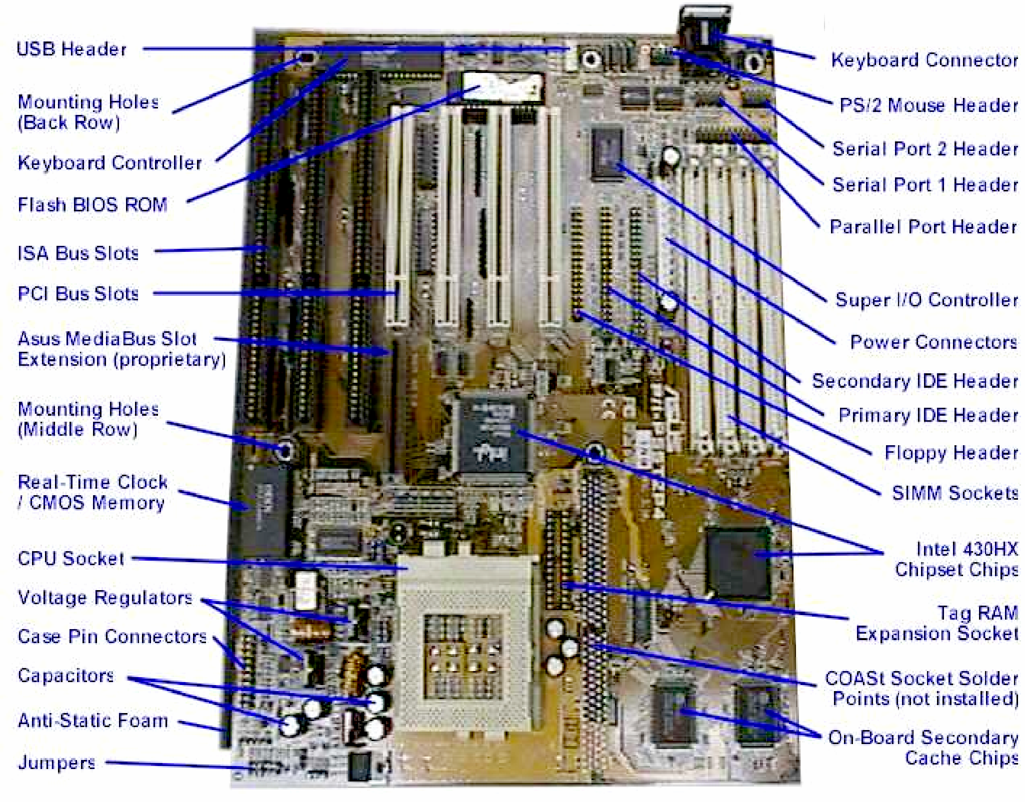
THE MOTHERBOARD

As can be seen from the plan of the computer system, the mother, or main board is at the center of the PC computer system. Effectively it is a printed circuit board containing the central processing unit (CPU) and the memory modules (SIMM's). It allows the CPU to interfaces with other parts of the computer via a 'BUS' system, into which sockets are fitted, for connection of various 'expansion' boards. Also on the motherboard is the RAM memory, normally in the form of SIMM, or DIMM modules in the modern PC computer, and cache memory in the form of integrated circuits (chips). All these various components will be examined in more detail in their appropriate sections.



Modern boards utilize a CPU with four PCI and three, or four, ISA expansion slots. Most video boards now use the PCI connection. So, if you are upgrading from a 386, or 486, motherboard that have a video board with VESA connectors, you may well have to purchase a new video panel.

Most Pentium boards are also fitted with controllers for Floppy Drives, 4 EIDE devices and Enhanced Parallel and Fast Serial Ports. This 'free's up' a couple of expansion slots for other devices

Motherboard and System Devices

The motherboard is, in many ways, the most important component in your computer (not the processor, even though the processor gets much more attention.) As mentioned earlier, if the processor is the brain of the computer, then the motherboard and its major components (the chipset, BIOS, cache, etc.) are the major systems that this brain uses to control the rest of the computer. Having a good understanding of how the motherboard and its contained subsystems works is probably the most critical part of getting a good understanding of how PCs work in general.

The motherboard plays an important role in the following important aspects of your computer system:

- Organization: In one way or another, everything is eventually connected to the motherboard. The way that the motherboard is designed and laid out dictates how the entire computer is going to be organized.
- Control: The motherboard contains the chipset and BIOS program, which between them control most of the data flow within the computer.
- Communication: Almost all communication between the PC and its peripherals, other PCs, and you, the user, goes through the motherboard.
- Processor Support: The motherboard dictates directly your choice of processor for use in the system.
- Peripheral Support: The motherboard determines, in large part, what types of peripherals you can use in your PC. For example, the type of video card your system will use (ISA, VLB, PCI) is dependent on what system buses your motherboard uses.
- Performance: The motherboard is a major determining factor in your system's performance, for two main reasons. First and foremost, the motherboard determines what types of processors, memory, system buses, and hard disk interface speed your system can have, and these components dictate directly your system's performance. Second, the quality of the motherboard circuitry and chipset themselves have an impact on performance.
- Upgradability: The capabilities of your motherboard dictate to what extent you will be able to upgrade your machine. For example, there are some motherboards that will accept regular Pentiums of up to 133 MHz speed only, while others will go to 200 MHz. Obviously, the second one will give you more room to upgrade if you are starting with a P133.

Motherboard Form Factors

The form factor of the motherboard describes its general shape, what sorts of cases and power supplies it can use, and its physical organization. For example, a company can make two motherboards that have basically the same functionality but that use a different form factor, and the only real differences will be the physical layout of the board, the position of the components, etc.

Form Factors Type:-

1. AT &BABY AT
2. ATX & Mini ATX
3. NLX & LPX

AT and Baby AT

Up until recently, the AT and baby AT form factors were the most common form factor in the motherboard world. These two variants differ primarily in width: the older full AT board is 12" wide. This means it won't typically fit into the commonly used "mini" desktop or minitower cases. There are very few new motherboards on the market that use the full AT size. It is fairly common in older machines, 386 class or earlier. One of the major problems with the width of this board (aside from limiting its use in smaller cases) is that a good percentage of the board "overlaps" with the drive bays. This makes installation, troubleshooting and upgrading more difficult.

The Baby AT motherboard was, through 1997, the most common form factor on the market. After three years and a heavy marketing push from Intel, the ATX form factor is now finally overtaking the AT form factor and from here out will be the most popular form factor for new systems. AT and Baby AT are not going anywhere, however, because there are currently just so many baby AT cases, power supplies and motherboards on the market.

A Baby AT motherboard is 8.5" wide and nominally 13" long. The reduced width means much less overlap in most cases with the drive bays, although there usually is still some overlap at the front of the case. One problem with baby AT boards is that many newer ones reduce cost by reducing the size of the board. While the width is quite standard, many newer motherboards are only 11" or even 10" long.

Baby AT motherboards are distinguished by their shape, and usually by the presence of a single, full-sized keyboard connector soldered onto the board. The serial and parallel port connectors are almost always attached using cables that go between the physical connectors mounted on the case, and pin "headers" located on the motherboard.

The AT and Baby AT form factors put the processor socket(s)/slot(s) and memory sockets at the front of the motherboard, and long expansion cards were designed to extend over them. When this form factor was designed, over ten years ago, this worked fine: processors and memory chips were small and put directly onto the motherboard, and clearance wasn't an issue. However, now we have memory in SIMM/DIMM sockets, not directly inserted onto the motherboard, and we have larger processors that need big heat sinks and fans mounted on them. Since the processor is still often in the same place, the result can be that the processor+heat sink+fan combination often blocks as many as three of the expansion slots on the motherboard! Most newer Baby AT style motherboards have moved the SIMM or DIMM sockets out of the way, but the processor remains a problem. ATX was designed in part to solve this issue.

ATX and Mini ATX

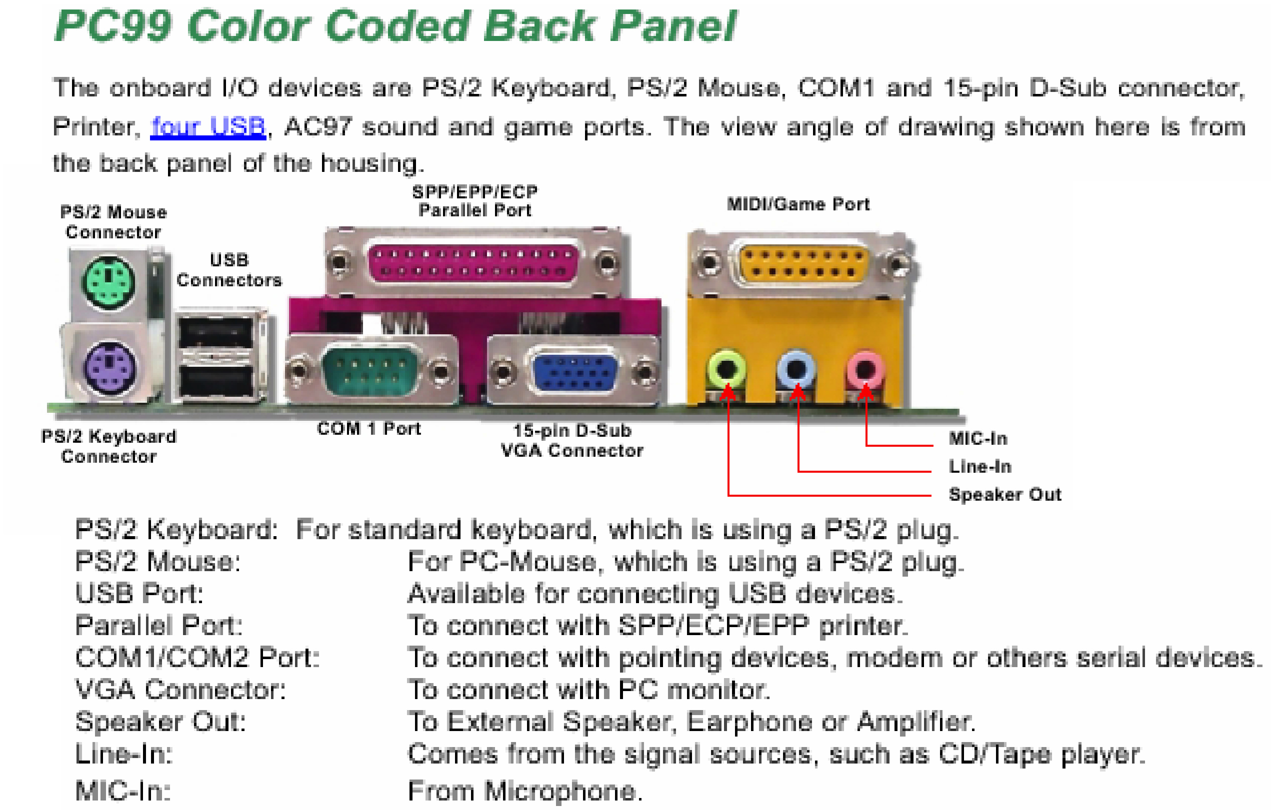
The first significant change in case and motherboard design in many years, the ATX form factor was invented by Intel in 1995. After three years, ATX is now finally overtaking AT as the default form factor choice for new systems (although AT remains popular for compatibility with older PCs, with homebuilders, and with some smaller PC shops). Newer Pentium Pro and Pentium II motherboards are the most common users of the ATX style motherboard (not surprisingly, since the Pentium II is the newest processor and uses the newest chipset families.) Intel makes the motherboards for many major name brands, and Intel only uses ATX.

The ATX design has several significant advantages over the older motherboard styles. It addresses many of the annoyances that system builders have had to put up with. As the Baby AT form factor has aged, it has increasingly grown unable to elegantly handle the new requirements of motherboard and chipset design. Since the ATX form factor specifies changes to not just the motherboard, but the case and power supply as well, all of the improvements are examined here:

- Integrated I/O Port Connectors: Baby AT motherboards use headers which stick up from the board, and a cable that goes from them to the physical serial and parallel port connectors mounted on to the case. The ATX has these connectors soldered directly onto the motherboard. This improvement reduces cost, saves installation time, improves reliability (since the ports can be tested before the motherboard is shipped) and makes the board more standardized.
- Integrated PS/2 Mouse Connector: On most retail baby AT style motherboards, there is either no PS/2 mouse port, or to get one you need to use a cable from the PS/2 header on the motherboard, just like the serial and parallel ports. (Of course most large OEMs have PS/2 ports built in to their machines, since their boards are custom built in large quantities). ATX motherboards have the PS/2 port built into the motherboard.
- Reduced Drive Bay Interference: Since the board is essentially "rotated" 90 degrees from the baby AT style, there is much less "overlap" between where the board is and where the drives are. This means easier access to the board, and fewer cooling problems.
- Reduced Expansion Card Interference: The processor socket/slot and memory sockets are moved from the front of the board to the back right side, near the power supply. This eliminates the clearance problem with baby AT style motherboards and allows full length cards to be used in most (if not all) of the system bus slots.
- Better Power Supply Connector: The ATX motherboard uses a single 20-pin connector instead of the confusing pair of near-identical 6-pin connectors on the baby AT form factor. You don't have the same risk of blowing up your motherboard by connecting the power cables backwards that most PC homebuilders are familiar with.
- "Soft Power" Support: The ATX power supply is turned on and off using signaling from the motherboard, not a physical toggle switch. This allows the PC to be turned on and off under software control, allowing much improved power management. For example, with an ATX system you can configure Windows 95 so that it will actually turn the PC off when you tell it to shut down.
- 3.3V Power Support: The ATX style motherboard has support for 3.3V power from the ATX power supply. This voltage (or lower) is used on almost all-newer processors, and this saves cost because the need for voltage regulation to go from 5V to 3.3V is removed.
- Better Air Flow: The ATX power supply is intended to blow air into the case instead of out of it. This means that air is pushed out of all the small cracks in the PC case instead of being drawn in through them, cutting down on dust accumulation. Further, since the processor socket or slot is on the motherboard

right next to the power supply, the power supply fan can be used to cool the processor's heat sink. In many cases, this eliminates the need to use (notoriously unreliable) CPU fans, though the ATX specification now allows for the fan to blow either into or out of the case.

- Improved Design for Upgradability: In part because it is the newest design, the ATX is the choice "for the future". More than that, its design makes upgrading easier because of more efficient access to the components on the motherboard.



PC99 Color Coded Back Panel

The onboard I/O devices are PS/2 Keyboard, PS/2 Mouse, COM1 and 15-pin D-Sub connector, Printer, four USB, AC97 sound and game ports. The view angle of drawing shown here is from the back panel of the housing.

PS/2 Keyboard:	For standard keyboard, which is using a PS/2 plug.
PS/2 Mouse:	For PC-Mouse, which is using a PS/2 plug.
USB Port:	Available for connecting USB devices.
Parallel Port:	To connect with SPP/ECP/EPP printer.
COM1/COM2 Port:	To connect with pointing devices, modem or others serial devices.
VGA Connector:	To connect with PC monitor.
Speaker Out:	To External Speaker, Earphone or Amplifier.
Line-In:	Comes from the signal sources, such as CD/Tape player.
MIC-In:	From Microphone.

Conventionally used in mass-produced "name brand" retail systems, the LPX motherboard form factor goes into the small Slimline or "low profile" cases typically found on these sorts of desktop systems. The primary design goal behind the LPX form factor is reducing space usage (and cost). This can be seen in its most distinguishing feature: the riser card that is used to hold expansion slots.

Instead of having the expansion cards go into system bus slots on the motherboard, like on the AT or ATX motherboards, LPX form factor motherboards put the system bus on a riser card that plugs into the motherboard. Then, the expansion cards plug into the riser card; usually, a maximum of just three. This means that the expansion cards are parallel to the plane of the motherboard. This allows the height of the case to be greatly reduced, since the height of the expansion cards is the main reason full-sized desktop cases are as tall as they are. The problem is that you are limited to only two or three expansion slots!

LPX form factor motherboards also often come with video display adapter cards built into the motherboard. If the card built in is of good quality, this can save the manufacturer money and provide the user with a good quality display. However, if the user wants to upgrade to a new video card, this can cause a problem unless the integrated video can be disabled. LPX motherboards also usually come with serial, parallel and mouse connectors attached to them, like ATX.

While the LPX form factor can be used by a manufacturer to save money and space in the construction of a custom product, these systems suffer from non-standardization, poor expandability, poor upgradability, poor cooling and difficulty of use for the do-it-yourselfer. They are not recommended for the homebuilder, but if you are upgrading one of these systems, you may not have many alternatives.

NLX

Much the way the AT form factor eventually became outdated and less suitable for use with the newest technologies, the LPX form factor has over time begun to show the same weaknesses. The need for a modern, small motherboard standard has lead to the development of the new NLX form factor. In many ways, NLX is to LPX what ATX is to AT: it is generally the same idea as LPX, but with improvements and updates to make it more appropriate for the latest PC technologies. Also like ATX, the NLX standard was developed by Intel Corporation and is being promoted by Intel. Intel of course is a major producer of large-volume motherboards for the big PC companies.

NLX still uses the same general design as LPX, with a smaller motherboard footprint and a riser card for expansion cards. To this basic idea, NLX makes the following main changes, most of which are familiar to those who have read about the enhancements introduced by ATX:

- Revised design to support larger memory modules and modern DIMM memory packaging.
- Support for the newest processor technologies, including the new Pentium II using SEC packaging.
- Support for AGP video cards.
- Better thermal characteristics, to support modern CPUs that run hotter than old ones.
- More optimal location of CPU on the board to allow easier access and better cooling.
- More flexibility in how the motherboard can be set up and configured.
- Enhanced design features, such as the ability to mount the motherboard so it can slide in or out of the system case easily.
- Cables, such as the floppy drive interface cable, now attach to the riser card instead of the motherboard itself, reducing cable length and clutter.
- Support for desktop and tower cases.

The NLX form factor is, like the LPX, designed primarily for commercial PC makers mass-producing machines for the retail market. Many of the changes made to it are based on improving flexibility to allow for various PC options and flavors, and to allow easier assembly and reduced cost. For homebuilders and small PC shops, the ATX form factor is the design of choice heading into the future.

Comparison of Form Factors

This table is a summary comparison of the sizes of the various motherboard form factors, and compatibility factors.

Style	Width	Depth	Where Found	Match to Case and Power Supply
Full AT	12"	11-13"	Very Old PCs	Full AT, Full Tower
Baby AT	8.5"	10-13"	Older PCs	All but Slimline, ATX
ATX	12"	9.6"	Newer PCs	ATX
Mini ATX	11.2"	8.2"	Newer PCs	ATX
LPX	9"	11-13"	Older Retail PCs	Slimline
Mini LPX	8-9"	10-11"	Older Retail PCs	Slimline
NLX	8-9"	10-13.6"	Newer Retail PCs	Slimline

Note: Some ATX cases will accept baby AT form factor motherboards.

BIOS

BIOS stands for Basic Input/Output System. The system BIOS is the lowest-level software in the computer; it acts as an interface between the hardware (especially the chipset and processor) and the operating system. The BIOS provides access to the system hardware and enables the creation of the higher-level operating systems (DOS, Windows 95, etc.) that you use to run your applications. The BIOS is also responsible for allowing you to control your computer's hardware settings, for booting up the machine when you turn on the power or hit the reset button, and various other system functions.

System Boot Sequence

The system BIOS is what starts the computer running when you turn it on. The following are the steps that a typical boot sequence involves. Of course this will vary by the manufacturer of your hardware, BIOS, etc., and especially by what peripherals you have in the PC. Here is what generally happens when you turn on your system power:

The internal power supply turns on and initializes. The power supply takes some time until it can generate reliable power for the rest of the computer, and having it turn on prematurely could potentially lead to damage. Therefore , the chipset will generate a reset signal to the processor (the same as if you held the reset button down for a while on your case) until it receives the Power Good signal from the power supply.

When the reset button is released, the processor will be ready to start executing. When the processor first starts up, it is suffering from amnesia; there is nothing at all in the memory to execute. Of course processor makers know this will happen, so they pre-program the processor to always look at the same place in the system BIOS ROM for the start of the BIOS boot program. This is normally location FFFF0h, right at the end of the system memory. They put it there so that the size of the ROM can be changed without creating compatibility problems. Since there are only 16 bytes left from there to the end of conventional memory, this location just contains a "jump" instruction telling the processor where to go to find the real BIOS startup program.

The BIOS performs the power-on self test (POST). If there are any fatal errors, the boot process stops.

The BIOS looks for the video card. In particular, it looks for the video card's built in BIOS program and runs it. This BIOS is normally found at location C000h in memory. The system BIOS executes the video card BIOS, which initializes the video card. Most modern cards will display information on the screen about the video card. (This is why on a modern PC you usually see something on the screen about the video card before you see the messages from the system BIOS itself).

The BIOS then looks for other devices' ROMs to see if any of them have BIOSes. Normally, the IDE/ATA hard disk BIOS will be found at C8000h and executed. If any other device BIOSes are found, they are executed as well. The BIOS displays its startup screen.

The BIOS does more tests on the system, including the memory count -up test, which you see on the screen. The BIOS will generally display a text error message on the screen if it encounters an error at this point.

The BIOS performs a "system inventory" of sorts, doing more tests to determine what sort of hardware is in the system. Modern BIOSes have many automatic settings and will determine memory timing (for example) based on what kind of memory it finds. Many BIOSes can also dynamically set hard drive parameters and access modes, and will determine these at roughly this time. Some will display a message on the screen for each drive they detect and configure this way. The BIOS will also now search for and label logical devices (COM and LPT ports). If the BIOS supports the Plug and Play standard, it will detect and configure Plug and Play devices at this time and display a message on the screen for each one it finds.

The BIOS will display a summary screen about your system's configuration. Checking this page of data can be helpful in diagnosing setup problems, although it can be hard to see because sometimes it flashes on the screen very quickly before scrolling off the top.

The BIOS begins the search for a drive to boot from. Most modern BIOSes contain a setting that controls if the system should first try to boot from the floppy disk (A:) or first try the hard disk (C:). Some BIOSes will even let you boot from your CD-ROM drive or other devices, depending on the boot sequence BIOS setting.

Having identified its target boot drive, the BIOS looks for boot information to start the operating system boot process. If it is searching a hard disk, it looks for a master boot record at cylinder 0, head 0, sector 1 (the first sector on the disk); if it is searching a floppy disk, it looks at the same address on the floppy disk for a volume boot sector.

If it finds what it is looking for, the BIOS starts the process of booting the operating system, using the information in the boot sector. At this point, the code in the boot sector takes over from the BIOS. If the first device that the system tries (floppy, hard disk, etc.) is not found, the BIOS will then try the next device in the boot sequence, and continue until it finds a bootable device.

If no boot device at all can be found, the system will normally display an error message and then freeze up the system. What the error message is depends entirely on the BIOS, and can be anything from the rather clear "No boot device available" to the very cryptic "NO ROM BASIC - SYSTEM HALTED". This will also happen if you have a bootable hard disk partition but forget to set it active.

This process is called a "cold boot" (since the machine was off, or cold, when it started). A "warm boot" is the same thing except it occurs when the machine is rebooted using {Ctrl}+{Alt}+{Delete} or similar. In this case the POST is skipped and the boot process continues roughly at step 8 above

BIOS Power-On Self Test (POST) :-

The first thing that the BIOS does when it boots the PC is to perform what is called the Power-On Self-Test, or POST for short. The POST is a built -in diagnostic program that checks your hardware to ensure that everything is present and functioning properly, before the BIOS begins the actual boot. It later continues with additional tests (such as the memory test that you see printed on the screen) as the boot process is proceeding.

The POST runs very quickly, and you will normally not even noticed that it is happening--unless it finds a problem. You may have encountered a PC that, when turned on, made beeping sounds and then stopped without booting up. That is the POST telling you something is wrong with the machine. The speaker is used because this test happens so early on, that the video isn't even activated yet! These beep patterns can be used to diagnose many hardware problems with your PC. The exact patterns depend on the maker of the BIOS; the most common are Award and AMI BIOSes.

Note: Some POST errors are considered "fatal" while others are not. A fatal error means that it will halt the boot process immediately (an example would be if no system memory at all is found). In fact, most POST boot errors are fatal, since the POST is testing vital system components.

Many people don't realize that the POST also uses extended troubleshooting codes that you can use to get much more detail on what problem a troublesome PC is having. You can purchase a special debugging card that goes into an ISA slot and accepts the debugging codes that the BIOS sends to a special I/O address, usually 80h. The card displays these codes and this lets you see where the POST stops, if it finds a problem. These cards are obviously only for the serious PC repairperson or someone who does a lot of work on systems.

Power Good Signal

When the power supply first starts up, it takes some time for the components to get "up to speed" and start generating the proper DC voltages that the computer needs to operate. Before this time, if the computer were allowed to try to boot up, strange results could occur since the power might not be at the right voltage. It can take a half-second or longer for the power to stabilize and this is an eternity to a processor that can run half a billion instructions per second! To prevent the computer from starting up prematurely, the power supply puts out a signal to the motherboard called "Power Good" (or "PowerGood", or "Power OK", or "PWR OK" and so on) after it completes its internal tests and determines that the power is ready for use. Until this signal is sent, the motherboard will refuse to start up the computer.

In addition, the power supply will turn off the Power Good signal if a power surge or glitch causes it to malfunction. It will then turn the signal back on when the power is OK again, which will reset the computer. If you've ever had a brownout where the lights flicker off for a split-second and the computer seems to keep running but resets itself, that's probably what happened. Sometimes a power supply may shut down and seem "blown" after a power problem but will reset itself if the power is turned off for 15 seconds and then turned back on.

The nominal voltage of the Power Good signal is +5 V, but in practice the allowable range is usually up to a full volt above or below that value. All power supplies will generate the Power Good signal, and most will specify the typical time until it is asserted. Some extremely el-cheapo power supplies may "fake" the Power Good signal by just tying it to another +5 V line. Such a system essentially has no Power Good functionality and will cause the motherboard to try to start the system before the power has fully stabilized. Needless to say, this type of power supply is to be avoided. Unfortunately, you cannot tell if your power supply is "faking" things unless you have test equipment. Fortunately, if you buy anything but the lowest quality supplies you don't really need to worry about this.

.

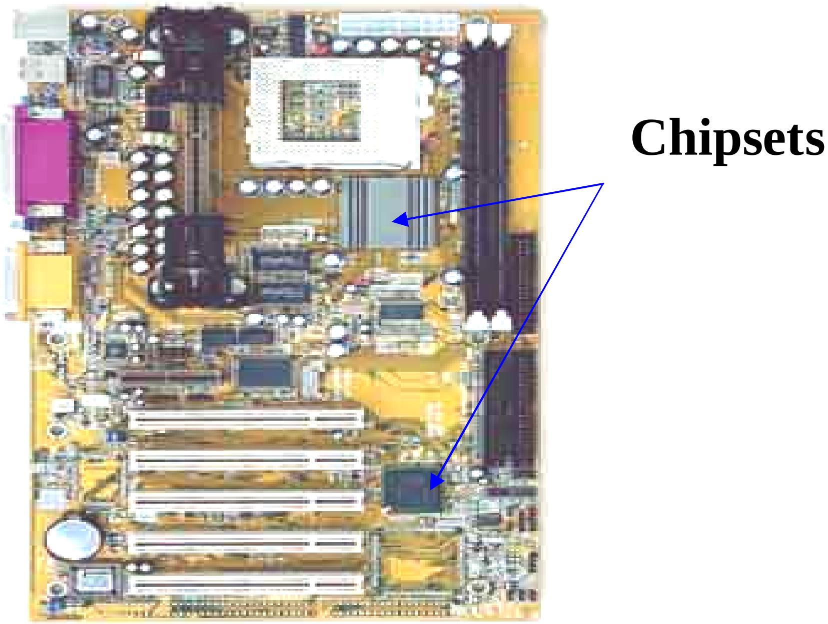
CHIPSET

A Chipset is a combination of chips that sit on the motherboard. It's responsible for most communication that takes place between different components on the motherboard; and decides many important features of a motherboard, like the AGP speed (2x or 4x), peak hard disk speed (ATA 66/100), memory type (RDRAM or SDRAM) and size. Chipsets differ in the features and functionality that they can offer to a motherboard, and consequently the entire system.

The motherboards are normally designed around the two chips, which are known as the "North bridge-South bridge architecture" one is located in the upper edge of the board (therefore the term north bridge) and controls the transfer and co-ordination of the information between the processor and the system memory. The other chip is located near the lower edge (hence the name south bridge) and takes care of the transfer of data between the interface slots and the system BIOS.

There are following chipset manufacturers:

Intel, AMD, VIA, Ali & SiS.



The following tasks are performed by the chipset:

1. CPU INTERFACE.

   The type of packaging – i.e. the casing that houses the CPU circuitry had done for microprocessors have been under constant development. Due to this, the type of interface a CPU uses to connect to a motherboard, which is identified by the chipset.

2. SOUND AND VIDEO SUPPORT.

   Some motherboards, like the i810, come with integrated sound and video. Thus motherboards based on the i810 & similar chipsets are a hit in the entry-level segment & on the corporate desktop. The i815e chipset has options for both on-board &external video. You can disable the onboard audio & video &put in your own cards. The chipset also determines the AGP port, which is normally used to house the video card.

3. FSB SETTINGS.

   Another critical function governed by the chipset is front side bus speed (FSB). This is the maximum speed at which memory on the motherboard can work. Three FSB speeds commonly available are 66MHz, 100MHz & 133MHz. Also the CPU speed is a multiple of FSB speed. So its again unto the chipset to build support for higher bus speeds & more multiples so that more CPU speeds can be incorporated.

4. HARD DRIVE CONTROLLER.

   When we say that a chipset has an ATA-66 controller, it means that it can support a peak transfer rate of 66 MB/sec for hard drives. ATA-66 support has been the standard in most of the motherboards. However ATA-100 was recently introduced. Moreover, these transfer rates are theoretical. Hard drives have a lot of other overheads that prevents them from achieving these.

5. RAM.

   Nowadays, not only does RAM run at different speeds, there are different kinds of memory. SDRAM is available in 100MHz & 133MHz FSB. Thus you can use slower memory even if higher speed is supported.

6. USB

   Chipsets also govern USB support on motherboard.

7. HARDWARE MONITORING.

   Hardware monitoring is a useful feature, though power users for the purpose of overclocking mostly use it. It can warn you in case your CPU fan dies all of sudden, the temperature of your CPU goes too high, or there is a problem in the voltage being supplied to the processor.

PERIPHERAL SLOTS

Industry Standard Architecture (ISA)

By 1984, the rudimentary design of the PC bus was already falling behind the times. As IBM's engineers were working on a revolutionary new product (for then) based on a fast 286 microprocessor designed to run at 8 MHz (though initially limited to 6 MHz), they confronted a bus unsuited for the performance level of the new machine. Because the 286 used a full 16-bit data bus, IBM decided to add more data signals (as well as address and control signals) to the PC bus to match the capabilities of the new and more powerful chip. The bus speed of the AT also was matched to the microprocessor, so again no performance penalty was incurred in connecting a peripheral—even expansion memory—to the bus.

Not only was the PC bus limited in its memory handling and the width of its data path to the capabilities of a microprocessor on the road to oblivion (the 8088), but also many of the available system services were in too short supply for growth of the PC beyond a desktop platform for simple, single-minded jobs. For example, most systems ran out of hardware interrupts long before they ran out of expansion slots and expansion boards needing interrupts for control. At the same time, engineers were faced by the profusion of PC bus-based expansion products, many of those made by IBM, which would be rendered incompatible if the bus were radically changed. A complete redesign required creating an entirely new line of expansion products for IBM and the compatibles industry, probably creating an outcry loud enough to weaken the IBM standard.

