Introduction to Computing

Lecture 5: Computer Hardware & Operating System

Objective

After studying this chapter, the student should be able to:

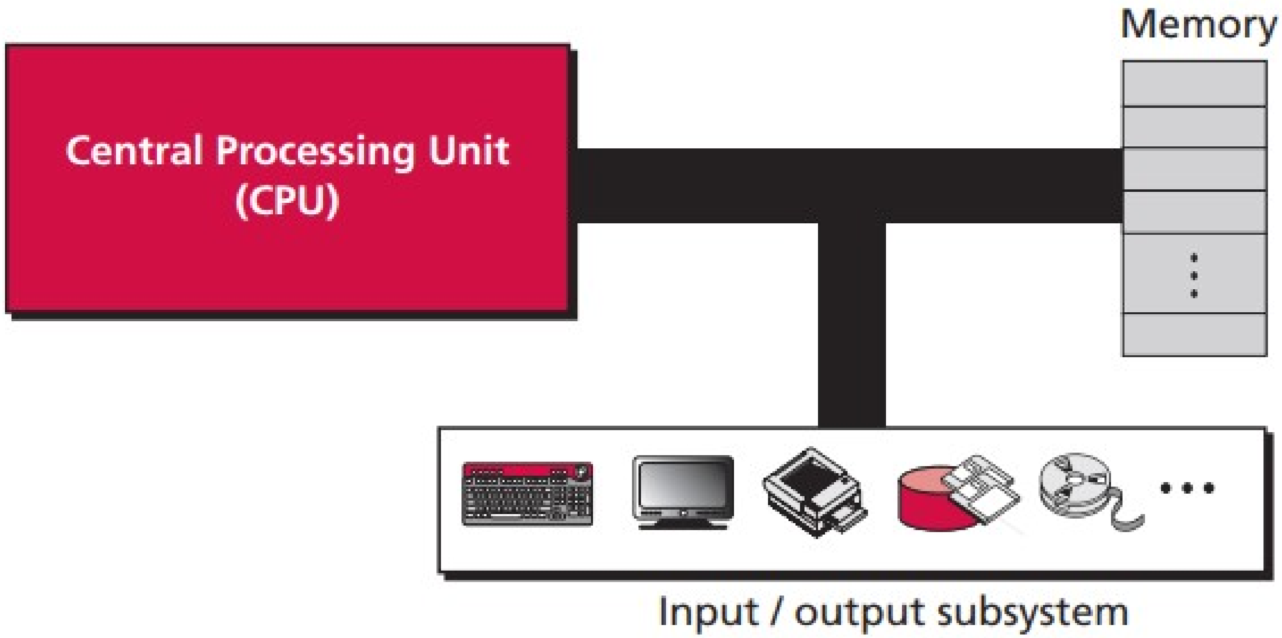
- ❑ Hardware
  - List the three subsystems of a computer.
  - Describe the role of the central processing unit (CPU) in a computer
  - Describe the main memory and its addressing space
  - Distinguish between main memory and cache memory
  - Define the input/output subsystem
- ❑ Operating Systems
  - Understand the role of the operating system in a computer system
  - Give a definition of an operating system
  - Understand the process of bootstrapping to load the operating system into memory
  - List the components of an operating system

