RAM BIOS	98	Shutdown A
46	Test cache and controller	9C	Shutdown B
48	Test memory		

ALR

See also *Phoenix*.

Code	Meaning
01	80[3,4]86 register test in progress
02	Real-time clock write/read failure
03	ROM BIOS Checksum failure
04	Programmable Internal Timer Failure (or no video card)
05	DMA initialization failure
06	DMA page register write/read failure
08	RAM refresh verification failure
09	1st 64-KB RAM test in progress
0A	1st 64-KB RAM chip or data line multi-bit
0B	1st 64-KB RAM odd/even logic failure
0C	Address line failure 1st 64 KB RAM
0D	Parity failure 1st 64-KB RAM
10	Bit 01st 64-KB RAM failure
11	Bit 11st 64-KB RAM failure
12	Bit 2 1st 64-KB RAM failure
13	Bit 3 1st 64-KB RAM failure
14	Bit 4 1st 64-KB RAM failure
15	Bit 5 1st 64-KB RAM failure
16	Bit 6 1st 64-KB RAM failure
17	Bit 7 1st 64-KB RAM failure
18	Bit 8 1st 64-KB RAM failure
19	Bit 9 1st 64-KB RAM failure
1A	Bit A 1st 64-KB RAM failure
1B	Bit B 1st 64-KB RAM failure
1C	Bit C 1st 64-KB RAM failure
1D	Bit D 1st 64-KB RAM failure
1E	Bit E 1st 64-KB RAM failure
1F	Bit F 1st 64-KB RAM failure
20	Slave DMA register failure
21	Master DMA register failure
22	Master interrupt mask register failure
23	Slave interrupt mask register failure
25	Interrupt vector loading in progress
27	Keyboard controller test failure
28	Real-time clock power failure and checksum calculation in progress
29	Real-time clock configuration validation in progress
2B	Screen memory test failure
2C	Screen initialization failure
2D	Screen retrace test failure
2E	Search for video ROM in Progress
30	Screen believed operational-screen believed running with video ROM
31	Mono display believed operable
32	Colour display (40 column) believed operable
33	Colour display (80 column) believed operable

AMBRA

See *Phoenix*.

AMI

Not all tests are performed by all AMI BIOSes. These refer to 2 Feb 91 BIOS.

POST Procedures

Procedure	Explanation
NMI Disable	NMI interrupt line to the CPU is disabled by setting bit 7 I/O port 70h (CMOS).
Power On Delay	Once keyboard controller gets power, it sets hard and soft reset bits. Check keyboard controller or clock generator.
Initialise Chipsets	Check the BIOS, CLOCK or chipsets.
Reset Determination	The BIOS reads bits in the keyboard controller to see if hard or soft reset is required (soft will not test mem above 64K). Failure could be BIOS or keyboard controller.
ROM BIOS Checksum	BIOS performs a checksum on itself and adds a preset factory value to make it equal 00. Failure is due to the BIOS chips.
Keyboard Test	A command is sent to the 8042 (keyboard controller) which performs a test and sets a buffer space for commands. After the buffer is defined the BIOS sends a command byte, writes data to the buffer, checks the high order bits (Pin 23) of the internal keyboard controller and issues a No Operation (NOP) command.
CMOS	Shutdown byte in CMOS RAM offset 0F is tested, the BIOS checksum calculated and diagnostic byte (0E) updated before the CMOS RAM area is initialised and updated for date and time. Check RTC/CMOS chip or battery.
8237/8259 Disable	DMA and PIC are disabled before POST proceeds. Check the 8237 or 8259 chips.
Video Disable	Vvideo controller is disabled and Port B initialised. Check adapter if problems here.
Chipset Init/ Memory Detect	Memory addressed in 64K blocks; failure would be in chipset. If all memory is not seen, failure could be in a chip in the block after the last one seen.
PIT test	The timing functions of the 8254 interrupt timer are tested. The PIT or RTC chips normally cause problems here.
Memory Refresh	PIT's ability to refresh memory tested (if an XT, DMA controller #1 handles this). Failure is normally the PIT (8254) in ATs or the 8237 (DMA #1) in XTs.
Address Lines	Test the address lines to the first 64K of RAM. An address line failure.
Base 64K	Data patterns are written to the first 64K, unless there is a bad RAM chip in which case you will get a failure.
Chipset Init	The PIT, PIC and DMA controllers are enabled.
Set Interrupt Table	Interrupt vector table used by PIC is installed in low memory, the first 2K.
8042 check	The BIOS reads the buffer area of the keyboard controller I/O port 60.
Video Tests	Ttype of video adapter is checked for then tests are done on adapter and monitor.
BIOS Data Area	The vector table is checked for proper operation and video memory verified before protected mode tests are entered into. This is done so that any errors found are displayed on the monitor.
Protected Mode Tests	Do reads and writes to mem below 1 Mb. Failures are bad RAM, 8042 or data line.
DMA Chips	The DMA registers are tested using a data pattern.
Final Initialisation	These differ with each version. Typically, floppy and hard drives are tested and initialised, and a check made for serial and parallel devices. Information is compared against the CMOS, and you will see the results of any failures on the monitor.
Boot	The BIOS hands over control to the Int 19 bootloader; this is where you would see error messages such as non-system disk.

AMI BIOS 2.2x

Code	Meaning	Code	Meaning
00	Flag test	5A	Verify LAR instruction
03	Register test	5D	Verify VERR instruction
06	System hardware initialisation	60	Address line 20 test
09	BIOS ROM checksum	63	Unexpected exception test
0C	Page register test	66	Start third protected mode test
0F	8254 timer test	69	Address line test
12	Memory refresh initialisation	6C	System memory test
15	8237 DMA controller test	6F	Shadow memory test
18	8237 DMA initialisation	72	Extended memory test
1B	8259 interrupt controller initialisation	75	Verify memory configuration
1E	8259 interrupt controller test	78	Display configuration error messages
21	Memory refresh test	7B	Copy system BIOS to shadow memory
24	Base 64K address test	7E	8254 clock test
27	Base 64K memory test	81	MC 146818 real time clock test
2A	8742 keyboard self test	84	Keyboard test
2D	MC 146818 CMOS test	87	Determine keyboard type
30	Start first protected mode test	8A	Stuck key test
33	Memory sizing test	8D	Initialise hardware interrupt vector
36	First protected mode test	90	Math coprocessor test
39	First protected mode test failed	93	Determine COM ports available
3C	CPU speed calculation	96	Determine LPT ports available
3F	Read 8742 hardware switches	99	Initialise BIOS data area
42	Initialise interrupt vector area	9C	Fixed/Floppy controller test
45	Verify CMOS configuration	9F	Floppy disk test
48	Test and initialise video system	A2	Fixed disk test
4B	Unexpected interrupt test	A5	External ROM scan
4E	Start second protected mode test	A8	System key lock test
51	Verify LDT instruction	AE	F1 error message test
54	Verify TR instruction	AF	System boot initialisation
57	Verify LSL instruction	B1	Interrupt 19 boot loader

AMI Colour BIOS

Code	Meaning	Code	Meaning
00	Control to Int 19 boot loader	4C	Clear extended memory locations
01	CPU flag test	4D	Update CMOS memory size
02	Power-on delay	4E	Base RAM size displayed
03	Chipset initialization	4F	Memory Read/Write test on 640K
04	Soft/hard reset	50	Update CMOS on RAM size
05	ROM enable	51	Extended memory tested
06	BIOS ROM checksum	52	Re-size extended memory
07	8042 keyboard controller tested	53	Return CPU to real mode
08	8042 keyboard controller tested	54	Restore CPU registers
09	8042 keyboard controller tested	55	A-20 gate disabled
0A	8042 keyboard controller tested	56	BIOS vector recheck
0B	8042 keyboard controller tested	57	BIOS vector check complete

Code	Meaning	Code	Meaning
0C	8042 keyboard controller tested	58	Clear BIOS display setup message
0D	8042 keyboard controller tested	59	DMA, PIT tested
0E	CMOS checksum tested	60	DMA page register tested
0F	CMOS initialization	61	DMA #1 tested
10	CMOS/RTC status OK	62	DMA #2 tested
11	DMA/PIC disable	63	BIOS data area check
12	DMA/PIC initialization	64	BIOS data area checked
13	Chipset/memory initialization	65	Initialize DMA chips
14	8254 PIT timer tested	66	8259 PIC initialization
15	8254 PIT channel 2 timer tested	67	Keyboard tested
16	8254 PIT channel 1 timer tested	80	Keyboard reset
17	8254 PIT channel 0 timer tested	81	Stuck key and batch test
18	Memory refresh test (PIC)	82	8042 keyboard controller tested
19	Memory refresh test (PIC)	83	Lock key check
1A	Check 15-microsecond refresh (PIT)	84	Compare memory size with CMOS
1B	Check 30-microsecond refresh (PIT)	85	Password/soft error check
20	Base 64K memory tested	86	XCMOS/CMOS equipment check
21	Base 64K memory parity tested	87	CMOS setup entered
22	Memory Read/Write	88	Reinitialize chipset
23	BIOS vector table initialization	89	Display power-on message
24	BIOS vector table initialization	8A	Display wait and mouse check
25	Turbo check 8042 keyboard controller	8B	Shadow any option ROMs
26	Global data table for kb controller; turbo	8C	Initialize XCMOS settings
27	Video mode tested	8D	Reset hard/floppy drives
28	Monochrome tested	8E	Floppy compare to CMOS
29	Color (CGA) tested	8F	Floppy disk controller initialization
2A	Parity-enable tested	90	Hard disk compare to CMOS
2B	Optional system ROMs check start	91	Hard disk controller initialization
2C	Video ROM check	92	BIOS data table check
2D	Reinitialize main chipset	93	BIOS data check hat halfway
2E	Video memory tested	94	Set memory size
2F	Video memory tested	95	Verify display memory
30	Video adapter tested	96	Clear all Interrupts
31	Alternate video adapter tested	97	Optional ROMs check
32	Alternate video adapter tested	98	Clear all Interrupts
33	Video mode tested	99	Setup timer data/RS232 base
34	Video mode tested	9A	RS232 test; Locate and test serial ports
35	Initialize BIOS ROM data area	9B	Clear all Interrupts
36	Power-on message display	9C	NPU test
37	Power-on message display	9D	Clear all Interrupts
38	Read cursor position	9E	Extended keyboard check
39	Display cursor reference	9F	Set numlock
3A	Display BIOS setup message	A0	Keyboard reset
40	Start protected mode tested	A1	Cache memory test
41	Build mode entry	A2	Display any soft errors
42	CPU enters protected mode	A3	Set typematic rate
43	Protected mode Interrupt enable	A4	Set memory wait states
44	Check descriptor tables	A5	Clear screen
45	Check memory size	A6	Enable parity/NMI

Code	Meaning	Code	Meaning
46	Memory Read/Write tested	A7	Clear all Interrupts
47	Base 640K memory tested	A8	Control to ROM at E0000
48	Check 640K memory size	A9	Clear all Interrupts
49	Check extended memory size	AA	Display configuration
4A	Verify CMOS extended memory	00	Interrupt 19 boot loader
4B	Check for soft/hard reset		

AMI Ez-Flex BIOS

Code	Meaning	Code	Meaning
01	NMI disabled; Start CPU flag test	4C	Memory above 1MB cleared for soft reset
02	Power-on delay	4D	Update CMOS memory size
03	Chipset initialization	4E	Base RAM size displayed
04	Check keyboard for soft/hard reset	4F	Memory Read/Write test on 640K
05	ROM enable	50	Update CMOS on RAM size
06	BIOS ROM checksum	51	Extended memory tested
07	8042 keyboard controller tested	52	System is prepared for real mode
08	8042 keyboard controller tested	53	Return CPU to real mode
09	8042 keyboard controller tested	54	Restore CPU registers
0A	8042 keyboard controller tested	55	A-20 gate disabled
0B	8042 protected mode tested	56	BIOS data area rechecked
0C	8042 keyboard controller tested	57	BIOS data area recheckcomplete
0D	CMOS RAM shutdown register tested	58	Display setup message
0E	CMOS checksum tested	59	DMA register page tested
0F	CMOS initialization	60	DMA page register tested
10	CMOS/RTC status OK	61	DMA #1 tested
11	DMA/PIC disable	62	DMA #2 tested
12	Video display disabled	63	BIOS data area check
13	Chipset/memory initialization	64	BIOS data area checked
14	8254 PIT timer tested	65	Initialize DMA chips
15	8254 PIT channel 2 timer tested	66	8259 PIC initialization
16	8254 PIT channel 1 timer tested	67	Keyboard tested
17	8254 PIT channel 0 timer tested	80	Keyboard reset
18	Memory refresh test (PIT)	81	Stuck key and batch test
19	Memory refresh test (PIT)	82	8042 keyboard controller tested
1A	Check 15-microsecond refresh (PIT)	83	Lock key check
1B	Base 64K memory tested	84	Compare memory size with CMOS
20	Address lines tested	85	Password/soft error check
21	Base 64K memory parity tested	86	CMOS equipment check
22	Memory Read/Write	87	CMOS setup entered if selected
23	Perform setups before init vector table	88	Main chipset reinitialized after CMOS setup
24	BIOS vector table initialization	89	Display power-on message
25	8042 keyboard controller tested	8A	Display wait and mouse check
26	Global data table for kb controller	8B	Shadow any option ROMs
27	Perform setups for vector table initialization	8C	Initialize CMOS settings
28	Monochrome tested	8D	Reset hard/floppy drives
29	Color (CGA) tested	8E	Floppy compare to CMOS
2A	Parity-enable tested	8F	Floppy disk controller initialization

Code	Meaning	Code	Meaning
2B	Optional system ROMs check start	90	Hard disk compare to CMOS
2C	Video ROM check	91	Hard disk controller initialization
2D	Determine if EGA/VGA is installed	92	BIOS data table check
2E	Video memory tested	93	BIOS data table check complete
2F	Video memory tested	94	Set memory size
30	Video adapter tested	95	Verify display memory
31	Alternate video adapter tested	96	Clear all Interrupts
32	Alternate video adapter tested	97	Optional ROMs checked
33	Video mode tested	98	Clear all Interrupts
34	Video mode tested	99	Timer data setup
35	Initialize BIOS ROM data area	9A	RS232 test; Locate and test serial ports
36	Power on display cursor set	9B	Clear all Interrupts
37	Power-on message display	9C	Math coprocessor checked
38	Read cursor position	9D	Clear all Interrupts
39	Display cursor reference	9E	Extended keyboard check
3A	Display BIOS setup message	9F	Set numlock
40	Protected mode tested	A0	Keyboard reset
41	Build descriptor tables	A1	Cache memory test
42	CPU enters protected mode	A2	Display any soft errors
43	Protected mode Interrupt enable	A3	Set typematic rate
44	Check descriptor tables	A4	Set memory wait states
45	Check memory size	A5	Clear screen
46	Memory Read/Write tested	A6	Enable parity/NMI
47	Base 640K memory tested	A7	Clear all Interrupts
48	Memory below 1MB checked for soft reset	A8	Control to ROM at E0000
49	Memory above 1MB checked for soft reset	A9	Clear all Interrupts
4A	ROM BIOS data area checked	AA	Display configuration
4B	Memory below 1MB cleared for soft reset	00	Interrupt 19 boot loader

AMI Old BIOS; 08/15/88-04/08/90

Code	Meaning	Code	Meaning
01	NMI disabled & 286 reg. test about to start	3E	About to go to real mode (shutdown)
02	286 register test over	3F	Shutdown successful, entered in real mode
03	ROM checksum OK	40	About to disable gate A-20 address line
04	8259 initialization OK	41	Gate A-20 line disabled successfully
05	CMOS pending interrupt disabled	42	About to start DMA controller test
06	Video disabled & system timer counting OK	4E	Address line test OK
07	CH-2 of 8253 test OK	4F	Processor in real mode after shutdown
08	CH-2 delta count test OK	50	DMA page register test OK
09	CH-1 delta count test OK	51	DMA unit-1 base register test about to start
0A	CH-0 delta count test OK	52	DMA unit-1 channel OK; to begin CH-2
0B	Parity status cleared	53	DMA CH-2 base register test OK
0C	Refresh & system timer OK	54	About to test f/f latch for unit-1
0D	Refresh link toggling OK	55	f/f latch test both unit OK
0E	Refresh period ON/OFF 50% OK	56	DMA unit 1 & 2 programmed OK
10	Confirmed refresh ON, starting 64K memory	57	8259 initialization over
11	Address line test OK	58	8259 mask register check OK

Code	Meaning	Code	Meaning
12	64K base memory test OK	59	Master 8259 mask register OK; start slave
13	Interrupt vectors initialized	5A	To check timer, keyboard interrupt level
14	8042 keyboard controller test OK	5B	Timer interrupt OK
15	CMOS read/write test OK	5C	About to test keyboard interrupt
16	CMOS checksum/battery check OK	5D	Timer/keyboard interrupt not proper level
17	Monochrome mode set OK	5E	8259 interrupt controller error
18	Colour mode set OK	5F	8259 interrupt controller test OK
19	About to look for optional video ROM	70	Start of keyboard test
1A	Optional video ROM control OK	71	Keyboard BAT test OK
1B	Display memory read/write test OK	72	Keyboard test OK
1C	Alt display memory read/write test OK	73	Keyboard global data initialization OK
1D	Video retrace check OK	74	Floppy setup about to start
1E	Global equipment byte set for video OK	75	Floppy setup OK
1F	Mode set call for Mono/Colour OK	76	Hard disk setup about to start
20	Video test OK	77	Hard disk setup OK
21	Video display OK	79	About to initialize timer data area
22	Power on message display OK	7A	Verify CMOS battery power
30	Virtual mode memory test about to begin	7B	CMOS battery verification done
31	Virtual mode memory test started	7D	About to analyze diag results for memory
32	Processor in virtual mode	7E	CMOS memory size update OK
33	Memory address line test in progress	7F	About to check optional ROM C000:0
34	Memory address line test in progress	80	Keyboard sensed to enable setup
35	Memory below 1MB calculated	81	Optional ROM control OK
36	Memory size computation OK	82	Printer global data initialization OK
37	Memory test in progress	83	RS-232 global data initialization OK
38	Memory initialization over below 1MB	84	80287 check/test OK
39	Memory initialization over above 1MB	85	About to display soft error message
3A	Display memory size	86	Give control to system ROM E000:0
3B	About to start below 1MB memory test	87	System ROM E000:0 check over
3C	Memory test below 1MB OK	00	Control given to Int-19; boot loader
3D	Memory test above 1MB OK		

AMI Plus BIOS

Code	Meaning	Code	Meaning
01	NMI disabled (Bit 7 of I/O port 70h)	3E	About to go to real mode (shutdown)
02	286 register test over	3F	Shutdown successful, entered in real mode
03	ROM checksum OK	40	About to disable gate A-20 address line
04	8259 initialization OK	41	Gate A-20 line disabled successfully
05	CMOS pending interrupt disabled	42	About to start DMA controller test
06	System timer (PIT) counting OK	4E	Address line test OK
07	Channel 0 of 8259 PIC test OK	4F	Processor in real mode after shutdown
08	DMA CH-2 delta count test OK	50	DMA page register test OK
09	DMA CH-1 delta count test OK	51	DMA unit-1 base register test about to start
0A	DMA CH-0 delta count test OK	52	DMA 1 channel OK; about to begin CH-2
0B	Parity status cleared (DMA/PIT)	53	DMA CH-2 base register test OK
0C	Refresh & system timer OK (DMA/PIT)	54	About to test both units OK
0D	Refresh link toggling OK (DMA/PIT)	55	f/f latch test both unit OK

Code	Meaning	Code	Meaning
0E	Refresh period ON/OFF 50% OK	56	DMA unit 1 & 2 programmed OK
10	About to start 64K memory	57	8259 initialization over
11	Address line test OK	58	8259 mask register check OK
12	64K base memory test OK	59	Master 8259 mask register OK
13	Interrupt vectors initialized	5A	Check timer, keyboard interrupt level
14	8042 keyboard controller test OK	5B	Timer interrupt OK
15	CMOS read/write test OK	5C	About to test keyboard interrupt
16	CMOS checksum/battery check OK	5D	Timer/keyboard interrupt not in proper level
17	Monochrome mode set OK	5E	8259 interrupt controller error
18	Colour mode set OK	5F	8259 interrupt controller test OK
19	Video ROM Search	70	Start of keyboard test
1A	Optional video ROM control OK	71	Keyboard test OK
1B	Display memory read/write test OK	72	Keyboard test OK
1C	Alternate display memory OK	73	Keyboard global data initialization OK
1D	Video retrace check OK	74	Floppy setup about to start
1E	Global equipment byte set for video OK	75	Floppy setup OK
1F	Mode set call for Mono/Colour OK	76	Hard disk setup about to start
20	Video test OK	77	Hard disk setup OK
21	Video display OK	79	About to initialize timer data area
22	Power on message display OK	7A	Verify CMOS battery power
30	Virtual mode memory test about to begin	7B	CMOS battery verification done
31	Virtual mode memory test started	7D	Analyze diag test results for memory
32	Processor in virtual mode	7E	CMOS memory size update OK
33	Memory address line test in progress	7F	About to check optional ROM C000:0
34	Memory address line test in progress	80	Keyboard sensed to enable setup
35	Memory below 1MB calculated	81	Optional ROM control OK
36	Memory size computation OK	82	Printer global data initialization OK
37	Memory test in progress	83	RS-232 global data initialization OK
38	Memory initialization over below 1MB	84	80287 check/test OK
39	Memory initialization over above 1MB	85	About to display soft error message
3A	Display memory size	86	Give control to system ROM E000:0
3B	About to start below 1MB memory test	87	System ROM E000:0 check over
3C	Memory test below 1MB OK	00	Control given to Int-19; boot loader
3D	Memory test above 1MB OK		

AMI BIOS 04/09/90-02/01/91

Code	Meaning
01	NMI disabled and 286 register test about to start.
02	286 register test passed.
03	ROM BIOS checksum (32K at F800:0) passed.
04	Keyboard controller test with and without mouse passed.
05	Chipset initialization over; DMA and Interrupt controller disabled.
06	Video disabled and system timer test begin.
07	CH-2 of 8254 initialization half way.
08	CH-2 of timer initialization over.
09	CH-1 of timer initialization over.
0A	CH-0 of timer initialization over.

Code	Meaning
0B	Refresh started.
0C	System timer started.
0D	Refresh link toggling passed.
10	Refresh on and about to start 64K base memory test.
11	Address line test passed.
12	64K base memory test passed.
15	Interrupt vectors initialized.
17	Monochrome mode set.
18	Colour mode set.
19	About to look for optional video ROM at C000 and give control to ROM if present.
1A	Return from optional video ROM.
1B	Shadow RAM enable/disable completed.
1C	Display memory read/write test for main display as in CMOS setup program over.
1D	Display memory read/write test for alternate display complete if main display memory test returns error.
1E	Global equipment byte set for proper display type.
1F	Video mode set call for mono/colour begins.
20	Video mode set completed.
21	ROM type 27256 verified.
23	Power on message displayed.
30	Virtual mode memory test about to begin.
31	Virtual mode memory test started.
32	Processor executing in virtual mode.
33	Memory address line test in progress.
34	Memory address line test in progress.
35	Memory below 1MB calculated.
36	Memory above 1MB calculated.
37	Memory test about to start.
38	Memory below 1MB initialized.
39	Memory above 1MB initialized.
3A	Memory size display initiated-updated when BIOS goes through memory test.
3B	About to start below 1MB memory test.
3C	Memory test below 1MB completed; about to start above 1MB test.
3D	Memory test above 1MB completed.
3E	About to go to real mode (shutdown).
3F	Shutdown successful and processor in real mode.
40	Cache memory on and about to disable A20 address line.
41	A20 address line disable successful.
42	486 internal cache turned on.
43	About to start DMA controller test.
50	DMA page register test complete.
51	DMA unit-1 base register test about to start.
52	DMA unit-1 base register test complete.
53	DMA unit-2 base register test complete.
54	About to check F/F latch for unit-1 and unit-2.
55	F/F latch for both units checked.
56	DMA unit 1 and 2 programming over; about to initialize 8259 interrupt controller.
57	8259 initialization over.
70	About to start keyboard test.
71	Keyboard controller BAT test over.

Code	Meaning
72	Keyboard interface test over; mouse interface test started.
73	Global data initialization for keyboard/mouse over.
74	Display 'SETUP' prompt and about to start floppy setup.
75	Floppy setup over.
76	Hard disk setup about to start.
77	Hard disk setup over.
79	About to initialize timer data area.
7A	Timer data initialized and about to verify CMOS battery power.
7B	CMOS battery verification over.
7D	About to analyze POST results.
7E	CMOS memory size updated.
7F	Look for <DEL> key and get into CMOS setup if found.
80	About to give control to optional ROM in segment C800 to DE00.
81	Optional ROM control over.
82	Check for printer ports and put the addresses in global data area.
83	Check for RS232 ports and put the addresses in global data area.
84	Coprocessor detection over.
85	About to display soft error messages.
86	About to give control to system ROM at segment E000.
00	System ROM control at E000 over now give control to Int 19h boot loader.

AMI New BIOS; 02/02/91-12/12/91

Code	Meaning
01	Processor register test about to start and NMI to be disabled.
02	NMI is Disabled. Power on delay starting.
03	Power on delay complete. Any initialization before keyboard BAT is in progress.
04	Init before keyboard BAT complete. Reading keyboard SYS bit to check soft reset/ power-on.
05	Soft reset/ power-on determined. Going to enable ROM. i. e. disable shadow RAM/Cache.
06	ROM enabled. Calculating BIOS checksum, waiting for KB controller input buffer to be free.
07	ROM Checksum passed. KB controller I/B free. Issuing BAT comd to kboard controller.
08	BAT command to keyboard controller issued. Going to verify BAT command.
09	Keyboard controller BAT result verified. Keyboard command byte to be written next.
0A	Keyboard command byte code issued. Going to write command byte data.
0B	Keyboard controller command byte written. Issuing Pin-23 & 24 blocking/unblocking command
0C	Pin 23 & 24 of keyboard controller is blocked/unblocked. NOP command to be issued next.
0D	NOP command processing done. CMOS shutdown register test to be done next.
0E	CMOS shutdown register R/W test passed. Calculating CMOS checksum, update DIAG byte.
0F	CMOS checksum calc done DIAG byte written. CMOS init. begins (If INIT CMOS IN EVERY BOOT is set).
10	CMOS initialization done (if any). CMOS status register about to init for Date and Time.
11	CMOS Status register initialised. Going to disable DMA and Interrupt controllers.
12	DMA Controller #1 & #2, interrupt controller #1 & #2 disabled. Disable Video display and init port-B.
13	Video display disabled and port-B initialized. Chipset init/auto mem detection about to begin.
14	Chipset initialization/auto memory detection over. 8254 timer test about to start.
15	CH-2 timer test halfway. 8254 CH-2 timer test to be complete.
16	Ch-2 timer test over. 8254 CH-1 timer test to be complete.
17	CH-1 timer test over. 8254 CH-0 timer test to be complete.
18	CH-0 timer test over. About to start memory refresh.

Code	Meaning
19	Memory Refresh started. Memory Refresh test to be done next.
1A	Memory Refresh line is toggling. Going to check 15 microsecond ON/OFF time.
1B	Memory Refresh period 30 microsec test complete. Base 64K memory test about to start.
20	Base 64k memory test started. Address line test to be done next.
21	Address line test passed. Going to do toggle parity.
22	Toggle parity over. Going for sequential data R/W test.
23	Base 64k sequential data R/W test passed. Setup before Interrupt vector init about to start.
24	Setup before vector init complete. Interrupt vector initialization about to begin.
25	Interrupt vector init done. Going to read I/O port of 8042 for turbo switch (if any).
26	I/O port of 8042 is read. Going to initialize global data for turbo switch.
27	Global data init is over. Any initialization after interrupt vector to be done next.
28	Initialization after interrupt vector is complete. Going for mono mode setting.
29	Monochrome mode setting is done. Going for Colour mode setting.
2A	Colour mode setting done. About to go for toggle parity before optional ROM test.
2B	Toggle parity over. About to give control for setup before optional video ROM.
2C	Processing before video ROM control is done. Look for optional video ROM and give control.
2D	Optional video ROM control done. Give control to do processing after video ROM returns control.
2E	Return from processing after video ROM control. If EGA/VGA not found do display mem R/W test.
2F	EGA/VGA not found. Display memory R/W test about to begin.
30	Display mem R/W test passed. About to look for retrace checking.
31	Display mem R/W test/ retrace check failed. About to do alternate Display memory R/W test.
32	Alternate Display memory R/W test passed. Looking for alternate display retrace checking.
33	Video display check over. Verification of display with switches and card to begin.
34	Verification of display adapter done. Display mode to be set next.
35	Display mode set complete. BIOS ROM data area about to be checked.
36	BIOS ROM data area check over. Going to set cursor for power on message.
37	Cursor setting for power on message id complete. Going to display power on message.
38	Power on message display complete. Going to read new cursor position.
39	New cursor position read and saved. Going to display the reference string.
3A	Reference string display is over. Going to display the Hit <Esc> message.
3B	Hit <Esc> message displayed. Virtual mode memory test about to start.
40	Preparation for virtual mode test started. Going to verify from video memory.
41	Returned after verifying from display memory. Going to prepare descriptor tables.
42	Descriptor tables prepared. Going to enter in virtual mode for memory test.
43	Entered in the virtual mode. Going to enable interrupts for diagnostics mode.
44	Interrupts enabled (if diag switch is on). Initialize data to check memory wrap around at 0:0.
45	Data initialized. Check for memory wrap around at 0:0and finding the total system memory size.
46	Memory wrap around test done. Memory size calculation over. Writing patterns to test memory.
47	Pattern to be tested written in extended memory. Going to write patterns in base 640k.
48	Patterns written in base memory. Going to find out amount of memory below 1Mb.
49	Amount of memory below 1Mb found and verified. Find out amount of memory above 1M memory.
4A	Amount of memory above 1Mb found and verified. Going for BIOS ROM data area check.
4B	BIOS ROM data area check over. Going to check <Esc> and clear mem below 1Mb for soft reset.
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.
4E	Memory test started. (NO SOFT RESET). About to display first 64k memory test.
4F	Memory size display started, for update during memory test. Sequential and random memory test.
50	Memory test below 1Mb complete. Going to adjust memory size for relocation/ shadow.
51	Memory size adjusted due to relocation/shadow. Memory test above 1Mb to follow.

Code	Meaning
52	Memory test above 1Mb complete. Going to prepare to go back to real mode.
53	CPU registers are saved including memory size. Going to enter in real mode.
54	Shutdown successful. CPU in real mode. Restore registers saved during preparation for shutdown.
55	Registers restored. Going to disable gate A20 address line.
56	A20 address line disable successful. BIOS ROM data area about to be checked.
57	BIOS ROM data area check halfway. BIOS ROM data area check to be complete.
58	BIOS ROM data area check over. Going to clear Hit <Esc>message.
59	Hit <Esc> message cleared. WAIT. . . message displayed.Sstart DMA and interrupt controller test.
60	DMA page register test passed. About to verify from display memory.
61	Display memory verification over. About to go for DMA #1 base register test.
62	DMA #1 base register test passed. About to go for DMA #2 base register test.
63	DMA #2 base register test passed. About to go for BIOS ROM data area check.
64	BIOS ROM data area check halfway. BIOS ROM data area check to be complete.
65	BIOS ROM data area check over. About to program DMA unit 1 and 2.
66	DMA unit 1 and 2 programming over. About to initialize 8259 interrupt controller
67	8259 initialization over. About to start keyboard test.
80	Keyboard test started. Clear output buffer, check for stuck key. About to issue keyboard reset
81	Keyboard reset error/stuck key found. About to issue keyboard controller i/f test command.
82	Keyboard controller interface test over. About to write command byte and init circular buffer.
83	Command byte written Global data init done. About to check for lock-key.
84	Lock-key checking over. About to check for memory size mismatch with CMOS.
85	Memory size check done. About to display soft error; check for password or bypass setup.
86	Password checked. About to do programming before setup.
87	Programming before setup complete. Going to CMOS setup program.
88	Returned from CMOS setup and screen cleared. About to do programming after setup.
89	Programming after setup complete. Going to display power on screen message.
8A	First screen message displayed. About to display WAIT. . . message.
8B	WAIT. . . message displayed. About to do Main and Video BIOS shadow.
8C	Main/Video BIOS shadow successful. Setup options programming after CMOS setup about to start.
8D	Setup options are programmed, mouse check and init to be done next
8E	Mouse check and initialisation complete. Going for hard disk floppy reset.
8F	Floppy check returns that floppy is to be initialized. Floppy setup to follow.
90	Floppy setup is over. Test for hard disk presence to be done.
91	Hard disk presence test over. Hard disk setup to follow.
92	Hard disk setup complete. About to go for BIOS ROM data area check.
93	BIOS ROM data area check halfway. BIOS ROM data area check to be complete.
94	BIOS ROM data area check over. Going to set base and extended memory size.
95	Memory size adjusted due to mouse support hdisk type 47. Going to verify from display memory.
96	Returned after verifying from display memory. Do any init before C800 optional ROM control.
97	Any init before C800 optional ROM control is over. Optional ROM check and control next.
98	Optional ROM control done. Give control to do required processing after ROM returns control.
99	Iinitialization 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 Copro 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 co-pro test complete. Going to check extd keyboard; ID and num-lock.
9F	Extd keyboard check done ID flag set. num-lock on/off. Keyboard ID command to be issued.
A0	Keyboard ID command issued. Keyboard ID flag to be reset.

Code	Meaning
A1	Keyboard ID flag reset. Cache memory test to follow.
A2	Cache memory test over. Going to display any soft errors.
A3	Soft error display complete. Going to set the keyboard typematic rate.
A4	Keyboard typematic rate set. Going to program memory wait states.
A5	Memory wait states programming over. Screen to be cleared next.
A6	Screen cleared. Going to enable parity and NMI.
A7	NMI and parity enabled. 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. Iinitialization required after E000 optional ROM control.
AA	Initialization after E000 optional ROM control is over. Going to display system configuration.
00	System configuration is displayed. Going to give control to INT 19h boot loader.

AMI New BIOS; 06/06/92-08/08/93

Code	Meaning
01	Processor register test about to start and NMI to be disabled.
02	NMI is Disabled. Power on delay starting.
03	Power on delay complete. Any initialization before keyboard BAT is in progress next.
04	Any init before keyboard BAT is complete. Reading keyboard SYS bit, to check soft reset/power on.
05	Soft reset/ power-on determined. Going to enable ROM; i.e. disable shadow RAM/Cache if any.
06	ROM enabled. Calculate ROM BIOS checksum - wait for 8042 keyboard controller input buffer to be free.
07	ROM BIOS checksum passed; KB controller input buffer free. Issue BAT command to keyboard controller.
08	BAT command to keyboard controller is issued. Going to verify the BAT command.
09	Keyboard controller BAT result verified. Keyboard command byte to be written next.
0A	Keyboard command byte code is issued. Going to write command byte data.
0B	Keyboard controller command byte written. Going to issue Pin 23/24 block/unblock command.
0C	Pin-23 & 24 of keyboard controller blocked/ unblocked. NOP command of controller to be issued next.
0D	NOP command processing is done. CMOS shutdown register test to be done next.
0E	CMOS shutdown register R/W test passed. Calculating CMOS checksum and update DIAG byte.
0F	CMOS checksum calc done; DIAG byte written. CMOS init begins (If INIT CMOS IN EVERY BOOT is set).
10	CMOS initialization done (if any). CMOS status register about to init for Date and Time.
11	CMOS Status register initialised. Going to disable DMA and Interrupt controllers.
12	DMA controller #1 & #2, interrupt controller #1 & #2 disabled. Disable Video display and init port-B.
13	Disable Video display and initialise port B. Chipset init/auto memory detection about to begin.
14	Chipset initialization/auto memory detection over. 8254 timer test about to start.
15	CH-2 timer test halfway. 8254 CH-2 timer test to be complete.
16	Ch-2 timer test over. 8254 CH-1 timer test to be complete.
17	CH-1 timer test over. 8254 CH-0 timer test to be complete.
18	CH-0 timer test over. About to start memory refresh.
19	Memory Refresh started. Memory Refresh test to be done next.
1A	Memory Refresh line is toggling. Going to check 15 microsecond ON/OFF time.
1B	Memory Refresh period 30 microsecond test complete. Base 64K memory test about to start.
20	Base 64k memory test started. Address line test to be done next.
21	Address line test passed. Going to do toggle parity.
22	Toggle parity over. Going for sequential data R/W test.
23	Base 64k sequential data R/W test passed. Any setup before Interrupt vector init about to start.
24	Setup required before vector initialization complete. Interrupt vector initialization about to begin.
25	Interrupt vector initialization done. Going to read I/O port of 8042 for turbo switch (if any).

Code	Meaning
26	I/O port of 8042 is read. Going to initialize global data for turbo switch.
27	Global data initialization is over. Any initialization after interrupt vector to be done next.
28	Initialization after interrupt vector is complete. Going for monochrome mode setting.
29	Monochrome mode setting is done. Going for Colour mode setting.
2A	Colour mode setting is done. About to go for toggle parity before optional ROM test.
2B	Toggle parity over. About to give control for any setup required before optional video ROM check.
2C	Processing before video ROM control done. Looking for optional video ROM and give control.
2D	Optional video ROM control done. Giving control for processing after video ROM returns control.
2E	Return from processing after video ROM control. If EGA/VGA not found test display mem R/W.
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 mem R/W test or retrace checking failed. About to do alternate Display memory R/W test.
32	Alternate Display memory R/W test passed. Looking for the alternate display retrace checking.
33	Video checking over. Verification of display type with switch setting and actual card to begin.
34	Verification of display adapter done. Display mode to be set next.
35	Display mode set complete. BIOS ROM data area about to be checked.
36	BIOS ROM data area check over. Going to set cursor for power on message.
37	Cursor setting for power on message complete. Going to display power on message.
38	Power on message display complete. Going to read new cursor position.
39	New cursor position read and saved. Going to display the reference string.
3A	Reference string display over. Going to display the Hit <ESC> message.
3B	Hit <ESC> message displayed. Virtual mode memory test about to start.
40	Preparation for virtual mode test started. Going to verify from video memory.
41	Returned after verifying from display memory. Going to prepare descriptor tables.
42	Descriptor tables prepared. Going to enter in virtual mode for memory test.
43	Entered in virtual mode. Going to enable interrupts for diagnostics mode.
44	Interrupts enabled (if diags switch on). Going to initialize data to check mem wrap around at 0:0.
45	Data initialized. Going to check for memory wrap around at 0:0 and finding total memory size.
46	Mem wrap around test done. Size calculation over. Going 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.
49	Amount of memory below 1Mb found and verified. Going to find amount of memory above 1Mb.
4A	Amount of memory above 1Mb found and verified. Going for BIOS ROM data area check.
4B	BIOS ROM data area check over. Going to check <Esc>, clear mem below 1 Mb for soft reset.
4C	Memory below 1Mb cleared. (SOFT RESET). Going to clear memory above 1 Mb.
4D	Memory above 1Mb cleared. (SOFT RESET). Going to save memory size.
4E	Memory test started. (NO SOFT RESET). About to display first 64K memory test.
4F	Memory size display started, updated during memory test. Going for sequential and random memory test.
50	Memory test below 1Mb complete. Going to adjust memory size for relocation/shadow.
51	Memory size adjusted due to relocation/shadow. Memory test above 1Mb to follow.
52	Memory test above 1Mb complete. Preparing to go back to real mode.
53	CPU registers saved including memory size. Going to enter real mode.
54	Shutdown successful; CPU in real mode. Restore registers saved during shutdown prep.
55	Registers restored. Going to disable gate A20 address line.
56	A20 address line disable successful. BIOS ROM data area about to be checked.
57	BIOS ROM data area check halfway. BIOS ROM data area check to be complete.
58	BIOS ROM data area check over. Going to clear Hit <ESC> message.
59	Hit <ESC> message cleared. <WAIT...> message displayed. About to start DMA and PIC test.
60	DMA page register test passed. About to verify from display memory.

Code	Meaning
61	Display memory verification over. About to go for DMA #1 base register test.
62	DMA #1 base register test passed. About to go for DMA #2 base register test.
63	DMA #2 base register test passed. About to go for BIOS ROM data area check.
64	BIOS ROM data area check halfway. BIOS ROM data area check to be complete.
65	BIOS ROM data area check over. About to program DMA unit 1 and 2.
66	DMA unit 1 and 2 programming over. About to initialize 8259 interrupt controller.
67	8259 initialization over. About to start keyboard test.
80	Keyboard test started. Clearing output buffer, checking for stuck key. Issue keyboard reset.
81	Keyboard reset error/stuck key found. About to issue keyboard controller interface command.
82	Keyboard controller interface test over. About to write command byte and init circular buffer.
83	Command byte written, Global data init done. About to check for lock-key.
84	Lock-key checking over. About to check for memory size mismatch with CMOS.
85	Memory size check done. About 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. Going to CMOS setup program.
88	Returned from CMOS setup program, screen is cleared. About to do programming after setup.
89	Programming after setup complete. Going to display power on screen message.
8A	First screen message displayed. About to display <WAIT...> message.
8B	<WAIT...> message displayed. About to do Main and Video BIOS shadow.
8C	Main/Video BIOS shadow successful. Setup options programming after CMOS setup about to start.
8D	Setup options programmed; mouse check and initialisation to be done next.
8E	Mouse check and initialisation complete. Going for hard disk and floppy reset.
8F	Floppy check returns that floppy is to be initialized. Floppy setup to follow.
90	Floppy setup is over. Test for hard disk presence to be done.
91	Hard disk presence test over. Hard disk setup to follow.
92	Hard disk setup complete. About to go for BIOS ROM data area check.
93	BIOS ROM data area check halfway. BIOS ROM data area check to be complete.
94	BIOS ROM data area check over. Going to set base and extended memory size.
95	Mem size adjusted due to mouse support, hard disk type 47. Going to verify from display memory.
96	Returned after verifying display memory. Doing any init before C800 optional ROM control
97	Any init before C800 option ROM control over. ROM check and control will be done next.
98	Optional ROM control done. Give control for required processing after ROM returns control.
99	Init 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 co-processor over. Going to initialize the coprocessor next.
9D	Coprocessor initialized. Going to do any initialization after coprocessor test.
9E	Initialization after copro test complete. Check extd keyboard, keyboard ID and num lock.
9F	Extd keyboard check is done, ID flag set. num lock on/off. Keyboard ID command to be issued.
A0	Keyboard ID command issued. Keyboard ID flag to be reset.
A1	Keyboard ID flag reset. Cache memory test to follow.
A2	Cache memory test over. Going to display soft errors.
A3	Soft error display complete. Going to set keyboard typematic rate.
A4	Keyboard typematic rate set. Going to program memory wait states.
A5	Memory wait states programming over. Screen to be cleared next.
A6	Screen cleared. Going to enable parity and NMI.
A7	NMI and parity enabled. Going to do any init 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. Do any initialisation after E000 optional ROM control.

Code	Meaning
AA	Initialization after E000 optional ROM control is over. Going to display the system configuration.
00	System configuration is displayed. Going to give control to INT 19h boot loader.

