

1. Thiết kế mạch giải mã

2. Thanh ghi

- **Các thanh ghi đa năng:**
 - 4 thanh ghi 16 bits:
 - AX: thanh ghi tổng, thường dùng để lưu kết quả
 - BX: thanh ghi cơ sở, thường dùng chứa địa chỉ ô nhớ
 - CX: thanh ghi đếm
 - DX: thanh ghi dữ liệu
 - Hoặc 8 thanh ghi 8 bits: AH, AL, BH, BL, CH, CL, DH, DL
- **Các thanh ghi con trỏ và chỉ số:**
 - SP: Con trỏ ngăn xếp. SP luôn chứa địa chỉ đỉnh ngăn xếp
 - BP: Con trỏ cơ sở - sử dụng với đoạn ngăn xếp
 - SI: Thanh ghi chỉ số nguồn.
 - DI: Thanh ghi chỉ số đích.
- **Các thanh ghi đoạn:**
 - CS (Code Segment): Thanh ghi đoạn mã. CS chứa địa chỉ bắt đầu đoạn mã
 - DS (Data Segment): Thanh ghi đoạn dữ liệu. DS chứa đại chỉ bdd đoạn dữ liệu
 - SS (Stack Segment): Thanh ghi đoạn ngăn xếp.
 - ES (Extra Segment): Thanh ghi đoạn dữ liệu mở rộng
- **Con trỏ lệnh (IP) và thanh ghi cờ (FR)**
- **Các cờ trạng thái:**
 - C (Carry): cờ nhớ. $C = 1 \rightarrow$ có nhớ, $C = 0 \rightarrow$ không nhớ
 - A (Auxiliary): cờ nhớ phụ. $A = 1 \rightarrow$ có nhớ phụ, $A = 0 \rightarrow$ không nhớ phụ
 - P (Parity): cờ chẵn lẻ. $P = 1$ khi tổng số bit 1 trong kết quả là lẻ, $P = 0$ khi tổng số bit 1 trong kết quả là chẵn
 - O (Overflow): cờ tràn. $O = 1$ khi kết quả bị tràn
 - Z (Zero): cờ zero. $Z = 1$ khi kết quả bằng 0, ngược lại $Z = 0$
 - S (sign): cờ dấu. $S = 1$ khi kết quả âm, $S = 0$ khi kết quả không âm
- **Các cờ điều khiển:**
 - D (Direction): cờ hướng, chỉ hướng tăng giảm địa chỉ với các lệnh chuyển dữ liệu. $D = 0 \rightarrow$ địa chỉ tăng. $D = 1 \rightarrow$ địa chỉ giảm

- T (Trap/ Trace): cờ bẫy/ lần vết, được dùng khi gỡ rối ct. $T = 1 \rightarrow$ COU ở chế độ chạy từng lệnh
- I (Interrupt): cờ ngắt. $I = 1 \rightarrow$ cho phép ngắt, $I = 0 \rightarrow$ cấm ngắt

3. Tập lệnh của cpu

4. Hợp ngữ 8086, 8088

Đổi chữ hoa \rightarrow thường, thường \rightarrow hoa,....

5. Chức năng đoạn lệnh

6. Chương 1-2 chủ yếu là lý thuyết

7. Raid (1-2 câu lý thuyết)

- Có 7 loại RAID (RAID 0 : RAID 6)
- K phân cấp RAID
- Tập hợp nhiều đĩa vật lý được HĐH coi như một đĩa (logic) duy nhất
- Dữ liệu được phân bố trên nhiều đĩa vật lý khác nhau
- Dung lượng RAID lên tới hàng nghìn GB
- Do dung lượng lớn \rightarrow cần có một phần đĩa dùng để lưu trữ thông tin an toàn

8. Nhiều bài tập tính toán, ít lý thuyết (10%)

9. K vào ROM, RAM

10. bài tập: cache, tập lệnh cpu, hợp ngữ, giải mã

- Nhị phân \rightarrow hexa: $11\ 1011\ 1110\ 0110\ (2) = 3BE6(16)$
- Hexa \rightarrow nhị phân: $3E8(16) = 11\ 1110\ 1000\ (2)$
- Thập phân \rightarrow hexa: $14988 = ?$
 - $14988 : 16 = 936$ dư 12 tức là C
 - $936 : 16 = 58$ dư 8
 - $58 : 16 = 3$ dư 10 tức là A
 - $3 : 16 = 0$ dư 3

$\rightarrow 14988(10) = 3A8C\ (16)$
- Hexa \rightarrow thập phân: $3A8C = ?$
 - $3A8C\ (16) = 3 \cdot 16^3 + 10 \cdot 16^2 + 8 \cdot 16^1 + 12 \cdot 16^0 = 14988(10)$
- Mã ASCII
 - 0: 30h – 0,...9 tăng dần, 48- 57 DEC (nhị phân)
 - A: 41h – A,...Z tăng dần, 65 – 90

- a:61h – a,...z tăng dần, 97 – 122

CHƯƠNG 1: GIỚI THIỆU CHUNG

1. LỊCH SỬ PHÁT TRIỂN MÁY TÍNH

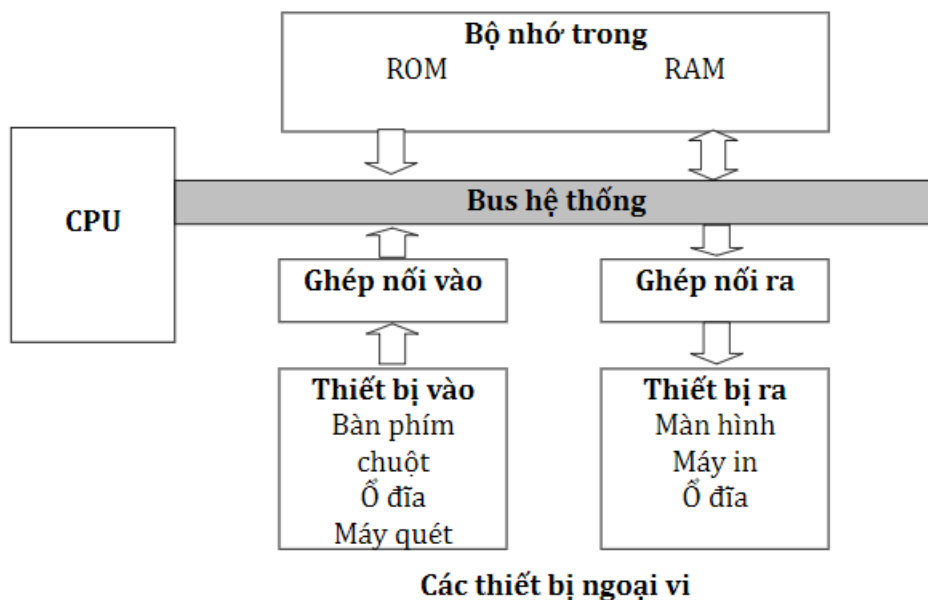
- Thế hệ 1 (1944-1959):
 - Sử dụng bóng đèn điện tử
 - Dùng băng từ làm các thiết bị đầu vào/ ra
 - Mật độ tích hợp: 1000 linh kiện/ foot³ (1 foot = 30.48 cm)
 - Vd: ENIAC_ 1946_ 500000 USD
- Thế hệ 2 (1960 – 1964):
 - Sử dụng transistors
 - 100.000 linh kiện/ foot³
 - Vd: UNIVAC 1107, UNIVAC III, IBM 7070, 7080, 7090, 1400 series, 1600 series.
- Thế hệ 3 (1964 – 1975):
 - Sử dụng mạch tích hợp (IC)
 - 10 triệu linh kiện/ foot³
 - Vd: UNIVAC 9000 series, IBM System/360, System 3, System 7
- Thế hệ 4 (1975 – 1989):
 - Sd LSI
 - 1 tỷ linh kiện/ foot³
 - Vd: IBM System 3090, IBM RISC 6000, IBM RT, Cray 2 XMP
- Thế hệ 5 (1990 -nay):
 - Sd VLSI
 - <10 nm – 0,045 μm

