

Link fanpage của huy nè : > [lạc lối vào sao lùn nâu siêu khổng lồ cùng huy](#)

Tiện thể follow luôn IG của momo nha, tym 2 bài viết đầu tiên nữa : >

<https://www.instagram.com/momo/?hl=vi>

38 trang LT + 11 chang BT

**NOTE: đang update thêm bài tập ở cuối file**

## 1. Tổng quan hệ thống máy tính

- a. Các thành phần cơ bản
  - i. Bộ xử lý trung tâm (CPU)
    - 1. **Điều khiển** hoạt động mt
    - 2. **Xử lý dữ liệu**
  - ii. Bộ nhớ chính (Main Memory)
    - 1. Chứa các **chương trình đang thực hiện**
  - iii. Hệ thống vào-ra (Input/Output)
    - 1. **Trao đổi** thông tin giữa máy tính với bên ngoài
  - iv. Bus hệ thống (System bus)
    - 1. **Kết nối và vận chuyển** thông tin
- b. CPU
  - i. Nguyên tắc hoạt động cơ bản:
    - 1. **theo chương trình** nằm trong bộ nhớ chính.
  - ii. Là thành phần **nhANH NHẤT** trong hệ thống
  - iii. Bao gồm:
    - 1. Đơn vị điều khiển (Control unit – CU)
      - a. **Điều khiển** hđ của mt theo **ctr đã định sẵn**
    - 2. Đơn vị số học và logic (Arithmetic and Logic Unit – ALU)
      - a. **Thực hiện** các phép toán **số học** và phép toán **logic**
    - 3. Tập thanh ghi (Register file – RF)
      - a. **Chứa các thông tin phục vụ** cho hđ of **CPU**

c. Bộ nhớ máy tính

i. Nhớ **ctr&dl** (**nhị phân**)

1. \* bộ nhớ chính thì chứa ctr **ĐANG THỰC HIỆN**

ii. Thao tác: đọc , ghi

iii. Thành phần chính:

d.

1. Bộ nhớ chính

a. All máy tính

**b. Lệnh và data of ctr ĐANG THỰC HIỆN**

c. Use **bộ nhớ bán dẫn**

d. Tổ chức -> **ngăn nhớ được đánh địa chỉ**  
(cho từng byte nhớ)

**e. Nội dung ngăn nhớ can change but địa chỉ vật lý can not**

f. CPU muốn đọc/ghi -> **cần biết địa chỉ** ngăn nhớ đó

2. Bộ nhớ đệm

a. Tốc độ nhanh (đặt đệm giữa CPU và bộ nhớ chính)

b. Dung lượng < **bộ nhớ chính**

c. Bộ nhớ **bán dẫn tốc độ nhanh**

**d. Chia thành 1 số mức (L1,L2,L3)**

**e. Tích hợp** trên cùng **chip bộ xử lý**

f. Có thể Có or không

3. Thiết bị lưu trữ

a. Bộ nhớ ngoài

b. Chức năng và đặc điểm:

i. Lưu giữ tài nguyên **phần mềm**

ii. Connect vs HT **dưới dạng i/o**

iii. Dung lượng lớn

**iv. Speed chậm**

c. Các loại:

i. Bộ nhớ từ: ổ đĩa cứng HDD

ii. Bộ nhớ bán dẫn: ổ thể rắn SSD, ổ nhớ flash, thẻ nhớ

iii. Bộ nhớ quang: CD, DVD

4. Hệ thống vào ra

a. Thao tác: vào dl, ra dl

b. Thành phần chính:

i. Thiết bị vào ra

1. **Thiết bị ngoại vi** (Peripherals)
2. **Chuyển đổi** dữ liệu giữa in và out mt
3. Các loại thiết bị:
  - a. in
  - b. out
  - c. lưu trữ
  - d. truyền thông

ii. Module vào ra

1. **Nối ghép** các thiết bị io vs mt
2. Có 1 or vài cổng io (io port)
3. Mỗi **cổng io** đánh **1 địa chỉ xác định**
4. Thiết bị io **connect và trao đổi** data vs mt **qua port io**
5. CPU muốn trao đổi -> biết **địa chỉ port io**

ii. Hoạt động cơ bản

1. 3 cái:

- a. Thực hiện chương trình
- b. HĐ ngắt
- c. HĐ io

2. Thực hiện chương trình

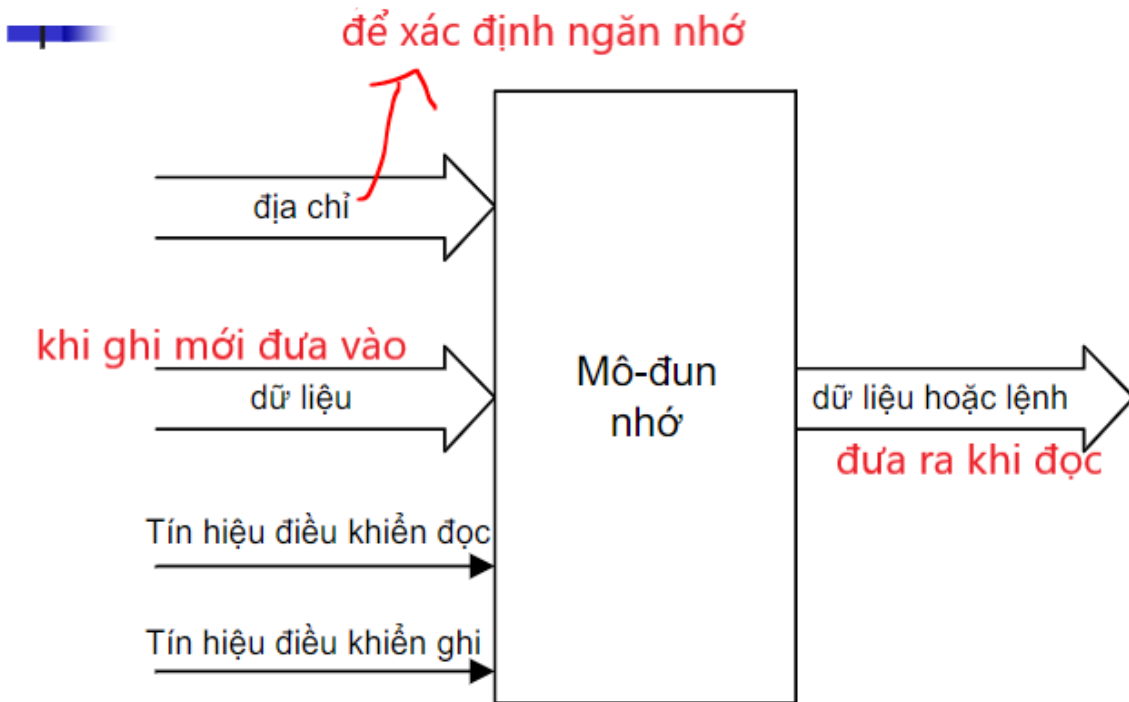
- a. Cơ bản of mt
- b. **Lặp đi lặp lại** chu trình: **nhận và thực hiện lệnh**
- c. Hở thực hiện ctr dừng nếu
  - i. Thực hiện lệnh lỗi
  - ii. Gặp lệnh dừng
  - iii. Tắt máy
- d. Nhận lệnh
  - i. Bắt đầu 1 chu trình -> CPU **nhận lệnh từ bộ nhớ chính**
  - ii. **Bộ đếm** chương trình PC (Program Counter): thanh ghi of CPU -> **giữ địa chỉ lệnh nhận**
  - iii. CPU phát ra địa chỉ từ bộ đếm => **tìm ra ngăn nhớ** của lệnh
  - iv. Lệnh được **đọc từ bộ nhớ** => **đưa vào thanh ghi lệnh IR** (Instruction Register)

- v. Nhận xong => nội dung **PC tự tăng** để trở đến lệnh tiếp
- e. Thực hiện lệnh
  - i. Bộ xử lý :
    1. **giải mã lệnh** đã được nhận
    2. **phát tín hiệu** điều khiển thực hiện thao tác mà lệnh yêu cầu
  - ii. các kiểu thao tác cơ bản:
    1. **trao đổi data** CPU vs BNC or CPU vs io module
    2. **Thực hiện các phép toán** số học hoặc phép toán logic với các dữ liệu
    3. **Chuyển điều khiển** trong chương trình: rẽ nhánh hoặc nhảy đến vị trí khác

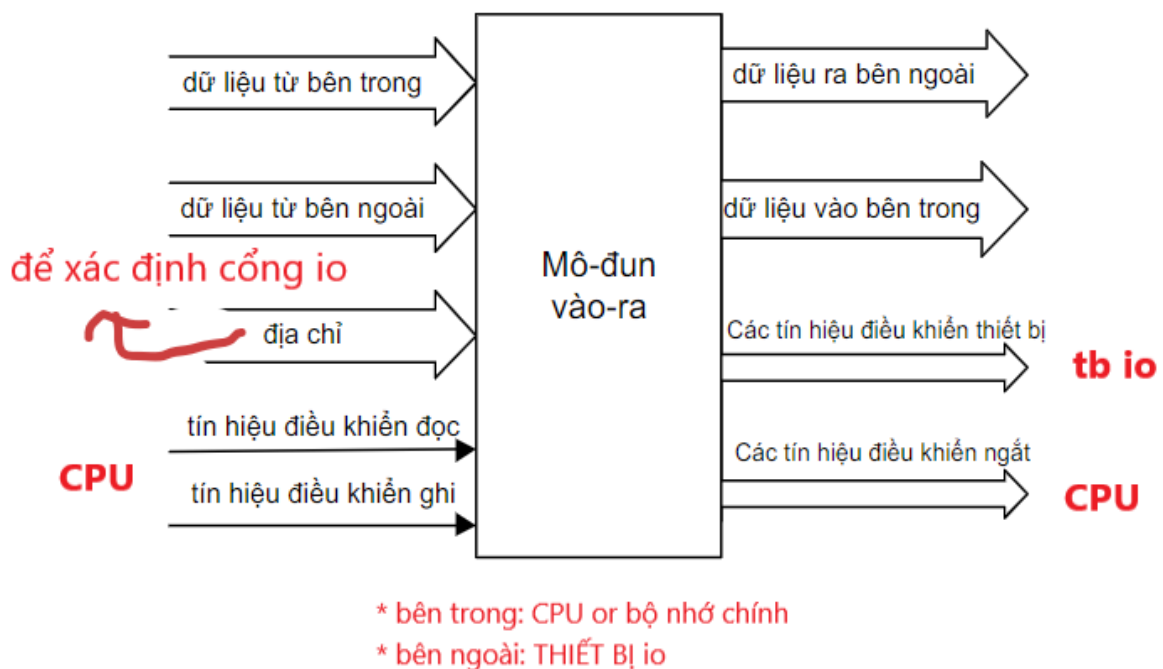
### 3. Ngắt

- a. Khái niệm
  - i. cơ chế cho phép CPU **tạm dừng chương trình đang thực hiện để chuyển sang** thực hiện một **chương trình con** có sẵn trong bộ nhớ.
- b. Các loại ngắt:
  - i. Biệt lệ (exception):
    1. gây ra do lỗi khi thực hiện chương trình
    2. VD: tràn số, mã lệnh sai, ...
  - ii. Ngắt từ bên ngoài (external interrupt):
    1. do thiết bị vào-ra (thông qua mô-đun vào-ra) gửi tín hiệu ngắt đến CPU để yêu cầu trao đổi dữ liệu
- c. Hết ngắt từ bên ngoài
  - i. Done 1 lệnh -> bộ xử lý ktra signal ngắt
  - ii. K có ngắt -> bxl nhận lệnh tiếp theo
  - iii. Có ngắt
    1. Tạm dừng ctr đang thực hiện
    2. Cắt ngữ cảnh (info ctr bị ngắt)
    3. Bộ đếm ctr PC trở đến ctr con xử lý ngắt tương ứng
    4. Thực hiện ctr con xử lý ngắt

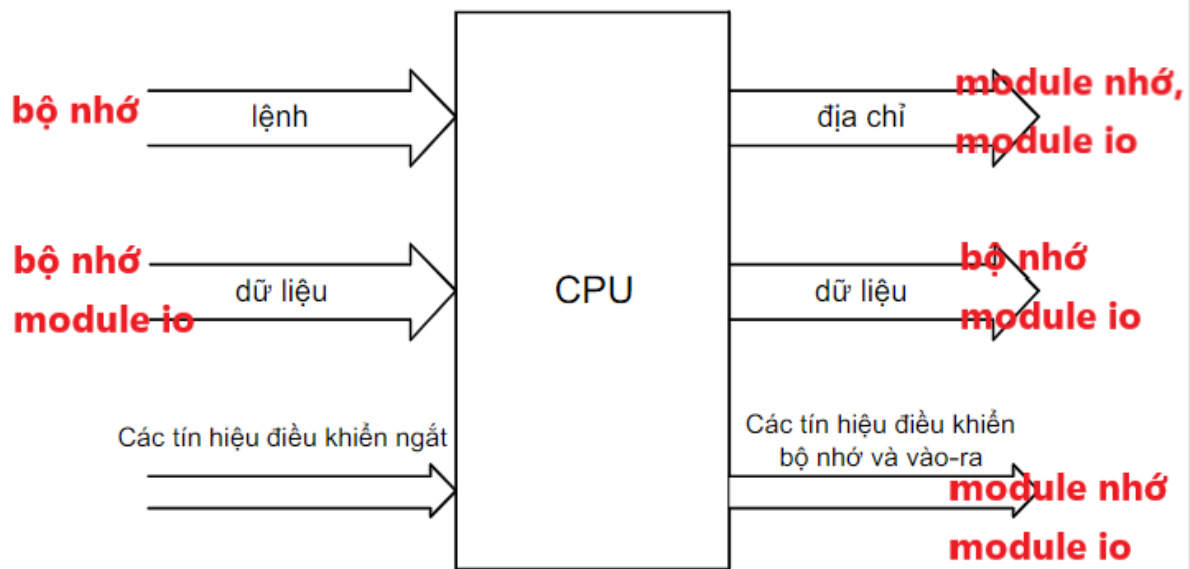
5. Khôi phục ngữ cảnh => tiếp tục thực hiện ctr đang bị dừng
- d. Xử lý với nhiều tín hiệu yêu cầu ngắt
  - i. XL ngắt tuần tự (**nói chung cái này là tuần tự mà đợi, chỉ 1 cái thực hiện thôi**)
    1. Khi một ngắt đang được thực hiện, các ngắt khác bị cấm (disabled interrupt)
    2. Bộ xử lý sẽ bỏ qua các yêu cầu ngắt tiếp theo
    3. Các yêu cầu ngắt tiếp theo vẫn đang đợi và được kiểm tra sau khi ngắt hiện tại được xử lý xong
  - ii. XL ngắt ưu tiên
    1. Các ngắt được định nghĩa mức ưu tiên khác nhau
    2. Ngắt có mức ưu tiên **thấp hơn có thể bị ngắt bởi** ngắt có mức ưu tiên **cao hơn**
    3. **Xảy ra ngắt lồng nhau**
4. Hoạt động vào-ra
  - a. Trao đổi data giữa MODULE io vs inside mt
  - b. Các kiểu hđ io:
    - i. CPU vs module by lệnh io trong ctr
    - ii. CPU trao quyền cho phép module io trao đổi trực tiếp với bộ nhớ chính (DMA)
- e. Bus máy tính
  - i. Luồng thông tin trong máy tính
    1. Các module trong máy tính cần được kết nối CPU
      - a. Module nhớ
      - b. Module io
    2. Kết nối module nhớ



