



## Chương 2 : Tổ chức CPU

---

### **Mục tiêu :**

- **Nắm được chức năng của CPU**
- **Hiểu được các thành phần bên trong CPU.**
- **Nắm được cách CPU giao tiếp với thiết bị ngoại vi.**
- **Biết được các đặc tính của CPU họ Intel**



# ***NỘI DUNG***

---

2.1 Giới thiệu hệ thống số

2.2 Bộ xử lý trung tâm CPU

2.3 Hệ thống Bus

2.4 Bộ thanh ghi

2.5 Cơ chế định vị địa chỉ

2.6 Các đặc tính thiết kế liên quan đến hiệu suất CPU họ Intel

2.7 Các đặc trưng của CPU họ Intel

2.8 Câu hỏi ôn tập

## 2.1 Hệ thống số

Hệ đếm	Cơ số	số ký số	dạng ký số và ký tự biểu diễn số
nhị phân	2	2	<b>0 1</b> <b>Ex : 1010<sub>b</sub></b>
bát phân	8	8	0 1 2 3 4 5 6 7      Ex : 24 <sub>o</sub>
thập phân	10	10	0 1 2 3 4 5 6 7 8 9      Ex : 12 <sub>d</sub>
thập lục phân	16	16	<b>0 1 2 3 4 5 6 7 8 9 A B C D E F</b> <b>Ex : 3F8<sub>h</sub></b>



# Hệ thống số

---

Hệ thống số là gì ?

Vào thời điểm đó, việc dùng các que để đếm là 1 ý tưởng vĩ đại!!  
Còn việc dùng các ký hiệu thay cho các que đếm còn vĩ đại hơn!!!!  
Một trong các cách để biểu diễn 1 số hiện nay là sử dụng hệ thống số đếm decimal.

Có nhiều cách để biểu diễn 1 giá trị số. Ngày xưa, con người dùng các que để

đếm sau đó đã học vẽ các hình trên mặt đất và trên giấy.

thí dụ số 5 lần đầu được biểu diễn bằng | | | | | (bằng 5 que).

Sau đó chữ số La Mã bắt đầu dùng các ký hiệu khác nhau để biểu diễn nhiều số gọn hơn.

Thí dụ số 3 vẫn biểu diễn bởi 3 que | | | nhưng số 5 thì được thay bằng V còn số 10 thì thay bằng X.



# Hệ thống số

---

**Sử dụng que để đếm là 1 ý nghĩa vĩ đại ở thời điểm này. Và việc dùng các ký hiệu để thay cho các que đếm càng vĩ đại hơn!!!.**

***Một trong những cách tốt nhất hiện nay là dùng hệ thống số thập phân (decimal system).***

# Decimal System

Con người ngày nay dùng hệ 10 để đếm. Trong hệ 10 có 10 digits

**0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

Những ký số này có thể biểu diễn bất kỳ 1 giá trị nào, thí dụ :

**754**

$$7 \cdot 10^2 + 5 \cdot 10^1 + 4 \cdot 10^0 = 700 + 50 + 4 = 754$$

base

digit  
position



Vị trí của từng ký số rất quan trọng, thí dụ nếu ta đặt "7"  
ở cuối thì:  
**547**  
nó sẽ là 1 giá trị khác :

$$5 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0 = 500 + 40 + 7 = 547$$

Diagram illustrating the positional value of digits in the number 547:

- The digit **10** in  $10^2$  is highlighted with a red box, and a red line points to a box labeled **base**.
- The digit **0** in  $10^0$  is highlighted with a green box, and a green line points to a box labeled **digit position**.

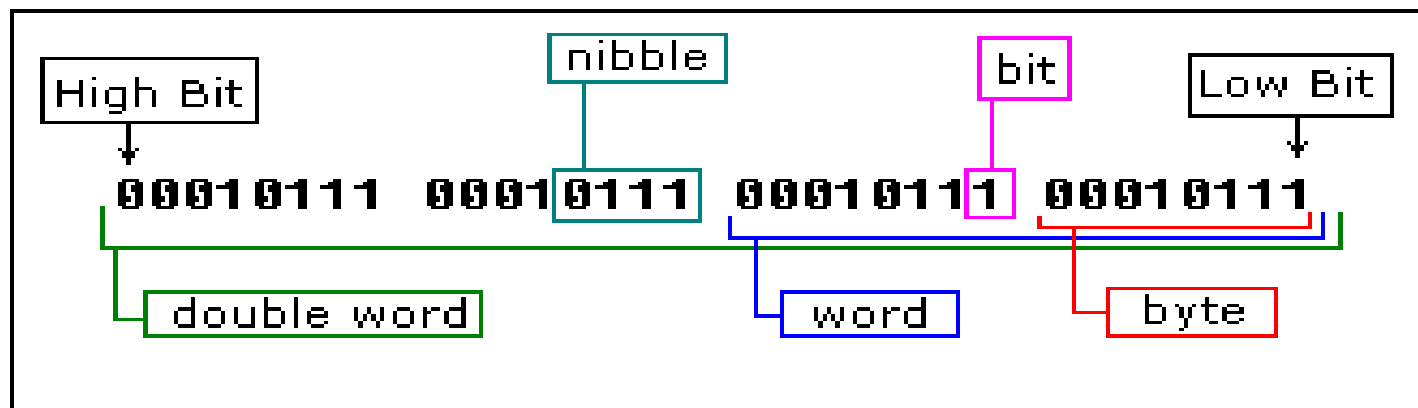
# Binary System

MT không thông minh như con người, nó dùng trạng thái của điện tử :  
on and off, or 1 and 0.

MT dùng binary system, binary system có 2 digits:  
0, 1

Như vậy cơ số (base) là 2.

Mỗi ký số (digit) trong hệ binary number được gọi là BIT, 4 bits nhóm  
thành 1 NIBBLE, 8 bits tạo thành 1 BYTE, 2 bytes tạo thành  
1 WORD, 2 words tạo thành 1 DOUBLE WORD (ít dùng):







# Hexadecimal System

---

## Hexadecimal System

Hexadecimal System dùng 16 digits:

**0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**

do đó cơ số (**base**) là **16**.

Hexadecimal numbers are compact and easy to read.

Ta dễ dàng biến đổi các số từ binary system sang hexadecimal system and và ngược lại, mỗi nibble (4 bits) có thể biến thành 1 hexadecimal digit :

Ex : **1234<sub>h</sub>** = 4660<sub>d</sub>



# Các phép toán trong hệ nhị phân

---

**cộng :**

$$0 + 0 = 0 \quad 0 + 1 = 1 \quad 1 + 0 = 1 \quad 1 + 1 = 0 \text{ nhớ } 1$$

**trừ :**  $0 - 0 = 0 \quad 0 - 1 = 1 \text{ mượn } 1 \quad 1 - 0 = 1 \quad 1 - 1 = 0$

**Nhân :** có thể coi là phép cộng liên tiếp

**Chia :** có thể coi là phép trừ liên tiếp

# Các phép toán trong hệ nhị phân ...

## Bảng phép tính Logic cho các số nhị phân

A	B	A and B	A or B	A xor B	Not A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

# Chuyển hệ từ 10 $\rightarrow$ hệ 2

Đổi từ hệ 10  $\rightarrow$  hệ 2 :

Ex : 12d = 1100b

Cách đổi : lấy số cần đổi chia liên tiếp cho 2, dừng khi số bị chia bằng 0. Kết quả là các số dư lấy theo chiều ngược lại.

$$12 : 2 = 6$$

0

$$6 : 2 = 3$$

0

$$3 : 2 = 1$$

1

$$1 : 2 = 0 \quad \text{dừng}$$

1



# Chuyển hệ từ hệ 2 $\rightarrow$ hệ 10

Đổi từ hệ 2  $\rightarrow$  hệ 10 :

Ex :  $1100_b = ?_d$

Cách đổi :  $\sum a_i * 2^i$  với  $i \in 0..n$

**a là ký số của số cần đổi.**

$$1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 = 12_d$$

$\downarrow$   
**a**



# Chuyển hệ từ hệ 10 $\rightarrow$ hệ 16

Đổi từ hệ 10  $\rightarrow$  hệ 16 :

$$\text{Ex : } 253_{\text{d}} = ?_{\text{h}}$$

**Cách đổi** : lấy số cần đổi chia liên tiếp cho 16, dừng khi số bị chia = 0. Kết quả là chuỗi số dư lấy theo chiều ngược lại.

$$253_{\text{d}} = \text{FD}_{\text{h}}$$





# Chuyển hệ từ hệ 2 $\rightarrow$ hệ 16

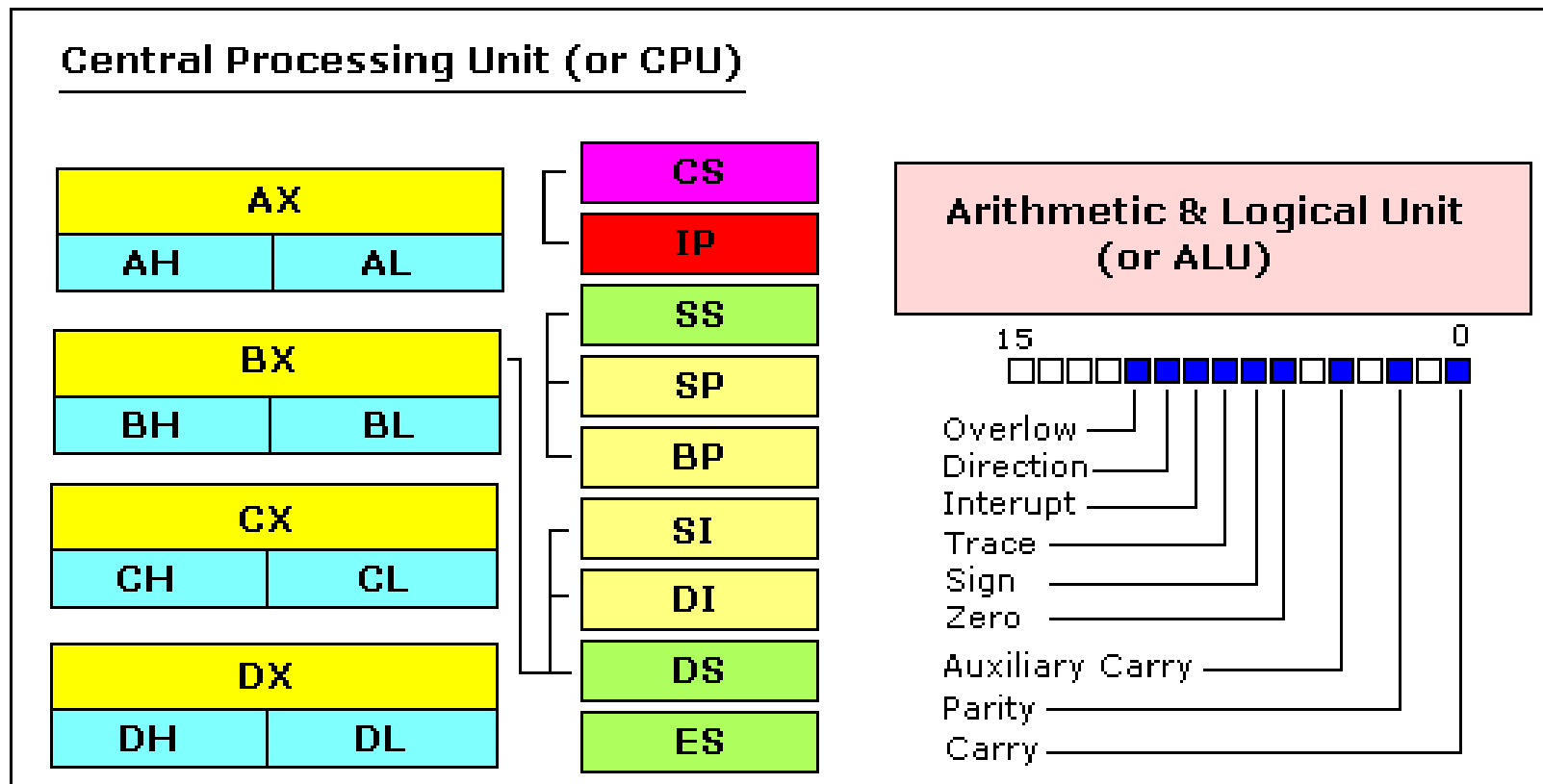
Đổi từ hệ 2  $\rightarrow$  hệ 16 :

**Ex :**  $101011010_b = ?_h$

**Cách đổi :** nhóm 4 chữ số nhị phân thành từng nhóm, rồi chuyển đổi từng nhóm sang số hệ thập lục phân.

$$\begin{array}{ccc} 0001 & 0101 & 1010 \\ \hline & 1 & 5 \end{array} = 15A_h$$

## 2.2 Bộ xử lý trung tâm CPU






## 2.2 Bộ xử lý trung tâm CPU

### CPU (Central Processing Unit) Bộ xử lý trung tâm –

**Chức năng : thực hiện chương trình lưu trong bộ nhớ chính bằng cách lấy lệnh ra - khảo sát - thực hiện lần lượt các lệnh.**

**Mỗi CPU có 1 tập lệnh riêng. Chương trình được thực thi ở CPU nào sẽ chỉ gồm các lệnh trong tập lệnh của CPU đó.**

**CPU gồm 1 số bộ phận tách biệt :**

- **Bộ điều khiển lấy lệnh ra từ bộ nhớ và xác định kiểu lệnh.** 
- **Bộ luận lý và số học (ALU) thực hiện phép toán như cộng, and.**
- **Các thanh ghi (Registers) : lưu kết quả tạm thời và các thông tin điều khiển. CPU giao tiếp với các bộ phận khác trong máy tính thông qua các tuyến gọi là Bus**

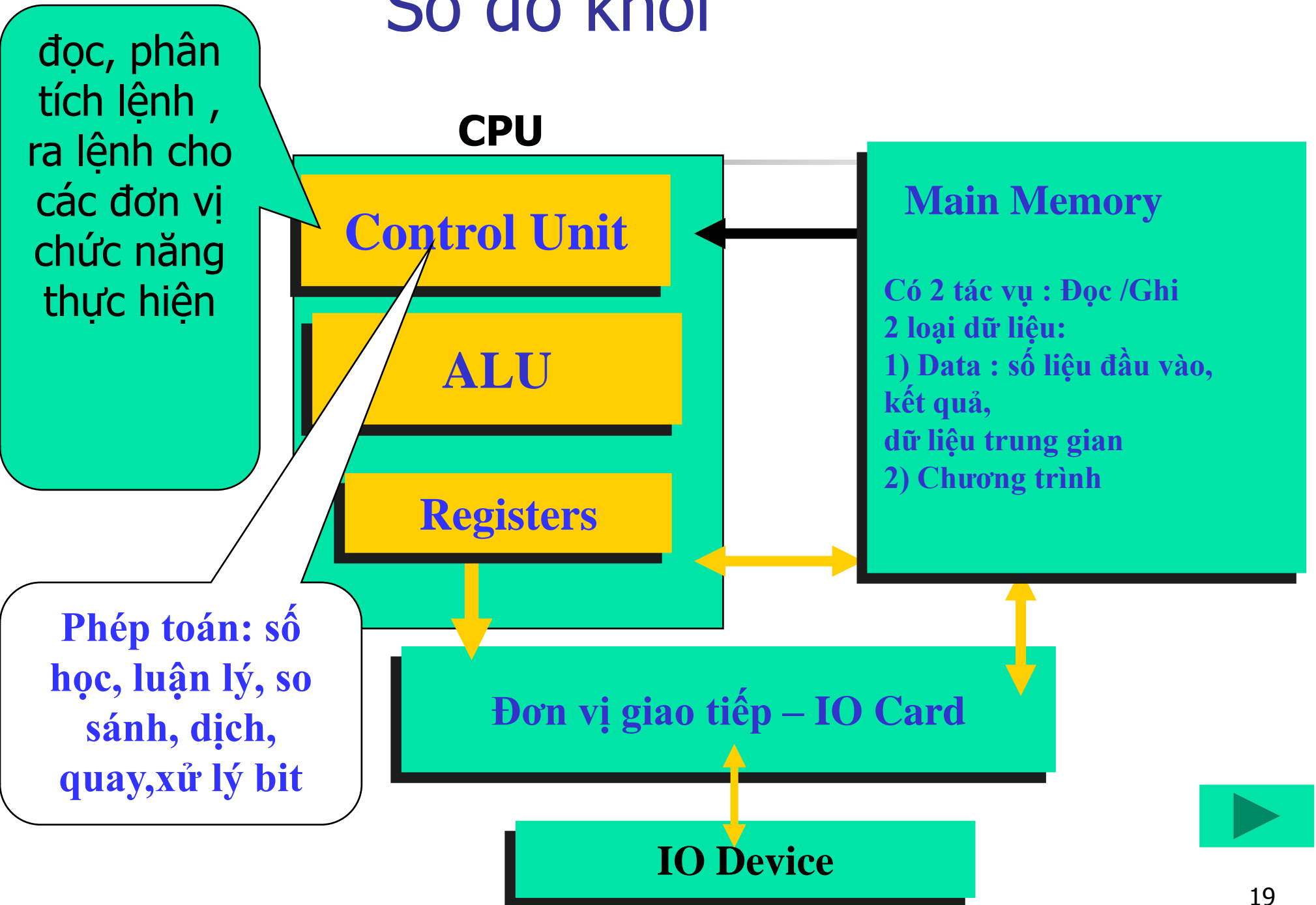


## CPU (cont)

---

- Các nhà chế tạo CPU qui định tốc độ thực hiện của từng chip phù hợp với nhịp tim của chip đó (clock speed) tốc độ đồng hồ, nhịp đồng hồ.
- Đơn vị đo tốc độ của chip CPU là Mhz cho biết chip đập bao nhiêu nhịp trong 1 s.  
Ex : CPU 500Mhz.

