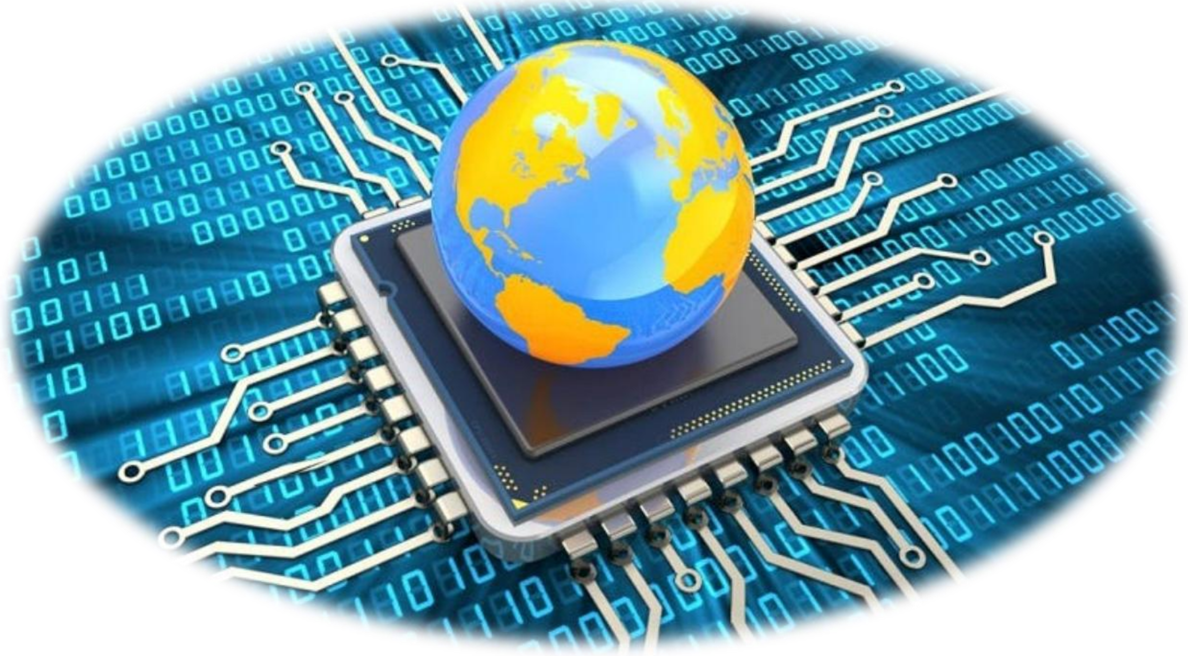


SE1101 : COMPUTER ORGANIZATION
DS1106 : COMPUTER SYSTEM ORGANIZATION

Ms.S.Sweshthika

Lecturer (Probationary)

Department of Software Engineering

Faculty of Computing

Sabaragamuwa University of SriLanka

Intended Learning Outcomes

- ILO1:- Outline the concepts of the construction of computer systems.
- ILO2:- Outline working knowledge of a low level & high-level programming of hardware devices.
- ILO3:- Present an overview of the main characteristics of computer memory systems and the use of a memory hierarchy.
- ILO4:- Explain the use of I/O modules as part of a computer organization.
- ILO5:- Present an overview of essential characteristics of machine instructions.

Course Contents

- Topic 01: Basic Concept and Computer evolution: Organization and Architecture, the evolution of the Intel x86 Architecture, Embedded Systems, ARM architecture.
- Topic 02: Computer Performance Issues: Multicore, MIC and GPGPUs, Basic Measures of Computer Performance, benchmark and SPEC.
- Topic 03: Computer Function and interconnection: Computer Bus Interconnection, Point to Point Interconnection.
- Topic 04: Computer Memory System: Cache Memory Principles, Semiconductor main memory, External memory.

Course Contents

- Topic 05: Input/output: External Devices, I/O Modules, Interrupt Driven I/O, Programmed I/O, I/O channels and processors, External Interconnection Standards.
- Topic 06: Arithmetic and Logic: number system, Integer Representation, Floating Point representation, Digital logic, Combinational Circuits, Sequential Circuits, Programmable Logic Devices.
- Topic 07: The central Processing Unit: Machine Instruction Characteristics, Addressing Modes, Assembly language, Processor, Instruction Level Parallelism and superscalar Processor.
- Topic 08: Parallel Organization: Parallel processing, Multicore computers, General purpose Graphic processing Unit.

Assessment Strategy

Continuous Assessment 30%			Final Assessment 70%		
Quizzes 20%	Mid-term 30%	Assignments 50%	Theory 100%	Practical	Other Specify

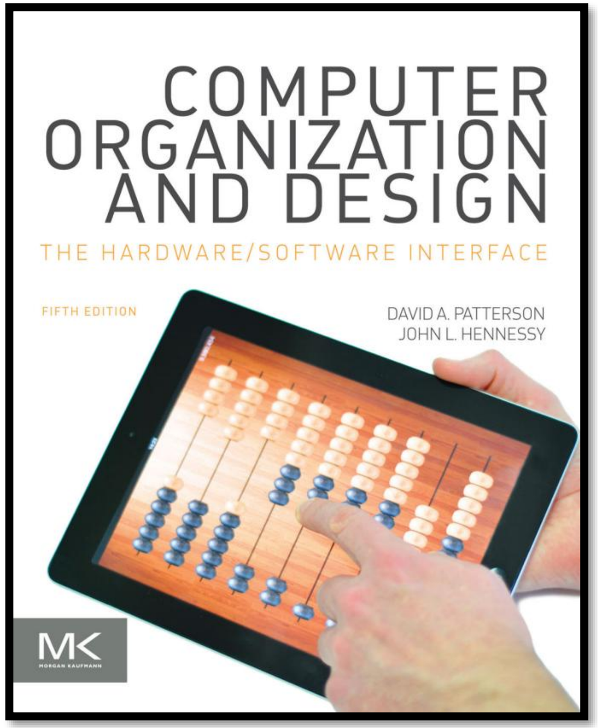
Semester	**1**		
Course Code:	SE1101		
Course Name:	Computer Organization		
Credit Value:	2		
Core/Optional	Core		
Hourly Breakdown	Theory	Practical	Independent Learning
	30		70

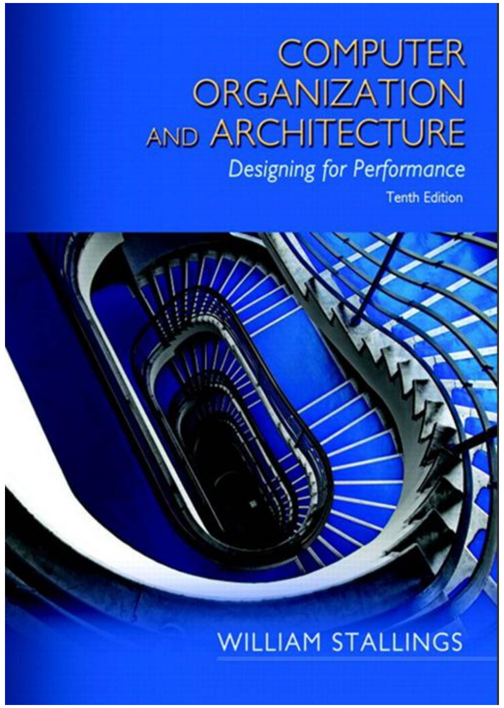
References

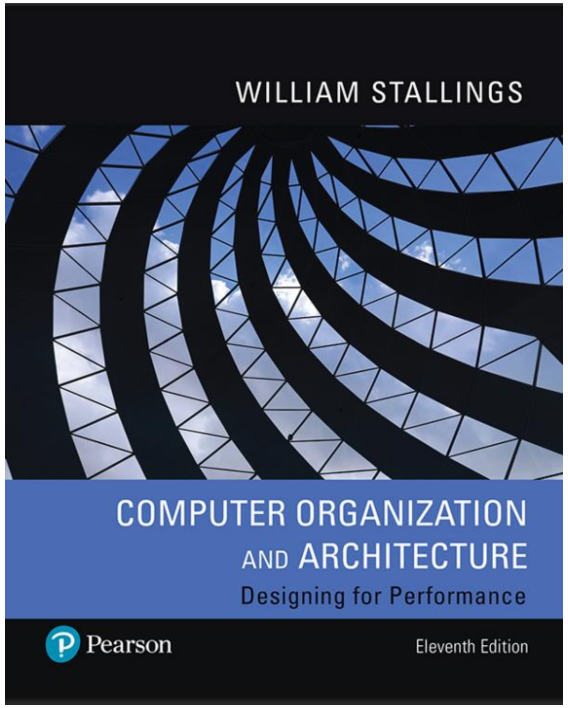
David A. Patterson and John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 5th ed., Morgan Kaufmann is an imprint of Elsevier, ISBN: 978-0-12-407726-3, 2014.

W.Stallings, *Computer Organization and Architecture: Designing for Performance*, 10th ed., Pearson Education, ISBN-13: 978-0-13-410161-3, 2015.

W.Stallings, Computer Organization & Architecture: Designing for Performance., 11th ed, Pearson Education, ISBN-13: 978-0-13-607373-4, 2019.





