

**G.K.S.M. Govt. College,  
Tanda Urmar**

**Department  
Of  
Computer Science**



**Presented By:-  
Shaveta Sangar  
(Assist. Professor)**

viruses



# Overview

- TYPES OF INFECTION
- DEFINITIONS
- DIFFERENCE BETWEEN VIRUS AND WORM
- ORIGINS
- TYPES OF VIRUSES
- WORMS



# Overview Cont...

- MELISSA VIRUS
- I LOVE YOU VIRUS
- CODE RED (WORM)
- SYMPTOMS OF AN INFECTION
- PROTECTION MEASURES
- CONCLUSION
- REFERENCES



# Types of Infection

- VIRUSES
- E-MAIL VIRUSES
- WORMS
- TROJAN HORSES



# Viruses

A virus is a small piece of software that piggybacks on real programs.

2 main characteristics of viruses

- It must execute itself.
- It must replicate itself.



Ralf responds quickly to his computer virus...

*Ralf*

# Virus

Virus might attach itself to a program such as spreadsheet. Each time the spreadsheet program runs, the virus runs too and replicate itself.



# E-mail Viruses

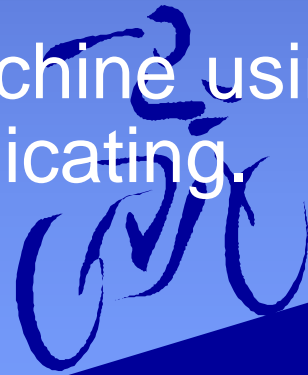
- Moves around in e-mail messages
- Usually replicate itself by automatically mailing itself to dozens of people in the victim's email address book.
- Example "MELISSA VIRUS"
- Example "I LOVE YOU VIRUS"





# WORMS

- Small piece of software that uses computer networks and security holes to replicate itself.
- Copy of the worm scans the network for another machine that has a specific security hole.
- Copy itself to the new machine using the security hole and start replicating.
- Example “CODE RED”



# Trojan Horses

- A simple computer program
- It claim to be a game
- Erase your hard disk
- No way to replicate itself.



# Difference between Virus and Worm

The difference between a worm and a virus is that a virus does not have a propagation vector. i.e., it will only effect one host and does not propagate to other hosts. Worms propagate and infect other computers. Majority of threats are actually worms that propagate to other hosts.



# Why do people do it ?

- For some people creating viruses seems to be thrill.
- Thrill of watching things blow up.



# Viruses

- Viruses show us how vulnerable we are
- A properly engineered virus can have an amazing effect on the Internet
- They show how sophisticated and interconnected human beings have become.



# Types of Viruses

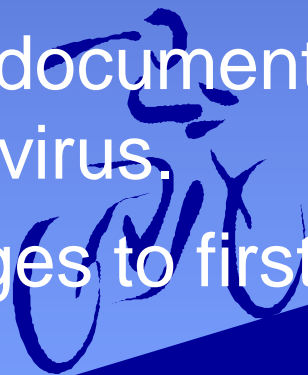
- File infector virus
  - Infect program files
- Boot sector virus
  - Infect the system area of a disk
- Master boot record virus
  - infect disks in the same manner as boot sector viruses. The difference between these two virus types is where the viral code is located.
- Multi-partite virus
  - infect both boot records and program files
- Macro virus
  - infect data files. Examples: Microsoft Office Word, Excel, PowerPoint and Access files



# Melissa Virus (March 1999)

Melissa virus spread in Microsoft Word documents sent via e-mail.

How it works ?

- Created the virus as word document
  - Uploaded to an internet newsgroup
  - Anyone who download the document and opened it would trigger the virus.
  - Send friendly email messages to first 50 people in person's address book.
- 

# Melissa Virus

Melissa Virus was the fastest spreading virus ever seen.

Forced a number of large companies to shut down their e-mail systems.





# I Love You Virus (May,2000)

- Contained a piece of code as an attachment.
- Double Click on the attachment triggered the code.
- Sent copies of itself to everyone in the victim's address book
- Started corrupting files on the victim's machine.



# Code Red (Worm)

- Code Red made huge headlines in 2001
- It slowed down internet traffic when it began to replicate itself.
- Each copy of the worm scanned the internet for Windows NT or Windows 2000 that don't have security patch installed.
- Each time it found an unsecured server, the worm copied itself to that server.

# Code Red Worm

Designed to do three things

- ❖ Replicate itself for the first 20 days of each month.
- ❖ Replace web pages on infected servers with a page that declares “Hacked by Chinese”
- ❖ Launch a concerted attack on the White House Web server



# Symptoms of Infection

- Programs take longer to load than normal.
- Computer's hard drive constantly runs out of free space.
- The floppy disk drive or hard drive runs when you are not using it.
- New files keep appearing on the system and you don't know where it come from.



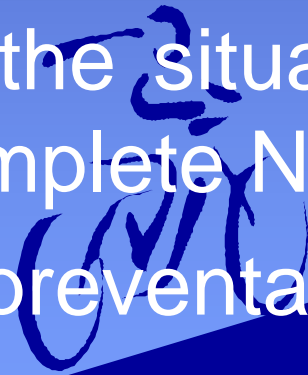
# Symptoms of Infection Cont..

- Strange sounds or beeping noises come from the computer.
- Strange graphics are displayed on your computer monitor.
- Unable to access the hard drive when booting from the floppy drive.
- Program sizes keep changing.



# Protection

- TO protect yourself you need to be “Proactive” about Security issues. Being reactive won't solve anything; Specially at crunch time and deadlines!! In matter of fact it can make the problem much more complex to solve, and the situation much worse, resulting in a complete Nightmare!!
- Best Measures are the preventative ones.



# Conclusion

Be aware of the new infections out there.

Take precaution measures.

Always backup your data.

Keep up-to-date on new Anti virus software.

