

## CHƯƠNG I : GIỚI THIỆU TỔNG QUÁT VỀ HỆ VI XỬ LÝ

### 1.1. Kiến trúc của hệ Vi xử lý

Kiến trúc hệ Vi xử lý là một thuật ngữ dùng để chỉ rõ những đặc trưng của hệ vi xử lý trong đó bao gồm có cấu trúc phần cứng và tổ chức phần mềm được cài đặt trong hệ. Một hệ thống vi xử lý hay gọi ngắn gọn hơn là hệ vi xử lý thường bao gồm các thành phần cơ bản như:

- Bộ xử lý trung tâm CPU (Central Processing Unit ) là trung tâm đầu não của hệ
- Bộ nhớ tâm bao gồm 2 thành phần là ROM và RAM
- Thiết bị vào/ra dữ liệu
- Các kênh thông tin hay Bus hệ thống...

Tất cả các thiết bị có các chức năng như vậy đều được gọi là một hệ vi xử lý. Trong thực tế có rất nhiều hãng chế tạo các bộ vi xử lý cho các máy vi tính như: IBM, Intel, Cyrix, AMD, Motorola.... nhưng thông dụng nhất vẫn là bộ vi xử lý của Intel. Các bộ vi xử lý của Intel được phát triển qua các thời kỳ như sau:

Năm 1971, Intel đưa ra bộ vi xử lý 4004 với 4 bit dữ liệu, 12 bit địa chỉ; 0,8MHz

Năm 1972, bộ vi xử lý Intel 8080 ra đời với 8bit dữ liệu, 12 bit địa chỉ;

Năm 1978, bộ vi xử lý Intel 8086 ra đời với 16bit dữ liệu, 20 bit địa chỉ; tốc độ 10MHz

Năm 1979, bộ vi xử lý Intel 8088 ra đời nhưng vẫn tương thích với hệ thống 8086

Năm 1982 bộ vi xử lý 80286 ra đời với 16bit dữ liệu, 20 bit địa chỉ, tốc độ 20MHz

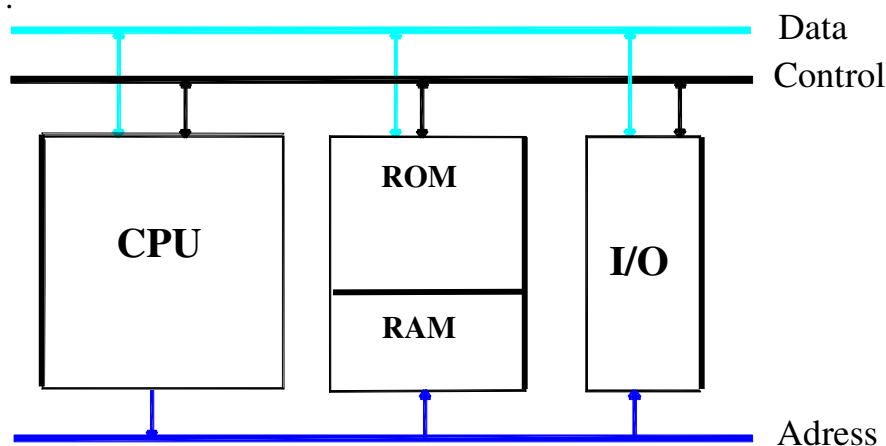
Năm 1985-1988, bộ vi xử lý 80386 ra đời với 32 bit dữ liệu và 32 bit địa chỉ ...

Năm 1989, bộ vi xử lý 80486 ra đời với 32 bit dữ liệu và 32 bit địa chỉ tốc độ đến 60M

Năm 1993, bộ vi xử lý Pentium ra đời với 64 bit dữ liệu, tốc độ xử lý 100MHz sau đó là các bộ vi xử lý Pentium Pro, Pentium II, Pentium III, Celeron, Pentium 4...