### 3. Kết nối module vào ra



### 4. Kết nối CPU



## ii. Cấu trúc bus cơ bản

1. Bus: tập hợp các **đường kết nối** để **vận chuyển** thông tin giữa các **mô-đun** của máy tính với nhau.
2. Các bus chức năng:
  - a. Bus địa chỉ
  - b. Bus dữ liệu
  - c. Bus điều khiển
3. Độ rộng bus: là **số đường dây** của bus có thể **truyền** các bit thông tin **đồng thời** (chỉ dùng cho bus địa chỉ và bus dữ liệu)
4. Bus địa chỉ:
  - a. Vận chuyển địa chỉ => xác định vị trí ngăn nhớ or io port
  - b. Độ rộng

$A_{N-1}, A_{N-2}, \dots, A_2, A_1, A_0$

- i. N bit:
- ii. Số lượng địa chỉ tối đa được sử dụng là:  $2^N$  địa chỉ (gọi là không gian địa chỉ)
- iii. Min: 000...00
- iv. Max: 111...111
5. Bus dữ liệu
  - a. Chức năng: vận chuyển
    - i. Lệnh: bộ nhớ => CPU
    - ii. Data: giữa các thành phần mt
  - b. Độ rộng
    - i. số bit được truyền đồng thời

- M bit:  $D_{M-1}, D_{M-2}, \dots, D_2, D_1, D_0$
- M thường là 8, 16, 32, 64 bit

ii.

## 6. Bus điều khiển

a. Chức năng: vận chuyển các tín hiệu điều khiển

b. Các loại tín hiệu điều khiển:

- i. Các tín hiệu điều khiển đọc/ghi
- ii. Các tín hiệu điều khiển ngắt
- iii. Các tín hiệu điều khiển bus

c. Ví dụ (dài quá, mong k vào)

i. Các tín hiệu (phát ra từ CPU) điều khiển đọc/ghi:

1. Memory Read(MEMR): Tín hiệu điều khiển đọc dữ liệu **từ một ngăn nhớ** có địa chỉ xác định đưa lên bus dữ liệu.
2. Memory Write(MEMW): Tín hiệu điều khiển ghi dữ liệu có sẵn trên bus dữ liệu **đến một ngăn nhớ** có địa chỉ xác định.
3. I/O Read(IOR): Tín hiệu điều khiển đọc dữ liệu **từ một cổng vào-ra** có địa chỉ xác định đưa lên bus dữ liệu.
4. I/O Write(IOW): Tín hiệu điều khiển ghi dữ liệu có sẵn trên bus dữ liệu **ra một cổng** có địa chỉ xác định.

ii. Các tín hiệu điều khiển ngắt:

1. Interrupt Request(INTR): Tín hiệu **từ bộ điều khiển vào-ra gửi** đến **yêu cầu** ngắt CPU để trao đổi vào-ra. Tín hiệu INTR có thể bị che.
2. Interrupt Acknowledge(INTA): Tín hiệu phát ra **từ CPU báo** cho bộ điều khiển vào-ra **biết CPU chấp nhận** ngắt để trao đổi vào-ra.



3. *Non Maskable Interrupt(NMI)*: tín hiệu ngắt **không che** được gửi đến ngắt **CPU**.

4. *Reset*: Tín hiệu từ bên ngoài gửi đến CPU và các thành phần khác để **khởi động lại** máy tính.

iii. Các tín hiệu điều khiển bus:

1. *Bus Request(BRQ)* : Tín hiệu từ **mô-đun vào-ra** gửi đến **yêu cầu CPU chuyển nhượng** quyền sử dụng bus.

2. *Bus Grant(BGT)*: Tín hiệu phát ra từ **CPU chấp nhận** chuyển nhượng quyền sử dụng bus cho mô-đun vào-ra.

3. *Lock/Unlock*: Tín hiệu **cấm/cho-phép** xin chuyển nhượng bus.

## 7. Phân cấp bus

a. Đơn bus: all module connect bus chung

i. Bus chỉ phục vụ **1 yêu cầu trao đổi data** at 1 thời điểm => độ trễ lớn

ii. Bus phải có speed = speed bus of module **nhANH NHẤT** trong hệ thống

b. Đa bus: Phân cấp thành nhiều bus cho các module khác nhau và có speed khác nhau

i. Bus của bộ xử lý

ii. Bus của RAM

iii. Các bus vào-ra

## 8. Kết nối điểm-điểm

a. Khắc phục nhược điểm của bus dùng chung

## 9. Một số bus điển hình trong máy tính (khó nhớ quá)

a. *QPI (Quick Path Interconnect)*

b. *PCI bus (Peripheral Component Interconnect)*: bus vào-ra đa năng

c. *PCIe: (PCI express)* kết nối điểm-điểm đa năng tốc độ cao

d. *SATA (Serial Advanced Technology Attachment)*: Bus kết nối với ổ đĩa cứng hoặc ổ đĩa CD/DVD

e. *USB (Universal Serial Bus): Bus nối tiếp đa năng*

## 2. Bộ nhớ máy tính

### a. Tổng quan

#### i. Đặc trưng

##### 1. Vị trí

- a. Bên trong CPU (tập thanh ghi)
- b. Bộ nhớ trong
  - i. Chính hệ
  - ii. Đệm
- c. Bộ nhớ ngoài
  - i. Các thiết bị lưu trữ

##### 2. Dung lượng

- a. Độ dài từ nhớ (tính = bit)
- b. Số lượng từ nhớ

##### 3. Đơn vị truyền

- a. Từ nhớ
- b. Khối nhớ

##### 4. Phương pháp truy nhập

- a. Truy nhập tuần tự (băng từ)
- b. Truy nhập trực tiếp (các loại đĩa)
- c. Truy nhập ngẫu nhiên (bộ nhớ bán dẫn)
- d. Truy nhập liên kết (cache)

##### 5. Hiệu năng

- a. Thời gian truy nhập
- b. Chu kỳ nhớ
- c. Tốc độ truyền

##### 6. Kiểu vật lý

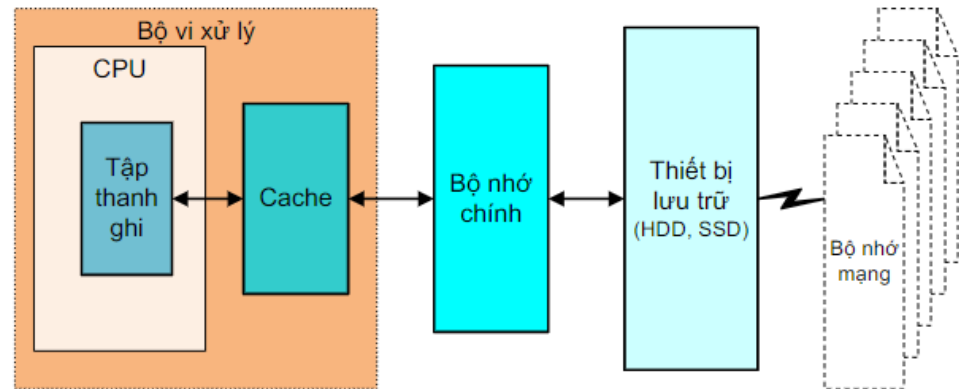
- a. Bán dẫn
- b. Từ
- c. Quang

##### 7. Các đặc tính vật lý

- a. Khả biến/ không khả biến
- b. Xóa được/ không xóa được

##### 8. Tổ chức

#### ii. Phân cấp bộ nhớ



Từ trái sang phải:

- dung lượng tăng dần
- tốc độ giảm dần
- giá thành cùng dung lượng giảm dần

### iii. Công nghệ bộ nhớ

Công nghệ bộ nhớ	Thời gian truy nhập	Giá thành/GiB (2012)
SRAM	0,5 – 2,5 ns	\$500 – \$1000
DRAM	50 – 70 ns	\$10 – \$20
Flash memory	5.000 – 50.000 ns	\$0,75 – \$1
HDD	5 – 20 ms	\$0,05 – \$0,1

#### ■ Bộ nhớ lý tưởng

- Thời gian truy nhập như SRAM
- Dung lượng và giá thành như ổ đĩa cứng

### iv. Nguyên lý cục bộ hoá tham chiếu bộ nhớ

1. Trong một khoảng thời gian đủ nhỏ CPU thường **chỉ tham chiếu các thông tin trong một khối nhớ cục bộ**
2. Ví dụ:
  - a. Cấu trúc chương trình tuần tự

- b. Vòng lặp có thân nhỏ
- c. Cấu trúc dữ liệu mảng

b. Bộ nhớ chính

## 1. Bộ nhớ bán dẫn

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

### i. ROM

#### 1. Lưu:

- a. Thư viện các chương trình con
- b. Các chương trình điều khiển hệ thống (BIOS)
- c. Các bảng chức năng
- d. Vi chương trình

#### 2. Các kiểu:

- a. ROM mặt nạ: thông tin được ghi khi sản xuất
- b. PROM (Programmable ROM):
  - i. Cần thiết bị chuyên dụng để ghi
  - ii. Chỉ ghi được một lần
- c. EPROM (Erasable PROM):
  - i. Cần thiết bị chuyên dụng để ghi
  - ii. Xóa được bằng tia tử ngoại
  - iii. Ghi lại được nhiều lần
- d. EEPROM (Electrically Erasable PROM)
  - i. Có thể ghi theo từng byte
  - ii. Xóa bằng điện

### ii. Bộ nhớ flash

1. Ghi theo khối
2. Xóa bằng điện
3. Dung lượng lớn

iii. RAM

1. Đọc-ghi
2. Khả biến
3. Lưu trữ thông tin tạm thời
4. 2 loại : SRAM DRAM

a. SRAM (tĩnh)

- i. Flip-flop => ổn định
- ii. Cấu trúc phức tạp
- iii. Dung lượng chip nhỏ
- iv. Tốc độ nhanh
- v. Đắt
- vi. Dùng làm bộ nhớ cache

b. DRAM (động)

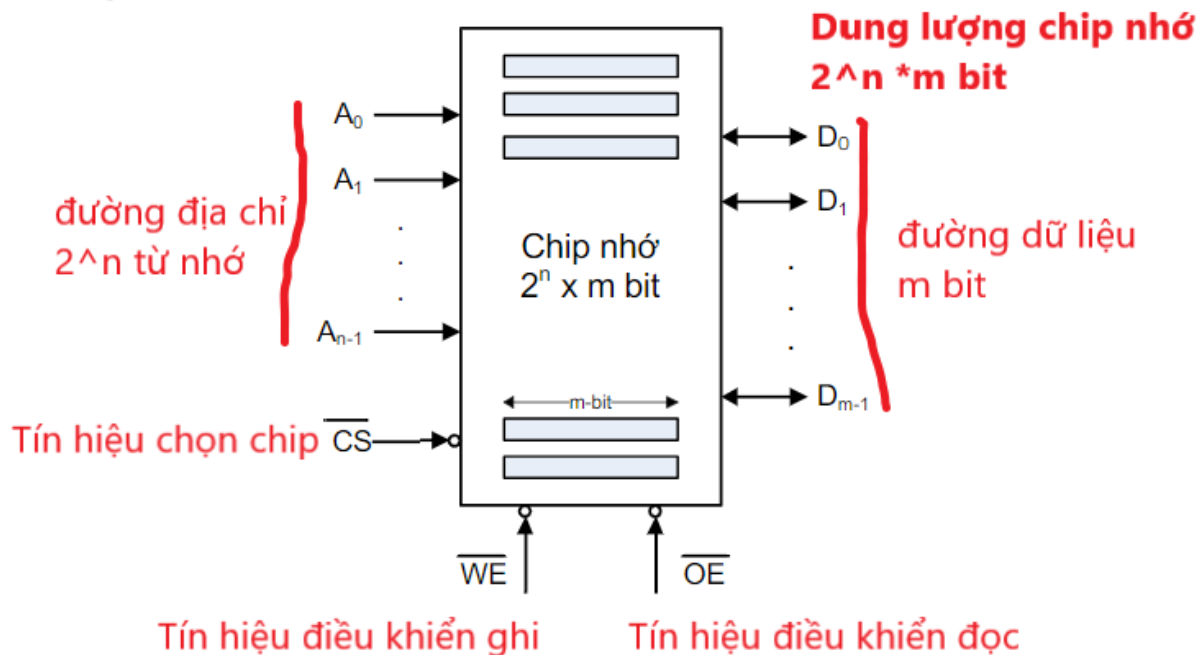
- i. Bit lưu trên tụ điện => cần có mạch làm tươi
- ii. Đơn giản
- iii. Dung lượng lớn
- iv. Tốc độ chậm
- v. Rẻ
- vi. Dùng làm bộ nhớ chính

c. 1 số DRAM tiên tiến:

- i. Cải tiến để tăng tốc độ
- ii. Synchronous DRAM (SDRAM): làm việc được đồng bộ bởi xung clock
- iii. DDR-SDRAM (Double Data Rate SDRAM)
- iv. DDR3, DDR4

d. Tổ chức của chip nhớ

## Sơ đồ cơ bản của chip nhớ



### e. Tổ chức của DRAM

- i. Dùng  $n$  đường địa chỉ dòng kênh  
=> cho phép truyền  $2n$  bit địa chỉ
- ii. Tín hiệu chọn địa chỉ hàng RAS (Row Address Select)
- iii. Tín hiệu chọn địa chỉ cột CAS (Column Address Select)

iv. **Dung lượng của DRAM =  $2^{(2n)} \times m$  bit (CẢN THẬN NHẦM - only DRAM mới  $2n$ )**

### iv. Đặc trưng của bộ nhớ chính

1. Chứa các **chương trình đang thực hiện** và các **dữ liệu đang được sử dụng**
2. Tồn tại trên **mọi hệ thống máy tính**
3. Bao gồm các **ngăn nhớ được đánh địa chỉ** trực tiếp bởi CPU
4. Dung lượng của bộ nhớ chính **nhỏ hơn** không gian địa chỉ bộ nhớ mà **CPU quản lý**.
5. Việc quản lý logic bộ nhớ chính **tuỳ thuộc vào hệ điều hành**
6. Độ rộng của bus dữ liệu để trao đổi với bộ nhớ:  $m = 8, 16, 32, 64, 128 \dots$  bit

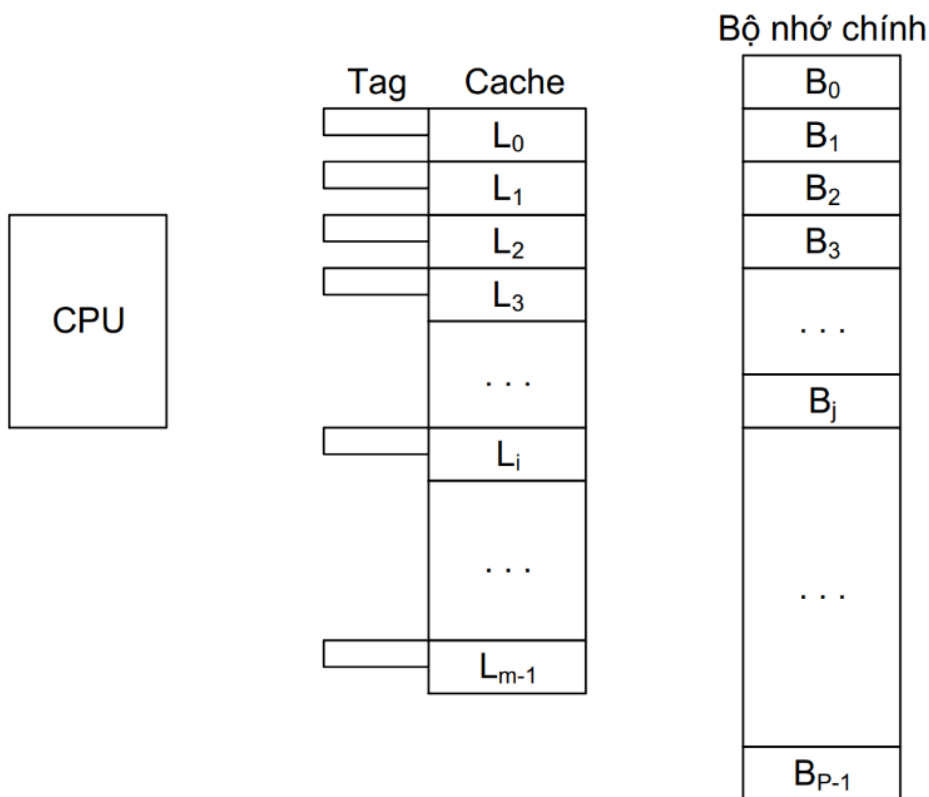
7. Các ngăn nhớ được **tổ chức theo byte**  
 => **tổ chức bộ nhớ vật lý khác nhau**

c. **Bộ nhớ đệm**

i. Nguyên tắc chung

1. Nhanh hơn BNC
2. Đặt giữa CPU và BNC
3. Có thể đặt trên chip CPU
4. Ví dụ: CPU yêu cầu nội dung => check cache trước  
 => có thì lấy luôn, không thì load lên cache rồi load từ cache lên CPU

ii. Cấu trúc



1. BNC: các block, Cache: các Line => **kích thước B và L bằng nhau**
2. Mỗi Line có 1 Tag => cho biết block nào đang ở line đó (nó là T bit địa chỉ cao của Block được nạp)
3. Block nạp vào Line => cập nhật Tag
4. Cách hoạt động:
  - a. CPU yêu cầu nội dung
  - b. Check Cache: có thì cache hit (lấy luôn), không có thì cache miss\*



c. Nếu cache miss => nạp block vào line và cập nhật tag, rồi đưa nội dung line đó cho CPU

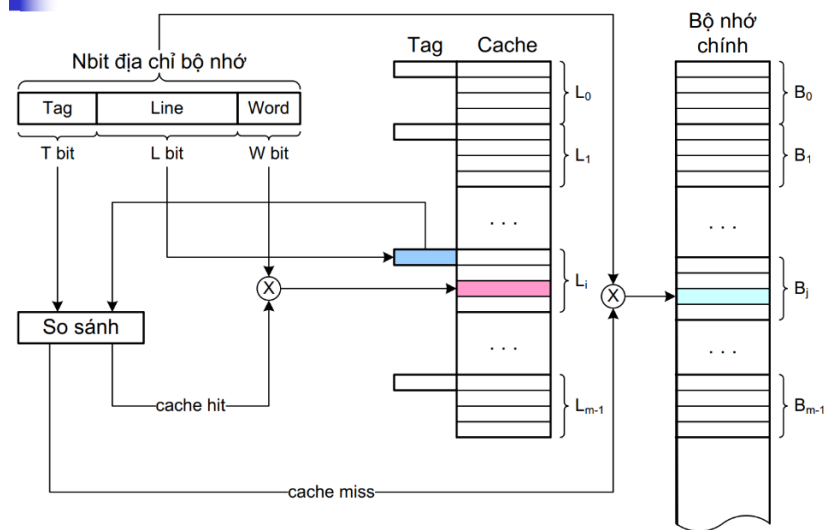
iii. Cách phương pháp ánh xạ

1. Ánh xạ trực tiếp

a. 1 Block chỉ có thể nạp 1 Line

$B_j$  chỉ có thể nạp vào  $L_{j \bmod m}$   
 $m$  là số *Line* của *cache*.

b.



c.

d. CPU phát ra cục N bit kia

i. Tag với Line để so sánh thôi

ii.  $2^L = \text{số Line}$

iii. Word là để lưu nội dung (phần này bằng kích thước Line or Block)

iv.  $2^W = \text{kích thước Block or Line}$

e. Ban đầu từ phần Line trong cục N bit kia để tìm ra Line tương ứng trong bộ Cache

f. Tìm Line xong rồi so sánh 2 Tag với nhau

i. Giống thì hit => Lấy/ghi luôn cache

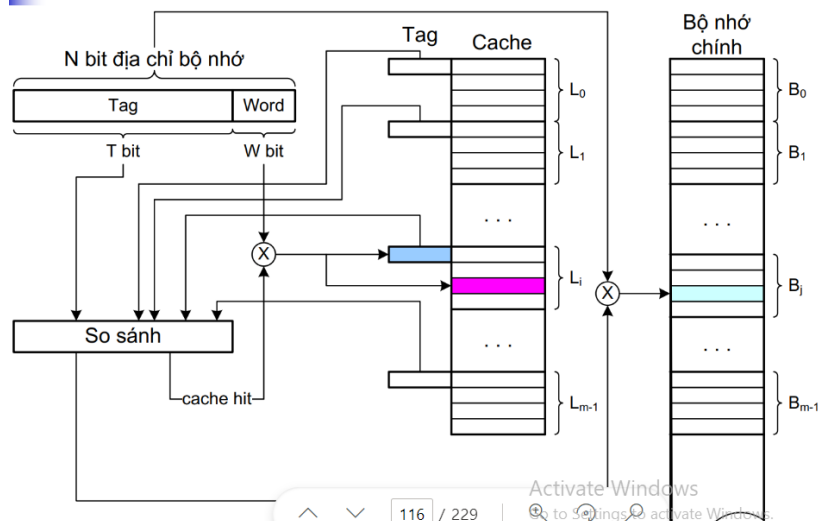
ii. Khác thì miss => Lấy/ghi BNC

g. Ưu: So sánh đơn giản

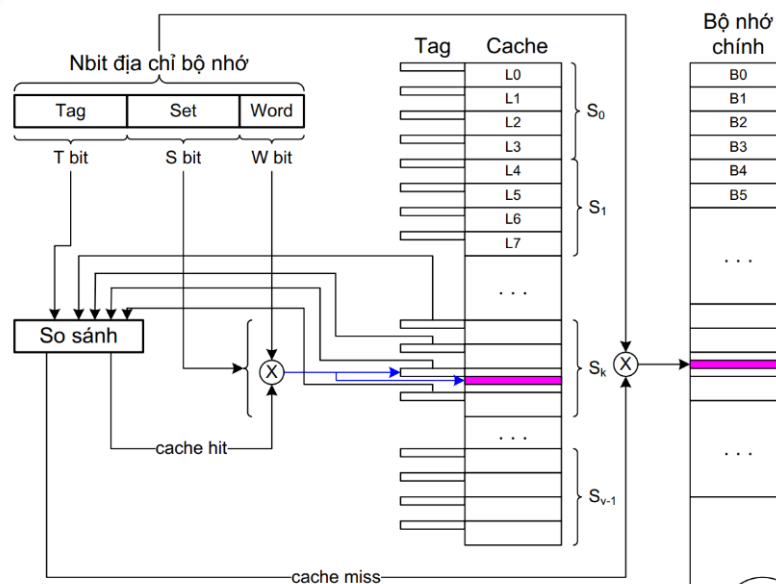
h. Nhược: Xác suất cache hit thấp

2. Ánh xạ liên kết toàn phần

a. Block nạp vào bất kỳ Line nào cũng được



- b.
  - c. CPU phát ra cục N bit kia
  - d. So sánh Tag lần lượt với các Tag của bộ Cache
    - i. Giống thì hit => lấy/ghi line
    - ii. Không thì miss
  - e. Ưu: Xác suất cache hit cao
  - f. Nhược:
    - i. Mất nhiều time
    - ii. So sánh phức tạp
    - iii. => Ít dùng
3. Ánh xạ liên kết tập hợp
- a. Dung hòa 2 thẳng trên (Một thẳng thì cứng nhắc quá, 1 thẳng thì không chia nhưng lại quét lần lượt)
  - b. Chia Cache thành các Set (ví dụ cứ 4 line = 1 Set)
  - c. Nguyên tắc:  $B_0 \rightarrow S_0, B_1 \rightarrow S_1, \dots$



d.

e. CPU phát ra cục N bit

- i. Từ S bit trong cục đó xác định được Set trong Cache (tương tự như tìm Line trong Ánh xạ trực tiếp)
- ii. So sánh T bit với lần lượt các Tag trong Set đó (na ná với Ánh xạ liên kết toàn phần)
- iii. Giống thì hit, khác thì miss

f. Thông dụng với 2,4,8,16 Lines/Set

4. Thay thế block trong cache

- a. Ánh xạ trực tiếp: không phải lựa chọn
- b. Ánh xạ liên kết: Có thuật giải thay thế:

**Random**: Thay thế ngẫu nhiên

**FIFO** (First In First Out): Thay thế *Block* nào nằm lâu nhất ở trong *Set* đó

**LFU** (Least Frequently Used): Thay thế *Block* nào trong *Set* có số lần truy nhập ít nhất trong cùng một khoảng thời gian

**LRU** (Least Recently Used): Thay thế *Block* ở trong *Set* tương ứng có thời gian lâu nhất không được tham chiếu tới

- i. Tối ưu nhất: **LRU**

5. Ghi dữ liệu khi cache hit

- a. Ghi xuyên qua
  - i. Ghi cả cache lẫn BNC => chậm
- b. Ghi trả sau
  - i. Chỉ ghi cache => nhanh

- ii. Block trong cache bị thay thế => ghi trả Block về BNC

d. Bộ nhớ ngoài

- i. Tài liệu dưới dạng các thiết bị lưu trữ

- ii. Các kiểu

- 1. Băng từ

- 2. Ổ đĩa từ (Ổ đĩa cứng HDD)

- a. Dung lượng lớn
    - b. Chậm
    - c. Tốn năng lượng
    - d. Dễ bị lỗi cơ học
    - e. Rẻ

- 3. Ổ đĩa quang (CD, DVD)

