Contents

[**I.** **LÝ THUYẾT** 3](#_Toc46568559)

[**Câu 1: Hãy trình bày sơ đồ khối, chức năng của các khối trong Kiến trúc máy tính Von Neumann?** 3](#_Toc46568560)

[**Câu 2: Hãy cho biết có mấy cách để phân loại máy tính, nêu các kiểu máy tính theo các cách đó?** 3](#_Toc46568561)

[**Câu 3: Hãy trình bày nguyên tắc chuyển đổi số ở hệ thập phân sang hệ nhị phân, bát phân và ngược lại? Cho ví dụ minh họa?** 4](#_Toc46568562)

[**Câu 4: Trình bày cấu trúc của bộ nhớ Cache và nguyên tắc đọc Cache.** 5](#_Toc46568563)

[**Câu 5: Hãy cho biết dựa vào tiêu chí nào người ta phân chia máy tính thành các thế hệ, đó là những thế hệ nào? Những máy tính ngày nay chúng ta đang sử dụng thuộc thế hệ nào?** 6](#_Toc46568564)

[**Câu 6: Các loại bus hệ thống trong máy tính? Chức năng của từng loại? Độ rộng của bus dữ liệu là 16 bit thì cho biết điều gì?** 7](#_Toc46568565)

[Câu 7: Hãy trình bày nguyên tắc chuyển đổi số ở hệ thập phân sang hệ nhị phân, thập lục phân và ngược lại? Cho ví dụ tương ứng? 7](#_Toc46568566)

[**Câu 8: Trình bày tổ chức của ổ đĩa cứng? Dung lượng của ổ đĩa cứng phụ thuộc vào yếu tố nào? Ứng dụng tính dung lượng của ổ đĩa cứng nếu biết Số byte trên sector là 512, số sector trên rãnh trung bình là 300, số rãnh trên một mặt là 20,000, số mặt trên một đĩa là 2, số đĩa trên ổ đĩa là 15?** 9](#_Toc46568567)

[**Câu 9: Hãy cho biết các phương pháp vào ra với máy vi tính?** 9](#_Toc46568568)

[**Câu 10: Các loại thanh ghi trong CPU? Chức năng của chúng?** 10](#_Toc46568569)

[**Câu 11: Trình bày nguyên tắc chuyển đổi số ở hệ nhị phân sang thập lục phân và ngược lại? Cho ví dụ tương ứng?** 11](#_Toc46568570)

[**Câu 12: Trình bày các phương pháp ánh xạ địa chỉ trong Cache?** 11](#_Toc46568571)

[**Câu 13: Khái niệm bộ nhớ trong của máy tính? Các loại bộ nhớ chính? Đặc trưng của bộ nhớ trong của máy tính? Cách giảm thiểu số chân của chíp nhớ?** 12](#_Toc46568572)

[**Câu 14: Trình bày cách quản lý trang trong bộ nhớ và phương pháp biên dịch địa chỉ tuyến tính sang địa chỉ vật lý?** 13](#_Toc46568573)

[**Câu 15: Hãy trình bày phương pháp biểu diễn số nguyên? Cho ví dụ minh họa?** 14](#_Toc46568574)

[**Câu 16: Trình bày mục đích của bộ nhớ ảo? Ánh xạ địa chỉ trong bộ nhớ ảo?** 16](#_Toc46568575)

[**Câu 17: Các loại bộ nhớ ROM của máy tính? Đặc trưng của từng loại?** 17](#_Toc46568576)

[**Câu 18: Vẽ sơ đồ thuật toán nhân nhanh Booth? Áp dụng tính 7 x 9?** 18](#_Toc46568577)

[**Câu 19: Hãy trình bày phương pháp biểu diễn số thực? Cho ví dụ?** 20](#_Toc46568578)

[**Câu 20: Trình bày khái quát thuật toán thay thế và chính sách ghi trong Cache?** 21](#_Toc46568579)

[**Câu 21: Các loại bộ nhớ trong của máy tính? Đặc trưng của từng loại? Ngày nay khi mua máy tính người ta thường quan tâm tới bộ nhớ trong nào và có những tham số gì đáng quan tâm nhất?** 22](#_Toc46568580)

[**Câu 22: Trình bày hiệu quả thực hiện lệnh sử dụng đường ống (Pipeline) trong kiến trúc RISC với đường ống 2 giai đoạn, 3 giai đoạn, 4 giai đoạn so với thực thi tuần tự?** 22](#_Toc46568581)

[**Câu 24: Trình bày các khối trong bộ vi xử lý của máy tính (CPU)? Chức năng, nhiệm vụ của từng khối?** 25](#_Toc46568582)

[**Câu 25: Các loại bộ nhớ RAM của máy tính? Đặc trưng của từng loại? Ngày nay máy tính thường sử dụng bộ nhớ RAM nào, các RAM của máy tính hay ghi DDR3 có ý nghĩa gì?** 25](#_Toc46568583)

[**Câu 26: Khái niệm Cache miss, thời gian truy cập cache, các nguyên nhân Cache miss? Cho ví dụ?** 26](#_Toc46568584)

[**Câu 27: Các chế độ địa chỉ của bộ vi xử lý? Nêu rõ ưu nhược điểm của từng chế độ?** 27](#_Toc46568585)

[**Câu 28: Hãy trình bày các kiểu hệ thống bộ vi xử lý song song và cách tổ chức song trong trong máy tính sử dụng bộ vi xử lý song song?** 28](#_Toc46568586)

[**Câu 29: Trình bày chi tiết các bước thực hiện lệnh trong máy tính?** 28](#_Toc46568587)

[**Câu 30: Các loại bus trong máy tính? Chức năng của từng loại? ( giống câu 6)** 29](#_Toc46568588)

[**Câu 31: Cấu trúc lệnh? Số địa chỉ trong lệnh? Cho ví dụ minh họa?** 30](#_Toc46568589)

[**Câu 32: Trình bày giải pháp đường ống lệnh (Pipeline) và Sơ đồ thời gian thực hiện lệnh bằng đường ống lệnh? Sơ đồ đường ống lệnh CPU 6 giai đoạn?** 30](#_Toc46568590)

[**II.** **BÀI TẬP** 32](#_Toc46568591)

# **LÝ THUYẾT**

# **Câu 1: Hãy trình bày sơ đồ khối, chức năng của các khối trong Kiến trúc máy tính Von Neumann?**

I/O

EQUIP-MENT

(I,O)

MAIN

MEMORY

(M)

ARITHMETIC-

LOGIC UNIT (CA)

PROGRAM CONTROL UNIT (CC)

Chức năng của các khối

* Main memory (bộ nhớ chính): Lưu trữ dữ liệu và các lệnh.
* Arithmetic Logic Unit (ALU): Khối đơn vị số học và logic: tính toán dữ liệu nhị phân.
* Control unit (CU): Dịch các lệnh trong bộ nhớ và thực thi chúng.
* I/O: Các thiết bị vào ra hoạt động dưới sự điều khiển của khối CU.

# **Câu 2: Hãy cho biết có mấy cách để phân loại máy tính, nêu các kiểu máy tính theo các cách đó?**

Có 3 cách phân loại máy tính:

**Phân loại theo khả năng**

* Máy tính lớn ( mainframe computer).
* Máy tính con ( mini computer).
* Máy vi tính (microcomputer): máy trạm làm việc (workstation), máy tính cá nhân PC (Personal computer).

**Phân loại theo nguyên lý:**

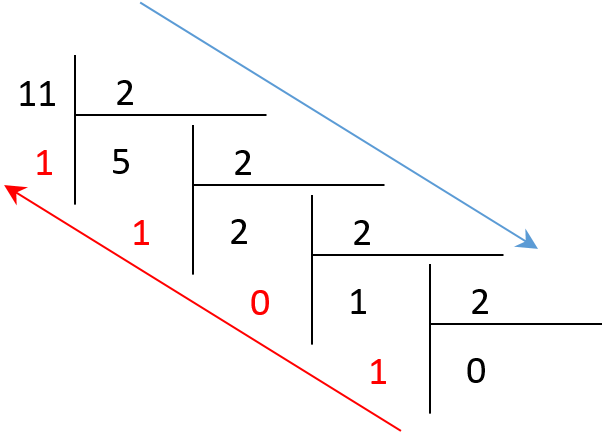
* Máy tính cơ khí.
* Máy tính tương tự.
* Máy tính số.

**Phân loại theo kiến trúc:**

* SISD( Single Instruction Stream-Single Data Stream).
* SIMD(Single Instruction Stream-Multiple Data Stream).
* MIMD( Multiple Instruction Stream-Multiple Data Stream).
* MISD(Multiple Instruction Stream-Single Data Stream).

