INDEX

CHAPTER 1

IBM PC CONFIGURATION

Introduction

The digital computer is an electronics machine. Its main capability is high-speed calculation. To solve a problem, we prepare and run a program as shown in fig 1.1. A computer program is a sequence of instructions. Each instruction specifies an operation to be performed. The computer interprets each instruction and executes the specified operation.

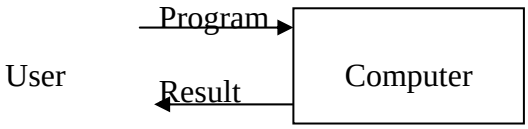


Fig 1.1 Using a Computer

The basic structure of a computer is shown in fig 1.2. The functional blocks in a computer are of five types:

- Arithmetic logic unit
- Control Unit
- Memory
- Input unit
- Output Unit

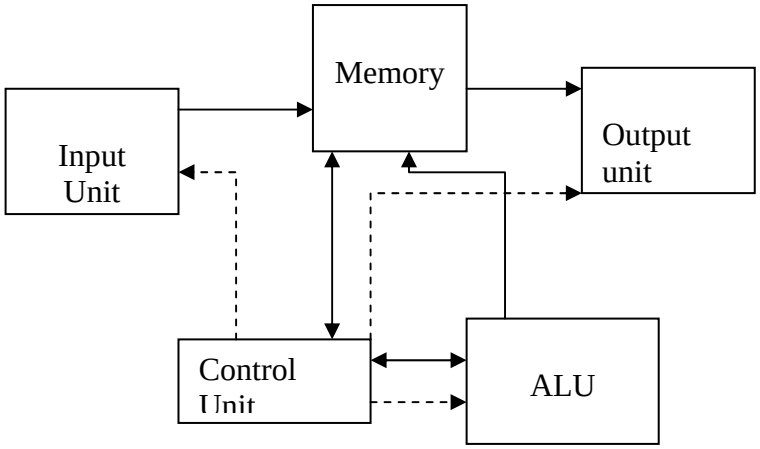


Fig 1.2 Basic Computer Structure

ASSIGNMENT

Q1 What are the main components of a computer, explain in brief?

CHAPTER 2

CENTRAL PROCESSING UNIT

Central Processing Unit

The ALU contains electronic circuits necessary to perform arithmetic and logical operations. The arithmetic operations are ADD, SUBTRACT, MULTIPLY, DIVIDE etc. The logical operations include COMPARE, SHIFT, ROTATE, AND, OR etc. The control unit analyses each instruction in the program and sends the relevant control signals to all other units - ALU, Memory, Input Unit and Output Unit. Figure 2.1 shows the internal communication inside a computer. A computer program consists of both instructions and data. The program is fed into the computer through the input unit and stored in the memory. In order to execute the program, the instructions have to be fetched from memory one by one. The control unit does this fetching of instructions. After an instruction is fetched, the control unit decodes the instruction. According to the instruction, the control unit issues control signals to other units. After an instruction is executed, the result of the instruction is stored in memory or stored temporarily in the control unit or ALU, so that this can be used by the next instruction. The results of a program are taken out of the computer through the output unit. The control unit and ALU are collectively known as Central Processing Unit (CPU).

CPU performance determines, in part, computer performance. CPUs vary in several ways that affect their performance. These variations are discussed in the subsequent paras.

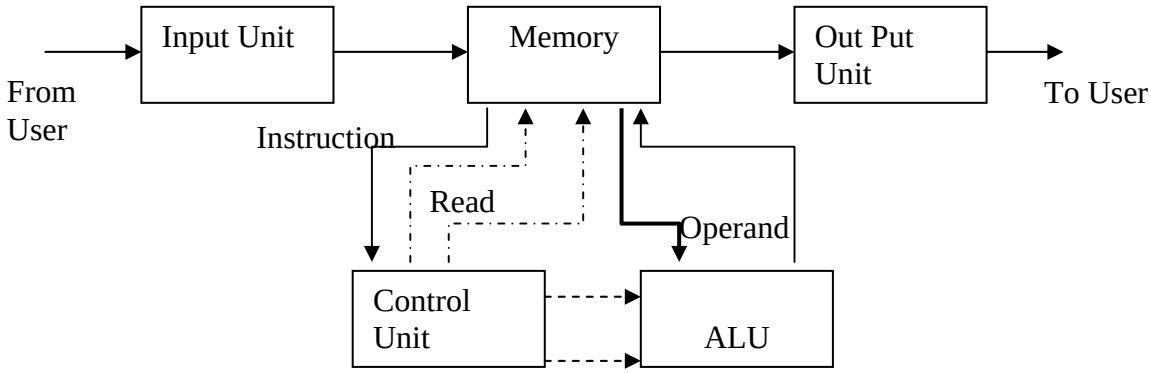


Fig 2.1 Communication inside a computer

CPU Speeds (Megahertz). Computers are a little like clockwork devices. A clock strikes a beat, and a certain small amount of work gets done. Just like a beginning piano player plays to the beat of a metronome, computers run to the beat of a clock (although it's an electronic clock). If you set the metronome too fast for the beginning piano player, he/she will become confused and the music won't come out right; the player won't have enough time to find the next piano

key, and the rendition will probably fall apart. Similarly, if you set the clock rate of a CPU too high, it will malfunction, but the result is a system crash.

Normally, running a CPU too fast won't damage the chip: the computer just won't function properly. Part of the design of a computer like the PC includes determining a clock rate.

CPU clocks generally "tick" more than a million times per second. A clock, which ticks at exactly one million times per second, is said to be a one megahertz clock (I MHz). The early PCs and XTs used a 4.77 MHz clock. IBM followed the PC with the AT, whose original model used a 6 MHz clock, and later IBM offered a version, which ran at 8 MHz.

CPU manufacturers strive to drive up the clock rate. As a result, the megahertz value of a computer is as important a measure of its speed as horsepower would be an approximate measure of a car's power. All other things being equal, a faster clock means faster execution and better performance.

The CPU is only part of what makes a computer fast. A really fast CPU paired with an amazingly slow hard disk would turn in a mediocre performance. The Pentium's hard disk and graphic boards were about 10 times faster than the XT's, but the CPU was almost two hundred times faster. A Pentium seems quick compared to an XT, of course, but not anywhere near 200 times faster, mostly because of the slower peripherals. If a computer were about 20 times faster than an XT in its disk, graphics board, and CPU, then it would probably feel faster than the Pentium system.

Nowadays, the slowest computer you are likely to come across will be 33 or 66 MHz. The fastest speeds you hear may get up around 800 MHz.

Word Size. Any computer can be programmed to manipulate any size number, but bigger the number, the longer it takes. The largest number that the computer can manipulate in one operation is determined by its word size. This is either 8, 16, or 32 bits. Think of it this way, if someone ask, "What is 5 times 6?" you answer "30," immediately-you did it in one Operation. "What is 55 times 66?" one will do a series of steps to arrive at the answer. 55 is larger than Your word size. That's one reason why a 386 with 32-bit registers, is faster than a 286, with 16-bit registers.

Data Path. No matter how large the computer's word size, the data must be transported into the CPU. This is the width of the computer's "loading door." it can be 8, 16, or 32 bits. A wider door will allow more data to be transported in less time than will a narrower door. An 8 MHz 8088 versus an 8 MHz 8086. The only difference between the 8088 and the 8086 is that the 8088 have an 8-bit data path, the 8086 a 16-bit data path. Now, both the 8088 and the 8086 have 16-bit registers, so a programmer would issue the same command to load 16 bits into either one the command **MOV AX, 0200** will move the 16-bit value 200 hex into a 16 bit register called AX. That will take twice as long on the 8088 as the 8086 because the 8086 can do it one operation while the 8088 takes two.

Although they're both 8 MHz Computers the 8088 machine computes more slowly for some operations.

The original 80386 was introduced in 1985 with a 32-bit data path. The 286 was not only a cheap chip, it was also a 16-bit chip, and a pure 16-bit chip, both word size and data path. That meant that 286 PCs could be built around 16-bit motherboards. The 80386SX was an 80386 with one difference: it was 16 bits, not 32 bits. Vendors liked that because they could use old 16-bit 80286 motherboards, modify the design a bit, and offer "386 technology." The original 80386 needed a name, so it became known as the 80386DX.

The 80486 line of chips also includes an 80486SX. It is a chip with reduced functionality, but the reduction is not due to changes in data path. Instead, the 486SX is different from the 486DX in the way that it calculates arithmetic.

The Pentium line of chips also includes an SX-like chip. The Pentium uses a 64-bit data path, but 486 manufacturers wanted to include Pentium compatibility on their motherboards. However, 486 motherboards are 32-bit in nature. To offer Pentium compatibility Intel's Pentium chip, which features a 32-bit data path, came in.

Internal Cache Memory. When we talk of RAM on a computer, we're talking about chips that the CPU uses to store its programs and data as it works, chips external to the CPU. But the increasing speed of CPUs has driven a corresponding need for faster RAM.

RAM is commonly designed to be dynamic RAM, a simpler and cheaper design than its alternative, static RAM. Static RAM, or SRAM, is considerably more expensive and can be made much faster than can DRAM.

PCs use a lot of dynamic RAM, which unfortunately sacrifices speed. To get back some of that speed, Intel puts a small amount of fast static RAM right into the CPU. Often used data need not be accessed via the relatively slow DRAM; instead, the CPU can keep the most important data in this small "cache" of storage. That is why it is called cache RAM.

The 80486 line of chips were the first in the x86 family to include cache RAM; with the exception of the DX4, they all contain 8K of internal cache. The DX4 doubled that amount to 16K. But even that small amount can significantly effect CPU performance. Many motherboard have from 64K to 512K more static RAM cache. It is external cache. Sometimes the internal cache is called the LI cache, and the external is called the L2 cache.

The Pentium's cache system is better than the 486s in four ways.

- The Pentium has twice as much cache, with two 8K caches, one for data, one for program code.
- The cache's method of organising its cached data is more efficient, employing a "write-back" algorithm. The older "write-through" algorithm, forces data written to the SRAM cache memory to be immediately written to the slower DRAM memory. That means that memory reads can come out of the cache quickly, but memory

- writes must always occur at the slower DRAM time. Reasoning that not every piece of information written to memory stays in memory very long. The Pentium's cache algorithm puts off writing data from SRAM to DRAM for as long as possible, unlike the 486, which uses a write-through cache.

- The cache controller wastes time in searching to see if an item is in the cache-the Pentium reduces that time by dividing the cache into smaller caches, each of which can be searched more quickly; that technique is called a two-way set associative cache.

- A cache must guess what data and program code the CPU will need soon, and then get that data before the CPU asks for it. But guessing what the CPU will need is not a straightforward task, particularly when there are decisions to be made. For example, suppose the cache sees that the CPU is currently executing some instructions that mean, "Com- pare value A with value B. If A is greater than B, then set the value maximum to A; otherwise, set the value maximum to B." That simple statement boils down to a bunch of instructions, instructions in memory that should be in the cache if the Pentium is to be able to continue running without delays. But since the cache controller can not know whether "A was greater than B" or "B was greater than A" fork in the road, then it doesn't know which results code grab and put in the cache. Pentium-has cache controller built into it with branch prediction capabilities, So that Pentium makes better use of your memory than the 486 did.

Numeric Coprocessors. Look at an early PC or XT and next to the 8088 is an empty, socket. It was for the Intel 8087, a special purpose microprocessor. The 8087 is a microprocessor that is only good for one class of tasks, floating point numeric operations. Intel markets a wide line of coprocessor chips, as seen in Table 6.1.

TABLE 6.1 Intel Coprocessor Chips

CHIP	MAX SPEED (MHZ)	PACKAGE TYPE	TYPICAL CPU
8087	10	DIP	8088/8086
80287	12	DIP	80286
80387DX	33	PGA	80386DX
80387SX	33	PLCC	80386SX
80486DX	66	PGA	80486
80487SX	33-	PGA	80486SX
Pentium	150	PGA	Pentium

In general, the 8088 and 8086 go with the 8087, the 80286 goes with the 80287, and the 80386 goes with the 80387.

Memory Addressable by a CPU. Megabytes are a unit of storage size, about the amount of space needed to store a million characters. When we say "memory," we are talking about primary memory, so memory means chips or RAM. We usually say "disk" when we refer

to disks rather than secondary memory. Disk is not volatile, which means that when you shut it off it retains its data. Remove power from a memory chip and whatever it contained is lost.

A particular chip can only address a certain size of memory. For the oldest chips, this amount was 65,536 bytes-a 64K memory. The original PC's CPU can address 1024K, or one megabyte (MB). The 80386, 80486, and Pentium can address 4 gigabytes or 4GB of memory. The amount of memory a processor can address depends on the address lines it has. A 8086 had 20 address lines hence it could support 2^{20} = 1MB, Pentium has 32 address lines hence it can address 2^{32} =4GB of memory.

