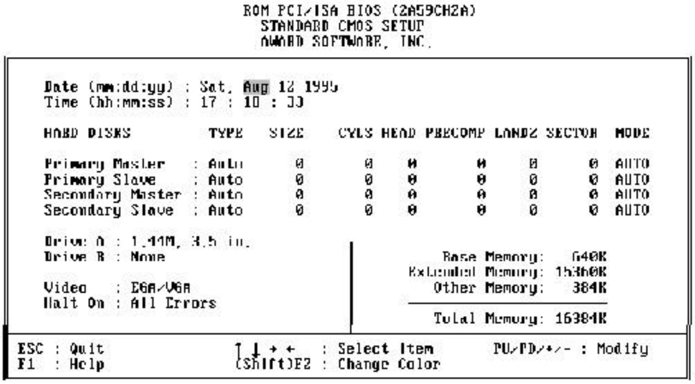
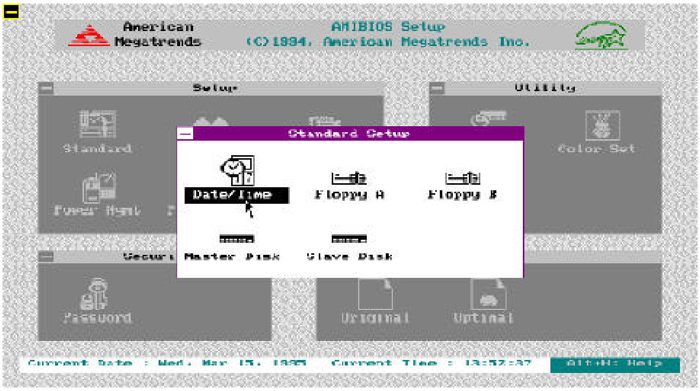
Standard CMOS Setup

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.



```
                    ROM PCI/ISA BIOS (2A59CH2A)
                      STANDARD CMOS SETUP
                      AWARD SOFTWARE, INC.

   Date (mm:dd:yy) : Sat, Aug 12 1995
   Time (hh:mm:ss) : 17 : 10 : 33

   HARD DISKS          TYPE   SIZE    CYLS HEAD PRECOMP LANDZ SECTOR  MODE

   Primary Master    : Auto     0        0    0       0     0      0  AUTO
   Primary Slave     : Auto     0        0    0       0     0      0  AUTO
   Secondary Master  : Auto     0        0    0       0     0      0  AUTO
   Secondary Slave   : Auto     0        0    0       0     0      0  AUTO

   Drive A : 1.44M, 3.5 in.
   Drive B : None                           |       Base Memory:    640K
                                            |   Extended Memory:  15360K
   Video   : EGA/VGA                        |      Other Memory:    384K
   Halt On : All Errors                     |  -----------------------
                                            |      Total Memory:  16384K

 ESC : Quit                 ↑ ↓ → ←  : Select Item      PU/PD/+/- : Modify
 F1  : Help               (Shift)F2  : Change Color
```



Current Date : Wed, Mar 15, 1995   Current Time : 13:52:37

Date and Time

Speak for themselves, really, except the timekeeping won't be wonderful, due to variations in voltages, etc (see *The Year 2000 Problem*).

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 would be forgiven for using 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.

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 (see *Known BIOS Problems*). 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. 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. You can use this capacity to increase performance of QIC80 or Travan tape drives on the floppy cable. They are known as *Extra Density* drives. Microsoft has yet another format which stores 1.7 Mb on a floppy, called *Distribution Media Format*, or DMF. Neither are supported by DOS.

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:

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.

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.

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.

Quick Boot

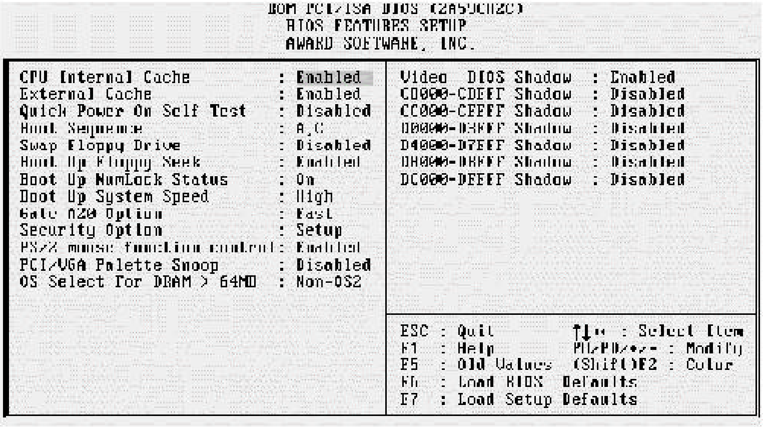
Enabled allows the system to boot within 5 seconds, but it skips just about everything.