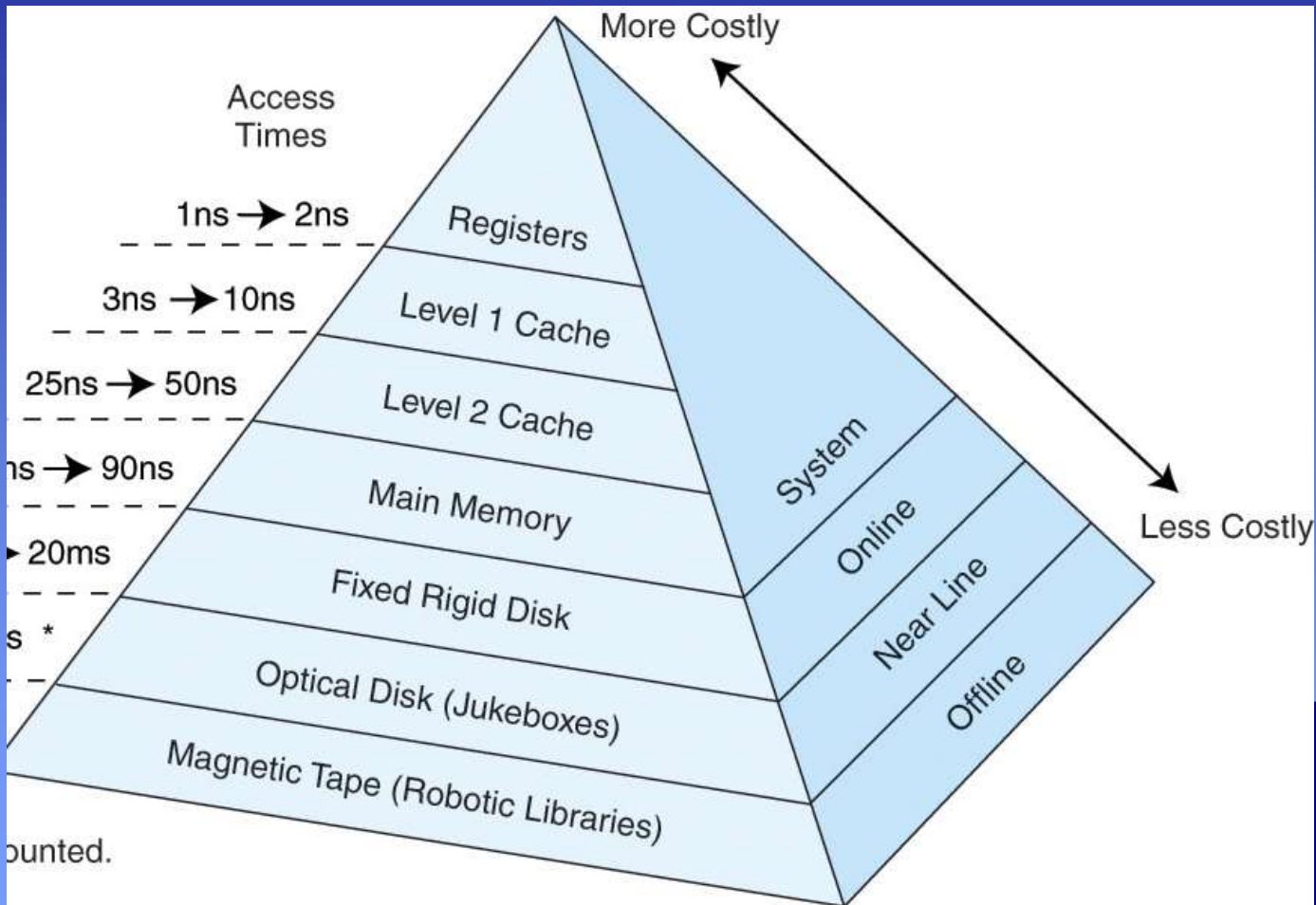
Simply avoid programs from unknown sources.



# MEMORY

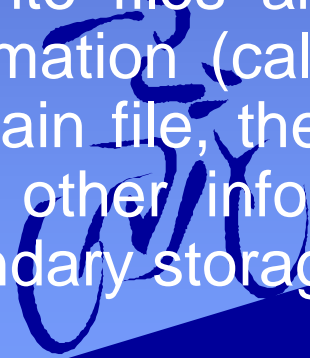






# Secondary Memory

- The computer usually uses its input/output channels to access secondary storage and transfers the desired data using intermediate area in primary storage. Secondary storage does not lose the data when the device is powered down—it is non-volatile. Per unit, it is typically also an order of magnitude less expensive than primary storage.
- The secondary storage is often formatted according to a file system format, which provides the abstraction necessary to organize data into files and directories, providing also additional information (called metadata) describing the owner of a certain file, the access time, the access permissions, and other information. Hard disk are usually used as secondary storage.