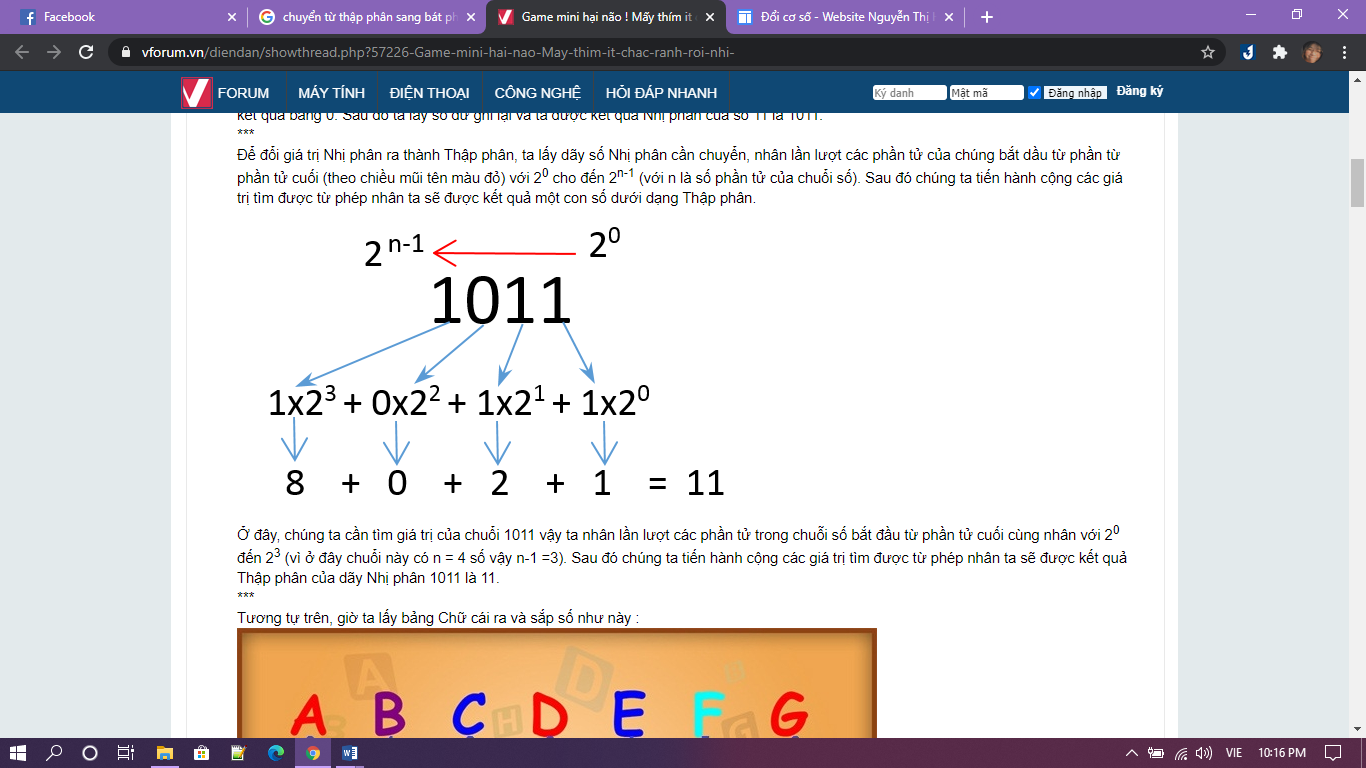
# **Câu 3: Hãy trình bày nguyên tắc chuyển đổi số ở hệ thập phân sang hệ nhị phân, bát phân và ngược lại? Cho ví dụ minh họa?**

-Đổi từ hệ thập phân sang hệ nhị phân: Nguyên tắc là cứ chia số đó cho 2 lấy phần dư rồi tiếp tục chia phần nguyên lấy phần dư cho đến khi số dư bằng 0, tiếp sau đó xếp thứ tự ngược từ dưới lên.



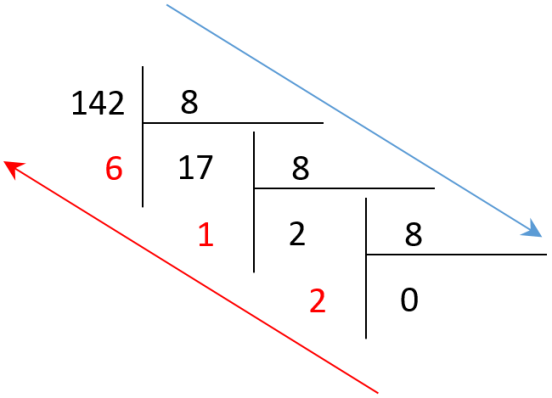
(11)10 = (1011)2

-Để đổi giá trị Nhị phân ra thành Thập phân, ta lấy dãy số Nhị phân cần chuyển, nhân lần lượt các phần tử của chúng bắt đầu từ phần từ phần tử cuối (theo chiều phải sang trái) với 20 cho đến 2n-1 (với n là số phần tử của chuỗi số). Sau đó chúng ta tiến hành cộng các giá trị tìm được từ phép nhân ta sẽ được kết quả một con số dưới dạng Thập phân.



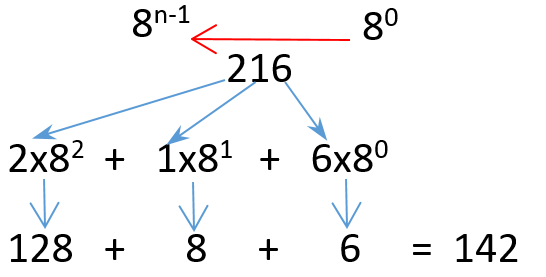
(1011)2= (11)10

-Đổi hệ Thập phân ra hệ Bát phân bằng cách chia con số Thập phân cần đổi với 8 rồi lấy kết quả chia với 8 liên tục cho đến khi kết quả bằng 0, sau đó ghi lại các số dư từ dưới lên để có được dãy Bát phân.



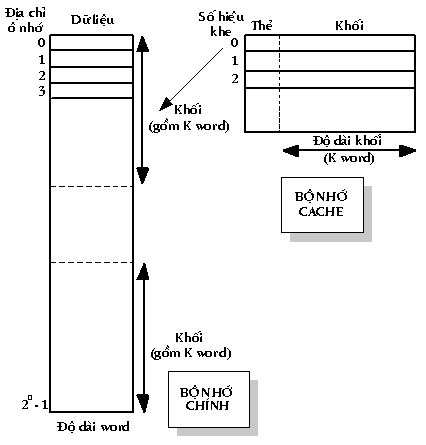
(142)10 = (216)8

Để đổi ngược lại Bát phân ra thập phân chúng ta nhân từ giá trị của chuỗi Bát phân với 80 đến 8n-1 theo chiều mũi tên phải qua trái.

(216)8= (142)10

# **Câu 4: Trình bày cấu trúc của bộ nhớ Cache và nguyên tắc đọc Cache.**

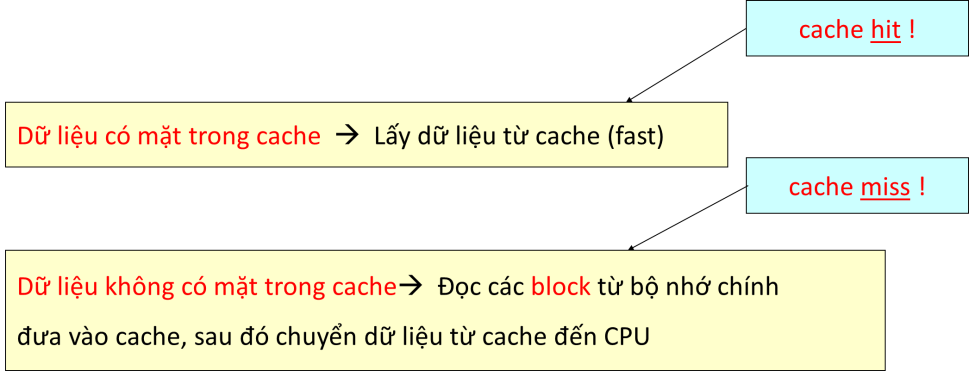
Cấu trúc bộ nhớ cache:



Bộ nhớ cache là nửa bên phải nhưng vẽ như thế này mới rõ được cấu trúc nó trong bộ nhớ. *(Nếu k đủ thời gian thì vẽ nửa bên phải thôi )*

Bộ nhớ cache bao gồm C khe của mỗi K word (từ nhớ) và số của các khe, hoặc các hàng, nó coi như ít hơn số khối nhớ của bộ nhớ chính (C <<M)

Hoạt động đọc của cache: khi CPU phát địa chỉ, từ nhớ sẽ được đọc. Nếu từ nhớ được chứa trong cache, nó sẽ được cung cấp cho CPU. Ngược lại, khối nhớ chứa từ nhớ đó sẽ được tải vào trong bộ nhớ và từ nhớ đó sẽ được cung cấp cho CPU.