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).

Advanced CMOS Setup

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



```
              ROM PCI/ISA BIOS (2A59CH2C)
                 BIOS FEATURES SETUP
                 AWARD SOFTWARE, INC.

CPU Internal Cache          : Enabled    Video  BIOS Shadow  : Enabled
External Cache              : Enabled    C8000-CBFFF Shadow  : Disabled
Quick Power On Self Test    : Disabled   CC000-CFFFF Shadow  : Disabled
Boot Sequence               : A,C        D0000-D3FFF Shadow  : Disabled
Swap Floppy Drive           : Disabled   D4000-D7FFF Shadow  : Disabled
Boot Up Floppy Seek         : Enabled    D8000-DBFFF Shadow  : Disabled
Boot Up NumLock Status      : On         DC000-DFFFF Shadow  : Disabled
Boot Up System Speed        : High
Gate A20 Option             : Fast
Security Option             : Setup
PS/2 mouse function control : Enabled
PCI/VGA Palette Snoop       : Disabled
OS Select For DRAM > 64MB   : Non-OS2

                                         ESC : Quit        ↑↓→← : Select Item
                                         F1  : Help        PU/PD/+/- : Modify
                                         F5  : Old Values  (Shift)F2 : Color
                                         F6  : Load BIOS  Defaults
                                         F7  : Load Setup Defaults
```

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 characters per second are generated (max 30 under DOS).

The **alt**, **shift**, **ctrl**, **numlock**, **caps lock** and **scroll lock** keys are excluded. Possibly disable for NetWare servers.

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 Priming

Found with the MR BIOS and similar to the above. The *Full Test* works at a rate of 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.

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 defective memory chips, but they could also be mismatched (in which case change the wait states), or the wrong ones for that 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 allows the correction of 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.

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 CMOS 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 start up tests. In other words, if disabled, the system does not halt after this message is displayed.

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.

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 one comes with the Award BIOS, 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.

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 searches all floppies (you may have more than 2) before defaulting to drive C:, which is 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.

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.

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, to cater for older peripherals. Normally, set High, but 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 is used to access memory above 1 Mb, which is usually handled through the keyboard controller chip (the 8042 or 8742).

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 contained 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 still 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, in which case 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* (overleaf).

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 you are using an add-on card that is 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.

CIH Buster Protection

Protects against viruses that try to destroy the BIOS. Disable before running anything like *Drive Image*.

Anti-Virus Protection

Protects against viruses that affect the boot sector and partition table (only). Disable before running anything like *Drive Image* or Windows. Doesn't work if your hard disk controller has its own BIOS (as with SCSI).

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*.

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

You get three attempts to get in, after which the system will have to be rebooted. The default is usually the manufacturer's initials (try **ami**), or **biostar, biosstar, AWARD?SW, AWARD?PW, LKWPETER, 589589, aLLy, condo**, **djonet**, **lkwpeter**, **j262 SWITCHES?SW, AWARD_SW**, or **Shift + S Y X Z** for Award (before 19 Dec 96), but if this doesn't work, or you forget your own password, you must discharge the CMOS. 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. Look for the chips mentioned below, under *Clearing Chips*.

You could try flooding the keyboard buffer to crash the password routine – just wait for the password prompt, then keep pressing **esc**.

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 so you can get it up to speed again. 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 your settings can remain even down to around 2.2v. Usually, the clock will stop first, as the oscillator needs a higher voltage to operate. 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 you can 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 will be denied without the correct password.

❑ *Setup*, where access to setup is denied without the password.

This can be disabled by selecting *Supervisor/User Password Setting* at the main menu and pressing **Enter** without entering anything below).

Supervisor/User Password

Gives two levels of security; *Supervisor* has higher priority, so the other doesn't work if it is enabled. To disable, press **Enter** without entering anything.

Network Password Checking

When set to 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 while trying to do something legitimate. Actually, it's useless for those drives that have their own BIOS in the controller (ESDI/SCSI). Disable when using Multiuser DOS, or installing software. Only available for operating systems such as DOS that do not trap INT 13.

Virus Warning

See *Boot Sector Virus Protection* (Award).

ChipAway Virus On Guard

See above. Guards against boot virus threats early in the boot cycle, before they have a chance to load.

