



UNESCO-NIGERIA TECHNICAL &  
VOCATIONAL EDUCATION  
REVITALISATION PROJECT-PHASE II



# **NATIONAL DIPLOMA IN ELECTRICAL ENGINEERING TECHNOLOGY**



## **Computer Hardware I**

**COURSE CODE: EEC117**

**YEAR I- SEMESTER I**

**THEORY/PRACTICAL**

Version 1: December 2008

## TABLE OF CONTENTS

<b>Department</b>	Electrical/Electronics Engineering
<b>Subject</b>	Computer Hardware I
<b>Year</b>	1
<b>Semester</b>	1
<b>Course Code</b>	EEC 117
<b>Credit Hours</b>	3
<b>Theoretical</b>	1
<b>Practical</b>	2

## INTRODUCTION TO COMPUTER HARDWARE I

WEEK 1.....	1
Computer History	
WEEK 2.....	10
Computer Components	
WEEK 3.....	18
CPU-Central Processing Unit	
WEEK 4.....	29
Motherboard	
WEEK 5.....	37
Motherboard Form factor	
WEEK 6.....	40
Serial Ports Parallel ports	
WEEK 7.....	45
USB Installation and Configuration	
WEEK 8.....	49
Visual Display Unit- CRT Monitors	
WEEK 9.....	54
LCD Monitors	
WEEK 10.....	57
Printers	
WEEK 11.....	63
Installing Printers	
WEEK 12.....	69
Modems	
WEEK 13.....	72
Installation of Dial-up modems	
WEEK 14.....	79
Introduction to Networking	
WEEK 15.....	84
Computer Network Sharing	

## Computer History

### Napier's Bones

c. 1600

Scotland

Napier's Bones are portable single-digit multiplication tables that can be arranged to show the product of multiplying almost any multi-digit number by a single digit. While the tables were often made of wood or paper, higher quality sets were fashioned from ivory or bone, giving the tables their name.



COMPUTER  
HISTORY  
MUSEUM

Although John Napier invented them in the late 1500s, a description of the “bones” was only published at the end of his life, in his book *Rabdologiæ*, because he was concerned that others would take credit for his idea.

While Napier also discovered logarithms, the bones are unrelated to that mathematical concept.

### Sets of Napier's bones ready for use



### J Lyons, Accounting office, c.1900

Devices to make calculation easier have existed for thousands of years. From the abacus to the mechanical desktop calculator, from analog bombsights to the WW II Enigma encryption

machine, this section highlights some of these devices and the problems they were invented to solve.

### Hollerith Census Machine

1889

Department of the Census, United States

Herman Hollerith invented the first automated tabulating system using punched cards. Initially designed to process the 1890 US census, his system became the basis for punched card accounting machines for most of the twentieth century. Hollerith became wealthy as his Tabulating Machine Company expanded beyond government customers to include railroads, insurance companies, and manufacturers.



Hollerith sold his patent rights in 1911 to a holding company (C-T-R) that was renamed International Business Machines Corporation (IBM) in 1924. Punched card machines grew more sophisticated and bridged the gap between the paper and electronic ages.

### Production Machines

The initial development of the electronic digital computer was by the military-funded academic and research establishment. The main focus of innovation and production soon moved to the commercial sector not only because of the large potential market but also because of the money and resources required. Building computers started to be a money-making business in the mid 1950's. Shown here is a mercury memory delay line memory from the Univac I, the first commercial electronic digital computer ever made. The Bendix G15 and LGP-30 represent two other early and influential drum-based computers. The Electrodata arithmetic unit shows a small section of a large mainframe computer system of the time. With the Cold War in full swing, the SAGE computer was developed to detect Russian manned bombers armed with nuclear weapons. SAGE computers represented the state-of-the-art in late 1950s computer technology: each installation had over 50,000 vacuum tubes, weighed 250 tons, and consumed three million watts of power—enough to power 2,000 homes.



### High Level Languages

The first computers had to be programmed in a numerical “machine language” that was directly executed by the computer from its main memory. Writing big programs was a tedious and error-prone task. In the early 1950s, “assembler language” programs were written using alphabetic symbols instead of numbers. The first “high-level” languages developed in the late 1950s allowed programs to be written using more understandable mathematical formulas or even English words. The most popular languages were FORTRAN, for scientific programming, and COBOL, for business programming. Eventually dozens of different high-level languages were created to ease the programming task for specific applications.

### 1970s - 1990s

During the 1970s, computers moved into the home in the form of microprocessor-based personal computers and game systems. The introduction of the IBM PC in 1981 was the most important event of that decade, creating a standard that is still in use today. More user-friendly machines such as the Apple Macintosh (based on the Xerox Alto), combined with more sophisticated software, resulted in computers that were inexpensive and powerful, yet easy to use. Here you will see some of these machines, as well as robots, printers, and computer graphics technologies of the time.

### IBM Personal Computer

1981

IBM Corporation, United States

Although IBM’s first personal computer arrived nearly ten years after others were available, the IBM Personal Computer (PC) instantly legitimized and expanded the market. Unlike most other contemporary IBM products, the PC incorporated both hardware (the Intel 8088 microprocessor) and software made by other companies. IBM published design details in their manuals that encouraged others to make copies or “clones” of the original machine, often with improved functionality. The IBM PC architecture quickly became an industry standard.



## Computer Types

### Desktop



**Figure 1: desktop computer**

A **desktop computer** is an independent personal computer that is made especially for use on a desk in an office or home. The term is used mainly to distinguish this type of personal computer from portable computers and laptops, but also to distinguish other types of computers like the PDA, server or mainframe.

Desktops are currently the more affordable and most common computers, and are frequently used by businesses, schools, households and other organizations. Nearly all modern *desktop computers* are modular, meaning that the components can easily be replaced or upgraded. A *desktop computer* can also refer to a computer whose case is oriented horizontally (usually, the monitor is placed on top of the case). Such cases are called *Desktops* as opposed to *Towers*.

### Hardware

The [hardware](#) in a desktop computer is modular, making it easy for someone with intermediate knowledge of a computer to modify one. The internal hardware of a basic desktop computer consists of [RAM](#), [CPU](#), a [motherboard](#), a [graphics card](#) and a [sound card](#) as well as additional onboard cards for [Ethernet](#) and other ports.

Desktop computers usually have a separate [monitor](#), the only modern exceptions being some Apple computers such as the [iMacs](#) and [eMacs](#). User-input peripherals such as [keyboards](#) and [mice](#) are attached to the computer's ports, as well as other peripherals such as [printers](#). Most of the functionality is contained within the case, but some components can be either external or internal (such as various [storage devices](#) or [modems](#)). The case is often placed on or under the desk. Sometimes, it can be placed underneath the monitor depending on the size and shape.

Desktop computers are generally more affordable than [notebook computers](#). This is because no extra effort is needed to miniaturize the components, or to manufacture components that use up less power.

## Components

- Fan- cools the computer
- Motherboard
- Hard drive- long-term internal data storage
- CD-ROM Drive- reads most or all types of CDs
- Processor
- CPU Cooler- cools the processing unit
- RAM- short-term data storage
- Video card- codes video output and delivers to monitor
- Sound card- codes sound output and delivers to speakers
- Speakers
- Floppy Drive
- Modem Used to access data via a telephone line. Used to access bulletin board systems and the Internet.
- Network card - Allows the computer to be connected to a computer network where it can communicate with other computers. Also used as a means of accessing the Internet.
- Power Supply- distributes power to the various components.

## Ports

All desktop computers have ports which are used to plug external devices into the computer such as monitors, keyboards, printers and scanners.

- USB - Used for the majority of peripherals
- Ethernet - Used for networking and broadband Internet connections
- Modem - Used for data access via a Telephone line (eg. dial-up Internet connections)
- Headphone jack - Used for connecting sound devices
- Serial - Used for connecting any device that uses a serial connector
- Parallel - Used for connecting any device that uses a parallel connector
- PS/2 - Used for computer mice and keyboards
- Video (VGA) - Used for connecting a monitor / projector
- Power - Used for connecting the power lead
- FireWire/IEEE 1394 - Used for connecting external hard drives and camcorders



## Notebook

A **laptop** computer (also known as **notebook** computer) is a small mobile personal computer, usually weighing from 1 to 3 kilograms (2 to 7 pounds). Terms for subtypes of notebooks (and related computer types) include:



- Notebooks smaller than an A4 sheet of paper and weighing around 1 kg are sometimes called **sub-notebooks** or **subnotebooks**.
- Notebooks weighing around 5 kg are sometimes termed **desknotes** (desktop/notebook).
- Powerful laptops (often heavy) designed to compete with the computing power offered by a typical desktop are often known as **desktop replacements**.
- Computers larger than PDAs but smaller than notebooks are also sometimes called **palmtops**.

Laptops usually run on batteries, but also from adapters which also charge the battery using mains electricity.

Laptops are capable of many of the same tasks that desktop computers perform, although they are typically less powerful for the same price. Laptops contain components that are similar to those in their desktop counterparts and perform the same functions but are miniaturized and optimized for mobile use and efficient power consumption. Laptops usually have liquid crystal displays and use SO-DIMM (Small Outline DIMM) modules (rather than the larger DIMMs used in desktop computers) for their RAM. In addition to a built-in keyboard, they may utilize a touchpad (also known as a trackpad) or a pointing stick for input, though an external mouse or keyboard can usually be attached.

## Parts

Many parts for a laptop computer are smaller, lighter, or otherwise adapted from the corresponding part in a desktop computer:

Most modern laptops use an active matrix display with resolutions of 1024 by 768 pixels (XGA) and above, screen sizes 10 inch (250 mm) or larger, and have a PC-Card expansion bay for expansion cards, formerly called PCMCIA. Internal hard disks are smaller—2.5 inch (64 mm) compared to the standard desktop 3.5 inch (90 mm) drive—and usually have lower performance and power consumption. Display adapters and sound cards are integrated. Modern laptops can often handle sophisticated games, but tend to be limited by their fixed screen resolution and display adapter type.

Notebook processor There are a wide range of notebook processors available from Intel (Pentium M (with Centrino technology), Celeron, Mobile Pentium 4 and Mobile Pentium 4-M), AMD (Athlon, Turion 64) which develops and manufactures for the different Microsoft operating systems. Motorola and IBM develops and manufactures the PowerPC chips for Apple notebooks. Generally, notebook processors are less powerful than their desktop counterparts, owing to the need to conserve electricity and reduce heat output. However, the PowerPC G3 and G4 processor generations have been able to offer almost the same performance as their desktop versions, limited mostly by lower performance in other parts of the system bus (bandwidth and peripheral units) in Apple's notebooks.

Some parts for a modern laptop have no corresponding part in a desktop computer:

Current models use lithium ion batteries, which have largely replaced the older nickel metal-hydride technology. Typical battery life for most laptops is two to five hours with light-duty use, but may drop to as little as one hour with intensive use. Batteries gradually degrade over time and eventually need to be replaced, depending largely on the charging and discharging pattern, from one to five years.

Docking stations may be used for expanding connectors and quickly connecting many components to the laptop, although they are falling out of favour as laptops' integral capabilities increase and USB allows several peripherals to be connected through one plug.

Most laptops are powered or recharged from an external AC converter that usually takes the form of a plain black rectangular box. These devices weigh about 500 g (about 1 lb) and often take the name "power brick." Apple laptops have a lighter, white and more stylish, charging device.

### **Upgradability**

Laptops generally cost more than a desktop computer of similar specification. Performance is usually lower than that of a comparable desktop because of the compromises necessary to keep weight and power consumption low.

Upgradability is severely limited: typically only the RAM and hard drive can be changed. Because nearly all functions are integrated into the proprietary-design mainboard theoretically to save space and power, laptops are difficult to repair and upgrade. Outright replacement of faulty parts can include the display screen, drives, daughterboards, modem,

[storage devices](#) and other components, but repair costs can be high, even when feasible (low upgradability). There is not a standard for [A4](#)-size laptops.

### Performance

However, newer types of laptops now rival desktops. These [desknets](#) or *desktop replacements* are the result of the development of more powerful [batteries](#), and the practice of installing desktop components directly into desknets, making them equivalent in performance with desktops of similar specifications, albeit much larger than their laptop predecessors. As a result desknets are generally too bulky to carry around, and most people who use these computers at their place of work will tend to carry them around less frequently.

The relative difference in performance between desktops and desknets has therefore gradually decreased as developers continually attempt to upgrade the performance of desknets.

However, while laptops continue to provide the [mobility](#) which desknets may not possess, sales of standard laptops have remained high regardless of the extra performance desknets provide.

Some companies who market "laptops" with the full power of a desktop, or even a server, often misrepresent what the concept of a laptop encompasses. For example, [Vertegri](#) of Canada once sold full [Macintosh clone](#) machines in a large laptop case, with no battery; and [Tadpole Computers](#) line of [SPARC](#) laptops have everything from dual processors to full-size [PCI](#) slots, but some models again have no battery. These machines are closer to the *Transportables* of an earlier time, but use a standard laptop form factor.

### Personal digital assistant PDA

**Personal digital assistants (PDAs or palmtops)** are handheld devices that were originally designed as personal organizers, but became much more versatile over the years. A basic PDA usually includes a clock, date book, address book, task list, memo pad, and a simple calculator. One major advantage of using PDAs is their ability to synchronize data with a PC or home computer.

The term "personal digital assistant" was coined on January 7, 1992 by John Sculley at the Consumer Electronics Show in Las Vegas, Nevada, referring to the Apple Newton. Earlier devices like the Psion and Sharp Wizard have the functionality to be considered PDAs, however.



The currently major PDA operating systems are:

- Palm OS - owned by Palm, Inc.

- Windows Mobile (Pocket PC), (based on the Windows CE kernel) - owned by Microsoft
- BlackBerry - owned by Research In Motion
- Many operating systems based on the Linux kernel - free (not owned by any company) These include:
  - GPE - Based on gtk/X11
  - OPIE/Qtopia - based on Qt/E Qtopia is developed by Trolltech, OPIE is a fork of Qtopia developed by volunteers
- Symbian OS (formerly EPOC) owned by Ericsson, Panasonic, Nokia, Samsung, Siemens and Sony Ericsson

Many PDAs run using a variation of the ARM architecture (usually denoted by the Intel Xscale trademark). This encompasses a class of RISC microprocessors that are widely used in mobile devices and embedded systems, and its design was influenced strongly by a popular 1970s/1980s CPU, the MOS Technology 6502. In its classic instant-on, silent operation, restricted storage and space, and single-tasking UI approach, the PDA can be seen as a logical descendent of the low power consumption, compact, limited capacity home computer popular during the late 1970s and 1990s. This class of machines has been largely replaced by descendents of the IBM PC that generally feature long boot-up times, fast execution CPUs that require active, noisy cooling, and very large capacity hard drives that produce additional noise and heat.

According to a Gartner market study, the overall market for PDAs shrank by 5% in the first quarter (Q1) of 2004, compared to Q1 2003, with marketshare resolving as follows (by operating system):

- Palm OS for Palm, Inc. PDAs and some other licensees- 40.5% (stable)
- Windows Mobile for PDAs that comply with the Microsoft's Pocket PC specifications - 40.4% (slightly increasing)
- BlackBerry OS for BlackBerry PDA (produced by Research In Motion) - 14.8% (strongly increasing)
- Various operating systems based on the Linux kernel for various special designed PDAs (many other supported) - 1.9% (stable)
- Other - 2.4% (strongly decreasing)

The reason usually cited for this decline is the growing capabilities of smartphones — mobile phones with PDA-like communication abilities.

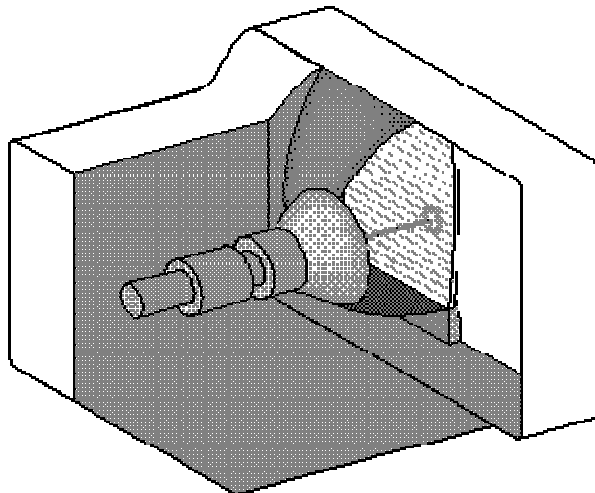
## Computer Components

### Computer Display (Monitor)

A **computer display, monitor** or **screen** is a [computer peripheral](#) device capable of showing still or moving images generated by a computer and processed by a [graphics card](#). Monitors generally conform to one or more [display standards](#). Sometimes the name "display" is preferred to the word "monitor", as the latter can be ambiguous alongside the other senses of "monitor" meaning "machine-level debugger" or "thread synchronization mechanism". Computer displays are sometimes called heads, especially when talking about how many are connected to a computer. Once an essential component of a [computer terminal](#), computer displays have long since become standardized peripherals in their own right.

#### CRT

A modern CRT display has considerable flexibility: it can often handle all resolutions from 640 by 480 pixels (640×480) up to 2048 by 1536 pixels (2048×1536) with 32-bit colour and a variety of refresh rates.

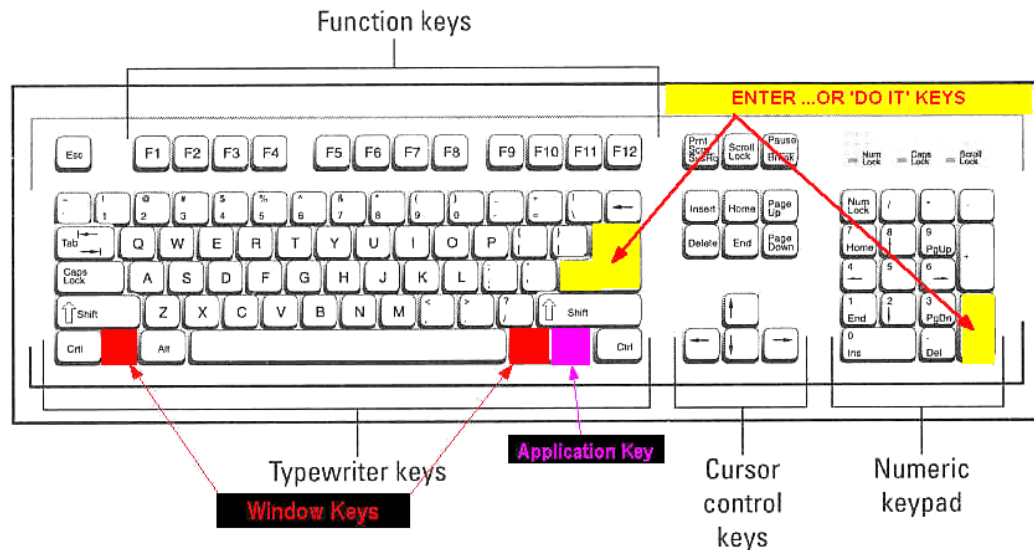


## LCD



A **liquid crystal display (LCD)** is a thin, flat [display device](#) made up of any number of color or monochrome [pixels](#) arrayed in front of a [light](#) source or reflector. It is prized by engineers because it uses very small amounts of electric power, and is therefore suitable for use in battery-powered [electronic](#) devices.

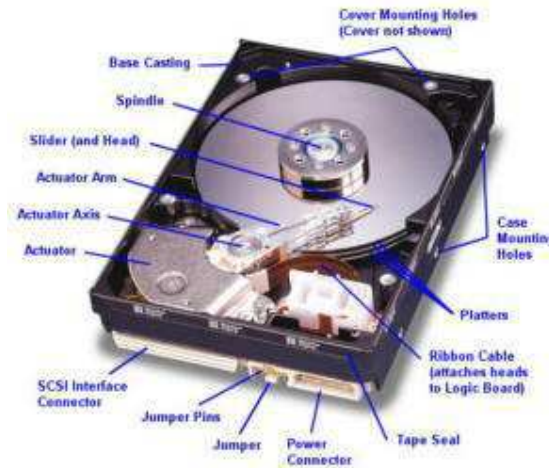
## Keyboard



Keyboards are designed for the input of text and characters, and also to control the operation of the computer. Physically, computer keyboards are an arrangement of rectangular or near-rectangular buttons, or "keys". Keyboards typically have characters engraved or printed on the keys; in most cases, each press of a key corresponds to a single written symbol. However, to produce some symbols requires pressing and holding several keys simultaneously, or in sequence; other keys do not <http://upload.wikimedia.org/wikipedia/en/1/19/QWERTY.png> produce any symbol, but instead affect the operation of the computer, or the keyboard itself. Mouse

## Hard Drive

A magnetic [disk](#) on which you can [store computer data](#). A hard disk uses rigid rotating [platters](#) (disks). It stores and retrieves digital data from a planar magnetic surface. Information is written to the disk by transmitting an electromagnetic flux through an antenna or *write head* that is very close to a magnetic material, which in turn changes its polarization due to the flux. The information can be read back in a reverse manner, as the magnetic fields cause electrical change in the coil or *read head* that passes over it.



## CD-Rom/DVD Drive

The **CD-ROM** (an abbreviation for "[Compact Disc Read-Only Memory](#)") is a non-volatile optical data [storage](#) medium using the same physical format as audio [compact discs](#), readable by a computer with a CD-ROM drive. A CD-ROM is a flat, metallized plastic disc with digital information encoded on it in a spiral from the center to the outside edge



## Floppy Drive

**3.5" Disk** - Created by IBM in [1987](#), which even today are still commonly used. Most 3 1/2" newer disks have a capacity of up to 1.44MB.

