

Contents

CHAPTER 1: BASIC CONCEPTS AND COMPUTER EVOLUTION	1
1.1 ORGANIZATION AND ARCHITECTURE	1
1.2 STRUCTURE AND FUNCTION	2
1.3 A BRIEF HISTORY OF COMPUTERS.....	8
First Generation: Vacuum Tubes (Bóng đèn chân không).....	8
Second Generation: Transistors (Linh kiện bán dẫn)	16
Third Generation: Integrated Circuits (IC) (Mạch tích hợp/ Vi mạch).....	17
Later Generations.....	19
1.4 EMBEDDED SYSTEMS.....	22
The Internet of Things (IoT)	22
Embedded Operating Systems.....	23
Application Processors versus Dedicated Processors.....	23
Microprocessors versus Microcontrollers	23
Deeply Embedded Systems	23
1.5 CLOUD COMPUTING.....	24
Cloud Networking	24
Cloud Storage.....	24
Cloud Services	24
PHỤ LỤC.....	26

1.1 ORGANIZATION AND ARCHITECTURE

- Phân biệt kiến trúc máy tính & tổ chức máy tính:

* **Kiến trúc máy tính (Computer Architecture):**

+ **Architectural** attributes include: **Instruction set** (tập lệnh) (điều khiển bộ xử lý, thực hiện các phép toán, thao tác trên đó), **number of bits** used to represent various *data types*, **I/O mechanisms**, techniques for **addressing memory** (Đánh địa chỉ bộ nhớ).

Kiến trúc bên trong = cốt lõi tạo nên máy tính: Tập lệnh → Mặt nào thuộc bộ xử lý sẽ hoạt động → Sự hữ này cần bao nhiêu bit → Xử lý xong thì đưa dữ liệu ra ngoài để giao tiếp (I/O, dùng ngoại vi nào) → DL có bao nhiêu bit? → Dùng KT đánh đ/chỉ → DL được truy xuất tại địa chỉ nào (từ...đến...) (lấy bao nhiêu bit).

+ Các thuộc tính của hệ thống mà người lập trình **có thể nhìn thấy**

+ Có **tác động trực tiếp đến việc thực hiện logic** của một chương trình

+ Một thuật ngữ thường được sử dụng thay thế cho kiến trúc máy tính là **kiến trúc tập lệnh (instruction set architecture) (ISA)**. ISA xác định các định dạng lệnh (instruction formats), mã lệnh (instruction opcodes), thanh ghi (registers), lệnh (instruction) và bộ nhớ dữ liệu; sự ảnh hưởng của các lệnh được thực hiện trên các thanh ghi và bộ nhớ; và một thuật toán để kiểm soát việc thực hiện lệnh.

+ Định nghĩa của Hennessy/ Patterson

Kiến trúc máy tính bao gồm:

■ **Kiến trúc tập lệnh (Instruction Set Architecture):** nghiên cứu máy tính theo cách nhìn của người lập trình (hardware/software interface).

■ **Tổ chức máy tính (Computer Organization) hay Vi kiến trúc (Microarchitecture):** nghiên cứu thiết kế máy tính ở mức cao, chẳng hạn như hệ thống nhớ, cấu trúc bus, thiết kế bên trong CPU.

■ **Phần cứng (Hardware):** nghiên cứu thiết kế logic chi tiết và công nghệ đóng gói của máy tính.

Kiến trúc tập lệnh thay đổi chậm, tổ chức và phần cứng máy tính thay đổi rất nhanh.

+ Kiến trúc tập lệnh của máy tính bao gồm:

■ **Tập lệnh:** tập hợp các chuỗi số nhị phân mã hoá cho các thao tác mà máy tính có thể thực hiện

■ **Các kiểu dữ liệu:** các kiểu dữ liệu mà máy tính có thể xử lý

* **Tổ chức máy tính (Computer Organization):**

+ Organizational attributes include: Hardware details **transparent** (trong suốt ~ không nhìn thấy) to the programmer, **control signals** (tín hiệu điều khiển ~ protocol = giao thức giao tiếp với hardware), **interfaces** between the computer and peripherals (/pəˈrɪf.ər.əl/) (thiết bị ngoại vi ~ cổng USB, LPT..., giao tiếp với 1 MT khác/ thiết bị khác), **memory technology** used.

+ Các đơn vị vận hành (operational units) và các kết nối của chúng (interconnections) nhằm thực hiện các đặc tả kiến trúc.

- Sự khác biệt giữa kiến trúc và tổ chức vẫn là một điều quan trọng. Nhiều nhà sản xuất máy tính cung cấp một dòng máy tính, tất cả đều có **cùng kiến trúc** nhưng có **sự khác biệt về cách tổ chức**. Do đó, các mẫu khác nhau trong dòng sản phẩm này có đặc điểm giá cả và hiệu suất khác nhau. Hơn nữa, **một kiến trúc cụ thể có thể kéo dài nhiều năm** và bao gồm một số mô hình máy tính khác nhau, **tổ chức của nó thay đổi theo sự thay đổi của công nghệ**.

- Kiến trúc có thể bao gồm một số mẫu (models). Khách hàng có yêu cầu khiêm tốn có thể mua mẫu rẻ hơn, chậm hơn và nếu nhu cầu tăng, sau đó sẽ nâng cấp lên mẫu đắt hơn, nhanh hơn mà **không cần phải từ bỏ phần mềm đã được phát triển**.

- Tổ chức máy tính phải được thiết kế để **thực hiện một đặc tả kiến trúc cụ thể** nên việc xử lý tổ chức một cách kỹ lưỡng cũng đòi hỏi phải kiểm tra chi tiết kiến trúc.

1.2 STRUCTURE AND FUNCTION

- Một hệ thống phân cấp (Hierarchical system /ˌhaɪəˈrɑː.kɪ.kəl/) là một **tập hợp các hệ thống con có liên quan với nhau**, mỗi hệ thống con lần lượt có cấu trúc phân cấp cho đến khi chúng ta đạt đến cấp độ thấp nhất của hệ thống con cơ bản.

- Ở mỗi cấp độ, người thiết kế quan tâm đến cấu trúc (structure) và chức năng (function):

+ Structure (Cấu trúc): Cách thức các thành phần liên kết với nhau.

+ Function (chức năng): **Thực hiện chương trình**; Hoạt động của từng bộ phận riêng lẻ như một phần của cấu trúc.

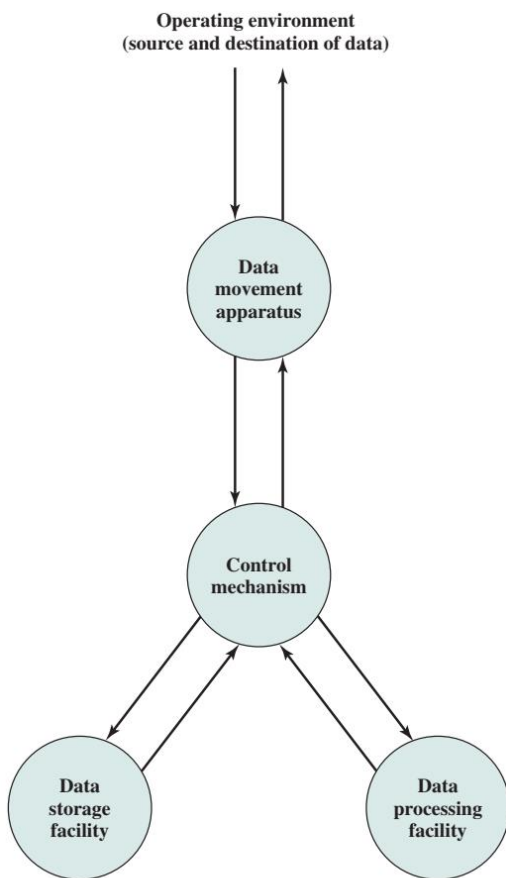


Figure 1.1 A Functional View of the Computer

- There are **four** basic **functions** that a computer can perform:

+ Data processing:

Data may take a wide variety of forms and the range of processing requirements (phạm vi yêu cầu xử lý) is broad.

There are only a few fundamental methods (1 vài pp cơ bản) or types of data processing.

+ Data storage: Short-term, Long-term

Ngay cả khi máy tính đang xử lý dữ liệu một cách nhanh chóng (**is processing data on the fly**) (tức là dữ liệu đến và được xử lý và kết quả sẽ xuất hiện ngay lập tức), máy tính phải **lưu trữ tạm thời** ít nhất những phần **dữ liệu đang được xử lý** tại bất kỳ thời điểm nào. Như vậy, ít nhất còn có chức năng lưu trữ **dữ liệu ngắn hạn (short-term)**.

Điều quan trọng không kém là máy tính thực hiện chức năng lưu trữ **dữ liệu lâu dài (long-term)**. Các tập tin dữ liệu được lưu trữ trên máy tính để **phục hồi và cập nhật sau này**.

+ 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**

+ Control: there must be control of these three functions. A **control unit manages** the computer's **resources** and **orchestrates the performance** (điều phối hoạt động) of its functional parts in response to those instructions.

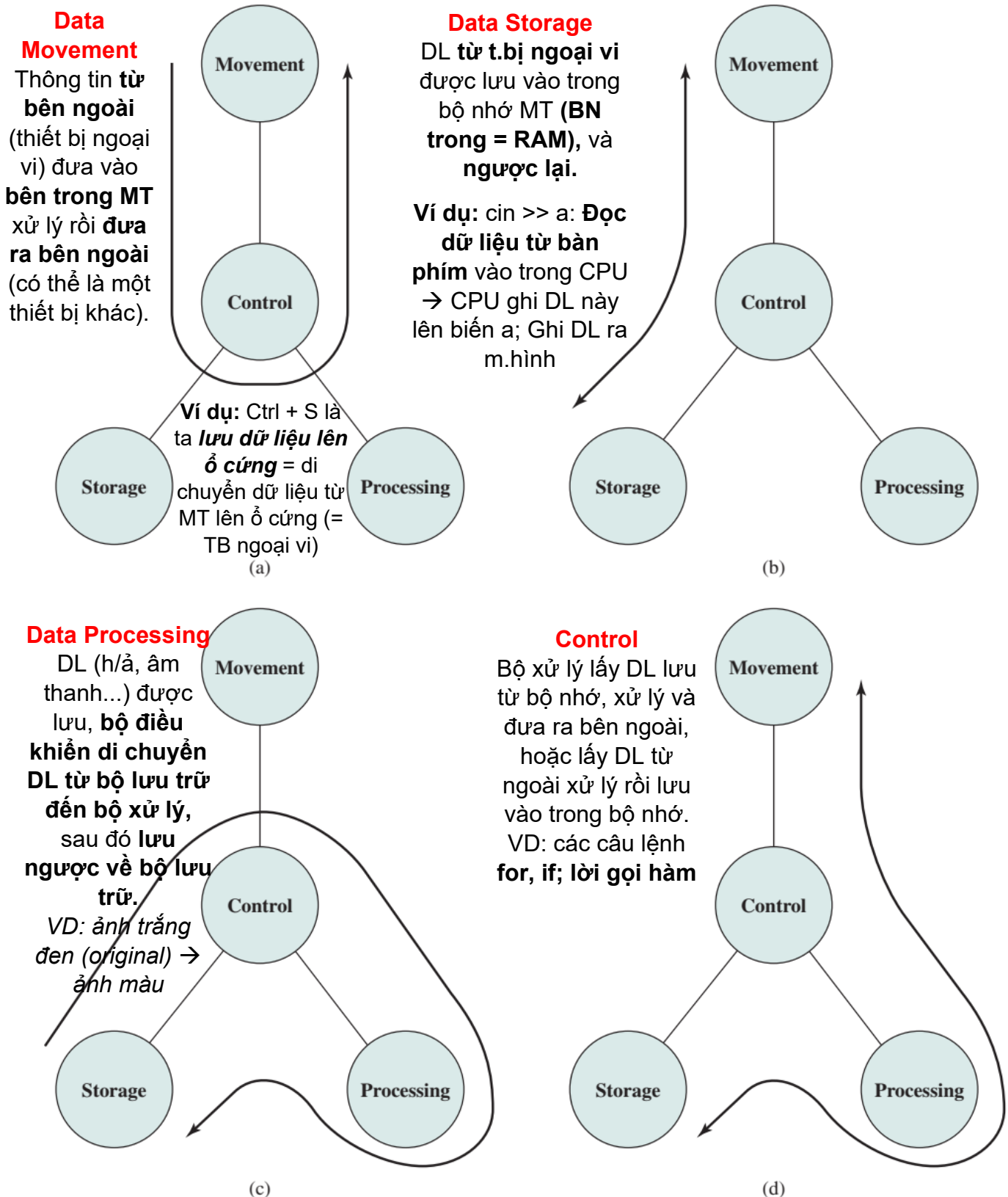


Figure 1.2 Possible Computer Operations

SIMPLE SINGLE-PROCESSOR COMPUTER

Figure 1.1 (below) cung cấp cái nhìn **phân cấp** về cấu trúc bên trong của máy tính **bộ xử lý đơn truyền thống**.

- There are **four** main **structural components** of the computer: (Các thành phần cơ bản của máy tính)

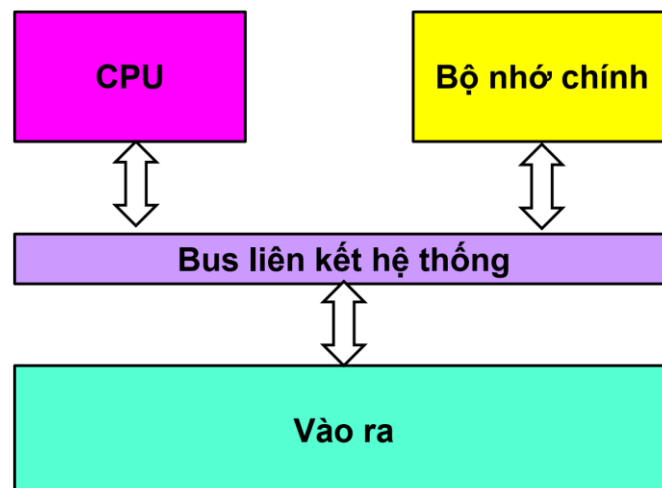
+ **CPU** – controls the **operation of the computer** and performs its **data processing functions**; thường được gọi đơn giản là bộ xử lý (**processor**); *Thực thi chương trình (lệnh), không có khả năng lưu trữ dữ liệu & lệnh*

- + **Main Memory** – stores data (*ROM/RAM: lưu trữ CT trong quá trình CPU thực thi*)
- + **I/O** – moves data between the computer (CPU) and its external environment (*T.bị n.vi*).