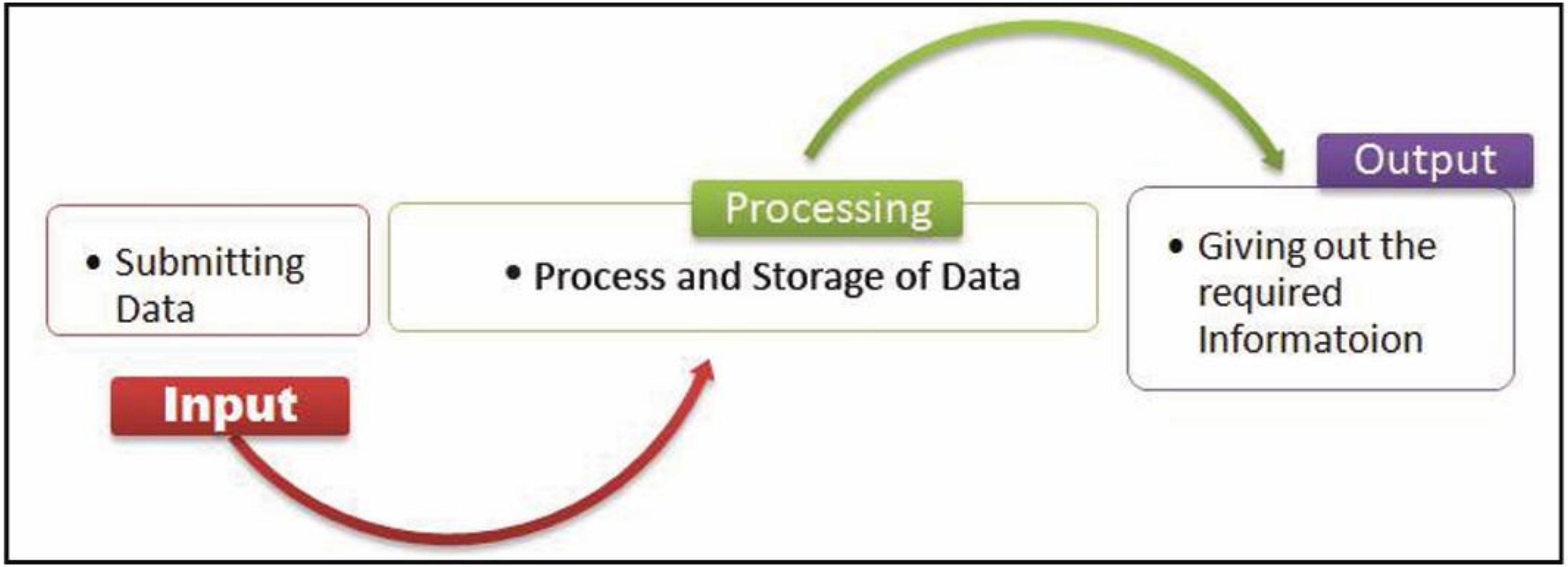


CHAPTER : 01
Introduction to Computer Organization

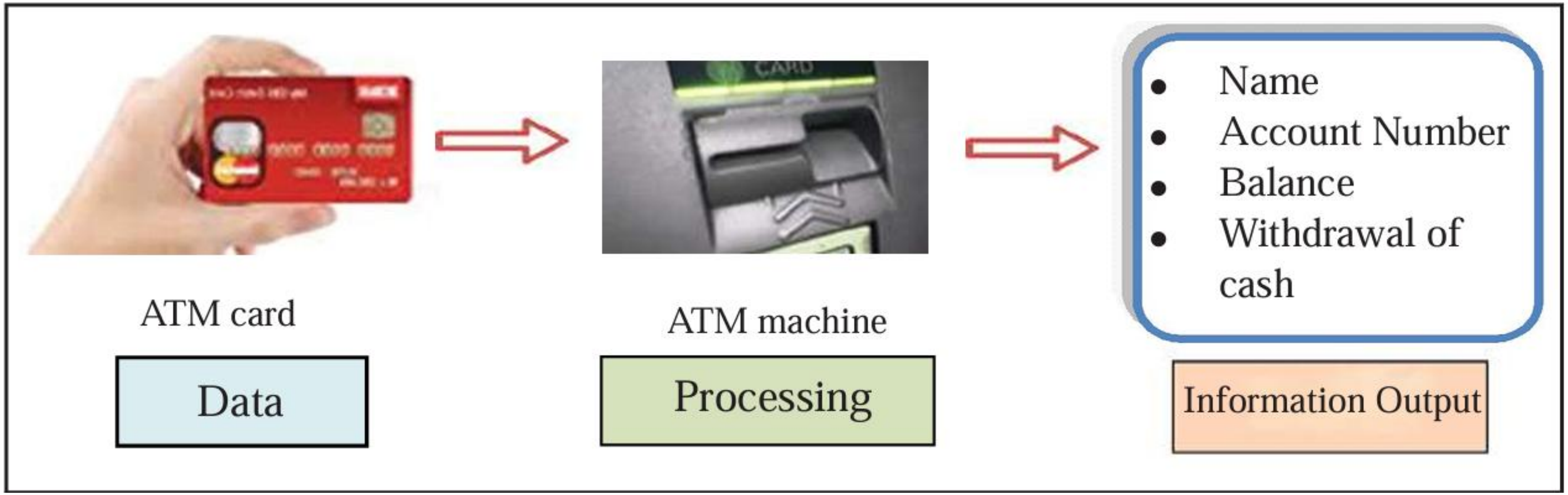
Objectives

- Understand the basic structure of a computer system.
- Distinguish between computer architecture and organization.
- Explain why both architecture and organization are important.
- Recognize key components: CPU, memory, I/O, and system interconnection.

What is Computer System?



Example of Computer System



Architecture Vs Organization Real World Analogy

Architecture

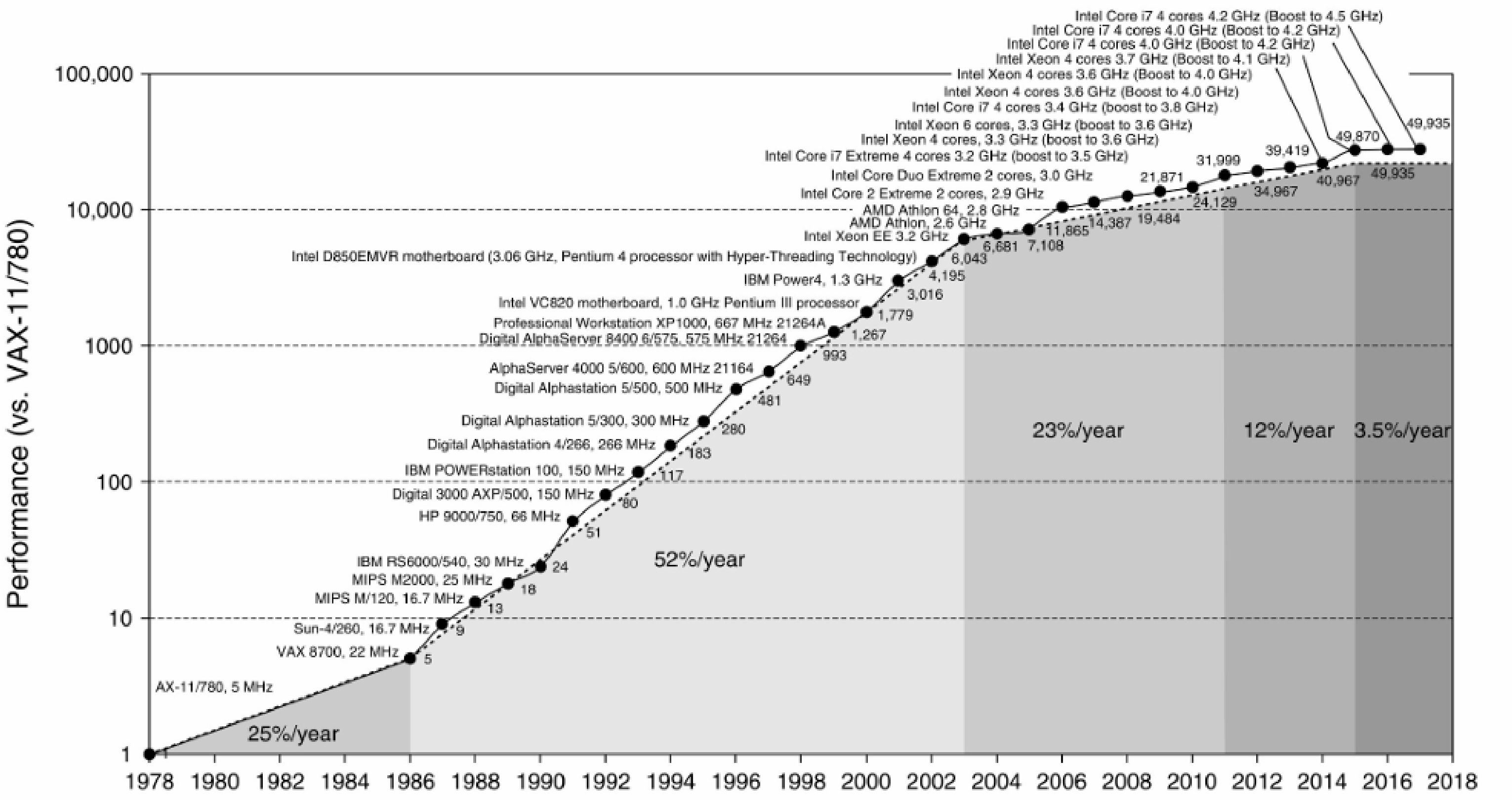
Organization

What is Computer Architecture?

- Attributes of the system visible to the programmer.
- Has a direct impact on how a program executes.
- Commonly referred to as Instruction Set Architecture (ISA).
- ISA includes:
  - Instruction formats and opcodes
  - Registers
  - Instruction & data memory
  - Effects of instructions on memory and registers
  - Control of instruction execution (algorithm)

Why Computer Architecture is important?

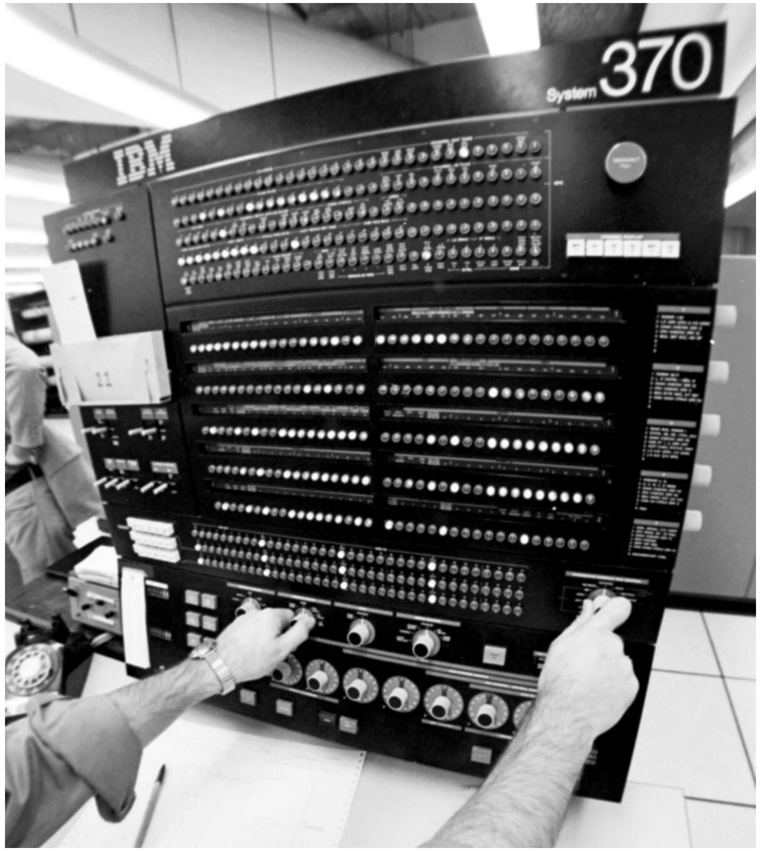
Why Computer Architecture : Historical Trends



What is Computer Organization?

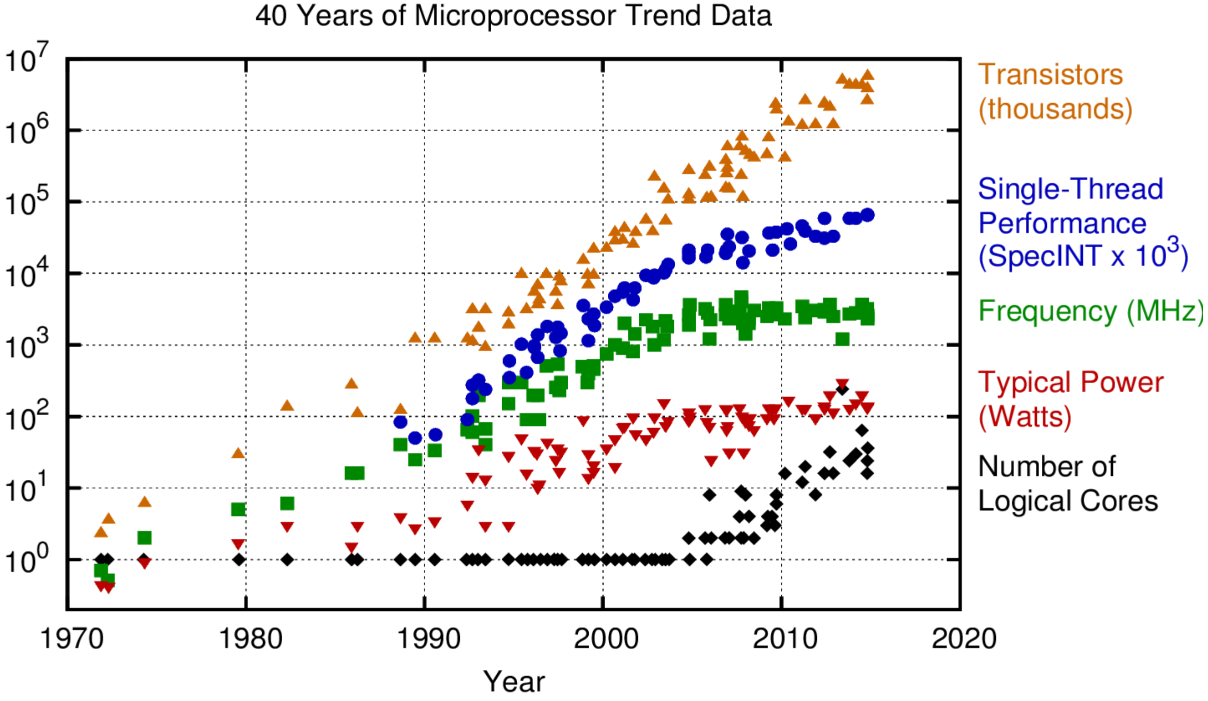
- Internal hardware details that implement the architecture.
- Deals with operational units and their interconnections.
- Focuses on how the system is built, not how it's programmed.
- Examples:
  - Control signals
  - Interfaces with peripherals
  - Memory technology used

Why Computer Organization is important?