2. KHÁI NIỆM KIẾN TRÚC VÀ TỔ CHỨC MÁY TÍNH

- Kiến trúc máy tính: là khoa học về lựa chọn và kết nối các thành phần phần cứng của máy tính nhằm đạt yêu cầu:
 - Hiệu năng: càng nhanh càng tốt
 - Chức năng: nhiều chức năng
 - Giá thành: càng rẻ càng tốt
- Tổ chức máy tính: là khoa học nghiên cứu các thành phần của máy tính và phương thức làm việc của chúng dựa trên kiến trúc cho trước
- 3 thành phần cơ bản của kiến trúc máy tính

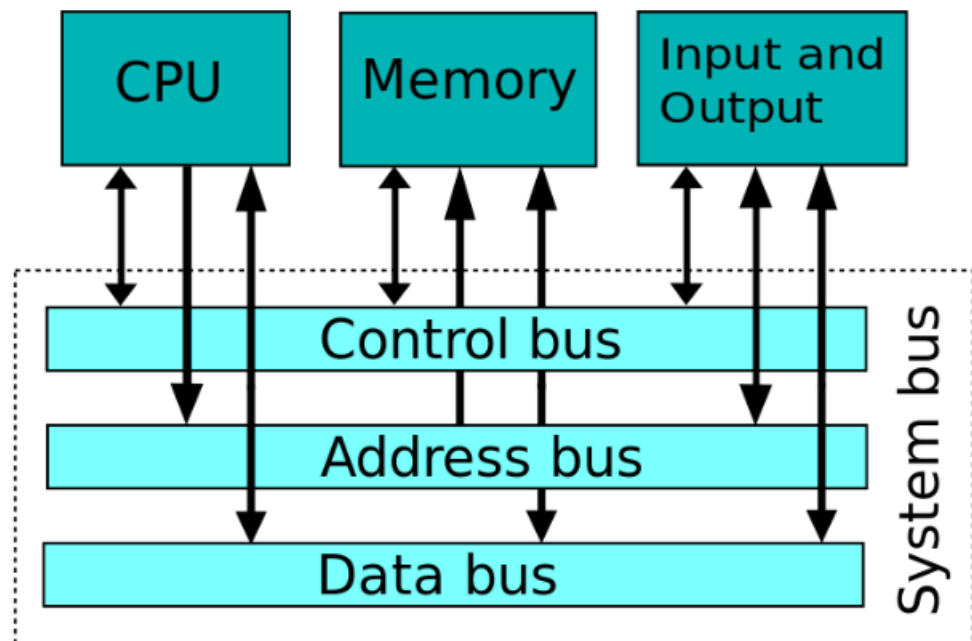
1. Kiến trúc tập lệnh: là hình ảnh trừu tượng của máy tính ở mức ngôn ngữ máy (hoặc hợp ngữ), bao gồm:
 - Tập lệnh
 - Các chế độ địa chỉ bộ nhớ
 - Các thanh ghi
 - Khuôn dạng địa chỉ và dữ liệu
2. Vi kiến trúc: mô tả hệ thống ở mức thấp
3. Thiết kế hệ thống, bao gồm tất cả các thành phần phần cứng khác trong hệ thống máy tính.

3. CẤU TRÚC VÀ CHỨC NĂNG CỦA MÁY TÍNH



- Bộ xử lý trung tâm (CPU):
 - Chức năng:
 - Đọc lệnh từ bộ nhớ
 - Giải mã và thực hiện lệnh
 - Bao gồm:
 - Khối điều khiển (CU)
 - Khối tính toán số học và logic (ALU)
 - Các thanh ghi
 - Bus trong CPU
- Bộ nhớ trong:
 - Lưu trữ lệnh và dữ liệu để CPU xử lý
 - Bao gồm:

- ROM
 - Lưu trữ lệnh và dữ liệu của hệ thống
 - Thông tin trong ROM vẫn tồn tại khi mất nguồn nuôi
- RAM
 - Lưu trữ lệnh và dữ liệu của hệ thống và người dùng
 - Thông tin trong RAM sẽ mất khi mất nguồn nuôi
- Các thiết bị vào ra:
 - Thiết bị vào: nhập dữ liệu và điều khiển (Bàn phím, chuột, ổ đĩa, máy quét)
 - Thiết bị ra: kết xuất dữ liệu (màn hình, máy in, ổ đĩa)
- Bus hệ thống, bao gồm 3 loại:
 - Bus địa chỉ (bus A)
 - Bus dữ liệu (bus D)
 - Bus điều khiển (bus C)



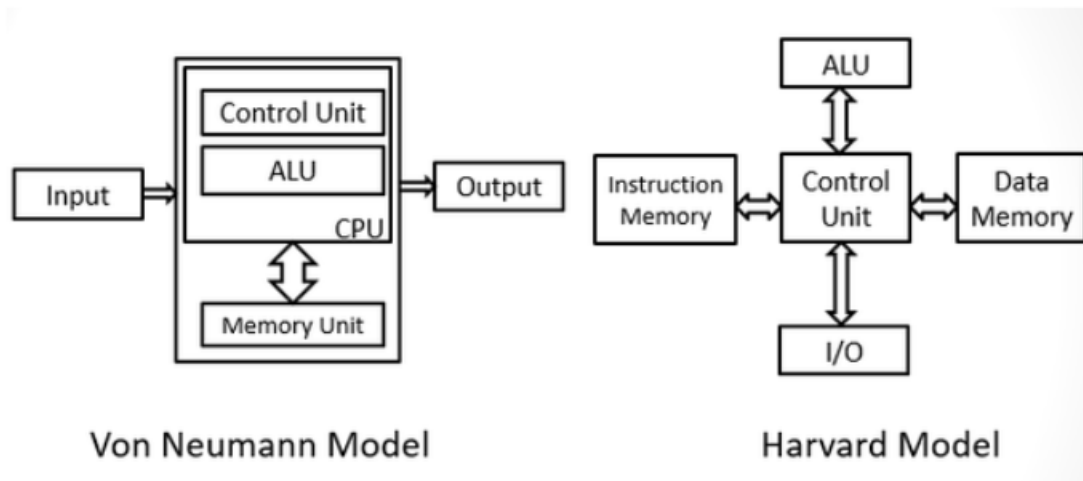
Luật Moore

- ❑ Quan sát bởi Gordon Moore, CEO của Intel, 1965
- ❑ Số transistors trên chip sẽ tăng gấp đôi trong ~18m
- ❑ Giá thành của chip hầu như không thay đổi
- ❑ Mật độ cao hơn, do vậy đường dẫn ngắn hơn
- ❑ Kích thước IC nhỏ hơn dẫn tới độ phức tạp tăng lên
- ❑ Điện năng tiêu thụ ít hơn
- ❑ Thiết kế tối ưu hơn