ASSIGNMENT

Q1 What is the requirement of a clock in a CPU? Should the clock speed of a CPU be same as that of the other components in a computer explain.

Q2 What is the difference between word size and data path? Is it necessary that these two should be same, explain?

Q3 Explain the role of cache memory in a CPU.

Q4 What was the role of a numeric co-processor in early PCs? Why is it that it is not found in the modern Pentium based machines?

Q5 Write briefly on the advantages of having registers in a CPU.

Q6 Classify the following interrupts into hardware and software interrupts:

(a) Interrupt from printer controller.

(b) NMI to memory parity error.

(c) Interrupt from Keyboard controller.

(d) Illegal Opcode.

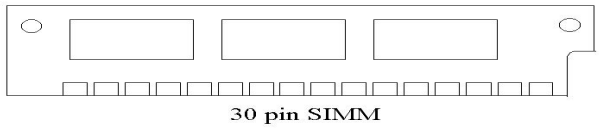
Q7 What is the first activity by CPU on recognising an interrupt?

CHAPTER 3

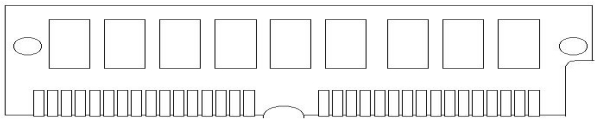
COMPUTER MEMORY

Introduction

The PC, like all computers, must have main memory. Main memory is high-speed memory that the CPU can read from or write to. "High speed" here means less than a microsecond to read/write. The other name for such memory is Random Access Memory or the RAM, a particular kind of chip on circuit boards. Memory is easy to pick out on a circuit board. It's packaged either as a "bank" of eight or nine small chips, or it's a mini-circuit board with several square chips mounted on it, called a SIMM-Single In line Memory Module. Memory is always organised into banks-either eight or nine discrete chips, or a SIMM. Most motherboards have room for four banks of memories. Some of the newer machines have no memory on the motherboard at all, but instead have a large circuit board with room for Megs and Megs of memory. As each SIMM is the equivalent of nine chips, SIMMs make replacing bad memory easier, but make repair options less flexible. Changing one chip is a lot cheaper than changing nine. You can see a SIMM in Figure 3.1.



30 pin SIMM



72 pin SIMM

Fig 3.1 SIMM

The familiar first 640K of a PC's memory is called conventional memory. It is supplemented by reserved areas containing ROM, Read Only Memory. The 286 and later machines can address memory beyond that called extended memory. And a small-but-important number of applications can use special memory called expanded memory or, as it is known as LIM memory.

Types of Memory

There are many kinds of memory these day, the three most common areas are:

- Conventional
- Extended-memory that Windows likes best
- Expanded, also called EMS (Expanded Memory Specification)or LIM (Lotus -Intel-Microsoft).

PCs tend to have a lot more disk space than they've got RAM space. As mentioned before, RAM is a lot faster than disk.

Designing a Computer's Memory: Zoning the First Megabyte

Many of the constraints that we face today have their roots in the past, a past 15 years ago.

In 1980, IBM commenced the PC development project. The goal of the small design team was to build a "home Computer". The chip that IBM selected was the Intel 8088, and one of the powerful features was that the 8088 could address up to 1024K one megabyte of RAM. The sad fact is that even Windows is constrained in some ways by the 8088's memory structure, which is why you've got to understand it.

So it is with PC design. Before any memory is placed in the system, the computer planner knows that the CPU can address a certain amount of memory-1024K, in the case of the 8088 CPU-and that groups of addresses must be set aside for particular functions. The actual memory chips that end up getting hooked up to the computer is basically identical, whether they are conventional, extended, expanded, or any other kind. What's important is the address of that memory.

In the early days of the PC's design, the entire universe consisted of just 1024K of addresses. You can see the rough overview of those addresses in Figure 3.2, called a memory map. A memory map is a common tool for diagramming memory addresses and the uses of memory put in those addresses.

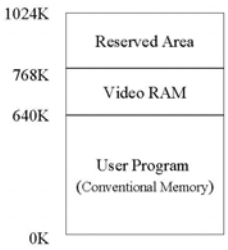


Fig 3.2 Memory Map upto 1024k

Conventional Memory

PCs can use 640K of main memory when running DOS and DOS programs. This first 640K of memory addresses are called conventional or, sometimes, user memory addresses. Any memory wired to those addresses serves, then, to store user programs, although other programs draw from those addresses as well. DOS loads in that memory, as do some "helper" pro- grams called TSRs and device drivers. The BIOS programs that form the lowest level of software in much of the PC world also keep a little data in the low 640K. Figure 3.3 is a hypothetical map of how my system might use memory when working with a simple old DOS applications like Lotus 1-2-3.

Interrupt Vectors and DOS. The bottom IK is an area used by the CPU as a kind of "table of contents" of hardware support programs called software interrupts; the table of contents is composed of pointers to those programs called interrupt vectors. This is a fixed size of 1024 byte. One interrupt points to

the program that controls your disk drives, another to the video board, and so on. Above that is DOS itself. It is hard to say exactly how much space DOS takes up in memory, as that varies by the version of DOS and what CONFIG.SYS options you choose.

Fig 3.3 Memory Mapping using DOS Application

Device Drivers. Directly above DOS load a special class of programs called device drivers. Device drivers are programs that either allow DOS to support a new piece of hardware, or add new capabilities to an existing piece of hardware. Device drivers are loaded in the CONFIG.SYS file with the DEVICE= statements. Some examples of device drivers include:

- The ANSI.SYS device driver that extends the ability of your system to respond to a new set of video commands.
- A mouse driver, like MOUSE.SYS, that enables your system to recognise and control your mouse.
- The memory manager drivers that many people use, HIMEM.SYS and EMM386.EXE, which make memory above 1024K into a useful resource for software.

Sometimes the order in which you load device drivers is important. In any case, device drivers change the way that your system reacts to information from your system's hardware.

Command Shell. Every operating system has a program that accepts inputs from users and reformulates them in a manner that the operating system can understand. In the case of DOS, the most common command shell is COMMAND.COM. It loads after the device drivers, but before TSRS.

TSR or Memory Resident Programs. TSR (terminate and stay resident) programs do much the same thing as device drivers, but they are loaded from AUTOEXEC.BAT, and so load after any device drivers. Here are a few examples of TSR programs,

- DOS comes with a small utility called DOSKEY that remembers the last twenty-or-so commands that one has typed so that if one makes a mistake he can recall a previous command, edit it, and resubmit it to the PC. Many virus and anti-virus programs are TSRS.

- Network shells and protocol stacks are usually TSRS.
- Many disk cache programs that speed up disk access are TSRS.
- Disk compression programs, like Stacker or DOS's DriveSpace, are memory resident either via device driver or TSR format.

TSRs and device drivers are useful, but they take up precious 640K conventional space.

User Programs. Above the TSRs you find the currently loaded program, example 1-2-3. The remaining memory is then available for 1-2-3 worksheets. Again, the top address is 640K. The vast majority of DOS programs can only run in the low 640K of your PC's memory. That's why the 640K conventional memory area is so important. if the program can only live in conventional memory and the conventional memory is full, getting more memory won't help.

Video RAM. DOS programs are almost all written to run on the 8088 chip and, its successors, the 286 and later. Any 8088 program can address up to 1024K of memory in theory. 640K is a very real barrier for most systems.

As used in the IBM world, the video board-the circuit board that acts as an interface between the CPU and the monitor-must have some memory on it. That memory is then shared between the circuitry on the video board and the CPU. The CPU "puts data on the screen" by putting data into this video RAM. The video circuitry sees the data in the memory and interprets it as graphical or textual information.

Video memory is memory used by video boards to keep track of what's to be displayed on the screen. When a program puts a character on the screen, or draws a circle on the screen, it is actually making changes to this video memory. IBM set aside 128K for video memory, but most video boards don't actually need or use that much memory space. Video RAM must go somewhere, and the original PC designers placed it from 640K through 768K. Even if there were more memory for user programs above 768K most programs could not use that memory, as most programs insist on contiguous blocks of memory.

A large number of programs are designed to directly manipulate the video hardware, and those programs are all written assuming that the video is where it's supposed to be-between 640K and 768K.

Video boards use the addresses from 640K through 768K, but how do they use these addresses? Table 5.1 shows the common video boards and their memory capacities.

The System Reserved Area. In addition to the DOS, device drivers, TSRS, user programs, and video, the PC needs to steal from the CPU's memory address space for the following:

- Small amounts of memory called buffers or frames used by some expansion boards
- Special memory containing system software called ROMs (Read Only Memory)

ROM (Read Only Memory). Another kind of memory exists which is not used as much as RAM, but is important. Unlike RAM, which the CPU can both write data to and read data from, this other kind of memory cannot be altered. It can only be read, so it is called Read Only Memory. This is memory that computer manufacturer loads just once with a special device called a PROM Blaster, EPROM programmer, or the like. You can read information from ROMS, but you can't write new information.

We use ROM to store software that won't change. In essence, we can say that ROM on a circuit board contains the software that tells the system how to use a circuit board.

ROMs are found on expansion boards like LAN, video, or scanner inter- face cards, to name a few examples. It is also found on the system board. The ROM on the system board contains a piece of software called BIOS, the Basic Input/Output Systems. BIOS is a set of low-level programs that directly manipulate your hardware. Those programs are called "software interrupts," and are pointed to by the interrupt vectors at the bottom of RAM. DOS relies on BIOS in that DOS doesn't communicate directly with your hardware, but rather issues commands through BIOS. Thus, when DOS reads your floppy, it does it by calling on the BIOS routine that reads the floppy drive. That's why the BIOS is so important: the BIOS determines in large measure how compatible your PC is.

ROMs can usually be identified because they are generally larger chips, 24 or 28-pin DIP chips, they are socketed (so they can be easily changed), and often have a paper label pasted on them with a version number or some such printed on it. ROMs are memories, albeit inflexible ones, and so require a place in the memory addresses in the reserved area from 640K to 1024N.

Extended Memory

From the 80286's introduction in 1981 onward, Intel chips could address megabytes and megabytes. An 80286 can address 16MB. An 80386 or 80486 can address 4GB of RAM. The term for normal RAM above the 1MB level is extended memory.
Why does memory above 1024K get a completely new name? Largely because the 286 and later chips have "split personalities": they can either address memory beyond 1024K, or run DOS programs. In order to use memory above 1024K, the 286 and later chips must shift to a new processor mode called protected mode. Protected mode has lots of virtues, but one big flaw: when a chip is in protected mode, it's incompatible with an older 8088 or 8086.

Expanded Memory or LIM memory

LIM stands for Lotus, Intel and Microsoft, The LIM memory isn't viewed by, the system as memory. All the PC knows is that there are "pages" of storage available 16K-sized pages. LIM can support up to 2000 of these pages, hence the 32MB maximum size. LIM boards allocate 64K enough space for four

pages-of memory somewhere in the reserved area between 640K and 1024K, so a program can manipulate up to four pages

at a time. LIM is manipulated, then, by pulling in a page from LIM memory to the memory in the reserved area. This memory is called a page frame, and moving data to and from LIM and page frames is called paging, reading and/or modifying the page frame, and possibly writing the page frame back to the LIM memory.

ASSIGNMENT

Q1 If the number of bits in Memory address register is 20, what is the maximum memory capacity?

Q2 Explain briefly how the first MB of memory is logically divided and the role of each division.

Q3 What is the difference between extended memory and expanded memory?

Q4 What are interrupt vectors and how are these different from TSR programs, explain?

CHAPTER 4

EXPANSION BUSES

What Is a Bus?

In order to be useful, the CPU must talk to memory, expansion boards, keyboard and the like. It communicates with other devices on the motherboard via metal traces in the printed circuit, the silver lines that you see running around a board. That is how SIMMs that probably live on your computer's motherboard communicate with the CPU which is also on your computer's motherboard-they talk back and forth by shooting electrons along these thin metal traces.

But how can expansion boards, which aren't part of the motherboard, be connected to the CPU, the memory and so on? Through the bus.

Back in the early days of microcomputers, some computers didn't allow easy expansion. Take, for example, the early Macintosh computers. To expand a 128K or 512K Mac, you had to do some extensive engineering which is why most people didn't mess around inside those computers. Any circuit boards that you wanted to add usually had to be mounted haphazardly inside the Mac's case; installing a hard disk actually involved disassembling the computer, soldering connections onto the Mac motherboard, and reassembling the machine. Making modifications difficult puts the user at the mercy of the modifier, as virtually all such modifications are done at the expense of the manufacturer's warranty and service agreement, if any-and they're expensive, as they require a technician.

You don't have to do such brain surgery on a PC, thankfully. PCs have expansion slots that allow easy upgrade. (By the way, today's Macs also have expansion slots, fortunately; in fact, Macs built after 1995 actually have the same bus as some PCs, the PCI bus that we'll discuss soon.)