- **3.5" Double Density** - 720KB
- **3.5" High Density** - 1.44MB
- **3.5" Extended Density (IBM ONLY)** - 2.88MB

Below is a graphic of a floppy diskette. As can be seen by the below picture this particular diskette is clear which enables you to see the inside of the diskette, you can notice that the floppy has a circular cloth that is located on both sides of the floppy. This cloth helps clean and protect the magnetic disk within the diskette.



Floppy diskette  
<http://www.computerhope.com>

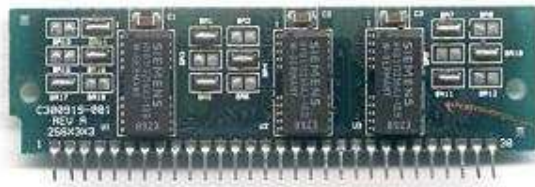
Additional information and help about floppies can be found on our [floppy page](#).

## RAM

**Random access memory** (sometimes **random-access memory**), commonly known by its acronym **RAM**, is a type of [computer storage](#) whose contents can be accessed in any (i.e., random) order. This is in contrast to *sequential memory devices* such as [magnetic tapes](#), discs and drums, in which the mechanical movement of the storage medium forces the computer to access data in a fixed order.

It is usually implied that RAM can be both written to and read from, in contrast to [read-only memory \(ROM\)](#). RAM is typically used for [primary storage](#) (main memory) in computers to hold actively used and actively changing information, although some devices use certain types of RAM to provide long-term [secondary storage](#)





## Modem

A **modem** (a [portmanteau word](#) constructed from **mod**ulator and **dem**odulator) is a [device](#) that [modulates](#) an [analog carrier signal](#) ([sound](#)) to encode [digital](#) information, and also [demodulates](#) such a carrier signal to decode the transmitted information. The goal is to produce a signal that can be transmitted easily and decoded to reproduce the original [digital data](#).



## Video Card

A **graphics card**, **video card**, **v card**, **video board**, **video display board**, **display adapter**, **video adapter**, or **graphics adapter** [1] is a component of a [computer](#) which is designed to convert a logical representation of an [image](#) stored in [memory](#) to a [signal](#) that can be used as input for a display medium, most often a [monitor](#) utilizing a variety of [display standards](#). Typically, it also provides functionality to manipulate the logical image in memory. The graphics card may be a stand-alone [expansion card](#), hence the name, but is often also built into the computer.



## Sound Card

A **sound card** is a [computer expansion card](#) that can input and output sound under program control.

A typical sound card includes a [sound chip](#) usually featuring a [digital-to-analog converter](#) that converts recorded or generated [digital](#) waveforms of sound into an [analog](#) format. This signal is led to a ([earphone](#)-type) connector where a cable to an amplifier or similar sound destination can be plugged in.



## Motherboard

A **motherboard**, also known as a **main board**, **mainboard**, **logic board** or **system board**, and sometimes abbreviated as **mobo**, is the central or primary [circuit board](#) making up a complex electronic system, such as a [computer](#).

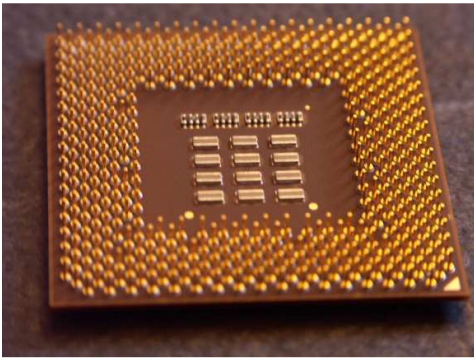
A typical computer is built with the [microprocessor](#), main [memory](#), and other basic components on the motherboard. Other components of the computer such as external storage, control circuits for video display and sound, and [peripheral](#) devices are typically attached to the motherboard via connectors or cables of some sort.



### CPU

central processing unit (CPU) refers to part of a computer that interprets and carries out, or processes, instructions contained in the software. The term processor can refer to a CPU as well; see processor (disambiguation) for other uses of this term. A microprocessor is a common type of CPUs that are manufactured on a single integrated circuit. Most, but not all, modern CPUs are microprocessors.

Historically, a single-processor CPU was a set of refrigerator-sized racks of electronics very similar to today's racks of processors in a server farm. One notable problem which continues to the current day is cooling the electronics which had to run at high speeds, which requires the dissipation of wasted energy, or heat.



## 1950s: early designs

Each of the computer designs of the early 1950s was a unique design; there were no upward-compatible machines or computer architectures with multiple, differing implementations. Programs written for one machine would not run on another kind, even other kinds from the same company. This was not a major drawback at the time because there was not a large body of software developed to run on computers, so starting programming from scratch was not seen as a large barrier.

The design freedom of the time was very important, for designers were very constrained by the cost of electronics, yet just beginning to explore how a computer could best be organized. Some of the basic features introduced during this period included [index registers](#) (on the [Ferranti Mark I](#)), a return-address saving instruction ([UNIVAC I](#)), immediate operands ([IBM 704](#)), and the detection of invalid operations ([IBM 650](#)).

By the end of the [1950s](#) commercial builders had developed factory-constructed, truck-deliverable computers. The most widely installed computer was the [IBM 650](#), which used [drum memory](#) onto which programs were loaded using either [paper tape](#) or [punch cards](#). Some very high-end machines also included [core memory](#) which provided higher speeds. [Hard disks](#) were also starting to become popular.

Computers are automatic [abaci](#). The type of number system affects the way they work. In the early [1950s](#) most computers were built for specific numerical processing tasks, and many machines used decimal numbers as their basic number system – that is, the mathematical functions of the machines worked in base-10 instead of base-2 as is common today. These were not merely [binary coded decimal](#). The machines actually had ten vacuum tubes per digit in each [register](#). Some early [Soviet](#) computer designers implemented systems based on ternary logic; that is, a bit could have three states: +1, 0, or -1, corresponding to positive, no, or negative voltage.

An early project for the [U.S. Air Force](#), [BINAC](#) attempted to make a lightweight, simple computer by using binary arithmetic. It deeply impressed the industry.

As late as 1970, major computer languages such as "[C](#)" were unable to standardize their numeric behavior because decimal computers had groups of users too large to alienate.

Even when designers used a binary system, they still had many odd ideas. Some used sign-magnitude arithmetic (-1 = 10001), rather than modern [two's complement](#) arithmetic (-1 = 11111). Most computers used six-bit character sets, because they adequately encoded [Hollerith](#) cards. It was a major revelation to designers of this period to realize that the data word should be a multiple of the character size. They began to design computers with 12, 24 and 36 bit data words.

In this era, [Grosch's law](#) dominated computer design: Computer cost increased as the square of its speed.

## 1960s: the computer revolution and CISC

One major problem with early computers was that a program for one would not work on others. Computer companies found that their customers had little reason to remain loyal to a particular brand, as the next computer they purchased would be incompatible anyway. At that point price and performance were usually the only concerns.

In 1962, IBM tried a new approach to designing computers. The plan was to make an entire family of computers that could all run the same software, but with different performances, and at different prices. As users' requirements grew they could move up to larger computers, and still keep all of their investment in programs, data and storage media.

In order to do this they designed a single *reference computer* called the [System 360](#) (or [S/360](#)). The System 360 was a virtual computer, a reference instruction set and capabilities that all

machines in the family would support. In order to provide different classes of machines, each computer in the family would use more or less hardware emulation, and more or less [microprogram](#) emulation, to create a machine capable of running the entire System 360 [instruction set](#).

For instance a low-end machine could include a very simple processor for low cost. However this would require the use of a larger microcode emulator to provide the rest of the instruction set, which would slow it down. A high-end machine would use a much more complex processor that could directly process more of the System 360 design, thus running a much simpler and faster emulator.

IBM chose to make the reference [instruction set](#) quite complex, and very capable. This was a conscious choice. Even though the computer was complex, its "[control store](#)" containing the [microprogram](#) would stay relatively small, and could be made with very fast memory. Another important effect was that a single instruction could describe quite a complex sequence of operations. Thus the computers would generally have to fetch fewer instructions from the main memory, which could be made slower, smaller and less expensive for a given combination of speed and price.

As the S/360 was to be a successor to both scientific machines like the [7090](#) and data processing machines like the [1401](#), it needed a design that could reasonably support all forms of processing. Hence the instruction set was designed to manipulate not just simple binary numbers, but text, scientific floating-point (similar to the numbers used in a calculator), and the [binary coded decimal](#) arithmetic needed by accounting systems.

Almost all following computers included these innovations in some form. This basic set of features is now called a "[complex instruction set computer](#)," or CISC (pronounced "sisk"), a term not invented until many years later.

In many CISCs, an instruction could access either registers or memory, usually in several different ways. This made the CISCs easier to program, because a programmer could remember just thirty to a hundred instructions, and a set of three to ten [addressing modes](#) rather than thousands of distinct instructions. This was called an "orthogonal instruction set." The [PDP-11](#) and [Motorola 68000](#) architecture are examples of nearly orthogonal instruction sets.

There was also the *BUNCH* (Burroughs, Univac, NCR, CDC, and Honeywell) that competed against IBM at this time though IBM dominated the era with [S/360](#).

The Burroughs Corporation (which later became Unisys when they merged with Sperry/Univac) offered an alternative to S/360 with their [B5000](#) series machines. The B5000 series [1961](#) had virtual memory, a multi-programming operating system (Master Control Program or MCP), written in [ALGOL 60](#), and the industry's first recursive-descent compilers as early as 1963.

[[edit](#)]

## 1970s: large scale integration

In the 1960s, the [Apollo guidance computer](#) and [Minuteman missile](#) made the [integrated circuit](#) economical and practical.

Around 1971, the first calculator and clock chips began to show that very small computers might be possible. The first [microprocessor](#) was the 4004, designed in 1971 for a calculator company, and produced by [Intel](#). The 4004 is the direct ancestor of the [Intel 80386](#), even now maintaining some code compatibility. Just a few years later, the word size of the 4004 was doubled to form the 8008.

By the mid-1970s, the use of integrated circuits in computers was commonplace. The whole decade consists of upheavals caused by the shrinking price of transistors.

It became possible to put an entire CPU on a single printed circuit board. The result was that minicomputers, usually with 16-bit words, and 4k to 64K of memory, came to be commonplace.

CISCs were believed to be the most powerful types of computers, because their microcode was small and could be stored in very high-speed memory. The CISC architecture also addressed the "semantic gap" as it was perceived at the time. This was a defined distance between the machine language, and the higher level language people used to program a machine. It was felt that compilers could do a better job with a richer instruction set.

Custom CISCs were commonly constructed using "bit slice" computer logic such as the AMD 2900 chips, with custom microcode. A bit slice component is a piece of an [ALU](#), register file or microsequencer. Most bit-slice integrated circuits were 4-bits wide.

By the early 1970s, the [PDP-11](#) was developed, arguably the most advanced small computer of its day. Almost immediately, wider-word CISCs were introduced, the 32-bit [VAX](#) and 36-bit [PDP-10](#).

Also, to control a cruise missile, Intel developed a more-capable version of its 8008 microprocessor, the 8080.

IBM continued to make large, fast computers. However the definition of large and fast now meant more than a megabyte of RAM, clock speeds near one megahertz [\[1\]\[2\]](#), and tens of megabytes of disk drives.

IBM's System 370 was a version of the 360 tweaked to run virtual computing environments. The [virtual computer](#) was developed in order to reduce the possibility of an unrecoverable software failure.

The Burroughs B5000/B6000/B7000 series reached its largest market share. It was a stack computer programmed in a dialect of Algol. It used 64-bit fixed-point arithmetic, rather than floating-point.

All these different developments competed madly for marketshare.

[\[edit\]](#)

## Early 1980s: the lessons of RISC

In the early [1980s](#), researchers at [UC Berkeley](#) and [IBM](#) both discovered that most computer language compilers and interpreters used only a small subset of the instructions of a [CISC](#). Much of the power of the CPU was simply being ignored in real-world use. They realized that by making the computer simpler and less orthogonal, they could make it faster and less expensive at the same time.

At the same time, CPUs were growing faster in relation to the memory they addressed. Designers also experimented with using large sets of internal registers. The idea was to [cache](#) intermediate results in the registers under the control of the compiler. This also reduced the number of [addressing modes](#) and orthogonality.

The computer designs based on this theory were called [Reduced Instruction Set Computers](#), or RISC. RISCs generally had larger numbers of registers, accessed by simpler instructions, with a few instructions specifically to load and store data to memory. The result was a very simple core CPU running at very high speed, supporting the exact sorts of operations the compilers were using anyway.

A common variation on the RISC design employs the [Harvard architecture](#), as opposed to the [Von Neumann](#) or Stored Program architecture common to most other designs. In a Harvard Architecture machine, the program and data occupy separate memory devices and can be accessed simultaneously. In Von Neumann machines the data and programs are mixed in a single memory device, requiring sequential accessing which produces the so-called "Von Neumann bottleneck."

One downside to the RISC design has been that the programs that run on them tend to be larger. This is because [compilers](#) have to generate longer sequences of the simpler instructions to accomplish the same results. Since these instructions need to be loaded from memory anyway, the larger code size offsets some of the RISC design's fast memory handling.



Recently, engineers have found ways to compress the reduced instruction sets so they fit in even smaller memory systems than CISCs. Examples of such compression schemes include [the ARM's "Thumb" instruction set](#). In applications that do not need to run older binary software, compressed RISCs are coming to dominate sales.

Another approach to RISCs was the "[niladic](#)" or "zero-address" instruction set. This approach realized that the majority of space in an instruction was to identify the operands of the instruction. These machines placed the operands on a push-down (last-in, first out) [stack](#). The instruction set was supplemented with a few instructions to fetch and store memory. Most used simple caching to provide extremely fast RISC machines, with very compact code. Another benefit was that the interrupt latencies were extremely small, smaller than most CISC machines (a rare trait in RISC machines). The first zero-address computer was developed by [Charles Moore](#). It placed six 5-bit instructions in a 32-bit word, and was a precursor to [VLIW](#) design (see below: 1990 to Today).

Commercial variants were mostly characterized as "[FORTH](#)" machines, and probably failed because that language became unpopular. Also, the machines were developed by defense contractors at exactly the time that the cold war ended. Loss of funding may have broken up the development teams before the companies could perform adequate commercial marketing.

RISC chips now dominate the market for 32-bit embedded systems. Smaller RISC chips are even becoming common in the cost-sensitive 8-bit embedded-system market. The main market for RISC CPUs has been systems that require low power or small size.

Even some CISC processors (based on architectures that were created before RISC became dominant) translate instructions internally into a RISC-like instruction set. These CISC chips include newer [x86](#) and [VAX](#) models.

These numbers may surprise many, because the "market" is perceived to be desktop computers. With Intel x86 designs dominating the vast majority of all desktop sales, RISC is found only in the [Apple](#) desktop computer lines. However, desktop computers are only a tiny fraction of the computers now sold. Most people own more computers in embedded systems in their car and house than on their desks.

[\[edit\]](#)

## Mid-1980s to today: exploiting instruction level parallelism

In the mid-to-late 1980s, designers began using a technique known as [instruction pipelining](#), in which the processor works on multiple instructions in different stages of completion. For example, the processor may be retrieving the operands for the next instruction while calculating the result of the current one. Modern CPUs may use over a dozen such stages.

A similar idea, introduced only a few years later, was to execute multiple instructions in parallel on separate arithmetic-logic units ([ALUs](#)). Instead of operating on only one instruction at a time, the CPU will look for several similar instructions that are not dependent on each other, and execute them in parallel. This approach is known as [superscalar](#) processor design.

Such techniques are limited by the degree of [instruction level parallelism](#) (ILP), the number of non-dependent instructions in the program code. Some programs are able to run very well on superscalar processors due to their inherent high ILP, notably graphics. However more general problems do not have such high ILP, thus making the achievable speedups due to these techniques to be lower.

Branching is one major culprit. For example, the program might add two numbers and branch to a different code segment if the number is bigger than a third number. In this case even if the branch operation is sent to the second ALU for processing, it still must wait for the results from the addition. It thus runs no faster than if there were only one ALU. The most common solution for this type of problem is to use a type of [branch prediction](#).

To further the efficiency of multiple functional units which are available in superscalar designs, operand register dependencies was found to be another limiting factor. To minimize these dependencies, [out-of-order execution](#) of instructions was introduced. In such a scheme, the

instruction results which complete out-of-order must be re-ordered in program order by the processor for the program to be restartable after an exception. *Out-of-Order* execution was the main advancement of the computer industry during the [1990s](#). A similar concept is [speculative execution](#), where instructions from both sides of a branch are executed at the same time, and the results of one side or the other are thrown out once the branch answer is known.

These advances, which were originally developed from research for RISC-style designs, allow modern CISC processors to execute twelve or more instructions per clock cycle, when traditional CISC designs could take twelve or more cycles to execute just one instruction.

The resulting instruction scheduling logic of these processors is large, complex and difficult to verify. Furthermore, the higher complexity requires more transistors, increasing power consumption and heat. In this respect RISC is superior because the instructions are simpler, have less interdependence and make superscalar implementations easier. However, as Intel has demonstrated, the concepts can be applied to a CISC design, given enough time and money.

Historical note: Some of these techniques (e.g. pipelining) were originally developed in the late [1950s](#) by [IBM](#) on their [Stretch](#) mainframe computer.

[\[edit\]](#)

## 1990 to today: looking forward

[\[edit\]](#)

## VLIW and EPIC

The instruction scheduling logic that makes a superscalar processor is just boolean logic. In the early 1990s, a significant innovation was to realize that the coordination of a multiple-ALU computer could be moved into the [compiler](#), the software that translates a programmer's instructions into machine-level instructions.

This type of computer is called a [very long instruction word](#) (VLIW) computer.

Statically scheduling the instructions in the compiler (as opposed to letting the processor do the scheduling dynamically) has many practical advantages over doing so in the CPU.

Oddly, speed is not one of them. With enough transistors, the CPU could do everything at once. However all those transistors make the chip larger, and therefore more expensive. The transistors also use power, which means that they generate heat that must be removed. The heat also makes the design less reliable.

Since compiling happens only once on the developer's machine, the control logic is "canned" in the final realization of the program. This means that it consumes no transistors, and no power, and therefore is free, and generates no heat.

The resulting CPU is simpler, and runs at least as fast as if the scheduling were in the CPU.

There were several unsuccessful attempts to commercialize VLIW. The basic problem is that a VLIW computer does not scale to different price and performance points, as a dynamically scheduled computer can.

Also, VLIW computers optimise for throughput, not low latency, so they were not attractive to the engineers designing controllers and other computers embedded in machinery. The [embedded systems](#) markets had often pioneered other computer improvements by providing a large market that did not care about compatibility with older software.

In January [2000](#), a company called [Transmeta](#) took the interesting step of placing a compiler in the central processing unit, and making the compiler translate from a reference byte code (in their case, [x86](#) instructions) to an internal VLIW instruction set. This approach combines the hardware simplicity, low power and speed of VLIW RISC with the compact main memory system and software reverse-compatibility provided by popular CISC.

[Intel](#) released a chip, called the [Itanium](#), based on what they call an [Explicitly Parallel Instruction Computing](#) (EPIC) design. This design supposedly provides the VLIW advantage of increased instruction throughput. However, it avoids some of the issues of scaling and complexity, by explicitly providing in each "bundle" of instructions information concerning their dependencies. This information is calculated by the compiler, as it would be in a VLIW design. The early versions are also backward-compatible with current [x86](#) software by means of an on-chip [emulation](#) mode. Integer performance has been disappointing as have sales in volume markets.

[\[edit\]](#)

## Multi-threading

Also, we may soon see multi-threaded CPUs. Current designs work best when the computer is running only a single program, however nearly all modern [operating systems](#) allow the user to run multiple programs at the same time. For the CPU to change over and do work on another program requires expensive [context switching](#). In contrast, a multi-threaded CPU could handle instructions from multiple programs at once.

To do this, such CPUs include several sets of registers. When a context switch occurs, the contents of the "working registers" are simply copied into one of a set of registers for this purpose.

Such designs often include thousands of registers instead of hundreds as in a typical design. On the downside, registers tend to be somewhat expensive in chip space needed to implement them. This chip space might otherwise be used for some other purpose.

[\[edit\]](#)

## Reconfigurable logic

Another track of development is to combine reconfigurable logic with a general-purpose CPU. In this scheme, a special computer language compiles fast-running subroutines into a bit-mask to configure the logic. Slower, or less-critical parts of the program can be run by sharing their time on the CPU. This process has the capability to create devices such as software [radios](#), by using digital signal processing to perform functions usually performed by analog [electronics](#).

[\[edit\]](#)

## Public domain processors

As the lines between hardware and software increasingly blur due to progress in design methodology and availability of chips such as [FPGAs](#) and cheaper production processes, even [open source hardware](#) has begun to appear. Loosely-knit communities like [OpenCores](#) have recently announced completely open CPU architectures such as the [OpenRISC](#) which can be readily implemented on FPGAs or in custom produced chips, by anyone, without paying license fees.

[\[edit\]](#)

## High end processor economics

Developing new, high-end CPUs is a **very** expensive proposition. Both the logical complexity (needing very large logic design and logic verification teams and simulation farms with perhaps thousands of computers) and the high operating frequencies (needing large circuit design teams and access to the state-of-the-art fabrication process) account for the high cost of design for this type of chip. The design cost of a high-end CPU will be on the order of US \$100 million. Since the design of such high-end chips nominally take about five years to complete, to stay competitive a company has to fund at least two of these large design teams to release products at the rate of 2.5 years per product generation. Only the personal computer mass market (with production rates in the hundreds of millions, producing billions of dollars in revenue) can support such economics. As of 2004, only four companies are actively designing and fabricating state of the art general

purpose computing CPU chips: [Intel](#), [AMD](#), [IBM](#) and [Fujiitsu](#). AMD is moving its CPU manufacturing over to IBM soon so that will only leave IBM and Intel as CPU-only foundries. [Motorola](#) has spun off its semiconductor division as [Freescale](#) as that division was dragging down profit margins for the rest of the company. [Texas Instruments](#), [TSMC](#) and [Toshiba](#) are a few examples of a companies doing manufacturing for another company's CPU chip design.