Computer Hardware

Computer Hardware

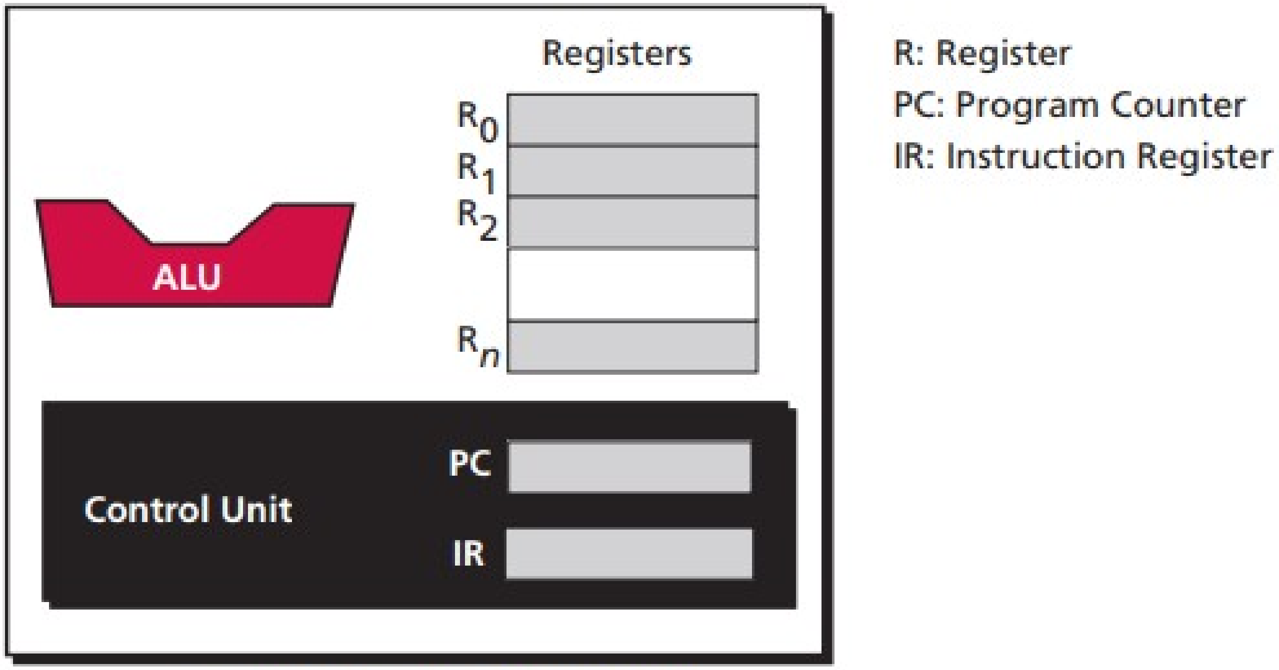
❖ MAIN PARTS of COMPUTER

- The parts that make up a computer into three broad categories or subsystems: the central processing unit (CPU), the main memory, and the input/output subsystem.



Central Processing Unit

- The **central processing unit (CPU)** performs operations on data.
- Three parts: an arithmetic logic unit (ALU), a control unit, and a set of registers



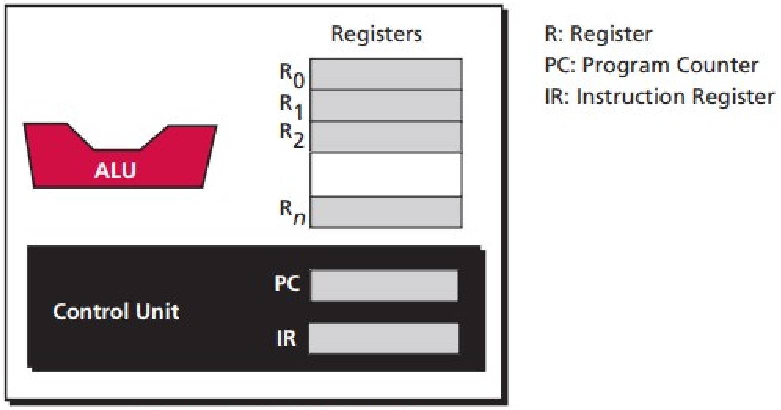
Central Processing Unit (CPU)

The arithmetic logic unit (ALU)

- The **arithmetic logic unit (ALU)** performs logic, shift, and arithmetic operations on data.
- ***Logic operations:*** NOT, AND, OR, and XOR. These operations treat the input data as bit patterns and the result of the operation is also a bit pattern.
- ***Shift operations:*** Logical shift operations and arithmetic shift operations.
  - Logical shift operations are used to shift bit patterns to the left or right
  - Arithmetic operations are applied to integers
  - Their main purpose is to divide or multiply integers by two
- ***Arithmetic operation***
  - Some arithmetic operations on integers and reals