# Sơ đồ khối





# Chu kỳ lệnh

---

Một chu kỳ thực hiện lệnh máy gồm 3 giai đoạn chính sau :

1. **Lấy lệnh : lệnh cất ở ô nhớ sẽ được lấy vào thanh ghi lệnh.**
2. **Giải mã và thực hiện lệnh : lệnh trong thanh ghi lệnh sẽ được giải mã và thực hiện theo mô tả của lệnh trong tập lệnh.**
3. **Xác định địa chỉ của lệnh tiếp theo : trong khi lệnh được thực hiện, giá trị của bộ đếm chương trình sẽ tự động tăng lên chỉ đến ô nhớ chứa lệnh sẽ được thực hiện tiếp theo.**

**Chu kỳ lệnh được xây dựng từ những đơn vị cơ bản là chu kỳ máy.**



# Chu kỳ máy

---

Chu kỳ máy là chu kỳ của 1 hoạt động cơ bản của máy tính như :

- Chu kỳ đọc bộ nhớ
- Chu kỳ ghi bộ nhớ
- Chu kỳ đọc toán hạng
- Chu kỳ ghi kết quả

Clock : xung làm nhiệm vụ định thì cho mạch tuần tự.



# Thực hiện lệnh

---

CPU thực hiện lệnh tuần tự theo chuỗi các bước :

- Lấy lệnh kể từ bộ nhớ. → thanh ghi lệnh.
- Thay đổi PC để chỉ đến lệnh kế tiếp.
- Xác định kiểu lệnh vừa lấy ra.
- Xác định kiểu dữ liệu vừa yêu cầu và xác định vị trí dữ liệu trong bộ nhớ.
- Nếu lệnh cần dữ liệu trong bộ nhớ, nạp nó vào thanh ghi của CPU

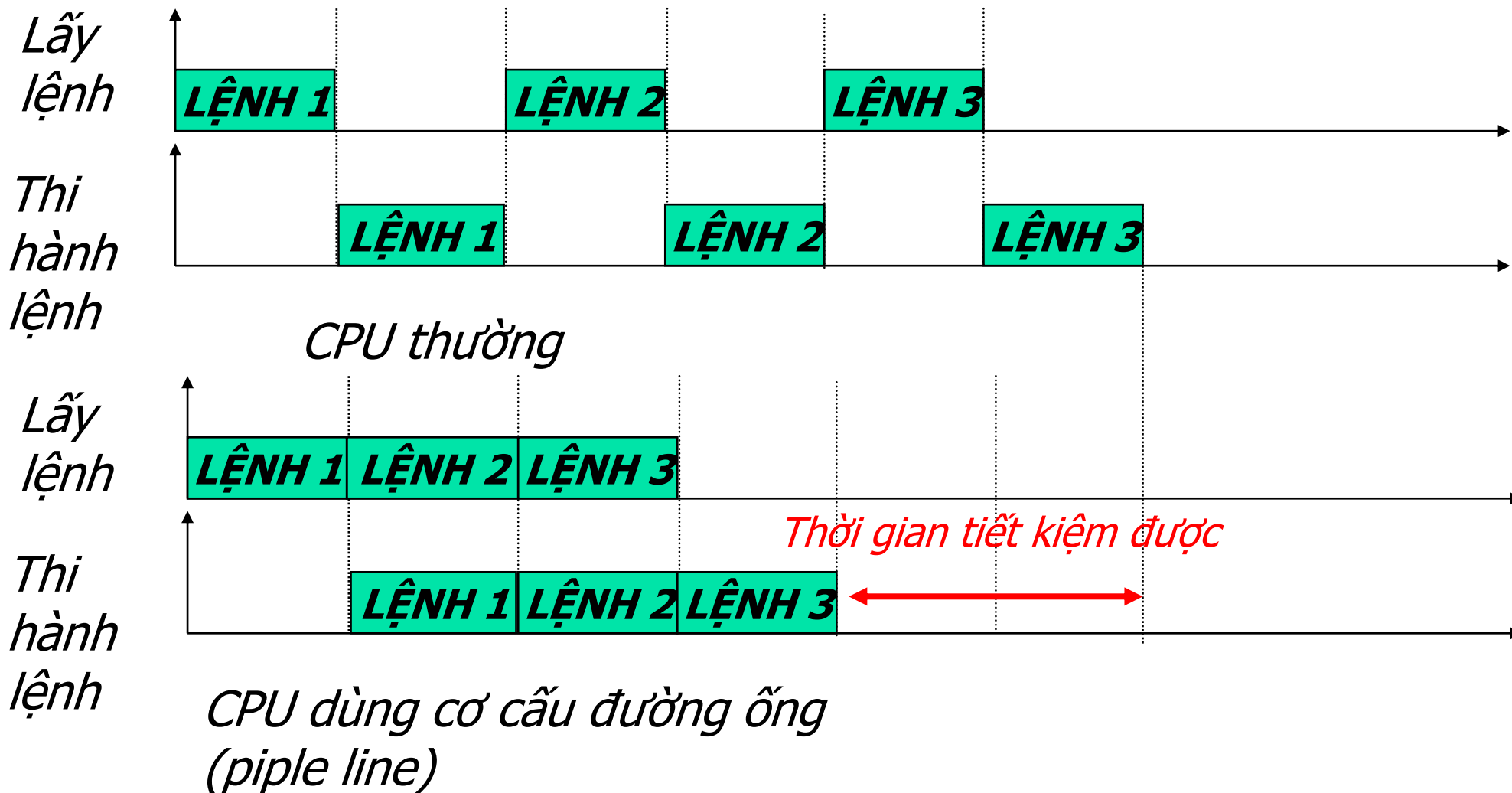


## Thực hiện lệnh (cont)

---

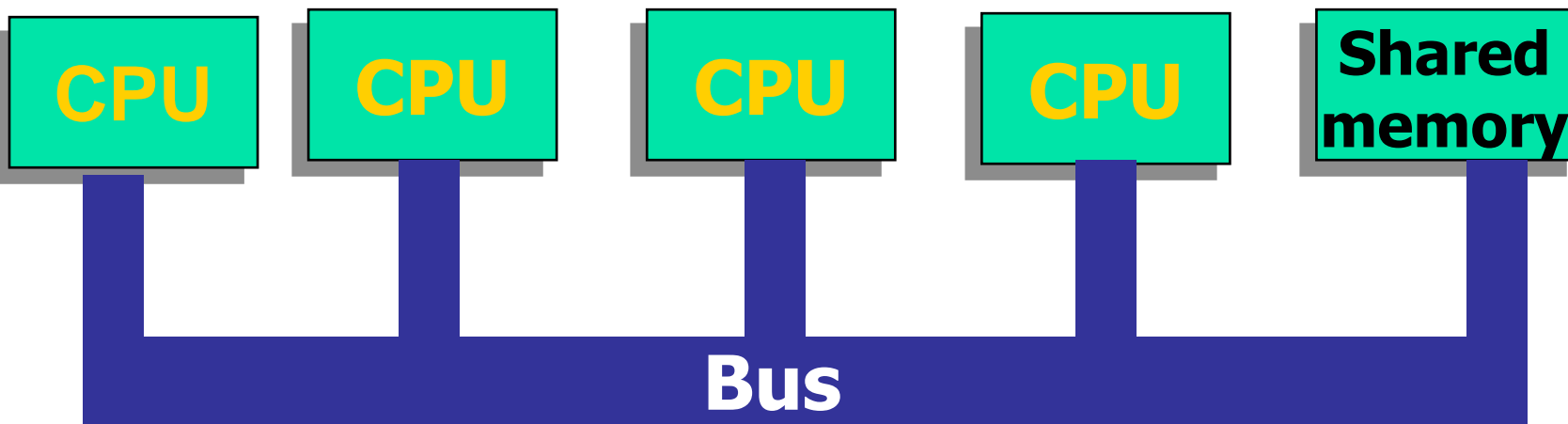
- Thực hiện lệnh..
- Lưu kết quả ở nơi thích hợp. .
- Trở về bước 1 để thực hiện lệnh kế.

## Sự phân phối thời gian cho 2 quá trình lấy lệnh và thi hành lệnh của CPU thường và CPU đường ống



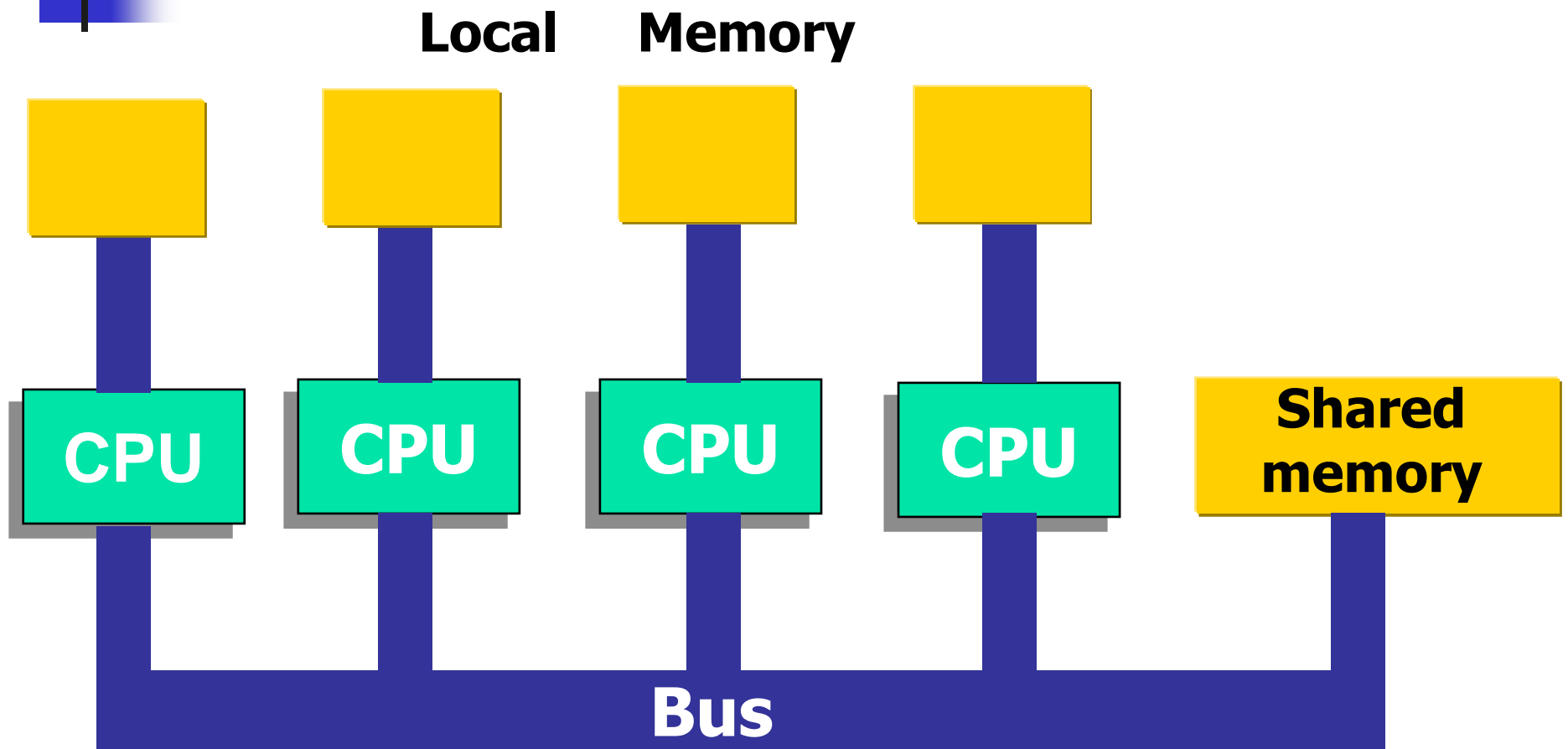


# Hệ đa bộ xử lý (MultiProcessor)



Hệ MultiProcessor sử dụng 1 đường Bus

# Hệ đa bộ xử lý (MultiProcessor)



Hệ MultiProcessor sử dụng nhiều bộ nhớ cục bộ



# Bus

**Bus là các đường truyền. Thông tin sẽ được chuyển qua lại giữa các thành phần linh kiện thông qua mạng lưới gọi là các Bus.**





## 2.3 Hệ thống Bus

---

**Các thiết bị ngoại vi kết nối với hệ thống nhờ các khe cắm mở rộng (expansion slot).**

**Bus hệ thống (Bus system) sẽ kết nối tất cả các thành phần lại với nhau.**

**Có 3 loại bus :bus dữ liệu (data bus), bus địa chỉ (address bus) và bus điều khiển (control bus).**



# Các loại Bus

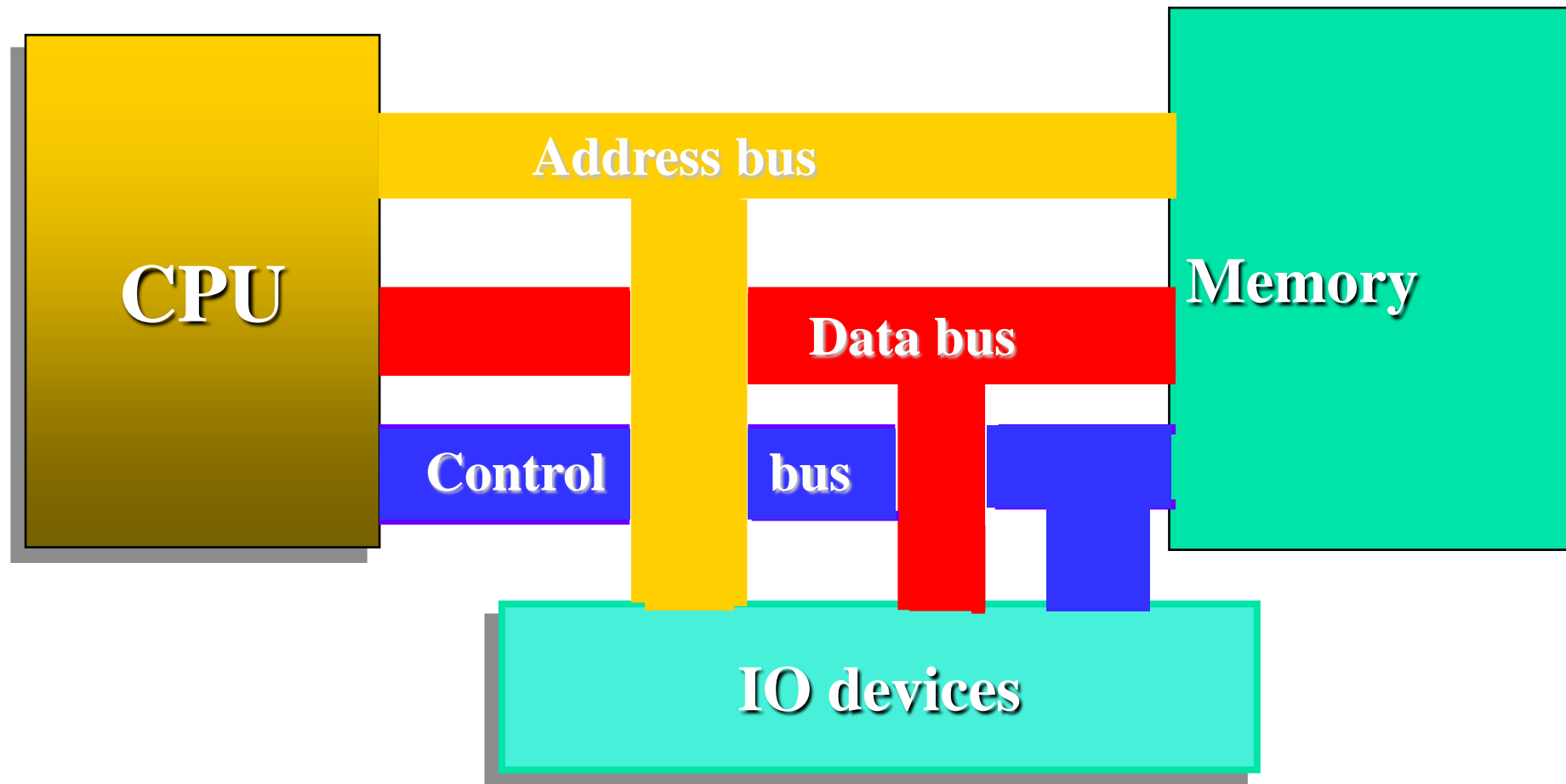
---

■ **Address Bus** : nhóm đường truyền nhận diện vị trí truy xuất trong thiết bị đích : thông tin được đọc từ đâu hoặc ghi vào đâu.