[\[edit\]](#)

## Embedded design

The majority of computer systems in use today are embedded in other machinery, such as telephones, clocks, appliances, vehicles, and infrastructure. An [embedded system](#) usually has minimal requirements for memory and program length and may require simple but unusual input/output systems. For example, most embedded systems lack keyboards, screens, disks, printers, or other recognizable I/O devices of a personal computer. They may control electric motors, relays or voltages, and reed switches, variable resistors or other electronic devices. Often, the only I/O device readable by a human is a single light-emitting diode, and severe cost or power constraints can even eliminate that.

In contrast to general-purpose computers, embedded systems often seek to minimize [interrupt latency](#) over instruction throughput.

When an electronic device causes an interrupt, the intermediate results, the registers, have to be saved before the software responsible for handling the interrupt can run, and then must be put back after it is finished. If there are more registers, this saving and restoring process takes more time, increasing the latency.

Low-latency CPUs generally have relatively few registers in their central processing units, or they have "shadow registers" that are only used by the interrupt software.

[\[edit\]](#)

## Other design issues

One interesting near-term possibility would be to eliminate the bus. Modern vertical [laser diodes](#) enable this change. In theory, an optical computer's components could directly connect through a holographic or phased open-air switching system. This would provide a large increase in effective speed and design flexibility, and a large reduction in cost. Since a computer's connectors are also its most likely failure point, a busless system might be more reliable, as well.

Another farther-term possibility is to use light instead of electricity for the digital logic itself. In theory, this could run about 30% faster and use less power, as well as permit a direct interface with quantum computational devices. The chief problem with this approach is that for the foreseeable future, electronic devices are faster, smaller (i.e. cheaper) and more reliable. An important theoretical problem is that electronic computational elements are already smaller than some wavelengths of light, and therefore even wave-guide based optical logic may be uneconomic compared to electronic logic. We can therefore expect the majority of development to focus on electronics, no matter how unfair it might seem. See also [optical computing](#).

Yet another possibility is the "clockless CPU" (asynchronous CPU). Unlike conventional processors, clockless processors have no central clock to coordinate the progress of data through the pipeline. Instead, stages of the CPU are coordinated using logic devices called "pipe line controls" or "FIFO sequencers." Basically, the pipeline controller clocks the next stage of logic when the existing stage is complete. In this way, a central clock is unnecessary. There are two advantages to clockless CPUs over clocked CPUs:

components can run at different speeds in the clockless CPU. In a clocked CPU, no component can run faster than the clock rate.

In a clocked CPU, the clock can go no faster than the worst-case performance of the slowest stage. In a clockless CPU, when a stage finishes quicker than normal, the next stage can immediately take the results rather than waiting for the next clock tick. A stage might finish quicker than normal because of the particular data inputs (multiplication can be very fast if it is multiplying by 0 or 1), or because it is running at a higher voltage or lower temperature than normal.

Two examples of asynchronous CPUs are the [ARM](#)-implementing [AMULET](#) and the asynchronous implementation of [MIPS](#) R3000, dubbed [MiniMIPS](#).

The biggest disadvantage of the clockless CPU is that most CPU design tools assume a clocked CPU, so making a clockless CPU involves modifying the design tools to handle clockless logic and doing extra testing to ensure the design avoids [metastable](#) problems. For example, the group that designs the aforementioned AMULET developed a tool called [LARD](#) to cope with the complex design of AMULET3.

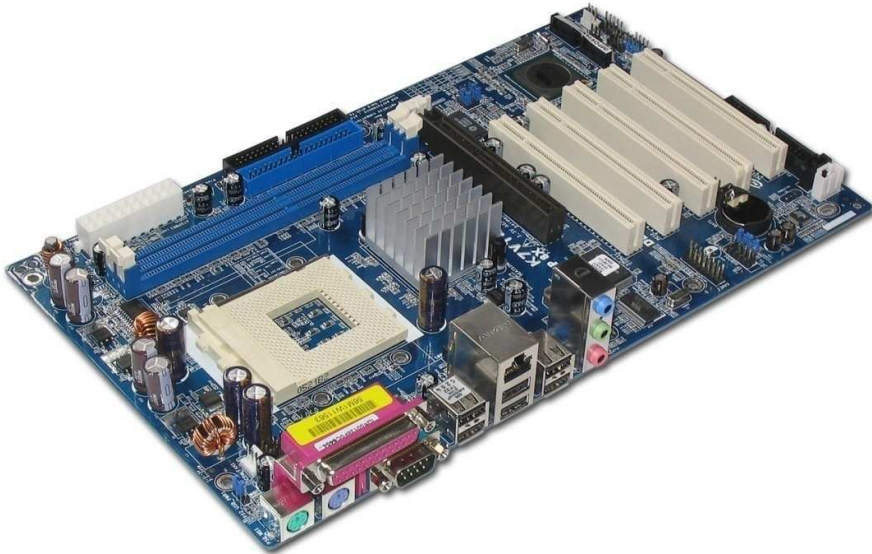
## Mother Board

A [motherboard](#) is a [printed circuit board](#) used in a [personal computer](#). It is also known as the [mainboard](#) and occasionally abbreviated to mobo or MB. The term **mainboard** is also used for the main circuit board in this and other electronic devices.

A typical motherboard provides attachment points for one or more of the following: [CPU](#), [graphics card](#), [sound card](#), [hard disk](#) controller, [memory \(RAM\)](#), and external peripheral devices.

All of the basic circuitry and components required for a computer to function sit either directly on the motherboard or in an expansion slot of the motherboard. The most important component on a motherboard is the chipset which consists of two components or chips known as the [Northbridge](#) and [Southbridge](#). These [chips](#) determine, to an extent, the features and capabilities of the motherboard.

The remainder of this article discusses the state of the so-called "[IBM compatible PC](#)" motherboard in the early [2000s](#). It contains the chipset, which controls the operation of the CPU, the [PCI](#), [ISA](#), [AGP](#), and [PCI Express](#) expansion slots, and (usually) the IDE/ATA controller as well. Most of the devices that can be attached to a motherboard are attached via one or more slots or sockets, although some modern motherboards support wireless devices using the [IrDA](#), [Bluetooth](#), or 802.11 ([Wi-Fi](#)) protocols



## CPU sockets

*Main article:* [CPU socket](#)

There are different slots and sockets for CPUs, and it is necessary for a motherboard to have the appropriate slot or socket for the CPU. Newer sockets, those with a three digit number, are named after the number of pins they contain. Older ones are simply named in the order of their invention, usually with a single digit.

A sample of sockets and associated processors:

### Sockets supporting Intel CPUs

- Socket 1 - [80486SX](#), [80486DX](#), [80486DX2](#), [80486DX4](#), and compatible processors from other manufacturers
- Socket 2 - [80486SX](#), [80486DX](#), [80486DX2](#), [80486DX4](#), and clones
- Socket 3 - [80486SX](#), [80486DX](#), [80486DX2](#), [80486DX4](#), and clones
- Socket 4 - early Intel Pentium processors
- Socket 5 - early Intel Pentium processors
- Socket 6 - [80486DX4](#)
- Socket 7 - [Intel Pentium](#) and Pentium MMX (also some AMD and Cyrix CPUs)
- Socket 8 - [Intel Pentium Pro](#)
- Slot 1 - [Intel Pentium II](#), older [Pentium III](#), and Celeron processors (233 MHz - 1.13 GHz)
- Slot 2 - [Intel](#) Xeon processors based on Pentium II/III cores
- Socket 370 - [Celeron](#) processors and newer [Pentium IIIs](#) (800 MHz - 1.4 GHz)
- Socket 423 - [Intel Pentium 4](#) and Celeron processors (based on the Willamette core)
- Socket 478 - [Intel Pentium 4](#) and Celeron processors (based on Northwood, Prescott, and Willamette cores)
- Socket 479 - [Intel Pentium M](#) and Celeron M processors (based on the Banias and Dothan cores)
- Socket 480 - Intel Pentium M processors (based on the Yonah core)
- Socket 603/604 - Intel [Xeon](#) processors based on the Northwood and Willamette Pentium 4 cores
- Socket T/LGA 775 ([Land Grid Array](#)) - [Intel Pentium 4](#) and Celeron processors (based on Northwood and Prescott cores)

## Sockets supporting AMD CPUs

- Slot A - original AMD Athlon processors
- Socket 462 (aka Socket A) - newer AMD Athlon, Athlon XP, Sempron, and Duron processors
- Socket 754 - lower end AMD Athlon 64 and Sempron processors with single-channel memory support
- Socket 939 - AMD Athlon 64 and AMD Athlon FX processors with dual-channel memory support
- Socket 940 - AMD Opteron and early AMD Athlon FX processors

## Peripheral card slots

There are usually a number of expansion card slots to allow [peripheral](#) devices and cards to be inserted. Each slot is compatible with one or more industry [bus](#) standards. Commonly available buses include: [PCI](#) (Peripheral Component Interconnect), [PCI-X](#), [AGP](#) (Accelerated Graphics Port), and [PCI Express](#).

ISA was the original bus for connecting cards to a PC. Despite significant performance limitations, it was not superseded by the more advanced but incompatible MCA ([Micro Channel Architecture](#)) (IBM's proprietary solution which appeared in their [PS/2](#) series of computers and a handful of other models) or the equally advanced and backward-compatible [EISA](#) (Extended Industry Standard Architecture) bus. It endured as a standard feature in PCs till the end of the 20th century, aided first by the brief dominance of the [VESA](#) (Video Electronic Standards Association) extension during the reign of the [486](#) and later by the need to accommodate the large number of existing ISA peripheral cards. The more recent PCI bus is the current industry standard, which initially was a high-speed supplement to ISA for high-bandwidth peripherals (notably graphics cards, network cards, and [SCSI](#) host adaptors), and gradually replaced ISA as a general-purpose bus. An AGP slot is a high speed, single-purpose port designed solely for connecting high performance graphics cards (which produce [video](#) output) to the monitor. Both AGP and PCI buses are marked for replacement by PCI Express, although this is unlikely to happen prior to 2006 because of the large established base of AGP/PCI motherboards and add-in cards.

A typical motherboard of 1999 might have had one AGP slot, four PCI slots, and one (or two) ISA slots; since about 2002 the last ISA slots in new boards have been replaced with extra PCI slots. Sometimes an [Advanced Communications Riser](#) slot is used instead on less expensive motherboards.

As of 2001, most PCs also support Universal Serial Bus ([USB](#)) connections, and the controller and ports required for this are usually integrated onto the motherboard. An [ethernet](#) interface and a basic audio processor are now almost universally integrated into current motherboards as well.

## BIOS

**BIOS**, in [computing](#), stands for basic input/output system. BIOS refers to the software code run by a computer when first powered on. The primary function of BIOS is to prepare the machine so other software programs stored on various media (such as hard drives, floppies, and CDs) can load, execute, and assume control of the computer.

While the name BIOS is an [acronym](#), it may also be a play on the [Greek](#) word βίος (*bios*) *life*. The term first appeared in the [CP/M](#) operating system, describing the part of CP/M loaded during boot time that interfaced directly with the hardware. (CP/M machines usually had a simple boot loader in ROM, and nothing else.) Most versions of [DOS](#) have a file called "[IBMBIO.COM](#)" or "[IO.SYS](#)" that is analogous to the CP/M disk BIOS.



## How the BIOS Boots

The BIOS runs off the onboard [flash memory](#) when the computer is powered on and it initializes the chipset and the memory subsystem. Subsequently, it uncompresses itself from flash memory into the system main memory and starts executing from there. PC BIOS code also contains diagnostics to assure critical hardware components, such as keyboard, [disk drive](#), I/O ports etc., are operational, and properly initialized. Nearly all BIOS implementations can optionally execute a setup program interfacing the [nonvolatile BIOS memory \(CMOS\)](#). This memory holds user-customizable configuration data (time, date, [hard drive](#) details, etc.) accessed by BIOS code. The [80x86](#) source code for early PC and AT BIOS was included with the IBM Technical Reference Manual.

In most modern BIOS implementations, one can select what boots first: [CD](#), [hard disk](#), [floppy disk](#), [flash keydrive](#) and so on. This is particularly useful for installing [operating systems](#) or booting to [LiveCDs](#), and for selecting the order of testing for the presence of bootable media.

Some BIOSes allow the user to select the operating system to load (e.g. load another OS from the second hard disk), though this is more often handled by a second-stage [boot loader](#).

## BIOS as firmware

BIOS is sometimes called [firmware](#) because it is an integral part of the system hardware. Before 1990 or so BIOSs were held on [ROM](#) chips that could not be altered. As their complexity and the need for updates grew, BIOS firmware was stored on [EEPROM](#) or [flash memory](#) devices that can be easily upgraded by the user. However, an improperly executed or aborted BIOS update can render the computer or device unusable. To avoid BIOS corruption, some new [motherboards](#) have a backup BIOS ("Dual BIOS" boards). Also, most BIOSes have a "boot block" which is a portion of the ROM that runs first and is not updateable. This code will verify that the rest of the BIOS is intact (via checksum, hash, etc.) before jumping to it. If the boot block detects that the main BIOS is corrupt, then it will typically boot to a floppy so that the user can try [flashing](#) again, hopefully with a better image. Hardware manufacturers frequently issue BIOS updates to upgrade their products and remove [bugs](#).

## Firmware on adapter cards

A computer system can contain several BIOS firmware chips. In addition to the boot BIOS, which contains code to access fundamental hardware components such as the keyboard or the [floppy drive](#), plug-in adapter cards such as [SCSI](#) or [USB](#) hard disk adapters or [network cards](#) or video boards may include their own BIOS, complementing or replacing the system BIOS code for the given component.

To find these memory mapped expansion ROMs during boot, PC BIOS implementations scan real memory from 0xC8000 to 0xF0000 on 2 kilobyte boundaries looking for a 0x55 0xaa signature, which is immediately followed by a byte indicating the number of 512 byte blocks the expansion ROM occupies in real memory. The BIOS then jumps to the offset immediately after the size byte, at which point the expansion ROM code takes over and uses BIOS services to provide a user configuration interface, register interrupt vectors for use by post-boot applications, or display diagnostic information.

For UNIX and Windows/DOS systems there is a utility with which you can dump your BIOS firmware software at <http://www.linuks.mine.nu/ree/>

## **The BIOS boot specification**

If the expansion ROM wishes to change the way the system boots (such as from a network device or a SCSI adapter for which the BIOS has no driver code), it can use the BIOS Boot Specification (BBS) API to register its ability to do so. Once the expansion ROMs have registered using the BBS APIs, the user can select among the available boot options from within the BIOS's user interface. This is why most BBS compliant PC BIOS implementations will not allow the user to enter the BIOS's user interface until the expansion ROMs have finished executing and registering themselves with the BBS API.

## Power-On Self Test

**Power-on Self Test** (POST) is the common term for a computer's pre-boot sequence. Though the same basic sequence is present on all computer architectures, it may go by different names such as [Initial Program Load](#) (IPL), [booting](#), or [bootstrapping](#). The term POST, however, has become popular in association with and as a result of the proliferation of the [PC](#). It can be used as a noun when referring to the [code](#) that controls the pre-boot phase or when referring to the phase itself. It can also be used as a verb when referring to the code or the system as it progresses through the pre-boot phase. Alternatively this may be called "POSTing".

## General Internal Workings

On the PC, the main duties of POST are handled by the main [BIOS](#) which may hand some of these duties to other programs designed to initialize very specific peripheral devices, notably for video and [SCSI](#) initialization. These other duty-specific programs are generally known collectively as Option ROMs or individually as the Video BIOS, SCSI BIOS, etc.

- The principal duties of the main BIOS during POST are as follows:
- verify the integrity of the BIOS code itself
- determine the reason POST is being executed
- find, size, and verify system [main memory](#)
- discover, initialize, and catalog all system buses and devices
- pass control to other specialized BIOSes (if and when required)
- provide a user interface for system configuration
- identify, organize, and select which devices are available for booting
- construct whatever system environment that is required by the target OS

The BIOS will begin its POST duties when the CPU is reset. The first memory location the CPU tries to execute is known as the [reset vector](#). In the case of a cold boot, the [northbridge](#) will direct this code fetch (request) to the BIOS located on the system [flash memory](#). For a warm boot, the BIOS will be located in the proper place in [RAM](#) and the northbridge will direct the reset vector call to the RAM.

During the POST flow of a contemporary BIOS, one of the first things a BIOS should do is determine the reason it is executing. For a cold boot, for example, it may need to execute all of its functionality. If, however, the system supports power savings or quick boot methods, the BIOS may be able to circumvent the standard POST device discovery, and simply program the devices from a preloaded system device table.

The POST flow for the PC has developed from a very simple straightforward process to one that is complex and convoluted. During POST, the BIOS must integrate a plethora of competing, evolving, and even mutually exclusive standards and initiatives for the matrix of hardware and OSes the PC is expected to support. However, the average user still only knows the POST and BIOS through its simple visible memory test and setup screen.

[\[edit\]](#)

## Fundamental Structure

In the case of the IBM PC compatible machines, the main BIOS is divided into two basic sections. The POST section, or POST code, is responsible for the tasks as mentioned above and the environment POST constructs for the OS is known as the [Runtime](#) code, The Runtime BIOS, or The Runtime footprint. Primarily these two divisions can be distinguished in that POST code should be flushed from memory before control is passed to the target OS while the Runtime code remains resident in memory. This division may be a misleading oversimplification, however, as many Runtime functions are executed while the system is “POSTing”.

## Standard Original IBM POST Error Codes

- 1 short beep - Normal POST - system is ok
- 2 short beeps - POST Error - error code shown on screen
- No beep - Power supply or system board problem
- Continuous beep - Power supply, system board, or keyboard problem
- Repeating short beeps - Power supply or system board problem
- 1 long, 1 short beep - System board problem
- 1 long, 2 short beeps - Display adapter problem (MDA, CGA)
- 1 long, 3 short beeps - Enhanced Graphics Adapter (EGA)
- 3 long beeps - 3270 keyboard card

## POST AMI BIOS Beep Codes

- 1 - Memory refresh timer error
- 2 - Parity error in base memory (first 64 KB block)
- 3 - Base memory read/write test error
- 4 - Mother board timer not operational
- 5 - Processor error
- 6 - 8042 Gate A20 test error (cannot switch to protected mode)
- 7 - General exception error (processor exception interrupt error)
- 8 - Display memory error (system video adapter)
- 9 - AMI BIOS ROM checksum error
- 10 - CMOS shutdown register read/write error
- 11 - Cache memory test failed

## IBM POST Diagnostic Code Descriptions

- 100 to 199 - System Board
- 200 to 299 - Memory
- 300 to 399 - Keyboard
- 400 to 499 - Monochrome Display
- 500 to 599 - Color/Graphics Display
- 600 to 699 - Floppy-disk drive and/or Adapter
- 700 to 799 - Math Coprocessor
- 900 to 999 - Parallel Printer Port
- 1000 to 1099 - Alternate Printer Adapter

- 1100 to 1299 - Asynchronous Communication Device, Adapter, or Port
- 1300 to 1399 - Game Port
- 1400 to 1499 - Color/Graphics Printer
- 1500 to 1599 - Synchronous Communication Device, Adapter, or Port
- 1700 to 1799 - Hard Drive and/or Adapter
- 1800 to 1899 - Expansion Unit (XT)
- 2000 to 2199 - Bisynchronous Communication Adapter
- 2400 to 2599 - EGA system-board Video (MCA)
- 3000 to 3199 - LAN Adapter
- 4800 to 4999 - Internal Modem
- 7000 to 7099 - Phoenix BIOS Chips
- 7300 to 7399 - 3.5" Disk Drive
- 8900 to 8999 - MIDI Adapter
- 11200 to 11299 - SCSI Adapter
- 21000 to 21099 - SCSI Fixed Disk and Controller
- 21500 to 21599 - SCSI CD-ROM System

### **Motherboard Form factor**

Form factor is the physical size and shape of a device. It is often used to describe the size of circuit boards

The form factor of a motherboard determines the specifications for its general shape and size. It also specifies what type of case and power supply will be supported, the placement of mounting holes, and the physical layout and organization of the board. Form factor is especially important if you build your own computer systems and need to ensure that you purchase the correct case and components.

The motherboard in a computer determines the type of form factor to be chosen. The same form factor has to be chosen for the case and the power supply also.

Using a matching form factor for the power supply and case ensures the following:

- The motherboard fits in the case
- The power supply cords to the motherboard provide correct voltages and the connectors match the connectors on the motherboard
- The holes in the motherboard align with the holes in the case for anchoring the board to the case.
- Holes in the case align with the parts coming off the motherboard

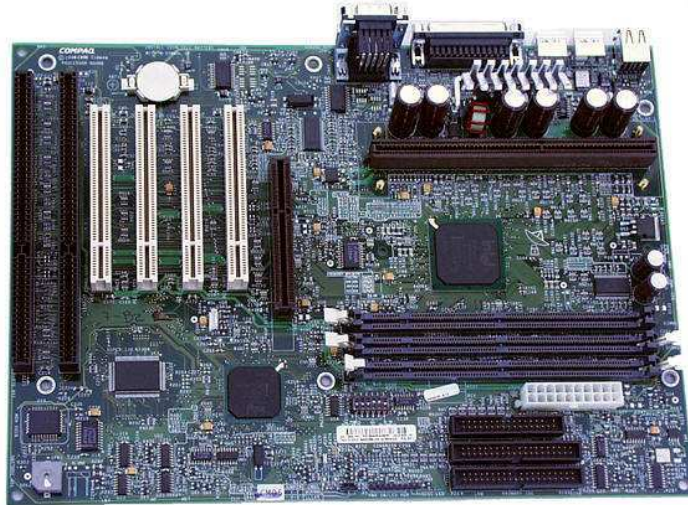
There are several types of form factors, The most popular among them being used today are:

- AT
- Baby AT
- ATX
- Mini ATX

### **AT Form factor**

The AT form factor is found in older computers (386 class or earlier).

