William Stallings Computer Organization and Architecture Compiled and Updated by: Bilal Ahmed

Chapter 1

Basic Concepts and Computer Evolution

Computer?

- An electronic machine, which:
  - Takes Input
  - Processes it
  - Produces output

Different Types of Computers?

- Super Computer
- Mainframe Computer
- Server Computer
- Mini Computer
- Micro Computer

Buzz words!

- Architecture
- Organization
- Instruction set
- Structure
- Function

Computer Organization Vs Architecture

- **Computer architecture**: what the programmer sees – things like instructions, data types, and how memory is accessed. It affects how programs run.
- **Computer organization:** how everything works behind the scenes – the parts inside the computer (like the processor, memory, and how they connect) that make the architecture work.

- **Architecture:** what the system *does* (from a programmer's view)
- **Organization:** how the system *does it* (the internal setup)

Computer Architecture

- **Computer Architecture** refers to features of a computer that a programmer can see and use, such as:
- The **instruction set** (what commands the computer understands)
- The **number of bits** used to represent data (e.g., 32-bit or 64-bit)
- **Input/Output (I/O)** methods
- **Addressing techniques** (how memory is accessed)
- It answers the question:
  "What can the computer do?"
- For example:
  Does it have a **multiply** instruction?

Computer Organization

Computer Organization is about **how** the computer's features are **built and work**.
It includes:

- The actual **hardware components** (like RAM, hard drive, graphics card)
- **Peripheral devices** (how things like the mouse and keyboard are connected)
- **Control signals** (used to manage how parts of the computer work together)
- How instructions are **carried out in hardware**

It answers the question:
"How does the computer do it?"
For example:
Does it have a **hardware unit** to multiply, or does it do it by **repeated addition**?

Real-life Analogy: House Design

Concept	Architecture	Organization
House Design	Blueprint (how many rooms, layout, etc.)	How the builder constructs it (materials, wiring, plumbing layout)
What You See	Floor plan, room types	Wall insulation, pipe layout, electrical routing
Consistency Across	All houses of same model look the same outside	But may use different building materials

Real-life Example: DLD

Aspect	Architecture	Organization
What it does	Performs logic and arithmetic ops	Uses logic gates, adders, MUXes
Defined by	Operation table / truth table / spec	Circuit diagram / internal wiring
Concerned with	Behavior visible to the outside	Physical implementation
Changes often?	No (spec is fixed for a design)	Yes (can optimize or redesign internals)

Real-life Example: Computing

Example	**Architecture** (What it does)	**Organization** (How it's built)
Computing	**Intel x86 ISA** – instructions like MOV, ADD, etc.	**Intel Core i5 vs i7** – different cores, cache size, clock speed
Mobile Chips	**ARMv8 ISA** – common instruction set in smartphones	**Apple M1 vs Snapdragon 8 Gen 2** – different designs, performance, battery life
RISC-V CPU	**RISC-V ISA** – open standard with defined instructions	One RISC-V CPU may use **pipelining**, another may be **single-cycle**
GPU Systems	**CUDA or OpenCL Programming Model** – defines parallel compute operations	**NVIDIA RTX vs AMD Radeon** – different core architecture, memory bandwidth, execution units

ARCHITECTURE	ARRIVAL	TYPE	IMPACT
Von Neumann	1945	Conceptual	Foundation of modern computing model
Harvard	~1950s	Conceptual	Crucial for embedded and parallel fetch systems
IBM System/360	1964	CISC	Standardized ISA, large influence on mainframes
Intel x86	1978	CISC	Dominated desktops and later servers
MIPS	1985	RISC	Simple and used in education, embedded
ARM	1985	RISC (Harvard-like)	Mobile/embedded dominance
SPARC	1987	RISC	High-performance servers
PowerPC	1992	RISC	Macs, consoles, industrial
x86-64	2003	CISC (64-bit)	Still dominant on desktop/server today
RISC-V	2010	RISC (Open-source)	Rapidly growing in education and industry

Tools

- Assembly
- Verilog
- Circuit Verse
- Digital JS
- Ripes

Concept of computer ‘Family’

A computer family is a category of computers with similar designs and microprocessors that are compatible.

Good examples of different computer families are the IBM or PC family versus the Apple or Mac family of computers

Mostly, all computer families have successive generations, they do share the same architecture. But there are always exceptions

Structure & Function

- Computer is a complex system with millions of elementary electronic components
- How can one describe them? Key is to recognize the hierarchical nature of the system
- Hierarchical system: a set of interrelated subsystems
- At each level of hierarchy designer is concerned with two things:
  - **Structure:** the way in which components relate to each other
  - **Function:** the operation of individual components as part of the structure
- The author takes a top down approach to explain the system

Computer

Major structural components:

CPU – controls the operation of the computer and performs its data processing functions

Main Memory – stores data

I/O – moves data between the computer and its external environment

System Interconnection – some mechanism that provides for communication among CPU, main memory, and I/O

CPU

Major structural components:

- Control Unit
  - Controls the operation of the CPU and hence the computer
- Arithmetic and Logic Unit (ALU)
  - Performs the computer's data processing function
- Registers
  - Provide storage internal to the CPU
- CPU Interconnection
  - Some mechanism that provides for communication among the control unit, ALU, and registers

Function

- There are four basic functions that a computer can perform:
  1. Data processing
     - Data may take a wide variety of forms and the range of processing requirements is broad
  2. Data storage
     - Short-term
     - Long-term
  3. Data movement
     - Input-output (I/O) - when data are received from or delivered to a device (peripheral) that is directly connected to the computer
     - Data communications – when data are moved over longer distances, to or from a remote device
  4. Control
     - A control unit manages the computer's resources and orchestrates the performance of its functional parts in response to instructions

Structure

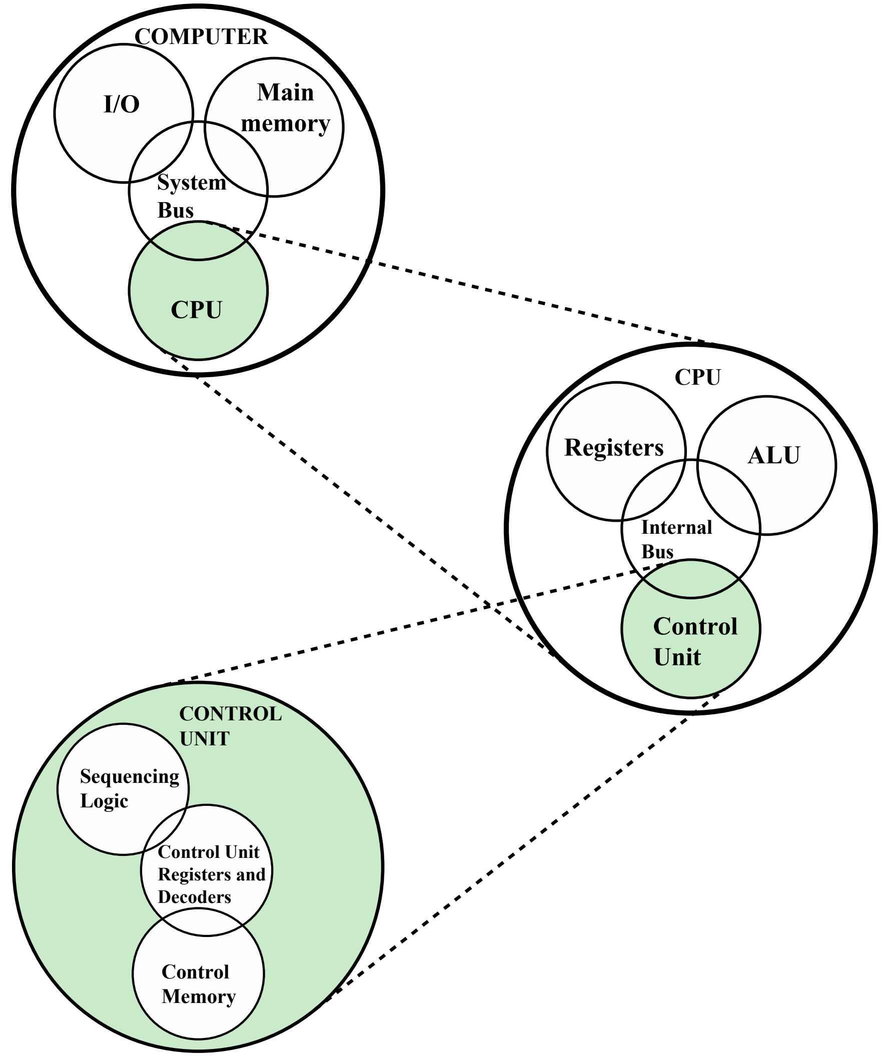


Figure 1.1 A Top-Down View of a Computer

The Control Unit

Sequence Logic Circuit: This type of circuit's output depends not only on the current input but also on past inputs.