# **Câu 5: Hãy cho biết dựa vào tiêu chí nào người ta phân chia máy tính thành các thế hệ, đó là những thế hệ nào? Những máy tính ngày nay chúng ta đang sử dụng thuộc thế hệ nào?**

-Để phân loại các thế hệ người ta vào 3 tiêu chí : về kỹ thuật , phần mềm và ứng dụng.

-Những thế hệ máy tính:

Thế hệ 1: (1946-1957) đèn điện tử

Thế hệ 2: (1958-1964) linh kiện bán dẫn chủ yếu là transistor

Thế hệ 3 (1965-1971) mạch tích hợp (IC)

Thế hệ 4 (1972-về sau) IC có mật độ tích hợp cao

Thế hệ 5: Đang phát triển. Dự kiến là thế hệ các máy tính xử lý song song.

(Nếu không có thời gian thì không cần trình bày chi tiết các thế hệ nêu được đặc trưng cơ bản nhất là phần cấu tạo đã đánh dấu màu đỏ thôi)

**Máy tính dùng đèn điện tử: thế hệ 1(1946-1957)**

+ **Về kỹ thuật: linh kiện dùng đèn điện tử**, độ tin cậy thấp, tổn hao năng lượng. Tốc độ tính toán từ vài nghìn đến vài trăm nghìn phép tính / giây.

+ Về phần mềm: chủ yếu dùng ngôn ngữ máy để lập trình.

+ Về ứng dụng: mục đích nghiên cứu khoa học kĩ thuật.

**Máy tính dùng transistor: thế hệ 2(1958-1964)**

+ **Về kỹ thuật: linh kiện bán dẫn chủ yếu là transistor**. Bộ nhớ có dung lượng khá lớn.

+ Về phần mềm: đã bắt đầu sd 1 số ngôn ngữ lập trình bậc cao: fortran, algol. Cobol...

+ Về ứng dụng: tham gia giải các bài toán kinh tế xã hội….

**Máy tính dùng vi mạch: thế hệ 3(1965-1971)**

+ **Về kỹ thuật: linh kiện chủ yếu sử dụng các mạch tích hợp (IC),** các thiết bị ngoại vi được cải tiến, đĩa từ được sd rộng rãi. Tốc độ tính toán đạt vài triệu phép toán trên giây, dung lượng bộ nhớ đạt vài MB.

+ Về phần mềm: xuất hiện nhiều hệ điều hành khác nhau. Xử lí song song. Phần mềm đa dạng chất lượng cao, cho phép khai thác máy tính theo nhiều chế độ khác nhau.

+ Về ứng dụng: tham gia trong nhiều lĩnh vực của xã hội.

**Máy tính dùng vi mạch mật độ cao: thế hệ thứ 4 ( 1972-về sau):**

**+ Về kỹ thuật: Sử dụng mạch tích hợp cỡ lớn VLSI, thiết kế các cấu trúc đa xử lí.** Tốc độ đạt tới hàng chục triệu phép toán trên giây.

+ Về ứng dụng: được áp dụng trong hầu hết các hoat động của xã hội

**Về thế hệ thứ 5: máy tính đa xử lí, đa máy tính, sử dụng kĩ thuật xử lý song song.**

-Những máy tính chúng ta đang sử dụng thuộc thế hệ thứ 4.

# **Câu 6: Các loại bus hệ thống trong máy tính? Chức năng của từng loại? Độ rộng của bus dữ liệu là 16 bit thì cho biết điều gì?**

Có 3 loại bus hệ thống trong máy tính đó là : bus dữ liệu, bus địa chỉ, bus điều khiển. (Trả lời luôn ý này sau đó liệt kê chức năng từng loại, dễ nhớ là bus có tên gọi như nào thì chức năng tương ứng để xác định vậy)

-**Bus dữ liệu:** cung cấp đường dẫn để truyền dữ liệu giữa các module hệ thống. Bus dữ liệu có thể là 32,64,128 đường, số lượng các đường này gọi là độ rộng của bus dữ liệu. Vì mỗi đường chỉ có thể truyền tải một bit tín hiệu tại 1 thời điểm, số lượng các đường cho biết bao nhiêu bit có thể truyền cùng 1 lúc. Độ rộng của bus dữ liệu là nhân tố quan trọng trong việc xác định hiệu suất hoạt động của hệ thống.

-**Bus địa chỉ:** được sử dụng để xác định nguồn hoặc đích của dữ liệu trên bus dữ liệu. Độ rộng của bus địa chỉ chỉ ra khả năng xác định bộ nhớ lớn nhất của hệ thống.

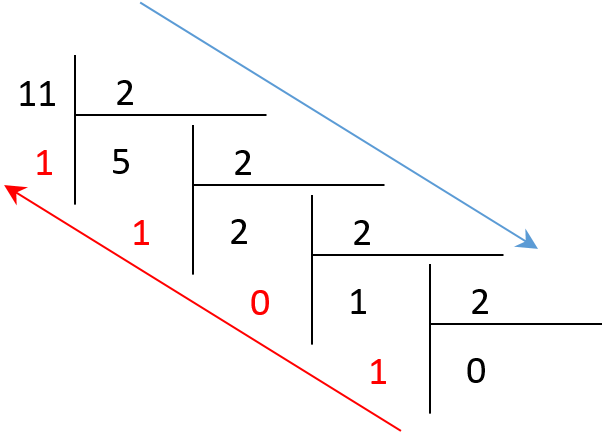
-**Bus điều khiển:** được sử dụng để điều khiển truy cập đến và sử dụng các đường dữ liệu và địa chỉ. Các tín hiệu điều khiển truyền cả lệnh và thông tin thời gian giữa các module hệ thống. Các tín hiệu thời gian xác định tính hợp lệ của dữ liệu và thông tin địa chỉ. Các tín hiệu lệnh xác định các phép toán được thực hiện.

Ngoài ra còn có phân loại bus theo đường truyền: bus đồng bộ và bus không đồng bộ.

Độ rộng của bits dữ liệu là 16 bít cho biết hệ thống bus có 16 đường để truyền 16bits dữ liệu cùng một lúc.

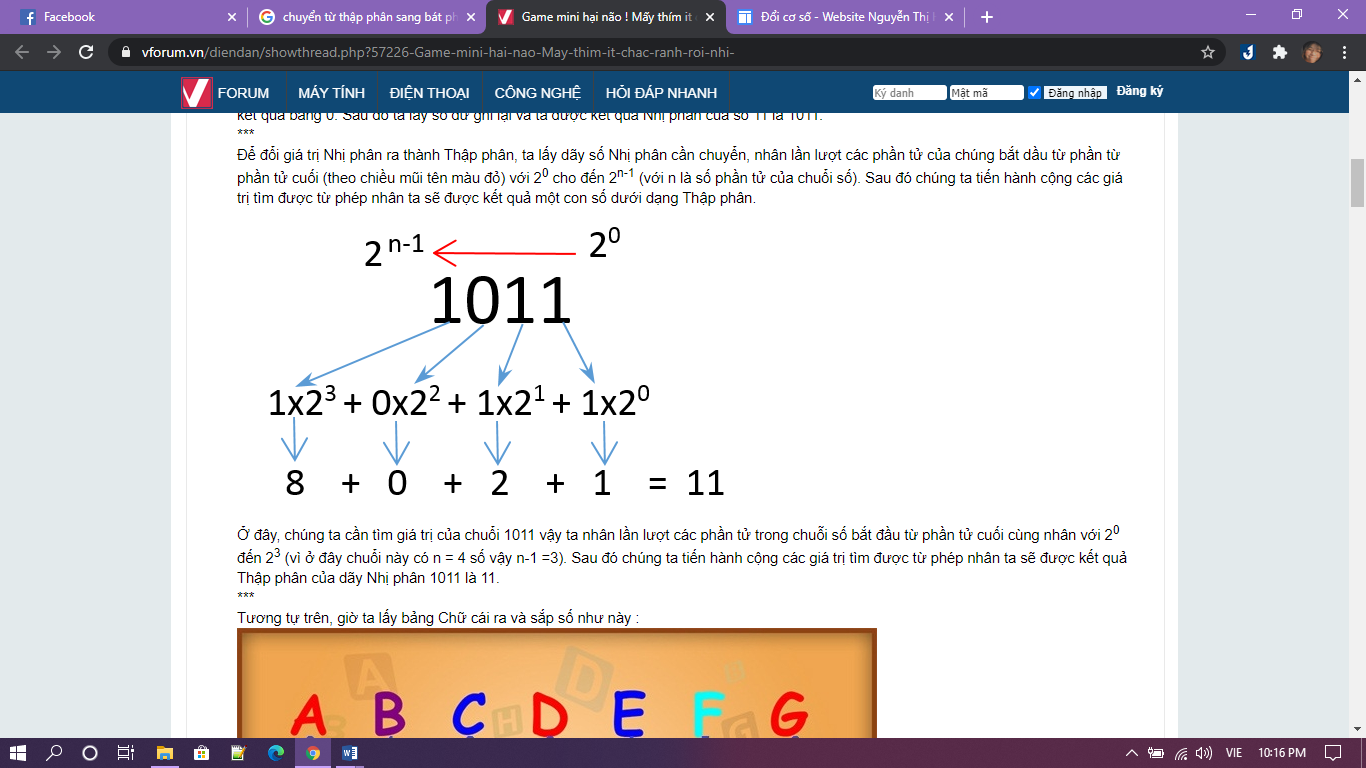
# Câu 7: Hãy trình bày nguyên tắc chuyển đổi số ở hệ thập phân sang hệ nhị phân, thập lục phân và ngược lại? Cho ví dụ tương ứng?