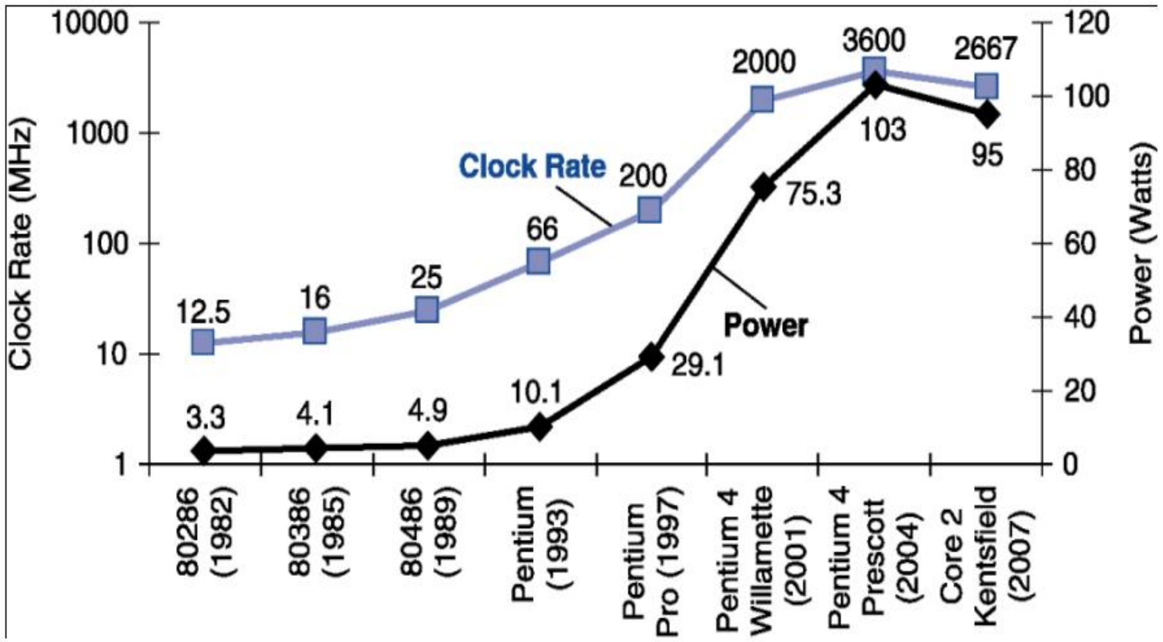


https://images.app.goo.gl/L4vgPgh8mqupwC189

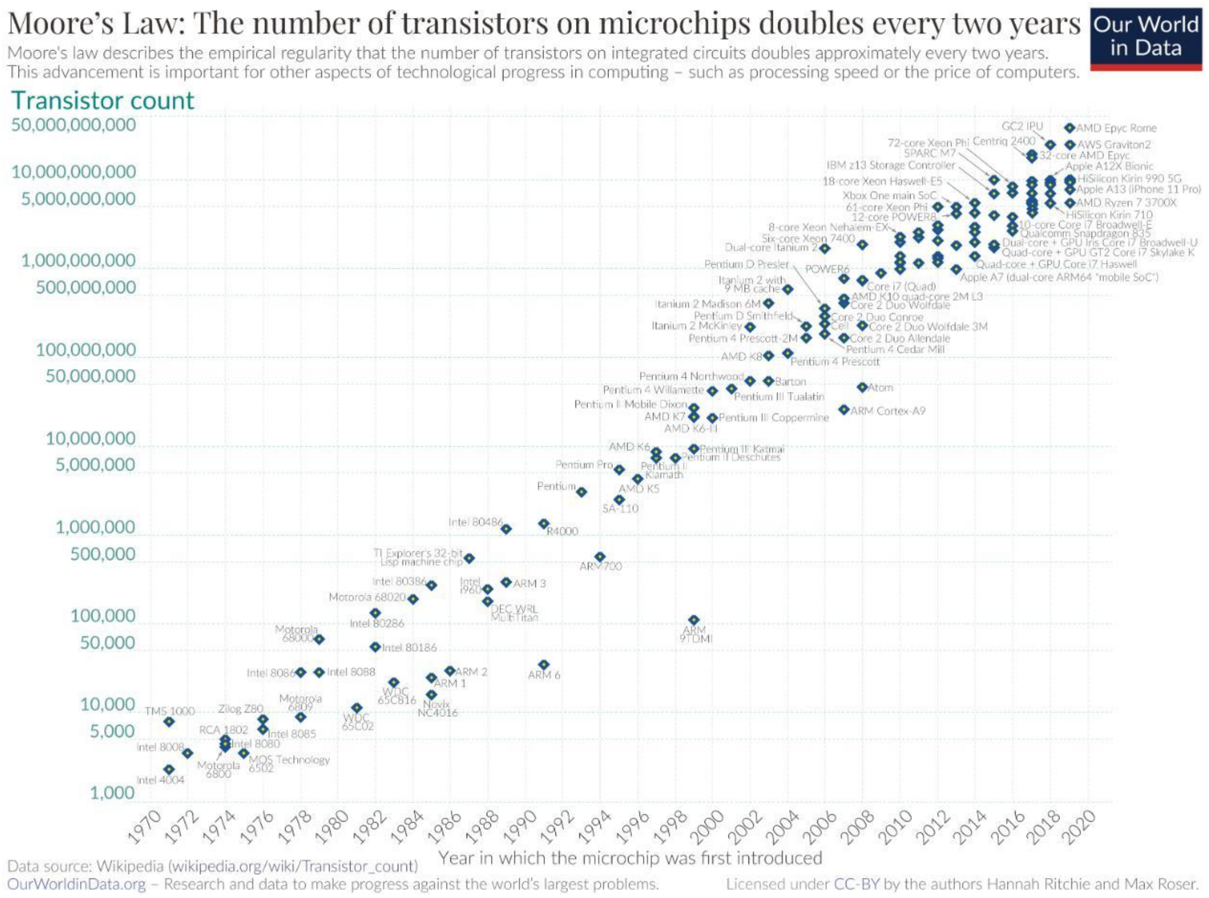
Why Computer Organization : Historical Trends

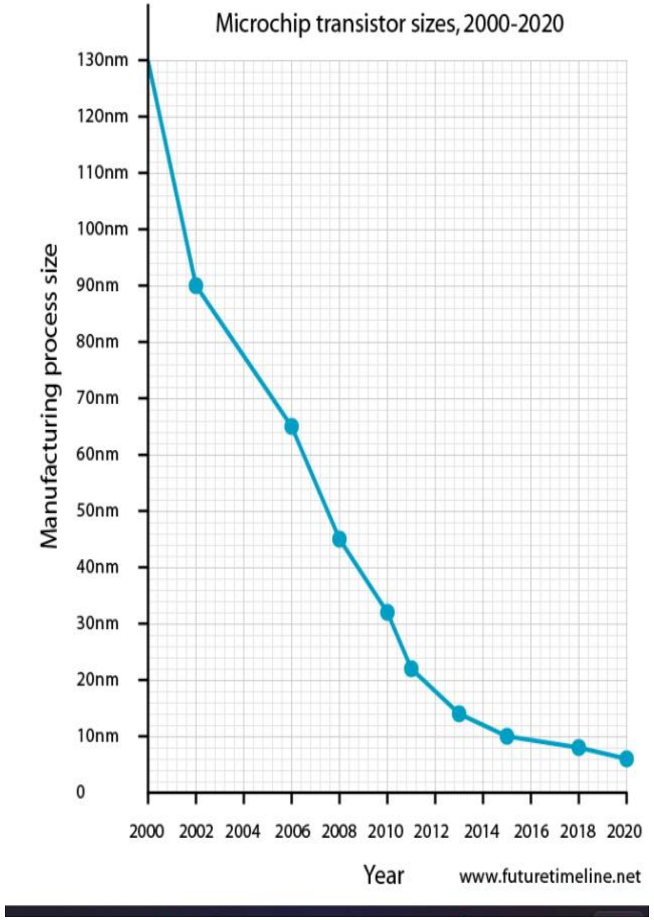


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp



Why Computer Organization : Historical Trends

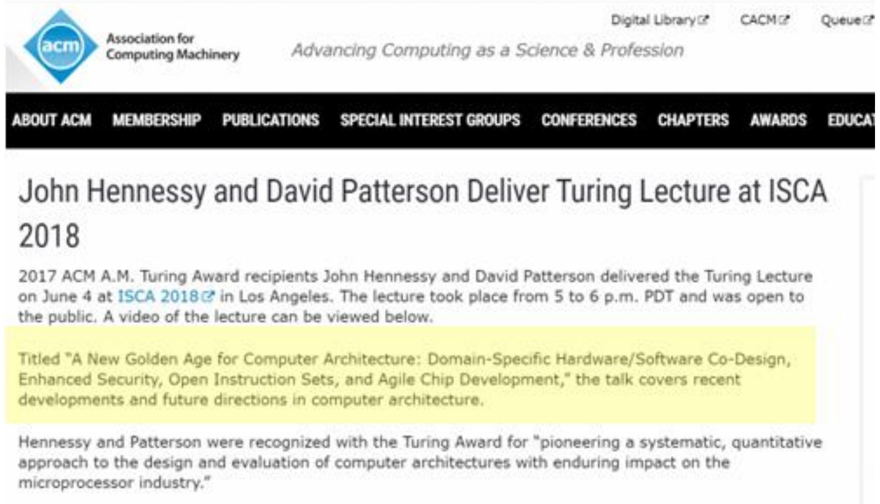




Today Trends : Computer Architecture and Organization

ACM Turing Awards

- The Turing Award is the most prestigious award in computer science – it is the Nobel Price of Computer Science.
- David A. Patterson and Joh L,. Hennessy received the Turing Award 2017 for their work on computer architecture and organization.



John Hennessy and David Patterson Deliver Turing Lecture at ISCA 2018

2017 ACM A.M. Turing Award recipients John Hennessy and David Patterson delivered the Turing Lecture on June 4 at ISCA 2018 in Los Angeles. The lecture took place from 5 to 6 p.m. PDT and was open to the public. A video of the lecture can be viewed below.

Titled "A New Golden Age for Computer Architecture: Domain-Specific Hardware/Software Co-Design, Enhanced Security, Open Instruction Sets, and Agile Chip Development," the talk covers recent developments and future directions in computer architecture.