4. KIẾN TRÚC VON NEUMANN

Cmp

5. KIẾN TRÚC HARVARD



Von Neumann: địa chỉ và bộ nhớ chia sẻ dữ liệu và lệnh từ CPU

Harvard: 2 địa chỉ và bộ nhớ cho dữ liệu và CPU → xử lý song song

Havard Architecture	Von Neumann Architecture
Dùng các đường truyền dữ liệu và truyền lệnh điều khiển riêng → linh hoạt	Đường truyền dữ liệu và truyền lệnh điều khiển chung
Có một tập địa chỉ và đường truyền (bus) riêng để đọc và ghi data vào bộ nhớ Và tập địa chỉ và đường truyền (bus) riêng để đọc lệnh	Có một tập địa chỉ và đường truyền (bus) chung để đọc / ghi data vào bộ nhớ; đồng thời đọc lệnh
Kiến trúc cho phép CPU có thể đọc lệnh và truy cập bộ nhớ đồng thời	Kiến trúc không cho phép CPU có thể đọc lệnh và truy cập bộ nhớ đồng thời. Tại mỗi thời điểm chỉ có một hệ thống truyền (bus) được hoạt động cho việc đọc lệnh hoặc đọc/ghi dữ liệu.
Có hai không gian địa chỉ cho mã lệnh và dữ liệu khác nhau: Địa chỉ 0 của địa chỉ khác địa chỉ lệnh 0 (instruction address 0) khác địa chỉ dữ liệu 0 (data address 0)	Kiến trúc Von Neumann có cùng không gian địa chỉ cho cả dữ liệu (data) và lệnh (instructions).

6. BIỂU DIỄN DỮ LIỆU TRONG MÁY TÍNH

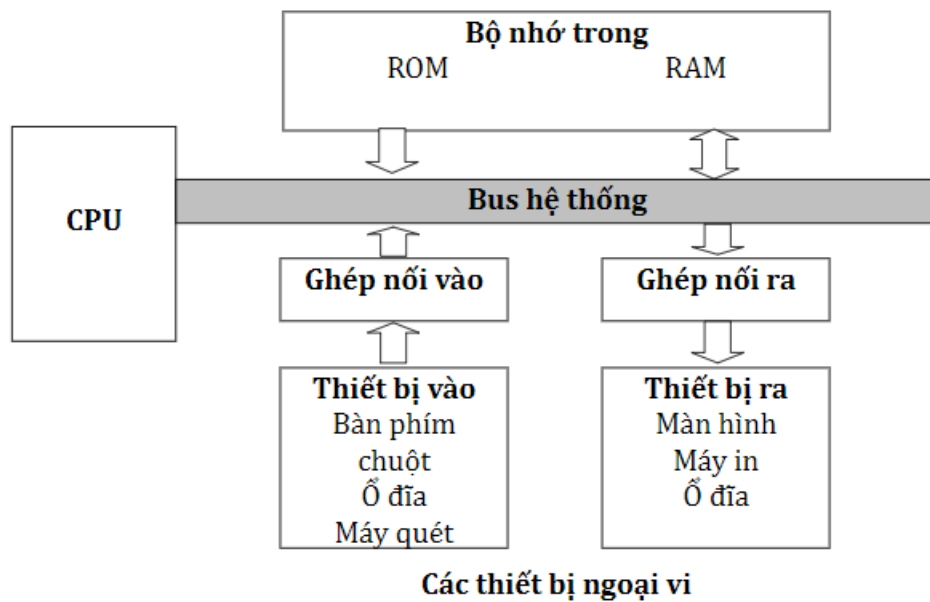
Số có dấu: $-2^{n-1} + 1 \rightarrow 2^{n-1} - 1$

Số không dấu: $0 \rightarrow 2^n - 1$

CHƯƠNG 1: GIỚI THIỆU CHUNG

1. MÁY TÍNH VÀ PHÂN LOẠI MÁY TÍNH

- Máy tính: nhận thông tin – xử lý thông tin – xuất thông tin
- Các thành phần cơ bản của máy tính
 - Bộ xử lý
 - Bus liên kết
 - Hệ thống vào/ra
 - Hệ thống nhớ



- Phân loại
 - Phân loại máy tính theo truyền thống
 - Bộ vi điều khiển
 - Máy vi tính
 - Máy tính nhỏ
 - Máy tính lớn
 - Siêu máy tính
 - Phân loại máy tính theo hiện đại
 - Máy tính để bàn
 - Máy chủ
 - Máy tính nhúng
 - Chức năng cơ bản của máy tính
 - Xử lý dữ liệu
 - Lưu trữ dữ liệu
 - Trao đổi dữ liệu
 - Điều khiển
- Bộ xử lý
 - Chức năng
 - Điều khiển hoạt động của máy tính
 - Xử lý dữ liệu
 - Nguyên tắc hoạt động cơ bản: BXL hoạt động theo ct nằm sẵn trong bộ nhớ, gồm 2 bước cơ bản:

- Nhận lệnh từ bộ nhớ
 - Giải mã và thực hiện lệnh
- Bộ vi xử lý: là bộ xử lý đc chế tạo trên một chip
- Các thành phần cơ bản của BXL
 - Khối điều khiển (CU)
 - ALU
 - Các thanh ghi
 - Bus bên trong
- Hệ thống nhớ
 - Chức năng: lưu trữ ct và dữ liệu
 - Các thao tác cơ bản với bộ nhớ
 - Đọc
 - Ghi
 - Các thành phần chính:
 - Bộ nhớ trong
 - Bộ nhớ ngoài
- Hệ thống vào/ra
- Bus liên kết
- Một số tín hiệu điều khiển điển hình
 - Các tín hiệu điều khiển đọc/ghi:
 - MEMR (memory read): đọc lệnh/dữ liệu từ ngăn nhớ
 - MEMW (memory write): ghi dữ liệu tới ngăn nhớ
 - IOR (I/O read): đọc dữ liệu từ một cổng vào/ra
 - IOW (I/O write): ghi dữ liệu đến một cổng vào/ra
 - Các tín hiệu điều khiển ngắt:
 - INTR (Interrupt Request): tín hiệu từ bên ngoài gửi đến CPU xin ngắt (ngắt chặn được)
 - INTA (Interrupt acknowledge): tín hiệu từ CPU gửi ra ngoài báo chấp nhận ngắt
 - NMI (Non maskable interrupt): tín hiệu ngắt gửi đến CPU (ngắt k chặn đc)
 - Reset: tín hiệu từ bên ngoài khởi động lại máy tính

2. KIẾN TRÚC VÀ TỔ CHỨC MÁY TÍNH

3. CHỨC NĂNG VÀ CẤU TRÚC CỦA MÁY TÍNH

4. HOẠT ĐỘNG CỦA MÁY TÍNH

5. LỊCH SỬ PHÁT TRIỂN MÁY TÍNH

CHƯƠNG 2: KHỐI XỬ LÝ TRUNG TÂM CPU

CPU - Sơ đồ khối tổng quát

CU: (Control Unit) Khối điều khiển

IR: (Instruction Register) Thanh ghi lệnh

PC: (Program Counter) Bộ đếm chương trình

MAR: (Memory Address Register) Thanh ghi địa chỉ bộ nhớ

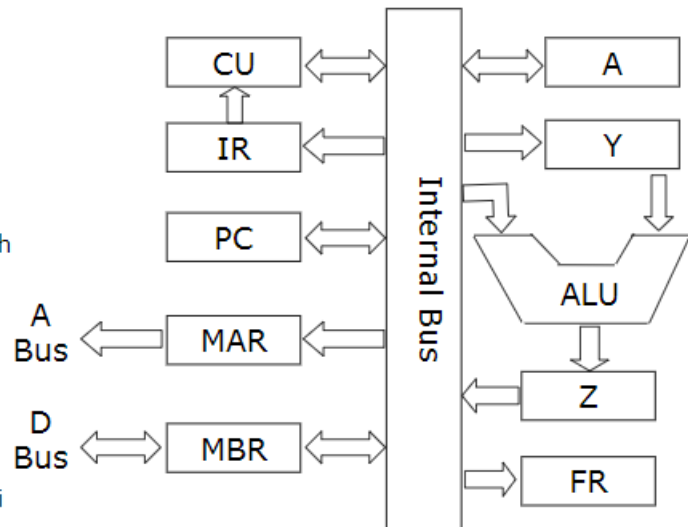
MBR: (Memory Buffer Register) Thanh ghi nhớ đệm