- 4. Bộ nhớ Flash

- a. SSD

- i. Bán dẫn Flash
      - ii. Không khả biến
      - iii. Nhanh
      - iv. Tiêu thụ ít năng lượng
      - v. Nhiều chip nhớ flash, truy cập song song được
      - vi. Ít bị lỗi
      - vii. Đắt

- b. USB Flash

- c. Thẻ nhớ

- 5. Hệ thống lưu trữ dung lượng lớn RAID

- a. Tập các ổ đĩa cứng vật lý được OS coi như 1 ổ logic duy nhất => **dung lượng lớn**
      - b. Data lưu trữ phân tán trên các ổ đĩa vật lý => truy cập song song **(nhanh)**
      - c. Lưu trữ thêm thông tin dự thừa, cho phép khôi phục lại thông tin trong trường hợp đĩa bị hỏng => **an toàn thông tin**
      - d. 7 loại phổ biến (RAID 0 – 6)

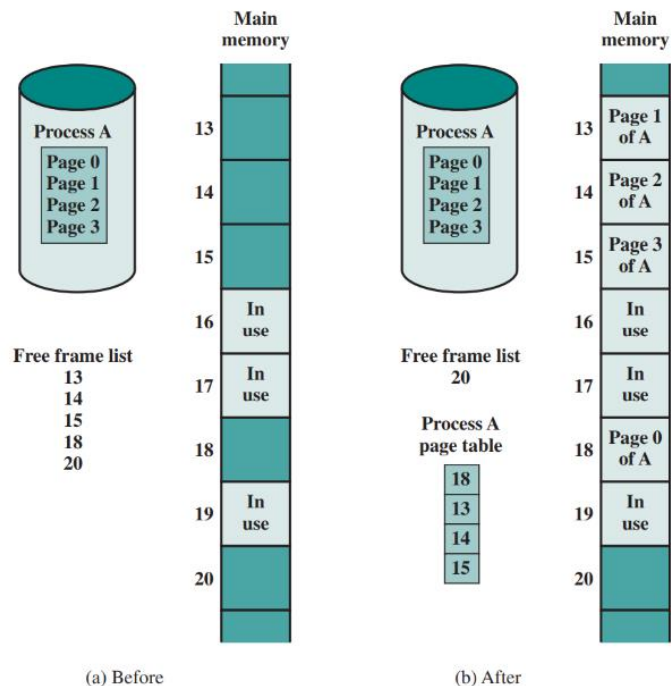
- iii. Bộ nhớ ảo

- 1. Khái niệm: BNC + Bộ nhớ ngoài (CPI coi tụi nó như 1 ổ duy nhất)

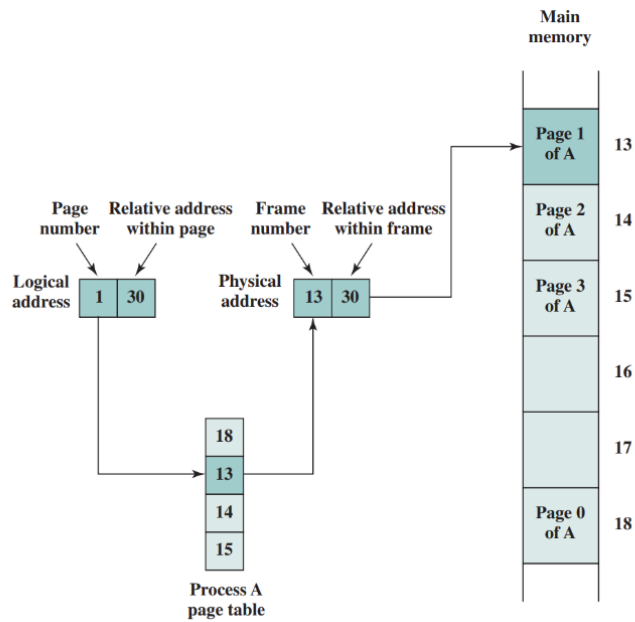
- 2. Kỹ thuật thực hiện:

- a. Phân trang: Chia thành các trang có kích thước = nhau và kề nhau

- b. Phân đoạn: Chia thành các đoạn có kích thước thay đổi, có thể đè nhau
3. Phân trang (Hơi khó hiểu nma hiểu **trang là thuộc về bộ nhớ ảo, khung trang là cái thuộc về bộ nhớ thực**)
- Chia :**
    - bộ nhớ thành các khung trang
    - chương trình (tiến trình) thành các trang
  - Cấp số hiệu** khung trang yêu cầu cho tiến trình
  - OS duy trì **list khung trang nhớ trống**
  - Tiến trình vào khung trang linh tinh cũng được chứ **k cần liên tiếp**
  - Dùng **bảng trang để quản lý**

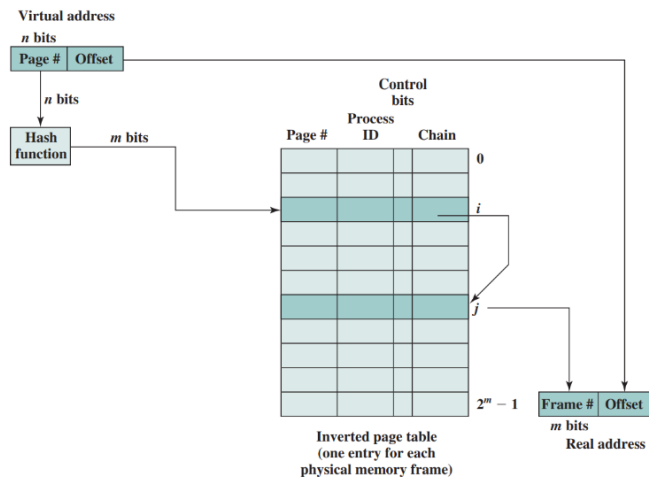


- f.
- g. Nhìn before và after xem, các trang đều vô các khung trang trống và dùng cái bảng trang để quản lý đó.



- h.  
i. Giải thích hình trên: Cái 13 là khung trang, thuộc về bộ nhớ thực, còn 1 là cái trang mình chia (thuộc về bộ nhớ ảo),  
j. Bảng trang (để quản lý)

## Cấu trúc bảng trang



- k.  
l. Giải thích: Trong cái bộ nhớ ảo lẫn bộ nhớ thực đều có 1 thằng **offset**, để biết **trang nào là của khung trang nào** (ánh xạ bộ nhớ thực với bộ nhớ ảo), **ProcessID chính là ID của Process A** đó, trong ví dụ bên trên thì cả 4 Page đó đều cũng 1 ProcessID. **Chain để xem Trang tiếp theo** của tiến trình ý nằm ở đâu. Và từ bảng trang này có thể quản lý,

trang nào ứng với khung trang nào, process ở những trang nào và có thứ tự các trang ra sao.

m. Nguyên tắc:

i. Phân trang theo yêu cầu

1. **Không yêu cầu all** trang of tiến trình nằm trong bộ nhớ

ii. Lỗi trang

1. Trang được yêu cầu không có trong bộ nhớ

2. OS cần hoán đổi trang yêu cầu vào

3. Có thể **hoán đổi để lấy chỗ**

4. Cần chọn trang để đưa ra

iii. Thất bại

- Quá nhiều tiến trình trong bộ nhớ quá nhỏ
- OS tiêu tốn toàn bộ thời gian cho việc hoán đổi
- Có ít hoặc không có công việc nào được thực hiện
- Đèn luôn luôn sáng
- Giải pháp:
  - Thuật toán thay trang
  - Giảm bớt số tiến trình đang chạy
  - Thêm bộ nhớ

1.

iv. Lợi ích

- Không cần toàn bộ tiến trình nằm trong bộ nhớ để chạy
- Có thể hoán đổi trang được yêu cầu
- Như vậy có thể chạy những tiến trình lớn hơn tổng bộ nhớ sẵn dùng
- Bộ nhớ chính được gọi là bộ nhớ thực
- Người dùng cảm giác bộ nhớ lớn hơn bộ nhớ thực

1.

e. Tổng kết

i. Cache: Tích hợp trên chip vi xử lý

1. L1: Cache lệnh và cache dữ liệu

2. L2, L3

ii. Bộ nhớ chính: tồn tại dưới dạng các module nhớ RAM

iii. Bộ nhớ trên PC (Khó nhớ ghê á)

- ROM BIOS chứa các chương trình sau:
  - Chương trình POST (Power On Self Test)
  - Chương trình CMOS Setup
  - Chương trình Bootstrap loader
  - Các trình điều khiển vào-ra cơ bản (BIOS)
- CMOS RAM:
  - Chứa thông tin cấu hình hệ thống
  - Đồng hồ hệ thống
  - Có pin nuôi riêng
- Video RAM: quản lý thông tin của màn hình
- Các loại bộ nhớ ngoài

1.

### 3. Chương 3: Hệ thống vào ra

#### a. Tổng quan:

- i. **Trao đổi thông tin giữa máy tính với bên ngoài**
- ii. Thao tác cơ bản: input, output
- iii. Các thành phần chính: **thiết bị vào ra + modun vào ra**

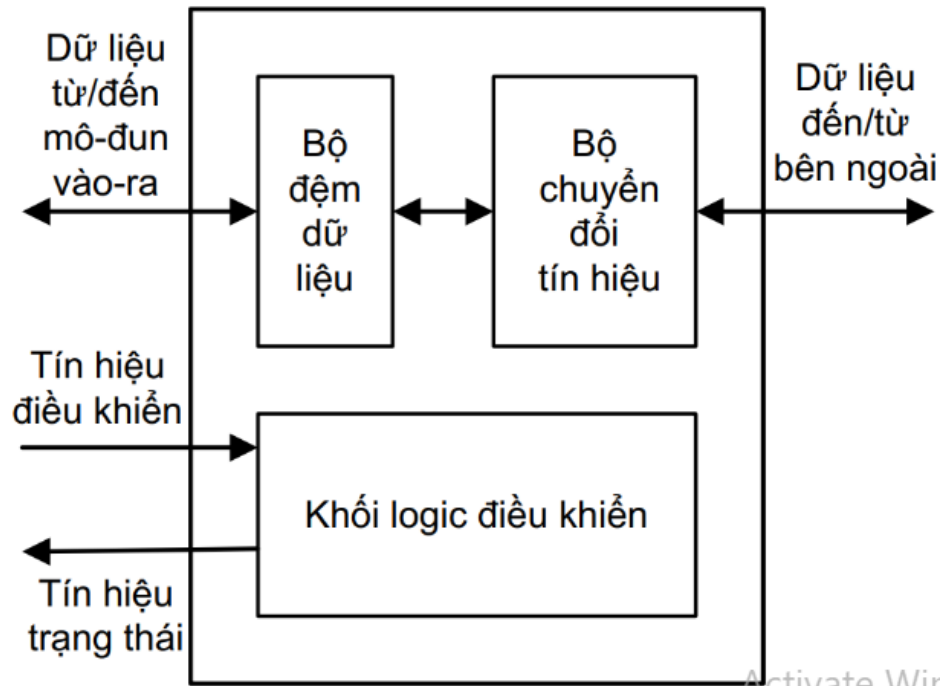
#### b. Đặc điểm

- i. Đa dạng:
  1. Nguyên tắc hoạt động
  2. Tốc độ
  3. Khuôn dạng dữ liệu
- ii. Thiết bị vào ra chậm hơn CPU và RAM
- iii. Cần Modun vào ra để nối ghép thiết bị với CPU, BNC

#### c. Thiết bị vào ra

- i. Thiết bị ngoại vi
- ii. Chức năng: Chuyển đổi dữ liệu giữa **trong và ngoài máy tính**
- iii. Phân loại:
  1. Thiết bị vào
  2. Thiết bị ra
  3. Thiết bị lưu trữ
  4. Thiết bị truyền thông
- iv. Giao tiếp:
  1. Người - máy
  2. Máy- máy
- v. Cấu trúc





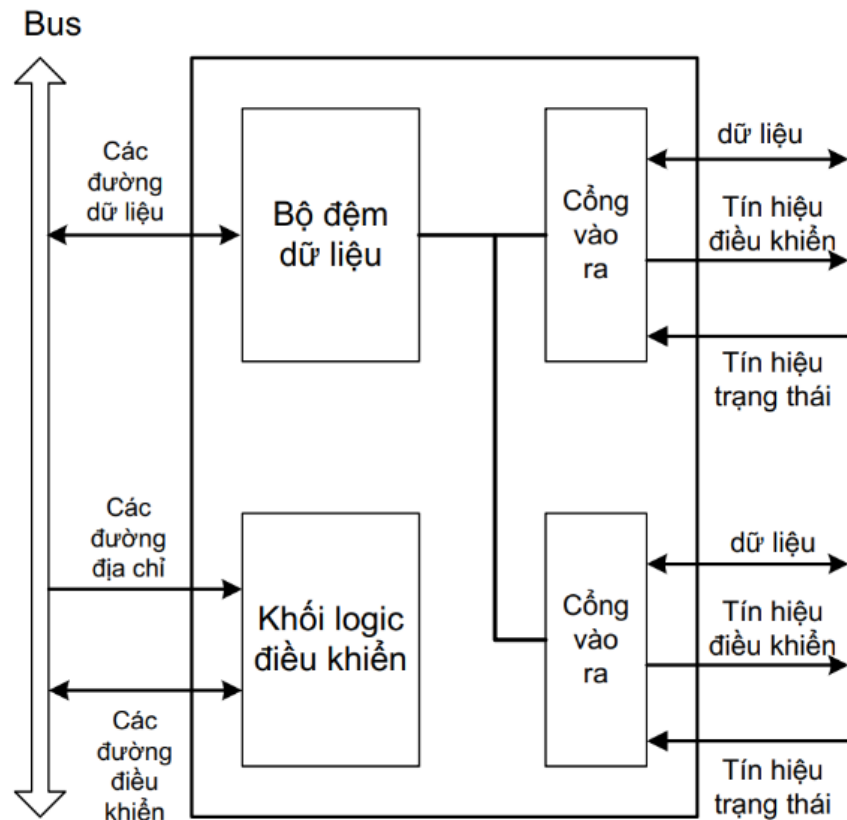
vi.