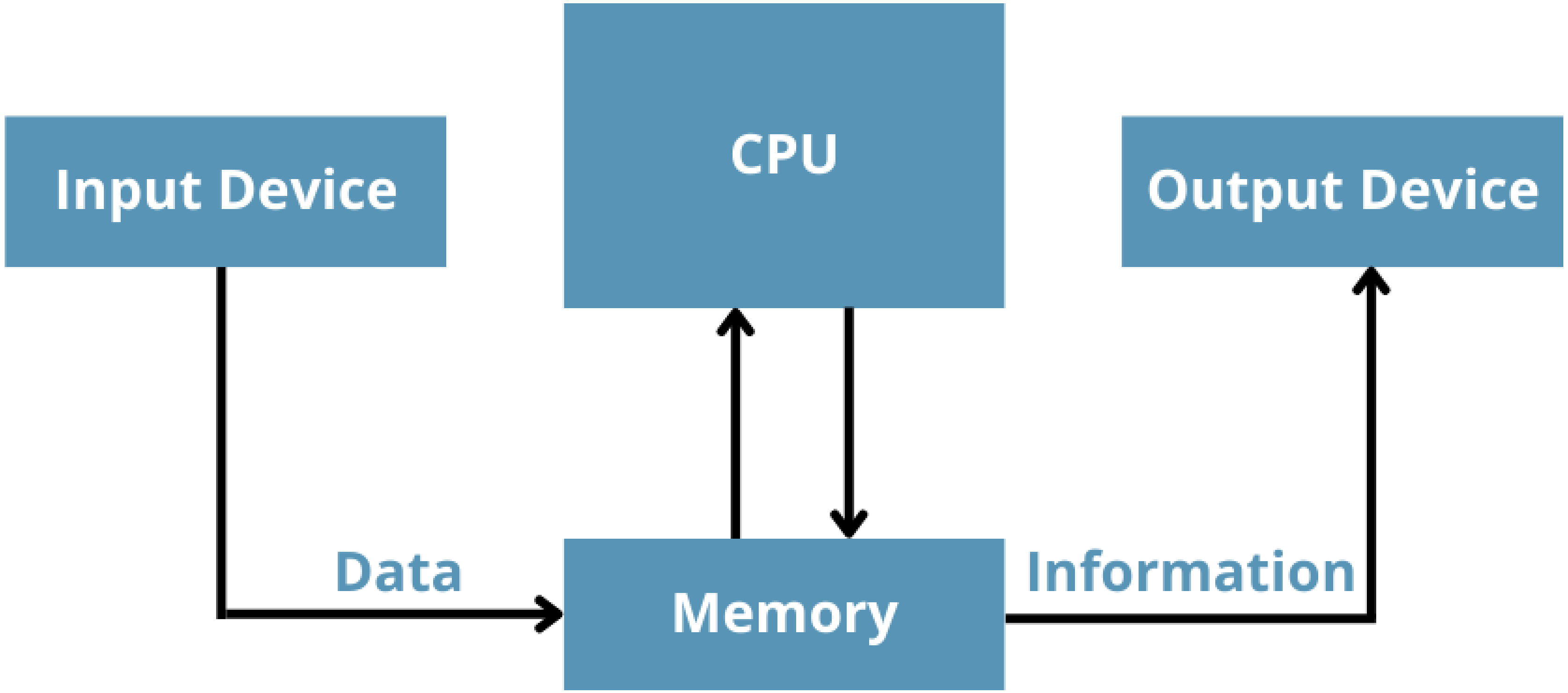
Hennessy and Patterson were recognized with the Turing Award for "pioneering a systematic, quantitative approach to the design and evaluation of computer architectures with enduring impact on the microprocessor industry."

See slides here: http://iscaconf.org/isca2018/docs/HennessyP attersonTuringLectureISCA4June2018.pdf

Computer Architecture Vs Computer Organization : Key Differences

Aspect	Computer Architecture	Computer Organization
Focus	What the computer does	How the computer does it
Key Components	Instruction Set Architecture (ISA), data types, registers, addressing	Control signals, memory tech, buses, ALU design, I/O mechanisms
User Interaction	Directly affects how a programmer writes software	Hidden from the programmer; affects performance & efficiency
Design Concern	Programming model	Hardware implementation & performance trade-offs
Influence Scope	Software and compiler developers	Hardware designers and architects
Analogy	Blueprint of a building	Construction and wiring of the building

Basic Functional Structure of a Computer System



Key Takeaways

•A computer system consists of interrelated components: CPU, memory, I/O devices, and system interconnections.
•Computer Architecture defines what a computer does (e.g., instruction set, data types, addressing modes).
• Computer Organization defines how the computer does it (e.g., control signals, memory technology, bus structures).
•Architecture is visible to programmers, while organization is more hardware-level and transparent to users.
• Understanding the distinction is critical for designing efficient, cost-effective, and compatible systems.

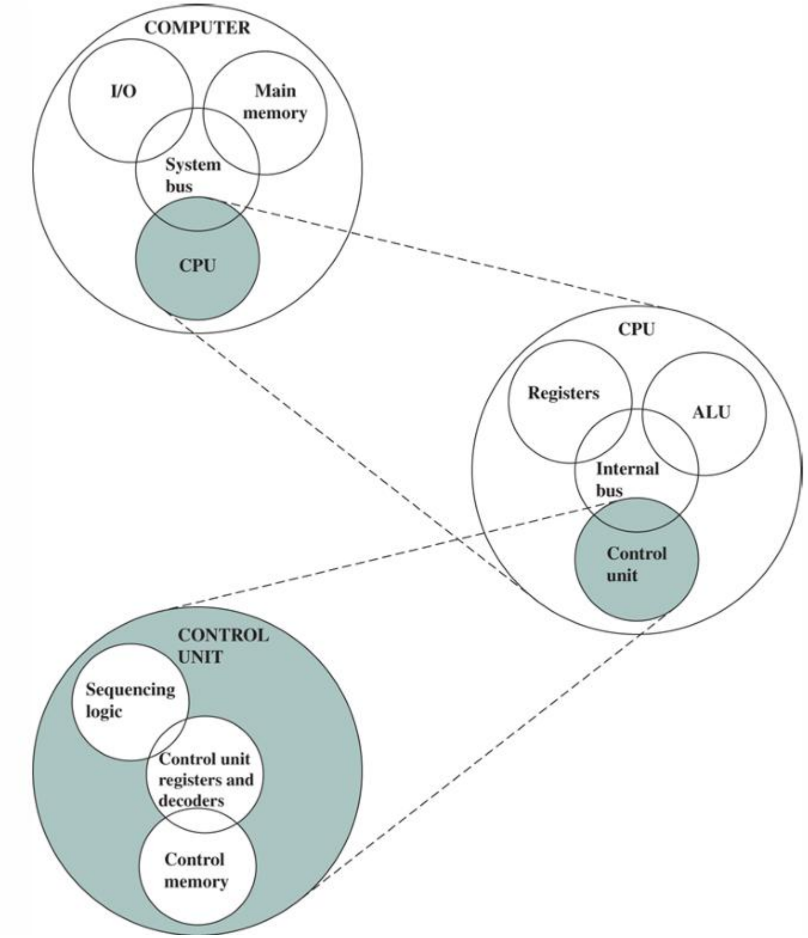
Evolution of Computer Systems:
Hardwired Control to Von Neumann Architecture

Objectives

At the end of this lecture you will be able to explain,

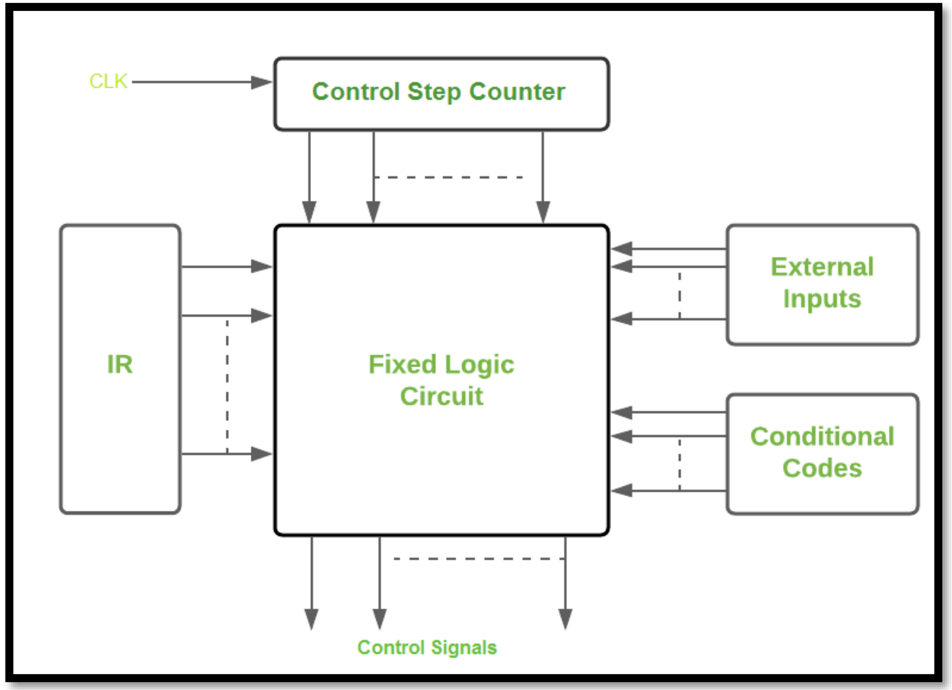
- Understand the basic components of the Von Neumann architecture.
- Explain the stored-program concept.
- Identify how data and instructions flow within a computer system.
- Recognize the limitations of the Von Neumann model

Simple Single-Processor Computer



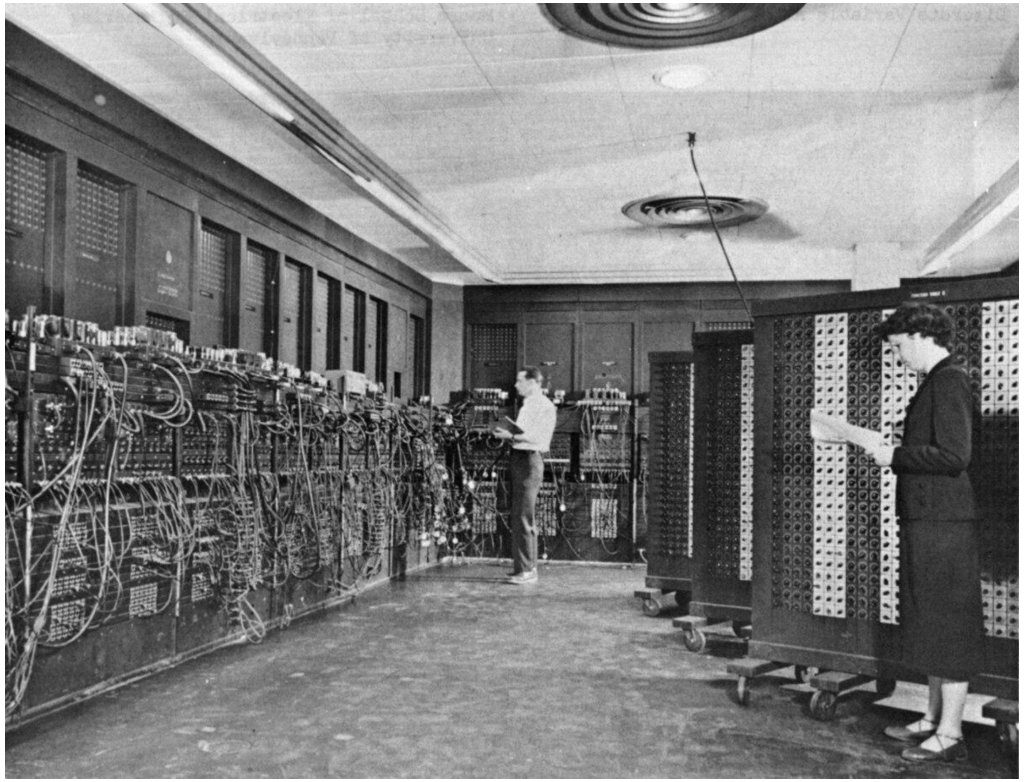
- **CPU (Central Processing Unit)**: Executes instructions and controls the operation of the computer.
- **Main Memory**: Stores both data and program instructions temporarily during execution.
- **I/O (Input/Output)**: Facilitates data exchange between the computer and the external environment.
- **System Interconnection**: Provides a communication path (like a system bus) for data transfer among CPU, memory, and I/O devices.