Registers

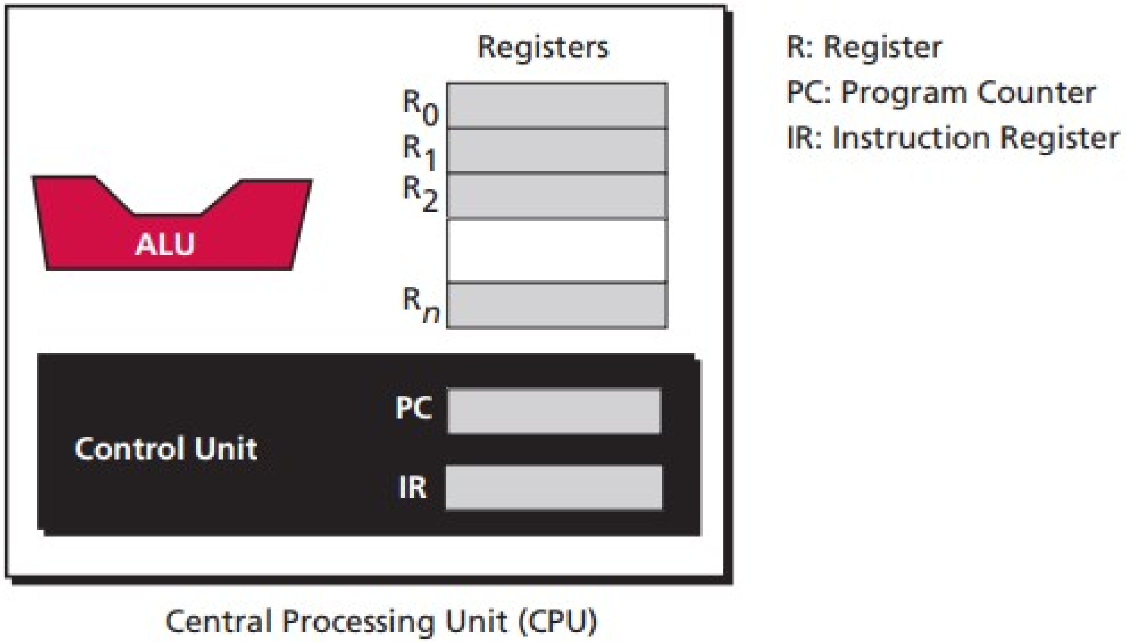
- **Registers** are fast stand-alone storage locations that hold data temporarily. Multiple registers are needed to facilitate the operation of the CPU
- ***Data registers:*** Computers use dozens of registers inside the CPU to speed up their operations. These require several registers to hold the intermediate results.
- ***Instruction registers***
  - Today computers store not only data, but also programs, in their memory.
  - The CPU is responsible for fetching instructions one by one from memory, storing them in the **instruction register** (IR), decoding them, and executing them.



Central Processing Unit (CPU)

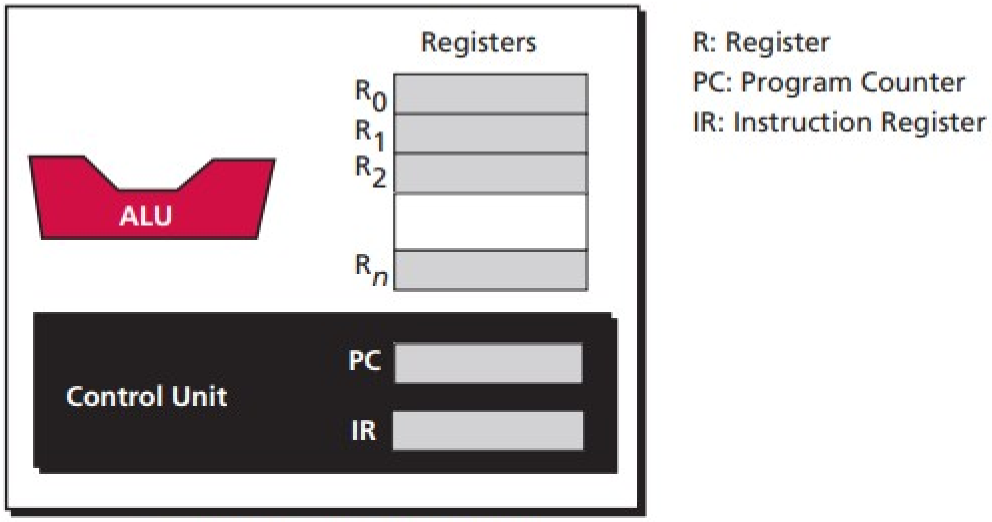
Program counter

- Another common register in the CPU is the **program counter**
- The program counter keeps track of the instruction currently being executed.
- After execution of the instruction, the counter is incremented to point to the address of the next instruction in memory.



