

# BÀI GIẢNG KỸ THUẬT VI XỬ LÝ

## \* *Giới thiệu môn học*

- Tên môn học: Vi xử lý
- Số đơn vị học trình: 4 đvht = 60 tiết (40 LT + 20 TH)
- Giáo trình chính: Kỹ thuật vi xử lý, Văn Thế Minh, NXB Giáo dục 1997.
- Tài liệu tham khảo:
  1. Kỹ thuật vi xử lý và lập trình assembly cho hệ vi xử lý, Đỗ Xuân Tiến, NXB KH&KT, 2003
  2. Lập trình hợp ngữ Assembly và máy tính IBM-PC, Quách Tuấn Ngọc, NXB GD, 1998.
  3. Cẩm nang lập trình hệ thống, Nguyễn Mạnh Hùng, NXB GD 1998.
- Những trường hợp sau không được thi:
  - Nghỉ quá 1/4 số tiết (15 tiết)
  - Không hoàn thành các bài tập, các bài kiểm tra.
- Hình thức thi và kiểm tra
  - Kiểm tra trên máy lập trình Assembly (giữa kỳ): hệ số 0,1
  - Thi viết (cuối kỳ): hệ số 0,9

## CHƯƠNG 1 CẤU TRÚC CỦA MỘT HỆ VI XỬ LÝ

### I - CÁC HỆ ĐẾM DÙNG TRONG MÁY TÍNH

#### 1. Hệ mười và hệ hai

##### a) Hệ đếm cơ số 10 (hệ thập phân, decimal number system)

Đây là hệ đếm mà chúng ta sử dụng hàng ngày. Hệ đếm này sử dụng mười chữ số ‘0’..’9’ để biểu diễn các số hệ 10. Chữ số ‘0’ biểu diễn số 0, chữ số ‘1’ biểu diễn số 1,... Khi đếm đến 10 thì hết số nên phải thêm 1 sang hàng bên trái để thành ‘10’ (mười) rồi lại tiếp tục đếm ‘11’, ‘12’, ‘13’,... Một số hệ 10 có giá trị bằng tổng giá trị của từng chữ số nhân với trọng số tương ứng của chữ số. Trọng số của chữ số thứ i là  $10^i$ .

Ví dụ:  $12345,67 = 1 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 7 \times 10^{-2}$

Khi làm việc với nhiều hệ đếm các số hệ 10 có thêm chữ D ở cuối để chỉ ra rằng đó là số hệ 10. Ví dụ: 12,25D

*Tổng quát hoá cho hệ đếm cơ số a bất kỳ ( $a \geq 2, a \in \mathbb{Z}$ ):*

- Sử dụng a chữ số để biểu diễn các số hệ a. Chữ số có giá trị nhỏ nhất là ‘0’, chữ số có giá trị lớn nhất tùy thuộc vào từng hệ đếm nhưng có giá trị là a-1.
- Giá trị (lượng) của một số hệ a bằng tổng giá trị của từng chữ số nhân với trọng số tương ứng của chữ số. Trọng số của chữ số thứ i là  $a^i$ .

##### b) Hệ đếm cơ số 2 (hệ nhị phân, binary number system)

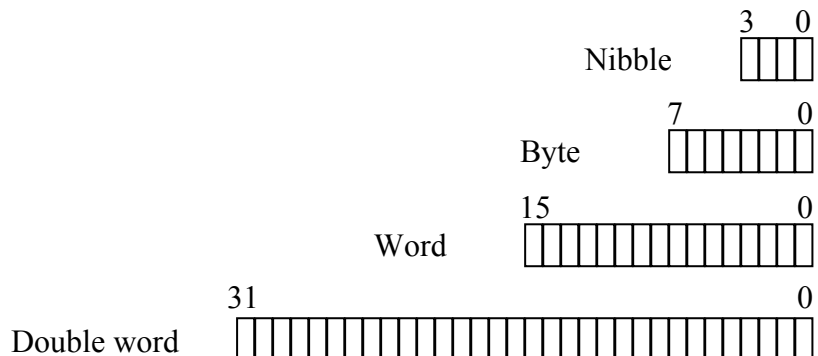
Với cơ số  $a = 2$ , hệ đếm này sử dụng hai chữ số ‘0’ và ‘1’ để biểu diễn các số nhị phân. Chữ số ‘0’ biểu diễn số 0, chữ số ‘1’ biểu diễn số 1. Khi đếm đến 2 thì hết chữ số nên ta phải thêm 1 sang bên trái tạo thành ‘10’,... Một số hệ 2 có giá trị bằng tổng giá trị của từng chữ số nhân với trọng số tương ứng của chữ số. Trọng số của chữ số thứ i trong một số nhị phân là  $2^i$ .