AMI WinBIOS; 12/15/93 Onwards

Code	Meaning
01	Processor register test about to start; disable NMI next.
02	NMI is Disabled. Power on delay starting.
03	Power on delay complete (to check soft reset/power-on).
05	Soft reset/power-on determined, enable ROM (i.e. disable shadow RAM cache, if any).
06	ROM is enabled. Calculating ROM BIOS checksum.
07	ROM BIOS checksum passed. CMOS shutdown register test to be done next.
08	CMOS shutdown register test done. CMOS checksum calculation next.
09	CMOS checksum calculation done; CMOS diag byte written; CMOS initialisation to begin.
0A	CMOS initialization done (if any). CMOS status register about to init for Date and Time.
0B	CMOS status register init done. Any initialization before keyboard BAT to be done next.
0C	KB controller I/B free. Going to issue the BAT command to keyboard controller.
0D	BAT command to keyboard controller is issued. Going to verify the BAT command.
0E	Keyboard controller BAT result verified. Any initialization after KB controller BAT next.
0F	Initialisation after KB controller BAT done. Keyboard command byte to be written next.
10	Keyboard controller command byte written. Going to issue Pin 23 & 24 block/unblock command.
11	Keyboard controller Pin 23/24 blocked/unblocked; check press of <INS> key during power-on .
12	Checking for pressing of <Ins> key during power-on done. Disabling DMA/Interrupt controllers.
13	DMA controller #1 and #2 and Interrupt controller #1 and #2 disabled; video display disabled and port B initialised; chipset init/auto memory detection next.
14	Chipset init/auto memory detection over. To uncompress the POST code if compressed BIOS.
15	POST code is uncompressed. 8254 timer test about to start.
19	8254 timer test over. About to start memory refresh test.
1A	Memory Refresh line is toggling. Going to check 15 micro second ON/OFF time.
20	Memory Refresh 30 microsecond test complete. Base 64K memory/address line test about to start.
21	Address line test passed. Going to do toggle parity.
22	Toggle parity over. Going for sequential data R/W test on base 64k memory.
23	Base 64k sequential data R/W test passed. Setting BIOS stack and any setup before Interrupt
24	Setup required before vector initialization complete. Interrupt vector initialization about to begin.
25	Interrupt vector initialization done. Going to read Input port of 9042 for turbo switch and clear password if POST diag switch is ON next.
26	Input port of 8042 is read. Going to initialize global data for turbo switch.
27	Global data init for turbo switch over. Any initialization before setting video mode next.
28	Initialization before setting video mode complete. Going for mono mode and colour mode setting.
2A	Mono and colour mode setting is done. About to go for toggle parity before optional ROM test.
2B	Toggle parity over. About to give control for setup before optional video ROM check next.
2C	Processing before video ROM control done. About to look for video ROM and give control.
2D	Video ROM control done. About to give control for processing after video ROM returns control.
2E	Return from processing after video ROM control. If EGA/VGA not found do display mem 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 mem R/W test or retrace checking failed. About to do alternate Display memory R/W test.
32	Alternate Display memory R/W test passed. Looking for the alternate display retrace checking.
34	Video display checking over. Display mode to be set next.

Code	Meaning
37	Display mode set. Going to display the power on message.
39	New cursor position read and saved. Going to display the Hit <DEL> message.
3B	Hit <DEL> message displayed. Virtual mode memory test about to start.
40	Going to prepare the descriptor tables.
42	Descriptor tables prepared. Going to enter in virtual mode for memory test.
43	Entered in virtual mode. Going to enable interrupts for diagnostics mode.
44	Interrupts enabled (if diags switch on). Going to initialize data to check mem wrap around at 0:0.
45	Data initialized. Going to check for memory wrap around at 0:0 and find total system memory.
46	Memory wrap around test done. Memory size calculation over. Go for 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 amount of memory below 1Mb.
49	Amount of memory below 1Mb found and verified. Finding amount of memory above 1Mb memory.
4B	Memory above 1Mb verified. Check for soft reset and going to clear memory below 1Mb for soft reset next (if power on go to POST # 4Eh).
4C	Memory below 1Mb cleared.(SOFT RESET)
4D	Memory above 1Mb cleared.(SOFT RESET); save memory size next (go to POST # 52h).
4E	Memory test started. (NOT SOFT RESET); display first 64K memory size next.
4F	Memory size display started.Will be updated during memory test; sequential and random memory test next.
50	Memory testing/initialisation below 1Mb complete. Adjust displayed memory size for relocation/ shadow.
51	Memory size display adjusted due to relocation/ shadow. Memory test above 1Mb to follow.
52	Memory testing/initialisation above 1Mb complete. Going to save size information.
53	Memory size is saved. CPU registers are saved. Going to enter real mode.
54	Shutdown successful, CPU in real mode, disable gate A20 line next.
57	A20 address line disable successful. Going to adjust memory size depending on relocation/shadow.
58	Memory size adjusted for relocation/shadow. Going to clear Hit <DEL> message.
59	Hit <DEL> message cleared. <WAIT...> message displayed. Start DMA and interrupt controller test.
60	DMA page register test passed. About to go for DMA #1 base register test.
62	DMA #1 base register test passed. About to go for DMA #2 base register test.
65	DMA #2 base register test passed. About to program DMA unit 1 and 2.
66	DMA unit 1 and 2 programming over. About to initialize 8259 interrupt controller.
67	8259 initialization over. About to start keyboard test.
F4	Extended NMI sources enabling is in progress (EISA).
80	Keyboard test. Clear output buffer; check for stuck key; issue reset keyboard command next.
81	Keyboard reset error/stuck key found. About to issue keyboard controller interface test command.
82	Keyboard controller interface test over. About to write command byte and init circular buffer.
83	Command byte written; global data init done; check for lock-key next.
84	Lock-key checking over. About to check for memory size mismatch with CMOS.
85	Memory size check done. About 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. Uncompress SETUP code and execute CMOS setup.
88	Returned from CMOS setup 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 msg displayed. <Wait...> message displayed. About to do Main/Video BIOS shadow.
8C	Main/Video BIOS shadow successful. Setup options programming after CMOS setup about to start.
8D	Setup options are programmed; mouse check and init next.
8E	Mouse check and initialisation complete. 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.
94	Hard disk setup complete. Going to set base and extended memory size.

Code	Meaning
96	Memory size adjusted for mouse support, hard disk type 47; init before C800, optional ROM next.
97	Init before C800 ROM control is over. Optional ROM check and control next.
98	Optional ROM control done. Give control for required processing after optional ROM returns control next.
99	Iinitialization required after optional ROM test over. 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 co-processor is over. Going to initialize the coprocessor next.
9D	Coprocessor initialized. Going to do any initialization after coprocessor test.
9E	Init after coprocessor test complete. Going to check extd keyboard; keyboard ID and NumLock.
9F	Extd keyboard check is done; ID flag set; NumLock on/off, issue keyboard ID command next.
A0	Keyboard ID command issued. Keyboard ID flag to be reset.
A1	Keyboard ID flag reset. Cache memory test to follow.
A2	Cache memory test over. Going to display any soft errors.
A3	Soft error display complete. Going to set the keyboard typematic rate.
A4	Keyboard typematic rate set. Going to program memory wait states.
A5	Memory wait states programming over. Going to clear the screen and enable parity/NMI.
A7	NMI and parity enabled.Ddo any initialization required before giving control to optional ROM at E000 next.
A8	Initialization before E000 ROM control over. E000 ROM to get control next.
A9	Returned from E000 ROM control. Going to do init required.
AA	Init after E000 optional ROM control is over. Going to display the system configuration.
B0	System configuration is displayed. Going to uncompress SETUP code for hot-key setup.
B1	Uncompressing of SETUP code is complete. Going to copy any code to specific area.
00	Copying of code to specific area done. Going to give control to INT 19h boot loader.

EISA

Code	Meaning
F0	Initialisation of I/O cards in slots is in progress (EISA).
F1	Extended NMI sources enabling is in progress (EISA).
F2	Extended NMI test is in progress (EISA).
F3	Display any slot initialisation messages.
F4	Extended NMI sources enabling in progress.

10/10/94

Code	Meaning
C2	NMI is Disabled. Power on delay starting
C5	Power on delay complete. Going to disable Cache if any
C6	Calculating ROM BIOS checksum
C8	CMOS shutdown register test done. CMOS checksum calculation to be done next
CA	CMOS checksum calc done, CMOS Diag byte written. CMOS status register to init for Date and Time
CB	CMOS status register init done. Any initialization before keyboard BAT to be done next
CD	BAT command to keyboard controller is to be issued
CE	Keyboard controller BAT result verified. Any initialization after KB controller BAT to be done next
CF	Initialization after KB controller BAT done. Keyboard command byte to be written next
D1	Keyboard controller command byte is written. Going to check pressing of INS key during power-on
D2	Checking for pressing of INS key during power-on done. Disabling DMA and Interrupt controllers
D3	DMA controller #1,#2, interrupt controller #1,#2 disabled. Chipset init auto memory detection now

Code	Meaning
D4	Chipset initialization/auto memory detection over. To uncompress the RUNTIME code
D5	RUNTIME code is uncompressed
DD	Transfer control to uncompressed code in shadow ram at F000:FFF0h

RUNTIME CODE IS UNCOMPRESSED IN F000H SHADOW RAM

Code	Meaning
03	NMI is Disabled. To check soft reset/power-on
05	Soft reset/power-on determined. Going to disable Cache if any
06	POST code to be uncompressed
07	POST code is uncompressed. CPU init and CPU data area init to be done next
08	CPU and CPU data area init done. CMOS checksum calculation to be done next
09	CMOS checksum done, Diag byte written. CMOS init begin (If "Init CMOS in every boot" set)
0A	CMOS initialization done (if any). CMOS status register about to init for Date and Time
0B	CMOS status register init done. Any initialization before keyboard BAT to be done next
0C	KB controller I/B free. Going to issue the BAT command to keyboard controller
0D	BAT command to keyboard controller is issued. Going to verify the BAT command
0E	Keyboard controller BAT result verified. Initialization after KB controller BAT to be done next
0F	Initialization after KB controller BAT done. Keyboard command byte to be written next
10	Keyboard controller command byte is written. Issue Pin-23,24 blocking/unblocking command
11	Pin-23,24 of keyboard controller blocked/ unblocked. Check pressing of INS key on power-on
12	Check for pressing of INS key during power-on done. Disable DMA and Interrupt controllers
13	DMA controller #1,#2, interrupt controller #1,#2 disabled. Video display is disabled and port-B is initialized.Chipset init about to begin
15	Chipset initialization over. 8254 timer test to start. 8254 timer test over. Start refresh test
1A	Memory Refresh line is toggling. Going to check 15 micro second ON/OFF time
20	Memory Refresh period 30 micro second test complete. Base 64K memory to be initialized
23	Base 64K memory initialized. SDet BIOS stack and to do any setup before Interrupt vector init
24	Setup required before interrupt vector init complete. Interrupt vector initialization to begin
25	Interrupt vector initialization done. Going to read Input port of 9042 for turbo switch (if any) and to clear password if post diag switch is on
26	Input port of 8042 is read. Going to initialize global data for turbo switch
27	Global data initialization for turbo switch over. Initialization before setting video mode next
28	Initialization before setting video mode complete. Monochrome mode and color mode setting
2A	Different Buses init (system, static, output devices) to start if present
2B	About to give control for any setup required before optional video ROM check
2C	Processing before video ROM control is done. Look for optional video ROM and give control
2D	Optional video ROM control done. Give control for processing after video ROM returns control
2E	Back after video ROM control. If EGA/VGA not found do display mem 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.Ddo alternate Display memory R/W test
32	Alternate Display memory R/W test passed. 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.
39	Display different Buses initialization error messages.
3A	New cursor position read and saved. Going to display the Hit DEL message
3B	Hit DEL message displayed. Virtual mode memory test about to start

Code	Meaning
40	Going to prepare the descriptor tables
42	Descriptor tables prepared. Going to enter in virtual mode for memory test
43	Entered protected mode. Enabling interrupts for diagnostics mode next
44	Interrupts enabled if diag switch is on. Initializing to check memory wraparound at 0:0 next
45	Data initialized. Check for memory wraparound at 0:0 and find the total system memory size
46	Memory wraparound test done. Size calculation done. 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 findout amount of memory below 1M memory
49	Memory below 1Mb found and verified.Find out amount of memory above 1M memory
4B	Amount of memory above 1Mb 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 1Mb cleared. (SOFT RESET) Going to clear memory above 1M
4D	Memory above 1Mb cleared. (SOFT RESET) Save the memory size. (Go to check point# 52h)
4E	Memory test started. (NOT SOFT RESET) About to display the first 64k memory size
50	Memory testing/initialization below 1Mb complete. Adjust 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 1Mb complete. Going to save memory size information
53	The memory size information and the CPU registers are saved. Entering real mode next
54	Shutdown successful, CPU in real mode. Disable gate A20 line and disable parity/NMI
57	A20 address line, parity/NMI disable successful. Adjust memory size on relocation/shadow
58	Memory size adjusted for relocation/shadow. Going to clear Hit DEL message
59	Hit DEL message cleared. WAIT... message displayed. Start DMA and interrupt controller test
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
67	8259 initialization over
7F	Extended NMI sources enabling is in progress
80	Keyboard test started. clearing output buffer, checking for stuck key. Issue keyboard reset
81	Keyboard reset error/stuck key. About to issue keyboard controller interface test command
82	Keyboard controller interface test over. About to write command byte and init circular buffer
83	Command byte written. Global data init done. About to check for lock-key
84	Lock-key checking over. About to check for memory size mismatch with CMOS
85	Memory size check done. About 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. Uncompress SETUP code and execute CMOS setup
88	Returned from CMOS setup 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 WAIT message displayed. About to do Video BIOS shadow
8C	Video BIOS shadow successful. Setup options programming after CMOS setup about to start
8D	Setup options are programmed, mouse check and init to be done next
8E	Mouse check and initialization complete. 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
94	Hard disk setup complete. To set base and extended memory size
95	Memory size adjusted for mouse support. Init of different buses optional ROMs from C800
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 next
98	Optional ROM control done. Give control for processing after optional ROM returns control

Code	Meaning
99	Any init required after optional ROM test over. 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 complete. Check extd keyboard, keyboard ID and num-lock
9F	Extd keyboard check is done, ID flag set, num-lock on/off. Keyboard ID command to be issued
A0	Keyboard ID command issued. ID flag to be reset. A1 Keyboard ID flag reset. Cache test next
A2	Cache memory test over. Going to display any soft errors
A3	Soft error display complete. Going to set keyboard typematic rate
A4	Keyboard typematic rate set. To program memory wait states
A5	Memory wait states programming over. Going to clear the screen and enable parity/NMI
A7	NMI and parity enabled. Do initialization required before giving control to ROM at E000h
A8	Initialization before E000 ROM control over. E000 ROM to get control next
A9	Returned from E000 ROM control. Initialization required after E000 optional ROM control
AA	Initialization after E000 ROM control is over. Going to display the system configuration
B0	System configuration is displayed
B1	Going to copy any code to specific area
00	Code copying to specific areas is done. Passing control to INT 19h boot loader next

Version 6.2 - 7/15/95

Valid for all AMI products with a core BIOS date of 7/15/95. Control is passed to different buses at 2A, 38, 39 and 95, where additional checkpoints are output to port 80 as word to identify the routines being executed. These are word checkpoints - the low byte is where control is passed, and the high byte contains this information:

Bits	Meaning
7-4	0000 Function 0. Disable all devices on the bus 0001 Function 1. Init static devices 0010 Function 2. Init output devices 0011 Function 3. Init input devices 0100 Function 4. Init IPL devices 0101 Function 5. Init general devices 0110 Function 6. Init error reporting 0111 Function 7. Init add-on ROMs
3-0	Specify the bus: 0 Generic DIM device Init Manager 1 Onboard system devices 2 ISA devices 3 EISA devices 4 ISA PnP devices 5 PCI devices

UNCOMPRESSED INITIALISATION CODES

Code	Meaning
D0h	NMI is disabled, power on delay starting. Init code checksum to be verified next
D1h	Init DMA controller,do keyboard controller BAT test, start memory refresh, 4 Gb flat mode next
D3h	Start memory sizing next
D4h	Return to real mode. Execute OEM patches and set stack.

Code	Meaning
D5h	Passing control to uncompressed code in Shadow RAM at E000:0000h. Init code is copied to segment 0 and control will be transferred to it.
D6h	Control is in segment 0.Next, checking if Ctrl-Home was pressed and verifying BIOS checksum. If keys were pressed or checksum is bad, next will go to checkpoint code E0h. Otherwise, go to D7h
DD	Early initialization super IO chips that contain disabled at power on RTC and KBC
D0	NMI disabled. Power on delay starting. Next, the initialization code checksum will be verified
D1	Initializing the DMA controller, performing the keyboard controller BAT test, starting memory refresh, and entering 4 GB flat mode next
D3	Starting memory sizing next
D4	Returning to real mode. Executing any OEM patches and setting the stack next
D5	Passing control to the uncompressed code in shadow RAM at E000:0000h. The initialization code is copied to segment 0 and control will be transferred to segment 0
D6	Control is in segment 0. Next, checking if CTRL HOME was pressed and verifying the system BIOS checksum. If either CTRL HOME was pressed or the system BIOS checksum is bad, next will go to checkpoint code E0h. Otherwise, going to checkpoint code D7h.
D7	Passing control to the interface module next
D8	The main system BIOS runtime code will be decompressed next
D9	Passing control to the main system BIOS in shadow RAM next

BOOTBLOCK RECOVERY CODES

Code	Meaning
E0h	FD controller initialized. Next, beginning base 512K memory test.
E1h	Init interrupt vector table next
E2h	Init DMA and interrupt controllers next
E6h	Enabling FD controller and timer IRQs. Enabling internal cache memory
EDh	Init floppy drive
EEh	Looking for floppy in drive A:. Reading first sector
EFh	Read error from floppy in A:
F0h	Next, searching for AMIBOOT ROM file in root directory
F1h	AMIBOOT ROM file not in root directory
F2h	Next, reading and analyzing floppy FAT to find clusters occupied by AMIBOOT ROM file
F3h	Next, reading AMIBOOT ROM file, cluster by cluster
F4h	AMIBOOT ROM file not the correct size
F5h	Next, disabling internal cache
FBh	Next, detecting type of flash ROM
FCh	Next, erasing flash ROM
FDh	Next, programming flash ROM
FFh	Flash ROM programming successful. Next, restarting System BIOS

UNCOMPRESSED INITIALISATION CODES IN F0000H SHADOW RAM

Code	Meaning
03	NMI disabled. Next, checking for soft reset or power on
05	BIOS stack has been built. Next, disabling cache
06	Uncompressing POST code
07	Init CPU and CPU data area
08	CMOS checksum calculation next

Code	Meaning
0A	CMOS checksum calculation done. Init CMOS status register for data and time next
0B	CMOS status register initialized. Next, performing init before keyboard BAT command issued
0C	Keyboard controller input buffer is free. Next, issuing BAT command to keyboard controller
0E	Keyboard controller BAT command verified. Next, performing any necessary init after BAT test
0F	init after BAT test done. Keyboard command byte written next
10	Keyboard controller command byte written. Next, pin 23/ 24 blocking/unblocking command
11	Next, checking if End or Ins keys were pressed during power on. Init CMOS RAM if if Initialise CMOS RAM in every boot POST option was set in AMIBCP or End key was pressed.
12	Next, disabling DMA controllers 1 and 2 and interrupt controllers 1 and 2
13	Video display disabled and port B initialized. Next, init chipset.
14	8254 timer test next
19	8254 timer test over. Starting memory refresh test.
1A	Memory refresh line toggling. Checking 15 second on/off time.
23	Reading 8042 input port and disabling MEGAKEY Green PC feature next. Making BIOS Code segment writable and performing necessary configuration before initializing interrupt vectors.
24	Configuration required before interrupt vector init completed. Interrupt vector init begins.
25	Interrupt vector init done. Clearing password if the POST DIAG switch is on.
27	Any init before setting video mode will be done next.
28	Init before setting video mode is complete.
2A	Bus init system, static, output devices will be done next, if present.
2B	Passing control to video ROM to perform any required configuration before video ROM test.
2C	All necessary processing before passing control to Video ROM is done. Looking for Video ROM next and passing control to it.
2D	Video ROM returned control to POST.Doing required processing after Video ROM had control.
2E	Completed post-video ROM test processing. If EGA/VGA controller is not found, performing display memory read/write test next.
2F	EGA/VGA controller was not found. Display memory read/write test is about to begin.
30	Display memory read/write test passed. Look for retrace checking next.
31	Display memory read/write test or retrace checking failed. Performing alternate display memory read/write test.
32	Alternate display memory read/write test passed. Look for alternat display retrace check next.
34	Video display checking over. Setting display mode next.
37	Display mode set. Displaying power on message next.
38	Initialising bus input, IPL, general devices next, if present.
39	Displaying bus init error messages.
3A	New cursor position read and saved. Displaying the Hit <Del> message next.
3B	Hit <Del> message displayed. Protected mode memory test about to start.
40	Preparing descriptor tables next.
42	Descriptor tables prepared. Entering protected mode for memory test next.
43	Entered protected mode. Enabling interrupts for diags mode next.
44	Interrupts enabled if diags switch is on. Initialising data to check memory wraparound at 0:0
45	Data initialized. Check for memory wraparound at 0:0 and find total system memory size next.
46	Memory wraparound test done. Mem size calculation done. Writing patterns to test memory
47	Memory pattern has been written to extended memory. Writing patterns to base 640K next.
48	Patterns written in base memory. Determining memory below 1 Mb next.
49	Memory below 1 Mb found and verified. Determining memory above 1 Mb next.
4B	Memory above 1 Mb verified. Checking for soft reset, clearing memory below 1 Mb for soft reset next. If power on, checking 4Eh next.
4C	Memory below 1 Mb cleared via soft reset. Clearing memory above 1 Mb next.
4D	Memory above 1 Mb cleared via soft reset. Saving memory size next. 52h next.

Code	Meaning
4E	Memory test started, but not as result of soft reset. Displaying first 64K memory size next.
4F	Memory size display started. Display updated during test.Performing sequential and random memory test next.
50	Memory below 1 Mb tested and initialized. Adjusting displayed memory size for relocation and shadowing next.
51	Memory size display adjusted for relocation and shadowing. Testing memory above 1 Mb.
52	Memory above 1 Mb tested and initialized. Saving memory size information next.
53	Memory size information and CPU registers are saved. Entering real mode next.
54	Shutdown successful, CPU in real mode. Disabling Gate A20 line, parity and NMI next.
57	Gate A20 line, parity and NMI disabled. Adjust memory size on relocation and shadowing next.
58	Adjusted memory size depending on relocation and shadowing. Clearing Hit <Del> message.
59	Hit <Del> message cleared. <Wait> message displayed.
60	DMA page register test passed.
62	DMA controller 1 base register test passed.
65	DMA controller 2 base register test passed.
66	Completed programming DMA controllers 1 and 2.
67	Completed 8259 interrupt controller init.
7F	Extended NMI source enabling in progress
80	Keyboard test started. Clearing output buffer and checking for stuck keys.
81	Keyboard reset error or stuck key found.
82	Keyboard controller interface test completed.
83	Command byte written and global data init completed
84	Locked key checking over.
85	Memory size check done.
86	Password checked.
87	Programming before WINBIOS setup complete.
88	Returned from WINBIOS setup and cleared screen.
89	Programming after WINBIOS setup completed.
8B	First screen power on message displayed, and <Wait> message. PS/2 mouse check and extended BIOS data area allocation check next.
8C	Programming WINBIOS setup options next.
8D	WINBIOS setup options programmed.
8F	HD controller reset.
91	FD controller configured.
95	Initialising bus option ROMs from C800 next.
96	Init before passing control to adapter ROM at C800
97	Init before C800 ROM gains control completed.
98	Adapter ROM has passed control back to BIOS POST. Performing required programming.
99	Init required after ROM test now complete.
9A	Set timer and printer base addresses.
9B	RS232 base address set.
9C	Required init before coprocessor test over.
9D	Coprocessor initialized.
9E	Init after copro test complete. Checking extended keyboard, keyboard ID and Num Lock key next. Issuing keyboard ID command next.
A2	Displaying soft errors next.
A3	Soft error display complete.
A4	Keyboard typematic rate set.
A5	Memory wait state programming over.

Code	Meaning
A7	Screen cleared, NMI and parity enabled.Init before passing control to ROM at E000 complete. Passing control to E000 next.
A8	Init before E000 control complete.
A9	Returned from E000 control.
AA	Init after E000 control complete. Displaying system configuration next.
AB	Uncompressing DMI data and executing DMI POST init next.
B0	System configuration displayed
B1	Copying code to specific areas.
00	Code copying done. Passing control to INT 19 boot loader next.

ARCHE TECHNOLOGIES

Legacy BIOS

Derives from AMI (9 April 90), using port 80; certain codes come up if a copy is made without AMI's copyright notice. The major differences are at the end.

Code	Explanation
01	Disable NMI and test CPU registers
02	Verify ROM BIOS checksum (32K at F800:0)
03	Initial keyboard controller and CMOS RAM communication
04	Disable DMA and interrupt controllers; test CMOS RAM interrupt
05	Reset Video
06	Test 8254 timer
07	Test delta count for timer channel 2 (speaker)
08	Test delta count for timer channel 1 (memory refresh)
09	Test delta count for timer channel 0 (system timer)
0A	Test parity circuit and turn on refresh
0B	Enable parity check circuit and test system timer
0C	Test refresh trace link toggle
0D	Test refresh timing synchronization of high and low period
10	Disable cache and shadow BIOS; test 64K base memory address lines
11	Test base 64K memory for random addresses and data read/write
12	Initialize interrupt vectors in lower 1K of RAM
14	Test CMOS RAM shutdown register read/write; disable DMA and interrupt controllers
15	Test CMOS RAM battery and checksum, and different options such as diagnostic byte
16	Test floppy information in CMOS RAM; initialize monochrome video
17	Initialise colour video
18	Clear parity status if any
19	Test for EGA/VGA video ROM BIOS at C000:0 and pass control to it if there
1A	Returned from video ROM. Clear parity status if any; update system parameters for any video ROM found; test display memory read/write
1B	Primary video adapter: check vertical/horizontal retrace; write/read test video memory
1C	Secondary video adapter: check vertical and horizontal retrace; write/read test video memory
1D	Compare and verify CMOS RAM video type with switches and video adapter; set equipment byte if correct
1E	Call BIOS to set mono/colour video mode according to CMOS RAM

Code	Explanation
20	Display CMOS RAM write/read errors and halt if any
21	Set cursor to next line and call INT 10 to display
22	Display Power on 386 BIOS message and check CPU speed is 25 or 33 MHz
23	Read new cursor position and call INT 10 to display
24	Skip 2 rows of text and display (C)AMI at bottom of screen
25	Refresh is off, so call shadow RAM test
F0	Failure inside shadow RAM test
30	Verify (C)AMI... and overwrite with blanks before entering protected mode
31	Enter protected mode and enable timer interrupt (IRQ0). Errors indicate gate A20 circuit failed
32	Size memory above 1Mb
33	Size memory below 640K
34	Test memory above 1Mb
35	Test memory below 1Mb
36	Unknown AMI function
37	Clear memory below 1Mb
38	Clear memory above 1Mb
39	Set CMOS shutdown byte to 3 and go back to real mode
3A	Test sequential and random data write/read of base 64K RAM
3B	Test RAM below 1Mb and display area being tested
3C	Test RAM above 1Mb and display area being tested
3D	RAM test OK
3E	Shutdown for return to real mode
3F	Back in real mode; restore all variables
40	Disable gate A20 since now in real mode
41	Check for (C)AMI in ROM
42	Display (C)AMI message
43	Clear <Esc> message; test cache
4E	Process shutdown 1; go back to real mode
4F	Restore interrupt vectors and global data in BIOS RAM area
50	Test 8237 DMA controller and verify (c)AMI in ROM
51	Initialize DMA controller
52	Test various patterns to DMA controller
53	Verify (C)AMI in ROM
54	Test DMA control flip-flop
55	Initialize and enable DMA controllers 1 and 2
56	Initialize 8259 interrupt controllers-clear write request and mask registers
57	Test 8259 controllers and setup interrupt mask registers
61	Check DDNIL status bit and display message if clear
70	Perform keyboard BAT (Basic Assurance Test)
71	Program keyboard to AT type
72	Disable keyboard and initialize keyboard circular buffer
73	Display DEL message for setup prompt and initialize floppy controller/drive
74	Attempt to access floppy drive
75	If CMOS RAM is good, check and initialize hard disk type identified in CMOS RAM
76	Attempt to access hard disk and set up hard disk
77	Shuffle any internal error codes
78	Verify (C)AMI is in ROM
79	Check CMOS RAM battery and checksum; clear parity status
7A	Compare size of base/extended memory to CMOS RAM info

Code	Explanation
7B	Unknown AMI function
7C	Display (C)AMI
7D	Set/reset AT compatible memory expansion bit
7E	Verify (C)AMI is in ROM
7F	Clear <DEL> message from screen and check if DEL pressed
80	Find option ROM in C800 to DE00 and pass control to any found
81	Return from adapter ROM; initialize timer and data area
82	Setup parallel and serial port base info in global data area
83	Test for presence of 80387 numeric coprocessor and initialize
84	Check lock key for keyboard
85	Display soft error messages if CMOS RAM data error was detected (battery or checksum)
86	Test for option ROM in E000:0 and pass control to any found
A0	Error in 256 Kbit or 1Mbit RAM chip in lower 640K memory
A1	Base 64K random address/data pattern test (only in 386APR and Presto 386SX)
A9	Initialize on-board VGA (Presto 386SX)
B0	Error in 256 Kbit RAM chip in lower 640K memory
B1	Base 64K random address/data pattern test (only in Presto 386SX BIOS)
E0	Returned to real mode; initialise base 64K RAM (Presto)
E1	initialize base 640K RAM (Presto)
EF	Configuration memory error in Presto -can't find memory
F0	Test shadow RAM from 0:4000 RAM area
00	Call INT 19 boot loader

AST

See also Phoenix or (mostly) Award. AST introduced an enhanced BIOS in 1992 with 3 beeps before all early POST failure messages, for Field Replaceable Unit identification. Otherwise, the most significant (left) digit of the POST code indicates the number of long beeps, and the least significant (right) digit indicates the short beeps. 17 therefore means 1 long beep and 7 short. Doesn't work after 20. Errors below 20 are generally fatal.

Early POST Codes

These are usually fatal and accompanied by a beep code:

Code	Meaning
1	System Board
2	SIMM Memory; System Board
3	SIMM Memory; System Board
4	SIMM Memory; System Board
5	Processor; System Board
6	Keyboard Controller; System Board
7	Processor; System Board
8	Video Adapter; Video RAM; System Board
9	BIOS; System Board
10	System Board
11	External cache; System Board

Code	Meaning
00	Reserved
	Beep and Halt if Error occurs
01	Test CPU registers and functionality
02	Test empty 8042 keyboard controller buffer
03	Test 8042 keyboard controller reset
04	Verify keyboard ID and low-level keyboard communication
05	Read keyboard input port (WS386SX16 only)
06	Initialise system board support chipset
09	Test BIOS ROM checksum; flush external cache
0D	Test 8254 timer registers (13 short beeps)
0E	Test ASIC registers (CLEM only, 14 short beeps)
0F	Test CMOS RAM shutdown byte (15 short beeps)
10	Test DMA controller 0 registers
11	Test DMA controller 1 registers
12	Test DMA page registers (see code 17)
13	see code 17
14	Test memory refresh toggle (see code l7)
15	Test base 64K memory
16	Set interrupt vectors in base memory
17	Init video; if EGA/VGA, issue code 12-13 if error, but only use this POST code beep pattern
12	EGA/VGA vertical retrace failed (different from normal beep)
13	EGA/VGA RAM test failed (different than normal beep tone)
14	EGA/VGA CRT registers failed (different than normal beep)
18	Test display memory
	Don't beep and don't halt if error occurs
20	EISA bus board power up (EISA Systems only)
30	Test interrupt controller #1 mask register
31	Test interrupt controller #2 mask register
32	Test interrupt controllers for stuck interrupt
33	Test for stuck NMI (P386 25/33, P486, CLEM and EISA)
34	Test for stuck DDNIL status bit (CLEM only)
40	Test CMOS RAM backup battery
41	Calculate and verify CMOS RAM checksum
42	Setup CMOS RAM options (except WS386SX16)
50	Test protected mode
51	Test protected mode exceptions
60	Calculate RAM size
61	Test RAM
62	Test shadow RAM (WS386SXl6, P386 25/33, P486, CLEM, EISA), or test cache (P386/l6)
63	Test cache (P38625/33, P486, CLEM, EISA), or copy system BIOS to shadow RAM (P386C, P386/l6, WS386SX16)
64	Copy system BIOS to shadow RAM (P386 25/33, P486, CLEM, EISA), or copy video BIOS to shadow RAM (P38616, SW386SX16)
65	Copy video BIOS to shadow RAM (P386 25/33, P486, CLEM, EISA), or test cache (WS386SX16)
66	Test 8254 timer channel 2 (P386 25/33, P486, EISA)
67	Initialize memory (Eagle only)

AT&T

Phoenix or Olivetti (M24 for early 6300s , and Phoenix for Intel boards). After 1991, NCR.

Code	Meaning	Code	Meaning
01	CPU Test	25	Unexpected Interrupt
02	System I/O Port	26	Expected Interrupt
03	ROM Checksum	30	Protected Mode for AT-Bus or Base Memory
05	DMA Page Register	31	Size of AT-Bus or External Memory
06	Timer 1	32	Address Lines A16..A23
07	Timer 2	33	Internal Memory or Conventional Memory Test
08	RAM Refresh	34	AT-Bus Memory Test or External Memory Test
09	8/19-Bit Bus Conversion	38	Shadow ROM BIOS
0A	Interrupt Controller 1	39	Shadow Extension BIOS
0B	Interrupt Controller 2	40	Enable/Disable Keyboard
0C	Keyboard Controller	41	Keyboard Clock and Data
0D	CMOS RAM/RTC	42	Keyboard Reset
0E	Battery Power Lost	43	Keyboard Controller
0F	CMOS RAM Checksum	44	A20 Gate
10	CPU Protected Mode	50	Initialize Interrupt Table
11	Display Configuration	51	Enable Timer Interrupt
12	Display Controller	60	Flexible (Floppy) Controller/Drive
13	Primary Display Error	61	Fixed (Hard) Disk Controller
14	Extended DMOS Test	62	Initialize Flexible (Floppy) Drives
15	AT-Bus Reset	63	Initialize Fixed (Hard) Drives
16	Initialize Tiger-Register	70	Real Time Clock (RTC)
17	Exists Extension ROM	71	Set Real Time Clock
18	Internal Mem Address Test	72	Parallel Interfaces
19	Remap Memory	73	Serial Interfaces
1A	Interleave Mode	74	External ROMs
1B	Remap Shadow Memory	75	Numeric Coprocessor
1C	Setup MRAM	76	Enable Keyboard and RTC Interrupts (IRQ9)
1D	Expanded Memory	F0	Display System Message
1E	AT Memory Error	F1	ROM at E000H
1F	Internal Memory Error	F2	Boot from Floppy or Hard Disk
20	Minimum Complete	F3	Setup Program
21	DMA Controller 1	F4	Password Program
22	DMA Controller 2	FC	DRAM Type Detection
23	Timer 0	FD	CPU Register Test
24	Initialize Internal Controllers		

Version 3.0

Code	Meaning
01	CPU test 1: verify CPU status bits
02	Powerup check - Init motherboard and chipset with default values; Check 8042 buffer
03	Clear 8042 keyboard controller - send command AA, fail if status is not 2 output buffer full
04	Reset 8042 keyboard controller
05	Get 8042 keyboard controller manufacturing status

Code	Meaning
06	Init motherboard chipset; disable color/mono video, 8237 DMA controller; reset 80x87; initialize 8255 timer 1; clear DMA/page registers/CMOS RAM shutdown byte
07	CPU test 2; read/write/verify CPU registers SS, SP, BP, with FF and 00
08	Initialize CMOS RAM/RTC
09	Checksum 32K of BIOS ROM
0A	Initialize video interface; initialize 6845 controller
0B	Test 8254 programmable interrupt timer channel 0
0C	Test 8254 programmable interrupt timer channel 1
0D	Test 8254 programmable interrupt timer channel 2
0E	Test CMOS RAM shutdown byte
0F	Test extended CMOS RAM, if present
10	Test 8237 DMA controller channel 0
11	Test 8237 DMA controller channel 1
12	Test 8237 DMA controller page registers
13	Test 8741 keyboard controller interface
14	Test memory refresh toggle
15	Test first 64K of base memory
16	Set up interrupt tables in low memory
17	Set up video I/O operations
18	(1 beep) Test MDA/CGA video memory unless EGA/VGA adapter is found
19	Test 8259 programmable interrupt timer channel 1
1A	Test 8259 programmable interrupt timer channel 0
1B	Test CMOS RAM battery level
1C	Test CMOS RAM checksum
1D	Set system memory size parameters
1E	Size base memory 64K at a time
1F	Test base memory found from 64K to 640K
20	Test stuck bit in 8259 programmable interrupt controller
21	Test for stuck NMI bits
22	Test 8259 programmable interrupt controller functionality
23	Test protected mode
24	Size extended memory above 1MB
25	Test all base and extended memory found, except the first 64K
26	Test protected mode exceptions
27	Init shadow RAM and move system BIOS and/or video BIOS into it if enabled by CMOS setup
28	Detect and initialize Intel 8242/8248 chip
2A	Detect and initialize keyboard
2B	Detect and initialize floppy drive
2C	Detect and initialize serial ports
2D	Detect and initialize parallel ports
2E	Detect and initialize hard drive
2F	Detect and initialize coprocessor
30	Reserved
31	Detect and initialize adapter ROMs
BD	Initialize Orvonton cache controller, if present
CA	Initialize 386 Micronics cache, if present
CC	Shutdown NMI handler
EE	Test for unexpected processor exception
FF	Interrupt 19 boot loader

Version 3.00-3.03 8/26/87 286 N3.03 Extensions

Code	Meaning
01	CPU test 1
02	Determine type of POST test
03	Clear 8042 keyboard controller
04	Reset 8042 keyboard controller
05	Get 8042 keyboard controller manufacturing status
06	Initialize LSI onboard chips
07	CPU test 2
08	Initialize CMOS RAM/RTC
09	Checksum 32K of BIOS ROM
0A	Initialize video interface
0B	Test 8254 programmable interrupt timer channel 0
0C	Test 8254 programmable interrupt timer channel 1
0D	Test 8254 programmable interrupt timer channel 2
0E	Test CMOS date and timer
0F	Test CMOS shutdown byte
10	Test DMA controller channel 0
11	Test DMA controller channel 1
12	Test DMA controller page registers
13	Test 8741 keyboard controller interface
14	Test memory refresh toggle
15	Test first 64K of base memory
16	Set up interrupt tables
17	Set up video I/O
18	Test video memory
19	Test 8259 programmable interrupt controller channel 1 mask bits
1A	Test 8259 programmable interrupt controller channel 2 mask bits
1B	Test CMOS battery level
1C	Test CMOS checksum
1D	Setup configuration byte for CMOS
1E	Sizing system memory & compare with CMOS
1F	Test found system memory
20	Test stuck 8259's interrupt bits
21	Test for stuck NMI bits
22	Test 8259 programmable interrupt controller functionality
23	Test protected mode and A20 gate
24	Size extended memory above 1MB
25	Test found system/extended memory
26	Test protected mode exceptions
2A	Detect and initialize keyboard
2B	Detect and initialize floppy drive
2C	Detect and initialize serial ports
2D	Detect and initialize parallel ports
2E	Detect and initialize hard drive
2F	Detect and initialize coprocessor
30	Test for unexpected processor exception
CC	POST_NMI

XT 8088/86 v3.1

Code	Meaning
01	CPU test 1
02	Determine type of POST test
06	Initialize 8259 programmable interrupt controller and 8237 DMA controller chips
07	CPU test 2
09	Checksum 32K of BIOS ROM
0A	Initialize video controller 6845 registers
15	Test first 64K of base memory
16	Set up interrupt tables
17	Set up video I/O
18	Test video memory
19	Test 8259 programmable interrupt controller channel 1 mask bits
1A	Test 8259 programmable interrupt controller channel 2 mask bits
1D	Setup configuration byte for CMOS
1E	Sizing system memory & compare with CMOS
1F	Test found system memory
20	Test stuck 8259's interrupt bits
21	Test for stuck NMI bits
22	Test 8259 programmable interrupt controller functionality
2A	Detect and initialize keyboard
2B	Detect and initialize floppy drive
2C	Detect and initialize serial ports
2D	Detect and initialize parallel ports
2E	Detect and initialize hard drive
2F	Detect and initialize coprocessor
31	Initialize option ROM's
FF	Interrupt 19 boot loader

386 v3.1

Code	Meaning	Code	Meaning
01	CPU test 1	1B	Test CMOS battery level
02	Determine type of POST test	1C	Test CMOS checksum data at 2E and 2Fh
03	Clear 8042 keyboard controller	1D	Configuration of CMOS if checksum good
04	Reset 8042 keyboard controller	1E	Sizing memory & compare with CMOS
05	Get 8042 keyboard controller status	1F	Tests memory from 64K to top of memory
06	Initialize LSI onboard chips	20	Test stuck 8259's interrupt bits
07	CPU test 2	21	Test for stuck NMI bits
08	Initialize CMOS RAM/RTC	22	Test 8259 PIC functionality
09	Checksum 32K of BIOS ROM	23	Test protected mode and A20 gate
0A	Initialize video interface	24	Size extended memory above 1MB
0B	Test 8254 PIC timer channel 0	25	Test extended memory with virtual 8086 mode and writing FFFF/AA55/0000 pattern
0C	Test 8254 PIC timer channel 1	26	Test protected mode exceptions
0D	Test 8254 PIC timer channel 2	27	Test cache controller or shadow RAM
0E	Test CMOS shutdown byte	28	Set up cache controller or 8042 controller
0F	Test extended CMOS	2A	Detect and initialize keyboard

Code	Meaning	Code	Meaning
10	Test DMA controller channel 0	2B	Detect and initialize floppy drive
11	Test DMA controller channel 1	2C	Detect and initialize serial ports
12	Test DMA controller page registers	2D	Detect and initialize parallel ports
13	Test 8741 keyboard controller interface	2E	Detect and initialize hard drive
14	Test memory refresh toggle	2F	Detect and initialize coprocessor
15	Test first 64K of base memory	31	Detect and initialize option ROMs
16	Set up interrupt tables	3B	Init sec cache with Opti chipset (486 only)
17	Set up video I/O	CC	NMI handler shutdown
18	Test video memory	EE	Test for unexpected processor exception
19	Test 8259 PIC channel 1 mask bits	FF	Interrupt 19 boot loader
1A	Test 8259 PIC channel 2 mask bits		

ISA/EISA v4.0

Code	Meaning	Code	Meaning
01	CPU test 1	4E	If POST loop pin set, reboot, otherwise display non-fatal error messages
02	CPU test 2	4F	Security check
03	Calculate BIOS EPROM	50	Write all CMOS values back to CMOS RAM
04	Test CMOS RAM interface	51	Preboot enabled
05	Initialize chipset	52	Initialize ROMs between C80000-EFFFF
06	Test memory refresh	53	Initialize time value at address 40 of BIOS
07	Setup low memory	55	Initialize DDNIL counter to NULL's
08	Setup interrupt vector table	63	Boot attempt
09	Test CMOS RAM checksum & load default	B0	Spurious interrupt in protected mode
0A	Initialize keyboard	B1	Unclaimed NMI
0B	Initialize video interface	BF	Program chipset
0C	Test video memory	C0	OEM specific
0D	OEM specific; init motherboard chips	C1	OEM specific
0F	Test DMA controller 0	C2	OEM specific
10	Test DMA controller 1	C3	OEM specific
11	DMA page registers	C4	OEM specific
14	Test 8254 timer 0 counter 2	C5	OEM specific
15	Verify 8259 PIC channel 1	C6	OEM specific
16	Verify 8259 PIC channel 2	C7	OEM specific
17	Test for stuck 8259 interrupt bits	C8	OEM specific
18	T Test 8259 functionality	C9	OEM specific
19	Test for NMI bits	D0	Debug
1F	Set EISA mode	D1	Debug
20	Initialize and enable EISA slot 0	D2	Debug
21	Initialize and enable EISA slot 1	D3	Debug
22	Initialize and enable EISA slot 2	D4	Debug
23	Initialize and enable EISA slot 3	D5	Debug
24	Initialize and enable EISA slot 4	D6	Debug
25	Initialize and enable EISA slot 5	D7	Debug
26	Initialize and enable EISA slot 6	D8	Debug
27	Initialize and enable EISA slot 7	D9	Debug
28	Initialize and enable EISA slot 8	DA	Debug
29	Initialize and enable EISA slot 9	DB	Debug

Code	Meaning	Code	Meaning
2A	Initialize and enable EISA slot 10	DC	Debug
2B	Initialize and enable EISA slot 11	DD	Debug
2C	Initialize and enable EISA slot 12	DE	Debug
2D	Initialize and enable EISA slot 13	DF	Debug
2E	Initialize and enable EISA slot 14	E1	Setup page 1
2F	Initialize and enable EISA slot 15	E2	Setup page 2
30	Size base memory from 256-640K and test	E3	Setup page 3
31	Test extended memory	E4	Setup page 4
32	If EISA mode flag set, test memory found during slot initialization	E5	Setup page 5
3C	Verify CPU switch in and out of protected, virtual 86 and 8086 page modes	E6	Setup page 6
3D	Check for mouse and initialize if present	E7	Setup page 7
3E	Initialize cache controller	E8	Setup page 8
3F	Enable shadow RAM	E9	Setup page 9
41	Initialize floppy drive controller and drives	EA	Setup page 10
42	Initialize hard drive controller and drives	EB	Setup page 11
43	Serial ports detected and initialized	EC	Setup page 12
44	Parallel ports detected and initialized	ED	Setup page 13
45	Coprocessor detected and initialized	EE	Setup page 14
46	Setup message print to screen	EF	Setup page 15
47	Boot speed set	FF	Boot via interrupt 19 if no errors detected

EISA BIOS

Code	Meaning	Code	Meaning
01	CPU Flags	2D	Initialize and enable EISA slot 13
02	CPU Registers	2E	Initialize and enable EISA slot 14
03	Initialize DMA	2F	Initialize and enable EISA slot 15
04	Memory refresh	30	Memory size 256K
05	Keyboard initialization	31	Memory test over 256K
06	ROM checksum	32	EISA memory
07	CMOS	3C	CMOS setup on
08	256K memory	3D	Mouse
09	Cache	3E	Cache RAM
0A	Set interrupt table	3F	Shadow RAM
0B	CMOS checksum	41	Initialize floppy drive controller and drives
0C	Keyboard initialization	42	Initialize hard drive controller and drives
0D	Video adapter	43	Serial ports detected and initialized
0E	Video memory	45	Coprocessor detected and initialized
0F	Test DMA controller 0	47	Boot speed set
10	Test DMA controller 1	4E	Manufacturing loop
11	DMA page registers	4F	Security check
14	Timer chip	50	Write all CMOS values back to CMOS RAM
15	Programmable interrupt controller 1	51	Enable NMI
16	Programmable interrupt controller 2	52	Adapter ROMs
17	Programmable interrupt controller stuck bits	53	Initialize time value at address 40 of BIOS
18	PIC maskable IRQs	63	Boot attempt
19	NMI bit check	B0	NMI in protected mode

Code	Meaning	Code	Meaning
1F	CMOS RAM	B1	Disable NMI
20	Initialize and enable EISA slot 0	BF	Program chipset
21	Initialize and enable EISA slot 1	C0	Cache on/off
22	Initialize and enable EISA slot 2	C1	Memory size
23	Initialize and enable EISA slot 3	C2	Base 256K test
24	Initialize and enable EISA slot 4	C3	DRAM page select
25	Initialize and enable EISA slot 5	C4	Video switch
26	Initialize and enable EISA slot 6	C5	Shadow RAM
27	Initialize and enable EISA slot 7	C6	Cache program
28	Initialize and enable EISA slot 8	C8	Speed switch
29	Initialize and enable EISA slot 9	C9	Shadow RAM
2A	Initialize and enable EISA slot 10	CA	OEM chipset
2B	Initialize and enable EISA slot 11	FF	Boot via interrupt 19 if no errors detected
2C	Initialize and enable EISA slot 12		

AWARD

The general procedures below are valid for greater than XT v3.0 and AT v3.02-4.2. The sequence may vary slightly between versions. If a failure occurs between 6- FF (unless it causes the computer to hang in the test), the system will keep outputting the POST sequence to the defined POST port. A normal error message will then be displayed on the screen when video is available. EISA codes typically go to 300h. ISA codes to 80h.