Report no FDD for Win 95

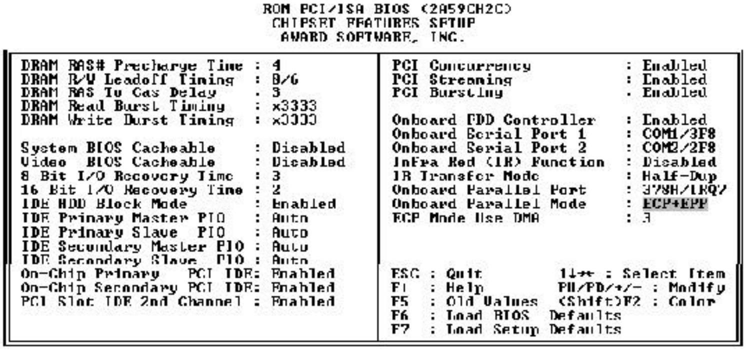
Set to *Yes* if using Windows 95/98 without a floppy to release IRQ6 (this is required to pass Windows 95/98's SCT test and get the logo). Also disable the Onboard FDC Controller in the Integrated Peripherals screen.

Turbo Frequency

Boosts your CPU speed by mildly overclocking your CPU (2-5%).

Advanced Chipset

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!



```
                    ROM PCI/ISA BIOS (2A59CH2C)
                     CHIPSET FEATURES SETUP
                      AWARD SOFTWARE, INC.

DRAM RAS# Precharge Time : 4          PCI Concurrency           : Enabled
DRAM R/W Leadoff Timing  : 8/6        PCI Streaming             : Enabled
DRAM RAS To Cas Delay    : 3          PCI Bursting              : Enabled
DRAM Read Burst Timing   : x3333
DRAM Write Burst Timing  : x3333      Onboard FDD Controller    : Enabled
                                      Onboard Serial Port 1     : COM1/3F8
System BIOS Cacheable    : Disabled   Onboard Serial Port 2     : COM2/2F8
Video  BIOS Cacheable    : Disabled   InFra Red (IR) Function   : Disabled
8 Bit I/O Recovery Time  : 3          IR Transfer Mode          : Half-Dup
16 Bit I/O Recovery Time : 2          Onboard Parallel Port     : 378H/IRQ7
IDE HDD Block Mode       : Enabled    Onboard Parallel Mode     : ECP+EPP
IDE Primary Master PIO   : Auto       ECP Mode Use DMA          : 3
IDE Primary Slave   PIO  : Auto
IDE Secondary Master PIO : Auto
IDE Secondary Slave  PIO : Auto
On-Chip Primary   PCI IDE: Enabled    ESC : Quit        ↑↓→← : Select Item
On-Chip Secondary PCI IDE: Enabled    F1  : Help        PU/PD/+/- : Modify
PCI Slot IDE 2nd Channel : Enabled    F5  : Old Values  (Shift)F2 : Color
                                      F6  : Load BIOS  Defaults
                                      F7  : Load Setup Defaults
```

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 downloaded from the MCCS BBS in Singen/Germany, on (49) 7731 69523 (use GAST as a username). Try also **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 also that 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 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 number of rows (typically 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 need to 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.

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 would like 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 setting 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 data 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, limiting the current and helping 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.

Act Bank A to B CMD Delay

This sets the delay between Active commands to different memory banks. The shorter the interval, the better the performance, at the expense of stability.

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 the write cycles on the bus, you can 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.

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 also 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.

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

Synchronous AT Clock

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

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 Plug and Play 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

The 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 appears separately 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 devices, such as the AHA 1542B, 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* specified 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 to do it, speeding up the transition 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 indicate the start of a CPU bus cycle, indicating that a valid command and address is stable on the bus. When enabled, more time will be 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. 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 now abandoned in favour of 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 that can be 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 to use 9, but many don’t, so you might save yourself 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.

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.

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)

Concurrent Mode

Allows DMA access for floppies and tapes, as QIC and other systems commonly share controllers with floppy disks. However, many computers will not support this.

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.

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 the 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* section.

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 with regard to read cycles.

Async SRAM Write WS

Choose the timing combination for your motherboard and memory with regard to 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 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, 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.

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.

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 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 above, but for writes. May be selected by the board designer.

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.

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 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 if you have 64 or 256K of cache memory installed; 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 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. 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 - you just won't get the benefits. It may be useful for 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.

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 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 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 that determines how much system memory is cacheable, since it can only store a certain amount of addresses.

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.

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, corresponding 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

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.

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 setting 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

Whether your computer wakes up before you do! Seriously, 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.