As a result of balancing these conflicting needs, the new AT bus was born a hybrid. It retained compatibility with most earlier PC expansion products, while adding the functionality needed to push forward into full 16-bit technology. In addition, The AT bus contained a few new ideas (at least for PC-compatible computers) that hinted at—and perhaps even foretold—the Micro Channel. Inherent in the AT bus but almost entirely unused are provisions for cohabiting microprocessors inside the system, able to take control and share resources.

The big physical difference between the PC/XT bus and the AT bus was the addition of a second connector to carry more data and address lines—four more address lines and eight data—for a total of 16 data lines and 24 address lines, enough to handle 16 megabytes, the physical addressing limit of the 80286 chip. To make up for some of the shortcomings of the PC, which limited its expandability, the new AT bus also included several new interrupt and DMA control lines. In addition, IBM added a few novel connections. One in particular helps make expansion boards compatible across the 8- and 16-bit lines of the IBM PC; it signals to the host that the card in the socket uses the PC or AT bus.

Maintaining physical compatibility with the earlier PC bus was accomplished with the simple but masterful stroke of adding the required new bus connections on a supplementary connector rather than redesigning the already entrenched 62-pin connector. Expansion cards that only required an 8-bit interface and needed no access to protected mode memory locations or the advanced system services of the AT could be designed to be compatible with the full line of 8- and 16-bit IBM-standard computers. Those needing the speed or power of the AT could get it through the supplemental connector. The design even allowed cards to use either 8- or 16-bit expansion depending on the host in which they were installed.

Because of its initial speed and data-path match with the 286 microprocessor, the original AT bus substantially out-performed the PC bus—its 16-bit data path combined with its 8 MHz clock (in its most popular form) yielded a potential peak transfer rate of 8MB/sec. Its 24 address lines put 16MB of memory within reach. However, the number of useful I/O ports was still limited to 1,024 because of compatibility concerns with PC bus expansion boards.

The AT bus design incorporated one major structural difference over the original PC bus, however. Where the PC had a single oscillator to control all its timing signals including bus and microprocessor, the AT used several separate oscillators. The microprocessor speed, time-of-day clock, system timer, and bus speed were separated and could be independently altered. As a result, separate clocks could be used for the microprocessor and the expansion bus (as well as the system timers). This change allowed expansion boards to operate at a lower speed from that of the microprocessor. Because of this change, the ultra-compatible AT bus could be used with higher performance PCs as they became available. Although expansion boards might not work at the 25 MHz or 33 MHz clock speed of 386 and newer microprocessors, the bus could be held back to its 8 MHz rate (or a slightly higher sub-multiple of the microprocessor clock frequency) to ensure backward compatibility with old expansion boards. At first, the lower speed of the bus was no problem because nothing anyone wanted to plug into the bus needed to transfer data faster than 8MB/sec. For example, the fastest devices of the time—state-of-the-art ESDI drives—pushed data around at a 1.25MB/sec rate, well within the peak 8MB/sec limit of ISA. Eventually, however, the speed needs of peripherals (and memory) left the AT bus design far behind.

One glaring problem with the original PC and AT expansion buses was that they were designed not just for peripherals but also for the basic memory expansion of the host PC. This worked at first, when both microprocessor and bus ran at the same speed, but became bothersome as microprocessors raced ahead of bus capabilities—to such extreme rates as 16 MHz! Adding memory for a fast microprocessor into a slow bus just doesn’t make sense. Every time the PC would need to access its bus-mounted memory, it would have to slow down to bus speed.

In early 1987, Compaq Computer Corporation cleverly sidestepped this problem with the introduction of its first Deskpro 386, which operated at 16 MHz. The first dual-bus PC, the Deskpro was the first machine to provide a separate bus for its memory, operating at microprocessor speed, and for input/output operations, operating at the lower speeds that expansion boards can tolerate. All modern PCs exploit this dual-bus concept, expanding on it with a third bus. The AT bus suffered another shortcoming. Although IBM documented the function of every pin on the AT bus, IBM never published a rigorous set of timing specifications for the signals on the bus. As a result, every manufacturer of AT expansion boards had to guess at timing and hope that their products would work in all systems. Although this empirical approach usually did not interfere with operation at 8 MHz, compatibility problems arose when some PC makers pushed the AT bus beyond that speed. The timing specifications of the AT bus were not officially defined until 1987 when a committee of the IEEE (Institute of Electrical and Electronic Engineers) formally approved a bus standard that became known as Industry

Standard Architecture or simply ISA. It also goes under several other names: ISA, classic bus, and its original name, AT bus.

The problem with holding the speed of the ISA bus at 8 MHz for backward expansion board compatibility first became apparent when people wanted to add extra memory to their higher speed PCs. When the microprocessor clock speed exceeded the bus speed, the microprocessor had to slow down (by adding wait states) whenever it accessed memory connected through the expansion bus. System performance consequently suffered, sometimes severely.

System designers at Compaq solved the problem by devoting a special, second bus to memory in the company's 1987 Deskpro 386. All current ISA-based PCs follow this design—a separate bus for high speed memory and another for I/O expansion.
Since the time the IEEE set the ISA specification, its bus signals have remained essentially unchanged. The introduction of the Plug-and-Play ISA specification on May 28, 1993, a joint development by Intel and Microsoft, alters the way expansion boards work in conjunction with the bus.

Plug-and-Play ISA is designed to give ISA systems the same, if not better, self-configuration capabilities enjoyed by more recent expansion bus designs. In fully compliant systems, you can plug in any combination of expansion boards and never have to worry about such things as DIP switch settings, jumper positions, interrupts, DMA channels, ports, or ROM ranges. Each Plug-and-Play ISA card can tell its computer host exactly what resource it requires. If the resource requests of two or more cards conflict, the Plug-and-Play system automatically straightens things out.

Instead of altering the bus, Plug-and-Play ISA substitutes an elaborate software-based isolation protocol. Effectively, it keeps an expansion board switched off until it can be uniquely addressed, so that one card can be queried at a time. The host system then can determine the resources the board needs; check to make sure that no other board requires the same resources; and reserve those resources for the target board.

Although Plug-and-Play ISA does not require them, it can make use of slot-specific address-enabled signals. The use of such signals—which are now not part of the ISA specification—can eliminate the complex software-query system used for isolating cards. While software-based Plug-and-Play configuration is possible with current systems, using the streamlined hardware-based scheme requires new motherboards.

Peripheral Component Interface

In July 1992, Intel Corporation introduced Peripheral Component Interconnect. Long awaited as a local bus specification, the initial announcement proved to be more and less than the industry hoped for. The first PCI specification fully documented Intel's conception of what local bus should be—and it wasn't a local bus. Instead, Intel defined mandatory design rules, including hardware guidelines to help ensure proper circuit operation of motherboards at high speeds with a minimum of design complication. It showed how to link PC circuits—including the expansion bus—for high speed operation. But the initial PCI announcement fell short exactly where the industry wanted the most guidance: the pinout of an expansion bus connector that allows the design of interchangeable expansion boards. In truth, PCI turned out not to be a local bus at all, but a high speed interconnection system a step removed from the microprocessor—but one that runs more closely to microprocessor speed than does a traditional expansion bus.

Although in its initial form, PCI was not incompatible with VL Bus, Intel positioned its design more as a VL Bus alternative by introducing PCI Release 2.0 in May 1993. The new specification extended the original document in two primary ways. It broadened the data path to 64 bits to match the new Pentium chip, and it gave a complete description of expansion connectors for both 32-bit and 64-bit implementations of a PCI expansion bus. The design is unlike and incompatible with the VL Bus. Foremost, PCI 2.0 was designed to be microprocessor-independent rather than limited to Intel's own chips. Instead of linking almost directly to the microprocessor, the PCI 2.0 specification provided a compatibility layer, making it what some industry insiders call a mezzanine bus. Whereas VL Bus was designed to augment more traditional expansion buses in a PC (the specification defines ISA, MCA, and EISA design alternatives), PCI tolerates older buses but can also replace them. In fact, machines that combine PCI with a traditional bus may serve as a foundation to move from ISA to PCI as the primary personal computer expansion standard.

The PCI bus provides superior performance to the VESA local bus; in fact, PCI is the highest performance general I/O bus currently used on PCs. This is due to several factors:

- Burst Mode: The PCI bus can transfer information in a burst mode, where after an initial address is provided multiple sets of data can be transmitted in a row. This works in a way similar to how cache bursting works.
- Bus Mastering: PCI supports full bus mastering, which leads to improved performance.
- High Bandwidth Options: The PCI bus specification version 2.1 calls for expandability to 64 bits and 66 MHz speed; if implemented this would quadruple bandwidth over the current design. In practice the 64-bit PCI bus has yet to be implemented on the PC (it does exist in non-PC platforms such as Digital Equipment's Alpha and is also found now on servers) and the speed is currently limited to 33 MHz in most PC designs, most likely for compatibility reasons. For mainstream PCI, we may be limited to 32 bits and 33 MHz for some time to come. However, it appears that the higher-performance PCI options are going to live on, albeit in modified form, through the new Accelerated Graphics Port.

Accelerated Graphics Port (AGP)

The need for increased bandwidth between the main processor and the video subsystem originally lead to the development of the local I/O bus on the PCs, starting with the VESA local bus and eventually leading to the popular PCI bus. This trend continues, with the need for video bandwidth now starting to push up against the limits of even the PCI bus.

Much as was the case with the ISA bus before it, traffic on the PCI bus is starting to become heavy on high-end PCs, with video, hard disk and peripheral data all competing for the same I/O bandwidth. To combat the eventual saturation of the PCI bus with video information, a new interface has been pioneered by Intel, designed specifically for the video subsystem. It is called the Accelerated Graphics Port or AGP.

AGP was developed in response to the trend towards greater and greater performance requirements for video. As software evolves and computer use continues into previously unexplored areas such as 3D acceleration and full-motion video playback, both the processor and the video chipset need to process more and more information. The PCI bus is reaching its performance limits in these applications, especially with hard disks and other peripherals also in there fighting for the same bandwidth.

Another issue has been the increasing demands for video memory. As 3D computing becomes more mainstream, much larger amounts of memory become required, not just for the screen image but also for doing the 3D calculations. This traditionally has meant putting more memory on the video card for doing this work. There are two problems with this:

- Cost: Video card memory is very expensive compared to regular system RAM.
- Limited Size: The amount of memory on the video card is limited: if you decide to put 6 MB on the card and you need 4 MB for the frame buffer, you have 2 MB left over for processing work and that's it (unless you do a hardware upgrade). It's not easy to expand this memory, and you can't use it for anything else if you don't need it for video processing.

AGP gets around these problems by allowing the video processor to access the main system memory for doing its calculations. This is more efficient because this memory can be shared dynamically between the system processor and the video processor, depending on the needs of the system.

The idea behind AGP is simple: create a faster, dedicated interface between the video chipset and the system processor. The interface is only between these two devices; this has three major advantages: it makes it easier to implement the port, makes it easier to increase AGP in speed, and makes it possible to put enhancements into the design that are specific to video.

AGP is considered a port, and not a bus, because it only involves two devices (the processor and video card) and is not expandable. One of the great advantages of AGP is that it isolates the video subsystem from the rest of the PC so there isn't nearly as much contention over I/O bandwidth as there is with PCI. With the video card removed from the PCI bus, other PCI devices will also benefit from improved bandwidth.

AGP is a new technology and was just introduced to the market in the third quarter of 1997. The first support for this new technology will be from Intel's 440LX Pentium II chipset. More information on AGP will be forthcoming as it becomes more mainstream and is seen more in the general computing market. Interestingly, one of Intel's goals with AGP was supposed to be to make high-end video more affordable without requiring sophisticated 3D video cards. If this is the case, it really makes me

wonder why they are only making AGP available for their high-end, very expensive Pentium II processor line. :^) Originally, AGP was rumored to be a feature on the

430TX Pentium socket 7 chipset, but it did not materialize. Via and other companies are carrying the flag for future socket 7 chipset development now that Intel has dropped it, and several non-Intel AGP-capable chipsets will be entering the market in 1998.

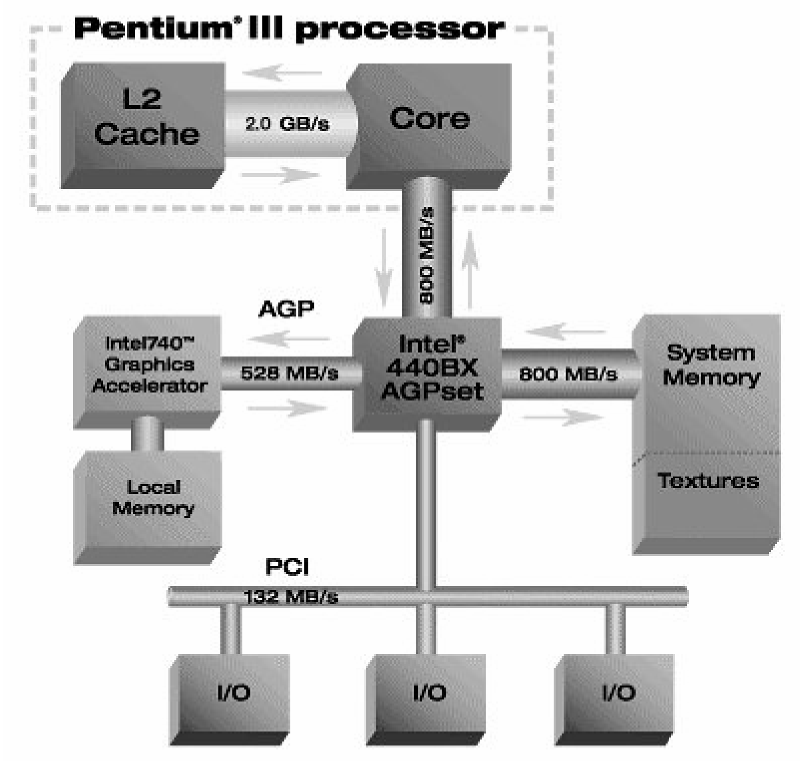


Table 7.1. Comparison of Expansion Bus Standards

Name	Date	Bus width	Clock speed	Addressing
PC bus	1981	8 bits	4.77 MHz.	1MB
ISA	1984	16 bits	8 MHz	16MB
Micro Channel	1987	32 bits	10 MHz	16MB
EISA	1988	32 bits	8 MHz	4GB
VL Bus	1992	32/64 bits	50 MHz	4GB
PCI	1992	32/64 bits	33 MHz	4GB
PC Card	1990	16 bits	8 MHz	64MB
CardBus	1994	32 bits	33 MHz	4GB

The processor (really a short form for microprocessor and also often called the CPU or CENTRAL PROCESSING UNIT) is the central component of the PC. It is the brain that runs the show inside the PC. All work that you do on your computer is performed directly or indirectly by the processor. Obviously, it is one of the most important components of the PC, if not the most important. It is also, scientifically, not only one of the most amazing parts of the PC, but one of the most amazing devices in the world of technology.

The processor plays a significant role in the following important aspects of your computer system:

- Performance: The processor is probably the most important single determinant of system performance in the PC. While other components also play a key role in determining performance, the processor's capabilities dictate the maximum performance of a system. The other devices only allow the processor to reach its full potential.

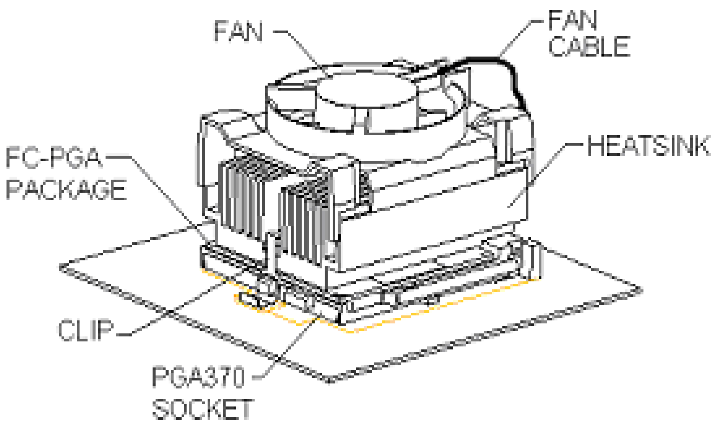
- Software Support: Newer, faster processors enable the use of the latest software. In addition, new processors such as the Pentium with MMX Technology, enable the use of specialized software not usable on earlier machines.

- Reliability and Stability: The quality of the processor is one factor that determines how reliably your system will run. While most processors are very dependable, some are not. This also depends to some extent on the age of the processor and how much energy it consumes.

- Energy Consumption and Cooling: Originally processors consumed relatively little power compared to other system devices. Newer processors can consume a great deal of power. Power consumption has an impact on everything from cooling method selection to overall system reliability.

- Motherboard Support: The processor you decide to use in your system will be a major determining factor in what sort of chipset you must use, and hence what motherboard you buy. The motherboard in turn dictates many facets of your system's capabilities and performance.

CPU Mounting



What Determines True Processor Performance?

The only measure of performance that really matters is the amount of time it takes to execute a given application. Contrary to a popular misconception, it is not clock frequency (MHz) alone or the number of instructions executed per clock (IPC) alone that equates to performance.

True performance is a combination of both clock frequency (MHz) and IPC:
Performance = MHz x IPC

This shows that the performance can be improved by increasing frequency, IPC or optimally both. It turns out that frequency is a function of both the manufacturing process and the micro-architecture. At a given clock frequency, the IPC is a function of processor micro-architecture and the specific application being executed. Although it is not always feasible to improve both the frequency and the IPC, increasing one and holding the other close to constant with the prior generation can still achieve a significantly higher level of performance.

In addition to the two methods of increasing performance described above, it is also possible to increase performance by reducing the number of instructions that it takes to execute the specific task being measured. Single Instruction Multiple Data (SIMD) is a technique used to accomplish this. Intel first implemented 64-bit integer SIMD instructions in 1996 on the Pentium ® processor with MMX ™ technology and subsequently introduced 128bit SIMD single precision floating point (SSE) on the Pentium III processor Applications can be broadly divided into two categories: integer/basic office productivity applications, and floating point/multimedia applications. The IPC achievable by these different application categories varies greatly, and this variance is strongly affected by the number of branches that the application code typically takes and the predictability of these branches. The more branches taken that are difficult to predict, the higher the possibility of mis-predicting and performing nonproductive work.

Integer and basic office productivity applications, such as word and spreadsheet processing, tend to have many branches in the codes that are difficult to predict, thus reducing overall IPC potential. As a result, performance increases on these applications are more resistant to improvements in micro-architectural means, such as deeper pipelines. Also, significantly raising the performance level on these types of applications does not necessarily increase the user's experience, as these types of applications only need to keep pace with the human level of read and write response time and today's higher end Pentium III processors satisfy this requirement.

Floating point and multimedia applications tend to have branches that are very predictable, and thus naturally have a higher average IPC potential. As a result, these types of applications generally scale very well with frequency and are inclined to benefit greatly from deeper pipelines. In addition, the processing power required by these applications tends to be unbounded: the more performance that is available, the better the user's experience.

Processor
Hall Of Fame

1971: 4004 Microprocessor

The 4004 was Intel's first microprocessor. This breakthrough invention powered the Busicom calculator and paved the way for embedding intelligence in inanimate objects including the personal computer.

1972: 8008 Microprocessor

The 8008 was twice as powerful as the 4004. A 1974 article in Radio Electronics referred to a device called the Mark-8, which used the 8008. The Mark-8 is known as one of the first computers for the home --one that by today's standards was difficult to build, maintain and operate.

1974: 8080 Microprocessor

The 8080 became the brains of one of the first personal computers -- the Altair, allegedly named for a destination of the Starship Enterprise from the Star Trek television show. Computer hobbyists could purchase a kit for the Altair for $395. Within months, it sold tens of thousands, creating the first PC back orders in history.

1978: 8086-8088 Microprocessor

A pivotal sale to IBM's new personal computer division made the 8088 the brains of IBM's new hit product--the IBM PC. The 8088's success propelled Intel into the ranks of the Fortune 500, and Fortune magazine named the company one of the "Business Triumphs of the Seventies."

1982: 286 Microprocessor

The 286, also known as the 80286, was the first Intel processor that could run all the software written for its predecessor. This software compatibility remains a hallmark of Intel's family of microprocessors. Within 6 years of it release, there were an estimated 15 million 286-based personal computers installed around the world.

1985: Intel 386 Microprocessor

The Intel386(TM) microprocessor featured 275,000 transistors--more than 100 times as many as the original 4004. It was a 32-bit chip and was "multi-tasking," meaning it could run multiple programs at the same time.

1989: Intel 486DX CPU Microprocessor

The 486(TM) processor generation really meant you go from a command-level computer into point-and-click computing. "I could have a color computer for the first time and do desktop publishing at a significant speed," recalls technology historian David K. Allison of the Smithsonian's National Museum of American History. The Intel 486(TM) processor was the first to offer a built-in math coprocessor, which speeds up computing because it offloads complex math functions from the central processor.

1993: Pentium® Processor

The Pentium® processor allowed computers to more easily incorporate "real world" data such as speech, sound, handwriting and photographic images. The Pentium brand, mentioned in the comics and on television talk shows, became a household word soon after introduction.

1995: Pentium® Pro Processor

Released in the fall of 1995, the Pentium® Pro processor is designed to fuel 32-bit server and workstation applications, enabling fast computer-aided design, mechanical engineering and scientific computation. Each Pentium® Pro processor is packaged together with a second speed-enhancing cache memory chip. The powerful Pentium® Pro processor boasts 5.5 million transistors.

1997: Pentium® II Processor

The 7.5 million-transistor Pentium® II processor incorporates Intel MMXTM technology, which is designed specifically to process video, audio and graphics data efficiently. It was introduced in innovative Single Edge Contact (S.E.C) Cartridge that also incorporated a high-speed cache memory chip. With this chip, PC users can capture, edit and share digital photos with friends and family via the Internet; edit and add text, music or between-scene transitions to home movies; and, with a video phone, send video over standard phone lines and the Internet.

1998: Pentium® II Xeon Processor

The Pentium® II XeonTM processors are designed to meet the performance requirements of mid-range and higher servers and workstations. Consistent with Intel's strategy to deliver unique processor products targeted for specific markets segments, the Pentium® II XeonTM processors feature technical innovations specifically designed for workstations and servers that utilize demanding business applications such as Internet services, corporate data warehousing, digital content creation, and electronic and mechanical design automation. Systems based on the processor can be configured to scale to four or eight processors and beyond.

1999: Celeron Processor

Continuing Intel's strategy of developing processors for specific market segments, the Intel Celeron processor is designed for the value PC market segment. It provides consumers great performance at an exceptional value, and it delivers excellent performance for uses such as gaming and educational software.

1999: Pentium® III Processor

The Pentium® III processor features 70 new instructions--Internet Streaming SIMD extensions-- that dramatically enhance the performance of advanced imaging, 3-D, streaming audio, video and speech recognition applications. It was designed to significantly enhance Internet experiences, allowing users to do such things as browse through realistic online museums and stores and download high-quality video. The processor incorporates 9.5 million transistors, and was introduced using 0.25-micron technology.

1999: Pentium® III Xeon Processor

The Pentium® III XeonTM processor extends Intel's offerings to the workstation and server market segments, providing additional performance for e-Commerce applications and advanced business computing. The processors incorporate the Pentium® III processor's 70 SIMD instructions, which enhance multimedia and streaming video applications. The Pentium® III Xeon processor's advance cache technology speeds information from the system bus to the processor, significantly boosting performance. It is designed for systems with multiprocessor configurations.

2000: Pentium® 4 Processor

Users of Pentium® 4 processor-based PCs can create professional-quality movies; deliver TV-like video via the Internet; communicate with real-time video and voice; render 3D graphics in real time; quickly encode music for MP3 players; and simultaneously run several multimedia applications while connected to the Internet. The processor debuted with 42 million transistors and circuit lines of 0.18 microns. Intel's first microprocessor, the 4004, ran at 108 kilohertz (108,000 hertz), compared to the Pentium® 4 processor's initial speed of 1.5-gigahertz (1.5 billion-hertz). If automobile speed had increased similarly over the same period, you could now drive from San Francisco to New York in about 13 seconds.

The purpose of the motherboard socket originally was just to provide a place to insert the processor into the motherboard. As such, it was no different than the sockets that were put on the board for most of the other PC components. However, over the last few years Intel, the primary maker of processors in the PC world, has defined several interface standards for PC motherboards. These are standardized socket and slot specifications to be used with various processors that are designed to use these standard sockets.

What is significant about the creation of these standards is that Intel's two main competitors, AMD and Cyrix, have been able to use these standards as well in their quest for compatibility with Intel. While packages and sockets/slots do change over time, the presence of standards allows for better implementations by motherboard makers, who can make boards that hopefully support future processors more easily than if each board had to be tailored to a specific chip.

Intel Socket 1 Specification

Designation: Socket 1
Number of Pins: 169
Pin Rows: 3
Voltage: 5 volts
Motherboard Class: 486
Supported Processors: 486SX, 486DX, 486DX2, 486DX4 OverDrive
Description: What is now called "Socket 1" was originally "the" OverDrive socket. It is found on most of the 486 systems that were originally designed to be upgradable with an OverDrive chip. It supports the older 5 volt 486SX, 486DX and 486DX2 class processors natively. The only OverDrive that will fit in the Socket 1 is the 486DX4 OverDrive; the Pentium OverDrive will not fit because it has four rows of pins and the socket only has three. Socket 1 has been obsolete for some time. Note that this is not the same as the 168-pin socket that was used for the original processor on many 486 systems, because that socket will not take an OverDrive chip.

Intel Socket 2 Specification

Designation: Socket 2
Number of Pins: 238
Pin Rows: 4
Voltage: 5 volts
Motherboard Class: 486
Supported Processors: 486SX, 486DX, 486DX2, 486DX4 OverDrive, Pentium OverDrive
Description: Socket 2 was the first OverDrive socket put on 486 systems that was intended to support the Pentium OverDrive chip. It supports the older 5 volt 486SX, 486DX and 486DX2 class processors directly, and the 486DX4 and Pentium OverDrives.

Intel Socket 3 Specification

Designation: Socket 3
Number of Pins: 237
Pin Rows: 4
Voltage: 5 volts / 3.3 volts
Motherboard Class: 486
Supported Processors: 486SX, 486DX, 486DX2, 486DX4, Pentium OverDrive, 5x86
Description: Socket 3 is the most recent and current socket for 486 class machines. The most important modification from the socket 2 design is support for 3.3 volt power; this allows the socket to use the most recent 486-class processors, including the AMD and Cyrix 5x86 processors. A jumper setting on the motherboard is normally used to select between 3.3 and 5 volt operation. The socket also supports the Pentium OverDrive processor.

Intel Socket 4 Specification

Designation: Socket 4
Number of Pins: 273
Pin Rows: 4
Voltage: 5 volts
Motherboard Class: 1st Generation Pentium
Supported Processors: Pentium 60-66, Pentium OverDrive
Description: Socket 4 was the first socket designed for native support of the early Pentium processors, running at 60 or 66 MHz. It is the only 5 volt Pentium socket. These machines had no real upgrade path to the faster versions of the Pentium because starting with the 75 MHz version, Intel switched to 3.3 volt power. This socket does support a special Pentium OverDrive, running at 120 MHz (for the 60 MHz) or 133 MHz (for the 66).

Intel Socket 5 Specification

Designation: Socket 5
Number of Pins: 320
Pin Rows: 5 (staggered)
Voltage: 3.3 volts
Motherboard Class: Pentium

Supported Processors: Pentium 75-133 MHz, Pentium OverDrive
Description: Socket 5 is the first socket designed for the mainstream (second generation) Pentium processors. It supports low-speed Pentiums from 75 to 133 MHz. Higher-speed Pentiums such as the 166 MHz and 200 MHz, and the newer Pentiums with MMX, will not work in a Socket 5 because they have an extra pin. They must be used in a Socket 7. Pentium OverDrives to upgrade Socket 5 Pentiums exist to allow upgrades to these motherboards. Socket 5 is now obsolete, replaced by Socket 7.

Intel Socket 6 Specification

Designation: Socket 6
Number of Pins: 235
Pin Rows: 4
Voltage: 3.3 volts
Motherboard Class: 486
Supported Processors: 486DX4, Pentium OverDrive
Description: Socket 6 is the last 486 class socket standard created by Intel. It is a slightly modified Socket 3, and it never caught on in the marketplace. Presumably, with Intel discontinuing the 486 line of processors, motherboard manufacturers did not see any need to incur the cost of changing their designs from the Socket 3 standard. Socket 6 is not used in modern motherboards.

Intel Socket 7 Specification

Designation: Socket 7
Number of Pins: 321
Pin Rows: 5 (staggered)
Voltage: 2.5-3.3 volts
Motherboard Class: Pentium
Supported Processors: Pentium 75-200 MHz, Pentium OverDrive, Pentium with MMX, Pentium with MMX OverDrive, K5, 6x86, K6, 6x86MX
Description: Socket 7 is the most popular socket for Pentium motherboards, and the closest thing to an industry standard socket on the market today. It supports a wide range of processors, including the highest performance fifth-generation chips from Intel. Furthermore, Socket 7 has been embraced by Intel competitors AMD and Cyrix, who have designed not only Pentium-class processors but also sixth-generation chips (AMD's K6 and Cyrix's 6X86MX) to fit the standard. For its part, Intel has moved on to newer designs, not intending to put sixth-generation technology on the Socket 7 standard.

Socket 7 motherboards were the first to incorporate integral voltage regulators modules, to supply the lower (sub 3.3 volt) voltages required to internally power newer generation processors. Not all Socket 7 motherboards support the lower voltages, however; it was up to the motherboard manufacturer to plan for the future and make this flexibility an option, and not all of them did this before the first Pentium with MMX was released that required sub-3-volt power. Intel produces a Pentium with MMX OverDrive to be used in motherboards that don't support the 2.8 volt power requirement natively.

Intel Socket 8 Specification

Designation: Socket 8
Number of Pins: 387
Pin Rows: 5 (dual pattern)
Voltage: 3.1 volts / 3.3 volts
Motherboard Class: Pentium Pro
Supported Processors: Pentium Pro, Pentium Pro OverDrive, Pentium II OverDrive
Description: Socket 8 is socket for the Pentium Pro processor, specially designed to handle its unusual dual-cavity, rectangular package. Since Intel has already decided to

move away from the Pentium Pro design for future processors, existing Socket 8 motherboards can only be upgraded through OverDrive chips. Intel has pledged to make available both higher-speed Pentium Pro OverDrives and also Pentium II OverDrive chips.

Socket 8 is the only one that supports the Pentium Pro. Since it is in essence a "dead end" technologically given Intel's decision to move to SEC (daughtercard) packaging starting with the Pentium II, some motherboard manufacturers have created a clever design for their newer boards that will support the Pentium Pro in a Pentium II slot. The Socket 8 is itself mounted into an SEC daughtercard similar to the one used by the Pentium II, which is inserted into a Slot 1 on the motherboard. Later, this card can be replaced by a Pentium II or later processor. This gives Pentium Pro buyers flexibility for future upgrades.

Intel Slot 1 Specification

Designation: Slot 1
Number of Pins: 242
Pin Rows: 2
Voltage: 2.8-3.3
Motherboard Class: Pentium Pro / Pentium II
Supported Processors: Pentium II, Pentium Pro (with Socket 8 on daughtercard)
Description: The most significant change in motherboard interfacing since the creation of the pin grid array with the 80286, Slot 1 is the first to use the new SEC daughtercard technology created for the Pentium II processor. The slot provides the interface to the processor and level 2 cache on the SEC card. In addition, many Slot 1 motherboards are being designed to accept a daughtercard carrying a Socket 8 for the Pentium Pro, to allow Pentium Pro buyers an upgrade path to Slot 1 processors later on.

Summary of Sockets and Slots for Specific Processors

The table below summarizes the main characteristics of the Intel socket and slot standards. Shown also are the main processors used with each socket, and the type of motherboard the socket is used on. Note that there are many different types of Pentium OverDrive processor, each geared specifically to the type of socket it is used in.