#### 1.1.1. Sơ đồ khối tổng quát của hệ Vi xử lý

❖ Sơ đồ khối :



**Hình 1.1** Cấu trúc các kênh thông tin trong hệ VXL

❖ Chức năng các khối :

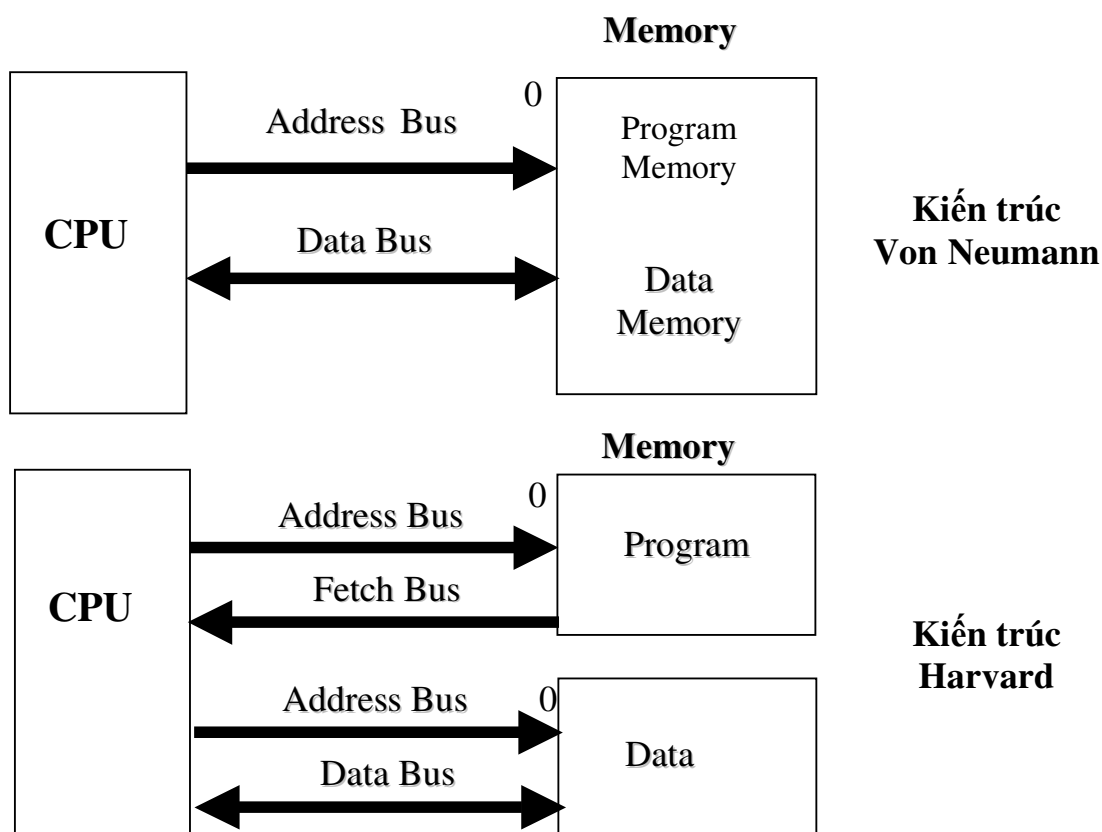
Bộ xử lý trung tâm CPU (Central Processing Unit ) là trung tâm đầu não của hệ, nó có chức năng thu thập, xử lý thông tin và điều khiển mọi hoạt động của hệ vi xử lý.

Bộ nhớ trung tâm có nhiệm vụ lưu trữ thông tin dữ liệu trước khi CPU xử lý

Thiết bị I/O thực hiện việc nhận dữ liệu từ các kênh thông tin từ bên ngoài vào để cho CPU xử lý và xuất ra các tín hiệu điều khiển hệ thống

Các kênh thông tin hay Bus hệ thống là cầu nối liên kết giữa các khối trong đó thực hiện 3 việc chính là liên kết các Bus địa chỉ, Bus điều khiển và Bus dữ liệu.

❖ Tổ chức các kênh thông tin trong hệ Vi xử lý



**Hình 1.2: Cấu trúc các kênh thông tin trong hệ VXL**

Trên đây là 2 cách tổ chức bộ nhớ theo kiểu Von Neumann và Harvard. Với kiểu tổ chức bộ nhớ chương trình và dữ liệu tách biệt cho phép tốc độ truy xuất thông tin nhanh hơn đáng kể. Các kênh dữ liệu đều là kênh song song và dùng chung cho tất cả các bộ nhớ, tuy nhiên nó phải được kiểm soát thông qua các cổng logic 3 trạng thái. Cổng này có nhiệm vụ tạo ra trạng thái đặc biệt khi có những thành phần không được kích hoạt làm việc, trạng thái đặc biệt này sẽ cách ly về mặt tín hiệu giữa kênh thông tin với từng thành phần trong hệ mặc dù chúng vẫn được kết nối với nhau về mặt vật lý.

