

C-book

Introduction

Computers

An electronic device for analyzing and storing data, making calculations, etc.

Originates from the word compute which means calculate. Computers can be programmed to process data following some finite set of instructions.

Types of computers

Personal computers

Or PCs in short. Intended to be used by a single person. First introduced by IBM. Built around a family of microprocessors referred to as “8086” (manufactured by Intel). Therefore its structure is referred to as “x86 architecture”.

An alternate system was Macintosh (introduced by Apple). Due to differences in hardware architecture between x86 and Macintosh, software used by these systems were not portable to the other.

Types of personal computers:

- Desktop - designed to sit on or under a desk. too big to be carried around.
- Workstation - desktop computers with high specs. powerful. built for specialized use cases.
- Laptop - portable. categorized under mobile computers.
- Tablet - even lighter than laptops. include touch input support.
- Handheld - can fit in our hands. example: personal digital assistant.
- Smartphone - cellular phones with advanced features
- Wearable - can be worn such as glasses

Network servers

Powerful personal computers with special hardware and software that enable them to function as primary computers in a network. They may be setup in groups called clusters or server farms.

Mainframe computers

Used in large organizations where many people, even thousands, frequently need to access the same data. Traditionally, each user accesses a mainframe computer through a device called a terminal.

Minicomputers

More powerful than personal computers but less powerful compared to mainframes. Aka. midrange computers.

Supercomputers

Most powerful, physically largest computers. Have thousands of processors.

Embedded computers

Computers that reside in a device and not directly visible. Aka. invisible computers. These devices are referred to as *smart* because they can interact with the environment in a more intelligent, adaptive, and efficient way.

Components of a computer system

A computer is a system. A complete computer system consists of:

- [Hardware](#): Tangible elements that can be seen and touched
- [Software](#): Programs that allow users to use the computer system and control its activities. Not tangible.
- [Data](#): Individual facts or pieces of information that is input to the computer for processing or produces as output after processing.
- [User\(s\)](#)

Computer Hardware

Major components of a PC

- Motherboard
Large circuit board that majority of the components are mounted on. Aka. main board.
- [Central Processing Unit](#)
- [Memory](#)
- [Secondary storage devices](#)
- Input / Output sub system enables interaction with the users
 - Display Monitor (VDU)
 - Keyboard
 - Mouse
 - Scanner
 - Printer
 - Microphone / Speakers
 - Joystick / Game controller
- Additional controllers
 - Graphic / display controller
 - Network / Wi-Fi controllers
 - Audio Interface (Sound card)
 - Universal Serial Bus (USB) controller
- Chassis the platform that holds all other components
 - Power supply unit
 - Fan / cooling system

Traditional computer system

Can be classified into 2 types.

- Single-user computer system
Can only be used by one user at a time.
- Multi-user computer system
Can be used by multiple users through multiple terminals.

CPU

CPU in today's computers are microprocessors. A CPU:

- performs logical & arithmetic operations
- controls all the other components and subsystems

Components of a CPU

Arithmetic & Logical Unit

Arithmetic unit handles arithmetic operations. Logical unit handles logical operations.

Control Unit

Controls the operation of the CPU and rest of the machine. Handles the decoding of instructions.

Registers

A type of memory that can hold a unit of data. Can be used for both data processing and control functionalities.

Type of registers

Program Counter

Keeps track of memory address of the next instruction to be executed.

Instruction Register

After an instruction is fetched into the CPU, it is stored in IR for execution. Keeping IR closer to CU (in the scale of micrometers) enables faster execution speed.

Accumulator

Where result of arithmetic or logical operation is stored immediately.

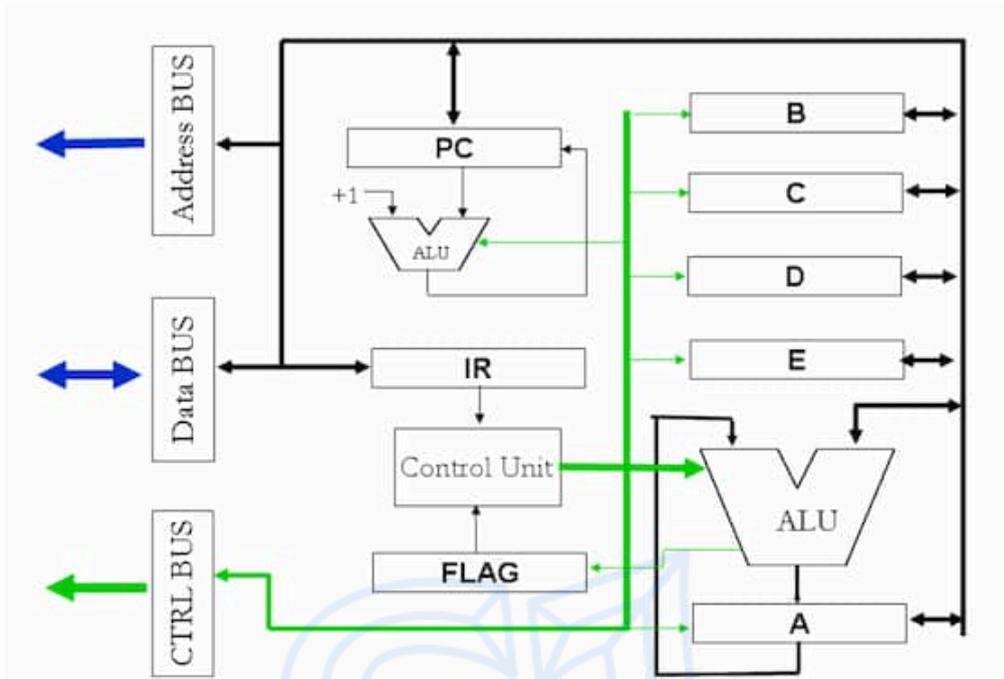
Flag Register

Stores the status of the last operation carried out by ALU.