Designation	# of Pins	Pin Rows	Voltage	Motherboard Generation	Supported Processors
Socket 1	169	3	5V	Fourth	80486DX, 80486SX, 80486DX2, 80486DX4 OverDrive
Socket 2	238	4	5V	Fourth	80486DX, 80486SX, 80486DX2, 80486DX4 OverDrive, Pentium OverDrive 63 and 83
Socket 3	237	4	5V / 3.3V	Fourth	80486DX, 80486SX, 80486DX2, 80486DX4, AMD 5x86, Cyrix 5x86, Pentium OverDrive 63 and 83
Socket 4	273	4	5V	Fifth (5V)	Pentium 60-66, Pentium OverDrive 120/133
Socket 5	320	5	3.3V	Fifth	Pentium 75-133 MHz, Pentium OverDrive 125-166, Pentium with MMX OverDrive 125-166
Socket 6	235	4	3.3V	Fourth	Not used
Socket 7	321	5	2.5-3.3V	Fifth	Pentium 75-200 MHz, Pentium OverDrive, Pentium with MMX, Pentium with MMX OverDrive, 6x86, K5, K6, 6x86MX
Socket 8	387	5	3.1V / 3.3V	Sixth	Pentium Pro
Slot 1	242	n/a	2.8V / 3.3V	Sixth	Pentium II, Pentium Pro (with Socket 8 on daughtercard

PENTIUM 4

Based on the all-new Intel® NetBurst™ micro-architecture, the Pentium 4 processor delivers breakthrough performance to handle next generation multi-tasking environments and unleashes the richness of the visual Internet. The Pentium 4 processor is optimized for Internet technologies such as JAVA* and XML--the new language of business. The Intel NetBurst micro-architecture allows the Pentium 4 processor to deliver this next-generation performance so it can be fully experienced and appreciated by the user, rather than focusing on simply speeding up applications such as word and spreadsheet processing.

The Intel NetBurst micro-architecture is the latest, true micro-architectural generation from Intel that implements the IA-32 architecture. The introduction of the Pentium 4 processor signifies a complete processor re-design that delivers new technologies and capabilities while advancing many of the innovative features introduced on prior Intel® micro-architectural generations.

Key Features

With the Pentium 4 processor, Intel delivers revolutionary change. Architectural innovations in the new design include the following features:

- Hyper pipelined technology to deliver significantly higher performance and frequency for scalability
- Rapid Execution Engine to execute integer instructions at lightning speed
- The technology to deliver an effective 400 MHz system bus
- Execution Trace Cache to deliver more instruction bandwidth to the core and make more efficient use of the cache storage.

Need more? The Pentium 4 processor also significantly builds upon the many of the features that the Pentium® III processor delivered:

- Faster processor clock speeds of up to 1.70 GHz
- 144 new SIMD instructions over Streaming SIMD Extensions (SSE) and MMX™ technologies to make a complete SIMD instruction set
- Advanced Dynamic Execution to deliver an enhanced branch prediction capability and a more efficient means to processing data
- Advanced Transfer Cache to provide a much higher data throughput channel between the Level 2 cache and the processor core
- Enhanced floating point and multimedia delivers a high bandwidth path into the floating point and multimedia units to keep executing.

Inside the NetBurst Micro-Architecture of The Intel Pentium 4 Processor

INTRODUCTION

The Intel ® NetBurst ™ micro-architecture is the foundation for the Intel Pentium ® 4 processor. It includes several important new features and innovations that will allow the Intel Pentium 4 processor and future IA-32 processors to deliver industry-leading performance for the next several years. This paper describes the most important features and innovations included in the Intel NetBurst micro-architecture.

Processor architecture versus micro-architecture

The architecture of a processor refers to the instruction set, registers, and memory-resident data structures that are public to a programmer and are maintained and enhanced from one generation of architecture to the next. The micro-architecture of a processor refers to implementation of processor architecture in silicon. Within a family of processors, like the Intel IA-32 processors, the micro-architecture typically changes from one processor generation to the next, while implementing the same public processor architecture. Intel's IA-32 architecture is based on the x86 instruction set and registers. It has been enhanced and extended through generations of IA-32 processors, while maintaining backward compatibility for code written to run on the earliest IA-32 processors.

New micro-architectures have historically been required to drive increases in processor performance for particular processor architecture. The early life cycle of each micro-architecture generation delivers a large performance gain over time. However, as the micro-architectural design matures, the performance delivered starts to diminish, requiring new micro-architectural advances in order to maintain the performance trajectory expected by the marketplace. The Intel NetBurst micro-architecture is the latest, true micro-architectural generation from Intel that implements the IA-32 architecture. This micro-architecture, along with several extensions to the IA-32 architecture, have been designed not only to increase the raw instruction processing speed of IA-32 processors, but also to unleash the richness of the visual internet. The Intel NetBurst micro-architecture allows the Pentium 4 processor to deliver this next-generation performance so it can be fully experienced and appreciated by the user, rather than focusing on simply speeding up applications such as word and spreadsheet.

Processing these types of applications need only to keep pace with a human level of response time, unlike multimedia applications, which have an almost unbounded, need for performance.

The NetBurst Micro-Architecture of the Intel Pentium 4 Processor

The Pentium 4 processor, utilizing the NetBurst micro-architecture, is a complete processor re-design that delivers new technologies and capabilities while advancing many of the innovative features, such as “out-of-order speculative execution” and “super-scalar execution”, introduced on prior Intel micro-architectural generations. Many of these new innovations and advances were made possible with the improvements in processor technology, process technology and circuit design and could not previously be implemented in high-volume, manufacturable solutions. The features and resulting benefits of the new micro-architecture are defined in the following sections.

Designed for Performance

A focused architectural definition effort was used to study the benefits of many advanced processor technologies and determine the best approach to improve the overall performance of the processor for many years to come. The result of this definition effort was a micro-architecture that significantly increased frequency capabilities to well above 40% higher than that of the P6 micro-architecture (on the same manufacturing process) while maintaining an average IPC that was within approximately 10% to 20% of the P6 micro-architecture. In this design, although the IPC is lower, the increase in frequency capability more than makes up (Performance = frequency x IPC) and delivers overall higher performance capability to the end user. This was done in the NetBurst micro-architecture by implementing a hyper-pipelined technology where the depth of the pipeline was doubled from that of the P6 micro-architectural generation.

Although this deeper pipeline delivers significantly higher levels of frequency, the potential performance impacts associated with the longer pipeline were comprehended and overcome in the design. The design effort focused on the following:

M*inimizing the Penalty Associated with Branch Mis-predicts*

Explanation of Branch Mis-predict Penalty: As with the P6 generation, the NetBurst micro-architecture takes advantage of out-of-order, speculative execution. This is where the processor routinely uses an internal branch prediction algorithm to predict the result of branches in the program code and then speculatively executes instructions down the predicted code branch. Although branch prediction algorithms are highly accurate, they are not 100% accurate. If the processor mis-predicts a branch, all the speculatively executed instructions must be flushed from the processor pipeline in order to restart the instruction execution down the correct program branch. On more deeply pipelined designs, more instructions must be flushed from the pipeline, resulting in a longer recovery time from a branch mis-predict. The net result is that applications that have many, difficult to predict branches will tend to have a lower average IPC.

Minimization of mis-predict penalty: To minimize the branch mis-prediction penalty and maximize the average IPC, the deeply pipelined NetBurst micro-architecture greatly reduces the number of branch mis-predicts and provides a quick method of recovering from any branches that have been mis-predicted. To minimize this penalty, the NetBurst micro-architecture has implemented an Advanced Dynamic Execution engine and an Execution Trace Cache. These features are both described later in this paper.

Keeping the High-Frequency Execution Units Busy (vs. Sitting Idle)

Although a processor may have a high frequency capability, it must provide a means to ensure that the execution units (integer and floating point) are continually being supplied with instructions for execution. This ensures that these high-frequency units are executing instructions (not sitting idle). With the high frequency of these execution units in the NetBurst micro-architecture and the implementation of the Rapid Execution Engine, where the Arithmetic Logic Units are running at two times the core frequency, Intel has implemented a number of features that ensure that these execution units have a continuous stream of instructions to execute.

Intel has implemented a 400-MHz system bus, an Advanced Transfer Cache, an Execution Trace Cache, an Advanced Dynamic Execution engine and a low-latency Level 1 Data Cache. These features work together to quickly provide instructions and data to the processor's high-performance execution units, thus keeping them executing code instead of just idling at high frequency.

Reducing the Number of Instructions Needed to Complete a Task or Program

Many applications often perform repetitive operations on large sets of data. Further, the data sets involved in these operations tend to be small values that can be represented with a small number of bits. These two observations can be combined to improve application performance by both compactly representing data sets and by implementing instructions that can operate in these compact data sets. This type of operation is called Single

Instruction Multiple Data (SIMD) and can reduce the overall number of instructions that a program is required to execute. The NetBurst micro-architecture implements 144 new SIMD instructions, called Streaming SIMD Extensions 2 (SSE2). The SSE2 instruction set enhances the SIMD instructions previously delivered with MMX technology and SSE technology. These new instructions support 128-bit SIMD integer operations and 128-bit SIMD double-precision floating-point operations. By doubling the amount of data on which a given instruction can operate, only half the number of instructions in a code loop need to be executed.

Intel NetBurst Micro-architecture Feature Details

Hyper-Pipelined Technology: The hyper-pipelined technology of the NetBurst micro-architecture doubles the pipeline depth, compared to the P6 micro-architecture. One of the key pipelines, the branch prediction/recovery pipeline, is implemented with a 20 stage pipeline in the NetBurst micro-architecture, compared to the equivalent pipeline in the P6 micro-architecture, which was implemented with a 10 stage pipeline. This technology significantly increases processor performance and frequency scalability of the base micro-architecture.

Execution Trace Cache: The Execution Trace Cache is an innovative way to implement a Level 1-instruction cache. It caches decoded x86 instructions (micro-ops), thus removing the latency associated with the instruction decoder from the main execution loops. In addition, the Execution Trace Cache stores these micro-ops in the path of program execution flow, where the results of branches in the code are integrated into the same cache line. This increases the instruction flow from the cache and makes better use of the overall cache storage space (12K micro-ops) since the cache no longer stores instructions that are branched over and never executed. The result is a means to deliver a high volume of instructions to the processor's execution units and a reduction in the overall time required to recover from branches that have been mis-predicted.

Rapid Execution Engine: Through a combination of architectural, physical and circuit designs, the simple Arithmetic Logic Units (ALUs) within the processor run at two times the frequency of the processor core. This allows the ALUs to execute certain instructions with latency that is ½ the duration of the core clock and results in higher execution throughput as well as reduced latency of execution.

***400-MHz System Bus*:** Through a physical signaling scheme of quad pumping the data transfers over a 100-MHz clocked system bus and a buffering scheme allowing for sustained 400-MHz data transfers, the Pentium 4 processor supports Intel's highest performance desktop system bus delivering 3.2GB of data per second in and out of the processor. This compares to 1.06GB/s delivered on the Pentium III processor's 133-MHz system bus.

Advanced Dynamic Execution: The Advanced Dynamic Execution engine is a very deep, out-of-order speculative execution engine that keeps the execution units executing instructions. It does so by providing a very large window of instructions from which the execution units can choose. The large out-of-order instruction window allows the processor to avoid stalls that can occur while instructions are waiting for dependencies to resolve. One of the more common forms of stalls is waiting for data to be loaded from memory on a cache miss. This aspect is very important in high frequency designs, as the latency to main memory increases relative to the core frequency. The NetBurst micro-architecture can have up to 126 instructions in this window (in flight) vs. the P6 micro-architecture's much smaller window of 42 instructions.

The Advanced Dynamic Execution engine also delivers an enhanced branch prediction capability that allows the Pentium 4 processor to be more accurate in predicting program branches. This has the net effect of reducing the number of branch mis-predictions by about 33% over the P6 generation processor's branch prediction capability. It does this by implementing a 4KB branch target buffer that stores more detail on the history of past branches, as well as by implementing a more advanced branch prediction algorithm. This enhanced branch prediction capability is one of the key design elements that reduce the overall sensitivity of the NetBurst micro-architecture to the branch mis-prediction penalty.

Advanced Transfer Cache: The Level 2 Advanced Transfer Cache is 256KB in size and delivers a much higher data throughput channel between the Level 2 cache and the processor core. The Advanced Transfer Cache consists of a 256-bit (32-byte) interface that transfers data on each core clock. As a result, a 1.4-GHz Pentium 4 processor can deliver a data transfer rate of 44.8GB/s (32 bytes x 1 (data transfer per clock) x 1.4 GHz = 44.8GB/s). This compares to a transfer rate of 16GB/s on the Pentium III processor at 1 GHz and contributes to the Pentium 4 processor's ability to keep the high-frequency execution units executing instructions vs. sitting idle.

Streaming SIMD Extensions 2 (SSE2): With the introduction of SSE2, the NetBurst micro-architecture now extends the SIMD capabilities that MMX technology and SSE technology delivered by adding 144 new instructions that deliver 128-bit SIMD integer arithmetic operation and 128-bit SIMD Double-Precision Floating Point. These new instructions deliver the capability to reduce the overall number of instructions required to execute a particular program task and as a result can contribute to an overall performance increase. They accelerate a broad range of applications, including video, speech, and image, photo processing, encryption, financial, engineering and Scientific Applications.

Resultant Performance Expectations

The Pentium 4 processor shows immediate performance improvements across most existing software applications available today, with performance levels varying depending on the application category type and the application's tendency to execute instructions and instruction sequences that are optimally executed on the new micro-architecture.

Over time, as more applications are optimized, either specifically for the micro-architecture via assembler-level optimizations, or are revised using the latest NetBurst micro-architecture optimized compilers and libraries, we will continue to see even greater levels of performance scaling when the software runs on the Pentium 4 processor.

In summary, the Pentium 4 processor, based upon the NetBurst micro-architecture, delivers an acceleration of performance across the applications and usage where users will truly be able to experience and appreciate it. These usage include 3D visualization, gaming, video, speech, and image photo processing, encryption, financial, engineering and Scientific Applications.

MEMORY

The system memory is the place where the computer holds current programs and data that are in use. The term "memory" is somewhat ambiguous; it can refer to many different parts of the PC because there are so many different kinds of memory that a PC uses. However, when used by itself, "memory" usually refers to the main system memory, which holds the instructions that the processor executes and the data that those instructions work with. Your system memory is an important part of the main processing subsystem of the PC, tied in with the processor, cache, motherboard and chipset.

Memory plays a significant role in the following important aspects of your computer system:

- Performance: The amount and type of system memory you have is an important contributing factor to overall performance. In many ways, it is more important than the processor, because insufficient memory can cause a processor to work at 50% or even more below its performance potential. This is an important point that is often overlooked.
- Software Support: Newer programs require more memory than old ones. More memory will give you access to programs that you cannot use with a lesser amount.
- Reliability and Stability: Bad memory is a leading cause of mysterious system problems. Ensuring you have high-quality memory will result in a PC that runs smoothly and exhibits fewer problems. Also, even high-quality memory will not work well if you use the wrong kind.
- Upgradability: There are many different types of memory available, and some are more universal than others. Making a wise choice can allow you to migrate your memory to a future system or continue to use it after you upgrade your motherboard.

Types Of Memory

Read-Only Memory (ROM)

One major type of memory that is used in PCs is called read-only memory or ROM for short. ROM is a type of memory that normally can only be read, as opposed to RAM, which can be both read and written. There are two main reasons that read-only memory is used for certain functions within the PC:

- Permanence: The values stored in ROM are always there, whether the power is on or not. A ROM can be removed from the PC, stored for an indefinite period of time, and then replaced, and the data it contains will still be there. For this reason, it is called non-volatile storage. A hard disk is also non-volatile, for the same reason, but regular RAM is not.
- Security: The fact that ROM cannot easily be modified provides a measure of security against accidental (or malicious) changes to its contents. You are not going to find viruses infecting true ROMs, for example; it's just not possible. (It's technically possible with erasable EPROMs, though in practice never seen.)

Read-only memory is most commonly used to store system-level programs that we want to have available to the PC at all times. The most common example is the system BIOS program, which is stored in a ROM, called the system BIOS ROM. Having this in a permanent ROM means it is available when the power is turned on so that the PC can use it to boot up the system. Remember that when you first turn on the PC the system memory is empty, so there has to be something for the PC to use when it starts up.

While the whole point of a ROM is supposed to be that the contents cannot be changed, there are times when being able to change the contents of a ROM can be very useful. There are several ROM variants that can be changed under certain circumstances; these can be thought of as "mostly read-only memory". The following are the different types of ROMs with a description of their relative modifiability:

- ROM**:** A regular ROM is constructed from hard-wired logic encoded in the silicon itself, much the way that a processor is. It is designed to perform a specific function and cannot be changed. This is inflexible and so regular ROMs are only used generally for programs that are static (not changing often) and mass-produced.

- Programmable ROM (PROM): This is a type of ROM that can be programmed using special equipment; it can be written to, but only once. This is useful for companies that make their own roms from software they write, because when they change their code they can create new proms without requiring expensive equipment. This is similar to the way a CD-ROM recorder works by letting you "burn" programs onto blanks once and then letting you read from them many times. In fact, programming a PROM is also called burning, just like burning a CD-R, and it is comparable in terms of its flexibility.

- Erasable Programmable ROM (EPROM): An EPROM is a ROM that can be erased and reprogrammed. A little glass window is installed in the top of the ROM package, through which you can actually see the chip that holds the memory. Ultraviolet light of a specific frequency can be shined through this window for a specified period of time, which will erase the EPROM and allow it to be reprogrammed again. Obviously this is much more useful than a regular PROM, but it does require the erasing light. Continuing the "CD" analogy, this technology is analogous to a reusable CD-RW.

- Electrically Erasable Programmable ROM (EEPROM): The next level of erasability is the EEPROM, which can be erased under software control. This is the most flexible type of ROM, and is now commonly used for holding BIOS programs. When you hear reference to a "flash BIOS" or doing a BIOS upgrade by "flashing", this refers to reprogramming the BIOS EEPROM with a special software program. Here we are blurring the line a bit between what "read-only" really means, but remember that this rewriting is done maybe once a year or so, compared to real read-write memory (RAM) where rewriting is done often many times per second!

> Note: One thing that sometimes confuses people is that since RAM is the "opposite" of ROM (since RAM is read-write and ROM is read-only), and since RAM stands for "random access memory", they think that ROM is not random access. This is not true; any location can be read from ROM in any order, so it is random access as well, just not writeable. RAM gets its name because earlier read-write memories were sequential, and did not allow random access.

Finally, one other characteristic of ROM, compared to RAM, is that it is much slower, typically having double the access time of RAM or more. This is one reason why the code in the BIOS ROM is often shadowed to improve performance.

Random Access Memory (RAM)

The kind of memory used for holding programs and data being executed is called RANDOM ACCESS MEMORY or RAM. RAM differs from read-only memory (ROM) in that it can be both read and written. It is considered volatile storage because unlike ROM, the contents of RAM are lost when the power is turned off. RAM is also sometimes called read-write memory or RWM. This is actually a much more precise name, so of course it (the name) is hardly ever used. It's a better name because calling RAM "random access" implies to some people that ROM isn't random access, which is not true. RAM is called "random access" because earlier read-write memories were sequential and did not allow random access. Sometimes old acronyms persist even when they don't make much sense any more (e.g., the "AT" in the old IBM AT stands for “advanced technology.”

Obviously, RAM needs to be write able in order for it to do its job of holding programs and data that you are working on. The volatility of RAM also means that you risk losing what you are working on unless you save it frequently.

RAM is much faster than ROM is, due to the nature of how it stores information. This is why RAM is often used to shadow the BIOS ROM to improve performance when executing BIOS code. There are many different types of RAMs, including static RAM (SRAM) and many flavors of dynamic RAM (DRAM).

Static RAM (SRAM)

Static RAM is a type of RAM that holds its data without external refresh, for as long as power is supplied to the circuit. This is contrasted to dynamic RAM (DRAM), which must be refreshed many times per second in order to hold its data contents. SRAMs are used for specific applications within the PC, where their strengths outweigh their weaknesses compared to DRAM:

- Simplicity: SRAMs don't require external refresh circuitry or other work in order for them to keep their data intact.
- Speed: SRAM is faster than DRAM.

In contrast, SRAMs have the following weaknesses, compared to DRAMs:

- Cost: SRAM is, byte for byte, several times more expensive than DRAM.
- Size: SRAMs take up much more space than DRAMs (which is part of why the cost is higher).

These advantages and disadvantages taken together obviously show that performance-wise, SRAM is superior to DRAM, and we would use it exclusively if only we could do so economically. Unfortunately, 32 MB of SRAM would be prohibitively large and costly, which is why DRAM is used for system memory. SRAMs are used instead for level 1 cache and level 2 cache memory, for which it is perfectly suited; cache memory needs to be very fast, and not very large.

SRAM is manufactured in a way rather similar to how processors are: highly-integrated transistor patterns photo-etched into silicon. Each SRAM bit is comprised of between four and six transistors, which is why SRAM takes up much more space compared to DRAM, which uses only one (plus a capacitor). Because an SRAM chip is comprised of thousands or millions of identical cells, it is much easier to make than a CPU, which is a large die with a non-repetitive structure. This is one reason why RAM chips cost much less than processors do.

Dynamic RAM (DRAM)

It is a type of RAM that only holds its data if it is continuously accessed by special logic called a refresh circuit. Many hundreds of times each second, this circuitry reads the contents of each memory cell, whether the memory cell is being used at that time by the computer or not. Due to the way in which the cells are constructed, the reading action itself refreshes the contents of the memory. If this is not done regularly, then the DRAM will lose its contents, even if it continues to have power supplied to it. This refreshing action is why the memory is called dynamic.

All PCs use DRAM for their main system memory, instead of SRAM, even though DRAMs are slower than SRAMs and require the overhead of the refresh circuitry. It may seem weird to want to make the computer's memory out of something that can only hold a value for a fraction of a second. In fact, DRAMs are both more complicated and slower than SRAMs.

The reason that DRAMs are used is simple: they are much cheaper and take up much less space, typically 1/4 the silicon area of SRAMs or less. To build a 64 MB core memory from SRAMs would be very expensive. The overhead of the refresh circuit is tolerated in order to allow the use of large amounts of inexpensive, compact memory. The refresh circuitry itself is almost never a problem; many years of using DRAM has caused the design of these circuits to be all but perfected.

DRAMs are smaller and less expensive than SRAMs because SRAMs are made from four to six transistors (or more) per bit, DRAMs use only one, plus a capacitor. The capacitor, when energized, holds an electrical charge if the bit contains a "1" or no charge if it contains a "0". The transistor is used to read the contents of the capacitor. The problem with capacitors is that they only hold a charge for a short period of time, and then it fades away. These capacitors are tiny, so their charges fade particularly quickly. This is why the refresh circuitry is needed: to read the contents of every cell and refresh them with a fresh "charge" before the contents fade away and are lost. Refreshing is done by reading every "row" in the memory chip one row at a time; the process of reading the contents of each capacitor re-establishes the charge.

DRAM is manufactured using a similar process to how processors are: a silicon substrate is etched with the patterns that make the transistors and capacitors (and support structures) that comprise each bit. DRAM costs much less than a processor because it is a series of simple, repeated structures, so there isn't the complexity of making a single chip with several million individually-located transistors.

There are many different kinds of specific DRAM technologies and speeds that they are available in. These have evolved over many years of using DRAM for system memory, and are discussed in more detail in other sections.

For the last decade the CPU has been the driving element in overall system performance. Today, as we move to specialized subsystems, a balanced system will determine the ultimate system performance. As new systems emerge and CPUs are packed with ever-greater resources (super-pipelined, superscalar, with multiple execution units, branch prediction and speculative execution techniques), Intel and other semiconductor companies have been concerned that the steady stream of instructions from memory to the processor may not be able to keep pace. Multiple resource demands on the CPU mean a single cache miss can affect into the halt of several instructions and cause unstable delivery of streamed data. In addition, new engines such as graphics accelerators, I/O servers and multimedia processors live on the same system bus as the SDRAM and each can concurrently demand direct access to the memory.

Synchronous DRAM provides the performance necessary to handle these tasks, alleviating the concerns of CPU manufacturers, and will become a driving force leading computing devices to a new level of functionality.

Basic DRAM operation

A DRAM memory array can be thought of as a table of cells. These cells are comprised of capacitors, and contain one or more ‘bits’ of data, depending upon the chip configuration. This table is addressed via row and column decoders, which in turn receive their signals from the RAS and CAS clock generators. In order to minimize the package size, the row and column addresses are multiplexed into row and column address buffers. For example, if there are 11 address lines, there will be 11 row and 11 column address buffers. Access transistors called ‘sense amps’ are connected to the each column and provide the read and restore operations of the chip. Since the cells are capacitors that discharge for each read operation, the sense amp must restore the data before the end of the access cycle.

The capacitors used for data cells tend to bleed off their charge, and therefore require a periodic refresh cycle or data will be lost. A refresh controller determines the time between refresh cycles, and a refresh counter ensures that the entire array (all rows) are refreshed. Of course, this means that some cycles are used for refresh operations, and has some impact on performance.

A typical memory access would occur as follows. First, the row address bits are placed onto the address pins. After a period of time the RAS\ signal falls, which activates the sense amps and causes the row address to be latched into the row address buffer. When the RAS\ signal stabilizes, the selected row is transferred onto the sense amps. Next, the column address bits are set up, and then latched into the column address buffer when CAS\ falls, at which time the output buffer is also turned on. When CAS\ stabilizes, the selected sense amp feeds its data onto the output buffer.

OPERATING MODES

Asynchronous: Operating mode where memory responds to input signals whenever they occur and are based on a clock which operates independently of the system clock; the memory runs on its own clock.

Synchronous: Operating mode where memory responds to input signals when they are present at specific time intervals regulated by the system clock; the memory is "in synch" with the system clock.

ASYNCHRONOUS OPERATION

An asynchronous interface is one where a minimum period of time is determined to be necessary to ensure an operation is complete. Each of the internal operations of an asynchronous DRAM chip are assigned minimum time values, so that if a clock cycle occurs any time prior to that minimum time another cycle must occur before the next operation is allowed to begin.

It should be fairly obvious that all of these operations require a significant amount of time and creates a major performance concern. The primary focus of DRAM manufacturers has been to either increase the number of bits per access, pipeline the various operations to minimize the time required or eliminate some of the operations for certain types of accesses.

Wider I/O ports would seem to be the simplest and cheapest method of improving performance. Unfortunately, a wider I/O port means additional I/O pins, which in turn means a larger package size. Likewise, the additional segmentation of the array (more I/O lines = more segments) means a larger chip size. Both of these issues mean a greater cost, somewhat defeating the purpose of using DRAM in the first place. Another drawback is that the multiple outputs draw additional current, which creates ringing in the ground circuit. This actually results in a slower part, because the data cannot be read until the signal stabilizes. These problems limited the I/O width to 4 bits for quite some time, causing DRAM designers to look for other ways to optimize performance.

Synchronous Operation

Once it became apparent that bus speeds would need to run faster than 66MHz, DRAM designers needed to find a way to overcome the significant latency issues that still existed. By implementing a synchronous interface, they were able to do this and gain some additional advantages as well.

With an asynchronous interface, the processor must wait idly for the DRAM to complete its internal operations, which typically takes about 60ns. With synchronous control, the DRAM latches information from the processor under control of the system clock. These latches store the addresses, data and control signals, which allows the processor to handle other tasks. After a specific number of clock cycles the data becomes available and the processor can read it from the output lines.

Another advantage of a synchronous interface is that the system clock is the only timing edge that needs to be provided to the DRAM. This eliminates the need for multiple timing strobes to be propagated. The inputs are simplified as well, since the control signals, addresses and data can all be latched in without the processor monitoring setup and hold timings. Similar benefits are realized for output operations as well.

All DRAMs that have a synchronous interface are known generically as SDRAM. This includes CDRAM (Cache DRAM), RDRAM (Rambus DRAM), ESDRAM (Enhanced SDRAM) and others, however the type that most often is called SDRAM is the JEDEC standard synchronous DRAM.

SDRAM was initially introduced as the answer to all performance problems, however it quickly became apparent that there was little performance benefit and a lot of compatibility problems. The first SDRAM modules contained only two clock lines, but it was soon determined that this was insufficient. This created two different module designs (2-clock and 4-clock), and you needed to know which your motherboard required. Though the timings were theoretically supposed to be 5-1-1-1 @ 66MHz, many of the original SDRAM would only run at 6-2-2-2 when run in pairs, mostly because the chipsets (i430VX, SiS5571) had trouble with the speed and coordinating the accesses between modules. The i430TX chipset and later non-Intel chipsets improved upon this, and the SPD chip (serial presence detect) was added to the standard so chipsets could read the timings from the module. Unfortunately, for quite some time the SPD EEPROM was either not included on many modules, or not read by the motherboards.

SDRAM chips are officially rated in MHz, rather than nanoseconds (ns) so that there is a common denominator between the bus speed and the chip speed. This speed is determined by dividing 1 second (1 billion ns) by the output speed of the chip. For example a 67MHz SDRAM chip is rated as 15ns. Note that this nanosecond rating is not measuring the same timing as an asynchronous DRAM chip. Remember, internally all DRAM operates in a very similar manner, and most performance gains are achieved by ‘hiding’ the internal operations in various ways.

The original SDRAM modules either used 83MHz chips (12ns) or 100MHz chips (10ns), however these were only rated for 66MHz bus operation. Due to some of the delays introduced when having to deal with the various synchronization of signals, the 100MHz chips will produce a module that operates reliably at about 83MHz, in many cases. These SDRAM modules are now called PC66, to differentiate them from those conforming to Intel’s PC100 specification

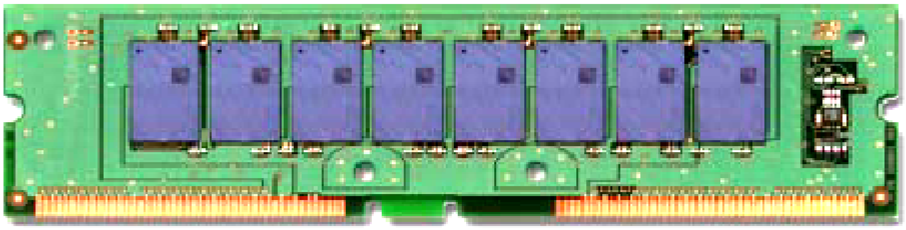
When Intel decided to officially implement a 100MHz system bus speed, they understood that most of the SDRAM modules available at that time would not operate properly above 83MHz. In order to bring some semblance of order to the marketplace, Intel introduced the PC100 specification as a guideline to manufacturers for building modules that would function properly on their upcoming i440BX. With the PC100 specification, Intel laid out a number of guidelines for trace lengths, trace widths and spacing, number of PCB layers, EEPROM programming specs, etc.

PC100 SDRAM on a 100MHz (or faster) system bus will provide a performance boost for Socket 7 systems of between 10% and 15%, since the L2 cache is running at system bus speed. Pentium II systems will not see as big a boost, because the L2 cache is running at ½ processor speed anyway, with the exception of the cache less Celeron chips of course.

RAMBUS MEMORY

RAMBUS TECHNOLOGY OVERVIEW

The explosive demands from the Internet and high-performance consumer products are driving the need for more bandwidth. As chip technology has crossed the 1 GHz boundary, the bottleneck in system design is now how fast data can be transferred between chips. Traditional component signaling is not evolving rapidly enough to keep up with these demands. The interconnect between chips inside these products is becoming a significant bottleneck in many of these products.



RAMBUS MEMORY MODULE

The adoption of Rambus technology will dramatically simplify computer architectures, since hardware traditionally used to increase the speed of the processor to memory interface will not be necessary. Rambus products use standard CMOS processes, low cost IC packaging and conventional PC board technologies in order to take advantage of high volume, low cost manufacturing processes. The results of these factors will enable faster, smaller and lower cost systems for computer, communications and consumer product applications.