-Đổi từ hệ thập phân sang hệ nhị phân, nguyên tắc là cứ chia số đó cho 2 lấy phần dư rồi tiếp tục chia phần nguyên lấy phần dư cho đến khi thương bằng 0, tiếp sau đó xếp thứ tự ngược từ dưới lên.



(11)10 = (1011)2

-Để đổi giá trị Nhị phân ra thành Thập phân, ta lấy dãy số Nhị phân cần chuyển, nhân lần lượt các phần tử của chúng bắt dầu từ phần từ phần tử cuối (theo chiều mũi tên phải sang trái) với 20 cho đến 2n-1 (với n là số phần tử của chuổi số). Sau đó chúng ta tiến hành cộng các giá trị tìm được từ phép nhân ta sẽ được kết quả một con số dưới dạng Thập phân.



(1011)2= (11)10

-Đổi từ hệ thập phân sang hệ thập lục phân :

+Bước 1: Nhóm 4 bits từ phía ngoài cùng bên phải của số nhị phân.

+Bước 2: Chuyển đổi mỗi nhóm trên sang số thập lục tương ứng.

Chuyển số nhị phân: 110101101010111001101010(2) = ?(16)

Bước 1: Nhóm lần lượt 4 bit từ phải sáng trái ta được:

1101 0110 1010 1110 0110 1010

Bước 2: chuyển mỗi nhóm 4 bits sang số thập lục phân ta được kết quả:

1101 0110 1010 1110 0110 1010

   D    6    A    E    6    A

Vậy ta được 110101101010111001101010(2) = D6AE6A(16)

***(Nếu thời gian ngắn thì ví dụ ghi tách ra 4 bit một và ghi kết quả dòng cuối luôn, hoặc lấy 8 bit thôi cho ngắn)***

# **Câu 8: Trình bày tổ chức của ổ đĩa cứng? Dung lượng của ổ đĩa cứng phụ thuộc vào yếu tố nào? Ứng dụng tính dung lượng của ổ đĩa cứng nếu biết Số byte trên sector là 512, số sector trên rãnh trung bình là 300, số rãnh trên một mặt là 20,000, số mặt trên một đĩa là 2, số đĩa trên ổ đĩa là 15?**

*Tổ chức của ổ đĩa cứng:*

* Ổ đĩa cứng: chứa nhiều lớp đĩa. Quay quanh truc khoảng 3.600-15000 vòng 1 phút.
* Các lớp đĩa này đc làm từ kim loại và đc phủ 1 chất từ tính, đường kính của đĩa thay đổi từ 1,3 – 8 inch. Mỗi mặt của đĩa chia làm nhiều đường tròn đồng trục gọi là rãnh.
* Thông thường mỗi mặt của 1 lớp đĩa chứa khoảng 10000-30000 rãnh.
* Mỗi rãnh chia làm nhiều cung (sector ) dùng để chứa thông tin. Mỗi rãnh có thế chứa từ 64-800 cung.
* Cung là đơn vị nhỏ nhất mà máy tính có thể đọc hoặc viết. Chuỗi thông tin ghi trên mỗi cung gồm có: số thứ tự của cung, một khoảng trống, số liệu của cung đó bao gồm cả các mã sửa lỗi, một khoảng trống, số thứ tự của cung tiếp theo.

Dung lượng ổ đĩa cứng được tính bằng: (số byte/sector) × (số sector/track) × (số cylinder) × (số đầu đọc/ghi) = 512 x 300 x 20000 x 2 x 15 =9,216.1010 byte = 85,83GB

*+Dung lượng của ổ đĩa phụ thuộc ?: Phụ thuộc vào những cái tính trong tích trên, liệt kê ra: Số byte trên sector, số sector trên rãnh, số rãnh trên một mặt, số mặt trên một đĩa, số đĩa trên ổ đĩa.*

# **Câu 9: Hãy cho biết các phương pháp vào ra với máy vi tính?**

*Vào ra riêng biệt*

* Cổng vào-ra được đánh địa chỉ theo không gian địa chỉ vào-ra
* CPU trao đổi dữ liệu với cổng vào-ra thông qua các lệnh vào-ra chuyên dụng (IN, OUT)
* Chỉ có thể thực hiện trên các hệ thống có quản lý không gian địa chỉ vào-ra riêng biệt

*Vào ra theo ánh xạ bộ nhớ*

* Cổng vào-ra được đánh địa chỉ theo không gian địa chỉ bộ nhớ.
* Vào-ra giống như đọc/ghi bộ nhớ.
* CPU trao đổi dữ liệu với cổng vào-ra thông qua các lệnh truy nhập dữ liệu bộ nhớ.
* Có thể thực hiện trên mọi hệ thống.

**Các phương pháp ĐIỀU KHIỂN VÀO - RA**

*Vào ra theo chương trình*

Nguyên tắc chung: CPU điều khiển trực tiếp vào-ra bằng chương trình → cần phải lập trình vào-ra.

*Vào ra bằng cách thăm dò ( hỏi vòng):* CPU luôn phải kiểm tra tính sẵn sàng của thiết bị ngoại vi thông qua tín hiệu bắt tay (handshake signal). Các tín hiệu này lấy từ mạch phối ghép do người thiết kế tạo ra để chương trình thăm dò hoạt động trên đó

*Vào ra bằng điều khiển ngắt*

* CPU không phải đợi trạng thái sẵn sàng của mô-đun vào-ra, CPU thực hiện một chương trình nào đó
* Khi mô-đun vào-ra sẵn sàng thì nó phát tín hiệu ngắt CPU
* CPU tạm dừng công việc hiện tại để phục vụ trao đổi dữ liệu (nhảy tới hàm phục vụ ngắt)
* Sau khi xong CPU trở lại tiếp tục thực hiện chương trình đang thực hiện.

*Vào ra bằng DMA (Truy cập bộ nhớ trực tiếp)*

* Thêm mô-đun phần cứng trên bus → DMAC (Controller)
* DMAC điều khiển trao đổi dữ liệu giữa môđun vào-ra với bộ nhớ chính
* Nhanh hơn các phương pháp nêu trên.

# **Câu 10: Các loại thanh ghi trong CPU? Chức năng của chúng?**

**Trả lời:**

- Có 2 loại thanh ghi trong CPU: User – Visible Register và Control and Status Registers

**+ User–Visible Register** (Thanh ghi người dùng lập trình được): Nó cho phép người lập trình ngôn ngữ máy hoặc ngôn ngữ Assembly thu nhỏ bộ nhớ chính bằng cách tối ưu hóa việc sử dụng các thanh ghi. Một số loại:

* Thanh ghi mục đích chung
* Thanh ghi Dữ liệu
* Thanh ghi địa chỉ

+ **Control and Status Registers** (Thanh ghi điều khiển và trạng thái): Các thanh ghi này được sử dụng bởi đơn vị điều khiển CU để điều khiển các hoạt động của CPU và bằng phân quyền, các chương trình điều khiển hệ thống điểu khiển sự thực thi của các chương trình khác. Có 4 loại chủ chốt:

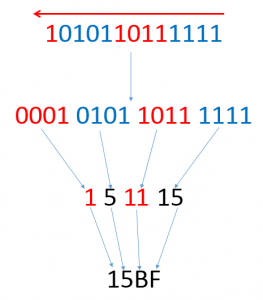
* Thanh ghi bộ đếm chương trình (Program Counter-PC): chứa địa chỉ của một chỉ lệnh được tìm nạp.
* Thanh ghi lệnh (IR - Instruction Register): chứa chỉ lệnh được tìm nạp gần nhất.
* Thanh ghi địa chỉ bộ nhớ (MAR - Memory Address Register): chứa địa chỉ của các vị trí trong bộ nhớ.
* Thanh ghi bộ nhớ đệm (MFR - Memory Fuffer Register): chứ một từ dữ liệu được ghi vào trong bộ nhớ hoặc từ được đọc gần đây nhất.