Award Test Sequence-up to v4.2

Procedure	Meaning
CPU	BIOS sets verifies and resets the error flags in the CPU (i.e. carry; sign; zero; stack overflow). Failure here is normally due to the CPU or system clock.
POST Determination	BIOS determines whether motherboard is set for normal operation or a continuous loop of POST (for testing). If the POST test is cycled 1-5 times over and over either the jumper for this function is set to burn-in or the circuitry involved has failed.
Keyboard Controller	BIOS tests internal operations of the keyboard controller chip (8042). Failure here is normally due to the keyboard chip.
Burn In Status	1-5 will repeat if the motherboard is set to burn in (you will see the reset light on all the time). If you haven't set the board for burn-in mode, there is a short in the circuitry.
Initialise Chipset	BIOS clears all DMA registers and CMOS status bytes 0E & 0F. BIOS then initialises 8254 (timer). Failure of this test is probably due to the timer chip.
CPU	A bit-pattern is used to verify the functioning of the CPU registers. Failure here is normally down to the CPU or clock chip.
RTC	Verifies that the real time clock is updating CMOS at normal intervals. Failure is normally the CMOS/RTC or the battery.
ROM BIOS Checksum	BIOS performs a checksum of itself against a predetermined value that will equal 00. Failure is down to the ROM BIOS.
Initialise Video	BIOS tests and initialises the video controller. Failure is normally the video controller (6845) or an improper setting of the motherboard or CMOS.
PIT	BIOS tests channels 0 1 2 in sequence. Failure is normally the PIT chip (8254/53).
CMOS Status	Walking-bit pattern tests CMOS shutdown status byte 0F. Failure normally in CMOS.

Procedure	Meaning
Extended CMOS	BIOS checks for extended information of chipset and stores it in the extended RAM area. Failure is normally due to invalid information and can be corrected by setting CMOS defaults. Further failure indicates either the chipset or the CMOS RAM.
DMA	Channels 0 and 1 are tested together with the page registers of the DMA controller chip(s)-8237. Failure is normally due to the DMA chips.
Keyboard	The 8042 keyboard controller is tested for functionality and for proper interfacing functions. Failure is normally due to the 8042 chip.
Refresh	Memory refresh is tested; the standard refresh period is 120-140 ns. Failure is normally the PIT chip in ATs or the DMA chip in XTs.
Memory	The first 64K of memory is tested with walking-bit patterns. Failure is normally due to the first bank of RAM or a data line.
Interrupt Vectors	BIOS interrupt vector table loaded to first bank of RAM. Failure here is not likely since memory in this area has been tested. If a failure does occur suspect the BIOS or the PIC.
Video ROM	Video ROM is initialised which performs an internal diagnostic before returning control to the System BIOS. Failure is normally the video adapter or the BIOS.
Video Memory	This is tested with a bit-pattern. This is bypassed if there is a ROM on the video adapter. Failure is normally down to the memory on the adapter.
PIC	Functionality of interrupt controller chip(s) is tested (8259). Failure is normally down to the 8259 chips but may be the clock.
CMOS Battery	BIOS verifies CMOS byte 0D is set which indicates CMOS battery power. Suspect the battery first and the CMOS second.
CMOS Checksum	A checksum is performed on the CMOS. Failure is either incorrect setup or CMOS chip or battery. If the test is passed the information is used to configure the system.
Determine System Memory	Memory up to 640K is addressed in 64K blocks. Failure is normally due to an address line or DMA chip. If all of the memory is not found there is a bad RAM chip or address line in the 64K block above the amount found.
Memory Test	Tests are performed on any memory found and there will normally be a message with the hex address of any failing bit displayed at the end of boot.
PIC	Further testing is done on the 8259 chips.
CPU protected mode	Processor is placed into protected mode and back into real mode; the 8042 is used for this. In case of failure suspect the 8042; CPU; CMOS; or BIOS in that order.
Determine Extended Memory	Memory above 1 Mb is addressed in 64K blocks. The entire block will be inactive if there is a bad RAM chip on a block.
Test Extended Memory	Extended memory is tested with a series of patterns. Failure is normally down to a RAM chip, and the hex address of the failed bit should be displayed.
Unexpected Exceptions	BIOS checks for unexpected exceptions in protected mode. Failure is likely to be a TSR or intermittent RAM failure.
Shadow/ Cache	Shadow RAM and cache activated; failure may be cache controller or chips. Check CMOS first for invalid information.
8242 Detection	BIOS checks for Intel 8242 keyboard controller and initialises. Failure may be an improper jumper setting or the 8242.
Initialise Keyboard	Failure could be the keyboard or the controller.
Initialise Floppy	All those set in the CMOS. Failure could be incorrect setup or controller or the drive.
Detect Serial Ports	BIOS searches for and initialises up to 4 serial ports at 3F8/2F8/3E8 and 2E8. Detection failure is normally due to an incorrect jumper setting somewhere or an adapter failure.
Detect Parallel Ports	BIOS searches for and initialises up to four parallel ports at 378/3BC and 278. Detection failure is normally due to an incorrect jumper setting somewhere or an adapter failure.
Initialise Hard Drive	BIOS initialises any hard drive set in CMOS. Failure could be due to invalid CMOS setup, hard drive or controller.
Detect NPU Coprocessor	Initialisation of any NPU Coprocessor found. Failure is due to either an invalid CMOS setup or the NPU is failing.
Initialise Adapter ROM	Any adapter ROMs between C800 and EFFF are initialised. The ROM will do an internal test before giving back control to the System ROM. Failure is normally due to the adapter ROM or the attached hardware.

Procedure	Meaning
Initialise External Cache	Any cache external to the 486 is enabled. Failure would indicate invalid CMOS setup, cache controller or chips.
NMI Unexpected Exceptions	A final check for unexpected exceptions before giving control to the Int 19 boot loader. Failure is normally due to a memory parity error or an adapter.
Boot Errors	Failure when the BIOS attempts to boot off the default drive set in CMOS is normally due to an invalid CMOS drive setup or as given by an error message. If the system hangs there is an error in the Master Boot Record or the Volume Boot Record.

Award Test Sequence-after v4.2 (386/486)

Procedure	Meaning
CPU	BIOS sets verifies and resets the error flags in the CPU then performs a register test by writing and reading bit patterns. Failure is normally due to the CPU or clock chip.
Initialise Support Chips	Video is disabled as is parity/DMA and NMI. Then the PIT/PIC and DMA chips are initialised. Failure is normally down to the PIT or DMA chips.
Init Keyboard	Keyboard and Controller are initialised.
ROM BIOS Test	A checksum is performed by the ROM BIOS on the data within itself and is compared to a preset value of 00. Failure is normally due to the ROM BIOS.
CMOS Test	A test of the CMOS chip which should also detect a bad battery. Failure is due to either the CMOS chip or the battery.
Memory Test	First 356K of memory tested with any routines in the chipsets. Failure normally due to defective memory.
Cache Initialisation	Any cache external to the chipset is activated. Failure is normally due to the cache controller or chips.
Initialise Vector Table	Interrupt vectors are initialised and the interrupt table is installed into low memory. Failure is normally down to the BIOS or low memory.
CMOS RAM	CMOS RAM checksum tested, BIOS defaults loaded if invalid. Check CMOS RAM.
Keyboard Init	Keyboard initialised and Num Lock set On. Check the keyboard or controller.
Video Test	Video adapter tested and initialised.
Video Memory	Tested on Mono and CGA adapters. Check the adapter card.
DMA Test	DMA controllers and page registers are tested. Check the DMA chips.
PIC Tests	8259 PIC chips are tested.
EISA Mode Test	A checksum is performed on the extended data area of CMOS where EISA information is stored. If passed the EISA adapter is initialised.
Enable Slots	Slots 0-15 for EISA adapters are enabled if the above test is passed.
Memory Size	Memory addresses above 265K written to in 64K blocks and addresses found are initialised. If a bit is bad, entire block containing it and those above will not be seen
Memory Test	Read/Write tests performed to memory over 256K; failure due to bad bit in RAM.
EISA Memory	Memory tests on any adapters initialised previously. Check the memory chips.
Mouse Initialisation	Checks for a mouse and installs the appropriate interrupt vectors if one is found. Check the mouse adapter if you get a problem.
Cache Init	The cache controller is initialised if present.
Shadow RAM Setup	Any Shadow RAM present according to the CMOS Setup is enabled.
Floppy Test	Test and initialise floppy controller and drive.
Hard Drive Test	Test and initialise hard disk controller and drive. You may have an improper setup or a bad controller or hard drive.
Serial/Parallel	Any serial/parallel ports found at the proper locations are initialised.
Maths Copro	Initialised if found. Check the CMOS Setup or the chip.
Boot Speed	Set the default speed at which the computer boots.
POST Loop	Reboot occurs if the loop pin is set; for manufacturing purposes.
Security	Ask for password if one has been installed. If not check the CMOS data or the chip.

Procedure	Meaning
Write CMOS	The BIOS is waiting to write the CMOS values from Setup to CMOS RAM. Failure is normally due to an invalid CMOS configuration.
Pre-Boot	BIOS is waiting to write the CMOS values from Setup to CMOS RAM.
Adapter ROM Initialise	Adapter ROMs between C800 and EFFF are initialised. The ROM will do an internal test before giving back control to the System ROM. Failure is normally due to the adapter ROM or the attached hardware.
Set Up Time	Set CMOS time to the value located at 40h of the BIOS data area.
Boot System	Control is given to the Int 19 boot loader.

3.0x

Uses IBM beep patterns. Version 3.xx sends codes 1-24 to port 80 and 300 and the system hangs up. Afterwards, codes are sent to the POST port and screen without hanging up.

Code	Meaning
01	CPU test 1: verify CPU status bits
02	Powerup check-Wait for chips to come up; initialize motherboard and chipset (if present) with defaults. Read 8042 status and fail if its input buffer contains data but output buffer does not.
03	Clear 8042 interface-send self-test command AA, fail if status not 2 output buffer full.
04	Reset 8042 Keyboard controller-fail if no data input (status not equal 1) within a million tries, or if input data is not 55 in response to POST 03.
05	Get 8042 manufacturing status-read video type and POST type bits from 8042 discrete input port; test for POST type = manufacturing test or normal; fail if no response from 8042.
06	Initialize on-board chips-disable colour & mono video, parity, and 8237 DMA; reset 80x87 math chip, initialize 8255 timer 1, clear DMA chip and page registers and CMOS RAM shutdown byte: initialize motherboard chipset.
07	CPU test 2: read/write/verify registers except SS, SP, BP with FF and 00 data
08	Initialize CMOS RAM/RTC chip-update timer cycle normally; disable PIE, AIE, UIE and square wave. Set BCD date and 24-hour mode.
09	Checksum 32K of BIOS ROM; fail if not 0
0A	Initialize video interface-read video type from 8042 discrete input port. Fail if can't read it. Initialize 6845 controller register at either colour or mono adapter port to 80 columns, 25 rows, 8/14 scan lines per row, cursor lines at 6/11 (first) & 7/12 (last), offset to 0.
0B	Test 8254 timer channel 0- this test is skipped; already initialized for mode 3.
0C	Test 8254 timer channel 1-this test is skipped; already initialized for mode 0.
0D	Test 8254 timer channel 2-write/read/verify FF, then 00 to timer registers; ini with 500h for normal ops.
0E	Test CMOS RAM shutdown byte (3.03: CMOS date and timer-this test is skipped and functions performed
0F	Test extended CMOS RAM if present (3.03: test CMOS shutdown byte-write/read/verify a walk-to-left l pattern at CMOS RAM address 8F)
10	Test 8237 DMA controller ch 0-write/read/verify pattern AA, 55, FF and 00.
11	Test 8237 DMA controller ch 1-write/read/verify pattern AA, 55, FF and 00.
12	Test 8237 DMA controller page registers-write/read/verify pattern AA, 55, FF and 00: use port addresses to check out address circuitry to select page registers. At this point, POST enables user reboot.
13	Test 8741 keyboard controller interface-read 8042 status, verify buffers are empty, send AA self-test command, verify 55 response, send 8741 write command to 8042, wait for 8042 acknowledgement, send 44 data for 8741 (keyboard enabled, system flag, AT interface), wait for ack, send keyboard disable command, wait for ack. Fail if no ack or improper responses.
14	Test memory refresh toggle circuits-fail if not toggling high and low.
15	Test first 64K of base memory-disable parity checking, zero memory, 64K at a time, to clear parity errors, enable parity checking, write/read/verify 00, 5A, FF and A5 at each address.
16	Set up interrupt vector tables in low memory.

Code	Meaning
17	Set up video I/O operations-read 8042 (motherboard switch or jumper) to find whether colour or mono adapter installed; validate by writing a pattern to mono memory B0000 and select mono I/O port if OK or colour if not, and initialize it via setting up the hardware byte and issuing INT 10. Then search for special video adapter BIOS ROM at C0000 (EGA/VGA), and call it to initialize if found. Fail if no 8042 response.
18, 1 beep	Test MDA/CGA video memory unless EGA/VGA adapter is found-disable video, detect mono video RAM at B0000 or colour at B8000, write/read/verify test it with pattern A5A5, fill it with normal attribute, enable the video card. No error halt unless enabled by CMOS. Beep once to let user know first phase of testing complete. From now on, POST will display test and error messages on screen.
19	Test 8259 PIC mask bits, channel 1-write/read/verify 00 to mask register.
1A	Test 8259 PIC mask bits, channel 2-write/read/verify 00 to mask register.
1B	Test CMOS RAM battery level-poll CMOS RTC/RAM chip for battery level status. Display error if level is low, but do not halt.
1C	Test CMOS RAM checksum-check battery level again, calculate checksum of normal and extended CMOS RAM. Halt if low battery or checksum not 0; otherwise reinitialize motherboard chipset if necessary.
1D	Set system memory size parameters from CMOS RAM data, Cannot fail.
1E	Size base memory 64K at a time, and save in CMOS. Cannot fail, but saves diagnostic byte in CMOS RAM if different from size in CMOS.
1F	Test base memory found from 64K to 640K-write/read/verify FFAA and 5500 patterns by byte. Display shows failing address and data.
20	Test stuck bits in 8259 PICs
21	Test for stuck NMI bits (parity /I0 check)
22	Test 8259 PIC interrupt functionality-set up counter timer 0 to count down and issue an interrupt on IRQ8. Fail if interrupt does not occur.
23	Test protected mode, A20 gate. and (386 only) virtual 86 & 8086 page mode.
24	Size extended memory above 1Mb; save into CMOS RAM. Cannot fail, but saves diagnostic byte in CMOS RAM if different from CMOS.
25	Test all base and extended memory found (except the first 64K) up to 16 Mb. Disable parity check but monitor for parity errors. Write/read/verify AA55 then 55AA pattern 64K at a time. On 386 systems use virtual 8086 mode paging system. Displays actual and expected data and failing address.
26	Test protected mode exceptions-creates the circumstances to cause exceptions and verifies they happen; out-of-bounds instruction, invalid opcode, invalid TSS (JMP, CALL, IRET, INT), segment not present on segment register instruction, generate memory reference fault by writing to a read-only segment.
27	Initialise shadow RAM and move system BIOS and/or video BIOS into it if enabled by setup. Also (386 only) initialise the cache controller if present in system. This is not implemented in some versions of 3.03
28	Detect and initialise Intel 8242/8248 chip (not implemented in 3.03)
29	Reserved
2A	Initialise keyboard
2B	Detect and initialise floppy drive
2C	Detect and initialise serial ports
2D	Detect and initialise parallel ports
2E	Detect and initialise hard drive
2F	Detect and initialise math coprocessor
30	Reserved
31	Detect and initialise adapter ROMs
BD	Initialize Orvonton cache controller if present
CA	Initialize 386 Micronics cache if present
CC	Shutdown NMI handler
EE	Test for unexpected processor exception
FF	INT 19 boot

3.00-3.03 8/26/87

Code	Meaning
01	Processor test part 1; Processor status verification. Tests following CPU status flags: set/clear carry zero sign and overflow (fatal). Output: infinite loop if failed; continue test if OK. Registers: AX/BP.
02	Determine POST test. Manufacturing (e.g. 01-05 in loop) or normal (boot when POST finished). Fails if keyboard interface buffer filled with data. Output: infinite loop if failed; continue if OK. Registers: AX/BX/BP.
03	Clear 8042 keyboard interface. Send verify TEST_KBRD command (AAh). Output: infinite loop if failed; continue test if OK. Registers: AX/BX/BP.
04	Reset 8042 keyboard controller. Verify AAh return from 03. Infinite loop if test fails.
05	Get 8042 keyboard controller manufacturing status. Read input port via keyboard controller to determine manufacturing or normal mode operation. Reset system if manufacturing status from 02. Output: infinite loop if failed; continue test if OK. Registers: AX/BX/BP.
06	Init chips on board LSI chips. Disable colour/mono video; parity and DMA (8237A). Reset coprocessor; initialise (8254) timer 1; clear DMA page registers and CMOS shutdown byte.
07	Processor test #2. read/write verify SS/SP/BP registers with FFh and 00h data pattern.
08	Initialize CMOS chip
09	EPROM checksum for 32 Kbytes
0A	Initialize video interface
0B	Test 8254 channel 0
0C	Test 8254 channel 1
0D	Test 8254 channel 2
0E	Test CMOS date and timer
0F	Test CMOS shutdown byte
10	Test DMA channel 0
11	Test DMA channel 1
12	Test DMA page registers
13	Test 8741 keyboard controller
14	Test memory refresh toggle circuits
15	Test 1st 64k bytes of system memory
16	Setup interrupt vector table
17	Setup video I/O operations
18	Test video memory
19	Test 8259 channel 1 mask bits
1A	Test 8259 channel 2 mask bits
1B	Test CMOS battery level
1C	Test CMOS checksum
1D	Setup configuration byte from CMOS
1E	Sizing system memory & compare w/CMOS
1F	Test found system memory
20	Test stuck 8259'S interrupt bits
21	Test stuck NMI (parity/IO chk) bits
22	Test 8259 interrupt functionality
23	Test protected mode and A20 gate
24	Sizing extended memory above 1MB
25	Test found system/extended memory
26	Test exceptions in protected mode

286 N3.03 Extensions

Code	Meaning
2A	POST_KEYBOARD present during reset keyboard before boot has no relationship to POST 19.
2B	POST_FLOPPY present during init of floppy controller and drive(s)
2C	POST_COMM present during init of serial cards.
2D	POST_PRN present during init of parallel cards
2E	POST_DISK present during init of hard disk controller and drive(s)
2F	POST_MATH present during init of coprocessor. Result remains after DOS boot; left on the port 80 display
30	POST_EXCEPTION present during protected mode access or when processor exceptions occur. A failure indicates that protected mode return was not possible
CC	POST_NMI present when selecting the F2 system halt option

Original XT

Code	Meaning	Code	Meaning
03	Flag resister test	30	V40 DMA if present
06	CPU register test	33	Verify system clock initialization
09	System hardware initialization	36	Keyboard test
0C	BIOS checksum	39	Setup interrupt table
0F	DMA page register initialization	3C	Read system configuration switches
12	Test DMA address and count registers	3F	Video test
15	DMA initialization	42	Serial port determination
18	Timer test	45	Parallel port determination
1B	Timer initialization	48	Game port determination
1E	Start RAM initialization	4B	Copyright message display
21	Test base 64K of RAM	4E	Calculation of CPU speed
24	Setup init. and temp stack	54	Test of system memory
27	Initialize PIC	55	Floppy drive test
2A	Interrupt mask register test	57	System initialized before boot
2D	Hot interrupt test	5A	Call to Int 19

XT 8088/86 BIOS v3.1

Code	Meaning
01	Processor test 1; processor status verification. Tests the following processor status flags, carry, zero, sign, overflow. The BIOS will set each flags, verify they are set, then turn each flag off and verify it is off. Failure of any flag will cause a fatal error.
02	Determine type of POST test, manufacturing or normal, which can be set by a jumper on some motherboards. If the status is normal, POST continues through and, assuming no errors, boot is attempted. If manufacturing, POST will run in continuous loop and boot will not be attempted.Failed if keyboard interface buffer filled with data.
03	Clear 8042 Keyboard Controller-Test by sending TEST_KBRD command (AAh) and verifying controller reads command. Reset Keyboard Controller then verifiy controller returns Aah.
04	Get Manufacturing Status
05	The last test in the manufacturing cycle. If test 2 found the status to be manufacturing, this POST will trigger a reset and POSTs 1-5 will be repeated continuously.
06	Init 8259 PIC and 8237 DMA controller chips. Disable colour and mono video, parity circuits and DMA chips. Reset math coprocessor. Initialise 8253 Timer channel 1. Clear DMA chip and page registers.
07	Processor test #2. Write, read and verify all registers except SS, SP and BP with data patterns 00 and FF.

Code	Meaning
08	Initialize CMOS Timer. Update timer cycle normally
09	EPROM checksum for 32 Kbytes. Test failed if sum not equal to 0. Also checksums the sign-on message.
0A	Initialize video controller 6845 registers as follows: 25 lines x 80 columns, first cursor scan line at 6/11 and last at 7/12, reset display offset to 0.
0B	Test Timer (8254) Channel 0. These three timer tests verify that the 8254 timer chip is functioning properly.
0C	Test Timer (8254) Channel 1
0D	Test Timer (8254) Channel 2
0E	Test CMOS Shutdown Byte. Use walking bit (1) algorithm to check interface to CMOS circuit.
0F	Test Extended CMOS and Initialize CHIPSET. On motherboards with chip sets that support extended CMOS configurations, such as Chips and Technologies, the BIOS tables of CMOS information configure the chip set. These chip sets have an extended storage mechanism that allows you to save a system configuration after power is turned off. A checksum verifies the validity of the extended storage and, if valid, permits the information to be loaded into extended CMOS RAM.
10	Test DMA Channel 0. These 3 three functions initialize the DMA chip and then test it using an AA, 55, FF, 00 pattern. Port addresses are used to check the address circuit to DMA page registers.
11	Test DMA Channel 1. Test DMA Page Registers.
12	Test DMA Page Registers.
13	Test Keyboard Controller. Test keyboard controller interface.
14	Test Memory Refresh.
15	Test 1st 64K of system memory. An extensive parity test is performed on the first 64K of system memory. This memory is used by the BIOS
16	Setup interrupt vector table in 1st 64K
17	Setup video I/O operations. If a CGA or MDA adapter is installed, the video is initialized by the system BIOS. If the system BIOS detects an EGA or VGA adapter, the option ROM BIOS installed on the video adapter is used to initialize and set up the video.
18	Test video memory for CGA and MDA video boards. This is not performed by the system BIOS on EGA or VGA video adapters-the board's own EGA or VGA BIOS will ensure that it is functioning properly.
19	Test 8259 channel 1 mask bits. These two tests verify 8259 masked interrupts by alternately turning off and on the interrupt lines. Unsuccessful completion will generate a fatal error.
1A	Test 8259 channel 2 mask bits.
1B	Test CMOS Battery Level. Verifies that the battery status bit is set to ``1''. A ``0'' can indicate a bad battery or some other problem, such as bad CMOS.
1C	Set Configuration from CMOS. If the checksum is good, the values are used to configure the system.
1D	Test CMOS Checksum. This function tests the CMOS checksum data (located at 2Eh and 2Fh), and Extended CMOS checksum, if present, to be sure they are valid.
1E	Size System Memory. The system memory size is determined by writing to addresses from 0K to 640K, starting at 0 and continuing until an address does not respond. This tells the BIOS that this is the end of the memory. This value is then compared to the CMOS value to ensure they are the same. If they are different a flag is set and at the end of POST an error message is displayed.
1F	Test found system memory. Tests from 64K to the top of memory found by writing the pattern FFAA and 5500 then reading the pattern back, byte by byte, and verifying that it is correct
20	Test stuck 8259's interrupt bits
21	Test stuck NMI (parity/IO chk) bits
22	Test 8259 interrupt functionality
23	Test Protected Mode. Verifies protected mode, 8086 virtual mode as well as 8086 page mode. Protected mode ensures that any data about to be written to extended memory (above 1MB) is checked to ensure that it is suitable for storage there.
24	Size Extended Memory. This function sizes memory above 1MB by writing to addresses starting at 1MB and continuing to 16MB on 286 and 386SX systems and 64MB on 386 systems until there is no response. This determines the total extended memory, which is compared with CMOS to ensure the values are the same. If they are different a flag is set and at the end of POST an error message is displayed.
25	Test Found Extended Memory using virtual 8086 paging mode and writing an FFFF, AA55, 0000 pattern.
26	Test Protected Mode Exceptions.

Code	Meaning
27	Setup Cache Control or Shadow RAM. Tests for Shadow RAM (286, 386SX, 386, and 486) and cache controller (386 and 486 only) functionality. Systems with CGA and MDA adapters will indicate that Video Shadow RAM is enabled, even though there is no BIOS ROM to shadow. This is normal.
28	Setup 8242. Optional 8242/8248 Keyboard Controller detection and support.
29	Reserved.
2A	Initialise keyboard
2B	Initialise floppy controller and drive
2C	Initialise COM ports
2D	Initialised LPT ports
2E	Initialize Hard Drive & Controller.
2F	Initialise maths coprocessor
30	Reserved.
31	Initialise option ROMs
3B	Initialize Secondary Cache w/ OPTi chip set
FF	Int 19 Boot attempt

Modular (386) BIOS v3.1

Also for PC/XT v3.0+ and AT v3.02+. Tests do not necessarily execute in numerical order.

Code	Meaning
01	Processor test part 1. Processor status verification. Tests the following processor-status flags: set/clear carry; zero; sign and overflow (fatal). BIOS sets each flag; verifies they are set and turns each flag off verifying its state. Failure of a flag means a fatal error. Output: infinite loop if failed; continue test if OK. Registers: AX/BP.
02	Determine POST type; whether normal (boot when POST finished) or manufacturing (run 01-05 in loop) which is often set by a jumper on some motherboards. Fails if keyboard interface buffer filled with data. Output: infinite loop if failed; continue test if OK. Registers: AX/BX/BP.
03	Clear 8042 keyboard interface. Send verify TEST_KBRD command (AAh). Output: infinite loop if failed; continue test if OK. Registers: AX/BX/BP.
04	Reset 8042 keyboard controller. Verify AAh return from 03. Infinite loop if test fails. Registers: AX/BX/BP.
05	Get 8042 keyboard controller manufacturing status; read input port via keyboard controller to determine manufacturing or normal mode operation. Reset system if manufacturing; i.e. if 02 found the status to be Manufacturing triggers a reset and 01-05 are repeated continuously. Output: infinite loop if failed; continue test if OK. Registers: AX/BX/BP.
06	Initialise chips on board LSI chips. Disables colour and mono video/parity circuits/DMA (8237) chips; resets maths copro; initialises timer 1 (8255); clears DMA chip and page registers and the CMOS shutdown byte.
07	Processor Test 2. Reads writes and verifies CPU registers except SS/SP/BP with data pattern FF and 00.
08	Initialises CMOS timer/RTC and updates timer cycle; normally CMOS (8254) timer; (8237A) DMA; (8259) interrupt and EPROM.
09	EPROM Checksum; test fails if not equal to 0. Also checksums sign-on message.
0A	Initialise Video Interface; specifically register 6845 to 80 characters per row and 25 rows per screen and 8/14 scan lines per row for mono/colour; first scan line of cursor 6/11; last scan line of cursor 7/12; reset display offset to 0.
0B	Test Timer (8254) Channel 0. See also below.
0C	Test Timer (8254) Channel 1.
0D	Test Timer (8254) Channel 2.
0E	Test CMOS Shutdown Byte using a walking-bit algorithm.
0F	Test Extended CMOS. On motherboards supporting extended CMOS configuration the BIOS tables of CMOS information configure the chipset which has an extended storage facility enabling you to keep the configuration with the power off. A checksum is used for verification.
10	Test DMA Channel 0. This and next two tests initialise the DMA chip and test it with an AA/55/FF/00 pattern. Port addresses used to check address circuit to DMA page circuit registers.

Code	Meaning
11	DMA Channel 1
12	DMA Page Registers
13	Test keyboard controller interface.
14	Test memory refresh toggle circuits.
15	First 64K of system memory which is used by the BIOS; an extensive parity test.
16	Interrupt Vector Table. Sets up and loads interrupt vector tables in memory for the 8259 PIC.
17	Video I/O operations. Initialises the video; EGA and VGA ROMs are used if present.
18	Video memory test for CGA and mono cards (EGA and VGA have their own procedures).
19	Test 8259 mask bits-Channel 1.Interrupt lines turned alternately off and on. Failure is fatal.
1A	8259 Mask Bits-Channel 2
1B	CMOS battery level; verifies battery status bit set to 1. 0 could indicate bad battery at CMOS.
1C	Tests the CMOS checksum data at 2E and 2Fh and extended CMOS checksum if present.
1D	Configuration of the system from CMOS values if the checksum is good.
1E	System memory size is determined by writing to addresses from 0-640K continuing till there is no response. The size is then compared to the CMOS and a flag set if they do not compare. An error message will then be displayed.
1F	Tests memory from the top of 64K to the top of memory found by writing patterns FFAA and 5500 and reading them back byte by byte
20	Stuck 8259 Interrupt Bits.
21	Stuck NMI bits (parity or I/O channel check).
22	8259 function.
23	Verifies protected mode; 8086 virtual and page mode.
24	As for 1E but for extended memory from 1-16Mb on 286/386SX systems and 64 Mb on 386s and above. The value found is compared to the CMOS settings.
25	Tests extended memory above using virtual 8086 paging mode and writing an FFFF/AA55/0000 pattern.
26	Protected Mode Exceptions; tests other aspects of protected mode operations.
27	Tests cache control (386/486) or Shadow RAM. Systems with CGA and MDA indicate that video shadow RAM is enabled even though there is no BIOS ROM to shadow.
28	Set up cache or 8242 keyboard controller. Optional Intel 8242/8248 controller detection and support.
29	Reserved.
2A	Initialise keyboard and controller.
2B	Initialise floppy drive(s) and controller.
2C	Detect and initialise serial ports.
2D	Detect and initialise parallel ports.
2E	Initialise hard drive and controller.
2F	Detect and initialise maths coprocessor.
30	Reserved.
31	Detect and initialise option ROMs. Initialises any between C800-EFFF.
3B	Initialise secondary cache with OPTi chipset (486 only).
CC	NMI Handler Shutdown. Detects untrapped NMIs during boot.
EE	Unexpected Processor Exception.
FF	Boot Attempt; if POST is complete and all components are initialised with no errors.

ISA/EISA BIOS v4.0

EISA codes may be sent to 300h.

Code	Meaning
01	Processor test 1: Verify CPU status flags-set, test, clear, and test the carry, zero, sign, overflow flags (fatal)
02	Processor test 2: Write/read/verify all CPU registers, except SS, SP and BP with data patterns FF and 00.

Code	Meaning
03	Calculate BIOS EPROM and sign-on message checksum; fail if not 0
04	Test CMOS RAM interface and verify battery power Is available.
05	Initialize chips: Disable NMI, PIE, AIE, UEI, SQWV; disable video, parity checking, and DMA: reset math coprocessor, clear all page registers and CMOS RAM shutdown byte: Initialize timers 0, 1 and 2, and set EISA timer to a known state: initialize DMA controllers 0 and 1: initialize interrupt controllers 0 and 1; initialise EISA extended registers.
06	Test memory refresh toggle to ensure memory chips can retain data.
07	Set up low memory; Initialize chipset early; test presence of memory; run OEM chipset initialization routines, clear lower 256K memory; enable parity checking and test in lower 256K; test lower 256K.
08	Setup interrupt vector table; initialize first 120 interrupt vectors with SPURIOUS_INT_HDLR and initialize INT 00-1F according to INT_TBL.
09	Test CMOS RAM checksum and load default; if checksum is bad.
0A	Initialize keyboard; detect type of keyboard controller (optional); set NumLock status.
0B	Initialize video interface; read CMOS RAM location 14 to find out type of video in use; detect and initialise the video adapter.
0C	Test video memory; write signon message to screen.
0D	OEM specific-initialise motherboard special chips as required by OEM; initialise cache controller early, when cache is separate from chipset.
0F	Test DMA controller 0 with AA, 55, FF, 00 pattern.
10	Test DMA controller 1 with AA, 55, FF, 00 pattern.
11	DMA page registers-use 1/O ports to test address circuits.
14	Test 3254 timer 0 counter 2.
15	Verify 8259 interrupt controller channel 1 by toggling interrupt lines off/on.
16	Verify 8259 interrupt controller channel 2 by toggling interrupt lines off/on.
17	Test stuck 8259 interrupt bits: turn interrupt bits off and verify no interrupt mask register is on.
18	Test 8259 functionality: force an interrupt and verify the interrupt occurred.
19	Test stuck NMI bits (parity I/O check): verify NMI can be cleared.
1F	Set EISA mode: If EISA non-volatile memory checksum is good, execute EISA init. If not, execute ISA tests and clear EISA mode flat. Test EISA config mem checksum and communication ability.
20	Initialize and enable EISA slot 0 (system board).
21-2F	Initialize and enable EISA slots 1-15.
30	Size base memory from 256-640K and test with various patterns.
31	Test extended memory above 1Mb using various patterns. Press Esc to skip.
32	If EISA mode flag set, test EISA memory found during slot initialization. Skip this by pressing Esc.
3C	Verify CPU can switch in/out of protected, virtual 86 and 8086 page modes.
3D	Detect if mouse is present, initialize it, and install interrupt vectors.
3E	Initialize cache controller according to CMOS RAM setup
3F	Enable shadow RAM according to setup or if MEM TYPE is SYS in the EISA configuration information.
41	Initialise floppy disk drive controller and any drives.
42	Initialise hard disk drive controller and any drives.
43	Detect and initialise serial ports.
44	Detect and initialize parallel ports.
45	Detect and initialise math coprocessor
46	Print Setup message (press Ctrl-Alt-Esc to enter Setup at bottom of the screen, and enable setup.
47	Set speed for boot.
4E	Reboot if manufacturing POST loop pin is set. Otherwise, display messages for non-fatal POST errors; setup if user pressed Ctrl-Alt-Esc.
4F	Security check (optional): Ask for password.
50	Write all CMOS RAM values back to CMOS RAM, and clear the screen.
51	Preboot enable: Enable parity, NMI, cache before boot.
52	Initialize ROMs between C80000-EFFFF. When FSCAN enabled, init from C80000 to F7FFF.

Code	Meaning
53	Initialize time value at address 40 of BIOS RAM area.
55	Initialize DDNIL counter to NULLs.
63	Boot attempt: Set low stack and boot by calling INT 19.
88	CPU failed to initialise
B0	Spurious interrupt occurred in protected mode.
B1	Unclaimed NMI. If unmasked NMI occurs, display Press F1 to disable NMI, F2 to boot.
BF	Program chipset: Called by POST 7 to program chipset from CT table.
C0	OEM specific-Turn on/off cache.
C1	OEM specific-Test for memory presence and size on-board memory.
C2	OEM specific-Initialize board and turn on shadow and cache for fast boot.
C3	OEM specific-Turn on extended memory DRAM select and initialize RAM.
C4	OEM specific-Handle display/video switch to prevent display switch errors.
C5	OEM specific-Fast Gate A20 handling.
C6	OEM specific-Cache routine for setting regions that are cacheable.
C7	OEM specific-Shadow video/system BIOS after memory proven good.
C8	OEM specific-Handle special speed switching.
C9	OEM specific-Handle normal shadow RAM operations.
D0-DF	Debug: available POST codes for use during development.
E1-EF	Setup pages: E1 = page 1, E2 = page 2, etc.
FF	If no error flags such as memory size are set, boot via INT 19-load system from drive A, then C; display error message if boot device not found.

EISA BIOS

Code	Meanings	Code	Meaning
1	CPU flags	2D	Slot 13
2	CPU registers	2E	Slot 14
3	Initialise DMA	2F	Slot 15
4	Memory refresh	30	Memory size 256K
5	Keyboard initialisation	31	Memory test over 256K
06	ROM checksum	32	EISA memory
07	CMOS	3C	CMOS setup on
08	256K memory	3D	Mouse
09	Cache	3E	Cache RAM
0A	Set interrupt table	3F	Shadow RAM
0B	CMOS checksum	40	N/A
0C	Keyboard initialisation	41	Floppy drive
0D	Video adapter	42	Hard drive
0E	Video memory	43	RS232/parallel
0F	DMA channel 0	45	NPU
10	DMA channel 1	47	Speed
11	DMA page register	4E	Manufacturing loop
14	Timer chip	4F	Security
15	PIC controller 1	50	CMOS update
16	PIC controller 2	51	Enable NMI
17	PIC stuck bits	52	Adapter ROMs
18	PIC maskable IRQs	53	Set time
19	NMI bit check	63	Boot
1F	CMOS XRAM	B0	NMI in protected

Code	Meanings	Code	Meaning
20	Slot 0	B1	Disable NMI
21	Slot 1	BF	Chipset program
22	Slot 2	C0	Cache on/off
23	Slot 3	C1	Memory size
24	Slot 4	C2	Base 256K test
25	Slot 5	C3	DRAM page select
26	Slot 6	C4	Video switch
27	Slot 7	C5	Shadow RAM
28	Slot 8	C6	Cache program
29	Slot 9	C8	Speed switch
2A	Slot 10	C9	Shadow RAM
2B	Slot 11	CA	OEM chipset
2C	Slot 12	FF	Boot

4.5x-non PnP

Code	Meaning
C0	Turn Off Chipset Cache; OEM specific cache control
01	Processor Test 1; Status (1Flags) Verification. Tests carry/zero/sign/overflow processor status flags.
02	Processor Test 2; Read/Write/Verify CPU registers except SS/SP and BP with pattern FF and 00.
03	Initialise Chips; Disable NMI/PIE/UEL/SQWV; video; parity checking; DMA; reset maths coprocessor. Clear all page registers and CMOS shutdown byte. Initialise timer 0 1 and 2 including set EISA timer to a known state. Initialise DMA controllers 0 and 1; interrupt controllers 0 and 1 and EISA extended registers.
04	Test Memory Refresh Toggle
05	Blank video; initialise keyboard
06	Reserved
07	Test CMOS Interface and battery status. Detects bad battery. BE and Chipset Default Initialisation. Program chipset registers with power-on BIOS defaults.
C1	Memory Presence Test; OEM specific test to size on-board memory
C5	Early Shadow; OEM specific-enable for fast boot
C6	Cache Presence Test; External cache size detection
08	Setup Low Memory; Early chipset initialisation. Memory presence test. OEM chipset routines. Clear low 64K of memory. Test first 64K memory
09	Early Cache Initialisation. Cyrix CPU Initialisation. Cache Initialisation
0A	Setup Interrupt Vector Table; Initialise first 120 interrupt vectors with SPURIOUS_INT_HDLR and initialise INT 00-FF according to INT_TBL.
0B	Test CMOS RAM Checksum if bad or Insert key depressed; load defaults.
0C	Initialise keyboard; Set NUM LOCK status.
0D	Initialise video interface; Detect CPU Clock. Read CMOS location 14h to find out type of video. Detect and initialise video adapter.
0E	Test Video Memory. Write signon message to screen. Set up Shadow RAM and enable as per Setup.
0F	Test DMA Controller 0. BIOS Checksum Test. keyboard detect and initialisation.
10	Test DMA Controller 1
11	Test DMA Page Registers
12-13	Reserved
14	Test Timer Counter 2. Test 8254 Timer 0 Counter 2
15	Test 8259-1 Mask Bits. Alternately turns on and off interrupt lines.
16	Test 8259-2 Mask Bits. Alternately turns on and off interrupt lines.
17	Test Stuck 8259 interrupt bits. Turn off interrupts then verify no interrupt mask register is on.
18	Test 8259 Interrupt Functionality. Force an interrupt and verify that it occurred.