OVERVIEW - RAMBUS TECHNOLOGY.

Intel Corporation has selected memory technology designed and licensed by Rambus Inc. to power the main memory platform for high-performance PC systems using Pentium-III and future processors.

Rambus is high-performance, chip-to-chip interface technology that enables semiconductor memory devices to keep pace with faster generations of processors and controllers. Rambus technology is incorporated onto dynamic-random-access-memory (DRAM) chips and the logic devices that control them. Rambus Inc. boasts that this new technology delivers ten times the performance of conventional DRAMs and three times the performance of today's PC 100 SDRAM DIMM modules. A single Rambus DRAM, referred to as RDRAM, transfers data at speeds up to 800MHz over a two-byte-wide channel.

There are three generations of Rambus Technology. The first and second generations, called base and concurrent, operate at a 600MHz data transfer rate and are currently used in the entertainment industry, graphic workstations and video graphics.

The third generation is called Direct Rambus. A Direct Rambus memory module is called a RIMM. The Direct Rambus RIMM module is a general purpose high-performance memory subsystem suitable for use in a broad range of applications including computer memory in personal computers, workstations and other applications where high bandwidth and low latency are required.

RAMBUS are avaliable at 600, 700, and 800 MHz.

The high-speed clock rate of 400 MHz enables an effective data rate of 800 Mbits per second (2 bits of data are transferred per each clock cycle, data is transferred at the leading and the trailing edge of the clock.) Since the Rambus Channel is 16 bit wide [2 bytes], the resulting data transfer rate is up to 1.6 GBytes per second per channel (2 x 800MB/sec = 1.6GB). There are Intel platforms that use more than one channel in the architecture.

What is a RIMM module?
Rambus trademarked the term, RIMM, as an entire word. It is the term used for a module using Rambus technology. It DOES NOT mean Rambus Inline Memory Module. RDRAM is the memory chip attached on the RIMM module.

The Direct Rambus™ RIMM™ Module is a high-performance plug-in memory module for PC main memory Developed in conjunction with Intel Corporation, Direct Rambus technology has the performance/cost ratio demanded by the high clock-rate microprocessors used in mainstream PCs starting to ship in 1999.
The RIMM module conforms to the standard DIMM form factor, but it is not pin-compatible. Its architecture is based on the electrical requirements of the Direct Rambus Channel, a high-speed bus operating at a clock rate of 400MHz which enables a data rate of 800MHz (data is clocked on both clock edges). A two byte-wide data channel is used resulting in a peak data transfer rate of 1.6 Gbytes per second. The bus uses transmission line characteristics to maintain high signal integrity.
Up to three RIMM modules may be used on a PC desktop motherboard. The Rambus Channel extends from the controller through each RIMM module in a continuous flow until the Channel termination is reached. Low-cost continuity modules are used to maintain Channel integrity in systems having less than three RIMM modules.
An on-board SPD (Serial Presence Detect) PROM chip is used to provide initialization information to the system processor on power-up. This technique assures compatibility across all Direct Rambus RDRAM manufacturing partners producing various density DRAM devices.

Heat Spreader

What is a heat spreader and why does a RIMM need one?

While a RIMM module does not dissipate more heat than a comparable SDRAM module, the RIMM Module can develop hot spots depending on the application. The heat spreader cover plate helps minimize hot spots as well as provides protection to the RDRAMs on the module.
What is the Continuity RIMM Module? Why is it needed in empty connector slots? For the Rambus memory system to operate properly, the signal traces connect through the memory controller chipset, the RIMM connectors to special “termination” devices on the motherboard. The Continuity RIMM Module is used to allow the signals to cross connectors that have no memory module installed.

Which type of systems take Rambus memory?
Initially Rambus will be on high end corporate desktops and workstations. Lower end PCs and higher end servers will take longer to incorporate the technology.

Rambus Memory Architecture Benefits

High Performance

- 1.6GB per second of peak bandwidth
- Multiple Channels can be used for even higher performance and bandwidth
- An individual 800MHz RDRAM device offers over ten times the bandwidth than a 66MHz X 16 SDRAM device
- A 64MB Rambus system has three times the effective bandwidth over a 64MB 64-bit wide 100MHz SDRAM system

Cost-competitive

- Uses conventional DRAM core; the 64MB RDRAM is comparable in die size to a 16 X 64Mb* SDRAM
- Low-cost industry-standard memory modules and connectors
- Uses existing industry-standard FR04 printed circuit board technology

Low Power consumption

- RDRAMs include low power modes that reduce overall power consumption
- Rambus devices inherently use less energy per byte transferred
- Configurations support six times the bandwidth of EDO DRAMs at comparable power consumption

Expandable/Granular

-

-
- 32Mb*, 64Mb*, 128Mb*, 256Mb*, 512Mb*, 1Gb* generations of RDRAMs are functionally and electronically compatible
- Memory can be incremented by a single RDRAM
- A single Channel supports 32 RDRAMs; expansion buffers allow support of additional 32 RDRAMs; a controller can support multiple Channels; using 256Mb* devices and repeaters, memory can be extended to 64Gb

Reduced Risk—Quick to Market

- Proven technology that ships in high volume in PCs and consumer products
- Support by the leading DRAM suppliers, assuring OEMs of an ample supply of RDRAMs
- Multiple sources for Rambus-compatible connectors, modules, clock chips and test systems
- A “Cookbook Solution” is provided to the system designer

Industry Standard Interface

- Rambus Validation Program ensures industry-wide compliance of all memory devices and modules
- All Rambus-based ICs are compatible
- Rambus-compatible modules, connectors and clock chips meet Rambus engineering and test specifications
- Intel® is developing chip sets for mainstream PCs to start shipping in 1999
- Kingston Technology will manufacture and distribute Rambus RIMM modules coinciding with chip set and RDRAM availability in 1999

What applications will require Rambus memory?

Internet applications, applications with Streaming SIMD Extensions, data visualization, streaming audio, photo digital editing, video capture, compression and decompression, video and speech recognition applications and any future applications requiring the additional headroom Rambus memory delivers.

Double Data Rate SDRAM (DDR SDRAM)

Only a few years ago, "regular" SDRAM was introduced as a proposed replacement for the older FPM and EDO asynchronous DRAM technologies. This was due to the limitations the older memory has when working with systems using higher bus speeds (over 75 MHz). In the next couple of years, as system bus speeds increase further, the bell will soon toll on SDRAM itself. One of the proposed new standards to replace SDRAM is Double Data Rate SDRAM or DDR SDRAM.

DDR SDRAM is similar in function to regular SDRAM, but doubles the bandwidth of the memory by transferring data twice per cycle--on both the rising and falling edges of the clock signal. The clock signal transitions from "0" to "1" and back to "0" each cycle; the first is called the "rising edge" and the second the "falling edge". Normally only one of these is used to trigger a data transfer; with DDR SDRAM both are used. Does this technique sound familiar? It is also used by the new AGP technology to double performance over the older PCI bus technology.

Direct Rambus DRAM (DRDRAM)

One of the two main competing standards to replace SDRAM is called Direct Rambus DRAM or DRDRAM (formerly called just "Rambus DRAM" or "RDRAM"). Unlike DDR SDRAM or SLDRAM, which are evolutionary designs based on regular SDRAM, DRDRAM is revolutionary design. It has received a lot of attention because of Intel's decision to pursue this technology for use in its future chipsets, in cooperation with its initial developer, a company unsurprisingly named Rambus.

DRDRAM works more like an internal bus than a conventional memory subsystem. It is based around what is called the Direct Rambus Channel, a high-speed 16-bit bus running at a clock rate of 400 MHz. As with DDR SDRAM, transfers are accomplished on the rising and falling edges of the clock, yielding an effective theoretical bandwidth of approximately 1.6 GBytes/second. This is an entirely different approach to the way memory is currently accessed over a wide 64-bit memory bus. It may seem counterproductive to narrow the channel since that reduces bandwidth, however the channel is then capable of running at much higher speeds than would be possible if the bus were wide. As with SDRAM, DRDRAM makes use of a serial presence detect (SPD) chip to tell the motherboard certain characteristics of the DRDRAM module when the system is booted. DRDRAM is proprietary, and is being designed to use a special type of module called a Rambus Inline Memory Module, or RIMM.

Rambus memory may become the next standard for future PCs, but the jury is still out. As with all new technology competitions, often marketing wins out over engineering. There is some concern that DRDRAM may not even be the best solution for systems in the future. In particular, some folks are unhappy about the prospects of having to pay licensing fees to Intel and Rambus to use the technology; recall that this requirement was one reason why the MCA bus standard died. Furthermore, some say that SLDRAM is a solution that is less revolutionary, providing the same (or more) improvements in performance with fewer radical changes required to the system architecture. Meanwhile, Intel is proceeding with plans to use the technology, so we will have to see what happens in 1999 and beyond.

Synchronous-Link DRAM (SLDRAM)

The main "competition" to the proposed DRDRAM standard is a new standard called Synchronous-Link DRAM or SLDRAM. This new technology is being developed by the SLDRAM Consortium, a group of about 20 major computer industry manufacturers, working to establish SDRAM as the next standard for high-speed PC memory.

SLDRAM is an evolutionary design that greatly improves the performance of the memory subsystem over SDRAM, without a completely new architecture such as that used by DRDRAM. The initial specifications for SLDRAM call for a 64-bit bus running at a 200 MHz clock speed. As with DDR SDRAM, transfers are made twice on each clock cycle, for an effective speed of 400 MHz. This yields a net theoretical bandwidth of about 3.2 Gbytes/second, double that of DRDRAM. Finally, SLDRAM is an open standard, meaning that no royalties need be paid to anyone in order to make use of it.

Interestingly enough, the DRDRAM and SLDRAM battle seems to be playing out in a manner similar to many prior technological skirmishes. One that comes immediately to mind is the fight for dominance between BEDO and SDRAM in the mid-90s; though many thought that BEDO was better technologically, Intel single-handedly sealed its fate by deciding to go with SDRAM instead. Today, we have Intel going with DRDRAM, against a consortium of companies trying to push SLDRAM as a better solution. However, as we enter 1999 we have more non-Intel choices in processors and chipsets than we did in 1996, so it is not clear at all if Intel will have its way in establishing DRDRAM over SLDRAM as the next standard. Another factor that will support SLDRAM is that it does not require the payment of royalties

the way DRDRAM does, something that could seriously harm the DRDRAM camp despite the presence of Intel.

Video RAM (VRAM) and Other Video DRAM Technologies

Modern video adapters use their own, specialized RAM that is separated from the main system memory. The demands placed on video memory are far greater than those placed on system memory. In addition to the video image being accessed and changed by the processor on a continual basis (many times a second when you are running a game for instance), the video card also must access the memory contents between 50 and 100 times per second to display the information on the monitor. Video cards have therefore spawned the creation of several new, innovative memory technologies, many of them designed to allow the memory to be accessed by the processor and read by the video card's refresh circuitry simultaneously. This is called dual porting and is found on Video RAM or VRAM memory. Cards using this type of memory are faster and more expensive than ones using FPM or EDO DRAM.

In addition to VRAM, several other new memory technologies and designs have evolved to maximize performance with video cards.

Memory Errors

Memory is an electronic storage device, and all electronic storage devices have the potential to incorrectly return information different than what was originally stored. Some technologies are more likely than others to do this. DRAM memory, because of its nature, is likely to return occasional memory errors. DRAM memory stores ones and zeros as charges on small capacitors that must be continually refreshed to ensure that the data is not lost. This is less reliable than the static storage used by SRAMs.

Every bit of memory is either a zero or a one, the standard in a digital system. This in itself helps to eliminate many errors, because slightly distorted values are usually recoverable. For example, in a 5 volt system, a "1" is +5V and a "0" is 0V. If the sensor that is reading the memory value sees +4.2V, it knows that this is really a "1", even though the value isn't +5V. Why? Because the only other choice would be a "0" and 4.2 is much closer to 5 than to 0. However, on rare occasions a+5V might be read as +1.9V and be considered a "0" instead of a "1". When this happens, a memory error has occurred.

There are two kinds of errors that can typically occur in a memory system. The first is called a repeatable or hard error. In this situation, a piece of hardware is broken and will consistently return incorrect results. A bit may be stuck so that it always returns "0" for example, no matter what is written to it. Hard errors usually indicate loose memory modules, blown chips, motherboard defects or other physical problems. They are relatively easy to diagnose and correct because they are consistent and repeatable.

The second kind of error is called a transient or soft error. This occurs when a bit reads back the wrong value once, but subsequently functions correctly. These problems are, understandably, much more difficult to diagnose! They are also, unfortunately, more common. Eventually, a soft error will usually repeat itself, but it can take anywhere from minutes to years for this to happen. Soft errors are sometimes caused by memory that is physically bad, but at least as often they are the result of poor quality motherboards,

memory system timings that are set too fast, static shocks, or other similar problems that are not related to the memory directly. In addition, stray radioactivity that is naturally present in materials used in PC systems can cause the occasional soft error. On a system that is not using error detection, transient errors often are written off as operating system bugs or random glitches.

The exact rate of errors returned by modern memory is a matter of some debate. It is agreed that the DRAMs used today are far more reliable than those of five to ten years ago. This has been the chief excuse used by system vendors who have dropped error detection support from their PCs. However, there are factors that make the problem worse in modern systems as well. First, more memory is being used; 10 years ago the typical system had 1 MB to 4 MB of memory; today's systems usually have 16 MB to 64 MB--or much more, since RAM prices have fallen dramatically in the last three years. Second, systems today are running much faster than they used to; the typical memory bus is running from 3 to 10 times the speed of those of older machines. Finally, the quality level of the average PC is way down from the levels of 10 years ago. Cheaply thrown-together PCs, made by assembly houses whose only concern is to get the price down and the machine out the door, often use RAM of very marginal quality.

Regardless of how often memory errors occur, they do occur. How much damage they create depends on when they happen and what it is that they get wrong. If you are playing your favorite game and one of the bits controlling the color of the pixel at screen location (520, 277) is inverted from a one to a zero on one screen redraw, who cares, right? However, if you are defragmenting your hard disk and the memory location containing information to be written to the file allocation table is corrupted, it's a whole different ball game...

The only true protection from memory errors is to use some sort of memory detection or correction protocol. (Well, that's not totally true. The other form of protection is prevention: buying quality components and not abusing or neglecting your system.) Some protocols can only detect errors in one bit of an eight-bit data byte; others can detect errors in more than one bit automatically. Others can both detect and correct memory problems, seamlessly.

Causes

The memory errors that your PC is likely to suffer fall into two broad classes, soft errors and hard errors. Either can leave you staring at an unflinching screen, sometimes but not always emblazoned with a cryptic message that does nothing to help you regain the hours’ work irrevocably lost. The difference between them is transience. Soft errors are little more than disabling glitches that disappear as fast as they come. Hard errors linger until you take a trip to the repair shop.

Soft Errors

For your PC, a soft memory error is an unexpected and unwanted change. Something in memory turn up different than it is supposed to be. One bit in a memory chip may suddenly, randomly change state. Or a glitch of noise inside your system may get stored as if it were valid data. In either case, one bit becomes something other than what it's supposed to be, possibly changing an instruction in a program or a data value.

With a soft error, the change appears in your data rather than hardware. Replace or restore the erroneous data or program code, and your system will operate exactly as it always has. In general, your system needs nothing more than a reboot—a cold boot being best to gain the assurance of your PC's self-test of its circuits (including memory). The only damage is the time you waste retracing your steps to get back to the place in your processing at which the error occurred. Soft errors are the best justification for the sage advice, "Save often."

Most soft errors result from problems either within memory chips themselves or in the overall circuitry of your PC. The mechanisms behind these two types of soft errors is entirely different.

Chip-Level Errors

The errors inside memory chips are almost always a result of radioactive decay. The problem is not nuclear waste (although nuclear waste is a problem) but something even more devious. The culprit is the epoxy of the plastic chip package, which like most materials may contain a few radioactive atoms. Typically, one of these minutely radioactive atoms will spontaneously decay and shoot out an alpha particle into the chip. (There are a number of radioactive atoms in just about everything—they don't amount to very much but they are there. And by definition, a radioactive particle will spontaneously decay sometime.) An alpha particle is a helium nucleus, two protons and two neutrons, having a small positive charge and a lot of kinetic energy. If such a charged particle hits a memory cell in the chip, the charge and energy of the particle can cause a cell to change state, blasting the memory bit it contains to a new and different value. This miniature atomic blast is not enough to damage the silicon structure of the chip itself, however.

Whether a given memory cell will suffer this kind of soft error is unpredictable, just as predicting whether a given radioactive atom will decay is unpredictable. When you deal with enough atoms, however, this unpredictability becomes a probability, and engineers can predict how often one of the memory cells in a chip will suffer such an error. They just can't predict which one.

In the early days of PCs, radioactive decay inside memory chips was the most likely cause of soft errors in computers. Thanks to improved designs and technology, each generation of memory chip has become more reliable no matter whether you measure per bit or per chip. For example, any given bit in a 16Kb might suffer a decay-caused soft error every billion or so hours. The likelihood that any given bit in a modern 16Mb chip will suffer an error is on the order of once in two trillion hours. In other words, modern memory chips are about 5,000 times more reliable than those of first generation PCs, and the contents of each cell is about 5 million times more reliable once you take into account that chip capacities have increased a thousand-fold. Although conditions of use influence the occurrence of soft errors, the error rate of modern memory is such that a typical PC with 8MB of RAM would suffer a decay-caused soft error once in ten to thirty years. The probability is so small that many computer makers now ignore it.

System Level Errors

Sometimes the data traveling though your PC gets hit by a noise glitch. If a pulse of noise is strong enough and occurs at an especially inopportune instant, it can be misinterpreted by your PC as a data bit. Such a system level error will have the same effect on your PC as a soft error in memory. In fact, some system level errors may be reported as memory errors, for example when the glitch appears in the circuitry between your PC's memory chips and the memory controller.

The most likely place for system level soft errors to occur is on your PC's buses. A glitch on a data line can cause your PC to try to use or execute a bad bit of data or program code, causing an error. Or your PC could load the bad value into memory, saving it to relish (and crash from) at some later time. A glitch on the address bus will make your PC similarly find the wrong bit or byte, and the unexpected value may have exactly the same effects as a data bus error.

The probability of a system level error occurring depends on the design of your PC. A careless designer can leave your system not only susceptible to system level errors but even prone to generating the glitches that cause them. Pushing a PC design to run too fast is particularly prone to causing problems. You can do nothing to prevent system level soft errors other than choose your PC wisely.

Hard Errors

When some part of a memory chip actually fails, the result is a hard error. For instance, a jolt of static electricity can wipe out one or more memory cells. As a result, the initial symptom is the same as a soft error—a memory error that may cause an error in the results you get or a total crash of your system. The operative difference is that the hard error doesn't go away when you reboot your system. In fact, your machine may not pass its memory test when you try to start it up again. Alternately, you may encounter repeated, random errors when a memory cell hovers between life and death.

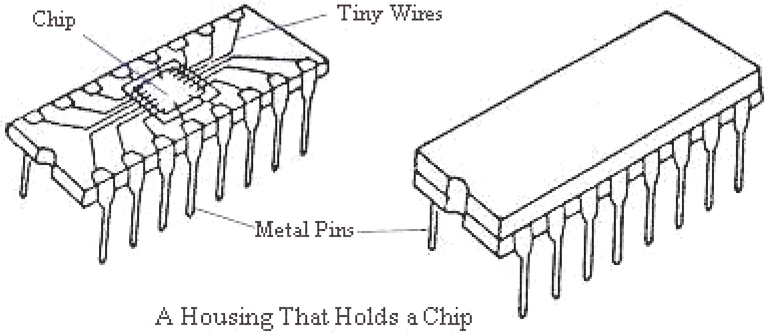
Hard errors require attention. The chip or module in which the error originates needs to be replaced.

Note, however, that operating memory beyond its speed capability often causes the same problem as hard errors. In fact, operating memory beyond its ratings causes hard errors. You can sometimes clear up such problems by adding wait states to your system's memory cycles, a setting many PCs allow you to control as part of their advanced setup procedure. This will, of course, slow down the operation of your PC so that it can accommodate the failing memory. The better cure is to replace the too-slow memory with some that can handle the speed.

Detection and Prevention

Most PCs check every bit of their memory for hard errors every time you switch your system on or perform a cold boot, although some PCs give you the option of bypassing this initial memory check to save time. Soft errors are another matter entirely. They rarely show up at boot time. Rather, they are likely to occur at the worst possible moment—which means just about any time you're running your PC. PC makers use two strategies to combat memory errors, parity and detection /correction. Either one will assure the integrity of your system's memory. Which is best—or whether you need any error compensation at all—is a personal choice.

Dual Inline Packages (DIPs) and Memory Modules



Most memory chips are packaged into small plastic or ceramic packages called dual inline packages or DIPs. A DIP is a rectangular package with rows of pins running along its two longer edges. These are the small black boxes you see on SIMMs, DIMMs or other larger packaging styles. The DIP has been the standard for packaging integrated circuits since the invention of the PC, and in fact the earliest processors were also packaged as (large) DIPs.

Older computer systems used DIP memory directly, either soldering it to the motherboard or placing it in sockets that had been soldered to the motherboard. At that time most systems had a small amount of memory (less than one megabyte) and this was the simplest way to do things. However, this arrangement caused many problems. Chips directly soldered onto the motherboard would mean the entire motherboard had to be trashed if any of the memory chips ever went bad.

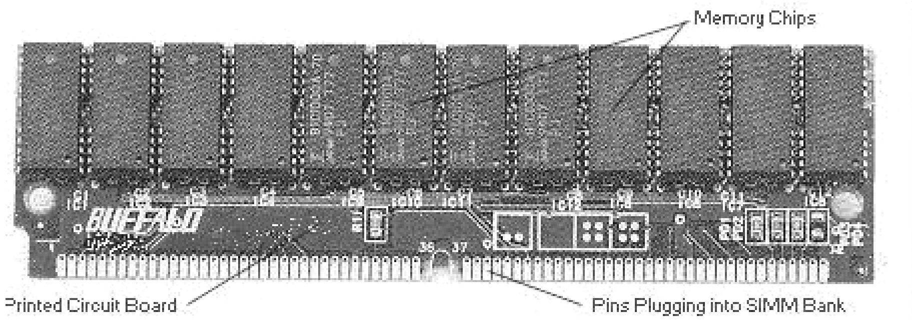
Chips inserted into sockets suffered reliability problems as the chips would (over time) tend to work their way out of the sockets. Due to thermal contraction and expansion as the machine was turned on and off, the chips would actually slowly come loose, a process called chip creep. Anyone who has worked at keeping an old XT running for many years probably remembers opening up the box and pushing all the memory chips back into their sockets with their thumbs to fix a memory problem. Dealing with individual chips also made upgrading or troubleshooting difficult.

Newer systems do not use DIP memory packaging directly. The DIPs are soldered onto small circuit boards called memory modules; the two most common being the single inline memory module or SIMM and the dual inline memory module or DIMM. The circuit boards are inserted into special sockets on the motherboard that are designed to eliminate the chip creep problem. This arrangement makes for better reliability and easier installation. Also, since SIMMs and DIMMs are (for most PCs) industry standard, it makes upgrades much simpler as well.

Standard and Proprietary Memory Modules

The three common sizes of memory modules (30-pin and 72-pin SIMMs and 168-pin DIMMs) are fortunately pretty close to being an industry standard. The vast majority of PCs use the "standard" or generic type of SIMM/DIMM. This gives the machine's owner the flexibility to shop the market and get the best deal on new memory.

Single Inline Memory Modules (SIMMs)



The single inline memory module or SIMM is still the most common memory module format in use in the PC world, largely due to the enormous installed base of PCs that use them (in new PCs, DIMMs are now overtaking SIMMs in popularity.) SIMMs are available in two flavors: 30 pin and 72 pin. 30-pin SIMMs are the older standard, and were popular on third and fourth generation motherboards. 72-pin SIMMs are used on fourth, fifth and sixth generation PCs.

SIMMs are placed into special sockets on the motherboard created to hold them. The sockets are specifically designed to ensure that once inserted, the SIMM will be held in place tightly. SIMMs are secured into their sockets (in most cases) by inserting them at an angle (usually about 60 degrees from the motherboard) into the base of the socket and then tilting them upward until they are perpendicular to the motherboard. Special metal clips on either side of the socket snap in place when the SIMM is inserted correctly. The SIMM is also keyed with a notch on one side, to make sure it isn't put in backwards.

The 30 pin SIMMs are generally available in sizes from 1 to 16 MB. Each one has 30 pins of course, and provides one byte of data (8 bits), plus 1 additional bit for parity with parity versions. 72-pin SIMMs provide four bytes of data at a time (32 bits) plus 4 bits for parity/ECC in parity/ECC versions.

SIMMs are available in two styles: single-sided or double-sided. This refers to whether or not DRAM chips are found on both sides of the SIMM or only on one side. 30-pin SIMMs are all (I am pretty sure) single-sided. 72-pin SIMMs are either single-sided or double-sided. Some double-sided SIMMs are constructed as composite SIMMs. Internally, they are wired as if they were actually two single-sided SIMMs back to back. This doesn't change how many bits of data they put out or how many you need to use. However, some motherboards cannot handle composite SIMMs because they are slightly different electrically.

72-pin SIMMs that are 1 MB, 4 MB and 16 MB in size are normally single-sided, while those 2 MB, 8 MB and 32 MB in size are generally double-sided. This is why there are so many motherboards that will only work with 1 MB, 4 MB and 16 MB SIMMs. You should always check your motherboard to see what sizes of SIMMs it supports. Composite SIMMs will not work in a motherboard that doesn't support them. SIMMs with 32 chips on them are almost always composite.

Warning: Lately, some 16 MB and 64 MB SIMMs have been seen that are composite. These can cause significant problems with some motherboards, since they are specified to support 16 MB SIMMs on the expectation that 16 MB SIMMs will all be single-sided. You may not be able to use double-sided 16 MB SIMMs in some systems, especially older or cheaper ones.

Most motherboards support either 30-pin or 72-pin SIMMs, but not both. Some 486 motherboards do support both, however. In many cases these motherboards have significant restrictions on how these SIMMs can be used. For example, only one 72-pin socket may be usable if the 30-pin sockets are in use, or double-sided SIMMs may not be usable.

Dual Inline Memory Modules (DIMMs)

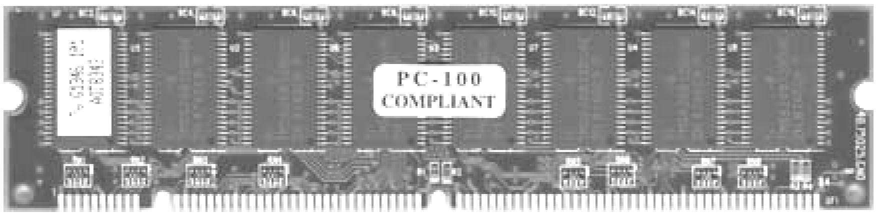
The dual inline memory module or DIMM is a newer memory module, intended for use in fifth- and sixth-generation computer systems. DIMMs are 168 pins in size, and provide memory 64 bits in width. They are a newer form factor and are becoming the de facto standard for new PCs; they are not used on older motherboards. They are also not generally available in smaller sizes such as 1 MB or 4 MB for the simple reason that newer machines are rarely configured with such small amounts of system RAM.

Physically, DIMMs differ from SIMMs in an important way. SIMMs have contacts on either side of the circuit board but they are tied together. So a 30-pin SIMM has 30 contacts on each side of the circuit board, but each pair is connected. This gives some redundancy and allows for more forgiving connections since each pin has two pads. This is also true of 72-pin SIMMs. DIMMs however have different connections on each side of the circuit board. So a 168-pin DIMM has 84 pads on each side and they are not redundant. This allows the packaging to be made smaller, but makes DIMMs a bit more sensitive to correct insertion and good electrical contact.

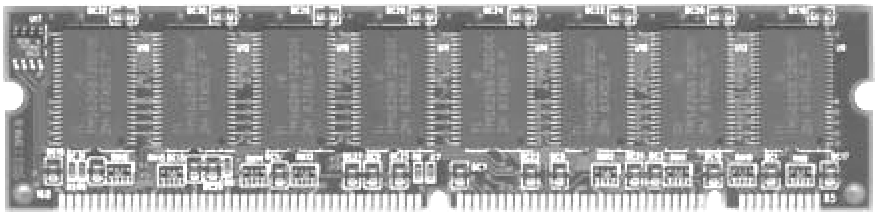
DIMMs are inserted into special sockets on the motherboard, similar to those used for SIMMs. They are generally available in 8 MB, 16 MB, 32 MB and 64 MB sizes, with larger DIMMs also available at a higher cost per megabyte. DIMMs are the memory format of

choice for the newest memory technology, SDRAM. DIMMs are also used for EDO and other technologies as well.

DIMMs come in different flavors, and it is important to ensure that you get the right kind for the machine that you are using. They come in two different voltages: 3.3V and 5.0V, and they come in either buffered or unbuffered versions. This yields of course a total of four different combinations. The standard today is the 3.3 volt unbuffered DIMM, and most machines will use these. Consult your motherboard or system manual. A smaller version of the DIMM is also sometimes seen; called the small outline DIMM or SODIMM, these packages are used primarily in laptop computers where miniaturization is key.



Front Side



Back Side

Memory Banks and Package Bit Width

As discussed in the section on memory buses in the memory section and processor section, data from the memory flows to and from the processor along the data bus. The width of the data bus dictates how much information can flow in each clock cycle. In order to take advantage of the full width of the processor's data bus, it is necessary to arrange the system memory so that each clock cycle, the full data bus width can be transferred at once. In fact, most systems require the system memory to be arranged so that this is the case.

A quantity of memory that is wide enough to match the bit width of the data bus is called a bank of memory. Most of today's PCs have a data bus width of 32 bits (fourth generation processors) or 64 bits (fifth and sixth generation CPUs). A computer will not read a partial bank of memory; the result of setting up a partial bank ranges from the memory in it being ignored, to the system not booting at all. The PC definitely will not start if the first bank is incomplete, since then it has no usable memory at all.

Most PCs have room for more than one bank of memory; some support two banks, some three or more. Banks are usually numbered starting from zero, although sometimes starting with bank one. The lowest-numbered bank should always be filled first, and they should always be filled sequentially.

Each of the different types of memory modules arranges its memory so that a certain bit width can be accessed simultaneously. 30-pin SIMMs have a width of 8 data bits, 72-pin SIMMs have 32 data bits, and DIMMs have 64 bits. In addition, when parity is used, an extra bit is added for error detection. So 30-pin parity SIMMs have 9 bits, 72-pin parity or ECC SIMMs 36, and parity or ECC DIMMs 72 bits. Each module can be made up of various types of DRAM chips, as long as the right width is maintained.