Another disadvantage of the old Macintosh approach is that the average Joe/Jane can't do the modifications him/herself. This would be like you having to cut a hole in the wall of your house to find a main power line every time you want to use an appliance. Without standard interface connectors (that is, an outlet) you would have to find the power line, then splice the appliance into it to get power for the appliance.

This scenario, as we know, is silly, as we have standard outlet plugs. Any manufacturer who wants to sell me a device requiring electrical power need only ensure that the device takes standard U.S. current, and add a two-prong plug. "Upgrading" my house, then (adding the new appliance) is a simple matter: just "plug and play." Many computers adopt a similar approach. Such computers have published a connector standard any vendor desiring to offer an expansion board for this computer need only follow the connector specifications, and the board will work in the computer. Even the earliest computers included such a connector, first called the "omnibus connector," as it gave access to virtually all important circuits in the computer. "Omnibus" was quickly shortened to ‘bus’ and bus it has remained.

So a bus is a communication standard, an agreement about how to build boards that can work in a standard PC. For various reasons, however, there are over a half-dozen such different standards in the PC world.

The First “PC” Bus

The PC wasn't the first computer based on a chip, not by about eight years. The first commercially available microcomputer was a computer called the Altair. It consisted of a case and a row of expansion slots. It was a back plane computer, with even the CPU on an expansion card. The bus that the Altar used became a standard in the industry for years, and in fact is still used in some machines: it was called the S-100 or Altar bus.

Although it was a standard, it wasn't ever true that every microcomputer used the S-100. The Apple 11 used a bus of its own, called the "Apple bus." The original 1981 PC model used a bus, with 62 lines. It came to be known as the PC bus. The 62 lines mentioned above are offered to the outside world through a standard connector as mentioned previously. These connectors are also called "expansion slots," as expansion boards must plug into these slot. Some PCs have had no slots at all, and so weren't expandable; other machines have three, and most clone type machines have eight slots. Some machines offer IO slots. The more slots, the better: expansion slots equal flexibility and upgradability. Let's take a minute, however, and look at what those 62 lines do.

Data Path. The original PC and XT were based on the 8088 chip. The 8088 had a data path of just 8 bits, so the PC bus only includes eight data lines. That means this bus is "8 bits wide," and so data transfers can 'only occur in 8-bit chunks on this bus. Expansion slots on a computer with this bus are called "8-bit" slots. Eight of the 62 wires then transport data around the PC. Consider the importance of data path in bus design. The 8 data bits supported by the original PC bus would be pretty inadequate for a Pentium based system; Pentium uses a 64-bit data path. Could someone actually build a Pentium computer with 8-bit expansion slots? Sure. But every time that the Pentium wanted to do a full 64-bit read of data, it would have to chop that request up into eight separate 8-bit reads. Really slow. But it could be done, and in fact there are, as you'll learn later, designs almost as bad: a fair number of Pentium systems in corporate a 16-bit bus.

Memory Size. The original PC bus included 20 wires to address memory. Each one of those address wires can either carry a 0 or a 1 signal, then each wire can only carry one of two possible values. Since there are 20 of the address lines, the total number of possibilities is 2 to the 20th power, or just over one million, As the 8088 can only address I MB of RAM. All of those address lines are duplicated on the PC bus, accounting for another 20 of the 62 bus lines.

Memory or Address. The 20 address lines actually do double duty, as there are two kinds of addresses: the memory's addresses and input/output addresses. The computer must be able to tell when the address lines are transmitting a memory address versus when the address lines are transmitting an 1/0 address; one line on the bus designates which one it is. Additionally, there are several other lines on the bus that tell whether the data on the bus

has been read from memory (or an 1/0 device), or that data is to be written to memory or 1/0.

Electronic Overhead. Some bus wires just transport simple electric power; there are +5 volts, -5 volts, + 12 volts, and electric ground lines as part of the bus. Why are those lines there? Simple: to power a board plugged into a bus slot. There are also a few control lines, like Reset (which, as you'd imagine, re-sets the processor), clock signals, and Refresh, which controls memory refresh.

Interrupts and Direct Memory Access Channels. Add-on cards sometimes need to demand the attention of the CPU; they do that via hardware interrupts or IRQ (interrupt request) levels. There are six IRQ levels on the PC bus, labelled IRQ2 through IRQ7. Each gets a wire on the bus. There are also IRQO and IRQ1, but they're not available on the bus.

Some of those add-in cards also need to transfer data to the system's memory quickly; they can do that via a Direct Memory Access (DMA) channel. There are three. DMA channels on the bus, labelled DMA1 through DMA3. There is also DMA channel 0, but, like IRQs 0 and 1,it's not accessible through the bus.

The AT (ISA) Bus

The first enhancement to the PC bus came with the IBM AT. When developing the AT, IBM saw that it had to upgrade the bus. One reason was because the 80826 is, a chip with a 16-bit data path. They certainly could have designed the AT with an 8-bit bus, but it would be very slow to make a 286 chip transfer data 8 bits at a time over the bus rather than utilise its full 16-bit data path. On the other hand 'there was backward compatibility with the PC and XT to think of. So IBM came up with a fairly good solution, they kept the old 62-line slot connectors and added another 36-wire connector, placing it in line with the older 62-line connector to provide some of these features:-

- Eight more data lines, bringing the data bus to 16 bits in width.
- Four more address lines, bringing the address bus to 24 bits in width. Two to the 24th power is around 16 million, so the AT's 16-bit slots could support up to 16MB of RAM, in theory.
- Four more DMA channels, 4 through 7.
- Five more IRQ levels: IRQIO, 11, 12,14, and 15.

These two-slot connectors are called, as you'd expect, 16-bit slots. You can see these two kinds of connectors in Figure 4.1.

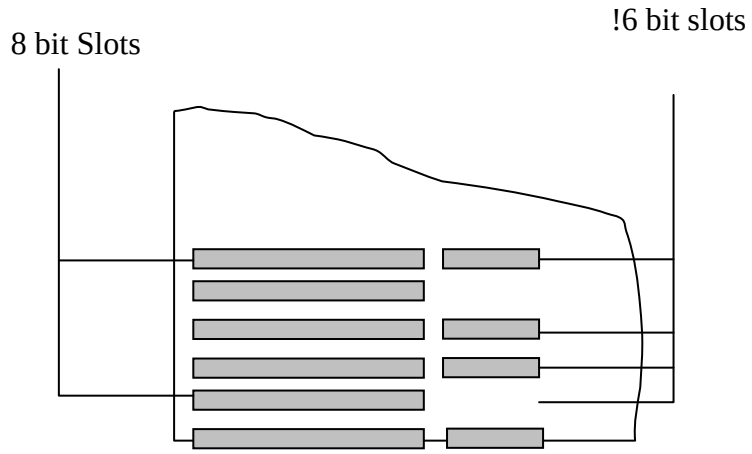


Fig 4.1 8-bit and 16-bit Connector

For a while, this 16-bit bus was called the AT bus. Since 1988, however most people have referred to these types of bus slots as Industry Standard Architecture, or ISA, slots. You can tell the difference between an 8-bit and a 16-bit ISA board by looking at the edge connector on the bottom of it.

As the 16-bit slots are just a superset of the 8-bit bus, 8-bit boards work just fine in 16-bit slots. Many PCs have only these 16-bit slots on their motherboard, you may; however sometimes see a motherboard with both 8-bit slots and 16-bit slots. If the 16-bit slots can use 8-bit boards, why have any 8-bit slots on an AT-type machine at all? The reason is not an electrical reason, but a physical reason. Some older 8-bit boards have a skirt that extends down and back on the circuit board, making it physically impossible to plug an 8-bit board with a skirt into a 16-bit connector.

Bus Speed. Buses have clocks, as well, and those. Clocks drive the boards inserted into their expansion slots. That suggests a question: how is it that any board works in any kind of computer? Why does a video board built for a 12 MHz computer still work in a 100 MHz Pentium? Part of the answer, is that the 100 MHz Pentium looks like a 66 MHz Pentium from the motherboard's point of view; the 100 MHz part is internal to the CPU chip. So the fastest bus you'd need for a 100 MHz Pentium would be 66 MHz. Prior to 1985, buses ran at the same speed as their CPUs. The PC ran at 4.77 MHz, and so did the PC bus. "Turbo" XT clones that ran at 7.16 MHz had buses that ran at 7.16 MHz. When IBM released the 6 MHz AT, then its bus ran at 6 MHz, and so on. A board designed for the 4.77 MHz PC might not work in the faster machine.

Then Compaq released its Deskpro 286/12. If they built a 12 MHz 286 computer with a 12 MHz bus, then in all probability no existing boards would work. Who'd buy a fast computer that wouldn't work with any of the add-in boards on the market?

Micro Channel Architecture (MCA)

IBM in 1987 announced the PS/2 line. In order to facilitate faster data transfer within the computer, and to lower noise levels the PS/2 Models 50 to 80 got a new bus called the Micro Channel Architecture (MCA) bus. MCA bus was completely incompatible with the old ISA bus. ISA expansion boards didn't work in the PS/2 line. MCA never caught strongly but some of the features that it offered have become essential for any advanced bus.

Better Speed and Data Path. MCA tried to better ISA, MCA runs 10MHz. Not a great improvement, but an improvement. MCA also supports either a 16-bit data path or a 32-bit data path. It actually has a "streaming" mode wherein it can transfer 64 bits at a time.

Software Board Configuration. Anyone who's installed an ISA board had to struggled with small DIP switches and jumpers. They are often hard to change around, and you have to remove the PC's cover to get to them. Micro Channel boards, in contrast, are software configurable. No jumpers, no DIP switches. You just run one central configuration program, and you can set up a computer by clicking things with the mouse, rather than rooting around in the machine.

Boards Can Share Interrupts. With ISA you could put a mouse card on the same IRQ level as your local area network board, or your network connection could crash the first time that you move the mouse. That is because of how ISA was designed. With MCA, it is possible to design a board that shares its interrupts with other boards.

Bus Mastering Improves upon DMA. Direct Memory Access (DNIA) a way for expansion boards to quickly transfer data from themselves to the system's RAM, or from the RAM to the boards. DMA's main goal is speed to make the PC faster. Boards transfer data directly to RAM, or RAM to boards using DMA, but not boards to boards. That's handled by a kind of "super" DMA called bus mastering. Bus mastering was not supported ISA but MCA supports bus mastering. Bus mastering is another one of those features that first appeared with MCA but is a must for any modern advanced bus.

It also includes something called Programmable Option Select, or POS. It allows circuit boards to be a lot smarter about how they interact with the computer. For one thing, DIP switch and configuration problems lessen considerably.

EISA (Extended Industry Standard Architecture

A group called "Watch zone" (Wyse, AST, Tandy, Compaq, Hewlett-Packard, Zenith, Olivetti, NEC, and Epson) got together forming a joint venture to respond to MCA. This new bus, called the Extended Industry Standard Architecture (EISA) has MCA's good features, without sacrificing compatibility with the old AT (ISA) bus. EISA is moderately successful.

EISA's features are summarised below:-

- They include: 32-bit data path
- Enough address lines for 4GB of memory

- More 1/0 addresses, 64K of them
- Software setup capability for boards, so no jumpers or DIP switches, similar to POS
- 8 MHz clock rate (unfortunately)
- No more interrupts or DMA channels
- Supports cards that are physically large, making them cheaper to build (smaller cards cost more to design)
- Bus mastering

Note that EISA is not a local bus, as it runs at 8 MHz. It runs at that speed because it must be hardware compatible with ISA, hence the need for slow bus slots. EISA will also run DMA at higher speeds than will ISA. The lack of local bus means, however, that EISA memory boards aren't a possibility.

Local Bus

In the XT and AT days, you expanded memory just by buying a memory expansion card, putting memory chips on it, and inserting the card into one of the PC's expansion slots. But by the time that PCs got to 12 MHz, that easy answer disappeared. No matter how fast your PC is 33,100, 200 MHz-the expansion slots still only run at 8 MHz.

Most boards in expansion slots communicate with things that are fairly slow anyway, like floppy drives, printer ports, modems, and the like. There are really only a handful of boards that benefit from really high speeds. Board that really needs to be able to blitz data around is a video graphics board. Hard disk interfaces benefit from high speed, and in particular SCSI hard disk interfaces. Video capture boards are another type of fast board, and local area network cards are yet another candidate for local bus.

There are two main kinds of local bus around these days: the VESA Local Bus (VLB) and the Peripheral Component Interconnect (PCI) bus.

VESA Local Bus

Vendors started including local bus slots for video cards, and then selling the video cards that fit into those slots. The problem was one of non-compatibility: the local bus video card on one machine was incompatible with the local bus video card on another machine. Since the lack of cross-compatibility was getting in the way of video board, industry group promulgated and promoted a local bus standard called VESA after the group's name-Video Electronic Standards Association.