What is Hardwired Programming?



- Hardwired programming is an early form of computing where the logic for executing a program is physically built into the hardware.
- No software instructions were stored — all control was done through fixed electronic circuits.
- A method where program logic is hardcoded into the system.
- Programs were implemented by manually wiring or configuring circuits.

Characteristics of Hardwired Programming



- Not programmable by software :- instructions are fixed in hardware.
- Rewiring needed to change tasks :- physical changes to circuits.
- Used switches, cables, or relays to “program” the system.
- Designed for a single or limited set of tasks.
- Very limited in terms of functionality and flexibility.

Why Von Neumann Architecture?

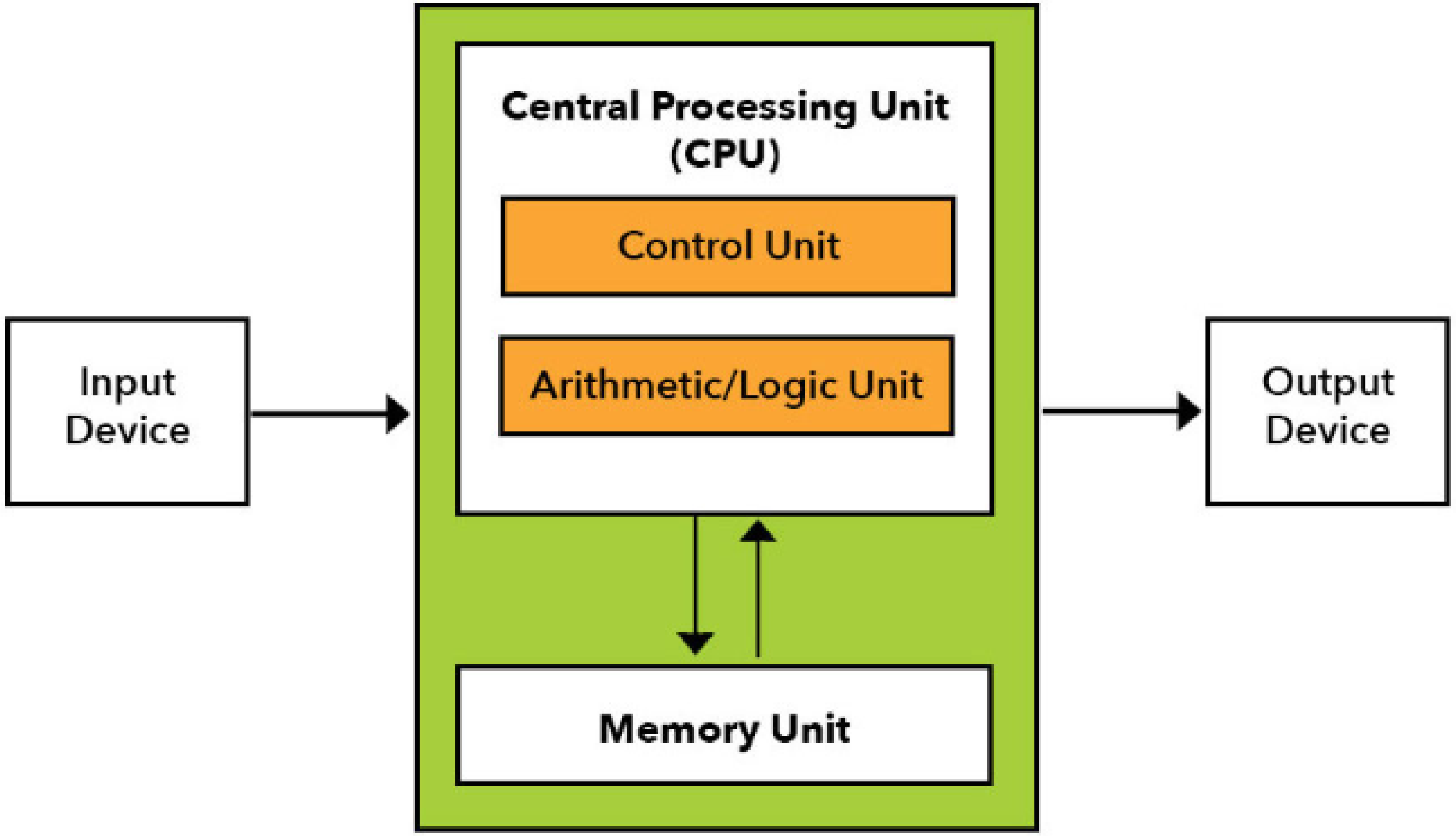
- **Hardwired systems were rigid and inflexible :-** Every new task required physical rewiring.
- **Reprogramming was slow and complex :-** Took days/weeks to change logic.
- **No memory for instructions :-** Instructions weren't stored, only executed through fixed circuits.
- **Von Neumann introduced the concept of a Stored-program computer** :- both data and instructions stored in memory.
- **This made computers more flexible, faster to reprogram, and general-purpose.**

First Stored Program Computer

John Von Neumann

- Father of Modern Computer Architecture.
- The Von Neumann architecture is a specific implementation of the stored program concept, proposed by John Von Neumann in 1945.

Von Neumann Architecture



Von Neumann Architecture Component Functions

- **Control Unit (CU) :-** Directs operations by fetching, decoding instructions, and controlling the execution cycle.
- **Arithmetic Logic Unit (ALU) :-** Performs all arithmetic operations and logical operations.
- **Memory Unit :-** Stores both data and instructions in the same memory.
- **Input Unit :-** Feeds data and instructions from the external world into the system.
- **Output Unit :-** Sends the processed data/results to the external world.
- **System Bus :-** Transfers data, addresses, and control signals among all units.

Main Components of Computer Functions

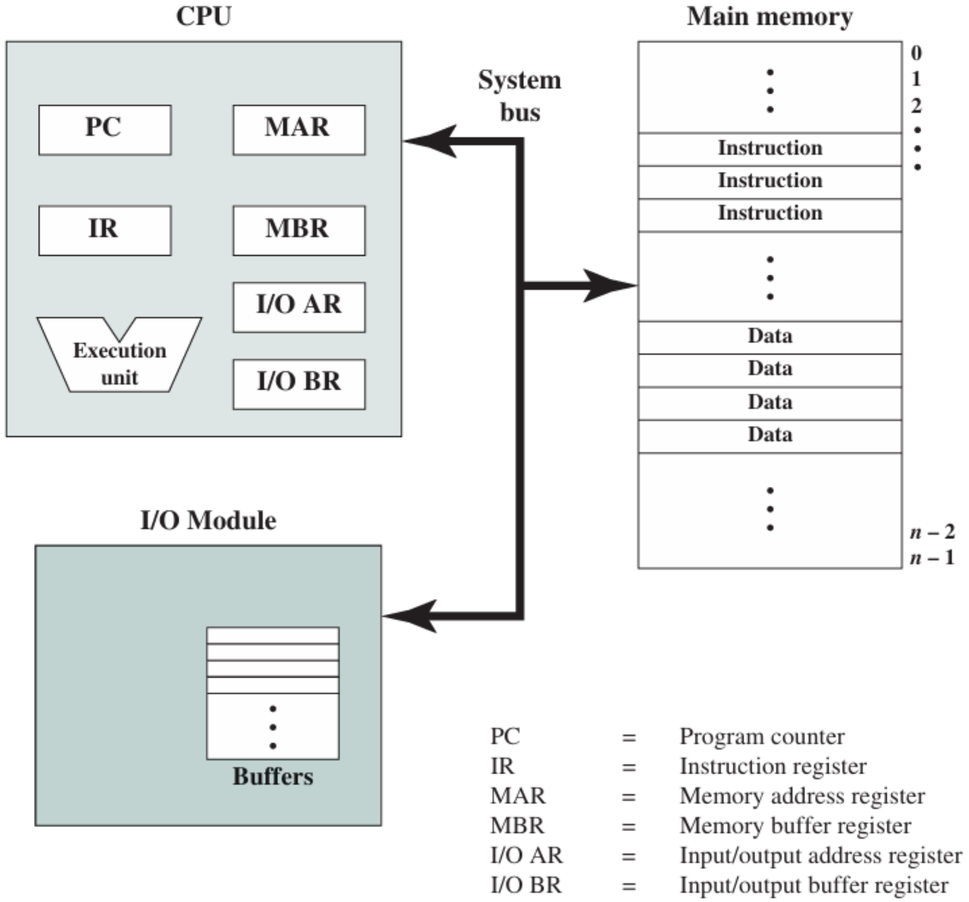


Figure 3.2 Computer Components: Top-Level View

Central Processing Unit

- CPU is like the brain of the computer.
- Main functions:- Fetch data and instructions from memory.
  :- Coordinate the complete execution of each instruction.
- Main Components of the CPU
  - ALU (Arithmetic and Logic Unit).
  - Control Unit.
  - Registers.

Registers in CPU

- Registers are small, high-speed storage units located inside the CPU.
- They temporarily store:

  -Data

  -Instructions

  -Intermediate results
- Modern CPUs have multiple registers, each with a specific role.
- Ex: AH, AL, BH, BL (General purpose registers in x86 architecture)

Types of Registers in CPU

Register Type	Purpose
General Purpose Registers	Temporarily store operands or results of operations
Special Purpose Registers	Used for controlling or monitoring program execution

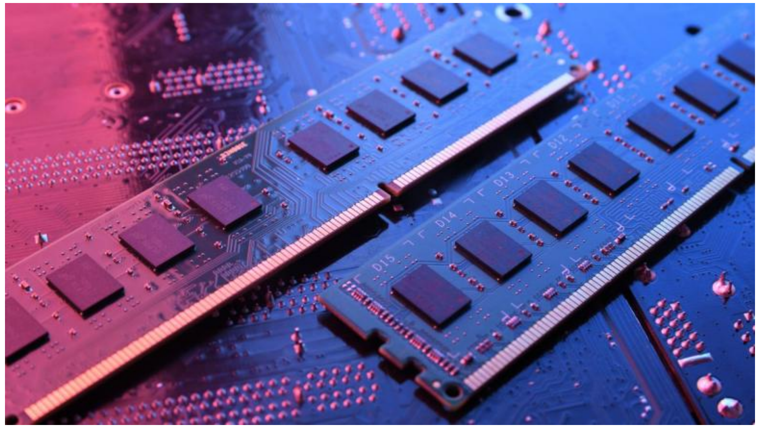
Note :- Special purpose registers include **PC, IR, MAR, MBR, Status Registers**.

Control Registers in CPU

Register	Function
Program Counter (PC)	Contains the address of the next instruction to be fetched
Instruction Register (IR)	Holds the currently executing instruction
Memory Address Register (MAR)	Stores the address in memory for the next read or write operation
Memory Buffer Register (MBR / MDR)	Temporarily holds data to/from memory during read/write
I/O Address Register (I/OAR)	Specifies the address of the I/O device involved in the data transfer
I/O Buffer Register (I/OBR)	Temporarily holds data being transferred to/from an I/O device and the CPU

Memory

- Memory is made up of storage cells, each capable of storing one binary digit (bit).
- A bit can be in one of two states: **0 or 1**, representing logical values.
- By combining multiple bits, we can store larger data such as numbers, characters, and instructions.
- Memory stores both data and instructions, as proposed in the Von Neumann model.