General Purpose Registers

Can be used to various tasks. Used to store immediate results of the ALU. Number of GPRs vary depending on the CPU. Usually denoted as “B”, “C” and so on.

Internal Structure



- There is a secondary ALU after PC. That increments the PC after fetching a new instruction.
- ALU accepts maximum 2 inputs. One input should come from accumulator.

Enhancing CPU Performance

Instruction pre-fetching

When an instruction is in the execution cycle, next instruction will be fetched.

Instruction pipelining

Instruction cycle is divided into sub-operations and different segments of CPU handle each sub-operations.

Increases through put of the microprocessor.

Hyper Threading

Allows different resources of the CPU to be used at the same time. CPU, BIOS, OS, and chipset have to support HT technology to use this.

Multicore processors

2 or more separate microprocessors, combined onto a single Silicon chip. Higher performance gain compared to HT.

CPU Instructions

Instruction set

The set of operations a CPU can perform. Varies with each CPU. Most operations fall into 4 categories.

Data movement operations

Moves data from one place to another.

Examples:

- LOAD: Read data into CPU from memory or IO
- STORE: Write data from CPU to memory or IO

Mathematical operations

Examples:

- ADD: Adds 2 values in CPU

Logical operations

Examples:

- COMP: Compare 2 values in CPU

CPU control instructions

Examples:

- JUMP: Jump to a program address
- JUMPZ: Jump to a program address only if the last operation had a zero value
- JUMPNZ: Jump to a program address only if the last operation had a non-zero value
- NOP: Do nothing but spend some CPU time



Note

NOP operation is required to keep the CPU idle while waiting for some data.

Composition of instructions

OpCode

A coded value which tells what the instruction is supposed to do. OpCodes are the examples mentioned above (such as ADD and COMP).

Operands

Number and type of operations depend on the OpCode. Sometimes optional.

Can either be a:

- Value
- Memory address

Note

Single-instruction-single-data architecture (SISD) means that CPU takes one instruction and data associated with the instruction, executes it and goes to the next instruction.

Execution of a program (Instruction cycle)

A program is executed as a series of steps.

Instruction Fetch

Next instruction is loaded into the CPU from memory. A memory read cycle is executed. Similar for any instruction.

Decode

CPU decides the instruction binary pattern and decides on what action to take. Similar for any instruction.

Operand Fetch

If the operation requires additional operands in memory, ready cycles are executed to fetch them. Depends on the operation. Sometimes optional.

Execution cycle

Instruction is executed by the CU. CPU generates the signal for its internal circuits to carry out the operation. Depends on the operation.

CPU Support Chips

Carries out tasks on behalf of the CPU.

Examples:

- Direct Memory Access controllers
- Disk controllers
- Real-Time Clock
- Display controllers
- Interrupt controllers
- Communication controllers

DMA controllers

Provides a way of bypassing the CPU when transferring data between memory and Input/Output (IO) devices. Resides between memory and CPU.

Disk controllers

When reading/writing to/from a disk the CPU will create a buffer (special memory area) containing the sector address and the data to be written or read. Then the CPU informs the controller about the location of the buffer. The controller then transfers the content of the buffer directly from memory to the disk sector.

Examples:

- Floppy Disk Controller
- ATA Controller

Real-Time Clocks

Used to keep track of time of the day. Usually backed-up by an extra power source such as a Li battery. Used to store some of the configuration information such as CMOS setup memory.

Display controllers

Used to generate images and text that you see on the displaying device on behalf of the CPU. The actual image (which is binary data) is generated in its memory called “refresh buffer”, then passed to video controller to be displayed.

Video controllers

Used to display the image on monitors. Classified based on their video processor and video memory.

Some video standards are:

- Video Graphics Array - minimum resolution: 320x240 in 256 colors or 640x480 in 16 colors
- Super VGA - supports resolution up to 800x600
- Video Electronics Standards Association SVGA - was developed to standardize SVGA. Also includes a video standard for connecting high-speed adaptors directly to the processor bus.

Computer Memory

2 types:

- Permanent or non-volatile
- Temporary or volatile - will be lost when power is off

Instructions and data for the CPU is sent from the memory. Results are sent back to the memory.

Consists of an array of consecutive memory locations. Each location is identified by a memory address and stores a single piece of data, usually a byte. CPU can either read or write a single memory location at a time.

Memory Bus

A set of electrical connections that connect memory locations with CPU.