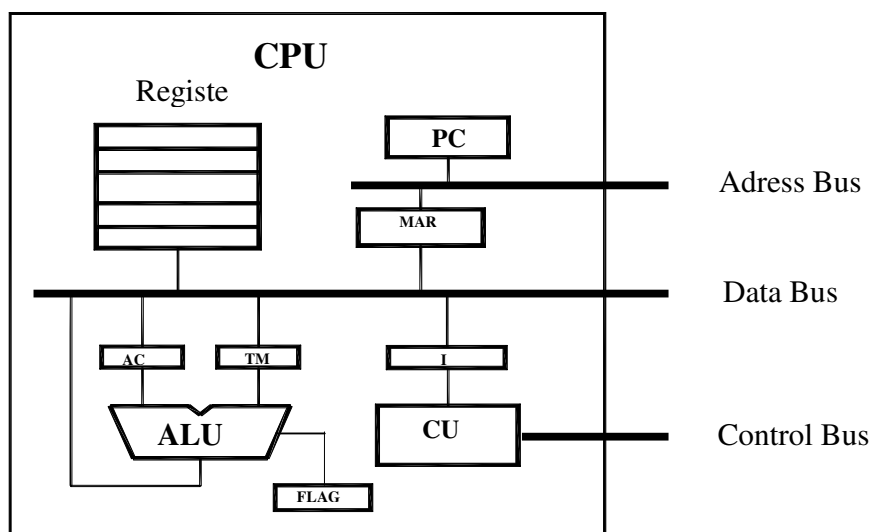
### 1.1.2. Khối xử lý trung tâm (Central Processing Unit)

Bộ vi xử lý là hạt nhân của hệ vi xử lý, nó là thành phần quan trọng nhất kiểm soát tất cả các hoạt động của hệ và thực hiện các thao tác trên dữ liệu. Hầu hết các CPU được hình thành từ các mạch logic nhằm thực hiện liên tục 2 thao tác là tìm nạp lệnh từ bộ nhớ để giải mã và thực thi lệnh. CPU có khả năng hiểu và thực thi các lệnh dựa trên một tập các mã nhị phân

gọi là mã máy trong đó mỗi mã nhằm thực hiện một thao tác nào đó. Tổ hợp các lệnh cho mỗi loại CPU gọi là tập lệnh và nó thường chia ra thành các nhóm lệnh như:

- Nhóm lệnh chuyển dữ liệu
- Nhóm lệnh số học
- Nhóm lệnh Logic
- Nhóm lệnh rẽ nhánh chương trình
- Nhóm lệnh xử lý bit

Cấu trúc đơn giản của một loại CPU được minh họa như sau:

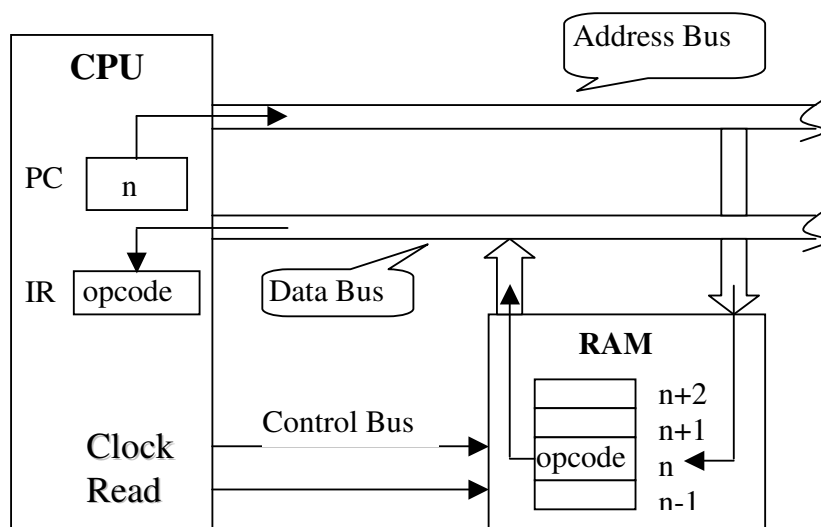


**Hình 1.3** Cấu trúc chung của một bộ CPU

- PC (Program Counter): Bộ đếm chương trình có vai trò như một con trỏ, trỏ đến ô nhớ chứa lệnh mà CPU cần truy nhập
- IR (Instruction Register): Thanh ghi lệnh IR (Instruction Register) Thanh ghi lệnh thực hiện chức năng chứa lệnh mà CPU đang thực hiện.
- CU (Control Unit) Đơn vị điều khiển có chức năng giải mã lệnh.
- MAR (Memory Address Register) Thanh ghi chỉ bộ nhớ thực hiện chức năng chứa địa chỉ của ô nhớ hiện thời mà CPU đang truy nhập.
- ALU (Arithmetic Logic Unit) đơn vị số học logic, thực hiện các phép tính số học, logic và các phép xử lý dữ liệu khác.
- ACC (Accumulator) Thanh chứa , chứa toán hạng của một phép tính hoặc kết quả của phép tính.
- TMP (Temporary) Thanh ghi tạm, chứa toán hạng thứ hai của phép tính.
- FLAGS Thanh ghi cờ chứa thông tin về trạng thái kết quả phép tính sau khi thực hiện lệnh.
- Address Bus : Bus địa chỉ
- Data Bus : Bus dữ liệu
- Control Bus : Bus điều khiển

### 1.1.3. Quá trình truy xuất và xử lý thông tin

Việc tìm nạp một lệnh từ Ram hệ thống là một trong những thao tác cơ bản nhất mà CPU phải thực hiện. Quá trình tìm nạp lệnh được thực hiện theo trình tự như sau:



**Hình1.4 : Hoạt động của Bus cho chu kỳ tìm nạp lệnh**

Địa chỉ đang chứa trong PC sẽ được gửi lên trên bus địa chỉ.

Tín hiệu cho phép đọc lệnh từ bộ nhớ sẽ được kích hoạt sang trạng thái tích cực

Dữ liệu hay mã lệnh sẽ được đọc từ bộ nhớ và gửi lên kênh dữ liệu rồi chuyển về thanh ghi lệnh IR. Tiếp theo đó là nội dung của PC (tức địa chỉ) sẽ được tăng lên 1 để trở tới địa chỉ kế tiếp trong bộ nhớ.

Mã lệnh sẽ được chuyển xuống bộ giải mã và căn cứ theo mã lệnh CPU sẽ triển khai thực hiện lệnh. Trường hợp toán hạng nằm trong bộ nhớ chương trình ngay sau mã lệnh. CPU tiếp tục đưa PC lên bus địa chỉ để trở tới toán hạng rồi đưa ra thanh ghi để thực hiện lệnh.

VD: ADD A,#3EH

Nếu toán hạng nằm ngay trong thanh ghi của CPU, khi đó CPU thực hiện lệnh ngay

VD : ADD A,R1

Toán hạng nằm trong bộ nhớ mà địa chỉ của nó trong 1 thanh ghi của CPU .CPU đưa địa chỉ cho thanh ghi địa chỉ (MAR) để đọc dữ liệu và thực hiện lệnh. VD : ADD A,@R0

### 1.1.4. Bộ nhớ chỉ đọc (*Read Only Memory - ROM*):

❖ *ROM cơ bản:*

ROM dùng để lưu trữ chương trình điều hành (Monitor) của hệ VXL. Chương trình này sẽ quy định mọi hoạt động của hệ VXL. Bộ VXL sẽ căn cứ vào các lệnh chứa trong chương trình để điều khiển hệ VXL thực hiện các chức năng, nhiệm vụ được ấn định trong lệnh. Nói cách khác, hệ VXL sẽ thực hiện một cách trung thực thuật toán mà người thiết kế phần mềm đã xây dựng và cài đặt vào ROM của hệ.

Ngoài ra, ROM trong hệ VXL còn dùng để lưu trữ các bảng biểu, tham số của hệ thống mà trong quá trình hoạt động không được thay đổi như: bảng địa chỉ cổng giao tiếp, các bảng tra cứu số liệu, các bộ mã cần sử dụng trong hệ.

ROM cũng được quản lý theo phương thức ma trận điểm, nó có nhiều chủng loại khác nhau: ROM, PROM, EPROM, EEPROM,...

ROM là bộ nhớ cố định có cấu trúc đơn giản nhất. Nội dung của nó do nhà sản xuất chế tạo, người sử dụng không thể thay đổi nội dung này được nữa.

❖ *PROM (Programmable ROM - ROM có khả năng lập trình được):*

**Đặc điểm chung:** Nội dung của PROM do nhà sản xuất hoặc người thiết kế hệ VXL nạp vào nhưng chỉ được 1 lần. Sau khi nạp xong nội dung này không thể thay đổi được nữa.

❖ *EPROM (Eraseable PROM ROM nạp/xoá được nhiều lần):*

**EPROM** là bộ nhớ cố định có cấu trúc đặc biệt. Nội dung của nó do nhà sản xuất hay người thiết kế hệ VXL nạp vào và có thể nạp/xoá nhiều lần. Người ta tạo ra 1 bit thông tin trong EPROM dựa trên nguyên tắc làm việc của Transistor trường có cực cửa cách ly kênh cảm ứng (MOSFET kênh cảm ứng).