# Memory Types

## I. Secondary Memory

## II. Primary Memory

### a) RAM

- i. SRAM
- ii. DRAM

### b) ROM

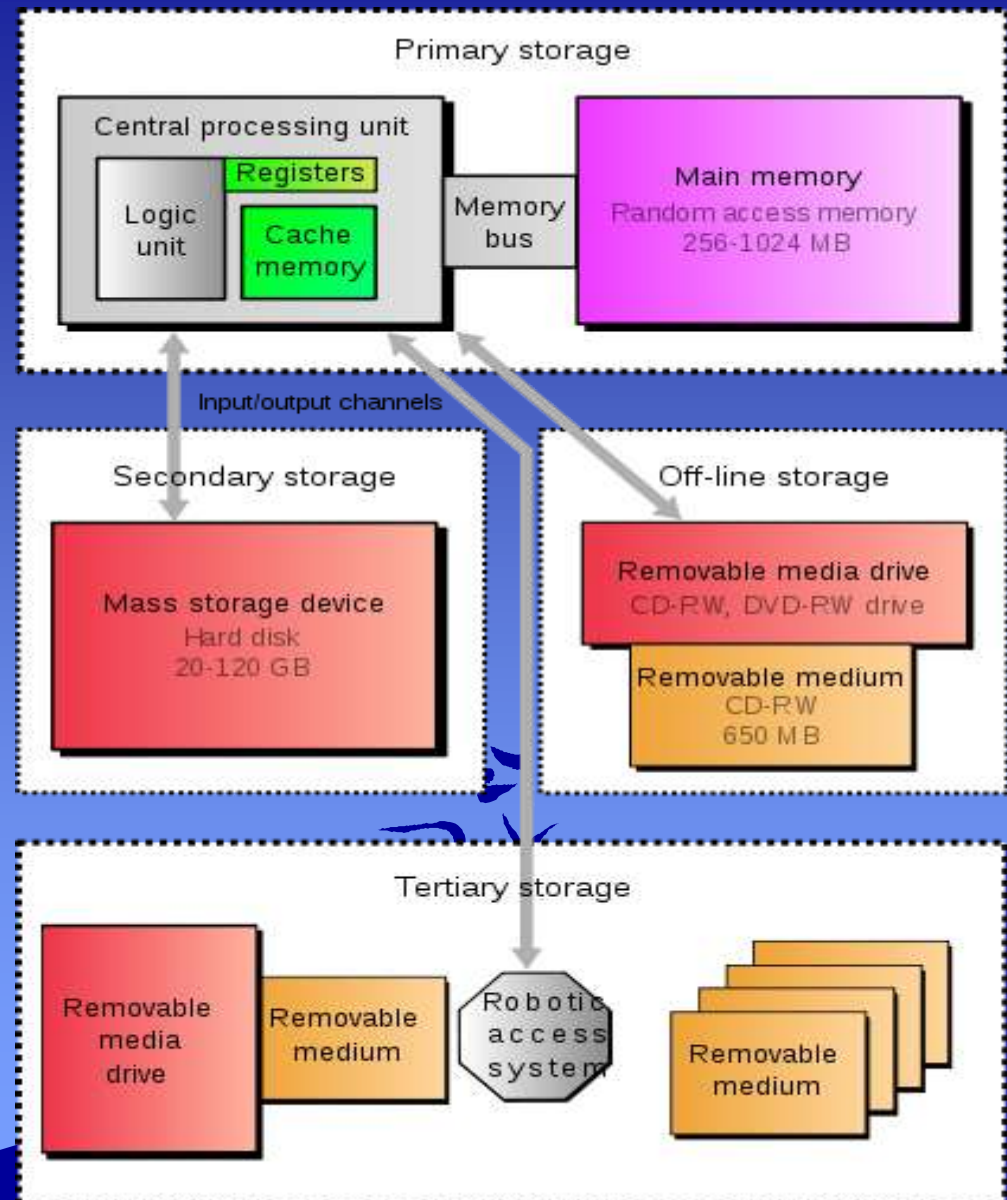
- i. PROM
- ii. EPROM

### c) Hybrid

- i. EEPROM
- ii. NVRAM
- iii. Flash Memory

### d) Cache Memory

### e) Virtual Memory

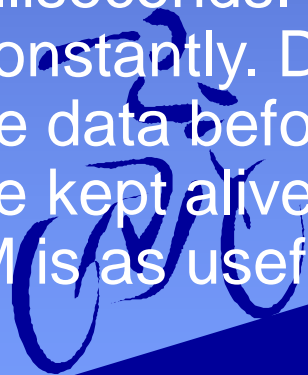


# Primary Memory

- Primary storage (or main memory or internal memory), often referred to simply as memory, is the only one directly accessible to the CPU. The CPU continuously reads instructions stored there and executes them as required.
- Main memory is directly or indirectly connected to the CPU via a *memory bus*. It is actually two buses (not on the diagram): an address bus and a data bus. The CPU firstly sends a number through an address bus, a number called memory address, that indicates the desired location of data. Then it reads or writes the data itself using the data bus.
- It is divided into RAM and ROM.


# RAM

The RAM family includes two important memory devices: static RAM (SRAM) and dynamic RAM (DRAM). The primary difference between them is the lifetime of the data they store.

- 1) SRAM retains its contents as long as electrical power is applied to the chip. If the power is turned off or lost temporarily, its contents will be lost forever.
  - 2) DRAM, on the other hand, has an extremely short data lifetime-typically about four milliseconds. This is true even when power is applied constantly. DRAM controller is used to refresh the data before it expires, the contents of memory can be kept alive for as long as they are needed. So DRAM is as useful as SRAM after all.
- 

# Types of RAM

**Double Data Rate synchronous dynamic random access memory** or also known as **DDR1 SDRAM** is a class of memory integrated circuits used in computers. The interface uses double pumping (transferring data on both the rising and falling edges of the clock signal) to lower the clock frequency. One advantage of keeping the clock frequency down is that it reduces the signal integrity requirements on the circuit board connecting the memory to the controller.



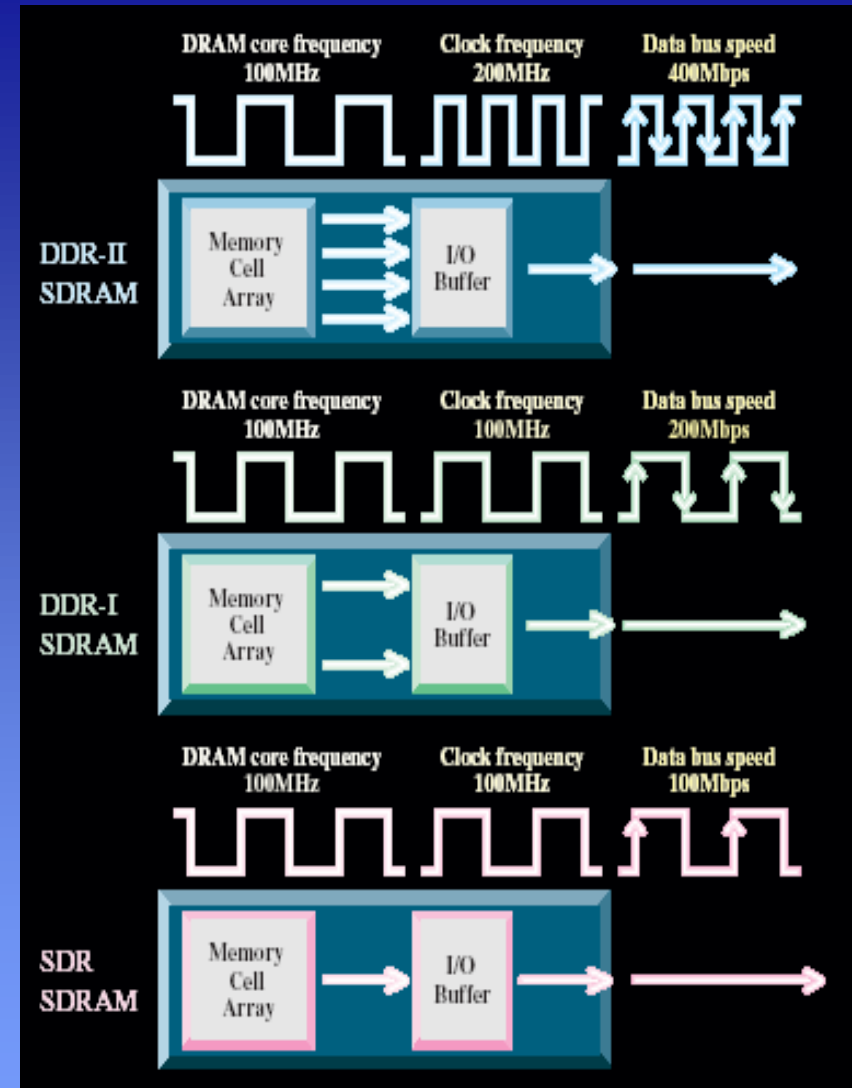
# DDR2, DDR and SDRAM

DDR2 memory is fundamentally similar to DDR SDRAM. Still, while DDR SDRAM can transfer data across the bus two times per clock, DDR2 SDRAM can perform four transfers per clock. DDR2 uses the same memory cells, but doubles the bandwidth by using the multiplexing technique.