# **Câu 11: Trình bày nguyên tắc chuyển đổi số ở hệ nhị phân sang thập lục phân và ngược lại? Cho ví dụ tương ứng?**

**Trả lời:**

**Đổi nhị phân sang thập lục phân:** Để chuyển đổi một dãy nhị phân thành số thập lục phân, ta nhóm lần lượt các bị nhị phân từ phải qua trái thành các nhóm 4 bit, sau đó dùng phương pháp [chuyển nhị phân sang thập phận](https://www.engisv.info/?p=220) để chuyển các nhóm 4bit này thành số thập phân, các số có kết quả từ 10 đến 15 sẽ chuyễn thành A, B, C, D, E, F và cuối cùng biểu diễn các giá trị tìm được thành dãy số thập lục phân.

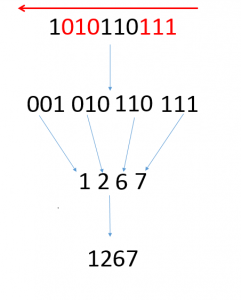
Ví dụ 1010110111111 đổi thành thập lục phân:



(1010110111111)2 = (15BF)16

**Đổi nhị phân thành bát phân:** Để chuyển đổi một dãy nhị phân thành số bát phân, ta nhóm lần lượt các bị nhị phân từ phải qua trái thành các nhóm 3 bit, sau đó dùng phương pháp [chuyển nhị phân sang thập phận](https://www.engisv.info/?p=220) để chuyển các bit này thành số thập phân (lưu ý là với hệ bát phân các số biểu diễn không lớn hơn 7) và cuối cùng biểu diễn các giá trị tìm được thành dãy số bát phân.

Ví dụ 1010110111 chuyển thành bát phân:



(1010110111)2 = (1267)8

# **Câu 12: Trình bày các phương pháp ánh xạ địa chỉ trong Cache?**

*(Câu này căn thời gian để làm chứ trình bày từng nào cũng không đủ)*

Có 3 kĩ thuật ánh xạ: ánh xạ trực tiếp, ánh xạ liên kết hoàn toàn và ánh xạ liên kết tập hợp.

- **Ánh xạ trực tiếp:** Mỗi block của bộ nhớ chính chỉ có thể được nạp vào 1 line duy nhất của cache.

Quy ước nạp:

* + block 0->line 0
  + block 1->line 1
  + block C->line 0
  + block i-> line (I mod C).

- **Ánh xạ liên kết hoàn toàn (toàn phần):**

* Mỗi block có thể được nạp vào bất kỳ line nào của cache.
* Địa chỉ bộ nhớ do CPU phát ra được chia thành 2 phần: tag và byte.
* Để kiểm tra xem một block có trong cache hay không, phải đồng thời kiểm tra tất cả tag của các line trong cache.
* Cần các mạch phức tạp để kiểm tra

- **Ánh xạ liên kết tập hợp:**

* Là phương pháp dung hòa của 2 phương pháp trên
* Chia cache thành các tập: S0, S1, S2 ...
* Mỗi Set có một số Line (2, 4, 8, 16 Line)
* Ví dụ mỗi Set có 2 line: 2-way Set Associative Mapping
* Mỗi block được nạp vào 1 line nào đó trong Set nhất định
* Địa chỉ do CPU phát ra có 3 trường: Tag, Set, Byte

# **Câu 13: Khái niệm bộ nhớ trong của máy tính? Các loại bộ nhớ chính? Đặc trưng của bộ nhớ trong của máy tính? Cách giảm thiểu số chân của chíp nhớ?**

**- Khái niệm:** bộ nhớ trong của máy tính là bộ nhớ cung cấp nơi để lưu trữ chính trong máy tính.

**- Phân loại:** Có 2 loại bộ nhớ chính là RAM (Random Access Memory) và ROM (Read Only Memory):

+RAM là bộ nhớ bán dẫn truy cập ngẫu nhiên.

* + Đặc điểm: có thể ghi hoặc đọc dữ liệu, khi máy tính tắt thông tin trong RAM sẽ bị mất. Như vậy Ram dùng để lưu trữ thông tin tạm thời.
  + RAM dc chia thành RAM tĩnh (SRAM) & RAM động (DRAM).

+ ROM là bộ nhớ bán dẫn chỉ dùng để đọc.

* + Đặc điểm: Chương trình và dữ liệu dc cất trong ROM từ lúc đầu, nên thông tin được lưu trữ trong bộ nhớ ROM không bị mất khi tắt nguồn điện nuôi cho mạch. Thường dùng để lưu trữ thông tin cố định đối với hệ thống (như: vi chương trình, các chương trình con thư viện cho các hàm thường xuyên được sử dụng, các chương trình hệ thống – *trong ngoặc này không nhớ thì thôi*).
  + ROM được chia thành: ROM không lập trình được (mask ROM) & ROM lập trình được (PROM –EPROM - EEPROM).

**- Cách giảm thiểu số chân chip nhớ:** Người ta thường “dồn kênh” (multiplex), tức là ghép chức năng, giữa bus địa chỉ và bus dữ liệu để giảm thiểu số chân cần thiết.

# **Câu 14: Trình bày cách quản lý trang trong bộ nhớ và phương pháp biên dịch địa chỉ tuyến tính sang địa chỉ vật lý?**

**Trả lời: Câu này chưa check. Sẽ update sau**

* Máy tính quản lý bộ nhớ phân trang dựa trên thư mục trang, bảng trang và trang. Chúng đều có cùng một kích thước là 4Kbyte.

+ Thư mục trang (Page Directory): Thư mục trang chứa tới 1024 lối vào thư mục trang PDE (Page Directory Entry). PDE có kích thước 32 bit (4 byte). Mỗi PDE chứa 20 bít địa chỉ nền bảng trang (trỏ đến nền bảng trang). Mỗi một thư mục trang có thể quản lý được tới 1024 bảng trang. Cấu trúc một PDE trong thư mục trang:

31 12 11 0

|  |  |
| --- | --- |
| Địa chỉ bảng trang (20bit) | Thông tin về bảng trang |

+ Bảng trang (Page Table): Bảng trang chứa 1024 lối vào bảng trang PTE (Page Table Entry). PTE có kích thước 32 bit (4 byte). Mỗi PTE chứa địa chỉ nền của một trang 4Kbyte, do vậy một bảng trang có thể quản lý được tới 1024 trang. Cấu trúc một PTE:

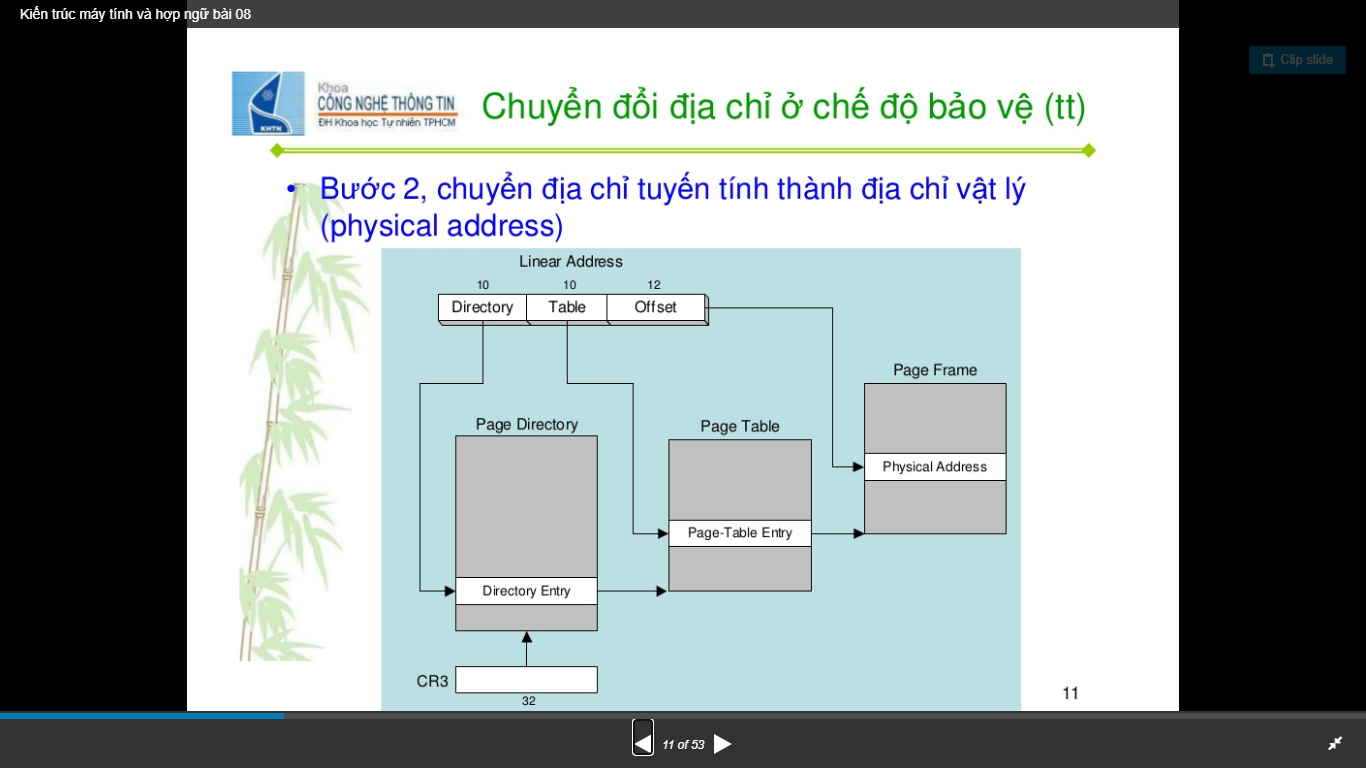
31 12 11 0

|  |  |
| --- | --- |
| Địa chỉ bảng trang(20 bit) | Thông tin về bảng trang |

+ Trang (page): không gian địa chỉ phẳng có kích thước 4 Kbyte hay 4Mbyte.