❖ *EEPROM (Electrical EPROM ROM có khả năng lập trình và xoá được bằng điện).*

### 1.1.5. Bộ nhớ W/R còn gọi là bộ nhớ truy cập ngẫu nhiên (RAM)

RAM là bộ nhớ có thể ghi và đọc được, thông tin trên RAM sẽ bị mất khi mất nguồn cung cấp. Theo phương thức lưu trữ thông tin, RAM được chia thành 2 loại cơ bản: RAM tĩnh và RAM động.

**RAM tĩnh:** Có thể lưu trữ thông tin lâu tùy ý miễn là được cung cấp điện năng - tất cả các loại phân tử nhớ bằng Trigon đều thuộc loại này.

**RAM động:** Chỉ lưu được thông tin trong 1 khoảng thời gian nhất định. Muốn kéo dài thời gian này cần có phương thức làm tươi lại thông tin trong phân tử nhớ RAM. Phân tử nhớ của RAM động đơn giản nhất là một linh kiện điện dung - tụ điện. Sử dụng RAM động có phức tạp nhưng về cấu trúc nhớ lại đơn giản, tiêu tốn ít năng lượng, tăng mật độ bộ nhớ và đôi khi còn làm tăng cả tốc độ làm việc của bộ nhớ.

Cấu trúc mạch điện của các bộ nhớ RAM rất đa dạng cả về công nghệ chế tạo chúng (TTL, MOS,...) và các yêu cầu sử dụng chúng như các yêu cầu về ghép nối, tốc độ làm việc, mật độ linh kiện và dung lượng cần thiết...

### 1.1.6. Các thiết bị xuất/nhập:

Các thiết bị xuất/nhập hay các thiết bị ngoại vi kết hợp với các mạch giao tiếp (Interface) sẽ tạo ra các đường truyền thông giữa hệ VXL với thế giới bên ngoài. Tuy nhiên để trao đổi thông tin giữa hệ VXL với các thiết bị ngoại vi, cần có các phương pháp điều khiển thích hợp như: Điều khiển vào/ra bằng chương trình, điều khiển vào/ra bằng ngắt, điều khiển vào/ra bằng phần cứng vv...

### 1.1.7. Cấu trúc kênh chung của hệ VXL:

Kênh (Bus) là tập hợp các đường thông tin có cùng mục đích. Để CPU có thể giao tiếp được với các bộ phận khác trong hệ VXL theo yêu cầu, mỗi hệ VXL cần sử dụng 3 kênh như :

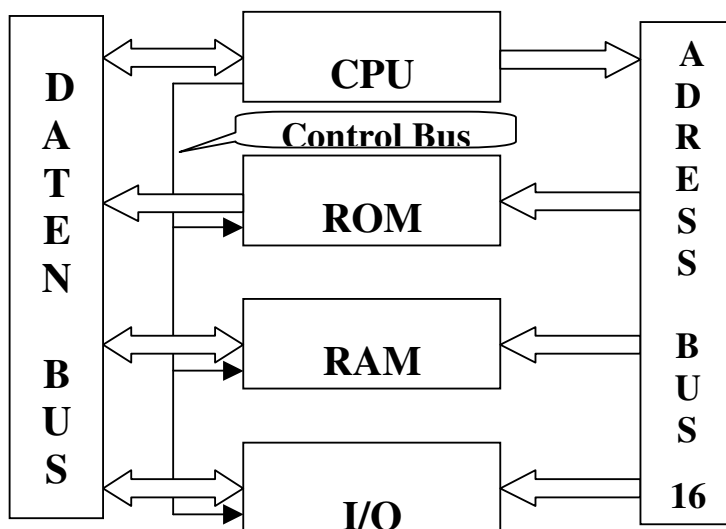
- Kênh địa chỉ (Address Bus).
- Kênh dữ liệu (Daten Bus).
- Kênh điều khiển (Control Bus).

Để thực hiện thao tác đọc hoặc ghi, CPU xác định rõ vị trí (địa chỉ) của dữ liệu (hoặc lệnh) bằng cách đặt địa chỉ đó lên kênh địa chỉ, sau đó kích hoạt tín hiệu **Read** hoặc **Write** trên kênh điều khiển để chỉ ra thao tác là **đọc** hay **ghi**.