Các loại thiết bị vào-ra:

- *Thiết bị vào (Input Devices)*
- *Thiết bị ra (Output Devices)*
- *Thiết bị lưu trữ (Storage Devices)*
 - Còn được gọi là bộ nhớ ngoài
 - Chức năng và đặc điểm
 - Lưu giữ tài nguyên phần mềm của máy tính
 - Được kết nối với hệ thống dưới dạng các thiết bị vào-ra
 - Dung lượng lớn
 - Tốc độ chậm
 - Các loại thiết bị lưu trữ
 - Bộ nhớ từ: ổ đĩa cứng HDD
 - Bộ nhớ bán dẫn: ổ thể rắn SSD, ổ nhớ flash, thẻ nhớ
 - Bộ nhớ quang: CD, DVD
- *Thiết bị truyền thông (Communication Devices)*

+ **System Interconnection** – some mechanism that provides for **communication among CPU, main memory, and I/O** (ví dụ phổ biến: **system bus**, bao gồm một số dây dẫn mà tất cả các thành phần khác gắn vào.)



- The most interesting and in some ways the most complex component is the **CPU**. Its major structural components are as follows:

- + **Control Unit (CU – khối điều khiển)**: Controls the *operation of the CPU and the computer*; *Giải mã & thực thi lệnh*.
- + **Arithmetic and Logic Unit (ALU)**: Performs the computer's **data processing function**; *Tính toán số học & logic*
- + **Registers (Thanh ghi)**: Provide *storage internal* to the CPU; chứa các thông tin **tạm thời** trong quá trình CU, ALU thực hiện công việc/ xử lý.
- + **CPU Interconnection**: Some mechanism that provides for communication among the control unit, ALU, and registers (Internal Bus).

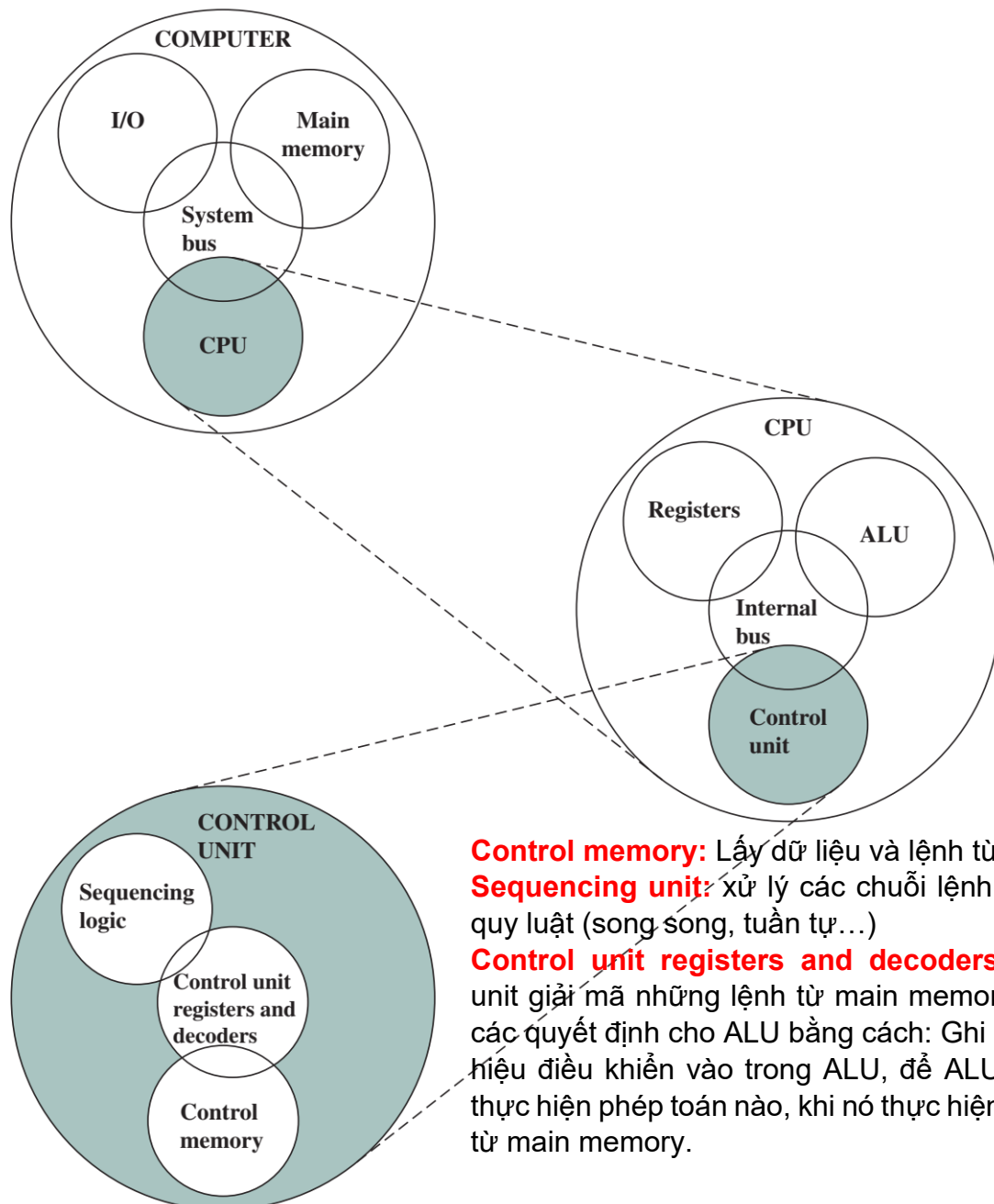


Figure 1.1 The Computer: Top-Level Structure

MULTICORE COMPUTER STRUCTURE

- Máy tính hiện đại thường có nhiều bộ xử lý. Khi tất cả các bộ xử lý này đều nằm trên một chip đơn, thuật ngữ máy tính đa lõi sẽ được sử dụng và **mỗi bộ xử lý** (bao gồm bộ điều khiển, ALU, các thanh ghi và có lẽ cả bộ đệm) **được gọi là lõi (Core)**.

- Central processing unit (Bộ xử lý trung tâm) (CPU)

- + Phần của máy tính (Portion of the computer) tìm nạp (fetch) và thực thi các lệnh
- + Bao gồm ALU, bộ điều khiển và các thanh ghi
- + Được coi là bộ xử lý trong hệ thống có **một đơn vị xử lý (single processing unit)**.

- Core (Lõi)

- + Một bộ xử lý riêng lẻ trên 1 chip xử lý
- + Có thể có *chức năng tương đương với CPU trên hệ thống một CPU*
- + Các đơn vị xử lý chuyên biệt (Specialized processing units) còn được gọi là các lõi

- Processor (Bộ xử lý)

- + Một mảnh silicon vật lý chứa *một hoặc nhiều lõi*
- + Là thành phần máy tính dịch (interprets) và thực thi các lệnh
- + Được gọi là **bộ xử lý đa lõi (multicore processor)** nếu nó chứa nhiều lõi.

- Một đặc điểm nổi bật khác của máy tính hiện đại là việc sử dụng nhiều lớp (layers) bộ nhớ, gọi là bộ nhớ đệm (**cache memory**), *giữa bộ xử lý và bộ nhớ chính*. (Chap 4).
- Bộ nhớ đệm có dung lượng nhỏ hơn & nhanh hơn bộ nhớ chính, **được đặt đệm giữa CPU và bộ nhớ chính** nhằm tăng tốc độ CPU & truy cập bộ nhớ.
- Có thể đạt được sự *cải thiện hiệu suất lớn hơn bằng cách sử dụng nhiều cấp độ bộ đệm*, với cấp 1 (L1) gần lõi nhất và các cấp bổ sung (L2, L3, v.v.) dần dần xa lõi hơn; *cấp n nhỏ hơn và nhanh hơn cấp $n + 1$* .
- Cache thường được tích hợp trên cùng chip bộ xử lý
- Cache có thể có hoặc không

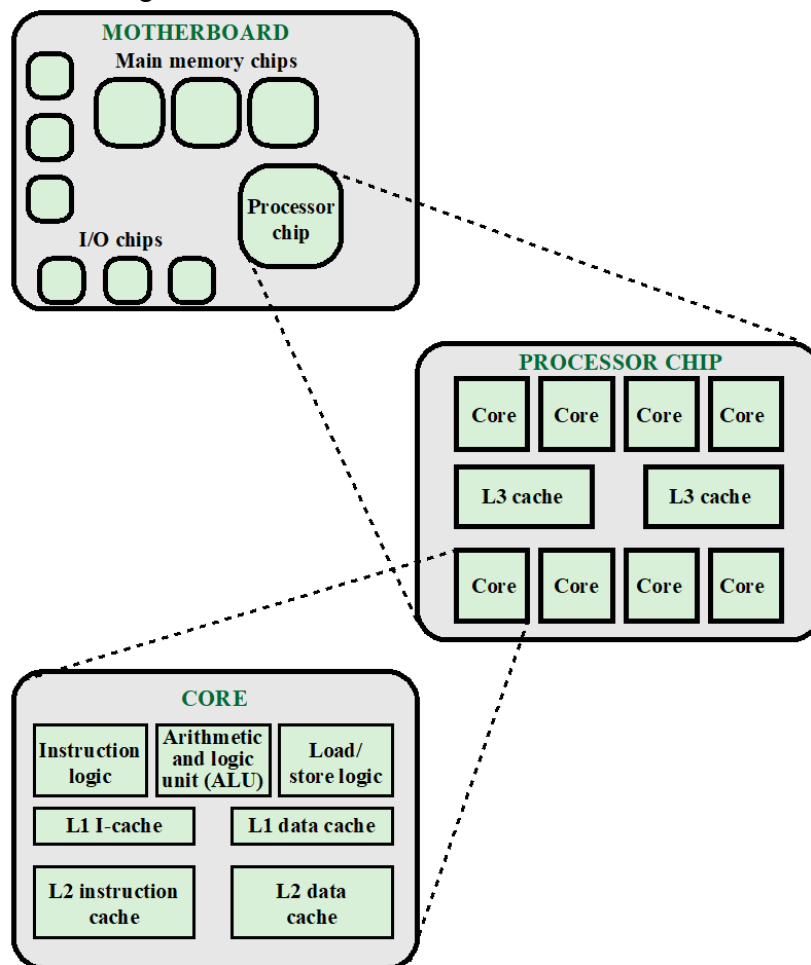


Figure 1.2 Simplified View of Major Elements of a Multicore Computer

- Most computers, including embedded computers in smartphones and tablets, plus personal computers, laptops, and workstations, are housed on (được đặt trên) a **motherboard**.
- Bảng mạch in (**printed circuit board**) (PCB) là một bảng phẳng, cứng **để giữ và kết nối các chip và các linh kiện điện tử khác**. Bảng mạch được làm từ các lớp, thường từ hai đến mười lớp, kết nối các thành phần thông qua các đường dẫn bằng đồng được khắc vào bảng mạch.

Bảng mạch in chính trong máy tính được gọi là bảng hệ thống hoặc bo mạch chủ (system board or **motherboard**), trong khi *những bảng mạch nhỏ hơn cắm vào các khe trên bảng mạch chính* được gọi là **bảng mở rộng (expansion boards)**.

- Các thành phần nổi bật nhất trên bo mạch chủ là các con chip. **Chip** là một phần vật liệu bán dẫn (semiconducting material), thường là silicon, trên đó các **mạch điện tử và cổng logic** được chế tạo. Sản phẩm tạo ra được gọi là **mạch tích hợp (integrated circuit)**.

- Bo mạch chủ có một khe hoặc ổ cắm cho chip xử lý (ngoài ra còn có các khe cắm cho chip bộ nhớ, chip điều khiển I/O và các thành phần máy tính quan trọng khác), thường chứa nhiều lõi riêng lẻ, được gọi là bộ xử lý đa lõi (a multicore processor).

- Một bo mạch chủ hiện đại chỉ kết nối một vài thành phần chip riêng lẻ, mỗi chip chứa từ vài nghìn đến hàng trăm triệu bóng bán dẫn, để khe cắm mở rộng (expansion slots) cho phép đưa nhiều thành phần hơn vào **bảng mở rộng (expansion boards)**.

Hình 1.2 cho thấy một chip xử lý có 8 lõi và bộ nhớ đệm L3. Hình vẽ chỉ ra rằng bộ nhớ đệm L3 chiếm hai phần riêng biệt trên bề mặt chip. Tuy nhiên, thông thường, *tất cả các lõi đều có quyền truy cập vào toàn bộ bộ nhớ đệm L3* thông qua các mạch điều khiển đã nói ở trên.

- Phân tích cấu trúc của một lõi đơn. Chức năng của lõi là:

■ **Instruction logic:** **Tim nạp lệnh và giải mã** từng lệnh để **xác định thao tác lệnh và vị trí bộ nhớ** của bất kỳ toán hạng nào.

■ **Đơn vị số học và logic (Arithmetic and logic unit - ALU):** Thực hiện hoạt động được chỉ định bởi một lệnh.

■ **Load/store logic:** Quản lý việc *truyền dữ liệu đến và đi từ bộ nhớ chính thông qua bộ đệm*.

Lõi cũng chứa bộ nhớ đệm L1, được phân chia thành **bộ đệm lệnh** (instruction cache) (I-cache) được sử dụng để **truyền các lệnh đến - từ bộ nhớ chính**, và **bộ đệm dữ liệu** L1 để **truyền toán hạng và kết quả**. Thông thường, các chip xử lý ngày nay cũng bao gồm bộ đệm L2 như một phần của lõi, cũng được phân chia thành bộ đệm lệnh và bộ đệm dữ liệu.

1.3 A BRIEF HISTORY OF COMPUTERS

First Generation: Vacuum Tubes (Bóng đèn chân không)

- Máy tính chỉ hiểu các dãy bit 0, 1. Lúc bấy giờ, để máy tính hiểu được các dãy này, người ta sử dụng các bóng đèn chân không (Đèn tia điện tử), với **1 bit tương đương 1 bóng đèn**.
- Máy tính đầu tiên sử dụng công nghệ này có tên là ENIAC, was the world's first general purpose electronic digital computer. The project was a response to U.S. needs during World War II.
- In 1943, the work began on the ENIAC. Result: weighing **30 tons**, occupying **1500 square feet** of floor space, and containing more than **18,000 vacuum tubes**. When operating, it consumed **140 kilowatts** of power, thực hiện **5000 phép cộng** mỗi giây.
- ENIAC là máy thập phân chứ không phải một máy nhị phân. Tức là, **số được biểu diễn dưới dạng thập phân**, và phép tính số học thực hiện trong hệ thập phân. Bộ nhớ của nó bao gồm 20 bộ tích lũy (accumulator), mỗi bộ tích lũy có thể trữ một số thập phân 10 chữ số. **Mỗi chữ số lại được biểu diễn bởi một vòng gồm 10 đèn ống chân không**. (Hệ thập phân gồm 10 số: 0-9) Tại một thời điểm, chỉ có một đèn ống chân không ở trạng thái ON, thể hiện một trong 10 chữ số.
- Hạn chế lớn nhất của ENIAC là nó phải được lập trình thủ công bằng cách thiết lập các công tắc chuyển mạch và cắm rút dây cáp. Bên cạnh đó nó còn *tỏa khá nhiều nhiệt*.