A: (Accumulator Register) Thanh ghi tích lũy

Y, Z: (Temporary Register) Thanh ghi tạm thời

FR: (Flag Register) Thanh ghi cờ

ALU: (Arithmetic and Logic Unit) Khối tính toán số học -logic



Chu kỳ xử lý lệnh

1. Khi một chương trình được chạy, hệ điều hành tải mã chương trình vào bộ nhớ trong
2. Địa chỉ lệnh đầu tiên của chương trình được đưa vào thanh ghi PC
3. Địa chỉ của ô nhớ chứa lệnh được chuyển tới bus A qua thanh ghi MAR
4. Tiếp theo, bus A truyền địa chỉ tới khối quản lý bộ nhớ MMU (Memory Management Unit)
5. MMU chọn ô nhớ và sinh ra tín hiệu READ

6. Lệnh chứa trong ô nhớ được chuyển tới thanh ghi MBR qua bus D
7. MBR chuyển lệnh tới thanh ghi IR. Sau đó IR lại chuyển lệnh tới CU
8. CU giải mã lệnh và sinh ra các tín hiệu xử lý cho các đơn vị khác, ví dụ như ALU để thực hiện lệnh
9. Địa chỉ trong PC được tăng lên để trỏ tới lệnh tiếp theo của chương trình sẽ được thực hiện
10. Thực hiện lại các bước 3->9 để chạy hết các lệnh của chương trình

CHƯƠNG 2: BIỂU DIỄN DỮ LIỆU VÀ SỐ HỌC MÁY TÍNH

- Biểu diễn số thực

Nguyên tắc chung

- Một số thực X có thể biểu diễn như sau:

$$X = (-1)^S * M * R^E$$

Trong đó:

- S (sign): dấu (0: số dương; 1: số âm)
- M (mantissa): phần định trị
- R (radix): cơ số
- E (exponent): số mũ

Vd: $X = -1234.567 = (-1)^1 * 1234567 * 10^{-3}$

$S = 1, M = 1234567, R = 10, E = -3$

Vd: $X = 0,13579 = (-1)^0 * 13579 * 10^{-5}$

$S = 0, M = 13579, R = 10, E = -5$

Các phép toán số học

$$X1 = (-1)^{S1} * M1 * R^{E1}$$

$$X2 = (-1)^{S2} * M2 * R^{E2}$$

- Phép nhân:

$$X1 * X2 = (-1)^{S1 \oplus S2} * (M1 * M2) * R^{E1 + E2}$$

- Phép chia:

$$X1 / X2 = (-1)^{S1 \oplus S2} * (M1 / M2) * R^{E1 - E2}$$

- Phép cộng / trừ:

$$X1 \pm X2 = R^{E1} * [(-1)^{S1} * M1 \pm (-1)^{S2} * M2 * R^{E2 - E1}]$$

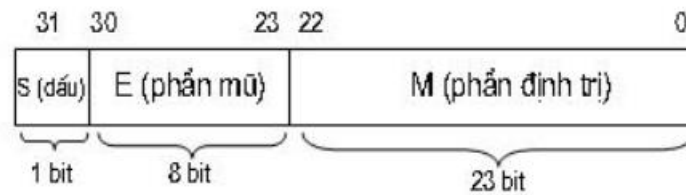
Chuẩn IEEE 754/85

- Có 3 dạng:

- Dạng đơn (single precision): 32 bit
- Dạng kép (double precision): 64 bit
- Dạng kép mở rộng (double-extended precision):
80 bit

- Cả 3 dạng đều có điểm chung: $R = 2$

Dạng đơn: 32 bit



- Khi đó, 1 số thực bất kỳ được biểu diễn:

$$X = (-1)^S * 1, M * 2^{E - 127}$$

- Dải giá trị biểu diễn:

$$2^{-127} < |X| < 2^{127},$$

hay: $10^{-38} < |X| < 10^{38}$

Chuyển đổi → IEEE 754/85 (32 bit)

- Bước 1: Đổi số thực đó → nhị phân
- Bước 2: Chuyển về dạng: $\pm 1, aa...a * 2^b$
- Bước 3: Xác định các giá trị:
 - S = 0 nếu số dương; S = 1 nếu số âm
 - $E - 127 = b \Rightarrow E = 127 + b \rightarrow$ nhị phân
 - M = aa...a00...0
23 bit

Ví dụ chuyển đổi

- Ví dụ 1: $17,625 \rightarrow \text{IEEE 754/85}$
 - B1: $A = 17,625 = 10001,101$
 - B2: $A = 1,0001101 * 2^4$ ($\pm 1,aa...a * 2^b$)
 - B3: Ta có:
 - $S = 0$, vì $A > 0$
 - $E - 127 = 4 \Rightarrow E = 131 = 1000\ 0011$
 - $M = 000\ 1101\ 0000\ 0000\ 0000\ 0000$
- $\Rightarrow A = 0100\ 0001\ 1000\ 1101\ 0000\ 0000\ 0000\ 0000$
 $= 41\ 8D\ 00\ 00H$

Ví dụ chuyển đổi

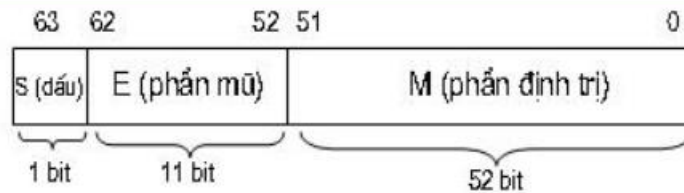
- Ví dụ 4: Tìm giá trị số thực: $C2\ 0D\ 00\ 00H$
- Giải:

Ta có: $X = C2\ 0D\ 00\ 00H =$
 $= 1100\ 0010\ 0000\ 1101\ 0000\ 0000\ 0000\ 0000$

 - $S = 1 \Rightarrow X < 0$
 - $E = 1000\ 0100 = 132 \Rightarrow E - 127 = 5$
 - $M = 000\ 1101\ 0000\ 0000\ 0000\ 0000$

$\Rightarrow X = -1,0001101 * 2^5 = -100011,01 = -35,25$

Dạng kép: 64 bit



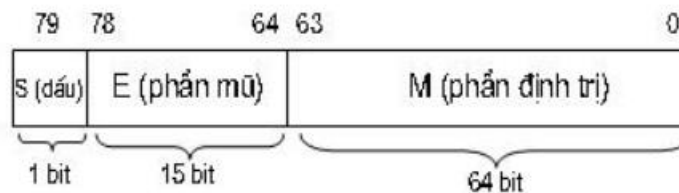
- Khi đó, 1 số thực bất kỳ được biểu diễn:

$$X = (-1)^S * 1,M * 2^{E - 1023}$$

- Dải giá trị biểu diễn:

$$2^{-1023} < |X| < 2^{1023},$$
$$\text{hay: } 10^{-308} < |X| < 10^{308}$$

Dạng kép mở rộng: 80 bit



- Khi đó, 1 số thực bất kỳ được biểu diễn:

$$X = (-1)^S * 1,M * 2^{E - 16383}$$

- Dải giá trị biểu diễn:

$$2^{-16383} < |X| < 2^{16383},$$
$$\text{hay: } 10^{-4932} < |X| < 10^{4932}$$

CHƯƠNG 5: HỆ THỐNG NHỚ

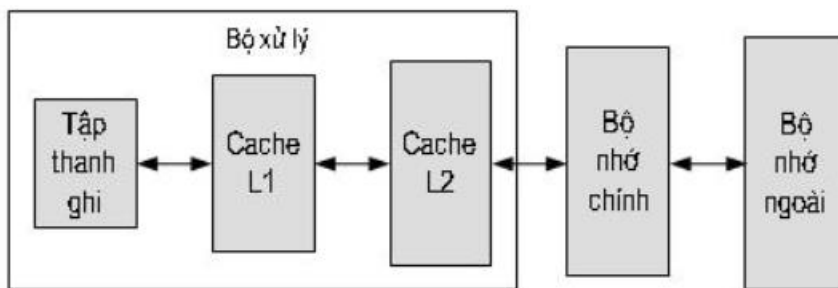
1. Các đặc trưng của hệ thống nhớ

- Vị trí:
 - Bên trong Bộ xử lý: Các thanh ghi
 - Bộ nhớ trong: Bộ nhớ chính

Bộ nhớ cache

- Bộ nhớ ngoài: Đĩa từ, băng từ
Đĩa quang
- Dung lượng
 - Độ dài ngăn nhớ
 - Số lượng ngăn nhớ
- Đơn vị truyền:
 - Truyền theo từ nhớ
 - Truyền theo khối nhớ
- Phương pháp truy nhập:
 - Truy nhập tuần tự (băng từ)
 - Truy nhập trực tiếp (đĩa từ, đĩa quang)
 - Truy nhập ngẫu nhiên (bộ nhớ trong)
 - Truy nhập liên kết (bộ nhớ cache)
- Kiểu vật lý của bộ nhớ:
 - Bộ nhớ bán dẫn
 - Bộ nhớ từ: băng từ và đĩa từ
 - Bộ nhớ quang: đĩa quang
- Các đặc trưng vật lý:
 - Bộ nhớ khả biến/ không khả biến
 - Bộ nhớ xóa được/ không xóa được

Phân cấp hệ thống nhớ



Từ trái sang phải:

- Dung lượng tăng dần
- Tốc độ trao đổi dữ liệu giảm dần
- Giá thành /1 bit giảm dần
- Tần suất BXL truy nhập giảm dần
- Mức trái chứa một phần dữ liệu của mức phải

2. Bộ nhớ bán dẫn

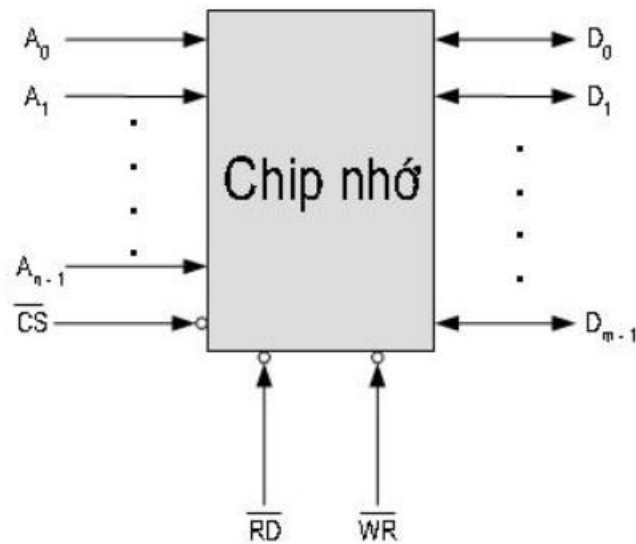
Kiểu bộ nhớ	Tiêu chuẩn	Khả năng xóa	Cơ chế ghi	Tính thay đổi
Read Only Memory (ROM)	Bộ nhớ chỉ đọc	Không xóa được	Mặt nạ	Không khả biến
Programmable ROM (PROM)				
Erasable PROM (EPROM)	Bộ nhớ hầu như chỉ đọc	Bảng tia cực tím, cả chip	Bảng điện	
Electrically Erasable PROM (EEPROM)		Bảng điện, mức từng byte		
Flash memory	Bộ nhớ đọc - ghi	Bảng điện, từng khối		Khả biến
Random Access Memory (RAM)		Bảng điện, từng byte		

- ROM:
 - Là loại bộ nhớ không khả biến
 - Không mất dữ liệu khi mất nguồn nuôi
 - Lưu trữ các thông tin:
 - Thư viện các ct con
 - Các ct hệ thống (BIOS)
 - Các bảng chức năng
 - Vi ct
- Các kiểu ROM:
 - ROM mặt nạ (ROM cố định):
 - Thông tin được ghi ngay khi sản xuất
 - Rất đắt
 - PROM:
 - Khi sản xuất chưa ghi dữ liệu
 - Cần thiết bị chuyên dụng để ghi bằng ct, chỉ ghi được 1 lần
 - EPROM:
 - Khi sản xuất chưa ghi dữ liệu
 - Cần thiết bị chuyên dụng để ghi bằng ct, ghi được nhiều lần
 - Trước khi ghi lại, phải xóa bằng tia cực tím
 - EEPROM:
 - Có thể ghi theo từng byte
 - Xóa bằng điện

- Ghi lâu hơn đọc
- Flash memory:
 - Ghi theo khối
 - Xóa bằng điện
- RAM
 - Bộ nhớ đọc/ ghi
 - Khả biến
 - Lưu trữ thông tin tạm thời
 - Mất dữ liệu khi mất nguồn nuôi
 - Có hai loại RAM:
 - SRAM (Static RAM)
 - Các bit được lưu trữ bằng các Flip – Flop
 - Không cần mạch làm tươi
 - Cấu trúc phức tạp hơn DRAM
 - Dung lượng nhỏ
 - Tốc độ nhanh hơn DRAM
 - Đắt hơn DRAM
 - Dùng làm bộ nhớ cache
 - DRAM (Dynamic RAM)
 - Các bit được lưu trữ trên tụ điện → cần phải có mạch làm tươi
 - Cấu trúc đơn giản
 - Dung lượng lớn
 - Tốc độ chậm hơn SRAM
 - Rẻ hơn SRAM
 - Dùng làm bộ nhớ chính

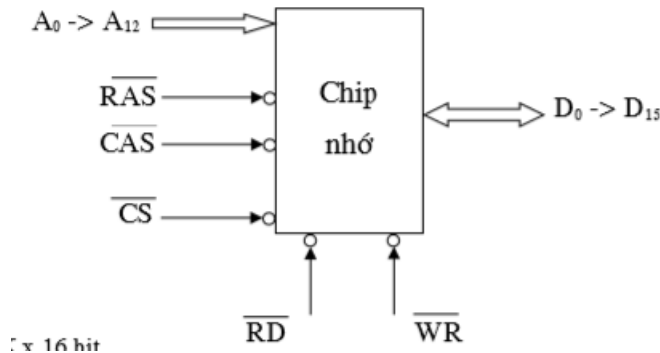
3. Tổ chức của chip nhớ

- SRAM



- Các đường địa chỉ: $A_0 \div A_{n-1} \rightarrow$ có 2^n ngăn nhớ.
- Các đường dữ liệu: $D_0 \div D_{m-1} \rightarrow$ độ dài ngăn nhớ là m bit.
- Dung lượng chip nhớ: $2^n \times m$ bit
- Các đường điều khiển:
 - Tín hiệu chọn chip: \overline{CS} (Chip Select)
 - Tín hiệu điều khiển đọc: $\overline{RD} / \overline{OE}$
 - Tín hiệu điều khiển ghi: $\overline{WR} / \overline{WE}$