Choosing memory packaging is an exercise in matching the width of the packaged RAM to the data bus width of the processor to make sure that a full bank of memory is provided. Fortunately, this is not as difficult as it sounds. The table below shows how this works (the 8088 and 8086 are not shown since they used individual memory chips, not SIMMs):

Processor Family	Data Bus Width (bits)	Non-Parity Bank Size (bits)	Parity/ECC Bank Size (bits)	30-Pin SIMMs Per Bank	72-Pin SIMMs Per Bank	168-Pin DIMMs Per Bank
80286, 80386SX	16	16	18	2	--	--
80386DX, 80486DX, 80486SX, 80486DX2, 80486DX4, AMD 5x86, Cyrix 5x86, Pentium OverDrive for 486s	32	32	36	4	1	--
Pentium, Pentium OverDrive for Pentiums, Pentium with MMX, Pentium with MMX OverDrive, 6x86, K5, Pentium Pro, Pentium II, K6, 6x86MX	64	64	72	--	2	1

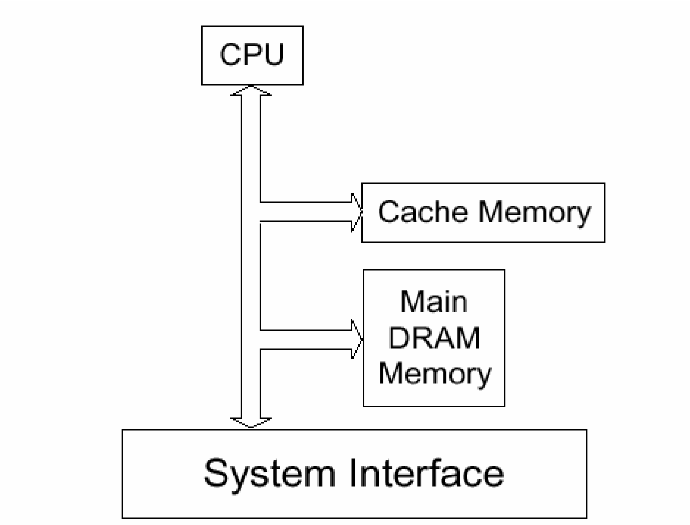
Note that a PC with a 64 bit data bus could use 8 30-pin SIMMs, except that this older technology is not supported on these newer machines; too much motherboard "real estate" is required with 30-pin SIMMs. Also, a 486 motherboard could actually make use of a single 168-pin DIMM to make up 2 banks of memory since the DIMM is 64 bits and the motherboard 32, but in practice this isn't done.

As you can see, Pentium-class and later PCs require two 72-pin SIMMs to make up a single bank. This is why you are always told to use a pair of SIMMs when buying memory for these machines. While this is generally true, there are in fact some Pentium motherboards that don't require a pair of 72-pin SIMMs. How is this possible? Basically, the chipset "cheats" by doing two consecutive accesses to 32 bits of memory at a time, allowing these machines to use a 32-bit bank size. This is a non-standard setup and leads to lower performance. It is found generally in older designs and is done mostly as a corner-cutting measure. In doing this, the bandwidth of the memory is cut in half for really no good reason. All else being equal, these motherboards should generally be avoided.

You should always use identical SIMMs when you require more than one to comprise a bank. Using different brands or speeds, or SIMMs with different types or quantities of DRAM chips, can cause motherboard system timing problems.

System Cache

Concepts of cache memory: When CPU accesses a piece of information in memory; there is a high possibility that this data will be accessed again. Rather than bring this from memory all the time, it is saved in a special, high-speed, on chip memory for CPU to use it later. This is called temporal locality. When a piece of data is accessed, it is also very likely that the data in nearby memory location will be accessed. This is called spatial locality



The system cache is responsible for a great deal of the system performance improvement of today's PCs. The cache is a buffer of sorts between the very fast processor and the relatively slow memory that serves it. (The memory is not really slow, it's just that the processor is much faster.) The presence of the cache allows the processor to do its work while waiting for memory far less often than it otherwise would.

There are in fact several different "layers" of cache in a modern PC, each acting as a buffer for recently-used information to improve performance, but when "the cache" is mentioned without qualifiers, it normally refers to the "secondary" or "level 2" cache that is placed between the processor and system RAM. The various levels of cache are discussed here, in the discussion on the theory and operation behind cache (since many of the principles are the same). However, most of the focus of this section is on the level 2 system cache.

How the CACHE concept come into existence.

In early PCs, the various components had one thing in common: they were all really slow. The processor was running at 8 MHz or less, and taking many clock cycles to get anything done. It wasn't very often that the processor would be held up waiting for the system memory, because even though the memory was slow, the processor wasn't a speed demon either. In fact, on some machines the memory was faster than the processor.

In the 15 or so years since the invention of the PC, every component has increased in speed a great deal. However, some have increased far faster than others. Memory, and memory subsystems, are now much faster than they were, by a factor of 10 or more. However a current top of the line processor has performance over 1,000 times that of the original IBM PC!

This disparity in speed growth has left us with processors that run much faster than everything else in the computer. This means that one of the key goals in modern system design is to ensure that to whatever extent possible, the processor is not slowed down by the storage devices it works with. Slowdowns mean wasted processor cycles, where the CPU can't do anything because it is sitting and waiting for information it needs. We want it so that when the processor needs something from memory, it gets it as soon as possible.

The best way to keep the processor from having to wait is to make everything that it uses as fast as it is. Wouldn't it be best just to have memory, system buses, hard disks and CD-ROM drives that just went as fast as the processor? Of course it would, but there's this little problem called "technology" that gets in the way.

Actually, it's technology and cost; a modern 2 GB hard disk costs less than $200 and has a latency (access time) of about 10 milliseconds. You could implement a 2 GB hard disk in such a way that it would access information many times faster; but it would cost thousands, if not tens of thousands of dollars. Similarly, the highest speed SRAM available is much closer to the speed of the processor than the DRAM we use for system memory, but it is cost prohibitive in most cases to put 32 or 64 MB of it in a PC.

There is a good compromise to this however. Instead of trying to make the whole 64 MB out of this faster, expensive memory, you make a smaller piece, say 256 KB. Then you find a smart algorithm (process) that allows you to use this 256 KB in such a way that you get almost as much benefit from it as you would if the whole 64 MB was made from the faster memory. How do you do this? The short answer is by using this small cache of 256 KB to hold the information most recently used by the processor. Computer science shows that in general, a processor is much more likely to need again information it has recently used, compared to a random piece of information in memory. This is the principle behind caching.

"Layers" of Cache

There are in fact many layers of cache in a modern PC. This does not even include looking at caches included on some peripherals, such as hard disks. Each layer is closer to the processor and faster than the layer below it. Each layer also caches the layers below it, due to its increased speed relative to the lower levels:

Level	Devices Cached
Level 1 Cache	Level 2 Cache, System RAM, Hard Disk / CD-ROM
Level 2 Cache	System RAM, Hard Disk / CD-ROM
System RAM	Hard Disk / CD-ROM
Hard Disk / CD-ROM	--

What happens in general terms is this. The processor requests a piece of information. The first place it looks is in the level 1 cache, since it is the fastest. If it finds it there (called a hit on the cache), great; it uses it with no performance delay. If not, it's a miss and the level 2 cache is searched. If it finds it there (level 2 "hit"), it is able to carry on with relatively little delay. Otherwise, it must issue a request to read it from the system RAM. The system RAM may in turn either have the information available or have to get it from the still slower hard disk or CD-ROM.

It is important to realize just how slow some of these devices are compared to the processor. Even the fastest hard disks have an access time measuring around 10 milliseconds. If it has to wait 10 milliseconds, a 200 MHz processor will waste 2 million clock cycles! And CD-ROMs are generally at least 10 times slower. This is why using caches to avoid accesses to these slow devices is so crucial.

Caching actually goes even beyond the level of the hardware. For example, your web browser uses caching itself, in fact, two levels of caching! Since loading a web page over the Internet is very slow for most people, the browser will hold recently-accessed pages to save it having to re-access them. It checks first in its memory cache and then in its disk cache to see if it already has a copy of the page you want. Only if it does not find the page will it actually go to the Internet to retrieve it.

Level 1 (Primary) Cache

Level 1 or primary cache is the fastest memory on the PC. It is in fact, built directly into the processor itself. This cache is very small, generally from 8 KB to 64 KB, but it is extremely fast; it runs at the same speed as the processor. If the processor requests information and can find it in the level 1 cache, that is the best case, because the information is there immediately and the system does not have to wait.

Note: Level 1 cache is also sometimes called "internal" cache since it resides within the processor.

Primary (Level 1) Cache and Cache Controller

All modern processors incorporate a small, high-speed cache right on the chip, to hold recently-used data and instructions from memory. A computer science principle called locality of reference states that if the processor recently referred to a location in memory, it is likely that it will refer to it again in the near future. Using a cache to hold recently used memory values saves the processor from going to memory each time to reload them. This provides a significant performance boost, because main memory is many times slower than the processor's cache.

The cache on the processor is called primary (or level 1) because it is the cache closest to the processor. Each time the processor requests information from memory, the cache controller on the chip uses special circuitry to first check if the memory data is already in the cache. If it is, then the system is spared a (time consuming) access to the main memory. Most computers also use a secondary (or level 2) cache, to catch some of the recently used data that doesn't fit in the smaller primary cache.

A full explanation of the value of caching, the principles behind it, the different levels of caching in a PC, and caching protocols and technologies, can be found in the section that discusses the secondary cache. The principles of operation of the primary and secondary caches, in terms of cache mapping, write policy etc. are pretty similar. The actual technology used for primary and secondary caches are of course different, as are their sizes and speeds.

The typical processor primary cache ranges in size from 8 KB to 64 KB, with larger amounts on the newer processors. Older processors (386 class and earlier) in fact have no primary cache at all. These caches are very fast because they run at the full speed of the processor and are integrated into it. In addition, most primary caches are set associative, which improves the chances of getting a "hit" on the cache.

There are two different ways that the processor can organize its primary cache: some processors have a single cache to handle both command instructions and program data; this is called a unified cache. Others have separate data and instruction caches. In some cases the capabilities of the data and instruction caches may be slightly different. For example, on the Pentium the data cache can use the write-back write policy, whereas the instruction cache is write-through only. Overall the performance difference between integrated and separate primary caches is not significant.

Level 2 (Secondary) Cache

The level 2 cache is a secondary cache to the level 1 cache, and is larger and slightly slower. It is used to catch recent accesses that are not caught by the level 1 cache, and is usually 64 KB to 2 MB in size. Level 2 cache is usually found either on the motherboard or a daughterboard that inserts into the motherboard. Pentium Pro processors actually have the level 2 cache in the same package as the processor itself (though it isn't in the same circuit where the processor and level 1 cache are) which means it runs much faster than level 2 cache that is separate and resides on the motherboard. Pentium II processors are in the middle; their cache runs at half the speed of the CPU.

Note: Level 2 cache is also sometimes called "external" cache since it resides outside the processor. (Even on Pentium Pros... it is on a separate chip in the same package as the processor.)

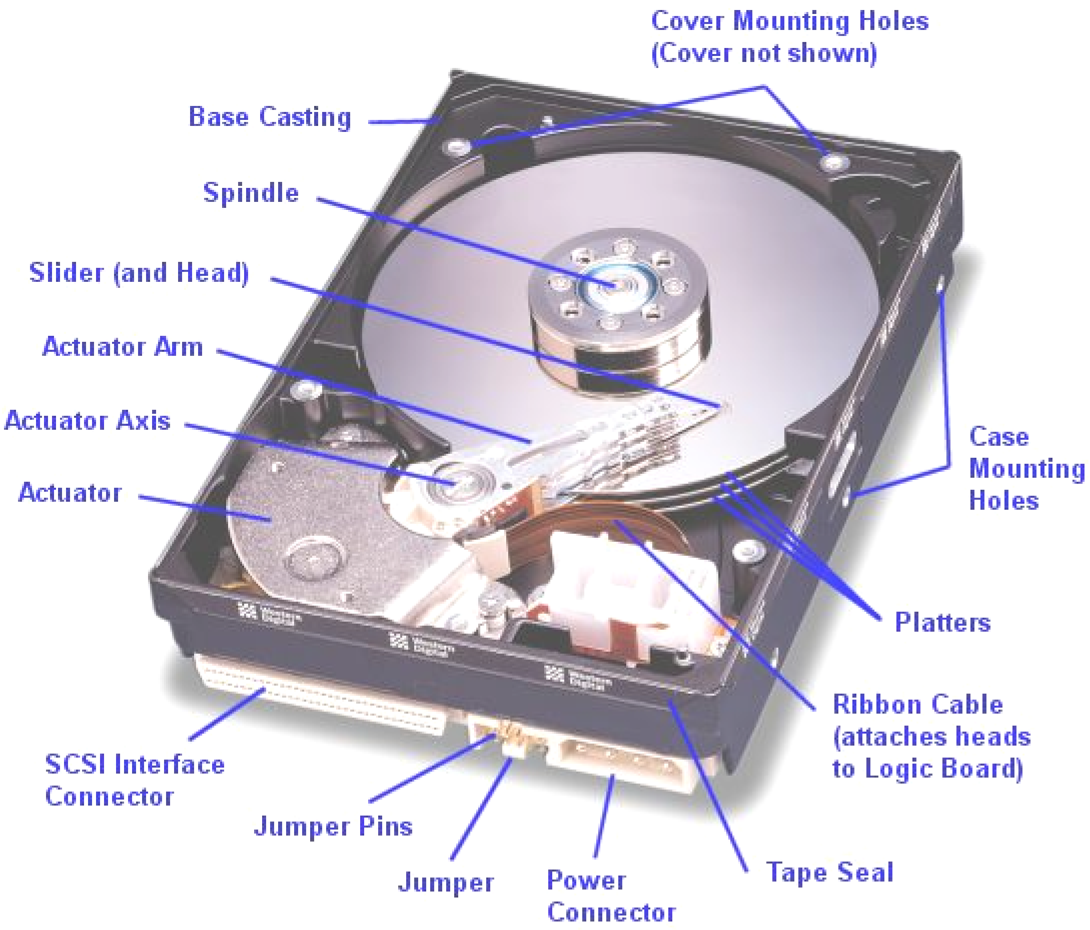
Disk Cache

A disk cache is a portion of system memory used to cache reads and writes to the hard disk. In some ways this is the most important type of cache on the PC, because the greatest differential in speed between the layers mentioned here is between the system RAM and the hard disk. While the system RAM is slightly slower than the level 1 or level 2 cache, the hard disk is much slower than the system RAM.

Unlike the level 1 and level 2 cache memory, which are entirely devoted to caching, system RAM is used partially for caching but of course for other purposes as well. Disk caches are usually implemented using software (like DOS's SmartDrive).

Hard Disk

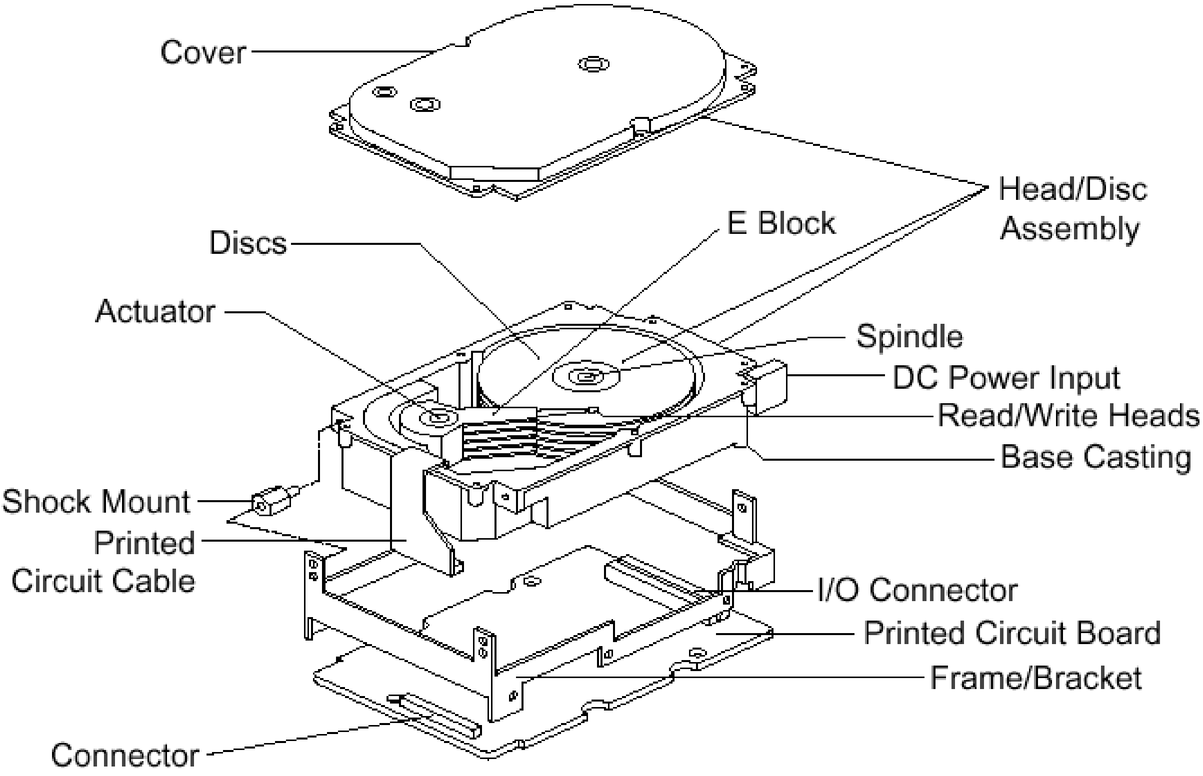
Hard Disk Operational Overview



A hard disk uses round, flat disks called platters, coated on both sides with a special media material designed to store information in the form of magnetic patterns. The platters are mounted by cutting a hole in the center and stacking them onto a spindle. The platters rotate at high speed, driven by a special spindle motor connected to the spindle. Special electromagnetic read/write devices called heads are mounted onto sliders and used to either record information onto the disk or read information from it. The sliders are mounted onto arms, all of which are mechanically connected into a single assembly and positioned over the surface of the disk by a device called an actuator. A logic board controls the activity of the other components and communicates with the rest of the PC.

Each surface of each platter on the diskan hold tens of billions of individual bits of data. These are organized into larger "chunks" for convenience, and to allow for easier and faster access to information. Each platter has two heads, one on the top of the platter and one on the bottom, so a hard disk with three platters (normally) has six surfaces and six total heads. Each platter has its information recorded in concentric circles called tracks. Each track is further broken down into smaller pieces called sectors, each of which holds 512 bytes of information.

The entire hard disk must be manufactured to a high degree of precision due to the extreme miniaturization of the components, and the importance of the hard disk's role in the PC. The main part of the disk is isolated from outside air to ensure that no contaminants get onto the platters, which could cause damage to the read/write heads.



Exploded line drawing of a modern hard disk, showing the major components.

Here's an example case showing in brief what happens in the disk each time a piece of information needs to be read from it. This is a highly simplified example because it ignores factors such as disk caching, error correction, and many of the other special techniques that systems use today to increase performance and reliability. For example, sectors are not read individually on most PCs; they are grouped together into continuous chunks called clusters. A typical job, such as loading a file into a spreadsheet program, can involve thousands or even millions of individual disk accesses, and loading a 20 MB file 512 bytes at a time would be rather inefficient:

1. The first step in accessing the disk is to figure out where on the disk to look for the needed information. Between them, the application, operating system, system BIOS and possibly any special driver software for the disk, do the job of determining what part of the disk to read.

2. The location on the disk undergoes one or more translation steps until a final request can be made to the drive with an address expressed in terms of its geometry. The geometry of the drive is normally expressed in terms of the cylinder, head and sector that the system wants the drive to read. (A cylinder is equivalent to a track for addressing purposes). A request is sent to the drive over the disk drive interface giving it this address and asking for the sector to be read.

3. The hard disk's control program first checks to see if the information requested is already in the hard disk's own internal buffer (or cache). It if is then the controller supplies the information immediately, without needing to look on the surface of the disk itself.

4. In most cases the disk drive is already spinning. If it isn't (because power management has instructed the disk to "spin down" to save energy) then the drive's controller board will activate the spindle motor to "spin up" the drive to operating speed.

5. The controller board interprets the address it received for the read, and performs any necessary additional translation steps that take into account the particular characteristics of the drive. The hard disk's logic program then looks at the final number of the cylinder requested. The cylinder number tells the disk which track to look at on the surface of the disk. The board instructs the actuator to move the read/write heads to the appropriate track.

6. When the heads are in the correct position, the controller activates the head specified in the correct read location. The head begins reading the track looking for the sector that was asked for. It waits for the disk to rotate the correct sector number under itself, and then reads the contents of the sector.

7. The controller board coordinates the flow of information from the hard disk into a temporary storage area (buffer). It then sends the information over the hard disk interface, usually to the system memory, satisfying the system's request for data.

Hard Disk Platters and Media

Every hard disk contains one or more flat disks that are used to actually hold the data in the drive. These disks are called platters (sometimes also "disks" or "discs"). They are composed of two main substances: a substrate material that forms the bulk of the platter and gives it structure and rigidity, and a magnetic media coating which actually holds the magnetic impulses that represent the data. Hard disks get their name from the rigidity of the platters used, as compared to floppy disks and other media which use flexible "platters" (actually, they aren't usually even called platters when the material is flexible.)

The platters are "where the action is"--this is where the data itself is recorded. For this reason the quality of the platters and particularly, their media coating, is critical. The surfaces of each platter are precision machined and treated to remove any imperfections, and the hard disk itself is assembled in a clean room to reduce the chances of any dirt or contamination getting onto the platters

Platter Size

The size of the platters in the hard disk is the primary determinant of its overall physical dimensions, also generally called the drive's form factor; most drives are produced in one of the various standard hard disk form factors. Disks are sometimes referred to by a size specification; for example, someone will talk about having a "3.5-inch hard disk". When this terminology is used it usually refers to the disk's form factor, and normally, the form factor is named based on the platter size. The platter size of the disk is usually the same for all drives of a given form factor, though not always, especially with the newest drives, as we will see below. Every platter in any specific hard disk has the same diameter.

The first PCs used hard disks that had a nominal size of 5.25". Today, by far the most common hard disk platter size in the PC world is 3.5". Actually, the platters of a 5.25" drive are 5.12" in diameter, and those of a 3.5" drive are 3.74"; but habits are habits and the "approximate" names are what are commonly used. You will also notice that these numbers correspond to the common sizes for floppy disks because they were designed to be mounted into the same drive bays in the case. Laptop drives are usually smaller, due to laptop manufacturers' never-ending quest for "lighter and smaller". The platters on these drives are usually 2.5" in diameter or less; 2.5" is the standard form factor, but drives with 1.8" and even 1.0" platters are becoming more common in mobile equipment.

Here are the main reasons why companies are going to smaller platters even for desktop units:

- **Enhanced Rigidity:** The rigidity of a platter refers to how stiff it is. Stiff platters are more resistant to shock and vibration, and are better suited for being mated with higher-speed spindles and other high-performance hardware. Reducing the hard disk platter's diameter by a factor of two approximately *quadruples* its rigidity.
- **Manufacturing Ease:** The flatness and uniformity of a platter is critical to its quality; an ideal platter is perfectly flat and consistent. Imperfect platters lead to low manufacturing yield and the potential for data loss due to the heads contacting uneven spots on the surface of a platter. Smaller platters are easier to make than larger ones.
- **Mass Reduction:** For performance reasons, hard disk spindles are increasing in speed. Smaller platters are easier to spin and require less-powerful motors. They are also faster to spin up to speed from a stopped position.
- **Power Conservation:** The amount of power used by PCs is becoming more and more of a concern, especially for portable computing but even on the desktop. Smaller drives generally use less power than larger ones.
- **Noise and Heat Reduction:** These benefits follow directly from the improvements enumerated above.
- **Improved Seek Performance:** Reducing the size of the platters reduces the distance that the head actuator must move the heads side-to-side to perform random seeks; this improves seek time and makes random reads and writes faster. Of course, this is done at the cost of capacity; you could theoretically achieve the same performance improvement on a larger disk by only filling the inner cylinders of each platter. In fact, some demanding customers used to partition hard disks and use only a small portion of the disk, for exactly this reason: so that seeks would be faster. Using a smaller platter size is more efficient, simpler and less wasteful than this sort of "hack".

The trend towards smaller platter sizes in modern desktop and server drives began in earnest when some manufacturers "trimmed" the platters in their 10,000 RPM hard disk drives from 3.74" to 3" (while keeping them as standard 3.5" form factor drives on the outside for compatibility.) Seagate's Cheetah X15 15,000 RPM drive goes even further, dropping the platter size down to 2.5", again trading performance for capacity (it is "only" 18 GB, less than half the size of modern 3.5" platter-size drives.) This drive, despite having 2.5" platters, still uses the common 3.5" form factor for external mounting (to maintain compatibility with standard cases), muddying the "size" waters to some extent (it's a "3.5-inch drive" but it doesn't have 3.5" platters.)

The smallest hard disk platter size available on the market today is a miniscule 1" in diameter! IBM's amazing Microdrive has a single platter and is designed to fit into digital cameras, personal organizers, and other small equipment. The tiny size of the

Platters enables the Micro drive to run off battery power, spin down and back up again in less than a second, and withstand shock that would destroy a normal hard disk. The downside? It's "only" 340 MB. :^)

Here's a summary table showing the most common platter sizes used in PCs, in order of decreasing size (which in most cases is also chronological order from their data of introduction, but not always) and also showing the most common form factors used by each technology:

Platter Diameter	Typical Form Factor	Application
5.12	5.25"	Oldest PCs, used in servers through the mid-1990s and some retail drives in the mid-to-late 1990s; now obsolete
3.74	3.5"	Standard platter size for the most common hard disk drives used in PCs
3.0	3.5"	High-end 10,000 RPM drives
2.5	2.5", 3.5"	Laptop drives (2.5" form factor); 15,000 RPM drives (3.5" form factor)
1.8	PC Card (PCMCIA)	PC Card (PCMCIA) drives for laptops
1.3	PC Card (PCMCIA)	Originally used on hand-held PCs (no longer made)
1.0	CompactFlash	Digital cameras, hand-held PCs and other consumer electronic devices

Magnetic Media

The substrate material of which the platters are made forms the base upon which the actual recording media is deposited. The media layer is a very thin coating of magnetic material which is where the actual data is stored; it is typically only a few millionths of an inch in thickness.

Older hard disks used oxide media. "Oxide" really means iron oxide--rust. Of course no high-tech company wants to say they use rust in their products, so they instead say something like "high-performance oxide media layer". :^) But in fact that's basically what oxide media is, particles of rust attached to the surface of the platter substrate using a binding agent. You can actually see this if you look at the surface of an older hard disk platter: it has the characteristic light brown color. This type of media is similar to what is used in audiocassette tape (which has a similar color.)

Oxide media is inexpensive to use, but also has several important shortcomings. The first is that it is a soft material, and easily damaged from contact by a read/write head. The second is that it is only useful for relatively low-density storage. It worked fine for older hard disks with relatively low data density, but as manufacturers sought to pack more and more data into the same space, oxide was not up to the task: the oxide particles became too large for the small magnetic fields of newer designs.

Today's hard disks use thin film media. As the name suggests, thin film media consists of a very thin layer of magnetic material applied to the surface of the platters. (While oxide media certainly isn't thick by any reasonable use of the word, it was much thicker than this new media material; hence the name "thin film".) Special manufacturing techniques are employed to deposit the media material on the platters. One method is electroplating, which deposits the material on the platters using a process similar to that used in electroplating jewelry. Another is sputtering, which uses a vapor-deposition process borrowed from the manufacture of semiconductors to deposit an extremely thin layer of magnetic material on the surface. Sputtered platters have the advantage of a more uniform and flat surface than plating. Due to the increased need for high quality on newer drives, sputtering is the primary method used on new disk drives, despite its higher cost.

Compared to oxide media, thin film media is much more uniform and smooth. It also has greatly superior magnetic properties, allowing it to hold much more data in the same amount of space. Finally, it's a much harder and more durable material than oxide, and therefore much less susceptible to damage.

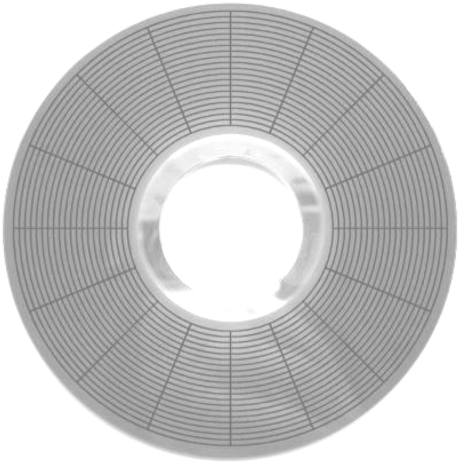
After applying the magnetic media, the surface of each platter is usually covered with a thin, protective, layer made of carbon. On top of this is added a super-thin lubricating layer. These materials are used to protect the disk from damage caused by accidental contact from the heads or other foreign matter that might get into the drive.

IBM's researchers are now working on a fascinating, experimental new substance that may replace thin film media in the years ahead. Rather than sputtering a metallic film onto the surface, a chemical solution containing organic molecules and particles of iron and platinum is applied to the platters. The solution is spread out and heated. When this is done, the iron and platinum particles arrange themselves naturally into a grid of crystals, with each crystal able to hold a magnetic charge. IBM is calling this structure a "nanocrystal super lattice". This technology has the potential to increase the areal density capability of the recording media of hard disks by as much as 10 or even 100 times! Of course it is years away, and will need to be matched by advances in other areas of the hard disk (particularly read/write head capabilities) but it is still pretty amazing and shows that magnetic storage still has a long way to go before it runs out of room for improvement.

Tracks and Sectors

Platters are organized into specific structures to enable the organized storage and retrieval of data. Each platter is broken into tracks--tens of thousands of them--which are tightly-packed concentric circles. These are similar in structure to the annual rings of a tree (but not similar to the grooves in a vinyl record album, which form a connected spiral and not concentric rings).

A track holds too much information to be suitable as the smallest unit of storage on a disk, so each one is further broken down into sectors. A sector is normally the smallest individually addressable unit of information stored on a hard disk, and normally holds 512 bytes of information. The first PC hard disks typically held 17 sectors per track. Today's hard disks can have thousands of sectors in a single track, and make use of zoned recording to allow more sectors on the larger outer tracks of the disk.



A platter from a 5.25" hard disk, with 20 concentric tracks drawn over the surface. This is far lower than the density of even the oldest hard disks; even if visible, the tracks on a modern hard disk would require high magnification to resolve. Each track is divided into 16 imaginary sectors. Older hard disks had the same number of sectors per track, but new ones use zoned recording with a different number of sectors per track in different zones of tracks.

cluster 1) In personal computer storage technology, a cluster is the logical unit of file storage on a hard disk; it's managed by the computer's operating system. Any file stored on a hard disk takes up one or more clusters of storage. A file's clusters can be scattered among different locations on the hard disk. The clusters associated with a file are kept track of in the hard disk's file allocation table (file allocation table). When you read a file, the entire file is obtained for you and you aren't aware of the clusters it is stored in.

Since a cluster is a logical rather than a physical unit (it's not built into the hard disk itself), the size of a cluster can be varied. The maximum number of clusters on a hard disk depends on the size of a FAT table entry. Beginning with DOS 4.0, the FAT entries were 16 binary digit in length, allowing for a maximum of 65,536 clusters. Beginnning with the Windows 95 OSR2 service release, a 32-bit FAT entry is supported, allowing an entry to address enough clusters to support up to two terabyte of data (assuming the hard disk is that large!).

The tradeoff in cluster size is that even the smallest file (and even a directory itself) takes up the entire cluster. Thus, a 10-byte file will take up 2,048 bytes if that's the cluster size. In fact, many operating systems set the cluster size default at 4,096 or 8,192 bytes. Until the file allocation table support in Windows 95 OSR2, the largest size hard disk that could be supported in a single partition was 512 megabyte. Larger hard disks could be divided into up to four partitions, each with a FAT capable of supporting 512 megabytes of clusters.

Hard Disk Formatting and Capacity

Two Formatting Steps

Many PC users don't realize that formatting a hard disk isn't done in a single step. In fact, three steps are involved:

1. **Low-Level Formatting:** This is the "true" formatting process for the disk. It creates the physical structures (tracks, sectors, control information) on the hard disk. Normally, this step begins with the hard disk platters "clean", containing no information.

2. **Partitioning:** This process divides the disk into logical "pieces" that become different hard disk volumes (drive letters). This is an operating system function

3. **High-Level Formatting:** This final step is also an operating-system-level command. It defines the logical structures on the partition and places at the start of the disk any necessary operating system files.