3 types:

- Address bus - used to indicate address of a memory location. goes from CPU to memory. determines the memory space
- Control bus - used to send control information (read request RD or write request WR) from CPU to memory.
- Data bus - actual data transmission. bidirectional. determines how much of data can be transferred at a time

Writing data:

1. Address bus is set with the memory address
2. Data bus is set with the data to be written
3. CPU activates WR in control bus

Reading data:

1. Address bus is set with the memory address
2. CPU activates RD in control bus
3. Data is fetched using data bus

ⓘ Note

Computer memory is called as bus-slave because it responds to the bus. CPU is called as bus-master because it controls the bus.

Types of Memory

Read Only Memory

Memory is written (hardcoded) when they are fabricated as ICs. Used to store initial start-up programs. Not economical to produce in small quantities.

Example: ICs in the market that have various melodies.

Programmable Read Only Memory

Similar to ROM but, the content can be written (using special equipment) once after the manufacturing process. Cost effective compared to ROMs.

UV Erasable PROM

Similar to PROM but can be written multiple times. The content should be erased using UV light before new content is written. Both erasing and programming process require special equipment.

Electrical Erasable PROM

Similar to UVEPROM but the content can be erased by applying a special high voltage.

Flash ROM

A special type of EEPROM that can be erased or programmed while in the application circuit. The contents remains unchanged even after a power failure.

Commonly used in modern PCs, various networking devices such as routers and firewalls and memory pens (also referred as memory sticks or USB pens).

Read Write Memory

Aka. RAM. Volatile.

2 types:

- Static RAM - uses transistors
- Dynamic RAM - uses capacitors. bulk of the PC memory is made using DRAM.

Transistors	Capacitors
Uses semiconductors	Uses semiconductors
High speed switching	Slower performance
Retains state forever (if power is supplied)	Discharges after some time, needs refreshing (in μs scale)
More reliable	Less reliable
Low transistor density	High capacitor density
High power consumption	Low power consumption
High cost per bit	Low cost per bit

Memory Modules

Set of memory ICs presented as a single memory block to the motherboard .

Type of memory modules:

- SIM - Single Inline Memory Module
- DIMM - Dual Inline Memory Module (64-bit wide 168-pin)
- DDR-DIMM - Double Data Rate-DIMM

Memory Characteristics

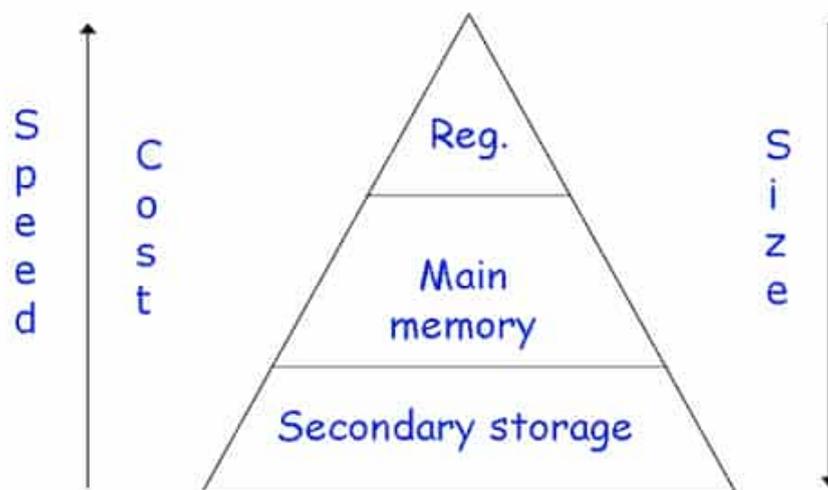
- Access speed – time taken for the CPU to read from or write to memory
- Cycle time – time taken to complete 1 memory access operation
- Packing Density - memory capacity per unit area
- Power consumption
- Cost - cost per unit of memory capacity

Memory Hierarchy

Modern CPUs are much faster than the speed of memory. The memory has to be organised in such a way that its slower speed does not reduce the overall system performance.

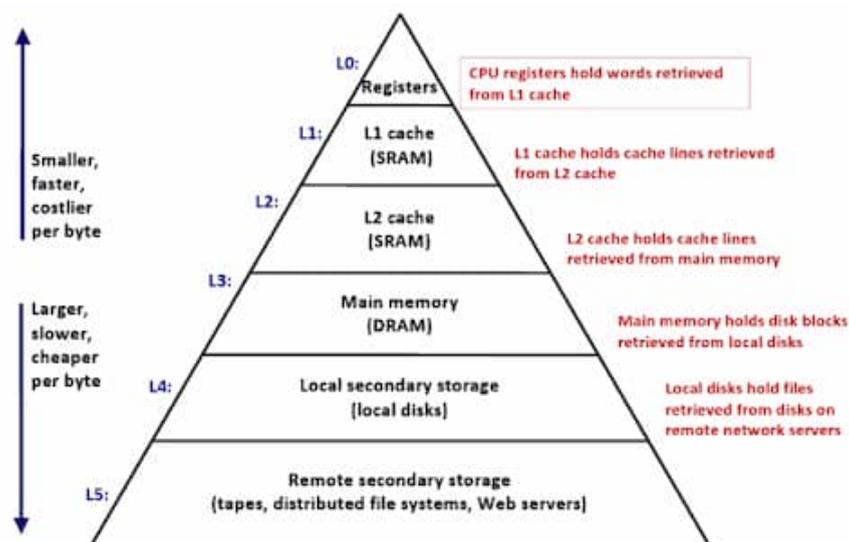
The objective of having a memory hierarchy is to minimize cost, maximize capacity and performance. Special methods are used to store the frequently used items in the faster devices and others in slower devices.