THE VON NEUMANN MACHINE

- Nhiệm vụ nhập và thay đổi chương trình cho ENIAC là cực kỳ tẻ nhạt. Nhưng giả sử nếu có một chương trình được thể hiện dưới dạng phù hợp để **lưu trữ trong bộ nhớ cùng với dữ liệu**. Khi đó, máy tính có thể đọc được lệnh từ bộ nhớ, và một chương trình có thể được thiết lập hoặc thay đổi bằng cách thiết lập giá trị của một phần trong bộ nhớ.
- Ý tưởng này được gọi là **chương trình lưu trữ được (mã hóa lệnh dạng nhị phân, lưu trong bộ nhớ)**, thường được biết đến là ý tưởng của các nhà thiết kế ENIAC, đặc biệt là nhà toán học John von Neumann, chuyên gia tư vấn cho dự án ENIAC. Cùng thời điểm đó Alan Turing cũng phát triển ý tưởng này. Công bố đầu tiên của ý tưởng này là do von Neumann đưa ra năm 1945 trong một máy tính mới, **EDVAC (Electronic Discrete Variable Computer)**.
- Năm 1946, von Neumann và các cộng sự của ông bắt đầu thiết kế một máy tính chương trình lưu trữ mới, gọi là **máy tính IAS**, tại Viện Nghiên cứu cao cấp Princeton (Princeton Institute for Advanced Studies). Mặc dù cho đến năm 1952 vẫn không hoàn thành, **máy tính IAS được ghi nhận là nguyên mẫu của tất cả các máy tính đa năng sau này**.

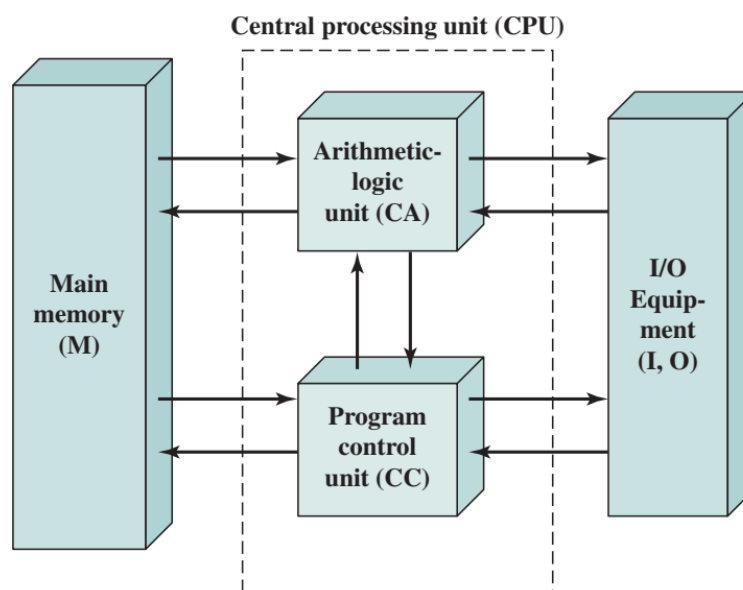


Figure 2.1 Structure of the IAS Computer

- The structure of the IAS computer consists of

■ A **main memory**, which stores both **data and instructions**

*the term **instruction** refers to a machine instruction that is directly interpreted and executed by the processor.*

- Tồn tại trên mọi máy tính
- Sử dụng bộ nhớ bán dẫn
- Tổ chức thành các ngăn nhớ được đánh địa chỉ (thường đánh địa chỉ cho từng byte nhớ)
- Nội dung của ngăn nhớ có thể thay đổi, song địa chỉ vật lý của ngăn nhớ luôn cố định
- CPU muốn đọc/ghi ngăn nhớ cần phải biết địa chỉ ngăn nhớ đó (Cụ thể ở Figure 1.6)

■ An **arithmetic and logic unit (ALU)** capable of operating on **binary data**

■ A **control unit**, which **interprets the instructions** in memory and causes them to be **executed**

■ **Input–output (I/O)** equipment **operated by the control unit**

Cấu trúc này đã được phác thảo trong đề xuất trước đó của von Neumann, được trích dẫn một phần:

2.2 Thứ nhất: Vì thiết bị chủ yếu là máy tính nên nó sẽ phải thường xuyên thực hiện các phép tính số học cơ bản: cộng, trừ, nhân và chia => nó phải chứa các cơ quan chuyên biệt.

Phần số học trung tâm của thiết bị có thể sẽ phải tồn tại và phần này tạo thành phần cụ thể đầu tiên: **CA**.

2.3 Thứ hai: Việc điều khiển logic của thiết bị = trình tự hoạt động thích hợp của thiết bị, có thể được thực hiện một cách hiệu quả nhất bởi *cơ quan điều khiển trung tâm (central control organ)*. Nếu thiết bị “mềm dẻo” (elastic), thì phải có *sự phân biệt giữa các lệnh cụ thể* được đưa ra và xác định *một vấn đề cụ thể*, và các cơ quan kiểm soát chung (general control organs) *xem xét các lệnh này được thực hiện*. (~ *Mỗi nơi trong cơ quan điều khiển đều có một công việc cụ thể riêng biệt để xử lý*).

Cái trước phải được lưu trữ theo cách nào đó; **cái sau** được thể hiện bằng các bộ phận **vận hành** của thiết bị. Khi nói đến cơ quan điều khiển trung tâm, chúng tôi chỉ muốn nói đến chức năng thứ hai này và các cơ quan thực hiện chức năng đó tạo thành phần cụ thể thứ hai: **CC**.

2.4 Thứ ba: Thiết bị thực hiện các chuỗi thao tác dài và phức tạp (cụ thể là các phép tính) đều **phải có bộ nhớ đáng kể**.

Các lệnh chi phối một vấn đề phức tạp **có thể tạo thành tài liệu đáng kể** (~*rất nhiều và đòi hỏi cần có nơi để lưu trữ các lệnh*), đặc biệt nếu mã mang tính chất “tình huống” (trong hầu hết các cách sắp xếp). Tài liệu này phải được ghi nhớ.

Tổng bộ nhớ cấu thành phần cụ thể thứ ba của thiết bị: **M**.

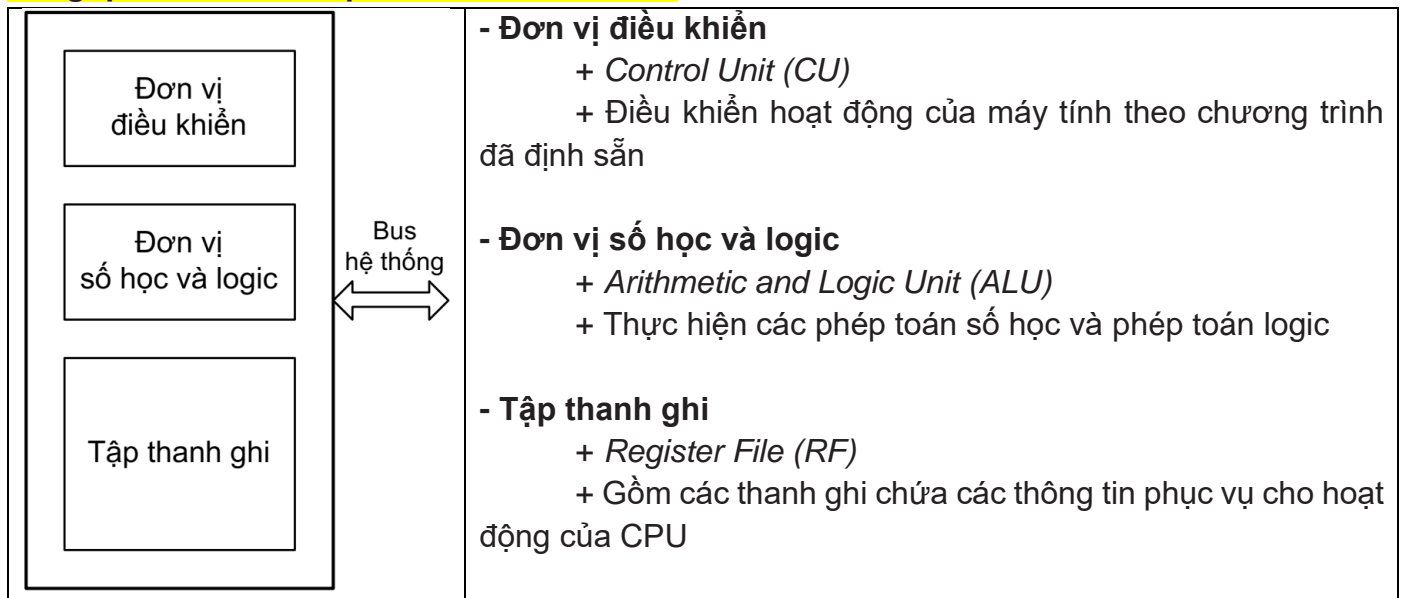
2.6 Ba phần cụ thể CA, CC (*gọi chung là C*) và M tương ứng với các nơ-ron liên kết trong hệ thần kinh của con người. Vẫn còn phải thảo luận về sự tương đương của tế bào thần kinh cảm giác hoặc hướng tâm và tế bào thần kinh vận động hoặc ly tâm. Đây là cơ quan **đầu vào và đầu ra** của thiết bị.

Thiết bị phải có khả năng duy trì liên lạc đầu vào và đầu ra (giác quan và động cơ) với một số phương tiện cụ thể thuộc loại này. Phương tiện sẽ được gọi là phương tiện ghi bên ngoài của thiết bị: **R**. (~*thiết bị ngoại vi*)

2.7 Thứ tư: Thiết bị phải có các cơ quan để **truyền thông tin từ R đến các phần cụ thể C và M**. Các cơ quan này tạo thành đầu vào của nó, phần cụ thể thứ 4: **I**. Sẽ thấy rằng tốt nhất nên thực hiện **mọi chuyển giao từ R (bằng cách I) vào M và không bao giờ trực tiếp từ C**.

2.8 Thứ năm: Thiết bị phải có các cơ quan để **chuyển từ các phần cụ thể C và M của nó sang R**. Các cơ quan này tạo thành đầu ra của nó, phần cụ thể thứ năm: **O**. Sẽ lại thấy rằng tốt nhất là thực hiện mọi **chuyển đổi từ M (bằng cách O) vào R và không bao giờ trực tiếp từ C**.

Tổng quát: Các thành phần cơ bản của CPU



- Tuy nhiên ở hình 1.6, ta thấy không có đường dẫn nào trực tiếp để chép dữ liệu từ I/O vào memory, mà đều phải thông qua CPU. Điều này gây chậm hiệu suất MT, do thời gian trì hoãn & năng lượng tiêu thụ khi **di chuyển** dữ liệu **từ bộ nhớ đến CPU hoặc ngược lại** thông qua các đường dẫn là khá lớn, chiếm nhiều hơn so với năng lượng & thời gian khi lưu trữ & xử lý.
→ Khắc phục: Đưa memory tích hợp lên CA (PIM: processing in memory), hoặc giảm khoảng cách đường truyền từ CPU đến memory.

- **Bộ nhớ** của IAS bao gồm **$4.096 = 2^{12}$ vị trí lưu trữ** (hình 1.6), được gọi là các **từ (words)**, mỗi vị trí có **40 chữ số nhị phân** (bit). Cả dữ liệu và lệnh đều được lưu trữ ở đó.

Các số được biểu diễn dưới dạng **nhị phân** và mỗi lệnh là một mã **nhị phân**. (Hình 1.7)

Mỗi số được biểu thị bằng **một bit dấu và một giá trị 39 bit**.

Ngoài ra, một từ có thể chứa **hai lệnh 20 bit**, với mỗi lệnh bao gồm **8 bit mã lệnh (opcode) chỉ định hoạt động** được thực hiện và **12 bit địa chỉ chỉ định một trong các từ trong bộ nhớ** (được đánh số từ 0 đến 999).).

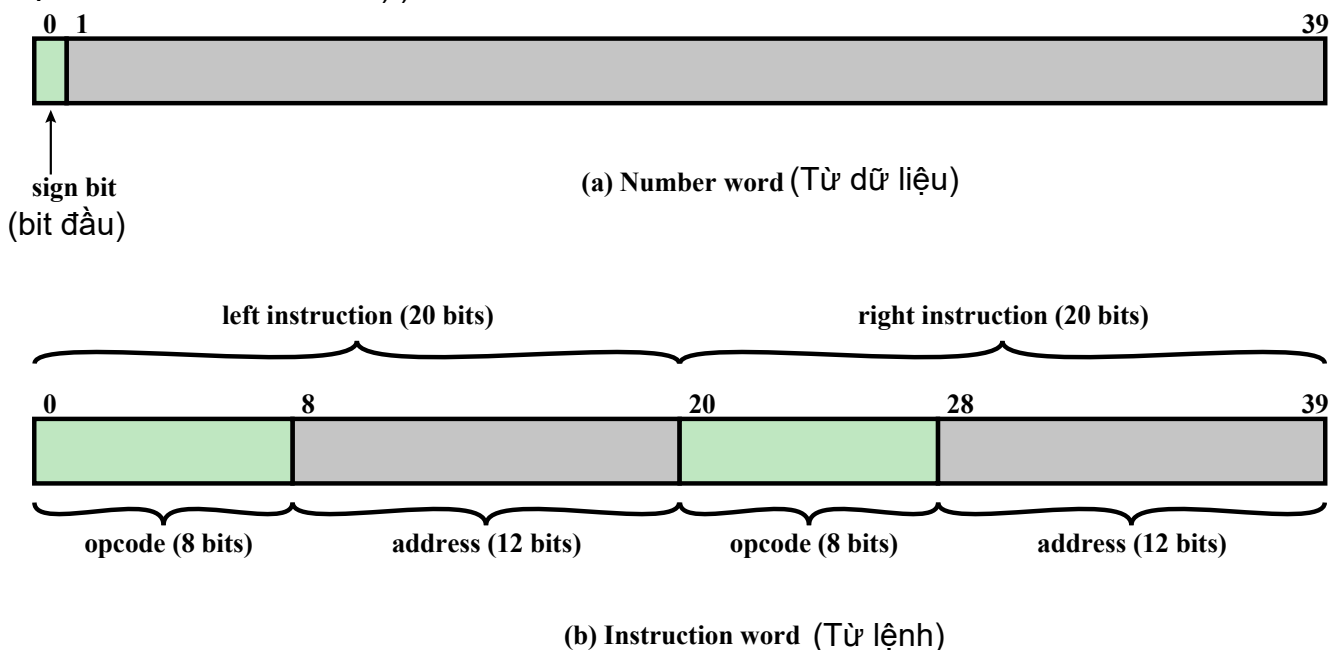


Figure 1.7 IAS Memory Formats

- Bộ điều khiển (Control unit) vận hành IAS bằng cách **tìm nạp các lệnh từ bộ nhớ và thực hiện từng lệnh một**. Các hoạt động này được giải thích bằng **Hình 1.6 (bên dưới)**. Hình này cho thấy cả thiết bị điều khiển và ALU đều chứa các vị trí lưu trữ, được gọi là các **thanh ghi (registers)**, được xác định như sau:

■ **Thanh ghi bộ nhớ đệm (Memory buffer register - MBR):**

- + Chứa một word sắp lưu trong bộ nhớ hoặc sắp gửi đến các cổng I/O,
- + Hoặc được sử dụng để nhận một word từ bộ nhớ hoặc từ các cổng I/O.