Low-Level Formatting

Low-level formatting is the process of outlining the positions of the tracks and sectors on the hard disk, and writing the control structures that define where the tracks and sectors are. This is often called a "true" formatting operation, because it really creates the physical format that defines where the data is stored on the disk. The first time that a low-level format ("LLF") is performed on a hard disk, the disk's platters start out empty. That's the last time the platters will be empty for the life of the drive. If an

LLF is done on a disk with data on it already, the data is permanently erased (save heroic data recovery measures which are sometimes possible).

Warning: You should never attempt to do a low-level format on an IDE/ATA or SCSI hard disk. Do not try to use BIOS-based low-level formatting tools on these newer drives. It's unlikely that you will damage anything if you try to do this (since the drive controller is programmed to ignore any such LLF attempts), but at best you will be wasting your time. A modern disk can usually be restored to "like-new" condition by using a zero-fill utility.

High-Level Formatting

After low-level formatting is complete, we have a disk with tracks and sectors--but nothing written on them. High-level formatting is the process of writing the file system structures on the disk that let the disk be used for storing programs and data. If you are using DOS, for example, the DOS FORMAT command performs this work, writing such structures as the master boot record and file allocation tables to the disk. High-level formatting is done after the hard disk has been partitioned, even if only one partition is to be used. See here for a full description of DOS structures, also used for Windows 3.x and Windows 9x systems.

IDE/ATA Transfer Modes and Protocols

Since performance is of utmost concern when using a hard disk, the different transfer modes and protocols that a drive (and interface) supports are very important.

Programmed I/O (PIO) Modes

The oldest method of transferring data over the IDE/ATA interface is through the use of programmed I/O.

The table below shows the five different PIO modes, along with the cycle time for each transfer and the corresponding throughput of the PIO mode:

PIO Mode	Cycle Time (nanoseconds)	Maximum Transfer Rate (MB/s)	Defining Standard
Mode 0	600	3.3	ATA
Mode 1	383	5.2	ATA
Mode 2	240	8.3	ATA
Mode 3	180	11.1	ATA-2
Mode 4	120	16.7	ATA-2

Programmed I/O is performed by the system CPU; the system processor is responsible for executing the instructions that transfer the data to and from the drive, using special I/O locations. This technique works fine for slow devices like keyboards and modems, but for performance components like hard disks it causes performance issues. Not only does PIO involved a lot of wasteful overhead, the CPU is "distracted" from its ordinary work whenever a hard disk read or write is needed. This means that using PIO is ideally suited for lower-performance applications and single tasking. It also means that the more data the system must transfer, the more the CPU gets bogged down. As hard disk transfer rates continue to increase, the load on the CPU would have continued to grow. This is the other key reason why PIO modes are no longer used on new systems, having been replaced by DMA modes, and then later, Ultra DMA

Direct Memory Access (DMA) Modes and Bus Mastering DMA

Direct memory access or DMA is the generic term used to refer to a transfer protocol where a peripheral device transfers information directly to or from memory, without the system processor being required to perform the transaction. DMA has been used on the PC for years over the ISA bus, for devices like sound cards and the floppy disk interface. Conventional DMA uses regular DMA channels, which are a standard system resource.

Several different DMA modes have been defined for the IDE/ATA interface; they are grouped into two categories. The first set of modes is single word DMA modes. When these modes are used, each transfer moves just a single word of data (a word is the techie term for two bytes, and recall that the IDE/ATA interface is 16 bits wide). There are (or were!) three single word DMA modes, all defined in the original ATA standard:

DMA Mode	Cycle Time (nanoseconds)	Maximum Transfer Rate (MB/s)	Defining Standard
Single Word Mode 0	960	2.1	ATA
Single Word Mode 1	480	4.2	ATA
Single Word Mode 2	240	8.3	ATA

Performing transfers of a single word at a time is horribly inefficient--each and every transfer requires overhead to set up the transfer. For that reason, single word DMA modes were quickly supplanted by multiword DMA modes. As the name implies, under these modes a "burst" of transfers occurs in rapid succession, one word after the other, saving the overhead of setting up a separate transfer for each word. Here are the multiword DMA transfer modes:

DMA Mode	Cycle Time (nanoseconds)	Maximum Transfer Rate (MB/s)	Defining Standard
Multiword Mode 0	480	4.2	ATA
Multiword Mode 1	150	13.3	ATA-2
Multiword Mode 2	120	16.7	ATA-2

Another important issue with DMA is that there are in fact two different ways of doing DMA transfers. Conventional DMA is what is called third-party DMA, which means that the DMA controllers on the motherboard coordinate the DMA transfers. (The "third party" is the DMA controller.) Unfortunately, these DMA controllers are old and very slow--they are basically unchanged since the earliest days of the PC. They are also pretty much tied to the old ISA bus, which was abandoned for hard disk interfaces for performance reasons. When multiword DMA modes 1 and 2 began to become popular, so did the use of the high-speed PCI bus for IDE/ATA controller cards. At that point, the old way of doing DMA transfers had to be changed.

Modern IDE/ATA hard disks use first-party DMA transfers. The term "first party" means that the peripheral device itself does the work of transferring data to and from memory, with no external DMA controller involved. This is also called bus mastering, because when such transfers are occurring the device becomes the "master of the bus". Bus mastering allows the hard disk and memory to work without relying on the old DMA controller built into the system, or needing any support from the CPU. It requires the use of the PCI bus--older buses like MCA also supported bus mastering but are no longer in common use. Bus-mastering DMA allows for the efficient transfer of data to and from the hard disk and system memory. Bus mastering DMA keeps CPU utilization low, which is the amount of work the CPU must do during a transfer.

The key technological advance introduced to IDE/ATA in Ultra DMA was double transition clocking. Before Ultra DMA, one transfer of data occurred on each clock cycle, triggered by the rising edge of the interface clock (or "strobe"). With Ultra DMA, data is transferred on both the rising and falling edges of the clock. (For a complete description of clocked data transfer and double transition clocking, see this fundamentals section.) Double transition clocking, along with some other minor changes made to the signaling technique to improve efficiency, allowed the data throughput of the interface to be doubled for any given clock speed.

In order to improve the integrity of this now faster interface, Ultra DMA also introduced the use of cyclical redundancy checking or CRC on the interface. The device sending data uses the CRC algorithm to calculate redundant information from each block of data sent over the interface. This "CRC code" is sent along with the data. On the other end of the interface, the recipient of the data does the same CRC calculation and compares its result to the code the sender delivered. If there is a mismatch, this means data was corrupted somehow and the block of data is resent. (CRC is similar in concept and operation to the way error checking is done on the system memory.) If errors occur frequently, the system may determine that there are hardware issues and thus drop down to a slower Ultra DMA mode, or even disable Ultra DMA operation.

The first implementation of Ultra DMA was specified in the ATA/ATAPI-4 standard and included three Ultra DMA modes, providing up to 33 MB/s of throughput. Several newer, faster Ultra DMA modes were added in subsequent years. This table shows all of the current Ultra DMA modes, along with their cycle times and maximum transfer rates:

Ultra DMA Mode	Cycle Time (nanoseconds)	Maximum Transfer Rate (MB/s)	Defining Standard
Mode 0	240	16.7	ATA/ATAPI-4
Mode 1	160	25.0	ATA/ATAPI-4
Mode 2	120	33.3	ATA/ATAPI-4
Mode 3	90	44.4	ATA/ATAPI-5
Mode 4	60	66.7	ATA/ATAPI-5
Mode 5	40	100.0	ATA/ATAPI-6?

16-Bit and 32-Bit Access

One of the options on some chipsets and BIOSes is so-called 32-bit access or 32-bit transfers. In fact, the IDE/ATA interface always does transfers 16 bits at a time, reflecting its name ("AT attachment"--the original AT used a 16-bit data bus and a 16-bit ISA I/O bus). For this reason, the name "32-bit" access or transfer is somewhat of a misnomer.

Since modern PCs use 32-bit I/O buses such as the PCI bus, doing 16-bit transfers is a waste of half of the potential bandwidth of the bus. Enabling 32-bit access in the BIOS (if available) causes the PCI hard disk interface controller to bundle together two 16-bit chunks of data from the drive into a 32-bit group, which is then transmitted to the processor or memory. This results in a small performance increase.

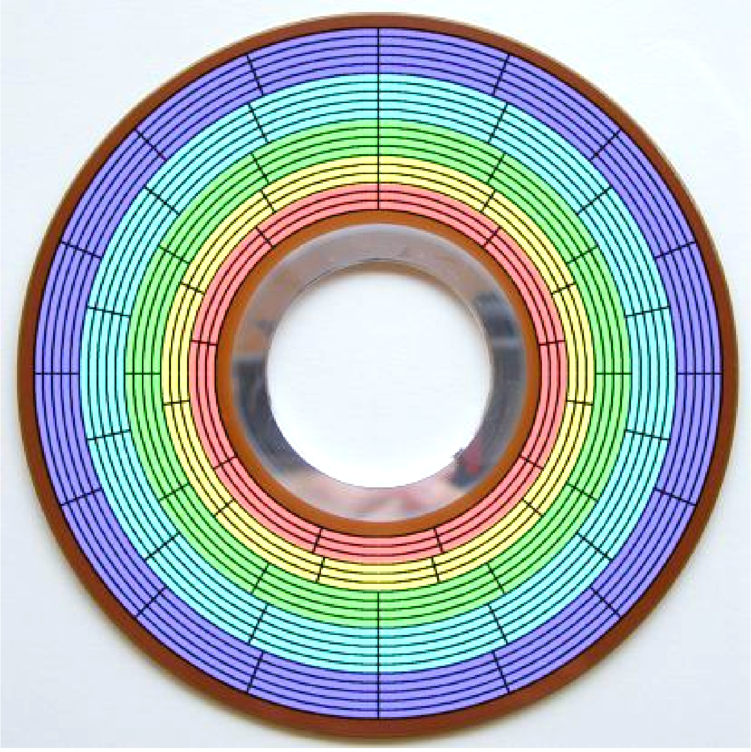
Block Mode

On some systems you will find an option in the system BIOS called block mode. Block mode is a performance enhancement that allows the grouping of multiple read or write commands over the IDE/ATA interface so that they can be handled on a single interrupt.

Interrupts are used to signal when data is ready to be transferred from the hard disk; each one, well, interrupts other work being done by the processor. Newer drives, when used with a supporting BIOS allow you to transfer as many as 16 or 32 sectors with a single interrupt. Since the processor is being interrupted much less frequently, performance is much improved, and more data is moving around with less command overhead, which is much more efficient than transferring data one sector at a time.

Zoned Bit Recording

To eliminate this wasted space, modern hard disks employ a technique called zoned bit recording (ZBR), also sometimes called multiple zone recording or even just zone recording. With this technique, tracks are grouped into zones based on their distance from the center of the disk, and each zone is assigned a number of sectors per track. As you move from the innermost part of the disk to the outer edge, you move through different zones, each containing more sectors per track than the one before. This allows for more efficient use of the larger tracks on the outside of the disk.



A graphical illustration of zoned bit recording. This model hard disk has 20 tracks. They have been divided into five zones, each of which is shown as a different color. The blue zone has 5 tracks, each with 16 sectors; the cyan zone 5 tracks of 14 sectors each; the green zone 4 tracks of 12 sectors; the yellow 3 tracks of 11 sectors, and the red 3 tracks of 9 sectors. You can see that the size (length) of a sector remains fairly constant over the entire surface of the If not for ZBR, if the inner-mostzone had its data packed as densely as possible, every track on this hard disk would be limited to only 9 sectors, greatly reducing capacity.

One interesting side effect of this design is that the raw data transfer rate (sometimes called the media transfer rate) of the disk when reading the outside cylinders is much higher than when reading the inside ones. This is because the outer cylinders contain more data, but the angular velocity of the platters is constant regardless of which track is being read (note that this constant angular velocity is not the case for some technologies, like older CD-ROM drives!) Since hard disks are filled from the outside in, the fastest data transfer occurs when the drive is first used. Sometimes, people benchmark their disks when new, and then many months later, and are surprised to find that the disk is getting slower! In fact, the disk most likely has not changed at all, but the second benchmark may have been run on tracks closer to the middle of the disk. (Fragmentation of the file system can have an impact as well in some cases.)

Integrated Drive Electronics / AT Attachment (IDE/ATA) Interface

Summary of IDE/ATA Standards

With the creation of several new ATA standards over the last few years, there are now quite a few of them "out there". The table below provides a quick summary of the different official IDE/ATA interfaces, showing their key attributes and features.

Interface Standard	ANSI Standard Number (includes date)	PIO Modes Added	DMA Modes Added	Ultra DMA Modes Added	Notable Features or Enhancements Introduced
ATA-1	X3.221-1994	0, 1, 2	Single word 0, 1, 2; multiword 0	--	--
ATA-2	X3.279-1996	3, 4	Multiword 1, 2	--	Block transfers, Logical block addressing, Improved identify drive command
ATA-3	X3.298-1997	--	--	--	Improved reliability, SMART, Drive security
ATA/ATAPI-4	NCITS 317-1998	--	--	0, 1, 2	Ultra DMA, 80-conductor IDE cable, CRC
ATA/ATAPI-5	NCITS 340-2000	--	--	3, 4	--
ATA/ATAPI-6	Under Development	--	--	5?	LBA expansion? Acoustic management? Multimedia streaming?

SCSI

SCSI stands for Small Computer Systems Interface, which is widely used in medium and large systems. SCSI is an industry-standard interface and offers faster transfer rates than does ATA/IDE, the interface most commonly used in desktop PCs.

In general, ATA/IDE is considered easier to implement and less expensive but does not offer as many features as SCSI. SCSI can support both the connection of many devices and the connection of many devices over long distances. SCSI features high transfer speeds, flexibility, as well as advanced functions such as command queuing and spindle sync. SCSI is completely backward compatible. Earlier versions include SCSI and Ultra SCSI, with the most current version, Ultra2 SCSI, offering even higher performance.

- 160 MB/s bus speed for SCSI compared to 100 MB/s for IDE.
- Command tagged queuing - SCSI hard disk drives re-order commands and data to offer the best performance on the bus.
- Domain validation - Test sequences are exchanged between the SCSI controller and the hard disk drive to test the communication path before sending the data. Failures, which are caused by cables, enclosures, etc. are limited. System integrators can take advantage of this technology.
- Hot swapping and the use of spare drives can provide automatic RAID rebuild and minimize the risks when hard disk drive failures occur.
- Packetization (Ultra 320 SCSI) - Command overhead is significantly reduced.
- In contrast to IDE, SCSI supports a multi-controller operation. This offers an additional level of security because it minimizes the risks of a system crash if a controller fails.

THE DIFFERENT TYPES OF SCSI INTERFACE ARE

a) Regular SCSI (SCSI 1)
b) Wide SCSI.
c) Fast SCSI.
d) Fast Wide SCSI.
e) Ultra SCSI.
f) Wide Ultra SCSI.
g) Ultra2 SCSI.
h) Wide Ultra2 SCSI.
i) Ultra3 SCSI.
j) Ultra 160 (Ultra 160/m) SCSI
k) Ultra 160+ SCSI.
l) Ultra320 SCSI.

Summary of SCSI Protocols and Transfer Modes

For easier comparison, the chart below shows all of the different SCSI transfer modes and feature sets, along with their key characteristics.

Transfer Mode	Defining Standard	Bus Width (bits)	Bus Speed (MHz)	Through- put (MB/s)	Special Features	Cabling	Signaling Method	Max Devices Per Bus	Max Cable Length (m)
"Regular" SCSI (SCSI-1)	SCSI-1	8	5	5		50-pin	SE	8	6
							HVD	8	25
Wide SCSI	SCSI-2	16	5	10		68-pin	SE	16	6
							HVD	16	25
Fast SCSI	SCSI-2	8	10	10		50-pin	SE	8	3
							HVD	8	25
Fast Wide SCSI	SCSI-2	16	10	20		68-pin	SE	16	3
							HVD	16	25
Ultra SCSI	SCSI-3 / SPI	8	20	20		50-pin	SE	8	1.5
								4	3
							HVD	8	25
Wide Ultra SCSI	SCSI-3 / SPI	16	20	40		68-pin	SE	8	1.5
								4	3
							HVD	16	25
Ultra2 SCSI	SCSI-3 / SPI-2	8	40	40		50-pin	LVD	8	12
								2	25
							HVD	8	25
Wide Ultra2 SCSI	SCSI-3 / SPI-2	16	40	80		68-pin	LVD	16	12
								2	25
							HVD	16	25
Ultra3 SCSI	SCSI-3 / SPI-3	16	40 (DT)	160	At least one of Fast-80, CRC, DV, QAS, Packet	68-pin	LVD	16	12
								2	25
Ultra160(/m) SCSI	SCSI-3 / SPI-3	16	40 (DT)	160	Fast-80, CRC, DV	68-pin	LVD	16	12
								2	25
Ultra160+ SCSI	SCSI-3 / SPI-3	16	40 (DT)	160	Fast-80, CRC, DV, QAS, Packet	68-Rpin	LVD	16	12
								2	25
Ultra320 SCSI	SCSI-3 / SPI-4	16	80 (DT)	320	Fast-160,	68-pin	LVD	16	12
								2	25

SCSI Host Adapters

Most IDE/ATA hard disks are controlled today by integrated IDE controllers that are built into the chipset on the motherboard. The SCSI interface is not, for the most part, controlled by built-in motherboard SCSI controllers, although some are and this is growing in popularity. Most systems require the addition of a special card which serves as the interface between the SCSI bus and the PC.

This device is called a SCSI host adapter, or alternately a host bus adapter (sometimes abbreviated HBA). It is sometimes called a SCSI controller or even just a SCSI card, though these are technically incorrect names. They are not accurate because SCSI is a systems-level interface, and every device on the bus has its own controller. Logically, the host adapter is just a SCSI device like any other. Its job is to act as the gateway between the SCSI bus and the internal PC's I/O bus. It sends and responds to commands and transfers data to and from devices on the bus and inside the computer itself. Since it is inside the PC, of course, the host adapter really isn't the same as the other devices on the bus--it's sort of a "first among equals", if you want to think about it that way.

Since SCSI is a very "intelligent" interface--meaning it has a lot of capabilities and the devices on it are able to interact in advanced ways--many SCSI host adapters have evolved rather exceptional capabilities, and can act in many ways to improve performance. In some ways, the host adapter is the key to good SCSI implementation in the PC, since no matter how advanced the peripherals are that you attach to the bus, everything goes through that host adapter.

Motherboard support for SCSI is actually on the rise, especially in higher-end systems, as SCSI becomes more "mainstream". It is still not common to find it in most motherboards because it increases cost, and most people still are not using SCSI. If you are building a new PC and want to go with SCSI, consider a motherboard with an integrated SCSI host adapter. When selecting such a motherboard, however, it is critical to pay specific attention to what SCSI transfer modes and feature sets the motherboard will support. While most built-in SCSI controllers can be disabled, having to buy a SCSI host adapter six months after you buy a SCSI-capable motherboard--because the motherboard-based controller doesn't do what you need it to--is just a waste of time and money.

SCSI Connectors

SCSI is a bus that supports both internal and external devices. To support these two types of devices, most SCSI host adapters come with both internal and external connectors. Internal connectors are usually mounted along the top edge of the SCSI host adapter, and are used for the ribbon cables employed for internal SCSI devices. External connectors are mounted along the outside edge of the host adapter (the part accessible from the back of the PC when the card is inserted into a system bus slot.)

The exact type of connectors provided on any given card depends on its design, and more specifically, the type of SCSI it is intended to support. A card that is designed to support narrow devices will have narrow (50-pin) connectors, while cards that are built to run wide devices will have 68-pin connectors. There are also different types of each of these two sizes of connector; for example, an older or lower-end host adapter may use the older high-density 68-pin connectors while high-end Ultra160 card may use the smaller very-high-density (VHDCI) connectors.

Multiple Segment and Channel Support

In its simplest form, a host adapter provides support for a single SCSI chain: that is, a single set of devices that are all connected together on the same SCSI bus. This is the way that many older and low-end SCSI host adapters work. They are fine for simple implementations, but are too limiting for complex SCSI setups. Especially with modern systems that need to use both LVD and single-ended devices, an adapter with support for just a single segment is insufficient for maximum performance. To expand capabilities, host adapter manufacturers make cards that support multiple segments, multiple channels, or both.

A segment is an electrically isolated "piece" of a SCSI bus; a single bus can be made up of one or more segments. Cards that implement multiple segments allow for more flexibility because the segments are electrically separate. Each segment can have a cable as long as the normal maximum cable length allowed for that particular type of SCSI, for example. One segment can use an internal cable within the PC and another an external cable. It's important to remember though that two segments on a single channel are logically considered to be part of the same SCSI bus even if they are electrically separate. This means all devices on all segments must have unique IDs, and that maximum bandwidth is shared between all devices on all segments that make up the bus.

The most expensive host adapters go beyond multiple segment support and actually have multiple channels. These are similar in concept to the way an IDE/ATA controller typically has two channels. Each channel is completely independent of the other, both electrically and logically. This means the two run in parallel with each other: you get support for twice as many devices, and twice as much throughput. In essence, a card with two channels is two host adapters in the same package. For example, an Ultra160 host adapter with dual channels will support 30 drives (16 per channel less one each per channel for the host adapter) and theoretical throughput of up to 320 MB/s (160 MB/s per channel). Note that each channel can itself have more than one electrical segment.

Host adapters that support multiple channels are not really needed for most applications, especially if already using high-performance SCSI like Ultra160; they are more common in servers than desktop PCs. Multiple segments, however, are commonly found even in desktop SCSI cards. One common use for multiple segments is to allow independent use of LVD and SE devices on the same host adapter without causing the LVD devices to degrade to SE operation.

Summary of IDE/ATA and SCSI Comparisons

The following table summarizes the comparison of SCSI and IDE/ATA. See the individual sections for a more thorough explanation of the summary conclusions below:

Interface Factor	IDE/ATA	SCSI
Cost	Low	Moderate to high
Performance	High for single devices or single tasking, moderate to low for multiple devices or multitasking	High in most situations
Ease of Configuration and Use	High for small number of devices, moderate for large number of devices	Low to moderate for both small and large numbers of devices
Expansion and Number of Devices	Moderate	High
Device Type Support	Moderate	High
Device Availability and Selection	High	Moderate
Software / Operating System Compatibility	High	Moderate to high
System Resource Usage	Moderate to poor	Good
Support for non-PC Platforms	Moderate	Good

FLOPPY DISKS

The invention of hard disks relegated floppy disks to the secondary roles of data transfer and software installation. The invention of the CD-ROM and the Internet, combined with the increasingly large size of software files, is threatening even these secondary roles. The floppy disk still persists, basically unchanged for over a decade, in large part because of its universality; the 3.5 inch 1.44 MB floppy is present on virtually every PC made in the last 10 years, which makes it still a useful tool. The floppy disk's current role is in these area:

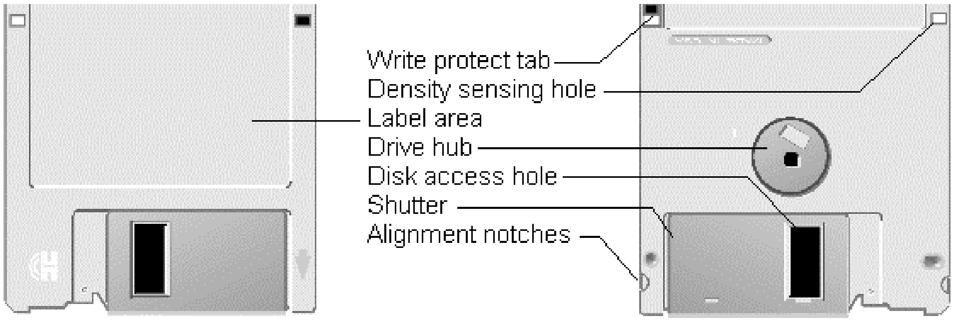
- **Data Transfer:** The floppy disk is still the most universal means of transferring files from one PC to another. With the use of compression utilities, even moderate-sized files can be shoehorned onto a floppy disk, and anyone can send anyone a disk and feel quite confident that the PC at the other end will be able to read it. The PC 3.5" floppy is such a standard, in fact, that many Apple and even UNIX machines can read them, making these disks useful for cross-platform transfer.

- **Small File Storage and Backup:** The floppy disk is still used for storing and backing up small amounts of data, probably more than you realize.

- **Software Installation and Driver Updates:** Many new pieces of hardware still use floppies for distributing driver software and the like, and some software still uses floppies (although this is becoming less common as software grows massive and CD-ROM drives become more universal.)

While floppy drives still have a useful role in the modern PC, there is no denying their reduced importance. Very little attention is paid to floppy "performance" any more, and even choosing makes or models involves a small fraction of the amount of care and attention required for selecting other components. In essence, the floppy drive today is a commodity item!

To understand how the data is organised on the disk, let us first consider the physical structure of the disk and the drive mechanism that reads floppy and writes to it.



Inside the square plastic jacket of the floppy disk is a circular platter made of tough plastic material. This plastic disk is coated with a magnetic material. A disk drive stores the data onto this disk by writing and reading magnetically encoded patterns that represent digital data. Since both the sides of the disk are coated both sides can be used to store the data.

A floppy drive contains a motor that rotates the disk at a constant speed. The drive has two read/write heads, one on each side of the disk. The heads are mounted on an arm that moves them in unison to nay position towards or away form the center of the disk.

The geometry of the fixed disk is similar to that of the floppy disk. Fixed disk rotate at a much higher speed, so the platters are made of magnetically coated metal, and not flexible plastic. Also, fixed disks consist a stack of several platters that rotate together, so fixed disks have multiple read/write heads-one for each disk surface.

Information is always stored on the disk surface in a series of concentric circles called tracks. Each track is further divided into segments called sectors. This process of dividing the disk into tracks and sectors is called formatting of disk. Any new disk has to be first formatted before writing data to it.

Had this scheme of tracks and sectors not been used it would have lead to delays in:

(a) Searching the information to be read from the disk.
(b) Searching the empty space for writing new information to the disk.

This division of the disk into tracks and sectors speeds up the operation of reading the data from the disk as well as writing the data onto the disk.A read/write head inside the disk drive does the process of reading information or writing information. To read/write information from the disk, the disk has to rotate and the appropriate sector of the appropriate track has to come below the read/write head.

Locating a particular track on the disk is a relatively uncomplicated matter. The disk drive merely moves the read/write head to the position where the specified track is located, much likely the way in which the needle of a record player is positioned on a location of a specific song on the record. The only difference being, the needle moves in an angle, whereas the read/write head moves linearly.

For locating a sector, a small hole on the floppy disk called the sector hole is made use of. This sector hole makes the location of the first sector and all other sectors that are referred to with reference to this hole.

The amount of information that can be stored on each side of the disk depends on the number of tracks, number of sectors and size of each sector.

How many tracks, how many sectors and what sector size depends upon the way in which the disk is formatted, which is under software control. This is the reason why a same disk can be formatted in a number of ways. Let us see the way in which a disk can be formatted.

Floppy Disk Formats.

We have seen earlier that the division of floppy into tracks and sectors is known as formatting a diskette. Unless and until a new disk is formatted data cannot be stored on it.

The table below shows the various formats supported by DOS, their specifications and the versions of DOS with they came into existence.

DOS versions	Format ID	Sides	Tracks /side	Sectors/ track	capacity	Media Desc.
1.0	S8	1	40	8	160 kb	FE
1.1	D8	2	40	8	320 kb	FF
2.0	S9	1	40	9	180 kb	FC
2.0	D9	2	40	9	360 kb	FD
3.0	QD15	2	80	15	1.2 Mb	F9
3.3	QD18	2	80	18	1.44 Mb	F0

In the above figure the format id is the usual way in which we refer a particular format, whereas the media descriptor is the way DOS identifies a particular format.

Logical Structure of Floppy Disks.

Regardless of the capacity of the disk we use, all disks are logically formatted in the same way, the disk sides, tracks and sectors are identified numerically with the same notation, and certain sectors are always reserved for some special programs that DOS uses to manage disk operations.

Every floppy disk is divided into four separate areas. These are:

(a) Boot Sector
(b) File Allocation Table (FAT)
(c) Directory
(d) Data Space

The size the size of each area on the disk varies from format to format but the structure and the order of these areas on the diskettes remains the same. Let us now dvelve more deeply into these four areas.

BOOT SECTOR :-

A "file system boot sector" is the first physical sector on a logical volume. A logical volume might be a primary partition, a logical drive in an extended partition, or a composite of two or more partitions, as is the case with mirrors, stripe sets, and volume sets.

On floppy disks, the boot sector is the first sector on the disk. In the case of hard drives, the first sector is referred to as the "Master Boot Record" or "MBR." This MBR is different from a file system boot sector and contains a partition table, which describes the layout of logical partitions on that hard drive. The file system boot sector would be the first sector in one of those partitions.

The Boot Process:-

The boot process of 80x86-based personal computers (as opposed to RISC- based systems) makes direct use of a file system boot sector for executing instructions. The initial boot process can be summarized as follows:

- Power On Self Test (or POST) initiated by system BIOS and CPU.
- BIOS determines which device to use as the "boot device."
- BIOS loads the first physical sector from the boot device into memory and transfers CPU execution to the start of that memory address.
- If the boot device is a hard drive, the sector loaded in step 3 is the MBR, and the boot process proceeds as follows:
- MBR code loads the boot sector referenced by the partition table for the "active primary partition" into memory and transfers CPU execution to the start of that memory address.
- Up to this point, the boot process is entirely independent of how the disk is formatted and what operating system is being loaded. From this point on, both the operating and file systems in use play a part.

In the case of FAT volumes which have Windows NT installed, the FAT boot sector is responsible for identifying the location of the file "NTLDR" on the volume, loading it into memory, and transferring control to it.

Inside the FAT Boot Sector

Because the MBR transfers CPU execution to the boot sector, the first few bytes of the FAT boot sector must be valid executable instructions for an 80x86 CPU. In practice these first instructions constitute a "jump" instruction and occupy the first 3 bytes of the boot sector. This jump serves to skip over the next several bytes which are not "executable."

Following the jump instruction is an 8 byte "OEM ID". This is typically a string of characters that identifies the operating system that formatted the volume.

Following the OEM ID is a structure known as the BIOS Parameter Block, or "BPB." Taken as a whole, the BPB provides enough information for the executable portion of the boot sector to be able to locate the NTLDR file. Because the BPB always starts at the same offset, standard parameters are always in a known location. Because the first instruction in the boot sector is a jump, the BPB can be extended in the future, provided new information is appended to the end. In such a case, the jump instruction would only need a minor adjustment. Also, the actual executable code can be fairly generic. All the variability associated with running on disks of different sizes and geometries is encapsulated in the BPB.

The BPB is stored in a packed (that is, unaligned) format. The following table lists the byte offset of each field in the BPB. A description of each field follows the table.

Field	**Offset**	**Length**
Bytes Per Sector	11	2
Sectors Per Cluster	13	1
Reserved Sectors	14	2
FATs	16	1
Root Entries	17	2
Small Sectors	19	2
Media Descriptor	21	1
Sectors Per FAT	22	2
Sectors Per Track	24	2
Heads	26	2
Hidden Sectors	28	4
Large Sectors	32	4

Bytes Per Sector: This is the size of a hardware sector and for most disks in use in the United States, the value of this field will be 512.

Sectors Per Cluster: Because FAT is limited in the number of clusters (or "allocation units") that it can track, large volumes are supported by increasing the number of sectors per cluster. The cluster factor for a FAT volume is entirely dependent on the size of the volume. Valid values for this field are 1, 2, 4, 8, 16, 32, 64, and 128. Query in the Microsoft Knowledge Base for the term "Default Cluster Size" for more information on this subject.

Reserved Sectors: This represents the number of sectors preceding the start of the first FAT, including the boot sector itself. It should always have a value of at least 1.

FATs: This is the number of copies of the FAT table stored on the disk. Typically, the value of this field is 2.