The size of the board is 12" X 13.8".



### **Disadvantage of AT Form factor**

- AT motherboards can not be used with smaller AT cases or with ATX cases.
- Dimensions make it difficult to install, upgrade or service.
- The CPU is placed on the motherboard in front of the expansion slots, thus long cards can not be used as they bump into the CPU.

### **Baby AT Form factor**

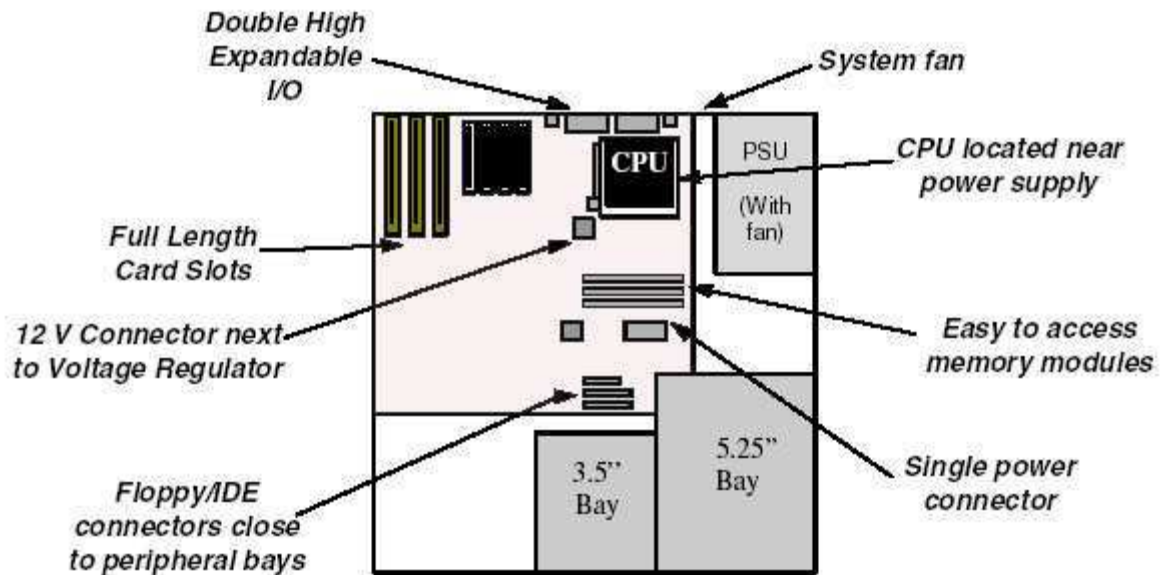
Following the AT form factor, the Baby AT form factor was introduced. With the Baby AT form factor the width of the motherboard was decreased from 12" to 8.5", limiting problems associated with overlapping on the drive bays' turf. Baby AT became popular and was designed for peripheral devices — such as the keyboard, mouse, and video — to be contained on circuit boards that were connected by way of expansion slots on the motherboard.

### **Disadvantage of Baby AT Form factor**

Baby AT was not without problems however. Computer memory itself advanced, and the Baby AT form factor had memory sockets at the front of the motherboard. As processors became larger, the Baby AT form factor did not allow for space to use a combination of processor, heatsink, and fan.

## **ATX Form factor**

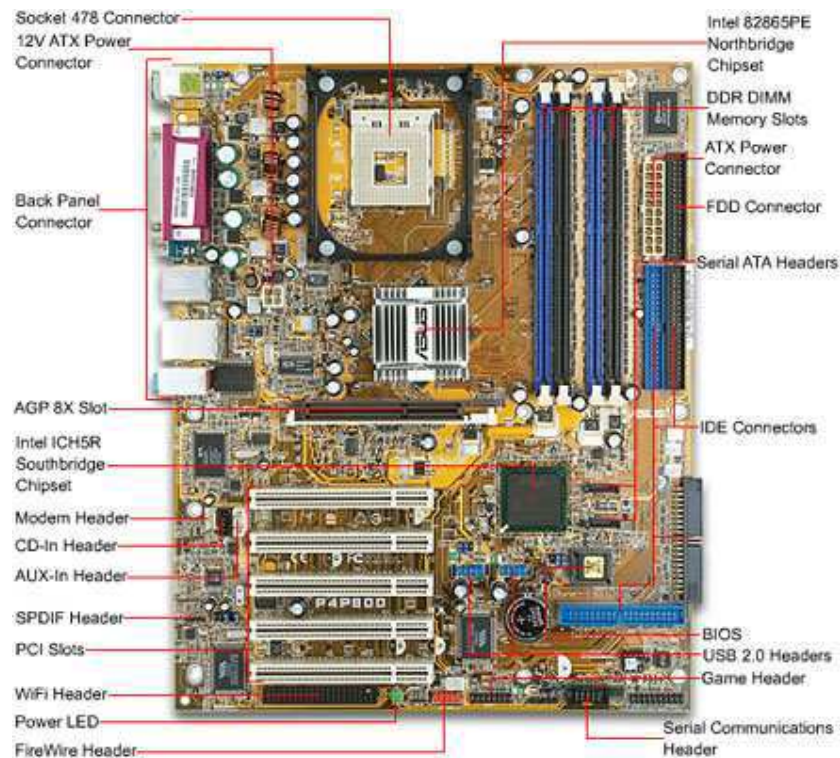
ATX is the most commonly used form factor today. An ATX motherboard measures 12" X 9.6" much smaller than that of AT motherboard. It is much easier to install, upgrade or service an ATX than an AT. On an ATX motherboard the CPU and memory slots are rotated through 90 degrees from the position



on the AT motherboard. This prevents interference with full-length expansion slots.

Additional differences between ATX and AT are that the power supply fan blows air into the case instead of out for better air flow, less overlap between the motherboard and drive bays, and integrated I/O Port connectors soldered directly onto the motherboard. The ATX form factor was an overall better design for upgrading.





### ***ATX Motherboard***

Another important feature about ATX form factor is the Soft Switch which shuts down the power to the system through the operating system. If the user switches off the power switch on the front case of the computer when the computer is on, the operating system will go through the normal shut down procedure.

### **Mini ATX Form factor**

Mini ATX form factor has a motherboard size, which is slightly smaller than the full-sized ATX specification. These boards use the same ATX form factor power supplies and cases. The main difference is that full ATX motherboards have a maximum size of 12"x9.6", and Mini ATX boards have maximum dimensions of 11.2"x8.2".

The different form factors in the ATX family are similar, and the main differences between motherboards of different ATX-style form factors are dimensions, and the placement of mounting holes. This means that most cases are "downward compatible" with the smaller ATX variants, as long as they have provided appropriate sets of places to put mounting hardware.

## Serial Ports and Parallel ports

Figure shows the Serial and Parallel ports in a computer.



### Serial Port:

Serial ports are a type of computer interface that complies with the RS-232 standard. They are 9-pin connectors that relay information, incoming or outgoing, one byte at a time. Each byte is broken up into a *series* of eight bits, hence the term *serial* port.

A typical Serial interface cable used for the serial port is shown in the figure.



Serial ports are one of the oldest types of interface standards. Before internal modems became commonplace, external modems were connected to computers via serial ports, also known as communication or "COM" ports. Computer mice and even keyboards also used serial ports.

The serial communications are used for transferring data over long distances, because parallel communications requires too many wires. Serial data received from a modem or other devices are converted to parallel so that it can be transferred to the PC bus.



### Parallel port:

The Parallel port (shown in figure) is a standard designed to connect a printer to a computer. It is used for the CPU to send data to a printer. This interface drives some input and output signals.

When a PC sends data to a printer or other device using a parallel port, it sends 8 bits of data (1 byte) at a time. These 8 bits are transmitted parallel to each

other, unlike the same eight bits being transmitted serially (all in a single row) through a serial port. Since the data (1 byte) is sent parallel at a time it is called a Parallel

### **BIOS (Basic Input Output Software) SETUP**

The BIOS (Basic Input/Output System) contains the microprocessor code necessary to start your computer and to perform basic functions like reading and writing to the hard disk drive and displaying text on the screen.

Most computers have a setup utility that allows certain parameters in the BIOS to be altered by the user depending on the specific configuration of the computer. These values can then be saved in non-volatile CMOS memory, so future boots of the computer retain the saved values. This is also called the CMOS setup.

### **BIOS Setup procedure**

To enter the BIOS Setup Utility, you must (re)start your computer and press a specific key sequence which is indicated by a prompt on the screen. Normally, the DEL key must be pressed while the memory is being tested. This is done just after the computer first starts up, but before Windows loads. Once the BIOS Setup Utility starts, you should see a screen similar to the following:

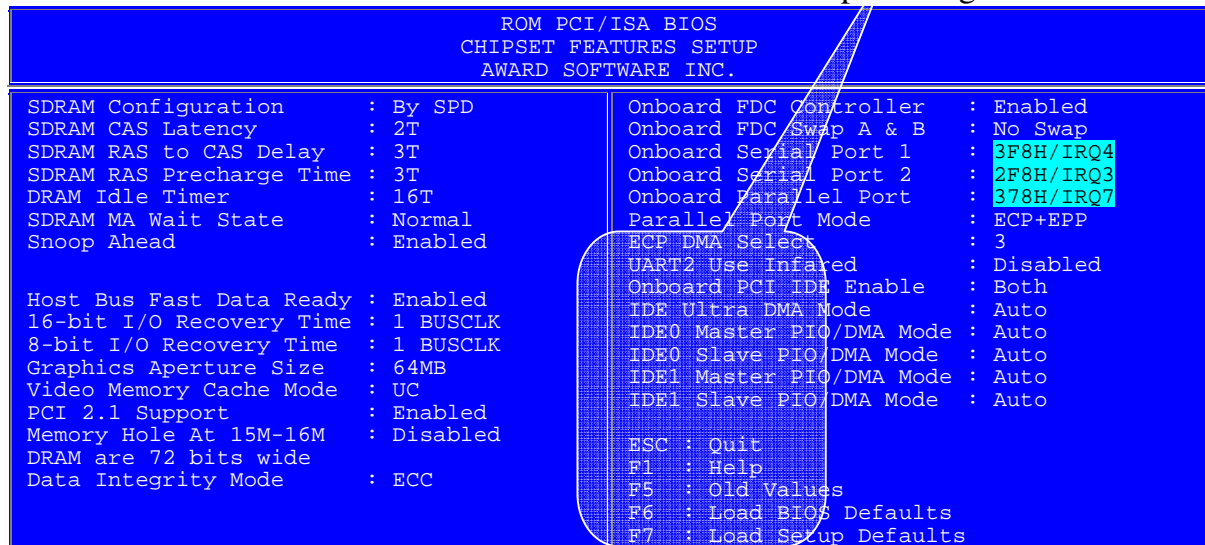
ROM PCI/ISA BIOS CMOS SETUP UTILITY AWARD SOFTWARE INC.	
STANDARD CMOS SETUP	SUPERVISOR PASSWORD
BIOS FEATURES SETUP	USER PASSWORD
<b>CHIPSET FEATURES SETUP</b>	IDE HDD AUTO DETECTION
POWER MANAGEMENT SETUP	SAVE & EXIT SETUP
PNP AND PCI SETUP	EXIT WITHOUT SAVING
LOAD BIOS DEFAULTS	
LOAD SETUP DEFAULTS	
Esc: Quit	
F10: Save & Exit Setup	

When the peripheral devices are connected to the computer the CPU can not respond to the commands or requests from these devices unless some settings are carried out in the BIOS. One such setting is called the IRQ (Interrupt Request).

An IRQ (Interrupt Request) is a physical signal between a device and the host processor (CPU) which informs the host processor that the device needs service.

We start by selecting **CHIPSET FEATURES SETUP** and press Enter.

### Serial and Parallel port assignments



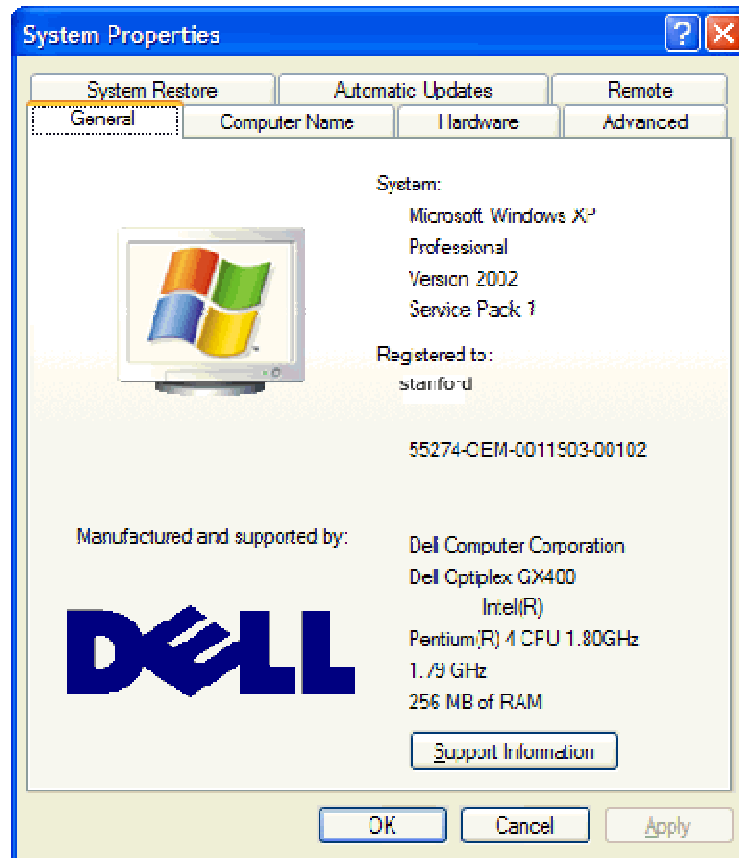
### Default port settings on any computers

Port	IRQ	I/O Address (Hex)	Type
COM1	IRQ 4	03F8 – 03FF	Serial
COM2	IRQ 3	02F8 – 02FF	Serial
COM3	IRQ 4	03E8 – 03EF	Serial
COM4	IRQ 3	02E8 – 02EF	Serial
LPT1	IRQ 7	0378 – 037F	Parallel
LPT2	IRQ 5	0278 – 027F	Parallel

To verify that the ports are configured correctly in a computer, we can use the Device manager in the Windows and view the properties as shown in the following figures.

1. Right Click on the My Computer icon on the desktop and select Properties  
It opens a window as shown in this figure.

2. Click on Hardware.

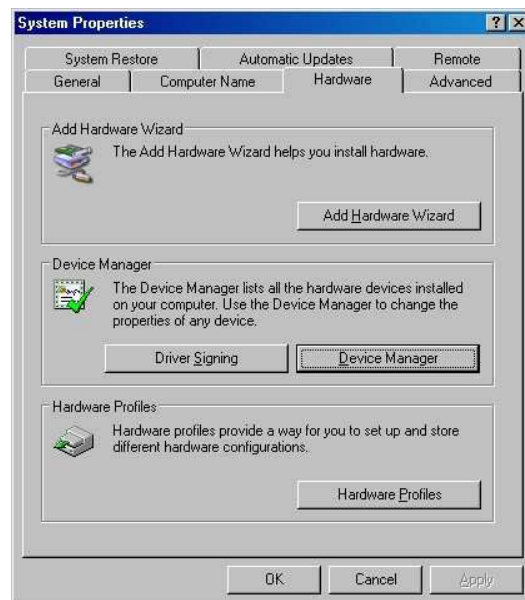


3. Click on the Device Manager.

4. A window opens up showing all the devices connected to the computer.

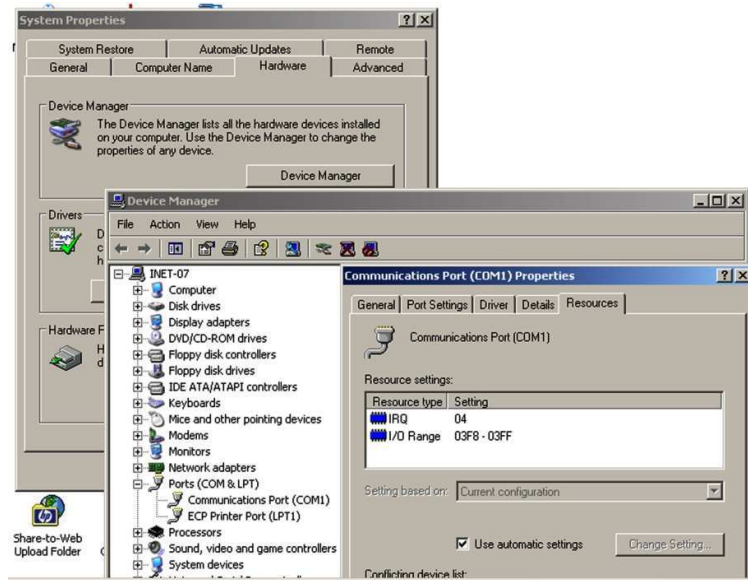
5. Click on the + beside the ports to see the list of ports.

6. Click on the COM port and then click on the properties.



7. The Properties dialog box gives all the details of the COM port.

8. Clicking on the Resources displays the Interrupt (IRQ) settings and the Address settings for this COM port.



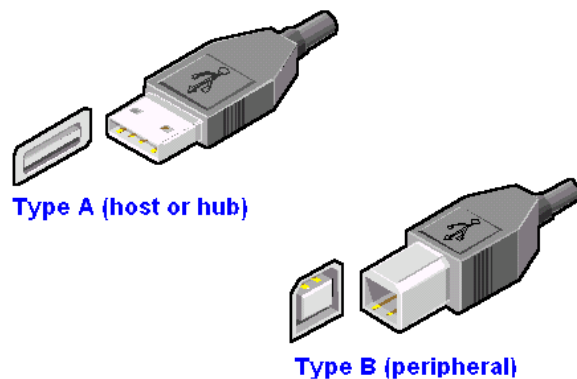
## **USB installation and Configuration**

The Universal Serial Bus, or **USB** for short, is an external bus (a hardwired connection linking two or more hardware components within a computer system) designed to provide a fast and functional means for adding external components to a PC.

The Universal Serial Bus gives a single, standardized, easy-to-use way to connect up to 127 devices to a computer. Just about every peripheral made now comes in a USB version.

The first and commonest USB PC peripherals are the standard human interface devices such as the keyboard, mouse and joystick. Other high speed units like ISDN modems, scanners, printers, external hard disks and CD ROM drives can also be connected. The high transmission rate of 12 Mbit/s permits even MPEG-2 based video products to be connected.

Figure shows a USB port in a PC and a USB connector.



### **Installing a USB Device**

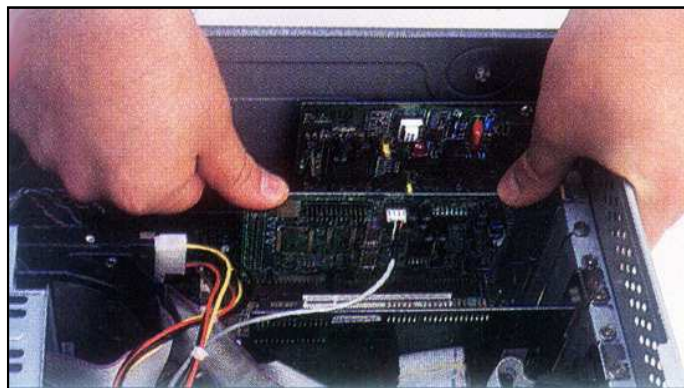
The following are the steps to be followed in installing a USB device.

1. The quickest way to find out if USB is supported by your motherboard is to check the BIOS. Look under the menus for a USB option. If it's there, enable it before switching off your PC to fit the connector.

2. If your motherboard has a USB connector, you'll need a USB port. Just open the casing and find a spare slot in the back as shown in the figure. You may have to sacrifice a PCI slot for this. Secure the plate to the backing plate.



3. Locate the connector on your motherboard. If you're lucky, the connector will only fit one way - if it doesn't you'll need to check your pin numbers carefully to establish the correct settings.
4. It's even easier to connect a PCI expansion card. Just plug it into an available slot. Push it down gently, but firmly, into place, then secure it to the backing plate with a screw. Figure shows insertion of a PCI expansion card.



5. Now restart your PC. When Windows loads, it should automatically detect your USB port and let you know about it. Windows will actually detect two devices - the Host Controller and the USB Hub.
6. Right-click on My Computer and select Properties. Switch to the Device Manager tab. If there's a warning sign on the Host Controller you need to free an IRQ. Click on Properties then Hardware Troubleshooter.

**IEEE 1394**



FireWire (also known as i.Link or IEEE 1394) is a personal computer and digital audio/video serial bus interface standard offering high-speed communications and isochronous (***Isochronous***: transport mechanism to ensure that data is delivered as fast as it is displayed) real-time data services. FireWire can be considered a successor technology to the obsolescent SCSI Parallel Interface. A single 1394 port can be used to connect up to 63 external devices. This is a very fast external bus standard that supports data transfer rates of up to 400Mbps (in 1394a) and 800Mbps (in 1394b).