■ **Thanh ghi địa chỉ bộ nhớ (Memory address register - MAR):** Chỉ định địa chỉ trong bộ nhớ của word chuẩn bị được ghi hoặc đọc vào MBR.

■ **Thanh ghi lệnh (Instruction register - IR):** Chứa mã tác vụ 8 bit của lệnh đang được thực thi.

■ **Thanh ghi đệm lệnh (Instruction buffer register - IBR):** Được sử dụng để lưu giữ tạm thời lệnh nằm bên tay phải của một word trong bộ nhớ.

■ **Bộ đếm chương trình (Program counter - PC):** Lưu giữ địa chỉ của cặp lệnh tiếp theo được lấy từ bộ nhớ.

■ **Bộ tích lũy (Accumulator - AC) và thương số nhân (multiplier quotient - MQ):** Được sử dụng để lưu giữ tạm thời các toán hạng và kết quả của các phép tính trong ALU.

Ví dụ: kết quả của phép nhân hai số 40 bit là số 80 bit; 40 bit có trọng số cao nhất được lưu trữ trong AC và các bit có trọng số thấp nhất được lưu trữ trong MQ.

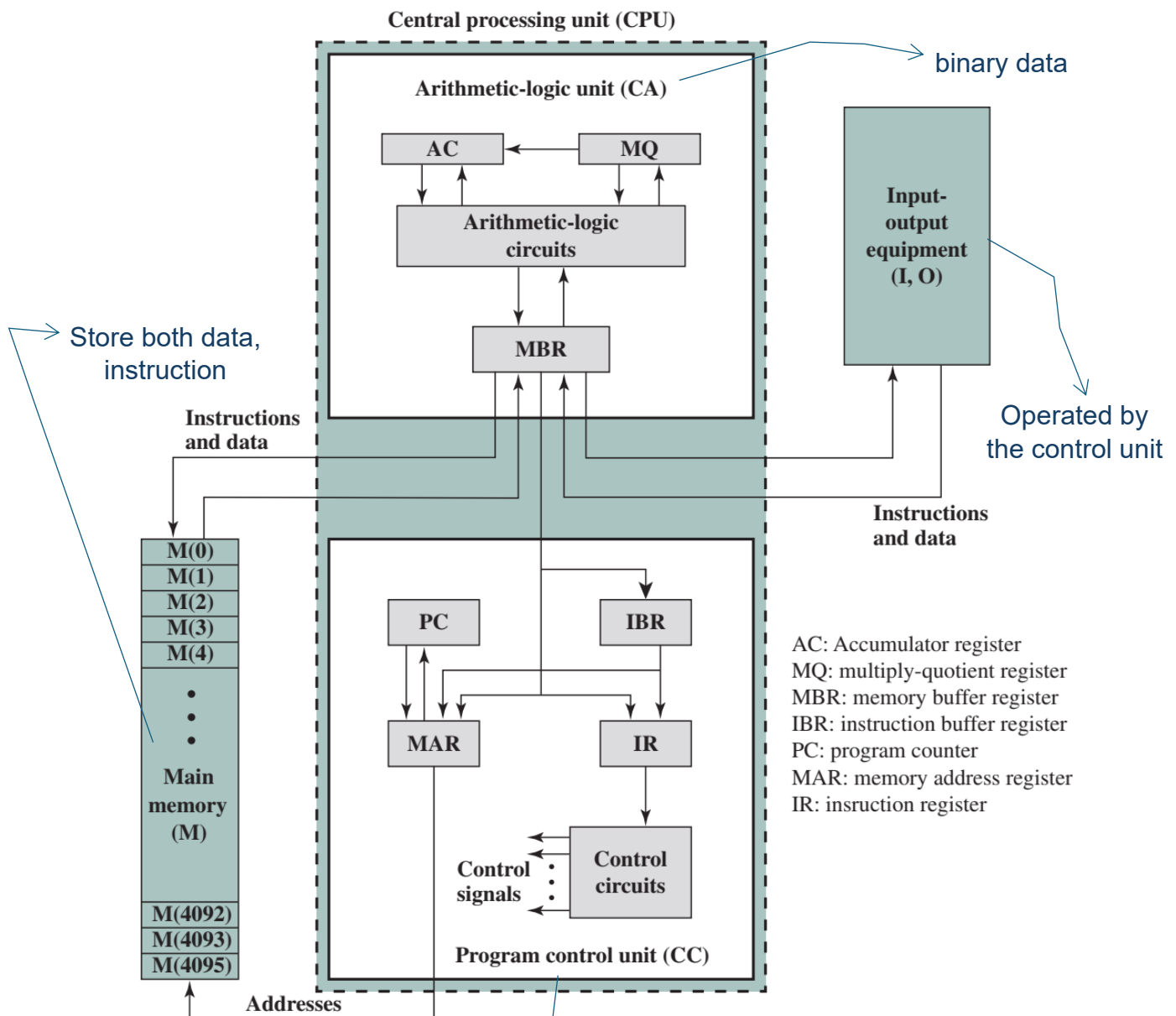


Figure 1.6 IAS Structure the instructions in memory and causes them to be executed

- IAS hoạt động bằng cách thực hiện **lặp đi lặp lại một chu kỳ lệnh**. Mỗi chu kỳ lệnh bao gồm hai chu kỳ con. Trong chu kỳ tìm nạp, **opcode của lệnh tiếp theo được tải vào IR và phần địa chỉ được tải vào MAR**. Lệnh này có thể được lấy từ IBR hoặc có thể được lấy từ bộ nhớ bằng cách tải một từ (word) vào MBR, sau đó xuống IBR, IR và MAR.
- Khi opcode nằm trong IR, chu trình thực thi sẽ được thực hiện. Mạch điều khiển (Control circuitry) dịch mã lệnh và thực hiện lệnh bằng cách gửi đi các tín hiệu điều khiển thích hợp để khiến dữ liệu được di chuyển hoặc một thao tác được thực hiện bởi ALU.
- Máy tính IAS có tổng cộng 21 lệnh được liệt kê trong bảng dưới, được phân loại như sau:
 - **Truyền dữ liệu (Data transfer)**: Di chuyển dữ liệu giữa bộ nhớ và thanh ghi ALU hoặc giữa hai thanh ghi ALU.
 - **Nhánh vô điều kiện (Unconditional branch)**: Bộ điều khiển thực hiện các lệnh theo trình tự từ bộ nhớ có thể được thay đổi bằng lệnh rẽ nhánh, để hoạt động được lặp đi lặp lại.
 - **Nhánh có điều kiện (Conditional branch)**: Nhánh có thể được tạo phụ thuộc vào một điều kiện, từ đó dẫn đến các quyết định.
 - **Số học (Arithmetic)**: Các phép toán được thực hiện bởi ALU.
 - **Sửa đổi địa chỉ (Address modify)**: Cho phép tính toán các địa chỉ trong ALU và sau đó chèn vào các lệnh được lưu trong bộ nhớ.

Kiểu lệnh	Opcode	Biểu diễn ký tự	Mô tả
Truyền dữ liệu	00001010	LOAD MQ	Truyền nội dung thanh ghi MQ tới thanh ghi AC
	00001001	LOAD MQ.M(X)	Truyền nội dung vị trí nhớ X tới MQ
	00100001	STOR M(X)	Truyền nội dung trong AC tới vị trí nhớ X
	00000001	LOAD M(X)	Truyền M(X) tới thanh ghi AC
	00000010	LOAD -M(X)	Truyền -M(X) tới thanh ghi AC
	00000011	LOAD M(X)	Truyền giá trị tuyệt đối của M(X) tới AC
	00000100	LOAD - M(X)	Truyền - M(X) tới thanh ghi AC
Nhánh không điều kiện	00001101	JUMP M(X,0:19)	Lấy lệnh tiếp theo từ nửa trái của M(X)
	00001110	JUMP M(X,20:39)	Lấy lệnh tiếp theo từ nửa phải của M(X)
Nhánh có điều kiện	00001111	JUMP +M(X,0:19)	Nếu số trong AC không âm, lấy lệnh tiếp theo từ nửa trái của M(X)
	00010000	JUMP +M(X,20:39)	Nếu số trong AC không âm, lấy lệnh tiếp theo từ nửa phải của M(X)

Số học	00000101	ADD M(X)	Cộng M(X) với AC; kết quả đặt vào AC
	00000111	ADD [M(X)]	Cộng [M(X)] với AC; kết quả đặt vào AC
	00000110	SUB M(X)	Lấy AC trừ M(X); kết quả đặt vào AC
	00001000	SUB [M(X)]	Lấy AC trừ [M(X)]; kết quả đặt vào AC
	00001011	MUL M(X)	Nhân M(X) với MQ; các bit quan trọng nhất của kết quả đặt vào AC, các bit ít quan trọng nhất của kết quả đặt vào MQ
	00001100	DIV M(X)	Chia AC cho M(X); thương số đặt vào MQ, phần dư đặt vào AC
	00010100	LSH	Nhân AC với 2; tức là dịch trái một vị trí bit
	00010101	RSR	Chia AC cho 2; tức là dịch phải một vị trí
Sửa đổi địa chỉ	00010010	STOR M(X,8:19)	Thay trường địa chỉ bên trái của M(X) bằng 12 bit ngoài cùng bên phải của AC
	00010011	STOR M(X,28:39)	Thay trường địa chỉ bên phải của M(X) bằng 12 bit ngoài cùng bên phải của AC

Table 1.1 The IAS Instruction Set

- Bảng trên trình bày các lệnh (không bao gồm các lệnh I/O) ở dạng tượng trưng, dễ đọc.
- Ở dạng nhị phân, mỗi lệnh phải tuân theo định dạng của Hình 1.7b.
 - + **Phần opcode (8 bit đầu tiên)** chỉ định lệnh nào trong số 21 lệnh sẽ được thực thi.
 - + **Phần địa chỉ (12 bit còn lại)** chỉ định vị trí nào trong số 4.096 vị trí bộ nhớ sẽ tham gia vào việc thực hiện lệnh.
- Phép nhân yêu cầu 39 phép toán con, một phép tính cho mỗi vị trí bit ngoại trừ bit dấu.

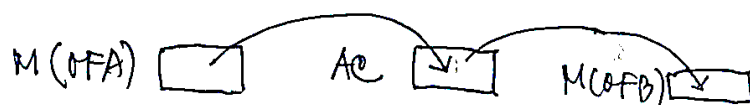
1.4 Given the **memory contents** of the IAS computer shown below,

Address	Contents (~ Instruction word)
08A	010FA210FB
08B	010FA0F08D
08C	020FA210FB

show the assembly language code for the program, starting at address 08A. Explain what this program does.

Thập phân (Decimal)	Nhị phân (Binary)	Thập lục phân (Hexadecimal)	Bát phân (Octal)
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

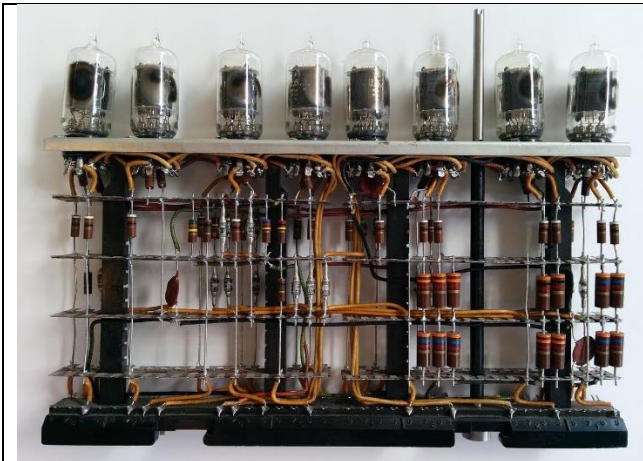
Address	Contents		
	<p>Cách làm:</p> <ul style="list-style-type: none"> Theo bài học ta chia lệnh này thành nửa trái & nửa phải, tương ứng với 5 chữ số mỗi bên. Đối với mỗi bên lấy 2 chữ số đầu tương ứng với opcode (mỗi chữ số thập lục phân tương ứng với 4 chữ số nhị phân theo bảng trên), 3 chữ số thập lục phân còn lại chính là địa chỉ ứng với 12 bits nhị phân. Đối với bài này ta giữ lại 3 chữ số thập lục phân cuối để làm địa chỉ. Tra 8 bit đầu trong bảng 1.1, cột Opcode, đối chiếu với các cột còn lại. 		
08A	Left instruction: 010FA = 0000 0001 0FA Right instruction: 210FB = 0010 0001 0FB	LOAD M(0FA) STOR M(0FB)	Transfer M(0FA) to the accumulator Transfer contents of accumulator to memory location 0FB
08B	010FA = 0000 0001 0FA 0F08D = 0000 1111 08D	LOAD M(0FA) JUMP + M(08D, 0:19)	Transfer M(0FA) to the accumulator If number in the accumulator is nonnegative, take next instruction from left half of M(08D) (but 08D has no instruction)
08C	020FA = 0000 0010 0FA 210FB = 0010 0001 0FB	LOAD – M(0FA) STOR M(0FB)	Transfer –M(X) to the accumulator Transfer contents of accumulator to memory location 0FB
08D	-	-	-



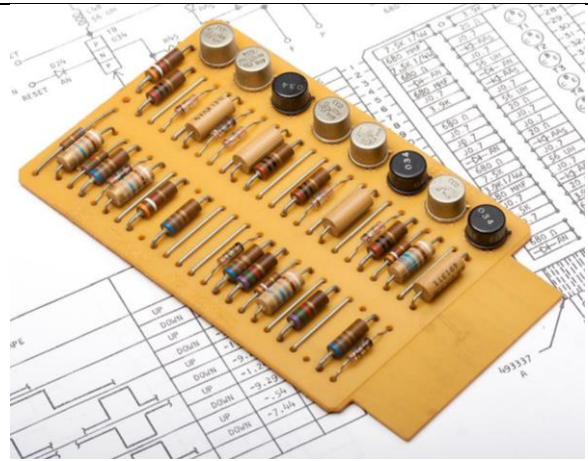
This program is to store the absolute value of content at memory location 0FA into memory location 0FB.