Control Memory: It stores the microprogram in a microprogrammed control unit.

Writable Control Memory: This is control memory that can be changed, allowing the microprogram and instruction set to be modified.

Control Unit: It uses registers and decoders to convert instructions into control signals that guide other parts of the computer, like memory, the arithmetic unit, and input/output devices.

Example: Instruction ADD R1, R2

- **1. Sequential Logic Circuit**
- This remembers **which step** we are on while executing the instruction.
- Example:
  - Step 1: Fetch the instruction
  - Step 2: Decode it
  - Step 3: Execute it
- The sequential logic keeps track of these steps (like a state machine).

Example: Instruction ADD R1, R2

2. Control Memory

- Inside control memory, there is a **microprogram** (a set of small steps) for ADD.
- Example micro-steps stored:
    1. Get operand from R1
    2. Get operand from R2
    3. Tell ALU to add them
    4. Store result back into R1

Example: Instruction ADD R1, R2

- **3. Writable Control Memory**
- If the CPU designer wants to **change how ADD works** (for example, to support floating-point addition instead of integer), they can update the writable control memory.
- This makes the CPU more flexible compared to hardwired logic.

Example: Instruction ADD R1, R2

4. Control Unit

- The control unit is the **traffic manager**:
    - It reads the ADD R1, R2 instruction.
    - It **decodes** it into signals (like "read R1," "read R2," "tell ALU to add," "write back to R1").
    - These signals are sent to the **registers** and **ALU** at the right time.

Summary

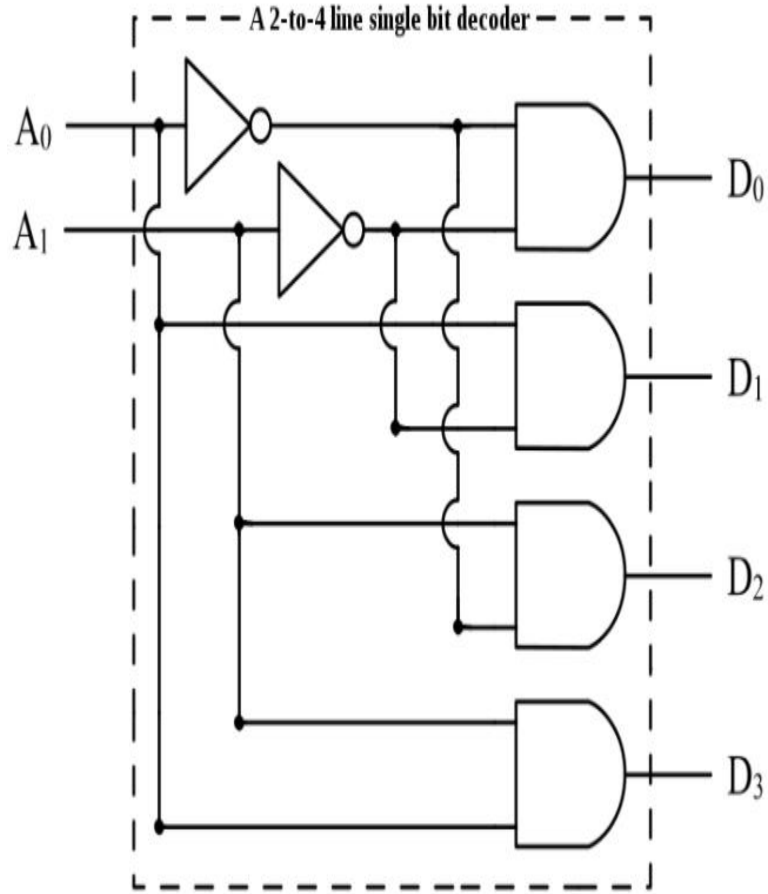
Sequential circuit → Keeps track of which micro-step we're in.

Control memory → Stores the recipe (set of micro instructions) for how to perform ADD.

Writable control memory → Lets us update/change that recipe.

Control unit → Sends actual signals to ALU, memory, registers.

Control Unit Registers and Decoders (2 to 4 bit Decoder)



Truth Table

A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Minterm Equations

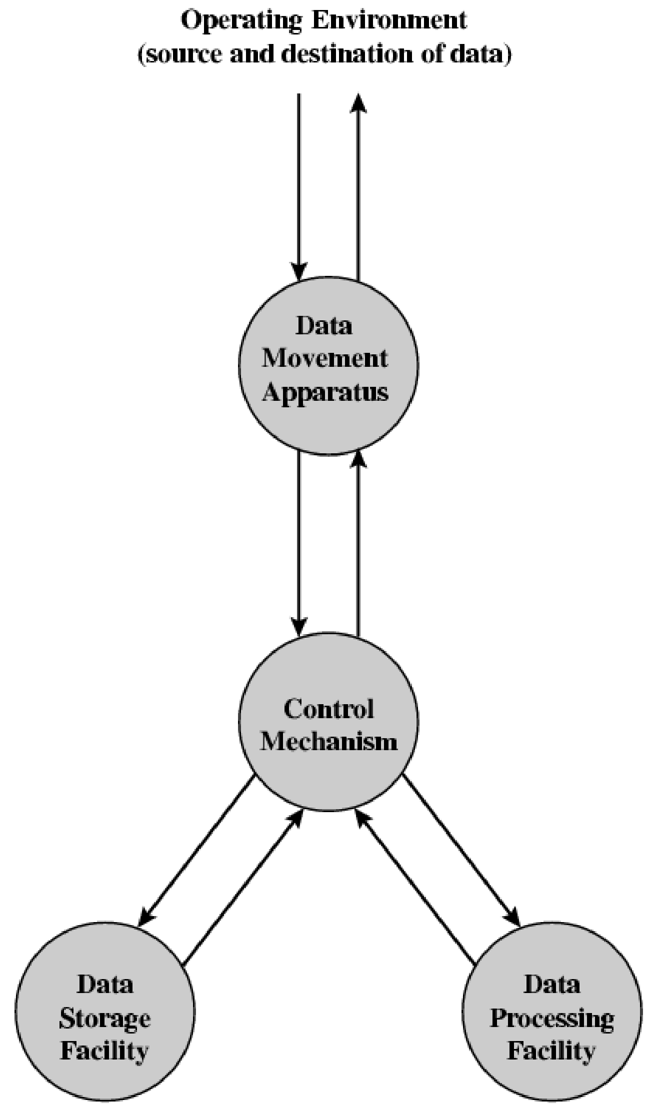
$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = \overline{A_1} \cdot A_0$$