Ví dụ:  $1101,01B = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 13,25$

Các số nhị phân có chữ B ở cuối để phân biệt với các số trong hệ đếm khác khi ta làm việc với nhiều hệ đếm.

*Các linh kiện điện tử cấu tạo nên máy tính chỉ có hai trạng thái: có điện và không có điện. Hai trạng thái này có thể được biểu diễn bằng 1 và 0. Chính vì lý do này mà hệ đếm cơ số 2 là hệ đếm duy nhất được dùng trong máy tính.*

Trong kỹ thuật máy tính, mỗi chữ số nhị phân được gọi là một bit (viết tắt của từ tiếng anh binary digit). Một cụm 4 bit tạo thành một nibble, cụm 8 bit tạo thành 1 byte, cụm 16 bit tạo thành 1 từ (word), cụm 32 bit tạo thành một từ kép (double word). Bit đầu tiên bên trái trong các số nhị phân gọi là bit có trọng số lớn nhất (Most Significant Bit, MSB), còn bit tận cùng bên phải gọi là bit có trọng số nhỏ nhất (Least Significant Bit, LSB). Vị trí của các bit trong nibble, byte, word, double word như sau:



Hệ hai là hệ đếm dùng trong các máy tính. Nhưng con người lại chỉ quen tính toán với hệ 10, bởi vậy để đảm bảo sự giao tiếp thuận tiện giữa người và máy phải có sự chuyển đổi qua lại giữa hai hệ đếm này.

**c) Chuyển đổi giữa hệ mười và hệ 2**

- Đổi số hệ hai sang hệ 10

Muốn đổi một số từ hệ hai sang hệ 10 ta tính tổng trọng số của các chữ số 1. Tổng này chính là số hệ mười cần đổi.

$$\begin{aligned} \text{Ví dụ: } 1101,01\text{B} &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 2^3 + 2^2 + 2^0 + 2^{-2} = 13,25 \end{aligned}$$

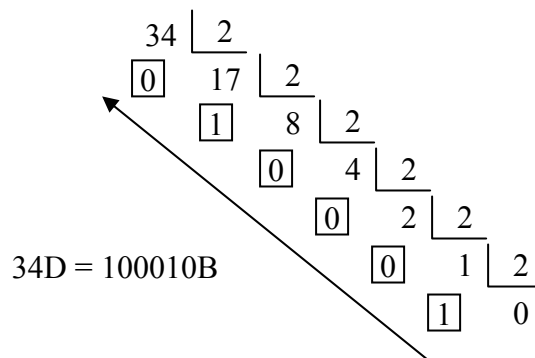
- Đổi số hệ mười sang hệ hai

Trường hợp tổng quát một số hệ mười có hai phần: phần nguyên và phần thập phân. Để đổi một số hệ mười sang hệ hai ta tách thành hai phần:

- *Đổi phần nguyên sang hệ hai:*

*Quy tắc:* Lấy số phần nguyên chia cho 2 và ghi nhớ phần dư, tiếp theo lấy thương của phép chia trước đó chia cho 2 và ghi nhớ phần dư. Cứ tiếp tục như vậy cho tới khi thương bằng 0 thì dừng lại. Lấy các số dư ghép lại theo thứ tự từ số dư nhận được cuối cùng đến số dư đầu tiên.

*Ví dụ:* Đổi số 34,125 hệ mười sang hệ 2. Ta đổi phần nguyên 34 trước:



- *Đổi phần thập phân sang hệ hai:*

*Quy tắc:* Lấy số cần đổi nhân với 2, tích nhận được sẽ gồm phần nguyên và phần lẻ thập phân, lấy phần lẻ thập phân của tích nhận được nhân tiếp với 2. Cứ tiếp tục như vậy cho tích chẵn bằng 1. Ghép các phần nguyên (phần trước dấu phẩy) của các tích thu được theo thứ tự nhận từ tích đầu tiên đến tích cuối cùng ta sẽ được các chữ số sau dấu phẩy của hệ hai cần tìm.