- DRAM



- Dùng n đường địa chỉ dẫn kênh \rightarrow cho phép truyền 2^n bit địa chỉ
- Tín hiệu chọn địa chỉ hàng \overline{RAS} (Row Address Select)
- Tín hiệu chọn địa chỉ cột \overline{CAS} (Column Address Select)
- Dung lượng của DRAM: $2^{2n} \times m$ bit

4. Bộ nhớ chính

- Các đặc trưng cơ bản
 - Chứa các ct đang thực hiện và các dữ liệu đang đc sử dụng
 - Tồn tại trên mọi hệ thống máy tính
 - Được đánh giá địa chỉ trực tiếp bởi CPU: có nhiều ngăn nhớ, mỗi ngăn nhớ được gán một địa chỉ xác định
 - Việc quản lý logic BNC tùy thuộc vào HĐH
 - Về nguyên tắc, người lập trình có thể can thiệp trực tiếp vào toàn bộ BNC của máy tính
 - Các ngăn nhớ tổ chức theo byte

5. Bộ nhớ cache

- Cache có tốc độ nhanh hơn bộ nhớ chính
- Cache đc đặt giữa CPU và BNC nhằm tăng tốc độ truy nhập bộ nhớ của CPU
- Cache có thể được đặt trên chip CPU
- Các khái niệm:
 - Nguyên lý định vị tham số bộ nhớ:

- Định vị về thời gian: thông tin vừa truy nhập thì xác suất lớn ngay sau đó nó đc truy nhập lại.
- Định vị về không gian: thông tin vừa truy nhập thì xác suất lớn là ngay sau đó các mục lân cận sẽ được truy nhập.
- Trao đổi thông tin giữa cache và BNC
 - BNC được chia thành các block nhớ
 - Cache được chia thành các line nhớ
 - Kích thước line bằng kích thước block
- Số lượng line << Số lượng block
- Mỗi line trong cache được gắn thêm một tag để xác định block nào đang ở trong line
- Các kỹ thuật ánh xạ địa chỉ:
 - Ánh xạ trực tiếp
 - Mỗi Block của BNC chỉ đc ánh xạ vào một Line duy nhất
 - Vd1: Cho máy tính có dung lượng :
 - BNC = 128 MB, cache = 256 KB, line = 32 byte
 - Độ dài ngăn nhớ = 1 byte

Tìm dạng địa chỉ do BXL phát ra?

BL:

$$\text{BNC} = 128 \text{ MB} = 2^7 * 2^{20} = 2^{27} \text{ byte}$$

$$\text{Cache} = 256 \text{ KB} = 2^8 * 2^{10} = 2^{18} \text{ byte}$$

$$\text{Line} = 32 \text{ byte} = 2^5 \text{ byte} \rightarrow w = 5$$

$$\text{Số lượng line trong cache: } 2^{18}/2^5 = 2^{13} \rightarrow r = 13$$

$$\text{Số bit của tag: } 27 - 13 - 5 = 9$$

$$\text{Vậy dạng địa chỉ do BXL phát ra: } 9 + 13 + 5$$

- Vd2: Cho máy tính có dung lượng:
 - BNC = 256 MB, cache = 64 KB, line = 16 byte
 - Độ dài ngăn nhớ = 4 byte

Tìm dạng địa chỉ do BXL phát ra?

BL:

$$\text{BNC} = 256 \text{ MB} = 2^{28} \text{ byte} = 2^{28}/2^2 = 2^{26} \text{ ng/nhớ}$$

$$\text{Cache} = 64 \text{ KB} = 2^{16} \text{ byte} = 2^{16}/2^2 = 2^{14} \text{ ng/ nhớ}$$

$$\text{Line} = 16 \text{ byte} = 2^4/2^2 = 2^2 \text{ ng/ nhớ} \rightarrow w = 2$$

$$\text{Số lượng line trong cache: } 2^{14}/2^2 = 2^{12} \rightarrow r = 10$$

$$\text{Số bit của tag: } 26 - 10 - 2 = 14, s - r = 14$$

○ Ánh xạ liên kết hoàn toàn

- Mỗi Block trong BNC được ánh xạ vào một Line bất kỳ trong cacche

- Vd1: Cho máy tính có dung lượng:

- BNC = 256 MB, cache = 128 KB, line = 32 byte

- Độ dài ngăn nhớ = 4 byte

Tìm dạng địa chỉ do B

XL phát ra?

BL:

$$\text{Line size} = 32 \text{ byte} = 2^5 / 2^2 \rightarrow \text{Word} = 3 \text{ bit}$$

$$\text{BNC size} = 256 \text{ MB} = 2^8 \cdot 2^{20} = 2^{28}$$

$$\text{Tag} = 28 - 5 = 23 \text{ bit}$$

$$\rightarrow 23 + 3$$

○ Ánh xạ liên kết tập hợp

- Cache được chia thành nhiều Set, mỗi Set gồm nhiều Line liên tiếp

- Một Block của BNC chỉ được ánh xạ vào một Set duy nhất trong cache, nhưng được ánh xạ vào Line bất kỳ trong set đó.

- Vd1: Cho máy tính có dung lượng:

- BNC = 512 MB, cache = 128 KB, line = 32 byte

- Độ dài ngăn nhớ = 2 byte, set: 4 line

Tìm dạng địa chỉ do BXL phát ra?

BL:

$$\text{Line} = 2^5/2^1 \text{ B} \rightarrow w = 4$$

$$\text{BNC} = 512 \text{ MB} = 2^{29} \text{ B} = 2^{29}/2^1 = 2^{28}$$

$$\text{Cache} = 128 \text{ KB} = 2^{17} \text{ B} = 2^{17}/2^1 = 2^{16}$$

$$\text{Dung lượng set: } 2^2 * 2^4 = 2^6 \rightarrow 2^{16}/2^6 = 2^{10}$$

$$\rightarrow d = 10$$

$$\text{số bit của tag: } 28 - 10 - 4 = 14$$

Các thuật toán thay thế

- Kỹ thuật ánh xạ trực tiếp: Không thay được
- Hai kỹ thuật ánh xạ liên kết: có 4 thuật toán
 - **Random**: thay ngẫu nhiên một Block cũ nào đó
 \Rightarrow Dễ thực hiện, nhanh nhất, tỉ lệ cache hit thấp.
 - **FIFO** (First In - First Out): thay Block ở đầu tiên trong số các Block đang có trong cache
 \Rightarrow tỉ lệ cache hit không cao
 - **LFU** (Least Frequently Used): thay Block được dùng với tần suất ít nhất \Rightarrow tỉ lệ cache hit tương đối cao
 - **LRU** (Least Recently Used): thay Block được dùng gần đây ít nhất \Rightarrow tỉ lệ cache hit cao

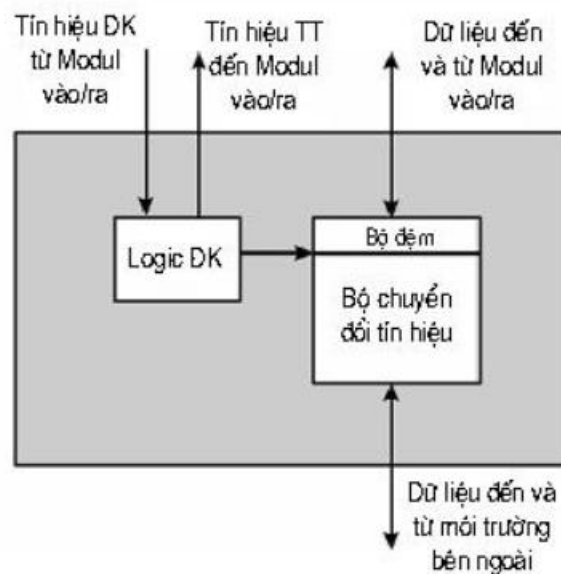
Hoạt động của cache

- **Đọc**:
 - Nếu cache hit: đọc gần nhớ từ cache
 - Nếu cache miss: thay Block \Rightarrow cache hit
- **Ghi**:
 - Nếu cache hit: có 2 phương pháp:
 - Write through: ghi dữ liệu vào cả cache và cả BNC
 \Rightarrow không cần thiết, tốc độ chậm, mạch đơn giản.
 - Write back: chỉ ghi vào cache, khi nào Block (trong cache) được ghi bị thay đi \rightarrow ghi vào BNC
 \Rightarrow tốc độ nhanh, mạch phức tạp.
 - Nếu cache miss: thay Block \Rightarrow cache hit