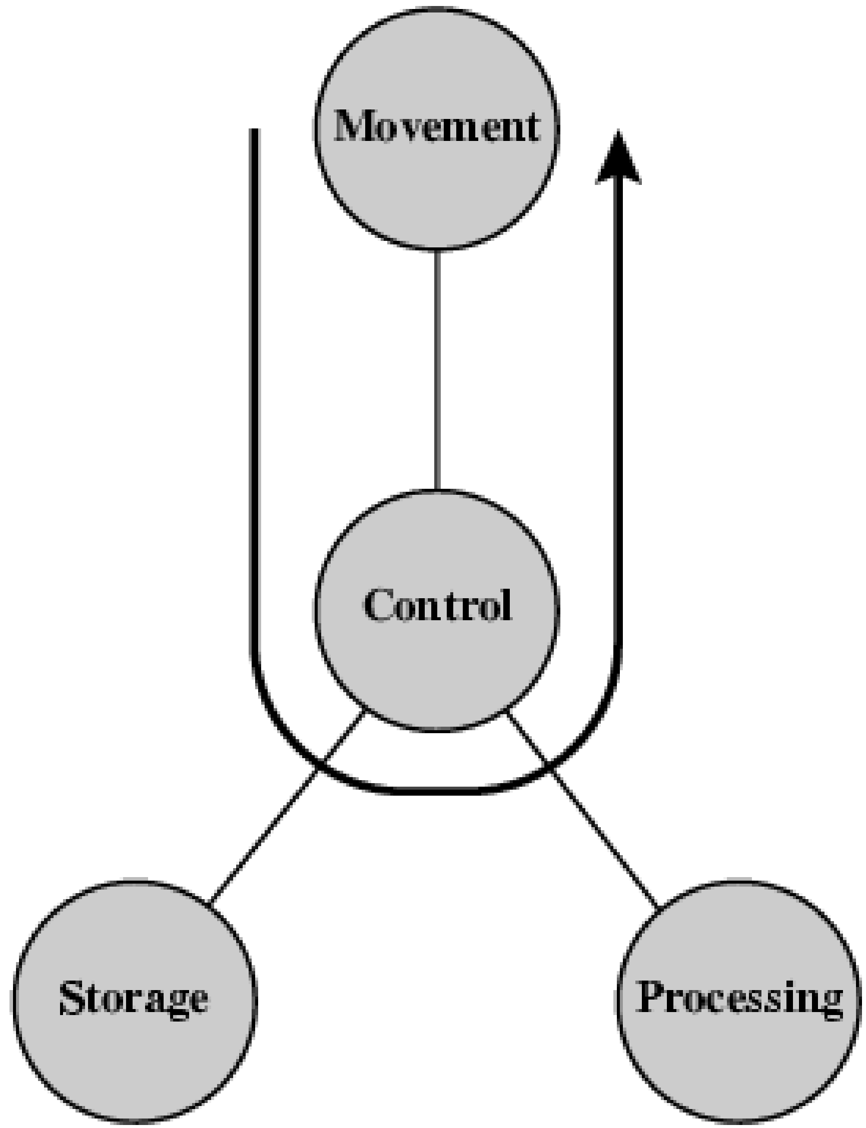
$$D_2 = A_1 \cdot \overline{A_0}$$

$$D_3 = A_1 \cdot A_0$$

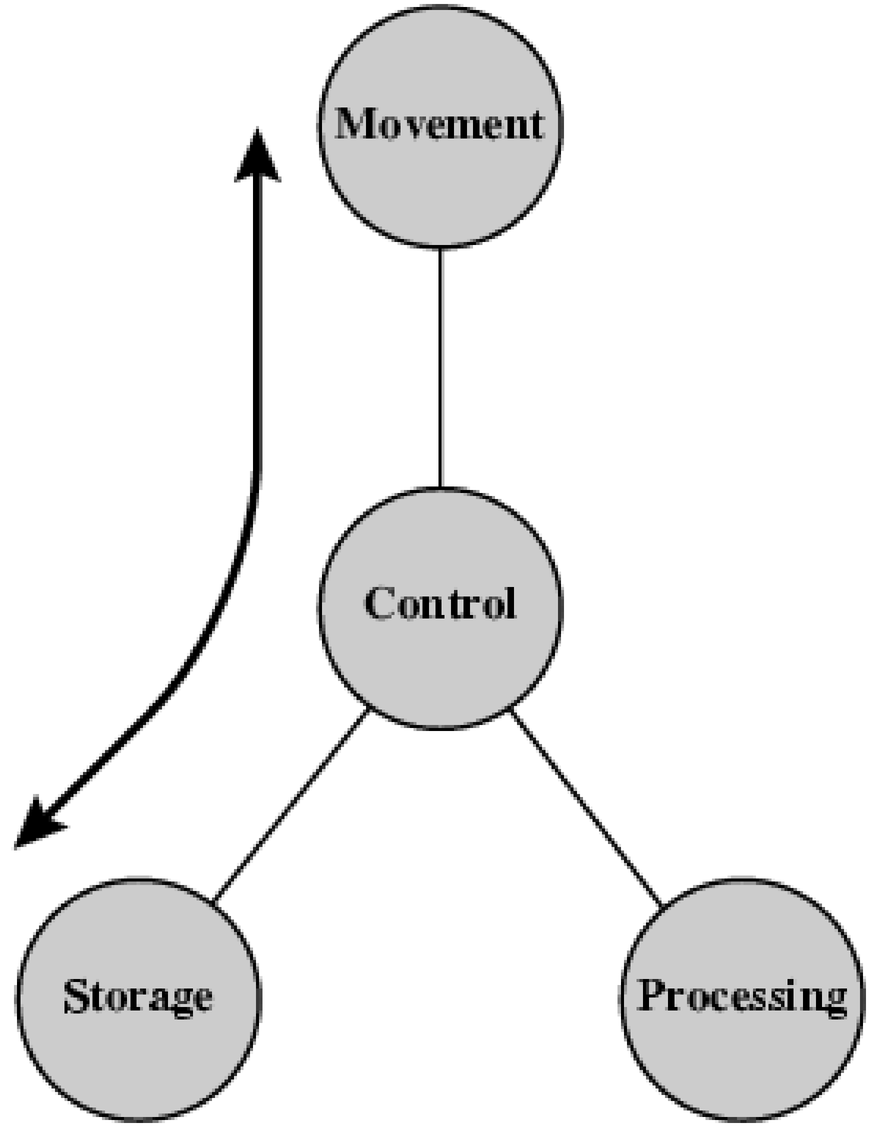
Functional View



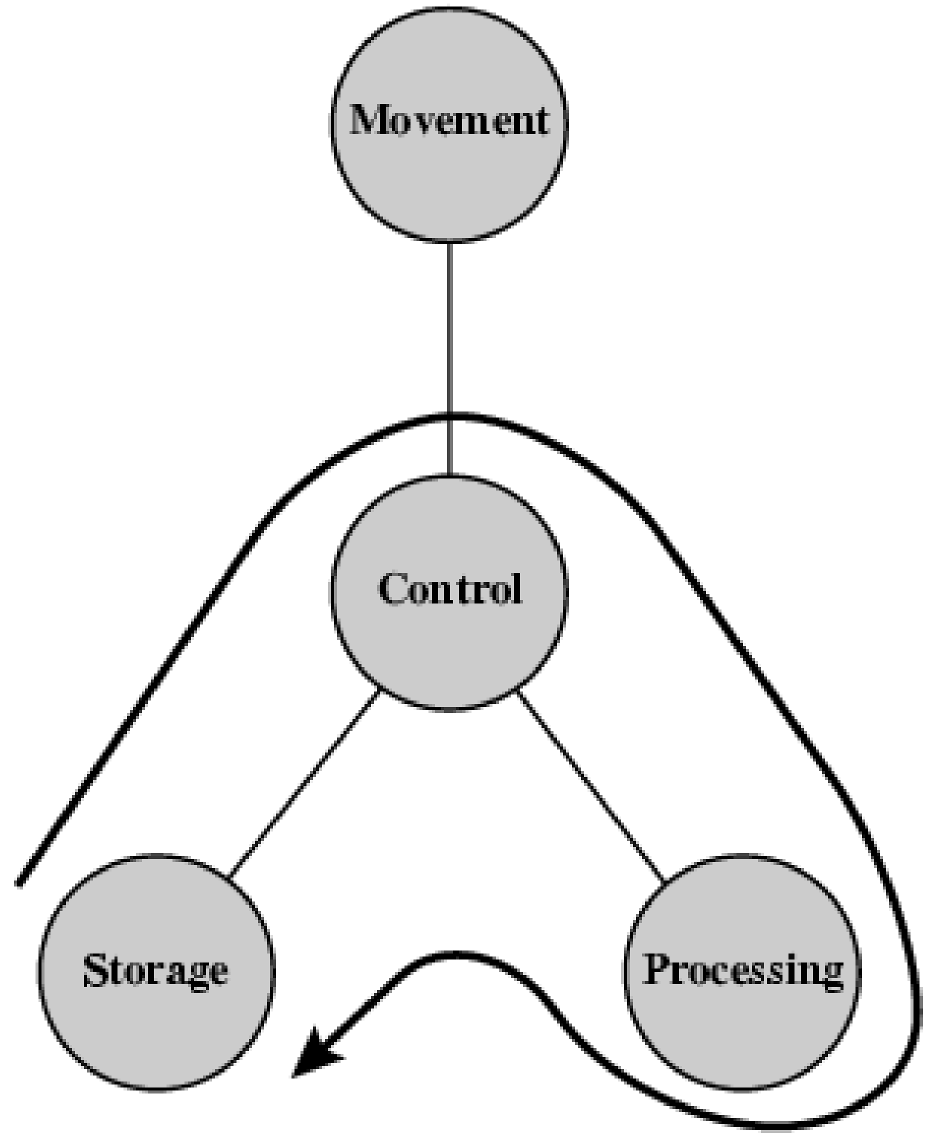
Data movement



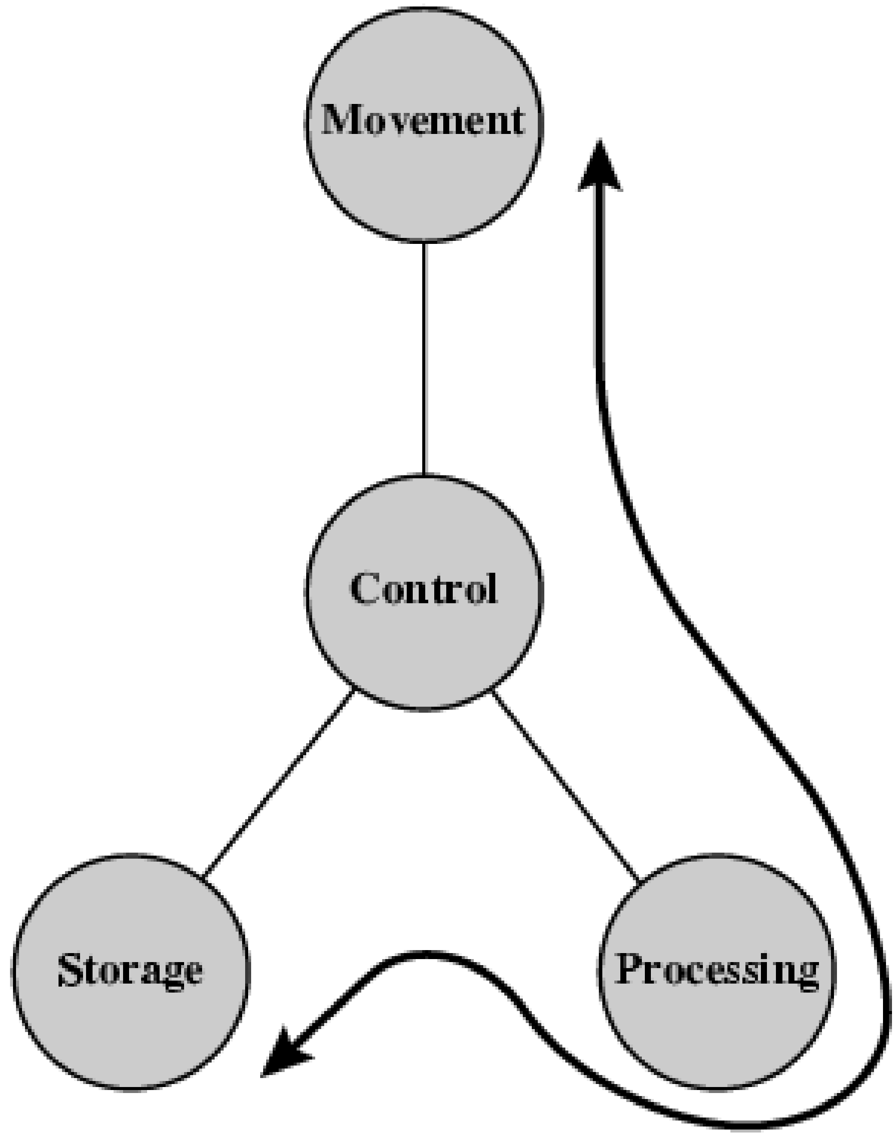
Storage



Processing (From or To Storage)



Processing
(From or
To I/O)



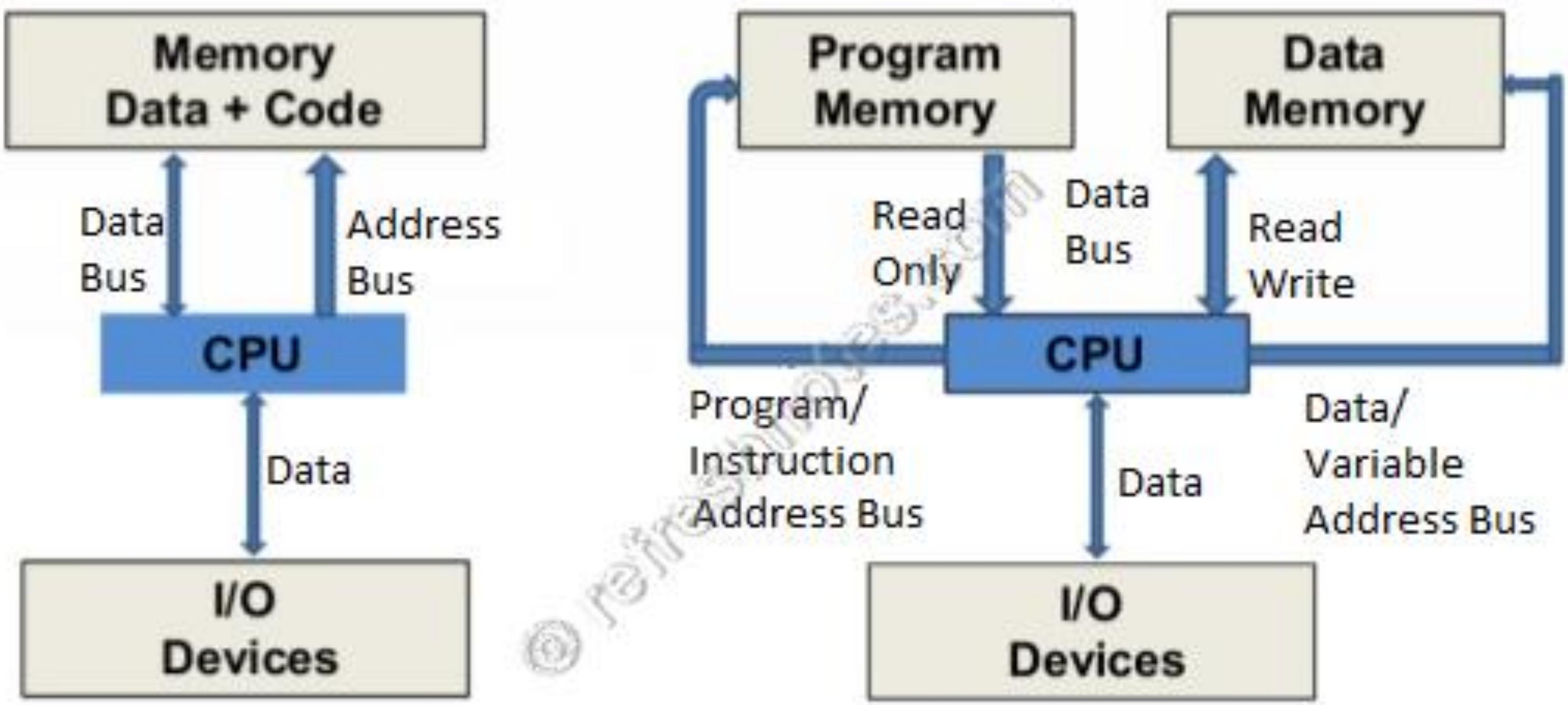
Why should we study CA?

- **Understand How Computers Work:** Learn how instructions are executed, how memory is accessed, and how hardware/software interact.
- **Optimize Software:** Writing efficient code requires knowing how CPUs, caches, and pipelines behave.
- **Design Better Hardware:** Knowledge of architecture is essential if you want to design processors, embedded systems, or robotics controllers.
- **Bridge Between Hardware & Software:** Architecture connects programming (software) with circuit design (hardware).
- **Evaluate & Choose Systems:** Helps compare CPUs (e.g., Intel vs ARM) for speed, power, and cost depending on the application.
- **Foundation for Advanced Fields:** Critical for Operating Systems, Compilers, AI accelerators, Robotics, High-performance Computing.