PCI: High Performance Local Bus

VESA Local Bus (VLB) is an important step in the evolution of computers , but it's not enough. VLB is really just a 32-bit, high-speed extension of the older, dumb ISA architecture. VLB offers improved speed, but no better ways of using that speed. VLB does not offer most of the attractive features of the Micro

Channel and EISA buses; it does not offer software setup of boards or bus mastering. VLB systems are still saddled with jumper-setting installations, and the CPU must baby-sit every single data transfer over the VLB bus. Forcing the CPU to manage each and every transfer keeps the CPU squarely in the middle of the system, making it a bottleneck to system performance.

Making computers faster, then, requires the near-impossible technology enhancement of creating cheap, reliable processors that run in the hundreds of megahertz. A bus that supported bus mastering, in contrast, could support a system composed of dozens of medium-speed processors, one on the disk controller, one on the video board, one on the serial port, and so on. A community of 33 MHz CPUs would be a lot faster than a fiercely centralised system dependent on a fragile, super-fast 300 MHz CPU driving slow peripherals.

Intel designed newer, faster bus slot called PCI, short for Peripheral Component Interconnect. Its features are given below.

Processor Independence. The PCI bus doesn't directly interface to the CPU. Rather, it communicates with the CPU via a "bridge circuit" that can act as a buffer between the specifics of a particular CPU and the bus. It means non-PC computer can also use this bus.

Wider Data Path. PCI distinguishes itself first because it is a 64-bit bus. PCI supports a data path appropriate for Pentium based computers, which require 64 bits at each clock cycle. PCI also supports a 32-bit data path, however, making it appropriate for use in older 486 systems.

High Speed. Like VLB, PCI runs to 66 MHz. The net throughput of a PCI bus can be 264 Mbps with a 32-bit board, or 512 Mbps with a 64-bit. Motherboards with 100Mhz PCI bus are used for PII and PIII processors.

Backward Compatibility. Although ISA or EISA boards cannot fit in PCI slots, the chipset that supports PCI also supports ISA and EISA. That means that it's easy to build a PC with PCI, ISA, and EISA slots all on the same motherboard.

Bus Mastering. Like EISA and Micro Channel and unlike VLB, PCI supports bus master adapter boards. Non bus-mastered data transfers require a lot of the CPU's time. For example, one author reports that a file transfer via an Ethernet network required over 40 percent CPU utilisation when the Ethernet card was ISA, but only 6 percent with a similar setup and a PCI Ethernet card.

Software Setup. PCI supports the Plug-and-Play standard developed in 1992. There are in general, no jumpers or DIP switches on PCI boards. To set up a PCI board, you just run the PCI Configuration Program. Reconfiguring a system can be done without opening the computer. The configuration program can be run to list information about all of the boards in the system and what resources they use. PCI is a good architecture, and it is relatively cheap to build PCI boards. It is no doubt the premier PC bus.

PC Card (PCMCIA): The Portable Bus

Laptop computers are an absolute must for travelling professionals. But there is one thing that laptops have always been difficult about supporting: add-on circuit boards. PCMCIA (or the PC Card) removes the objection. It has been customary to add two kinds of hardware to laptops: an internal modem and more memory. Personal Computer Memory Card Industry (PCMCIA) also called as PC Cards were developed in Japan. PC Card boards are about the dimensions of a credit card, but a mite thicker.

Type 1, Type 2, and Type 3 PC Card Slots. The standard proved extremely popular, so popular in fact that hardware vendors said to the PCMCIA and soon modems and hard disks were also interfaced to it. So the memory card interface became a "PC Card Type I slot."

Type I slot is 3.3 millimetres thick, with a 68-pin connector. Most Type I cards are memory cards, either normal RAM or "flash" memory cards loaded with a piece of software. The need for internal modems drove the Type 2 slots. While developing Type 2 ,an important software standard called Card Services and Socket Services was developed. Type 2 cards can be designed to act as an object placed directly into the PC's memory. Type 2 cards are 5 millimetres thick, allowing more space for more complex circuitry. Type I cards will work in Type 2 slots.

The PCMCIA has defined a Type 3 specification, one flexible enough to support removable hard disks. The main difference of Type 3 is that it is a lot thicker. Type 3 cards can be 10.5 millimetres thick.

Socket and Card Services. The PC Card standard supports the ability to remove and install a PC Card "on the fly". All other buses require to power down the computer before installing or removing a card, but PC Card supports "hot swap. The computer supports this capability with two levels of software support.

- Socket services is the PCMCIA name for the BIOS-like software that handles the low-level hardware calls to the card. They are loaded like a device driver. While cards can be swapped without powering down, changes in cards do require a reboot.
- Card services is a higher-layer set of routines that manage how the PC Card memory areas map into the CPU's memory area. They also provide a high-level interface supporting simple commands that are common to almost all PCMCIA cards, commands like erase, copy, read, and write data.

In Windows 95, however, you can change most PC Card cards at will, using them and removing them without rebooting.

PC Card Features. Let's compare PC Card to the other buses that we've discussed, feature for feature.

- Memory address space: PC Card supports a 64MB addressing ability. This is because the bus uses 26 bits for addressing.(*-Bus mastering: PC Card does not support bus mastering or DMA.

- Plug-and-play setup: PC Card allows that hardware setups be done with software. Because of the physical size of a PC Card, you'll never see jumpers or DIP switches.
- Number of PCMCIA slots possible in a single system: Most of the other buses support no more than 16 slots. The PC Card standard can, theoretically, support 4080 PC Card slots on a PC.
- Data path: The data path for PC Card is only 16 bits, a real shame but one that will probably be fixed in the next version of the standard.
- Speed: Like other modern bus standards, PC Card is limited to 33 MHz clock rate.

The smaller size of PC Card cards, coupled with their low power usage, makes the new bus quite attractive not only for laptops, but also for the so-called "green" PCs, desktop computers designed to use as little power as possible. For that reason, PC Card could become an important desktop standard as well as a laptop standard.

ASSIGNMENT

Q1 What is a bus? What are the general categories of line comprising the bus?

Q2 What was the requirement of the first PC bus? What were the improvements in AT bus with respect to the PC bus?

Q3 Which are the two most popular buses found on the modern PCs? Why are two types of buses required on the same PC and how can these coexist?

Q4 What is the local bus? Explain the structure of VESA local bus in brief.

Q5 Which is the bus available on a Laptop? Explain briefly sockets and card services.

Q6 IS it important that the bus speed be same as the CPU speed? Explain your reasons.

CHAPTER 5

PERIPHERALS AND CONTROLLERS

PERIPHERALS

Introduction

Peripheral devices are the agents through which one interacts with a computer. Earlier the peripheral devices were huge electromechanical machines with several circuits and mechanisms. Thanks to the Progress in the field of electronics and the other related fields, the present day peripheral devices are small and powerful. The evolution of the microprocessor has brought in several improvements in computer peripherals.

Keyboards

The keyboard is the most friendly input Peripheral. Both program and data can be keyed in through it. In addition, certain commands to software can be given from the keyboard. It is almost impossible to use a computer without a keyboard.

The keyboard consists of a set of key switches. There is one key switch for each letter, number, symbol etc. When a key is pressed, the key switch is activated. The keyboard has an electronic circuit to determine, which key has been pressed. Then a standard 8-bit code is generated and sent to the computer. Detecting which key is pressed and generating corresponding code known as encoding.

There are two types of keyboards. A serial keyboard sends the data, bit by bit, in a serial fashion. The computer converts the data into a parallel byte. A parallel keyboard sends the data as a byte in parallel form, all the bits are sent simultaneously on different lines (wires). The cable between the keyboard and the computer should have more wires in a parallel keyboard.

Keyboard Function. Figure 2.2 shows the block diagram of a keyboard Generally, the key switches are connected in a matrix of rows and columns. Each key switch has a fixed set of co-ordinates: row number and column number.

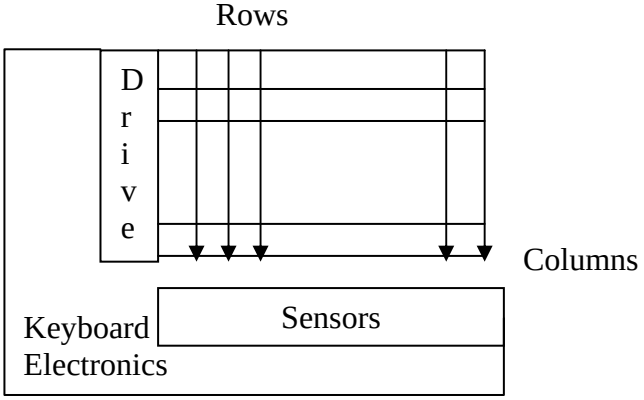


Fig 2.2 Keyboard Block Diagram

The functions to be performed by the keyboard electronics are:

- Sensing a key depression
- Encoding
- Sending the code to computer.

A standard technique known as scanning is followed by the keyboard electronics. The rows are used as inputs to the matrix. The keyboard electronics sends signals to the matrix through the rows. The columns are used as outputs from the matrix. The electronics circuit senses the column lines.

There are different types of key switches. Some of the common types are:

- Mechanical key switch
- Membrane key switch
- Capacitive key switch
- Hall effect key switch
- Reed relay key switch

CRT Display Monitor

The Cathode Ray Tube (CRT) display is a widely used visual display unit (VDU) for the past several years. The CRT display is also called CRT monitor. Figure2.3 present a block diagram of a CRT monitor. The CRT monitor receives video signals from the computer, and displays the video information as dots on the CRT screen. The computer has CRT Controller cct, which works in synchronisation with a CRT monitor. The main unit in the CRT monitor is the CRT itself. CRT is an evacuated glass tube it is usually called a picture tube. The CRT is an evacuated glass tube with a fluorescent coating on the inner front surface, called screen. An electron gun at one end (neck) emits an electron beam. This beam is directed towards the screen. When the beam strikes the screen, the phosphor coating on the screen produces illumination at the, glass the spot where the electron beam strikes. The beam is deflected by an electro-magnetic deflection in order to produce illumination at various spots on the screen. Horizontal deflection coils deflect the beam in the horizontal direction and the vertical coils deflect the beam in the vertical direction. The illumination caused on the screen exists for a few milliseconds due to the persistence of the phosphor. To create a permanent image on the screen, it is necessary to cause illuminations repeatedly. This is done by scanning the CRT screen with the electron beam. The common method of scanning is called Raster scan. In this method, the electron beam is moved back and forth across the screen. On reaching the extreme right, the beam is brought back to the left. It then moves right from the next scan line. On reaching the bottom of the screen, the beam is brought to the top of the screen. To produce an image, the beam is turned on or off. The video information from the computer is used for turning the beam on or off at appropriate places when the beam scans the screen. The HSYNC from the computer provides horizontal synchronisation for each scan line, i.e., when the beam starts from the left side of the screen, after returning from the right. VSYNC

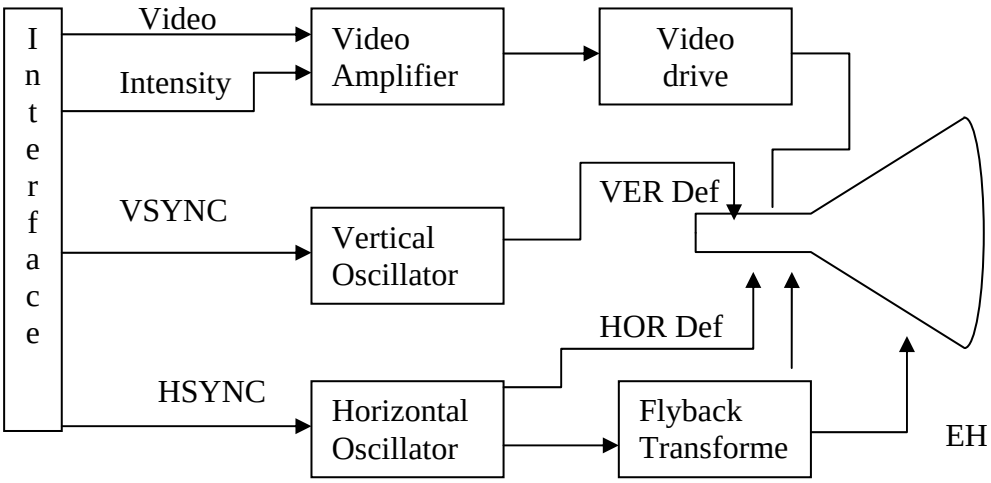


Fig 2.3 Block Diagram of CRT Monitor

from the computer is used for vertical synchronisation for each raster.

There are two types of images on CRT displays : Alphanumeric displays (or text) and graphics. The alphanumeric display system generally follows the dot matrix scheme for each character generation, as in a dot matrix printer.

Printer

The printer is an electromechanical device. It has both electronic circuits and mechanical assemblies. The electronic circuits control the mechanical assemblies. Hence the electronic mechanism to print data received from the computer. The mechanical assemblies include print head assembly, print carriage motor, ribbon assembly, paper movement assembly, sensor assemblies etc.