Traditional



Modern

Cache memory is added to this hierarchy to bridge the speed difference between processor and memory. It is used by the microprocessor to store frequently used instructions and data.



Swapping

When free space is needed in cache memory, infrequently used data and instructions are sent back to the main memory. This process is called swapping.

Paging

When free space is needed in main memory, some of the data and instructions are sent to the secondary storage. This process is called paging.

Von Neumann Architecture

An architecture on how a computer can be built. Simple and implementable proposal but not the best mechanism for operations such as disk access or time critical operations.

Von Neumann Machine

A computer structure resulting from Von Neumann architecture.

Parts of a Von Neumann Machine

- A memory - contains both data and instructions
- A calculating unit - capable of performing both arithmetic and logical operations on the data
- A control unit - which could interpret an instruction retrieved from the memory and select alternative courses of action based on the results of previous operations

CPU centric system

- Each operation is carried only by the CPU
- Every movement of data is made via the CPU
- Memory is the only “direct access” storage device for the CPU
- Only one operation is carried out by the CPU at any time

Secondary Storage

Non-volatile or permanent. High capacity. Low cost-per-megabyte.

Can be categorized into 3 types (Magnetic storages are included in a separate page):

- [Magnetic](#)
- [Optical](#)
- Floptical - combination of optical and magnetic devices

Optical

Uses tiny visible light beams or laser.

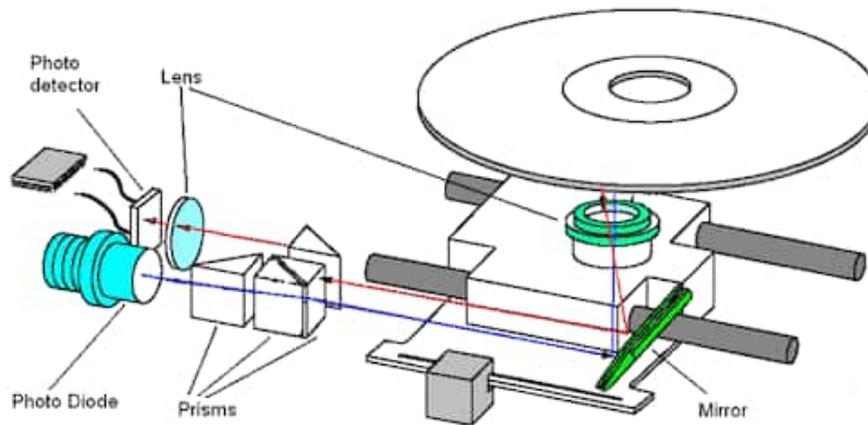
Examples:

- Compact Disk - Read Only Memory
- Compact Disk - Recordable
- Compact Disk - ReWritable
- Digital Video Disk
- Digital Video Disk - Recordable
- Digital Video Disk - ReWritable

Compact Disk

Made of polycarbonate wafer, 120mm in diameter and 1.2mm thick, with a 15mm hole in the centre.

The wafer base is stamped or moulded with a single physical track in a spiral configuration starting from the inside of the disk and spiralling outwards. When examined under a microscope, pits (raised bumps along the track) and lands (flat areas between the pits) can be seen. Pits are 0 and lands are 1.



These components are used in CD technology.

- Photodiode - to generate the light beam
- Servomotor - to position the beam onto the correct track
- Phototransistor - to convert light into electrical impulses

Flash storage

Built using EEPROMs. High capacity and high transfer speed compared to other forms of portable media. Low latency, low noise, low power consumption, high reliability.

HDDs are now being replaced with SSDs which uses flash memory technology. RAM modules are also starting to use flash memory technology.

Categorized into 2 divisions based on operational characteristics:

- NAND Flash
- NOR Flash

Flash memory stores information in an array of memory cells made from floating gate transistors. In the traditional flash drives, each cell stores a single bit of data. However, newer flash devices known as multi-level cell devices can store more than one bit per cell by choosing between multiple levels of electrical charge.