CHƯƠNG 6: HỆ THỐNG VÀO/ RA

- Tại sao cần Modul vào/ra?
 - Không thể nối trực tiếp các thiết bị ngoại vi với bus hệ thống, vì:
 - BXL không thể điều khiển đc tất cả TBNV
 - Tốc độ trao đổi dữ liệu khác nhau
 - Khuôn dạng dữ liệu khác nhau
 - Tất cả có tốc độ chậm hơn BXL và RAM
 - Chức năng của Modul vào/ra:
 - Nối ghép với BXL và hệ thống nhớ
 - Nối ghép với một hoặc nhiều TBNV
- Các thiết bị ngoại vi
 - Chức năng: chuyển đổi dữ liệu giữa bên trong và bên ngoài máy tính
 - Phân loại:
 - TBNV giao tiếp người – máy: màn hình, bàn phím, máy in,..
 - TBNV giao tiếp máy – máy: các thiết bị theo dõi và kiểm tra
 - TBNV truyền thông: modem, card giao tiếp mạng (NIC)

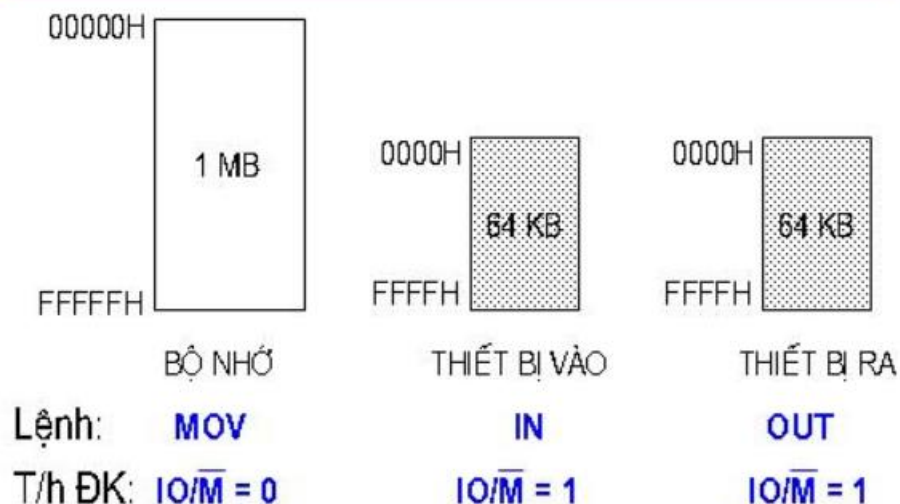
Sơ đồ khối của TBNV



- Các thành phần cơ bản của TBNV:

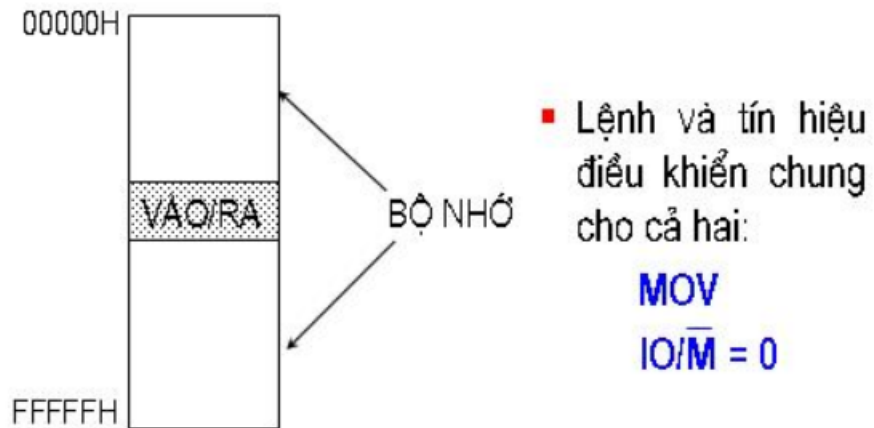
- Bộ chuyển đổi tín hiệu
- Logic điều khiển
- Bộ đếm
- Modul vào/ra:
 - Chức năng:
 - Điều khiển và định thời gian
 - Trao đổi thông tin với BXL
 - Trao đổi thông tin với TBNV
 - Bộ đếm dữ liệu
 - Phát hiện lỗi
 - Thành phần cơ bản:
 - Thanh ghi dữ liệu
 - Các cổng vào/ra
 - Thanh ghi điều khiển/ trạng thái
 - Logic điều khiển
- Các pp địa chỉ hóa cổng vào/ra
 - Vào/ra cách biệt (isolated IO)
 - Đặc điểm: Không gian địa chỉ cổng vào/ra nằm ngoài không gian địa chỉ bộ nhớ
 - Cách truy nhập:
 - Cần có tín hiệu phân biệt truy nhập cổng vào/ra hay truy nhập bộ nhớ
 - Sử dụng lệnh vào/ra chuyên dụng: IN, OUT

Ví dụ đối với BXL 8088



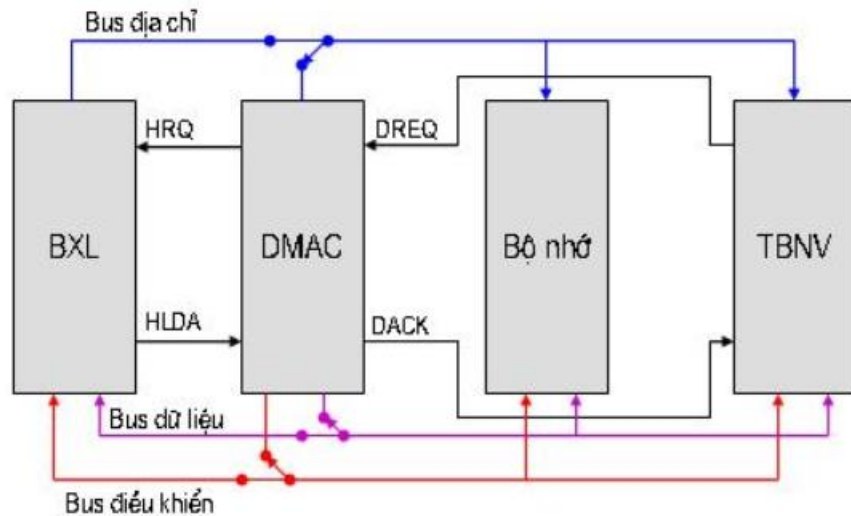
- Vào/ra theo bản đồ bộ nhớ (memory mapped IO)
 - Đặc điểm: Không gian địa chỉ cổng vào/ra nằm trong không gian địa chỉ bộ nhớ
 - Cách truy nhập:
 - Dùng chung tín hiệu như truy nhập bộ nhớ
 - Dùng chung lệnh trao đổi dữ liệu với bộ nhớ