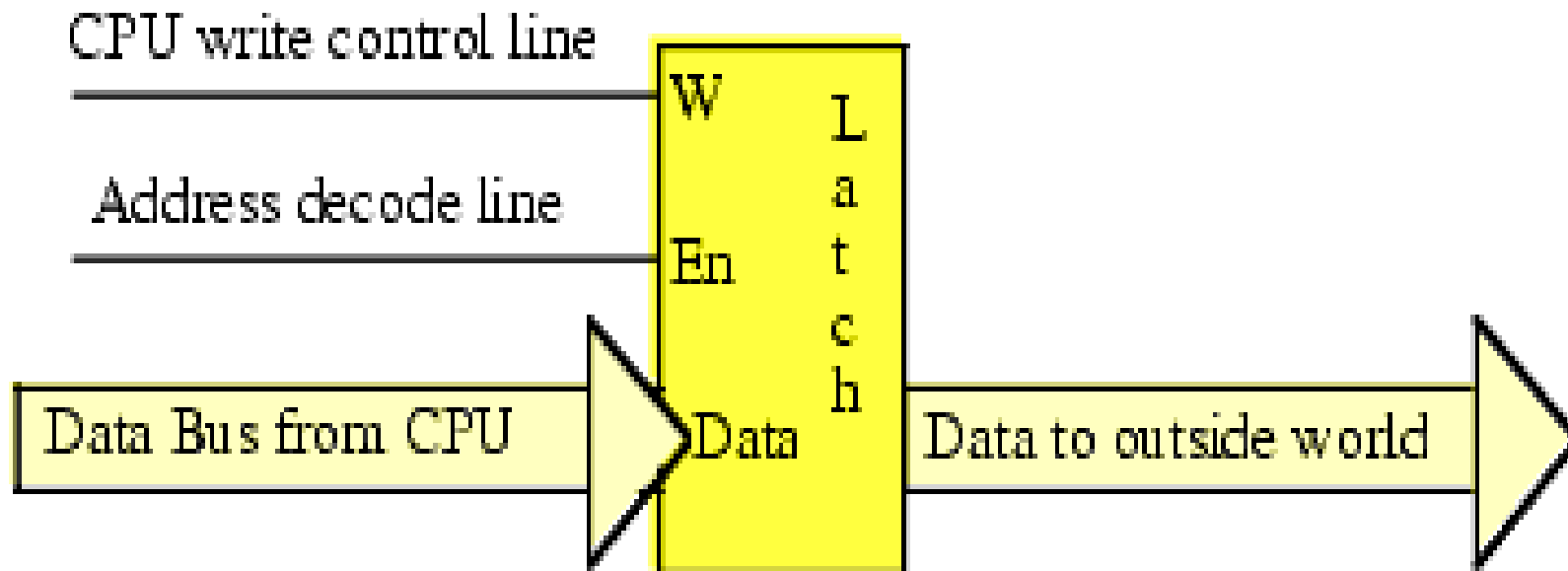
■ **Data Bus** : nhóm đường truyền để tải data thực sự giữa các thiết bị hệ thống do địa chỉ trên address bus đã xác định. Độ rộng của data bus (số đường dây dẫn) xác định data trong mỗi lần truyền là bao nhiêu.

■ **Control Bus** : nhóm đường truyền cho các tín hiệu điều khiển như : tác vụ là đọc hay ghi, tác vụ thực thi trên bộ nhớ hay trên thiết bị ngoại vi, nhận dạng chu kỳ bus và khi nào thì hoàn tất tác vụ...

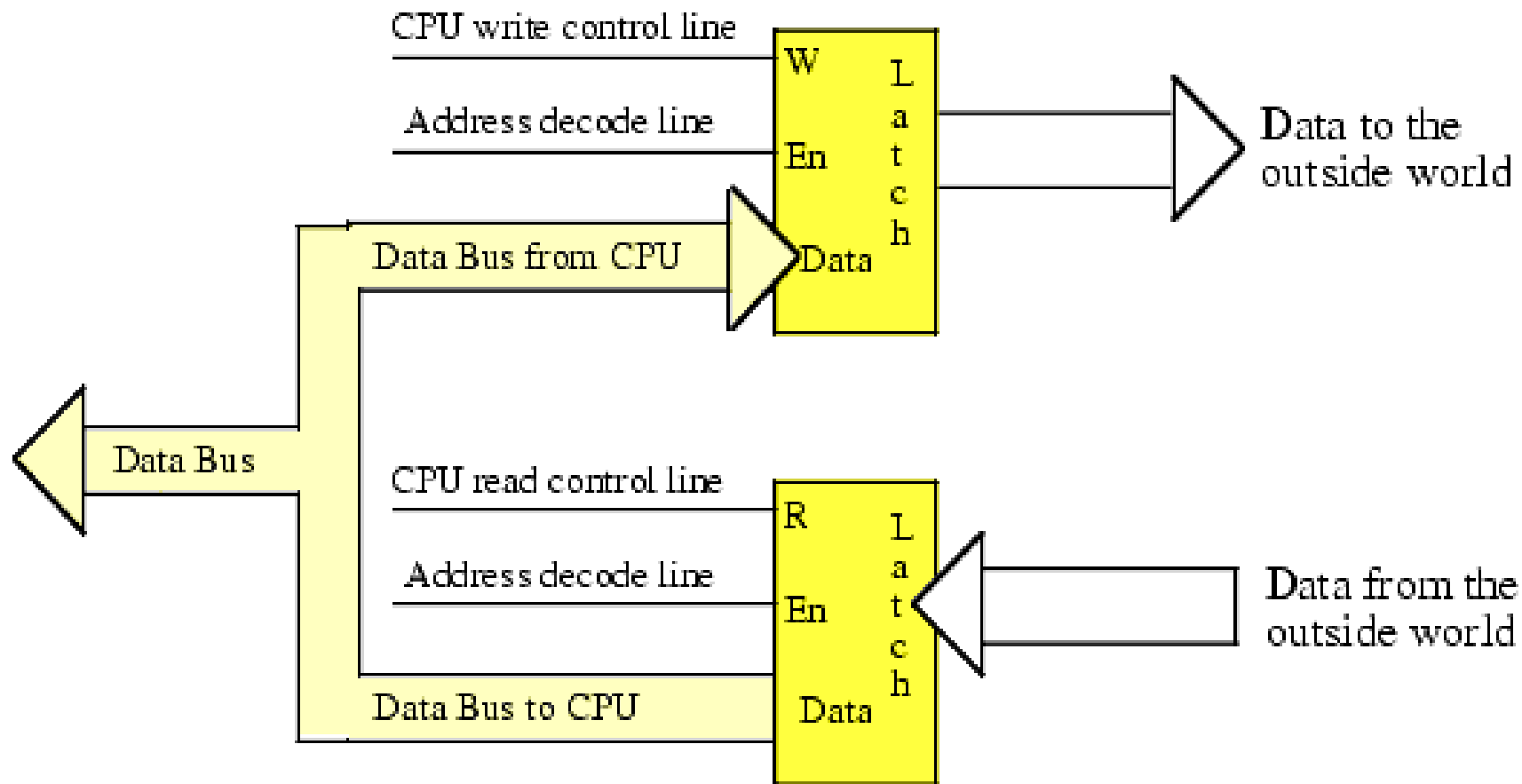
# Minh họa hệ thống Bus



# A Typical Output Port

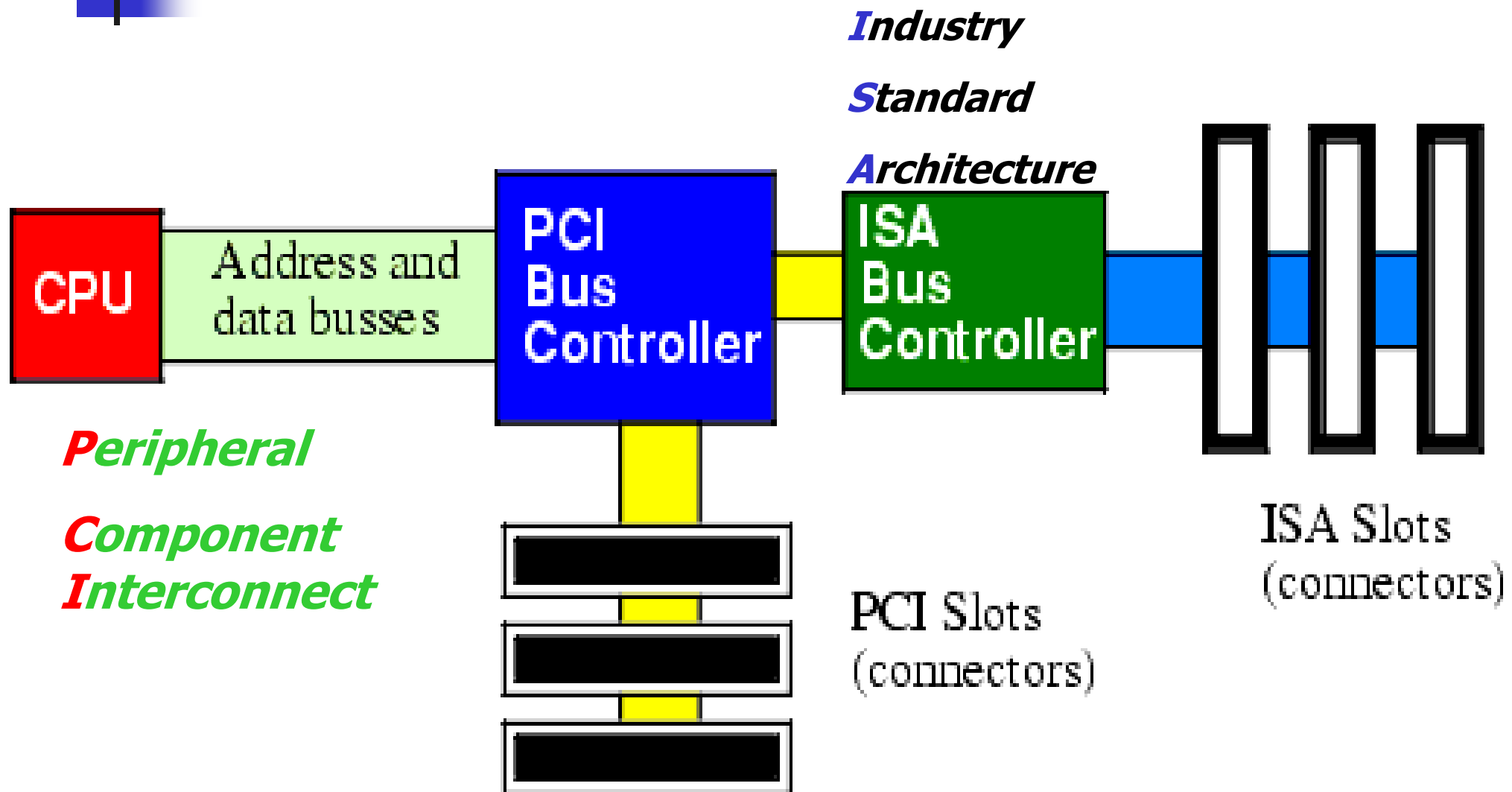


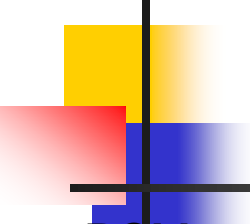
# An Input and an Output Device That Share the Same Address (a Dual I/O Port)





# Connection of the PCI and ISA Busses in a Typical PC





---

**PCI local bus n.** Short for Peripheral Component Interconnect local bus. A specification introduced by Intel Corporation that defines a local bus system that allows up to 10 PCI-compliant expansion cards to be installed in the computer. A PCI local bus system requires the presence of a PCI controller card, which must be installed in one of the PCI-compliant slots. Optionally, an expansion bus controller for the system's ISA, EISA, or Micro Channel Architecture slots can be installed as well, providing increased synchronization over all the system's bus-installed resources. The PCI controller can exchange data with the system's CPU either 32 bits or 64 bits at a time, depending on the implementation, and it allows intelligent, PCI-compliant adapters to perform tasks concurrently with the CPU using a technique called bus mastering. The PCI specification allows for multiplexing, a technique that permits more than one electrical signal to be present on the bus at one time.



# Bus PCI

---

*PCI chuẩn nối ghép các thiết bị ngoại vi với bộ VXL tốc độ cao của Intel như 486/Pentium*

- *Tốc độ tối đa 33MHz*
- *Data bus 32 bits và 64 bits*
- *Hỗ trợ cho 10 thiết bị ngoại vi*
- *Plug and Play*



# Plug and Play

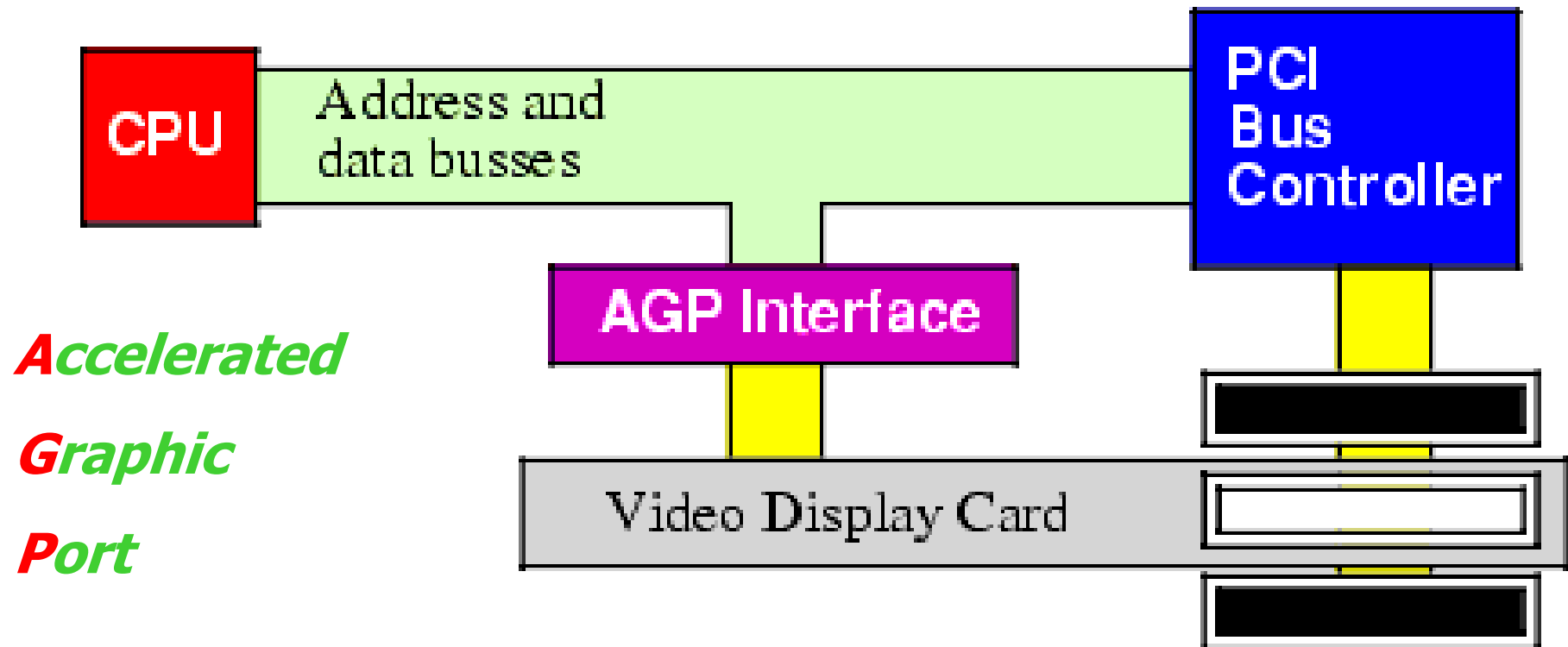
---

***1. Cả BIOS trên mainboard và Card bổ sung đều không phải là Plug and Play.***

***2. BIOS trên mainboard Plug and Play nhưng Card bổ sung thì không → phần mềm cài đặt sẽ giúp sắp xếp địa chỉ I/O, IRQ và các kênh DMA.***

***3. BIOS trên mainboard và Card bổ sung là Plug and Play → cấu hình tự động thực hiện mọi công việc.***

# AGP Bus Interface





## AGP (**Accelerated Graphics Port**)

---

Acronym for Accelerated Graphics Port. A high-performance bus specification designed for fast, high-quality display of 3-D and video images. Developed by Intel Corporation, AGP uses a dedicated point-to-point connection between the graphics controller and main system memory. This connection enables AGP-capable display adapters and compatible chip sets to transfer video data directly between system memory and adapter memory, to display images more quickly and smoothly than they can be displayed when the information must be transferred over the system's primary (PCI) bus. AGP also allows for storing complex image elements such as texture maps in system memory and thus reduces the need for large amounts of memory on the adapter itself. AGP runs at 66 MHz—twice as fast as the PCI bus—and can support data transfer speeds of up to 533 Mbps..