Second Generation: Transistors (Linh kiện bán dẫn)

- Là một thiết bị rắn (solid-state device) làm từ silicon.
- smaller, cheaper, and generates less heat than a vacuum tube, can be used in the same way as a vacuum tube.
- Was invented at Bell Labs in 1947
- Mãi đến cuối những năm 1950, máy tính bán dẫn hoàn toàn mới được thương mại hóa.
- Hiệu năng xử lý tốt hơn, dung lượng bộ nhớ lớn hơn, kích thước nhỏ hơn.
- CU và ALU phức tạp hơn
- Sử dụng các ngôn ngữ lập trình bậc cao
- Xuất hiện các phần mềm hệ thống (giống như các hệ điều hành hiện đại như Window hay Linux) cho phép:
 - Tải chương trình
 - Di chuyển dữ liệu tới các thiết bị ngoại vi và thư viện
 - Thực hiện các tính toán thông thường



Vacuum tube logic module from a 700 series IBM computer



Transistorized IBM Standard Modular System (SMS) card used in the 7000 series

- Kích thước của bộ nhớ chính, theo bội số của 2^{10} 36-bit words, tăng từ 2k ($1k = 2^{10}$) lên 32k từ; thời gian truy cập một từ của bộ nhớ = thời gian chu kỳ bộ nhớ, giảm từ 30 micro giây xuống 1,4 micro giây. Số lượng opcode tăng từ 24 lên 185.
- Tốc độ tương đối của CPU tăng lên theo ước số của 50. Cải thiện tốc độ đạt được nhờ cải tiến thiết bị điện tử và mạch điện phức tạp hơn.
- Bộ điều khiển lấy **hai từ liên kế từ bộ nhớ để tìm nạp lệnh**. Bộ điều khiển phải **truy cập bộ nhớ cho một lệnh chỉ trong một nửa chu kỳ lệnh** → Giảm đáng kể thời gian chu kỳ lệnh trung bình.
- Kênh dữ liệu (Data channel) là một mô-đun I/O **độc lập** với **bộ xử lý và tập lệnh riêng**. CPU không thực thi các lệnh I/O chi tiết, chúng được lưu trữ trong bộ nhớ chính để được **thực thi bởi bộ xử lý trong chính kênh dữ liệu**.
CPU bắt đầu quá trình truyền I/O bằng cách gửi tín hiệu điều khiển đến kênh dữ liệu, hướng dẫn nó thực hiện một chuỗi lệnh trong bộ nhớ. Kênh dữ liệu thực hiện nhiệm vụ của nó một cách độc lập với CPU và báo hiệu cho CPU khi hoạt động hoàn tất. Sự sắp xếp này giúp giảm bớt gánh nặng xử lý đáng kể cho CPU.

Mô-đun vào-ra

- Chức năng: nối ghép các thiết bị vào-ra với máy tính
- Mỗi mô-đun vào-ra có **một hoặc một vài cổng vào-ra** (I/O Port)
- **Mỗi cổng** vào-ra được **đánh một địa chỉ** xác định

- Các thiết bị vào-ra được kết nối và trao đổi dữ liệu với máy tính thông qua các cổng vào-ra
- CPU muốn trao đổi dữ liệu với thiết bị vào-ra, cần phải **biết địa chỉ của cổng vào-ra tương ứng**

- Một tính năng mới khác là bộ ghép kênh (multiplexor), đó là điểm cuối trung tâm cho các kênh dữ liệu, CPU và bộ nhớ. Bộ ghép kênh **lên lịch truy cập vào bộ nhớ từ CPU và các kênh dữ liệu**, cho phép các thiết bị này hoạt động độc lập.

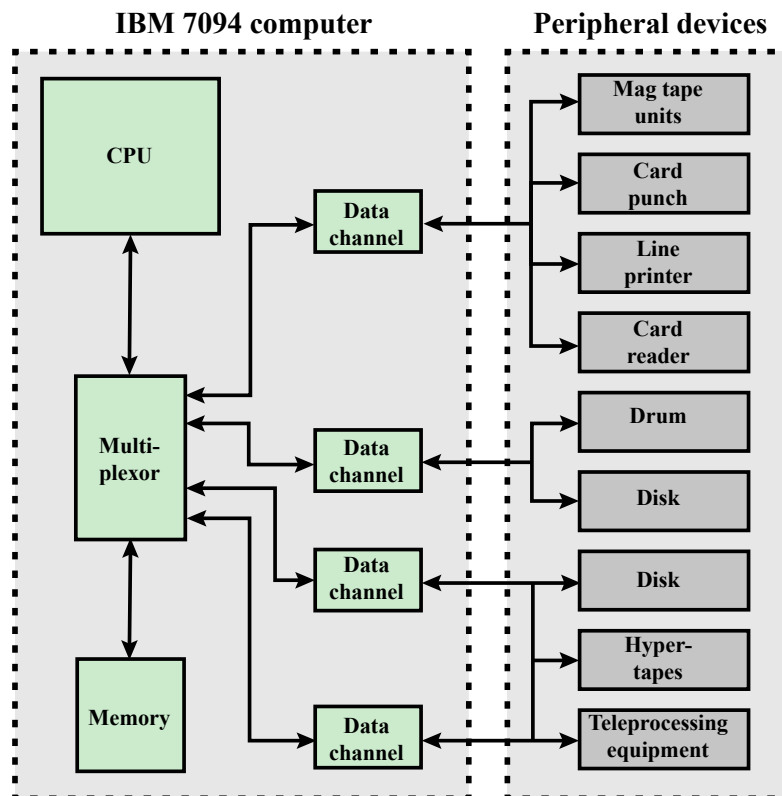


Figure 1.9 An IBM 7094 Configuration

Third Generation: Integrated Circuits (IC) (Mạch tích hợp/ Vi mạch)

- Từ năm 1950 đến 1960, máy tính được chế tạo từ các **linh kiện rời (discrete component)** (transistor, điện trở, tụ điện)
- Các linh kiện sản xuất đơn lẻ, độc lập, đóng gói riêng.
- Sau đó được hàn lại hoặc nối với nhau lên trên một bảng mạch masonite.
- Quá trình sản xuất tốn kém và cồng kềnh.
- Các máy tính thế hệ thứ hai gồm khoảng 10000 transistor, sau đó con số này lên tới hàng trăm nghìn

- 1958 - phát minh ra mạch tích hợp → công nghệ nền tảng cho máy tính thế hệ thứ 3
- Hai thành viên quan trọng nhất của thế hệ máy tính thứ ba là IBM System/360 và DEC PDP-8
- Các thành phần cơ bản của máy tính số được chia thành 2 loại: **gate (cổng logic)** và **memory cell (ô nhớ)**

- **Gate:** là thiết bị thực hiện các **hàm logic và Boolean** đơn giản
- **Memory cell:** là **thiết bị lưu trữ một bit** của dữ liệu, mang hai giá trị (trạng thái) **0 hoặc 1**

Chúng thực hiện **bốn chức năng** cơ bản của máy tính: lưu trữ, truyền dữ liệu, xử lý dữ liệu và điều khiển.

- **Lưu trữ dữ liệu:** sử dụng các memory cell

- **Xử lý dữ liệu:** sử dụng các gate
 - **Truyền dữ liệu:** Di chuyển dữ liệu - Dữ liệu được di chuyển vào và ra bộ nhớ trên các đường dẫn giữa các bộ phận của máy tính và *thông qua các gate*.
 - **Điều khiển:** các tín hiệu điều khiển hoạt động của gate và memory cell
- Máy tính gồm các gate, memory cell và sự liên kết giữa chúng

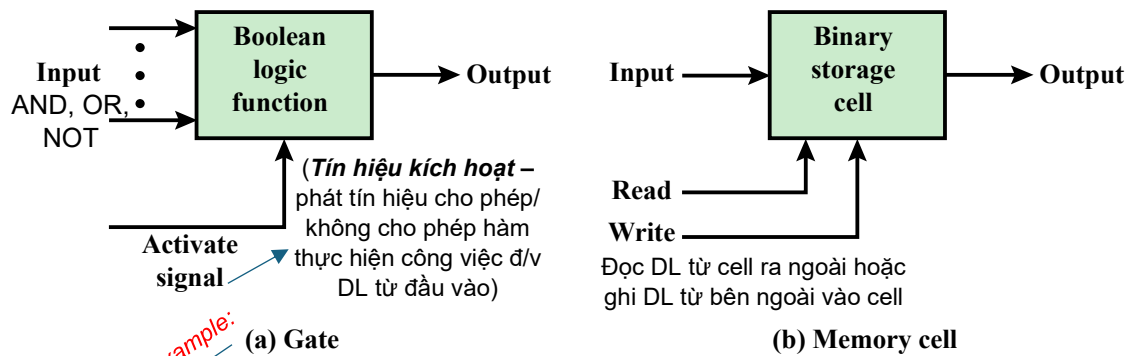
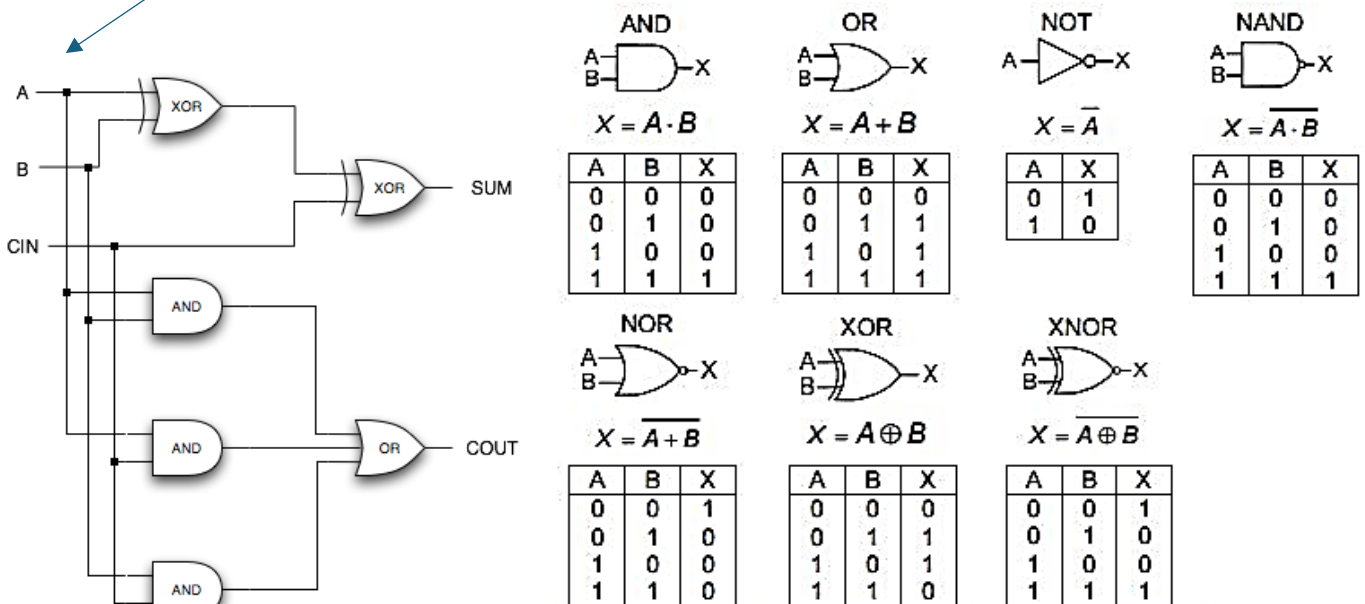


Figure 1.10 Fundamental Computer Elements



Hình 8.1: Các phép toán đại số bool với bảng sự thật và cổng logic

- Mạch tích hợp: một tấm **wafer** silicon mỏng, chia thành ma trận các vùng nhỏ, **mỗi vùng chứa một mạch** giống hệt nhau được gọi là **chip**.

- Một chip chứa nhiều gate hoặc memory cell
- Chip được đóng gói (packaged) và cung cấp các chân để gắn vào các thiết bị ngoài chip.