Although extremely fast and flexible, 1394 is also expensive. Like USB, 1394 supports both Plug-and-Play and hot plugging, and also provides power to peripheral devices

Figure shows IEEE 1394 cable connectors.

IEEE 1394 is widely used for Multimedia applications and is used for connecting the Digital cameras, video cameras and web cameras to the computer.

Figure below illustrates the use of IEEE 1394 support



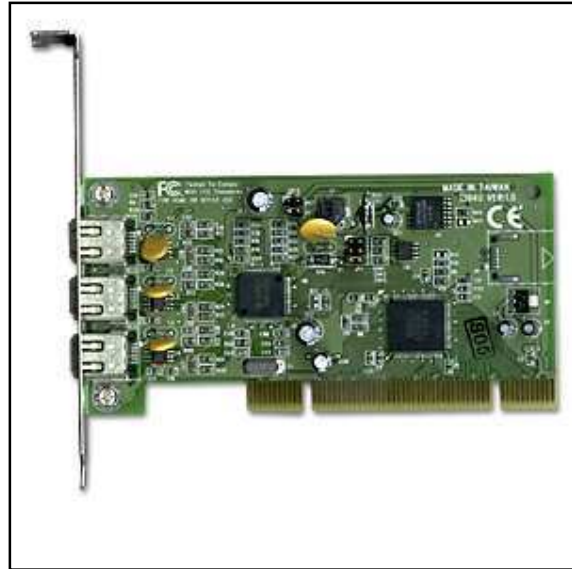
## Installing a IEEE 1394 port

The following are the steps to be followed in installing a USB device.

1. Installation of the IEEE 1394 card is not unlike other PCI cards.

A typical IEEE 1394 card is shown in the figure.

Simply install the Firewire IEEE 1394 card into an available PCI slot in your PC as shown to the right and secure it.



2. Verify that Windows recognizes that an IEEE 1394 controller is present on the motherboard. Using Device manager, look for the 1394 Bus controller listed as an installed device.
3. Click the + sign beside the controller in the device manager to see the specific brand of the 1394 controller the board contains. If the controller is not installed or is not working reinstall the driver.
4. In Control panel, double-click the **Add New Hardware** icon. If there is any problem in installing the driver, verify that 1394 is enabled in the setup.
5. Plug the device into the 1394 port. Install the device drivers for the device connected. Use the Add New Hardware icon in the control panel and incase if the device is not listed in the Device manager, reboot the computer.
6. Install the application software for the IEEE 1394 compliant device. (CD or Floppy is normally supplied along with the device).

## Visual Display Unit (VDU)

VDU is a device, such as a television screen, which produces a visible display of data. The VDU is also called Monitor.

The computer monitor is an output device that is part of the computer's display system. A cable connects the monitor to a **video adapter** (video card) that is installed in an expansion slot on the computer's motherboard. This system converts signals into text and pictures and displays them on a TV-like screen (the monitor).

The computer sends a signal to the video adapter, telling it what character, image or graphic to display. The video adapter converts that signal to a set of instructions that tell the display device (monitor) how to draw the image on the screen.

## Types of Monitors

There are many ways to classify monitors. The most basic is in terms of color capabilities, which separates monitors into three classes:

- **Monochrome** : Monochrome monitors actually display two colors, one for the background and one for the foreground. The colors can be black and white, green and black, or amber and black.
- **Gray-scale** : A gray-scale monitor is a special type of monochrome monitor capable of displaying different shades of gray.
- **Colour**: Colour monitors can display anywhere from 16 to over 1 million different colors. Color monitors are sometimes called RGB monitors because they accept three separate signals -- red, green, and blue.

There are mainly two types of monitors in use. They are:

1. CRT monitor



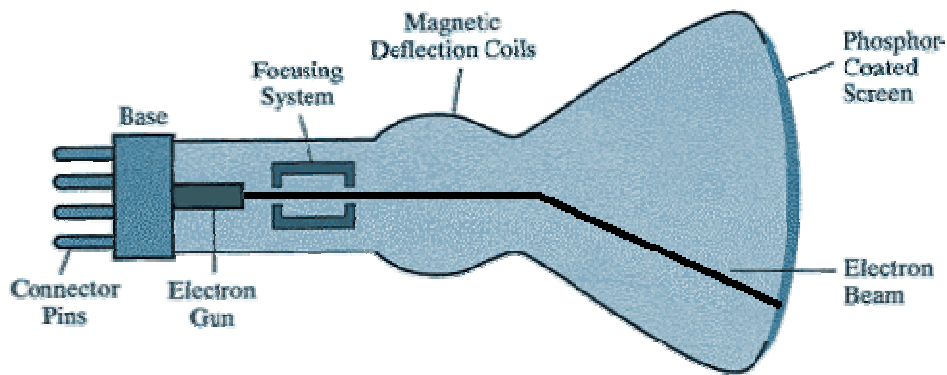
2. LCD Monitor



# 1. CRT Monitor

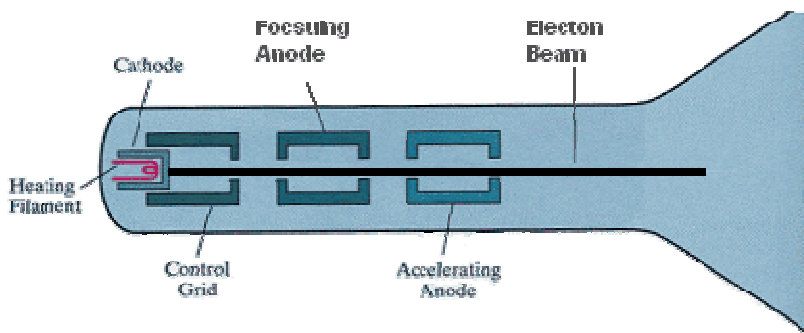
## Working Principle

A CRT is a vacuumed sealed tube with no air inside. In a CRT monitor, the electron gun produces a beam of electrons that travels through a focusing system, deflection coils, and then into the screen to display a picture. The rest of this website is dedicated to explaining the details of the system.



The above figure is the inside of a Cathode-ray tube. A beam of electrons (cathode rays) is emitted by the electron gun, passes through various focusing and deflection systems, and then hits specific areas on a phosphor-coated screen.

## Electron Gun



The electron gun consists of a metal cathode, control grid, and various anodes as labeled above in the figure. It is important to remember that electrons are small negatively particles, because their direction is controlled by the type of voltage. Negative charges repel each other, and opposites attract.

- First, heat is generated by an electric current passing through a *heating filament*, and it causes electrons to fly off the *cathode*.
- Intensity of the electron beam is controlled by the voltage at the *control grid*. A high negative voltage in the *control grid* will stop the electron beam.
- Next, inside the vacuum of the CRT, the *focusing anode* concentrates the electrons into a small beam so they can be accelerated.
- Finally, the negatively charged electrons are accelerated towards the screen by a high positive voltage at the *accelerating anode*.

### Focusing System

After the electron beam leaves the electron gun, the electrons go through another focusing system. The focusing system, a metal cylinder, uses a positive electric field that causes the electrons to converge into a small point. This assures that the electron beam will only hit one spot on the monitor at a time. Improving the focusing system increases the sharpness of the picture on the screen.

### Deflection Coils

The magnetic deflection coils are used to hit the correct part of the screen. They are mounted on all sides of the cathode-ray tube, and they control the horizontal and vertical direction of the electron beam. Varying the electricity running through the coils aims the beam at the proper screen location.

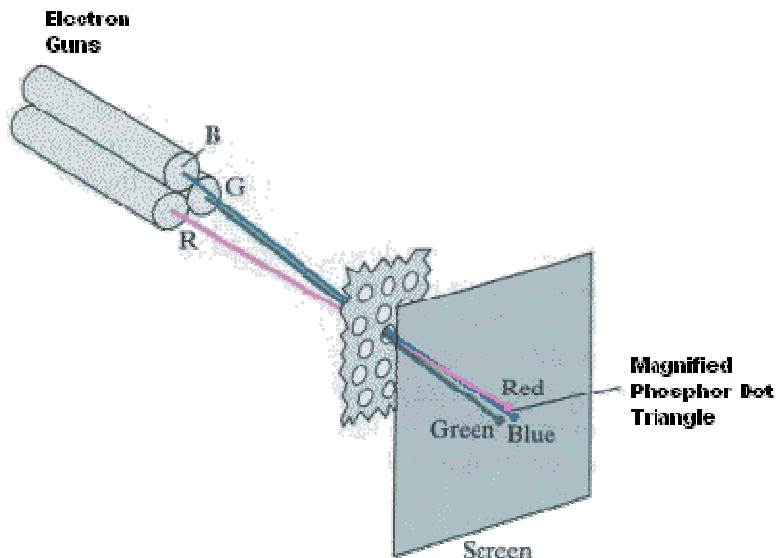
### The Phosphor Coating

The light on the screen that a user sees is caused by electrons illuminating a phosphor coating. Part of the energy from the electrons is converted to heat by friction, and the rest of the energy causes the phosphor to become "excited." The phosphor does not hold its excited state for long, and the light quickly dwindles. Different phosphors hold the light for different

amounts of time. The amount of time it takes for the phosphor to lose 9/10ths of its original intensity is called the *persistence*.

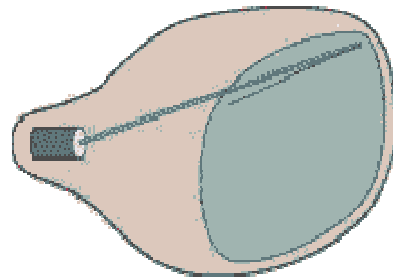
## Color

A CRT monitor displays color by using the phosphor and the shadow-mask method. This is the same system that televisions use. It is based on the RGB model, which means there is a red, green, and blue dot at each position on the screen. Three electron guns are used to activate each color separately. Varying the intensity of each electron gun, or shutting the beams off, determines the color. This is shown in the figure below. If all beams are off, then the dot is black. If all beams are on, then the color of the dot is white. Today, the average monitor has the capability to display millions of different colors.



## Raster Scanning

Since the phosphor dots lose their color and light very quickly, a system must be incorporated to refresh, or redraw, the picture at a high rate. The picture is refreshed by raster-scanning, which is based off television technology. The electron beams are swept along each row activating each spot on the screen to display the proper colors. As you can see in the figure below,



it activates each spot from top to bottom as well. Each of these spots is called a *pixel*. Increasing the refresh rate, number of pixels, and colors creates a better quality picture on the screen.

### The Color CRT Monitor

A CRT monitor is a system with many parts and methods, working in perfect unison. The process starts with electrons being emitted by the electron gun. They are pushed into the cathode-ray tube, and then the electrons light up phosphor pixels on the screen. So every picture that a computer user sees is not solid at all; it is actually made up of tiny individual pixels constantly be refreshed. This entire cycle is controlled behind the scenes by a complex video card and computer.

## 2. LCD Monitor (Liquid Crystal Display)

A Liquid crystal display or LCD is a digital display technology that produces images on a flat surface by shining light through liquid crystals and colored filters.

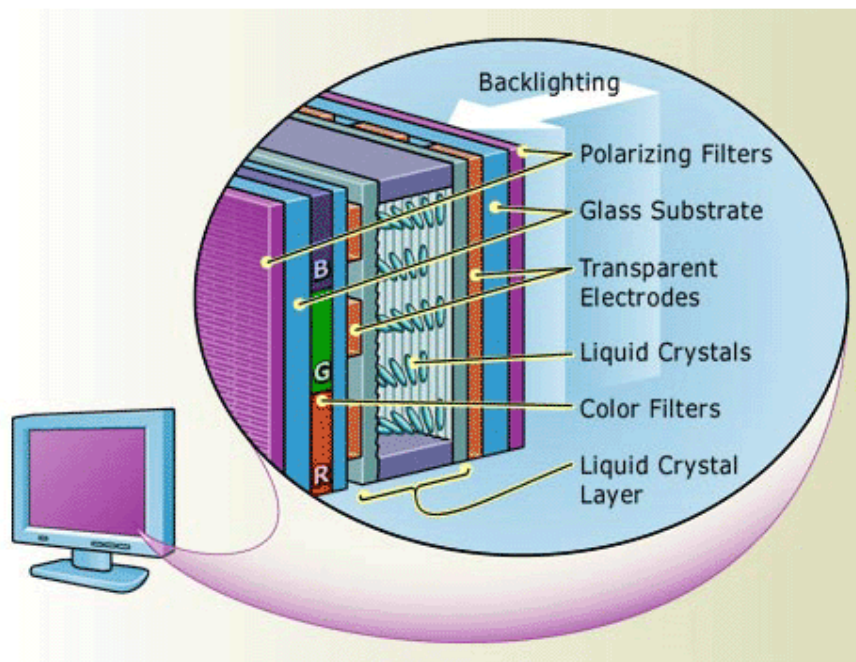
- Takes up less space, consumes less power, and produces less heat than traditional cathode-ray tube monitors.
- Lack of flicker and low glare reduce eyestrain.
- Much more expensive than CRTs of comparable size.

### Working principle

liquid crystals are liquid chemicals whose molecules can be aligned precisely when subjected to electrical fields--much in the way metal shavings line up in the field of a magnet. When properly aligned, the liquid crystals allow light to pass through.

Whether on a laptop or a desktop, an LCD screen is a multilayered, sideways sandwich. A fluorescent light source, known as the *backlight*, makes up the rearmost slice of bread. This light passes through the first of two polarizing filters. The polarized light then passes through a layer that contains thousands of liquid crystal blobs arrayed in tiny containers called *cells*. The cells are, in turn, arrayed in rows across the screen; one or more cells make up one pixel (the smallest discernible dot on a display). Electric leads around the edge of the LCD create an electric field that twists the crystal molecule, which lines the light up with the second polarizing filter and allows it to pass through. Figure below shows the constructional details of an LCD panel.





In an active-matrix color LCD, the backlight shines through a sandwich of filters, glass, and liquid crystals. First, a layer of polarizing filters align the light rays. The light then passes through cells filled with liquid crystal that, when twisted by an electrical field, bend the light. Finally, varying amounts of light drift through colored filters; these colors combine to produce the specific hue of each pixel.

For a simple monochrome LCD, such as you'd find on a PalmPilot, that's it: The protective cover goes on, and it's finished. But color LCDs, such as you'd find on a laptop PC, are more complex.

In a color LCD panel, each pixel is made up of three liquid crystal cells. Each of those three cells is fronted by a red, green, or blue filter. Light passing through the filtered cells creates the colors you see on the LCD. Occasionally the mechanism that sends the electrical current to one or more pixels fails; in those instances you'll see a completely dark, "bad" pixel.

### **Graphics Accelerator Cards**

Early graphics adapters received data from the processor and basically forwarded the signals to the monitor, leaving the CPU to do all the work related to processing and calculating. In non-accelerated graphics adapters, the computer needed to change each pixel individually to change the image on the screen. After

graphical user interfaces (like Windows for example) became popular, systems began to slow down as the CPU was left trying to move large amounts of data from the system RAM to the video card.

Today all new video cards are accelerated and are connected to the system's CPU through high-speed buses such as PCI or AGP.

Also known as a 3D accelerator, the graphics accelerator card is an internal board that generally is installed into the PCI or AGP slot and reduces the time it takes to produce images on the computer screen by incorporating its own processor and memory.

The biggest difference between accelerated and non-accelerated cards is that with accelerated video cards, the CPU no longer has to carry the bulk of the processing burden from graphics calculations. Since the video card has its own processor, it is able to perform most of the work, leaving your CPU free to process other tasks.

### **Synchronous Graphics RAM (SGRAM)**

**SGRAM** is a type of video adapter that contains its own processor to boost performance levels. These processors are specialized for computing graphical transformations, so they achieve better results than the general-purpose CPU used by the computer. In addition, they free up the computer's CPU to execute other commands while the graphics accelerator is handling graphics computations. The popularity of graphical applications, and especially multimedia applications and computer games, has made graphics accelerators not only a common enhancement, but a necessity.

The popularity of graphical applications, and especially multimedia applications, has made graphics accelerators not only a common enhancement, but a necessity. Most computer manufacturers now bundle a graphics accelerator with their mid-range and high-end systems.

## Printers

Printer is a very common output device which prints information and data from the computer on to paper. There are different types of printers.

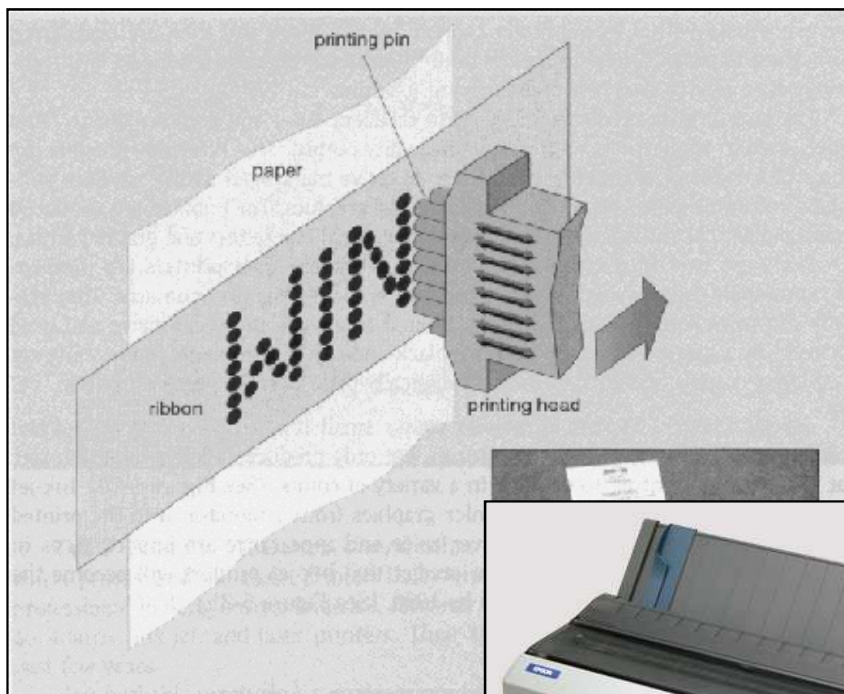
The various types of printers are:

1. Dot matrix printer.
2. Inkjet printer.
3. Thermal printer.
4. Laser printer.

### 1. Dot Matrix printer

It is an impact printer that forms characters as a series of dots.

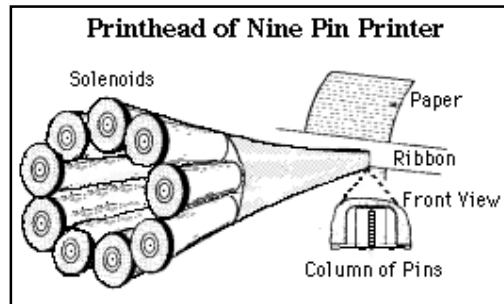
The print head contains a grid of pins which form a character. This heads stamps on to the paper through an inked ribbon to produce a character (for example, "H") that is made up of dots. The illustration below shows the working of a Dot matrix printer.



The print heads can have from 9 to 24 pins in them. The more pins the print head has, the better the quality of the printed output. Printers with 24 pins produce letter quality output.

Figure shows a print head.

The quality of the printed output is measured in dpi (dots per inch). The higher the dpi, the better the quality of the printed output.



Dot Matrix printers come with a standard parallel interface. They can also use a serial interface. Parallel interfaces are faster, but serial interfaces can be used over greater distances.

Impact printers are the only printers that will make an impression on carbon paper. Any printing that is done on paper with a carbon behind it or on self carbonated paper needs to be done on an impact printer for the printing come through on the second layer of paper.

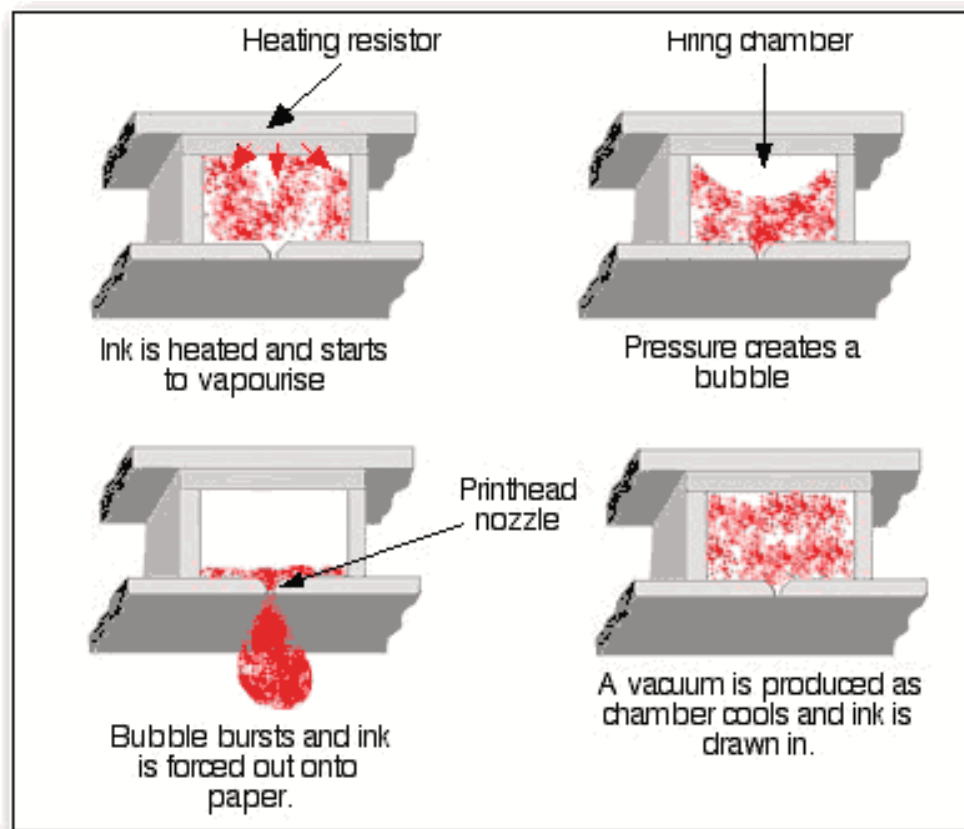
## **2. Ink Jet printer**

An Inkjet printer prints characters by spraying very fine jets of ink on to the paper. It works best with non-absorbent paper. Figure below shows a typical Ink jet printer.