First Generation (1940s–1950s) – Vacuum Tubes

- Hardware: **Vacuum tubes** for circuits, **punch cards** for input/output.
- Memory: Magnetic drums.
- Programming: **Machine language (0s/1s)**, very hard to program.
- Example machines: **ENIAC (1945)**, **UNIVAC (1951)**.
- Architecture concept: **Von Neumann architecture** (stored-program computers) introduced.

Von Neumann vs Harvard



Von Neumann

Harvard

Von Neumann vs Harvard

Feature	Von Neumann Architecture	Harvard Architecture
Memory	Single memory for data & instructions	Separate memory for data & instructions
Bus System	One bus (shared)	Two buses (separate)
Instruction & Data	Fetched one at a time (sequential)	Can fetch instruction & data simultaneously
Speed	Slower due to bottleneck	Faster (no bottleneck)
Complexity	Simpler, cheaper	More complex, costly
Usage	General-purpose computers (PCs, servers)	Microcontrollers, DSPs, embedded systems

IAS Computer

- The **IAS machine** was the first electronic computer to be built at the Institute for Advanced Study (IAS) in Princeton, New Jersey. It is sometimes called the von Neumann machine, since the paper describing its design was edited by John von Neumann

IAS Computer

- The IAS machine was a binary computer with a 40-bit word, storing two 20-bit instructions in each word.
- The memory was 1,024 words (5.1 kilobytes). Negative numbers were represented in two's complement format.
- It was an asynchronous machine, meaning that there was no central clock regulating the timing of the instructions. One instruction started executing when the previous one finished.
- Von Neumann showed how the combination of instructions and data in one memory could be used to implement loops, by modifying branch instructions when a loop was completed, for example.
- The requirement that instructions, data and input/output be accessed via the same bus later came to be known as the Von Neumann bottleneck.

Early Computers

Feature / Machine	**Harvard Mark I (1944)**	**ENIAC (1945)**	**EDVAC (1949)**	**UNIVAC I (1951)**
Generation	1st (Electromechanical)	1st (Electronic, vacuum tubes)	1st (Electronic)	1st (Commercial Electronic)
Architecture	Harvard (separate program & data, punched tape)	Hard-wired, no stored program	Von Neumann (stored-program)	Von Neumann (stored-program)
Programming	Punched tape input (fixed)	Manual rewiring of plugboards	Instructions stored in memory	Instructions stored in memory
Flexibility	Very limited	Faster but hard to reprogram	Flexible (load different programs in memory)	Flexible & commercial use
Tech	Electromechanical relays	~18,000 vacuum tubes	Vacuum tubes	Vacuum tubes
Significance	Transition from calculators → early computers	First general-purpose electronic computer	First to implement **stored-program concept**	First **commercially available** stored-program computer (software era begins)

Harvard Architecture

- **Harvard Mark I (1944)** – electromechanical computer, designed at Harvard University by Howard Aiken.
- Programs were stored on punched tape (instructions).
- Data was stored in electromechanical counters (separate from instructions).

Von Neumann Architecture (Implemented Early)

1949

EDVAC (1949) → first true Von Neumann machine.

1952

IBM 701 (1952) → first IBM scientific computer.

UNIVAC I (1951) → first commercial computer.

1951

IBM

Was the major manufacturer of punched-card processing equipment

Delivered its first electronic stored-program computer (701) in 1953

- Intended primarily for scientific applications

Introduced 702 product in 1955

- Hardware features made it suitable to business applications

Series of 700/7000 computers established IBM as the overwhelmingly dominant computer manufacturer

Second Generation (1950s–1960s) – Transistors

- Hardware: **Transistors** replaced vacuum tubes → smaller, faster, more reliable.
- Memory: Magnetic cores.
- Programming: **Assembly language** introduced, plus early **high-level languages** (FORTRAN, COBOL).
- Example machines: **IBM 7090, PDP-1**.
- Architecture: More emphasis on **instruction sets** and control units.

Members of the IBM 700/7000 Series

Model Number	First Delivery	CPU Technology	Memory Technology	Cycle Time (μs)	Memory Size (K)	Number of Opcodes	Number of Index Registers	Hardwired Floating-Point	I/O Overlap (Channels)	Instruction Fetch Overlap	Speed (relative to 701)
701	1952	Vacuum tubes	Electrostatic tubes	30	2–4	24	0	no	no	no	1
704	1955	Vacuum tubes	Core	12	4–32	80	3	yes	no	no	2.5
709	1958	Vacuum tubes	Core	12	32	140	3	yes	yes	no	4
7090	1960	Transistor	Core	2.18	32	169	3	yes	yes	no	25
7094 I	1962	Transistor	Core	2	32	185	7	yes (double precision)	yes	yes	30
7094 II	1964	Transistor	Core	1.4	32	185	7	yes (double precision)	yes	yes	50

Important Terms

- **Cycle Time (µs)**: The time, in microseconds, it takes to complete one basic machine operation.
- **Memory Size (K)**: The total capacity of the main memory, in thousands of units.
- **Number of Opcodes**: The count of unique instructions the CPU can execute.
- **Number of Index Registers**: The number of specialized registers used for address modification.
- **Hardwired Floating-Point**: A dedicated hardware unit for performing floating-point arithmetic.
- **I/O Overlap (Channels)**: The ability to perform input/output operations simultaneously with CPU processing.
- **Instruction Fetch Overlap**: The capability to retrieve the next instruction while the current one is still executing.
- **Speed (relative to 701)**: A measure of the model's processing speed compared to the original 701 model.

IBM-7094

- Includes an Instruction Backup Register, used to buffer the next instruction.
- The control unit fetches two adjacent words from memory for an instruction fetch.
- A data channel is an independent I/O module with its own processor and instruction set.
  - CPU does not execute detailed I/O instructions.
  - The CPU initiates an I/O transfer by sending a control signal to the data channel, instructing it to execute a sequence of instructions in memory.
- The multiplexor schedules access to the memory from the CPU and data channels, allowing these devices to act independently.