# Độ rộng Bus

---

**Độ rộng bus chính là số đường dây dẫn hợp thành bus.**

**Với address bus** : trên mỗi đường dây chỉ có thể có 1 trong 2 trạng thái 0 hoặc 1 nên bus có độ rộng  $n$  thì có thể nhận biết được  $2^n$  địa chỉ.

**Với data bus** : được thiết kế theo nguyên tắc là bội của 8 (8,16,32,64 bit) như thế mỗi lần truyền 1 byte/2 bytes/4 bytes tùy theo máy. Bề rộng Data bus càng lớn thì data truyền càng nhanh.



---

*Bus PC/XT có khe cắm 62 chân bao gồm :*

*Data bus D0-D7*

*Address Bus A0-A19*

*Các tín hiệu điều khiển ....*

*Bus PC/AT : bus XT + 36 chân nữa để làm việc với data bus 16 bit, bus địa chỉ 24 bit.*

*36 chân bổ sung được dùng làm các đường dữ liệu D8-D15, các đường địa chỉ A21-A23,...  
D0-D7 : là bus dữ liệu 8 bit, 2 chiều nối giữa bộ VXL với bộ nhớ, I/O.*





# Nhược điểm của Bus ISA

---

*Data bus bị hạn chế ở 16 bits → không thể phối hợp với data bus 32 bits của bộ VXL 386/486/Pentium.*

*Address bus địa chỉ 24 bits giới hạn khả năng truy cập bộ nhớ cực đại qua khe cắm mở rộng 16MB → không thể phối hợp được với bus địa chỉ 32 bit của 386/486/Pentium.*



# Chu kỳ Bus

---

**Mỗi chu kỳ bus là 1 tác vụ xảy ra trên bus để truyền tải data.**

**Mỗi lần CPU cần lệnh (hoặc data) từ bộ nhớ hoặc I/O, chúng phải thực thi 1 chu kỳ bus để có được thông tin hoặc ghi thông tin ra bộ nhớ hoặc ra I/O.**

**Mỗi chu kỳ bus gồm 2 bước :**

**bước 1 : gửi địa chỉ**

**bước 2 : truyền data từ địa chỉ đã được định vị.**



---

**4 chu kỳ bus cơ bản :**

**đọc bộ nhớ (memory Read)**

**ghi bộ nhớ (memory Write)**

**đọc I/O (I/O Read)**

**ghi I/O (I/O Write).**

**Các tín hiệu cần thiết để thực hiện các chu kỳ bus được sinh ra bởi CPU hoặc DMA Controller hoặc bộ làm tươi bộ nhớ.**



# Chu kỳ Bus

---

**Mỗi chu kỳ Bus đòi hỏi tối thiểu trọn vẹn 2 xung đồng hồ hệ thống.**

**Đây là mốc tham chiếu theo thời gian để đồng bộ hoá tất cả các tác vụ bên trong máy tính. Xung đầu tiên gọi là Address time , địa chỉ truy xuất sẽ được gửi đi cùng với tín hiệu xác định loại tác vụ sẽ được thực thi (đọc/ghi/đến mem/đến I/O).**



# Chu kỳ Bus

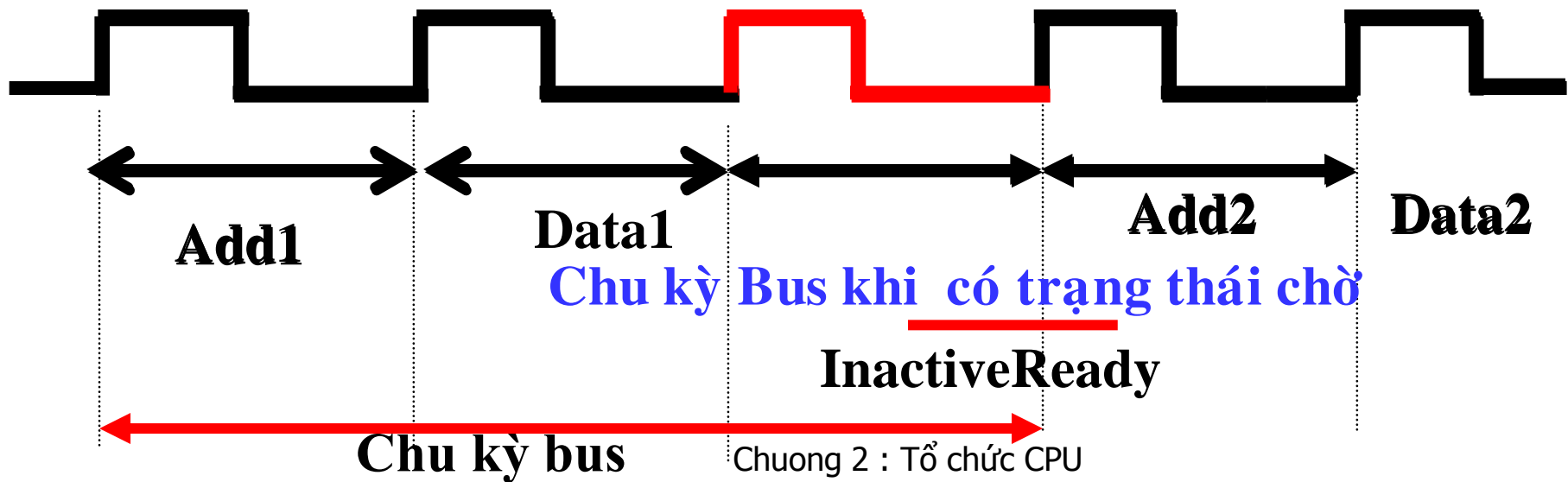
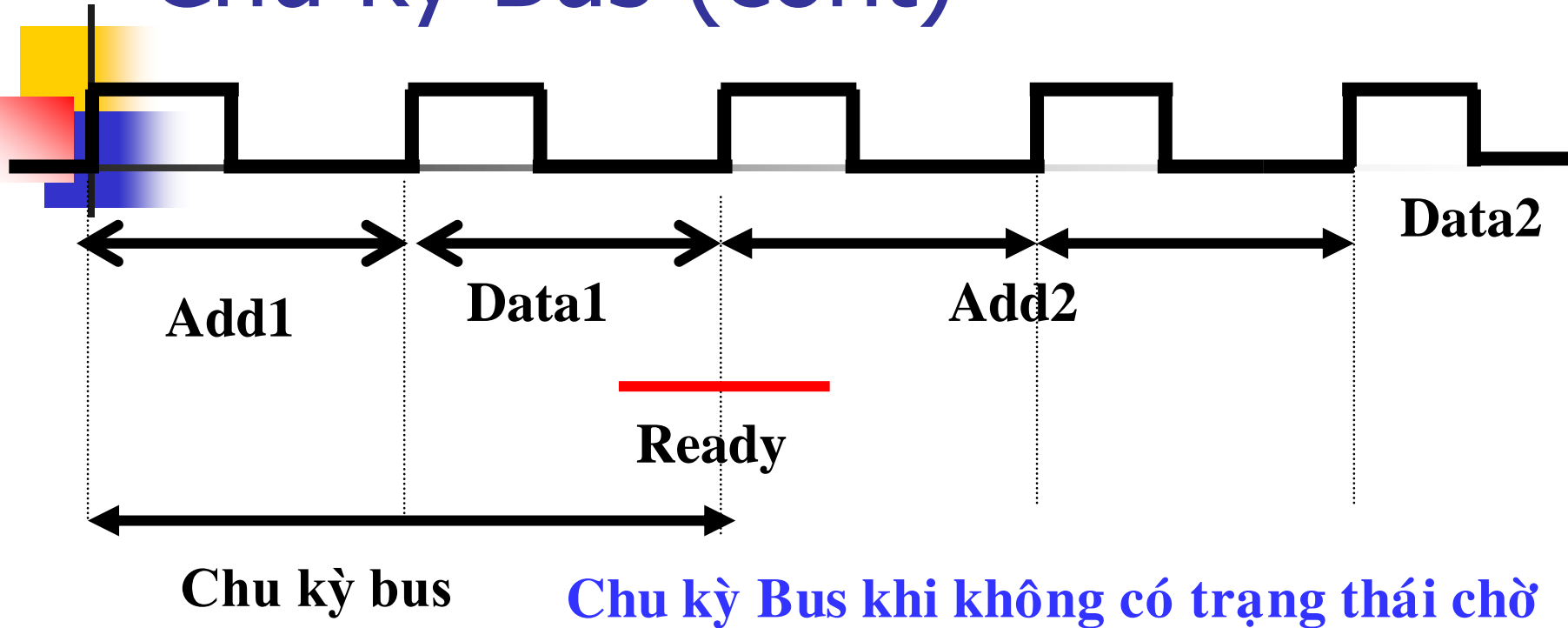
---

**Cuối xung thứ 2, CPU sẽ kiểm tra đường tín hiệu Ready.**

**Nếu thiết bị cần truy xuất sẵn sàng đáp ứng tác vụ, thiết bị này sẽ kích 1 tín hiệu lên đường Ready để báo cho CPU biết và chu kỳ bus hoàn tất.**

**Khi 1 thiết bị không sẵn sàng, không có tín hiệu trên đường Ready, CPU phải chờ, có thể phải tiêu tốn thêm 1 hay nhiều xung clock.**

# Chu kỳ Bus (cont)





# Chu kỳ Bus (cont)

---

**Chú ý :**

**Trong 1 số hệ thống, cho phép ta Setup một số **wait states** trong phần Extend Setup của Bios.**

**Nếu ta cho giá trị này nhỏ thì có thể ngoại vi không theo kịp CPU và hệ thống bị treo.**

**Còn nếu cho giá trị này lớn thì tốc độ chung của hệ thống bị chậm lại.**

**Wait states mặc định là 4 cho các vĩ mạch 8 bit và là 1 cho các vĩ mạch 16 bit.**

**tốc độ truyền tải tối đa :**

**$\text{tốc độ truyền tải} = \text{tốc độ bus (MHz)} \times \text{số bytes trong 1 lần truyền} / \text{số chu kỳ xung clock cho mỗi lần truyền}$**



## 2.4 Hệ thống thanh ghi

---

■ Là các phần tử có khả năng lưu trữ thông tin với dung lượng 8, 16 , 32, 64 bit.

■ Được xây dựng từ các FlipFlop nên có tốc độ truy xuất rất nhanh.

**Phân loại thanh ghi :**

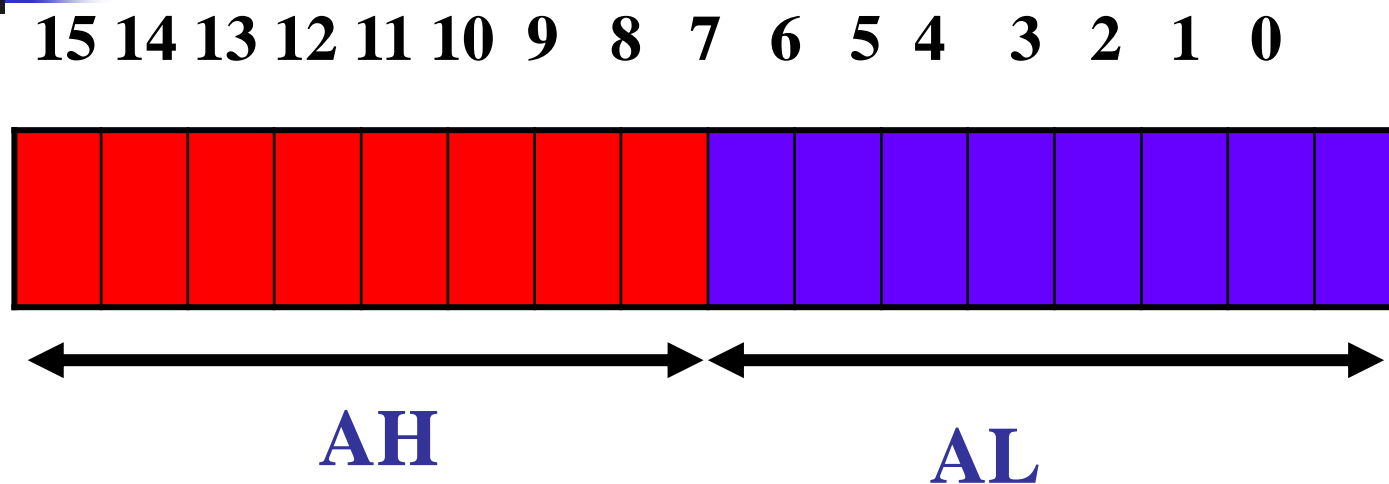
■ Thanh ghi tổng quát : chủ yếu dùng để lưu trữ dữ liệu trong quá trình thực thi CT, nhưng mỗi thanh ghi còn có 1 số chức năng riêng.

■ Thanh ghi điều khiển : các bit của nó qui định tác vụ của các đơn vị chức năng của MT.

■ Thanh ghi trạng thái : lưu trữ thông tin mô tả trạng thái.



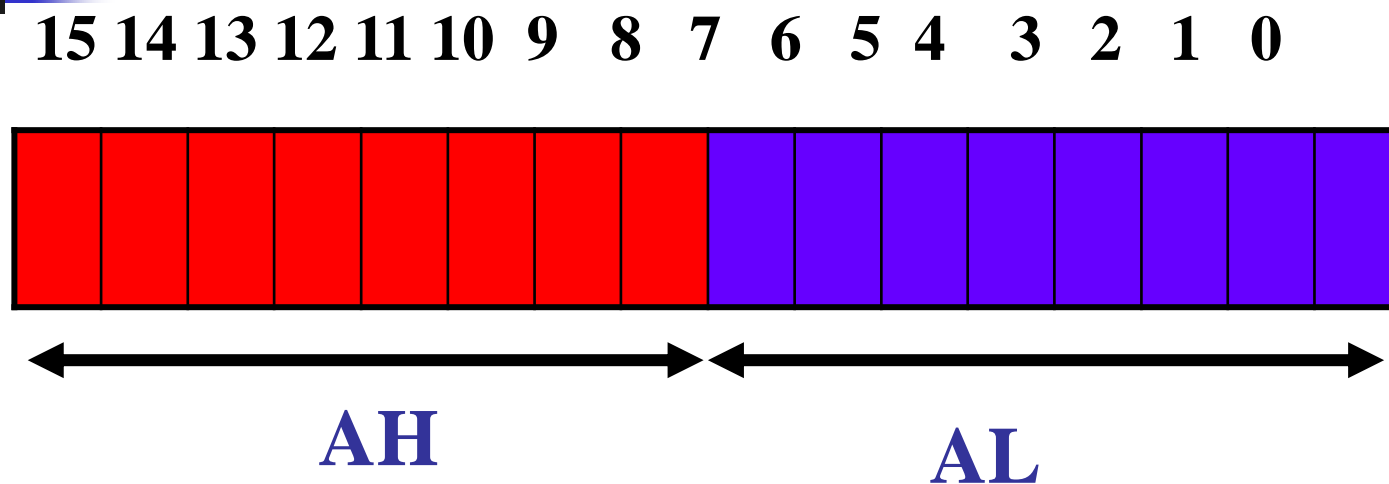
# AX Register



**Thanh ghi AX (Accumulator register) : thanh ghi tích lũy, dài 16 bit nhưng nó cũng có thể chia làm 2 thanh ghi 8 bit AH và AL**