Nếu kích hoạt tín hiệu điều khiển Read, thao tác đọc lấy 1 byte dữ liệu từ bộ nhớ ở vị trí đã xác định và đặt byte này lên kênh dữ liệu. CPU sẽ đọc dữ liệu và cất dữ liệu vào 1 trong các thanh ghi nội của CPU.

Nếu kích hoạt tín hiệu điều khiển Write, CPU sẽ thực hiện thao tác ghi bằng cách xuất dữ liệu lên kênh dữ liệu. Nhờ vào tín hiệu điều khiển, bộ nhớ nhận biết được đây là thao tác ghi và lưu dữ liệu vào vị trí đã được xác định.

Kênh dữ liệu cho phép trao đổi thông tin giữa CPU và bộ nhớ, cũng như giữa CPU với thiết bị ngoại vi. Thông thường các hệ VXL dành hầu hết thời gian cho việc di chuyển dữ liệu, đa số các thao tác di chuyển dữ liệu xảy ra giữa 1 thanh ghi của CPU với ROM và RAM ngoài. Do đó độ lớn của kênh dữ liệu ảnh hưởng rất lớn tới hiệu suất của hệ VXL. Nếu bộ nhớ của hệ thống rất lớn và CPU có khả năng tính toán cao, nhưng việc truy xuất dữ liệu – di chuyển dữ liệu giữa bộ nhớ và CPU thông qua kênh dữ liệu lại bị nghẽn thì hiện tượng “nghẽn cổ chai” này chính là hậu quả của độ rộng kênh dữ liệu không đủ lớn. Để khắc phục hiện tượng này, cần tăng đường tín hiệu cho kênh dữ liệu.



**Hình 1.5 : Cấu trúc kênh chung của hệ thống VXL**

Như ở hình 1.3, kênh dữ liệu là kênh 2 chiều, còn kênh địa chỉ là kênh 1 chiều. Các thông tin về địa chỉ luôn được cung cấp bởi CPU, trong khi các dữ liệu di chuyển theo cả 2 hướng tùy thuộc vào thao tác thực hiện là đọc hay ghi. Thuật ngữ “**dữ liệu**” được sử dụng theo nghĩa tổng quát: “**thông tin**” di chuyển trên kênh dữ liệu có thể là **lệnh** của chương trình, **địa chỉ** theo sau lệnh hoặc **dữ liệu** được sử dụng bởi chương trình.

Kênh điều khiển là tập hợp các tín hiệu, mỗi tín hiệu có một vai trò riêng trong việc điều khiển có trật tự hoạt động của hệ thống. Các tín hiệu điều khiển được cung cấp bởi CPU để đồng bộ việc di chuyển thông tin trên các kênh địa chỉ và dữ liệu. Các bộ VXL thường có 3 tín hiệu điều khiển: **Read**, **Write**, **Clock**. Tuy nhiên tùy vào yêu cầu cụ thể cũng như cấu trúc phần cứng của từng hệ VXL mà số lượng tín hiệu điều khiển có thể khác nhau.

## 1.2. Các hệ thống số liên quan VĐK

❖ Hệ đếm thập phân (Decimal): Hệ đếm thập phân còn gọi là hệ đếm cơ số mười và nó được biểu diễn bởi 10 con số từ (0,1,2,3,4,5,6,7,8,9) những con số này được sử dụng rất nhiều trong khoa học kỹ thuật cũng như trong đời sống hàng ngày, khi biểu diễn số thập phân thì đứng sau dãy số thường có chữ D.

Ví dụ: Ba nghìn Chín trăm Bảy mươi Tám được biểu diễn như sau

$$\begin{aligned} 3978 &= 3 \times 10^3 + 9 \times 10^2 + 7 \times 10^1 + 8 \times 10^0 \\ &= 3000 + 900 + 70 + 8 \end{aligned}$$

❖ Hệ đếm thập lục phân (Hexadecimal): Hệ đếm thập lục phân còn gọi là hệ đếm cơ số mười sáu và nó được biểu diễn bởi 16 ký số (0,1,2,3,4,5,6,7,8,9,B,C,D,E,F) những con số này được sử dụng rất nhiều trong khoa học kỹ thuật đặc biệt là khoa học máy tính vì biểu diễn mã Hexa rất ngắn gọn, khi biểu diễn số thập lục phân thì sau dãy số phải có chữ H.

Ví dụ: 3978h, 12CCh, 1998h, ABCDh, 2008h ...