*Ví dụ:* Tiếp ví dụ trên, bây giờ ta đổi phần thập phân 0,125 sang hệ hai:

$$\begin{array}{rcl} 0,125 & \times & 2 & = & \boxed{0},250 \\ 0,250 & \times & 2 & = & \boxed{0},500 \\ 0,500 & \times & 2 & = & \boxed{1},000 \end{array}$$

↓

$$0,125\text{D} = 0,001\text{B}$$

Kết hợp cả hai phần ta có 34,125D = 100010,001B

**3. Số BCD (số hệ mười mã hóa bằng hệ hai)**

Giữa hệ mười và hệ hai còn tồn tại một loại số gọi là số BCD. Số BCD là các số hệ mười được mã hóa bằng hệ hai (Binary Coded Decimal number). Các số BCD rất thích hợp cho các thiết bị đo có phần hiển thị số ở đầu ra. Ở loại số này người ta dùng 4 chữ số hệ hai để mã hoá

một số hệ mười có giá trị trong khoảng 0 .. 9. Vì chỉ mã hoá mười số nên không dùng hết các tổ hợp của 4 bit.

*Vi dụ:* + Số 495 nếu biểu diễn theo số BCD thì được 0100 1001 0101  
 + Số BCD 0111 0010 0000 0001 biểu diễn số hệ mười là 7201

Ví dụ trên cho ta cách chuyển đổi giữa số hệ mười và số BCD.

Số BCD có hai dạng: dạng BCD gói (còn được gọi là BCD chuẩn) và dạng BCD không gói. Dạng BCD gói biểu diễn hai chữ số hệ mười trong một byte, còn dạng BCD không gói biểu diễn một chữ số hệ mười trong một byte.

*Vi dụ:*

Số hệ 10	BCD gói	BCD không gói
12	0001 0010	0000 0001 0000 0010
623	0000 0110 0010 0011	0000 0110 0000 0010 0000 0011

#### 4. Hệ đếm cơ số 16 (hệ hex, hexa-decimal number system)

Nếu dùng hệ 2 để biểu diễn các số có giá trị lớn ta sẽ gặp điều bất tiện là số hệ 2 thu được quá dài. Bởi vậy, trong thực tế người ta tìm cách nhóm 4 số hệ 2 thành một số hệ 16,

Hệ đếm cơ số 16 sử dụng 16 chữ số để biểu diễn các số hệ 16. Mười chữ số thập phân ‘0’...‘9’ để biểu diễn các giá trị từ 0 đến 9, còn các giá trị từ 10 đến 15 vì không còn chữ số nữa nên phải dùng thêm các chữ cái:

A	biểu diễn cho	10
B	biểu diễn cho	11
C	biểu diễn cho	12
D	biểu diễn cho	13
E	biểu diễn cho	14
F	biểu diễn cho	15

Khi đếm đến 16 vì không còn chữ số nên phải thêm 1 sang bên trái để tạo thành ‘10’,... Giá trị của một số hệ 16 bằng tổng các tích giữa giá trị của từng chữ số nhân với trọng số của nó. Trọng số của chữ số thứ i trong một số hệ 16 là  $16^i$ .

*Vi dụ:*  $2AE4 = 2 \times 16^3 + 10 \times 16^2 + 14 \times 16^1 + 4 \times 16^0$

Để phân biệt với các số trong hệ đếm khác, các số hệ 16 có thêm chữ H ở cuối. *Vi dụ:* 2AE4H

- Đổi số hệ 16 sang hệ 2
  - Đổi từng chữ số hệ 16 thành nhóm 4 chữ số hệ 2.
  - Đổi số hệ 2 sang hệ 16
- Nhóm 4 chữ số hệ 2 một từ phải qua trái rồi chuyển từng nhóm này thành một chữ số hệ mười sáu.