**AX ngoài chức năng lưu trữ dữ liệu, nó còn được CPU dùng trong phép toán số học như nhân, chia.**

# AX Register

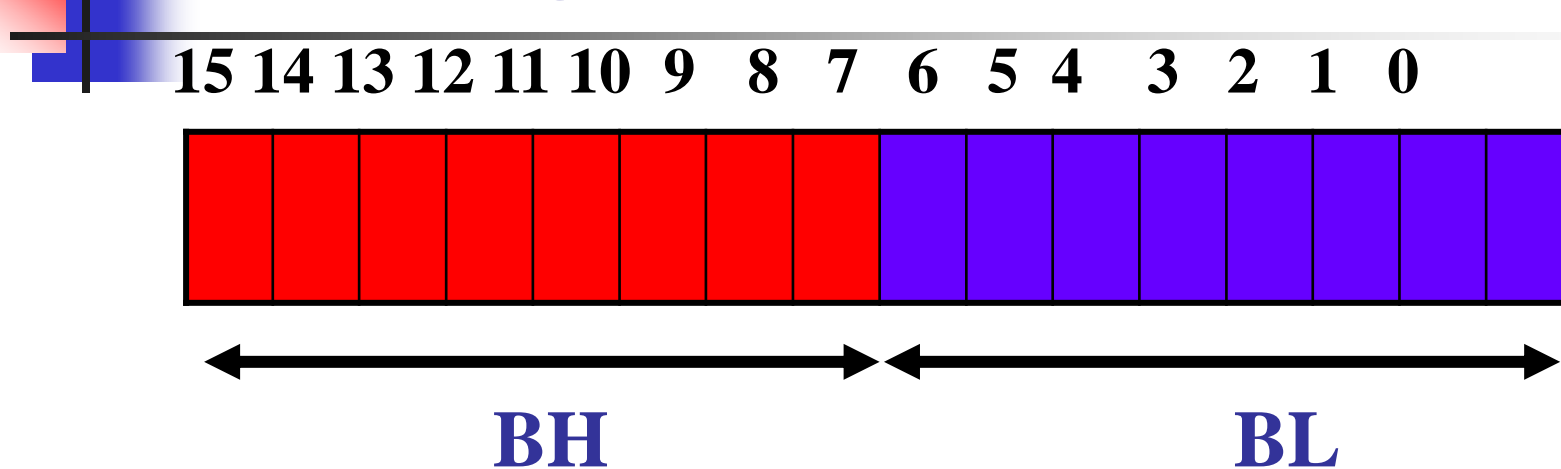


**Thanh ghi AH là nửa cao của thanh ghi AX**

**Thanh ghi AL là nửa thấp của thanh ghi AX**

**Thí dụ nếu AX=1234h thì AH=12H AL=34h**

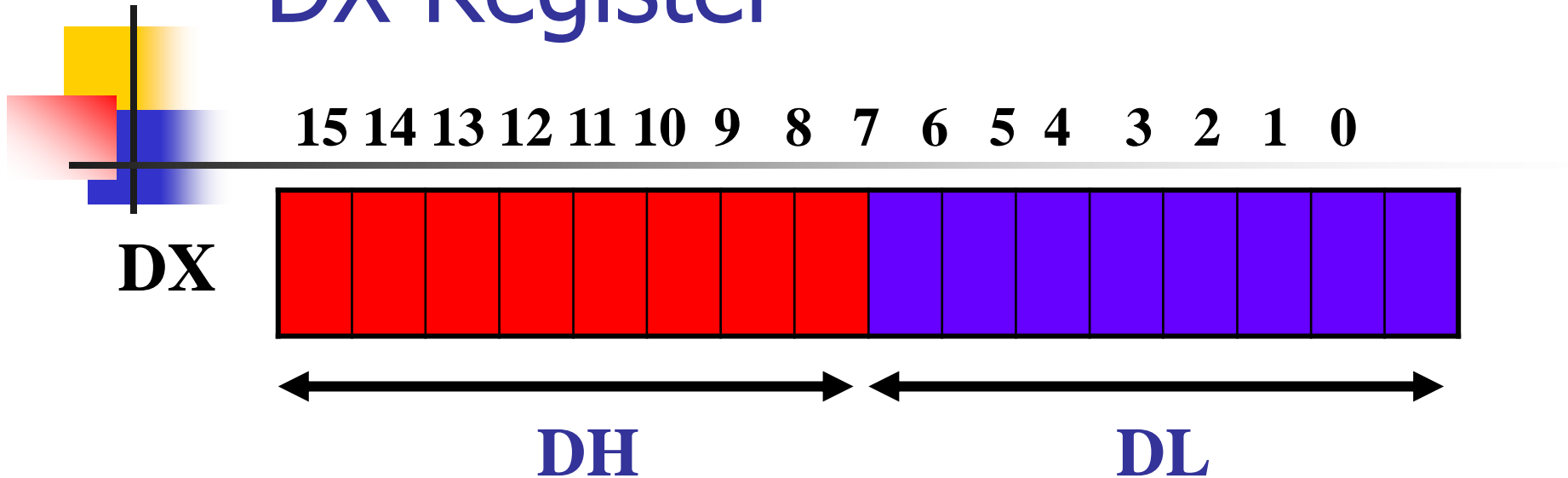
# BX Register



**Thanh ghi BX (Base register) : dài 16 bit nhưng nó cũng có thể chia làm 2 thanh ghi 8 bit BH và BL**

**BX lưu giữ địa chỉ của 1 thủ tục hay biến, nó cũng được dùng thực hiện các phép dời chuyển số học và dữ liệu.**

# DX Register

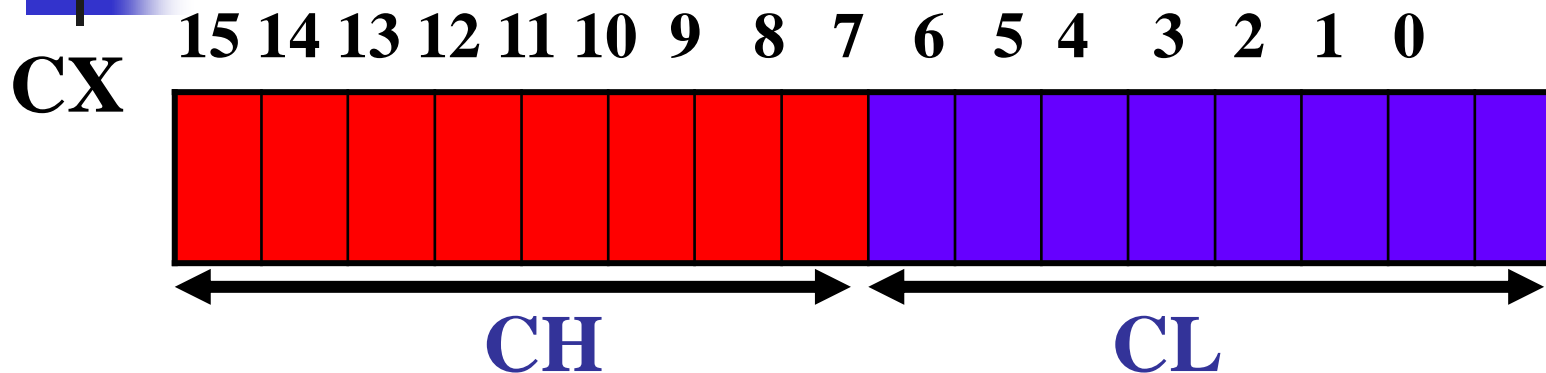


**Thanh ghi DX (Data register) : dài 16 bit nhưng nó cũng có thể chia làm 2 thanh ghi 8 bit DH và DL**

**Thanh ghi DX : có vai trò đặc biệt trong phép nhân và phép chia ngoài chức năng lưu trữ dữ liệu.**

**Ex : khi nhân DX sẽ lưu giữ 16 bit cao của tích.**

# CX Register



**CX (Counter register)** : thanh ghi này dùng làm bộ đếm trong các vòng lặp. Các lệnh tự động lặp lại và sau mỗi lần lặp giá trị của CX tự động giảm đi 1.

**CL** thường chứa số lần dịch, quay trong các lệnh dịch, quay thanh ghi

**CX** dài 16 bit, nó cũng có thể chia làm 2 thanh ghi 8 bit là **CH** và **CL**



# Các thanh ghi Segment

---

**CPU có 4 thanh ghi segment dài 16 bit, các thanh ghi này không thể chia làm 2 thanh ghi 8 bit như 4 thanh ghi AX, BX, CX và DX.**

**Các thanh ghi đoạn được sử dụng như là địa chỉ cơ sở của các lệnh trong chương trình, stack và dữ liệu.**

**4 thanh ghi đoạn : CS (Code Segment), DS (Data Segment), SS (Stack Segment) và ES (Extra Segment).**

**CS : chứa địa chỉ bắt đầu của code trong chương trình.**

**DS : chứa địa chỉ của các biến khai báo trong chương trình.**

**SS : chứa địa chỉ của bộ nhớ Stack dùng trong chương trình**

**ES : chứa địa chỉ cơ sở bổ sung cho các biến bộ nhớ.**



# Thanh ghi 32 bit

---

- Đối với một số CPU đời mới, có các thanh ghi dài 32, 64 bit. Ta ghi thêm E đứng trước tên các thanh ghi 16 bit...

**EAX, EBX, ECX, EDX ....**



## 2.5 Thanh ghi đoạn và sự hình thành địa chỉ

---

■ 8088 sử dụng 20 bit để đánh địa chỉ bộ nhớ → quản lý trên 1Mb bộ nhớ. Nhưng 8088 lại không có thanh ghi nào 20 bit, tất cả là 16 bit do đó 1 thanh ghi chỉ có thể đánh địa chỉ tối đa là 64 kB bộ nhớ.

■ Như vậy phải kết hợp 2 thanh ghi mới địa chỉ hoá toàn bộ bộ nhớ. 8088 sử dụng 1 trong các thanh ghi dùng chung và 1 trong các thanh ghi đoạn (CS,DS,SS,ES) để tạo thành 1 địa chỉ 20 bit.





# SỰ PHÂN ĐOẠN BỘ NHỚ

---

*CPU 8086 dùng phương pháp phân đoạn bộ nhớ để quản lý bộ nhớ 1MB của nó.*

*Địa chỉ 20 bit của bộ nhớ 1MB không thể chứa đủ trong các thanh ghi 16 bit của CPU 8086 → bộ nhớ 1MB được chia ra thành các đoạn (segment) 64KB.*

*Địa chỉ trong các đoạn 64KB chỉ có 16 bit nên CPU 8086 dễ dàng xử lý bằng các thanh ghi của nó.*

*→ PHÂN ĐOẠN BỘ NHỚ : là cách dùng các thanh ghi 16 bit để biểu diễn cho địa chỉ 20 bit.*



## 2.5 Địa chỉ vật lý & địa chỉ luận lý

---

***Địa chỉ 20 bits được gọi là địa chỉ vật lý.***

***Địa chỉ vật lý dùng như thế nào ?***

***Dùng trong thiết kế các mạch giải mã địa chỉ cho bộ nhớ và xuất nhập.***

***Còn trong lập trình , địa chỉ vật lý không thể dùng được mà nó được thay thế bằng địa chỉ luận lý (logic).***



# Địa chỉ luận lý

---

*Địa chỉ của 1 ô nhớ được xác định bởi 2 phần:*

*Segment : offset* —→ *Địa chỉ trong  
đoạn (độ dài)*

↑  
*Địa chỉ đoạn*

*Ex : B001:1234*

*Mỗi địa chỉ thành phần là 1 số 16 bit và được viết  
theo cách sau :*

*Segment : offset*



# Sự hình thành địa chỉ

---

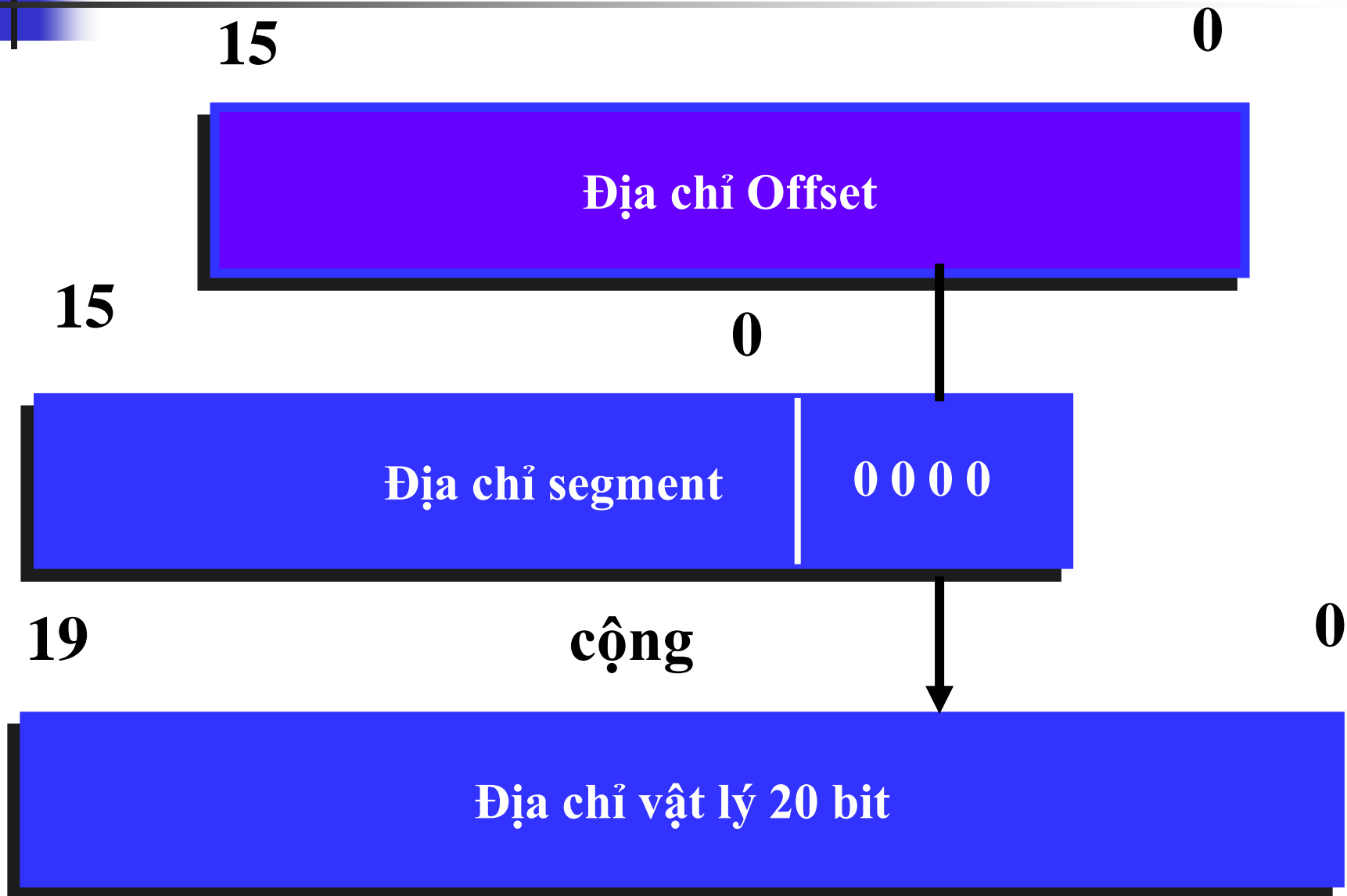
**Hãng Intel đề xuất 1 phương pháp để hình thành địa chỉ.**

**Mỗi địa chỉ ô nhớ được hình thành từ 1 phép tính tổng 1 địa chỉ cơ sở và 1 địa chỉ offset.**

**Địa chỉ cơ sở lưu trong 1 thanh ghi segment, còn địa chỉ offset nằm trong 1 thanh ghi chỉ số hay thanh ghi con trỏ.**

**Phép cộng này sẽ tạo 1 địa chỉ 20 bit gọi là địa chỉ vật lý.**

# Thí dụ minh họa hình thành địa chỉ



# Sự hình thành địa chỉ tuyệt đối

địa chỉ  
segment

địa chỉ Offset

Giả sử ta có địa chỉ **08F1 : 0100**

địa chỉ tương đối

CPU tự động lấy địa chỉ segment x 10 (hệ 16) thành **08F10**

Sau đó nó cộng với địa chỉ Offset **0100**

➔ địa chỉ tuyệt đối : **09010**



# Cách tính địa chỉ vật lý từ địa chỉ luận lý

$$\text{Địa chỉ vật lý} = (\text{segment} * 16) + \text{offset}$$

Segment	0
---------	---

+

offset
--------

---

Địa chỉ vật lý
----------------

*Ex : tính địa chỉ vật lý tương ứng địa chỉ luận lý B001:1234*

$$\text{Địa chỉ vật lý} = B0010h + 1234h = B1244h$$



# Sự chồng chất các đoạn

---

*Địa chỉ segment hay còn gọi là địa chỉ nền của đoạn. Nó cho biết điểm bắt đầu của đoạn trong bộ nhớ.*