Root Entries: This is the total number of file name entries that can be stored in the root directory of the volume. On a typical hard drive, the value of this field is 512. Note, however, that one entry is always used as a Volume Label, and that files with long file names will use up multiple entries per file. This means the largest number of files in the root directory is typically 511, but that you will run out of entries before that if long file names are used.

Small Sectors: This field is used to store the number of sectors on the disk if the size of the volume is small enough. For larger volumes, this field has a value of 0, and we refer instead to the "Large Sectors" value which comes later.

Media Descriptor: This byte provides information about the media being used. The following table lists some of the recognized media descriptor values and their associated media. Note that the media descriptor byte may be associated with more than one disk capacity.

Byte	Capacity	Media Size and Type
F0	2.88 MB	3.5-inch, 2-sided, 36-sector
F0	1.44 MB	3.5-inch, 2-sided, 18-sector
F9	720 KB	3.5-inch, 2-sided, 9-sector
F9	1.2 MB	5.25-inch, 2-sided, 15-sector
FD	360 KB	5.25-inch, 2-sided, 9-sector
FF	320 KB	5.25-inch, 2-sided, 8-sector
FC	180 KB	5.25-inch, 1-sided, 9-sector
FE	160 KB	5.25-inch, 1-sided, 8-sector
F8	-----	Fixed disk

Sectors Per FAT: This is the number of sectors occupied by each of the FATs on the volume. Given this information, together with the number of FATs and reserved sectors listed above, we can compute where the root directory begins. Given the number of entries in the root directory, we can also compute where the user data area of the disk begins.

Sectors Per Track and Heads: These values are a part of the apparent disk geometry in use when the disk was formatted.

Hidden Sectors: This is the number of sectors on the physical disk preceding the start of the volume. (that is, before the boot sector itself) It is used during the boot sequence in order to calculate the absolute offset to the root directory and data areas.

Large Sectors: If the Small Sectors field is zero, this field contains the total number of sectors used by the FAT volume.

Some additional fields follow the standard BIOS Parameter Block and constitute an "extended BIOS Parameter Block." The next fields are:

Field	Offset	Length
Physical Drive Number	36	1
Current Head	37	1
Signature	38	1
ID	39	4
Volume Label	43	11
System ID	54	8

Physical Drive Number: This is related to the BIOS physical drive number. Floppy drives are numbered starting with 0x00 for the A: drive, while physical hard disks are numbered starting with 0x80. Typically, you would set this value prior to issuing an INT 13 BIOS call in order to specify the device to access. The on-disk value stored in this field is typically 0x00 for floppies and 0x80 for hard disks, regardless of how many physical disk drives exist, because the value is only relevant if the device is a boot device.

Current Head: This is another field typically used when doing INT13 BIOS calls. The value would originally have been used to store the track on which the boot record was located, but the value stored on disk is not currently used as such. Therefore, Windows NT uses this field to store two flags:

- The low order bit is a "dirty" flag, used to indicate that autochk should run chkdsk against the volume at boot time.

- The second lowest bit is a flag indicating that a surface scan should also be run.

Signature: The extended boot record signature must be either 0x28 or 0x29 in order to be recognized by Windows NT.

ID: The ID is a random serial number assigned at format time in order to aid in distinguishing one disk from another.

Volume Label: This field was used to store the volume label, but the volume label is now stored as a special file in the root directory.

System ID: This field is either "FAT12" or "FAT16," depending on the format of the disk.

On a bootable volume, the area following the Extended BIOS Parameter Block is typically executable boot code. This code is responsible for performing whatever actions are necessary to continue the boot-strap process. On Windows NT systems, this boot code will identify the location of the NTLDR file, load it into memory, and transfer execution to that file. Even on a non-bootable floppy disk, there is executable code in this area. The code necessary to print the familiar message, "Non-system disk or disk error" is found on most standard, MS-DOS formatted floppy disks that were not formatted with the "system" option.
Finally, the last two bytes in any boot sector always have the hexidecimal values: 0x55 0xAA.

The File Allocation Table

A file allocation table (FAT) is a table that an operating system maintains on a hard disk that provides a map of the cluster (the basic unit of logical storage on a hard disk) that a file has been stored in. When you write a new file to a hard disk, the file is stored in one or more clusters that are not necessarily next to each other; they may be rather widely scattered over the disk. A typical cluster size is 2,048 byte, 4,096 bytes, or 8,192 bytes. The operating system creates a FAT entry for the new file that records where each cluster is located and their sequential order. When you read a file, the operating system reassembles the file from clusters and places it as an entire file where you want to read it. For example, if this is a long Web page, it may very well be stored on more than one cluster on your hard disk.

Until Windows 95 OSR2 (OEM Release 2), DOS and Windows file allocation table entries were 16 bits in length, limiting hard disk size to 128 megabyte, assuming a 2,048 size cluster. Up to 512 megabyte support is possible assuming a cluster size of 8,192 but at the cost of using clusters inefficiently. DOS 5.0 and later versions provide for support of hard disks up to two gigabyte with the 16-bit FAT entry limit by supporting separate FATs for up to four partitions.

With 32-bit FAT entry (FAT32) support in Windows 95 OSR2, the largest size hard disk that can be supported is two terabyte! However, personal computer users are more likely to take advantage of FAT32 with 5 or 10 gigabyte drives.

Virtual File Allocation Table VFAT (Virtual File Allocation Table) is the part of the Windows 95 and later operating system that handles long file names, which otherwise could not be handled by the original file allocation table (file allocation table) programming. A file allocation table is the means by which the operating system keeps track of where the pieces of a file are stored on a hard disk. Since the original FAT for the Disk Operating System operating system assumed file names were limited to a length of eight characters, a program extension was needed to handle the longer names allowed in Windows 95. Microsoft refers to this extension as a since other operating systems may need to install and use it in order to access FAT partitions written by Windows 95 and later Windows systems. The VFAT extension runs in protected mode, uses 32-bit code, and uses VCACHE for disk cache.

FAT Types

There are four types of FATs (File Allocation Table). FAT12 is now obsolete, used on floppy disks and partitions below 16Mb only. FAT 16 is the next step. It can be used if a DOS partition is between 16Mb and 32Mb. BIGDOS is also a 16-bit type, but allows larger partition sizes. Plain DOS and Windows95 use this type nowadays (OS/2, WindowsNT and Linux can also be installed over FAT, but there is no point for doing that). The maximum partition size is 2Gb, FAT 32 is the newest. It is introduced in Windows 95 OEM Service Relase 2 (known as W95b), It allows big partition sizes (2Tb), but it's incompatible with older types.

FAT - File Allocation Table

The file system that is used / or ordinarily designed for floppies and used by DOS, W 3.x, W95, Windows NT and OS/2. A FAT directory holds info such as name, file size, date & time stamp, the starting cluster number and the file attributes like (hidden, system & etc.). It's file system can support up to 65,525 clusters and is limited to 2 gigabytes. This works best on small 500mb drives because of the cluster size. It seems to be about 2% faster than FAT32 and NTFS but Windows is faster if confined to a small area. FAT performance drops off after 400mb's and over.

FAT32 - File Allocation Table 32

FAT32 will not recognize FAT or NTFS volumes of other operating systems so you can't use them. It supports drives up to 2 terabytes and it uses smaller clusters which are 4096bytes in size.

The Difference (FAT12/16 or FAT32)

Remember that DOS 6.x or even versions of Windows prior to SR2 won't recognize the front end of a FAT32 partition. So if you must run the occasional old DOS app, move it into a FAT16 drive partition and then restart from an old DOS boot diskette.FAT16 does not support partitions larger than 2GB. FAT32 is an improvement, as it supports drives up to 2 Terabytes in size, and cluster sizes are 4K for partitions smaller than 8GB. So, if you can get FAT32 on the drive, it will work. Fat12/16 and Fat32 is a Partition size/cluster size issue. FAT32 solves this problem by reducing to 4KB the default file cluster size for partitions between 260MB and 8GB. (Drives or partitions under 260MB use .5KB clusters.) Up to 16GB, FAT32's cluster size is 8KB; to 32GB, it's 16KB; and for partitions of 32GB and greater, the cluster size holds steady at 32KB. FAT32 adds a few other improvements. The root directory on a FAT32 drive is now an ordinary cluster chain, so it can be located anywhere on the drive. This removes FAT16's previous limitation of 512 root directory entries. In addition, the boot record on FAT32 drives has been expanded to allow a backup of critical data structures. This makes FAT32 drives less susceptible to failure.

FAT32 partitions are also invisible to other operating systems, including other versions of Windows. To access a FAT32 partition from a boot floppy, you must create an SR2 start-up disk. You won't see your C: drive if you boot from an older Win95 or DOS start-up disk. If you start out with SR2 on a FAT32 partition and subsequently install Windows NT or OS/2, neither OS will be able to access the FAT32 partition.
In addition, you can't run disk-compression software (such as Microsoft's DriveSpace) on a FAT32 partition. But it is possible to include both FAT32 and FAT16 partitions on a single hard disk and use DriveSpace compression on FAT16 partitions. (so SR2 includes the same DriveSpace 3 compression Microsoft ships with its Plus pack.)

NTFS - New Technology File System

This systems structure is the (MFT) or master file table. It uses too much space to use on a (ex; 400mb) hard-drive because it keeps multiple copies of files in the MFT to protect against data loss. It also uses clusters to store data in small noncontiguous clusters and isn't broken up resulting in good performance on large hard-drives. It also supports Hot Fixing where bad sectors are automatically detected and marked.

HPFS - High Performance File System

This system sorts the directory based on names and is better organized, is faster and is a better space saver. It allocates data to sectors instead of clusters, organized into 8mb bands. This banding improves performance because the read/write heads don't have to return to track zero each time for access.

NetWare File System

This is quick because Novell developed it for NetWare servers being Netware 3.x and 4.x partitions.

Linux Ext2

This is also quick because it is a developed version of UNIX. The Linux Ex12 volume supports up to 2 terabytes.

FATx

FAT32x is a proprietary file system developed by Microsoft to enable FAT32 partitions to exist beyond 1024 cylinders. Windows 95 versions 'B' (OSR2) and later and Windows 98 are the only operating systems currently using FAT32x partitions. The movement to drives that have more than 1024 translated cylinders (i.e. 8Gb and larger) has been the catalyst for this development.

Working in FAT32x partitions is essentially the same as working in FAT32 partitions. However, when attempting to manipulate a FAT32x partition, problems may occur. Procedures such as copying, imaging, resizing, and moving FAT32x partitions require different methods than those used for FAT32 partitions.

Many new computers have pre-installed FAT32x partitions. This has created numerous problems for individuals wishing to modify their partitions on their new systems. FAT32x partitions have a different file system flag in the partition table. Sometimes a FAT32x partition is erroneously created entirely within 1024 cylinders. This can be corrected, in some cases, by using a disk editing utility.

Floppy Types

5.25" Media Construction

The first floppy disks were actually not 3.5" or 5.25" at all--they were 8" in size. (And what beasts they are, if you've ever seen them. They are still in use on some very old non-PC equipment.) The 5.25" is the younger cousin of the original floppy and retains for the mostpart the same basic design as that media, in a smaller size.

The 5.25" disk is comprised of two basic pieces: the actual, round disk media, sometimes called a "cookie", and the protective jacket. The actual disk is made from a thin piece of plastic and is coated with a magnetic material. It has a large hole in its center that is used by the drive to grasp the disk and spin it--the jacket of course does not spin. A slot is cut in the jacket to expose the disk for the read/write heads; it is wide enough for the heads and long enough to allow the actuator to move the heads over all of the tracks on the disk. A notch at the side of the disk acts as a write-protect control; it is somewhat crude however in that you must use tape over the notch to write-protect the disk.

The 5.25" disk earns its name: "floppy". These disks are notoriously fragile. The jacket provides inadequate protection for the disk itself; this, combined with the large size of the disk, makes it very easy to bend. Special care must be taken not to damage them accidentally; basically, they need to be kept inside a plastic box most of the time to avoid destroying them. They do not take kindly to being sent in the mail unless in a larger box. The read/write "window" of the disk is exposed and for this reason the disks can be easily damaged if not kept in their protective paper "pockets". They can even be damaged by writing on the jacket with a ball-point pen, because the jacket is so thin that the pen can cause an impression in the disk media itself.

The lack of durability of the 5.25" media helped contribute to the downfall of the 5.25" floppy disk drive, compared to the 3.5" disks.

3.5" Media Construction

3.5" floppy disks are similar in concept of course to 5.25" disks, but offer several improvements in implementation. The three main improvements over the older style of disk all have to do with durability. First, the jacket is made of a much sturdier material that can withstand a reasonable amount of abuse without destroying the disk within. Second, the read/write window of the disk itself is protected by a sliding metal cover that is engaged when the media is inserted into the drive. Finally, the disk itself is smaller, which makes it much sturdier as well.

The 3.5" disk has several other improvements over the 5.25" media as well. The write-protect notch is replaced by a hole with a sliding plastic piece; when the hole is open the disk is write-protected and when it is closed the disk is write-enabled, and switching from one state to the other is simple. The large hole in the center of the 5.25" disk is replaced by a small metal disk with an indexing hole in it, improving durability further.

File System Parameter	360 KB 5.25"	1.2 MB 5.25"	720 KB 3.5"	1.44 MB 3.5"	2.88 MB 3.5"
Cluster Size	2 sectors	1 sector	2 sectors	1 sector	2 sectors
Maximum Number of Root Directory Entries	112	224	112	224	448

Summary of Floppy Disk Types and pecifications

The following table shows a summary of the various floppy disk specifications provided in other sections of this chapter, for each of the five major floppy disk types:

Category	Specification	360 KB 5.25"	1.2 MB 5.25"	720 KB 3.5"	1.44 MB 3.5"	2.88 MB 3.5"
Drive	Read/Write Heads (Data Surfaces)	2	2	2	2	2
	Spindle Motor Speed	300 RPM	360 RPM	300 RPM	300 RPM	300 RPM
Controller	Minimum Controller Transfer Rate	250 Kbits/s	500 Kbits/s	250 Kbits/s	500 Kbits/s	1 Mbits/s
Media	Track Density (TPI)	48	96	135	135	135
	Bit Density (BPI)	5,876	9,869	8,717	17,434	34,868
	Density Name	Double Density (DD)	High Density (HD)	Double Density (DD)	High Density (HD)	Extra-High Density (ED)
Geometry	Tracks (Cylinders)	40	80	80	80	80
	Sectors Per Track/Cylinder	9	15	9	18	36
	Total Sectors Per Disk	720	2,400	1,440	2,880	5,760
File System	Cluster Size	2 sectors	1 sector	2 sectors	1 sector	2 sectors
	Maximum Root Directory Entries	112	224	112	224	448
Capacity	Unformatted Capacity	~480 KB	~ 1.6 MB	~1 MB	~2 MB	~4 MB
	Formatted Capacity (binary kilobytes)	360	1,200	720	1,440	2,880
	Formatted Capacity (bytes)	368,640	1,228,800	737,280	1,474,560	2,949,120
	File System Overhead (bytes)	6,144	14,848	7,168	16,896	17,408
	Total Usable Capacity (bytes)	362,496	1,213,952	730,112	1,457,664	2,931,712
	Total Usable Capacity (binary KB)	354	1,185.5	713	1,423.5	2,863
	Total Usable Capacity (binary MB)	0.346	1.158	0.696	1.390	2.796

CD-ROM Drives

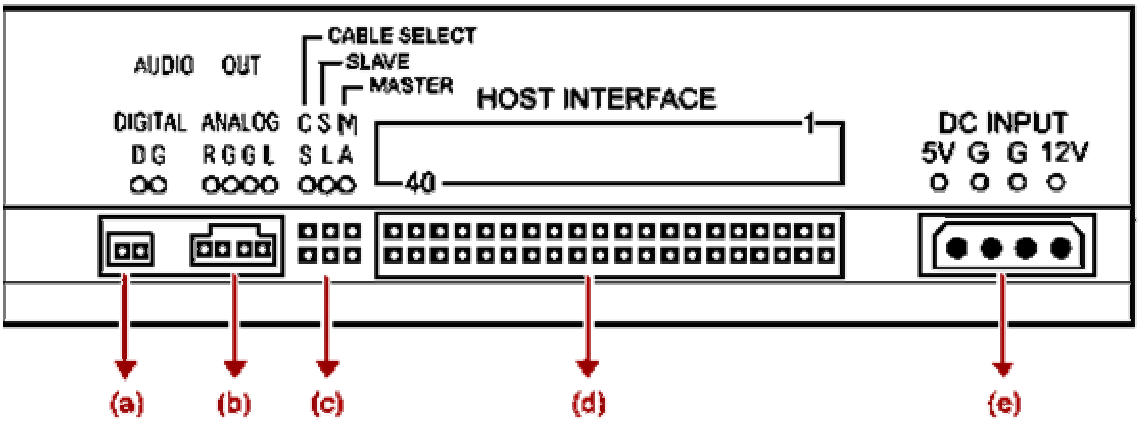
In a few short years, the Compact Disk - Read Only Memory (CD-ROM) drive has gone from pricey luxury to inexpensive necessity on the modern PC. The CD-ROM has opened up new computing vistas that were never possible before, due to its high capacity and broad applicability. In many ways, the CD-ROM has replaced the floppy disk drive, but in many ways it has allowed us to use our computers in ways that we never used them before. In fact, the "multimedia revolution" was largely a result of the availability of cheap CD-ROM drives.

As the name implies, CD-ROMs use compact disks, in fact, the same physical disk format as the ones we use for music. Special formatting is used to allow these disks to hold data. As CD-ROMs have come down in price they have become almost as common in a new PC as the hard disk or floppy disk, and they are now the method of choice for the distribution of software and data due to their combination of high capacity and cheap and easy manufacturing. Recent advances in technology have also improved their performance to levels approaching those of hard disks in many respects. CD-ROM drives play a significant role in the following essential aspects of your computer system:

- Software Support: The number one reason why a PC today basically must have a CD-ROM drive is the large number of software titles that are only available on CD-ROM. At one time there were a few titles that came on CD-ROM, and they generally came on floppy disks as well. Today, not having a CD-ROM means losing out on a large segment of the PC software market. Also, some CD-ROMs require a drive that meets certain minimum performance requirements.

- Performance: Since so much software uses the CD-ROM drive today, the performance level of the drive is important. It usually isn't as important as the performance of the hard drive or system components such as the processor or system memory, but it is still important, depending on what you use the drive for. Obviously, the more you use the CD-ROM, the more essential it is that it perform well.

CD-ROM Drive Construction and Operation



In terms of construction and basic components, CD-ROMs are rather similar in most regards to other storage devices that use circular, spinning media, which isn't that much of a surprise. The big difference of course is the way the information is recorded on the media, and the way that it is read from the media as well. This section takes a look at the basics of how CD-ROM drives are constructed and how they work.

Optical "Head" Assembly

The middle two letters in "CD-ROM" stand for "read only", so it shouldn't be any surprise that standard CD-ROM drives are read only devices, and cannot be written to. (Newer variants of CD-ROMs, CD-R and CD-RW drives, break this long-standing rule of this type of device.)

The reason that the word "head" is in quotes is that CD-ROM drives do not use a read head in the conventional sense the way a floppy disk or hard disk does. It isn't just that the head cannot record, it really isn't a single solid head that moves over the surface of CD-ROM media, reading it. The head is a lens--sometimes called a pickup--that moves from the inside to the outside of the surface of the CD-ROM disk, accessing different parts of the disk as it spins. This is just like how a hard disk or floppy disk head works, but the CD-ROM lens is only one part of an assembly of components that together.

Here's how the CD-ROM works:

1. A beam of light energy is emitted from an infrared laser diode and aimed toward a reflecting mirror. The mirror is part of the head assembly, which moves linearly along the surface of the disk.
2. The light reflects off the mirror and through a focusing lens, and shines onto a specific point on the disk.

3. A certain amount of light is reflected back from the disk. The amount of light reflected depends on which part of the disk the beam strikes: each position on the disk is encoded as a one or a zero based on the presence or absence of "pits" in the surface of the disk. A series of collectors, mirrors and lenses accumulates and focuses the reflected light from the surface of the disk and sends it toward a photodetector.
4. The photodetector transforms the light energy into electrical energy. The strength of the signal is dependent on how much light was reflected from the disk.

Most of these components are fixed in place; only the head assembly containing the mirror and read lens moves. This makes for a relatively simplified design. CD-ROMs are of course single-sided media, and the drive therefore has only one "head" to go with this single data surface.

Since the read head on a CD-ROM is optical, it avoids many of the problems associated with magnetic heads. There is no contact with the media as with floppy disks so there is no wear or dirt buildup problem. There is no intricate close-to-contact flying height as with a hard disk so there is no concern about head crashes and the like. However, since the mechanism uses light, it is important that the path used by the laser beam be unobstructed. Dirt on the media can cause problems for CD-ROMs, and over time dust can also accumulate on the focus lens of the read head, causing errors as well.

Head Actuator Mechanism

Most people don't think of a CD-ROM drive as having a head actuator, in the sense that a hard disk or floppy disk drive does. In fact, however, the lens assembly does move across the CD-ROM media in a similar way to how the heads on a hard disk or floppy disk drive do.

As describedearlier, only part of the whole mechanism used to read the CD-ROM actually moves. This is the lens and mirror assembly that focuses the laser energy onto the surface of the disk. The technology used to move the read head on a CD-ROM drive is in some ways a combination of those used for floppy disk drives and for hard disk drives.

Mechanically, the head moves in and out on a set of rails, much as the head of a floppy disk drive does. At one end of its travel the head is positioned on the outermost edge of the disk, and on the other end it is near the hub of the CD. However, due to the dense way the information is recorded on the CD, CD-ROM drives cannot use the simple stepper motor positioning of a floppy disk. CD-ROM media actually use a tighter density of tracks than even hard disks do! Instead, the positioning of the head is controlled by an integrated microcontroller and servo system. This is similar to the way the actuator on a hard disk is positioned. This means that the alignment problems found on floppy drives (and much older hard disks) are not generally a concern for CD-ROM drives, and there is some tolerance for a CD that is slightly off center (but not a lot).

Like a floppy disk, the head actuator on a CD-ROM is relatively slow. The amount of time taken to move the heads from the innermost to the outermost tracks--called a full-stroke seek--is about an order of magnitude higher than it is for hard disks.

Spindle Motor, Constant Linear Velocity (CLV) and Constant Angular Velocity (CAV)

Like all spinning-disk media, the CD-ROM drive includes a spindle motor that turns the media containing the data to be read. The spindle motor of a standard CD-ROM is very different from that of a hard disk or floppy drive in one very important way: it does not spin at a constant speed. Rather, the speed of the drive varies depending on what part of the disk (inside vs. outside) is being read.

Standard hard disks and floppy disks spin the disk at a constant speed. Regardless of where the heads are, the same speed is used to turn the media. This is called constant angular velocity (CAV) because it takes the same amount of time for a turn of the 360 degrees of the disk at all times. Since the tracks on the inside of the disk are much smaller than those on the outside of the disk, this constant speed means that when the heads are on the outside of the disk they will traverse a much longer linear path than they do when on the inside. Hence, the linear velocity is not constant. Newer hard disks take advantage of this fact by storing more information on the outer tracks of the disk than they do on the inner tracks, a process called zoned bit recording. They also have higher transfer rates when reading data on the outside of the disk, since more of it spins past the head in each unit of time.

CD-ROMs take a different approach. They adjust the speed of the motor so that the linear velocity of the disk is always constant. When the head is on the outside of the disk, the motor runs slower, and when it is on the inside, it runs faster. This is done to ensure that the same amount (rate) of data always goes past the read head in a given period of time. This is called constant linear velocity or CLV.

The reason that CD-ROMs work this way is based on their heritage of being derived from audio CDs. Early CD players did not have the necessary smarts or buffer memory to allow them to deal with bits arriving at a different rate depending on what part of the disk they were using. Therefore, the CD standard was designed around CLV to ensure that the same amount of data would be read from the disk each second no matter what part of it was being accessed. CD-ROMs were designed to follow this methodology.

The speed of the spindle motor is controlled by the microcontroller, tied to the positioning of the head actuator. The data signals coming from the disk are used to synchronize the speed of the motor and make sure that the disk is turning at the correct rate.

The first CD-ROMs operated at the same speed as standard audio CD players: roughly 210 to 539 RPM, depending on the location of the heads. This results in a standard transfer rate of 150 KB/s. It was realized fairly quickly that by increasing the speed of the spindle motor, and using sufficiently powerful electronics, it would be possible to increase the transfer rate substantially. There's no advantage to reading a music CD at double the normal speed, but there definitely is for data CDs. Thus the double-speed, or 2X CD-ROM was born. It followed in short order with 3X, 4X and even faster drives.

Virtually all of these drives up to about 12X or so still vary the motor speed to maintain constant linear velocity. As the speed of the drives has increased, many newer drives have come out that actually revert back to the CAV method used for hard disks. In this case, their transfer rate will vary depending on where on the disk they are working, again, just like it does for a hard disk. The "X" rating can be somewhat specious for these drives, since they achieve it only--at best--at the outer edge of the disk. No CAV drive claiming to be 24X actually transfers at that rate over the whole disk. Of course, hard disk drives are the same way and nobody seems to complain about their claims. Some drives actually use a partial CLV or mixed CLV/CAV implementation where the speed of the disk is varied but not as much as in a true CLV drive.

Why change back to CAV?

The change back to CAV as the drives get faster and faster is being done due to the tremendous difficulty in changing the speed of the motor when it is going so fast. It is one thing to change a disk spinning at 210 RPM to 539 and back again, but quite another to change it from 5,040 to 12,936 and then back to 5,040! This spin-up and spin-down action is actually one factor contributing to the slow performance of CD-ROMs especially on random accesses.

This table summarizes the differences between CLV and CAV:

Characteristic	Constant Linear Velocity (CLV)	Constant Angular Velocity (CAV)
Drive Speed	Variable	Fixed
Transfer Rate	Fixed	Variable
Application	Conventional CD-ROM drives	Faster and newer CD-ROM drives, hard disk drives, floppy disk drives

There are in fact some drives that use a mixture of CLV and CAV. This is a compromise design that uses CAV when reading the outside of the disk, but then speeds up the spin rate of the disk while reading the inside of the disk. This is done to improve the transfer rates at the inside edge of the disk, which can be 60% lower than the rates at the outside of the disk in a regular CAV drive.

Loading Mechanism

The loading mechanism refers to the mechanical components that are responsible for loading CDs into the CD-ROM drive. There are two different ways that CD-ROM media are normally loaded into the CD-ROM drive.The most popular loading mechanism used today is the tray. With this system, a plastic tray, driven by gears, holds the CD. When the eject button is pressed the tray slides out of the drive, and the CD is placed upon it. The tray is then loaded back into the drive when the eject button is pressed a second time. Most drives will also respond to a slight "push" on the drive tray by activating the mechanism and retracting the tray.

Many older CD-ROM drives, and many higher-end drives even today, use caddies. These are small carriers made of plastic. A hinge on one side opens up to let you put a disk within the caddy, and a metal cover on the bottom slides out of the way to allow access to the CD by the drive. The caddy is inserted into the CD-ROM drive as a sort of "virtual cartridge". In fact, the CD inside the caddy is pretty similar to the way a 3.5" floppy disk works within its jacket--a media disk inside a plastic protective carrier with a sliding metal access panel. Of course the CD is still removable. Also, the CD caddies are much more solidly built.

Of the two mechanisms, the tray is far more common because it makes for a cheaper drive and also for cheaper use of the media. Most consumer-grade drives use trays for this reason. There are problems with these tray drives however:

- **Fragile Mechanism:** One problem is that the mechanism for moving the tray in and out of the drive is really not hard to break if it is mishandled. The CD, when placed in the tray, just sort of "sits there" loose, and if you put it in the tray off-center it is possible for the disk to get stuck in the tray when it retracts, potentially damaging both disk and drive.

- **Increased Handling:** Trays mean each disk must be handled a fair bit, which can increase the chances of wear, dirt accumulation and scratches on the media. Caddies eliminate virtually all handling of the individual disk.

- **No Vertical Orientation:** These drives cannot be side-mounted, as the CD would fall right out of the tray. This isn't a concern for most people but it is for some. There are in fact some CD-ROMs that have four tabs around the perimeter of the tray for holding the disk in place. This might work mounted vertically.

- Caddies are used on many high-end drives and are a much better mechanism, if you can afford to use them properly. This means that you basically need a caddy for each CD you use on a regular basis. Unfortunately, this can be an expensive proposition.

Connectors and Jumpers

The connectors and jumpers on a CD-ROM are similar in most ways to what you will find on a hard disk drive. Mercifully, CD-ROM drive manufacturers have done a much better job of being at least somewhat standardized in the use of jumpers and connectors, and even in where they are located on the drive. All CD-ROMs generally have their jumpers and connectors located at the back of the drive.

You will find a standard 4-pin power connector on the back of a regular internal CD-ROM drive, the same kind that is used for hard disk drives and most other internal devices. This is pretty universal and is found on most every drive. The other connections and jumpers depend on the interface that the drive is using; an IDE/ATAPI drive will use different ones than a SCSI drive for example. For ATAPI, you will find the standard 40-pin data connector, along with jumpers to select the drive as a master or slave device. For SCSI, you will find a 50-pin connector and jumpers to set the device ID and termination.

One connector that is found on a CD-ROM and not on a hard disk drive is the audio connector that goes to the sound card. This three- or four-wire cable is used to send CD audio output directly to the sound card so it can be recorded or played back on the computer's speakers.

Media Construction and Manufacture

Compact disks start as round wafers made from a polycarbonate substrate, measuring 120 mm (about 4.75 inches) in diameter and about 1.2 mm in thickness, which is less than 1/20th of an inch. These blanks are made into production CDs using a process not dissimilar to how old vinyl records were made.

The first step in the creation of a CD is the production of a master. The data to be recorded on the disk (either audio or computer data, there are many different formats) is created as an image of ones and zeros. The image is etched into the master CD using a relatively high-power laser (much more powerful than the one you would find in a regular CD player) using special data encoding techniques that use microscopic pits to represent the data. The master CD is then used to create duplicate master stamps.

The actual CDs are produced by pressing them with the master stamp. This creates a duplicate of the original master, with pits in the correct places to represent the data. After stamping, the entire disk is coated with a thin layer of aluminum (which is what makes the disk shine, and is what the laser reflects off when the disk is read) and then another thin layer of plastic. Then, the printed label is applied to the disk.

Many people don't realize that the data surface of the CD is actually the top of the disk. The media layer is directly under the CD label, and the player reads the CD from the bottom by focusing the laser through the 1.2 mm thickness of the CD's substrate. This is one reason why the bottom of the disk can have small scratches without impeding the use of the disk; they create an obstacle that the laser must look through, but they don't actually damage the data layer. On the other hand, scratches on the top of the disk can actually remove strips of the reflective aluminum coating, leaving the disk immediately unusable.

CDs are fairly hardy but are far from indestructible. They are reasonably solid but overly flexing them can make them unreadable. CD media should always be cared for properly. The use of caddies or jewel cases will protect them; in general, the less handling, the better.

Data Encoding and Decoding

Like hard disks and floppy disks, the compact disk is a digital storage medium. At the very lowest level, only two different values can be recorded on a disk: a one, or a zero. Magnetic disks record data using tiny magnetic fields, and the flux reversals that are detected by the read head as the disk moves from one type of field to another. Compact disks use a physical recording technique instead of a magnetic one.

The disk starts out totally flat. At each data-holding position on the disk, the CD is either left flat (these areas are called "lands") or is imprinted with a "pit", which is burned by a laser into the CD master, and then stamped into production CDs using a metal stamp made from the master. So as the disk spins, the laser traverses from lands to pits, many thousands per second. When the laser hits a land, it reflects cleanly off the aluminum coating, but when it hits a pit much of the light is diffused. The photodetector in the read head senses the difference and this is how it knows if the bit was a one or a zero.