Printer Functions. Printer receives data characters from the computer and prints the characters on the paper. In addition, the printer also receives control characters from the computer. The control characters are not printable characters. They convey some sort of control information to the printer. Some of the control characters widely used are CR (Carriage Return), LF (Line Feed) and FF (Form Feed). CR specifies that the printer head carriage should return to the first print column. Any subsequent data character received will be printed starting from the first column. LF informs the printer to skip one line on the paper. FF instructs the printer to skip the paper to the beginning of the next page (or form). The printer stationery (paper) is available as continuous sheets folded into pages. Each page is known as a form.

Magnetic Storage Devices

Figure 2.4 shows different types of magnetic storage devices used as auxiliary or secondary memories. The magnetic disk drive has been used as an input/output device for the computer since 1956. The invention of disk drives has made the magnetic drum less attractive, and hence obsolete. It has also reduced the use of the magnetic tape drive. The tape drive is used nowadays mainly for two purposes:

- As a back up unit it is used to take copies of the disk contents. These are useful, in case the disk contents are destroyed due to some programming error or hardware problem.
- To transport files from one site to another site, the tape media is more convenient.

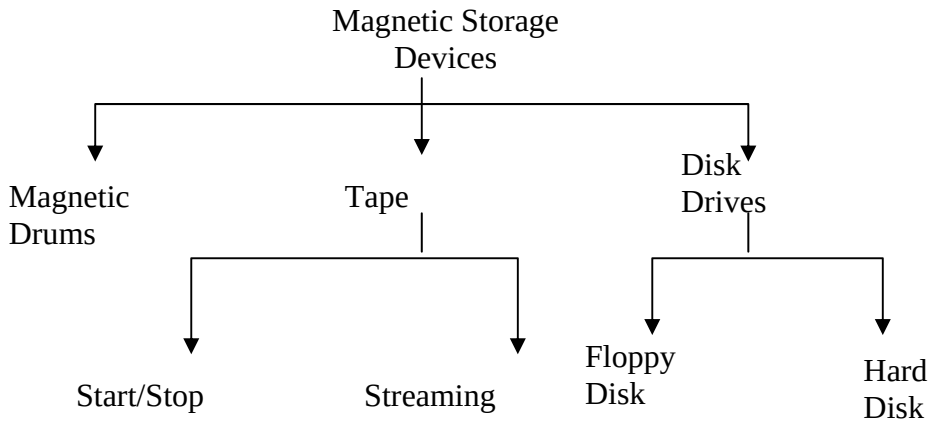


Fig 2.4 Magnetic Storage Devices

Most commonly used magnetic storage devices are the hard disk drives and the floppy disk drives. The hard disk is a fixed drive while the floppy drive is a removable drive. The capacity of hard disk varies from few hundred MB to many GB nowadays. Floppy drives are available in different sizes and capacities. The two most widely used are 3½ inch 1.44 MB floppy and 5¼ inch 1,2 MB floppy. Block diagram of a disk drive is shown in fig 2.5.

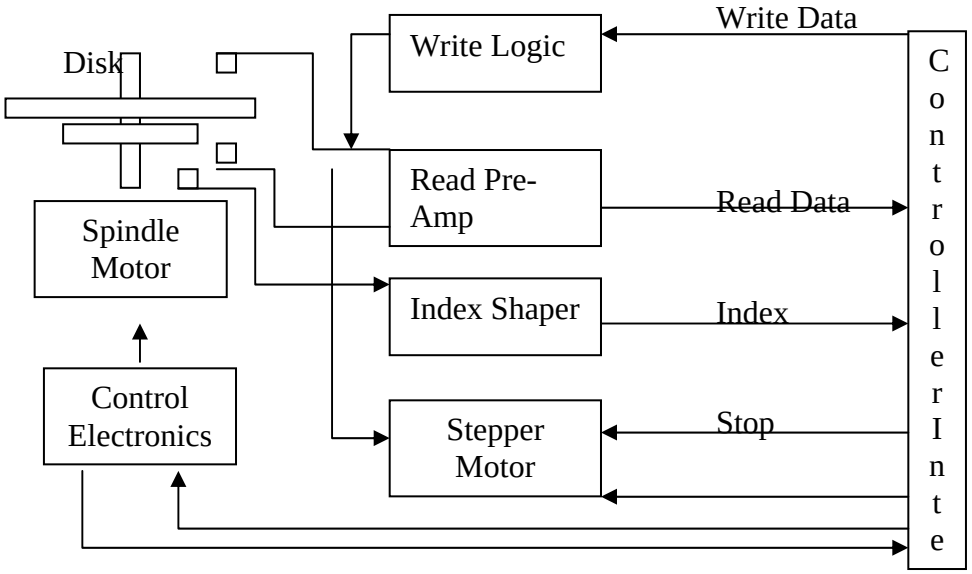


Fig 2.5 Block Diagram of Disk Drive

CONTROLLERS OR INTERFACE

Keyboard Interface

The keyboard interface block diagram is shown in Fig. 2.6. The keyboard is a serial keyboard. On typing a key, the keyboard sends a scan code corresponding to that key, as serial data, bit by bit. There is a Serial In Parallel Out (SIPO) shift register on the motherboard which converts this serial data into a parallel byte (S bit scan code). Once a byte is assembled here, an interrupt (IRQI) is raised to the CPU through the interrupt controller, PIC. The keyboard interrupt service routine reads this scan code through the PPI-port A, at which the scan code from the SIPO is available. It then stores this scan code in RAM. This scan code is ultimately converted into ASCII code by the software.

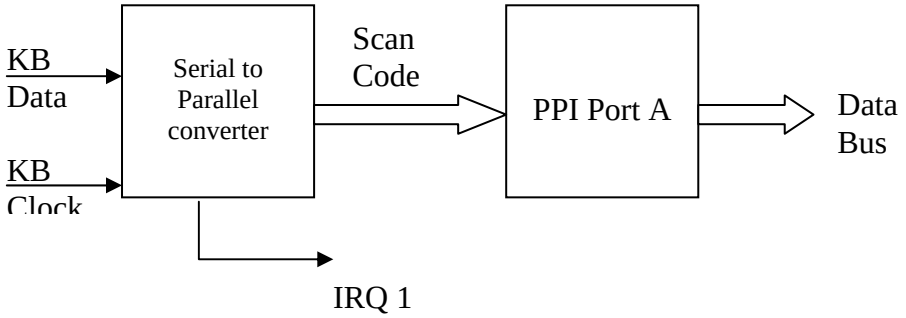


Fig 2.6 Keyboard Interface

Parallel Interface

The printer controller or parallel interface block diagram is shown in Fig. 2.7. The parallel interface between the printer controller and the printer is known as **Centronics Parallel Interface**. The print data and command signals sent by the software is simply passed on to the printer by the printer controller. Similarly the status signals sent by the printer are simply made available to the software on an input port. Perhaps because the printer controller does not have much intelligence and acts only as a coupler between the CPU and the printer, IBM has termed it a printer adapter.

The printer controller supports data transfer in two different modes:

- Programmed mode
- Interrupt mode.

Different software routines operate the printer controller in one of these two modes. The software chooses the mode of data transfer and informs the printer controller of this by an appropriate command. Three different printer controllers are available in the IBMPC. These are named LPTI, LPT2 and LPT3 by software. Parallel interface is also used for connecting scanners, cameras and for direct cable connect between two computers.

Serial Interface

The serial interface block diagram is shown in Fig. 2.8. The PC supports two serial interfaces. Each is a RS232C standard interface, which is a modem interface. A serial printer or a terminal or another computer can be connected to each of the two serial interfaces. If these are at distant places, a set of modems and telephone lines are used. If these are close, they can be connected directly, without modems but with an appropriate cable.

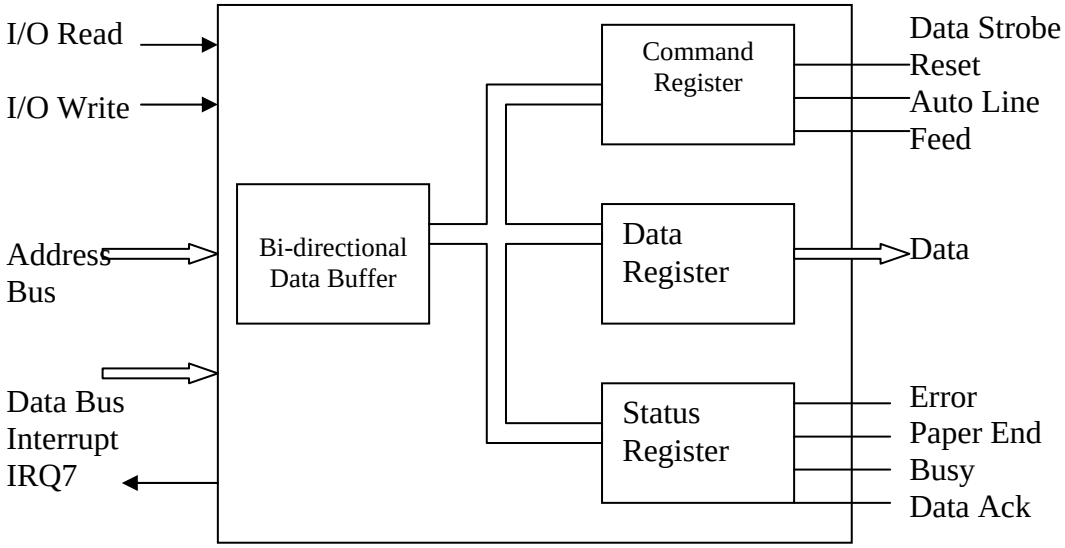


Fig 2.6 Printer Controller

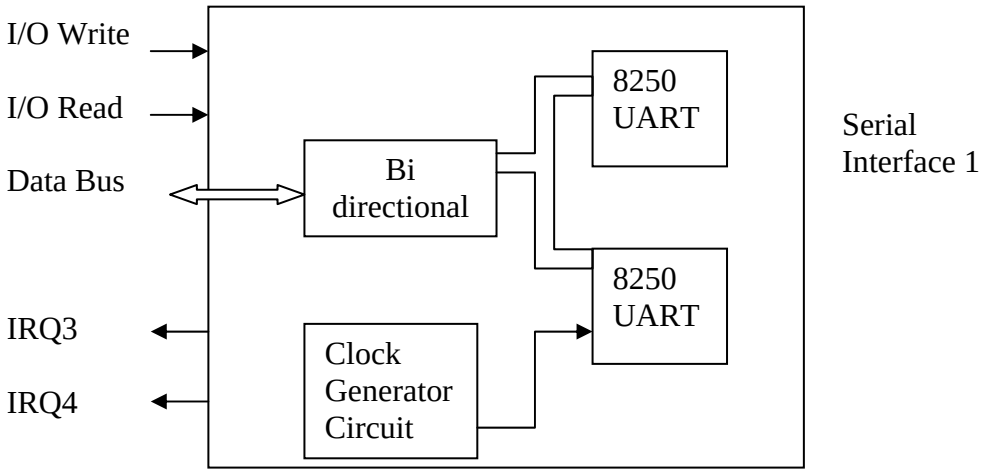


Fig 2.7 Serial Interface

The IBM PC supports both asynchronous and synchronous communication. Synchronous communications is used very rarely, usually for high-speed communication between one computer to another remote computer. Asynchronous communication is widely used in PCs. Hence we generally mean asynchronous communication when we refer to a serial interface.

The serial interface takes care of converting the parallel data from the CPU into serial bits. It also adds parity bit and start-stop bits to the data stream. The parity bit is used for error detection. The start-stop bits are used for achieving synchronisation between sending end and receiving end. In this method (asynchronous communication), there can be time gaps between one byte to the next.

When data bits are received from the other end, the serial interface controller converts them into a parallel data byte. It also removes start-stop bits and checks for parity error. It can be operated both in interrupt mode and in program mode, for data transfer.

The serial interface controller, in the IBM PC as well as in clones, is achieved by using a communication controller, IC 8250. The modem interface is obtained by connecting a set of MC1488 and MC1489 ICs to an 8250.

The two serial interfaces are referred to as COMI and COM2 by software. One of the very common uses of the serial interface in a PC is for connecting a serial mouse.

CRT Display Controller

Figure 2.9 presents the block diagram of the CRT controller in a PC.