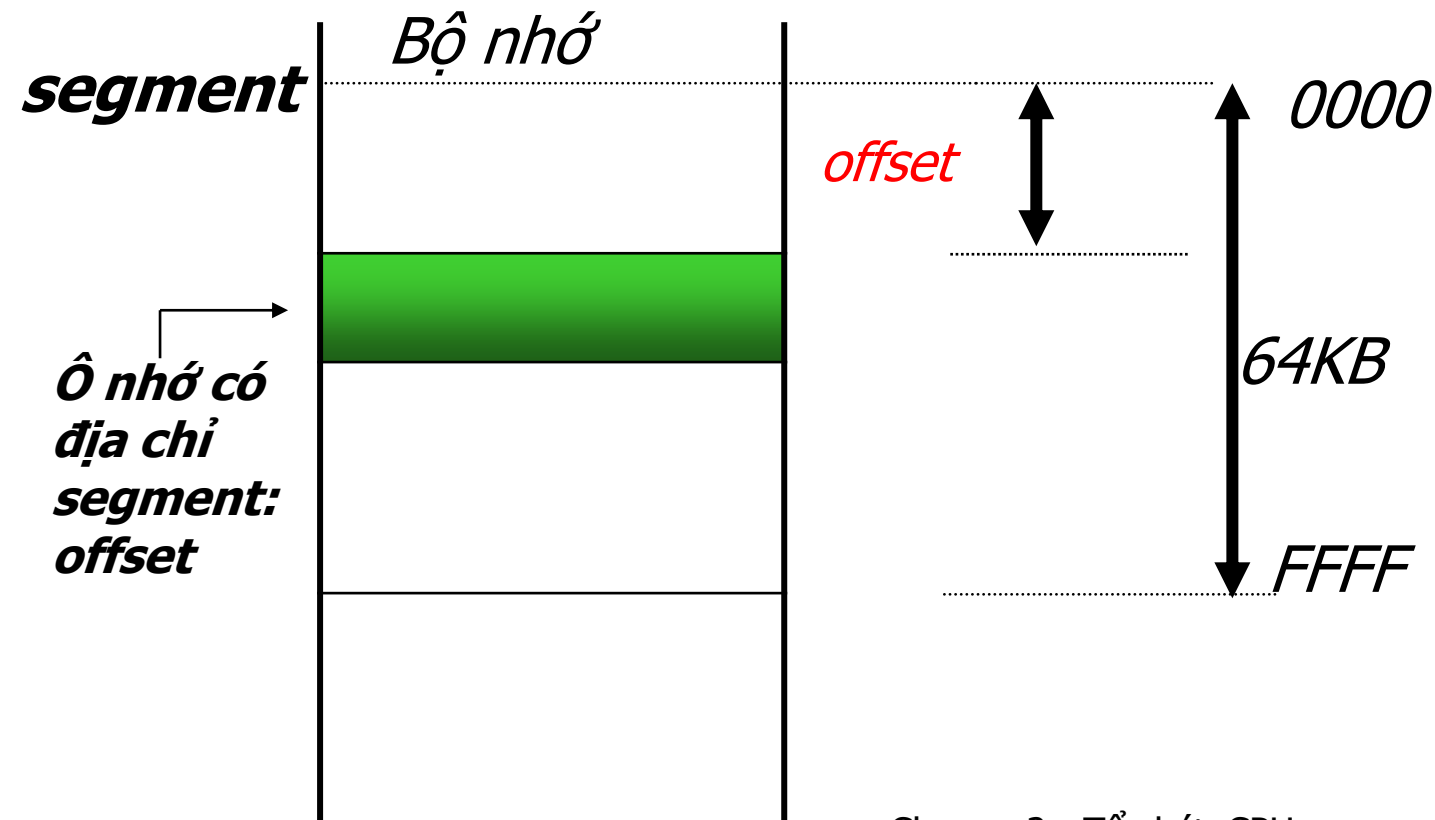
*Địa chỉ offset thể hiện khoảng cách kể từ đầu đoạn của ô nhớ cần tham khảo.*


*Do offset dài 16 bit nên chiều dài tối đa của mỗi đoạn là 64K.*



# Sự chồng chất các đoạn

*Trong mỗi đoạn, ô nhớ đầu tiên có offset là 0000h và ô nhớ cuối cùng là FFFFh.*





---

*Mỗi ô nhớ chỉ có địa chỉ vật lý nhưng có thể có nhiều địa chỉ luận lý.*

*Ex : 1234:1234*

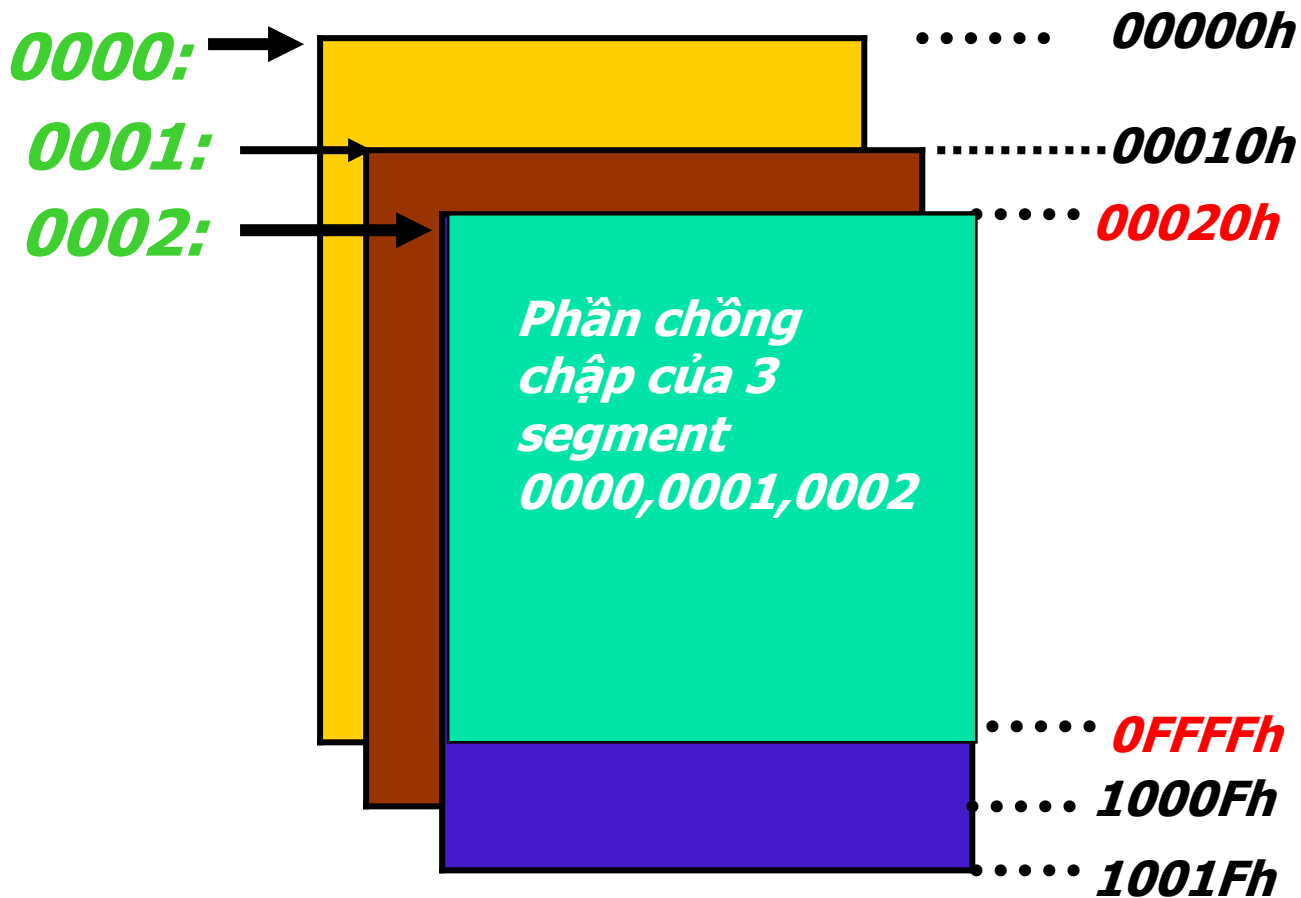
*1334:0234*

*1304:0534*

*Đều có chung địa chỉ vật lý 13574h*

***Tại sao ?***

Để hiểu rõ tại sao ta hãy xét mối quan hệ giữa địa chỉ vật lý với segment và offset





## Giải thích

---

*0000:0000 → 00000h*

*Giữ nguyên phần segment, tăng phần offset lên 1 thành ra địa chỉ luận lý là 0000:0001*

*Địa chỉ vật lý tương ứng là 00001h*

*Tương tự với địa chỉ luận lý là 0000:0002 ta có địa chỉ vật lý là 00002h*

*Khi offset tăng 1 đơn vị thì địa chỉ vật lý tăng 1 địa chỉ hoặc là tăng 1 byte.  
Như vậy có thể xem đơn vị của offset là byte*



---

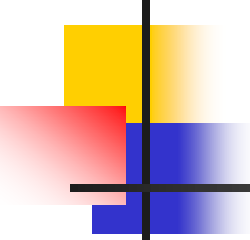
*Làm lại quá trình trên nhưng giữ nguyên phần offset chỉ tăng phần segment.*

*0001:0000 → 00010h*

*0002:0000 → 00020h*

*Khi segment tăng 1 đơn vị thì địa chỉ vật lý tăng 10h địa chỉ hoặc là tăng 16 bytes*

*Đơn vị của segment là paragraph*



---

***Ta thấy segment 0000 nằm ở đầu vùng nhớ nhưng segment 0001 bắt đầu cách đầu vùng nhớ chỉ có 16 bytes, segment 0002 bắt đầu cách đầu vùng nhớ 32 bytes.....***

***Phần chồng chập 3 segment 0000,0001,0002 trên hình vẽ là vùng bộ nhớ mà bất kỳ ô nhớ nào nằm trong đó (địa chỉ vật lý từ 00020h đến 0FFFFh) đều có thể có địa chỉ luận lý tương ứng trong cả 3 segment.***



---

*Ex : ô nhớ có địa chỉ 0002Dh sẽ có địa chỉ logic trong segment 0000 là 0000:002D*

***Trong segment 0001 là 0001:001D***

***Trong segment 0002 là 0002:000D***

***→ nếu vùng bộ nhớ nào càng có nhiều segment chồng chập lên nhau thì các ô nhớ trong đó càng có nhiều địa chỉ luận lý.***



***Một ô nhớ có bao nhiêu  
địa chỉ luận lý***

---

***Một ô nhớ có ít nhất 1 địa chỉ  
luận lý và nhiều nhất là  
 $65536/16 = 4096$  địa chỉ luận lý***





## Các thanh ghi đoạn CS, DS, SS, ES

---

- 3 trong 4 thanh ghi đoạn được dùng trong các mục đích đặc biệt sau
- **CS** : xác định đoạn lệnh – nơi chứa chương trình được thi hành.
- **DS** : xác định đoạn dữ liệu – nơi chứa chương trình được thi hành.
- **SS** : xác định đoạn stack – vùng làm việc tạm thời dùng để theo dõi các tham số và các địa chỉ đang được chương trình hiện hành sử dụng.
- Còn thanh ghi ES : trỏ đến đoạn thêm, thường được dùng để bổ sung cho đoạn dữ liệu → có vùng nhớ >64k cho đoạn dữ liệu.



## Các thanh ghi đoạn CS, DS, SS, ES

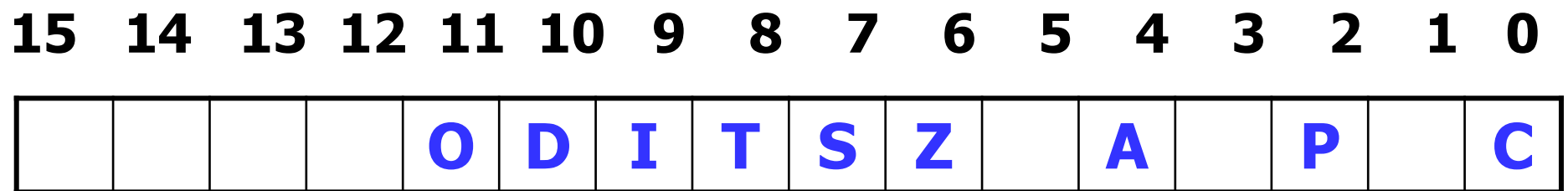
---

- 3 trong 4 thanh ghi đoạn được dùng trong các mục đích đặc biệt sau
- **CS** : xác định đoạn lệnh – nơi chứa chương trình được thi hành.
- **DS** : xác định đoạn dữ liệu – nơi chứa chương trình được thi hành.
- **SS** : xác định đoạn stack – vùng làm việc tạm thời dùng để theo dõi các tham số và các địa chỉ đang được chương trình hiện hành sử dụng.
- Còn thanh ghi ES : trỏ đến đoạn thêm, thường được dùng để bổ sung cho đoạn dữ liệu → có vùng nhớ >64k cho đoạn dữ liệu.



## Thanh ghi trạng thái (thanh ghi cờ)

- Thanh ghi cờ là thanh ghi 16 bit nằm bên trong EU (Excution Unit). Tuy nhiên chỉ có 9 trong 16 bit được sử dụng. 7 bit còn lại không dùng.



O OverFlow flag

D : Direction flag

I : Interrupt flag

T : Trap flag

S : Sign flag

Z : Zero flag

A : Auxiliary flag

P : Parity flag

C : Carry flag



## Thanh ghi trạng thái (thanh ghi cờ)

---

**Giải thích :**

**Cờ CF : chỉ thị cộng có nhớ, trừ có mượn.**

**Cờ PF : On khi kết quả của tác vụ có số bit 1 là số chẵn.**

**Nếu số bit 1 là số lẻ thì PF là Off.**

**Cờ AF : có nhớ trong phép cộng hoặc có mượn trong phép trừ với 4 bit thấp sang 4 bit cao.**

**Cờ ZF : On khi tác vụ luận lý cho kết quả là 0.**

**Cờ SF : bit cao nhất của kết quả sẽ được copy sang SF.  $SF = 1$  kết quả là số âm.  $SF = 0$  khi kết quả là số dương.**



## Thanh ghi trạng thái (thanh ghi cờ)

**Giải thích :**

**Cờ OF :  $OF=1$  khi kết quả bị tràn số (vượt quá khả năng lưu trữ ).  
Nếu kết quả không bị tràn thì  $OF=0$ .**

**3 bit còn lại là 3 bit điều khiển :**

**Cờ TF : báo CPU thi hành từng bước. Cung cấp công cụ debug chương trình.**

**Cờ IF :  $IF=1$  giúp 8086 nhận biết có yêu cầu ngắt quãng có che.**

**Cờ DF : xác định hướng theo chiều tăng/giảm trong xử lý chuỗi.**

**8086 cho phép User lập trình bật tắt các cờ CF,DF,IF,TF**



## Thanh ghi chỉ số (Index)

---

5 thanh ghi offset dùng để xác định chính xác 1 byte hay 1 word trong 1 đoạn 64K. Đó là :

- IP : thanh ghi con trỏ lệnh, cho biết vị trí của lệnh hiện hành trong đoạn lệnh. Con trỏ lệnh IP còn được gọi là bộ đếm chương trình.

Thường được dùng kết hợp với CS để theo dõi vị trí chính xác của lệnh sẽ được thực hiện kế tiếp.

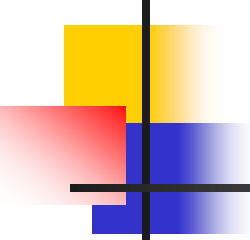


# Thanh ghi chỉ số (Index)

---

- Các thanh ghi con trỏ Stack : SP và BP, mỗi thanh ghi dài 16 bit.
- SP (Stack pointer) cho biết vị trí hiện hành của đỉnh Stack.
- BP (Basic Pointer) dùng để truy cập dữ liệu trong Stack.
- SI (source index) : trỏ đến ô nhớ trong đoạn dữ liệu được định địa chỉ bởi thanh ghi DS.
- DI (destination) : chức năng tương tự SI.  
Hai thanh ghi này thường dùng trong xử lý chuỗi.