The DDR2 memory cell is still clocked at the same frequency as DDR SDRAM and SDRAM cells, but the frequency of the input/output buffers is higher with DDR2 SDRAM (as shown in Fig. on next Slide). The bus that connects the memory cells with the buffers is twice wider compared to DDR. Thus, the I/O buffers perform multiplexing: the data is coming in from the memory cells along a wide bus and is going out of the buffers on a bus of the same width as in DDR SDRAM, but of a twice bigger frequency. This allows to increase the memory bandwidth without increasing the operational frequency.

The interface uses double pumping (transferring data on both the rising and falling edges of the clock signal) to lower the clock frequency.

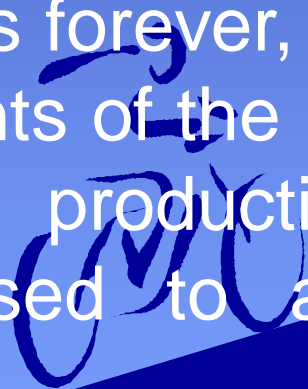
One advantage of keeping the clock frequency down is that it reduces the signal integrity requirements on the circuit board connecting the memory to the controller.





# Types of ROM

Memories in the ROM family are distinguished by the methods used to write new data to them (usually called programming), and the number of times they can be rewritten. This classification reflects the evolution of ROM devices from hardwired to programmable to erasable-and-programmable. A common feature is their ability to retain data and programs forever, even during a power failure. The contents of the ROM had to be specified before chip production, so the actual data could be used to arrange the transistors inside the chip.



# PROM

One step up from the masked ROM is the PROM (programmable ROM), which is purchased in an unprogrammed state. If you were to look at the contents of an unprogrammed PROM, the data is made up entirely of 1's. The process of writing your data to the PROM involves a special piece of equipment called a device programmer. The device programmer writes data to the device one word at a time by applying an electrical charge to the input pins of the chip. Once a PROM has been programmed in this way, its contents can never be changed. If the code or data stored in the PROM must be changed, the current device must be discarded. As a result, PROMs are also known as one-time programmable (OTP) devices.

# EPRM

An EPROM (erasable-and-programmable ROM) is programmed in exactly the same manner as a PROM. However, EPROMs can be erased and reprogrammed repeatedly. To erase an EPROM, you simply expose the device to a strong source of ultraviolet light. (A window in the top of the device allows the light to reach the silicon.) By doing this, you essentially reset the entire chip to its initial-unprogrammed-state. Though more expensive than PROMs, their ability to be reprogrammed makes EPROMs an essential part of the software development and testing process.

# Hybrid types

As memory technology has matured in recent years, the line between RAM and ROM has blurred. Now, several types of memory combine features of both. These devices do not belong to either group and can be collectively referred to as hybrid memory devices. Hybrid memories can be read and written as desired, like RAM, but maintain their contents without electrical power, just like ROM. Two of the hybrid devices, EEPROM and flash, are descendants of ROM devices. These are typically used to store code. The third hybrid, NVRAM, is a modified version of SRAM. NVRAM usually holds persistent data.



□ **EEPROMS** are electrically-erasable-and-programmable. Internally, they are similar to EPROMs, but the erase operation is accomplished electrically, rather than by exposure to ultraviolet light. Any byte within an EEPROM may be erased and rewritten. Once written, the new data will remain in the device forever-or at least until it is electrically erased. The primary tradeoff for this improved functionality is higher cost, though write cycles are also significantly longer than writes to a RAM. So you wouldn't want to use an EEPROM for your main system memory.

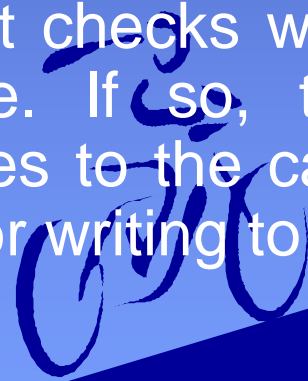
□ **Flash memory** combines the best features of the memory devices described thus far. Flash memory devices are high density, low cost, nonvolatile, fast (to read, but not to write), and electrically reprogrammable. These advantages are overwhelming and, as a direct result, the use of flash memory has increased dramatically in embedded systems. From a software viewpoint, flash and EEPROM technologies are very similar. The major difference is that flash devices can only be erased one sector at a time, not byte-by-byte. Typical sector sizes are in the range 256 bytes to 16KB. Despite this disadvantage, flash is much more popular than EEPROM and is rapidly displacing many of the ROM devices as well.



□ The third member of the hybrid memory class is **NVRAM** (non-volatile RAM). Nonvolatility is also a characteristic of the ROM and hybrid memories discussed previously. However, an NVRAM is physically very different from those devices. An NVRAM is usually just an SRAM with a battery backup. When the power is turned on, the NVRAM operates just like any other SRAM. When the power is turned off, the NVRAM draws just enough power from the battery to retain its data. NVRAM is fairly common in embedded systems. However, it is expensive-even more expensive than SRAM, because of the battery-so its applications are typically limited to the storage of a few hundred bytes of system-critical information that can't be stored in any better way.

# Cache Memory

- A CPU cache is a cache used by the central processing unit of a computer to reduce the average time to access memory. The cache is a smaller, faster memory which stores copies of the data from the most frequently used main memory locations. As long as most memory accesses are cached memory locations, the average latency of memory accesses will be closer to the cache latency than to the latency of main memory.
- When the processor needs to read from or write to a location in main memory, it first checks whether a copy of that data is in the cache. If so, the processor immediately reads from or writes to the cache, which is much faster than reading from or writing to main memory





# Locality of Reference

- The better the hit rate for level 0, the better off we are
  - Similarly, if we use 2 caches, we want the hit rate of level 1 to be as high as possible
  - We want to implement the memory hierarchy to follow *Locality of Reference*
    - accesses to memory will generally be near recent memory accesses and those in the near future will be around this current access



– Three forms of locality:

- Temporal locality – recently accessed items tend to be accessed again in the near future (local variables, instructions inside a loop)
- Spatial locality – accesses tend to be clustered (accessing  $a[i]$  will probably be followed by  $a[i+1]$  in the near future)
- Sequential locality – instructions tend to be accessed sequentially

– How do we support locality of reference?