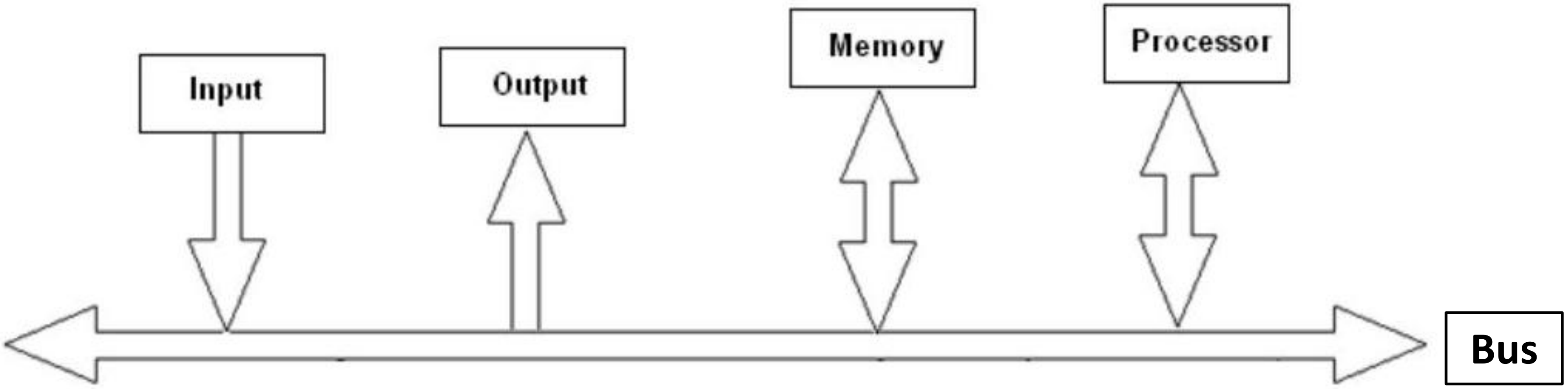


I/O Modules

- Acts as the communication bridge between the computer and the external world.
- **Input Function**: Accepts data and instructions from input devices (e.g., keyboard, mouse) and converts them into internal signals understandable by the system.
- **Output Function**: Converts internal signals into human- or machine-readable form and sends them to output devices (e.g., monitor, printer).
- Collectively called I/O components, they allow user interaction and external device integration.
- Essential for feeding data into the system and reporting results after processing.

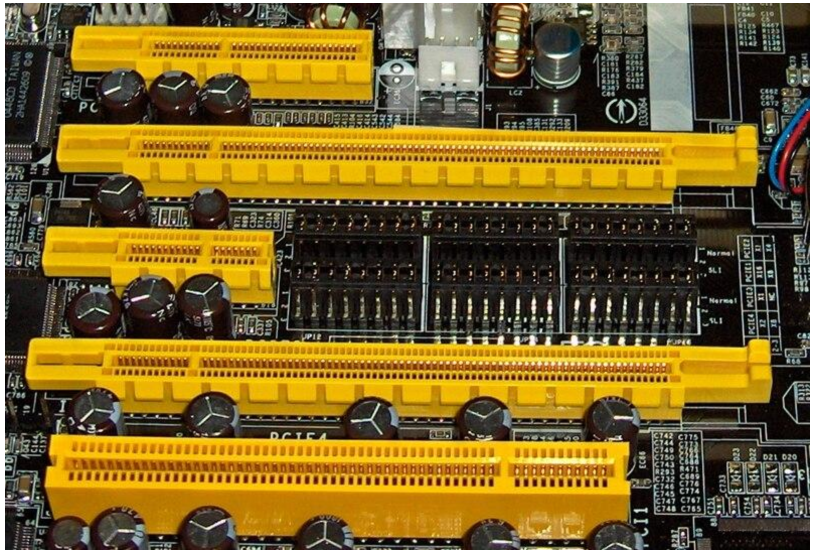
Bus Interconnection in Computer Systems

- A bus is a communication pathway connecting two or more devices.
- It is a shared transmission medium.
- Used in general-purpose computers and still common in embedded systems like microcontrollers.



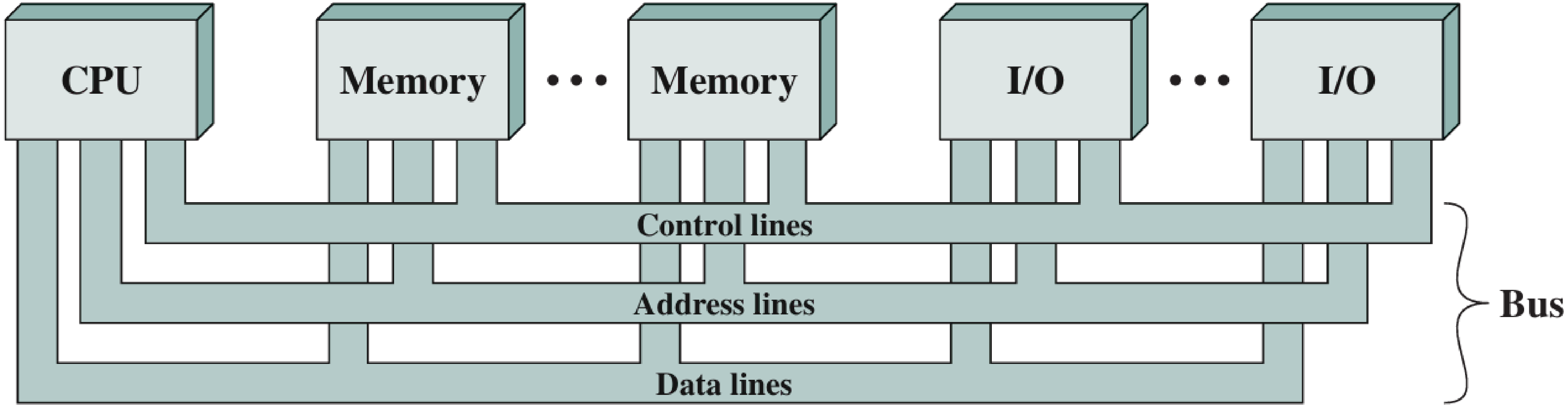
Key Characteristics of a Bus

- Shared medium: only one device can transmit at a time.
- If two devices transmit simultaneously, data gets garbled.
- Multiple lines transmit binary data — parallel communication.



Types of Computer Buses

- **System Bus**: Connects CPU, Memory, and I/O modules.
- A system bus typically contains 50–100+ lines.



What is Data Bus?

- Carries data between system modules.
- Width (Eg: 32-bit, 64-bit) determines how many bits are transferred at a time.
- Impacts system performance.
- Bidirectional

What is Address Bus?

- Specifies the source or destination of data.
- Determines memory capacity.
- Used for both memory and I/O addressing.
- Uni-directional

What is Control Bus?

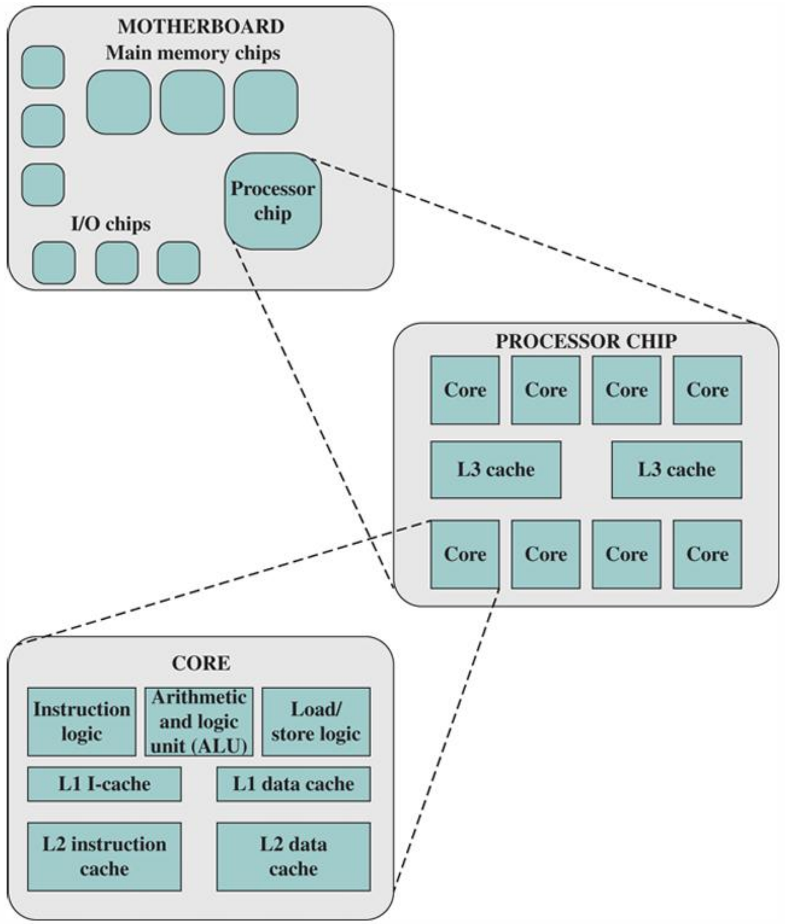
- Coordinates the operations of system modules.
- Sends command and timing signals.
- Important control signals:
  - Memory Read / Write
  - I/O Read / Write
  - Bus Request / Grant
  - Interrupt Request / ACK
  - Clock & Reset

Note :- Control Bus is crucial for managing the flow of data and instructions.
:- May be bi-directional depending on system design.

Operation of a Bus

- To send data:
  - Request control of the bus.
  - Transmit data over the data bus.
- To receive data:
  - Request control of the bus.
  - Send read request.
  - Wait for data response.

Multicore Computer Structure



- **Core**: Each core functions like an independent CPU — it includes instruction logic, an ALU, registers, and local cache (L1/L2), capable of executing programs independently.
- **L3 Cache**: Shared, high-speed memory used by all cores to reduce access time to frequently used data and instructions.
- **Processor Chip**: A physical silicon component that integrates multiple cores and shared cache, enabling parallel processing and efficient multitasking.
- **System Board (Motherboard)**: Hosts the processor chip, memory slots, I/O controllers, and expansion slots to interconnect all major components of a computer system.

Key Takeaways

- The Von Neumann Architecture introduced the stored-program concept, enabling flexible and general-purpose computing.
- Core components of the Von Neumann model:
- CPU, Memory, I/O modules, and Buses.
- The CPU consists of Control Unit, ALU, and Registers that manage instruction execution.
- Memory stores both data and instructions in binary format.
- I/O Modules handle data exchange with external devices.
- The System Bus (data, address, control lines) enables communication among all system parts.