On CRT monitors, programs can display either alphanumeric text or both alphanumeric text and graphics (figures). The original IBM PC design offered two types of CRT display controllers. These are

- Monochrome Display Adapter (MDA)
- Colour Graphics Display Adapter (CGA)

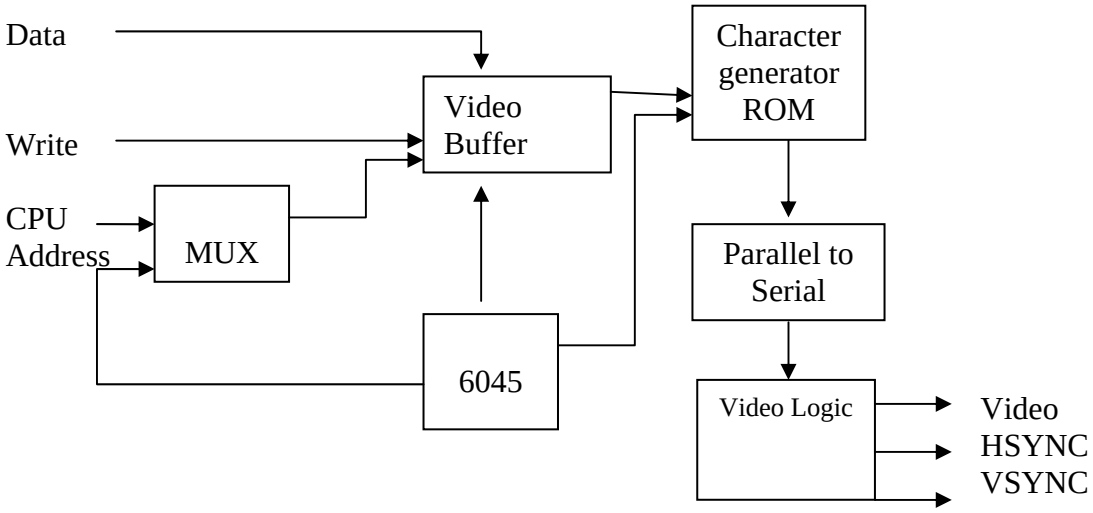


Fig 2.9 Block Diagram Display Adapter

The MDA supported only text mode display on monochrome monitors. The CGA supported both monochrome displays and colour displays. All modern computers use Super Video graphic array (SVGA) video adapter or interface. It

offers both text mode and graphics mode display. The basic principle of operation of MDA, CGA, VGA or SVGA is identical, as shown in Fig. 2.10. Each of them has a video buffer (Screen memory). This is a shared memory between the CPU and the CRT controller. The text or graphics pattern to be displayed on the CRT screen is stored in the video buffer memory by the CPU. The CRT controller reads the contents of the video buffer memory, gets the dot patterns from ROM using video buffer contents and sends the video signal to the CRT monitor along with synchronisation signals HSYNC and VSYNC.

A SVGA adapter also called as Video Board is a plug and play device and can be configured using software. It has a device driver software, which controls it. A video chip on the video board examines the data in the video memory and creates a digital image signal. That digital signal is then converted to an analog signal by a chip called the DAC, the Digital-to-Analog Converter, another chip on the video board, and the resultant signal goes out the connector on the back of the board, and into the monitor.

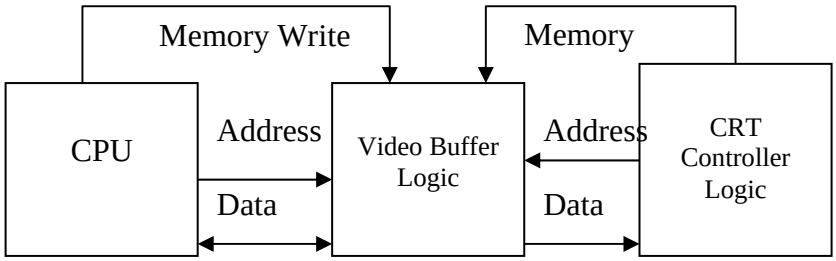


Fig 2.10 Video Buffer as Shared Memory

Video boards are distinguished by their resolution, which is the number of dots (pixels) that they can put on the video screen. More dots means sharper pictures. They are also distinguished by how many different colours they can display on those dots, and by how much work the CPU must do in order to create images.

The CPU must do all of the work of picture creation, it has to place each and every one of the pixels on the computer's screen. But some video boards contain special circuitry called accelerator or bit blitter chips that can speed up video operations considerably.

Floppy Disk Controller (FDC)

Figure 2.11 presents the block diagram of a floppy disk controller. The FDC can support upt6 four floppy disk drives. The FDC is connected to the system bus and to the DMA controller. It performs data transfer in DMA mode.

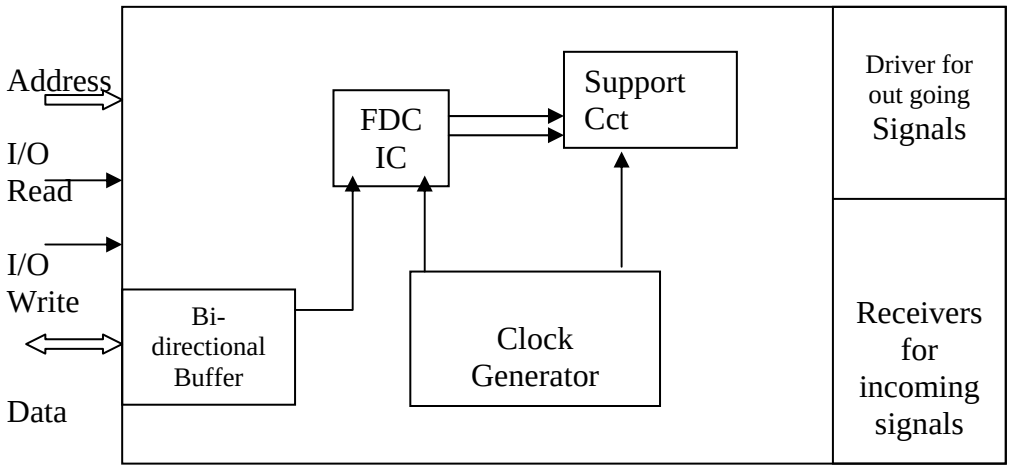


Fig 2.11 Floppy Disk Controller

The FDC is based on the programmable IC, NEC 765 or Intel 8272. It is an intelligent floppy disk controller IC. It converts data from parallel to serial and vice versa. It also has logic to generate CRC characters during writing on floppy diskette and also to verify CRC characters during reading. CRC stands for cyclic redundancy check a technique used for checking the integrity of data.

The FDC has the capability to support different sector formats. It can handle floppy disk drives of different speeds. The desired choice of parameters is made known to the FDC by the software/BIOS. The IBM PC was designed to support 5¼-inch floppy diskettes of 360 KB. Some of the clones have also added the capability for high-density floppy diskettes (1.2 MB) to the PC. All modern PCs have 3½ inch 1.44 MB floppy drive and some also use a 5¼ inch drive.

The BIOS issues commands and associated parameters to the FDC. The FDC executes the command by generating appropriate control signals to the FDD. After a command is completed, the FDC presents the status of completion to the BIOS. For each command a set of command parameters are to be supplied to the FDC IC. This is achieved by executing a series of output instructions by the CPU. The FDC IC has several registers to store the command and command parameters.

For a read or write command, the system, FDC and FDD are all involved as data transfer takes place. For commands like SEEK or RECALIBRATE, only the FDC and FDD are involved, as there is no data transfer.

The FDC issues STEP pulses and DIRECTION signals to the FDD. On completing the command execution, the FDC interrupts the CPU. The ISR reads the status from the FDC IC and finds out whether the command is completed successfully or not. For this purpose, the FDC IC stores various status parameters related to the command in its internal registers. The ISR reads all this information by a series of input instructions.

In addition to the FDC IC, external hardware circuits used are:

- Address decoder
- Control port

- Data separator
- Write pre-compensation circuit
- Drivers
- Receivers

The address decoder enables the FDC when 8088 performs an input instruction or output instruction. The control port is used to achieve certain controls over the FDC IC and the FDDS. These include resetting the FDC IC, enabling DMA request and interrupt request from FDC IC, selecting a FDD and turning on the spindle motor in a FDD. The data separator works together with the FDC IC to separate DATA pulses and clock pulses. The write pre-compensation circuit compensates for the shifting of pulses due to peak shift during the read operation. The drivers are used for the control signals to the FDD and the receivers are used for the status signals from the FDD. The floppy disk drives are connected to the FDC by a 34 pin flat cable. A single cable is used in daisy chained mode to connect a number of FDDs. In most modern computers the IDE controller is used for both FDC as well as HDC. In modern computers the controller is placed on the motherboards and connectors are available on the motherboards to connect the drives to it. These drives can be configured with the help of CMOS setup.

Hard Disk Controller

There was no uniformity in the LSIs used in the HDC. Most PCs used WDIOIO IC and some custom LSIs for additional circuits. Some PCs used a microprocessor along with WD1010. Others used only custom LSIS.

Conceptually, the HDC is identical to the FDC. It has certain additional circuits, which are not present in the FDC. These are Sector Buffer, ECC Logic, Retry Logic, and Diagnostics Logic.
The sector buffer is used to store the data bytes of one full sector both during the read operation and write operation. During the write operation the data bytes received from the memory in DMA mode are stored in the sector buffer. The HDC takes these bytes from the sector buffer, serialises them into data bits, mixes them with clock bits and sends the data to the hard disk drive. During the read operation, the data consists of DATA bits and CLOCK bits. The HDC separates them, converts the data bits into parallel bytes and stores them in the sector buffer. From the sector buffer, these data are transferred to memory in DMA mode. Due to the sector buffer, the DMA transfer and hard disk data transfer are isolated.

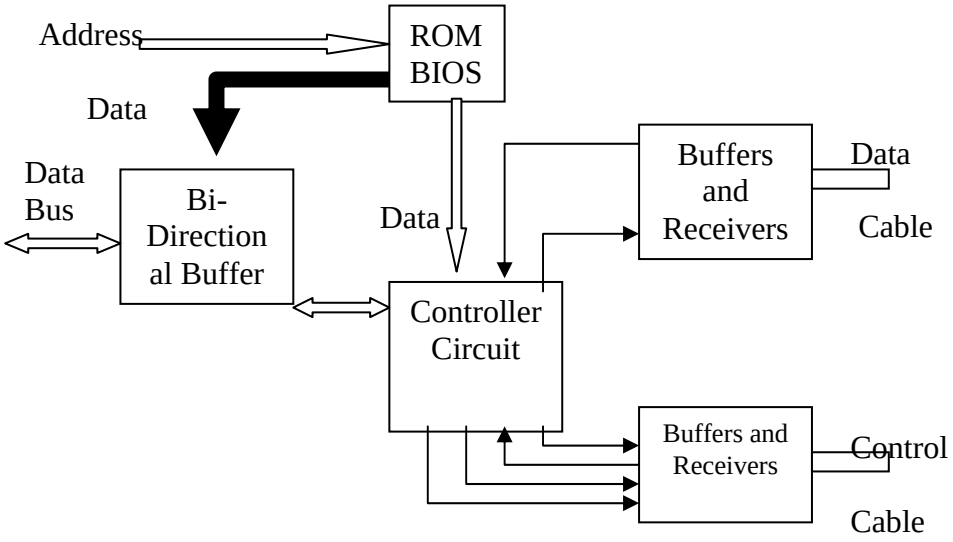


Fig 2.12 Hard Disk Controller

The ECC logic generates a 32-bit ECC pattern for the data bytes, which is also written in the sector along with the data bytes. During a read operation, the HDC checks the ECC pattern to verify whether the data bytes are error-free. If there is any error, the HDC corrects it. The ECC is used for the data field in each sector whereas CRCC is used for the ID field in each sector.

As brought out earlier in most modern computers the IDE controller is used for both FDC as well as HDC. The controller is placed on the motherboards and connectors are available on the motherboards to connect the drives to it. Upto four HDD can be connected to one IDE controller including the CD-ROM drives. These can be software configured using the CMOS setup. The drives are named as Primary Master and Slave, Secondary Master and Slave. Primary and Secondary drives connected using different cables. Block diagram of Hard Disk Controller is shown in Figure 2.12.

ASSIGNMENT

Q1 Classify the following statements into valid or invalid:

(a) In FDD there are two read write heads for each side of the diskette.

(b) In HDD the read write heads do not touch the platters but fly at a height from it.

(c) The index hole is present only in a soft sectored floppy and not in hard sectored floppy.

(d) A character printer should have a serial interface.

(e) The keyboard sends clock and data bots on the same wire.

(f) A graphics monitor cannot display text.

(g) The advantage in daisy chaining is reduction in cable length and hardware circuits.

Q2 What is a controller and why is it required?

Q3 What is the basic difference in the functioning of FD controller and HD controller?

Q4 Explain briefly the working of a video card. How does the amount of memory on the video card effect the performance of the card.

Q5 Explain briefly how a key strike by the user results in s display on the screen.

Q6 Why do we need two types of communication ports explain in brief?

CHAPTER 6

OPERATING SYSTEM

Multi-programmed Batched Systems

Spooling results in several jobs that have already been read waiting on disk, ready to run. A pool of jobs on disk allows the operating system to select which job to run next, in order to increase CPU utilisation. When jobs come in directly on magnetic tape, it is not possible to run jobs in a different order. Jobs must be run sequentially, on a first-come , first-served basis. However, when several jobs are on a direct-access device, such as a disk, job scheduling becomes possible.