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 setting chooses which pin the dirty data is tied to.

I/O	means Bi-directional input/output (default)
IN	means Input only

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.

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 motherboard, 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).

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 combination of RAS and CAS therefore specifies a particular RAM location in a particular RAM chip, where they intersect. Unfortunately, as can be seen with the above, a lot of time is taken up with 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 time 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 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 DRAM 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 that adjusting one setting may change several parameters in the background.

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.

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.

SDRAM TRP SRAS Precharge

The delay from the precharge command to the bank activate command.

SDRAM TRAS Timing

The minimum bank active time.

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 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 TRCD

The cycles from a bank activate command until the acceptance of a read or write command.

DRAM (Read/Write) Wait States

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:

$$\frac{\text{Wait States} = \text{ns} + 10 \text{ x Clock Speed}}{1000 -2}$$

So:

$$\frac{.97 = 80 + 10 \text{ x } 33}{1000 -2}$$

gives you (almost) 1 wait state for 80 ns RAM at 33 MHz. For machines with 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 Read/Write Timing

See above.

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.

Add Extra Wait for RAS#

Same as above.

Add Extra Wait for CAS#

Same as above.

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).

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.

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 Write Burst Timing

See *FP Mode DRAM Read WS*.

DRAM Timing Option

See *DRAM Speed*.

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 Post Write

An Award setting. Still working on it, but see *Posted Write Enable*.

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 Burst (B/E/P)

See *DRAM Read Burst (B/E/P)*, above.

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 Last Write to CAS#

Sets how much time (or how many cycles) will elapse between the last data signalled to when CAS# is asserted. This time is used as setup time for the CAS signal. Choices are 2 (default), 3 or 4.

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 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 your code is not sequential, you may be better off without this enabled.

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 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 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 to insert 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 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 Precharge Wait State

Use 0 for 60-70 ns and 1 for 70 ns DRAM.

DRAM Wait State.

Same as above.

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***.**

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.

CAS Low Time for Write/Read

The number of clock cycles CAS is pulled low for memory operations, very dependent on memory timing.

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 its charge before EDO DRAM is allowed to 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.

EDO Back-to-Back Timing

The number of timer ticks needed for back-to-back accesses, depending on your memory. (SiS).

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*.

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	will 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	read and write DRAM timings are X-3-3-3

Try the lowest figures first, until your machine is running successfully.

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.

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 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.

RAS Precharge @Access End

When enabled, RAS# remains asserted at the end of access ownership. Otherwise, it is deasserted.

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

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 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# 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. The powersaving occurs because an internal counter is used, not an external address, and 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 the 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.

The 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.

Extended Read Around Write

When Enabled, reads can bypass writes within the 82450GX memory interface component(s), provided their addresses do not match.

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. If it knows the next data 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.

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, being the number of 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 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.

Enhanced Page Mode

Enable or Disable, according to your memory.

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 (refers to cacheing). Choices are:

Disabled	(E segment default)
Enabled	(F segment default)
Cached	L2 cache?
Into-486	L1 cache

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).

Disable Shadow Memory Size

Sets a shadow memory size for *Disable Shadow Memory Base*, above. It doesn't actually disable anything.

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*.

DRAM Relocate (2, 4 & 8 M)

Remaps 256K of upper memory to the top of DRAM size. Only applicable when the D and E segments are not shadowed, and with 2, 4 or 8 Mb of on-board memory.

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)—default ***EMS 1*** (218, 219, 21A)

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

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.

Extended Memory Boundary

Where extended memory ends, and expanded memory begins. Possibly for expanded memory cards.

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.

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. Set disabled 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 applies only to EDO memory and only needs to be changed if you get memory addressing errors.

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*.

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.

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 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 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 cache 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 be filled up (there's a lot of data). In addition, 3D operations need to use the FPU (maths copro), which can 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 also above. If disabled, all 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, which reduces 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 possible 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 the chips at higher than rated speeds will mean dropping a CAS/RAS level.

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.

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.....

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.

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.

RAS Pulse Width

The number of 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. *Enabled* allows the SDRAM signal controller to run at 1T (that is 1 clock cycle). Otherwise, it runs slower, at 2T.

SDRAM Burst X-1-1-1-1-1-1-1

Allows burst mode. Enabled is best for performance.

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 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, having 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.

DRAM Clock

Allows the DRAM to work concurrently with the host bust clock. If you disable this, it will align itself to the AGP Clock.

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.

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 pages, 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 being accessed.