#### d. Modun vào ra

##### i. Chức năng:

1. Điều khiển và định thời ???
2. Trao đổi tt với CPU, BNC, thiết bị vào ra
3. Đệm giữa bên trong mt với thiết bị vào ra
4. Phát hiện lỗi của thiết bị vào ra

##### ii. Cấu trúc



- iii.
- iv. Nhìn thì giống cấu trúc thiết bị vào ra thế thôi nhưng mà khác, cái này sẽ đặt **bên trái** cấu trúc của thiết bị vào ra sẽ ra được cái hoàn chỉnh

#### e. Địa chỉ hóa cổng vào-ra (Cách đánh địa chỉ cho port đó)

- i. Không gian dùng cho **ngăn nhớ** và **cổng vào ra**: dùng **chung or tách biệt**
- ii. Phương pháp
  - 1. IO theo bản đồ bộ nhớ
  - 2. IO riêng biệt
- iii. IO theo bản đồ bộ nhớ (dùng chung với bộ nhớ)
  - 1. IO port **coi như ngăn nhớ** (cũng được đánh địa chỉ)
  - 2. Trao đổi dữ liệu với cổng IO = **lệnh truy nhập bộ nhớ**
  - 3. Thực hiện trên all hệ thống
  - 4. Như MIPS hồi xưa lưu cổng vào ra vào địa chỉ nào đó (load/store)
- iv. IO riêng biệt (dùng riêng)
  - 1. Đánh **địa chỉ riêng**
  - 2. Trao đổi = **lệnh riêng biệt**
  - 3. Intel x86 (In/Out)

## f. Các phương pháp điều khiển IO

### i. Có 3 cái

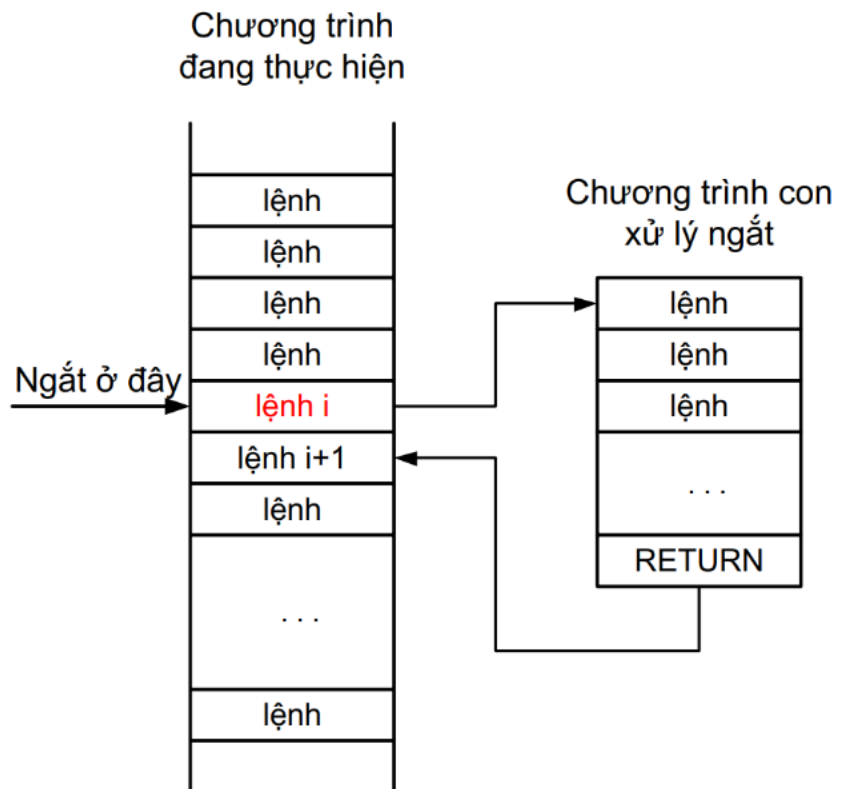
1. IO bằng chương trình
2. IO bằng ngắt
3. Truy nhập bộ nhớ trực tiếp - DMA

### ii. IO bằng chương trình (như mình vẫn lập trình printf scanf ấy)

1. CPU **điều khiển trực tiếp** bằng chương trình
2. Trước khi thực hiện lệnh IO, chương trình cần **đọc và check trạng thái sẵn sàng** của modul vào ra
3. Các tín hiệu điều khiển IO (nhớ 2 cấu trúc io bên trên)
  - a. Điều khiển: kích hoạt thiết bị io
  - b. Kiểm tra: ktra state của modul io và tb io
  - c. Điều khiển đọc: yêu cầu modul io nhận data từ tb io => đưa vào bộ đệm data => CPU nhận data đó
  - d. Điều khiển ghi: yêu cầu modul io lấy data trên bus data => đưa đến bộ đệm data => chuyển ra tb io
4. Các lệnh io
  - a. Io theo bản đồ nhớ: lệnh trao đổi dữ liệu với bộ nhớ
  - b. Io riêng biệt: lệnh chuyên dụng (IN, OUT)
5. Đặc điểm
  - a. Io theo ý muốn người lập trình
  - b. CPU trực tiếp điều khiển trao đổi giữa CPU với modul io
  - c. CPU đợi modul io => tốn time CPU

### iii. IO điều khiển bằng ngắt

1. Nguyên tắc chung:
  - a. CPU **không phải đợi** trạng thái sẵn sàng của modul io => CPU làm 1 ctr khác
  - b. Modul io sẵn sàng => **phát tín hiệu ngắt** cho CPU
  - c. CPU thực hiện **ctr con xử lý ngắt** io tương ứng => trao đổi dữ liệu
  - d. CPU **quay lại làm tiếp** ctr đang bị ngắt

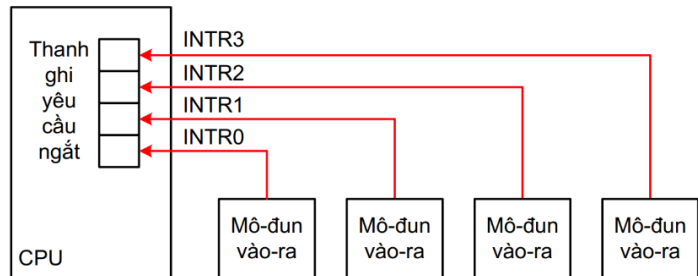


- 2.
3. Hoạt động io dữ liệu: nhìn từ modul io
  - a. Modul io nhận tín hiệu ĐỌC từ CPU
  - b. Modul io nhận data từ thiết bị io (lúc này CPU làm việc khác)
  - c. Có data rồi => modul io phát tín hiệu ngắt CPU
  - d. CPU yêu cầu data
  - e. Modul io chuyển data cho CPU
4. Hoạt động io dữ liệu: nhìn từ CPU
  - a. Phát tín hiệu ĐỌC
  - b. Làm việc khác
  - c. Cuối mỗi chu trình lệnh, ktra tín hiệu yêu cầu ngắt
  - d. Nếu bị ngắt:
    - i. Cất ngữ cảnh
    - ii. Thực hiện ctr con xử lý ngắt để lấy dữ liệu
    - iii. Khôi phục ngữ cảnh của ctr đang thực hiện
5. Phương pháp nối ghép ngắt (ví dụ nhiều yêu cầu ngắt)

a. Có các cách:

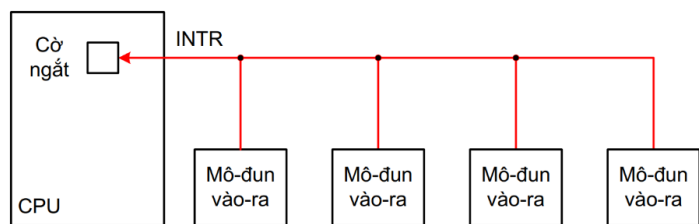
- i. Sử dụng nhiều đường yêu cầu ngắt
- ii. Hồi vòng bằng phần mềm
- iii. Hồi vòng bằng phần cứng
- iv. Sử dụng bộ điều khiển ngắt

b. Sử dụng nhiều đường yêu cầu ngắt



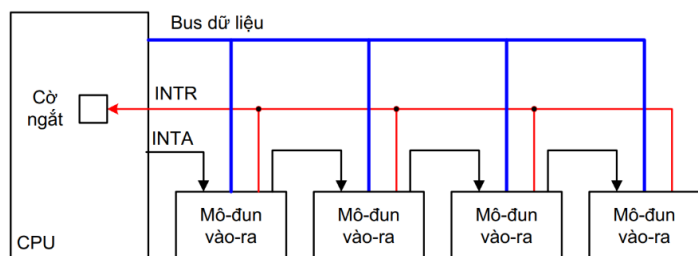
- i.
- ii. 1 modul ~ 1 đường yêu cầu ngắt
- iii. Hạn chế số lượng modul vào ra
- iv. Các đường ngắt được quy định mức ưu tiên

c. Hồi vòng bằng phần mềm



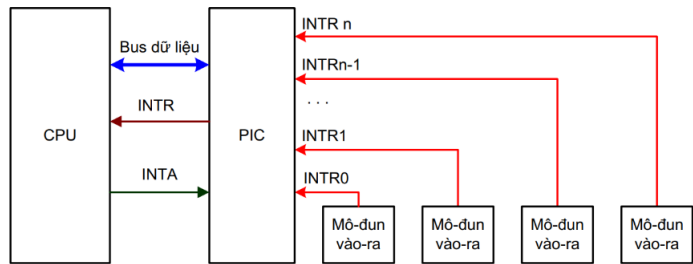
- i.
- ii. CPU thực hiện phần mềm hồi lần lượt từng modul io
- iii. Chậm
- iv. Thứ tự được hồi là thứ tự ưu tiên

d. Hồi vòng bằng phần cứng



- i.
- ii. CPU phát tín hiệu chấp nhận ngắt đến modul io đầu tiên
- iii. Nếu nó k ngắt thì gửi đến modul tiếp
- iv. Thứ tự modul io kết nối ~ thứ tự ưu tiên

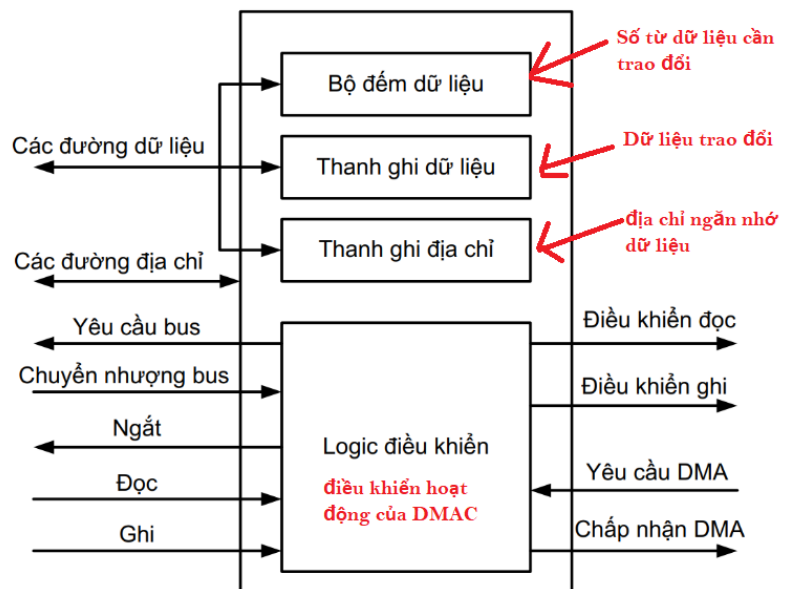
e. Dùng bộ điều khiển ngắt lập trình được (PIC)



- i.
- ii. PIC có **nhiều đường yêu cầu ngắt** có quy định mức ưu tiên (na ná cách 1)
- iii. PIC **chọn 1 yêu cầu ngắt** không bị cấm có **mức ưu tiên cao nhất** => gửi CPU
- iv. Kết hợp phần cứng + mềm:
  1. Cứng: gây ngắt CPU
  2. Mềm: Trao đổi dữ liệu giữa CPU với modun vào ra
- v. CPU trực tiếp điều khiển io
- vi. CPU không phải đợi => Hiệu quả tốt

#### iv. DMA

1. Mấy cái trên IO bằng chương trình và ngắt do CPU trực tiếp điều khiển => chiếm time of CPU
2. Khắc phục: bằng kỹ thuật DMA
3. DMAC (bộ điều khiển):



- a.
- b. Hoạt động
  - i. CPU nói cho DMAC
    1. vào/ra data
    2. Địa chỉ tb io (io port)