The most important aspect of job scheduling, is the , ability to multi-program. Off0line operation and spooling for overlapped I/O have their limitations. A single user cannot, in general, keep either the CPU or the I/O devices busy at all times. Multiprogramming increases CPU utilisation by organising jobs so that the CPU always has something to execute.

The operating system keeps several jobs in memory at a time(Fig 6.1). The operating system picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as tape to be mounted, a command to be typed on a keyboard, or an I/o operation to complete. In a non multi-programmed system, the CPU would another job. When that job needs to wait , the CPU is switched to another job and so on. Eventually , the first job finishes waiting and gets the CPU back. As long as there is always some job to execute, the CPU will never be idle.

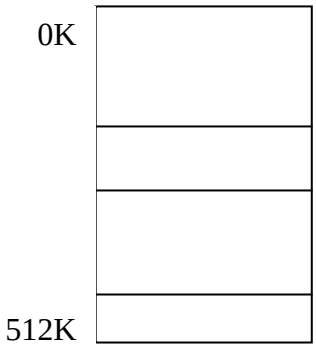


Fig 6.1 Memory Layout for a multi programming system

Multiprogramming is the first instance where the operating system must make decisions for the users. Multi-programmed operating systems are therefore fairly sophisticated. All the jobs that enter the system are kept in the job pool. The pool consists of all processes residing on mass storage awaiting allocation of main memory. If several jobs are ready to be brought into memory, and there is no enough room for all of them, then the system must chose among them. This decision is job scheduling. When the operating system selects a job from

the job pool, it loads it into memory for execution. Having several programs in memory at the same time requires some form of memory management. In addition, if several jobs are ready to run at the same time, the system must choose among them. This decision is CPU scheduling. Finally, multiple jobs running concurrently require that their ability to affect one another be limited in all phases of the operating system, including process scheduling, disk storage, and memory management.

Time-Sharing- Systems

Multi-programmed batched systems provide an environment where the various system resources (for example, CPU, memory, and peripheral devices) are utilised effectively. There are some difficulties with a batch system from the point of view of the programmer or user, however. Since the user cannot interact with the job when it is executing, the user must set up the control cards to handle all possible outcomes. In a multi-step job, subsequent steps may depend on the result of earlier ones. The running of a program, for example, may depend on successful compilation. It can be difficult to define completely what to do in all cases.

Another difficulty is that programs must be debugged statically, from snapshot dumps. A programmer cannot modify a program as it executes to study its behavior. A long turnaround time inhibits experimentation with a program.

Time-sharing (or multitasking) is a logical extension of multiprogramming. Multiple jobs are executed by the CPU switching between them, but the switches occur so frequently that the users may interact with each program while it is running. In an interactive, or hands-on, computer system provides on-line communication between the user and the system. The user gives instructions to the operating system or to a program directly, and receives an immediate response. Usually, a keyboard is used to provide input, and a display screen (such as a cathode-ray tube (CRT), or monitor) is used to provide output. When the operating system finishes the execution of one command, it seeks the next "control statement" from the user's keyboard. The user gives a command, waits for the response, and decides on the next command, based on the result of the previous one. The user can easily experiment, and can see results immediately. Most systems have an interactive text editor for entering programs, and an interactive debugger for assisting in debugging programs.

Batch systems are quite appropriate for executing large jobs that need little interaction. The user can submit jobs and return later for the results; it is not necessary to wait while the job is processed. Interactive jobs tend to be composed of many short actions, where the results of the next command may be unpredictable. The user submits the command and then waits for the results. Accordingly, the response time should be quite short, on the order of seconds at most. An interactive system is used when a short response time is required.

Early computers were interactive systems. That is, the entire system was at the immediate disposal of the programmer/operator. This situation allowed the

programmer great flexibility and freedom in program testing and development. But this arrangement resulted in substantial idle time while the CPU waited for some action to be taken by the programmer/operator. Because of the high cost of these early computers, idle CPU time was undesirable. Batch operating systems were developed to avoid this problem. Batch systems improved system utilisation for the owners of the computer systems.

Time-sharing systems were developed to provide interactive use of a computer system at a reasonable cost. A time-shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer. Each user has at least one separate program in memory. A program that is loaded into memory and is executing is commonly referred to as a process. When a process executes, it typically executes for only a short time before it either finishes or needs to perform I/O. I/O may be interactive; that is, output is to a display for the user and input is from a user keyboard. Since interactive I/O typically runs at people speeds, it may take a long time to complete. Input, for example, may be bounded by the user's typing speed; five characters per second are fairly fast for people, but are very slow for computers. Rather than let the CPU sit idle when this interactive input takes place, the operating system will rapidly switch the CPU to the program of some other user.

A time-shared operating system allows the many users to share the computer simultaneously. Since each action or command in a time-shared system tends to be short, only a little CPU time is needed for each user. As the system switches rapidly from one user to the next, each user is given the impression that she has her own computer, whereas actually one computer is being shared among many users.

The idea of time-sharing was demonstrated as early as 1960. Time- shared systems are more difficult and expensive to build due to the numerous I/O devices needed, As the popularity of time sharing has grown, researchers have attempted to merge batch and tinic-shared systems. Many computer systems that were designed as primarily batch systems have been modified to create a time-sharing subsystem. At the same time, time-sharing systems have often added a batch subsystem.

Time-sharing operating systems are even more complex than are multi-programmed operating systems. As in multiprogramming, several jobs must be kept simultaneously in memory, which requires some form of memory management and protection. So that a reasonable response time can be obtained, jobs may have to be swapped in and out of main memory to the disk that now serves as a backing store for main memory. A common method for achieving this goal is virtual memory, which is a technique that allows the execution of a job that may not reside completely in memory. The main visible advantage of this scheme is that programs can be larger than the physical storage. Further, it abstracts main memory into large, uniform array storage, separating logical memory as viewed by the user from physical memory. This frees programmers from concern over memory storage limitations. Time-sharing systems must also provide an on-line file system. The file system

resides on a collection of disks, hence, disk management must also be provided. Also, time-sharing systems provide a mechanism for concurrent execution, which requires sophisticated CPU scheduling schemes. To ensure orderly execution, the system must provide mechanisms for job synchronization and communication, and must ensure that jobs do not get stuck in a deadlock, forever waiting for each other. Multiprogramming and time-sharing are the central themes of modem operating systems

PROCESS MANAGEMENT

A process can be thought of as a program in execution. A process will need certain resources such as CPU time, memory, files, and 110 devices to accomplish its task. These resources are allocated to the process either when it is created, or while it is executing. A process is the unit of work in most systems. Such a system consists of a collection of processes: Operating-system processes execute system code, and user processes execute user code. All these processes can potentially execute concurrently.

The operating system is responsible for the following activities in connection with process management:

- Creation and deletion of both user and system processes.
- Scheduling of processes.
- Provision of mechanisms for synchronization, communication, and deadlock handling for processes.

Early computer systems allowed only one program to be executed at a time. This program had complete control of the system, and had access to all of the system's resources. Current-day computer systems allow multiple programs to be loaded into memory and to be executed concurrently. This evolution required firmer control and more compartmentalization of the various programs.

The more complex the operating system, the more it is expected to do on behalf of its users. Although its main concern is the execution of user programs, it also needs to take care of various system tasks that are better left outside the kernel itself A system therefore consists of a collection of processes: Operating-system processes executing system code, and user processes executing user code. All these processes can potentially execute concurrently, with the CPU multiplexed among them. By switching the CPU between processes, the operating system can make the computer more productive.

MEMORY MANAGEMENT

The main purpose of a computer system is to execute programs. These programs, together with the data they access, must be in main memory (at least partially) during execution.
To improve both the utilisation of CPU and the speed of its response to its users, the computer must keep several processes in memory. There are many different memory-management schemes. These schemes reflect various approaches to memory management, and the effectiveness of the different algorithms depends on the particular situation. Selection of a memory-management scheme for a specific system depends on many factors, especially on the hardware design of the system. Each algorithm requires its own hardware support.

Since main memory is usually too small to accommodate all data and programs permanently, the computer system must provide secondary storage to back up main memory. Most modern computer systems use disks as the primary on-line storage medium for information (both programs and data). The file system provides the mechanism for on-line storage of and access to both data and programs residing on the disks.

FILE MANAGEMENT

A file is a collection of related information defined by its creator. Files are mapped, by the operating system, onto physical devices. Files are normally organised into directories to ease their use. For most users, the file system is the most visible aspect of an operating system. It provides the mechanism for on-line storage of and access to both data and programs belonging to the operating system and all the users of the computer system. The file system consists of two distinct parts. A collection of files, each storing related data, and a directory structure, which organises and provides information about all the flies in the system. Some file systems have a third part, partitions, which are used to separate physically or logically large collections of directories. Different ways to handle file protection are employed, which is necessary in an environment where multiple users have access to files, and where it is usually desirable to control by whom and in what ways files may be accessed.

File Concept. Computers can store information on several different storage media, such as magnetic disks, magnetic tapes, and optical disks. So that the computer system will be convenient to use, the operating system provides a uniform logical view of information storage. The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, the file. Files are mapped, by the operating system, onto physical devices. These storage devices are usually non- volatile, so the contents are persistent through power failures and system reboots.

A file is a named collection of related information that is recorded on secondary storage. From the uses perspective, a **file** is the smallest allotment of logical secondary storage that is data cannot be written to secondary storage unless

they are within a file. Commonly, files represent programs both (source and object) and data. Data files may be numeric, alphabetic, alphanumeric, or binary. Files may be free form, such as text files, or may be formatted rigidly. In general, a file is a sequence of bits, bytes, lines, or records whose meaning is defined by the file's creator and user. The concept of a file is thus extremely general.

The information in a file is defined by its creator. Many different types of information may be stored in a files source programs, object programs, executable programs, numeric data, text, payroll records, graphic images, sound recordings, and so on. A file has a certain, defined structure according to its type. A text file is a sequence of characters organised into lines (and possibly pages); a source file is a sequence of subroutines and 'functions, each of which is further organised as declarations followed by executable statements; an object file is a sequence of bytes organised into blocks understandable by the system's linker; an executable file is a series of code sections that the loader can bring into memory and execute. If users are to be able to access both data and code conveniently, an on-line file system must be available. The operating system implements the abstract concept of a file by managing mass-storage devices, such as tapes and disks. Files are normally organised into logical clusters, or directories, which makes them easier to use. Since multiple users have access to files, it is desirable to control by whom and in what ways files may be accessed.

ASSIGNMENT

Q1. Write a short note on multi-programmed systems.

Q2. What are Time sharing systems, Explain?

Q3. How are Multi-programmed systems different from time-sharing systems? Can these co-exist on the same computer?

Q4. What are the functions of an 0S?

Q5. Write short notes on the following:-
- (a) Memory Management?
- (b) Process Management?
- (c) File Management?

CHAPTER 7

NETWORK

Network Types

There are basically two types of networks: local-area networks and wide-area networks. The main difference between the two is the way in which they are geographically distributed. Local-area networks are composed of processors that are distributed over small geographical areas, such as a single building or a number of adjacent buildings. Wide-area networks, on the other hand, are composed of a number of autonomous processors that are distributed over a large geographical area. These differences imply major variations in the speed and reliability of the communications network, and are reflected in the distributed operating- system design.

Local-Area Networks

Local-area networks (LANs) emerged in the early 1970s, as a substitute for large mainframe computer systems. It had become apparent that, for many enterprises, it is more economical to have a number of small computers, each with its own self-contained applications, rather than a single large system. Because each small computer is likely to need a full complement of peripheral devices (such as disks and printers), and because some form of data sharing is likely to occur in a single enterprise, it was a natural step to connect these small systems into a network.

LANs are usually designed to cover a small geographical area (such as a single building, or a few adjacent buildings) and are generally used in an office environment. All the sites in such systems are close to one another, so the communication links tend to have a higher speed and lower error rate than do their counterparts in wide-area networks. So that this higher speed and reliability can be attained, high-quality (expensive) cables are needed. It is also possible to use the cable exclusively for data network traffic. Over longer distances, the cost of using high-quality cable is enormous, and the exclusive use of the cable tends to be prohibitive.

The most common links -in a local-area network are twisted pair, base band coaxial cable, broadband coaxial cable, and fibre optics. The most common configurations are multi-access bus, ring, and star networks. Communication speeds range from I megabyte per second, for networks such as Appletalk and IBM's slow token ring, to I gigabit per second for experimental optical networks. Ten megabits per second is most common, and is the speed of Ethernet. Recently, the optical-fibre-based FDDI network has been increasing its market share. This network is token based and runs at 100 megabits per second.

A typical LAN may consist of a number of different minicomputers or workstations, various shared peripheral devices (such as laser printers or magnetic-tape units), and one or more gateways (specialised processors) that provide access to other networks (Figure 7.1). An Ethernet scheme is commonly used to construct LANs. There is no central controller in an Ethernet

network, because it is a multi-access bus, so new hosts can be added easily to the network.