## Operation

Inkjet printing, like laser printing, is a non-impact process. Ink is emitted from nozzles while they pass over media. The operation of an inkjet printer is easy to visualize: liquid ink in various colors being squirted onto paper and other media, like plastic film and canvas, to build an image. A print head scans the page in horizontal strips, using the printer's motor assembly to move it from left to right and back again, while the paper is rolled up in vertical steps, again by the printer. A strip (or row) of the image is printed, then the paper moves on,



ready for the next strip. To speed things up, the print head doesn't print just a single row of pixels in each pass, but a vertical row of pixels at a time. The working of an Inkjet printer is shown by the illustration below.

### **3. Thermal printer**

In a Thermal printer characters are formed by heated elements being placed in contact with special heat sensitive paper forming darkened dots when the elements reach a critical temperature.

Thermal printer paper tends to darken over time due to exposure to sunlight and heat. The standard of print produced is poor.

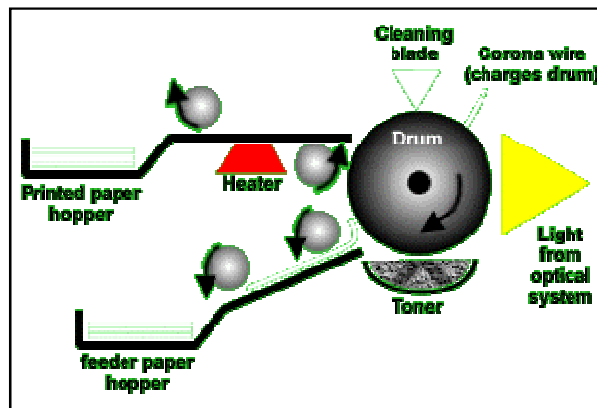
Thermal printers are widely used in battery powered equipment such as portable calculators. Figure shows a Thermal printer.



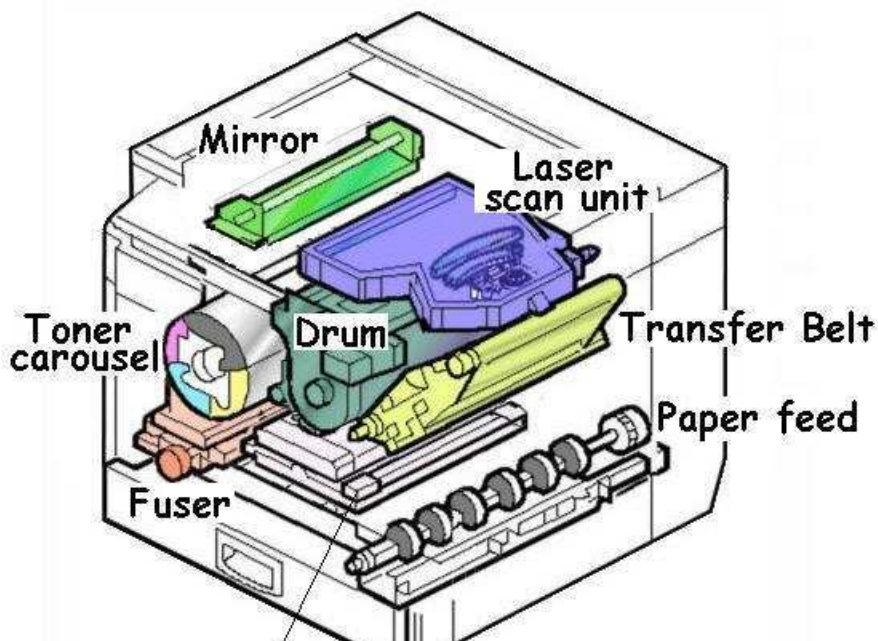
#### 4. Laser printer

A Laser printer uses a laser beam to print. The laser creates an image on a drum inside the printer. This picks up toner and prints the image on to the paper like a photocopier does as shown in the figure.

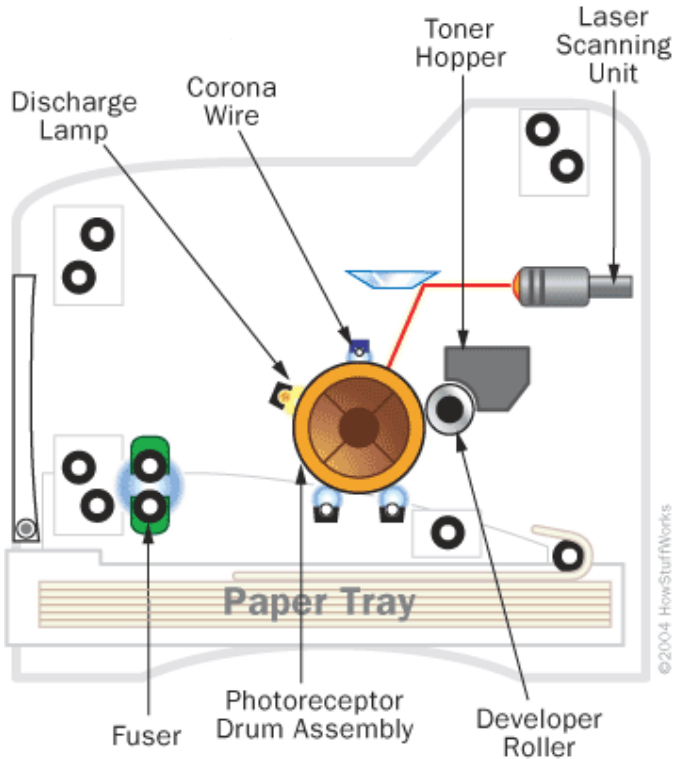
A photoconductive drum is initially charged and then a high intensity laser beam is used to discharge selected areas on the drum. These discharged areas correspond to the white areas of the printed document.



Toner is attracted to parts of the drum with a high charge. The drum rotates and transfers the toner to the paper which has an even greater electrical charge. Finally a heater fixes the toner onto the paper.



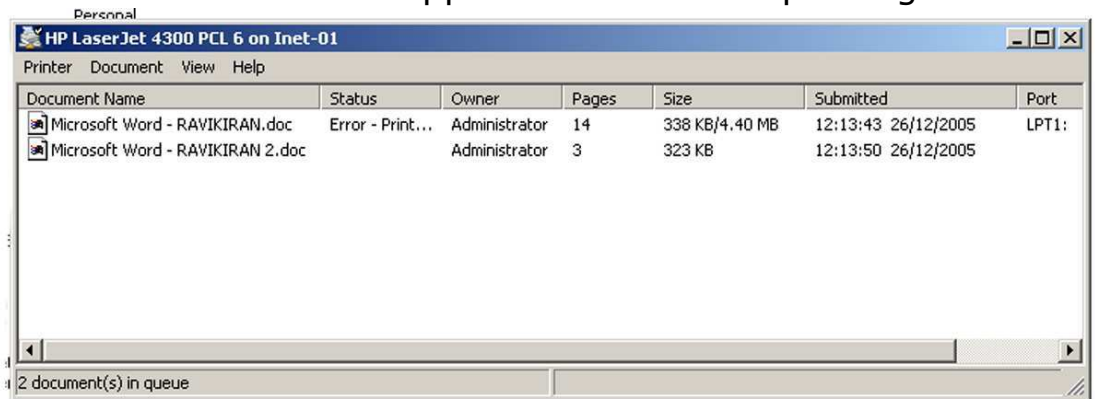
The main parts of a Laser printer are shown in the figures below:



## How Windows handles a Printing job

When windows receives a print job from an application, it places the job in the queue and prints from

the queue. This is to release the application from the printing job as soon as possible. Several print jobs can



accumulate in the queue and can be viewed in the Printer window as shown in the figure below.

The process of queue printing is called **Spooling**.

**Spooling** means simultaneous peripheral operations online. Most of the windows printing jobs are done by spooling.



If the Printer port, printer cable and printer all support bidirectional (Two way) communication, the printer can communicate with the Windows. For example, Windows XP can ask the printer how much printer memory is available and what fonts are installed. The printer can send messages like 'Paper-jam' or 'Out-of-paper' to the Windows XP.

Windows manages the printing jobs using one of these methods:

- For Windows 9x or Windows NT/2000/XP using a PostScript printer, the print job data is converted to the PostScript language. PostScript is a language used to communicate how a page has to be printed.
- For Windows 9x applications using non-PostScript printer, the print job data is converted to Enhanced Meta File (EMF). This format contains the print command along with the data to help in speed printing.
- For Windows 2000/XP, a printer language used is PCL (**P**rinter **C**ontrol **L**anguage). Most of the Printer manufacturers use PCL.
- Text data that contains no embedded control characters (With out fomating) is sent to the printer as it is. When DOS applications use this type of printing, the data is called raw data and is sent to the printer directly. This will even bypass the printer queue.

## Installing Printers

A Printer connected to the output port of a computer is called a **Local Printer**. and then the computer can share this printer with others on the network. When the printer is accessed by way of the network it is called a **Network Printer**. A computer can have several printers installed. Windows designates one printer to be the **default printer**.

The procedure for installing a printer is as follows:

1. Turn off the computer. If necessary, remove any spacers or tape that is used to secure the internal components of the printer as shown in the figure. Assemble the printer referring to the manual provided.



2. Connect the printer cable to the port on the printer as shown in figure.



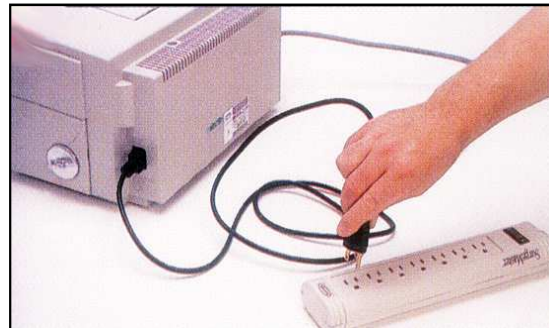
3. Connect the other end of the printer cable to the parallel port at the back of the computer as shown in figure.



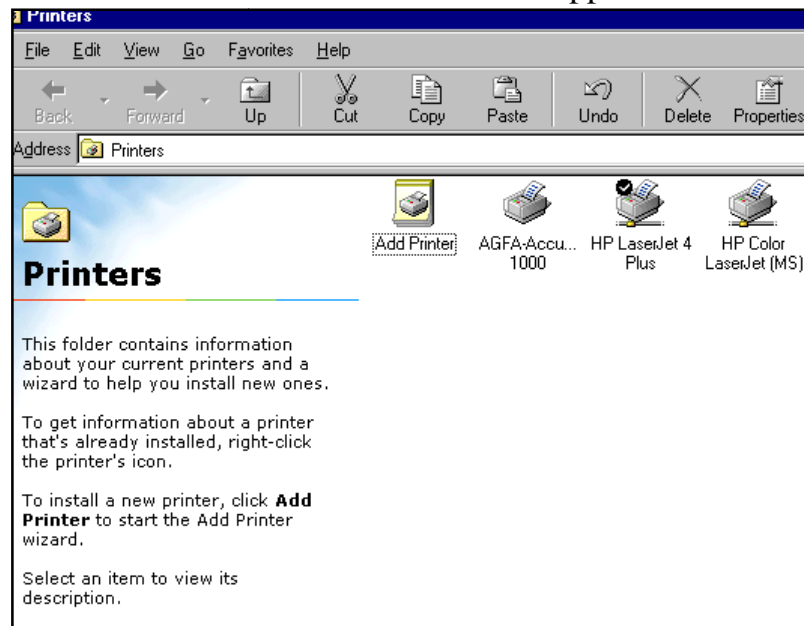
4. Connect the power cable to the power connector on the printer as shown in figure.



5. Plug the power cable into an electrical outlet as shown in figure.



6. Switch on the power for the printer and computer.
7. After the computer boots-up, Click the Start button, point to Settings, and then click Printers. The window shown below appears.



8. Double-click Add Printer, and then click Next.
9. Click Local Printer or Network Printer as appropriate, and then click Next.
10. Click the appropriate manufacturer and model for your printer, and then click Next
11. If you chose to install a local printer, click the correct port and then click Next

12. Type a name for the printer (or accept the default name), and then click either Yes or No in the "Do you want your Windows-based programs to use this printer as the default printer?" area. Click Next.
13. To print a test page, click Yes. Click Finish

### **Sharing a printer between multiple computers**

Users who have multiple computers in their home or work environment may wish to share a printer with the other computers to help save on cost and for convenience. Below are some of the ways you can share a printer on a network for different types of setups.

Before setting up the printer to be shared on a network, you must decide what type of setup you plan on doing. Below are the different types of setups and their advantages and disadvantages.

- 1) Printer connected to a computer / server
- 2) Dedicated print server
- 3) Network printer

#### **1) Printer connected to computer / server**

Connecting a printer to a computer and sharing the printer from that computer is the most common solution for sharing a printer because of the ease and price. The primary disadvantage of this is that the computer must always be on in order for the printer to work. Although this solution may not be the best solution for everyone it is usually the easiest and cheapest solution for sharing your printer between all the computers on your network.

Below are steps for sharing your printer with each of the major Microsoft Windows operating systems. For these steps to work you must already have a network setup at your home or office.

### **Setting up the printer to be shared on the network**

#### **Microsoft Windows 2000 / XP**

1. Open the Windows Control Panel
2. Double-click the "Printers" or "Printers and Faxes" icon.
3. If you currently have a printer already installed on the computer right click the printer icon and click Properties. In the Properties window click the sharing tab and make sure the printer is shared.
4. Once the printer has been shared on the computer skip to the next section for adding the shared printer on each of the computers on your network.

## Microsoft Windows 95 / 98

1. Open the Windows Control Panel
2. Double-click Network
3. In the Network window under the Configuration tab click the File and Print Sharing button.
4. In the File and Print Sharing Window check the "I want to be able to allow others to print to my printer(s)" option. Note: If the computer asks to restart the computer after selecting this option make sure to restart.
5. Click Ok and then Ok again to close out of the Network Window.
6. Once back in the Control Panel double-click printers
7. If you currently have a printer already installed on the computer right click the printer icon and click Properties. In the Properties window click the sharing tab and make sure the printer is shared.
8. Once the printer has been shared on the computer skip to the next section for adding the shared printer on each of the computers on your network.

## Adding the shared printer

### Microsoft Windows 2000 / XP

1. Open the Windows Control Panel
2. Double-click the "Printers" or "Printers and Faxes" icon.
3. Double-click the "Add a printer" icon.
4. Once in the Printer Wizard click the Next button to get started.
5. Select either "A network printer, or a printer attached to another computer" or "Network Printer" and click Next. This option is different in Windows 2000 and Windows XP.
6. Type the name of the printer's network address. This will be the name of the computer and then the name of the printer. For example, if the computer on our network was called "Ravi" and our printer was called "hp" the network path for our printer would be: "\\Ravi\hp", if you are adding the printer on a Microsoft Windows XP computer you can click the Browse icon to browse your network and locate the printer.
7. Once the path has been typed in click Next.
8. If the printer was located over the network Windows should either install the printer or you may be required to install the drivers for that printer on the computer before completing the installation.

## Microsoft Windows 95 / 98

1. Open the Windows Control Panel

2. Double-click Printers
3. Double-click the "Add a printer" icon.
4. Click Next to start the Add a printer wizard.
5. Select Network Printer and click Next.
6. Type the network name of the printer. This will be the name of the computer and then the name of the printer. For example, if the computer on our network was called "hope" and our printer was called "hp" the network path for our printer would be: "\\Ravi\hp", or click the browse icon to browse to the location of the printer.
7. If the printer was located over the network Windows should either install the printer or you may be required to install the drivers for that printer on the computer before completing the installation.

## **2) Dedicated print server**

If we do not wish to have a computer always turned on and maintained you may want to get a hardware print server. A print server enables you to connect a small network appliance to your network and connect each your printers to it.

## **3) Network printer**

Some printers, often high-end printers have the capability to connect directly to a network or have hardware installed into them that enables the printer to be connected to the Network. If your printer supports this option and has the necessary hardware you should be able to connect the printer directly to the Network and have it detected by the computers.

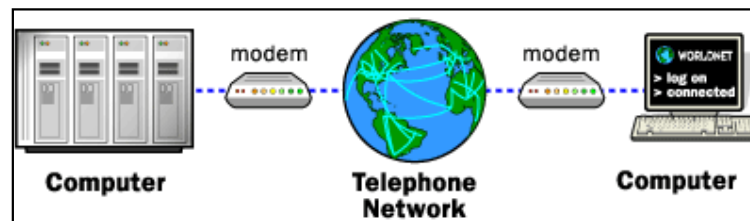
-

## Modems

The word "modem" is a contraction of the words **modulator-demodulator**. A modem is typically used to send digital data over a phone line.

The sending modem **modulates** the data into a signal that is compatible with the phone line, and the receiving modem **demodulates** the signal back into digital data. **Wireless modems** convert digital data into radio signals and back.

Modems came into existence in the 1960s as a way to allow terminals to connect to computers over the phone lines. A typical arrangement is shown below:



## Modem standards

Most modem standards are referred to by a code assigned by the **Consultative Committee for International Telephony and Telegraphy (CCITT)**.

The standards fall into three categories:

1. Modulation (speed)
2. Error correction
3. Data compression

### **1. Modulation (speed) standards**

Modulation (or speed) standards involve the rates and ways modems communicate with each other, and how they negotiate the best communication speed they can both use for the connection. These are common modulation

standards:

**V.22.**

The CCITT standard for data transmission at speeds up to 2400 bps.

**HST**

The USR proprietary standard for data transmission at speeds from 9600 to 16800 bps, depending on the model.

The reverse channel is much slower. If you are connecting to another type of modem, the best speed you can expect is 2400 bps.

**Hayes V-series**

The Hayes proprietary standard for data transmission at speeds up to 9600 bps.

The reverse channel is much slower. If you are connecting to another type of modem, the best speed you can expect is 2400 bps.

**V.32**

The CCITT standard for data transmission at speeds up to 9600 bps.

**V.32bis**

The CCITT standard for data transmission at speeds up to 14400 bps.

**V.32terbo**

The AT&T proprietary standard for data transmission at speeds up to 19200 bps.

**V.fc**

The Rockwell chip set proprietary standard for data transmission at speeds up to 28800 bps.

**V.34**

The CCITT standard for data transmission at speeds up to 28800 bps.

**K56flex**

The Lucent and Rockwell proprietary standard for 56K modems.

**x.2**

The US Robotics proprietary standard for 56K modems.

**V.90**

The CCITT standard for 56K modems.

**2. Error correction standards**



Error correction standards provide a way of correcting errors that result from outside interference, such as noise on the phone line. Error correction ensures that data coming out of the receiving modem is exactly the same as data going into the sending modem.

Error correction standards correct only those errors occurring between the two modems. They cannot correct errors occurring between the modem and the computer (a connection which is considerably more reliable when proper cables are used and connections secured).

### **3. Data compression standards**

Data compression standards provide a way of compressing data at the sending modem, transmitting it across the modem link in compressed form, and then expanding it at the receiving modem.

If the data can be compressed, data compression increases the effective throughput. If the data cannot be compressed (for example, if it has already been compressed with a utility such as Stuffit or PKZip), then modems with data compression give little benefit.

### **Baud rate**

The measure of how fast a modem transfers data. The faster the baud rate, the faster the data transfer. If you divide the baud rate by 10, you get a rough estimate of the number of bytes (or characters) transferred per second, and this provides an idea of how long a file transfer will take. For example, a 2400 bps modem can transfer about 240 bytes per second (that is, about 1Kb every four seconds). Therefore, a 25Kb file would take about 100 (25x4) seconds to transfer.

### **Flow control and handshaking**

Most high speed modems can still connect to older, lower speed modems. When they do, they receive information from the compute at high speed, and send the data out at a lower

speed. Therefore, they may have to instruct the computer to slow down periodically.

Flow control is the method a modem uses to control the quantity of data the computer sends to the modem. It ensures that data is not lost if it is sent to the modem faster than the modem can accept it.

There are two widely recognized standard methods for flow control. They are:

1. Software handshaking (using XON/XOFF)
2. Hardware handshaking (using RTS/CTS).

## Installation of a Dial-up Modem

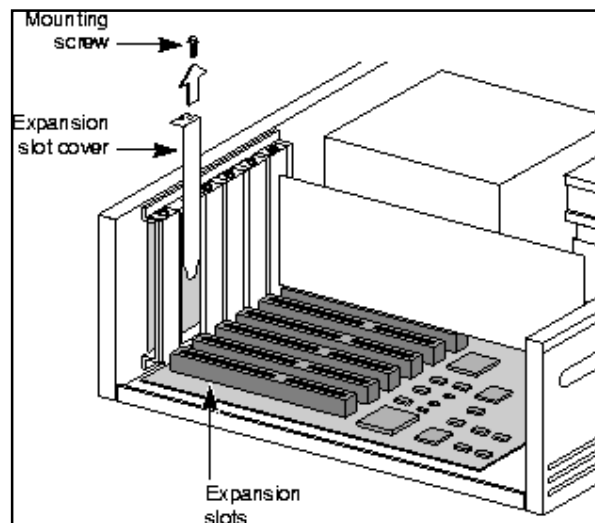
### Procedure Steps:

1. Turn System Off and unplug it.
2. If your modem is a conventional type that uses jumpers to set the COM port and IRQ number, you need to configure the jumpers according to the instructions in your modem manual. If your modem is plug and play (most newer ones are) then you do not need to configure any jumpers on it.