*Vi dụ:* 100101101011B = 1001 0110 1011 = 96BH

#### 4. Các phép toán số học đối với hệ 2 và hệ 16

Sinh viên tự nghiên cứu

## II - MÃ HOÁ THÔNG TIN

### 1. Khái niệm về mã hoá

Mã hoá thông tin là quy ước về cách biểu diễn thông tin trong máy tính. Trong máy tính người ta dùng mã nhị phân (số nhị phân) có độ dài (số bit) cố định để biểu diễn thông tin. Với độ dài từ mã là n ta có thể biểu diễn được  $2^n$  trạng thái khác nhau.

*Xét trường hợp biểu diễn số:*

\* Nếu dùng 1 byte (8 bit) để biểu diễn các số nguyên không dấu (số tự nhiên) thì ta có thể biểu diễn được  $2^n = 2^8 = 256$  số từ 0 đến 255.

Mã nhị phân	Số
0000 0000	0
0000 0001	1
0000 0010	2
.....	....
1111 1111	255

\* Nếu dùng 1 byte để biểu diễn các số nguyên có dấu thì có thể biểu diễn được  $2^n = 2^8 = 256$  số từ -128 đến 127. Sở dĩ khoảng số nguyên biểu diễn được là từ -128 đến 127 là vì phải dùng mất 1 bit để biểu diễn dấu. Bit biểu diễn dấu là bit MSB. MSB = 1 cho số âm, MSB = 0 cho số dương. Khi MSB = 0 thì ta có số dương lớn nhất là 0111 1111 = 127. Khi MSB = 1 ta có số âm, số âm có nhiều cách biểu diễn nhưng hay dùng nhất là kiểu số bù hai. Với kiểu số bù hai số âm nhỏ nhất là 1000 0000 = -128 (ta sẽ nói cách tìm số -128 sau).

Cách tìm số bù hai biểu diễn một số âm như sau:

- Đổi trị tuyệt đối của số âm ra mã nhị phân (với độ dài từ mã cố định, ví dụ 8, 16, 32)
- Tìm số bù một của số nhị phân bằng cách đảo các bit: 0 thành 1, 1 thành 0
- Cộng số bù một với 1 ta sẽ được số bù hai

*Ví dụ:* Biểu diễn số -128 trong dạng số bù hai 8 bit

128D = 1000 0000B

Bù một = 0111 1111

+ 1

Bù hai = 1000 0000

Ngược lại, từ một số nhị phân hãy tìm số nguyên mà nó biểu diễn (giả sử số nhị phân biểu diễn số nguyên có dấu). Khi đó ta làm như sau:

- Xét bit MSB. Nếu MSB = 0 thì chỉ cần đổi số nhị phân sang hệ mười. Nếu MSB = 1 thì chuyển sang bước tiếp theo.

- Tìm số bù một
- Lấy số bù một cộng với 1 để được số bù hai
- Đổi số nhị phân bù hai sang hệ mười

*Ví dụ:* Cho số nhị phân 1101 1001, số này biểu diễn một số nguyên có dấu, hãy tìm số nguyên đó.

- Bit MSB = 1 => Số biểu diễn là số âm
- Số bù một = 0010 0110
- Số bù hai = 0010 0111
- Số nguyên cần tìm là -39

## 2. Mã ASCII

Thông tin, dữ liệu xung quanh chúng ta có rất nhiều loại như văn bản, số liệu, âm thanh, hình ảnh, ... Muốn đưa các loại thông tin này vào máy tính chúng ta phải dùng mã nhị phân để biểu diễn. Trong thực tế thông tin được truyền đi, được lưu giữ trong các bộ nhớ máy tính hoặc để hiển thị trên màn hình đều ở dưới dạng ký tự và tuân theo một loại mã được dùng rộng rãi trên thế giới gọi là mã ASCII (American Standard Code for Information Interchange, mã chuẩn của Mỹ dùng để trao đổi thông tin). Việc dùng các ký tự để mã hoá thông tin theo bảng mã ASCII cho phép các máy tính và các bộ phận của một máy tính có thể trao đổi thông tin với nhau.

Các từ mã trong bảng mã ASCII có độ dài 1 byte (8 bit). Bảng mã ASCII được chia thành 2 nửa: nửa đầu có mã từ 0 đến 127 gọi là bảng mã ASCII tiêu chuẩn, nửa sau có mã từ 128 đến 255 gọi là bảng mã ASCII mở rộng. Hầu hết các nước trên thế giới có bảng mã ASCII tiêu chuẩn giống nhau. Bảng mã ASCII mở rộng thường khác nhau và dùng để chứa bộ ký tự của riêng từng nước.