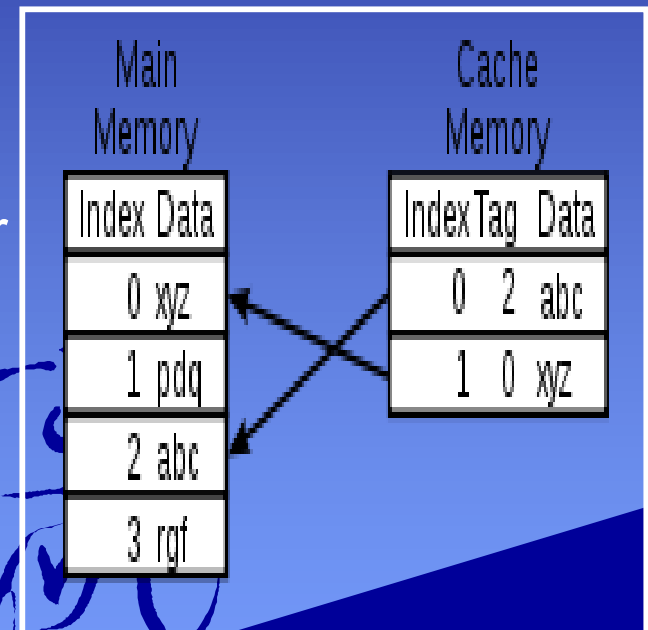
- If we bring something into cache, bring in neighbors as well
- Keep an item in the cache for awhile as we hope to keep using it



# Cache Memory

The diagram on the right shows two memories. Each location in each memory has a datum (a *cache line*), which in different designs ranges in size from 8 to 512 bytes. The size of the cache line is usually larger than the size of the usual access requested by a CPU instruction, which ranges from 1 to 16 bytes.

Each location in each memory also has an index, which is a unique number used to refer to that location. The index for a location in main memory is called an address. Each location in the cache has a tag that contains the index of the datum in main memory that has been cached. In a CPU's data cache these entries are called *cache lines* or *cache blocks*.



# Cache and Memory Organization

- Group memory locations into lines (or refill lines)
  - For instance, 1 line might store 16 bytes or 4 words
    - The line size varies architecture-to-architecture
  - All main memory addresses are broken into two parts
    - the line #
    - the location in the line
      - If we have 256 Megabytes, word accessed, with word sizes of 4, and 4 words per line, we would have 16,777,216 lines so our 26 bit address has 24 bits for the line number and 2 bits for the word in the line

- The cache has the same organization but there are far fewer line numbers (say 1024 lines of 4 words each)
  - So the remainder of the address becomes the *tag*
    - The tag is used to make sure that the line we want is the line we found

Block	Tag	Data	Valid
0	00000000	words A, B, C,...	1
1	11110101	words L, M, N,...	1
2	-----		0
3	-----		0

The valid bit is used to determine if the given line has been modified or not (is the line in memory still valid or outdated?)

# Types of Cache

- Direct-mapped – each entry in memory has 1 specific place where it can be placed in cache
  - this is a cheap and easy cache to implement (and also fast), but since there is no need for a replacement strategy it has the poorest hit rate
- Associative – any memory item can be placed in any cache line
  - this cache uses associative memory so that an entry is searched for in parallel – this is expensive and tends to be slower than a direct-mapped cache, however, because we are free to place an entry anywhere, we can use a replacement strategy and thus get the best hit rate

- Set-associative – a compromise between these two extremes
  - by grouping lines into sets so that a line is mapped into a given set, but within that set, the line can go anywhere
  - a replacement strategy is used to determine which line within a set should be used, so this cache improves on the hit rate of the direct-mapped cache
  - while not being as expensive or as slow as the associative cache