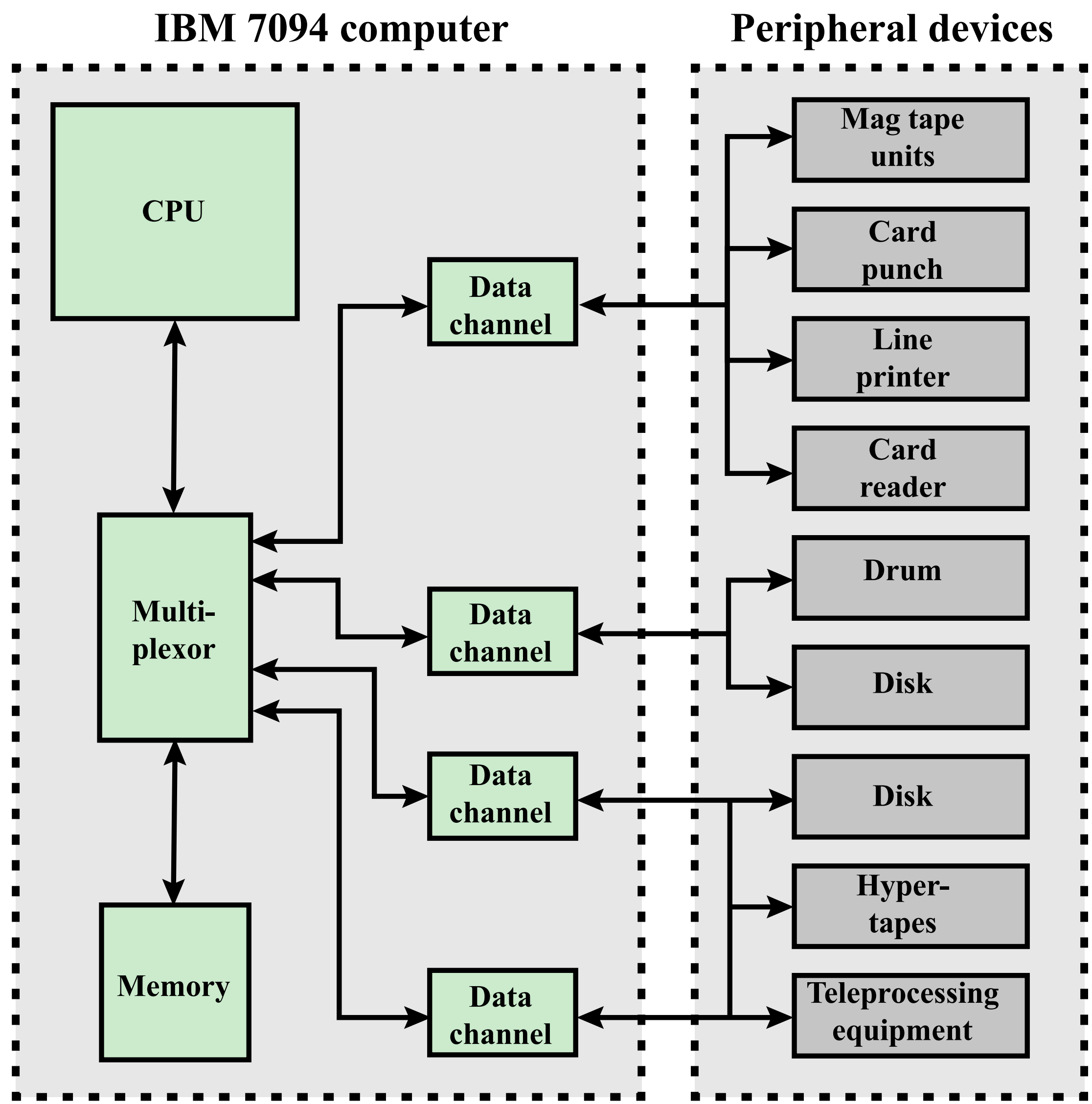
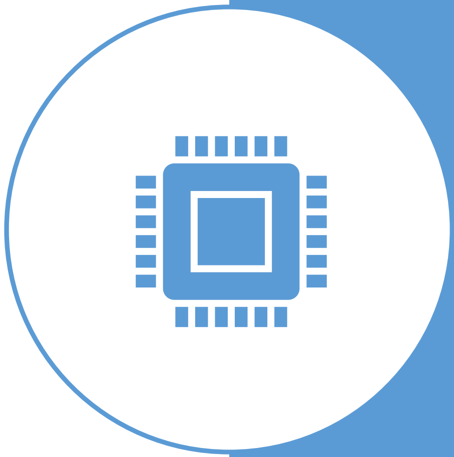


Figure 1.9 An IBM 7094 Configuration

Third Generation (1960s–1970s) – Integrated Circuits (ICs)

- •Hardware: **Integrated Circuits (ICs)** replaced individual transistors.
- •OS: Multiprogramming & time-sharing introduced.
- •Programming: More high-level languages (C, BASIC).
- •Example machines: **IBM System/360**, **PDP-8**, **DEC VAX**.
- •Architecture: **CISC (Complex Instruction Set Computers)** become standard.



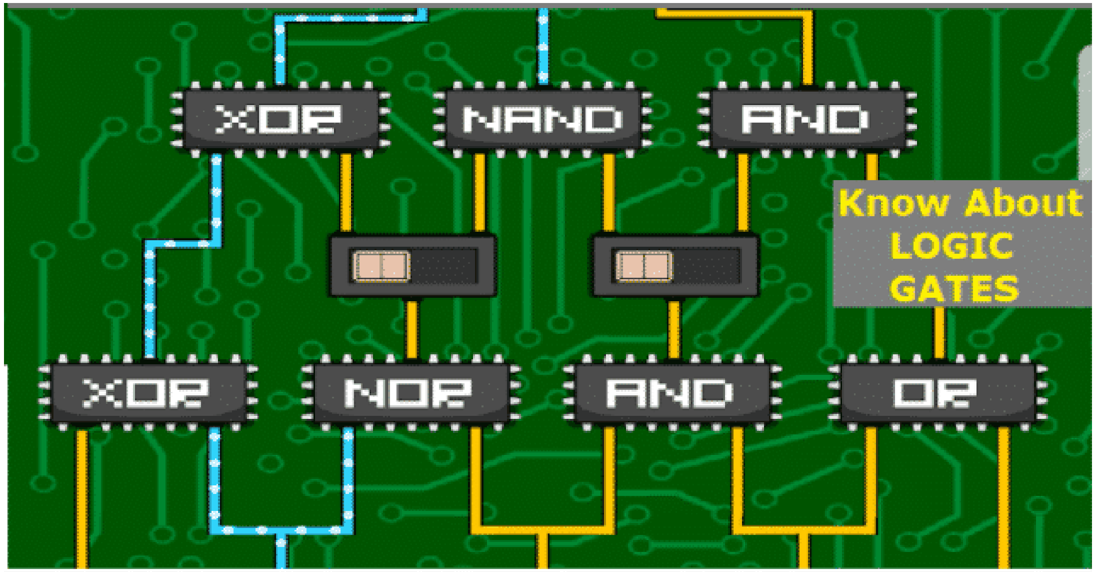
Integrated Circuits

- Data storage – provided by memory cells
- Data processing – provided by gates
- Data movement – the paths among components are used to move data from memory to memory and from memory through gates to memory
- Control – the paths among components can carry control signals
- The two most important members of the third generation were the IBM System/360 and the DEC PDP-8

Integrated Circuits

- A computer consists of gates, memory cells, and interconnections among these elements

- The gates and memory cells are constructed of simple digital electronic components