Code	Meaning
19	Test Stuck NMI Bits (Parity/I/O check). Verify NMI can be cleared.
1A	Display CPU Clock
1B-1E	Reserved
1F	Set EISA Mode. If EISA NVR checksum is good execute EISA initialisation. If not execute ISA tests and clear EISA mode flag. Test EISA configuration memory integrity (checksum and communication interface).
20	Enable Slot 0. Motherboard
21-2F	Enable Slots 1-15
30	Size Base and Extended Memory. From 256-640K and that above 1 Mb.
31	Test Base and Extended Memory. Various patterns are used on that described above. This will be skipped in EISA mode and can be skipped in ISA mode with Esc.
32	Test EISA Extended Memory. If EISA Mode flag is set then test EISA memory found in slots initialisation. This will be skipped in ISA mode and can be skipped in EISA mode with Esc.
33-3B	Reserved
3C	Setup Enabled
3D	Initialise and Install Mouse
3E	Setup Cache Controller
3F	Reserved
BF	Chipset Initialisation. Program registers with Setup values.
40	Display virus protect enable or disable.
41	Initialise floppy drive(s) and controller
42	Initialise hard drive(s) and controller
43	Detect and initialise Serial/Parallel Ports and game port.
44	Reserved
45	Detect and Initialise Maths Coprocessor
46	Reserved
47	Reserved
48-4D	Reserved
4E	Manufacturing POST Loop or Display Messages. Reboot if manufacturing POST Loop Pin is set. Otherwise display any messages (i.e. non-fatal errors during POST) and enter Setup.
4F	Security Check. Ask password (optional)
50	Write CMOS. Write all CMOS values back to RAM and clear screen.
51	Pre-boot Enable. Enable Parity Checker; NMI and cache before boot.
52	Initialise Option ROMs. Between C800-EFFF. When FSCAN is enabled will initialise between C800-F7FF
53	Initialise Time Value In 40h BIOS area.
60	Setup Virus Protect. According to Setup
61	Set Boot Speed
62	Setup NumLock. According to Setup
63	Boot attempt. Set Low Stack. Boot via INT 19
88	CPU failed to initialise
B0	Spurious. If interrupt occurs in protected mode
B1	Unclaimed NMI. If unmasked NMI occurs display Press F1 to disable NMI; F2 reboot
E1-EF	Setup Pages. E1=Page 1; E2=Page 2 etc
FF	Boot

4-5x PnP Elite

Code	Meaning
C0	1.Turn off OEM specific cache, shadow 2.Initialize standard devices with default values: DMA controller (8237) Programmable Interrupt Controller (8259) Programmable Interval Timer (8254) RTC chip
C1	Auto detection of onboard DRAM & Cache
C3	1. Test the first 256K DRAM 2. Expand the compressed codes into temporary DRAM area including compressed BIOS & Option ROMs
C5	Copy BIOS from ROM into E000FFFF shadow RAM so that POST will go faster
01-02	Reserved
03	Initialize EISA registers (EISA BIOS only)
04	Reserved
05	1. Keyboard Controller Self Test 2. Enable Keyboard Interface
07	Verifies CMOS's basic R/W functionality
BE	Program defaults values into chipset according to the MODBINable Chipset Default Table
09	1.Program configuration register of Cyrix CPU according to the MODBINable Cyrix Register Table 2.OEM specific cache initialization
0A	1.Initialize the first 32 interrupt vectors with corresponding interrupt handlers. Initialize INT No from 33120 with Dummy (Spurious) interrupt handler 2.Issue CPUID instruction to identify CPU type 3.Early Power Management initialization (OEM specific)
0B	1.Verify the RTC time is valid or not 2.Detect bad battery 3.Read CMOS data into BIOS stack area 4.PnP init including (PnP BIOS only) . Assign CSN to PnP ISA card. Create resource map from ESCD 5.Assign IO & Memory for PCI devices (PCI BIOS only)
0C	Initialization of the BIOS data area (40:040:FF)
0D	1.Program some chipset's value according to setup.(Early setup value program). 2.Measure CPU speed for display & decide the system clock speed 3.Video initialization including Mono, CGA, EGA/VGAIf no display device found, the speaker will beep.
0E	1.Initialize the APIC (MultiProcessor BIOS only) 2.Test video RAM (If Monochrome display device found) 3.Show message including:Award logo Copyright string BIOS date code & Part No OEM specific sign on messages Energy Star logo (Green BIOS only) CPU brand, type & speed
0F	DMA channel 0 test
10	DMA channel 1 test
11	DMA page registers test
14	Test 8254 timer 0 counter 2
15	Test 8259 interrupt mask bits for channel 1
16	Test 8259 interrupt mask bits for channel 2
19	Test 8259 functionality
1E	If EISA NVM checksum is good, execute EISA initialization (EISA BIOS only)
30	Get base memory & extended memory size
31	1.Test base memory from 256K to 640K 2.Test extended memory from 1M to the top of memory
32	1.Display the Award Plug & Play BIOS extension message(PnP BIOS only) 2.Program all onboard super I/O chips(if any) including COM ports, LPT ports, FDD port according to setup
3C	Set flag to allow users to enter CMOS setup utility
3D	1.Initialise keyboard 2.Install PS2 mouse
3E	Try to turn on level 2 cache Note: Some chipset may need to turn on the L2 cache in this stage. But usually, the cache is turn on later in Post 61h

Code	Meaning
BF	1.Program the rest of the chipset's value according to setup (later setup value program) 2.If auto configuration is enabled, programmed the chipset with predefined values in the MODBINable AutoTable
41	Initialize floppy disk drive controller
42	Initialize hard drive controller
43	If it is a PnP BIOS, initialize serial & parallel ports
45	Initialize math coprocessor
4E	If there is any error detected (such as video, KB....), show all the error messages on the screen & wait for user to press <F1> key
4F	1.If password is needed, ask for password 2.Clear the Energy Star logo (Green BIOS only)
50	Write all the CMOS values currently in the BIOS stack are back into the CMOS
52	1.Initialize all ISA ROMs 2.Later PCI initializations(PCI BIOS only), assign IRQ to PCI devices, initialize all PCI ROMs 3.PnP inits (PnP BIOS), assign IO, Memory, IRQ & DMA to PnP ISA devices, initialize all PnP ISA ROMs 4.Program shadow RAM according to setup setting 5.Program parity according to setup setting 6.Power Management initialization. Enable/Disable global PM, APM interface initialization
53	1.If it is not a PnP BIOS, initialize serial & parallel ports 2.Initialize time value in BIOS data area by translate the RTC time value into a timer tick value
60	Setup virus protection (boot sector) functionality according to setup setting
61	1.Try to turn on level 2 cache (if L2 cache already turned on in post 3D, this part will be skipped) 2.Set the boot up speed according to setup setting 3.Last chance for chipset initialization 4.Last chance for Power Management initialization (Green BIOS only) 5.Show the system configuration table
62	1.Setup daylight saving according to setup values 2.Program NUM lock, typematic rate & speed according to setup setting
63	1.If any change in hardware configuration, update ESCD infor (PnP BIOS only) 2.Clear memory used 3.Boot system via INT 19h
88	CPU failed to initialise-
FF	Boot

Version 6.0 (Jan 29, 1999)

Code	Name	Description
C0	Turn Off Chipset and CPU test	OEM Specific-Cache control
		Processor Status (1FLAGS) Verification. Tests the following processor status flags: Carry, zero, sign, overflow. The BIOS sets each flag, verifies They are set, then turns each flag off and verifies it is off. Read/Write/Verify all CPU registers except SS, SP, and BP with data pattern FF and 00. RAM must be periodically refreshed to keep the memory from decaying. This function ensures that the memory refresh function is working properly.
C1	Memory Presence	First block memory detect OEM Specific-Test to size on-board memory. Early chip set initialization Memory presence test OEM chip set routines Clear low 64K of memory Test first 64K memory.
C2	Early Memory Init	OEM Specific- Board Initialization
C3	Extend MemoryDRAM Select	OEM Specific- Turn on extended memory Initialization. Cyrix CPU initialization. Cache initialization.
C4	Special Display Handling	OEM Specific- Display/Video Switch Handling so that Switch Handling display switch errors never occurs
C5	Early Shadow	OEM Specific- Display/Video Switch Handling Handling so Switch Handling display switch errors never occur

Code	Name	Description
C6	Cache presence test	External cache size detection
CF	CMOS Check	CMOS checkup
B0	Spurious	If interrupt occurs in protected mode.
B1	Unclaimed NMI	If unmasked NMI occurs, display Press F1 to disable NMI, F2 reboot.
BF	Program Chip Set	To program chipset from default values
E1-EF	Setup Pages	E1- Page 1, E2 - Page 2, etc.
1	Force load Default to chipset	Chipset defaults program
2	Reserved	
3	Early Superio Init	Early Initialized the super IO
4	Reserved	
5	Blank Video	Reset Video controller
6	Reserved	
7	Init KBC	Keyboard controller init
8	KB test	Test the keyboard
9	Reserved	
A	Mouse Init	Initialized the mouse
B	Onboard Audio init	Onboard audio controller initialize if exist
C	Reserved	
D	Reserved	
E	CheckSum Check	Check the intergrity of the ROM,BIOS and message
F	Reserved	
10	Auto detec EEPROM	Check Flash type and copy write/erase routines to 0F000h segments
11	Reserved	
12	Cmos Check	Check CMOS Circuitry and reset
13	Reserved	
14	Chipset Default load	Program the chipset registers with CMOS Values
15	Reserved	
16	Clock Init	Init onboard clock generator
17	Reserved	
18	Identify the CPU	Check the CPU ID and init L1/L2 cache
19	Reserved	
1A	Reserved	
1B	Setup Interrupt Vector Table	Initialize first 120 interrupt vectors with SPURIOUS_INT_HDLR and Initialize INT 00h-1Fh according to INT_TBL
1C	Reserved	
1D	Early PM Init	First step initialize if single CPU Onboard
1E	Reserved	
1F	Re-initial KB	Re-init KB
20	Reserved	
21	HPM init	If support HPM, HPM get initialized here
22	Reserved	
23	Test CMOSInterface and battery status	Verifies CMOS is working correctly, and detects bad battery. If failed,load CMOS defaults and load into chipset
24	Reserved	
25	Reserved	
26	Reserved	
27	KBC final Init	Final Initial KBC and setup BIOS data area
28	Reserved	
29	Initialize Video Interface	Read CMOS location 14h to find out type of video in use. Detect and Initialize Video Adapter.

Code	Name	Description
2A	Reserved	
2B	Reserved	
2C	Reserved	
2D	Video memory test	Test video memory, write sign-on message. Setup shadow RAM - Enable shadow according to Setup.
2E	Reserved	
2F	Reserved	
30	Reserved	
31	Reserved	
32	Reserved	
33	PS2 Mouse setup	Setup PS2 Mouse and reset KB
34	Reserved	
35	Test DMA Controller 0	Test DMA channel 0
36	Reserved	
37	Test DMA Controller 1	Test DMA channel 1
38	Reserved	
39	Test DMA Page Registers	Test DMA Page Registers.
3A	Reserved	
3B	Reserved	
3C	Test Timer Counter 2	Test 8254 Timer 0 Counter 2.
3D	Reserved	
3E	Test 8259-1 Mask Bits	Verify 8259 Channel 1 masked interrupts by alternately turning off and on the interrupt lines.
3F	Reserved	
40	Test 8259-2 Mask Bits	Verify 8259 Channel 2 masked interrupts by alternately turning off and on the interrupt lines.
41	Reserved	
42	Reserved	
43	Test Stuck 8259's Interrupt Bits. Test 8259 Interrupt Functionality	Turn off interrupts then verify no interrupt mask register is on. Force an interrupt and verify the interrupt occurred.
44	Reserved	
45	Reserved	
46	Reserved	
47	Set EISA Mode	If EISA non-volatile memory checksum is good, execute EISA initialization. If not, execute ISA tests an clear EISA mode flag.
48	Reserved	
49	Size Base and Ext Memory	Size base memory from 256-640K and extended mem above 1MB.
4A	Reserved	
4B	Reserved	
4C	Reserved	
4D	Reserved	
4E	Size Base and Extended Memory	Size base memory from 256K to 640K and extended memory above 1MB using various patterns (this test is skipped in EISA mode and can be skipped with ESC key in ISA mode).
4F	Reserved	
50	USB init	Initialize USB controller
51	Reserved	
52	Memory Test	Test all memory of memory above 1MB using Virtual 8086 mode, page mode and clear the memory
53	Reserved	

Code	Name	Description
54	Reserved	
55	CPU display	Detect CPU speed and display CPU vendor specific version string and turn on all necessary CPU features
56	Reserved	
57	PnP Init	Display PnP logo and PnP early init
58	Reserved	
59	Setup Virus protect	Setup virus protect according to Setup
5A	Reserved	
5B	Awdflash Load	If required, will auto load Awdflash.exe in POST
5C	Reserved	
5D	Onboard I/O Init	Initializing onboard superIO
5E	Reserved	
5F	Reserved	
60	Setup enable	Display setup message and enable setup Functions
61	Reserved	
62	Reserved	
63	Initialize & install mouse	Detect if mouse is present, initialize mouse, install interrupt vectors.
64	Reserved	
65	PS2 Mouse special	Special treatment to PS2 Mouse port
66	Reserved	
67	ACPI init	ACPI sub-system initializing
68	Reserved	
69	Setup Cache controller	Initialize cache controller.
6A	Reserved	
6B	Setup Entering	Enter setup check and autoconfiguration check up
6C	Reserved	
6D	Initialise floppy drive & controller	Initialize floppy disk drive controller and any drives.
6E	Reserved	
6F	FDD install	Install FDD and setup BIOS data area parameters
70	Reserved	
71	Reserved	
72	Reserved	
73	Initialise hard drive & controller	Initialize hard disk drive controller and any drives.
74	Reserved	
75	Install HDD	IDE device detection and install
76	Reserved	
77	Detect & Initialize Serial/Parallel ports	Initialize any serial and parallel ports (also game port).
78	Reserved	
79	Reserved	
7A	Detect & Initialize Math Coprocessor	Initialize math coprocessor.
7B	Reserved	
7C	HDD Check for Write Protection	HDD check out
7D	Reserved	
7E	Reserved	
7F	POST error check	Check POST errors and display them and ask for user intervention
80	Reserved	
81	Reserved	

Code	Name	Description
82	Security Check	Ask password security (optional).
83	Write CMOS	Write all CMOS values back to RAM and clear screen.
84	Pre-boot Enable	Enable parity checker Enable NMI, Enable cache before boot.
85	Initialise Option ROMs	Initialize any option ROMs present from C8000h to EFFFFh. NOTE: When FSCAN option is enabled, ROMs initialize from C8000h to F7FFFh.
86-92	Reserved	
93	Boot Medium detection	Read and store boot partition head and cylinders values in RAM
94	Final Init	Final init for last micro details before boot
95	Special KBC patch	Set system speed for boot Setup NumLock status according to Setup
96	Boot Attempt	Set low stack Boot via INT 19h.
FF	Boot	

Version 6.0 (Jan 29, 1999) Quick Post

Code	Name	Description
65	Init onboard device	Early Initialized the super IO. Reset Video controller Keyboard controller init. Test the Keyboard Initialized the mouse. Onboard audio controller initialize if exist. Check intergraty of the ROM, BIOS and messageCheck Flash type and copy flash write/erase routines to 0F000h segments Check Cmos Circuitry and reset CMOS Program the chipset registers with CMOS values. Init onboard clock generator
66	Early Sytem setup	Check the CPU ID and init L1/L2 cache. Initialize first 120 interrupt vectors with SPURIOUS_INT_HDLR and initialize INT 00h-1Fh according to INT_TBL First step initialize if single CPU onboard. Re-init KBIf support HPM, HPM get initialized here
67	KBC and CMOS Init	Verifies CMOS working, detect bad battery. If failed, load defaults and load into chipset Final Initial KBC and setup BIOS data area.
68	Video Init	Read CMOS location 14h to find out type of video in use. Detect and Initialize Video Adapter. Test video memory, write sign-on message to screen. Setup shadow RAM - Enable shadow according to Setup.
69	8259 Init	Init 8259 channel 1 and mask IRQ 9
6A	Memory test	Quick Memory Test
6B	CPU Detect and IO init	Detect CPU speed and display CPU vendor specific version stringand turn on necessary CPU features. Display PnP logo and PnP early initSetup virus protect according to Setup. If required, auto load Awdflash.exe in POST Initializing onboard super IO
6C	Reserved	
6D	Reserved	
6E	Reserved	
6F	Reserved	
70	Setup Init	Display setup message and enable setup functions. Detect if mouse is present, initialize mouse, install interrupt vectors. Special treatment to PS2Mouse port ACPI sub-system initializing.
71	Setup Cache controller	Initialize cache controller.
72	Install FDD	Enter setup and auto-configuration check up. Initialize floppy controller and drives. Install FDD and setup BIOS data area
73	Install HDD	Initialize hard drive controller and any drives. IDE device detection and install. Initialize any serial and parallel ports (also game port).
74	Detect & Initialize MathCoprocessor	Initialize math coprocessor.
75	HDD Check for Write Protection	HDD check out
76	Reserved	

Code	Name	Description
77	Display POST error	Check POST error and display them and ask for user interventionAsk password security (optional).
78	CMOS & Option ROM Init	Write all CMOS values back to RAM and clear screen. Enable parity checker. Enable NMI, Enable cache before boot. Initialize ROMs present from C8000h to EFFFFh. NOTE: When FSCAN option is enabled, ROMs initialize from C8000h to F7FFFh.
79	Reserved	
7A	Reserved	
7B	Reserved	
7C	Reserved	
7D	Boot Medium detection	Read and store boot partition head and cylinders values in RAM
7E	Final Init	Final init for last micro details before boot
7F	Special KBC patch	Set system speed for boot. Setup NumLock status as per Setup
80	Boot Attempt	Set low stackBoot via INT 19h.
FF	Boot	

Version 6.0 S4 Codes

Code	Name	Description
5A	Early Chipset Init	Early Initialized the super IO. Reset Video controller. Keyboard controller init. Test the Keyboard. Initilized the mouse
5B	CMOS Check	Check CMOS Circuitry and reset CMOS
5C	Chipset default Prog	Program the chipset registers with CMOS values.Init onboard clock generator
5D	Identify the CPU	Check the CPU ID and init L1/L2 cache
5E	Setup Interrupt Vector Table	Initialize first 120 interrupt vectors with SPURIOUS_INT_HDLR andINT 00h-1Fh according to INT_TBL. First step initialize if single CPU Onboard. Re-init KB. If support HPM, HPM get initialized Here.
5F	Test CMOS Interface & battery status	Verifies CMOS is working correctly, and detects bad battery. If failed, load CMOS defaults and load into chipset.
60	KBC final Init	Final Initial KBC and setup BIOS data area
61	Initialize Video Interface	Read CMOS location 14h to find out type of video in use. Detect and Initialize Video Adapter.
62	Video Memory Test	Test video memory, write sign-on message to screen.Setup shadow RAM - Enable shadow according to Setup.
63	Setup PS2 mouse and test DMA	Setup PS2 Mouse and reset KBTest DMA channel 0
64	Test 8259	Test 8259 channel 1 and mask IRQ 9
65	Init Boot Device	Detect if mouse is present, initialize mouse, install interrupt vectors.Special treatment to PS2 Mouse port ACPI sub-system initializingInitialize cache controller.
66	Install Boot Devices	Enter setup check and autoconfiguration check up. Initialize floppy disk drive controller and any drives. Install FDD and setup BIOS data area Parameters. Initialize hard drive controller and any drives. IDE device detection and install
67	Cache Init	Cache init and USB init
68	PM init	PM initialization
69	PM final Init and issue SMI	Final init Before resume
FF	Full on	

Version 6.0 Boot Block Codes

Code	Name	Description
1	Base memory test	Clear base memory area (0000:0000--9000:ffffh)
5	KB init	Initialized KBC
12	Install interrupt vectors	Install int. vector (0-77), and initialized 00-1fh to their proper place
0D	Init Video	Video initializing
41	Init FDD	Scan floppy and media capacity for onboard super IO
FF	Boot	Load boot sector

Award 6.0 Rev 1.0 11/03/99 Medallion (i810)

BOOT BLOCK CODES

Code	Meaning
CF	Test CMOS R/W functionality
C0	Early chipset initialization
C1	Detect memory
0C	BIOS checksum verify
C5	OEM Specific-Early Shadow enable for fast boot. Copy BIOS from ROM into shadow.
01	Clear base memory 0-640 Kb
05	Enable Keyboard Interface
0C	Initial interrupt vector 00-1Fh
0D	Detect and Initialize Video Adapter. If no display device, speaker will beep
41	Initialize floppy disk drive controller and detect media type
FF	SystemBooting

Code	Description
CFh	Test CMOS R/W functionality.
C0h	Early chipset init: Disable shadow RAM, L2 (socket 7 or below), Program basic chipset register
C1h	Detect memory, autodetect DRAM size, type, ECC. Autodetect L2 cache (socket 7 or below)
C3h	Expand compressed BIOS code to DRAM
C5h	Call chipset hook to copy BIOS back to E000 & F000 shadow RAM.
0h1	Expand the Xgroup codes locating in physical address 1000:0
03h	Initial Superio_Early_Init switch.
05h	1. Blank out screen 2. Clear CMOS error flag
07h	1. Clear 8042 interface 2. Initialize 8042 self-test
08h	Test special keyboard controller for Winbond 977 Super I/O chips. Enable keyboard interface.
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.
0Eh	Test F000h segment shadow to see whether R/W-able. If test fails, keep beeping speaker.
10h	Auto detect flash type to load appropriate flash R/W codes into the run time area in F000 for ESCD & DMI support.
12h	Use walking 1s to check interface in CMOS circuitry. Set RTC power status, check override.
14h	Program chipset default values into chipset. Default values are MODBINable by OEMs.
16h	Initial onboard clock generator if Early_Init_Onboard_Generator defined. See also 26h.
18h	Detect CPU information including brand, SMI type (Cyrix or Intel) and CPU level (586 or 686).
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.

Code	Description
1Dh	Initial EARLY_PM_INIT switch.
1Fh	Load keyboard matrix (notebook platform)
21h	HPM initialization (notebook platform)
23h	Check validity of RTC value- e.g. 5Ah is an invalid value for RTC minute. Load CMOS settings into BIOS stack. If CMOS checksum fails, use default value.
24h	Prepare BIOS resource map for PCI & PnP use. If ESCD is valid, take into consideration of the ESCD's legacy information.
25h	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
26h	If Early_Init_Onboard_Generator not defined Onboard clock generator init. Disable respective clock resource to empty PCI & DIMM slots. Init onboard PWM and H/W monitor devices
27h	Initialize INT 09 buffer
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. Example: onboard IDE controller. Measure CPU speed.
2Bh	Invoke Video BIOS
2Dh	Initialize double-byte language font (Optional) Put information on screen display, including Award title, CPU type, CPU speed, full screen logo.
33h	Reset keyboard if Early_Reset_KB is defined e.g. Winbond 977 Super I/O chips. See 63h.
35h	Test DMA Channel 0
37h	Test DMA Channel 1.
39h	Test DMA page registers.
3Ch	Test 8254
3Eh	Test 8259 interrupt mask bits for channel 1.
40h	Test 8259 interrupt mask bits for channel 2.
43h	Test 8259 functionality.
47h	Initialize EISA slot
49h	Calculate total memory by testing the last double word of each 64K page.Program write allocation for AMD K5 CPU.
4Eh	Program MTRR of M1 CPU. Initialize L2 cache for P6 class CPU & program CPU with proper 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.
50h	Initialize USB Keyboard & Mouse.
52h	Test all memory (clear all extended memory to 0)
53h	Clear password according to H/W jumper (Optional)
55h	Display number of processors (multi-processor platform)
57h	Display PnP logo. Early ISA PnP initialization -Assign CSN to every ISA PnP device.
59h	Initialize the combined Trend Anti-Virus code.
5Bh	(Optional Feature) Show message for entering AWDFLASH.EXE from FDD
5Dh	Initialize Init_Onboard_Super_IO. Initialize Init_Onbaard_AUDIO.
60h	OK to enter Setup utility; i.e. not until this POST stage can users enter the CMOS setup utility.
63h	Reset keyboard if Early_Reset_KB is not defined.
65h	Initialize PS/2 Mouse
67h	Prepare memory size information for function call: INT 15h ax=E820h
69h	Turn on L2 cache
6Bh	Program chipset registers according to items described in Setup & Auto-configuration table.
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".
6Fh	Initialize floppy controller. Set up floppy related fields in 40:hardware.
75h	Detect & install all IDE devices: HDD, LS120, ZIP, CDROM…..
76h	(Optional) Enter AWDFLASH.EXE if found in floppy drive & ALT+F2 is pressed.

Code	Description
77h	Detect serial ports & parallel ports.
7Ah	Detect & install co-processor
7Ch	Init HDD write protect.
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 to continue: Clear EPA or customization logo.

E8POST.ASM STARTS

Code	Description
82h	Call chipset power management hook. Recover 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	USB final Initialization. Switch screen back to text mode
87h	NET PC: Build SYSID Structure.
89h	Assign IRQs to PCI devices. Set up ACPI table at top of the memory.
8Bh	Invoke all ISA adapter ROMs. Invoke all PCI ROMs (except VGA)
8Dh	Enable/Disable Parity Check according to CMOS setup. APM Initialization
8Fh	Clear noise of IRQs
93h	Read HDD boot sector information for Trend Anti-Virus code
94h	Enable L2 cache. Program Daylight Saving. Program boot up speed. Chipset final initialization.Power management final initialization. Clear screen & display summary table. Program K6 write allocationProgram P6 class write combining
95h	Update keyboard LED & typematic rate
96h	Build MP table. Build & update ESCD. Set CMOS century to 20h or 19h. Load CMOS time into DOS timer tickBuild MSIRQ routing table.
FFh	Boot attempt (INT 19h)

Unexpected Errors

Code	Meaning
B0	If interrupt occurs in protected mode
B1	Unclaimed NMI occurs

v3.3

Code	Meaning	Code	Meaning
1-5	Keyboard controller	1F	Memory verifier
06	On board LSI	20-23	CPU support chips
07	CPU	24	Extended memory size
8-0E	CMOS; 8254; 8237; 8259; EPROM	25	Extended memory size
0F	Extended CMOS	26	Protected mode
10-14	Refresh	27-28	Shadow RAM
15	First 64K RAM	2A	Initialise keyboard
16	Interrupt vector tables	2B	Floppy drive initialisation
17	Video initialisation	2C	Serial port initialisation
18	Video memory	2D	Parallel port initialisation
19-1A	Interrupt line mask	2E	Hard disk initialisation

Code	Meaning	Code	Meaning
1B	Battery good	2F	Maths coprocessor
1C	CMOS checksum	30	Reserved
1D	CMOS chip	31	Optional ROMs
1E	Memory size	FF	Boot

CHIPS AND TECHNOLOGIES

Some displayed as decimal as well as to port 80 in hex. Micro Channel uses 680 and 3BC.

POST Procedures

Procedure	Meaning
Power On Tests	CPU synchronises with clock. Check the CPU or clock.
System ROM Check	The BIOS runs a checksum on itself. Check the BIOS chips.
DMA Controller Fail	DMA Controllers are initialised and tested. Check the DMA chips.
System Timer Failed	Channels 0/1/2 are tested in sequence. Check the PIT chips.
Base 64K Memory Testing	Walking-bit test on 1st 64K of RAM. Check for bad RAM chips or data or address line.
Interrupt Contr Failed	Test the 8259 chip.
CPU Still In Prot Mode	Attempts made to read configuration of system through 8042 keyboard controller.
Refresh Not Occurring	Memory refresh is tested; standard refresh is 120-140 ns. Check the PIT chip.
Keyboard Controller Not Responding	Tests are run on the keyboard controller. Check the 8042 chip.
Could Not Enter Protected Mode	BIOS attempts to enter protected mode to test extended memory. Check 8042 or A20.
Initialise Timer	Attempts are made to initialise the PIT.
Init DMA Controller	Attempts are made to initialise the DMA Controller.
Entering/Exiting Protected Mode	The transition is handled by the keyboard controller and the A20 line. Check the 8042 or the A20.
Relocate Shadow RAM	BIOS attempts to shadow itself into extended memory. Check for memory problems.
Test For EMS	Check the EMS adapter or an improper CMOS/Jumper setting.
Test Video Capabilities	Normally includes a memory test on the adapter memory up to 256K.
Test Memory	Extensive testing of Base, Extended, Expanded memory. Check for defective memory modules; 8042 chip; A20 line or an improper CMOS/Jumper setting.
Check System Options	The hardware in the system is compared with the values stored in CMOS. The PIT/PIC/8042/RTC and other system board chips are tested again.
Peripheral Check/Test	Checks made for peripherals at standard I/O ports including serial and parallel ports keyboards and coprocessors. Should see an error message on screen at this point.
Floppy Test	Floppy devices set in CMOS are checked and initialised. If a bootable floppy is found the fixed disks are tested and BIOS will boot to floppy. Check for defective controllers or an improper CMOS Setup.
Fixed Disk Test	Checks for fixed disks in CMOS. If no bootable floppy in A: drive, BIOS loads first sector off the first fixed disk and jumps to memory where the sector was loaded. You may just see a flashing cursor or an error message from the potential operating system. Check for improper CMOS setup/defective controller/fixed disk or corruption of bootloader software on the fixed disk.
Advanced Options	include mouse/cache etc. You should see an error message on the screen at this point, except that a defective cache may hang the system; in most cases, the cache will be disabled by the BIOS.

POST Codes

NEAT, PEAK/DM, OC8291, ELEAT BIOS

Hex	Dec	Code
00	00	Error in POS register.
01	01	Flag register failed.
02	02	CPU register failed.
03	03	System ROM did not checksum
04	04	DMA controller failed
05	05	System timer failed
06	06	Base 64K RAM failed address test,: not installed, misconfigured, bad addressing
07	07	Base 64K RAM failed data test
08	08	Interrupt controller failed
09	09	Hot (unexpected) interrupt occurred
0A	10	System timer does not interrupt
0B	11	CPU still in protected mode
0C	12	DMA page registers failed
0D	13	Refresh not occurring
0E	14	Keyboard controller not responding
0F	15	Could not enter protected mode
10	16	GDT or IDT failed
11	17	LDT register failed
12	18	Task register failed
13	19	LSL instruction failed
14	20	LAR instruction failed
15	21	VERR/VERW failed
16	22	Keyboard controller gate A20 failed
17	23	Exception failed/unexpected exception
18	24	Shutdown during memory test
19	25	Last used error code
1A	26	Copyright checksum error
1B	27	Shutdown during memory sizing
1C	28	CHIPSet initialization
50	80	Initialize hardware
51	81	Initialize timer
52	82	Initialize DMA controller
53	83	Initialize interrupt controller
54	84	Initialize CHIPSet
55	85	Setup EMS configuration
56	86	Entering protected mode for first time
57	87	Size memory chips
58	88	Configure memory chip interleave
59	89	Exiting protected mode for first time
5A	90	Determine system board memory size
5B	91	Relocate shadow RAM
5C	92	Configure EMS
5D	93	Set up wait state configuration
5E	94	Re-test 64K RAM
5F	95	Test shadow RAM

Hex	Dec	Code
60	96	Test CMOS RAM
61	97	Test video
62	98	Test and initialize DDNIL bits
63	99	Test protected mode interrupt
64	100	Test address line A20
65	101	Test memory address lines
66	102	Test memory
67	103	Test extended memory
68	104	Test timer interrupt
69	105	Test real time clock (RTC)
6A	106	Test keyboard
6B	107	Test 80x87 math chip
6C	108	Test RS232 serial ports
6D	109	Test parallel ports
6E	110	Test dual card
6F	111	Test floppy drive controller
70	112	Test hard drive controller
71	113	Test keylock
72	114	Test pointing device
90	144	Setup RAM
91	145	Calculate CPU speed
92	146	Check configuration
93	147	Initialize BIOS
94	148	POST Bootstrap
95	149	Reset ICs
96	150	PEAK: System board POS. NEAT/OC8291 ELEAT: Test/init cache and controller.
97	151	VGA Power on Diagnostics and setup
98	152	Adapter POS
99	153	Re-initialize DDNIL bits
A0	160	Exception 0
A1	161	Exception 1
A2	162	Exception 2
A3	163	Exception 3
A4	164	Exception 4
A5	165	Exception 5
A6	166	Exception 6
A7	167	Exception 7
A8	168	Exception 8
A9	169	Exception 9
AA	170	Exception A
AB	171	Exception B
AC	172	Exception C
AD	173	Exception D
C0	224	System board memory failure
C1	225	I/O Channel Check activated
C2	226	Watchdog timer timeout
C3	227	Bus timer timeout

COMPAQ

Port 84 codes show errors - 85 shows the category:

```
00      System BIOS
01      Error after boot
05      Video POST
```

General

Code	Meaning	Code	Meaning
00	Initialise flags	09	Reset code in CMOS byte
01	Read manufacturing jumper	0A	Vector Via 40:67 reset function
02	8042 Received Read command	0B	Vector Via 40:67 with E01 function
03	No response from 8042	0C	Boot reset function
04	Look for ROM at E000	0D	Test #2 8254 Counter 0
05	Look for ROM at C800	0E	Test #2 8254 Counter 2
06	Normal CMOS reset code	0F	Warm Boot
08	Initialise 8259		

Overall Power Up Sequence

Code	Meaning	Code	Meaning
10	PPI disabled	20	Test real and extended memory
11	Initialise (blast) VDU controller	21	Initialise time-of-day
12	Clear Screen; turn on video	22	Initialise 287 coprocessor
13	Test time 0	23	Test the keyboard and 8042
14	Disable RTC interrupts	24	Reset A20
15	Check battery power	25	Test diskette subsystem
16	Battery has lost power	26	Test fixed disk subsystem
17	Clear CMOS diags	27	Initialise parallel printer
18	Test base memory (first 128K)	28	Perform search for optional ROMs
19	Initialise base memory	29	Test valid system configuration
1A	Initialise VDU adapters	2A	Clear screen
1B	The system ROM	2B	Check for invalid time and date
1C	CMOS checksum	2C	Optional ROM search
1D	DMA controller/page registers	2D	Test timer 2
1E	Test keyboard controller	2F	Write to diagnostic byte
1F	Test 286 protected mode		

Base RAM Initialisation

Code	Meaning	Code	Meaning
30	Clear first 128K bytes of RAM	36	Check battery power
31	Load interrupt vectors 70-77	37	Check for game adapters
32	Load interrupt vectors 00-1F	38	Check for serial ports
33	Initialise MEMSIZE and RESETWD	39	Check for parallel printer ports
34	Verify CMOS checksum	3A	Initialise port and comm timeouts
35	CMOS checksum not valid	3B	Flush keyboard buffer

Base RAM Test

Code	Meaning	Code	Meaning
40	Save RESETWD value	44	Start verify of 128K RAM test
41	Check RAM refresh	45	Check for parity errors
42	Start write of 128K RAM test	46	No RAM errors
43	Rest parity checks	47	RAM error detected

VDU Initialisation and Test

Code	Meaning	Code	Meaning
50	Check for dual frequency in CMOS	58	Start of VDU test (each adapter)
51	Check CMOS VDU configuration	59	Check existence of adapter
52	Start VDU ROM search	5A	Check VDU registers
53	Vector to VDU option ROMs	5B	Start screen memory test
54	Initialise first display adapter	5C	End test of adapter
55	Initialise second display adapter	5D	Error detected on an adapter
56	No display adapters installed	5E	Test the next adapter
57	Initialise primary VDU mode	5F	All adapters successfully tested

Memory Test

Code	Meaning	Code	Meaning
60	Start of memory tests	6B	Display error message
61	Enter protected mode	6C	End of memory test
62	Start memory sizing	6D	Initialise KB OK string
63	Get CMOS size	6E	Determine size to test
64	Start test of real memory	6F	Start MEMTEST
65	Start test of extended memory	70	Display XXXXXKB OK
66	Save size memory (base	71	Test each RAM segment
67	128K option installed CMOS bit	72	High order address test
68	Prepare to return to Real Mode	73	Exit MEMTEST
69	Back in Real Mode-successful	74	Parity error on bus
6A	Protected mode error during test		

80286 Protected Mode

Code	Meaning	Code	Meaning
75	Start protected mode test	7B	Exit protected test
76	Prepare to enter protected mode	7C	High order address test failure
77	Test software exceptions	7D	Entered cache controller test
78	Prepare to return to Real Mode	7E	Programming memory cache
79	Back in Real Mode-successful	7F	Copy system ROM to high RAM
7A	Back in Real Mode-error occurred		

8042 and Keyboard

Code	Meaning	Code	Meaning
80	Start of 8042 test	88	Got result
81	Do 8042 self test	89	Test for stuck keys
82	Check result received	8A	Key seems to be stuck
83	Error result	8B	Test keyboard interface
84	OK 8042	8C	Got result
86	Start test	8D	End of Test
87	Got acknowledge		

System Board Test

Code	Meaning	Code	Meaning
90	Start of CMOS test	94	Page registers seem OK
92	CMOS seems to be OK	95	DMA controller is OK
92	Error on CMOS read/write test	96	8237 initialisation is complete
93	Start of DMA controller test	97	Start of NCA RAM test

Diskette Test

Code	Meaning	Code	Meaning
A0	Start of diskette tests	B2	Combo controller failed-exit
A1	FDC reset active (3F2h bit 2)	B3	Testing drive 1
A2	FDC reset inactive (3F2h bit 2)	B4	Testing drive 2
A3	FDC motor on	B5	Drive error (error condition)
A4	FDC timeout error	B6	Drive failed (failed to respond)
A5	FDC failed reset	B7	No fixed drives-exit
A6	FDC passed reset	B8	Fixed drive tests complete
A8	Start to determine drive type	B9	Attempt to boot diskette
A9	Seek operation initiated	BA	Attempt to boot fixed drive
AA	Waiting for FDC seek status	BB	Boot attempt failed FD/HD
AF	Diskette tests completed	BC	Boot record read, jump to boot record
B0	Start of fixed disk drive tests	BD	Drive error, retry booting
B1	Combo board not found-exit	BE	Weitek coprocessor test

EISA TESTS (Deskpro/M, /LT, /33L, P486c)

Code	Meaning	Code	Meaning
C0	EISA non-volatile memory checksum	C5	EISA display config error messages
C1	EISA DDF map initialization	C6	EISA PZ initialization begun
C2	EISA IRQ initialization	C7	EISA PZ initialization done
C3	EISA DMA initialization	C8	System manager board self-test
C4	EISA slot initialization		

LT, SLT, LTE

Code	Meaning	Code	Meaning
C0	Disable NMI	C6	Update BIOS time of day
C1	Turn off hard disk subsystem	C7	Turn on hard disk/modem subsystems
C2	Turn off video subsystem	C8	Turn on floppy disk subsystem
C3	Turn off floppy disk subsystem	C9	Turn on video subsystem
C4	Turn off hard disk/modem subsystems	CB	Flush keyboard input buffer
C5	Go to standby	CC	Re-enable MNI

Standard POST Functions

Code	Meanings	Code	Meaning
D0	Entry to clear memory routine	D5	Clr base mem, CLIM reg init fail SLT/286
D1	Ready to go to protected mode	D7	Scan and clear DDNIL bits
D2	Ready to clear extended memory	D9	Orvonton 4-way cache detect
D3	Ready to reset back to real mode	DD	Built-in self-test failed
D4	Back in real mode, ready to clear		

Option ROM Replacement

Code	Meaning	Code	Meaning
E0	Ready to replace E000 ROM	E9	Receiving for serial external boot sector
E1	Completed E000 ROM replacement	EA	Looking for parallel external boot ID str
E2	Ready to replace EGA ROM	EB	Receiving parallel external boot sector
E3	Completed EGA ROM replacement	EC	Boot record read, jump to boot record
E8	Looking for serial external boot ID str		

Port 85=05 (Video POST)

Code	Meaning	Code	Meaning
00	Entry into video option ROM	50	Slot type conflict error
01	Alternate adapter tests	51	Video memory conflict error
02	Vertical sync tests	52	ROM conflict error
03	Horizontal sync tests	60	Red DAC stuck low error
04	Static tests	61	Green DAC stuck low error
05	Bus tests	62	Blue DAC stuck low error
06	Configuration tests	63	DAC stuck high error
07	Alternate ROM tests	64	Red DAC fault error
08	Colour gun off tests	65	Green DAC fault error
09	Colour gun on tests	66	Blue DAC fault error
0A	Video memory tests	70	Bad alternate ROM version
0B	Board present tests	80	Colour gun stuck ON base code
10	Illegal configuration error	90	Colour gun stuck OFF base code
20	No vertical sync present	A0	Video memory failure base code
21	Vertical sync out of range	F0	Equipment failure base code
30	No horizontal sync present	00	Video POST over (also send 00 to 85)
40	Colour register failure		

After POST, the OS is booted. If a run-time error is detected, code 01 is sent to port 85, and the error code to port 84 before booting. Here are run-time codes:

Code	Meaning	Code	Meaning
10	Entered dummy end-of-interrupt routine	14	Illegal opcode instruction encountered
11	Entered int 2 module (parity error)	15	Entered dum iret module
12	Emulating lock instruction	16	Entered irg9 module
13	Emulating loadall instruction	17	Entered 287err module

286 DeskPro

Code	Meaning	Code	Meaning
01	CPU	42	Printer Date
02	Coprocessor	43	Printer Pattern Test
03	DMA	48	Printer Failed
04	Interrupt Controller	51	VDU Controller Test
05	Port 61	52	VDU Controller Test
06	Keyboard Controller	53	VDU Attribute Test
07	CMOS	54	VDU Character Set Test
08	CMOS	55	VDU 80x25 Mode
09	CMOS	56	VDU 80x25 Mode
10	Programmable Timer	57	VDU 40x25 Mode
11	Refresh Detect Test	60	Diskette Drive ID Test
12	Speed Test	61	Format
14	Speaker Test	62	Read Test
21	Memory Read/Write	63	Write/Read Compare Test
24	Memory Address	64	Random Seek
25	Walking I/O	65	ID Media Test
31	Keyboard Short Test	66	Speed Test
32	Keyboard Long Test	67	Wrap Test
33	Keyboard LED Test	68	Write Protect Test
35	Security Lock Test	69	Reset Controller Test
41	Printer Failed		

386 DeskPro

Code	Meaning	Code	Meaning
01	I/O ROM Error	41	Printer Error
02	System Memory Board Failure	42	Mono Adapter Failure
12	System Option Error	51	Display Adapter Failure
13	Time and Date not set	61	Diskette Controller Error
14	Memory Size Error	62	Diskette Boot Recorder Error
21	Memory Error	65	Diskette Drive Error
23	Memory Address Error	67	Ext FDC Failed-Go To Internal F
25	Memory Error	6A	Floppy Port Address Conflict
26	Keyboard Error	6B	Floppy Port Address Conflict
33	Keyboard Controller Error	72	Coprocessor Detection
34	Keyboard or System Unit Error		

486 DeskPro

Code	Meaning	Code	Meaning
01	CPU Test Failed	34	Keyboard Typematic Test Failed
02	Coprocessor or Weitek Error	41	Printer Failed or Not Connected
03	DMA Page Registers	42	Printer Data Register Failed
04	Interrupt Controller Master	43	Printer Pattern Test
05	Port 61 Error	48	Printer Not Connected
06	Keyboard Controller Self Test	51	Video Controller Test Failed
07	CMOS RAM Test Failed	52	Video Memory Test Failed
08	CMOS Interrupt Test Failed	53	Video Attribute Test Failed
09	CMOS Clock Load Data Test	54	Video Character Set Test Failed
10	Programmable Timer	55	Video 80x25 Mode
11	Refresh Detect Test Failed	56	Video 80x25 Mode
12	Speed Test Slow Mode out of range	57	Video 40x25 Mode Test Failed
13	Protected Mode Test Failed	58	Video 320x200 Mode Colour Set 1
14	Speaker Test Failed	59	Video 320x200 Mode Colour Set 1
16	Cache Memory Configuration	60	Diskette ID Drive Types Test
19	Installed Devices Test	61	Diskette Format Failed
21	Memory Machine ID Test Failed	62	Diskette Read Test Failed
22	Memory System ROM Checksum	63	Diskette Write
23	Memory Write/Read Test Failed	65	Diskette ID Media Failed
24	Memory Address Test Failed	66	Diskette Speed Test Failed
25	Walking I/O Test Failed	67	Diskette Wrap Test Failed
26	Increment Pattern Test Failed	68	Diskette Write Protect Test
31	Keyboard Short Test, 8042	69	Diskette Reset Controller Test
32	Keyboard Long Test Failed	82	Video Memory Test Failed
33	Keyboard LED Test, 8042	84	Video Adapter Test Failed

DELL

OEM version of Phoenix, Port 80. Also uses Smartvu display on front.