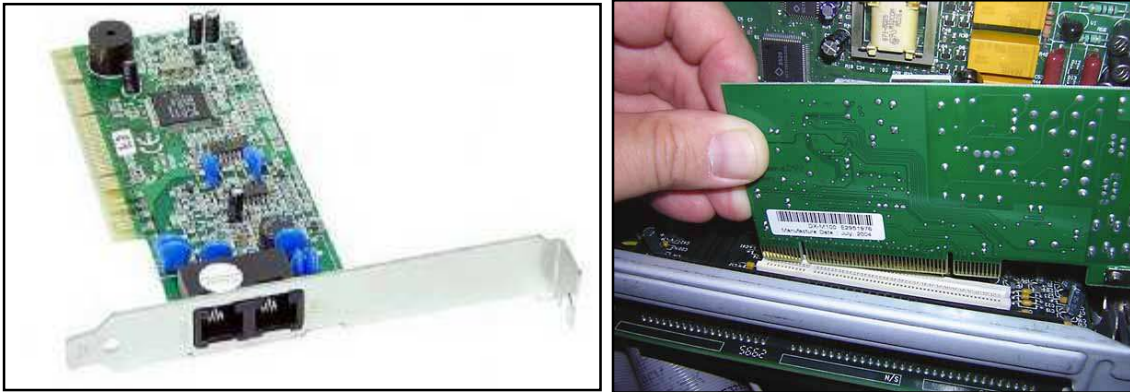
3. Select an open expansion slot (shown above), preferably away from as many other components as possible.



4. Unscrew and remove the metal insert on the back of the system case that corresponds to this expansion slot.



5. Insert the card into the expansion slot carefully. You may have to rock it back and forth from front to back to get it to go in. Figure below shows a modem card and insertion of the card into the slot.



6. Secure the card by screwing it into place.
7. Connect one end of the phone cord to the back of the modem (in the "Wall" or "Line" jack) and the other end to the wall socket.



*Included with your new modem should be a CD (or perhaps a floppy diskette) that contains the software the computer needs to recognize and work with the modem. This small utility program is called a **device driver**. Most computers come with preinstalled device drivers for a selection of the most common modems, but it's best to work with the software the modem manufacturer provides, as it is specifically created to match your modem. (It also may be more up-to-date than that supplied with your computer's operating system.)*

8. When your computer starts up after the new modem has been installed, you should see the Found New Hardware Wizard. Follow the instructions on your screen, and the wizard will configure the computer to work with the new modem.

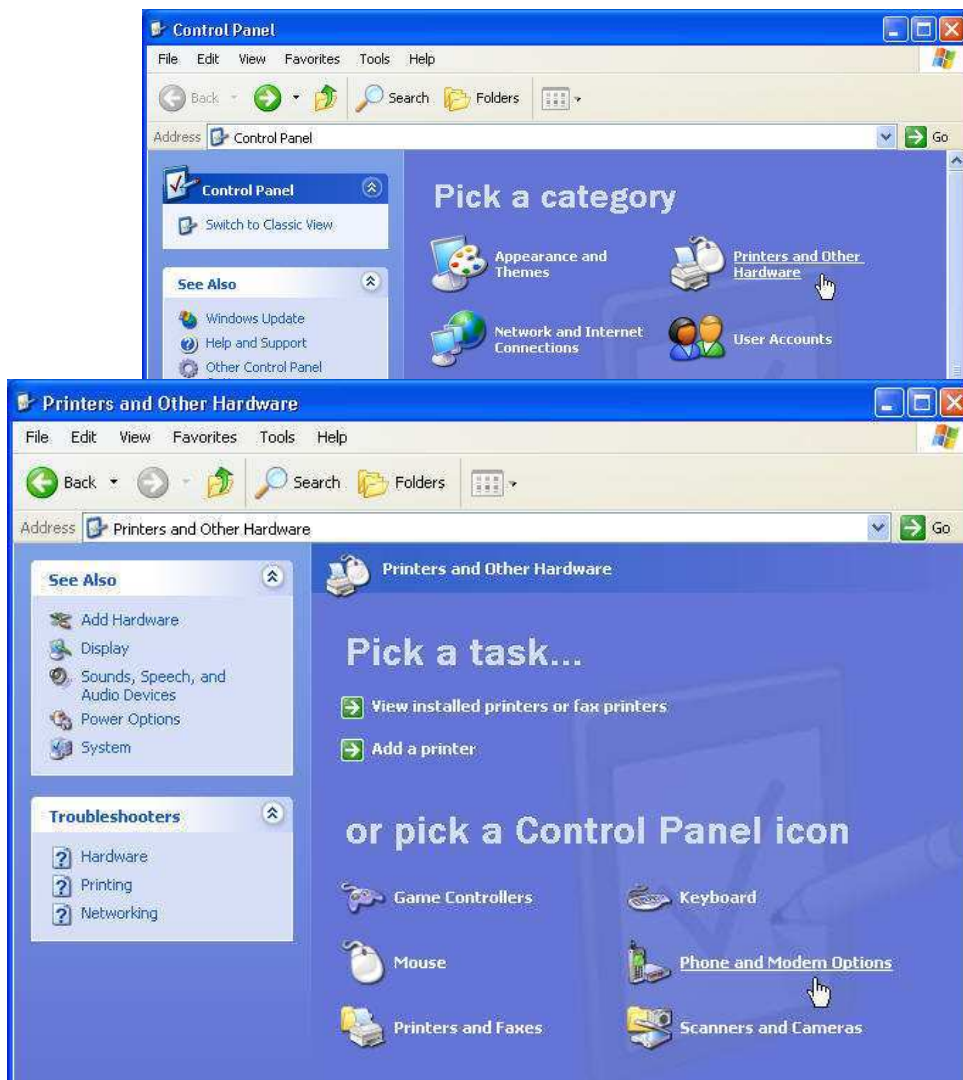
9. Insert the disc into the CD (or floppy) drive on your computer when the wizard prompts you to do so. When the installation is complete, the wizard will ask you to restart your computer.

10. If the Found New Hardware Wizard doesn't start up, Follow the steps given below (For Windows XP)

a) Click Start -> Control Panel



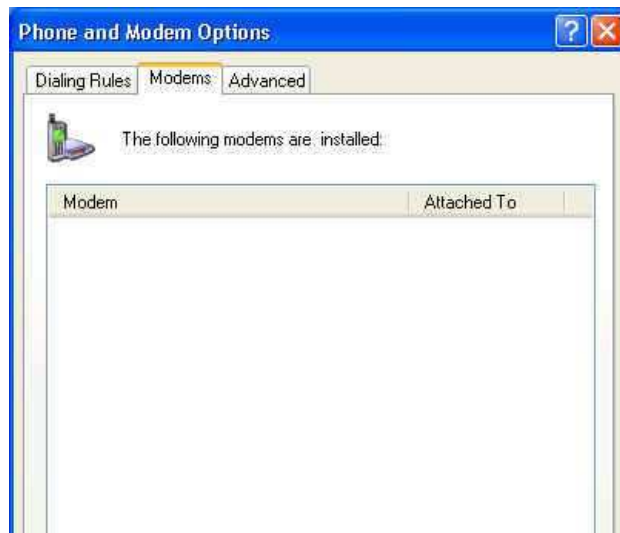
b) Double click on the Printers and Other Hardware icon



c)

Double click on **Phone and modem Options**

d) To install a new modem, click on Modems tab and the Add button.



the press

e) Click **Next**



f) Windows will now detect your modem.



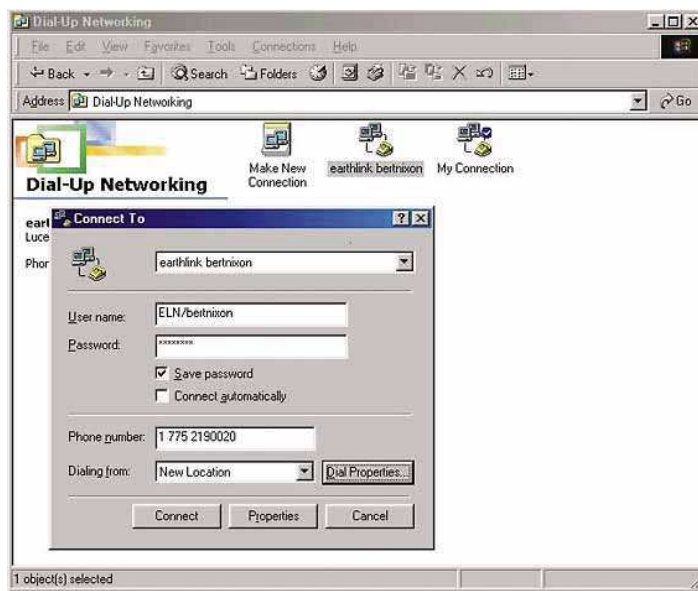
g) Press Finish to complete the installation procedure.



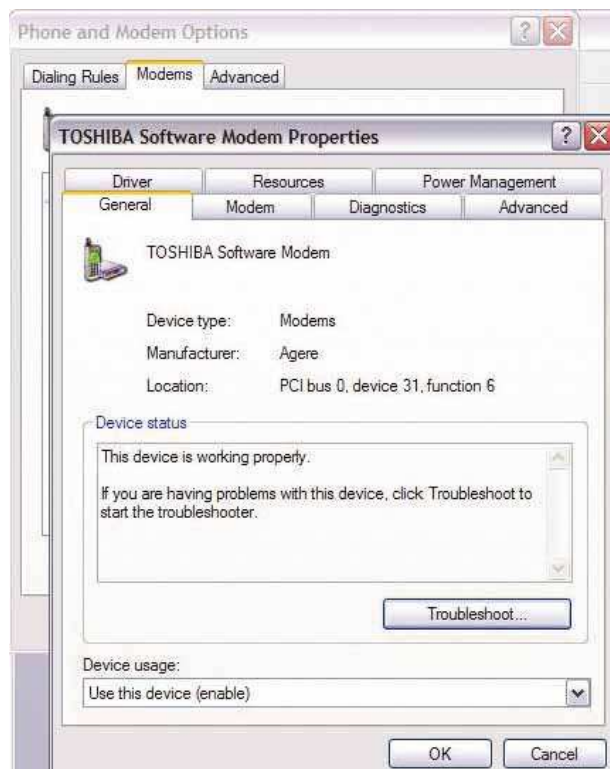
Note: By following the instructions above, you can install the majority of the modems that can be found in the market. However, if for any reason the installation of your modem fails, you will have to consult the manual that came with the modem for further instructions. There is also a possibility that your modem gets installed automatically by your O/S, since Windows XP uses the Plug and Play technology.

## 11. Set Up Your Modem

- a) Click Start, Settings, and Control Panel and select Dial-Up Networking.



- b) Select the icon labeled Make New Connection. You will need to name the connection (the ISP name, for instance) and supply the ISP's dial-up information, as well as the username and password you use with the ISP.





c) you can set up the connection by opening the Internet Options Icon on the Control Panel. When you select the Connections tab, you will see a button labeled Setup; clicking the button will guide you through setting up your connection. You will need some information from your ISP to do this, including the phone number to call, the names of the ISP's email servers, the type of service, and your password for initial login.

## Introduction to networking

### What Is a Network?

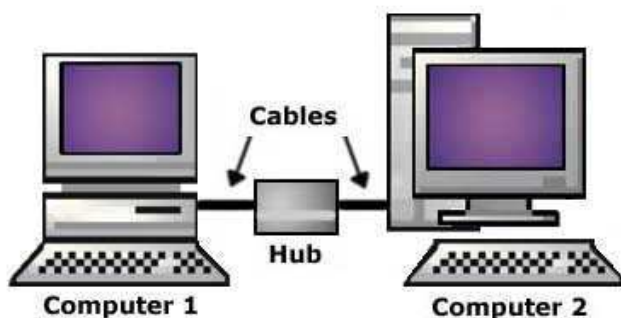
A network is created when two or more computers are connected to share information and resources.

### Basic Components

- **Cables** - carry electronic signals from one computer to another
- **Connectors** - attach cables to network interface cards
- **Hub(s)** - connects all computers in a network to each other and helps direct messages to appropriate recipients; as more users join the network, more hubs may be added
- **Individual computers** - for users to compose and retrieve data
- **Protocol** - accepted common format (i.e., computer language) that all computers on a network use to transmit data. There are many standard protocols, among them: AppleTalk, Ethernet, NetBEUI, and TCP/IP
- **Network Interface Cards (NICs)** - inserted into each computer (some newer models will come with the cards already in place) to enable the computers to send and receive messages; this is where the cables will be connected
- **Network Software** - computer applications used to operate the network and its many components
- **Server** - computer which is the central data storage unit for the network; all other computers connected to the network work off of the server

### Basic Set-up

The basic concept of networking is quite simple. The computers are usually connected via cables\* to a central hub.



\*Large networks also use satellites and radio waves to cover great distances.



**Shared drive:** a computer's permanent storage drive that allows remote access to multiple users

### **Benefits of a Networked Office**

Networks can improve the efficiency and cost effectiveness of office work by allowing you to:

- Store commonly used files on a shared drive for easier access.
- Share files in electronic form to reduce printing costs and paper waste.
- Share resources, such as hard drives, printers, scanners, etc., to get more use out of office equipment.
- Share software with multi-user licenses.
- Send files to individual co-workers with intra-office e-mail.
- Perform data backups quicker and easier with files on shared drives.
- Give co-workers access to view files on your hard drive from their computer.
- Connect your entire office to the Internet through one line.

This article explains what a Local Area Network (LAN) is and describes the different types of LANs and how each can be helpful to your organization.

## Introduction

By definition a network is simply two or more computers communicating with each other. By understanding the different kinds of networks and how they work you will be able to determine which form of LAN will suit your program's needs best. Some LANs are rather simple and easy to set up while others are much more complex and will require the help of a trained professional.

## Definitions

**Network-enabled** - hardware device with the necessary ports for network cables.

## Peer-to-Peer

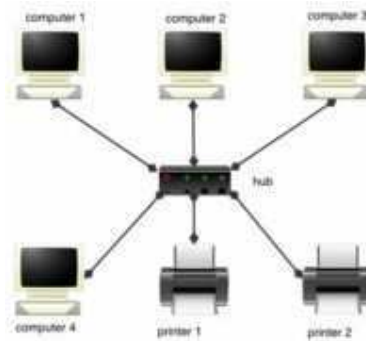
Peer-to-Peer networks are the simplest of all LAN set-ups. Not only are they easy to configure and require few resources, they are also inexpensive to implement. Peer-to-Peer, also called P2P networks, are the most direct method of connecting computers, requiring only a network cable for the simplest of networks

A simple Peer-to-Peer network can be a simple collection of two computers and a printer. Computer 1 will be able to print on the printer connected to Computer 2, if Computer 2 sets up the sharing permissions for the printer. Already we can see the advantages of building a small LAN: Files can be shared between the two computers and there is only one printer that must be purchased and maintained. A LAN like this is very simple to set up. In fact, many households may have LANs just like this



**Diagram 1**

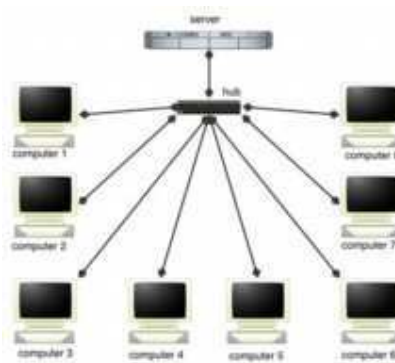
When a LAN needs to include a larger number of computers, a hub may be used. This hub will enable all the computers connected to the network to communicate with each other without having to connect each individual computer to each and every other computer on the network. This saves space and considerable set-up time and expense. Again, this P2P network is fairly simple to setup and allows all the computers to communicate with each other and to print using one or two printers. One note of caution - Printer 1 and Printer 2 in this diagram must be network enabled. If they are not, one could simply connect Printer 1 to Computer 4 and connect Printer 2 to Computer 3 and the printers could be shared as in Diagram 1.



**Diagram 2**

#### **Client/Server**

Client/Server LANs are generally much larger in scope than P2P networks. A file server is the most common server used. It frees up hard drive space on all the client machines and provides a single place to store files for everyone to access. Other popular servers include mail servers to send and deliver email, web servers to host Internet sites, database servers, and print servers. The complexity of client/server LANs can range from fairly simple networks to networks comprised of thousands of client machines and numerous servers that require an entire staff dedicated to keeping the network up and running.



**Diagram 3**

**Summary**

There are benefits to both client/server and peer-to-peer LANs. If a simple communication method is desired, a P2P network can be set up quickly, easily, and inexpensively. When the size of the network grows quite large one should consider implementing a client/server configuration. This can provide a central storage area for files (thus keeping client machines free from large numbers of files), ensuring that the most current file will be available to everyone on the network. This central storage area also makes backing up files much simpler since only the server needs to be backed up. The costs of larger client/server networks increase with size, however, and once they reach a certain size one must also take into consideration the personnel cost to maintain the network.

## Computer Network Sharing

With the availability and power of today's personal computers, you might ask why networks are needed. From the earliest networks to today's high-powered personal computers, the answer has remained the same: networks increase efficiency and reduce costs. Computer networks achieve these goals in three primary ways:

- Sharing information (or data)
- Sharing hardware and software
- Centralizing administration and support

More specifically, computers that are part of a network can share:

- Documents (memos, spreadsheets, invoices, and so on).
- E-mail messages.
- Word-processing software.
- Project-tracking software.
- Illustrations, photographs, videos, and audio files.
- Live audio and video broadcasts.
- Printers.
- Fax machines.
- Modems.
- CD-ROM drives and other removable drives, such as Zip and Jaz drives.
- Hard drives.

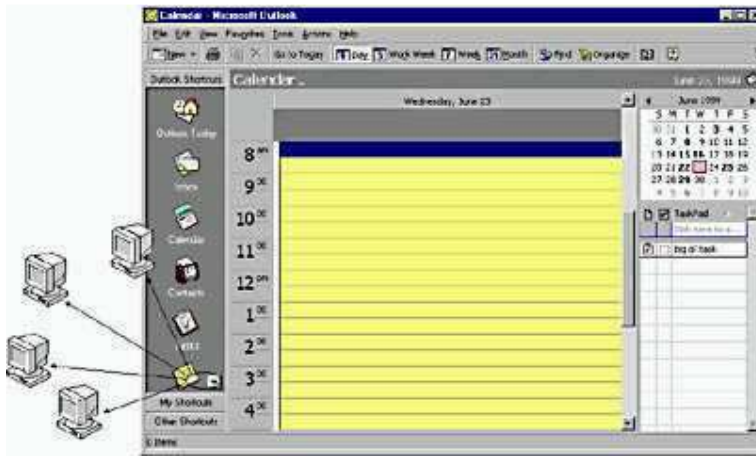
And more sharing options exist. The capabilities of networks are constantly expanding as new ways are found to share and communicate by means of computers.

## Sharing Information (or Data)

The ability to share information quickly and inexpensively has proven to be one of the most popular uses of networking technology. It has been reported that e-mail is by far the number-one activity of people who use the Internet. Many businesses have invested in networks specifically to take advantage of network-based e-mail and scheduling programs.

By making information available for sharing, networks can reduce the need for paper communication, increase efficiency, and make nearly any type of data available simultaneously to every user who needs it. Managers can use these utilities to communicate quickly and effectively with large numbers of people and to organize and schedule meetings with people drawn from an entire company or business enterprise far more easily than was previously possible. (See Figure 1.4.)

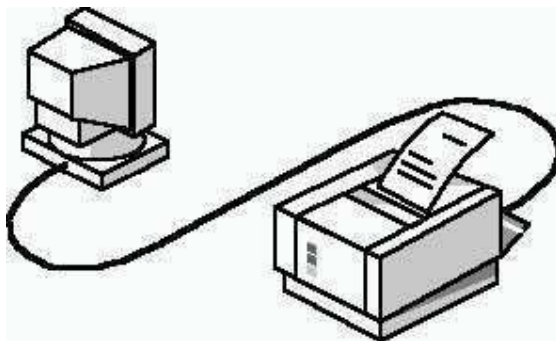




**Figure 1.4** *Scheduling a meeting with Microsoft Outlook*

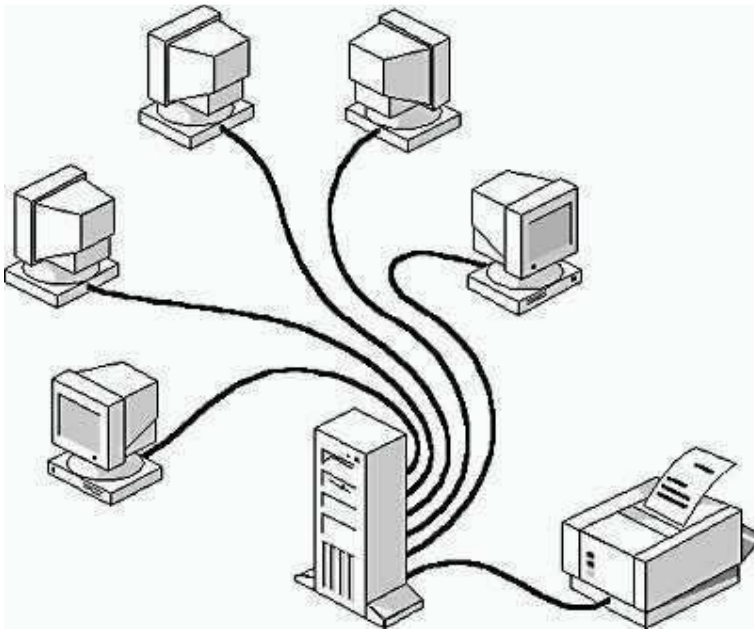
### **Sharing Hardware and Software**

Before the advent of networks, computer users needed their own printers, plotters, and other peripherals; the only way users could share a printer was to take turns sitting at the computer connected to the printer. Figure 1.5 shows a typical stand-alone workstation with a printer.