3. Địa chỉ đầu của mảng nhớ data => nạp vào thanh ghi địa chỉ

4. Số từ dữ liệu cần truyền => nạp vào bộ đếm dữ liệu

ii. CPU làm việc khác

iii. DMAC điều khiển trao đổi

iv. Sau khi truyền được 1 từ dữ liệu

1. Nội dung thanh ghi địa chỉ tăng

2. Nội dung bộ đếm dữ liệu giảm

v. Bộ đếm dữ liệu giảm về 0 => DMAC gửi tín hiệu ngắt CPU để báo kết thúc DMA

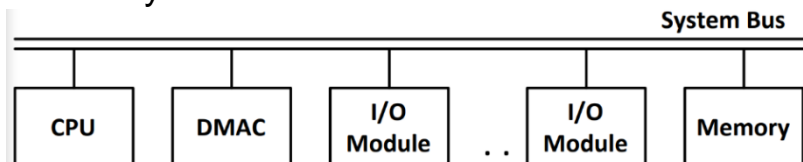
c. Các kiểu thực hiện DMA

i. DMA truyền theo khối: dùng bus để truyền xong cả khối dữ liệu

ii. DMA lấy chu kỳ: DMAC cưỡng bức CPU treo tạm thời từng chu kỳ bus => DMAC chiếm bus thực hiện truyền 1 từ dữ liệu

iii. DMA trong suốt: DMAC nhận biết những chu kỳ nào CPU không sử dụng bus thì chiếm bus để trao đổi 1 từ dữ liệu

d. DMA truyền theo khối

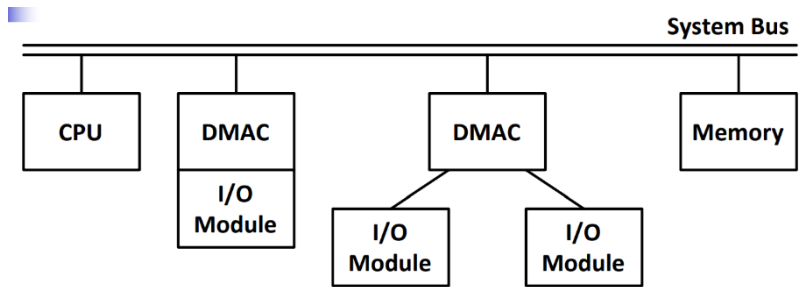


■ Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus hai lần

- Giữa mô-đun vào-ra với DMAC
- Giữa DMAC với bộ nhớ

e.

f. DMA lấy chu kỳ

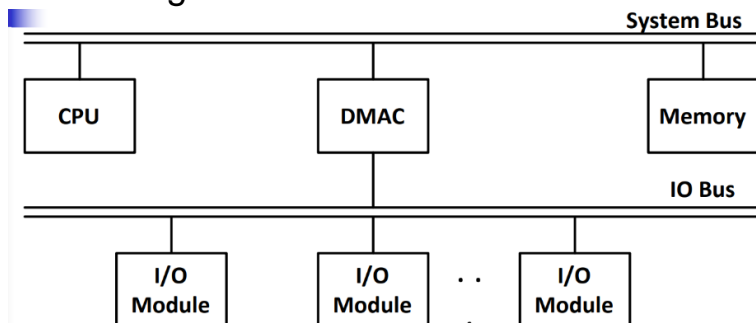


DMAC điều khiển một hoặc vài mô-đun vào-ra  
 Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus một lần

- Giữa DMAC với bộ nhớ

g.

h. DMA trong suốt



Bus vào-ra tách rời hỗ trợ tất cả các thiết bị cho phép DMA  
 Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus một lần

- Giữa DMAC với bộ nhớ

i.

#### 4. Đặc điểm

- a. CPU không tham gia trong quá trình trao đổi dữ liệu
- b. DMAC điều khiển trao đổi dữ liệu giữa bộ nhớ chính với modul io (bằng phần cứng) => nhanh
- c. Phù hợp với yêu cầu trao đổi data lớn

#### g. Bộ xử lý vào ra

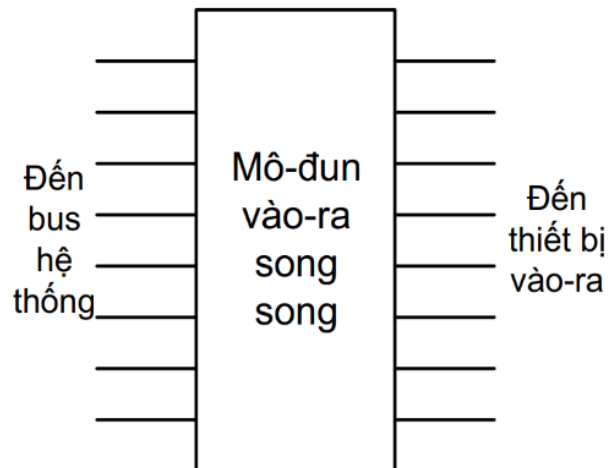
- i. Điều khiển io được thực hiện bởi 1 bộ xử lý io chuyên dụng
- ii. Chương trình bộ xử lý io nằm trong bộ nhớ chính or bộ nhớ riêng

#### h. Nối ghép thiết bị vào ra

- i. Có:
  1. Nối ghép song song
  2. Nối ghép nối tiếp



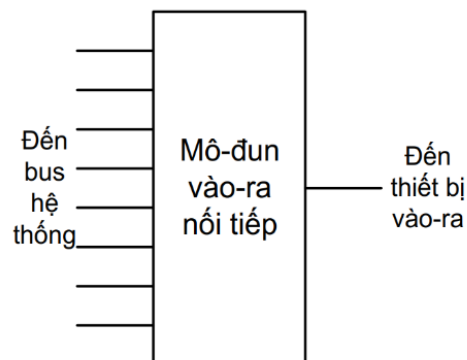
ii. Nối ghép song song



- Truyền nhiều bit song song
- Tốc độ nhanh
- Cần nhiều đường truyền dữ liệu

iii.

iv. Nối ghép nối tiếp



- Truyền lần lượt từng bit
- Cần có bộ chuyển đổi từ dữ liệu song song sang nối tiếp hoặc/và ngược lại
- Tốc độ chậm hơn
- Cần ít đường truyền dữ liệu

v.

vi. Cấu hình nối ghép

1. Điểm tới điểm: Thông qua 1 io port nối ghép với 1 thiết bị
2. Điểm tới đa điểm: Thông qua 1 cổng io cho phép nối ghép được với nhiều thiết bị

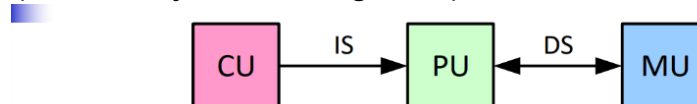
## 4. Chương 4: Các kiến trúc song song (Khó quá nhớ slide thôi)

### a. Kiến trúc máy tính

#### i. 4 cái:

1. SISD - Single Instruction Stream, Single Data Stream
2. SIMD - Single Instruction Stream, Multiple Data Stream
3. MISD - Multiple Instruction Stream, Single Data Stream
4. MIMD - Multiple Instruction Stream, Multiple Data Stream

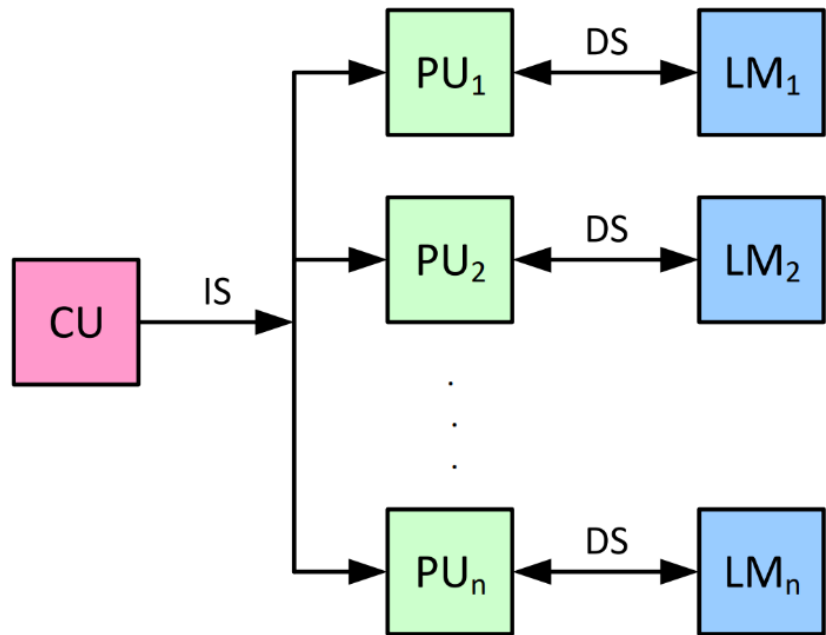
#### ii. SISD (1 bộ xử lý, đơn dòng lệnh)



- CU: Control Unit
- PU: Processing Unit
- MU: Memory Unit
- Một bộ xử lý
- Đơn dòng lệnh
- Dữ liệu được lưu trữ trong một bộ nhớ
- Chính là Kiến trúc von Neumann (tuần tự)

1.

#### iii. SIMD (đơn bộ xử lý, đơn dòng lệnh điều khiển nhiều đơn vị xử lý)



1.

Đơn dòng lệnh điều khiển đồng thời các đơn vị xử lý PUs

Mỗi phần tử xử lý có một bộ nhớ dữ liệu riêng LM (local memory)

Mỗi lệnh được thực hiện trên một tập các dữ liệu khác nhau

Các mô hình SIMD

- Vector Computer
- Array processor

2.

iv. MISD (đa bộ xử lý 1 luồng dữ liệu)

- Một luồng dữ liệu cùng được truyền đến một tập các bộ xử lý
- Mỗi bộ xử lý thực hiện một dãy lệnh khác nhau.
- Chưa tồn tại máy tính thực tế
- Có thể có trong tương lai

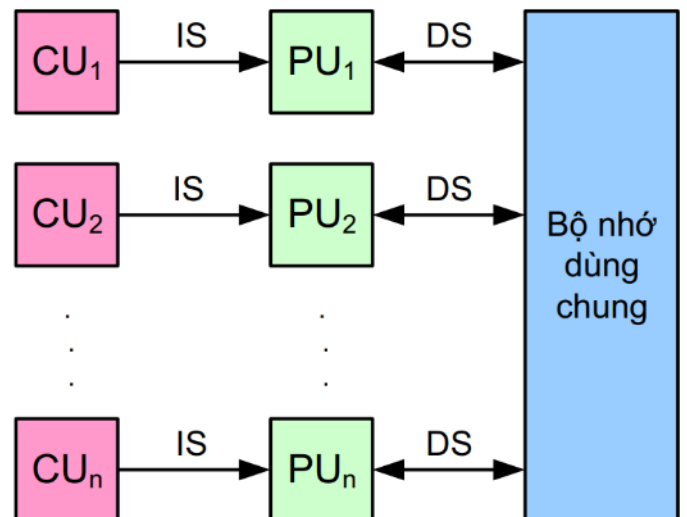
1.

v. MIMD (đa bộ xử lý đa luồng dữ liệu)

- Tập các bộ xử lý
- Các bộ xử lý đồng thời thực hiện các dãy lệnh khác nhau trên các dữ liệu khác nhau
- Các mô hình MIMD
  - Multiprocessors (Shared Memory)
  - Multicomputers (Distributed Memory)

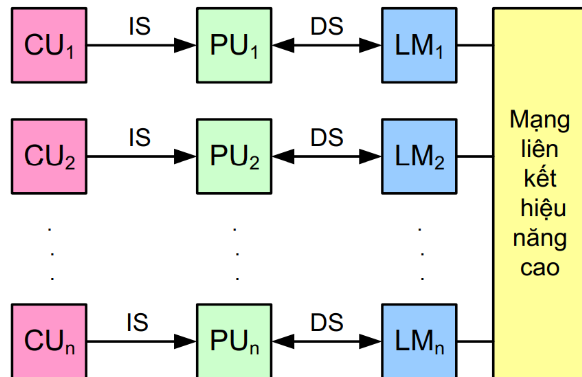
- 1.
2. MIMD-Shared Memory

Đa xử lý bộ nhớ dùng chung  
(shared memory multiprocessors)



- a.
3. MIMD Distributed Memory

## Đa xử lý bộ nhớ phân tán (distributed memory multiprocessors or multicomputers)



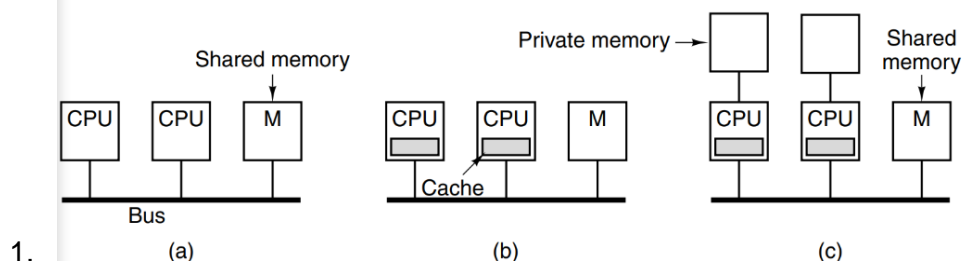
a.  
b.

### b. Phân loại các kỹ thuật song song

- i. Song song mức lệnh
  1. Pipeline
  2. Superscalar
- ii. Song song mức dữ liệu: SIMD
- iii. Song song mức luồng: MIMD
- iv. Song song mức yêu cầu: Cloud computing

### c. Đa xử lý bộ nhớ dùng chung

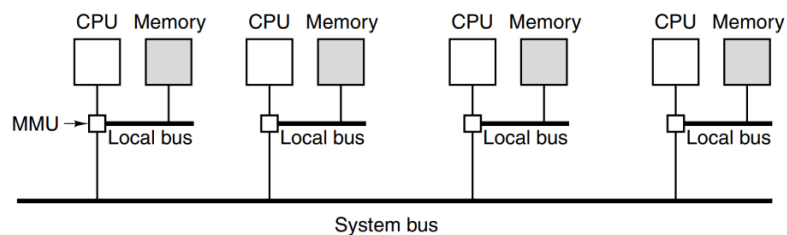
- i. Gồm:
  1. Hệ thống đa xử lý đối xứng SMP
  2. Hệ thống đa xử lý không đối xứng NUMA
  3. Bộ xử lý đa lõi
- ii. SMP (UMA)



- Một máy tính có  $n \geq 2$  bộ xử lý giống nhau
- Các bộ xử lý dùng chung bộ nhớ và hệ thống vào-ra
- Thời gian truy cập bộ nhớ là bằng nhau với các bộ xử lý
- Các bộ xử lý có thể thực hiện chức năng giống nhau
- Hệ thống được điều khiển bởi một hệ điều hành phân tán
- Hiệu năng: Các công việc có thể thực hiện song song
- Khả năng chịu lỗi

2.  
3.