Trong bảng mã ASCII tiêu chuẩn người ta chỉ dùng hết 7 bit (từ bit 0 đến bit 6) để mã hoá các ký tự, còn 1 bit MSB có thể cho liên tục bằng 0, hoặc bằng 1, hoặc có thể dùng để chứa bit parity phục vụ việc phát hiện lỗi khi truyền.

Khi xem xét bảng mã ASCII tiêu chuẩn ta có thể rút ra mấy nhận xét sau:

Hai cột đầu (cột 0 và cột 1) của bảng mã dùng cho các ký tự điều khiển

Các ký tự số nằm ở cột 3. Mã của ký tự '0' là 30H, của ký tự '9' là 39H. Giữa giá trị số và mã ASCII của số đó có khoảng cách là 30H.

Các chữ cái hoa nằm ở cột 4 và 5, các chữ cái thường nằm ở cột 6 và 7. Khoảng cách giữa chữ thường và chữ hoa cùng tên là 20H.

**BẢNG MÃ ASCII TIÊU CHUẨN**

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

### 3. Quan hệ giữa mã ASCII và số BCD

Khi lưu trữ, hiển thị hoặc truyền giữa các thiết bị các giá trị số 0 .. 9 thực chất ta làm việc với mã ASCII của các số đó, tức là các số 30H .. 39H. Ta thấy trong một byte biểu diễn các giá trị số 0 .. 9 có 4 bit thấp ứng với mã BCD của chính số đó, còn 4 bit cao luôn luôn là mã BCD của số 3. Nếu ta thay 4 bit cao trong mã ASCII của một số bằng 0000B thì ta thu được số BCD không gói của số đó.

## III - SƠ LƯỢC VỀ CẤU TRÚC VÀ HOẠT ĐỘNG CỦA HỆ VI XỬ LÝ

### 1. Sự phát triển của các bộ vi xử lý

Bộ vi xử lý là thành phần rất cơ bản không thể thiếu được để tạo nên máy vi tính và các hệ vi xử lý. Trước khi tìm hiểu cấu trúc của một hệ vi xử lý ta lướt qua về lịch sử phát triển của các bộ vi xử lý.

#### a) Thế hệ 1 (1971 - 1973)

Năm 1971, hãng Intel đã cho ra đời bộ vi xử lý đầu tiên là 4004 (4 bit số liệu, 12 bit địa chỉ). Sau đó Intel và các hãng sản xuất khác lần lượt cho ra đời các bộ vi xử lý khác: 4040 (4 bit) và 8008 (8 bit) của Intel, PPS-4 (4 bit) của Rockwell International, IPM-16 (16 bit) của National Semiconductor.

Đặc điểm chung của các bộ vi xử lý thế hệ này là:

- + Độ dài từ thường là 4 bit (có thể dài hơn)
- + Công nghệ chế tạo PMOS có mật độ tích hợp nhỏ
- + Tốc độ thực hiện lệnh: 10 - 60  $\mu$ s/lệnh với tần số đồng hồ fclk = 0,1 - 0,8 MHz
- + Tập lệnh đơn giản và phải cần nhiều vi mạch phụ trợ mới tạo nên một hệ vi mạch hoàn chỉnh.

#### b) Thế hệ 2 (1974 - 1977)

Các bộ vi xử lý đại diện cho thế hệ này là các bộ vi xử lý 8 bit 8080 và 8085 của Intel, 6800 và 6809 của Motorola và Z80 của Zilog.

Đặc điểm:

- + Độ dài từ là 8 bit
- + Công nghệ chế tạo: NMOS (có mật độ tích hợp cao hơn PMOS) hoặc CMOS
- + Tốc độ thực hiện lệnh: 1 - 8  $\mu$ s/lệnh với tần số đồng hồ fclk = 1 - 5 MHz
- + Tập lệnh phong phú hơn thế hệ trước.

#### c) Thế hệ 3 (1978 - 1982)

Các bộ vi xử lý đại diện cho thế hệ này là các bộ vi xử lý 16 bit 8086, 80186, 80286 của Intel, 68000 và 68010 của Motorola.