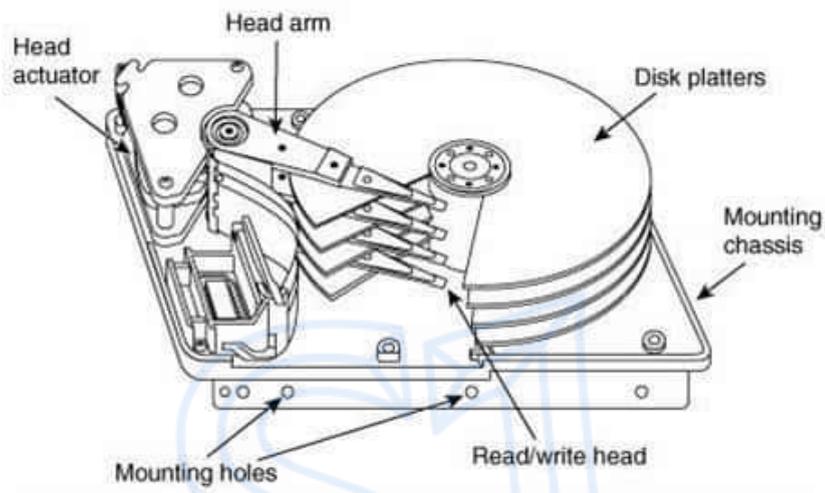
Magnetic Storage

Tape drive

Stores data on magnetic tapes. high capacity. cost-effective. slow. mainly used for data backups nowadays. Aka. DAT drives.

Hard disk drive

Magnetic storage. Operationally identical to a conventional radio cassette tape. Uses a disk coated with magnetic medium. High data transfer speed.



Capacity can be improved by adding more platters. Platters are kept in dust free environment to avoid damages because of high speed. Each side of the platter require a read/write head.

Categorized by their capacity, controller, and platter rotation speed. Capacity ranges from 500GB to 8TB. Platter rotation speeds are 3600, 5400 and 7200 RPM.

Available disk controllers:

- Integrated Device Electronics
- Small Computer System Interface
- Serial AT Attachment Interface

Track

Single ring of data on one side of a platter.

Sector

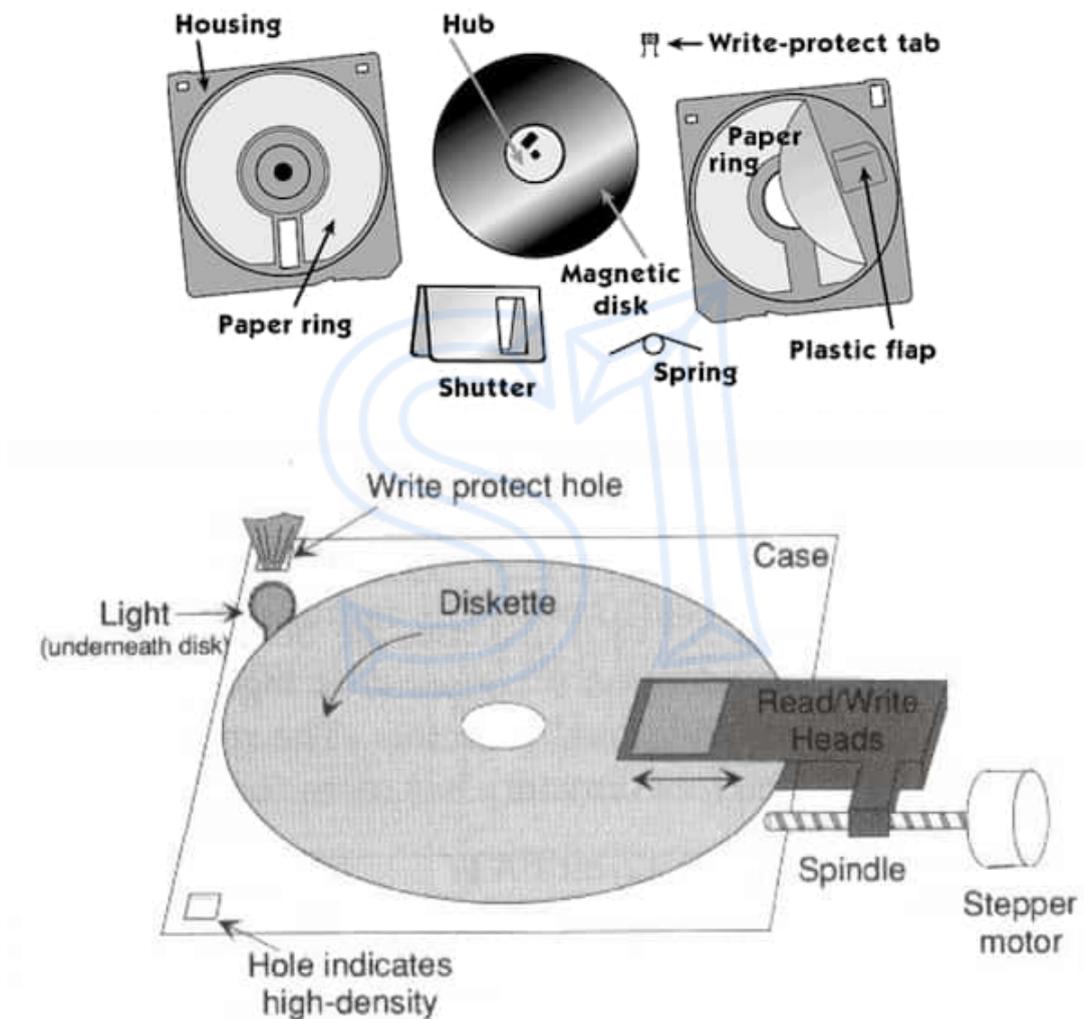
Fixed size (usually 512 bytes) divisions (usually about 900) of a track.