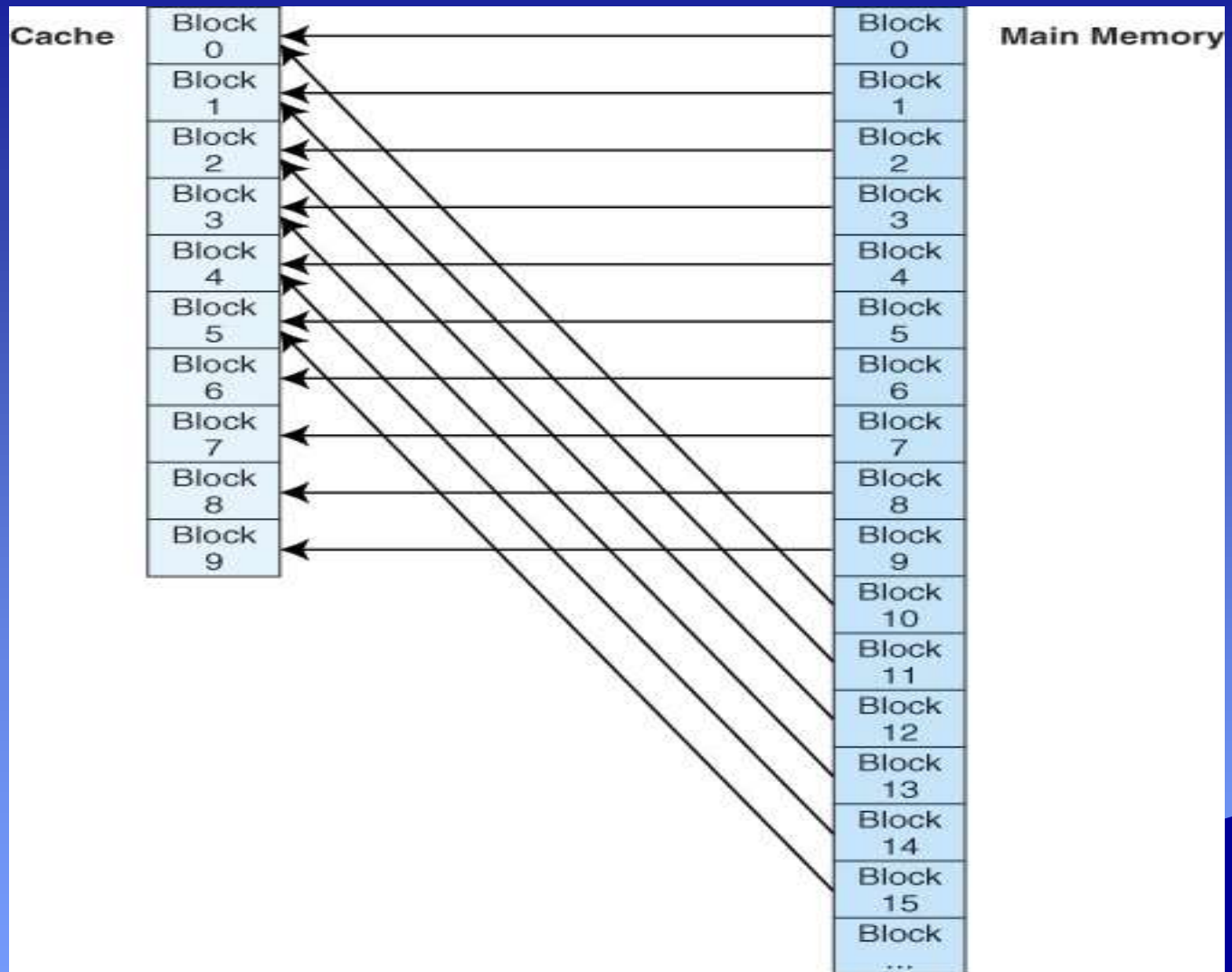


# Direct Mapped Cache

- Assume  $m$  refill lines
  - A line  $j$  in memory will be found in cache at location  $j \bmod m$ 
    - Since each line has 1 and only 1 location in cache, there is no need for a replacement strategy
  - This yields poor hit rate but fast performance (and cheap)
  - All addresses are broken into 3 parts
    - a line number (to determine the line in cache)
    - a word number
    - the rest is the tag – compare the tag to make sure you have the right line







# Associative Cache

- Any line in memory can be placed in any line in cache
  - No line number portion of the address, just a tag and a word within the line
  - Because the tag is longer, more tag storage space is needed in the cache, so these caches need more space and so are more costly
- All tags are searched simultaneously using “associative memory” to find the tag requested
  - This is both more expensive and slower than direct-mapped caches but, because there are choices of where to place a new line, associative caches require a replacement strategy which might require additional hardware to implement

Tag 22 bit

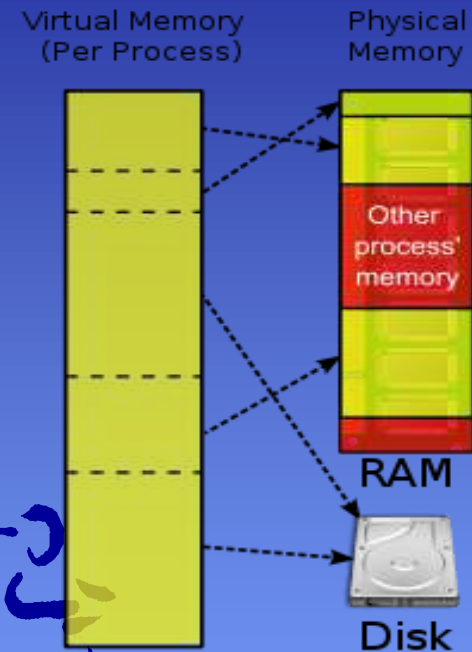
Word  
2 bit



# Virtual Memory

It is a computer system technique which gives an application program the impression that it has contiguous working memory (an address space), while in fact it may be physically fragmented and may even overflow on to disk storage.

computer operating systems generally use virtual memory techniques for ordinary applications, such as word processors, spreadsheets, multimedia, players accounting, etc., except where the required hardware support (memory management unit) is unavailable or insufficient.



Handwritten signature or initials in blue ink.

# Memory is used in

- ✓ Computer
- ✓ Mobile
- ✓ Printer
- ✓ Digital Camera
- ✓ CD/DVD Player
- ✓ Many other appliances like TV, Washing Machine, Oven, Digital Diaries etc.



# The Paging Process

- When the CPU generates a memory address, it is a logical (or virtual) address
  - The first address of a program is 0, so the logical address is merely an offset into the program or into the data segment
    - For instance, address 25 is located 25 from the beginning of the program
    - But 25 is not the physical address in memory, so the logical address must be translated (or *mapped*) into a physical address

- Assume memory is broken into fixed size units known as frames (1 page fits into 1 frame)
  - We know the logical address as its page # and the offset into the page
- We have to translate the page # into the frame # (that is, where is that particular page currently be stored in memory – or is it even in memory?)
  - Thus, the mapping process for paging means finding the frame # and replacing the page # with it

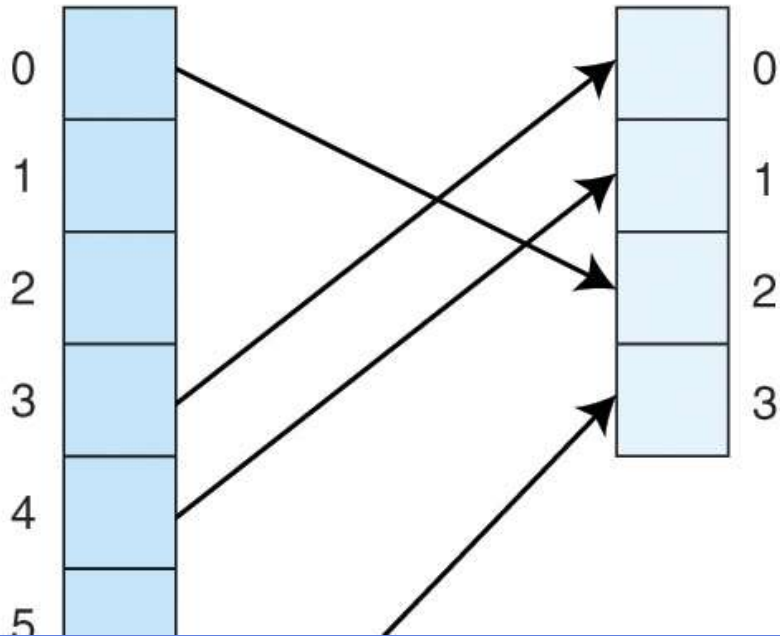


Virtual Memory

Physical Memory

Page

Page Table



	Frame #	Valid Bit
0	2	1
1	-	0
2	-	0
3	0	1
4	1	1
5	-	0
6	-	0
7	3	1





- Here, we have a process of 8 pages but only 4 physical frames in memory – therefore we must place a page into one of the available frames in memory whenever a page is needed
- At this point in time, pages 0, 3, 4 and 7 have been moved into memory at frames 2, 0, 1 and 3 respectively
- This information (of which page is stored in which frame) is stored in memory in a location known as the Page Table. The page table also stores whether the given page has been modified (the valid bit – much like our cache)

# Page Faults

- Just as cache is limited in size, so is main memory – a process is usually given a limited number of frames
- What if a referenced page is not currently in memory?
  - The memory reference causes a *page fault*
    - The page fault requires that the OS handle the problem
  - The process' status is saved and the CPU switches to the OS
  - The OS determines if there is an empty frame for the referenced page, if not, then the OS uses a replacement strategy to select a page to discard

if that page is dirty, then the page must be written to disk instead of discarded

- The OS locates the requested page on disk and loads it into the appropriate frame in memory
  - The page table is modified to reflect the change
- Page faults are time consuming because of the disk access – this causes our effective memory access time to deteriorate badly!