❖ Hệ đếm nhị phân (Binary): Hệ đếm nhị phân còn gọi là hệ đếm cơ số hai và nó được biểu diễn bởi 2 con số là 0 và 1, trong kỹ thuật điện tử số thì số 0 gọi là mức logic thấp ứng với điện áp thấp, số 1 gọi là mức logic cao tương ứng với điện áp cao nhất. Mỗi ký hiệu 0 hoặc 1 được gọi là 1 Bit (Binary Digit), khi biểu diễn số nhị phân thì đứng sau dãy số phải có chữ B.

Ví dụ:

1100b ; gọi là 1 nibble  
10011001b ; gọi là 1 Byte  
1010101111001101b ; gọi là 1 Word

Trong dãy số nhị phân được biểu diễn thì số nhị phân sát phải gọi là bit LSB còn số nhị phân sát trái gọi là bit MSB

Ví dụ: 1010101010101010

MSB                      LSB

Số nhị phân thường được biểu diễn ở 2 dạng là số nhị phân có dấu và số nhị phân không dấu, nếu số nhị phân không dấu sẽ chỉ biểu diễn các số không âm ( $\geq 0$ ) còn số nhị phân có dấu thì biểu diễn được cả giá trị âm

$$\text{Ví dụ : } (1101) = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$$

Dải giá trị của các số có dấu 8 bit là [-128, +127]

Dải giá trị của các số có dấu 16 bit là [-32768, +32767]

Trong khi biểu diễn dãy số nhị phân có dấu thì người ta sử dụng bit MSB để quy ước cho bit dấu, với bit 0 cho dãy số nhị phân dương còn bit 1 cho dãy số nhị phân âm.

❖ Hệ đếm BCD (Binary Coded Decimal): Số đếm BCD được định nghĩa là số thập phân nhưng được biểu diễn dưới dạng nhị phân 4 bit, nhưng dãy số nhị phân 4 bit này khi quy sang hệ thập phân thì giá trị của nó phải  $\leq 9$ . Trong kỹ thuật điện tử nói chung thì mã BCD được sử dụng để giải mã hiển thị LED bảy thanh...

Ví dụ: (0011), (0100), (0101), (0110), ... (1001) ; đây gọi là số BCD không nén

Ví dụ: (00110100), (01010110), ... (01111000) ; đây gọi là số BCD nén

Dưới đây là bảng các mã BCD

Thập phân	BCD	Thập phân	BCD
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

Ví dụ: Cho số thập phân là 15 , biểu diễn dưới dạng số BCD là 00010101

❖ Số bù 2: Trong kỹ thuật Vi xử lý để biểu diễn một con số nào đó dưới dạng dãy số nhị phân thì ngoài việc biểu diễn số không dấu, số có dấu thì người ta còn sử dụng cách biểu diễn số bù 2. Vậy số bù 2 sẽ biểu diễn như thế nào?

Ví dụ: Hãy biểu diễn dãy số A = 10011001 sang số bù 2 của nó:

- b1: Tìm số bù 1 của A(bằng cách lấy bù tất cả các bit của A): 01100110
- b2: Tìm số bù 2 của A (bằng cách lấy số bù 1 cộng cho 1) : 01100111

đến đây ta nhận thấy rằng số bù 2 của một số nào đó nó chính là số đối của nó và tổng = 0

❖ Phép cộng nhị phân không dấu :

Ví dụ : Cho 2 số nhị phân như sau A = 10010011 ; B = 00111001 hãy tìm tổng Y của 2 số nhị phân đã cho

$$\begin{array}{rcl}
 A : & 10010011b & \\
 B : & 00111001b & \\
 \hline
 Y = A+B & = 11001100b & 
 \end{array}
 \begin{array}{l}
 0 + 0 = 0 \\
 0 + 1 = 1 \\
 1 + 0 = 1 \\
 1 + 1 = 0 \quad ; \text{nhớ } 1
 \end{array}$$

❖ Phép trừ nhị phân:

Ví dụ : Cho 2 số nhị phân như sau A = 10010011 ; B = 00111001 hãy tìm hiệu Z của 2 số nhị phân đã cho (Lưu ý rằng phép trừ có thể thực hiện bằng cách biến thành phép cộng)

$$\begin{array}{rcl}
 A : & 10010011b & \\
 B : & 00111001b & \\
 \hline
 Y = A - B & = 01011010b & 
 \end{array}
 \begin{array}{l}
 0 - 0 = 0 \\
 0 - 1 = 1 \quad ; \text{mượn } 1 \\
 1 - 0 = 1 \\
 1 - 1 = 0
 \end{array}$$

❖ Phép nhân nhị phân :