Code	Beeps	SmartVu	Meaning
01	1-1-2	Regs xREG xCPU(2)	CPU register test in progress
02	1-1-3	CMOS xCMS	CMOS write/read test failed
03	1-1-4	BIOS xROM	ROM BIOS checksum bad
04	1-2-1	Timr xTMR	Programmable interval timer failed
05	1-2-2	DMA xDMA	DMA initialization failed
06	1-2-3	Dpge xDPG	DMA page register write/read bad
08	1-3-1	Rfsh xRFH	RAM refresh verification failed
09	1-3-2	Ramp RAM?	First 64K RAM test in progress
0A	1-3-3	xRAM	First 64K RAM chip or data line bad, multi-bit
0B	1-3-4	xRAM	First 64K RAM odd/even logic bad
0C	1-4-1	xRAM	Address line bad first 64K RAM
0D	1-4-2	64K? x64K	Parity error detected in first 64K RAM
10	2-1-1		Bit 0 first 64K RAM bad
11	2-1-2		Bit 1 first 64K RAM bad

Code	Beeps	SmartVu	Meaning
12	2-1-3		Bit 2 first 64K RAM bad
13	2-1-4		Bit 3 first 64K RAM bad
14	2-2-1		Bit 4 first 64K RAM bad
15	2-2-2		Bit 5 first 64K RAM bad
16	2-2-3		Bit 6 first 64K RAM bad
17	2-2-4		Bit 7 first 64K RAM bad
18	2-3-1		Bit 8 first 64K RAM bad
19	2-3-2		Bit 9 first 64K RAM bad
1A	2-3-3		Bit 10 first 64K RAM bad
1B	2-3-4		Bit 11 first 64K RAM bad
1C	2-4-l		Bit 12 first 64K RAM bad
1D	2-4-2		Bit 13 first 64K RAM bad
1E	2-4-3		Bit 14 first 64K RAM bad
1F	2-4-4		Bit 15 first 64K RAM bad
20	3-1-1	SDMA xDMS	Slave DMA register bad
21	3-1-2	MDMA xDMM	Master DMA register bad
22	3-1-3	PICO xICO	Master interrupt mask register bad
23	3-1-4	PIC1 xIC1	Slave interrupt mask register bad
25	3-2-2	Intv	Interrupt vector loading in progress
27	3-2-4	Kybd xKYB	Keyboard controller test failed
28	3-3-1	CmCk	CMOS power bad; calculating checksum
29	3-3-2	Cnfg	CMOS configuration validation in progress
2B	3-3-4		Video memory test failed
2C	3-4-1	CRTI	Video initialization failed
2D	3-4-2		Video retrace failure
2E	3-4-3	CRT?	Search for video ROM in progress
30	none		Screen operable, running with video ROM
31	none		Monochrome monitor operable
32	none		Colour monitor (40 column) operable
33	none		Colour monitor (80 column) operable

Non-Fatal Error Meanings for ATs

Only if Manufacturing Jumper is on POST

Code	Beeps	Smartvu	Meaning
34	4-2-1	Tick	Timer tick interrupt test in progress or bad
35	4-2-2	Shut	Shutdown test in progress or bad
36	4-2-3	A20	Gate A20 bad
37	4-2-4		Unexpected interrupt in protected mode
38	4-3-1	Emem	RAM test in progress or high address line bad > FFFF
3A	4-3-3	Tmr2	Interval timer channel 2 test or bad
3B	4-3-4	Time	Time-of-Day clock test or bad
3C	4-4-1	Asyn	Serial port test or bad
3D	4-4-2	Prnt	Parallel port test or bad
3E	4-4-3		Math coprocessor test or bad
3F	4-4-4	XCsh	Cache test failure

DTK

Evolved from ERSO (Taiwan).

Post Procedures-Symphony 486 BIOS

Procedure	Meaning
Init Interrupt Controller	Check the PIC chips.
Initialise Video Card	
Initialise DMA Controller	
Initialise Page Register	Check the 74612 chips.
Test Keyboard Controller	Internal operations of the keyboard controller are tested (8042).
Initialise DMA Contr/Timer	DMA registers and CMOS status bytes 0E/0F cleared. BIOS initialises 8254. Check the DMS or PIT chips.
DRAM Refresh Testing	
Base 64K Memory Testing	Walking-bit test of 1st 64K RAM. Check bad chips or data or address line.
Set System Stack	Part of memory is set aside by BIOS as a stack. Check bad DMA/memory.
Read System Configuration via 8042	Check for incorrect setup or bad keyboard controller or CMOS chip.
Test Keyboard Clock and Data Line	Keyboard's handling of A20 tested, and internal clock. Check keyboard controller or a bad address line.
Determine Video Type	
Check RS232/Printer	Test serial/parallel ports. Check I/O cards.
FDC Check	Test floppy controller. Check the drive as well.
Count Shadow RAM	Run memory tests. Check for bad memory chips address lines or data lines.
Show Total Mem/Return Real Mode	Total memory displayed and return to real mode. Keyboard or A20.
Back to Real Mode	Transition attempted through A20 line and keyboard controller.
Check HDC	The hard drive controller is tested.
Check FDD	Attempts are made to initialise the floppy drives.
Turn off Gate A20 and Test CoPro	Ttransition back to real mode by disabling A20 then copro tested. Check keyboard controller coprocessor or improper setup.
Set Time and Date	Time and date will be read from the RTC.

POST Codes

Code	Meaning	Code	Meaning
01	Power on start	48	Video 80 x 25 mode initialisation
03	Initialise interrupt controller-8259	4D	Display DTK BIOS title
05	Initialise video card-MCA and CGA	4F	Check RS232 and printer
0D	Initialise DMA controller-8237	50	FDC check
0E	Initialise page register-74612	55	Count shadow RAM
12	Test keyboard controller-8042	58	Display total memory, return to real mode
16	Initialise DMA controller and timer	5A	Back to real mode
22	DRAM refresh testing	60	Check HDC
25	Base 64K memory testing	62	Check FDD
30	Set system stack	65	Check HDC
33	Read system configuration through 8042	67	Initialise FDC and HDC
37	Test keyboard clock and data line	6A	Turn off gate A20 and test coprocessor
40	Determine video type	70	Set time and date according to RTC
44	Testing MGA and CGA if existing	77	Boot

EUROSOFT

See Mylex/Eurosoft.

FARADAY A-TEASE

Owned by Western Digital.

Code	Meaning	Code	Meaning
01	CPU test failed	1E	Check CMOS real time clock
02	BIOS ROM checksum test	1F	Generate and verify CMOS RAM checksum
03	Shutdown	21	Initialize PROM drivers
04	DMA page register test	22	Test parallel port loopback
05	8254 timer test	23	Test serial port loopback
06	Start refresh	24	Test CMOS real time clock
07	8042 keyboard controller test	25	Test shutdown
08	Test lower 128K RAM	26	Test memory over 1mb; output codes for errors 80-FF
09	Setup video	80	Divide overflow
0A	Test 128K-640K	81	Single step
0B	Test DMA controller #1	82	NMI
0C	Test DMA controller #2	83	Breakpoint
0D	Test interrupt controller #1	84	Int 0 detect
0E	Test interrupt controller #2	85	Bound error
0F	Test control port	86	Invalid opcode
10	Test parity	87	Processor extension not available
11	Test CMOS RAM	88	Double exception
12	Test for manufacturing mode	89	Processor ext segment error
13	Set up interrupt vectors	8A	Invalid task state segment
14	Test keyboard	8B	Segment not present
15	Configure parallel port	8C	Stack segment not present
16	Configure serial ports	8D	General protection error
17	Configure lower 640K RAM	8E	General protection error
18	Configure RAM above 1 Mb	8F	General protection error
19	Configure keyboard	90	Processor extension error
1A	Configure floppy drive	91-FF	Spurious interrupts (except F3 and F0)
1B	Configure hard drive	F3	CPU virtual (protected mode) test error
1C	Configure game card	F0	Virtual block move error
1D	Configure 80287 math chip		

HEADSTART

See Philips.

HP

Derived from Phoenix, all POST information is sent to the screen.

Vectra

A failure during POST will emit four beeps, and a 4-digit hex code to the monitor. Failures that occur before EGA/VGA monitors are initialised will not be displayed, so use a mono instead. BIOSes prior to March 1989 initialised the video before getting on with the POST.

POST PROCEDURES

Code	Meaning
CPU	Registers in CPU tested with data patterns; error flags are set, verified and reset.
BIOS Checksum	Checksums are performed on High and low BIOS Chips.
PIC Test	Test Timer Channels 0-2 then the memory refresh signal. Initialise timer if tests are passed. Check the 8254 chip.
64K Test	Walking-bit and address collision tests are performed on the first 64K of memory. Check for a bad memory chip or address line.
Cache Controller	Test the CPU cache controller and memory.
Video Adapter	Initialise the video adapter. If EGA/VGA is present wait for adapter to finish internal diagnostics. check the adapter or for improper setup.
DMA Test	Bit-patterns written to all DMA controller registers (inc page registers) and verifies the patterns written. If tests pass, registers are reset and the controller initialised.
PIC Test	Test mask register of master and slave interrupt controllers. Generate interrupt and monitor CPU to test success. Failure is normally down to the PIC but the interrupt test uses the BIOS clock (interrupt) and the RTC so check those.
Keyboard Controller	Perform several tests on the 8042 keyboard controller then send a series of interrupt request commands via the 8259 PIC.
HP-HIL Test	Test HP-HIL (Hardware Interrupt Level) controller with data patterns and verify it.
CMOS Test	Perform a checksum on the standard and extended CMOS RAM areas; perform a register test and check Byte 0D to determine power status. Check the CMOS extended CMOS RAM or battery respectively.
Manufacturing Test	Search for diagnostic tool used in manufacturing and run predetermined tests if found. Otherwise continue POST.
Base Memory Test	Test RAM between 64-640K with several pattern tests; the bit failure and bank can be determined by the displayed hex code.
Ext Memory Test	Test extended memory found. Bank and failing bit displayed by the hex code.
RTC Test	Test the RTC portion of the CMOS chip.
Keybrd Contr	Test keyboard controller; initialise k/b if no errors.
Floppy Disk	Test and initialise floppy controllers and drives found; check specific errors with displayed hex code. Check for correct setup or defective CMOS chip or battery.
Maths Copro	Test NPU registers and interrupt request functions.
CPU Clock Test	Test interface between CPU and system at different speeds. Check for incorrect clock setting for system peripherals or a bad CPU or clock generator chip.
Serial/Parallel Test	Test and initialise serial/parallel ports. Failure here will not halt the POST. The Vectra RS BIOS does not test the parallel port.
Boot	Initialise the BIOS vector table; standard and extended CMOS data areas and any adapter ROMs present. Then call Int 19 and give control to the boot loader. Failures past this point are usually down to the hard drive or corrupt OS code.

POST CODES

Code	Meaning	Code	Meaning
01	LED test	18	RAM address line independence test
02	Processor test	19	Size extended memory
03	System (BIOS) ROM test	20	Real-Mode memory test (first 640K)
04	RAM refresh timer test	21	Shadow RAM test
05	Interrupt RAM test	22	Protect Mode RAM test (extended RAM)
06	Shadow System ROM BIOS	23	Real Time clock test
07	CMOS RAM test	24	Keyboard test
08	Internal cache memory test	25	Mouse test
09	Initialize the Video Card	26	Hard disk test
10	Test external cache	27	LAN test
11	Shadow option ROMs	28	Flexible disk controller subsystem test
12	Memory Subsystem test	29	Internal numeric coprocessor test
13	Initialize EISA/ISA hardware	30	Weitek coprocessor test
14	8042 self-test	31	Clock speed switching test
15	Timer 0/Timer 2 test	32	Serial Port test
16	DMA Subsystem test	33	Parallel Port test
17	Interrupt controller test		

Vectra ES

Code	Meaning	Code	Meaning
000F	80286 CPU is bad		X=9 Y>0=Bad U42 Z>0=Bad U32
0010	Bad checksum on ROM 0	5XYZ	Lower 640K failed marching ones test
0011	Bad checksum on ROM 1		X = bbbx = > bbb (0-7) is # of 128K bank
011X	One RTC register is bad; Register = x(0-D)		bbb0 = > Indicate even byte bad
0120	RTC failed to tick		bbb1 = > Indicate odd byte bad
0240	CMOS/RTC has lost power		YZ = bbbb bbbb = > Bits (b = 1 are bad)
0241	Invalid checksum, IBM CMOS area	61XY	RAM address line XY stuck
0280	Invalid checksum, HP CMOS area		Some address lines to RAM stuck to 0 or 1
02XY	A CMOS register is bad; Register = XY - 40		XY = 00bb bbbb = > address bbbbbb stuck
0301	8042 failed to accept the reset command		XY = 01bb bbbb = > Multiple address lines are stuck (bbbbbb is the first bad one)
0302	8042 failed to respond to reset command	620X	Lower 640K parity error; Bank X
0303	8042 failed to reset		X = Address in 64K bank with parity error
0311	8042 failed to accept "WRITE CMD BYTE"		if X = 0 to y, U21 and/or U31 is/are bad
0312	8042 failed to accept the data of above cmd		if X = 8 to 9, U11 and/ore U41 is/are bad
0321	8042 failed to accept scancode from port 68	63XY	Parity error above 1MB; Bank XY
0322	8042 failed to respond to above scancode		Parity error during RAM test above first MB
0323	8042 responded incorrectly to scancode		XY = Address in 64K bank with parity error
0331	8042 failed to accept command from port 6A	6400	Parity generator failed to detect error
0332	8042 failed to generate SVC on port 67	71XY	Master 8259 failed R/W; bits XY
0333	8042 generated HPINT type on port 65		XY = bbbb bbbb + > bits (b = 1 is bad)
0334	8042 failed the R/W register on port 69	72XY	Slave 8259 failed R/W; bits XY
0335	8042 failed to generate HPINT on IRQ 15		XY = bbbb bbbb = > bits (b = 1 is bad)
0336	8042 failed to generate HPINT on IRQ 12	7400	Master 8259 failed interrupt
0337	8042 failed to generate HPINT on IRQ 11	7500	Slave 8259 failed interrupt

Code	Meaning	Code	Meaning
0338	8042 failed to generate HPINT on IRQ 10	9XYZ	Floppy drive controller error
0339	8042 failed to generate HPINT on IRQ 7		X=drive #
033A	8042 failed to generate HPINT on IRQ 5		Y=0=1st level error
033B	8042 failed to generate HPINT on IRQ 4		Z=0 Unsuccessful input from FD
033C	8042 failed to generate HPINT on IRQ 3		Z=1 Unsuccessful output to FDC
0341	8042 failed keyboard interface test comd		Z=2 Error while executing seek
0342	8042didn't respond to interface command		Z=3 Error during recalibrate
0343	Keyboard clock line stuck low		Z=4 Error verifying RAM buffer
0344	Keyboard clock line stuck high		Z=5 Error while resetting FDC
0345	Keyboard data line stuck low		Z=6 Wrong drive identified
0346	Keyboard data line stuck high		Z=7 Wrong media identified
0350	No ACK from keyboard self test command		Z=8 No interrupt from FDC
0351	Bad ACK from keyboard self test command		Z=9 Failed to detect track 0
0352	Keyboard is dead or not connected		Z=A Failed to detect index pulse
0353	No result from keyboard self test command		Y>1=Higher level error
0354	Keyboard self test failed		Y=1=Read sector error, side 0
0401	8042 failed to enable gate A-20		Y=2=Read sector error, side 1
0503	Serial port dead or non existent		Y=3=Write sector error, side 0
0505	Serial port fails port register tests		Y=4=Write sector error, side 1
0543	Parallel port dead or non-existent		Y=5=Format sector error, side 0
06XX	Stuck key; XX=scancode of key		Y=6=Format sector error, side 1
0700	Failed to switch to slow mode		Y=7=Read ID error, side 0
0701	Failed to switch to dynamic mode		Y=8=Read ID error, side 1
0702	Timer (channel 0) failed to interrupt		Z=1=No ID address mark
0703	Memory cycles too slow in slow mode		Z=2=No data address mark
0704	Memory cycles too fast in slow mode		Z=3=Media is write protected
0705	I/O cycles too slow in slow mode		Z=4=Sector # wrong
0706	I/O cycles too fast in slow mode		Z=5=Cylinder # wrong
0707	Memory cycles too slow in dynamic mode		Z=6=Bad cylinder
0708	Memory cycles too fast in dynamic mode		Z=7=DMA overrun
0709	I/O cycles too slow in dynamic mode		Z=8=ID CRC error
070A	I/O cycles too fast in dynamic mode		Z=9=Data CRC error
110X	Timer channel failed to register test (X)		Z=A=End of cylinder
1200	Memory refresh signal stuck high		Z=B=Unrecognizable error
1201	Memory refresh signal stuck low	A001	No 80287 detected
211X	DMA 1 failed R/W test at register x (0-7)	A002	80287 failed stack register R/W test
212X	DMA 2 failed R/W test at register x (0-7)	A00C	No zero-divide interrupt from 80287
221X	Bad DMA page register; X=register 0-7	CXYZ	R/W error on extended RAM in XY bank
300X	HP-HIL controller failed self test; X=data		Read/Write test failure on extended RAM
	X = xx1 = >read/write fail with data = 0DA5h		X = 0 = > Even byte is bad
	X = xx1x = >R/W fail with data = 0DA5h		X = 1 = > Odd byte is bad
	X = x1xx = >R/W fail with data = 0DA5h		XY = 64K bank where RAM failed
	X = 1xxx = >R/W fail with data = 0DA5h	CFFF	Extended RAM marching ones failed
3010	HP-HIL device test failed		Marching on test failure on extended RAM
4XYZ	Lower 640K failed R/W test;		X = 0 = > Even byte bad
	X=0,2,4,6 Y>0=Bad U23 Z>0=Bad U13		X = 1 = > Odd byte bad
	X=1,3,5,7 Y>0=Bad U43 Z>0=Bad U33		XA = 64K bank where RAM failed
	X=8 Y>0=Bad U22 Z>0=Bad U12		

Vectra QS & RS

Code	Meaning	Code	Meaning
000F	386 CPU bad	620X	Lower 640K parity error; Bank X
0010	Bad checksum on ROM 0		X = Address in 64K bank where parity occurred
0011	Bad checksum on ROM 1	63XY	Parity error above 1MB; Bank XY
011X	RTC register is bad		XY = Address in 64K bank where parity occurred
0120	RTC failed to tick	6500	Shadow RAM bad at BIOS segment
0240	CMOS/RTC lost power	5XYZ	Lower 640K failed marching ones test
0241	Invalid checksum, IBM CMOS area		RAM in lower 640K failed read/write test
0280	Invalid checksum, HP CMOS area		X = bbcc = > bb is # 64K of 32 bit word bank
02XY	Bad CMOS register, at XY-40		cc = 00 = > byte 0 is bad
0301	8042 failed to accept reset command		cc = 01 = > byte 1 is bad
0302	8042 failed to respond to reset		cc = 02 = > byte 2 is bad
0303	8042 failed on reset		cc = 03 = > byte 3 is bad
0311	8042 didn't accept "WRITE CMD BYTE"		YZ = bbbb bbbb = > bits (b = 1 are bad)
0312	8042 didn't accept data		
0321	8042 failed to accept scancode, port 68	6510	Shadow RAM bad at HP EGA segment
0322	8042 failed to respond to scancode	71XY	Master 8259 failed R/W; bits XY
0323	8042 responded incorrectly to scancode		XY = bbbb bbbb = > bits which b = 1 is bad
0331	8042 failed to accept command from port 6A	72XY	Slave 8259 failed R/W; bits XY
0332	8042 failed to generate SVC on port 67		XY = bbbb bbbb = > bits which b = 1 is bad
0333	8042 generated HPINT type on port 65	7400	Master 8259 failed interrupt
0334	8042 failed the R/W register on port 69	7500	Slave 8259 failed interrupt
0335	8042 failed to generate HPINT on IRQ 15	9XYZ	Floppy drive controller error
0336	8042 failed to generate HPINT on IRQ 12		X=drive #
0337	8042 failed to generate HPINT on IRQ 11		Y=0=1st level error
0338	8042 failed to generate HPINT on IRQ 10		Z=0 Unsuccessful input from FD
0339	8042 failed to generate HPINT on IRQ 7		Z=1 Unsuccessful output to FDC
033A	8042 failed to generate HPINT on IRQ 5		Z=2 Error while executing seek
033B	8042 failed to generate HPINT on IRQ 4		Z=3 Error during recalibrate
033C	8042 failed to generate HPINT on IRQ 3		Z=4 Error verifying RAM buffer
0341	8042 failed interface test command		Z=5 Error while resetting FDC
0342	8042didn't respond to interface command		Z=6 Wrong drive identified
0343	Keyboard clock line stuck low		Z=7 Wrong media identified
0344	Keyboard clock line stuck high		Z=8 No interrupt from FDC
0345	Keyboard data line stuck low		Z=9 Failed to detect track 0
0346	Keyboard data line stuck high		Z=A Failed to detect index pulse
0350	No ACK from keyboard self test command		Y>1=Higher level error
0351	Bad ACK from keyboard self test command		Y=1=Read sector error, side 0
0352	Keyboard is dead or not connected		Y=2=Read sector error, side 1
0353	No result from keyboard self test command		Y=3=Write sector error, side 0
0354	Keyboard self test failed		Y=4=Write sector error, side 1
0401	8042 failed to enable gate A-20		Y=5=Format sector error, side 0
0503	Serial port dead or non-existent		Y=6=Format sector error, side 1
0505	Serial port fails port register tests		Y=7=Read ID error, side 0
06XX	Stuck key; XX=scancode of key		Y=8=Read ID error, side 1
0700	Failed to switch to slow speed		Z=1=No ID address mark

Code	Meaning	Code	Meaning
0701	Failed to switch to fast speed		Z=2=No data address mark
0702	Timer failed to interrupt		Z=3=Media is write protected
0703	CPU clock too slow in slow speed		Z=4=Sector # wrong
0704	CPU clock too fast in slow speed		Z=5=Cylinder # wrong
0707	CPU clock too slow in fast speed		Z=6=Bad cylinder
0708	CPU clock too fast in fast speed		Z=7=DMA overrun
0709	Failed to switch bus clock to ATCLK		Z=8=ID CRC error
110X	Timer X (0-2) failed to register test		Z=9=Data CRC error
1200	Memory refresh signal stuck high		Z=A=End of cylinder
1201	Memory refresh signal stuck low		Z=B=Unrecognizable error
211X	DMA 1 failed R/W test at register x (0-7)	A001	No 80287 detected
212X	DMA 2 failed R/W test at register x (0-7)	A002	80287 failed stack register R/W test
221X	Bad DMA page register; X=register 0-7	A00C	No zero-divide interrupt from 80287
300X	HP-HIL controller failed self test; X=data	AF00	Weitek coprocessor didn't enter protected mode
	X = xxx1 = > R/W fail with data = 0DA5Ah	AF01	Weitek coprocessor nor present
	X = xx1x = > R/W fail with data = 0DA5Ah	AF02	Weitek coprocessor fails register test
	X = x1xx = > R/W fail with data = 0DA5Ah	AF05	Weitek coprocessor fails addition test
	X = 1xxx = > R/W fail with data = 0DA5Ah	AF06	Weitek coprocessor fails interrupt test
3010	HP-HIL device test failed	AF0C	Weitek coprocessor fails interrupt test
4XYZ	Lower 640K failed R/W test;	CXYZ	R/W error on extended RAM in XY bank
	X=0,2,4,6 Y>0=Bad U23 Z>0=Bad U13		X = 0 = > Even byte bad
	X=1,3,5,7 Y>0=Bad U43 Z>0=Bad U33		X = 1 = > Odd byte bad
	X=8 Y>0=Bad U22 Z>0=Bad U12		XY = 64K bank where RAM failed
	X=9 Y>0=Bad U42 Z>0=Bad U32	CFFF	No extended RAM found
61XY	RAM address line XY stuck	EXYZ	Extended RAM marching 1s failure at XYZ
	Some address lines to RAM stuck to 0 or 1		X = 0 = > Byte 0 is bad
	XY = 00bb bbbb = > line bbbbbb is stuck		X = 1 = > Byte 1 is bad
	XY = 01bb bbbb = > Multiple address lines are stuck bbbbbb is the first bad one		X = 2 = > Byte 2 is bad
			X = 3 = > Byte 3 is bad

Pavilion Series 3100 & 8000

Code	Meaning	Code	Meaning
02	Verify real mode	7C	Set up hardware interrupt vectors
03	Disable NMI	7E	Initialize coprocessor, if present
04	Get processor type	80	Disable onboard super I/O ports
06	Initialize system hardware	81	Late POST device initialization
08	Initialize chipset with POST values	82	Detect & install external RS-232 ports
09	Set IN-POST flags	83	Configure non-MDC IDE controllers
0A	Initialize CPU registers	84	Detect & install external parallel ports
0B	Enable CPU registers	85	Initialize PnP ISA devices
0C	Initialize cache to POST values	86	Reinitialize onboard I/O ports
0E	Initialize I/O component	87	Set motherboard configurable devices
0F	Initialize local IDE bus	88	Initialize BIOS data area
10	Initialize power management	89	Enable NMI's
11	Load alternate registers	8A	Initialize extended BIOS data area
12	Restore CPU control word during warm boot	8B	Test & initialize PS/2 mouse

Code	Meaning	Code	Meaning
13	Initialize PCI bus mastering devices	8C	Initialize floppy controller
14	Initialize keyboard controller	8F	Determine number of ATA drives
16	BIOS ROM checksum	90	Initialize hard disk controllers
17	Initialize cache before memory size	91	Initialize local BUS HD controllers
18	Initialize 8254 timer	92	Jump to user patch 2
1A	Initialize DMA controller	93	Build MPTABLE for multiprocessor boards
1C	Reset PIC	94	Disable A-20 line
20	Test DRAM refresh	95	Install CD-ROM for boot
22	Test 8742 keyboard controller	96	Clear huge ES segment register
24	Set ES segment register to 4GB	97	Fix up multiprocessor table
26	Enable A-20 line	98	Search for options ROM's
28	Autosize DRAM	99	Check for smart drive
29	Initialize POST memory manager	9A	Shadow ROM option
2A	Clear 512K base RAM	9C	Set up power management
2C	RAM address line failure	9E	Enable hardware interrupts
2E	RAM data failure, low byte	9F	Determine number of ATA & SCSI drives
2F	Enable cache before BIOS shadow	A0	Set time of day
30	RAM data failure, high byte	A2	Check key lock
32	Test CPU, BUS clock frequency	A4	Initialize typematic rate
33	Initialize POST dispatch manager	A8	Erase F2 prompt
34	Test CMOS RAM	AA	Scan for F2 keystroke
35	Initialize alternate chipset registers	AC	Enter SETUP
36	Warm start shut-down	AE	Clear IN-POST flag
37	Reinitialize chipset (MB only)	B0	Check for errors
38	Shadow system BIOS ROM	B2	POST done, prepare for boot
39	Reinitialize cache (MB only)	B4	One short beep before boot
3A	Autosize cache	B5	Terminate quiet boot
3C	Configure advanced chipset registers	B6	Check password (optional)
3D	Load alternate registers new CMOS values	B8	Clear global descriptor table
40	Set initial CPU speed	B9	Clean up all graphics
42	Initialize interrupts	BA	Initialize DMI parameters
44	Initialize BIOS interrupts	BB	Initialize PnP option ROM's
45	POST device initialization	BC	Clear parity checkers
46	Check ROM copyright notice	BD	Display multi boot menu
47	Initialize manager for PCI option ROM's	BE	Clear screen optional
48	Check video config against CMOS	BF	Check virus and backup reminders
49	Initialize manager for PCI option ROM's	C0	Try to boot with Int 19
4A	Initialize all video adapters	C1	Initialize POST error manager
4B	Display quiet boot screen	C2	Initialize error logging
4C	Shadow video BIOS	C3	Initialize error display function
4E	Display BIOS copyright notice	C4	Initialize system error handler
50	Display CPU type & speed	E0	Initialize the chipset
51	Initialize	E1	Initialize the bridge
52	Test keyboard	E2	Initialize the processor
54	Set key click if enabled	E3	Initialize system timer
56	Enable keyboard	E4	Initialize system I/O
58	Test for unexpected interrupts	E5	Check force recovery boot
59	Initialize POST display service	E6	Checksum BIOS ROM
5A	Display "Press F2 to Enter Setup"	E7	Got to BIOS

Code	Meaning	Code	Meaning
5B	Disable CPU cache	E8	Set huge segment
5C	Test RAM, 512-640K	E9	Initialize multiprocessor
60	Test extended memory	EA	Initialize OEM special code
62	Test extended memory address lines	EB	Initialize PIC & DMA
64	Jump to user patch 1	EC	Initialize memory type
66	Configure advanced cache registers	ED	Initialize memory type
67	Initialize multi-processor APIC	EE	Shadow boot block
68	Enable external & processor caches	EF	System memory test
69	Set up SMM area	F0	Initialize interrupt vectors
6A	Display external L2 cache size	F1	Initialize runtime clock
6C	Display shadow area message	F2	Initialize video
6E	Display high address for UMB recovery	F3	Initialize beeper
70	Display error message	F4	Initialize BOOT
72	Check for configuration errors	F5	Clear huge segment
74	Test real time clock	F6	Boot to mini-DOS
76	Check for keyboard errors	F7	Boot to full DOS
7A	Test for key lock on		

IBM

Tests are performed by PC/XT/AT and PS/2 machines. There will be POST Codes (below), beep codes and screen displays if possible, but the XT does not give POST codes. ATs emit codes to 80h, while PS/2 models 25 and 30 emit to 90h, and 35 and above to 680. The BIOS will test main system components first, then non-critical ones. If there is an error, the BIOS will look for a reference diskette in drive A: so diagnostics can be performed.

IBM POST I/O Ports

Architecture	Typical Computer	Port
PC	PC	none
ISA	XT	60
	AT	80
	PS/2 25,30	90, 190
MCA	PS/2 50 up	680, 3BC
EISA	none	none

POST Procedures

Procedure	Meaning
CPU	Perform register test on the CPU by writing data patterns to the registers and reading the results of the write.
BIOS Checksum	The value of bits inside the BIOS chip(s) is added to a preset value that should create a total of 00.
CMOS RAM	RAM within the CMOS chip is tested by writing data patterns to the area and verifying that the data was stored correctly.
DMA	Test DMA chips by forcing control inputs to the CPU to an active state and verifying that the proper reactions occur.

Procedure	Meaning
8042/8742 Keyboard Controller	Test including Gate A20 and the reset command. The buffer space is prepared and data is sent to the determined area via the keyboard controller to see if commands are received and executed correctly.
Base 64K System RAM.	Perform a walking-bit test of the first 64K of RAM so the BIOS vector area can be initialised. Check for bad RAM chips or a data/address line.
8259A PIC	Determine if commands to interrupt CPU processes are carried out correctly. Check the PIC/PIT/RTC/CMOS or Clock chip(s).
8254 PIT	Check that proper setup and hold times are given to the PIC for interrupts of the CPU processes. Check the PIT or Clock chip.
82385 Cache Controller	This is normally responsible for cache and shadow memory.
CMOS RAM Configuration Data	Check information in CMOS RAM before further testing so any failures after this could also be down to the CMOS chip.
CRT controllers	Test any video adapters listed in the CMOS.
RAM above 64K	Perform a walking-bit test on memory above 64K listed in the CMOS.
Keyboard	Test interface to keyboard including scan code stuck keys etc.
Pointing Device (mouse etc)	Test and init vector for pointing devices. Failure to see a device may be the device itself but there may be a problem with the CMOS or 8042/8742.
Diskette Drive A:	Test and initialise the A: drive.
Serial Interface Circuitry	Test any RS232 devices found at the proper I/O address.
Diskette Controllers	If an A: drive has been found further testing is performed before proceeding to the bootloader. This test includes reading the first sector of any diskette in the drive to see if a valid boot code is there.
Fixed Disk Controllers	Test and initialise any hard drives set in the CMOS including reading the first sector of the hard drive to see if a valid boot code exists.

XT (Port 60)

The PC uses an irregular way of sending codes to ports 10 and 11, which is impractical to monitor on a POST card. The XT, on the other hand, uses three methods; before initializing the display, it issues a few codes to port 60 (the 8255 controller for the keyboard) for critical system board errors. It beeps to indicate successful or unsuccessful POST, and displays error messages. After initializing the display, it writes error codes to memory address 15, which are sent to the screen as part of other error messages.

Code	Meaning
00 or FF	CPU register test failed
01	BIOS ROM (ROS) checksum failed
02	Timer I failed
03	8237 DMA register write/read failed or unexpected timer 1 request for DMA ch 1
04	After enabling port 213 expansion box, base 32K memory write/read of AA, 55, FF, 01 and 00 test failed; POST output alternates between POST code and failing bit pattern.
	Size memory, init 8259 PIC, setup interrupt vectors in RAM, read configuration switches, poll jumper. If installed, load manufacturing test via keyboard port and run. If not, initialize rest of system.

AT POST Codes

Code	Meaning
00	Main board damaged
01	80286 test in real mode; verify read/write registers, flags and conditional jumps.
02	ROM checksum test-test 32K ROMs; POST BASIC and BIOS.
03	Test CMOS shutdown byte-rolling bit pattern and verified at shutdown address.
04	8254 timer 1; all bits on; set timer count; check all bits on.

Code	Meaning
05	8254 timer 1; all bits off; set timer count; check all bits off.
06	8237 DMA 0 init channel register test. Disable DMA controller; r/w current address to all channels
07	8237 DMA 1 init channel register test. Disable DMA controller; r/w current address to all channels
08	DMA page register test-r/w all page registers. Check 74LS612.
09	Storage refresh test. 8042 i/face test I/O issue self test; check 55H received
0A	Keyboard controller test 1; Soft reset
0B	Keyboard controller test 2; Reset 8042
0C	Keyboard controller test 3; Test switch settings
0D	Keyboard controller test 4: Write byte 0 of 8042 mem; issue comd to 8042, await response.
0E	Base 64K r/w memory test-r/w data patterns AAh, 55h.
0F	Get I/P buffer switch setting. Also Base 64K r/w memory test #2-r/w data patterns AAh, 55h.
10	Roll error code to MFG Port
11	Initialise display row count. Verify 286 LGDT.SGDT LIDT/SIDT instruction
12	Protected mode register test failure
13	Initialise 8259
14	Setup interrupt vector to temp interrupt
15	Establish BIOS interrupt call subroutine vectors. CMOS checksum/battery OK
16	Set data segment or Check CMOS battery condition.
17	Set defective battery flag or CMOS checksum error.
18	Ensure CMOS dividers set or enable protected mode.
19	Set return address byte in CMOS.
1A	Set temporary stack or protected mode test. Determine memory size; verify parity.
1B	Segment address 01-0000 (second 64K memory test)
1C	Set or reset; check 512-640 memory installed
1E	Set (expanded?) memory size determined in CMOS; or determine memory size above 1024K.
1F	Test address lines 19-23
20	Fatal addressing error; Shutdown.
21	Return 1 from shutdown. Initialise and start CRT controller (6845); test video r/w; test video enable; select alpha mode; w/r patterns; or check CMOS config data.
22	Enable video signal and set mode; CRT interface test; verify video enable and horizontal sync. Video card init failure or invalid switch setting.
23	Check for advanced video card; Video card initialisation failure or invalid switch setting.
24	8259 PIC test -r/w interrupt mask register with 1s and 0s; mask device interrupts off.
25	Check for hot interrupts; test interrupt mask registers.
26	Display 101 error; Check for unexpected interrupts.
27	Check the converting logic (106 error)
28	Check hot NMI interrupts (error 107)
29	Test data bus to timer 2 (error 108). 8253 timer register failure.
2A	8253 Timer speed failure (error 102)
2B	Too fast; or 8253 Timer interrupt initialisation.
2C	Too slow; or Timer 0 interrupt failure (error 103)
2D	Check 8042 (k/b controller) for last command excepted (error 105)
2F	Check for warm boot
30	Set shutdown return 2; Protected mode r/w memory test step 1.
31	Enable protected mode; Protected mode r/w memory test step 2.
32	Address lines 0-15
33	Next block of 64K; Protected mode r/w memory test step 3.
34	Restore checkpoint; Protected mode r/w memory test step 4.
35	Keyboard test; Check for manufacturing burn in test.

Code	Meaning
36	Check <AA> scan code; keyboard clock error.
38	Error-check 8042 working; also 37 and 39
3A	Initialise 8042; keyboard locked
3B	Check for ROM in 2K blocks
3C	Check for floppy diskette drive
3D	Initialise floppy for drive type
3E	Initialise hard drive
3F	Initialise printer; non-fatal error; press F1 to continue.

Additional Protected Mode Tests

Code	Meaning
40	Enable hardware interrupt if 80287; initialisation
41	System code @ segment code E000.0
42	Exit to system code
43	Go to boot loader diskette attachment test
44	Boot from fixed disk
45	Unable to boot; go to BASIC
81	Build descriptor table
82	Switch to virtual mode
90-B6	EXEC_00 to EXEC_31 & SYS_32 to SYS_38 tests; memory test; boot loader.
DD	Transmit error code to MFG_PORT
F0	Set data segment
F1	Interrupt test (programming interrupt 32)
F2	Exception interrupt test
F3	Verify 286 LDT/SDT and LTR/STR instructions.
F4	Verify 286 bound instruction
F5	Verify push and pop all instruction; stack/register test.
F6	Verify access rights function correctly.
F7	Verify Adjust RPL field of selector instructions (ARPL) functions
F8	Verify LAR function
F9	Verify LSL i(Load Segment Limits) instruction
FA	Low meg chip select test

PS/2 (Micro Channel) POST Codes

Code	Meaning
00	CPU test; FFAA0055 pattern
01	32 bit CPU register test; setup system timer
02	System ROM checksum
03	Test system enable/system port 94 enable/check
04	Test system POS register; port 102 enable/check
05	Test adapter setup port; POS port 96 enable/check
06	Test RTC/CMOS shutdown byte; Byte 0F CMOS (NMI disable)
07	Test extended CMOS location; ports 74-76 test
08	Test DMA & page register 8 channels; ports 2
09	Initialise DMA command & mode registers
0A	Test refresh (port 61)

Code	Meaning
0B	Test keyboard controller buffers (8042-port 61
0C	Keyboard controller self test (8042-port 60)
0D	Keyboard controller test continuation (8042)
0E	Keyboard self test error indicated (port 64)
0F	Setup system memory configuration
10	Test first 512K RAM in real mode
11	Half system if memory test error
12	Verify LGDT/SGDR LIDT/SIDT (keyboard commands)
13	Initialise PIC #1 (Master)
14	Initialise PIC #2 (Slave)
15	Initialise A20 interrupt vectors
16	Setup extended vector table
17	Check power RTC/CMOS power good signal (byte 0D)
18	Check RTC/CMOS checksum
19	RTC/CMOS lost power (0D 80h)
1A	Skip memory test in protected mode if warm reset
1B	Prepare for shutdown; protected mode initialisation
1C	Setup stack pointer point to the end of first 64K
1D	Decide low memory size in protected mode; Size base memory
1E	Save memory size detected
1F	Setup system memory split address
20	Check for extended memory beyond 64 Mb
21	Test memory address bus lines
22	Clear parity error and channel check; Disable NMI
23	Initialise interrupt 00; system timer
24	Determine CMOS validity
25	Write keyboard controller (8042) command byte
40	Check valid CMOS and video
41	Display error code 160. Check CMOS, AC ripple.
42	Test PIC #1 & PIC #2 registers; Master/Slave test
43	Test PIC #1 & PIC #2 registers with another pattern
44	Check for interrupt with interrupt masked; check for NMI when disabled.
45	Test NMI
46	NMI error detected
47	Test system timer 0
48	Check stuck speaker clock; speaker bitstuck test
49	Test timer 0 count
4A	Test timer 2 output
4B	Check if timer interrupt occurred
4C	Test timer 0 for count too fast or slow
4D	Verify timer 0 interrupt
4E	Check 8042 ready for command; buffer free
4F	Check for soft reset
50	Prepare for shutdown/protected mode
51	Start protected mode test
52	Test memory in 64K increments
53	Check if memory test done
54	Shutdown system and return to real mode
55	Test for manufacture or regular test; test for loop. Check jumper.

Code	Meaning
56	Disable keyboard
57	Check for keyboard self test
58	Keyboard test passed; check for errors
59	Test keyboard interface
5A	Initialise mouse
5B	Disable mouse
5C	Initialise interrupt vectors
5D	Initialise interrupt vectors
5E	Initialise interrupt vectors
5F	BIOS data area
60	Determine diskette rate
61	Reset floppy controller/drive
62	Floppy drive test
63	Turn floppy motor off
64	Serial port setup
65	Enable/test RTC interrupt
66	Configure floppy drives
67	Configure hard drive
68	Enable system CPU arbitration; wait states
69	Scan for optional ROMs
6A	Verify serial and parallel ports
6B	Setup equipment byte
6C	Setup configuration errors reported
6D	Set keyboard typematic rate
6E	Reset page register; boot up system (Int 19 bootloader)
70	Reset disk
71	Read bootcode for E6/E9
72	Control to bootcode
73	Bootcode/ROM Basic

INTEL

CA810E

Also RC440BX & SR 440BX. Possibly based on AMI.