Evolution of Computer Generations

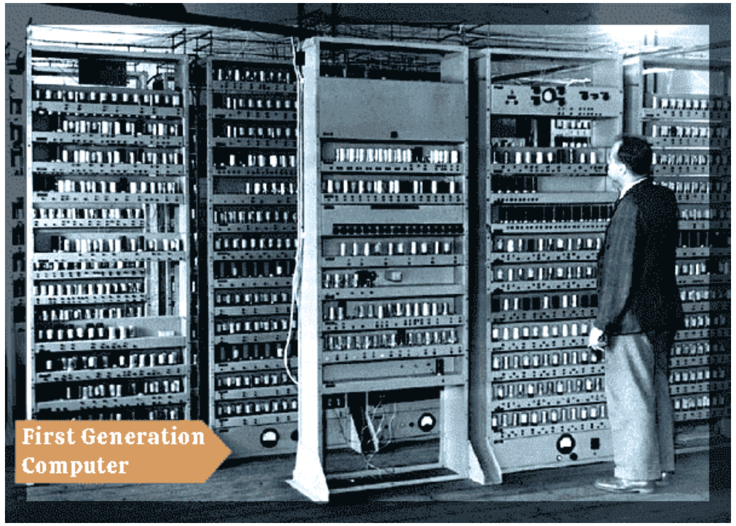
Objectives

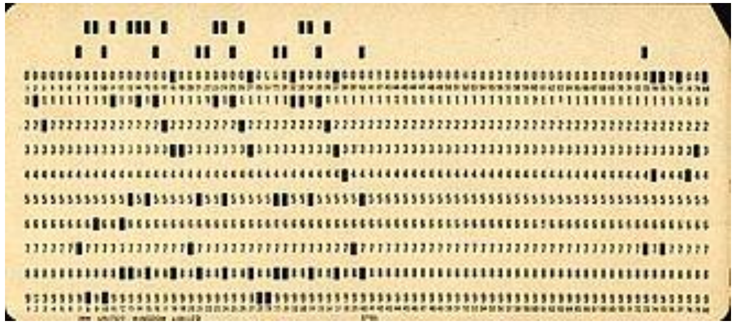
At the end of this lecture you will be able to explain,

- Understand the evolution of computer systems from the first to the sixth generation.
- Distinguish between key hardware and software trends across generations.
- Explain the importance and evolution of Intel x86 and ARM architectures.
- Recognize the characteristics and applications of embedded systems.
- Describe the fundamentals of cloud computing, its service models, and benefits/challenges.

First Generation (≈1940–1956) – Vacuum Tubes

- **Hardware:** Vacuum tubes
  : Magnetic drums
  : Punched cards
- **Software:** Only machine language
  : no OS
- **Systems:** ENIAC, UNIVAC, IBM 701
- **Characteristics:** Massive size
  : High power consumption
  : Reliable
  : Slow I/O





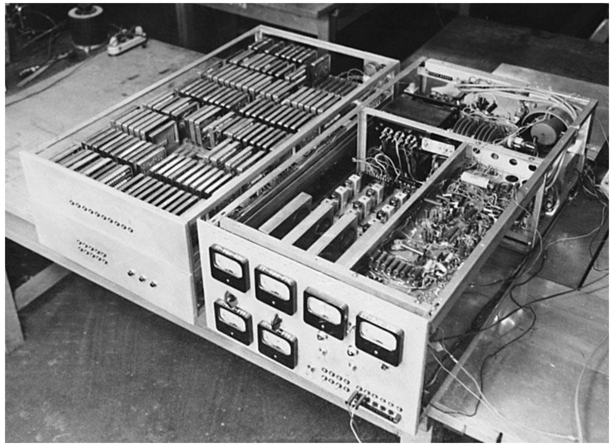
Punched Card

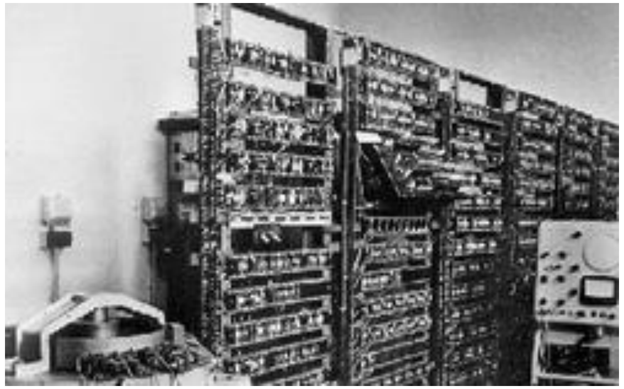
What is IAS Computer?

•**Built at**: Princeton Institute for Advanced Studies (designed 1946, completed 1952).
•**Key Designer**: John von Neumann (with colleagues).
•**Concept**: Introduced the **stored-program concept**
•**Prototype**: Served as a model for most later general-purpose computers.
•**Architecture** (Von Neumann architecture):- Main memory
:- Arithmetic and Logic Unit
:- Control unit.
:- Input/output equipment.
•**Legacy**: Almost all modern computers still follow this same architecture.

Second Generation (≈1956–1963) – Transistors

- **Hardware:** Transistors
  : Magnetic-core memory
- **Software:** Assembly language
  : Batch-processing OSs
- **Systems:** IBM 1401, IBM 7090
- **Characteristics:** Smaller
  : Faster
  : More Reliable
  : Lower Heat/Power





What is Batch Processing OS?

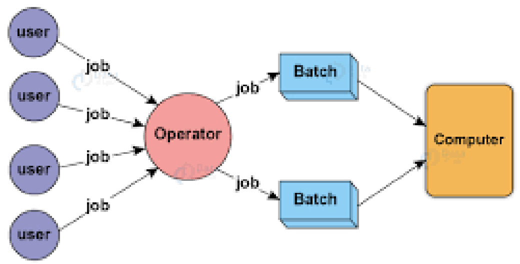
•**Definition**: A type of operating system where jobs (programs + data) are collected together (in batches) and executed one after another, without user interaction during execution.

•**How it worked**:- Users submitted their programs (often on punched cards).

:- The OS grouped these jobs into a batch.

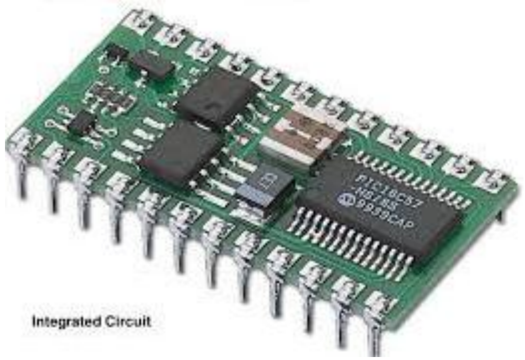
:- The computer executed them sequentially.

•**Key Point**: No real-time interaction. The computer works on one batch until it's done, then moves to the next.



Third Generation (≈1964–1975) – Integrated Circuits

•**Hardware:** ICs (SSI, MSI)
: Keyboards/monitors
•**Software:** High-level languages
: Early OS
: Time-sharing
•**Systems:** IBM 360, PDP-8, CDC 6600
•**Characteristics:** Multiprogramming
: Improved efficiency
: Lower cost



Forth Generation (≈1975–1989) – Microprocessors

- **Hardware:** LSI/VLSI chips
  : Microprocessors
  : Optical/floppy disks
- **Software:** GUI-based OS (e.g. UNIX, early MS-DOS)
  : More interactive
- **Systems:** IBM PC, Apple II
- **Characteristics:** Personal computing begins
  : Compact
  : Affordable
  : Upgradeable

Fifth Generation (≈1989–2010) – ULSI, Networking & AI Foundations

•**Hardware:** ULSI chips
: Internet connectivity
: Optical storage
: Multimedia support

•**Software:** GUI OS with emerging AI features
: Voice/handwriting recognition

•**Systems:** Laptops, notebooks, SUN workstations, early AI-oriented systems

•**Characteristics:** Rich multimedia
: Portable
: Highly reliable
: Network-capable

Sixth Generation (≈2010–Now)

Hardware Trends: Ultra-scale integration
: Multicore
: GPGPUs (graphics processing)
: IoT devices
: Cloud servers

Software Trends: AI and machine learning–embedded software
: Parallel computing frameworks
: Virtualization
: Containerization

Systems: Smartphones, embedded systems, high-performance servers, edge devices, RISC-V and ARM platforms

Characteristics: Massive parallelism, energy-efficient performance, intelligent systems, ubiquitous connectivity

Summary

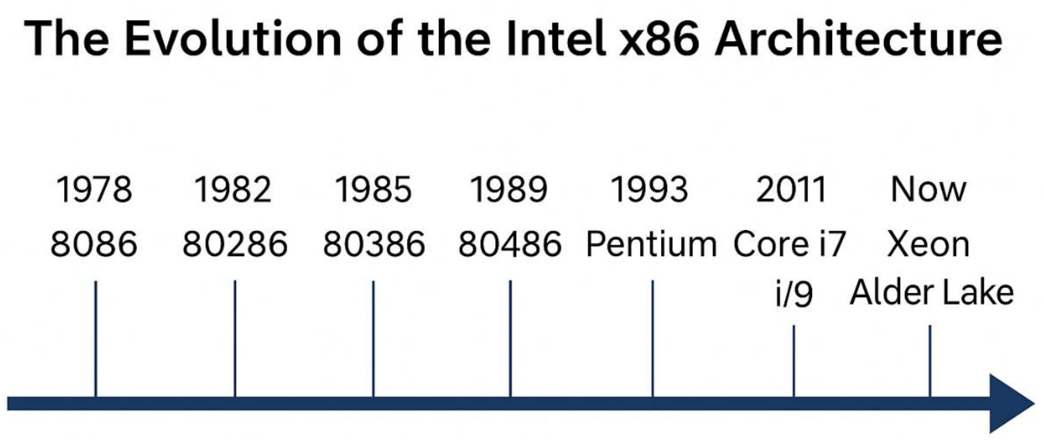
Generation	Time Frame	Hardware	Software/OS	Key Systems	Characteristics
1st	1940–1956	Vacuum tubes, drums	Machine language	ENIAC, UNIVAC	Huge, unreliable, high power
2nd	1956–1963	Transistors, core memory	Assembly, batch OS	IBM 1401, 7090	More reliable, lower cost/power
3rd	1964–1975	ICs, input/output devices	HLL, OS, time-sharing	IBM 360, PDP-8	Efficient, programmable
4th	1975–1989	Microprocessors, LSI/VLSI	GUI OS, interactive tools	IBM PC, Apple II	Personal computing era
5th	1989–2010s	ULSI, network connectivity	AI-capable GUIs	Laptops, workstations	Multimedia, portable, networked
6th+	2010s–Present	Multicore, GPGPU, cloud, IoT	Virtualization, AI frameworks	Smartphones, servers, IoT	Parallel, intelligent, connected

Intel x86 Architecture

- x86: a family of backward-compatible instruction set architectures.
- First introduced by Intel in 1978 (8086).
- Still dominant in desktops, laptops, and servers.
- Backward compatibility means code from decades ago can still run.

Why Study x86 Evolution?

- Shows how computer hardware evolved.
- Demonstrates the impact of Moore's Law.
- Helps understand modern processor features.
- Links to computer organization concepts: registers, memory, pipelines, parallelism.



Intel x86 Evolution

Generation	Year	Word Size	Clock Speed	Key Features
8086	1978	16-bit	5–10 MHz	Segmented memory
80286	1982	16-bit	6–25 MHz	Protected mode
80386	1985	32-bit	12–40 MHz	Virtual memory
80486	1989	32-bit	20–100 MHz	On-chip FPU
Pentium	1993	32-bit	60–300 MHz	Superscalar
Core	2006+	64-bit	GHz range	Multi-core

Embedded System

An embedded system is a computer system designed for a specific function within a larger device, often with real-time computing constraints.

Example :- Washing Machine Controller
:- ATM Machine
:- Digital Camera
:- Automotive engine control

Key Characteristics of Embedded System

- **Specific Function:** Designed for a dedicated purpose.
- **Real-Time Operation:** Must respond within strict timing limits.
- **Resource Constraints:** Limited memory, CPU speed, and power.
- **Reliability:** Must operate continuously without failure.
- **Low Power Consumption:** Especially in battery-powered devices.