Control Unit

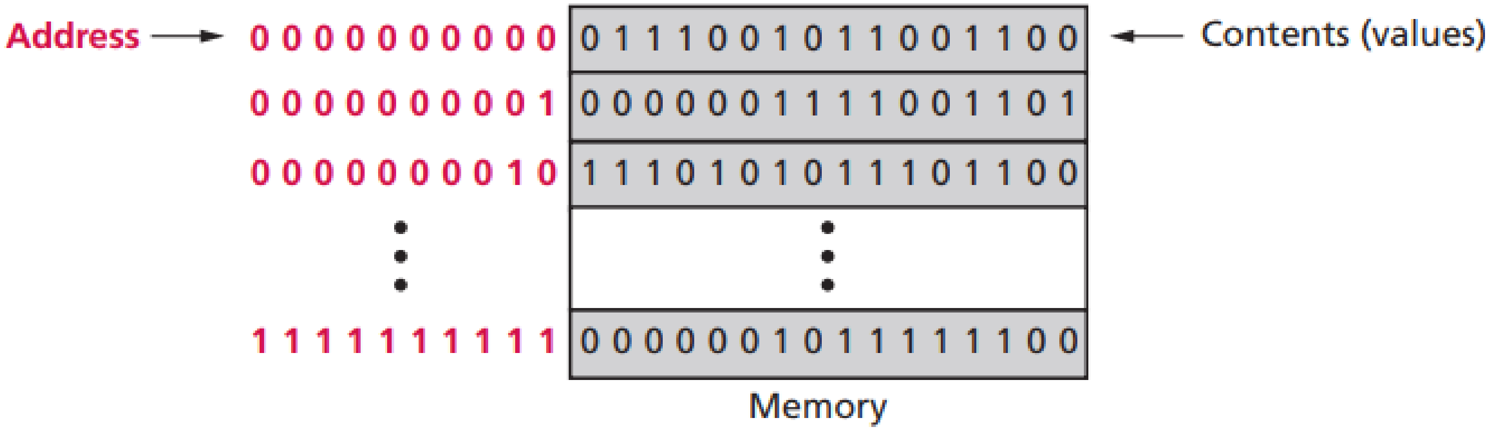
- The third part of any CPU is the control unit.
- The **control unit** controls the operation of each subsystem.
- Controlling is achieved through signals sent from the control unit to other subsystems



Central Processing Unit (CPU)

Main Memory

- **Main memory** consists of a collection of storage locations, each with a unique identifier, called an *address*.
- Data is transferred to and from memory in groups of bits called *words*.
- A word can be a group of 8 bits, 16 bits, 32 bits, or 64 bits (and growing).
- The word is 8 bits, it is referred to as a *byte*. The term ‘byte’ is so common in computer science that sometimes a 16-bit word is referred to as a 2-byte word, or a 32-bit word is referred to as a 4-byte word.



Address Space

- To access a word in memory requires an identifier.
- Although programmers use a name to identify a word (or a collection of words), at the hardware level each word is identified by an address.
- The total number of uniquely identifiable locations in memory is called the **address space**.
- For example, a memory with 64 kilobytes and a word size of 1 byte has an address space that ranges from 0 to 65535.

Table 5.1 Memory units

Unit	*Exact Number of Bytes*	*Approximation*
kilobyte	2^{10} (1024) bytes	10^3 bytes
megabyte	2^{20} (1 048 576) bytes	10^6 bytes
gigabyte	2^{30} (1 073 741 824) bytes	10^9 bytes
terabyte	2^{40} bytes	10^{12} bytes

Examples

Memory addresses are defined using unsigned binary integers.

Example 5.1

A computer has 32 MB (megabytes) of memory. How many bits are needed to address any single byte in memory?

Solution

The memory address space is 32 MB, or 2^{25} ($2^5 \times 2^{20}$). This means that we need $\log_2 2^{25}$, or 25 bits, to address each byte.

Example 5.2

A computer has 128 MB of memory. Each word in this computer is eight bytes. How many bits are needed to address any single word in memory?

Solution

The memory address space is 128 MB, which means 2^{27}. However, each word is eight (2^3) bytes, which means that we have 2^{24} words. This means that we need $\log_2 2^{24}$, or 24 bits, to address each word.