- Ban đầu, số lượng gate/memory cell trong một chip còn ít => công nghệ này được gọi là **SSI – small-scale integration: mạch tích hợp kích thước nhỏ**

- Về sau, số lượng G/C trong một chip ngày càng nhiều

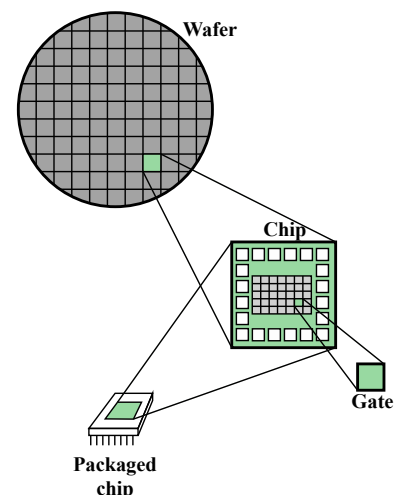


Figure 1.11 Relationship Among Wafer, Chip, and Gate



- Phân loại vi mạch theo qui mô tích hợp:

SSI - Small Scale Integration

MSI - Medium Scale Integration

LSI - Large Scale Integration

VLSI - Very Large Scale Integration

ULSI - Ultra Large Scale Integration

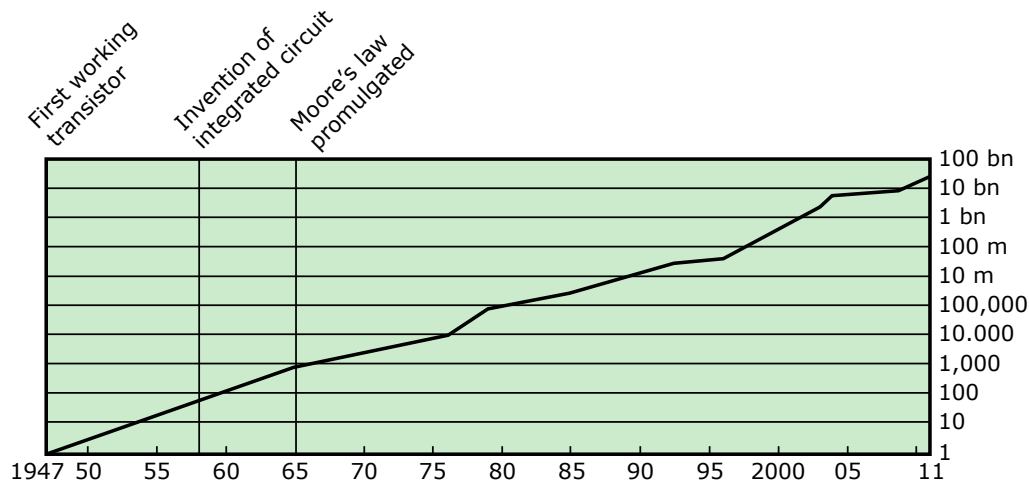


Figure 1.12 Growth in Transistor Count on Integrated Circuits (DRAM memory)

Hình vẽ này phản ánh **định luật Moore** nổi tiếng, do Gordon Moore, người đồng sáng lập của Intel, đưa ra vào năm 1965.

Ông Moore quan sát thấy **số lượng transistor có thể đặt trên một con chip sẽ tăng gấp đôi sau mỗi năm** và dự đoán chính xác tốc độ này sẽ duy trì trong tương lai gần. Đáng ngạc nhiên, tốc độ tăng trưởng này tiếp tục năm này qua năm khác, thập kỷ này qua thập kỷ khác. Tốc độ sau đó chậm lại còn tăng gấp đôi sau mỗi 18 tháng vào những năm 1970 và duy trì cho đến ngày nay.

Hệ quả:

1. Giá thành của mạch bộ nhớ và logic máy tính đã giảm rất mạnh – Giá thành của một con chip hầu như không thay đổi.
2. Chiều dài đường dẫn điện được rút ngắn, tốc độ hoạt động tăng (Do các phần tử logic và bộ nhớ được đặt gần nhau hơn trên các chip có mật độ dày đặc hơn).
3. Máy tính trở nên nhỏ gọn hơn và thuận tiện cho sử dụng ở các môi trường khác nhau
4. Giảm tiêu thụ điện năng và yêu cầu bộ làm mát
5. Các kết nối trên mạch tích hợp có độ tin cậy cao hơn nhiều so với mối hàn. Với nhiều mạch điện hơn trên mỗi chip, kết nối giữa các chip sẽ ít hơn

Later Generations

- Các thế hệ sau dựa trên sự phát triển của các công nghệ mạch tích hợp:

- + Large-scale integration (LSI) - mạch tích hợp cỡ lớn: hơn 1000 thiết bị tích hợp trong m chip **(Fourth Generation)**
 - + Very-large-scale integration (VLSI): 10000 thiết bị/chip **(Fifth Generation)**
 - + Hiện nay, Ultra-large-scale integration (ULSI): hơn 1 tỉ thành phần/chip **(Sixth Generation)**
- Các công nghệ này là nền tảng cho sự phát triển của các thế hệ máy tính và là công nghệ cơ bản cho việc sản xuất và chế tạo các linh kiện cơ bản:
- + Bộ nhớ bán dẫn: mạch tích hợp ban đầu được sử dụng để chế tạo Bộ xử lý, tuy nhiên, sau này người ta cũng sử dụng công nghệ đó để chế tạo bộ nhớ máy tính
 - + Vi xử lý: các bộ xử lý có kích thước nhỏ

Evolution of Intel Microprocessors

- Vi xử lý 4004 có thể **cộng hai số 4 bit** và có thể thực hiện **phép nhân chỉ bằng việc lặp đi lặp lại phép cộng**. Theo tiêu chuẩn ngày nay, 4004 quả là thô sơ, nhưng nó đánh dấu điểm khởi đầu trong quá trình phát triển liên tục về khả năng và hiệu suất của vi xử lý.
Sự phát triển này có thể nhận thấy dễ dàng nhất qua số lượng bit mà bộ xử lý xử lý được cùng một lúc. Tham số đánh giá tốt nhất là **chiều rộng của bus dữ liệu: số bit dữ liệu có thể đưa vào hoặc gửi ra khỏi bộ xử lý** tại một thời điểm. Một tham số khác là **số bit trong thanh ghi tích lũy (accumulator)** hoặc trong tập hợp thanh ghi đa năng. Thông thường, các tham số này trùng nhau, nhưng cũng có trường hợp khác biệt. Ví dụ, một số vi xử lý được phát triển để xử lý các số 16 bit trong thanh ghi nhưng mỗi lần chỉ có thể đọc và ghi 8 bit.
- Bước tiến lớn tiếp theo trong quá trình phát triển của vi xử lý là **sự ra đời của Intel 8008** vào năm 1972. Đây là **vi xử lý 8 bit đầu tiên** và độ phức tạp gần gấp đôi so với 4004.
- Tuy nhiên, sự kiện lớn tiếp theo có sức ảnh hưởng lớn hơn cả: sự ra đời của **Intel 8080** vào năm 1974. Đây là **vi xử lý đa năng đầu tiên**. Trong khi 4004 và 8008 được thiết kế cho các ứng dụng cụ thể, 8080 được thiết kế **để trở thành CPU** của máy vi tính đa năng. Giống như 8008, 8080 là vi xử lý 8 bit. Tuy nhiên, 8080 nhanh hơn, có **tập lệnh phong phú hơn**, và có **khả năng định địa chỉ lớn**.
- Cũng trong thời gian đó, vi xử lý 16 bit đã bắt đầu được phát triển. Tuy nhiên, cho đến những năm cuối thập kỷ 1970, **vi xử lý 16 bit đa năng mới xuất hiện. Một trong số đó là 8086**. Bước tiến tiếp theo diễn ra vào năm 1981, khi cả Bell Labs và Hewlett-Packard cùng phát triển bộ **vi xử lý 32 bit**. Intel đã giới thiệu vi xử lý 32 bit của mình, **80386**, vào năm 1985.

Table 1.3 Evolution of Intel Microprocessors (page 1 of 2)**(a) 1970s Processors**

	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	8	6	3	6
Addressable memory	640 bytes	16 KB	64 KB	1 MB	1 MB

(b) 1980s Processors

	80286	386TM DX	386TM SX	486TM DX CPU
Introduced	1982	1985	1988	1989
Clock speeds	6–12.5 MHz	16–33 MHz	16–33 MHz	25–50 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

(c) 1990s Processors

	486TM SX	Pentium	Pentium Pro	Pentium II
Introduced	1991	1993	1995	1997
Clock speeds	16–33 MHz	60–166 MHz,	150–200 MHz	200–300 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

(d) Recent Processors

	Pentium III	Pentium 4	Core 2 Duo	Core i7 EE 4960X
Introduced	1999	2000	2006	2013
Clock speeds	450–660 MHz	1.3–1.8 GHz	1.06–1.2 GHz	4 GHz
Bus width	64 bits	64 bits	64 bits	64 bits
Number of transistors	9.5 million	42 million	167 million	1.86 billion
Feature size (nm)	250	180	65	22
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 L2/15 MB L3
Number of cores	1	1	2	6

1.4 EMBEDDED SYSTEMS

- Liên quan đến việc **sử dụng thiết bị điện tử và phần mềm trong một sản phẩm**
- Hàng tỷ hệ thống máy tính được sản xuất mỗi năm được nhúng trong các thiết bị lớn hơn
- Ngày nay nhiều thiết bị **sử dụng năng lượng điện** đều có hệ thống máy tính nhúng
- Các hệ thống nhúng thường được liên kết chặt chẽ với môi trường của chúng
Điều này có thể làm phát sinh những hạn chế về thời gian thực do nhu cầu tương tác với môi trường.

Các ràng buộc như tốc độ chuyển động cần thiết, độ chính xác đo lường cần thiết và khoảng thời gian cần thiết quyết định thời gian hoạt động của phần mềm

- Nếu nhiều hoạt động phải được quản lý đồng thời thì điều này đặt ra **những hạn chế về thời gian thực** phức tạp hơn

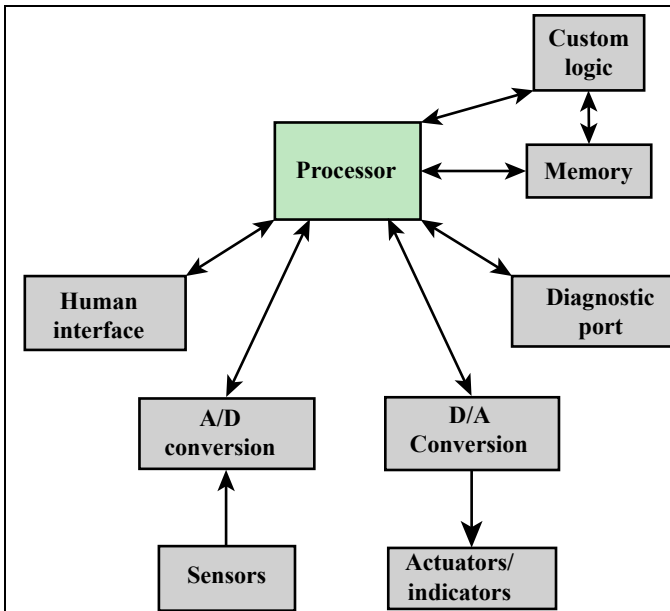


Figure 1.14 Possible Organization of an Embedded System

Ngoài bộ xử lý và bộ nhớ, còn có một số thành phần khác với máy tính để bàn hoặc máy tính xách tay thông thường:

- **Nhiều giao diện khác nhau** cho phép hệ thống đo lường, thao tác và tương tác với môi trường bên ngoài.
- **Giao diện con người (Human interface)** có thể như đèn nhấp nháy hoặc tầm nhìn của robot trong thời gian thực. Có thể không có giao diện của con người.
- **Cổng chẩn đoán (diagnostic port)** sử dụng để chẩn đoán hệ thống đang được điều khiển.
- Lập trình trường mục đích đặc biệt (Special-purpose field programmable, FPGA), dành riêng cho ứng dụng (application-specific, ASIC), hoặc thậm chí phần cứng phi kỹ thuật số có thể được sử dụng để **tăng hiệu suất hoặc độ tin cậy**.
- Phần mềm có **chức năng cố định và dành riêng** cho ứng dụng.
- Tối ưu hóa về năng lượng, kích thước mã, thời gian thực hiện, trọng lượng và kích thước cũng như chi phí.

Có một số điểm tương đồng đáng chú ý với các hệ thống máy tính đa năng:

- Ngay cả với phần mềm có chức năng cố định, **khả năng nâng cấp để sửa lỗi, cải thiện bảo mật và bổ sung chức năng đã trở nên rất quan trọng** đối với hệ thống nhúng.
- Nền tảng hệ thống nhúng **hỗ trợ nhiều loại ứng dụng**. Ví dụ: điện thoại thông minh và thiết bị nghe nhìn: TV thông minh...

The Internet of Things (IoT)

- Thuật ngữ đề cập đến khả năng kết nối mở rộng của các thiết bị thông minh, từ thiết bị gia dụng đến các cảm biến nhỏ
- Chủ yếu được điều khiển bởi các thiết bị nhúng sâu
- Các thế hệ triển khai đạt đến đỉnh cao trong IoT:
 - **Công nghệ thông tin (IT):** PC, máy chủ, bộ định tuyến, tường lửa, v.v., được những người làm CNTT trong doanh nghiệp mua dưới dạng thiết bị CNTT và chủ yếu sử dụng kết nối có dây

- **Công nghệ vận hành (OT):** Máy móc/thiết bị có CNTT nhúng do các công ty không thuộc CNTT xây dựng, chẳng hạn như máy móc y tế, SCADA, điều khiển quy trình và ki-ốt, được những người OT của doanh nghiệp mua làm thiết bị và chủ yếu sử dụng kết nối có dây
- **Công nghệ cá nhân:** Điện thoại thông minh, máy tính bảng và thiết bị đọc sách điện tử được người tiêu dùng mua dưới dạng thiết bị CNTT chỉ sử dụng kết nối không dây và thường có nhiều hình thức kết nối không dây
- **Công nghệ cảm biến/thiết bị truyền động:** Các thiết bị đơn mục đích được người tiêu dùng, nhân viên CNTT và OT mua độc quyền sử dụng kết nối không dây, thường ở dạng duy nhất, như một phần của các hệ thống lớn hơn

Embedded Operating Systems

Có hai cách tiếp cận chung để phát triển hệ điều hành nhúng (HĐH):

- + Sử dụng hệ điều hành hiện có và điều chỉnh nó cho ứng dụng nhúng
- + Thiết kế và triển khai một hệ điều hành chỉ dành cho mục đích sử dụng nhúng

Application Processors versus Dedicated Processors (Bộ xử lý ứng dụng và Bộ xử lý chuyên dụng)

- Bộ xử lý ứng dụng

- + Được xác định bởi khả năng của bộ xử lý để thực thi các hệ điều hành phức tạp
- + Một ví dụ là điện thoại thông minh – hệ thống nhúng được thiết kế để hỗ trợ nhiều ứng dụng và thực hiện nhiều chức năng khác nhau

- Bộ xử lý chuyên dụng

- + Được dành riêng cho một hoặc một số ít nhiệm vụ cụ thể theo yêu cầu của thiết bị chủ
- + Nên bộ xử lý và các thành phần liên quan có thể được thiết kế để giảm kích thước và chi phí.

Microprocessors versus Microcontrollers

- Các chip vi xử lý đời đầu (microprocessor):

- + Bao gồm các thanh ghi, ALU và một số loại đơn vị điều khiển hoặc logic xử lý lệnh.
- + Khi mật độ bóng bán dẫn tăng lên, có thể tăng độ phức tạp của kiến trúc tập lệnh và cuối cùng là thêm bộ nhớ và nhiều bộ xử lý.
- + Các chip vi xử lý hiện đại, bao gồm nhiều lõi và một lượng bộ nhớ đệm đáng kể.