Basic Architecture of Embedded System

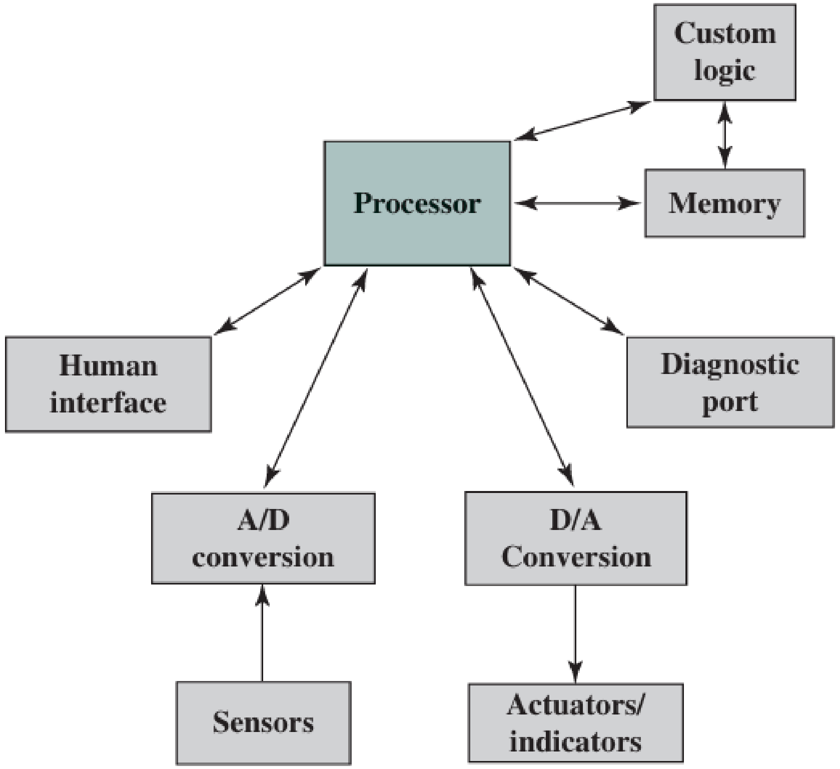


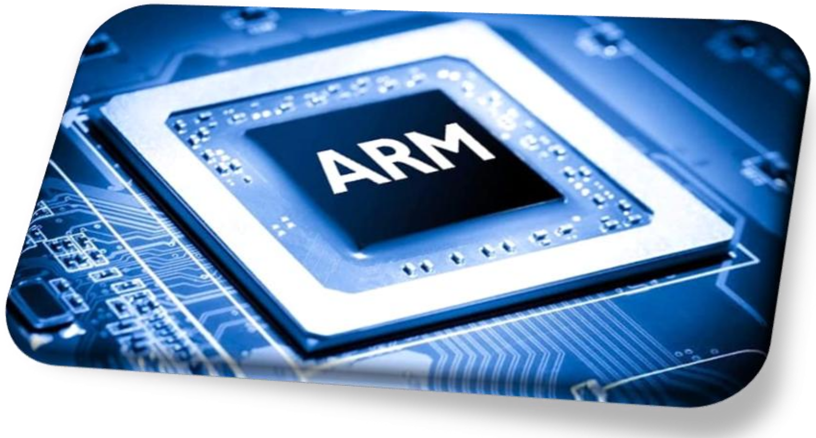
Figure 1.14 Possible Organization of an Embedded System

Types of Embedded System

Type	Processor / Capability	Real-World Examples
Small-Scale	8- or 16-bit microcontroller, limited memory	Simple home appliances (microwave ovens, washing machines)
Medium-Scale	16- to 32-bit controllers, RTOS-capable	ATMs, industrial controllers
Sophisticated / Complex	32- to 64-bit processors, multi-function, networked	Industrial robots, advanced instrumentation

ARM Architecture

- ARM = Advanced RISC Machines (originally Acorn RISC Machine)
- Based on RISC principles (Reduced Instruction Set Computer)
- Designed by ARM Holdings (Cambridge, UK)
- ARM Holdings licenses designs - does not manufacture chips
- Used in embedded systems, mobile devices, and IoT products



What is RISC Machine?

- RISC = **Reduced Instruction Set Computer**
- Processor with a **small, simple set of instructions**
- Goal: **faster execution** (usually 1 instruction per clock cycle)

Key Features:- Simple instructions

:- Load/Store architecture (memory only with LOAD / STORE)

:- Many registers → less memory use

:- Pipeline friendly → multiple instructions in parallel

Why ARM Architecture is Popular?

- High performance with low power consumption
- Small die size → fits in compact devices
- Cost-effective for mass production
- Found in:- Smartphones (Android, iPhone)
  :- Gaming systems
  :- Smart home devices
  :- Industrial controllers

Evolution of ARM Architecture

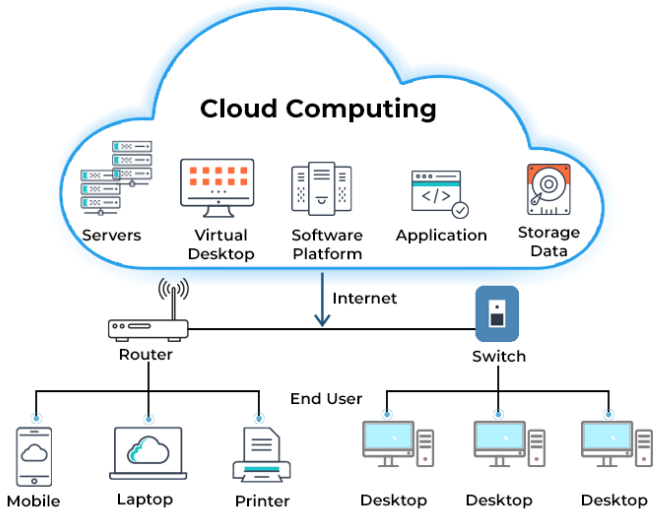
- Early 1980s: Developed by Acorn Computers for BBC Micro project
- 1985: ARM1 — first RISC processor
- 1990: ARM Ltd formed (Acorn + VLSI + Apple)
- Evolved into the world's most widely used processor architecture

ARM Cortex Product Families

Family	Target Use	Key Features	Memory Management	Examples
Cortex-A	Application processors (smartphones, tablets, smart TVs, home gateways)	High performance, supports OS (Linux, Android, Windows), runs at >1 GHz	MMU (supports virtual memory & paging)	Cortex-A53 (64-bit), Cortex-A72
Cortex-R	Real-time systems (automotive control, storage controllers, industrial automation)	Low latency, predictable timing, high reliability	MPU only (no MMU)	Cortex-R4, Cortex-R7
Cortex-M	Microcontrollers (IoT devices, sensors, embedded systems)	Ultra-low power, deterministic interrupts, compact design	MPU only, Thumb-2 ISA	Cortex-M0, M0+, M3, M4

Cloud Computing

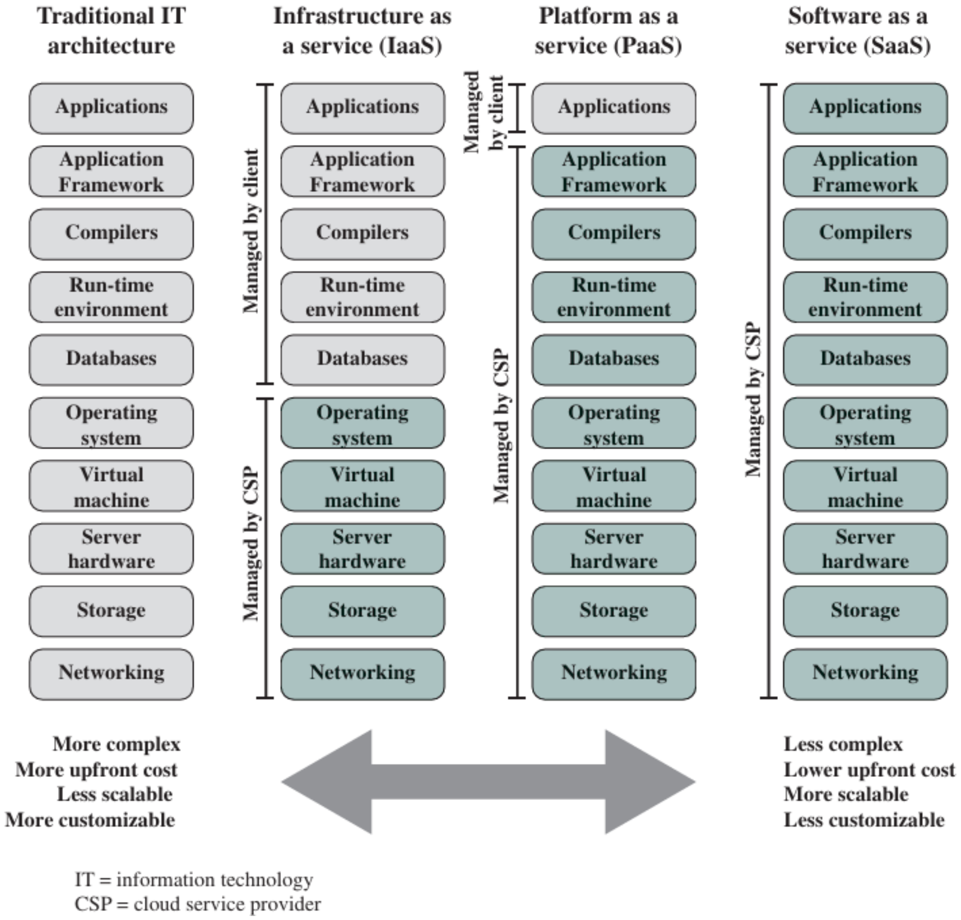
- A model for enabling ubiquitous, convenient, on- demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction *(NIST SP- 800-145)*



Key Characteristics of Cloud Computing

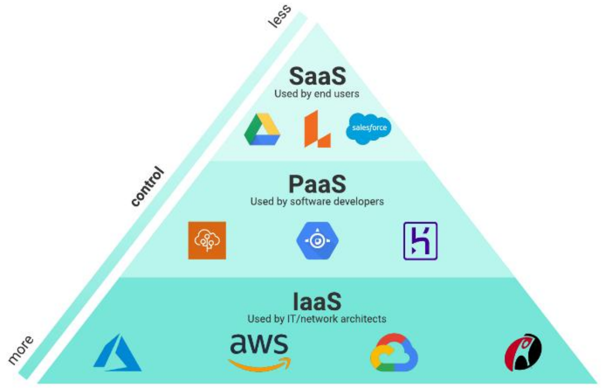
- On-Demand Self-Service – resources available instantly
- Broad Network Access – accessible from anywhere with internet
- Resource Pooling – multiple customers share the same infrastructure
- Rapid Elasticity – scale resources up/down quickly
- Measured Service – pay for what you use

Cloud Service Models



Examples

- **IaaS :-** AWS EC2, Google Compute Engine
- **PaaS :-** Google App Engine, Microsoft Azure App Service
- **SaaS :-** Gmail, Microsoft 365, Dropbox



Advantages And Disadvantages of Cloud Computing

Advantages:- Reduced hardware & maintenance costs
:- Scalability & flexibility
:- Accessible anywhere
:- Disaster recovery & backup options

Challenges:- Security and Privacy Concerns
:- Internet Dependency
:- Downtime/Service Outages
:- Limited control over resources
:- Hidden Costs
:- Compliance and Legal Issues

Key Takeaways

•Computer generations reflect advancements in hardware, software, and system capabilities.

•Intel x86 remains a cornerstone in desktop, laptop, and server computing due to backward compatibility.

•Embedded systems are specialized, efficient, and found in numerous everyday devices.

•ARM architecture dominates mobile and IoT markets due to its performance–power balance.

•Cloud computing offers scalable, on-demand access to shared resources with diverse service models.