Ví dụ đối với BXL 8088



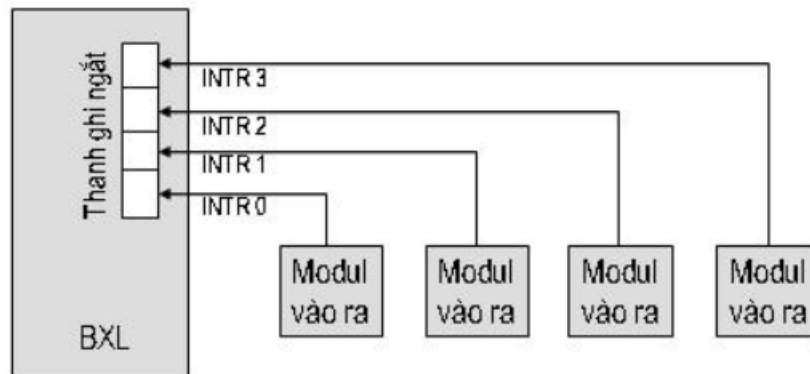
- Các pp điều khiển vào/ra
 - Vào/ra bằng ct
 - Nguyên tắc chung:
 - Sử dụng lệnh vào/ra trong ct để trao đổi dữ liệu với cổng vào/ra
 - Khi BXL thực hiện ct, gặp lệnh vào/ra thì BXL điều khiển trao đổi dữ liệu với TBNV
 - Vào/ra bằng ngắt
 - Vào/ra bằng DMA
 - DMAC điều khiển vào/ra không qua BXL (CPU)
 - Đây là pp có tốc độ trao đổi dữ liệu nhanh
 -

Sơ đồ

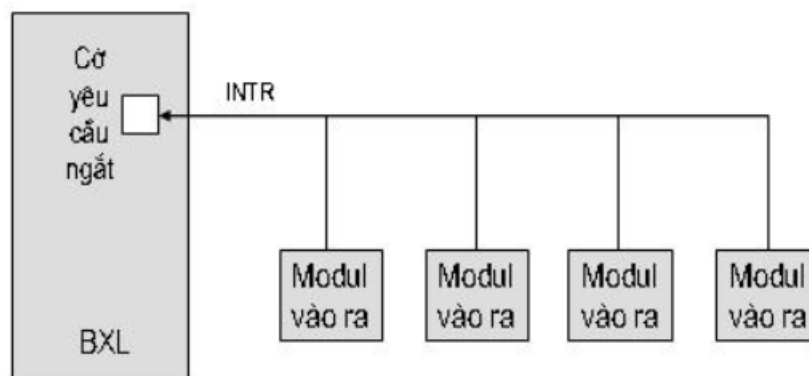


- Quy trình hoạt động
 - B1: TBNV gửi tín hiệu DREQ (Dma REQuest) tới DMAC
 - B2: DMAC gửi tín hiệu HRQ (Hold ReQuest) để xin dùng các đường bus
 - B3: BXL sẽ thực hiện xong chu kỳ bus hiện tại và trả lời đồng ý bằng việc gửi tín hiệu HLDA (HoLD Acknowledge) tới DMAC
 - B4: DMAC gửi tín hiệu DACK (Dma ACKnowledge) tới TBNV báo chuẩn bị truyền dữ liệu
 - B5: DMAC thực hiện điều khiển quá trình truyền dữ liệu giữa bộ nhớ và TBNV
 - B6: DMAC thực hiện xong công việc, nó bỏ kích hoạt tín hiệu HRQ. Hệ thống trở lại bình thường.
- Các kiểu thực hiện DMA:
 - DMA truyền theo khối: sd bus để truyền cả khối dữ liệu
 - DMA ăn trộm chu kỳ: ép buộc BXL treo tạm thời chu kỳ để thực hiện truyền một byte dữ liệu
 - DMA trong suốt: nhận biết những chu kỳ nào BXL k dùng bus thì lấy bus để tranh thủ truyền 1 byte dữ liệu
- Phân loại ngắt
 - Ngắt cứng
 - Ngắt cứng NMI

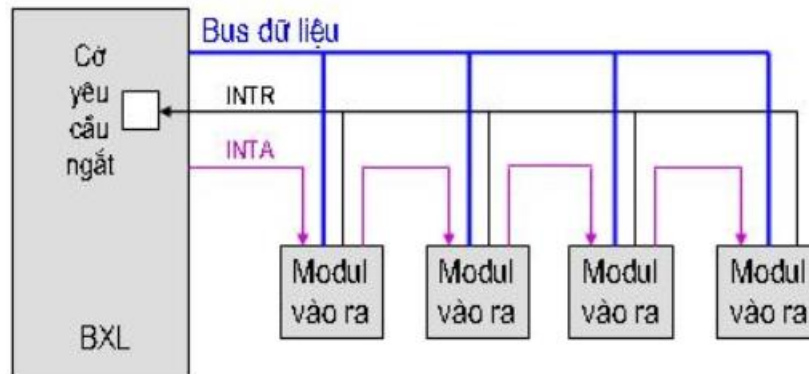
- Ngắt cứng MI là ngắt cứng chắn được, ngắt cứng MI dùng để trao đổi dữ liệu với TBNV
- Ngắt mềm, do lệnh ngắt nằm trong ct sinh ra
- Ngắt ngoại lệ, là ngắt do lỗi ct sinh ra
 - Lệnh chia cho 0 sinh ra ngắt ngoại lệ
 - Tràn số sinh ra ngắt ngoại lệ
 - Lỗi bộ nhớ sinh ra ngắt ngoại lệ
- Các pp xác định modul ngắt
 - Nhiều đường yêu cầu ngắt



- BXL phải có các đường yêu cầu ngắt khác nhau cho mỗi modul vào/ra
- Hạn chế số lượng thiết bị
- Kiểm tra vòng bằng phần mềm

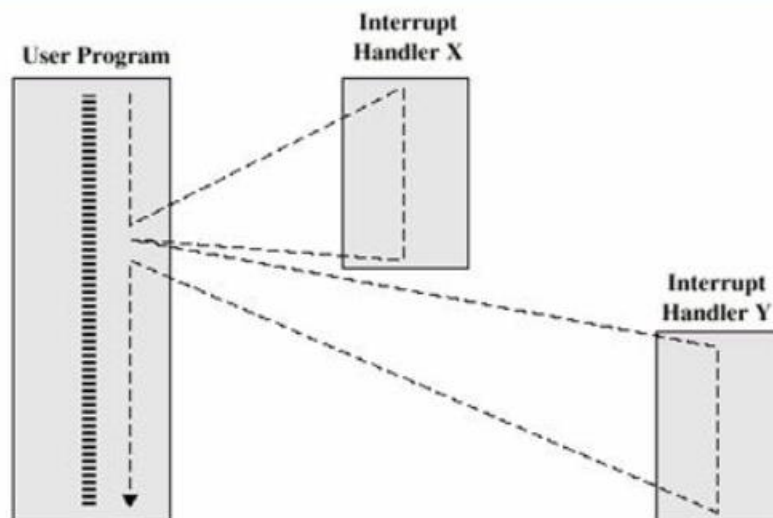


- BXL thực hiện phần mềm kiểm tra từng modul
- Tốc độ chậm
- Kiểm tra vòng bằng phần cứng



- BXL phát tín hiệu chấp nhận ngắt đến chuỗi các modul vào/ra
- Modul sẽ đáp ứng bằng cách đặt vectơ ngắt lên bus dữ liệu
- BXL sử dụng vectơ để xác định CTC điều khiển ngắt
- Chiếm bus
 - Modul vào ra cần chiếm bus trước khi nó phát tín hiệu yêu cầu ngắt
 - Vd: PCI, SCSI

Ngắt tuần tự



Ngắt lồng nhau