# ĐỊA CHỈ LUẬN LÝ VÀ THANH GHI



***Để tham khảo đến bộ nhớ trong chương trình, VXL 8086 cho phép sử dụng các địa chỉ luận lý 1 cách trực tiếp hoặc thông qua các thanh ghi của nó.***

*Thanh ghi đoạn dùng để chứa segment*

*Thanh ghi tổng quát dùng để chứa địa chỉ trong đoạn offset*

***Để tham khảo đến địa chỉ luận lý có segment trong thanh ghi DS, offset trong thanh ghi BX, ta viết **DS:BX*****

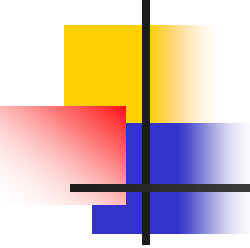




---

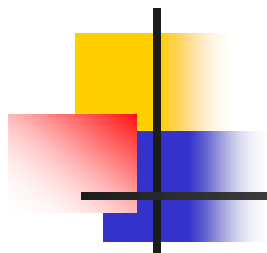
***Ex : nếu lúc tham khảo***

***DS = 2000h BX = 12A9h thì địa chỉ luận lý  
DS:BX chính là tham khảo đến ô nhớ  
2000:12A9***



*Trong cách sử dụng địa chỉ luận lý thông qua các thanh ghi có 1 số cặp thanh ghi luôn phải dùng chung với nhau 1 cách bắt buộc :*

***CS:IP*** lấy lệnh (địa chỉ lệnh sắp thi hành)  
***SS:SP*** địa chỉ đỉnh Stack  
***SS:BP*** thông số trong Stack  
(dùng trong chương trình con)  
***DS:SI*** địa chỉ chuỗi nguồn  
***ES:DI*** địa chỉ chuỗi đích



*Chương trình mà VXL 8086 thi hành thường có 3 đoạn :*

*Đoạn chương trình có địa chỉ trong thanh ghi CS.*

*Đoạn dữ liệu có địa chỉ trong thanh ghi DS.*

*Đoạn stack có địa chỉ trong thanh ghi SS.*



## Các đặc tính của CPU Intel

---

- Hiệu quả của CPU thuộc họ Intel khi xử lý và chuyển giao thông tin được xác định bởi các yếu tố sau :
- Tần số mạch xung đồng hồ của CPU.
- Độ rộng của Data bus
- Độ rộng của Address bus



## Các đặc tính của CPU Intel

---

- Tần số mạch xung đồng hồ của CPU.càng nhanh thì tốc độ xử lý càng nhanh.
- Độ rộng của Data bus càng rộng thì càng nhiều data được chuyển giao trong 1 lần giao dịch.
- Độ rộng của Address bus càng rộng thì khả năng quản lý bộ nhớ càng lớn.



## Các đặc tính của CPU Intel

Loại CPU	Data Bus ( bit )	Address bus ( bit )	Khả năng quản lý bộ nhớ
<b>8088</b>	<b>8</b>	<b>20</b>	<b>1 MB</b>
<b>8086</b>	<b>16</b>	<b>20</b>	<b>1MB</b>
<b>80286</b>	<b>16</b>	<b>24</b>	<b>16Mb</b>
<b>80386</b>	<b>32</b>	<b>32</b>	<b>4 GB</b>
<b>80486</b>	<b>32</b>	<b>32</b>	<b>4 GB</b>
<b>Pentium</b>	<b>64</b>	<b>32</b>	<b>4GB</b>
Chương 2 : Tổ chức CPU			86



# Tóm tắt CPU họ Intel

---

- CPU 80286 : Data bus 16 bit nên mỗi lần chuyển giao 2 bytes → quản lý 16MB bộ nhớ.

Chỉ có khả năng thực hiện các phép toán đối với các số nguyên, có thể dùng tập lệnh 80286 để mô phỏng các phép toán số học dấu chấm động nhưng điều này sẽ làm giảm hiệu suất hệ thống.

- Nếu muốn có khả năng thực hiện các phép toán dấu chấm động phải gắn CoProcessor 8087.

80286 làm việc theo 2 chế độ : chế độ thực và chế độ bảo vệ.



# Tóm tắt CPU họ Intel

---

- CPU 80386 : Data bus 32 bit nên có thể quản lý 4GB bộ nhớ.  
Các thanh ghi dài 32 bit → tăng độ chính xác của các phép toán.  
Độ rộng Bus → tăng tốc độ thực thi.

**CPU 80386 hoàn toàn tương thích với các CPU trước nó.**





# Tóm tắt CPU họ Intel

---

- CPU 80486 : có bus 32 bit . 1 Coprocessor 387, bộ phận điều khiển Cache, 1 Cache 8K, dùng phối hợp tập lệnh rút gọn RISC và tập lệnh phức tạp CISC.

**CPU 80486 phần lớn các lệnh chỉ dùng 1 số ít xung.**

**Sử dụng cơ chế đường ống có khả năng xử lý 5 lệnh đồng thời :**

- **Lấy lệnh trước PreFetch**
- **Giải mã lần 1 Decode 1**
- **Giải mã lần 2 Decode 2**
- **Thực thi lệnh Execution**
- **Ghi lại trạng thái. WriteBack**



# RISC & CISC

---

## ■ Nguyên lý CISC :

Complex Intruction Set Computer

- Tập lệnh khá lớn >300 lệnh
- Khả năng định vị phức tạp
- Một số lệnh cần phải vi lệnh hoá

quá nhiều lệnh → nạp lâu → làm chậm hệ thống

lệnh phức tạp → nên time giải mã lệnh nhiều khi lớn hơn time thực thi.

Chỉ có hơn 20% lệnh thường dùng tới



# RISC & CISC

---

**Nguyên lý RISC : tập lệnh thu gọn**

**R**educe **I**nstruction **S**et **C**omputer

**tập lệnh nhỏ → thi hành ngay không cần giải mã.**

**lệnh làm việc theo cơ chế đường ống (pipeline).**

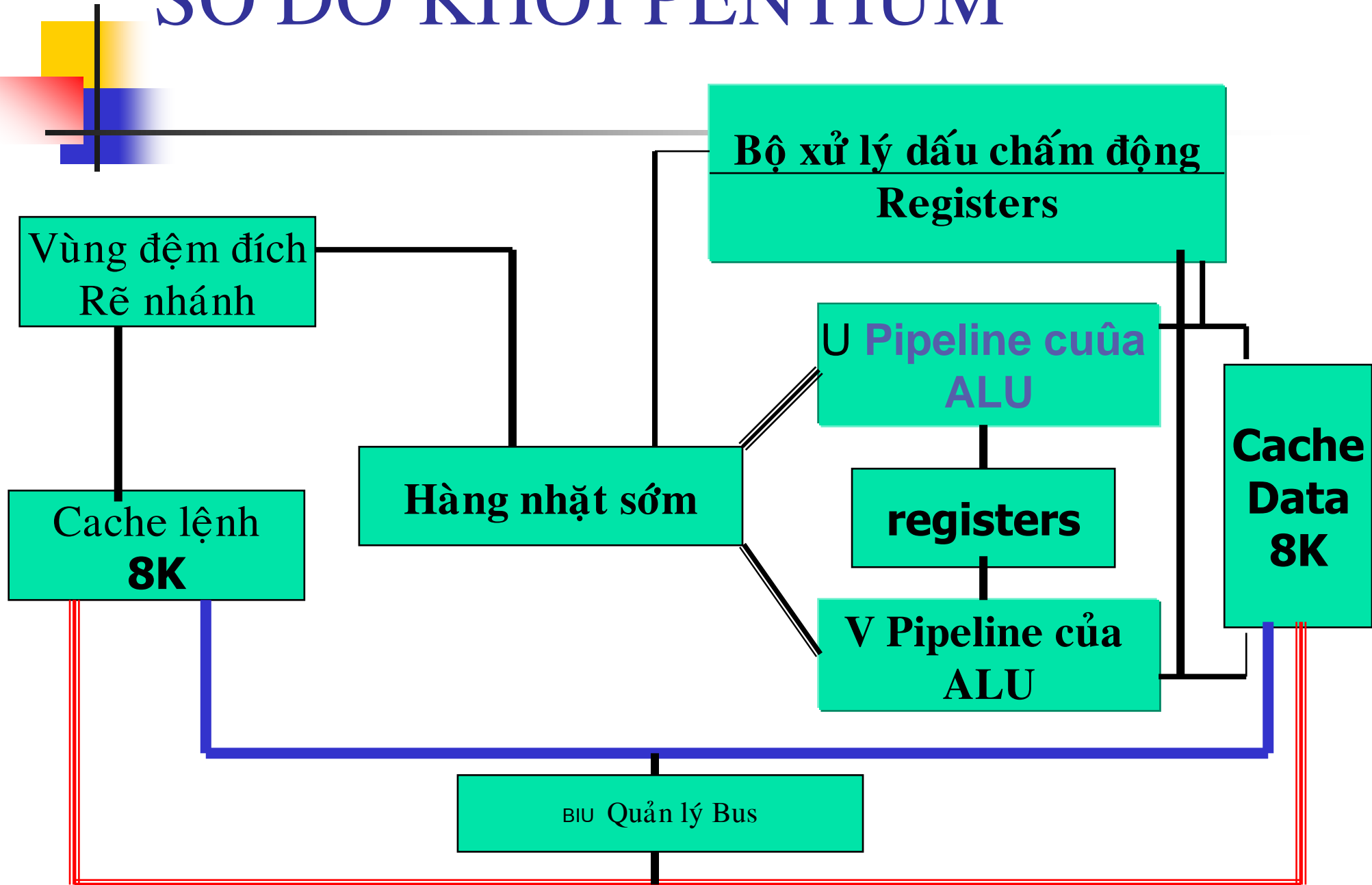


# CPU Pentium

---

- 3 thành phần góp sức tăng tốc độ xử lý của Pentium :
  - Đơn vị tính toán số nguyên supercallar
  - Bộ nhớ Cache cấp 1 ở bên trong CPU.
  - Đơn vị tính toán số chấm động supercallar

# SƠ ĐỒ KHỐI PENTIUM





# Câu hỏi ôn tập

---

- Bus là gì? Trong các loại Bus, Bus nào là Bus 2 chiều.
- Cho 1 ô nhớ có địa chỉ vật lý là 1256H, cho biết địa chỉ dạng segment:offset với các đoạn 1256H và 1240H.
- Ô nhớ có địa chỉ vật lý 80FD2H, ở trong đoạn nào thì nó có offset = BFD2H?
- Xác định địa chỉ vật lý của ô nhớ có địa chỉ logic 0A51H:CD90H



# Câu hỏi ôn tập

---

- Thế nào là biên giới đoạn?
- Sự khác nhau cơ bản giữa bộ vi xử lý 8086 và 80286?
- Thuyết minh trình tự CPU thực hiện câu lệnh  $\text{Mem}(b) \leftarrow \text{Not Mem}(a)$
- Chu kỳ lệnh, chu kỳ máy. Cho biết quan hệ giữa chu kỳ clock, chu kỳ máy và chu kỳ lệnh.
- Quan hệ giữa tập lệnh và kiến trúc của CPU



## Câu hỏi ôn tập

---

- Giải thích tại sao khi tăng tần số xung clock, giảm chu kỳ wait state của bộ nhớ, thêm cache cho CPU lại làm cho hệ thống chạy với hiệu suất cao hơn. ?
- Trình bày chiến lược chính lưu trữ thông tin trong Cache?
- Tính tốc độ chuyển giao dữ liệu của máy tính có CPU 486DX-66MHz và máy Pentium 100MHz.
- Phân biệt RISC và CISC.
- Trình bày cơ chế đường ống trong thực thi của CPU

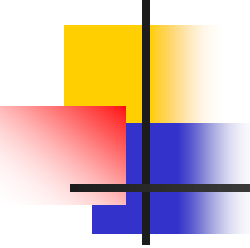




---

*Bus ISA-8 bits :*

- a. chạy ở tốc độ đồng hồ là 8 MHz truyền tải dữ liệu tối đa 8 MB/s.*
- b. chạy ở tốc độ đồng hồ là 4.77 MHz truyền tải dữ liệu tối đa 6MB/s.*
- c. chạy ở tốc độ đồng hồ là 4.77 MHz truyền tải dữ liệu tối đa 1MB/s.*
- d. chạy ở tốc độ đồng hồ là 4.77 MHz truyền tải dữ liệu tối đa 12MB/s.*



---

*Bus ISA-16 bits :*

- a. chạy ở tốc độ đồng hồ là 8→12 MHz truyền tải dữ liệu tối đa 8 MB/s.*
- b. chạy ở tốc độ đồng hồ là 32 MHz truyền tải dữ liệu tối đa 12MB/s.*
- c. chạy ở tốc độ đồng hồ là 4.77 MHz truyền tải dữ liệu tối đa 12MB/s.*
- d. chạy ở tốc độ đồng hồ là 16MHz truyền tải dữ liệu tối đa 12MB/s.*



---

*Bus PCI :*

- a. truyền tải dữ liệu tối đa 528 MB/s.*
- b. truyền tải dữ liệu tối đa 128MB/s.*
- c. truyền tải dữ liệu tối đa 512MB/s.*
- d. truyền tải dữ liệu tối đa 64MB/s.*



---

*Dẫn đầu về Chipset hiện có trên thị trường là :*

*a.AMD*

*b.ALI*

*c.Intel*

*d.Mac*



---

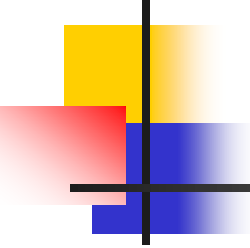
*Hệ thống Bus là hệ thống xa lộ thông tin bên trong PC giúp trao đổi:*

*a. thông tin giữa các máy tính*

*b. dữ liệu giữa các thiết bị ngoại vi*

*c. dữ liệu giữa bộ VXL và các thiết bị khác*

*d. tất cả đều đúng*



---

*Mọi hoạt động của máy tính từ CPU đến bộ nhớ RAM và những thiết bị I/O đều phải thông qua sự nối kết được gọi chung là :*

*a. Chuẩn giao tiếp*

*b. Bus*

*c. BIOS*

*d. CMOS*



# BÀI TẬP

---

*Bài 1 : Cho biết giá trị chuỗi 'XY' được lưu trữ trong MT dưới dạng số hex và dạng số bin?*

*Bài 2 : Cho biết giá trị ở hệ 10 của các số nguyên có dấu sau :*

*a.10000000<sub>b</sub>                      b.01111111<sub>b</sub>*

*Bài 3 : Cho đoạn code sau :*

*MOV AH,7F                                      INT 20H*

*MOV AX,1234                      Hãy cho biết giá trị của*

*MOV BH,AL                      các thanh ghi AX,BX ?*

*MOV BL,AH*



# BÀI TẬP

---

*Bài 4: Cho đoạn code sau :*

*MOV AL,81*

*ADD AL, 0FE*

*INT 20H*

*Giả sử các số đều là số có dấu. Giải thích kết quả chứa trong thanh ghi AL khi đoạn code trên được thực thi. Sử dụng giá trị ở hệ 10 để giải thích.*





# BÀI TẬP

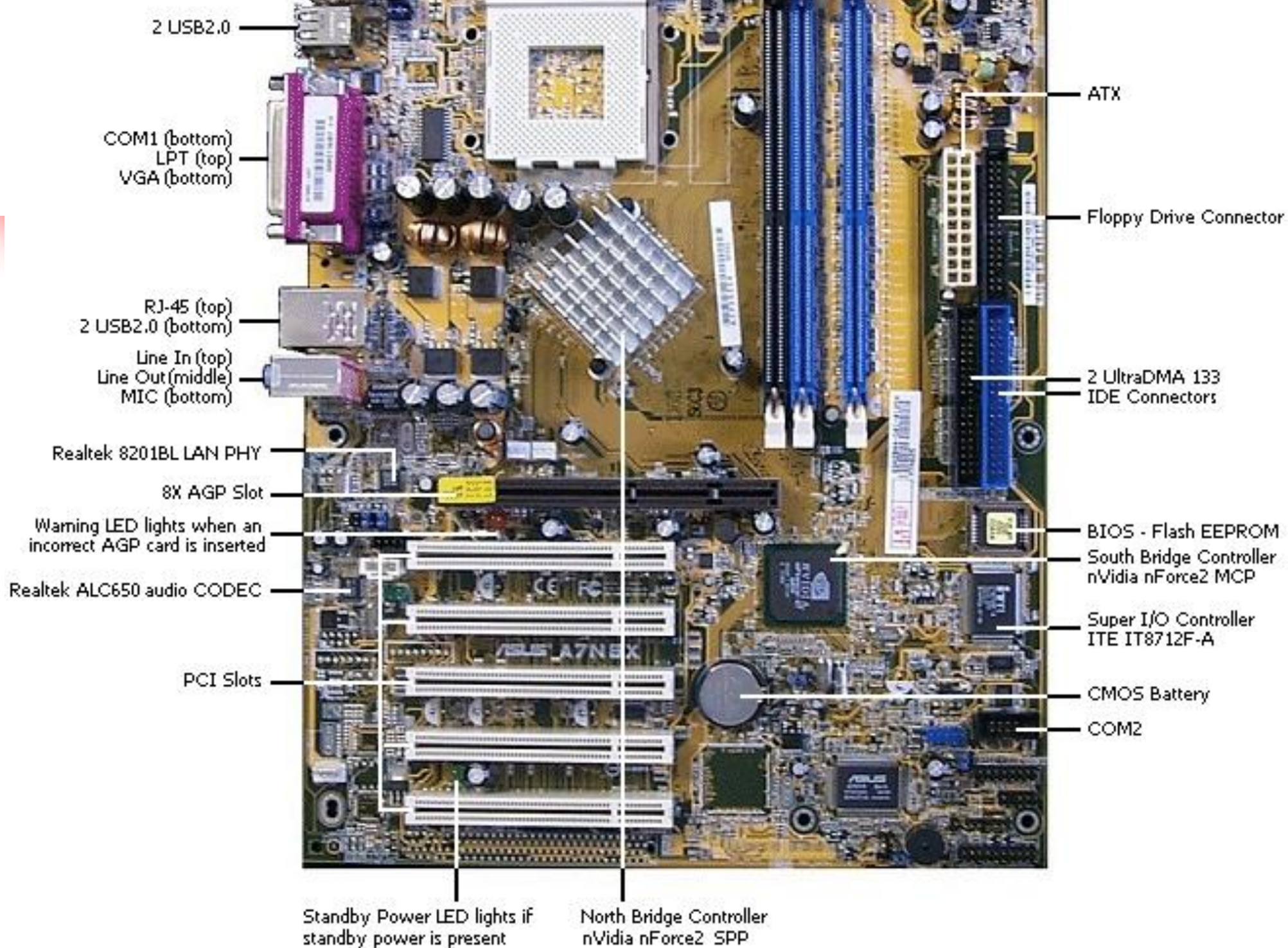
---

*Bài 5: Giả sử thanh ghi trong MT của bạn dài 24 bits, cho biết giá trị của số dương lớn nhất mà thanh ghi này có thể chứa ở 2 hệ 2 và hệ 16?*