Đặc điểm:

- + Độ dài từ là 16 bit
- + Công nghệ chế tạo: HMOS (có mật độ tích hợp cao hơn PMOS) hoặc CMOS
- + Tốc độ thực hiện lệnh: 0,1 - 1  $\mu$ s/lệnh với tần số đồng hồ fclk = 5 - 10 MHz
- + Tập lệnh đa dạng với các lệnh nhân, lệnh chia và các lệnh thao tác với chuỗi ký tự.

#### d) Thế hệ 4 (1983 - ?)

Các bộ vi xử lý đại diện cho thế hệ này là các bộ vi xử lý 32 bit 80386, 80486 và 64 bit Pentium của Intel, các bộ vi xử lý 32 bit 68020, 68030, 68040, 68060 của Motorola.

Đặc điểm của các bộ vi xử lý thế hệ này là bus địa chỉ đều là 32 bit (có khả năng đánh địa chỉ cho bộ nhớ tới 4 GB) và có khả năng làm việc với bộ nhớ ảo. Người ta cũng áp dụng các cơ chế hoặc các cấu trúc đã được sử dụng trong các máy tính lớn vào các bộ vi xử lý: cơ chế xử lý xen kẽ liên tục dòng mã lệnh (pipeline), bộ nhớ cache (bộ nhớ ẩn), bộ nhớ ảo. Các bộ vi

xử lý thế hệ này đều có bộ quản lý bộ nhớ (MMU) và nhiều khi cả bộ đồng xử lý toán học ở bên trong. Phần lớn các bộ vi xử lý thế hệ này đều sản xuất bằng công nghệ HCMOS.

Bên cạnh các bộ vi xử lý vạn năng truyền thống thường được sử dụng để xây dựng các máy tính với tập lệnh đầy đủ (Complex Instruction Set Computer, CISC) mà chúng ta đã nói ở trên, trong thời gian này cũng đã xuất hiện các bộ vi xử lý cải tiến dùng để xây dựng các máy tính với tập lệnh rút gọn (Reduced Instruction Set Computer, RISC) với nhiều tính năng có thể so sánh với các máy tính lớn ở thế hệ trước. Đó là các bộ vi xử lý Alpha của hãng Digital, PowerPC của ba hãng Apple-Motorola-IBM.

Dưới đây là một số thông số chính của các bộ vi xử lý Intel

### Các bộ vi xử lý 16 bit của Intel

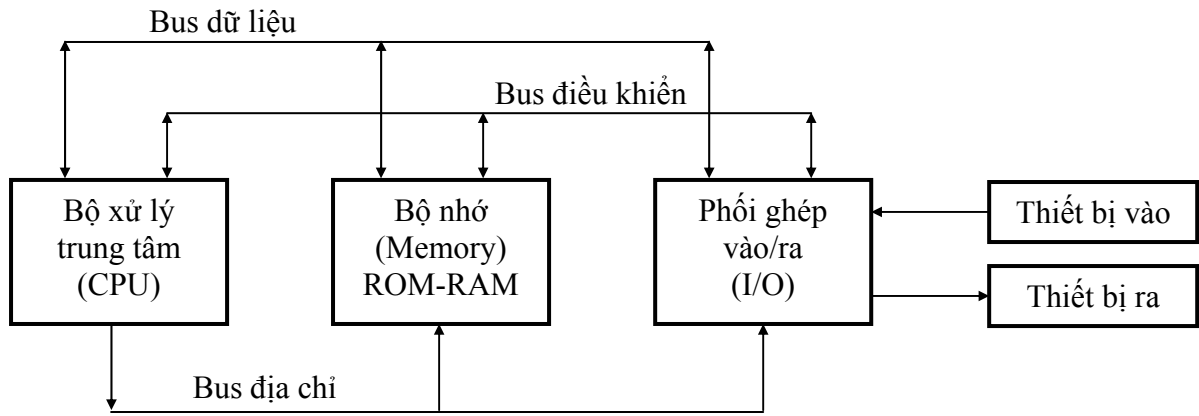
	<b>8086</b>	<b>8088</b>	<b>80286</b>
Năm sản xuất	6/1978	1979	2/1982
$f_{clk}$ max (đồng hồ nhịp)	10 MHz	10 MHz	20 MHz
MIPS (triệu lệnh / s)	0,33	0,33	1,2
Số tranzitor	29.000	29.000	134.000
Bus số liệu	16 bit	8 bit	16 bit
Bus địa chỉ	20 bit	20 bit	24 bit
Khả năng địa chỉ hoá	1 MB	1 MB	16 MB
Số chân	40	40	68
Chế độ bộ nhớ ảo	không	không	có
Có bộ quản lý bộ nhớ bên trong	không	không	có
Đồng xử lý toán học	8087	8087	80287