Code	Meaning
D0	NMI disabled. Keyboard controller and RTC enabled. Initialization code checksum verification starting
D1	Keyboard controller BAT test, CPU ID saved, and going to 4GB flat mode
D3	Initialize chipset, start memory refresh, and determine memory size
D4	Verify base memory
D5	Initialization code to be copied to segment 0 and control to be transferred to segment 0
D6	Control is in segment 0. Used to check if in recovery mode and verify main BIOS checksum. If in recovery mode or if checksum is wrong, go to E0 for recovery. Otherwise, got o D7 to give control to main BIOS
D7	Find main BIOS module in ROM image
D8	Uncompress the main BIOS module
D9	Copy main BIOS image to F000 shadow RAM and give control to main BIOS in F000 shadow RAM

Boot Block Recovery Code Checkpoints

Code	Meaning
E0	Onboard diskette controller (if any) is initialized. Compressed recovery code is uncompressed at F000:0000 in shadow RAM. Give control to recovery code at F000 in shadow RAM. Initialize interrupt vector tables, system timer, DMA controller, and interrupt controller
E8	Initialize extra (Intel recovery) module
E9	Initialize diskette drive
EA	Try to boot from diskette. If reading of boot sector is successful, give control to boot sector code
EB	Boot from diskette failed; look for ATAPI (LS-120, Zip) devices
EC	Try to boot from ATAPI device. If reading boot sector is successful, give control to boot sector code
EF	Boot from diskette and ATAPI device failed. Give2 beeps. Retry booting procedure (go to checkpoint E9)

Runtime Code Uncompressed in F000 Shadow RAM

Code	Meaning
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	Keyboard controller I/B free. To issure the BAT command to the keyboard controller
0E	Any initialization after keyboard controller BAT to be done next
0F	Keyboard command byte written
10	Going to issue Pin 23, 24 blocking/unblocking command
11	Going to check pressing of <INS>, <END> key during power-on
12	To init CMOS if "Init CMOS in every boot" is set or <END> 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 15 µs On/OFF time
23	To read 8042 input port and disable megakey GreenPC feature. make BIOS cade segment writeable
24	To do any setup before int vector init
25	Interrupt vector initialization 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
2B	To give control for any setup required before optional 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
2F	EGA/VGA not found. Display memory R/W test about to begin
30	Display memoryR/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 Disaply 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

Code	Meaning
39	Display different buses initialization error messages
3A	New cursor position read and saved. To display the Hit <DEL> message
40	To prepare the descriptor tables
42	To enter 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 test memory
46	Memory wrap around test done. 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 640KB 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 # 4E)
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 checl point # 52
4E	Memory test started. (NOT SOFT RESET) About to display the first 64KB memory size
4F	Memory size display started. Will be updated during memory test. going for sequential and random memory test
50	Memory testing/iniialization below 1MB complete. Adjust displayed memory size for relocation shadow
51	Memory size display adjusted due to relocation/shadow. Memory test above 1MB to follow
52	Memory testing/initialization above 1MB complete. Going to save memory size information
53	Memory size information is saved. CPU registers are saved. Going to enter 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 <DEL> message
59	Hit <DEL> message cleared. <WAIT...> message displayed. Start DMA and interrupt controller test
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 #1 and #2
66	DMA #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	Passwoed 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. <WAIT...> message. PS/2 mouse check and extended BIOS data area allocation
8C	Setup options programming after CMOS setup about to start
8D	Going to 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.
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

Code	Meaning
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	Iinitialization after optional ROM test over. Going to setup timer data area and printer base address
9A	Return after setting timer and printer base addresses. 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 initializae the coprocessor test
9D	Coprocessor initialized. Goin to do any initialization after coprocessor test
9E	Initialization after copro test is complete. Goin to check extended keyboard, keyboard ID and num-lock
A2	Going to display any soft errors
A3	Soft error display complete. Going to set keyboard typematic rate
A4	Keyboard typematic rate set. To program memory wait states
A5	Going to enable parity/NMI
A7	NMI and parity enabled. 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	Put Int13 module runtime image to shadow
AC	Generate MP for multiprocessor support (if present)
AD	Put CGA Int10 module (if present) in shadow
AE	Uncompress SMBIOS module and init SMBIOS code and form the runtime SMBIOS image in shadow
B1	Going to copy any code to specific area
00	Copying of code to specific area done. Going to give control to Int19 boot loader

440 Series

Code	Meaning	Code	Meaning
02	Verify real mode	82	Detect and install external RS232 ports
03	Disable non-maskable interrupt (NMI)	83	Configure non-MCD IDE controllers
04	Get processor type	84	Detect and install external parallel ports
06	Initialize system hardware	85	Initialize PC compatible PnP ISA devices
08	Initialize chipset with initial POST values	86	Reinitialize onboard I/O ports
09	Set IN POST flag	87	Set motherboard configurable devices
0A	Initialize processor registers	88	Initialize BIOS Data Area
0B	Enable processor cache	89	Enable non-maskable interrupts (NMI)
0C	Initialize caches to initial POST values	8A	Initialize extended BIOS data area
0E	Initialize I/O component	8B	Test and initialize PS/2 mouse
0F	Initialize the local bus IDE	8C	Initialize diskette controller
10	Initialize power management	8F	Determine number of ATA drives
11	Load alternate registers initial POST values	90	Initialize hard disk controllers
12	Restore proc control word in warm boot	91	Initialize local bus hard disk controllers
13	Initialize PCI bus mastering devices	92	Jump to UserPatch2
14	Initialize keyboard controller	93	Build MPTABLE for multiprocessor boards
16	BIOS ROM checksum	94	Disable A20 address line
17	Initialize cache before memory autosize	95	Install CD-ROM for boot
18	8254 timer initialization	96	Clear huge ES segment register
1A	8237 DMA controller initialization	97	Fix up multiprocessor table
1C	Reset programmable interrupt controller	98	Search for option ROMs
20	Test DRAM refresh	99	Check for SMART drive
22	Test keyboard controller	9A	Shadow option ROMs

Code	Meaning	Code	Meaning
24	Set ES segment register to 4GB	9C	Setup power management
26	Enable A20 line	9E	Enable hardware interrupts
28	Autosize DRAM	9F	Determine number of ATA and SCSI drives
29	Initialize POST memory manager	A0	Set time of day
2A	Clear 512KB base RAM	A2	Check key lock
2C	RAM failure on address line xxxx	A4	Initialize typematic rate
2E	RAM failure on bits xxxx of low byte	A8	Erase F2 prompt
2F	Enable cache before system BIOS shadow	AA	Scan for F2 key stroke
30	RAM failure on bits xxxx of high byte	AC	Enter SETUP
32	Test processor bus-clock frequency	AE	Clear IN POST flag
33	Initialize POST dispatch manager	B0	Check for errors
34	Test CMOS RAM	B2	POST done - prepare to boot system
35	Initialize alternate chipset registers	B4	One short beep before boot
36	Warm start shut down	B5	Terminate QuietBoot
37	Reinitialize the chipset (motherboard only)	B6	Check password (optional)
38	Shadow system BIOS ROM	B8	Clear global descriptor table
39	Reinitialize the cache (motherboard only)	B9	Clean up all graphics
3A	Autosize cache	BA	Initialize DMI parameters
3C	Configure advanced chipset registers	BB	Initialize PnP Option ROMs
3D	Load alternate registers with CMOS values	BC	Clear parity checkers
40	Set initial processor speed new	BD	Display MultiBoot menu
42	Initialize interrupt vectors	BE	Clear screen (optional)
44	Initialize BIOS interrupts	BF	Check virus and backup reminders
45	POST device initialization	C0	Try to boot with Int19h
46	Check ROM copyright notice	C1	Initialize POST Error Manager (PEM)
47	Initialize manager for PCI option ROMs	C2	Initialize error logging
48	Check video configuration against CMOS	C3	Initialize error display function
49	Initialize PCI bus and devices	C4	Initialize system error handler
4A	Initialize all video adapters in system	D0	Interrupt handler error
4B	Display QuietBoot screen	D2	Unknown interrupt error
4C	Shadow video BIOS ROM	D4	Pending interrupt error
4E	Display BIOS copyright notice	D6	Initialize option ROM error
50	Display processor type and speed	D8	Shutdown error
51	Initialize EISA motherboard	DA	Extended Block Move
52	Test keyboard	DC	Shutdown 10 error
54	Set key click if enabled	E0	Initialize the chipset
56	Enable keyboard	E1	Initialize the bridge
58	Test for unexpected interrupts	E2	Initialize the processor
59	Initialize POST display service	E3	Initialize system timer
5A	Display prompt "Press F2 to enter "SETUP"	E4	Initialize system I/O
5B	Disable processor cache	E5	Check force recovery boot
5C	Test RAM between 512 and 640KB	E6	Checksum BIOS ROM
60	Test extended memory	E7	Go to BIOS
62	Test extended memory address lines	E8	Set huge segment
64	jump to UserPatch1	E9	Initialize multiprocessor
66	Configure advanced cache registers	EA	Initialize OEM special code
67	Initialize multiprocessor APIC	EB	Initialize PIC and DMA
68	Enable external and processor caches	EC	Initialize memory type
69	Setup SMM area	ED	Initialize memory speed

Code	Meaning	Code	Meaning
6A	Display external L2 cache size	EE	Shadow boot block
6C	Display shadow area message	EF	System memory test
6E	Display possible high address for UMB recovery	F0	Initialize interrupt vectors
70	Display error messages	F1	Initialize runtime clock
72	Check for configuration errors	F2	Initialize video
74	Test real-time clock	F3	Initialize beeper
76	Check for keyboard errors	F4	Initialize boot
7A	Test for key lock on	F5	Clear huge segment
7C	Setup hardware interrupt vectors	F6	Boot to mini-DOS
7E	Disable onboard Super I/O ports and IRQs	F7	Boot to full DOS
81	Late POST device initialization		

440GX

Codes in Execution Sequence.

Code	Meaning
02	Verify real mode
12	Restore processor control word during warm boot (only occurs on warm boot)
24	Set ES segment register to 4GB
04	Get processor type
06	Initialize system hardware
18	8254 timer initialization
08	Initialize PCIset registers with initial POST values
C4	Initialize system flags in CMOS
11	Load alternate registers with initial POST values
0E	Initialize I/O
0C	Initialize caches to initial POST values
16	BIOS ROM checksum
17	Turn cache off
28	Autosize DRAM
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 (1st 4Meg)
2F	Initialize L2 cache if enabled in CMOS
38	Shadow system BIOS ROM
20	Test DRAM refresh
29	Post Memory Manager initialization (PMM)
33	Post Dispatch Manager initialization
34	Test CMOS
C1	Post error manager initialization
09	Set IN POST flag
0A	Initialize processor registers and processor microcode
3A	Autosize cache
0B	Enable processor cache
0F	Initialize the local bus IDE (not used anymore but here for Phoenix standard)
10	Initialize Power Management (APM not used in L440GX+)
14	Initialize keyboard controller
1A	8237 DMA controller initialization

Code	Meaning
1C	Reset programmable interrupt controller (PIC)
22	Test 8742 keyboard controller
32	Read processor bus-clock frequency and compute boot processor speed
67	Initialize and register via SMM through APIC bus
69	Initialize SMI handler for all processors
00	Wait for secondary processor to execute init SMI handler
F4	Exit SMI handler (secondary processor executed halt in SMI)
3C	Configure advanced PCIset registers and reset coprocessor
3D	Load alternate registers with CMOS values
42	Initialize interrupt vectors
46	Check ROM copyright notice
45	Initialize all pre-PnP devices
49	Initialize PCI bus and devices (also read ESCD and allocate resources)
48	Check video configuration against CMOS (VGA or MDA)
4A	Initialize all video adapters in system
4C	Shadow video BIOS ROM
24	Put processor in big real mode (flat mode memory addressing - up to 4GB)
59	Post display manager initialization (video screen error codes now visible)
22	Reset and test keyboard first try (only warm reset)
52	Reset and test keyboard controller (both warm and cold reset)
54	Set key click if enabled
76	Enable keyboard
58	Test for unexpected interrupts
4B	QuietBoot start (not used in L440GX+)
4E	Display copyright notice
50	Display processor(s) type and speed
51	EISA init (not used in L440GX+)
5A	Display prompt "Press F2 to enter SETUP"
5B	Disable processor L1 cache for memory test
5C	Test RAM between 512KB and 640KB
60	Test extended memory (4MB to top of memory)
62	Test extended memory address lines
64	Jump to UserPatch1
66	Configure advanced cache registers
68	Enable external and processor caches
6A	Display external cache size
6C	Display shadow message
6E	Display non-disposable segments
70	Display error messages to video
72	Check for configuration errors
74	Test real time clock
7C	Setup hardware interrupt vectors
7E	Test coprocessor if present
80	Not used
88	Initialize BIOS Data Area, time-outs for detecting parallel, serial and HDD controller. Clear CMOS shutdown flag
8A	Initialize Extended BIOS Data Area
81	Late POST core initialization of devices
87	Configure MCD devices

Code	Meaning
85	Initialize and detect PC compatible PnP ISA devices (parallel, serial, etc..)
82	Not used
84	Clear interrupts from COM port detection
86	Console redirection initialized
83	Configure onboard hard disk controller
89	Enable NMI
8C	Initialize floppy controller
90	Initialize and detect hard disks
8B	Detect and test for mouse and auxiliary device on keyboard controller
95	Install CD-ROM for boot
92	Jump to UserPatch2
C5	Initialize GPNV areas in DMI
98	Search for option ROMs. One long, two short beeps on checksum failure of an option ROM
93	Scan for User Flash ROMs. MP Table initialization (wake up secondary processor and halt it)
9C	Setup Power Management (not used)
9D	Enable security
9E	Enable hardware interrupts
A0	Set time of day
A2	Check key lock
A4	Initialize typematic rate
C2	Initialize DMI tables
C3	Log post errors with POST error manager and to SEL in BMC. Update VID bits and memory presence to BMC. Display and FRB errors (watchdog time-outs, bits or processor failures)
A8	Erase F2 prompt
AA	Scan for F2 keystroke
AC	Initialize EMP port if selected. Remove COM2 from BDA if EMP is enabled. Enter SETUP
AE	Clear IN POST flag
B0	Secure boot if enabled (secure front panel, blank video, floppy write protect). Check for errors
B2	POST done - prepare to boot Operating System
B4	One short beep before boot
B5	Display QuietBoot (not used)
BE	Clear screen
B6	Check password (optional)
BC	Clear parity checkers
BA	Not used
B7	ACPI configuration (table configuration in memory and BDA)
BD	Display MultiBoot menu if ESC is hit
BF	Display system config summary (if enabled in CMOS)
8F	Get total # of hard drives and put in BDA
91	Program IDE hard drives (timing, PIO modes, etc...)
9F	Save total # of hard drives (SCSI and ATA) in BDA
97	Fixup MP table (checksum)
99	Check SMART hard drive
C7	Prepare to boot to OS. Clean up graphics and PMM areas
C0	Try to boot with Int19h. Return to video mode 3, disable PMM, return to real mode, disable gate A20, clears system memory, resets stack, invokes Int19h
D0	Interrupt handler error
D2	Unknown interrupt error
D4	Pending interrupt error

Code	Meaning
D6	Initialize option ROM error
D8	Shutdown error
DA	Extended Block Move
DC	Shutdown 10 error

440BX

Codes in Execution Sequence.

Code	Meaning
02	Verify real mode
12	Restore processor control word during warm boot (only occurs on warm boot)
24	Set ES segment register to 4GB
04	Get processor type
06	Initialize system hardware
18	8254 timer initialization
08	Initialize PCIset registers with initial POST values
C4	Initialize system flags in CMOS
11	Load alternate registers with initial POST values
0E	Initialize I/O
0C	Initialize caches to initial POST values
16	BIOS ROM checksum
17	Turn off cache
28	Autosize DRAM
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 (first 4 meg)
2F	Initialize L2 cache if enabled in CMOS
38	Shadow system BIOS ROM
20	Test DRAM refresh
29	Post Memory Manager initialization (PMM)
33	Post Dispatch Manager initialization
34	Test CMOS
C1	Post error manager initialization
09	Set IN POST flag
0A	Initialize processor registers and CPU microcode
3A	Autosize cache
0B	Enable processor cache
0F	Initialize the local bus IDE
10	Initialize Power Management (APM not used in Nightshade)
14	Initialize keyboard controller
1A	8237 DMA controller initialization
1C	Reset Programmable Interrupt Controller
22	Test 8742 Keyboard Controller
32	Read processor bus-clock frequency and compute boot processor speed
67	Initialize and register other CPU via SMM through APIC bus
69	Initialize SMI handler for all processors
00	Wait for secondary processor to execute init SMI handler
F4	Exit SMI handler (secondary processor executed halt in SMI)

Code	Meaning
3C	Configure advanced PCIset registers and reset coprocessor
3D	Load alternate registers with CMOS values
42	Initialize interrupt vectors
46	Check ROM copyright notice
45	Initialize all pre-PnP devices
49	Initialize PnP bus and devices (also read ESCD and allocate resources)
48	Check video configuration against CMOS (VGA or MDA)
4A	Initialize all video adapters in system
4C	Shadow video BIOS ROM
24	Put CPU in big real mode (flat mode memory addressing - up to 4GB)
59	Post display manager initialization (video screen error codes now visible)
22	Reset and test keyboard first try (only warm reset)
52	Reset and test keyboard controller (both warm and cold reset)
54	Set key click if enabled
76	Enable keyboard
58	Test for unexpected interrupts
4B	QuietBoot start (not used in N440BX)
4E	Display copyright notice
50	Display CPU(s) type and speed
51	EISA initialized (not used in N440BX)
5A	Display prompt "Press F2 to enter SETUP"
5B	Disable CPU L1 cache for memory test
5C	Test RAM between 512KB and 640KB
60	Test extended memory (4MB to top of memory)
62	Test extended memory address lines
64	Jump to UserPatch1
66	Configure advanced cache registers
68	Enable external and processor caches
6A	Display external cache size
6C	Display shadow message
6E	Display non-disposable segments
70	Display error messages to video
72	Check for configuration errors
74	Test real time clock
7C	Setup hardware interrupt vectors
7E	Test coprocessor if present
80	Not used
88	Init Data Area, time-outs for detecting parallel, serial and HD controller. Clear shutdown flag
8A	Initialize Extended BIOS Data Area
81	Late POST core initialization
87	Configure MCD devices
85	Initialize and detect PC compatible PnP ISA devices (serial, parallel, etc.)
82	Not used
84	Clear interrupts from COM port detection
86	Console redirection initialized
83	Configure onboard hard disk controller
89	Enable NMI
8C	Initialize floppy controller
90	Initialize and detect hard disks

Code	Meaning
8B	Detect and test for Mouse or Auxiliary device on keyboard controller
95	Install CD-ROM for boot
92	Jump to UserPatch2
C5	Initialize GPNV areas of DMI
98	Search for option ROMs. One long, two short beeps for checksum failure of an option ROM
93	Scan for User flash ROMs. MP Table initialization (wake up secondary processor and halt it)
9C	Setup Power Management (not used)
9D	Enable security
9E	Enable hardware interrupts
A0	Set time of day
A2	Check key lock
A4	Initialize typematic rate
C2	Initialize DMI tables
C3	Log POST errors and to SEL in BMC. Update VID bits and memory presence to BMC. Display any FRB errors (watchdog time-outs, bits or CPU failures)
A8	Erase F2 prompt
AA	Scan for F2 keystroke
AC	Initialize EMP port if selected. Remove COM2 from BDA of EMP is enabled. Enter SETUP
AE	Clear IN POST flag
B0	Turn on secure boot if enabled (secure front panel, blank video, floppy WP). Check for errors
B2	POST done - prepare to boot Operating System
B4	One short beep before boot
B5	Display QuietBoot (not used)
BE	Clear screen
B6	Check password (optional)
BC	Clear parity checkers
BA	Not used
B7	ACPI configuration (table configuration in memory and BDA)
BD	Display MultiBoot menu if ESC is hit
BF	Display system configuration summary (if enabled in CMOS)
8F	Get total # of hard drives and put in BDA
91	Program IDE hard drives (timing, PIO modes, etc.)
9F	Save total # of hard drives (SCSI and ATA) in BDA
97	Fixup MP Table (checksum)
99	Check SMART harddrive
C7	Prepare to boot OS, clean up graphics and PMM areas
C0	Try to boot Int19h. Return to video mode 3, disable PMM, return to real mode, disable gate A20, clear system memory, reset stack, invoke Int19h

OR840

Port 80	Port 81	Description
01		Transition to protected mode complete
	30	Seek relevant processor patch
	31	Apply processor patch
	32	Setup ICH GPIO base, SIO PME base, miscellaneous ICH settings
	33	Setup ICH and SIO GPIO attributes and values
	3A	Set ICH GPIO attributes and values
	3B	Set SIO GPIO attributes for GPIO10-17

Port 80	Port 81	Description
	3C	Set SIO GPIO attributes for GPIO20-27
	3D	Set SIO GPIO attributes for GPIO30-37
	3E	Set SIO GPIO attributes for GPIO40-43
	3F	Set SIO GPIO attributes for GPIO50-57
	40	Set SIO GPIO attributes for GPIO60-61
	41	Set initial values for SIO GPIOs
	34	SMBUS initialization
	35	Initialize L1 cache for use as memory during memory initialization
02		Start Minimum memory establishment
	00	Memory not supported (Not RDRAM)
	01	Memory not supported (SPD contains invalid width - not 16 or 18)
	02	No memory devices were found on one or both channels
	03	More than 32 devices on the channel
	04	Memory failure (number of devices detected does not match SPD data)
	05	Memory not supported (FRAS data in SPD is invalid)
	0A	Memory not supported (Populated memory requires too many time domains)
	0B	Memory not supported (No valid channel frequency)
	0C	Memory failure (levelization failure 0 ran out of time domains)
	0D	Memory not supported (unsupported memory technology)
	0E	Memory failure (Continuity module missing or chipset failure)
	0F	Memory not supported (could no find valid refresh rate
	10	Memory not supported (invalid refresh information in SPD
	11	Memory not supported (TCDC invalid)
	12	Memory not supported (does not support enough time domains)
	13	Memory not supported (TRDC invalid)
	14	Memory not supported (invalid SPD TCLS or TCAS)
	15	Memory not supported (SPD mismatch between channel A and B)
	17	Memory not supported (SPD mismatch between channel A and B)
	18	Memory not supported (SPD mismatch between channel A and B)
	19	Memory not supported (SPD mismatch between channel A and B)
	1A	Memory not supported (SPD mismatch between channel A and B)
	1B	Memory not supported (SPD mismatch between channel A and B)
	1C	Memory not supported (SPD mismatch between channel A and B)
	1F	Memory not supported (SPD mismatch between channel A and B)
	20	Memory not supported (invalid number of devices on RIMM)
	22	Memory not supported (SPD mismatch between channel A and B)
	24	Memory failure (Detected bad chipset configuration)
	25	Memory not supported (RIMMs must support ECC)
	28	Memory not supported (unsupported memory technology)
	29	Memory not supported (unsupported memory technology)
	2A	Memory not supported (could not find valid CAS latency)
	2C	Memory not supported (can not mix registered and non-registered memory
	2D	Memory not supported (could not find valid CAS latency)
	2E	Memory failure (levelization phase 1)
	2F	Memory failure (levelization phase 2)
	38	Start memory discovery
	40	STEP: Setup for memory discovery
	41	STEP: SIO reset
	42	STEP: Serial enumeration

Port 80	Port 81	Description
	44	RDRAM STEP: Detect RIMM presence using SPD
	45	RDRAM STEP: DRCG bypass mode - SIO RESET/SET RESET/CLEAR RESET
	46	RDRAM STEP: Determine RAMBUS frequency and set DRCG
	47	RDRAM STEP: MCH RAC initialization
	48	RDRAM STEP: Page Policy and power management (initialization mode)
	4A	RDRAM STEP: Test 77
	4B	RDRAM STEP: Serial enumeration
	4C	RDRAM STEP: Group enumeration
	4D	RDRAM STEP: Program timing parameters
	4E	RDRAM STEP: Power down exit
	4F	RDRAM STEP: Fast clock initialization
	50	RDRAM STEP: RDRAM core initialization
	51	RDRAM STEP: Levelization
	52	RDRAM STEP: Program power down configuration options
	53	RDRAM STEP: Begin normal operations - Page Policy and power management and IC bit
	54	RDRAM STEP: memory initialization complete - reenter MIT
	55	RDRAM STEP: Program power down configuration options
	56	RDRAM STEP: Begin normal ops - Page Policy, power management and set IC bit
	57	RDRAM STEP: Memory initialization complete - reenter MIT
03		Found quick start VM0 FMM object
	30	Seek relevant processor patch
	31	Apply processor patch
	32	Setup ICH GPIO base, SIO PME base, miscellaneous ICH settings
	33	Setup ICH GPIO attributes and values
	3A	Set ICH GPIO attributes and values
	3B	Set SIO GPIO attributes for GPIO10-17
	3C	Set SIO GPIO attributes for GPIO20-27
	3D	Set SIO GPIO attributes for GPIO30-37
	3E	Set SIO GPIO attributes for GPIO40-43
	3F	Set SIO GPIO attributes for GPIO50-57
	40	Set SIO GPIO attributes for GPIO60-61
	41	Set initial values for SIO GPIOs
	34	SMBUS initialization
	35	Initialize L1 cache for use as memory during memory initialization
04		Memory established
	60	Prepare for ECC scrubbing
	61	ECC scrubbing
	62	Restore context after ECC scrubbing
	63	Exit SMBUS
	64	ICH settings
	66	PIC
	68	SIO - Parallel port
	69	SIO - Serial port #1
	6A	SIO - Serial port #2 / DMA
	6B	SIO - Game port
	6C	SIO - MIDI
	6D	SIO - Exit configuration
	6E	Local APIC, FWH decode, runtime enable
05		Enable paging

Port 80	Port 81	Description
06		Unpack VM2
07		Transition to VM2
30		Interposer entry
	01	Interposer - Phase 0 - CMOS initialization
	02	Interposer - Phase 0 - IVT initialization
	03	Interposer - Phase 0 - Initialize compatibility table
	04	Interposer - Phase 0 - Runtime mouse fixes
	11	Interposer - Phase 1 - Reserved
	12	Interposer - Phase 1 - SCSI device numbering
	21	Interposer - Phase 2 - E820 data transfer
	22	Interposer - Phase 2 - Data repackaging
	23	Interposer - Phase 2 - Int 13h post
	24	Interposer - Phase 2 - Reserved
3A	31	Interposer - VM3 entry - Reserved
	32	Interposer - VM3 entry - Entering Int 19h
3F		DCC
40	XY	Scan for devices on PCI BUS where XY : X = Device(bits 7 - 3), Y = Bus(bits 2 - 0)
41		Route PCI IRQs to devices
42		Allocate memory resources to PCI devices
43		Allocate IO resources to PCI devices
44		Detect and shadow PCI Option ROM - Add in card
45		Detect and shadow PCI Option ROM - Embedded
4F		PCI Enumeration complete
50		SEL_FLASH_GetMaxRecSize had a severe parity error reading flash
70		Enumerate primary IDE channel
	02	Test for empty primary IDE channel
	03	Primary IDE channel discovery complete
	04	Primary IDE channel - check for ATAPI signature - master
	05	Primary IDE channel - master found - look for slave
	16	Primary IDE channel - ATAPI slave found
	26	Primary IDE channel - slave found
	36	Primary IDE channel - slave found
71		Enumerate secondary IDE channel
	82	Test for empty secondary IDE channel
	83	Secondary IDE channel discovery complete
	84	Secondary IDE channel - check for ATAPI signature - master
	85	Secondary IDE channel - master found - look for slave
	96	Secondary IDE channel - ATAPI slave found
	A6	Secondary IDE channel - slave found
	C6	Secondary IDE channel - slave found
72		Program IDE chipset settings
73		Program IDE devices
74		Setup IDE runtime data
75		Hard disk spin-up delay and Drive diagnostics
90		SIO initialization - Flex card detection
91		SIO initialization - serial
92		SIO initialization - parallel
93		SIO initialization - keyboard controller
94		SIO initialization - FDC

Port 80	Port 81	Description
95		Initialization - audio
A0		SMI handler - ACPI mode enable
A1		SMI handler - ACPI mode disable
A2		ACPI_LOAD_FACS
A3		ACPI_LOAD_FACD
A4		ACPI_LOAD_RSDT
A5		ACPI_LOAD_DSDT
A6		ACPI_LOAD_SSDT
A7		ACPI_LOAD_APIC
A8		TBD
A9		TBD
AA		TBD
AB		TBD
AC		ACPI_NO_CATALOG
AD		TBD
AE		ACPI_E820
AF		ACPI_FIXUPS
B8	00	AGP Pro Detected, prevent system from booting
B9	00	FMM initialization failed - Flash corruption - BIOS crisis recovery required
DE	AD	Double BIT ECC error detected (forced hang) - This code may be only temporary
F0		Enter BIOS recovery mode
F2		BIOS recovery - initialize flash
F4		BIOS recovery - enumerate PCI buses
F6		BIOS recovery - initialize floppy controller
F8		BIOS recovery - Extract BIOS update file from floppy
F9		BIOS recovery - Validate BIOS update contents
FA		BIOS recovery - Erase FWH blocks
FB		BIOS recovery - Enable FWH security
FC		BIOS recovery - Write buffer to FWH
FE		BIOS recovery - Operation successful
FF	F1	BIOS recovery - Flash initialization failure
	F2	BIOS recovery - Flash update operation failed
	F3	BIOS recovery - Read file from floppy operation failed
	F4	BIOS recovery - Flash erase operation failed
	F5	BIOS recovery - Flash write operation failed
	F6	BIOS recovery - File verify operation (checksum) failed
	F7	BIOS recovery/flash update - processor patch installation failed
	F8	BIOS recovery - File verify operation (invalid BIOS) failed
	F9	BIOS recovery - File verify operation (mismatched platform BIOS) failed
	FA	BIOS recovery - Boot block incompatible with BIOS
	FB	BIOS recovery - Flash verify after write failed

LANDMARK

Same as BIOSYS BIOS. Beeps as for IBM AT. Codes sent to ports 280 and 80.

XT Jumpstart

Code	Meaning	Code	Meaning
01	Jump to reset area in ROM BIOS	16	Setup and init cassette function
02	Initialize DMA page register	17	Setup and init bootstrap function
03	Initialize DMA refresh register	18	Setup and init keyboard function
04	Clear all RAM	19	Enable speaker
05	Perform RAM test on 1st 64k	1A	Setup timer 0 for the real time clock
06	Clear 1st 64k	1B	Enable RTC
07	Initialize BIOS stack to 0:FC0	1C	Setup timer 2 for the beeper
08	Set the equipment flag based on switches	1D	Size memory: write 55AA/AA55 to 1st/last word in segment
09	Initialize default interrupt vectors	1E	Read 1st and last word of segment
0A	Initialize 8255 if it exists and enable parity	1F	Compare 1st and last words
0B	Initialize 8259 and enable interrupts	20	Report determined memory size to screen
0C	Setup adapters and peripherals	21	Perform checksum on ROM BIOS
0D	Setup video	22	If cold boot perform complete RAM testing
0E	Initialize video	23	Move system stack to bottom of memory and save pointer at 40:0E
0F	Initialize equipment	24	Reset parity after RAM sizing
10	Initialize memory configuration in RAM (currently = 64K)	25	Enable timer and keyboard interrupts
11	Setup timer function	26	Setup the serial and parallel ports
12	Initialize timer function	27	Setup the game port
13	Setup time of day function	28	Setup the floppy disk controller
14	Initialize time of day function	29	Scan for optional ROM in 2K chunks from C8000 to start of BIOS
15	Setup and init print screen function	2A	Boot System

AT Jumpstart

Code	Meaning	Code	Meaning
03	1 short beep when first awake	32	Size memory by testing it
04	Initialize bell tone	33	Adjust memory configuration
05	Enable CMOS RAM	33	Verify CMOS RAM size
06	Reset video controller	34	Enable I/O parity
07	Disable I/O parity	35	Test 8259
08	Start memory refresh	36	Bytes swap test
09	Clear reset flag in RAM	37	Test NMI
0A	Test DMA page registers	38	Timer test
10	Use CMOS to determine if soft reset	39	Initialize timer A
11	Perform ROM checksum	3A	Protected mode memory test
12	Test timer A	3B	Test keyboard
13	Test DMA channel A	3C	Test keyboard interrupt
14	Test DMA channel B	3D	Enable A20
15	Test refresh	3E	Reset hard disk controller

Code	Meaning	Code	Meaning
16	Flush 8042 input buffer	3F	Setup floppy controller
17	Reset 8042	40	Test floppies
18	Get keyboard switch	41	Setup keyboard (NumLock)
19	Initialise keyboard	42	Enable timer interrupt
1A	Clear any existing parity	43	Check for dual floppy/hard disk controller
1B	Enable on-board parity	44	Find floppy drive A type
1C	Test base 64K memory	45	Find floppy drive B type
1D	Test base 64k parity	46	Reset hard disk
1E	Initialize POST stack	47	Enable slave DMA
20	Put keyboard # in RAM	63	Set video interrupt vector
65	Set video speed	48	Call any external ROMs
21	Test protected mode registers	49	Initialize printer
22	Initialize 8259 interrupts	4A	Initialize serial
23	Zero all 256 interrupts	4B	Initialize 80287
24	Initialize interrupts 0-1fh	4C	Read CMOS RAM status
25	Perform DRAM checksum	4D	Check CMOS configuration against hardware found
26	Adjust configuration based on hardware	70	Check CMOS against memory found
27	Check manu switch (may exit POST)	4E	Initialize timer ticks
28	Initialize video controller	4F	Enable IRQ9
2A	Test video memory	50	Enable on-board parity
2B	Test video sync	51	Call add-on card ROM
2C	Look for external video	52	Enable keyboard interrupt
2D	Change video configuration if external video	53	Reset printer
2E	Unused	60	Check for any errors
2F	Initialize video controller	61	One short beep
30	Change video interrupt	62	Print sign-on message
31	Print any POST messages	64	Perform boot

MAGNAVOX

See Philips.

MICRONICS

Makes its own upgrades for Phoenix.

MR BIOS

The last code emitted is the one that failed. There may also be a message on screen. Beep codes are in a binary format and are preceded by a high and low tone (described elsewhere). Check also Nasty Noises for more codes.

POST Procedures

Procedure	Meaning
Reset	See if a warm boot (Ctrl+Alt+Del) or a cold boot (Reset) is needed.
Chipset Initialisation	Reset the support chips (8259) DMAs and timers to defaults before proceeding.
Disable Chips	Disable NMI/DMA and Video (6845) for accurate results later. Failure here normally a NMI generated by one of the disabled chips.
ROM BIOS Checksum	Perform checksum test, add a preset value stored in BIOS to create value of 00.
DMA Test	Perform a test of the page registers in the DMA controller.
Keyboard Controller Test	Send a command to the 8042 keyboard controller to perform a selftest. The keyboard controller will return a buffer and error buffer address.
Chipset Initialisation	Initialise the DMA (8237)/PIC (8259)/PIT (8254) and RTC chips.
DMA Test	Test the registers of the master 16-bit and slave 8-bit DMA controllers by writing bit patterns and reading the results.
Cache/Shadow Disable	Disable cache and shadow RAM before processing with POST.
Refresh	Test interval in which PIT (8254) chip sends a refresh signal to the DMA chips.
Base 64K Memory	Test the first 64K of system memory with a walking-bit pattern.
PIC Test	Test the mask registers of the master and slave interrupt controllers by setting the mask-bit in the registers and generating an interrupt to see if the interrupt is trapped. Then test the additional registers in the PICs with a walking-bit pattern.
PIT Test	Test interrupt timer channels 0-2 and initialise if no failures occur.
RTC	Perform read/write test of RTC portion of CMOS and initialise if no failures occur.
Video	Test and initialise the video adapter, which will perform an internal diagnostic and sign on before returning an OK status.
CMOS Checksum	Perform a checksum on the system RAM.
Keybd Initialisation	Initialise the keyboard and read the buffer address for errors.

OEM Specific

Procedure	Meaning
Base Memory Test	Test memory addresses between 64-640K with a walking-bit pattern. There may be a hex display of the failing it.
Keyboard 2nd Init	Tries again if the first failed.
Protected Mode Test	Test the ability of the keyboard controller address line 20 to respond to commands that switch the CPU in and out of protected mode.
Extended Memory	Test addresses above 1 Mb in 64K blocks and perform pattern tests.
OEM Memory	Normally test the cache controller and shadow RAM.
RTC Time Test	Test the write active line of the RTC/CMOS chip. Check bad CMOS/battery
Serial Port	Generate an interrupt of the CPU through I/O ports reserved for RS232 devices. Failure to see a device could be the device itself or more than one set to the same port. Checks are only made for two devices.
Parallel	Check for parallel devices. Failure to see a device could be the device itself or more than one using the same port. Checks are only made for three.
NPU Test	Perform a register test on the NPU then initialise if passed.
Floppy Test	Test floppy controller and drive.

Procedure	Meaning
Fixed Disk	Test fixed disk controller and drive and compare the results against CMOS. This is skipped if no drive is installed.
CMOS Update	Update information in CMOS RAM based on the previous results.

Non-Fatal Errors

Procedure	Meaning
Lock Check	Check if a system lock-byte is set and wait for user response if an error is generated. Check the panel lock or circuitry.
NumLock/Pwd/Setup	Set NumLock on (if set) and ask for password (if set) and display setup message.
Typematic Rate	Set the typematic rate.
Floppy Disk	Perform any further initialisation needed.
Hard Disk	Perform any further initialisation needed.
Video Mode	Set primary video mode and display any errors found during initialisation routines.
Shadow/Cache Enable	
Adapter ROM	Initialise adapters with a ROM signature of 55AA. Self tests will be performed by the equipment concerned before handing back control to the POST.
Video Monitor Mode	Set the video mode based on the information in the CMOS and update the time variables from the RTC.
Parity/NMI Enable	Enable NMI by setting bit 7 of CMOS address 41 and enable parity.
Set Stack	Set last significant byte of stack pointer and install shadow RAM at E000 if in CMOS.
Acknowledge	Acknowledge errors and set primary video mode before calling Int 19 boot loader. Errors reported will await a keyboard response before proceeding. Errors beyond this point are normally software related.

3.3

Code	Meaning
00	Cold-Boot commences (Not seen with warm-boot).
01	HOOK 00 OEM specific typically resets chipset to default
02	Disable critical I/O: 6845s CRT; 8237s DMA; 7675 floppy and parity latches
03	BIOS checksum test
04	DMA Page register test (Ports 81-8F)
05	8042 (Keyboard Controller) Self test.
06	Game Port init: 8237 master/slave; 8254 ch2/1; RTC Reg3 F/A; 8259 master/slave
07	HOOK 01. OEM specific; typically disables cache/shadow
08	Refresh toggle test (PORTB)
09	Pattern test master/slave 8237s; eight 16-bit regs each
0A	Base 64K memory test
0B	Pattern test master/slave 8259 mask regs
0C	8259/IRQ tests purge powerup ints
0D	8254 channel-0 test and initialization
0E	8254 channel-2 toggle test speaker circuitry
0F	RTC tests/inits: Init REG-B; write/readback NVRAM. PIE test
10	Video Initialization.
11	CMOS Checksum test
12	Sign-on msg. Accept KB BAT; perform 1st try KB unit; cold boot delay
13	HOOK 02. OEM specific; select 8MHz bus
14	Size/Test base memory (low 64K already done)

Code	Meaning
15	Perform 2nd try KB init if necessary
16	HOOK 03. OEM specific. Size/Test cache
17	Test A20 gate off; then on.
18	Size/Test extended memory
19	HOOK 04 and Size/Test system memory (special OEM memory)
1A	Test RTC Update-In-Progress and validate time
1B	Serial port determination off-board/on-board
1C	Parallel port determination off-board/on-board
1D	Copro determination/initialization
1E	Floppy controller test/determination CMOS validation
1F	Fixed Disk controller test/determination CMOS validation
20	Rigorous CMOS parameter validation, display other config changes
21	Front-Panel lock check; wait for user to acknowledge errors
22	Set NumLock; Password-Security Trap; despatch to setup utility
23	HOOK 05. OEM specific.
24	Set typematic rate. 28 HOOK 6. OEM specific, typically enables shadow, cache, turbo
25	Floppy subsystem initialization
26	Fixed subsystem initialization
27	ACK errors; set primary adapter video mode
29	Disable A20-gate; set low stack, install C800, E000 ROMs.
2A	ACK errors; set video mode, set DOS time variables from RTC.
2B	Enable parity checking and NMI
2C	Set low stack, Install E000 ROM
2D	ACK errors, set primary video mode.
2E	HOOK 07. OEM specific. Log-in EMS (if built-in).
2F	Pass control to INT 19.

3.4

Code	Meaning
00	Cold Start. Output EDX register to I/O ports 85h, 86h, 8Dh, 8Eh for later use
01	Init Custom KBD controller, disable CPU cache, cold init onboard I/O chipset, size & test RAM & cache
02	Disable critical I/O: 6845s CRT; 8237s DMA; 7675 floppy and parity latches (monitor, DMA, FDC, I/O ports, Speaker, NMI).
03	BIOS checksum test
04	DMA Page register test (Ports 81-8F)
05	8042 (Keyboard Controller) Self test. Enable A20 Gate.
06	Init ISA I/O
07	Warm initialize custom KBD controller, warm initialize onboard I/O chipset.
08	Refresh toggle test (PORTB)
09	Pattern test master/slave 8237s
0A	Base 64K memory test. Test Master 8259 mask, test Slave 8259 mask
0B	Pattern test master/slave 8259 mask
0C	Test 8259 Slave, test 8259 slave's interrupt range, initialize interrupt vectors 00-77h, init KBD buffer variables.
0D	8254 channel-0 test
0E	8254 channel-2
0F	RTC test, CMOS RAM r/w test
10	Turn on monitor, show possible error messages.

Code	Meaning
11	CMOS Checksum test
12	Call video ROM init routine. Sign-on msg.
13	Set 8MHz AT bus
14	Size/Test base memory, Stuck NMI
15	No KB and power on: Perform 2nd try KB init if necessary
16	Size/Test cache
17	Test A20 gate off; then on.
18	Size/Test external memory, Stuck NMI
19	Size/Test system memory, Stuck NMI
1A	Test RTC time
1B	Serial port determination off-board/on-board
1C	Parallel port determination off-board/on-board
1D	Copro initialization
1E	Floppy controller determination
1F	IDE determination
20	Display CMOS config changes
21	Clear Screens
22	Set NumLock LED; perform security functions
23	Final determination of onboard Serial/Parallel ports.
24	Set typematic rate
25	Floppy subsystem initialization
26	ATA disks initialization
27	Set primary adapter video mode
28	WB-CPU support, Green PC: purge 8259 slave, relieve trapped IRRs before enabling PwrMgmt, set 8042 pins, Ctrl-Alt-Del possible, Enable CPU Features.
29	Disable A20-gate; install C800, E000 ROMs.
2A	Clear primary screen, convert RTC to system ticks, set final DOS timer variables.
2B	Enable NMI and latch
2C	Reserved
2D	Reserved.
2E	Fast A20: Fix A20.
2F	Purge 8259 slave; relieve any trapped IRRs before enabling Green-PC. Pass control to INT 19.
32	Test CPU Burst
33	Reserved
34	Determine 8042, Set 8042 Warm-Boot flag STS.2
35	Test HMA Wrap, Verify A20 enabled via F000:10 HMA
36	Reserved
37	Validate CPU: CPU Step NZ, CPUID Check. Disable CPU features
38	Set 8042 pins (Hi-Speed, Cache-off)
39	PCI Bus: Load PCI; Processor Vector init'd, BIOS Vector init'd, OEM Vector init'd
3A	Scan PCI Bus
3B	Initialize PCI Bus with intermediate defaults
3C	Initialize PCI OEM with intermediate defaults, OEM bridge
3D	PCI Bus or PLUGnPLAY: Initialize AT Slotmap from AT-Bus CDE usage
3E	Find phantom CDE ROM PCI-cards
3F	PCI Bus: final Fast-Back-to-Back state
40	OEM POST Initialization, Hook Audio
41	Allocate I/O on PCI-Bus, logs-in PCI-IDE
42	Hook PCI-ATA chips

Code	Meaning
43	Allocate IRQs on the PCI Bus
44	Allocate/enable PCI Memory/ROM space
45	Determine PS/2 Mouse
46	Map IRQs to PCI Bus per user CMOS, Enable ATA IRQs.
47	PCI-ROM install, note user CMOS
48	IfSetup conditions: execute setup utility
49	Test F000 Shadow integrity, Transfer EPROM to Shadow-RAM
4A	Hook VL ATA Chip
4B	Identify and spin-up all drives
4C	Detect Sec IRQ, if VL/AT-Bus IDE exists but its IRQ not known yet, then autodetect it
4D	Detect/log 32-bit I/O ATA devices
4E	ATAPI drive M/S bitmap to Shadow-RAM, Set INT13 Vector
4F	Finalize Shadow-RAM variables
50	Chain INT 13
51	Load PnP, Processor Vector init'd, BIOS Vector init'd, OEM Vector init'd
52	Scan PLUGnPLAY, update PnP Device Count
53	Supplement IRQ usage-AT IRQs
54	Conditionally assign everything PnP wants
58	Perform OEM Custom boot sequence just prior to INT 19 boot
59	Return from OEM custom boot sequence. Pass control to 1NT 19 boot
5A	Display MR BIOS logo
88	Dead motherboard and/or CPU and/or BIOS ROM.
FF	BIOS POST Finished.

Msg	Low-High	Problem
03	LH-LLL	ROM-BIOS Checksum Failure
04	LH-HLL	DMA Page Register Failure
05	LH-LHL	Keyboard Controller Selftest Failure
08	LH-HHL	Memory Refresh Circuitry Failure
09	LH-LLH	Master (16 bit) DMA Controller Failure
09	LH-HLH	Slave (8 bit) DMA Controller Failure
0A	LH-LLLL	Base 64K Pattern Test Failure
0A	LH-HLLL	Base 64K Parity Circuitry Failure
0A	LH-LHLL	Base 64K Parity Error
0A	LH-HHLL	Base 64K Data Bus Failure
0A	LH-LLHL	Base 64K Address Bus Failure
0A	LH-HLHL	Base 64K Block Access Read Failure
0A	LH-LHHL	Base 64K Block Access Read/Write Failure
0B	LH-HHHL	Master 8259 (Port 21) Failure
0B	LH-LLLH	Slave 8259 (Port A1) Failure
0C	LH-HLLH	Master 8259 (Port 20) Interrupt Address Error
0C	LH-LHLH	Slave 8259 (Port A0) Interrupt Address Error
0C	LH-HHLH	8259 (Port 20/A0) Interrupt Address Error
0C	LH-LLHH	Master 8259 (Port 20) Stuck Interrupt Error
0C	LH-HLHH	Slave 8259 (Port A0) Stuck Interrupt Error
0C	LH-LHHH	System Timer 8254 CH0 / IRQ0 Interrupt Failure
0D	LH-HHHH	8254 Channel 0 (System Timer) Failure
0E	LH-LLLLH	8254 Channel 2 (Speaker) Failure

Msg	Low-High	Problem
0E	LH-HLLLH	8254 OUT2 (Speaker Detect) Failure
0F	LH-LHLLH	CMOS RAM Read/Write Test Failure
0F	LH-HHLLH	RTC Periodic Interrupt / IRQ8 Failure
10	LH-LLHLH	Video ROM Checksum Failure at Address XXXX Mono Card Memory Error at Address XXXX Mono Card Memory Address Line Error at XXXX CGA Card Memory Error at Address XXXX CGA Card Address Line Error at Address XXXX
11	(None)	Real Time Clock (RTC) Battery is Discharged
11	(None)	Battery Backed Memory (CMOS) is Corrupt
12	LH-HLHLH	Keyboard Controller Failure
14/18/19	LH-LHHLH	Memory Parity Error
14/18/19	LH-HHHLH	I/O Channel Error
14		
18		
19	(None)	RAM Pattern Test Failed at XXXX Parity Circuit Failure in Bank XXXX Data Bus Test Failed: Address XXXX Address Line Test Failed at XXXX Block Access Read Failure at Address XXXX Block Access Read/Write Failure: Address XXXX Banks Decode to Same Location: XXXX and YYYY
15	(None)	Keyboard Error-Stuck KeyKeyboard Failure or no Keyboard Present
17	LH-LLLHH	A20 Test Failure Due to 8042 Timeout
17	LH-HLLHH	A20 Gate Stuck in Disabled State (A20=0)
17	(None)	A20 Gate Stuck in Asserted State (A20 Follows CPU)
1A	LH-LHLHH	Real Time Clock (RTC) is Not Updating
1A	(None)	Real Time Clock (RTC) Settings are Invalid
1E	(None)	Diskette CMOS Configuration is Invalid Diskette Controller Failure Diskette Drive A: Failure Diskette Drive B: Failure
1F	(None)	Fixed Disk CMOS Configuration is Invalid Fixed Disk C: (80) Failure Fixed Disk D: (81) Failure Please Wait for Fixed Disk to Spin Up
20	(None)	Fixed Disk Configuration Change Diskette Configuration Change Serial Port Configuration Change Parallel Port Configuration Change Video Configuration Change Memory Configuration Change Numeric Coprocessor Configuration Change
21	(None)	System Key in Locked Position-Turn Key to Unlocked Posn
29	(None)	Adapter ROM Checksum Failure at Address XXXX

MYLEX/EUROSOFT

Derived from Eurosoft BIOS, mainly for Mylex EISA boards.