Memory Types: RAM and ROM

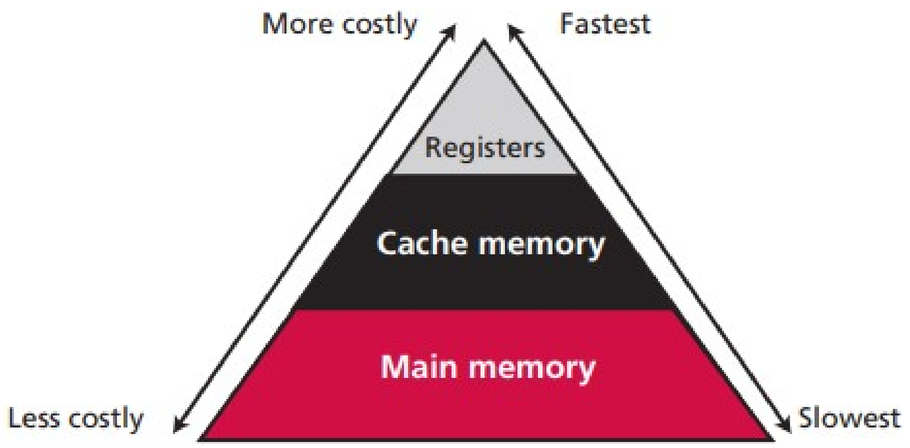
- ***RAM:*** **Random access memory (RAM)** makes up most of the main memory in a computer.
- A data item can be accessed randomly—using the address of the memory location—without the need to access all data items located before it.
- All information in RAM is erased if you turn off the computer or if there is a power outage.
- RAM technology is divided into two broad categories: SRAM and DRAM.
  - **Static RAM (SRAM)** technology uses traditional *flip-flop gates* to hold data. The gates hold their state (0 or 1), which means that data is stored as long as the power is on and there is no need to refresh memory locations. SRAM is fast but expensive.
  - **Dynamic RAM (DRAM)** technology uses capacitors, electrical devices that can store energy, for data storage. If a capacitor is charged, the state is 1; if it is discharged, the state is 0. Because a capacitor loses some of its charge with time, DRAM memory cells need to be refreshed periodically. DRAMs are slow but inexpensive.

ROM

- The contents of **read-only memory (ROM)** are written by the manufacturer, and the CPU can read from, but not write to, ROM.
- Contents are not lost if turn off the computer. It is used for programs or data that must not be erased or changed even if you turn off the computer.
- ***PROM***
  - One variation of ROM is **programmable read-only memory (PROM)**. This type of memory is blank when the computer is shipped. The user of the computer, with some special equipment, can store programs on it.
  - When programs are stored, it behaves like ROM and cannot be overwritten. This allows a computer user to store specific programs in PROM
- ***EPROM***
  - A variation of PROM is **erasable programmable read-only memory (EPROM)**. It can be programmed by the user, but can also be erased with a special device that applies ultraviolet light. To erase EPROM memory requires physical removal and reinstallation of the EPROM.
- ***EEPROM***
  - A variation of EPROM is **electrically erasable programmable read-only memory (EEPROM)**. EEPROM can be programmed and erased using electronic impulses without being removed from the computer.

Memory Hierarchy

- ❖ Computer users need a lot of memory, especially memory that is very fast and inexpensive.
- ❖ This demand is not always possible to satisfy—very fast memory is usually not cheap
- ❖ The solution is hierarchical levels of memory.
    - o Use a very small amount of costly high-speed memory where speed is crucial. The registers inside the CPU are of this type.
    - o Use a moderate amount of medium-speed memory to store data that is accessed often. *Cache memory* is of this type.
    - o Use a large amount of low-speed memory for data that is accessed less often. Main memory is of this type



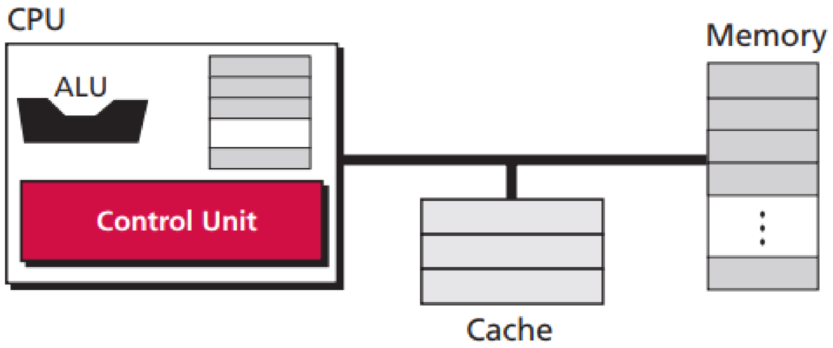
Cache Memory

❖ **Cache memory** is faster than main memory but slower than the CPU and its registers.

❖ Cache memory, which is normally small in size, is placed between the CPU and main memory

Cache memory at any time contains a copy of a portion of main memory. When the CPU needs to access a word in main memory, it follows this procedure:

1. The CPU checks the cache.
2. If the word is there, it copies the word: if not, the CPU accesses main memory and copies a block of memory starting with the desired word. This block replaces the previous contents of cache memory.
3. The CPU accesses the cache and copies the word.