+ bảng con trỏ danh mục trang: tổ hợp 3 giá trị 64 bit, mỗi giá trị trỏ đến thư mục trang. Cấu trúc này chỉ được dùng khi mở rộng không gian địa chỉ lên 36 bit.

* Phương pháp biên dịch địa chỉ tuyến tính sang địa chỉ vật lý:



Để biên dịch địa chỉ tuyến tính sang địa chỉ vật lý, địa chỉ tuyến tính được chia thành 3 phần:

+ Chỉ số thư mục trang: bit 22 đến 31, là giá trị lệch của một vị trí (PDE) trong bảng danh mục (có tất cả PDE), PDE cho biết vị trí bảng trang cần truy nhập.

+ Giá trị bảng trang: bit 12 đến bit 21 là giá trị lệch của một vị trí (PTE) trong bảng trang PTE cho biết địa chỉ cơ sở của một trang trong bộ nhớ vật lý.

+ Địa chỉ lệch trong trang: bit 0 đến bit 11 cho biết vị trí của byte cần truy nhập trong trang được chọn bằng giá trị PTE.

Địa chỉ cơ sở của thư mục trang hiện tại được lưu trữ trong thanh ghi điều khiển CR3

Mức thấp nhất trong hai mức của PDE và PTE được lấy làm mức đặc quyền và quyền ghi/đọc của từng trang.

Địa chỉ vật lý của ô nhớ trong trang được tính bằng cách kết hợp 20 bit cao (phần địa chỉ trang) trong PTE với 12 bit thấp nhất trong địa chỉ tuyến tính.

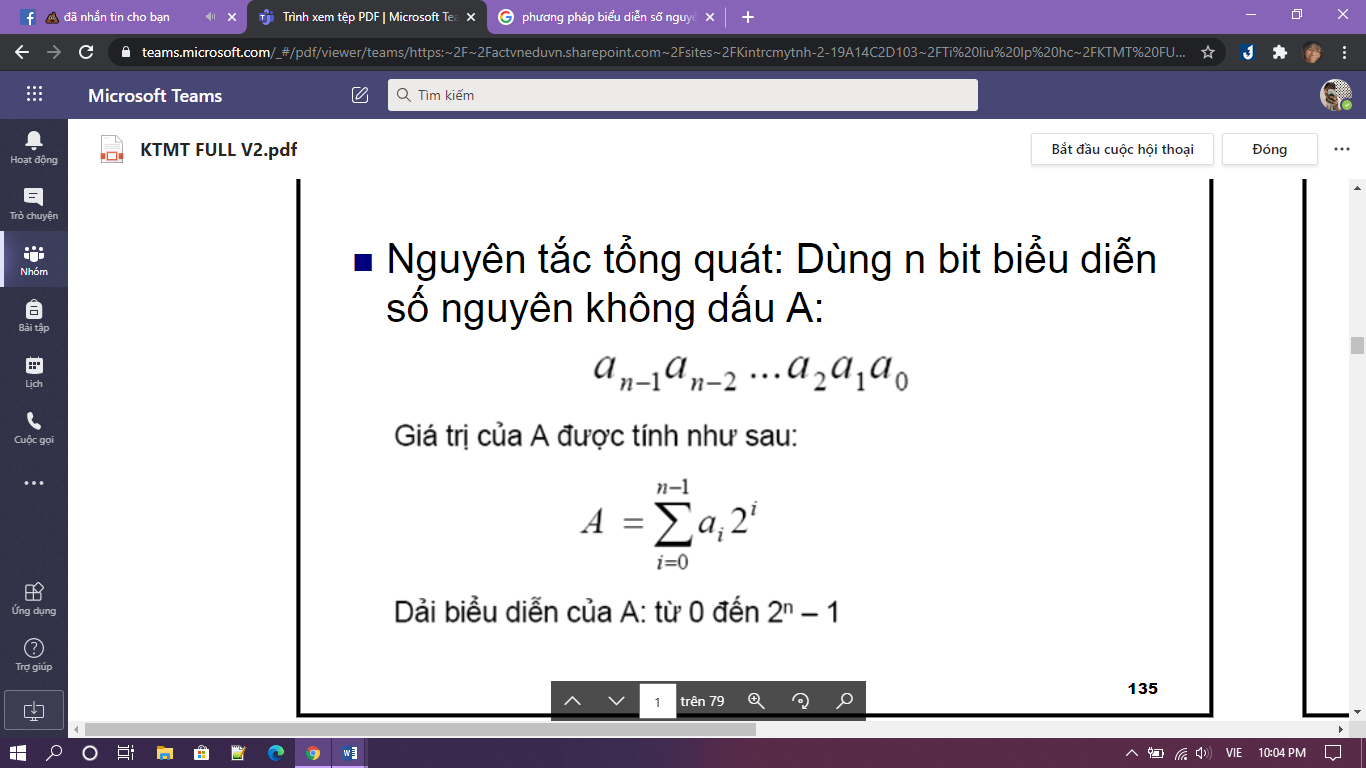
# **Câu 15: Hãy trình bày phương pháp biểu diễn số nguyên? Cho ví dụ minh họa?**

TL:

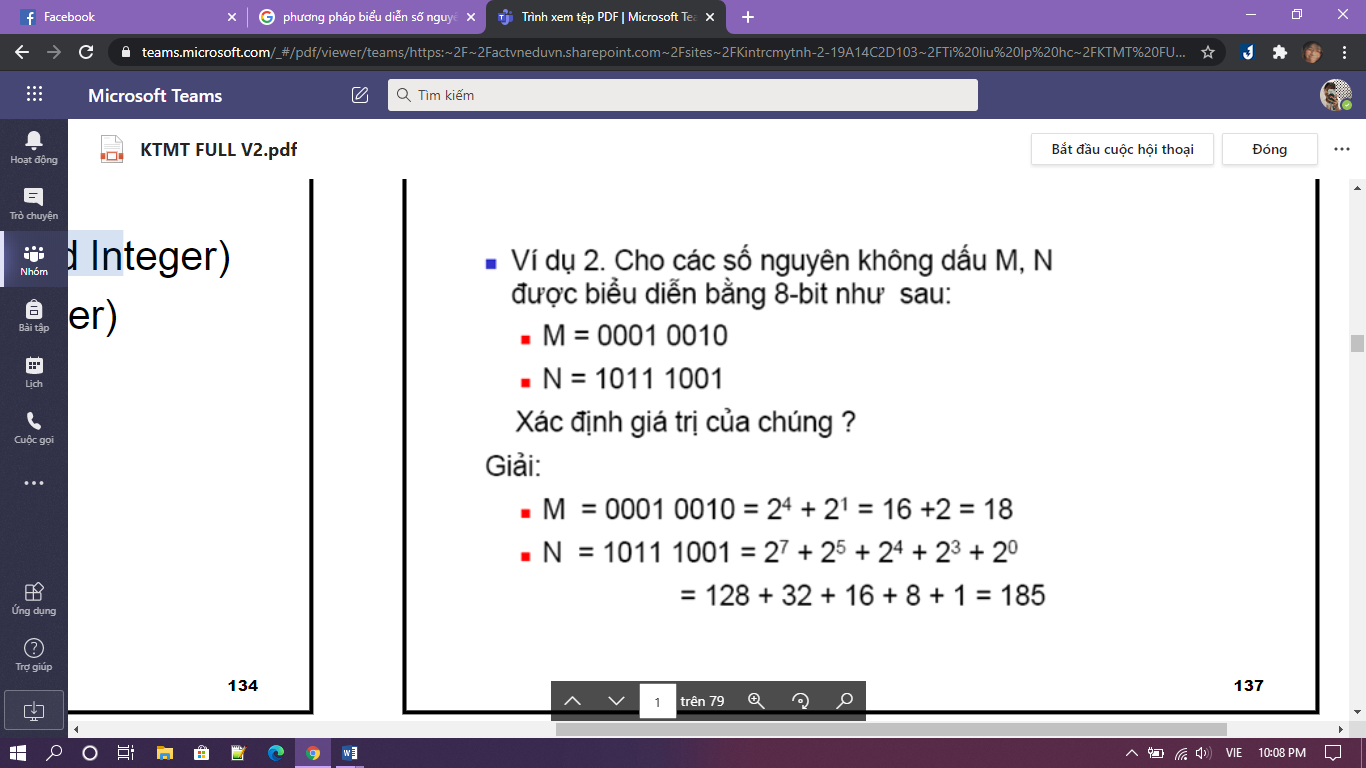
Biểu diên số nguyên chia thành biểu diễn số nguyên không dấu và số nguyên có dấu