Microelectronics

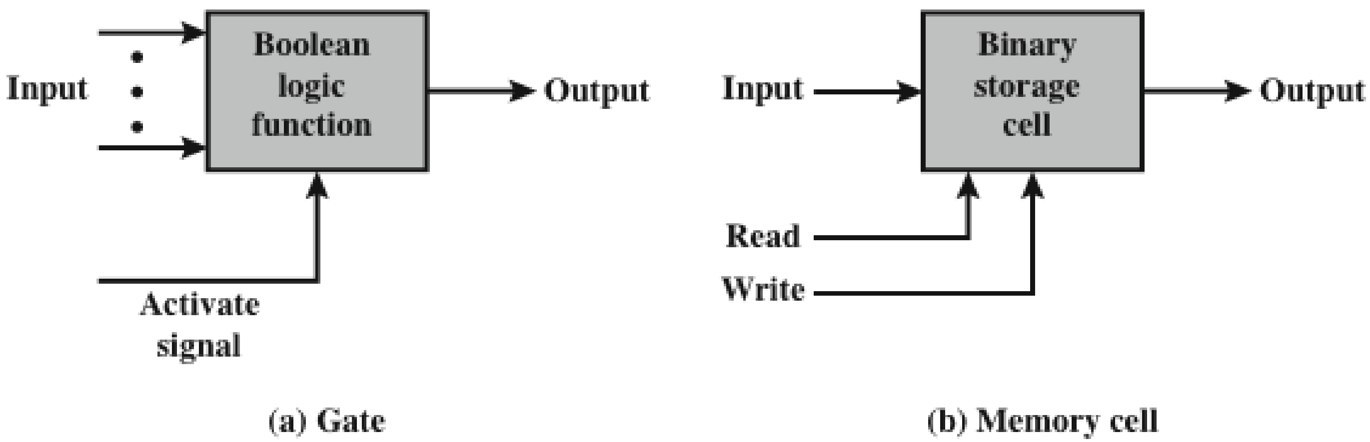


Figure 2.6 Fundamental Computer Elements

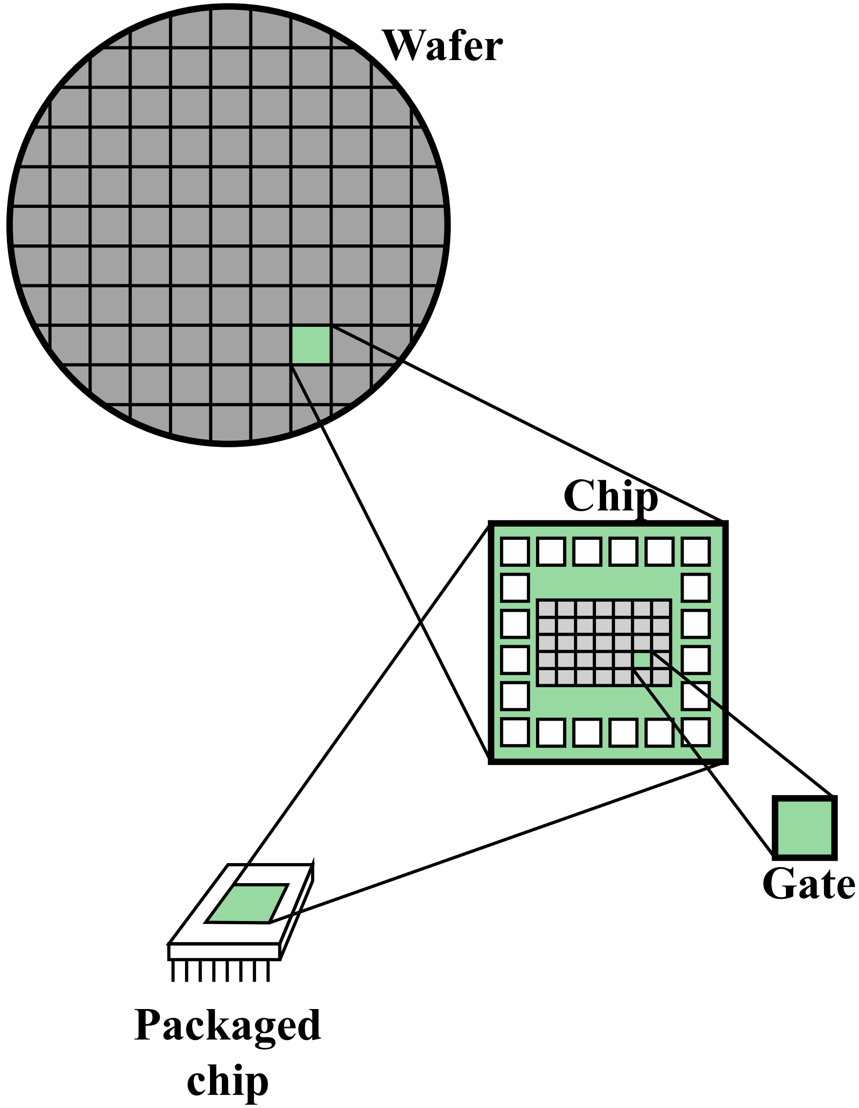


Figure 1.11 Relationship Among Wafer, Chip, and Gate

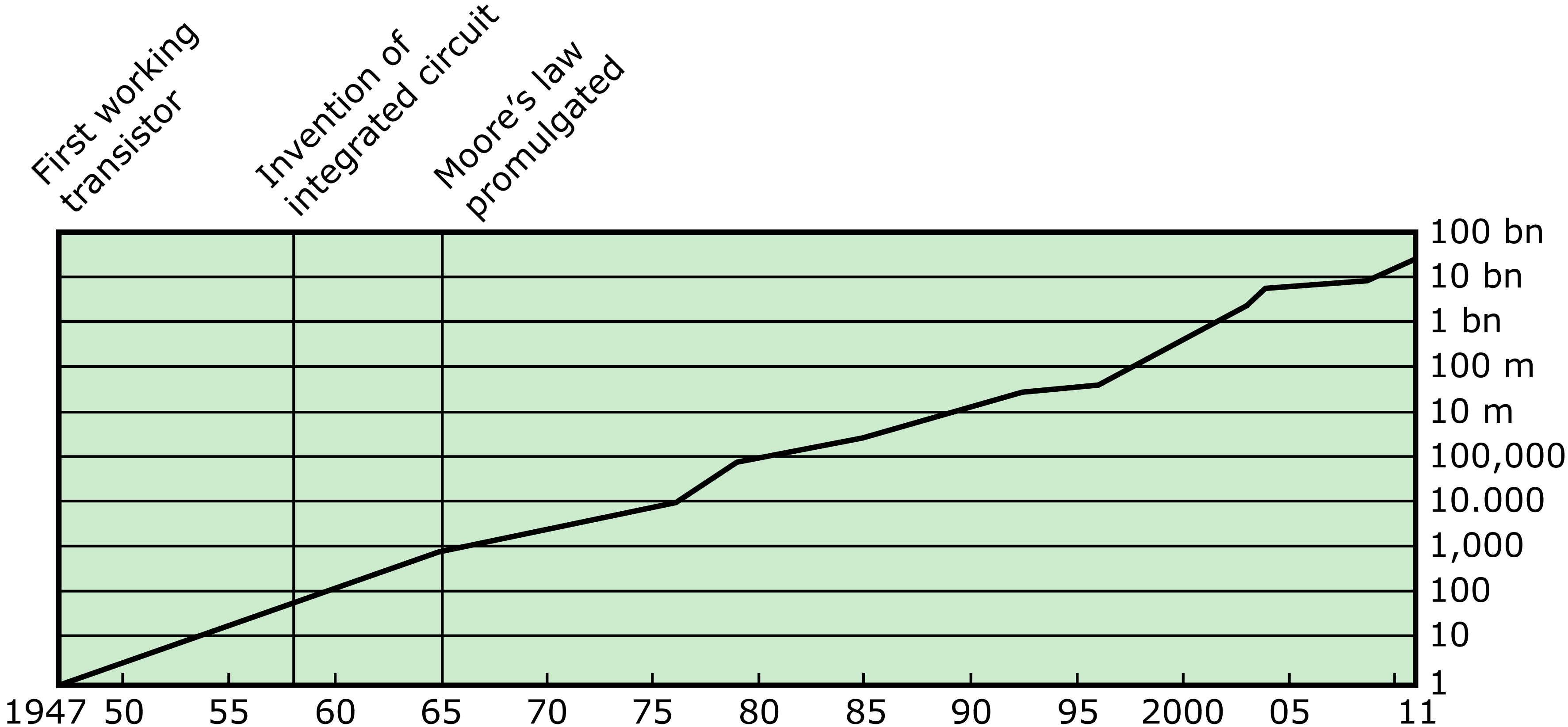


Figure 1.12 Growth in Transistor Count on Integrated Circuits (DRAM memory)

Moore's Law

Proposed by: Gordon Moore (Intel, 1965)

Statement: *Transistor count on a chip doubles every ~18–24 months, while cost per transistor halves.*

Faster & cheaper computers

Smaller devices (miniaturization)

Growth of PCs, smartphones, AI

1971: Intel 4004 → 2,300 transistors

2023: Apple M2 → ~20 billion transistors

IBM System/360

- Announced in 1964
- Product line was incompatible with older IBM machines
- Was the success of the decade and cemented IBM as the overwhelmingly dominant computer vendor
- The architecture remains to this day the architecture of IBM's mainframe computers
- Was the industry's first planned family of computers
  - Models were compatible in the sense that a program written for one model should be capable of being executed by another model in the series

Characteristics of the System/360 Family

Characteristic	Model 30	Model 40	Model 50	Model 65	Model 75
Maximum memory size (bytes)	64K	256K	256K	512K	512K
Data rate from memory (Mbytes/sec)	0.5	0.8	2.0	8.0	16.0
Processor cycle time μs)	1.0	0.625	0.5	0.25	0.2
Relative speed	1	3.5	10	21	50
Maximum number of data channels	3	3	4	6	6
Maximum data rate on one channel (Kbytes/s)	250	400	800	1250	1250

DEC-PDP-8

- Same year(IBM system/360) –PDP-8 from Digital Equipment Corporation
- Small enough to be placed on top of bench
- No air-conditioned room required
- Cheap enough (Not exact functionality like mainframe)
- Introduced the concept of Omnibus(Bus structure for microcomputers)

DEC-PDP-8

- Consist of 96 separate signal paths
- Used to carry data, address and control signals
- All system components share a common set of signal paths
- Their use controlled by CPU
- Flexible architecture→ modules to be plugged into the bus