Cylinder

Set of tracks on a disk, that are on each side of all platters in a stack and are at the same distance from the center of the disk.

Floppy disk drive

Removable. Has a flexible, magnetic medium that is enclosed in a (semi-)rigid plastic case.



Paper rings are used for protection.

Flap - a small opening for the read/write head to access the disk

The head actuator mechanism is slightly different to the mechanism in a hard disk. In a hard disk the head arm moves laterally from centre of the disk towards to the edge while in a floppy disk the movement is horizontal.

The read/write head is mounted on a spindle and the spindle is controlled by a stepper motor. The high density hole is only available in High Density floppy disks.

Computer Software

Program

An ordered sequence of instructions that the hardware can execute.

Software

The programs that allow users to use a computer system and control its activities.

Process

An instance of a running program.

Categorization of Software

Software can be categorized into system software and application software. Categorizing software into these 2 types is a difficult task.

System software

Any software that satisfies any one of the following conditions:

- manages the computer hardware
- can be used to maintain the computer so that it runs efficiently
- helps to do tasks easily and quickly
- helps to create new software
- may not be targeted for end-users

Example: [operating system](#)

Systems programming requires a greater degree of hardware awareness compared to programming application software.

Utility software

Performs very specific tasks, usually related to managing computer system resources. Widely used utility software today can be incorporated into operating systems in the future.

Software development tools

Software that are required to create new software.

Examples:

- Compilers
- Assemblers
- Linkers
- Debuggers

Some operating systems (like Linux) provide such software by default.

Firmware

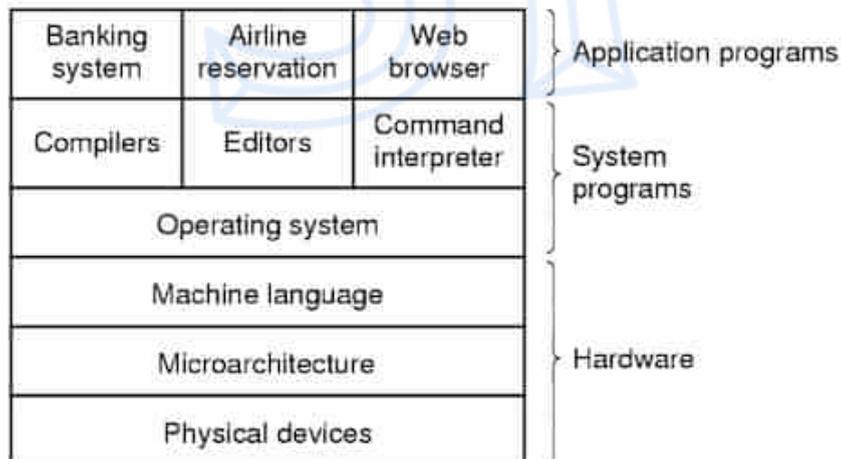
System software embedded in a hardware device (i.e., stored on a non-volatile storage).

Examples:

- BIOS in personal computers
- controlling software in DVD players

Application Software

All software except system software. Designed to be used by end-users.



Operating System

A layer of (systems) software that manages the hardware resources. Also provides a friendly interface to the users by hiding the complexity of the hardware. Acts as a virtual machine to application software. Microsoft Windows, Linux, and Mac OS are a few examples.

OS provides common interfaces of the hardware to software regardless of the specific features of its underlying hardware. Thus, the program needed only to be compatible with this virtual machine interface.

Handles:

- CPU scheduling
- Memory management
- Storage management

Kernel

Core of an OS. Loads before any of the user programs and remains in the memory. Responsible for managing CPU, memory, disk, processes.

Major services of OS

An operating system provides two major services.

OS as an Extended Machine

OS hides specific details about the underlying hardware. Instead, provide a simpler and uniform interfaces for application software and users.

For example: saving a file to a secondary storage device. Writing mechanism is physically different for floppy disk, hard disk, and magnetic tape. However, application software authors or the end users do not need to be aware of those details. For them, saying “save this file in that place” to the OS is enough. All other details will be handled by the OS.

OS as a Resource Manager

OS manages all hardware resources. OS decides which programs run at which time, how much memory is allocated to each program, where to write data, how to deal with concurrent users, and many other things.

History

1st generation

Cannot be considered as real operating systems. Those days programs were written by changing the wire in a plugboard. When a different program is to be executed the plugboard needs to be rewired.

A plugboard is a large circuit board where different vacuum tubes are connected to form a circuit. The tip of the wire includes a plug, therefore instead of soldering those wires they were directly plugged into the plugboard.

2nd generation

Were mainly batch systems. Very low performant.

Steps

- A “job” is submitted in the form of punch cards
- It is transferred into a magnetic tape. Multiple jobs with similar needs are batched together.
- Tape is inserted into main computer which does the processing.
- Output is written to a tape (which might then be converted to a hardcopy)

3rd generation

Uses multiprogramming for better performance.

Multiple jobs are kept in the memory at the same time. A job is run by the CPU until it is blocked due to some I/O activity. Then OS switches to one of the remaining jobs. No user interaction.