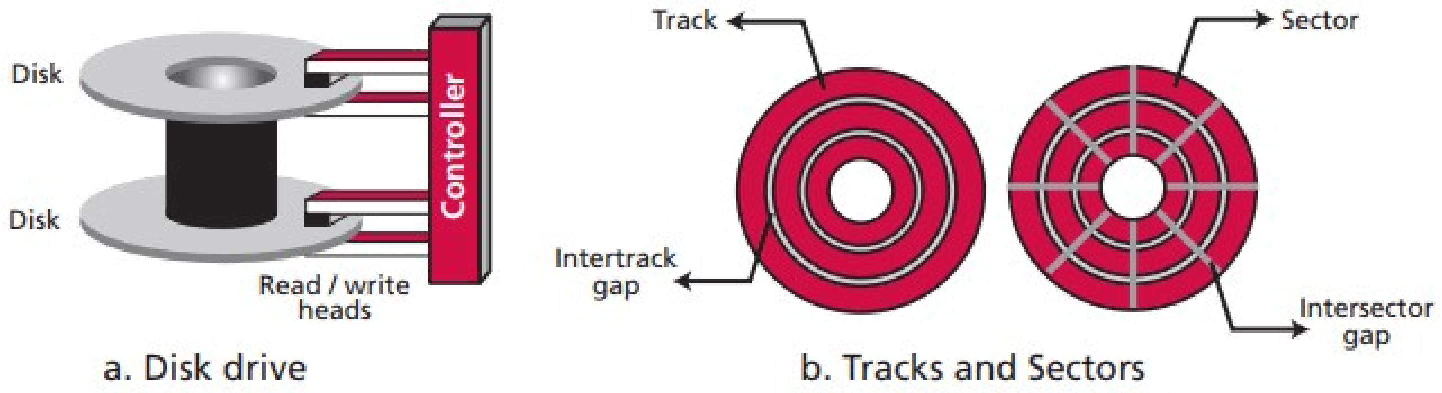
Input/Output Subsystem

❖ **Nonstorage devices**

- **Nonstorage devices** allow the CPU/memory to communicate with the outside world, but they cannot store information. ***Keyboard and monitor, printer***

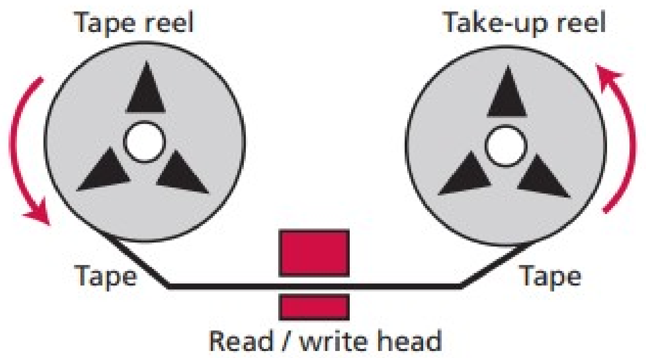
❖ **Storage devices**

- **Storage devices** can store large amounts of information to be retrieved at a later time. Their contents are not erased when the power is turned off.
- ***Magnetic storage devices***
  - Magnetic storage devices use magnetization to store bits of data. If a location is magnetized, it represents 1, if not magnetized, it represents 0.

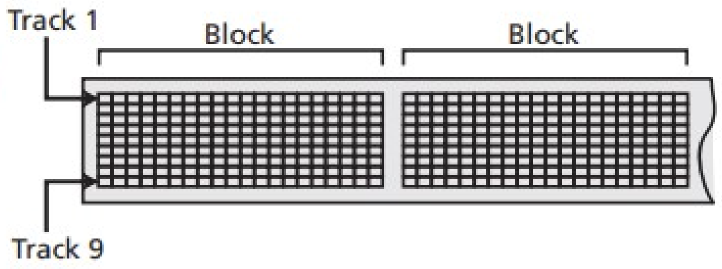


Magnetic Tape

- **Surface organization**. The width of the tape is divided into nine tracks, each location on a track storing 1 bit of information. Nine vertical locations can store 8 bits of information related to a byte plus a bit for error detection.
- **Data access.** A magnetic tape is considered a sequential access device. Although the surface may be divided into blocks, there is no addressing mechanism to access each block. To retrieve a specific block on the tape, we need to pass through all the previous blocks.
- **Performance**. Although magnetic tape is slower than a magnetic disk, it is cheaper.
- Today, people use magnetic tape to back up large amounts of data



a. Tape drive



b. Surface orgnization

Others

- ***Optical storage devices***
  - CD-R
  - CD-RW
  - DVD
  - …

Q & A