### Các bộ vi xử lý 32 bit của Intel

	<b>386DX</b>	<b>386SX</b>	<b>486DX</b>	<b>486SX</b>	<b>486DX2</b>	<b>Pentium</b>
Năm sản xuất	10/1985	6/1988	4/1989	4/1991	3/1992	5/1993
$f_{clk}$ max (đồng hồ nhịp)	40 MHz	33 MHz	50 MHz	25 MHz	66 MHz	100 MHz
MIPS (triệu lệnh / s)	6	2,5	20	16,5	52	112
Số tranzitor	275.000	275.000	1,2 triệu	1,18 triệu	1,2 triệu	3,1 triệu
Bộ nhớ Cache	Bên ngoài, do 82385 đ.khiển	Bên ngoài, do 82385 đ.khiển	I/DCache 8 KB	I/DCache 8 KB	I/DCache 8 KB	ICache 8KB DCache 8KB
Bus số liệu	32 bit	16 bit	32 bit	32 bit	32 bit	64 bit
Bus địa chỉ	32 bit	24 bit	32 bit	32 bit	32 bit	32
Khả năng địa chỉ hoá	4 GB	16 MB	4 GB	4 GB	4 GB	4 GB
Số chân	132	100	168	168	168	273
Chế độ bộ nhớ ảo	có	có	có	có	có	có
Có bộ quản lý bộ nhớ và bảo vệ bộ nhớ bên trong	có	có	có	có	có	có
Đồng xử lý toán học	80387DX	80387SX	bên trong	80487SX	bên trong	bên trong

## 2. Sơ lược cấu trúc và hoạt động của hệ vi xử lý

Để tạo thành một hệ vi xử lý hoàn chỉnh, bộ vi xử lý cần phải kết hợp với các bộ phận điện tử khác như bộ nhớ và các bộ phối ghép vào/ra. Cần lưu rằng máy vi tính chỉ là một trong những ứng dụng cụ thể của hệ vi xử lý. Dưới đây là sơ đồ khối tổng quát của một hệ vi xử lý.



Trong sơ đồ này ta thấy các khối chức năng chính của hệ vi xử lý gồm có:

- + Khối xử lý trung tâm (Central Processing Unit, CPU)
- + Bộ nhớ trong RAM-ROM (Memory, M)
- + Khối phối ghép với các thiết bị ngoại vi (Input/Output, I/O)
- + Các bus truyền thông tin giữa các khối

\* *Khối xử lý trung tâm (CPU, còn gọi là bộ vi xử lý)* đóng vai trò chủ đạo trong hệ vi xử lý. Khi hoạt động, nó đọc mã lệnh được ghi dưới dạng các bit 0 và 1 từ bộ nhớ, sau đó nó sẽ giải mã các lệnh này thành dãy các xung điều khiển ứng với các thao tác trong lệnh để điều khiển các khối khác thực hiện từng bước các thao tác đó. Để làm được việc này bên trong CPU có các thanh ghi (ô nhớ): thanh ghi con trỏ lệnh IP dùng để chứa địa chỉ của các lệnh sắp thực hiện, các thanh ghi con trỏ và chỉ số, các thanh ghi đa năng cùng bộ tính toán số học và logic (ALU, Arithmetic and Logic Unit) để thao tác với dữ liệu. Ngoài ra bên trong CPU còn có các hệ thống mạch điện tử rất phức tạp để giải mã lệnh và từ đó tạo ra các xung điều khiển cho toàn hệ.