4.71

Pass	Fail	Meaning
03	04	DMA page registers test
05	06	Keyboard reply test
07	08	Keyboard self-test
09	0A	8042 keyboard controller able to read links
0B		RATMOD/DIAG link
0C	0D	Keyboard acceptance of 60H
0E	0F	Keyboard acceptance of parameter
10	11	Read keyboard command byte
12	13	Keyboard command byte came back
14	15	RAM refresh toggle test
16	17	RAM bit test
18	19	RAM parity test
1A	1B	CMOS RAM test
1C	1D	CMOS RAM battery test
1E	1F	CMOS RAM checksum test
	20	CMOS RAM battery fault bit set
21	22	Master DMA controller test
21	23	Slave DMA controller 2 test
24		Protected mode entered safely
25		RAM test completed
26	27	BIOS ROM checksum test
28		Protected mode exit
29	2A	Keyboard power-up reply received test
2B	2C	Keyboard disable command acceptance test
	2D	Video display presence check
	2E	POST Errors were reported
	2F	About to halt
30		Protected mode entered safely (2)
31		RAM test complete
33		Master interrupt controller test
34	35	Slave interrupt controller test
36	37	Chipset initialization
38	39	System BIOS shadowed
3A	3B	Video BIOS shadowed

EISA/ISA

Code	Meaning	Code	
01	Processor test	16	Initialise output port of keyboard controller
02	DMA Page Register	17	Keyboard interrupt test
03	8042 keyboard controller	18	Initialise keyboard
04	BIOS ROM Checksum error	19	RTC clock test failure

Code	Meaning	Code	
05	Send keyboard command test bad	1A	Maths copro test failure
06	CMOS RAM Test	1B	Reset hard/floppy controller
07	RAM Refresh Test	1C	Initialise floppy drive
08	1st 64K memory test	1D	Initialise hard drive
09	8237 DMA controller test	1E	Initialise ROMs in C000-DFFF
0A	Initialise DMA controller	1F	Initialise serial and parallel ports
0B	Interrupt Test	20	Initialise time of day in RTC
0C	Determine RAM size	21	Initialise ROMs in E000-EFFF
0D	Initialise video	22	Look for boot device
0E	EGA/VGA ROM checksum test failed	23	Boot from floppy disk
10	Search for monochrome card	24	Boot from hard disk
11	Search for colour card	25	Gate A20 enable/disable failure
12	Word splitter and byte shifter test failed	26	Parity error occurred
13	Keyboard Test	30	DDNIL bit scan failure
14	RAM Test failed	FF	Fatal error occurred and system halted
15	Timer test error		

NCR

Purchased 1991 by AT&T. See AMI pre-0490 for PC386, others below. NCR ones use LPT1.

Architecture	Typical PC	BIOS	POST Code Port
XT	PC6	NCR	378 or 3BC (LPT 1)
AT (ISA)	3728, 3204, PC 916	NCR	80 and 378 or 3BC (LPT 1)
	PC386	AMI Pre-0490	80
Micro Channel	3421	Phoenix	680 and 3BC

PC6

Code	Meanings
AA	8088 CPU failure
B1	2764 EPROM checksum failure
B2	8237 DMA controller failure
B3	8253 timer failure
B4	RAM failure. Halts if error in first 64K, otherwise displays MEMORY ERROR.
B5	8259 interrupt controller failure. Displays INTERRUPT FAILURE
B6	RAM parity error. Displays ERROR IN BASE MEMORY or ERROR ON EXPANSION CARD.
BB	All tests passed

3302/3304/3728/PC916SX

Code	Meaning
01	Test CPU registers
02	Test system I/0 port-write and read port 61 to confirm will handle RAM refresh.
03	Test ROM BIOS checksum
04	Test DMA page registers

Code	Meaning
05	Test timer channel 1 (refresh)
06	Test timer channel 2 (speaker)
07	Test RAM refresh logic. Also verifies timer is working.
08	Test base 64K RAM
09	Test 8/16 bit bus conversion
0A	Test interrupt controller 1
0B	Test interrupt controller 2
0C	Test I/O controller
0D	Test CMOS RAM read/write
0E	Test for battery power low or interrupted since last test
0F	Test CMOS RAM checksum
10	Test CPU protected mode
11	Test video configuration in CMOS RAM or display switch
12	Test primary video controller
13	Test secondary video controller
20	Display results of tests to this point
21	Test DMA controller 1
22	Test DMA controller 2
23	Test Timer channel 0 (system timer tick)
24	Initialize interrupt controllers
25	Test interrupts
26	Test interrupts
30	Check base 640K memory size
31	Check extended memory size
32	Test higher 8 address lines
33	Test base memory
34	Test extended memory
40	Test keyboard-enable/disable
41	Test keyboard-reset
42	Test keyboard-clock low
43	Test keyboard-for interrupt, enable keyboard, init pointers, write out subcommand
44	Test 8086 address overrun compatibility (gate A20)
50	Set up hardware interrupt vectors
51	Enable interrupt from timer channel 0
52	Security ROM
60	Test floppy disk controller and drive
61	Test hard disk controller
62	Initialize floppy drives
63	Initialize hard drives
70	Test real time clock
71	Set time of day in real time clock
72	Check parallel interfaces
73	Check serial interfaces
74	Check for and execute adapter option ROMs
75	Check if math coprocessor is installed and enable interrupt
76	Enable keyboard and real time clock interrupts
F0	System not configured correctly, or hardware defect
F1	Scan for and execute motherboard option ROMs
F2	INT 19 to boot operating system-No POST errors.

PC916 5/6

*halt on error if loop jumper installed in keyboard connector

Code	Meaning
01	Test CPU registers, reset video cards, display diagnostic messages
02	Verify port 61, disable non-maskable interrupt, start speaker timer channel 2
03	Test ROM BIOS checksum
04	Test DMA page registers
05	Test timer channel 1 (refresh)
06	Test timer channel 2 (speaker)
07	Test refresh logic by reading port 61 bit 4 every 15 microseconds
08	Test base 64K RAM
09	Test 8/16-bit bus converting logic, initialize both interrupt controllers
0A	Test interrupt mask register A
0B	Test interrupt mask register B, write temporary interrupt vector table for INT 00-77
0C	Test 8042/8742 keyboard controller
0D	Test CMOS RAM shutdown byte
0E*	Test CMOS RAM battery power low or interrupted since last test
0F*	Test CMOS RAM checksum; initialize periodic rate
10	Test CPU protected mode
11	Test video configuration in CMOS RAM or display switch, look for advanced video card ROM in segment C000, initialize interrupt vectors.
12	Initialize and test primary video controller
13	Primary video error, test secondary video controller
14	Test disabling Speed stretch enable/disable port 69 bit 0=1
15	Start refresh timer 1 counter 1, disable speed switch timer 2, counter 2
16	Enable then disable speed stretch enable/disable port 69 bit 0
17	Clear write protect bit
18	Write/verify global/local/interrupt descriptor table registers; copy ROM BIOS to shadow RAM F000
19	Verify RAM to ROM BIOS copy OK; reinitialize restart vector, check and execute for burn-in ROM D000. Disable real time clock in CMOS status reg B, reset and initialize video cards.
IA	Command 8042 to execute self-test and verify result
1B	Test 64K Shadow RAM in segment F000
20	Display results of tests to this point
21	Test DMA controller 1
22	Test DMA controller 2 and initialize all 8 channels
23	Test timer 1 counter 0 840 ns clock timer for IRQ0 (INT8)
24	Initialize both interrupt controllers
25	Check for unexpected (hot) interrupts
26	Wait for interrupt
27*	Test timer 2 counter 0 for NMI (INT02), failsafe
28*	Test timer 2 counter 1 (INT72-74)
30	Check base 640K memory size
31	Check extended memory size (max 256M RAM on 5.2, 6 BIOS)
32	Test higher 8 address lines for mirror addresses (5.x BIOS)
33*	Test base memory
34*	Test extended memory (up to 256M)
35*	Test RAM in E000 (v6 BIOS-also test keyboard shutdown command FE-shutdown path 0B)
40	Test keyboard-enable/disable
41	Test keyboard-reset command FF (halt on error if loop jumper not installed)

Code	Meaning
42	Test keyboard-clock low (halt on err if loop jumper not installed)
43	Test keyboard-check for interrupt, enable keyboard, initialize buffer pointers, verify keyboard unlocked, disable external interrupts mask A=F, turn on write protect for RAM E000-FFFF, write out subcommand (halt on error if loop jumper not installed).
44	Test address overrun compatibility (turn off gate A20, 8042 P2 bit 1 = 0)
45	v6 BIOS-Init mouse, en IRQ1 (INT09)keyboard (15 IRQs, 1 disabled), disp Press F1 for Setup.
50	Set up hardware interrupt vectors 0-15, 70-77
51	Enable IRQ0 interval interrupt 08 from timer channel 0; enable ext interrupts (STI)
60	Test for floppy/hard disk controller and drive
61	Test cylinder register for disk controller
62	Initialize floppy drives
63	Initialize hard drives
70*	Test real time clock
71	Set interval timer RAM counts
72	Configure and test parallel interfaces
73	Configure and test serial interfaces
74	Check for and execute adapter option ROMs C8000-DFFFF
75*	Test math coprocessor if installed, and enable interrupt
76	Enable keyboard and real time clock IRQ8 (INT 70) interrupts; enable slave interrupt controller 2 via PIC 1 mask bit 2=0.
F0	Display logged errors. Halt if locked; loop if loop jumper installed
F1	Test system code at segment E000 (v5.x BIOS only); v6 BIOS-copy video ROM BIOS (if present) to shadow RAM if system ROM is absent and switch pack switch 1 is on
F2	INT 19 to boot operating system-No POST errors
F3	Go to setup if F1 key pressed. v6 BIOS: execute floppy diagnostic if Ctrl-D pressed, enable failsafe NMI port 61 bit 2=0, enable parity error port 61 bit 3=0, enable NMI.
F4	v5.x BIOS only-Display speed setting
F4	v6 BIOS-Display speed setting Auto, high, fixed
F5	v5.x BIOS only-initialize counter 2 for speed requested
F6	v5.x BIOS only-Test base memory (long test in 5.2 BIOS)
F6	v6 BIOS only-Test base memory (long test) if F2 pressed
F7	v5.x BIOS only-Test extended memory (long test in 5.2 BIOS)
F7	v6 BIOS only-Long test extended memory if F2 pressed

OLIVETTI

For EISA and PS/2, code is issued after the test has passed, so a stuck code indicates the next test failed. Codes are sent to printer ports 3BC (mono adapter's parallel port), 278, or 378; they will not be printed as no strobe is sent. AT&Ts using the Olivetti motherboard and BIOS (e.g. the AT&T 6300) do the same.

1076/AT&T 6312/WGS 80286

The first checkpoint, 40, resets and initializes a test monitoring device on the parallel port. When an error occurs, the most recent checkpoint code sent to 378 is exclusive-ored with 3F to complement the lower 6 bits, and then sent to 378, so if the refresh test fails (45), the POST card will show 7B because the most recent code sent before the failure was 44. If an error occurs, the POST tries to run through activities that display a message on the monitor, showing tttt Error: xx, where tttt is the name of the failing routine, and xx is a suberror number. If the error is fatal, the display will show *Unrecoverable Power-Up Error*, wait for F1 to be pressed, and return to failing test. If video has failed, there are beep codes.

Pass	Fail	Meaning
40		Dummy check-reset black box
41	7F	80286 CPU flags and register test
42	7E	Check and verify shutdown code-read keyboard status from port 64. if shutdown bit is set, read shutdown byte from CMOS (and clear location there), check for illegal shutdown condition, initialize the 8259s unless shutdown is 9 or A, and jump to the correct routine to handle the shutdown: 0= warm boot (go to next test), 1= return to advanced protected mode test, 2= return to memory test above 1 Mb, 3=return to protected mode test 2, 4=INT19, 5=send EOI to 8259 and return to user routine, 9=int15 block move, A=return to user routine.
43	7D	Checksum test the BIOS ROMs-verify contents add up to 0.
44	7C	Test the 8253 timer-check all 3 timers for not counting, counting too slowly, or counting too fast. Suberror display is the bad timer number 0, 1, or 2.
45	7B	Start memory refresh and verify it occurs every 15.1 microseconds. Init the manufacturing test byte in RAM.
46	7A	Command the 8041 keyboard controller to do a self-test. Suberror display is 1 if error return, 2 if self-test times out.
47	79	Test the first 8K of RAM in 4 passes: 1) write into each word a data value corresponding to the address; 2) invert all bits written; 3) write an odd parity pattern; 4) write zeros. Only pass 4 is done on a warm boot. Beep once when this test passes. Install dummy interrupt vectors, set up the stack and other memory areas. display power-on banner on screen.
48	78	Test 80286 in protected mode 1-pattern test all IDT and GDT registers, verify LIDT, SIDT, LGDT, and SGDT instructions.
49	77	Test CMOS RAM shutdown byte with a pattern, then clear it.
4A	76	Test 80286 in protected mode 2-put CPU into protected mode, check it's there, then return to real mode
4B	75	Test RAM from 8K to 640K (cold boot only)-display progress for each 128K block; write, read, and compare the address and inverted address into each word.
4C	74	Test RAM above 1 Mb - same as below 1 Mb test. Also verify CPU runs properly in prot mode.
4D	73	Test for NMI-installs NMI vector in interrupt table and small service routine. Disables I/O and memory parity errors, then checks for hot NMI.
4E	72	Test for RAM parity-turn NMI parity checking back on, and run a pattern test on the parity checking circuit, monitoring for a parity error.
50	71	Test 8259 interrupt controller 1-pattern test the mask register, install interrupt vectors for IRQs, mask them all off. look for hot interrupt coming through mask, set timer 0 to issue an interrupt, unmask it, count down, and expect the interrupt. Suberror display is l=no in, 2=timer doesn't count, 3=int occurred when masked, 4=bad mask register.
51	6F	Test 8259 interrupt controller 2-same as # 1, but no timer test is done. Suberror display is 5=int occurs wen masked, 6=bad mask register. When the test passes, install the interrupt service routine pointer in the vector table, mask off all interrupts. and display PASS message.
52	6E	Test DMA page register-marching bit test on all page registers.
53	6D	Test 8237 DMA controller 1-pattern test all read/write registers. Initialize each channel into the correct mode for BIOS. Suberror 1 display if failure.
54	6C	Test 8237 DMA controller 2-pattern test all read/write registers. Initialize each channel into the correct mode for BIOS. Suberror 3 display if failure.
55	6B	Test PIC port-write/read pattern test speaker port 61.
56	6A	Test keyboard controller-reset the keyboard and initiate self-test Suberror display is l=bad keyboard self-test completion code. 2=stuck key. 3=no keyboard interrupt Otherwise, display pass message, and set up keyboard id flags and buffer in BIOS RAM area.
57	69	Test CMOS clock/calendar chip-verify accurate time keeping and display pass message.
59	68	Test 80286 advanced protected mode-tests LDT, SDT, LTR, STR, VERR, VERW, LAR, SLR, ARPL instructions; forces exception ints 13 and 5. Suberror display is 3=instruction error, 4=no exception or protection violation. Otherwise display prot mode pass message.

Pass	Fail	Meaning
5A	66	Test CMOS RAM battery and display message if low.
5B	65	Test CMOS RAM non-destructively-copy contents to base memory, write/read pattern test CMOS RAM, restore contents. Suberror 2 if failure.
5C	64	Verify CMOS RAM checksum.
5D	63	Test parallel port by writing AA to 3BC, 278 and 378, and set config info in BIOS RAM.
5E	62	Test serial port configuration-read 3FA and 3FA and assume a UART is present if values not FF. Set up port addresses and timeout values in BIOS RAM area.
5F	61	Test configuration of memory below 640K-compare memory size stored in CMOS RAM with result of earlier test. Display message to run setup if different.
60	60	Test configuration of memory above 1M-compare memory size stored in CMOS RAM with result of earlier test. Display message to run setup if different.
61	5F	Test configuration of 80287 math coprocessor chip -verify math chip same as in CMOS RAM info. Display pass or run setup message.
62	5E	Test configuration of game port at 201 and set equipment bit in BIOS RAM data area.
62	5D	Test keylock switch and wait till unlocked.
63	5D	Test hard drive configuration-initialize controller and drive. Display whether drives are present, and message to run setup if not same as CMOS RAM info.
64	5C	Configure floppy drives A and B-initialize controller and drive. Display whether drives are present, and message to run setup if not same as CMOS RAM info.
66	5B	Test option ROMs-look for signature AA5 each 2K beginning at C8000, run checksum and display error if it occurs. Otherwise pass control to the ROM so it can initialize, and display pass message when done.
		INT 19-boot the system.

M20

Not a true IBM clone, as it had a Zilog Z8001 CPU. Also, a typical POST card will not fit in a slot, so you can only monitor codes from the parallel port. The POST shows a triangle, diamond, or 4 lines on the screen to indicate early POST failure, as shown in the table.

Code	Meaning
	Program video controller using load, output, and jump relative instructions (need video).
Triangle	Test Z8001 CPU registers and instructions; infinite loop if failure.
Triangle	Test RAM module; infinite loop if failure; also send msg to printer: E Mc bb ssss wwww. c = RAM configuration # (3 = 1 32K memory card); bb = hex 16K bank # (0,4,5,6,9,A=motherboard; 1,7,B=expansion board 1; 2=expansion board 2; 3,11,12=cxpansion board 3); ssss = what data should be; wwww = what data was (hx).
4 vertical lines	Test CPU call and trap instructions; infinite loop if failure.
Diamond	Initialize screen and printer drivers.
	Program UARTs (serial chips) and 8253 baud rate generator for keyboard at 1200 baud and RS232 at 9600. Now test remaining circuits and send codes to display and printer.
EC0	8255 parallel interface chip test failed
EC1	6845 CRT controller chip test failed
EC2	1797 floppy disk controller chip test failed
EC3	8253 timer chip test failed
EC4	8251 keyboard serial interface chip test failed
EC5	8251 RS232 serial interface chip test failed
EC6	8259 interrupt controller chip test failed
EK0	Keyboard did not respond
EK1	Keyboard responded, but self-test failed
ED1	Disk drive 1 test failed
ED0	Disk drive 0 test failed

Code	Meaning
E10	Non-vectored interrupt error
E11	Vectored interrupt error

M21/M24 (AT&T 6300)

The M24 went to the US as the AT&T 6300. It had an 8086, so was faster than the PC, albeit difficult to work on. codes are sent to 378 (LPT1). If a fatal error occurs, it performs more initialization of DMA and interrupt controller circuits, tries to display an error message, complements the lower 6 bits of the POST code, sends the result to port 378, and halts the CPU, so numbers will flicker on the POST display with bit 6 on and the lower bits running from 0 upward. The codes start at 40 because a black box was used to monitor POST status at the parallel port. Bit 6 was set true (1) to alert the box the POST was starting.

Code	Meaning
40	CPU flags and register test failed (fatal)
41	BIOS ROM checksum test failed (fatal)
42	Disable pdma controller comd and test 8253 timer channel 1, mode 2, refresh counter (fatal); display sub-error code of 1 if interval is below window, 2 if above, and 3 if timer does not reply.
43	8237 DMA controller test failed (fatal)-master clear the controller, set the mask register, read the control registers, test all 8 read/writeable channel registers. Test registers 0-3 DMA address and count with FFFF then 0000.Set up channel 0 for 64K RAM address refresh. Set up memory-to-I/O transfer, unmask the RAM refresh, and let refresh begin for the first time. Set up the 8253 for proper refresh count. Test for unexpected DMA request (suberror 3), and init DMA channel 1 (not used), 2 (floppy), 3 (display), and init nibble latches. Check for proper DMA transfer into lowest 64K bank of RAM (suberror 4 if parity error).
44	8259 PIC test failed (halt)-init stack to lower 64K RAM area just tested, init and disable 8259A, set up interrupt vectors in RAM, set up software then hardware diagnostic interrupt vectors, test software interrupts, then hardware interrupts. Disable interrupts via 8259 mask register, check for hot interrupts, convert hot mask to IRQ number, save any error code, install interrupt vectors, initialize video, and display error messages (H:#, where # is the hot IRQ#).
45	Install real interrupt vectors, determine system configuration from switches, and initialize video mono and colour. Set video mode 3, clear screen, and display passing error messages for CPU, ROM, DMA, or PIC. Size and clear RAM at every 64K bank past the lowest 64K, displaying the tested RAM as test progresses. Display errors in form cc:y000:zzz:wwww:rrrr, where cc is the config number, y the failing segment, z the offset, w the written data and r the read data. Test MM58174 clock calendar, and display message if fails Test 8253 real time clock count capability, and tone generator. Display errors, halt if failure.
48	Send beep to display and initialize all basic hardware. Init 8041 keyboard controller, determine parallel port configurations and test their registers, determine serial 8250 and Z8530 configurations, check for game card, set up interrupt controller, set all 4 Z8530 serial controllers to 9600 baud, no parity, 1 stop and 8 data. Set up interrupt vectors, initialize RAM variables, clear the screen, initialize the hard disk controller, test for and initialize option ROMs, verify ROM checksums okay, initialize floppy disk controller, allow user to select alternate Z8000 processor if installed and perform INT 19 cold boot.

EISA 2.01

Port 278, 378, Or 3BC (i.e. printer ports)

Code	Meaning	Code	Meaning
01	Test CPU flags, registers. Initialize PIC	1E	Test IDTR and GDTR
02	Test memory refresh	1F	Test CMOS shutdown byte
03	Test CMOS RTC periodic interrupt	20	Test real/protected mode
04	Test gate A20 line	21	Check system memory configuration
05	Test mapping memory SRAM	22	Size memory

Code	Meaning	Code	Meaning
06	Test first 128K RAM. Stack has now been established	23	Test 640K base memory
07	Test for console presence and initialize	24	Verify base memory configuration
08	Verify system BIOS ROM checksum	25	Test extended memory (above 1 Mb)
09	Test 8042 keyboard controller Normal burn-in/manufacturing mode established	26	Verify extended memory configuration
0A	Test timer ratio	27	Check for contiguous extended memory
0B	Test CMOS RAM battery	28	Test cache memory. Extended BIOS data area created and POST errors logged
0C	Verify CMOS RAM checksum	29	Test protected mode instructions
0D	Test for unexpected NMI	2A	Test CMOS RAM
0E	Test interrupt controller #1	2B	Test real time clock
0F	Test interrupt controller #2	2C	Check calendar values
10	Test timer 1 counter 0	2D	Test keyboard/AUX device fuse
11	Test system control port B	2E	Test keyboard
12	Test system control port A	2F	Initialize keyboard typematic rate and delay
13	Verify checksum of NVRAM configuration	30	Test auxiliary device
14	Initialize system board	31	Test 80x87 math coprocessor
15	Initialize adapter	32	Test and initialize Weitek math coprocessor
16	Initialize ESC SCSI adapter	33	Run 1860 CPU basic and advanced diagnostics
17	Initialize system video	34	Test and configure serial ports
18	Test and copy shadow RAM. Video init-display banner and non-fatal errors	35	Test and configure parallel ports
19	Test DMA page registers	36	Detect game port
1A	Test DMA address registers	37	Test and initialize hard drives
1B	Test DMA count registers	38	Test and initialize floppy drives
1C	Test DMA mask registers	39	Scan for and pass control to adapter ROMs
1D	Test DMA stop registers. Initialize DMA controllers	3A	INT 19 boot-load operating system

PS/2 Compatible

Code	Meaning	Code	Meaning
01	Processor test	24	Watchdog timer test
02	Shutdown	25	Test RAM from 64K to 640K
03	Interrupt controller initialisation	26	Configure memory 640K
04	Refresh test	27	Text expansion memory
05	CMOS periodic interrupt test	28	Init ext BIOS data segment, log errors
06	Timer ratio	29	Configure memory above 1 Mb
07	Test first 64k RAM	2A	Dummy checkpoint
08	Test the KBC (8742)	2B	Test RAM parity
09	NMI test	2C	Test DMA page registers
0A	8254 test	2D	Test DMA controller base/current address registers
0B	Port 94h test	2E	Test DMA transfer count register
0C	Port 103h test	2F	Initialize DMA controller
0D	Port 102h test	30	Test PIO 61
0E	Port 96h test	31	Test keyboard
0F	Port 107h test	32	Initialize keyboard typematic rate and delay

Code	Meaning	Code	Meaning
10	Blank the screen	33	Test AUX device
11	KB/Aux device fuse check	34	Test advanced protected mode
12	CMOS battery test	35	Configure parallel ports
13	CMOS RAM checksum test	36	Configure 8250 serial ports
14	Extended CMOS checksum 0-8K	37	Configure coprocessor
15	System board and adapter initialisation	38	Configure game card
16	RAM test and initialisation	39	Configure and initialize hard disk
17	Protected mode register test	3A	Floppy disk configuration
18	CMOS RAM shutdown byte test	3B	Initialize ROM drivers
19	80286 protected mode test	3C	Display total memory and hard drives
1A	Video option ROM scan	3D	Final initialization, Checkpoints complete
1B	EPROM checksum test	3E	Detect and initialize parallel ports
1C	Interrupt controller #1 test	3F	Initialize hard drive and controller
1D	Interrupt controller #2 test	40	Detect and initialize math coprocessor
1E	Interrupt vector initialisation	41	Reserved
1F	CMOS RAM test	42	Initiate adapter ROM scan
20	Extended CMOS r/w test	CC	Unexpected processor exception occurred
21	CMOS clock test	DD	Save DDNIL status
22	Clock calendar test	EE	NMI handler shutdown
23	Dummy checkpoint	FF	INT 19 boot

PACKARD BELL

See Phoenix.

PHILIPS/MAGNAVOX/HEADSTART

Philips, Magnavox, and HeadStart use motherboards designed by Philips Home Electronics in Montreal. Most use a Philips-designed BIOS, although at least one of their portables uses one from Award Software. The beep pattern consists of a series of long and short beeps that correspond to the binary representation of the POST code where leading zeroes are omitted; a zero means a short and a one means a long beep. The various Philips platforms do not all execute the same POST tests.

Philips Platform Cross Reference

Platform	CPU	System Model/Name
Avenger	80286	Magnavox MaxStation 286, Magnum GL; Headstart Series 300
P3212	80286	Magnavox MaxStation 480, Headstart System 380
P 3239	8028680386SX	Magnavox Headstart/Maxstation/Magnum/Professional 1200, 48CD, 1600, 64CD, P160, SR16CD
P 3349	80386SX-20	Magnavox Headstart/Maxstation/Magnum/Professional SX20, 80CD
P3345	80386SX	Magnavox Maxstation 386SX, Magnum SX; Headstart Series 500
P33711	80386DX	Headstart/Maxstation/Magnum/Professional 3300

Code	Beeps0=sh 1=lng	Meanings (Port 80)
0A	1010	DMA page register write/read bad
10	1 0000	CMOS RAM read/write error (only after hard reset)
11	1 0001	System ROM BIOS checksum error
12	1 0010	Timer A error
13	1 0011	DMA controller A error
14	1 0100	DMA controller B error
15	1 0101	Memory refresh error
16	1 0110	Keyboard controller error
17	1 0111	Keyboard controller error
19	1 1001	Keyboard controller error
1C	1 1100	Base 64K RAM error
1D	1 1101	Base 64K RAM parity error
1F	1 1111	Orvonton LSI sync missing
21	10 0001	PVAM register error
25	10 0101	System options error
2B	10 1011	Video sync error (incorrect switch or CMOS RAM)
2C	10 1100	Video BIOS ROM error
2D	10 1101	Monochrome/colour configuration error
2E	10 1110	No video memory
35	11 0101	Interrupt controller error
36	11 0110	Byte swapper error
37	11 0111	NMI error
38	11 1000	Timer interrupt
39	11 1001	LSI timer halted
3A	11 1010	Main memory test error
3B	11 1011	Keyboard error
3C	11 1100	Keyboard interrupt error (only after hard reset)
3D	11 1101	DDNIL scan halted, cache disabled
40	100 0000	Diskette error
48	100 1000	Adapter card error
4c	100 1100	CMOS battery/checksum error (run SETUP)
4D	100 1101	System options error (run Setup)
52	101 0010	Keyboard controller error
6A	110 1010	Failure shadowing BIOS ROM
70	111 0000	Memory size configuration error (run SETUP)

PHOENIX

On 4.3 and above, the system will generate a code with four groups of beeps, with 1-4 per group. Micro channel ones send codes to 680.

Architecture	Typical Computer	POST Port
ISA	XT	60
	AT	80
	PS/2 25/30	90
EISA	Intel chipset	80
MCA	PS/2 50 up	68O

POST Procedures

Procedure	Meaning
CPU	Check internal operations e.g. ALE/IRQ status; Request; ALU and Memory Read/Write.
CMOS RAM	Test with walking-bit pattern.
ROM BIOS	Perform checksum on ROM BIOS where all bits are added and compared to a factory-set total.
PIT	Check to ensure interrupt requests are properly executed.
DMA	Check DMA from CPU to memory without BIOS. Also check page registers.
Base 64K	Check first 64K block.
Serial and Parallel	I/O data areas for any devices found are assigned; they are not tested.
PIC	Check that proper interrupt request levels are addressed.
Keyboard Controller	Check 8240 for proper operationluding scan code response and Gate A20 which allows CPU operation in protected mode.
CMOS	Check data within CMOS and compare to BIOS information. Failure of the extended area is often due to wrong data setup. Constant failure after resetting CMOS is either battery CMOS chip or RTC.
Video Controller	Test and initialise controller and ROM on the video adapter.
RTC	Check to ensure proper frequencies are on proper lines for the Video Colour CPU and DMA Frequency. Check RTC/PIT or system crystal.
CPU	Return From Protected Mode. CPU is put into protected mode and returns to the POST at the point indicated by the CMOS ROM data area byte 0F. Failure here is normally due to the CPU/keyboard controller/CMOS chip or an address line.
PIC	Test Counter 2.
NMI	Check the Non-Maskable Interrupt request vector for active status. Failure is normally due to the CMOS but could also be the BIOS IRQ or CPU chips.
Keyboard	Check for NumLock/Caps and Shift Keys.
Mouse	Initialise through the keyboard controller; this is only done if a mouse is present and it is initialised in this way.
RAM above 64K	Test in 64K blocks with a walking-bit pattern and parity enabled.
Fixed/Floppy Controllers	Test for proper response to BIOS calls.
Shadow RAM Areas	Look in CMOS for settings on which adapter or system ROMs are to be shadowed.
Option ROM	Look for ROM signatures of 55AA in extended memory then initialise the ROM and halt testing while internal checks are carried out.
External Cache	Check controller chip for external cache.
CPU Int Cache	
Hardware Adapters	Initialise and test video/floppy/hard I/O adapters/serial and parallel.
Cassette	Test internal or external cassette drives.
Boot Code Errors	Errors occurring after this point are normally a corrupt boot record.

2.52 BNP XT

Code	Meaning
01	Test 8253 timer
02	First 64K RAM failed
03	First 1K parity check failed
04	Initialize 8259 interrupt controller
05	Second 1K RAM test (BIOS data area) failed

BIOS Plus or v1.0 POST/Beep Codes

Only for BIOS PLUS or A286/A386/A486 Version 1.xx on an AT-class (80286 or higher) systems. Codes in the 50h range or beyond are chipset or platform specific, and will vary from system to system.

Code	Beeps	Meaning
01	none	CPU register test in progress.
02	1-1-3	CMOS write/read failure.
03	1-1-4	ROM BIOS Checksum Failure.
04	1-2-1	Programmable interval timer failure.
05	1-2-2	DMA Initialisation failure.
06	1-2-3	DMA page register write/read failure.
08	1-3-1	RAM refresh verification failure.
09	none	1st 64K RAM test in progress.
0A	1-3-3	1st 64K RAM chip or data line failure multi-bit.
0B	1-3-4	1st RAM odd/even logic failure.
0C	1-4-1	Address line failure 1st 64K RAM.
0D	1-4-2	Parity failure 1st 64K RAM.
10	2-1-1	Bit 0 1st 64K RAM failure.
11	2-1-2	Bit 1 1st 64K RAM failure.
12	2-1-3	Bit 2 1st 64K RAM failure.
13	2-1-4	Bit 3 1st 64K RAM failure.
14	2-2-1	Bit 4 1st 64K RAM failure.
15	2-2-2	Bit 5 1st 64K RAM failure.
16	2-2-3	Bit 6 1st 64K RAM failure.
17	2-2-4	Bit 7 1st 64K RAM failure.
18	2-3-1	Bit 8 1st 64K RAM failure.
19	2-3-2	Bit 9 1st 64K RAM failure.
1A	2-3-3	Bit A(10) 1st 64K RAM failure.
1B	2-3-2	Bit B(11) 1st 64K RAM failure.
1C	2-4-2	Bit C(12) 1st 64K RAM failure.
1D	2-4-2	Bit D(13) 1st 64K RAM failure.
1E	2-4-3	Bit E(14) 1st 64K RAM failure.
1F	2-4-4	Bit F(15) 1st 64K RAM failure.
20	3-1-1	Slave DMA register failure.
21	3-1-2	Master DMA register failure.
22	3-1-3	Master interrupt mask register failure.
23	3-1-4	Slave interrupt mask register failure.
25	none	Interrupt vector loading in progress.
27	3-2-4	8042 keyboard controller test failure.
28	none	CMOS power failure/checksum calculation in progress.
29	none	CMOS configuration validation in progress.
2B	3-3-4	Screen memory test failure.
2C	3-4-1	Screen initialisation failure.
2D	3-4-2	Screen retrace test failure.
2E	none	Search for video ROM in progress.
30	none	Screen believed running with video ROM.
31	none	Mono monitor believed operable.
32	none	Colour monitor (40 col) believed operable.
33	none	Colour monitor (80 col) believed operable.
34	4-2-1	Timer tick interrupt test in progress or failed (non-fatal).

Code	Beeps	Meaning
35	4-2-2	Shutdown failure (non-fatal).
36	4-2-3	Gate A20 failure (non-fatal).
37	4-2-4	Unexpected interrupt in protected mode (non-fatal).
38	4-3-1	Mem high address line fail at 01000-0A000 (non-fatal).
39	4-3-2	Mem high addr line fail at 100000-FFFFFF (non-fatal).
3A	4-3-3	Timer chip counter 2 failed (non-fatal).
3B	4-3-4	Time-of-day clock stopped
3C	4-4-1	Serial port test
3D	4-4-2	Parallel port test
3E	4-4-3	Maths coprocessor test
41	low 1-1-2	System board select bad
42	low 1-1-3	Extended CMOS RAM bad

PCI

Code	Meaning
02	If the CPU is in protected mode turn on A20 and pulse the reset line; forcing a shutdown 0.
04	On a cold boot save the CPU type information value in the CMOS.
06	Reset DMA controllers. Disable videos. Clear pending interrupts from RTC. Setup port B register.
08	Initialise chipset control registers to power on defaults.
0A	Set a bit in the CMOS that indicates POST; used to determine if the current configuration causes the BIOS to hang. If so default values will be used on next POST.
0C	Initialise I/O module control registers.
0E	External CPU caches are initialised. Cache registers are set to default.
10/12/14	Verify response of 8742.
16	Verify BIOS ROM checksums to zero.
18	Initialise all three of 8254 timers.
1A	Initialise DMA command register. Initialise 8 DMA channels.
1C	Initialise 8259 interrupt controller to :ICW4 needed; Cascade and edge-triggered mode.
20	Test DRAM refresh by polling refresh bit in PORTB.
22	Test 8742 keyboard controller. Send self test command to 8742 and await results. Also read the switch inputs from the 8742 and write the keyboard controller command byte.
24	Set ES segment register to 4 Gb
26	Enable Address Line A20
28	Autosize DRAM
2A	Clear first 64K of RAM
2C	Test RAM address lines
2E	Test first 64K bank of memory consisting of a chip address line test and a RAM test.
30/32	Find true MHz value
34	Clear CMOS diagnostic byte (register E). Check RTC and verify battery has not lost power. Checksum the CMOS and verify it has not been corrupted.
36/38/3A	External cache is autosized and its configuration saved for enabling later in POST.
3C	Configure advanced cache features. Configure external cache's configurable parameters.
3E	Read hardware configuration from keyboard controller
40	Set system power-on speed to rate determined by CMOS. If CMOS is invalid use a conservative speed.
42	Initialise interrupt vectors 0-77h to the BIOS general interrupt handler.
44	Initialise interrupt vectors 0-20h to proper values from the BIOS interrupt table.
46	Check copyright message checksum.
48	Check video configuration.

Code	Meaning
4A	Initialise both monochrome and colour graphics video adapters.
4C/4E	Display Copyright message.
50	Display CPU type and speed
52	Test for the self-test code if a cold start. When powered the keyboard performs a self-test and sends an AA if successful.
54	Initialise keystroke clicker during POST.
56	Enable keyboard
58	Test for unexpected interrupts. First do an STI for hot interrupts; secondly test NMI for unexpected interrupt. Thirdly enable parity checkers and read from memory checking for unexpected interrupt.
5A	Display prompt Press F2 to Enter Setup
5C	Determine and test the amount of memory available. Save the total memory size in the BIOS variable called bdaMemorySize.
5E	Perform address test on base memory. Following address lines are tested based on the memory size.
60	Determine and test the amount of extended memory available. Save the total extended memory size in the CMOS at CMOSExtended.
62	Perform an address line test on A0 to the amount of memory available. This test is dependent on the processor since the test will vary depending on the width of memory (16 or 32 bits). This test will also use A20 as the skew address to prevent corruption of the system memory.
68	External and CPU caches if present are enabled. Non-cacheable regions are configured if necessary.
6A	Display cache size on screen if non-zero.
6C	Display BIOS shadow status.
6E	Display the starting offset of the non-disposable section of the BIOS.
70	Check flags in CMOS and in the BIOS data area to see if any errors have been detected during POST. If so, display error messages on the screen.
72	Check status bits for configuration errors. If so display error messages on the screen.
74	Test RTC if the battery has not lost power. If the RTC is not running or the battery has lost powerset the incorrect time bit in register E of the CMOS.
76	Check status bits for keyboard errors. If so display error messages on the screen.
78	Check for stuck keys on the keyboard. If so display error messages on the screen.
7A	Enable keylock
7C	Set up hardware interrupt vectors
7E	Test coprocessor if present
80-82	Detect and install RS232 ports
84	Detect and install parallel ports
86-88	Initialise timeouts/key buffer/soft reset flag.
8A	Initialise extended BIOS data area and initialise the mouse.
8C	Initialise both floppy disks and display an error message if failure was detected. Both drives are checked so the appropriate diskette types are established in the BIOS data area.
8E	Hard disk autotype configuration
90	If the CMOS RAM is valid and intact and fixed disks are defined call the fixed disk init routine to initialise the fixed disk system and take over the appropriate interrupt vectors.
92-94	Disable A20 address line
96-98-	Scan for ROM BIOS extensions.
9E	Enable hardware interrupts
A0	Set time of day
A2	Set up NumLock indicator. Display a message if key switch is locked.
A4	Initialise typematic rate.
A6	Initialise hard disk autoparking.
A8	Erase F2 prompt.
AA	Scan for F2 key strokes.
AC	Check to see if SETUP should be executed.

Code	Meaning
AE	Clear ConfigFailedBit and InPostBit in CMOS.
B0	Check for POST errors
B2	Set/clear status bits to reflect POST complete.
B4	One beep.
B6	Check for password before boot.
B8	Clear global descriptor table (GDT).
BA	Initialise the screen saver.
BC	Clear parity error latch.
BE	Clear screen.
C0	Try to boot with INT 19
D0-D2	If an interrupt occurs before interrupt vectors have been initialised this interrupt handler will try to see if the interrupt caused was an 8259 interrupt and which one. If unknown, InterruptFlag will be FF. Otherwise it will contain the IRQ number that occurred
D4	Clear pending timer, kbd interrupts, transfer control to double word address at RomCheck.
D6-D8-DA	Return from extended block move.

Phoenix v3.07

see Quadtel.

ISA/EISA/MCA BIOS POST/Beep Codes (fatal)

Msg	Beeps	Meaning
01	none	CPU register test in progress.
02	1-1-3	CMOS write/read failure.
03	1-1-4	ROM BIOS Checksum Failure.
04	1-2-1	Programmable interval timer failure.
05	1-2-2	DMA Initialisation failure.
06	1-2-3	DMA page register write/read failure.
08	1-3-1	RAM refresh verification failure.
09	none	1st 64K RAM test in progress.
0A	1-3-3	1st 64K RAM chip or data line failure multi-bit.
0B	1-3-4	1st RAM odd/even logic failure.
0C	1-4-1	Address line failure 1st 64K RAM.
0D	1-4-2	Parity failure 1st 64K RAM.
0E	1-4-3	Fail-safe timer failure.
0F	1-4-4	Software NMI port failure.
10	2-1-1	Bit 0 1st 64K RAM failure.
11	2-1-2	Bit 1 1st 64K RAM failure.
12	2-1-3	Bit 2 1st 64K RAM failure.
13	2-1-4	Bit 3 1st 64K RAM failure.
14	2-2-1	Bit 4 1st 64K RAM failure.
15	2-2-2	Bit 5 1st 64K RAM failure.
16	2-2-3	Bit 6 1st 64K RAM failure.
17	2-2-4	Bit 7 1st 64K RAM failure.
18	2-3-1	Bit 8 1st 64K RAM failure.
19	2-3-2	Bit 9 1st 64K RAM failure.
1A	2-3-3	Bit A 1st 64K RAM failure.
1B	2-3-2	Bit B 1st 64K RAM failure.

Msg	Beeps	Meaning
1C	2-4-2	Bit C 1st 64K RAM failure.
1D	2-4-2	Bit D 1st 64K RAM failure.
1E	2-4-3	Bit E 1st 64K RAM failure.
1F	2-4-4	Bit F 1st 64K RAM failure.
20	3-1-1	Slave DMA register failure.
21	3-1-2	Master DMA register failure.
22	3-1-3	Master interrupt mask register failure.
23	3-1-4	Slave interrupt mask register failure.
25	none	Interrupt vector loading in progress.
27	3-2-4	Keyboard controller test failure.
28	none	CMOS pwr failure; checksum calculation in progress.
29	none	CMOS RAM configuration validation in progress.
2B	3-3-4	Screen memory test failure.
2C	3-4-1	Screen initialisation failure.
2D	3-4-2	Screen retrace test failure.
2E	none	Search for video ROM in progress.
30	none	Screen believed running with video ROM.
31	none	Mono monitor believed operable.
32	none	Colour monitor (40 col) believed operable.
33	none	Colour monitor (80 col) believed operable.

ISA/EISA/MCA BIOS POST/Beep Codes (non-fatal)

Non-fatal if manufacturing jumper is on.