Time-sharing systems

Uses multitasking which is a variation of multiprogramming. Improves resource utilization. CPU is never idle. Allows users to interact with their programs while it is being executed.

4th generation

Aka. desktop systems or personal computer OSs. Currently used.

Functions of an Operating System

Functionality of an OS slightly varies based on the type.

Common tasks carried out by a typical OS:

- **Memory management** - allocates memory for programs. handles [swapping](#) and reversing. The objective is to utilize limited memory in an efficient manner.
- **Spooling print jobs**: All the print jobs will be collected onto the hard disk and later they are scheduled by the OS for printing.
- **Configuring devices**: allows easy access to devices, their installation and configuration.
- **Monitoring system performance**: A modern OS can display how system resources are utilized.
- **Administering security**: supports multiple concurrent users while making sure each user uses the system without interference by others. It also enforces authentication and authorization.
- **Managing storage media and files**: makes sure the secondary storage devices are optimally used. all the hard work of saving and retrieving files to and from disks, timestamping, setting various file attributes such as read-only, hidden and so on are handled by the OS.

Popular Operating Systems

Main classes of operating systems:

- Microsoft OS - Started with MS-DOS. Later evolved into Windows.
- UNIX - Led to different variants such as Linux, Sun Solaris, FreeBSD, AIX, HP-UX and so on.
- Mainframe OS - From IBM. Some of the well-known ones are OS/360, OS/390 and OS/400.
- MacOS - For Apple Macintosh computers

DOS

DOS was the first disk-based OS developed for IBM-PC by Microsoft. The objective was to keep the OS, application programs and all user files on a disk and managing them through set of commands called DOS commands. Simple to use and learn. Closed source and paid.

There are 2 well-known variants:

- Personal Computer DOS - developed and sold with IBM PCs
- Microsoft-DOS - sold in open market.

Windows

DOS with a GUI. Inspired by Apple Lisa. Initial versions of Windows were just an application over DOS; GUI was issuing DOS commands. Windows 98 also supported multitasking. Closed source and paid.

Supports most of the hardware and software components. Mostly used OS in the consumer market. However, compared to some other OSs Windows is not so stable, does not use resources in an effective way and have many security issues.

UNIX

Has many variants and versions which are still being used. It was developed as a time-sharing system for minicomputers and was initially mostly used by universities. Very stable. Highly reputed for its security, reliability, robustness, and performance.

It was developed by engineers for engineers. Therefore, it was harder for an average person to effectively use UNIX or any of its variants. As a result, UNIX was not so popular among the average users. However, modern variants with GUIs are much better in terms of user friendliness.

Linux

A UNIX-like OS developed originally for home PCs. Main objective was to make it simple so that anyone can understand and improve. Free and open source. Stable, reliable, and efficient. Most used OS in servers. Requires less disk space, memory and processing power. Better at handling multiple users.

Now, runs on many platforms including Intel, PowerPC, Macintosh, and Sun Sparc.

Embedded System

A special-purpose computer which is completely encapsulated by the device it control. A typical embedded system has a single specialized function.

The computer sits behind the equipment and controls its various components based on the input provided by the user and obtained from number of sensors attached to the system. The way this interaction occurs is usually implemented by the software that run inside the embedded computer. Due to the flexibility and advanced processing capabilities provided by the software, manufacturers of these equipment increasingly prefer to use embedded systems rather than their traditional hardwired controllers.

Cross compiling

The process of compiling a software on a different computer and later using that on a program on a different computer.

Differences with normal computers

- Processing unit is typically a microcontroller
- End users wouldn't feel the existence of embedded computer most of the time.
- Used over a very long period of time and generally they cannot be programmed or maintained by the end user.
- Have many design constraints such as:
 - limited memory
 - requirement of low cost
 - strict performance guarantee
 - fail-safe operations
 - low power consumption
 - reliability
 - guaranteed real-time behaviour
- Can be optimized as they are dedicated to a specific task.
- Often use simple executives (OS kernels) or real-time OSs, support for real-time scheduling and no hard drives.
- Often interact with their physical environment using a variety of sensors and/or actuators.

Microcontroller

Can be considered as a very simple and a small-scale computer in a single chip.

Has a program memory, a processing unit and a random access memory (registers) which maps to hard disk, CPU and RAM in a typical personal computer respectively. Lightweight.

Programming and designing circuits with microcontrollers are quite complex and not friendly for beginners. Microcontroller programs have to be written separately and inserted into the device using a separate tool (a programmer).

Examples

- thermostats
- calculators
- ATMs
- printers
- video game consoles
- Handheld computers
- PDAs
- cell phones

The line of definition continues to blur as devices get more famous and expanded.

Challenges

Embedded system research field is rich with potential because it combines two factors.

1. First, the system designer usually has control over both the hardware design and the software design.
2. Embedded systems are built upon a wide range of disciplines, including computer architecture (processor architecture and micro architecture, memory system design), compiler, scheduler/operating system, and real-time systems.

Combining these two factors means that barriers between these fields can be broken down, enabling synergy between multiple fields, and resulting in optimizations which are greater than the sum of their parts.

Delivering good performance

Many embedded systems have real-time requirements. Unfortunately, many of the performance enhancing features make it difficult to predict their performance accurately.