- Một chip vi điều khiển (microcontroller):

- + Là một chip đơn chứa bộ xử lý, bộ nhớ cố định cho chương trình (ROM), bộ nhớ khả biến cho đầu vào và đầu ra (RAM), đồng hồ và bộ điều khiển I/O.
- + Phần bộ xử lý của bộ vi điều khiển có diện tích silicon thấp hơn nhiều so với các bộ vi xử lý khác và hiệu suất sử dụng năng lượng cao hơn nhiều.

Deeply Embedded Systems

• Tập hợp con của hệ thống nhúng

- Có bộ xử lý mà cả người lập trình và người dùng đều khó quan sát được hành vi của nó
- Sử dụng **bộ vi điều khiển** thay vì bộ vi xử lý
- Không thể lập trình được khi logic chương trình của thiết bị đã được ghi vào ROM
- **Không có tương tác với người dùng**
- Các thiết bị chuyên dụng, **có mục đích duy nhất** phát hiện thứ gì đó trong môi trường, thực hiện **mức xử lý cơ bản** và sau đó thực hiện điều gì đó với kết quả
- Thường có khả năng **không dây và xuất hiện trong các cấu hình nối mạng**, chẳng hạn như mạng cảm biến được triển khai trên một khu vực rộng lớn

- Thường có những **hạn chế** về tài nguyên cực lớn về bộ nhớ, kích thước bộ xử lý, thời gian và mức tiêu thụ điện năng

1.5 CLOUD COMPUTING

- NIST định nghĩa điện toán đám mây là:

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.

“Một mô hình cho phép truy cập mạng theo yêu cầu, thuận tiện, ở mọi nơi vào nhóm tài nguyên máy tính có thể định cấu hình chung, có thể được cung cấp và phát hành nhanh chóng với nỗ lực quản lý hoặc tương tác tối thiểu của nhà cung cấp dịch vụ.”

- Bạn có được lợi thế nhờ quy mô, quản lý mạng chuyên nghiệp và quản lý bảo mật chuyên nghiệp
- Cá nhân hoặc công ty chỉ cần trả tiền cho dung lượng lưu trữ và dịch vụ họ cần
- Nhà cung cấp đám mây đảm nhiệm vấn đề bảo mật

Cloud Networking

- Liên quan đến các mạng và chức năng quản lý mạng phải **có sẵn để kích hoạt điện toán đám mây**.
- Tập hợp các khả năng mạng cần thiết để truy cập đám mây, bao gồm việc sử dụng các **dịch vụ chuyên biệt** qua Internet, liên kết trung tâm dữ liệu doanh nghiệp với đám mây và **sử dụng tường lửa cũng như các thiết bị bảo mật mạng** khác tại các điểm quan trọng để thực thi các chính sách bảo mật truy cập

Cloud Storage

- **Tập hợp con của điện toán đám mây**
- Bao gồm lưu trữ cơ sở dữ liệu và các ứng dụng cơ sở dữ liệu được lưu trữ từ xa trên máy chủ đám mây
- Cho phép các doanh nghiệp nhỏ và người dùng cá nhân tận dụng khả năng lưu trữ dữ liệu có quy mô phù hợp với nhu cầu của họ và tận dụng nhiều ứng dụng cơ sở dữ liệu khác nhau mà **không cần phải mua, bảo trì và quản lý tài sản lưu trữ**

Cloud Services

- Mục đích cơ bản của điện toán đám mây là **cung cấp dịch vụ cho thuê tài nguyên máy tính** một cách thuận tiện. Nhà cung cấp dịch vụ đám mây (cloud service provider - CSP) duy trì tài nguyên điện toán và lưu trữ dữ liệu có sẵn trên Internet hoặc mạng riêng. Khách hàng có thể thuê một phần tài nguyên này khi cần thiết.

- Hầu như tất cả dịch vụ đám mây đều được cung cấp bằng **một trong ba mô hình** (Hình 1.17): **SaaS, PaaS và IaaS**.

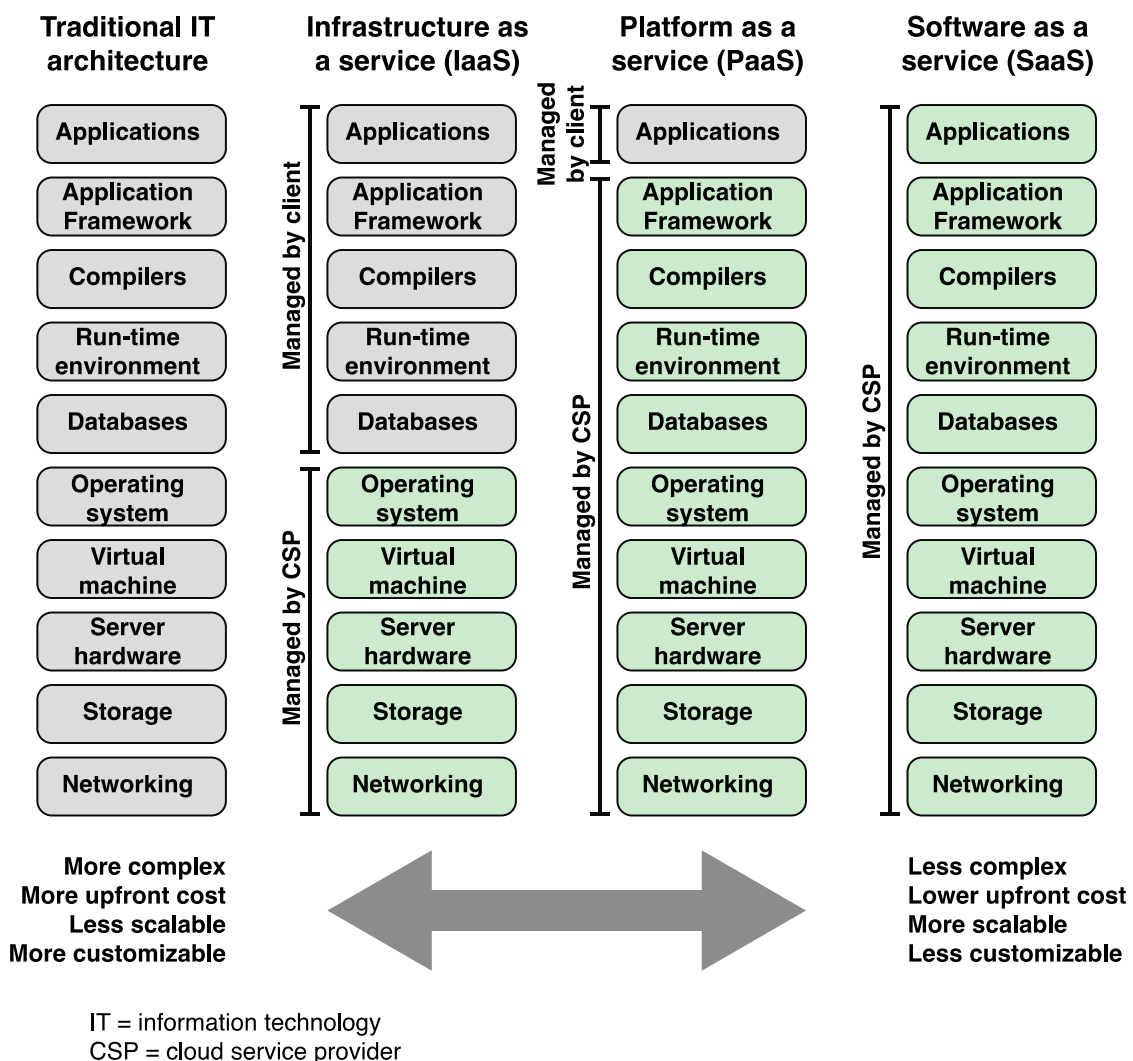


Figure 1.17 Alternative Information Technology Architectures

SOFTWARE AS A SERVICE Phần mềm dưới dạng dịch vụ (SaaS)

- Cung cấp dịch vụ cho khách hàng dưới dạng phần mềm (phần mềm ứng dụng) chạy trên đó và có thể truy cập được trên đám mây.
- Tuân theo mô hình dịch vụ Web quen thuộc được áp dụng cho tài nguyên đám mây.
- Cho phép khách hàng sử dụng các ứng dụng của nhà cung cấp đám mây chạy trên cơ sở hạ tầng đám mây của nhà cung cấp.

Các ứng dụng có thể truy cập được từ nhiều thiết bị khác nhau thông qua một giao diện đơn giản như trình duyệt Web.

- Giảm bớt sự phức tạp của việc cài đặt, bảo trì, nâng cấp và vá lỗi phần mềm.
- Ví dụ: **Gmail, và Salesforce.com.**
- Những người đăng ký phổ biến là các tổ chức muốn cung cấp cho nhân viên của họ quyền truy cập vào phần mềm văn phòng, chẳng hạn như quản lý tài liệu và email.
- Các cá nhân cũng thường sử dụng mô hình SaaS để có được tài nguyên đám mây.
- Nhà cung cấp đám mây cũng thường cung cấp các tính năng liên quan đến dữ liệu như sao lưu tự động và chia sẻ dữ liệu giữa những người đăng ký.

PLATFORM AS A SERVICE Nền tảng như một dịch vụ (PaaS)

- Cung cấp dịch vụ cho khách hàng dưới dạng một nền tảng mà trên đó các ứng dụng của khách hàng có thể chạy, các ứng dụng đó do khách hàng tạo hoặc mua.
- Cung cấp các khối xây dựng phần mềm hữu ích, cùng với một số công cụ phát triển, chẳng hạn như ngôn ngữ lập trình, môi trường thời gian chạy và các công cụ khác hỗ trợ triển khai các ứng dụng mới.

- Trên thực tế, **PaaS là một hệ điều hành trên đám mây**.
- Hữu ích cho một tổ chức muốn **phát triển các ứng dụng mới** hoặc phù hợp với nhu cầu trong khi chỉ trả tiền cho các tài nguyên máy tính cần thiết khi cần thiết và chỉ trong thời gian cần thiết.
- **Google App Engine và Nền tảng Salesforce1** từ Salesforce.com là những ví dụ về PaaS.

INFRASTRUCTURE AS A SERVICE Cơ sở hạ tầng dưới dạng dịch vụ (IaaS)

- IaaS cung cấp các **máy ảo, phần cứng và hệ điều hành trừu tượng** khác, có thể được điều khiển **thông qua giao diện lập trình ứng dụng dịch vụ (application programming interface - API)**.
- Cung cấp khả năng **xử lý, lưu trữ, mạng và các tài nguyên điện toán cơ bản** khác để khách hàng có thể triển khai và chạy phần mềm tùy ý, có thể bao gồm các hệ điều hành và ứng dụng.
- Cho phép **kết hợp các dịch vụ điện toán cơ bản**, chẳng hạn như xử lý số và lưu trữ dữ liệu, để xây dựng các hệ thống máy tính có khả năng thích ứng cao.
- Ví dụ về IaaS là **Amazon Elastic Computer Cloud (Amazon EC2) và Windows Azure**.

PHỤ LỤC

- **desktop computer**: A computer designed for use by an individual, usually incorporating a graphics display, a keyboard, and a mouse.
- **server** A computer used for **running larger programs for multiple users**, often simultaneously, and typically **accessed only via a network**.
- **supercomputer** A class of computers with the **highest performance and cost**; they are configured as servers and typically cost millions of dollars.
- **terabyte** Originally **1,099,511,627,776 (2^{40}) bytes**, although some communications and secondary storage systems have redefined it to mean **1,000,000,000,000 (10^{12}) bytes**.
- **petabyte** Depending on the situation, either **1000 or 1024 terabytes**.
- **gigabyte** Traditionally **1,073,741,824 (2^{30}) bytes**, although some communications and secondary storage systems have redefined it to mean **1,000,000,000 (10^9) bytes**. Similarly, depending on the context, megabyte is either 2^{20} or 10^6 bytes.
- **datacenter** A room or building designed to **handle the power, cooling, and networking needs** of a large number of servers.
- **acronym** A word constructed by **taking the initial letters of a string of words**. For example: **RAM** is an acronym for Random Access Memory, and **CPU** is an acronym for Central Processing Unit.

Hiệu suất của một chương trình phụ thuộc vào sự kết hợp giữa hiệu quả của các thuật toán được sử dụng trong chương trình, hệ thống phần mềm được sử dụng để tạo và dịch chương trình thành các lệnh máy và hiệu quả của máy tính trong việc thực hiện các lệnh đó, có thể bao gồm các hoạt động ra/vào (I/O).

Hardware or software component	How this component affects performance
Algorithm	Determines both the number of source-level statements and the number of I/O operations executed
Programming language, compiler, and architecture	Determines the number of computer instructions for each source-level statement
Processor and memory system	Determines how fast instructions can be executed
I/O system (hardware and operating system)	Determines how fast I/O operations may be executed

Hình bên dưới cho thấy các lớp phần mềm này được tổ chức chủ yếu theo kiểu phân cấp, với các ứng dụng là vòng ngoài cùng và nhiều phần mềm hệ thống nằm giữa phần cứng và phần mềm ứng dụng.