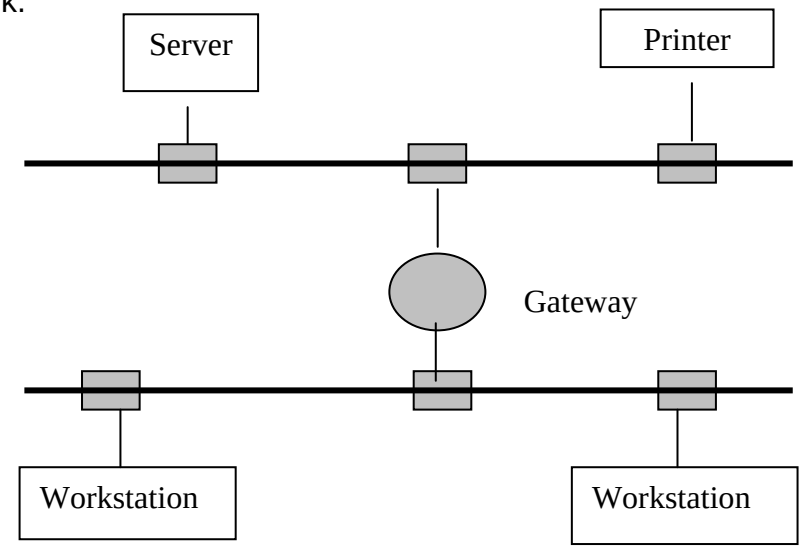


Fig 7.1 Local Area Network

Wide-Area Networks

Wide-area networks (WANs) emerged in the late 1960s, mainly as an academic research project to provide efficient communication between sites, allowing hardware and software to be shared conveniently and economically by a wide community of users. The first WAN to be designed and developed was the Arpanet. Work on the Arpanet began in 1968. The Arpanet has grown from a four-site experimental network to a worldwide network of networks, the Internet, comprising thousands of computer systems. Recently, several commercial networks have also appeared on the market. The Telnet system is available within the continental United States; the Datapac system is available in Canada. These networks provide their customers with the ability to access a wide range of hardware and software computing resources.

Because the sites in a WAN are physically distributed over a large geographical area, the communication links are by default relatively slow and unreliable. Typical links are telephone lines, microwave links, and satellite channels. These communication links are controlled by special communication processors (Figure 7.2), which are responsible for defining the interface through which the sites communicate over the network, as well as for transferring information among the various sites.

As an example, let us consider the Internet WAN. The system provides an ability for hosts at geographically separated sites to communicate with one another. The host computers typically differ from one another in type, speed, word length, operating system, and so on. Hosts are generally on LANs, which are in turn connected to the Internet via regional networks. The regional networks, such as NSFnet in the Northeast United States, are interlinked with routers (described in Section 15.5.2) to form the worldwide network. Connections between networks frequently use a telephone system service called TI, which provides a transfer rate of 1.544 megabits per second. The

routers control the path each message takes through the net. This routing may be either dynamic, to increase communications efficiency, or static, to reduce security risks or to allow communications charges to be computed.

Other WANs in operation use standard telephone lines as their primary means of communication. Modems are the devices that accept digital data from the computer side and convert it to the analog signals that the telephone system uses. A modem at the destination site converts the analog signal back to digital and the destination receives the data. The UNIX news network, UUCP, allows systems to communicate with each other at predetermined times, via modems, to exchange messages. The messages are then routed to other nearby systems and in this way either are propagated to all hosts on the network (public messages) or are transferred to their destination (private messages). WANs are generally slower than LANs; their transmission rates range from 1200 bits per second to over I megabit per second.

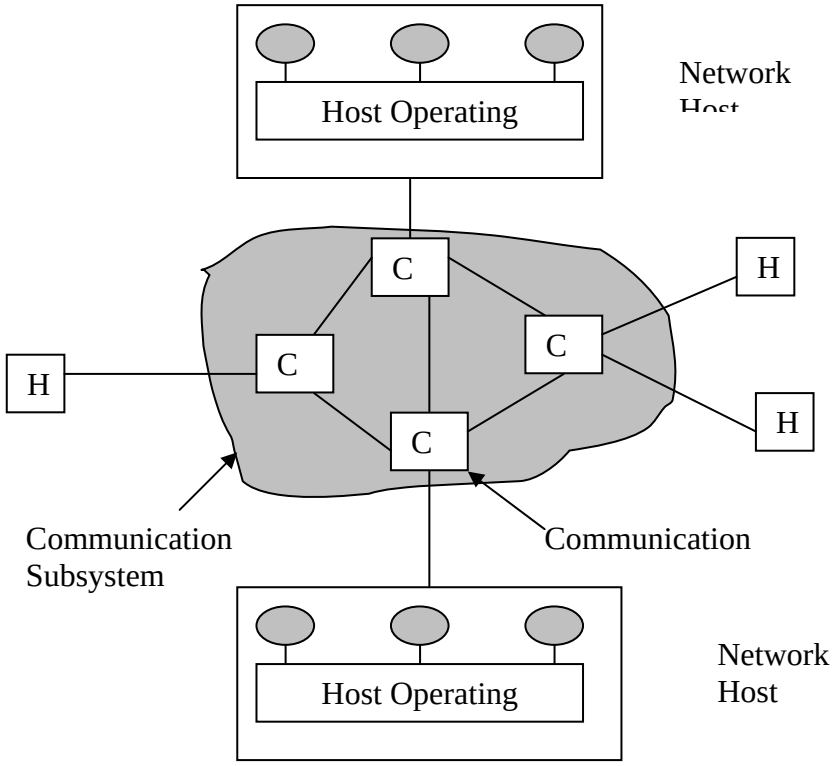


Fig 7.2 Wide Area Network

Network Topology

The sites in a network system can be connected in a variety of ways. Each configuration has advantages and disadvantages. Most common configurations are described in the succeeding paras, and compare them with the following criteria:-

- Basic cost. How expensive is it to link the various sites in the system?
- Communication cost. How long does it take to send a message from site A to site B?

- Reliability. If a link or a site in the system fails, can the remaining sites still communicate with one another?

The various topologies are depicted as graphs whose nodes correspond to sites. An edge from node A to node B corresponds to a direct connection between the two sites.

Fully Connected Networks

In a fully connected network, each site is directly linked with all other sites in the system (Figure 7.3). The basic cost of this configuration is high, since a direct communication line must be available between every two sites. The basic cost grows as the square of the number of sites. In this environment, however, messages between the sites can be sent fast; a message needs to use only one link to travel between any two sites. In addition, such systems are reliable, since many links must fail for the system to become partitioned. A system is partitioned if it has been split into two (or more) subsystems that lack any connection between them.

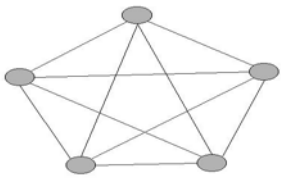


Figure 7.3 Fully Connected Network

Partially Connected Networks

In a partially connected network, direct links exist between some, but not all, pairs of sites (Figure 7.4). Hence, the basic cost of this configuration is lower than that of the fully connected network. However, a message from one site to another may have to be sent through several intermediate sites, resulting in slower communication. For example, in the system depicted in Figure 7.4, a message from site A to site D must be sent through sites B and C.

In addition, a partially connected system is not as reliable as is a fully connected network. The failure of one link may partition the network. For the example in Figure 7.3, if the link from B to C fails, then the network is partitioned into two subsystems. One subsystem includes sites A, B, and E; the second subsystem includes sites C and D. The sites in one partition cannot communicate with the sites in the other. So that this possibility is minimised, each site is usually linked to at least two other sites. For example, if we add a link from A to D, the failure of a single link cannot result in the partition of the network.

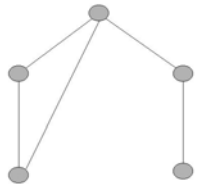


Figure 7.4 Partially connected Network

Hierarchical Networks

In a hierarchical network, the sites are organised as a tree (Figure 7.5). This organisation is commonly used for corporate networks. Individual offices are linked to the local main office. Main offices are linked to regional offices; regional offices are linked to corporate headquarters.

Each site (except the root) has a unique parent, and some (possibly zero) number of children. The basic cost of this configuration is generally less than that of the partially connected scheme. In this environment, a parent and child communicate directly. Siblings may communicate with each other only through their common parent. A message from one sibling to another must be sent up to the parent, and then down to the sibling. Similarly, cousins can communicate with each other only through their common grandparent. This configuration matches well with the generalisation that systems near each other communicate more than those that are distant. For instance, systems within a building are more likely to transfer data than those at separate installations.

If a parent site fails, then its children can no longer communicate with each other or with other processors. In general, the failure of any node (except a leaf) partitions the network into several disjoint sub trees.

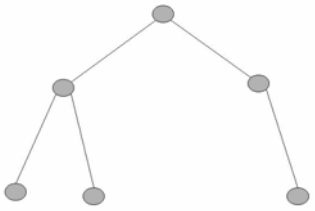


Fig 7.5 Tree Structured Network

Star Networks

In a star network, one of the sites in the system is connected to all other sites (Figure 7.6). None of the other sites are connected to any other. The basic cost of this system is linear in the number of sites. The communication cost is also low, because a message from process A to B requires at most two transfers (from A to the central site, and then from the central site B). This simple

transfer scheme, however, may not ensure speed, since the central site may become a bottleneck. Consequently, though the number of message transfers needed is low, the time required to send these messages may be high. In many star systems, therefore, the central site is completely dedicated to the message-switching task.

If the central site fails, the network is completely partitioned.

Ring Networks

In a ring network, each site is physically connected to exactly two other sites (Figure 7.7a). The ring can be either unidirectional or bi-directional. In a unidirectional architecture, a site can transmit information to only one of its neighbours. All sites must send information in the same direction. In a bi-directional architecture, a site can transmit information to both of its neighbours. The basic cost of a ring is linear in the number of sites. However, the communication cost can be high. A message from one site to another travels around the ring until it reaches its destination. In a unidirectional ring, this process could require n - 1 transfer. In a bi-directional ring, at most n/2 transfers are needed.

In a bi-directional ring, two links must fail before the network will be partitioned. In a unidirectional ring, a single site failure (or link failure) would partition the network. One remedy is to extend the architecture by providing double links, as depicted in Figure 7.7b. The IBM token ring network is a ring network.

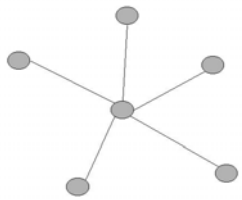


Fig 7.6 Star Network

Multi-Access Bus Networks

In a multi-access bus network, there is a single shared link (the bus). All the sites in the system are directly connected to that link, which may be organised as a straight line (Figure 7.8a) or as a ring (Figure 7.8b). The sites can communicate with each other directly through this link. The basic cost of the network is linear in the number of sites. The communication cost is quite low, unless the link becomes a bottleneck. Notice that this network topology is similar to that of the star network with a dedicated central site. The failure of one site does not affect communication among the rest of the sites. However, if the link fails, the network is partitioned completely. The ubiquitous Ethernet network, used by many institutions worldwide, is based on the multi-access bus model.

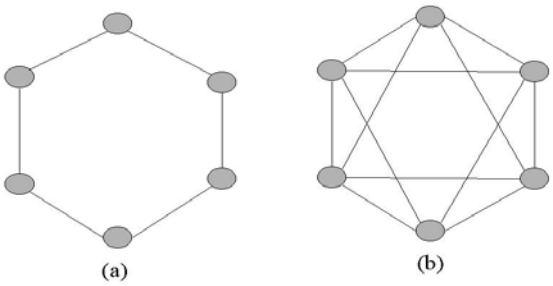


Figure 7.6 Ring Networks. (a) Single Links (b) Double Links

Hybrid Networks

It is common for networks of differing types to be connected together. For example, within a site, a multi-access bus such as Ethernet may be used, but between sites, a hierarchy may be used. Communications in such an environment can be tricky because the multiple protocols must be translated to one another and the routing of data is more complicated.

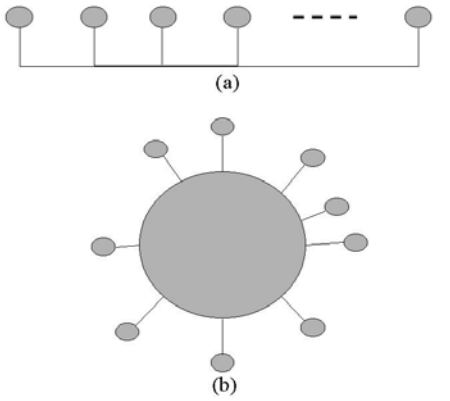


Fig 7.7 Bus Network (a) Linear Bus (b) Ring Bus

ASSIGNMENT

Q1 Explain Local Area network in brief with the help of a neat diagram.

Q2 What is the need of a wide area network? Explain in light of its structure.

Q3 Name the different topologies of a network. Which is the topology best suited for your unit, explain with reasons?