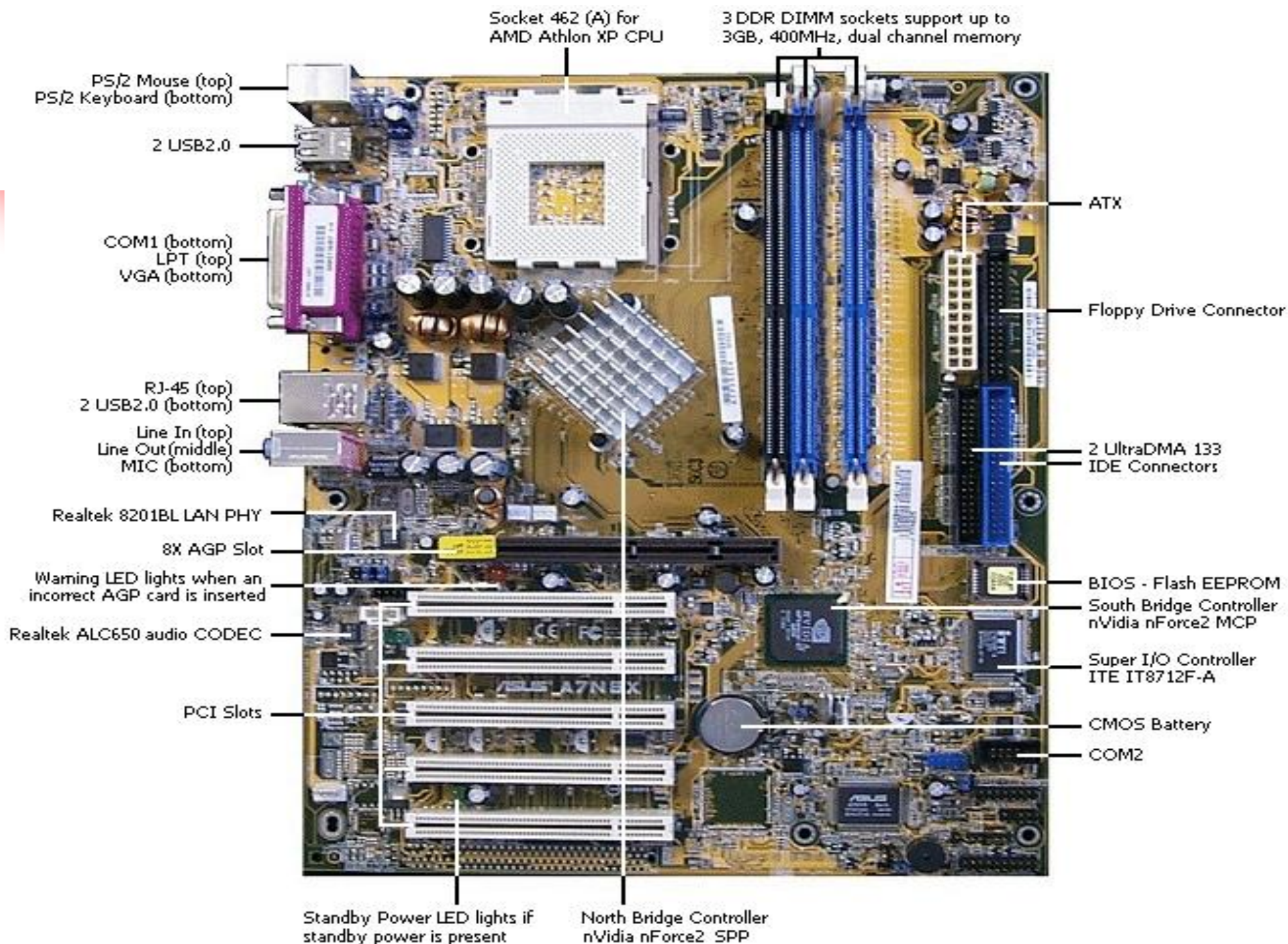
*Bài 6 : Biến đổi địa chỉ sau thành địa chỉ tuyệt đối*

*a. 0950:0100*

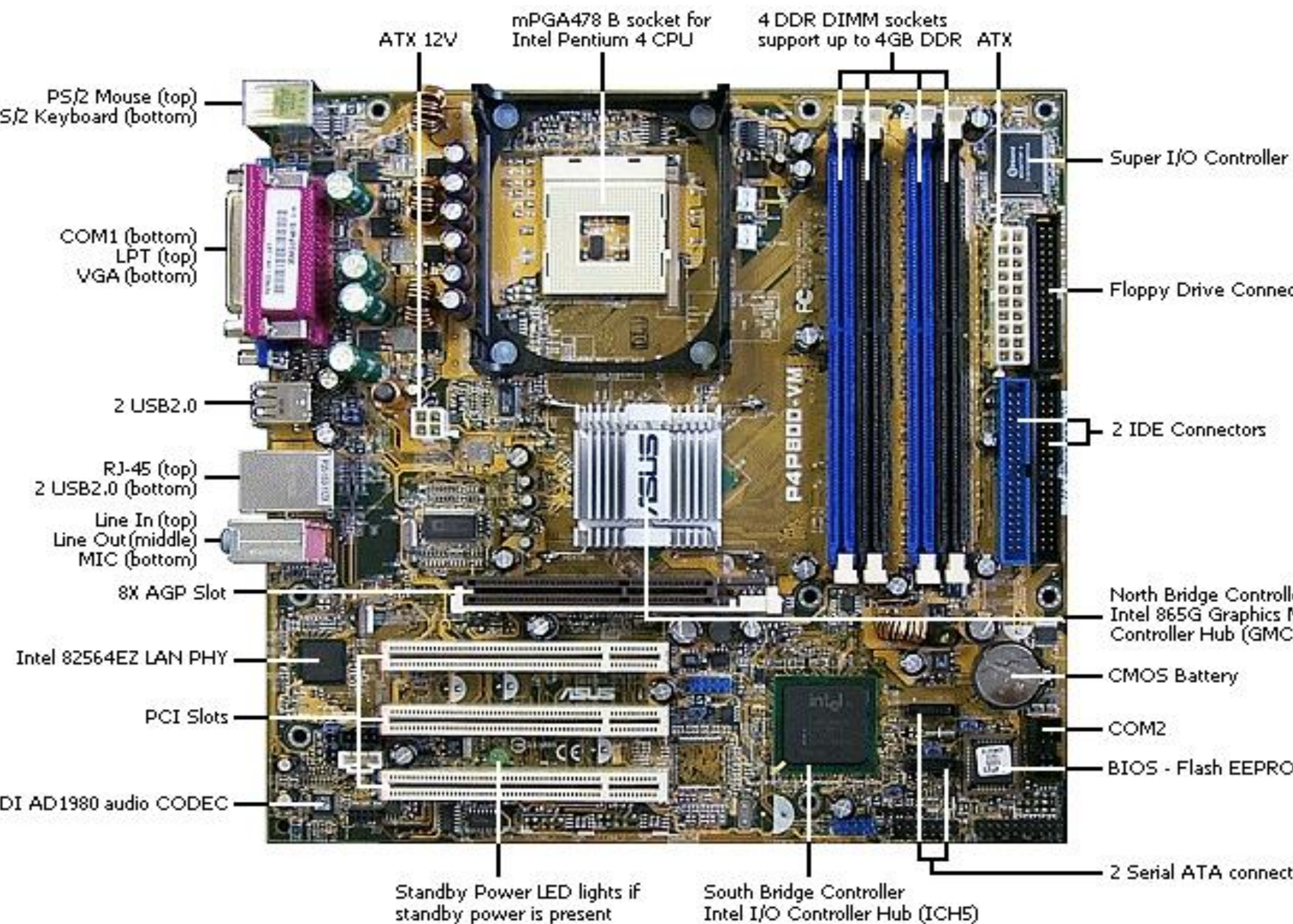
*b. 08F1:0200*







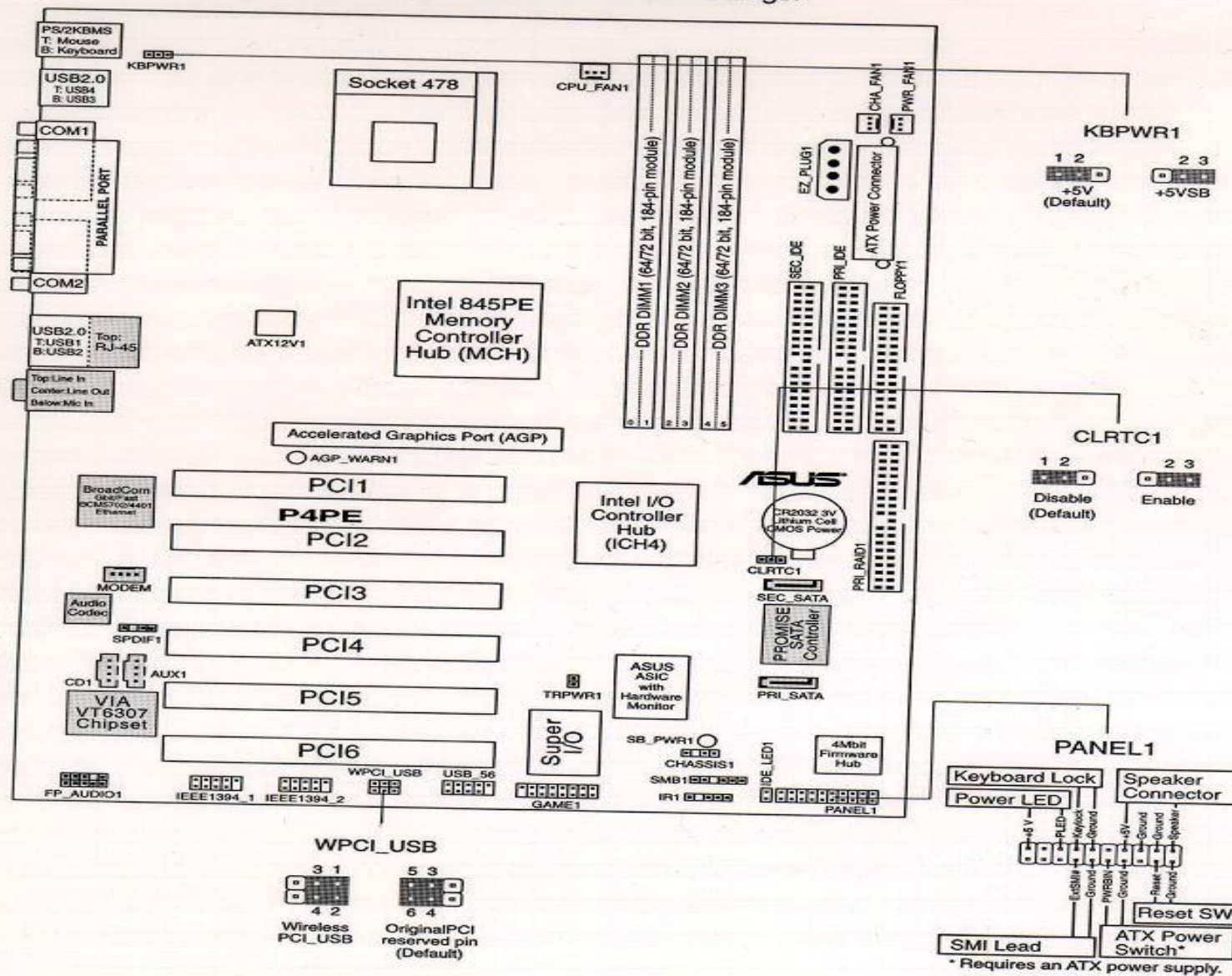


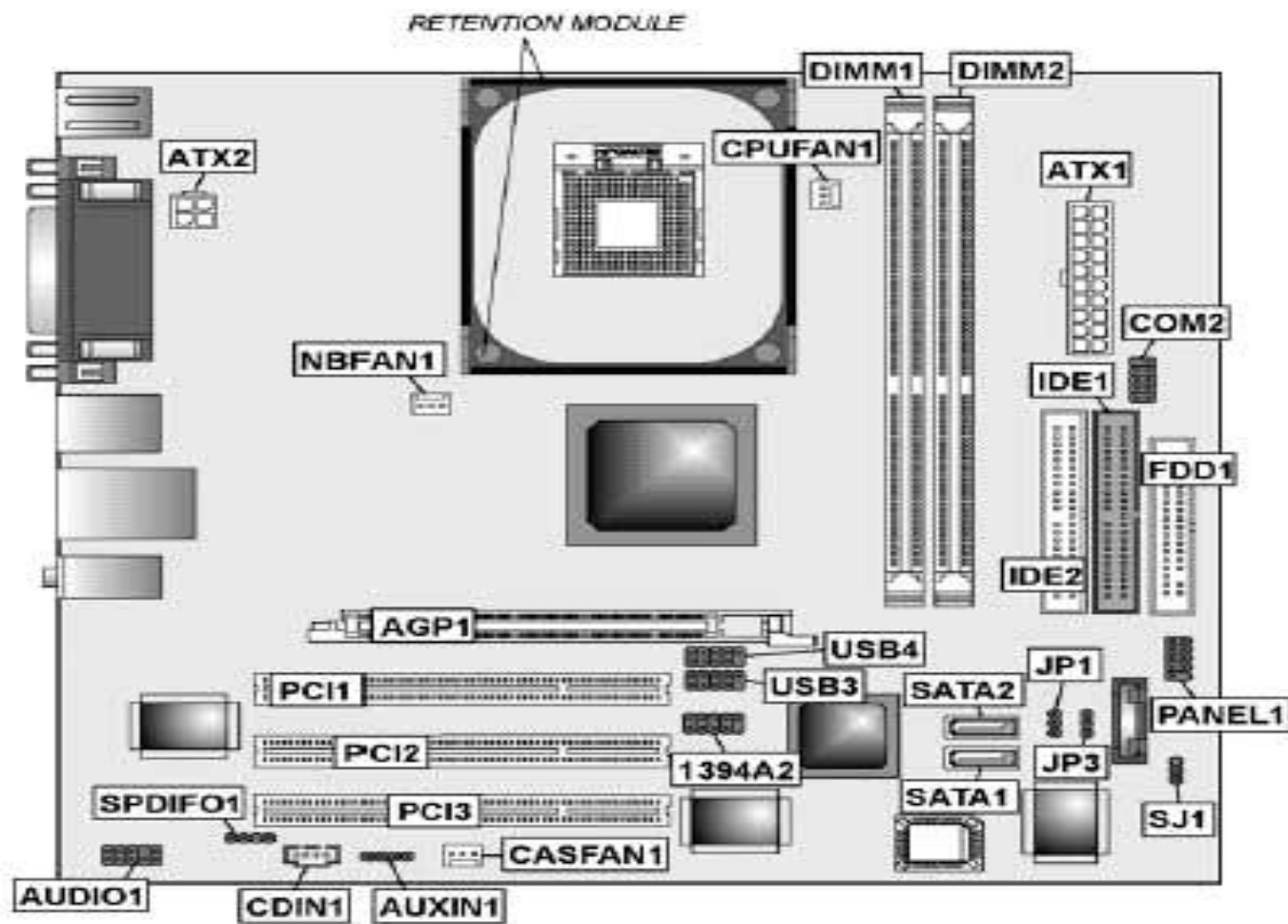




# P4PE Motherboard Settings

**IMPORTANT!** This diagram presents the general layout of your motherboard, and a summary of the switch and jumper settings. Before changing any setting, refer to the motherboard user guide for detailed descriptions and important notes on the settings.







# MAINBOARD