Msg	Beeps	Meaning
34	4-2-1	No time tick.
35	4-2-2	Shutdown test in progress or failure.
36	4-2-3	Gate A20 failure.
37	4-2-4	Unexpected interrupt in protected mode.
38	4-3-1	Memory high address line fail at 01000-0A000. Also RAM test in progress or address failure >FFFH.
39	4-3-2	Memory high address line failure at 100000-FFFFFF.
3A	4-3-3	Interval Timer channel 2 test or failure.
3B	4-3-4	Time-of-day clock test or failure.
3C	4-4-1	Serial port test or failure.
3D	4-4-2	Parallel port test or failure.
3E	4-4-3	Maths coprocessor test
3F		Cache test (Dell)
41	lw 1-1-2	System board select bad (Micro Channel only)
42	Low 1-1-3	Extended CMOS RAM bad (Micro Channel only)

PicoBIOS v4.0 R6/UMC Chipset PCI

Beeps	Code	Meaning
1-1-1-3	02	Verify Real Mode
1-1-2-1	04	Get CPU type
1-1-2-3	06	Initialize system hardware
1-1-3-1	08	Initialize chipset registers with initial POST values

Beeps	Code	Meaning
1-1-3-2	09	Set in POST flag
1-1-3-3	0A	Initialize CPU registers
1-1-4-1	0C	Initialize cache to initial POST values
1-1-4-3	0E	Initialize I/O
1-2-1-1	10	Initialize Power Management
1-2-1-2	11	Load alternate registers with initial POST values
1-2-1-3	12	Jump to UserPatch0
1-2-2-1	14	Initialize keyboard controller
1-2-2-3	16	BIOS ROM checksum
1-2-3-1	18	8254 timer initialization
1-2-3-3	1A	8237 DMA controller initialization
1-2-4-1	1C	Reset Programmable Interrupt Controller
1-3-1-1	20	Test DRAM refresh
1-3-1-3	22	Test 8742 Keyboard Controller
1-3-2-1	24	Set ES segment to register to 4 GB
1-3-3-1	28	Autosize DRAM
1-3-3-3	2A	Clear 512K base RAM
1-3-4-1	2C	Test 512 base address lines
1-3-4-3	2E	Test 512K base memory
1-4-1-3	32	Test CPU bus-clock frequency
1-4-2-1	34	CMOS RAM read/write failure (check ISA card seating)
1-4-2-4	37	Reinitialize the chipset
1-4-3-1	38	Shadow system BIOS ROM
1-4-3-2	39	Reinitialize the cache
1-4-3-3	3A	Autosize cache
1-4-4-1	3C	Configure advanced chipset registers
1-4-4-2	3D	Load alternate registers with CMOS values
2-1-1-1	40	Set Initial CPU speed
2-1-1-3	42	Initialize interrupt vectors
2-1-2-1	44	Initialize BIOS interrupts
2-1-2-3	46	Check ROM copyright notice
2-1-2-4	47	Initialize manager for PCI Options ROMs
2-1-3-1	48	Check video configuration against CMOS
2-1-3-2	49	Initialize PCI bus and devices
2-1-3-3	4A	Initialize all video adapters in system
2-1-4-1	4C	Shadow video BIOS ROM
2-1-4-3	4E	Display copyright notice
2-2-1-1	50	Display CPU type and speed
2-2-1-3	52	Test keyboard
2-2-2-1	54	Set key click if enabled
2-2-2-3	56	Enable keyboard
2-2-3-1	58	Test for unexpected interrupts
2-2-3-3	5A	Display prompt Press F2 to enter SETUP
2-2-4-1	5C	Test RAM between 512 and 640k
2-3-1-1	60	Test expanded memory
2-3-1-3	62	Test extended memory address lines
2-3-2-1	64	Jump to UserPatch1
2-3-2-3	66	Configure advanced cache registers
2-3-3-1	68	Enable external and CPU caches

Beeps	Code	Meaning
2-3-3-2	69	Initialise SMI handler
2-3-3-3	6A	Display external cache size
2-3-4-1	6C	Display shadow message
2-3-4-3	6E	Display non-disposable segments
2-4-1-1	70	Display error messages
2-4-1-3	72	Check for configuration errors
2-4-2-1	74	Test real-time clock
2-4-2-3	76	Check for keyboard errors
2-4-4-1	7C	Set up hardware interrupts vectors
2-4-4-3	7E	Test coprocessor if present
3-1-1-1	80	Disable onboard I/O ports
3-1-1-3	82	Detect and install external RS232 ports
3-1-2-1	84	Detect and install external parallel ports
3-1-2-3	86	Re-initialize onboard I/O ports
3-1-3-1	88	Initialize BIOS Data Area
3-1-3-3	8A	Initialize Extended BIOS Data Area
3-1-4-1	8C	Initialize floppy controller
3-2-1-1	90	Initialize hard-disk controller
3-2-1-2	91	Initialize local-bus hard-disk controller
3-2-1-3	92	Jump to UserPatch2
3-2-2-1	94	Disable A20 address line
3-2-2-3	96	Clear huge ES segment
3-2-3-1	98	Search for option ROMs
3-2-3-3	9A	Shadow option ROMs
3-2-4-1	9C	Set up Power Management
3-2-4-3	9E	Enable hardware interrupts
3-3-1-1	A0	Set time of day
3-3-1-3	A2	Check key lock
3-3-3-1	A8	Erase F2 prompt
3-3-3-3	AA	Scan for F2 key stroke
3-3-4-1	AC	Enter SETUP
3-3-4-3	AE	Clear in-POST flag
3-4-1-1	B0	Check for errors
3-4-1-3	B2	POST done--prepare to boot operating system
3-4-2-1	B4	One beep
3-4-2-3	B6	Check password (optional)
3-4-3-1	B8	Clear global descriptor table
3-4-4-1	BC	Clear parity checkers
3-4-4-3	BE	Clear screen (optional)
3-4-4-4	BF	Check virus and backup reminders
4-1-1-1	C0	Try to boot with INT 19
4-2-1-1	D0	Interrupt handler error
4-2-1-3	D2	Unknown interrupt error
4-2-2-1	D4	Pending interrupt error
4-2-2-3	D6	Initialize option ROM error
4-2-3-1	D8	Shutdown error
4-2-3-3	DA	Extended Block Move
4-2-4-1	DC	Shutdown 10 error

FLASH BIOS INTEGRITY TEST

4-3-1-3	E2	Initialize the chipset
4-3-1-4	E3	Initialize refresh counter
4-3-2-1	E4	Check for Forced Flash
4-3-2-2	E5	Check HW status of ROM
4-3-2-3	E6	BIOS ROM is OK
4-3-2-4	E7	Do a complete RAM test

FLASH RECOVERY

4-3-3-1	E8	Do OEM initialization
4-3-3-2	E9	Initialize interrupt controller
4-3-3-3	EA	Read in bootstrap code
4-3-3-4	EB	Initialize all vectors
4-3-4-1	EC	Boot the Flash program
4-3-4-2	ED	Initialize the boot device
4-3-4-3	EE	Boot code was read OK

v4.0 R6

Code	Meaning	Code	Meaning
02	Verify Real Mode	6B	Load custom defaults (optional)
03	Disable NMI	6C	Display shadow area message
04	Get CPU type	6E	Show possible high address UMB recovery
06	Init system hardware	70	Display error messages
08	Init chipset registers with initial POST values	72	Check for configuration errors
09	Set IN POST flag	76	Check for keyboard errors
0A	Init COU registers	7C	Set up hardware interrupt vectors
0B	Enable CPU cache	7E	Init copro if present
0C	Init caches to initial POST values	80	Disable onboard super I/O ports and IRQs
0E	Init I/O component	81	Late POST device inti
0F	Init local bus IDE	82	Detect and install external RS232 ports
10	Init power management	83	Configure non-MCD IDE controllers
11	Load alternate registers with initial POST values	84	Detect and install external parallel ports
12	Restore CPU control word during warm boot	85	Init PC-compatible PnP ISA devices
13	Init PCI bus mastering devices	86	Re-initialise onboard I/O ports
14	Init keyboard controller	87	Set motherboard configurable devices
16	BIOS ROM checksum (beep 1-2-2-3)	88	Init BIOS data area
17	Init cache before memory autosize	89	Enable NMIs
18	8254 timer init	8A	Init BIOS extended data area
1A	8237 DAM controller init	8B	Test/init PS/2 mouse
1C	Reset programmable interrupt controller	8C	Init floppy controller
20	Test DRAM refresh (beep 1-3-1-1)	8F	Determine number of ATA drives (optional)
22	Test 8742 controller (beep 1-3-1-3)	90	Init HD controller
24	Set ES segment register to 4 Gb	91	Init local bus HD controller
26	Enable A20 line	92	Jump to Userpatch2
28	Autosize DRAM	93	Build MPTABLE for multi-processor boards

Code	Meaning	Code	Meaning
29	Init POST memory manager	95	Install CD ROM for boot
2A	Clear 512K base RAM	96	Clear huge ES segment register
2C	RAM failure on line XXXX (beep 1-3-4-1)	97	Fix up multiprocessor table
2E	RAM failure on data bits xxxx of low byte of memory bus (beep 1-3-4-3)	98	Search for option ROMs (beep 1-2)
2F	Enable cache before system BIOS shadow	99	Check for SMART Drive (optional)
30	RAM failure on data bits xxxx of high byte of memory bus (beep 1-4-1-1)	9A	Shadow Option ROMs
32	Test CPU bus clock frequency	9C	Set up power management
33	Init Phoenix Dispatch Manager	9D	Init security engine (optional)
36	Warm start shut down	9E	Enable hardware interrupts
38	Shadow system BIOS ROM	9F	Determine number of ATA and SCSI drives
3A	Autosize cache	A0	Set time of day
3C	Advanced conguration of chipset registers	A2	Check key lock
3D	Load alternate registers with CMOS values	A4	Init typematic rate
42	Init interrupt vectors	A8	Erase F2 prompt
45	POST device init	AA	Scan for F2 keystroke
46	Check ROM copyright notice (beep 2-1-2-3)	AC	Enter setup
48	Check video configuration against CMOS	AE	Clear boot flag
49	Init PCI bus and devices	B0	Check for errors
4A	Init all video adapters	B2	POST done, prepare to boot OS
4B	QuietBoot start (optional)	B4	One short beep before boot
4C	Shadow Video BIOS ROM	B5	Terminate Quickboot (optional)
4E	Display BIOS copyright notice	B6	Check password (optional)
50	Display CPU type and speed	B9	Prepare Boot
51	Init EISA board	BA	Init DMI parameters
52	Test keyboard	BB	Init PnP Option ROMs
54	Set key click if enabled	BC	Clear parity checkers
58	Test for unexpected interrupts (2-2-3-1)	BD	Display multiboot menu
59	Init POST display service	BE	Clear Screen (optional)
5A	Display prompt "Press F2 to enter setup"	BF	Check virus and backup reminders
5B	Disable CPU cache	C0	Try to boot with INT 19
5C	Test RAM between 512K and 640K	C1	Init POSR error message (PEM)
60	Test expanded memory	C2	Init error logging
62	Test extended memory address lines	C3	Init error display function
64	Jump to user patch 1	C4	Init system error handler
66	Configure advanced cache registers	C5	PnPnd dual CMOS (optional)
67	Init multi processor APIC	C6	Init notebook docking (optional)
68	Enable external and CPU caches	C7	Init notebook docking late
69	Setup SMM area	C8	Force Check (optional)
6A	Display external L2 cache size	C9	Extended Checksum (optional)

FOR BOOT BLOCK IN FLASH ROM

Code	Meaning	Code	Meaning
E0	Init chipset	EA	Init OEM special code
E1	Init bridge	EB	Init PIC and DMA
E2	Init CPU	EC	Init memory type
E3	Init system timer	ED	Init memory size

Code	Meaning	Code	Meaning
E4	Init system I/O	F0	Init interrupt vectors
E5	Check force recovery boot	F1	Init RTC
E6	Checksum BIOS ROM	F2	Init video
E7	Go to BIOS	F3	Init system management mode
E8	Set huge segment	F4	Out put 1 beep before boot
E9	Init multi-processor	F5	Boot to mini DOS
EE	Shadow boot block	F6	Clear Huge Segment
EF	System memory test	F7	Boot to full DOS

V4 Rel 6

Code	Meaning	Code	Meaning
000h	TP_NULL	045h	TP_DEVICE_INIT
001h	TP_IPMI_INIT	046h	TP_COPYRIGHT
002h	TP_VERIFY_REAL	047h	TP_I2O_INIT
003h	TP_DISABLE_NMI	048h	TP_CONFIG
004h	TP_GET_CPU_TYPE	049h	TP_PCI_INIT
006h	TP_HW_INIT	04Ah	TP_VIDEO
007h	TP_CS_BIOS_DESHAD	04Bh	TP_QUIETBOOT_START
008h	TP_CS_INIT	04Ch	TP_VID_SHADOW
009h	TP_SET_IN_POST	04Eh	TP_CR_DISPLAY
00Ah	TP_CPU_INIT	04Fh	TP_MULTBOOT_INIT
00Bh	TP_CPU_CACHE_ON	050h	TP_CPU_DISPLAY
00Ch	TP_CACHE_INIT	051h	TP_EISA_INIT
00Eh	TP_IO_INIT	052h	TP_KB_TEST
00Fh	TP_FDISK_INIT	054h	TP_KEY_CLICK
010h	TP_PM_INIT	055h	TP_USB_INIT
011h	TP_REG_INIT	056h	TP_ENABLE_KB
012h	TP_RESTORE_CR0	057h	TP_1394_INIT
013H	TP_PCI_BM_RESET	058h	TP_HOT_INT
014h	TP_8742_INIT	059h	TP_PDS_INIT
016h	TP_CHECKSUM	05Ah	TP_DISPLAY_F2
017h	TP_PRE_SIZE_RAM	05Bh	TP_CPU_CACHE_OFF
018h	TP_TIMER_INIT	05Ch	TP_MEMORY_TEST
01Ah	TP_DMA_INIT	05Eh	TP_BASE_ADDR
01Ch	TP_RESET_PIC	060h	TP_EXT_MEMORY
020h	TP_REFRESH	062h	TP_EXT_ADDR
022h	TP_8742_TEST	064h	TP_USERPATCH1
024h	TP_SET_HUGE_ES	066h	TP_CACHE_ADVNCD
026h	TP_ENABLE_A20	067h	TP_MP_INIT_MIN
028h	TP_SIZE_RAM	068h	TP_CACHE_CONFIG
029h	TP_PMM_INIT	069h	TP_PM_SETUP_SMM
02Ah	TP_ZERO_BASE	06Ah	TP_DISP_CACHE
02Bh	TP_ENH_CMOS_INIT	06Bh	TP_CUST_DFLT
02Ch	TP_ADDR_TEST	06Ch	TP_DISP_SHADOWS
02Eh	TP_BASERAML	06Eh	TP_FAST_ZERO
02Fh	TP_PRE_SYS_SHADOW	070h	TP_ERROR_MSGS
030h	TP_BASERAMH	072h	TP_TEST_CONFIG

Code	Meaning	Code	Meaning
032h	TP_COMPUTE_SPEED	074h	TP_RTC_TEST
033h	TP_PDM_INIT	076h	TP_KEYBOARD
034h	TP_CMOS_TEST	07Ah	TP_KEYLOCK
035h	TP_REG_REINIT	07Ch	TP_HW_INTS
036h	TP_CHK_SHUTDOWN	07Dh	TP_ISM_INIT
037h	TP_CS_REINIT	07Eh	TP_COPROC
038h	TP_SYS_SHADOW	080h	TP_IO_BEFORE
039h	TP_CACHE_REINIT	0BDh	TP_BOOT_MENU
03Ah	TP_CACHE_AUTO	0BEh	TP_CLEAR_SCREEN
03Bh	TP_DBGSRV_INIT	0BFh	TP_CHK_RMDR
03Ch	TP_ADV_CS_CONFIG	0C0h	TP_INT19
03Dh	TP_ADV_REG_CONFIG	0C1h	TP_PEM_INIT
03Eh	TP_READ_HW	0C2h	TP_PEM_LOG
03Fh	TP_ROMPILOT_MEMORY	0C3h	TP_PEM_DISPLAY
040h	TP_SPEED	0C4h	TP_PEM_SYSER_INIT
041h	TP_ROMPILOT_INIT	0C5h	TP_DUAL_CMOS
042h	TP_VECTOR_INIT	0C6h	TP_DOCK_INIT
044h	TP_SET_BIOS_INT	0C7h	TP_DOCK_INIT_LATE
08Ch	TP_FLOPPY	0C8h	TP_FORCE
08Eh	TP_AUTOTYPE	0C9h	TP_EXT_CHECKSUM
08Fh	TP_FDISK_FAST_PREINIT	0CAh	TP_SERIAL_KEY
090h	TP_FDISK	081h	TP_LATE_DEVICE_INIT
091h	TP_FDISK_FAST_INIT	082h	TP_RS232
092h	TP_USERPATCH2	083h	TP_FDISK_CFG_IDE_CTRLR
093h	TP_MP_INIT	084h	TP_LPT
095h	TP_CD	085h	TP_PCI_PCC
096h	TP_CLEAR_HUGE_ES	086h	TP_IO_AFTER
097h	TP_MP_FIXUP	087h	TP_MCD_INIT
098h	TP_ROM_SCAN	088h	TP_BIOS_INIT
099h	TP_FDISK_CHECK_SMART	089h	TP_ENABLE_NMI
09Ah	TP_MISC_SHADOW	08Ah	TP_INIT_EXT_BDA
09Bh	TP_PMCPUSPEED	08Bh	TP_MOUSE
09Ch	TP_PM_SETUP	0CDh	TP_PCMATA
09Dh	TP_SECURITY_INIT	0CEh	TP_PEN_INIT
09Eh	TP_IRQS	0CFh	TP_XBDA_FAIL
09Fh	TP_FDISK_FAST_INIT2	0D1h	TP_BIOS_STACK_INIT
0A0h	TP_TIME_OF_DAY	0D3h	TP_SETUP_WAD
0A2h	TP_KEYLOCK_TEST	0D4h	TP_CPU_GET_STRING EQU
0A4h	TP_KEY_RATE	0D5h	TP_SWITCH_POST_TABLES
0A8h	TP_ERASE_F2	0C1h	TP_CHKBOOTTYPE
0AAh	TP_SCAN_FOR_F2	0C2h	TP_SAVEBOOTTYPE
0ACh	TP_SETUP_CHECK	0C3h	TP_CHKREQBOOTTYPE
0AEh	TP_CLEAR_BOOT	0C4h	TP_HOTKEY_START
0B0h	TP_ERROR_CHECK	0C5h	TP_HOTKEY_END
0B1h	TP_ROMPILOT_UNLOAD	0C6h	TP_CONSOLE_INIT
0B2h	TP_POST_DONE	0C7h	TP_CONSOLE_COMPORT
0B3h	TP_ENH_CMOS_STORE	0C8h	TP_A20_TEST
0B4h	TP_ONE_BEEP	0C9h	TP_EISA_BEFORE_INIT
0B5h	TP_QUIETBOOT_END	0CAh	TP_EISA_AFTER_INIT

Code	Meaning	Code	Meaning
0B6h	TP_PASSWORD	0CBh	TP_SAVE_MEMCFG
0B7h	TP_ACPI	0CCh	TP_RESTORE_MEMCFG EQU
0B8h	TP_SYSTEM_INIT	0CDh	TP_CONSOLE_VECTOR EQU
0B9h	TP_PREPARE_BOOT	0CEh	TP_ERRLOG_INIT
0Bah	TP_DMI	0CFh	TP_ERRLOG_MSG
0BBh	TP_INIT_BCVS	0CDh	TP_PCMATA
0BCh	TP_PARITY		

QUADTEL

v3.07 AT BIOS (Phoenix 3.07)

Code	Meaning	Code	Meaning
02	Flag test	4A	Start second protected mode test
04	Register test	4C	Verify LDT instruction
06	System hardware initialisation	4E	Verify TR instruction
08	Initialise chipset registers	50	Verify LSL instruction
0A	BIOS ROM checksum	52	Verify LAR instruction
0C	DMA page register test	54	Verify VERR instruction
0E	8254 timer test	56	Unexpected exception
10	8254 timer initialisation	58	Address line 20 test
12	8237 DMA controller test	5A	Keyboard ready test
14	8237 DMA initialisation	5C	Determine AT or XT keyboard
16	Initialise 8259/reset coprocessor	5E	Start third protected mode test
18	8259 interrupt controller test	60	Base memory test
1A	Memory refresh test	62	Base memory address test
1C	Base 64K address test	64	Shadow memory test
1E	Base 64K memory test	66	Extended memory test
20	Base 64K test (upper 16 bits) for 386s	68	Extended address test
22	8742 keyboard self test	6A	Determine memory size
24	MC 146818 CMOS test	6C	Display error messages
26	Start first protected mode test	6E	Copy BIOS to shadow memory
28	Memory sizing test	70	8254 clock test
2A	Autosize memory chips	72	MC 146818 RTC test
2C	Chip interleave enable test	74	Keyboard stuck key test
2E	First protected mode test exit	76	Initialise hardware interrupt vectors
30	Unexpected shutdown	78	Maths coprocessor test
31	DDNIL bit scan failure	7A	Determine COM ports available
32	System board memory size	7C	Determine LPT ports available
34	Relocate shadow RAM if configured	7E	Initialise BIOS data area
36	Configure EMS system	80	Determine floppy/fixed disk controller
38	Configure wait states	82	Floppy disk test
3A	Retest 64K base RAM	84	Fixed disk test
3C	CPU speed calculation	86	External ROM scan
3E	Get switches from 8042	88	System key lock test
40	Configure CPU speed	8A	Wait for <F1> key pressed

Code	Meaning	Code	Meaning
42	Initialise interrupt vectors	8C	Final system initialisation
44	Verify video configuration	8E	Interrupt 19 boot loader
46	Initialise video system	B0	Unexpected interrupt before or after boot.
48	Test unexpected interrupts		

16K XT

Code	Meaning
03	Test flag register
06	Test CPU Register
09	Initialise system hardware
0C	Test BIOS ROM checksum
0F	Initialise 8237 DMA page register
12	Test 8237 address and count registers
15	Initialise 8237 DMA
18	Test 8253 timer
1B	Initialise 8253 timer
1E	Start memory refresh test
21	Test base 64K RAM, Cycling POST display shows code, upper then lower bytes of failing address
24	Set up common INT temp stack
27	Initialize 8259 interrupt controller
2A	Test interrupt mask register
2D	Test for hot (unexpected) interrupt
30	Test V40 DMA if present
31	Test for DDNIL bits present
33	Verify system clock interrupt
36	Test keyboard
39	Set up interrupt table
3C	Read system configuration switches
3F	Test video
42	Determine COM ports available
45	Determine LPT ports available
48	Determine if game port available
4B	Display copyright message
4E	Calculate CPU speed
54	Test system memory
55	Test floppy drive
57	Initialize system before boot
5A	Call Interrupt 19 boot loader

SUPERSOFT

PC/XT/AT

	XT	AT
11	CPU register or logic error	CPU register or logic
12	ROM POST checksum error	ROMPOST A checksum error
13	8253 timer channel 0 error	ROMPOST B checksum error
14	8253 timer channel 1 error	8254 timer channel 0 error
15	8253 timer channel 2 error	8254 timer channel 1 error
16	8237A DMA controller error	8254 timer channel 2 error
17	8255 parity error detected	8237A DMA controller 1 err
18	16K critical RAM region error	8237A DMA controller 2 err
19	Memory refresh error	DMA page registers error
1A	-	8042 parity error detected
21	8259 Interrupt controller error	16K critical RAM region
22	Unexpected interrupt detected	Memory refresh error
23	Interrupt 0 (timer) error	CPU protected mode error
24	Nonmaskable interrupt error	8259 Interrupt controller 1 err
25	MDA video memory error	8259 Interrupt controller 2 err
26	CGA video memory error	Unexpected interrupt detected
27	EGA/VGA memory error	Interrupt 0 (timer) error
28	8087 math chip error	CMOS real time clock error
29	Keyboard controller error	Nonmaskable interrupt error
2A	-	8Ox87 math chip error
31	Keyboard scan lines/stuck key	Keyboard controller error
32	Floppy controller error	Stuck key or CMOS RAM err
33	Floppy disk read error	Floppy controller error
34	Memory error at address x	Floppy disk read error
35	Slow refresh, address x	MDA video memory error
36, 37	-	CGA, EGA/VGA RAM error
38	-	BIOS checksum error
41	BIOS checksum error	Memory error at address x
42	BASIC ROM 1 checksum	Slow refresh, address x
43-45	BASIC ROM 2, 3, 4	Display pass count
59	No monitor	No monitor

TANDON

Slimline 286, 386SX and 486; 486 EISA

Type A AT 29 Feb 1988

Code	Meaning
01	Test 80286 CPU flags and registers
02	Test BIOS ROM checksum
03	Test MC146818 CMOS RAM battery (RTC)
04	Test 8254 timer
05	8254 timer test failed
06	Initialize RAM refresh
07	Test first 16K RAM
08	Initialize cold boot interrupt vectors
09	Test 8259 interrupt controller and interrupt vectors
0A	Fill in temporary interrupt vectors
0B	Initialize interrupt vector table 1
0C	Initialize interrupt vector table 2
0D	Initialize fixed disk vector
0E	Interrupt vector test failed
0F	Clear keyboard controller input buffer
10	Keyboard controller input buffer clearing failed
11	Run keyboard controller self-test
12	Initialize equipment check data area
13	Determine presence of and install 80287 math coprocessor
14	Test MC146818 CMOS RAM disk value range
15	Test for and install parallel port
16	Test for and install serial port
17	Invoke INT 19 to boot operating system

Type B AT-1992

Code	Meaning
01	Cold boot started
06	Initialize chipset if any
07	Warm boot entry. About to start 8042 keyboard controller self-test
08	Part of cold boot keyboard initialization passed
09	Keyboard self-test finished. Test ROM BIOS checksum.
0A	Test CMOS RAM battery level
0B	Save CMOS RAM battery condition in CMOS diagnostic/status register
0C	Finished saving CMOS RAM battery condition
0D	Test 8254 PIT. Disable RAM parity, I/O parity, DMA controllers, and speaker; enable timer channel 2.
0E, AA, xx	8245 test failed. xx is the failing channel number.
0F	Initialize 8254 timer channels (0 to mode 3 for 55 ms square wave, 1 to mode 2 as rate generator for refresh) and conduct memory refresh test.
10	Refresh test failed
11	Test base 64K RAM and fill with zeros
12	64K RAM test failed. 3 long beeps and halt.

Code	Meaning
13	RAM test passed
14	Set up stack, disable mappers for systems that support EMS drivers (warm boot), initialize battery beep flag parameters for notebook, perform read/write test of CMOS , enable error message if failed.
15	CMOS RAM read/write test complete
16	Calculating CPU speed; may set to low if CMOS RAM failed
18	Test and initialize both 8259 interrupt controllers
1A	8259 initialization complete
1B	Install interrupt handler and vector for INT 0F to check for unexpected (spurious) interrupts. Halt if spurious interrupt occurs.
1C	Spurious interrupt did not occur (test pass). Test 8254 timer channel 0, IRQ0, and INT8 tests.
1D	Error. Timer 0 interrupt did not occur when expected. Halt system.
1E	Both 8259 interrupt controllers passed the tests
20	Set up interrupt vectors 02-1F
21	Set up interrupt vectors 70-77
22	Clear interrupt vectors for 41 and 46 (disk parameter pointers).
23	Read 8042 self-test result, DMA page reg ch 2 (port 81).
24	Test for proper 8042 self-test result (55).
25	Error: Keyboard controller self-test failed, display message and halt.
26	Keyboard controller self-test passed
27	Confirm DMA working; prepare DMA channel 2 for floppy data transfer
28	Reinitialize video (cold boot)
29	Reinitialize video with cursor off (warm boot)
2A	Video parameters are initialized
2B	Enable NMI and I/O channel check, disable 8254 timer channel 2 and speaker
2C	Run RAM test to determine size of RAM
2D	RAM sizing complete
2E	Send reset command to keyboard controller to initiate a keyboard scan cycle
2F	Keyboard has been initialized. Initialize the CMOS RTC
30	CMOS RTC has been initialized. Initialize on-board floppy if any
31	Install the hard disk controller
32	Disk controller has been installed; prepare DMA channel 2 for floppy transfers
33	Perform equipment check and initialize numeric data processor (math chip)
34	Install the serial/parallel ports
35	Test CMOS RAM battery level
36	Check for keypress-Esc=Setup, Spacebar=menu; do speed beeps 2=high, l=low
37	Enable 8254 timer channel 0 for system tick, enable keyboard and slave interrupt controller 8259 #2
38	Timer tick, keyboard and 8259 #2 enabled; enable/disable cache per CMOS RAM
39	Enable keyboard interface and interrupts. Go to Setup as necessary; shadow ROMs as appropriate.
3A	Setup finished, so clear the screen and display Please Wait message
3B	Test the fixed and floppy drives
3C	Scan for and invoke the adapter ROMs in C800-E000
3D	Turn off Gate A20; restore vectors 3bh-3fh with temporary interrupt service routines.
3E	Gate A20 is turned off
3F	Invoke INT19 to boot operating system.

These accompanied by 5 long beeps:

Code	Meaning
BF	486-based, 386SX/20c or 386SX/25c processor module boards are used in a system where the WD76C10 chipset is not revision F or above.
CF	CPU on a 486-based processor module has failed its internal self-test.
DF	386SX/20c or 386SX/25c module board failed correctly to initialize its on-board cache (bad cache RAM. illegal configuration, etc., or unknown module ID).
EF	Extended CMOS RAM within the WD76C10 chipset failed its self-test

486 EISA-10 Oct 1989

Code	Meaning
	Power on or system reset: enable 8042, RTC; disable 82C601 chip serial, parallel, floppy, hard drive, NMI; check 8042 status.
AA, 01, xx	Show 80486 BIST (built-in self-test) result: xx=00 if OK, FF if not.
01	Disable cache, enable ROM, high speed on, turn off caches, disable EISA NMIs, set master and slave IRQs to edge-triggered, disable reset chaining. disable 82C601 chip but set it valid.
05	Initialize address decoder, 640K RAM; set BIOS as cacheable, enable extended memory.
06	Clear Shutdown Flag.
07	8042 and keyboard test: wait till 8042 buffer empty, disable 8042 command, read 8042 output buffer, set response OK to DMA page reg channel 2.
08	Send 8042 NOP command, self-test command; get 8042 self-test result, send to DMA page reg channel 2.
AA, 01, xx	Show 8042 self-test result: xx=55 if OK
09	Test BIOS ROM checksum; 3 short beeps and halt if bad
0A	Read CMOS registers 3 times to clear pending RTC interrupts, and disable them. Check battery.
0B	Bad CMOS RAM battery.
0C	Send command to port 61 to disable parity and speaker, enable timer; disable DMA.
0D	Test 8254 counter timer: set all 3 counters to mode 3 (square wave), start them and read the counts.
0E	A counter timer is bad (at least one is 0 and not counting).
AA, 01, xx	Show failing counter address (xx = 40, 41, or 42), then beep L-S-L-S and halt.
0F	Enable and check memory refresh (set timer 1 to mode 2 for 15 microsecond refresh, and turn on DMA to perform it); delay 1 ms and check bit 4 of port 61 for 0-to-1 toggle.
10	Memory refresh failed (no toggle); beep short-long-short, and halt.
11	Check and clear the first 64K of RAM in real mode: disable NMI, clear parity latches, fill 64K with 5555 and check it, then AAAA and check it, then 0000.
AA, 06, mmnn, oopp, qqrr	First 64K memory test failed. mmnn=location lsb, msb; oopp= value read lsb, msb; qqrr=value expected lsb, msb.
AA, 01, xx	Test port 61 for parity error (bits 7, 6=1) and display xx=value read from port 61 if parity error occurred.
12	First 64K memory test failed. Clear parity latches, give 3 long beeps, and halt.
13	First 64K memory test passed.
14	Reset warm boot flag (40:72) and test CMOS. Turn off caches, shadow BIOS, set speed high, calculate high speed and initialize GP flag, set speed low and turn off cache if CMOS not good or speed not high, otherwise turn on cache and set speed high.
16	Check Shutdown Flag 123x.
17	Reset was cold boot. Set 40:e9 bit 7 (disk_status).
18	Prepare 8259 interrupt controllers; send FF to mask register and check.
19	Interrupt controller initialization failed; initialize video, display the error message, and halt.
1A	Test interrupt controller: set all 256 ints to slipped interrupt vector. If warm boot (40:e9 bit 7), skip to 1E.
1B	Set int 0F to spurious interrupt vector, check for spurious interrupts.

Code	Meaning
1C	Set int 08 (timer 0) to timer 0 int vector, enable timer and int, wait for int from timer.
1D	Timer interrupt did not occur. Init video, display error message and halt.
1E	Initialize interrupt vectors.
1F	Initialize interrupt vectors 00-6F to temporary interrupt service routine.
20	Set vectors for interrupt 02-1F.
21	Set interrupt vectors for 70-77, clear vectors 60-67 and 78-FF.
22	Clear interrupt vectors for 41 and 46 (disk parameter pointers).
23	Read 8042 self-test result from DMA page reg ch 2 (port 81).
24	Test for proper 8042 self-test result (55).
25	8042 self-test failed. Get keyboard controller status, init video, display error msg, and halt.
26	Initialize 8042 keyboard controller, transfer 128K mem. exp. bit from 8042 to CMOS RAM (IBM compatible, but not used), read state of security switch and initialize RAM variable.
27	Check Shutdown Flag = 123x. No= cold boot.
28	If cold boot or CMOS RAM is bad, install video ROM and establish video, initialize equipment flags according to primary video adapter and CMOS RAM content, initialize POST status, initialize video.
29	If not cold boot and CMOS RAM is OK, install Video ROM and establish video for mono/CGA, initialize equipment flags according to primary video adapter and CMOS RAM contents, initialize video warm boot, initialize video.
2A	Check for bad CMOS RAM and queue the message if so; command port 61 to clear parity latches, disable the speaker and disable timer channel 2; enable NMI.
2B	Check Shutdown Flag = 123x. if warm boot, use memory sizes from CMOS RAM.
2C	If cold boot, turn caches off, test memory for appropriate size, and restore cache status.
2D	Turn off POST Fail CMOS RAM bit and display any queued error messages; initialize keyboard RAM (40:17-30) + (40:E0-E7).
2E	Initialize 8042 keyboard controller and test keyboard.
2F	Initialize time of day in the real time clock chip.
30	Test for and install floppy controller.
31	Enable C&T 82C601 chip IDE interface, test for and install hard drive.
32	Test 8259 DMA registers with 55 then AA, and initialize them to 0 (ports D2 and D4).
33	Test for and initialize math coprocessor chip
34	Test for and initialize parallel and serial ports, on and off board.
35	Initialize RAM variables for bad CMOS time, date, checksum, and battery condition.
36	Wait for user to press Esc, space. Check keyboard lock, clear the keyboard lock override, beep to indicate speed, display any queued messages. Esc=setup, space=boot menu.
37	Enable system clock tick (IRQ0), keyboard (IRQ1), and slave interrupt controller (IRQ2)
38	Initialize RAM variables for Ctrl-Alt-Esc, Ctrl-Alt-Ins
39	Enter setup if user pressed Ctrl-Alt-Esc. If EISA, revert to ISA if tab key pressed.
3A	Clear screen and update equipment flags according to CMOS contents (may have changed during setup). Shadow any ROMs per setup. Enable/disable cache per CMOS RAM.
3B	Initialize floppy and fixed disk drives.
3C	Set POST Fail bit in CMOS RAM, then scan for and invoke adapter option ROMs.
3D	Clear the Shutdown Flag to 0, turn off gate A20 to enable memory wrap in real mode.
3E	Set vectors for interrupts 3B-3F, clear Post Fail bit in CMOS RAM, home the cursor, display any error messages, clear MSW of 32-bit registers (ISC Unix).
3F	Invoke INT 19 to boot operating system.

TANDY

Uses OEM version of Phoenix BIOS.

WYSE

Uses OEM version of Phoenix BIOS.

ZENITH

LEDs on system board indicate status of various stages of boot-up. All will light up first, then go out in sequence when the test is completed. May also use an AMI (Plus, normally) or a Phoenix BIOS.

Post Procedures

Procedure	Meaning
CPU	Perform read/write test on internal register. Check for defective CPU or clock generator.
ROM BIOS	Check ROM CRC value against computed value of this test. Check BIOS & I/O circuitry.
RAM	Check first 64K of memory to see if data can be stored in it so the BIOS can use it later.
DMA	Test the register functions of the DMA chips.
PIT/PIC	Perform tests on main support chips and enable appropriate interrupts. Check AC ripple.
RTC/CMOS	Check the validity of the CMOS RAM and compare value in CMOS with appropriate devices. BIOS will use the values from the CMOS to set up appropriate IRQ routines for disk and other I/O access. Check for defective CMOS/battery/adapter or CMOS setting.
Video Display	Attempts will be made to initialise video to a mono screen early on so error messages can be displayed. This test is for initialising upper video modes available with EGA/VGA.
Test/Boot to Diskette	Check floppy subsystem and prepare drive for boot if there is a bootable floppy in A:.
Boot to Fixed Disk	Initialise any fixed disks in the CMOS and give control to the first one if a bootable floppy has not been detected previously. Check for corrupt boot code if not a hardware error.

POST Codes

Code	Meaning
01	VGA check
02	MDA initialise
03	Initialise video
05	Set hard reset
07	Check ROM at E000
08	Check ROM shadow at F000
09	Remap video to E000
0B	Keyboard controller test
0C	CMOS/8042 test
0D	DMA test
0E	DMA page register
1D	Testing system board
1E	Testing system board
1F	Bus sizing
20	Set BIOS data area
21	Testing DMA
22	Checking C800 for ROM
24	Testing base memory
25	8042 test
26	8042 test
27	8042 test
28	Memory parity test

Code	Meaning	Code	Meaning
0F	Test 64K memory	29	PIT test
10	Test base memory	2A	Testing floppy disk
11	Second VGA unit	2B	Testing FDC/drives
12	Mono initialisation	2C	Testing HDC/drives
13	RTC/CMOS test	2D	Checking CMOS settings
15	CPU register test	2E	Soft configuration
16	CPU add test	30	Checking adapter ROM
17	RTC/8042 test	31	Checking CMOS settings
18	Enter protected mode	32	Enabling interrupts
19	Testing memory	33	Soft configuration
1A	Testing extended memory	34	Soft configuration
1B	Leaving protected mode	35	Jump to boot code
1C	Testing system board	00	Boot to OS

Orion 4.1E-1992

00h-1Fh and F0h-FFh are displayed after the indicated function is completed.

Code	Meaning
02	Cold Boot, Enter Protected Mode
03	Do Machine Specific Initialization
F0	Start of Basic HW Initialization for Boot
F1	Clear CMOS Pre-Slush Status Location
F2	Starting CLIO Initialization
F3	Initialize SYSCFG Register
F4	DXPI Initialization for Boot Block
F5	Turning OFF Cache
F6	Configure CPU Socket Pins
F7	Checking for 387SX
F8	82C206 DEFAULT Initialization
F9	Superior Default Initialization
FF	End of Machine-specific Boot Block
04	Check Flash Checksum
05	Flash OK, jump into Flash (FFFD Flash Code
06	Reset or Power-Up
07	CLIO Default init command
08	SYSCFG REG initialised
09	CMOS Pre-slush error words initialisation
10	SCP initialised
11	DRAM autosizing complete
12	Parity check enabled. Enable Memory Parity (EMP) LED turned off
13	Start of slushware test
14	Slushware at 000F0000h OK
15	BIOS ROM copied to slushware
16	Back in Real Mode
17	ROM BIOS Slushing is finished. CPU LED Turned off
18	Video ROM (C0000 Slushware Test
19	Internal Video ROM Slushed
1A	Back in Real Mode

Code	Meaning
1B	Internal video hardware enabled.
1C	CPU clock frequency determined
1E	BIOS RAM cleared

20-EF displayed before function has been attempted. 20-2A indicate restart after shutdown, usually return to real mode from protected mode. CMOS RAM shutdown byte (0F) has value indicating reason.

Code	Meaning
20	RESET (CMOS 0)
21	Continue after Setting Memory Size (CMOS 0F=1)
22	Continue after Memory Test (CMOS 0F=2)
23	Continue after Memory Error (CMOS 0F=3)
24	Continue with Boot Loader Request (CMOS 0F=4)
25	Jump to execute User Code (flush) (CMOS 0F=5)
26	Continue after Protected Mode Test Passed (CMOS 0F=6)
27	Continue after Protected Mode Test Failed (CMOS 0F=7)
28	Continue after Extended Protected Mode Test (CMOS 0F=8)
29	Continue after Block Move (CMOS 0F=9)
2A	Jump to execute User Code (CMOS 0F=A)
2B	Reserved
2C	Reserved
2D	Reserved
2E	Reserved
2F	Reserved
30	Exit from Protected Mode
31	TEST-RESET passed (80386). Warm Boot
32	Check the ROM Checksum. ROM LED Turned Off
33	Clear the Video Screen On
34	Check System DRAM Config Update CMOS-TOTAL-MEM-SIZE Value
35	Pro-load CMOS if CMOS is
36	Turn Off the UMB RAM
37	Turn Parity Generation
38	Initialize System Variable
39	Check for errors in POWER
3A	Initialize SCP MODE
3B	Test CMOS Diag. Power Reset
3C	Test CPU Reset 80386 & Determine State Number
3D	Save CPU ID & Processor-T
3E	Init the Video & Timers
3F	Init DMA Ports, Clear Page
40	Set Speed too Fast for Now
41	Test EEPROM Checksum
42	Enable/Disable Superior's Parallel, FDC & HDC Per CMOS
43	Slush External Video BIOS if on CMOS
44	Turn Cache off for Memory
45	Test Extended RAM (1-16Mb)
46	Test BASE RAM (0-64 OK). RAM LED turned off by Base RAM Test
47	Determine Amount of System

Code	Meaning
48	Set WARM-BOOT Flag if RES Indicates Cold Boot
49	Clear 16K of Base RAM
4A	Install BIOS Interrupt Vector
4B	Test System Timer. INT LED turned off if CLOCK Test passes
4C	(Re)Initialize Interrupt
4D	Enable Default Hardware Initialization
4E	Determine Global I/O Configuration
4F	Initialize Video
50	Init WD90C30 Scratchpad
51	Check for Errors before Boot
52	Reserved
53	Test (Ext Only) and Initialize
54	Reserved
55	Initialize the Keyboard Processor
56	Initialize the PS/2 Mouse
57	Configure CLIO for Mouse
58	Configure CLIO for LAN
59	Configure CLIO for SCSI
5A	Configure CLIO for WAM
5B	Wait for User to Enter Code
5C	Init System Clock TOD, Enable
5D	Test, Init Floppy Drive Sensor. Disk LED Turned off
5E	Check for Z150 Style Disk
5F	Init Winchester Subsystem
60	Set Default I/O Device Parameters
61	Get LAN ID Info from LAN
62	*Install ROMs at 0C8000h
63	*Install ROMs at 0E000h
64	Initialize SCSI Interface
65	Run with A2O off in PC Mode
66	Really turn off the SCP
67	Set Machine Speed using CMOS
68	Turn on Cache
69	Calibrate 1ms Constants
6A	*Enable Non-Maskable Interpreter
6B	Reserved
6C	Clear the warm-boot flag
6D	Check for Errors before Boot
6E	Boot

191 BIOS -1992

Code	Meaning	Code	Meaning
0	Start of Slush Test	34	Initialize System Variables
1	Processor Test	35	Init Interrupt Controllers
2	CACHE and CLIO	36	Check Error that Occurred
3	ISP Defaults Set	37	Reinitialize SCP Warm Boot
4	Into Protected Mode	38	Test CMOS Diag, Power, Reset
5	Memory SIMMs Count	39	Reserved, or DDNIL status flag check
6	Memory Controller	3A	Test CPU Reset (80386)
7	Preped to Test Block	3B	Save the CPU ID in GS
8	First 1Mb of Ram	3C	Slush Video ROM to C0000
9	Checksum OEM ROM	3D	Init the Video and Timers
10	Low Flash ROM Checks	3E	Init CMA Ports, Clear Page
11	F000 ROM Checks	3F	Set Speed too Fast for now
12	Aurora VIDEO ROM	40	Checksum the Nonvolatile RAM
13	F000 ROM Slushed	41	Initialize Configuration
14	Sep Initialized	42	Init Expansion Boards from VRAM
15	Language Slushed	43	Turn Cache off for Memory Test
16	Do VIDEO Specific tests	44	Init Memory Ctrlr, test Extd Memory
17	Done Slushing	45	Test Base RAM
32	Point Interrupt Vectors	46	Determine amount of System RAM
33	Turn on Parity Generation		

INDEX

D

E

M

N

O

Q