Ví dụ : Cho 2 số nhị phân như sau A = 00100101 ; B = 00000100 hãy tìm tích F

Khi nhân 2 dãy số nhị phân với nhau thì ta đặt phép toán nhân giống như nhân số thập phân, kết quả của phép nhân 2 dãy số nhị phân 8 bit sẽ thu được dãy số nhị phân là 16 bit, như vậy ta có  $F = A * B = 0000000010010100b$



❖ Phép chia nhị phân :

Ví dụ : Cho 2 số nhị phân như sau  $A = 10010110$  ;  $B = 00000100$  hãy tìm thương  
 Khi chia 2 dãy số nhị phân với nhau thì ta đặt phép toán chia cũng giống như khi chia 2 số thập phân, kết quả của phép chia cũng như phần dư (nếu có) thu được tương tự như khi làm phép chia đối với số thập phân, như vậy ta có  $M = A/B = 00100101b$  dư  $0010b$

❖ Chuyển đổi số thập phân sang nhị phân :

Để chuyển đổi số thập phân sang số nhị phân người ta thường dùng phương pháp lấy số thập phân cần chuyển rồi chia 2 liên tiếp đến khi không thể chia được nữa thì dừng

Ví dụ : Chuyển số thập phân 25 sang số nhị phân không dấu

Chia 2	Thương số	Dư số	
25/2 =	12	1	LSB
12/2 =	6	0	
6/2 =	3	0	
3/2 =	1	1	
1/2 =	0	1	MSB

Kết quả thu được là: 11001 với số dư lần thứ nhất là bit có trọng số nhỏ nhất

❖ Chuyển đổi số nhị phân sang thập phân :

Để chuyển đổi số nhị phân sang số thập phân người ta thường dùng phương pháp lấy tổng của tích n các số nhị phân cần chuyển nhân với  $2^0$  đến  $2^{N-1}$  hay theo biểu thức tổng quát như sau:  $A = B_{(N-1)} * 2^{(N-1)} + B_{(N-2)} * 2^{(N-2)} + B_{(N-3)} * 2^{(N-3)} + \dots + B_{(1)} * 2^{(1)} + B_{(0)} * 2^{(0)}$

Ví dụ : Chuyển số nhị phân không dấu 01011110b sang số thập phân

$$A = 0*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0 = 94$$

như vậy ta có  $01011110b = 94$

❖ Chuyển đổi số nhị phân sang Hexa :

Ví dụ : Chuyển số nhị phân 1100101011111110 sang số hexa

Trước hết ta chia số nhị phân đã cho thành các nhóm 4-bit tính từ bit có trọng số nhỏ nhất, sau đó thay thế mỗi nhóm 4-bit bằng ký hiệu hexa tương ứng với nó ta sẽ thu được kết quả như sau:

$$\begin{array}{cccc} 1100 & 1010 & 1111 & 1110 \\ C & A & F & E \end{array} \text{ hay } 1100101011111110b = \text{CAFEh}$$

❖ Chuyển đổi số Hexa sang nhị phân:

Ví dụ : Chuyển số hexa 2F8h và ABBAh sang số nhị phân

Tương tự như trường hợp trên ta sẽ thay thế mỗi ký hiệu hexa bằng 4-bit nhị phân tương ứng với nó và ta sẽ thu được kết quả như sau:

$$\begin{array}{cccc} 2 & F & 8 & \\ 0010 & 1111 & 1000 & \text{ hay } 2F8h = 001011111000b \\ A & B & B & A \\ 1010 & 1011 & 1011 & 1010 \text{ hay ABBAh} = 1010101110111010b \end{array}$$

❖ Mã ASCII: (American Standard Code for Information Interchange )

Quá trình trao đổi thông tin trong máy tính nói chung cũng như quá trình xử lý thông tin của các bộ vi xử lý, tất cả các thông tin đều được biểu diễn dưới dạng các số 0 và 1. Mỗi tổ hợp số 0 hoặc 1 được gán một ký tự chữ cái, chữ số hoặc một ký tự đặc biệt nào đó. Khi thông tin được truyền đi, được lưu giữ trong bộ nhớ và hiển thị trên màn hình đều ở dưới dạng ký tự và tuân theo một loại mã chuẩn được sử dụng rất rộng rãi gọi là mã ASCII.

**BẢNG MÃ ASSII (American Standard Code for Information Interchange)**

Decimal Hex ASCII			Decimal Hex ASCII			Decimal Hex ASCII			Decimal Hex ASCII		
0	0	NUL	32	20		64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	HT	41	29	)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SOH	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	□