\* *Bộ nhớ bán dẫn* hay còn gọi là *bộ nhớ trong* là một bộ phận rất quan trọng của hệ vi xử lý. Bộ nhớ bán dẫn gồm có ROM và RAM. ROM dùng để chứa chương trình điều khiển của toàn hệ, khi bật điện CPU sẽ lấy lệnh từ đây để khởi động hệ thống. RAM chứa một phần chương trình điều khiển hệ thống, các chương trình ứng dụng, dữ liệu và các kết quả của chương trình. Các dữ liệu và chương trình muốn lưu trữ lâu dài sẽ được để ở bộ nhớ ngoài (đĩa từ, băng từ). Khi hoạt động CPU sẽ lấy các lệnh trong RAM ra thực hiện, bởi vậy các chương trình muốn CPU thực hiện phải được đưa vào RAM. Bộ nhớ trong được tổ chức thành các ô nhớ có độ dài (số bit) bằng nhau, mỗi ô nhớ được gán một địa chỉ cố định và duy nhất.

\* *Khối phối ghép vào/ra (I/O)* là bộ phận giao tiếp giữa hệ vi xử lý với thế giới bên ngoài. Các thiết bị ngoại vi như bàn phím, chuột, màn hình, máy in, đĩa từ,... đều liên hệ với hệ vi xử lý qua bộ phận này. Mỗi giao diện với một thiết bị bên ngoài gọi là cổng. Có 2 loại cổng: cổng vào và cổng ra, cổng vào để lấy thông tin từ ngoài vào và cổng ra để đưa thông tin từ trong hệ ra ngoài. Nếu nhìn từ trong CPU thì khối I/O có thể xem như bộ nhớ, mỗi cổng có một địa chỉ cố định và có thể đọc dữ liệu vào hoặc ghi dữ liệu ra cổng.

\* *Bus* là một nhóm các đường dây dẫn điện để truyền tín hiệu giữa các khối trong hệ. Các đường dây này gọi chung là bus hệ thống. Bus hệ thống gồm có ba bus thành phần: bus địa chỉ chuyên tải tín hiệu địa chỉ, bus dữ liệu chuyên tải tín hiệu dữ liệu và bus điều khiển chuyên tải tín hiệu điều khiển.

+ Bus địa chỉ thường có từ 16, 20, 24 đến 32 đường dây song song chuyên tải các bit địa chỉ (mỗi đường dây vận chuyển một bit). Khi đọc/ghi bộ nhớ hoặc cổng I/O CPU sẽ đưa ra trên bus này địa chỉ của ô nhớ hay cổng I/O cần đọc/ghi. Khả năng phân biệt địa chỉ của CPU (số địa chỉ mà CPU có khả năng quản lý) phụ thuộc vào số bit của bus địa chỉ. Số bit của bus địa chỉ là n thì khả năng phân biệt địa chỉ của CPU là  $2^n$  địa chỉ khác nhau. Ví dụ, bus địa chỉ

có 20 bit thì CPU có khả năng phân biệt địa chỉ là  $2^{20} = 1 \text{ M}$  địa chỉ khác nhau. Cần lưu ý là chỉ CPU mới có khả năng đưa ra địa chỉ trên bus địa chỉ, bởi vậy trên sơ đồ bus địa chỉ có một chiều đi từ CPU tới các khối khác.

+ Bus dữ liệu thường có từ 4, 8, 16, 32 đến 64 đường dây song song tùy thuộc vào bộ vi xử lý cụ thể. Các đường dây này chuyển tải tín hiệu dữ liệu giữa các khối trong hệ. Số lượng đường dây này quyết định số bit dữ liệu mà CPU có khả năng xử lý cùng một lúc. Trên sơ đồ bus dữ liệu có hai chiều để chỉ ra rằng dữ liệu có thể truyền đi từ CPU (dữ liệu ra) hoặc truyền đến CPU (dữ liệu vào).

+ Bus điều khiển thường gồm hàng chục đường dây song song chuyển tải các tín hiệu điều khiển khác nhau. Mỗi tín hiệu điều khiển truyền trên một đường dây và có một chiều nhất định. Vì khi hoạt động CPU đưa tín hiệu điều khiển tới các khối khác trong hệ, đồng thời nó cũng nhận tín hiệu điều khiển từ các khối đó để phối hợp hoạt động. Trên sơ đồ bus điều khiển có mỗi tên 2 chiều là để thể hiện tính hai chiều của cả nhóm tín hiệu chứ không phải một tín hiệu.