Fast Command

Controls the internal timing of the CPU – *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, whatever that is. Don't change unless you change DRAM or CPU. MA means *Memory Access*. Low is best for performance.

Fast RAS to CAS Delay

Determines the timing of the 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 in the relevant banks.

386 DRAM Quick Write Mode

As above.

DRAM Page Idle Timer

The time in HCLKs that 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 if there are 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 DRAM 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.

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, if any (zero wait states).

PCI-to-DRAM Prefetch

Allows the prefetching of large parts of memory, assuming coherent data, so the contents can be accessed with very low latency and 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.

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

CPU Low Speed Clock

Or *Low Speed CPU Clock Select* selects whatever speed you want to use as the slow speed when you select Turbo Off on the front panel 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.

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*.

Check ELBA# Pin

Sets when the ELBA# pin is checked, during T1 or T2. Should mostly be set to T2, that is, later in the cycle for better reliability, but this can depend on other settings. The *External Local Bus Access#* 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!

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 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.

Appian Controller

An advanced IDE controller. You also need special software to activate it.

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 set *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 testing only. Your machine will hang without the right equipment attached to the board and you will need to discharge the CMOS (see *Password*). 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. Should be 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 board, similar to the above. Disabling appears to improve performance slightly.

NA# Enable

Allows pipelining, where the chipset signals the CPU for a new memory address before all data transfers for the current cycle are complete, resulting in 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 chipset.

Host Bus Slave Device

This allows you to use an Intel 486 Host Bus Slave (e.g. a graphics device).

Polling Clock Setting

The rate the system polls all sub-systems (buses, memory, etc.) for service requests. Choices are:

```
14.318 MHz
CLK2 (Default)
CLK2/2
CLK2/3
CLK2/4
28.636 MHz
```

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.

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. Normally, disable.

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 true compatibles) support a streaming mode where larger amounts of data are moved to/from memory during a single 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.

Set Mouse Lock

You can lock the PS/2 Mouse as a security precaution.

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, but worse for stability.

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. 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.

POST Testing

Found on AST machines, determines whether POST testing will be *normal*, or *in-depth*. Normal just checks the memory.

MPS 1.1 Mode

The version of the multiprocessor specification.

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 that must be processed in sequence, as the Pentium Pro (or above) is able to execute out-of-order for smoother processing. Can be set to 1 or 8, meaning you can track up to 8 pipelined bus transactions.

Large Disk Access Mode

Choices are *DOS*, or *Other*. This was found on a Packard Bell with A Phoenix BIOS. Select the appropriate operating system.

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, like the Matrox Mystique, 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.

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. Some utilities can check a drive – Micro House EZ-S.M.A.R.T. and Symantec S.M.A.R.T. Doctor.

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. If you get problems with Win 98, check out article Q199886 in the MSN.

Spread Spectrum Modulated

There are techniques (developed by the US government, amongst others) for collecting intelligence from PC transmissions, as microprocessors (and screens) can radiate for some distance—you can expect to receive a PC's signals for up to ½ mile, and a mainframe's for anywhere between 3-4 (scan the area between 2-12 MHz).

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, as mentioned above, 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.

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.

You may get a *Smart Clock* option, which 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.

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.

Audio DMA

Selects a DMA Channel for motherboard sound systems.

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.

Language

Sets the language on BIOS setup screens and error messages. Has no affect on the language used by the Operating System or applications.

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.

PCI Primary IDE INT# Line

Assigns an interrupt line to an add-on PCI primary IDE controller.

PCI Secondary IDE INT# Line

See *PCI Primary IDE INT# Line (above).*

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.

Power-Supply Type

AT or ATX. It seems a bit late to set this after the machine has started, but it really concerns enabling sof-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.

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.

IN0-IN6(V)

The current voltage of up to seven 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.

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 TimerI

Specifies the number of clocks that the DRAM controller will remain in the IDLE state before precharging all pages.

Starting Point of Paging

Specifies the number of clocks required for starting of page miss cycles. Or controls the 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.

Turbo External Clock

Disable for AMD CPUs.

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.

DREQ6 PIN as

Invokes a software suspend routine by toggling the DREQ6 signal. Select *Suspend SW* only if your board has such a feature.

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.

Linear Merge

When enabled, only consecutive linear addresses can be merged.

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 this 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, so increases 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.

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.

CPU Ratio/Vcore (V)

Two items that adjust the CPU clock multiplier and core voltage, for overclocking.

DDR Voltage

Adjusts the DDR voltage to increase the DDR rate. Naturally, the recommendation is not to use it long term.

Notes