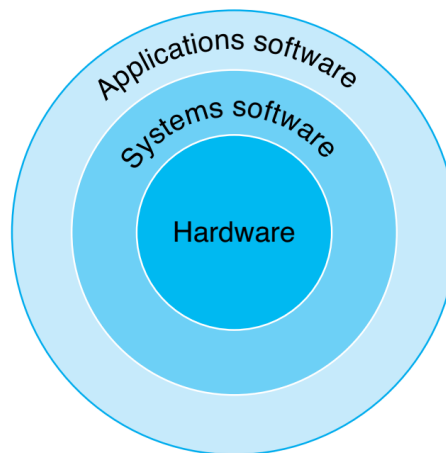


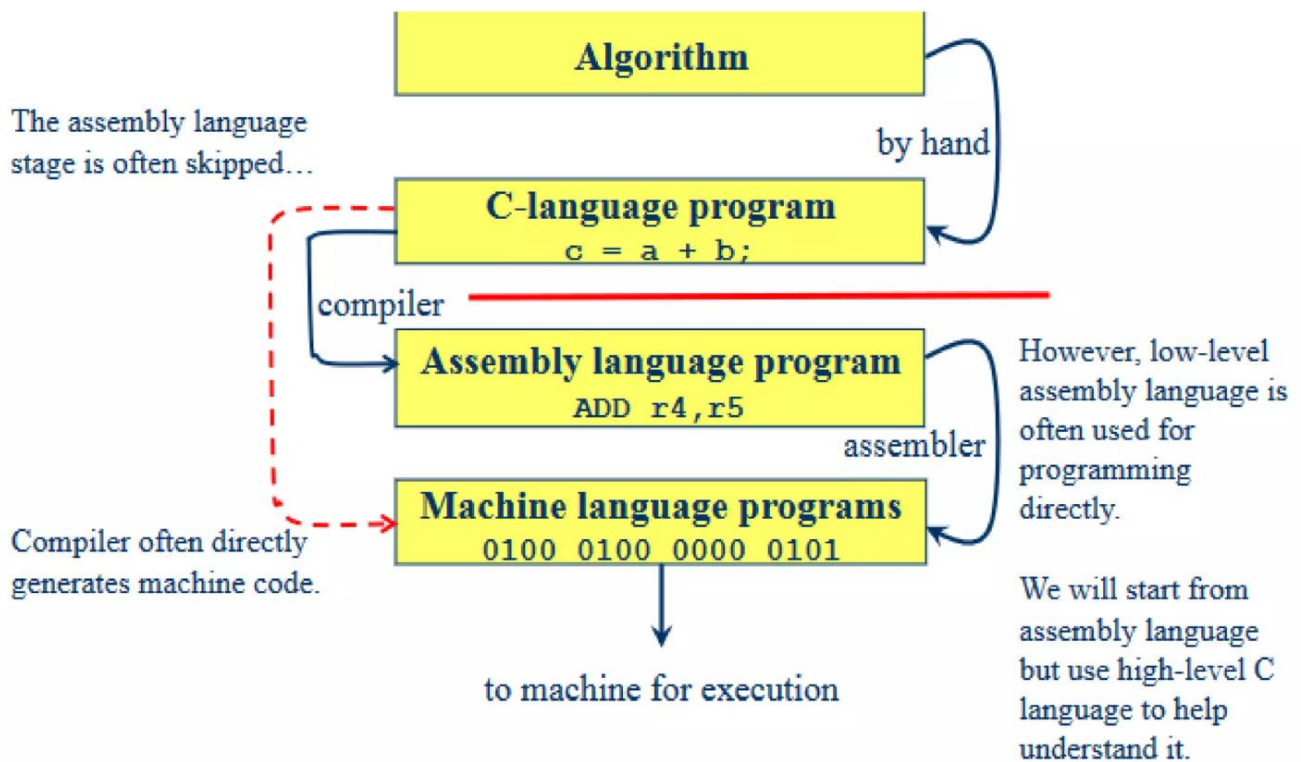
FIGURE 1.2 A simplified view of hardware and software as hierarchical layers, shown as concentric circles with hardware in the center and applications software outermost.

Có nhiều loại phần mềm hệ thống, nhưng có hai loại phần mềm hệ thống đóng vai trò trung tâm trong mọi hệ thống máy tính ngày nay: **hệ điều hành và trình biên dịch**.

Một hệ điều hành **giao tiếp giữa chương trình của người dùng và phần cứng** và cung cấp nhiều dịch vụ và chức năng giám sát khác nhau. Trong số các chức năng quan trọng nhất là

- Xử lý các thao tác đầu vào và đầu ra cơ bản
- Phân bổ lưu trữ và bộ nhớ
- Cung cấp khả năng chia sẻ máy tính được bảo vệ giữa nhiều ứng dụng sử dụng đồng thời.

Trình biên dịch thực hiện một chức năng quan trọng khác: **dịch một chương trình được viết bằng ngôn ngữ cấp cao**, chẳng hạn như C, C++, Java hoặc Visual Basic **thành các lệnh mà phần cứng có thể thực thi**. Do sự phức tạp của các ngôn ngữ lập trình hiện đại và sự đơn giản của các lệnh được thực thi bởi phần cứng, việc dịch từ một chương trình ngôn ngữ cấp cao sang các lệnh phần cứng rất phức tạp.



- **liquid crystal display (LCD)** Công nghệ hiển thị sử dụng một lớp polyme lỏng mỏng có thể được sử dụng để truyền hoặc chặn ánh sáng tùy theo lượng điện tích được áp dụng.
- **active matrix display** Màn hình tinh thể lỏng sử dụng bóng bán dẫn để điều khiển việc truyền ánh sáng ở từng pixel riêng lẻ.
- **pixel** The **smallest individual picture element**. Screens are composed of hundreds of thousands to millions of pixels, organized in a matrix.

The image is composed of a matrix of picture elements, or **pixels**, which can be represented as a matrix of bits, called a *bit map*. Depending on the size of the screen and the resolution, the display matrix ranges in size from 640×480 to 2560×1600 pixels in 2008.

A color display might use **8 bits for each of the three colors (red, blue, and green), for 24 bits per pixel**, permitting millions of different colors to be displayed.

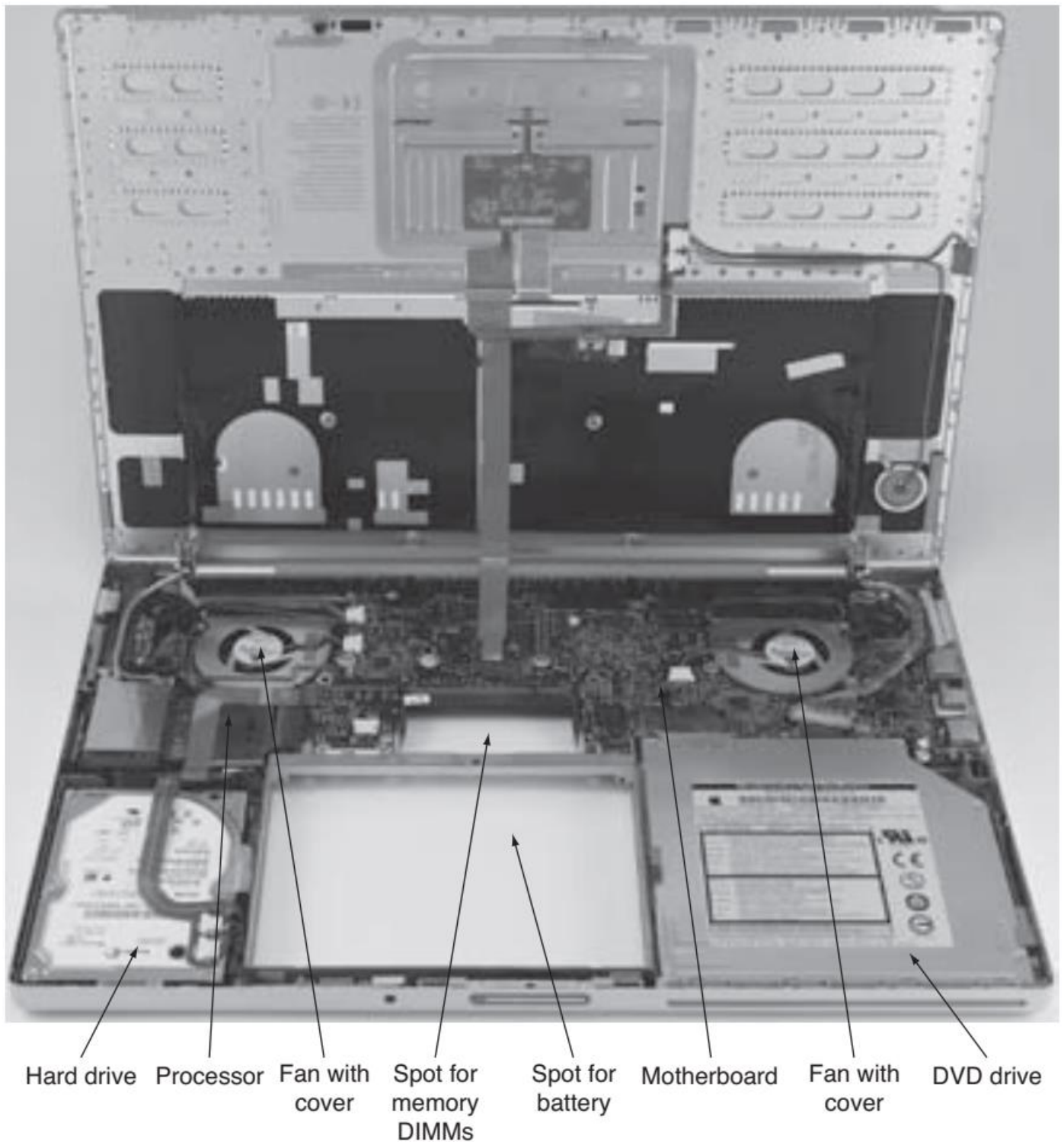


FIGURE 1.7 Inside the laptop computer of Figure 1.5.

- **dynamic random access memory (DRAM)** Memory built as an integrated circuit; it provides random access to any location.
- **dual inline memory module (DIMM)** A small board that contains DRAM chips on both sides. (SIMMs have DRAMs on only one side.)
- **static random access memory (SRAM)** Also memory built as an integrated circuit, but faster and less dense (it is *ít ì hơn*) than DRAM.
- **datapath** The component of the processor that performs arithmetic operations

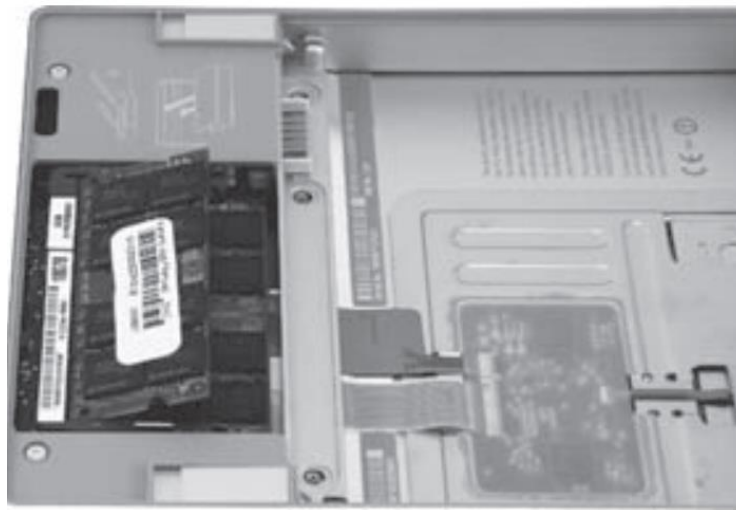
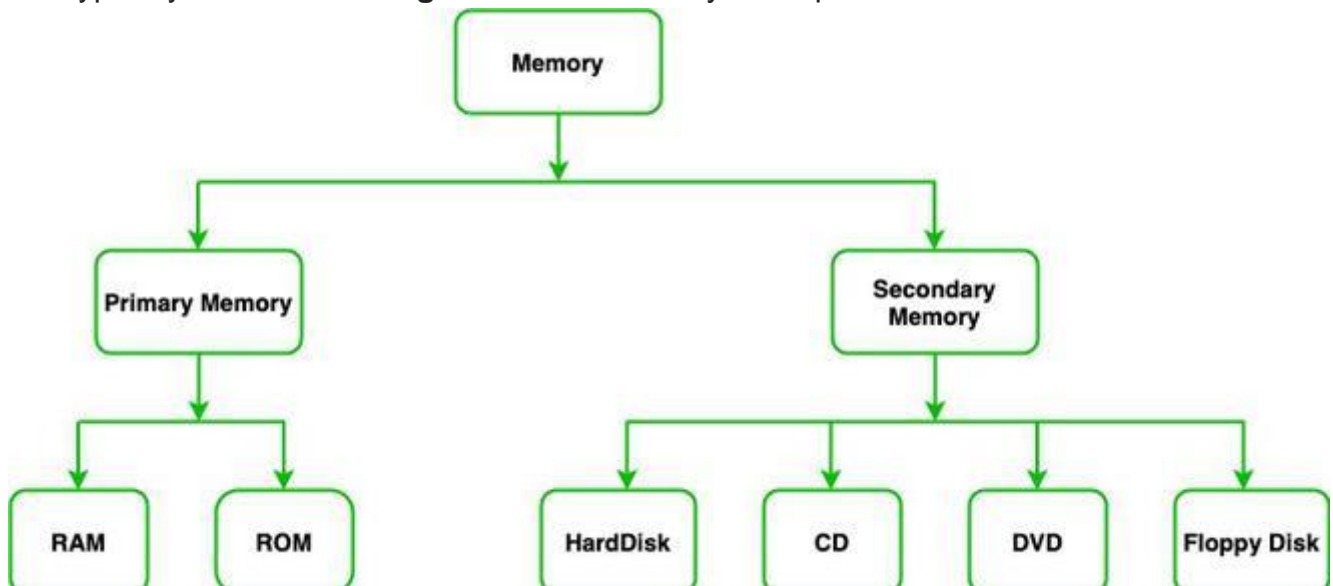


FIGURE 1.8 Cận cảnh mặt dưới của laptop cho thấy bộ nhớ.

The main memory is contained on one or more small boards shown on the left. The hole for the battery is to the right. The DRAM chips are mounted on these boards (called DIMMs, for dual inline memory modules) and then plugged into the connectors.

- **volatile memory (Bộ nhớ dễ bay hơi)** Storage, such as DRAM, that **retains (lưu giữ) data only if it is receiving power.**
 - là bộ nhớ máy tính cần năng lượng để duy trì thông tin được lưu trữ; nó vẫn giữ lại nội dung của nó khi bật nguồn nhưng khi **nguồn điện bị ngắt, dữ liệu được lưu trữ sẽ nhanh chóng bị mất**
- **nonvolatile memory** Một dạng bộ nhớ **lưu giữ dữ liệu ngay cả khi không có nguồn điện** và được sử dụng để **lưu trữ các chương trình giữa các lần chạy**. Ổ đĩa từ là một ví dụ.
- **main memory** Also called **primary memory**. Memory used to **hold programs while they are running**; typically **consists of DRAM** in today's computers.
- **secondary memory** **Nonvolatile memory** used to **store programs and data** between runs; typically **consists of magnetic disks** in today's computers.



- **magnetic disk** Also called **hard disk**. A form of **nonvolatile** secondary memory, bao gồm các đĩa quay được phủ một vật liệu ghi từ tính.
- **flash memory** A **nonvolatile** semiconductor memory. It is **cheaper and slower than DRAM** but more expensive and faster than magnetic disks.
 - một loại bộ nhớ máy tính kiểu bộ nhớ điện tĩnh, có thể bị xóa và lập trình lại.

<https://www.youtube.com/playlist?list=PLPsodm1vdWCWDQ9P7f-zBcM2Eiruh41tj>

<https://www.youtube.com/playlist?list=PLZTZImDGs5YcfMtlONvLtW090fqMMOzYt>