### iii. NUMA



- Có một không gian địa chỉ chung cho tất cả CPU
- Mỗi CPU có thể truy cập từ xa sang bộ nhớ của CPU khác
- Truy nhập bộ nhớ từ xa chậm hơn truy nhập bộ nhớ cục bộ

1.  
2.

### iv. Bộ xử lý đa lõi

#### 1. Các thay đổi của bộ xử lý:

- a. Tuần tự
- b. Pipeline
- c. Siêu vô hướng
- d. Đa luồng
- e. Đa lõi: nhiều CPU trên 1 chip

### v. Đa bộ xử lý bộ nhớ phân tán

#### 1. Máy tính quy mô lớn

- Hệ thống qui mô lớn
- Đắt tiền: nhiều triệu USD
- Dùng cho tính toán khoa học và các bài toán có số phép toán và dữ liệu rất lớn
- Siêu máy tính

a.  
b.

## 2. Máy tính cụm (Cluster)

Nhiều máy tính được kết nối với nhau bằng mạng liên kết tốc độ cao (~ Gbps)

Mỗi máy tính có thể làm việc độc lập (PC hoặc SMP)

Mỗi máy tính được gọi là một node

Các máy tính có thể được quản lý làm việc song song theo nhóm (cluster)

Toàn bộ hệ thống có thể coi như là một máy tính song song

Tính sẵn sàng cao

Khả năng chịu lỗi lớn

3.

## vi. Bộ xử lý đồ họa đa dụng

1. SIMD

2. GPGPU

3. Hệ thống lai CPU/GPGPU

a. CPU là host: tuần tự

b. GPGPU: song song

4.

## 5. Bài tập:

a. Bus địa chỉ  $\Rightarrow 2^n$

■ Ví dụ:

- Máy tính sử dụng bus địa chỉ 32-bit ( $A_{31}-A_0$ ), bộ nhớ chính được đánh địa chỉ cho từng byte  
 $\rightarrow$  Có khả năng đánh địa chỉ cho  $2^{32}$  bytes nhớ = 4GiB

b. Bus dữ liệu  $\Rightarrow$  chia 8

■ Ví dụ:

- Máy tính có bus dữ liệu kết nối CPU với bộ nhớ là 64-bit  
 $\rightarrow$  Có thể trao đổi 8 byte nhớ ở một thời điểm

c. BT chương 1:

i. Dạng 1:

1. Viết tắt:

- a. P: Hiệu năng
- b. t: thời gian thực hiện
- c.  $T_0$ : Chu kỳ xung nhịp
- d.  $F_0$ : Tốc độ xung nhịp
- e. N: số chu kỳ xung nhịp
- f.  $t_{CPU}$ : Thời gian thực hiện của CPU

2. Công thức:

- a.  $P = 1/t$
- b.  $F_0 = 1/T_0$

$$t_{CPU} = n \times T_0 = \frac{n}{f_0}$$

3.

4. Ví dụ:



“Máy tính A nhanh hơn máy B k lần”

$$P_A / P_B = t_B / t_A = k$$

■ Ví dụ: Thời gian chạy chương trình:

- 10s trên máy A, 15s trên máy B
- $t_B / t_A = 15s / 10s = 1.5$
- Vậy máy A nhanh hơn máy B 1.5 lần

a.

b.

Hiệu năng được tăng lên bằng cách:

- Giảm số chu kỳ xung nhịp n
- Tăng tốc độ xung nhịp  $f_0$

c.

- Hai máy tính A và B cùng chạy một chương trình
- Máy tính A:
  - Tốc độ xung nhịp của CPU:  $f_A = 2GHz$
  - Thời gian CPU thực hiện chương trình:  $t_A = 10s$
- Máy tính B:
  - Thời gian CPU thực hiện chương trình:  $t_B = 6s$
  - Số chu kỳ xung nhịp khi chạy chương trình trên máy B ( $n_B$ ) nhiều hơn 1.2 lần số chu kỳ xung nhịp khi chạy chương trình trên máy A ( $n_A$ )
- Hãy xác định tốc độ xung nhịp cần thiết cho máy B ( $f_B$ )?

d.

Ta có:  $t = \frac{n}{f}$

Số chu kỳ xung nhịp khi chạy chương trình trên máy A:

$$n_A = t_A \times f_A = 10s \times 2GHz = 20 \times 10^9$$

Số chu kỳ xung nhịp khi chạy chương trình trên máy B:

$$n_B = 1.2 \times n_A = 24 \times 10^9$$

Tốc độ xung nhịp cần thiết cho máy B:

$$f_B = \frac{n_B}{t_B} = \frac{24 \times 10^9}{6} = 4 \times 10^9 Hz = 4GHz$$

e.

ii. Dạng 2:

1. Viết tắt:

- a. IC-số lệnh của chương trình (Instruction Count)
- b. CPI -số chu kỳ trên một lệnh (Cycles per Instruction)

2. Công thức:

Vậy thời gian thực hiện của CPU:

$$t_{CPU} = IC \times CPI \times T_0 = \frac{IC \times CPI}{f_0}$$

- Nếu loại lệnh khác nhau có số chu kỳ khác nhau, ta có tổng số chu kỳ:

$$n = \sum_{i=1}^K (CPI_i \times IC_i)$$

- CPI trung bình:

$$CPI_{TB} = \frac{n}{IC} = \frac{1}{IC} \sum_{i=1}^K (CPI_i \times IC_i)$$

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

*Thời gian CPU = Số lệnh của chương trình x Số chu kỳ/lệnh x Số giây của một chu kỳ*

$$t_{CPU} = IC \times CPI \times T_0 = \frac{IC \times CPI}{f_0}$$

- Hiệu năng phụ thuộc vào:

- Thuật giải
- Ngôn ngữ lập trình
- Chương trình dịch
- Kiến trúc tập lệnh
- Phần cứng

## MIPS như là thước đo hiệu năng

- MIPS: Millions of Instructions Per Second  
(Số triệu lệnh trên 1 giây)

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} = \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

$$\text{MIPS} = \frac{f_0}{\text{CPI} \times 10^6}$$

$$\text{CPI} = \frac{f_0}{\text{MIPS} \times 10^6}$$

## MFLOPS

- Sử dụng cho các hệ thống tính toán lớn
- Millions of Floating Point Operations per Second
- Số triệu phép toán số dấu phẩy động trên một giây

$$\text{MFLOPS} = \frac{\text{Executed floating point operations}}{\text{Execution time} \times 10^6}$$

GFLOPS ( $10^9$ )

TFLOPS ( $10^{12}$ )

PFLOPS ( $10^{15}$ )

### 3. Ví dụ

- a.
- b.

Ta có:  $t_{CPU} = IC \times CPI_{TB} \times T_0$

Hai máy cùng kiến trúc tập lệnh, vì vậy số lệnh của cùng một chương trình trên hai máy là bằng nhau:

$$IC_A = IC_B = IC$$

Thời gian thực hiện chương trình đó trên máy A và máy B:

$$t_A = IC_A \times CPI_A \times T_A = IC \times 2.0 \times 250ps = IC \times 500ps$$

$$t_B = IC_B \times CPI_B \times T_B = IC \times 1.2 \times 500ps = IC \times 600ps$$

Từ đó ta có:  $\frac{t_B}{t_A} = \frac{IC \times 600ps}{IC \times 500ps} = 1.2$

- c. Kết luận: máy A nhanh hơn máy B 1.2 lần
- Hai máy tính A và B có cùng kiến trúc tập lệnh
  - Máy tính A có:
    - Chu kỳ xung nhịp:  $T_A = 250ps$
    - Số chu kỳ/ lệnh trung bình:  $CPI_A = 2.0$
  - Máy tính B:
    - Chu kỳ xung nhịp:  $T_B = 500ps$
    - Số chu kỳ/ lệnh trung bình:  $CPI_B = 1.2$
  - Hãy xác định máy nào nhanh hơn và nhanh hơn bao nhiêu ?

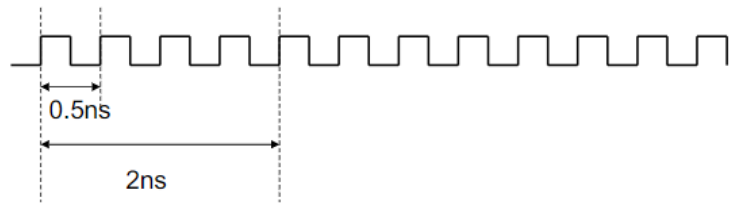
- d. Cho bảng chỉ ra các dãy lệnh sử dụng các lệnh thuộc các loại A, B, C. Tính CPI trung bình?

Loại lệnh	A	B	C
CPI theo loại lệnh	1	2	3
IC trong dãy lệnh 1	20	10	20
IC trong dãy lệnh 2	40	10	10

- e.
- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>■ Dãy lệnh 1: Số lệnh = 50<ul style="list-style-type: none"><li>■ Số chu kỳ =<br/><math>= 1 \times 20 + 2 \times 10 + 3 \times 20 = 100</math></li><li>■ <math>CPI_{TB} = 100/50 = 2.0</math></li></ul></li></ul> | <ul style="list-style-type: none"><li>■ Dãy lệnh 2: Số lệnh = 60<ul style="list-style-type: none"><li>■ Số chu kỳ =<br/><math>= 1 \times 40 + 2 \times 10 + 3 \times 10 = 90</math></li><li>■ <math>CPI_{TB} = 90/60 = 1.5</math></li></ul></li></ul> |
|---|---|

f.

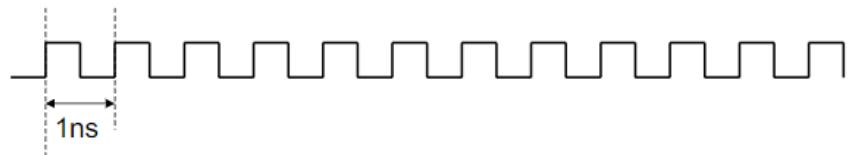
- Tính MIPS của bộ xử lý với:  
clock rate = 2GHz và CPI = 4



- Chu kỳ  $T_0 = 1/(2 \times 10^9) = 0.5\text{ns}$
- $\text{CPI} = 4 \rightarrow$  thời gian thực hiện 1 lệnh =  $4 \times 0.5\text{ns} = 2\text{ns}$
- Số lệnh thực hiện trong 1s =  $(10^9\text{ns})/(2\text{ns}) = 5 \times 10^8$  lệnh
- Vậy bộ xử lý thực hiện được 500 MIPS

g.

- Tính CPI của bộ xử lý với:  
clock rate = 1GHz và 400 MIPS



- Chu kỳ  $T_0 = 1/10^9 = 1\text{ns}$
- Số lệnh thực hiện trong 1 s là 400MIPS =  $4 \times 10^8$  lệnh
- Thời gian thực hiện 1 lệnh =  $1/(4 \times 10^8)\text{s} = 2.5\text{ns}$
- Vậy ta có:  $\text{CPI} = 2.5$

h.

d. BT chương 2

i. Dạng 1

1. Đường địa chỉ =  $n$
2. Đường dữ liệu =  $m$
3. Dung lượng chip nhớ =  $2^n \times m$  bit (CẦN THẬN ĐỀ LỪA LÀ DRAM thì phải  $2^{(2n)} \times m$  bit)

## Tăng độ dài từ nhớ

VD1:

- Cho chip nhớ SRAM 4K x 4 bit
- Thiết kế mô-đun nhớ 4K x 8 bit

Giải:

- Dung lượng chip nhớ =  $2^{12} \times 4$  bit
- chip nhớ có:
  - 12 chân địa chỉ
  - 4 chân dữ liệu
- mô-đun nhớ cần có:
  - 12 chân địa chỉ
  - 8 chân dữ liệu

- Giải thích:
    - $1k = 1024$ 
      - $\Rightarrow 4k = 4 \times 1024 = 2^{12}$
      - $\Rightarrow$  đánh được cho 12 chân địa chỉ
      - $\Rightarrow$  có  $2^{12}$  địa chỉ
    - ở chip nhớ 1 địa chỉ đánh được 4 bit (từ đề bài)
      - $\Rightarrow$  có 4 chân dữ liệu
    - Áp dụng CT với bit nhớ
      - $\Rightarrow$  dung lượng:  $2^{12} \times 4$  (bit)
    - Tương tự với module
  - Bonus: 4k x 4 bit kia cũng là dung lượng từ nhớ (tương đương với cái dung lượng nhớ đc của cache)
4. Vd2:

## VD2:

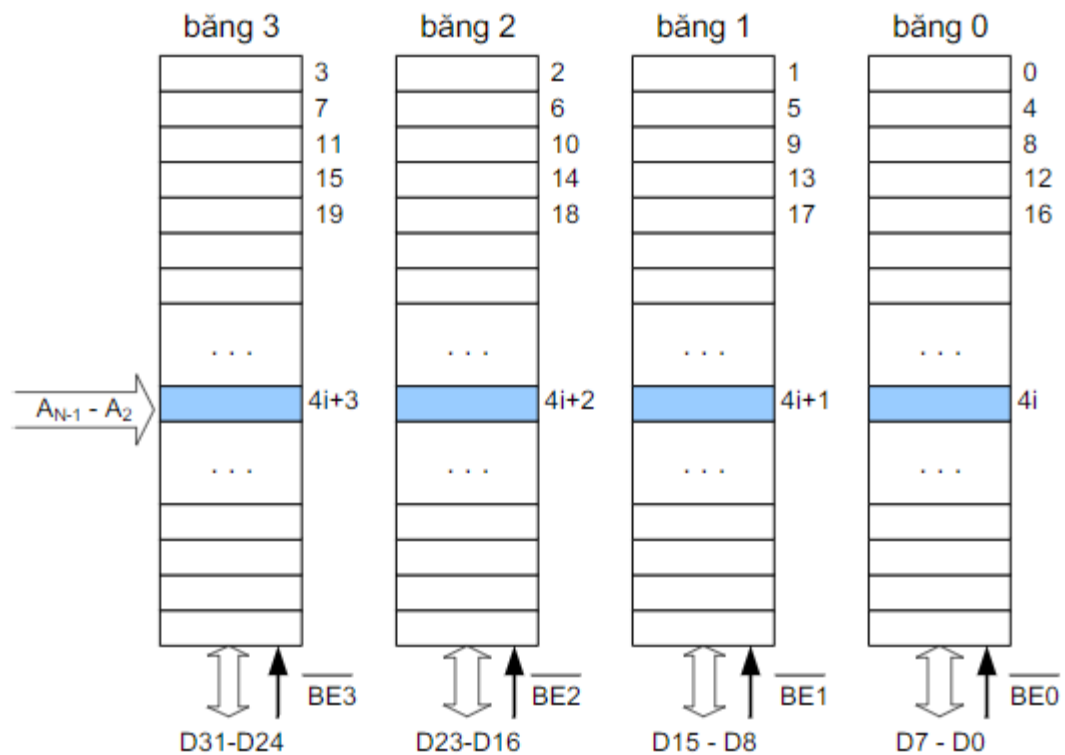
- Cho chip nhớ SRAM 4K x 8 bit
- Thiết kế mô-đun nhớ 8K x 8 bit

## Giải:

- Dung lượng chip nhớ =  $2^{12} \times 8$  bit
- chip nhớ có:
  - 12 chân địa chỉ
  - 8 chân dữ liệu
- Dung lượng mô-đun nhớ =  $2^{13} \times 8$  bit
  - 13 chân địa chỉ
  - 8 chân dữ liệu

- ii. Dạng 2 (hên xui): Mấy cái mạch vẽ các thứ (chịu)
- iii. Dạng 3

$m = 32\text{bit} \rightarrow$  bốn băng nhớ đơn xen



iv.

M

v.

vi. Kiểu kiểu này:

1. Cho địa chỉ 127 chẳng hạn

$M = 32\text{ bit}$

cái địa chỉ 127 thuộc băng nhớ nào?

2. Giải:

a. 32 bit  $\Rightarrow$  Có 4 băng nhớ

b. Lấy 127 chia 4 dư 3  $\Rightarrow$  Băng nhớ thứ 3 (kết quả)

vii. Bài tập phần Cache (Cộng trừ nhân chia thôi)



Giả sử máy tính đánh địa chỉ cho từng byte

Không gian địa chỉ bộ nhớ chính = 4GiB

Dung lượng bộ nhớ *cache* là 256KiB

Kích thước *Line (Block)* = 32byte.

Xác định số bit của các trường địa chỉ cho ba trường hợp tổ chức:

- Ánh xạ trực tiếp
- Ánh xạ liên kết toàn phần
- Ánh xạ liên kết tập hợp 4 đường

1.

### Với ánh xạ trực tiếp

Bộ nhớ chính = 4GiB =  $2^{32}$  byte → Số bit địa chỉ của bộ nhớ chính là:  $N = 32$  bit

*Cache* = 256 KiB =  $2^{18}$  byte

Kích thước *Line* = 32 byte =  $2^5$  byte → số bit địa chỉ của trường Word là:  $W = 5$  bit

Số *Line* trong *cache* =  $2^{18} / 2^5 = 2^{13}$  *Line* → số bit địa chỉ trường Line là:  $L = 13$  bit

Số bit địa chỉ của trường Tag là:

$$T = 32 - (13 + 5) = 14 \text{ bit}$$

Tag	Line	Word
-----	------	------

14 bit

13 bit

5 bit

2.

## Với ánh xạ liên kết toàn phần

Bộ nhớ chính = 4GiB =  $2^{32}$  byte → số bit địa chỉ của bộ nhớ chính là:  $N = 32$  bit

Kích thước *Line* = 32 byte =  $2^5$  byte → số bit địa chỉ của trường Word là:  $W = 5$  bit

Số bit địa chỉ của trường *Tag* là:

$$T = 32 - 5 = 27 \text{ bit}$$

Tag	Word
-----	------

27 bit

5 bit

3.

## Với ánh xạ liên kết tập hợp 4 đường

- Bộ nhớ chính = 4GiB =  $2^{32}$  byte → số bit địa chỉ của bộ nhớ chính là:  $N = 32$  bit

- Kích thước *Line* = 32 byte =  $2^5$  byte → số bit địa chỉ của trường Word là:  $W = 5$  bit

- Số *Line* trong *cache* =  $2^{18} / 2^5 = 2^{13}$  *Line*

- Một *Set* có 4 *Line* =  $2^2$  *Line*

→ số *Set* trong *cache* =  $2^{13} / 2^2 = 2^{11}$  *Set*

→ số bit địa chỉ của trường *Set* là:  $S = 11$  bit

- Số bit địa chỉ của trường *Tag* là:

$$T = 32 - (11 + 5) = 16 \text{ bit}$$

Tag	Set	Word
-----	-----	------

16 bit

11 bit

5 bit

4.

### viii. Bài tập RAID

1. raid 0 thì ko có hồi phục nên = tổng dung lượng các đĩa
2. raid 1 là mirror nên = tổng dung lượng chia đôi
3. raid từ 3 -> 5 thì = dung lượng của đĩa nhỏ nhất \* (số đĩa - 1)
4. raid 6 = dung lượng của đĩa nhỏ nhất \* (số đĩa - 2)

**Ví dụ:**

#7ad73f

1/1 point (graded)

Một hệ thống RAID gồm 4 ổ đĩa cứng HDD 2 Terabyte, ghép với nhau để tạo thành 1 hệ thống lưu trữ với dung lượng tối đa 4 Terabyte. Hỏi loại RAID nào đã được áp dụng?

- ☐ RAID 6
- ☐ RAID 4
- ☐ RAID 5
- ☒ RAID 1



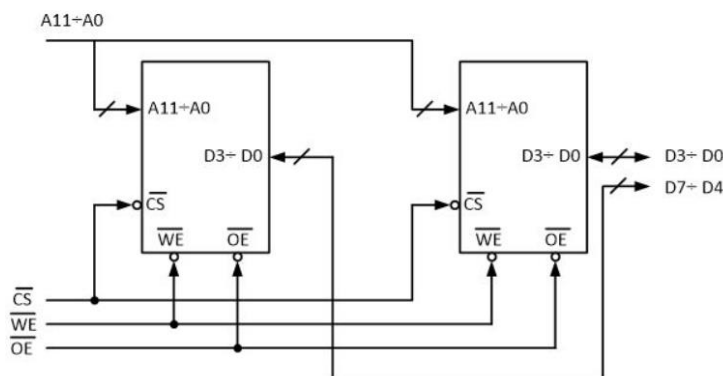
4 ổ đĩa cứng x 2 Terabyte => Nó có 8 Terabyte. Hệ thống lưu trữ với dung lượng 4 Terabyte => Gấp đôi (Vì  $8/4 = 2$ )

NHƯNG MÀ dung lượng \* (số đĩa - 2) =  $2 \times (4 - 2) = 4$  (cũng bằng cái dung lượng hệ thống kia) => CŨNG LÀ RAID 6 chứ nhỉ???

## BÀI TẬP SƠ ĐỒ MẠCH

#6b1003

1/1 point (graded)



Cách thiết kế module nhớ từ các chip nhớ như trong ảnh có tác dụng gì?

- ☐ Số lượng ngăn nhớ giảm. Kích thước ngăn nhớ không đổi
- ☐ Số lượng ngăn nhớ tăng. Kích thước ngăn nhớ tăng
- ☒ Số lượng ngăn nhớ không đổi. Kích thước ngăn nhớ tăng
- ☐ Số lượng ngăn nhớ tăng. Kích thước ngăn nhớ không đổi



A: Số từ nhớ

D: Kích thước

=> Thấy 1 dây A mà 2 dây D nên là số lượng không đổi nhưng kích thước tăng 😊

## BÀI TẬP THÊM (Đội 1 xúu)

1. Chip nhớ có 10 đường địa chỉ, 8 đường dữ liệu. Modul nhớ có 11 đường địa chỉ, 16 đường dữ liệu. Cần bao nhiêu chip nhớ để thiết kế modul nhớ này?

a. CTTQ:

i. Coi  $n$  là đường địa chỉ,  $m$  là đường dữ liệu

ii.  $2^n \cdot m$  của modul /  $2^n \cdot m$  của chip nhớ

iii.  $\Rightarrow 2^{11} \cdot 16 / (2^{10} \cdot 8) = 4$

iv. Đáp án là 4

b. Giải thích kỹ hơn:

i. Do đường địa chỉ của modul nhiều hơn đường địa chỉ của chip nhớ  $\Rightarrow$  Có ghép song song

ii. Ban đầu phải ghép nối tiếp 2 thanh chip nhớ kia để ra 1 cái 16 đường dữ liệu, sau đó phải ghép song song thêm 1 cục (có 2 thanh chip nhớ) tương tự để tăng đường địa chỉ lên 11  $\Rightarrow$  Có  $2 \cdot 2 = 4$

c. Trường hợp nối tiếp only thì đường địa chỉ modul và chip nhớ sẽ bằng nhau  $\Rightarrow$  Lấy thương của đường dữ liệu thôi (Nhưng mà cứ áp dụng CTTQ vẫn đúng cho mọi trường hợp)

2. Chương trình gì đó viết bằng Java,

a. có 6 lệnh

b. biên dịch thành 180 lệnh mã máy

c. CPI trung bình = 2

d. CPU có tốc độ 1.5GHz

e. Hỏi thời gian thực hiện phần mềm ?

f. Giải:

i. Lệnh bình thường  $\Rightarrow$  Khum quan tâm (Cẩn thận bị lừa nha)

ii. Quan tâm lệnh mã máy thôi  $\Rightarrow IC = 180$

iii. CPI trung bình = 2

iv. Thấy Hz kia  $\Rightarrow f = 1.5 \text{ GHz}$

v. Time thực hiện =  $IC \cdot CPI \cdot T = IC \cdot CPI / f = \dots$

3. Note: Cùng cấu trúc tập lệnh  $\Rightarrow$  Mặc định 2 IC bằng nhau (hay cho trong bài toán so sánh máy nào chạy nhanh hơn và mình phải dùng  $CT \text{ time} = IC \cdot CPI \cdot T$ )

4. Bài tập kiểu dữ liệu số nguyên

a. 511 là số nguyên KHÔNG DẤU lớn nhất trong phạm vi biểu diễn

- i. Có 9 bit vì  $2^9 = 512$
  - b. 511 là số nguyên CỐ DẤU lớn nhất trong phạm vi biểu diễn
    - i. Có 10 bit vì  $2^{(10-1)} = 512$
- 5. Bài tập về băng nhớ
  - a. Họ cho 2 ngăn nhớ liên tiếp nằm ở băng nhớ số 7 và số 0
  - b.  $\Rightarrow$  Có 8 băng nhớ
  - c.  $\Rightarrow$  Bus dữ liệu có độ rộng  $8 \times 8 = 64$  bit
- 6. Bộ xử lý X có bus dữ liệu 16bit, bus địa chỉ 26 bit, quản lý bộ nhớ theo byte. Hỏi X có thể ghép với RAM tối đa là bao nhiêu
  - a. Cần thận bị lừa chỗ RAM nhé, vì đề có thể hỏi 2 kiểu
  - b. Nếu là RAM, thì sẽ chỉ tính qua bus địa chỉ thôi
    - i.  $\Rightarrow$  Dung lượng =  $2^{26} = 64\text{MiB}$
    - ii.  $\Rightarrow$  Chọn đáp án 64MiB
  - c. Nếu thay RAM bằng chip nhớ thì sẽ dùng công thức như slide
    - i. Dung lượng =  $2^{26} \times 16$
- 7. Bus hệ thống của máy tính có đường bus địa chỉ = 34 bit. Máy tính đã cắm 1 thanh RAM 4GiB. Hỏi máy có thể lắp 1 thanh RAM có kích thước lớn nhất bằng bao nhiêu và không bị thừa (máy tính không có cổng vào ra)
  - a. Dung lượng =  $2^{34} = 16 \text{ GiB}$
  - b. Đã lắp 4GiB  $\Rightarrow$  Còn 12GiB
  - c. Đừng chọn 12GiB nha vì chả ai lắp RAM 12GiB vì RAM chỉ có những loại kiểu 4 / 8 / 16 thôi
  - d.  $\Rightarrow$  Lắp 8 GiB
- 8. Máy tính bus địa chỉ 12bit, bus dữ liệu 32 bit. Hỏi ngăn nhớ ở địa chỉ 0x12345A thuộc băng nhớ nào
  - a. Khum cần quan tâm bus địa chỉ khi làm về băng nhớ
  - b. Bus dữ liệu 32 bit  $\Rightarrow$  Có  $32 / 8 = 4$  băng nhớ
  - c. Bình thường thì phải đổi 0x12345A ra thập phân rồi chia 4 lấy dư NHƯNG DÀI NÊN ĐỪNG LÀM
  - d. Cách làm: Vì nếu đổi 0x12345A ra thập phân thì nguyên đoạn 0x12345 nó sẽ phải nhân với 1 số mũ của 8 nên luôn chia hết cho 4 r  $\Rightarrow$  CHỈ XÉT SỐ CUỐI THÔI
  - e. Xét A  $\Rightarrow A = 10$  chia 4 dư 2
  - f.  $\Rightarrow$  Ở ngăn nhớ thứ 2