**Figure 1.5** *A printer in a stand-alone environment*

Networks make it possible for several people to share data and peripherals simultaneously. If many people need to use a printer, they can all use the printer available on the network. Figure 1.6 shows a typical network environment in which five workstations share a single printer.



**Figure 1.6** *Sharing a printer in a networking environment*

Networks can be used to share and standardize applications, such as word processors, spreadsheets, inventory databases, and so on, to ensure that everyone on the network is using the same applications and the same versions of those applications. This allows documents to be shared easily and creates training efficiencies: it is easier for people to master one word processing application thoroughly than to try to learn four or five different word processing applications.

### **Centralizing Administration and Support**

Networking computers can simplify support tasks as well. It is far more efficient for technical personnel to support one version of one operating system or application and to set up all computers in the same manner than to support many individual and unique systems and setups.

## 1.15 Network Security

It's very important to understand that in security, one simply cannot say "what's the best firewall?" There are two extremes: absolute security and absolute access. The closest we can get to an absolutely secure machine is one unplugged from the network, power supply, locked in a safe, and thrown at the bottom of the ocean. Unfortunately, it isn't terribly useful in this state. A machine with absolute access is extremely convenient to use: it's simply there, and will do whatever you tell it, without questions, authorization, passwords, or any other mechanism. Unfortunately, this isn't terribly practical, either: the Internet is a bad neighborhood now, and it isn't long before some bonehead will tell the computer to do something like self-destruct, after which, it isn't terribly useful to you.

This is no different from our daily lives. We constantly make decisions about what risks we're willing to accept. When we get in a car and drive to work, there's a certain risk that we're taking. It's possible that something completely out of control will cause us to become part of an accident on the highway. When we get on an airplane, we're accepting the level of risk involved as the price of convenience. However, most people have a mental picture of what an acceptable risk is, and won't go beyond that in most circumstances. If I happen to be upstairs at home, and want to leave for work, I'm not going to jump out the window. Yes, it would be more convenient, but the risk of injury outweighs the advantage of convenience.

Every organization needs to decide for itself where between the two extremes of total security and total access they need to be. A policy needs to articulate this, and then define *how* that will be enforced with practices and such. Everything that is done in the name of security, then, must enforce that policy uniformly.

### Types And Sources Of Network Threats

Now, we've covered enough background information on networking that we can actually get into the security aspects of all of this. First of all, we'll get into the types of threats there are against networked computers, and then some things that can be done to protect yourself against various threats.

#### Denial-of-Service

*DoS* (Denial-of-Service) attacks are probably the nastiest, and most difficult to address. These are the nastiest, because they're very easy to launch, difficult (sometimes impossible) to track, and it isn't easy to refuse the requests of the attacker, without also refusing legitimate requests for service.

The premise of a DoS attack is simple: send more requests to the machine than it can handle. There are toolkits available in the underground community that make this a simple matter of running a program and telling it which host to blast with requests. The attacker's program simply makes a connection on some service port, perhaps forging the packet's header information that says where the packet came from, and then dropping the connection. If the host is able to answer 20 requests per second, and the attacker is sending 50 per second, obviously the host will be unable to service all of the attacker's requests, much less any legitimate requests (hits on the web site running there, for example).

Such attacks were fairly common in late 1996 and early 1997, but are now becoming less popular.

Some things that can be done to reduce the risk of being stung by a denial of service attack include

- Not running your visible-to-the-world servers at a level too close to capacity
- Using packet filtering to prevent obviously forged packets from entering into your network address space.

Obviously forged packets would include those that claim to come from your own hosts, addresses reserved for private networks as defined in RFC 1918 [4], and the *loopback* network (127.0.0.0).

- Keeping up-to-date on security-related patches for your hosts' operating systems.

### Unauthorized Access

“Unauthorized access” is a very high-level term that can refer to a number of different sorts of attacks. The goal of these attacks is to access some resource that your machine should not provide the attacker. For example, a host might be a web server, and should provide anyone with requested web pages. However, that host should not provide command shell access without being sure that the person making such a request is someone who should get it, such as a local administrator.

### Executing Commands Illicitly

It's obviously undesirable for an unknown and untrusted person to be able to execute commands on your server machines. There are two main classifications of the severity of this problem: normal user access, and administrator access. A normal user can do a number of things on a system (such as read files, mail them to other people, etc.) that an attacker should not be able to do. This might, then, be all the access that an attacker needs. On the other hand, an attacker might wish to make configuration changes to a host (perhaps changing its IP address, putting a start-up script in place to cause the machine to shut down every time it's started, or something similar). In this case, the attacker will need to gain administrator privileges on the host.

### Confidentiality Breaches

We need to examine the threat model: what is it that you're trying to protect yourself against? There is certain information that could be quite damaging if it fell into the hands of a competitor, an enemy, or the public. In these cases, it's possible that compromise of a normal user's account on the machine can be enough to cause damage (perhaps in the form of PR, or obtaining information that can be used against the company, etc.)

While many of the perpetrators of these sorts of break-ins are merely thrill-seekers interested in nothing more than to see a shell prompt for your computer on their screen, there are those who are more malicious, as we'll consider next. (Additionally, keep in mind that it's possible that someone who is normally interested in nothing more than the thrill could be persuaded to do more: perhaps an unscrupulous competitor is willing to hire such a person to hurt you.)

### Destructive Behavior

Among the destructive sorts of break-ins and attacks, there are two major categories.

### **Data Diddling.**

The data diddler is likely the worst sort, since the fact of a break-in might not be immediately obvious. Perhaps he's toying with the numbers in your spreadsheets, or changing the dates in your projections and plans. Maybe he's changing the account numbers for the auto-deposit of certain paychecks. In any case, rare is the case when you'll come in to work one day, and simply know that something is wrong. An accounting procedure might turn up a discrepancy in the books three or four months after the fact. Trying to track the problem down will certainly be difficult, and once *that* problem is discovered, how can any of your numbers from that time period be trusted? How far back do you have to go before you think that your data is safe?

### **Data Destruction.**

Some of those perpetrators are simply twisted jerks who like to delete things. In these cases, the impact on your computing capability -- and consequently your business -- can be nothing less than if a fire or other disaster caused your computing equipment to be completely destroyed.

### **Where Do They Come From?**

How, though, does an attacker gain access to your equipment? *Through any connection that you have to the outside world.* This includes Internet connections, dial-up modems, and even physical access. (How do you know that one of the temps that you've brought in to help with the data entry isn't really a system cracker looking for passwords, data phone numbers, vulnerabilities and anything else that can get him access to your equipment?)

In order to be able to adequately address security, all possible avenues of entry must be identified and evaluated. The security of that entry point must be consistent with your stated policy on acceptable risk levels.

### **Lessons Learned**

From looking at the sorts of attacks that are common, we can divine a relatively short list of high-level practices that can help prevent security disasters, and to help control the damage in the event that preventative measures were unsuccessful in warding off an attack.

#### **Hope you have backups**

This isn't just a good idea from a security point of view. Operational requirements should dictate the backup policy, and this should be closely coordinated with a disaster recovery plan, such that if an airplane crashes into your building one night, you'll be able to carry on your business from another location. Similarly, these can be useful in recovering your data in the event of an electronic disaster: a hardware failure, or a break-in that changes or otherwise damages your data.

#### **Don't put data where it doesn't need to be**

Although this *should* go without saying, this doesn't occur to lots of folks. As a result, information that doesn't need to be accessible from the outside world sometimes is, and this can needlessly increase the severity of a break-in dramatically.

### **Avoid systems with single points of failure**

Any security system that can be broken by breaking through any one component isn't really very strong. In security, a degree of redundancy is good, and can help you protect your organization from a minor security breach becoming a catastrophe.

### **Stay current with relevant operating system patches**

Be sure that someone who knows what you've got is watching the vendors' security advisories. Exploiting old bugs is still one of the most common (and most effective!) means of breaking into systems.

### **Watch for relevant security advisories**

In addition to watching what the vendors are saying, keep a close watch on groups like [CERT](#) and [CIAC](#). Make sure that at least one person (preferably more) is subscribed to these mailing lists

### **Have someone on staff be familiar with security practices**

Having at least one person who is charged with keeping abreast of security developments is a good idea. This need not be a technical wizard, but could be someone who is simply able to read advisories issued by various incident response teams, and keep track of various problems that arise. Such a person would then be a wise one to consult with on security related issues, as he'll be the one who knows if web server software version such-and-such has any known problems, etc.

This person should also know the ``dos" and ``don'ts" of security, from reading such things as the ``Site Security Handbook."[\[5\]](#)

## **Firewalls**

As we've seen in our discussion of the Internet and similar networks, connecting an organization to the Internet provides a two-way flow of traffic. This is clearly undesirable in many organizations, as proprietary information is often displayed freely within a corporate *intranet* (that is, a TCP/IP network, modeled after the Internet that only works within the organization).

In order to provide some level of separation between an organization's intranet and the Internet, *firewalls* have been employed. A firewall is simply a group of components that collectively form a barrier between two networks.

A number of terms specific to firewalls and networking are going to be used throughout this section, so let's introduce them all together.

### **Bastion host.**

A general-purpose computer used to control access between the internal (private) network (intranet) and the Internet (or any other untrusted network). Typically, these are hosts running a flavor of the Unix operating system that has been customized in order to reduce its functionality to only what is necessary in order to support its functions. Many of the general-purpose features have been turned off, and in many cases, completely removed, in order to improve the security of the machine.

### **Router.**

A special purpose computer for connecting networks together. Routers also handle certain functions, such as *routing* , or managing the traffic on the networks they connect.

#### **Access Control List (ACL).**

Many routers now have the ability to selectively perform their duties, based on a number of facts about a packet that comes to it. This includes things like origination address, destination address, destination service port, and so on. These can be employed to limit the sorts of packets that are allowed to come in and go out of a given network.

#### **Demilitarized Zone (DMZ).**

The DMZ is a critical part of a firewall: it is a network that is neither part of the untrusted network, nor part of the trusted network. But, this is a network that connects the untrusted to the trusted. The importance of a DMZ is tremendous: someone who breaks into your network from the Internet should have to get through several layers in order to successfully do so. Those layers are provided by various components within the DMZ.

#### **Proxy.**

This is the process of having one host act in behalf of another. A host that has the ability to fetch documents from the Internet might be configured as a *proxy server* , and host on the intranet might be configured to be *proxy clients* . In this situation, when a host on the intranet wishes to fetch the `<http://www.interhack.net/>` web page, for example, the browser will make a connection to the proxy server, and request the given URL. The proxy server will fetch the document, and return the result to the client. In this way, all hosts on the intranet are able to access resources on the Internet without having the ability to direct talk to the Internet.

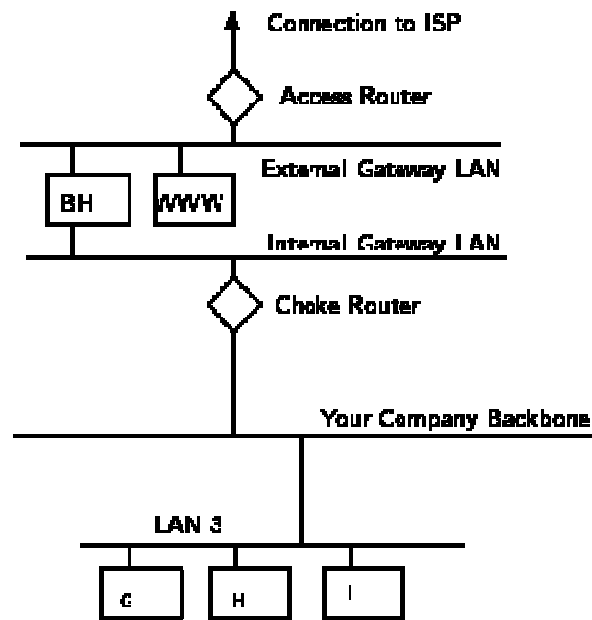
### **Types of Firewalls**

There are three basic types of firewalls, and we'll consider each of them.

#### **Application Gateways**

The first firewalls were application gateways, and are sometimes known as proxy gateways. These are made up of bastion hosts that run special software to act as a proxy server. This software runs at the *Application Layer* of our old friend the ISO/OSI Reference Model, hence the name. Clients behind the firewall must be *proxitized* (that is, must know how to use the proxy, and be configured to do so) in order to use Internet services. Traditionally, these have been the most secure, because they don't allow anything to pass by default, but need to have the programs written and turned on in order to begin passing traffic.

**Figure 5:** A sample application gateway



These are also typically the slowest, because more processes need to be started in order to have a request serviced. Figure 5 shows a application gateway.

### Packet Filtering

Packet filtering is a technique whereby routers have *ACLs* (Access Control Lists) turned on. By default, a router will pass all traffic sent it, and will do so without any sort of restrictions. Employing ACLs is a method for enforcing your security policy with regard to what sorts of access you allow the outside world to have to your internal network, and vice versa.

There is less overhead in packet filtering than with an application gateway, because the feature of access control is performed at a lower ISO/OSI layer (typically, the transport or session layer). Due to the lower overhead and the fact that packet filtering is done with routers, which are specialized computers optimized for tasks related to networking, a packet filtering gateway is often much faster than its application layer cousins. Figure 6 shows a packet filtering gateway.

Because we're working at a lower level, supporting new applications either comes automatically, or is a simple matter of allowing a specific packet type to pass through the gateway. (Not that the *possibility* of something automatically makes it a good idea; opening things up this way might very well compromise your level of security below what your policy allows.)

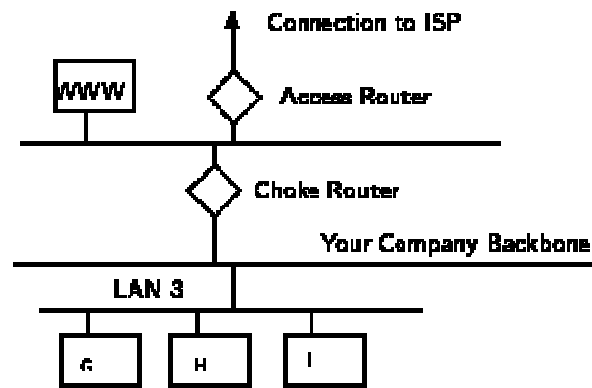
There are problems with this method, though. Remember, TCP/IP has absolutely no means of guaranteeing that the source address is really what it claims to be. As a result, we have to use layers of packet filters in order to localize the traffic. We can't get all the way down to the actual host, but with two layers of packet filters, we can differentiate between a packet that came from the Internet and one that came from our internal network. We can identify which network the packet came from with certainty, but we can't get more specific than that.

### Hybrid Systems

In an attempt to marry the security of the application layer gateways with the flexibility and speed of packet filtering, some vendors have created systems that use the principles of both.



**Figure 6:** A sample packet filtering gateway



In some of these systems, new connections must be authenticated and approved at the application layer. Once this has been done, the remainder of the connection is passed down to the session layer, where packet filters watch the connection to ensure that only packets that are part of an ongoing (already authenticated and approved) conversation are being passed.

Other possibilities include using both packet filtering and application layer proxies. The benefits here include providing a measure of protection against your machines that provide services to the Internet (such as a public web server), as well as provide the security of an application layer gateway to the internal network. Additionally, using this method, an attacker, in order to get to services on the internal network, will have to break through the access router, the bastion host, and the choke router.

### So, what's best for me?

Lots of options are available, and it makes sense to spend some time with an expert, either in-house, or an experienced consultant who can take the time to understand your organization's security policy, and can design and build a firewall architecture that best implements that policy. Other issues like services required, convenience, and scalability might factor in to the final design.

### Some Words of Caution

The business of building firewalls is in the process of becoming a commodity market. Along with commodity markets come lots of folks who are looking for a way to make a buck without necessarily knowing what they're doing. Additionally, vendors compete with each other to try and claim the greatest security, the easiest to administer, and the least visible to end users. In order to try to quantify the potential security of firewalls, some organizations have taken to firewall certifications. The certification of a firewall means nothing more than the fact that it *can* be configured in such a way that it can pass a series of tests. Similarly, claims about meeting or exceeding U.S. Department of Defense "Orange Book" standards, C-2, B-1, and such all simply mean that an organization was able to configure a machine to pass a series of tests. This doesn't mean that it was loaded with the vendor's software at the time, or that the machine was even usable. In fact, one vendor has been claiming their operating system is "C-2 Certified" didn't make mention of the fact that their operating system only passed the C-2 tests without being connected to any sort of network devices.

Such gauges as market share, certification, and the like are no guarantees of security or quality. Taking a little bit of time to talk to some knowledgeable folks can go a long way in providing you a comfortable level of security between your private network and the big, bad Internet.

Additionally, it's important to note that many consultants these days have become much less the advocate of their clients, and more of an extension of the vendor. Ask any consultants you talk to about their vendor affiliations, certifications, and whatnot. Ask what difference it makes to them whether you choose one product over another, and vice versa. And then ask yourself if a consultant who is certified in technology XYZ is going to provide you with competing technology ABC, even if ABC best fits your needs.

### Single Points of Failure

Many ``firewalls" are sold as a single component: a bastion host, or some other black box that you plug your networks into and get a warm-fuzzy, feeling safe and secure. *The term ``firewall" refers to a number of components that collectively provide the security of the system.* Any time there is only one component paying attention to what's going on between the internal and external networks, an attacker has only one thing to break (or fool!) in order to gain complete access to your internal networks.

See the [Internet Firewalls FAQ](#) for more details on building and maintaining firewalls.

### Secure Network Devices

It's important to remember that the firewall only one entry point to your network. Modems, if you allow them to answer incoming calls, can provide an easy means for an attacker to sneak *around* (rather than *through* ) your front door (or, firewall). Just as castles weren't built with moats only in the front, your network needs to be protected at all of its entry points.

### Secure Modems; Dial-Back Systems

If modem access is to be provided, this should be guarded carefully. The *terminal server* , or network device that provides dial-up access to your network needs to be actively administered, and its logs need to be examined for strange behavior. Its password need to be strong -- not ones that can be guessed. Accounts that aren't actively used should be disabled. In short, it's the easiest way to get into your network from remote: guard it carefully.

There are some remote access systems that have the feature of a two-part procedure to establish a connection. The first part is the remote user dialing into the system, and providing the correct userid and password. The system will then drop the connection, and call the authenticated user back at a known telephone number. Once the remote user's system answers that call, the connection is established, and the user is on the network. This works well for folks working at home, but can be problematic for users wishing to dial in from hotel rooms and such when on business trips.

Other possibilities include one-time password schemes, where the user enters his userid, and is presented with a ``challenge," a string of between six and eight numbers. He types this challenge into a small device that he carries with him that looks like a calculator. He then presses enter, and a ``response" is displayed on the LCD screen. The user types the response, and if all is correct, he login will proceed. These are useful devices for solving the problem of good passwords, without requiring dial-back access. However, these have their own problems, as they require the user to carry them, and they must be tracked, much like building and office keys.

No doubt many other schemes exist. Take a look at your options, and find out how what the vendors have to offer will help you *enforce your security policy effectively*.

### Crypto-Capable Routers

A feature that is being built into some routers is the ability to session encryption between specified routers. Because traffic traveling across the Internet can be seen by people in the middle who have the resources (and time) to snoop around, these are advantageous for providing connectivity between two sites, such that there can be secure routes.

See the Snake Oil FAQ [\[6\]](#) for a description of cryptography, ideas for evaluating cryptographic products, and how to determine which will most likely meet your needs.

### Virtual Private Networks

Given the ubiquity of the Internet, and the considerable expense in private leased lines, many organizations have been building *VPNs* (Virtual Private Networks). Traditionally, for an organization to provide connectivity between a main office and a satellite one, an expensive data line had to be leased in order to provide direct connectivity between the two offices. Now, a solution that is often more economical is to provide both offices connectivity to the Internet. Then, using the Internet as the medium, the two offices can communicate.

The danger in doing this, of course, is that there is no privacy on this channel, and it's difficult to provide the other office access to ``internal" resources without providing those resources to everyone on the Internet.

VPNs provide the ability for two offices to communicate with each other in such a way that it looks like they're directly connected over a private leased line. The session between them, although going over the Internet, is private (because the link is encrypted), and the link is convenient, because each can see each others' internal resources without showing them off to the entire world.

A number of firewall vendors are including the ability to build VPNs in their offerings, either directly with their base product, or as an add-on. If you have need to connect several offices together, this might very well be the best way to do it.

### Conclusions

Security is a very difficult topic. Everyone has a different idea of what ``security" is, and what levels of risk are acceptable. The key for building a secure network is to *define what security means to your organization* . Once that has been defined, everything that goes on with the network can be evaluated with respect to that policy. Projects and systems can then be broken down into their components, and it becomes much simpler to decide whether what is proposed will conflict with your security policies and practices.

Many people pay great amounts of lip service to security, but do not want to be bothered with it when it gets in their way. It's important to build systems and networks in such a way that the user is not constantly reminded of the security system around him. Users who find security policies and systems too restrictive will find ways around them. It's important to get their feedback to understand what can be improved, and it's important to let them know *why* what's been done has been, the sorts of risks that are deemed unacceptable, and what has been done to minimize the organization's exposure to them.