# PHYSICAL COMPONENTS OF COMPUTER SYSTEM



# Computer System



- **Input** - getting data into the computer
- Input Devices
  - – enable users to get data into the computer for processing



# Processing

transforming data  
into information

Microprocessor is  
simply a small  
processor fabricat  
ed on a chip of  
silicon

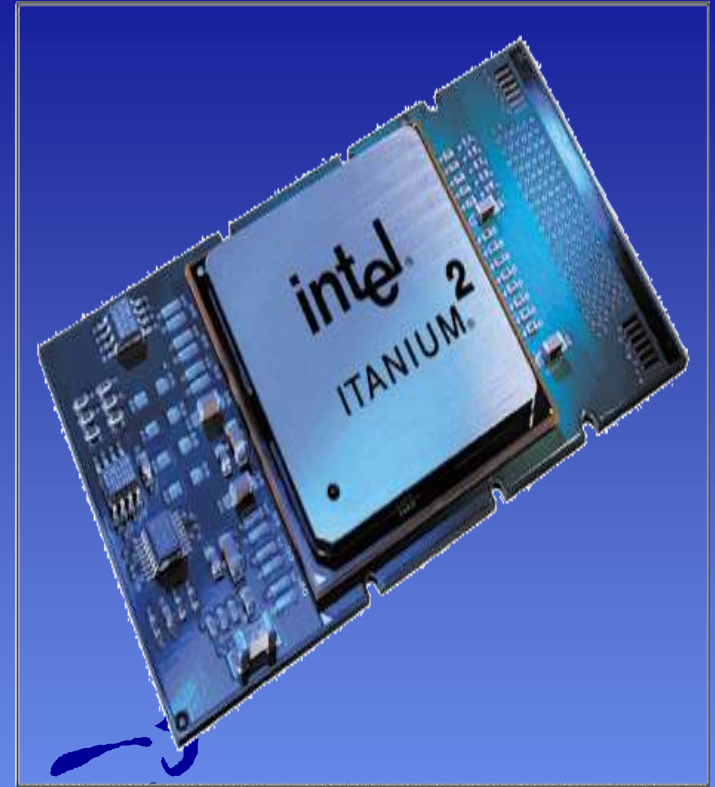


Figure 5-9

The Itanium 2

Output –  
displaying the  
information



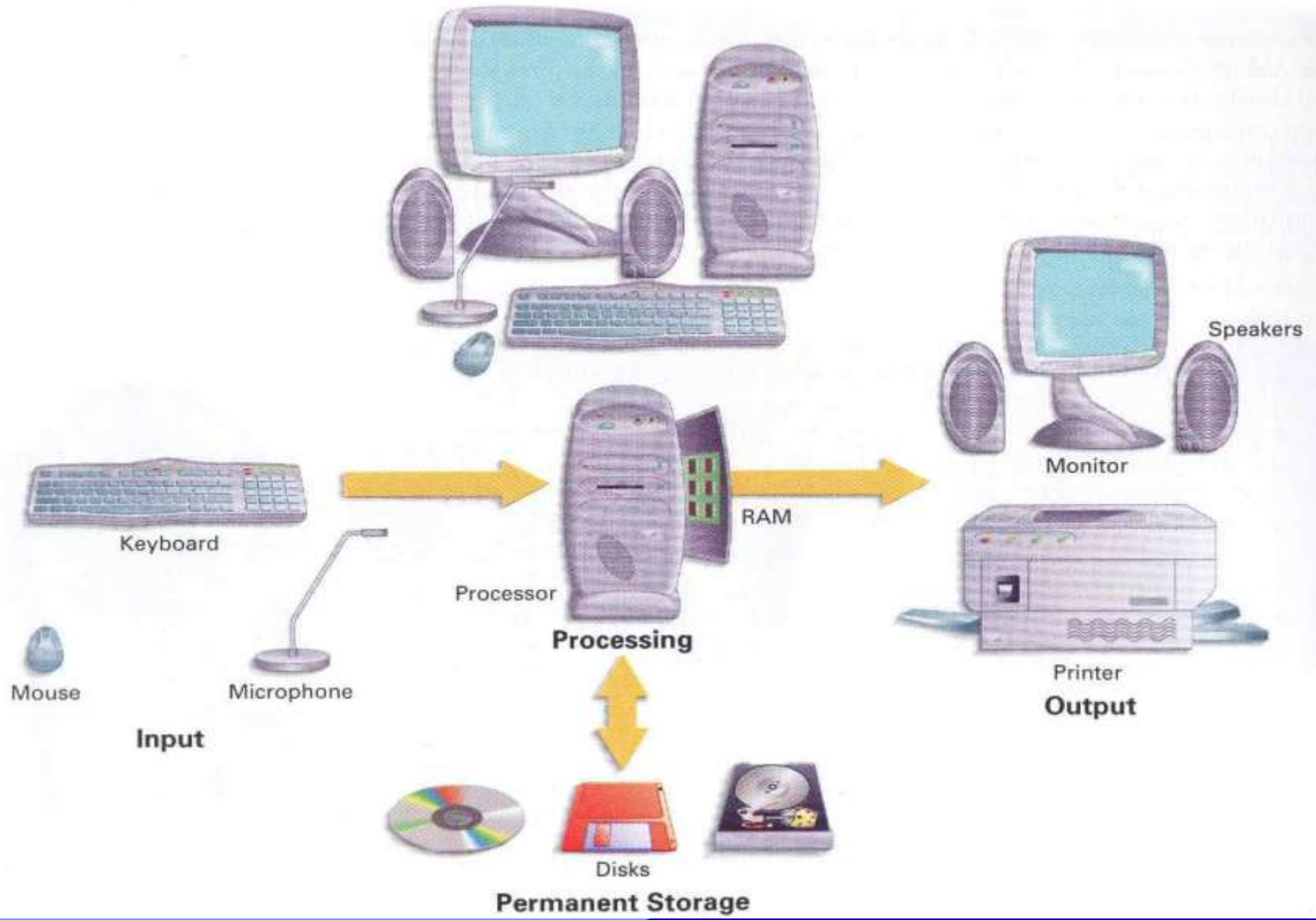
# OUTPUT DEVICE

- Output devices – enable users to see and produce processed informations



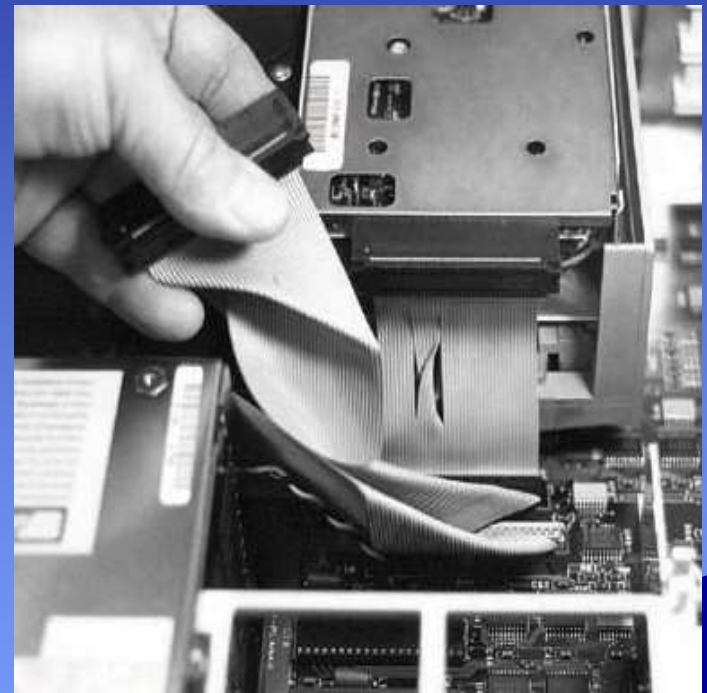
© 2000



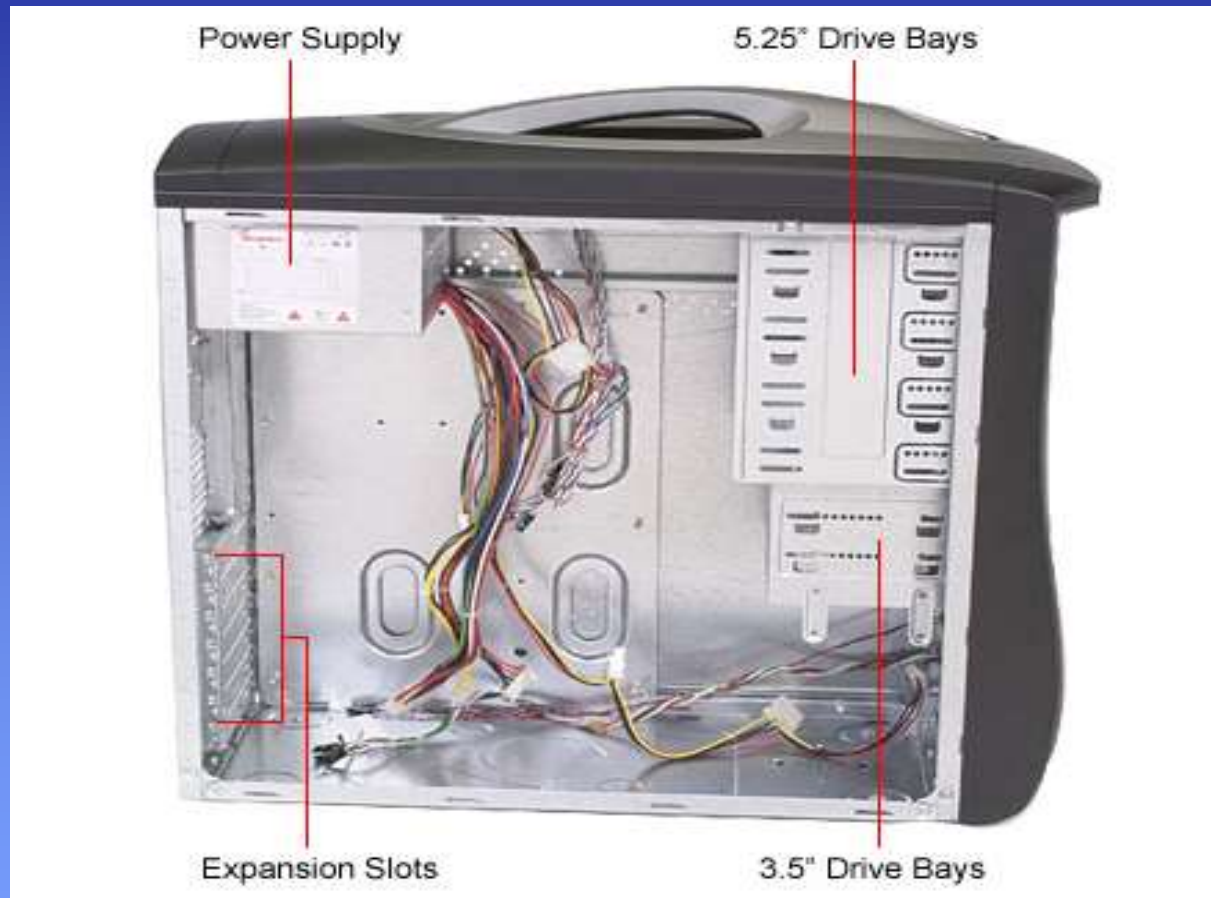


# Moving Information Within the Computer

- Bits that compose a word are passed in parallel from place to place.
  - **Ribbon cables:**
    - Consist of several wires, molded together.
    - One wire for each bit of the word or byte.
    - Additional wires coordinate the activity of moving information.
    - Each wire sends information in the form of a **voltage**



# Inside the Computer Case



# Video Card

- Connects the computer to the monitor. It is a circuit board attached to the motherboard that contains the memory and other circuitry necessary to send information to the monitor for display on screen.



# CD Rom Drive

- The drive that plays CDs and reads data that has been stored on



# CD

- Compact Disk – A type of optical storage device



# Floppy Disk Drive

- A device that holds a removable floppy disk when in use; read/write head data to the diskette.



# Hard Disk

- Magnetic storage device in the computer





# RAM

- Random Access Memory

RAM is a computer's temporary memory. It consists of memory chips on the motherboard near the processor. RAM stores data and programs while they are being used by the computer. RAM is volatile memory, meaning it loses its data when power is lost.



# Printer

- An output device that produces a hard copy on paper. It gives information in a hard copy form.



# Microphone

- Allows the user to record computer.