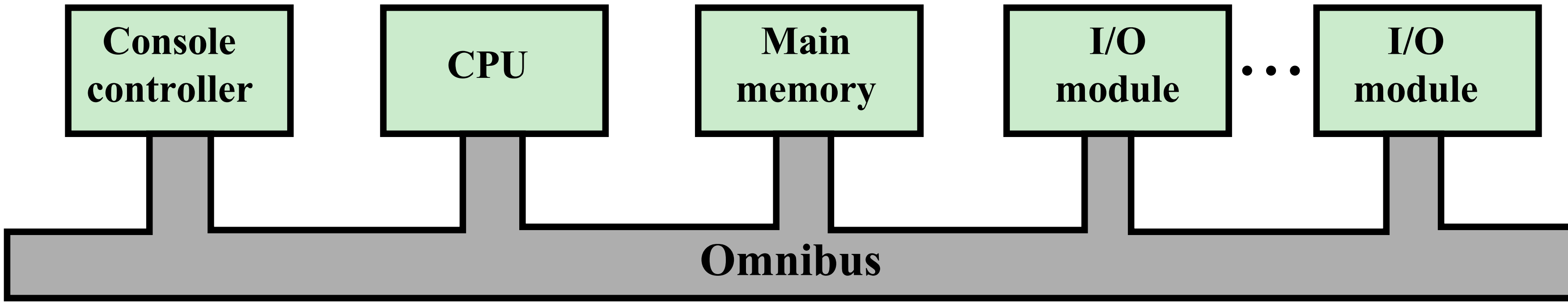


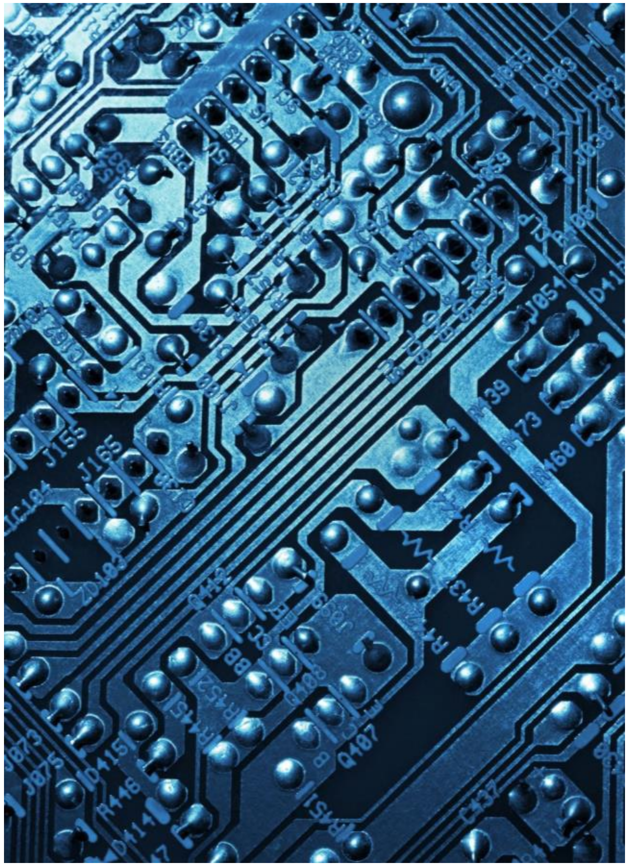
Figure 1.13 PDP-8 Bus Structure

Computer Generations

Generation	Approximate Dates	Technology	Typical Speed (operations per second)
1	1946–1957	Vacuum tube	40,000
2	1957–1964	Transistor	200,000
3	1965–1971	Small and medium scale integration	1,000,000
4	1972–1977	Large scale integration	10,000,000
5	1978–1991	Very large scale integration	100,000,000
6	1991-	Ultra large scale integration	>1,000,000,000

Fourth Generation (1970s–1990s) – Microprocessors

- Hardware: **Microprocessors** (entire CPU on a single chip).
- PCs introduced: Apple II, IBM PC (1981).
- Memory: Semiconductor memory (RAM, ROM).
- Examples: **Intel 8086 (1978)** → basis of today's x86.
- Architecture:
    - **CISC (x86)** dominant in PCs.
    - **RISC (Reduced Instruction Set Computer)** ideas emerge in 1980s (MIPS, SPARC, ARM).



Microprocessors

- The density of elements on processor chips continued to rise
  - More and more elements were placed on each chip so that fewer and fewer chips were needed to construct a single computer processor
- 1971 Intel developed 4004
  - First chip to contain all of the components of a CPU on a single chip
  - Birth of microprocessor
- 1972 Intel developed 8008
  - First 8-bit microprocessor
- 1974 Intel developed 8080
  - First general purpose microprocessor
  - Faster, has a richer instruction set, has a large addressing capability

Evolution of Intel Microprocessors

	4004	8008	8080	8086	8088
Introduced	1971	1972	1974	1978	1979
Clock speeds	108 kHz	108 kHz	2 MHz	5 MHz, 8 MHz, 10 MHz	5 MHz, 8 MHz
Bus width	4 bits	8 bits	8 bits	16 bits	8 bits
Number of transistors	2,300	3,500	6,000	29,000	29,000
Feature size (μm)	10		6	3	6
Addressable memory	640 Bytes	16 KB	64 KB	1 MB	1 MB

a. 1970s Processors

	80286	386TM DX	386TM SX	486TM DX CPU
Introduced	1982	1985	1988	1989
Clock speeds	6 MHz - 12.5 MHz	16 MHz - 33 MHz	16 MHz - 33 MHz	25 MHz - 50 MHz
Bus width	16 bits	32 bits	16 bits	32 bits
Number of transistors	134,000	275,000	275,000	1.2 million
Feature size (μm)	1.5	1	1	0.8 - 1
Addressable memory	16 MB	4 GB	16 MB	4 GB
Virtual memory	1 GB	64 TB	64 TB	64 TB
Cache	—	—	—	8 kB

b. 1980s Processors

Evolution of Intel Microprocessors

	486TM SX	Pentium	Pentium Pro	Pentium II
Introduced	1991	1993	1995	1997
Clock speeds	16 MHz - 33 MHz	60 MHz - 166 MHz,	150 MHz - 200 MHz	200 MHz - 300 MHz
Bus width	32 bits	32 bits	64 bits	64 bits
Number of transistors	1.185 million	3.1 million	5.5 million	7.5 million
Feature size (μm)	1	0.8	0.6	0.35
Addressable memory	4 GB	4 GB	64 GB	64 GB
Virtual memory	64 TB	64 TB	64 TB	64 TB
Cache	8 kB	8 kB	512 kB L1 and 1 MB L2	512 kB L2

c. 1990s Processors

	Pentium III	Pentium 4	Core 2 Duo	Core i7 EE 990
Introduced	1999	2000	2006	2011
Clock speeds	450 - 660 MHz	1.3 - 1.8 GHz	1.06 - 1.2 GHz	3.5 GHz
Bus width	64 bits	64 bits	64 bits	64 bits
Number of transistors	9.5 million	42 million	167 million	1170 million
Feature size (nm)	250	180	65	32
Addressable memory	64 GB	64 GB	64 GB	64 GB
Virtual memory	64 TB	64 TB	64 TB	64 TB
Cache	512 kB L2	256 kB L2	2 MB L2	1.5 MB L2/12 MB L3

d. Recent Processors

Fifth Generation (1990s–2000s) – Parallelism & Networking

Hardware: **Very Large Scale Integration (VLSI)** → millions of transistors.

Multi-core processors emerge.

Networking + Internet shape architecture demands.

Examples: Intel Pentium series, AMD Athlon, early ARM processors in mobile devices.

Architecture: Superscalar, pipelining, branch prediction, caches for speed.

Sixth Generation (2010s–Now) – Heterogeneous & AI-focused

•Hardware: **Multi-core, many-core, GPUs, TPUs**.

•ARM becomes dominant in **mobile devices** (low power).

•Apple shifts Macs to **ARM (Apple Silicon M1, M2, M3)**.

•Cloud computing and data centers demand specialized processors.

•Architecture: **RISC-V** (open-source ISA) gaining popularity.

•Specialized chips: AI accelerators, neural processing units (NPUs).