Hence the challenge for real-time system researchers is to develop approaches to design fast systems with easily predicted performance, or to measure existing complex but fast systems more accurately.

Arduino

An open-source physical computing platform based on a simple general purpose microcontroller board, and a development environment (Arduino IDE) for writing software for the board. Can be used to develop and implement embedded systems.

Ready to go platform that can be use to simply plug into the computer, upload a program and run it without any additional circuit development.

- Can be used to develop interactive devices, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs.
- Can be stand-alone, or can be run through a software (e.g., Flash, Processing, MaxMSP)

The boards can be assembled by hand or purchased pre-assembled; the open-source IDE can be downloaded for free.

Differences

Programming an embedded platform like the Arduino is somewhat different from programming a PC.

- Arduino does not have a keyboard, or a display screen
- Memory and storage are limited

Arduino programs are usually written and compiled in a different computer (usually a PC) and later downloaded into the Arduino board via the USB cable.

Advantages

- Learning is easy (compared to embedded system development) as it hides messy details of microcontroller programming
- Comes with a lot of hardware facilities ready to use

Hardware

The Arduino board is where the code is executed. The board can only control and respond to electricity, so specific components are attached to the board to enable it to interact with the real world.

These components can be

- sensors - converts some aspect of the physical world to electricity so that the board can sense it.
Examples: sensors include switches, accelerometers, and ultrasound distance sensors.
- actuators - gets electricity from the board and converts it into something that changes the world.
Examples: lights and LEDs, speakers, motors, and displays.

The most popular boards contain a USB connector that is used to provide power and connectivity for uploading your software onto the board. Figure 4.3 shows the basic components of an Arduino Uno.

The “Mini USB Socket” is used to connect the board to a computer using a standard USB mini cable. This can be used to power the board. The “External Power Input” socket can be used to power the Arduino board with a 6-20V supply.

The “5V Input/Output” pin can be used either to power the board using a 5V supply when there is no other mode of power supplying is used or to get a 5V supply to provide to any other device when some other power supplying mode is in use on the board.

The “Reset Button” can be used to force the board to re-start executing the uploaded program from the very beginning. The “Digital I/O Pins” can be used to provide digital (0 or 5V) outputs and to read digital inputs by configuring them in the program.

Certain pins (3, 5, 6, 9, 10 & 11) are marked with a tilde (~). These pins can be used as PWM (Pulse-Width Modulation) outputs. In PWM configuration, these pins can be used to effectively output voltage values in the range 0-5V.

The “Analog Input Pins” can be used either as digital I/O pins or as input pins which can read voltage values in the range 0 – 5V by configuring them in the program.

The “Ground Pins” can be used as ground connections for external circuitry. This is a very basic introduction to the pins and other components of the Arduino Uno board which will be needed in this module.

Software

Software programs for Arduino is created on a computer using Arduino IDE. They are called as sketches. The IDE can be used to write and edit code and convert this code into instructions that Arduino hardware understands. The IDE also transfers those instructions to the Arduino board (a process called uploading).

Trends in Computing

Note

I don't like including stuff like these in the curriculum. The trends are changing quickly.

Included those for the sake of completeness.

Current Trends

Cloud Computing

Cloud computing is a subscription-based service which provides a pool of configurable computer resources: storage, networks, servers, applications, and services, over the internet.

Advantages:

- On-demand
- Cost effective for users

Disadvantages:

- Possible downtime
- Security & Privacy concerns

Augmented Reality

Augments / supplements user's physical, real-world environment with computer generated sensory inputs such as video, sound, graphics, or location (E.g.: GPS) data. Users can visualize information about the environment and its objects which are laid on top the real world.

Applicable for many fields including architecture, art, commerce, education, gaming, medical, etc.

Ubiquitous Computing

The concept of enabling computing everywhere at any time. Aka. pervasive computing.

Examples: smart phones, PDA (Personal Digital Assist), smart glasses, digital watches, and interactive white boards.



Characteristics

- Very tiny in physical size
- Being integrated into any shape of device and displacing services of desktop systems
- Augment the original use-value of devices they are embedded
- Communicating through increasing interconnected networks

Future Trends

GPU-Accelerated Computing

GPU-accelerated computing is the use of a graphics processing unit (GPU) together with a CPU to accelerate scientific, engineering, and enterprise applications. Compute-intensive work is offloaded to a GPU while the remaining work is carried out by CPU. Enables applications to run significantly faster.

Reconfigurable Computing

Reconfigurable computing is a computer architecture combining some of the flexibility of software with the high performance of hardware by processing with very flexible, high speed, computing fabrics like field-programmable gate arrays (FPGAs).

The principal difference when compared to using ordinary microprocessors is the ability to make substantial changes to the data path and to the control flow. On the other hand, the main difference with custom hardware, i.e., application-specific integrated circuits (ASICs) is the possibility to adapt the hardware during runtime by “loading” a new circuit on the reconfigurable fabric.

Quantum Computing

A quantum computer makes direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data. Quantum computation uses qubits (quantum bits), which can be in a superposition of states. A theoretical model is the quantum Turing machine, also known as the universal quantum computer.