While CDs are often referred to as having "tracks", this is actually imprecise. In fact, the entire CD is one very long, tightly-packed spiral. This is just like the single track on a phonograph record in concept, but there is a huge difference in scale. A standard CD has a spiral comprised of about 20,000 "tracks", so the spiral is in fact about three miles long! The tracks of the spiral are spaced about 1.6 microns apart. This is equivalent to a track density of about 16,000 tracks per inch, which exceeds that of even high-end hard disks today.

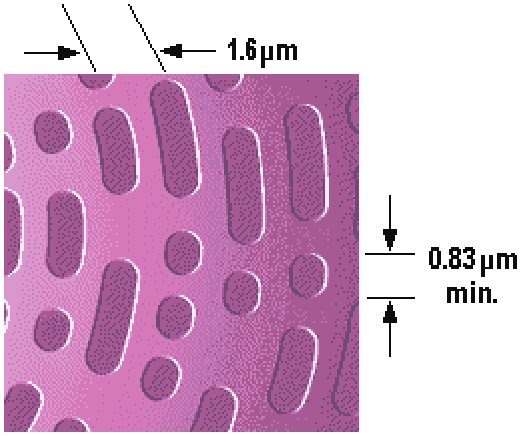
CD Capacity

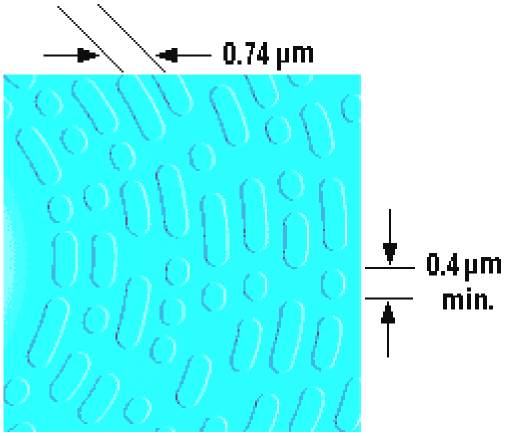
A standard CD has a capacity of about 74 minutes of standard CD audio music. There are extended CDs that can actually exceed this limit and pack more than 80 minutes on a disk, but these are non-standard. Regular CD-ROM media hold about 650 MB of data, but the actual storage capacity depends on the particular CD format used.

DIGITAL VERSATILE DISK (DVD)

At first glance, a DVD disc can easily be mistaken for a CD both are plastic discs 120mm in diameter and 1.2mm thick and both rely on lasers to read data stored in pits in a spiral track. And whilst it can be said that the similarities end there, it's also true that DVD's seven-fold increase in data capacity over the CD has been largely achieved by tightening up the tolerances throughout the predecessor system.

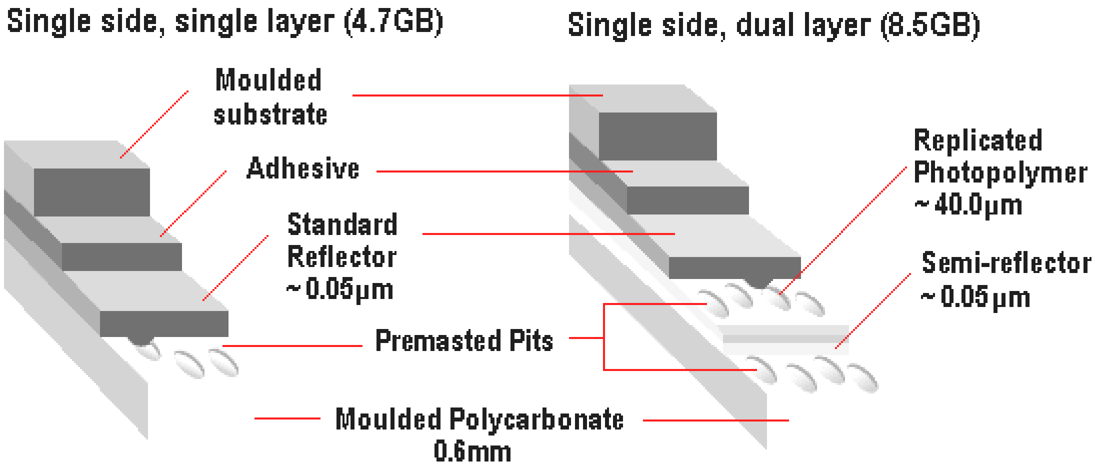
Technology



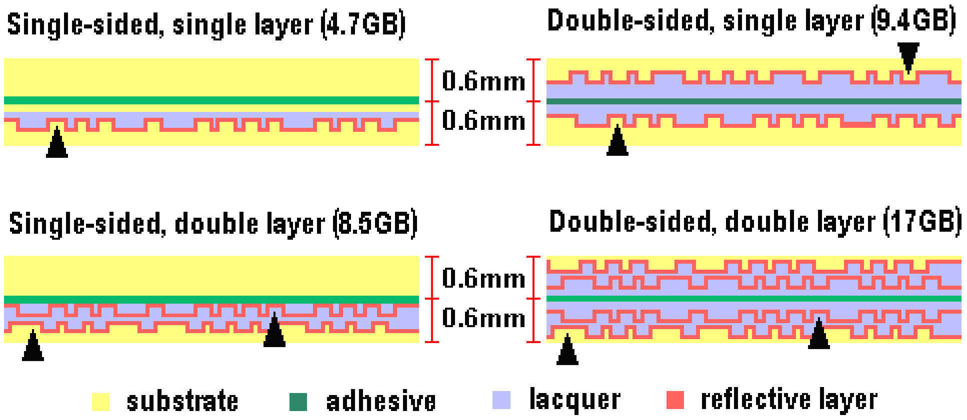


Firstly, the tracks are placed closer together, thereby allowing more tracks per disc. The DVD track pitch (the distance between each) is reduced to 0.74 micron, less than half of CD's 1.6 micron. The pits, in which the data is stored, are also a lot smaller, thus allowing more pits per track. The minimum pit length of a single layer DVD is 0.4 micron as compared to 0.834 micron for a CD. With the number of pits having a direct bearing on capacity levels, DVD's reduced track pitch and pit size alone give DVD-ROM discs four times the storage capacity of CDs.

The packing of as many pits as possible onto a disc is, however, the simple part and DVD's real technological breakthrough was with its laser. Smaller pits mean that the laser has to produce a smaller spot, and DVD achieves this by reducing the laser's wavelength from the 780nm (manometers) infrared light of a standard CD, to 635nm or 650nm red light.



Secondly, the DVD specification allows information to be scanned from more than one layer of a DVD simply by changing the focus of the read laser. Instead of using an opaque reflective layer, it's possible to use a translucent layer with an opaque reflective layer behind carrying more data. This doesn't quite double the capacity because the second layer can't be quite as dense as the single layer, but it does enable a single disc to deliver 8.5GB of data without having to be removed from the drive and turned over. An interesting feature of DVD is that the discs' second data layer can be read from the inside of the disc out, as well as from the outside in. In standard-density CDs, the information is always stored first near the hub of the disc. The same will be true for single- and dual-layer DVD, but the second layer of each disc can contain data recorded 'backwards', or in a reverse spiral track. With this feature, it takes only an instant to refocus a lens from one reflective layer to another. On the other hand, a single-layer CD that stores all data in a single spiral track takes longer to relocate the optical pickup to another location or file on the same surface.



Thirdly, DVD allows for allows for double-sided discs. To facilitate the focusing of the laser on the smaller pits, manufacturers used a thinner plastic substrate than that used by a CD-ROM, thereby reducing the depth of the layer of plastic the laser has to travel through to reach the pits. This reduction resulted in discs that were 0.6mm thick - half the thickness of a CD-ROM. However, since these thinner discs were too thin to remain flat and withstand handling, manufacturers bonded two discs back-to-back - resulting in discs that are 1.2mm thick. This bonding effectively doubles the potential storage capacity of a disc. Note that single-sided discs still have two substrates, even though one isn't capable of holding data.

Finally, DVD has made the structure of the data put on the disc more efficient. When CD was developed in the late 1970s, it was necessary to build in some heavy-duty and relatively crude error correction systems to guarantee the discs would play. When bits are being used for error detection they are not being used to carry useful data, so DVD’s more efficient and effective error correction code (ECC) leaves more room for real data.

File Systems

One of the major achievements of DVD is that it has brought all the conceivable uses of CD for data, video, audio, or a mix of all three, within a single physical file structure called UDF, the Universal Disc Format. Promoted by the Optical Storage Technology Association (OSTA), the UDF file structure ensures that any file can be accessed by any drive, computer or consumer video. It also allows sensible interfacing with standard operating systems as it includes CD standard ISO 9660 compatibility. UDF overcomes the incompatibility problems from which CD suffered, when the standard had to be constantly rewritten each time a new application like multimedia, interactivity, or video emerged.

The version of UDF chosen for DVD which - to suit both read-only and writable versions - is a subset of the UDF Revision 1.02 specification known as MicroUDF (M-UDF). Because UDF wasn't supported by Windows until Microsoft shipped Windows 98, DVD providers were forced to use an interim format called UDF Bridge. UDF Bridge is a hybrid of UDF and ISO 9660. Windows 95 OSR2 supports UDF Bridge, but earlier versions do not. As a result, to be compatible with Windows 95 versions previous to OSR2, DVD vendors had to provide UDF Bridge support along with their hardware.
DVD-ROM discs use the UDF Bridge format. (Note, Windows95 was not designed to read UDF but can read ISO 9660). The UDF Bridge specification does not explicitly include the Joliet extensions for ISO 9660, which are needed for long filenames. Most current Premastering tools do not include the Joliet extensions but it is expected that this feature will be added in due course. Windows98 does read UDF so these systems have no problem with either UDF or long filenames.

DVD-Video discs use only UDF with all required data specified by UDF and ISO 13346 to allow playing in computer systems. They do not use ISO 9660 at all. The DVD-Video files must be no larger than 1 GB in size and be recorded as a single extent (i.e. in one continuous sequence). The first directory on the disc must be the VIDEO_TS directory containing all the files, and all filenames must be in the 8.3 (filename.ext) format.

DAT

Acronym for DIGITAL AUDIO TAPE, a type of magnetic tape that uses a scheme called helical scan to record data. A DAT cartdrige is slightly larger than a credit card in width and height and contains a magnetic tape that can hold from 2 to 24 GB of data. It can support data transfer rates of about 2 Mbps. Like other types of tapes, DATs are sequential-access media.
The most common format for DAT cartridges is DDS(Digital Data Storage).

DAT Technology Overview

First developed for the audio electronics market, DAT technology was first applied in computer peripherals in the late 1980's. Unlike traditional magnetic tape audio cartridge products, DAT technology proves inherently reliable through the helical scan recording method, which provides a high recording density with a very low error rate.

All DAT products, including computer implementations, use the helical scan recording method. This recording method has been used in professional video tape recorders (VTR's) since 1956 and in home video cartridge recorders (VCR's) since 1974. In 1986, DAT products using helical scan technology were first developed for audio applications. DAT consumer products are specifically designed for digital audio recording and playback.

Helical Scan Recording

Helical scan recording was originally developed as a method of efficiently recording high-quality television signals on a relatively slow moving tape. It requires that both the tape and the recording head move simultaneously. This recording method results in an extremely high recording density, far higher than can be achieved with stationary-head devices such as 1/2-inch open-reel or 1/4-inch cartridge tapes.

In helical scan recording, both the read and write heads are located on a rapidly rotating cylinder or drum. The cylinder is tilted at an angle in relation to the vertical axis of the tape. As the tape moves horizontally, it wraps around the part of the circumference of the cylinder (102°) so that the head enters at one edge of the tape and exits at the other edge before the tape unwraps. The horizontal movement of the tape in combination with the angular movement of the cylinder causes the track to be recorded diagonally across the tape, rather than straight down its length. The resulting recorded track, nearly one-inch, is approximately eight times longer than the width of the tape.

DDS Recording Format

This standard format was co-developed by DDS manufacturers to support DAT devices as computer peripherals. The objectives of DDS are to maximize storage capacity and performance; to facilitate data interchange; to provide compatibility with existing tape storage command sets; and to provide extremely fast random access.

The DDS format also takes advantage of the helical scan recording method and the inherent error correction capability of the DAT technology to augment error detection and correction. The format consists of a finite sequence of data groups with each data group being a fixed-length recording area. A data group is made up of 22 data frames and 1 ECC frame; each frame is made up of two helical scan tracks. The advantages of the fixed-length data group are that ECC is easily generated, and buffering requirements are simplified. Although data groups are fixed-length and always contain 22 data frames, the DDS format is designed such that variable-length computer records can be stored in the fixed-length data groups.

The media onto which the information is stored is in the form of a tape; similar to that of an audio casette(tape). A magnetically coated strip of plastic on which data can be encoded. Tapes for computer are similar to tapes used to store music.

Storing data on tapes is considerably cheaper than storing data on disk. Tapes also have large storage capacities, ranging from a few hundred kilobytes to several gigabytes. Accessing data on tapes, however, is much slower than accessing data on disks. Tapes are sequential-access media, which means that to get to a particular point on the tape, the tape must go through all the preceding points. In contrast, disks are random-access media because a disk drive can access any point at random without passing through intervening points.

Because tapes are so slow, they are generally used only for long-term storage and backup. Data to be used regularly is almost always kept on a disk. Tapes are also used for transporting large amounts of data. Tapes come in a variety of sizes and format.

Read-After-Write

The Read-After-Write (RAW) technique provides a means of verifying that host data was written on the tape correctly by applying a read check immediately after writing the data to tape. The read check is a comparison of the actual signal quality versus a predetermined acceptable threshold level. If a frame is identified as bad, it is rewritten later down the tape. The bad frame is not necessarily rewritten immediately. It can be rewritten after three, four, or five other frames have been written. Any frame can be rewritten multiple times to provide for skipping over bad areas on the tape.

Excessive consecutive rewrites typically signal a degraded media condition. In these cases it is best to discontinue use of the tape in question, and continue with a piece of good media. During a read or restore operation the threshold level is reduced to maximize the likelihood that data can be successfully retrieved from tape. The combination of the elevated read threshold during write and reduced threshold during read ensures that data is written with the highest possible margin, and that recorded data can be read or retrieved with the highest possible confidence.

UNIVERSAL SERIAL BUS (USB)

Universal Serial Bus (USB) is a connectivity specification developed by the USB Promoter Group. USB is aimed at peripherals connecting outside the computer in order to eliminate the hassle of opening the computer case for installing cards needed for certain devices. USB provides for ease of use, expandability, and speed for the end user.

USB is enjoying broad adoption in the marketplace today. Over 1000 devices have passed compliance testing. The next version of USB, dubbed USB 2.0, is a higher speed (480Mbs) version that is also fully compatible with USB 1.1. Recently at the USB 2.0 Developers Conference the USB 2.0 Promoters Group stated that they are expecting shipments of USB 2.0 systems and peripherals by the 2nd half of 2000.

ADVANTAGES OF USB

- Supported peripheral units

USB pilots peripheral units via a cable whose length cannot exceed five metres per segment. The units include mouse, keyboards, printers, modems, joysticks, scanners, telephones, video cameras, network interfaces, graphic tablets and a host of others. USB lets you expand your personal computer, and is an economic data transfer system up to a maximum speed of 460 Mbit/second, conveying voice, audio and video in compressed format.

- Automatic detection of peripheral units

One of USB's strikingly interesting characteristics is ease of installation - quicker and surer also when automatically detecting peripheral units.The devices, connected to Universal Serial Bus, make use of Plug and Play technology. They can be fitted and/or removed without switching off the computer, thus cutting down on recognition time of connected hardware and increasing output in both work and amateur environments.

- Connection

Similarly to the SCSI bus, USB is able to connect cascade-wise up to 127 devices and has a band width reaching 460 Megabits/second. In addition to conveying data, the cable also supplies power to low voltage appliances. The connection lay-out is a multi-level star, with maximum length of five meters per single segment. Each level consists of a star that linked to a single concentrator (hub) reprising the layout of twisted-pair local networks. The hub can either be passive, in which case it receives power from the computer and re-distributes it to down stream devices, or active, in which case it generates power itself for the downstream devices. So USB is a quicker communication standard than the out-of-date parallel and serial ports which it is expected to replace in the future.

- Power supply

The USB port supplies a maximum of 500 milliAmperes at 5 Volt. When connected to a passive hub 100 mA are absorbed by the hub and the other 400 are distributed to the output ports, 100 mA for each.
If an active hub is used, a maximum of 500 mA can be supplied to each ouptut port.

- Data transmission

It can reach 1.5 Mbit/second (low-speed USB) for slow devices (keyboard, mouse, pen, joystick and the like, remotely configurable monitors, and virtual devices), attains 12 Mbps (Full-speed USB) for medium speed devices (audio appliances, ISDN interfaces, telephones and exchanges) and ultimately reaches a speed of 460 Mbit/second (high-speed USB) in devices such as hard-disks and digital monitors. However, slow and rapid peripheral units can coexist on the same chain.

USB HOST

The host of a USB chain is the computer from which the chain spreads out. The host controls chain operation and all data transiting on that given chain are exchanged at the computer. No peripheral units may enter the chain without explicit permission by the host, which first checks the peripheral unit's compatibility with the chain. The interface used by the computer to connect to the chain (USB sockets and relevant control circuits) and the relevant software are known as USB controller.

USB Controller

The electronic and logic interface the computer uses to link up to a chain of USB peripheral units. It includes the USB sockets, relevant circuits, and control software. In computers without a USB controller integrated in the motherboard, a small expansion card for PCI bus can be used as a controller.

A Technical Introduction to USB 2.0

This document introduces the features and benefits of USB 2.0 and describes its impact to users, PC manufacturers and PC peripheral manufacturers. Following a recap of USB 1.1, this paper overviews the technical aspects of USB 2.0 whose details are in the specification draft released in October.

USB 2.0 Summary

A core team from Compaq, Hewlett Packard, Intel, Lucent, Microsoft, NEC and Philips is leading the development of the USB Specification, version 2.0, that will increase data throughput by a factor of 40. This backwards-compatible extension of the USB 1.1 specification uses the same cables, connectors and software interfaces so the user will see no change in the usage model. They will, however, benefit from an additional range of higher performance peripherals, such as video-conferencing cameras, next-generation scanners and printers, and fast storage devices, with the same ease-of-use features as today's USB peripherals.

Impact to User

From a user's perspective, USB 2.0 is just like USB, but with much higher bandwidth. It will look the same and behave the same, but with a larger choice of more interesting, higher performance devices available. Also, all of the USB peripherals the user has already purchased will work in a USB 2.0-capable system.

Impact to PC Manufacturer

USB 2.0 will provide system manufacturers the ability to connect to high performance peripherals in the least expensive way. The additional performance capabilities of USB 2.0 can be added with little impact to overall system cost.Indeed, high-bandwidth interfaces such as SCSI adapters may no longer be required in some systems, leading to a net saving of system cost. Simpler construction will result since only USB connectors will be needed on many future PCs. Today's ubiquitous USB connectors will become USB 2.0, superceding USB 1.1.

Impact to Peripheral Manufacturer

Today's USB devices will operate with full compatibility in a USB 2.0 system. The added capabilities of USB 2.0 will expand the market segment for USB peripherals, while enabling retail products to transition with the installed base. Support of USB 2.0 is recommended for hubs and higher bandwidth peripherals. Designing a USB 2.0 peripheral will be a similar engineering effort to that of designing a USB 1.1 peripheral. Some low-speed peripherals, such as HID devices, may never be redesigned to support the USB 2.0 high-speed capability in order to maintain the absolute lowest cost.

Historical Perspective – Universal Serial Bus

The Universal Serial Bus was originally developed in 1995 by many of the same industry leading companies currently working on USB 2.0. The major goal of USB was to define an external expansion bus which makes adding peripherals to a PC as easy as hooking up a telephone to a wall-jack. The program's driving goals were ease-of-use and low cost. These were enabled with an external expansion architecture, as shown in

Figure 1, which highlights:

- PC Host Controller Hardware And Software,
- Robust Connectors And Cable Assemblies,
- Peripheral Friendly Master-Slave Protocols,
- Expandable Through Multi-Port Hubs.

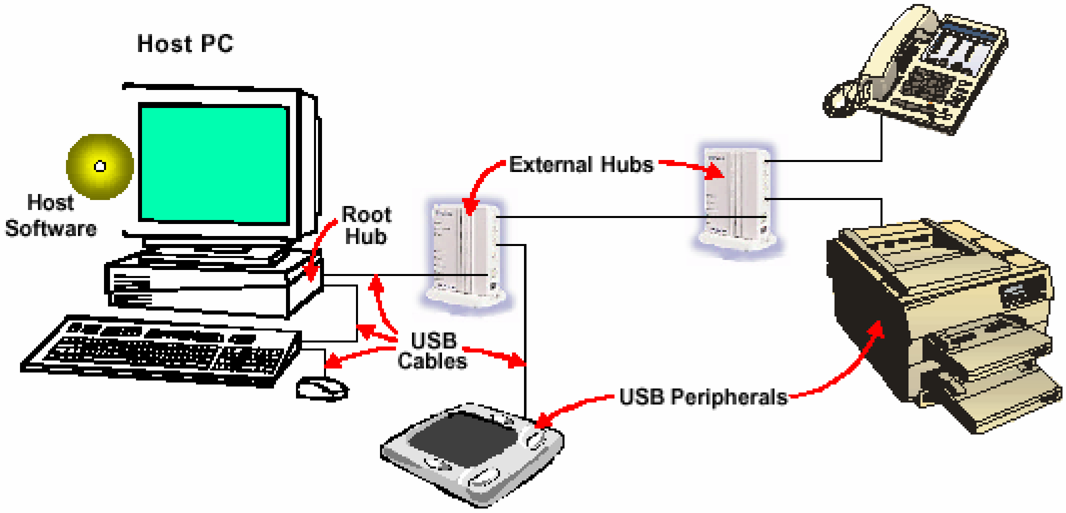


Figure 1. Example USB 1.1 System Configuration

Today, USB is enjoying tremendous success in the marketplace, with most peripheral vendors around the globe developing products to this specification. Virtually all new PCs come with one or more USB ports on the box. In fact, USB has become a key enabler of the Easy PC Initiative, an industry initiative led by Intel and Microsoft to make PCs easier to use. This effort sprung from the recognition that users need simpler, easier to use PCs that don't sacrifice connectivity or expandability. USB is one of the key technologies used to provide this.

Recap of USB 1.1 Operation

An understanding of the roles of each of the major elements within a USB 1.1 system will better show the evolutionary step that USB 2.0 provides.

Role Of Host PC Hardware And Software:

The role of the system software is to provide a uniform view of IO system for all applications software. It hides hardware implementation details so that application software is more portable. For the USB IO subsystem in particular, it manages the dynamic attach and detach of peripherals. This phase, called ENUMERATION, involves communicating with the peripheral to discover the identity of a device driver that it should load, if not already loaded. A unique address is assigned to each peripheral during enumeration to be used for run-time data transfers. During run-time the host PC initiates transactions to specific peripherals, and each peripheral accepts it's transactions and responds accordingly. Additionally the host PC software incorporates the peripheral into the system power management scheme and can manage overall system power without user interaction.

Role Of The Hub:

Besides the obvious role of providing additional connectivity for USB peripherals, a hub provides managed power to attached peripherals. It recognizes dynamic attachment of a peripheral and provides at least 0.5W of power per peripheral during initialization. Under control of the host PC software, the hub may provide more device power, up to a maximum of 2.5W, for peripheral operation. A newly attached hub will be assigned its unique address, and hubs may be cascaded up to five levels deep. During run-time a hub operates as a bi-directional repeater and will repeat USB signals as required on upstream (towards the host) and downstream (towards the device) cables. The hub also monitors these signals and handles transactions addressed to itself. All other transactions are repeated to attached devices. A hub supports both 12Mb/s (full-speed) and 1.5Mbs (low-speed) peripherals.

Role Of The Peripheral.

All USB peripherals are slaves that obey a defined protocol. They must react torequest transactions sent from the host PC. The peripheral responds to control transactions that, for example, request detailed information about the device and it's configuration. The peripheral sends and receives data to/from the host using a standard USB data format. This standardized data movement to/from the PC host and interpretation by the peripheral gives USB it's enormous flexibility with little PC host software changes. USB 1.1 peripherals can operate at 12Mb/s or 1.5Mb/s.

What does USB 2.0 add?

USB 2.0 is an evolution of the USB 1.1 specification, providing a higher performance interface. Today's USB 1.1 connectors and full-speed cables will support the higher speeds of USB 2.0 without any changes. Characterization that has already been done on these cables confirms his compatibility. Analysisthat has been done by the electrical team suggests that a target of 480Mbs is achievable on USB 2.0. USB 2.0 will specify a microframe, which will be 1/8 th of a 1msec frame. This will allow USB 2.0 devices to have small buffers even at high data rates. Support of higher speed USB 2.0 peripherals connected to a hub assumes USB 2.0 hubs as shown in Figure 2. The higher transmission speed is negotiated on a device-by-device basis and if the higher speed is not supported by a peripheral, then the link operates at a lower speed of 12Mb/s or 1.5Mb/s as determined by the peripheral.

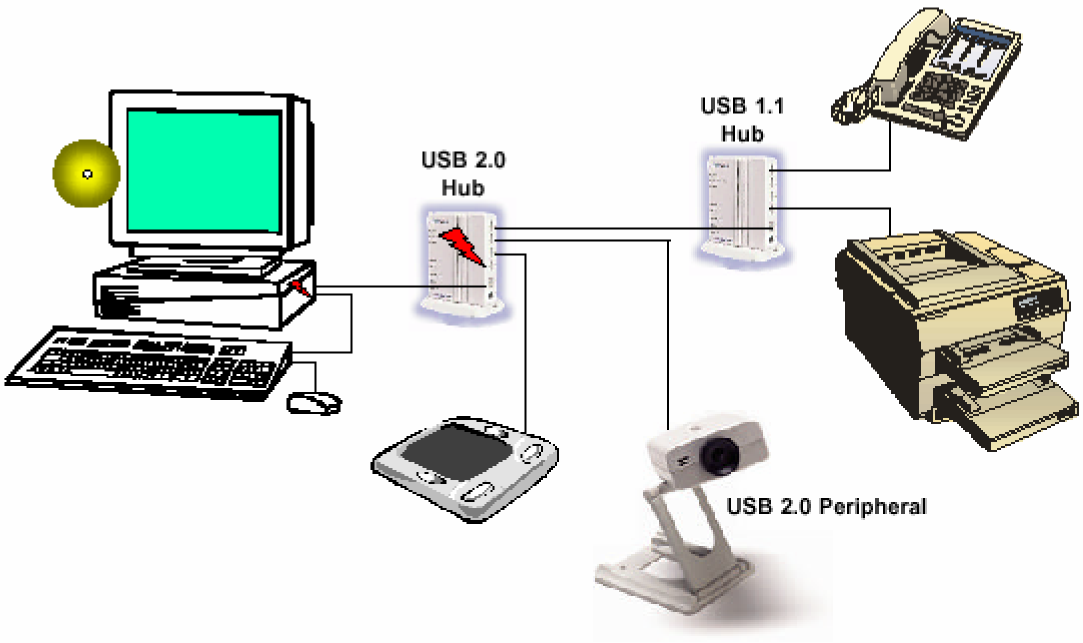


Figure 2. Example Future USB 2.0 System Configuration

Figure 2. Example Future USB 2.0 System Configuration

As shown in Figure 2, high-speed connections were negotiated between the root hub and the external USB 2.0 hub and between the external USB 2.0 hub and the video-conferencing camera (a USB 2.0 peripheral). All other connections are at USB 1.1 data rates, i.e. 12Mb/s automatically downshifting to 1.5Mb/s for low-speed peripherals. Note that the external USB 2.0 hub has different signaling rates on its ports. Using a 40x multiplier for USB 2.0, the USB 2.0 hub example in Figure 2 has an input rate of 480Mb/s and output rates of 480Mb/s for attached high speed USB 2.0 peripherals, and 12Mb/s or 1.5Mb/s for attached USB 1.1 peripherals. Any downstream port of a USB 2.0 hub can support attachment of any speed USB device. The USB 2.0 hub must match the data rates sent out of its downstream ports to the data rate appropriate to the attached device. This increases the hub's role in a USB 2.0 system as outlined below.

Overview of USB 2.0 Operation

The external view of a USB 2.0 system looks no different from a USB 1.1 system as evidenced by comparing Figures 1 and 2. A casual observer will not be able to discriminate between the two system versions – which is exactly the view the user should have. However, the user will have to be able to distinguish between USB 2.0 hubs and USB 1.1 hubs in order to optimize the placement of USB 2.0 high-speed devices. The roles of the components of the 2.0 system have minor changes from the roles in a USB 1.1 system.

Role of Host PC software.

Current applications software on the PC continues to operate with USB 1.1 peripherals and is unchanged. The system software will comprehend the increased capabilities of USB 2.0 peripherals so that it can optimize performance. The system software will also detect sub-optimal configurations, i.e. a USB 2.0 peripheral attached to a USB 1.1 hub, and will alert the user and recommend a better configuration for attaching the peripherals. New applications will be written to take advantage of the higher speed capabilities and ease-of-use of USB 2.0 peripherals and drivers.

Role Of The Hub.

A USB 2.0 hub accepts high-speed transactions at the faster frame rate and must deliver them to high-speed USB 2.0 peripherals and USB 1.1 peripherals. This data rate matching responsibility will require some increased hub complexity and temporary buffering of the incoming high-speed data. In the simplest case of communicating with an attached USB 2.0 peripheral, the hub repeats the high-speed signals on appropriate USB 2.0 upstream and downstream cables just as a USB 1.1 hub repeats full and low-speed signals today on USB 1.1 devices. This allows USB 2.0 peripherals to utilize the majority of USB 2.0 bandwidth.

To communicate with USB 1.1 peripherals, a USB 2.0 hub contains a mechanism that supports the concept of matching the data rate with the capabilities of the downstream device. In other words, the hub manages the transition of the data rate from the high speed of the host controller to the lower speed of a USB 1.1 device. This feature of USB 2.0 hubs means that USB 1.1 devices can operate along with USB 2.0 devices and not consume disproportionate amounts of USB 2.0 bandwidth. This new hub architecture is intended to be as simple and cost effective as possible, and yet deliver the full capabilities of 1.1 connections. The new USB 2.0 hub will be completely defined in the USB 2.0 specification providing clear implementation guidelines for hub vendors and allowing a single software driver to service USB 2.0 hub products from multiple vendors.

Role of the peripheral.

Current peripheral products do not require ant changes to operate in a USB 2.0 system. Many human interface devices such as mice, keyboard, and game pads will not require the additional performance that USB 2.0 offers and will remain as full or low speed peripherals as defined by USB 1.1. The higher data rate of USB 2.0 will however open up the possibilities of exciting new peripherals. Video conferencing cameras will perform better with access to higher bandwidth. Next generation higher speed and higher resolution printer and scanner devices will be enabled at the high end. High-density storage devices such as R/W DVD and high capacity CD-ROM jukeboxes will also be enabled by USB 2.0. These devices require minor changes to the peripheral interface, as defined in the USB 2.0 specification. Overall the additional cost to support USB 2.0 is expected to be minimal to the peripheral. Both USB 1.1 and USB 2.0 devices will inter operate in a USB 2.0 system.

Summary

The USB specification is currently at version 1.1 and supports a wide range of products. Many vendors are now moving towards USB drawn by its inclusion on virtually all PC platforms and its ease of use. More and more types of innovative new peripherals are taking advantage of USB, which further enhance the available USB product portfolio.

The version 2.0 specification that is under development is an evolutionary step that increases performance capabilities at low cost for USB peripherals in a backward compatible fashion. It is expected to be broaden the market for new and higher performance PC peripherals and supercede USB 1.1 on future PC’s.