- Biểu diễn số nguyên không dấu :

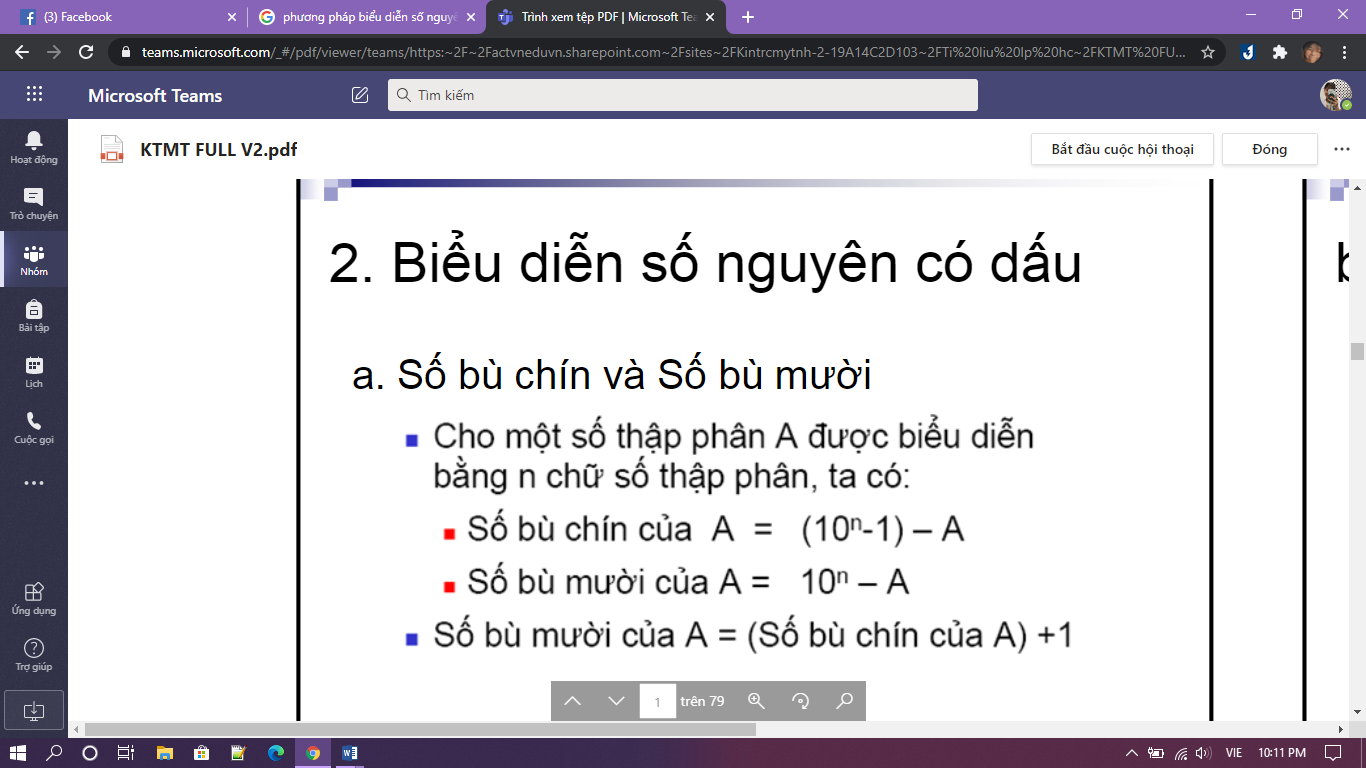
Nguyên tắc tổng quát: Dùng n bit biểu diễn số nguyên không dấu A:



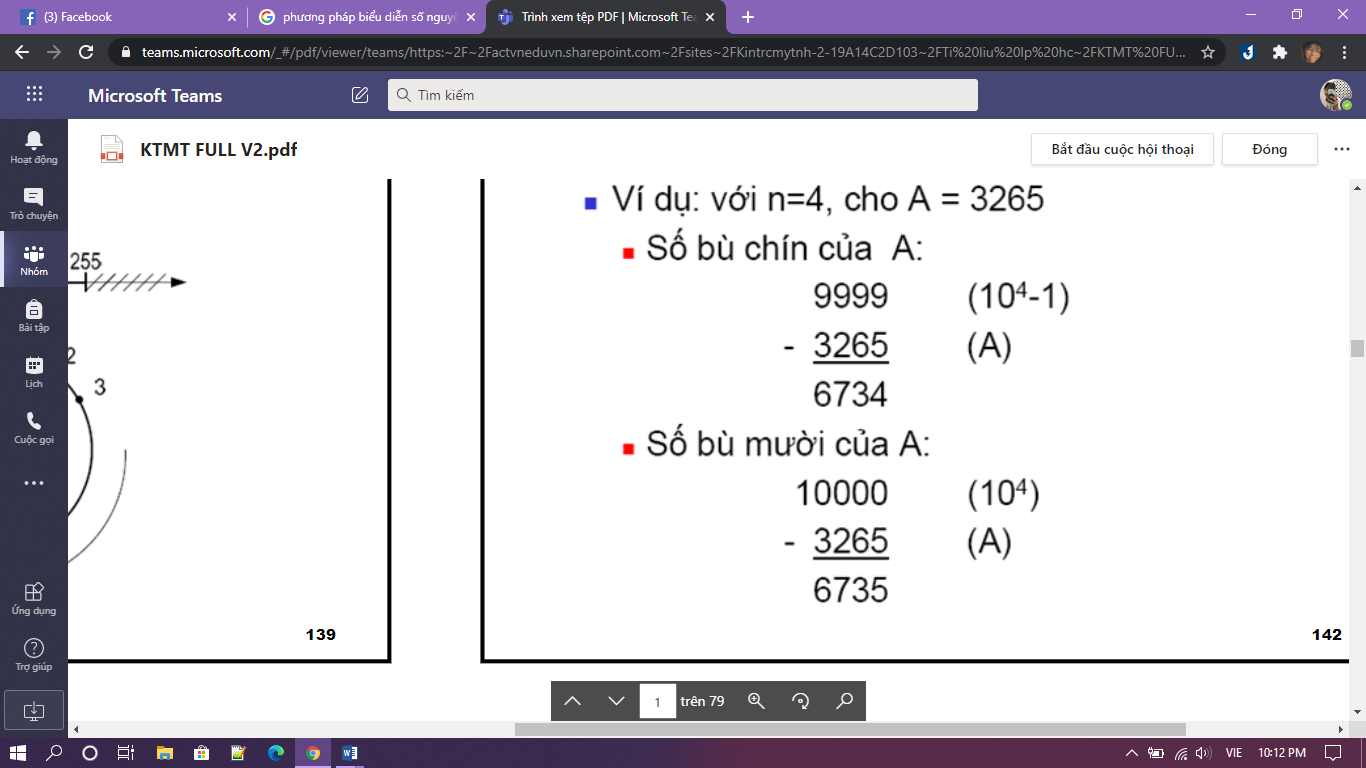
VD:



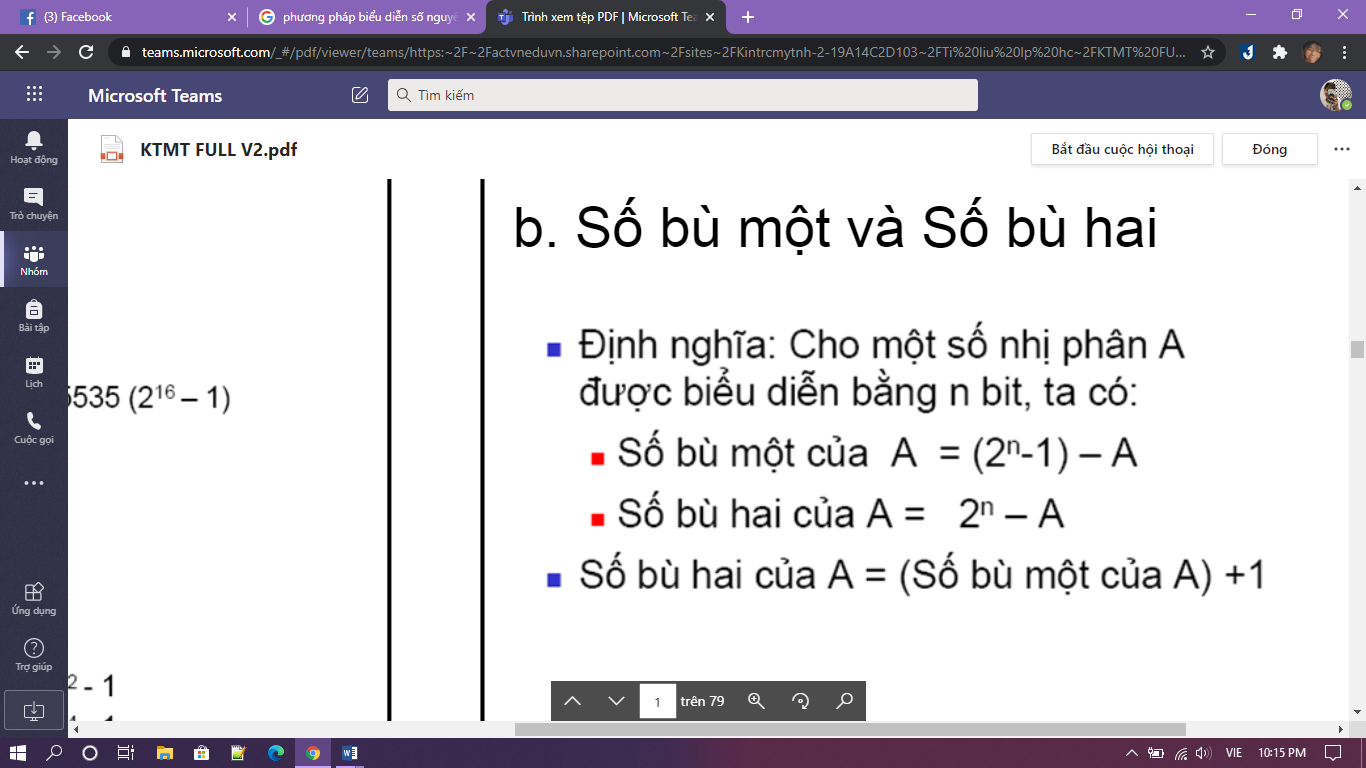
-Biểu diễn số nguyên có dấu: a, Số bù chín và số bù mười

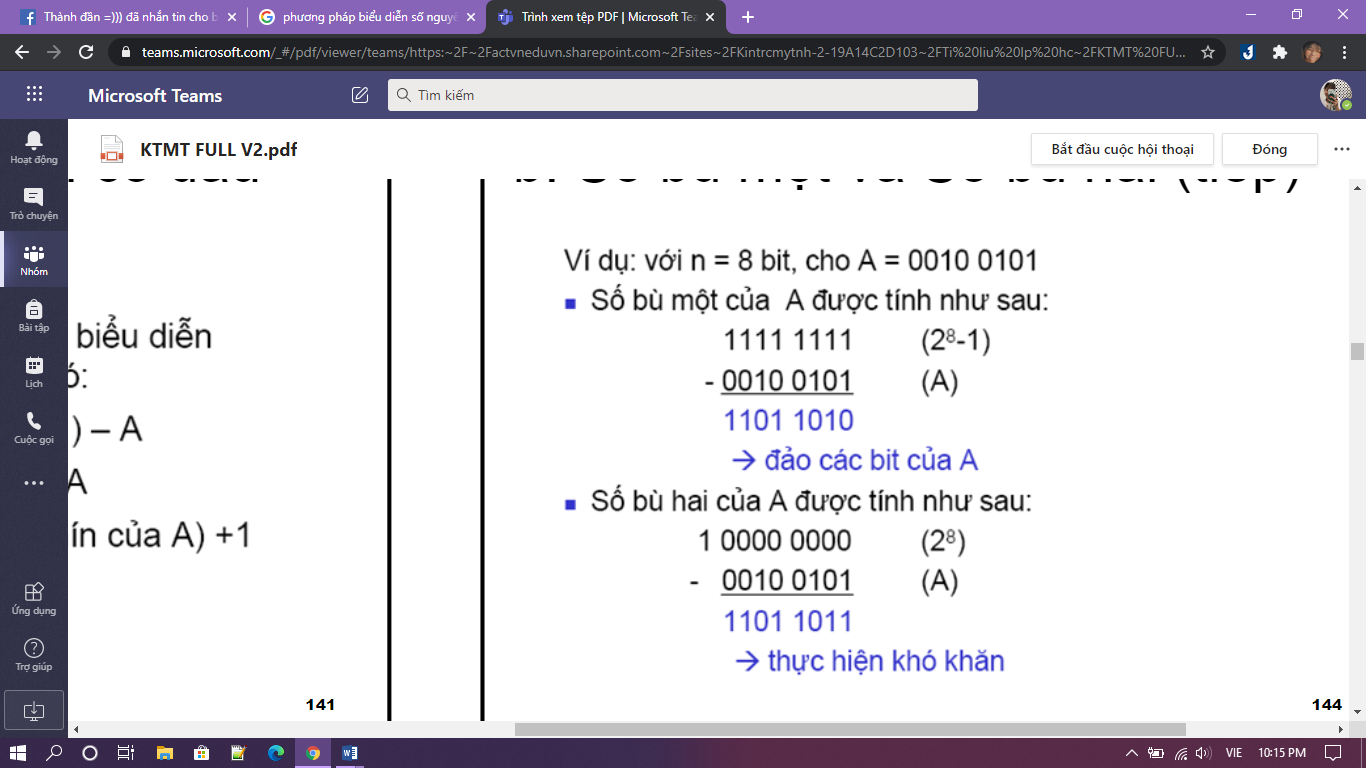


VD:



b, Số bù một và số bù hai





# **Câu 16: Trình bày mục đích của bộ nhớ ảo? Ánh xạ địa chỉ trong bộ nhớ ảo?**

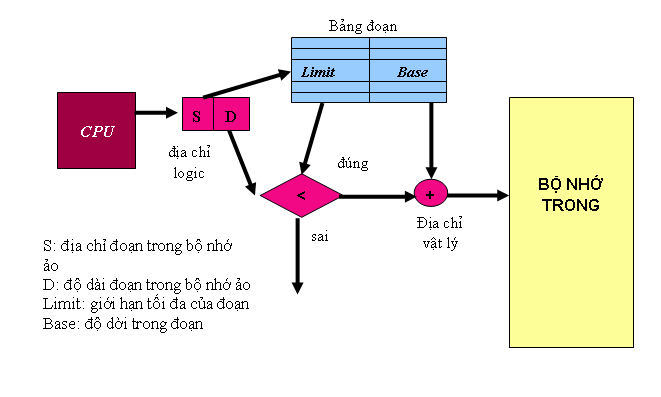
**Trả lời:**

Mục đích:

Bộ nhớ ảo được dùng để cho phép thực hiện cùng lúc nhiều tiến trình (process), mỗi tiến trình có một không gian định vị riêng.

Ngoài việc phân chia không gian bộ nhớ, cần bảo vệ và quản lý tự động các cấp bộ nhớ, bộ nhớ ảo đơn giản hoá việc nạp chương trình vào bộ nhớ để thi hành nhờ một cơ chế được gọi là sự tái định địa chỉ (address relocation). Cơ chế này cho phép một chương trình có thể được thi hành khi nó nằm ở bất cứ vị trí nào trong bộ nhớ.

Định vị trang và định vị đoạn đều dựa vào một cấu trúc dữ liệu trong đó số thứ tự trang hoặc số thứ tự đoạn được có chỉ số. Cho định vị trang, dựa vào bảng trang, địa chỉ trong bộ nhớ vật lý được xác lập cuối cùng là việc đặt kề nhau số thứ của trang vật lý với địa chỉ trong trang. Cho định vị đoạn, dựa vào thông tin trên bảng đoạn, việc kiểm tra tính hợp lệ của địa chỉ được tiến hành. Địa chỉ vật cuối cùng được xác lập bằng cách cộng địa chỉ đoạn và địa chỉ trong đoạn (độ dời trong đoạn).

Ánh xạ địa chỉ giữa bộ nhớ ảo và bộ nhớ vật lý trong cách định vị đoạn 

# **Câu 17: Các loại bộ nhớ ROM của máy tính? Đặc trưng của từng loại?**

**Trả lời:**

Các loại bộ nhớ ROM của máy tính và đặc trưng từng loại như sau:

PROM (Programmble ROM):

• Khi sản xuất chưa ghi dữ liệu

• Cần thiết bị chuyên dùng để ghi bằng chương trình, chỉ ghi được một lần

EPROM (Erasable PROM):

• Khi sản xuất chưa ghi dữliệu

• Cần thiết bị chuyên dụng để ghi bằng chương trình, ghi được nhiều lần

• Trước khi ghi lại, phải xóa bằng tia cực tím

EEPROM (Electrically Erasable PROM):

• Có thể ghi theo từng byte

• Xóa bằng điện

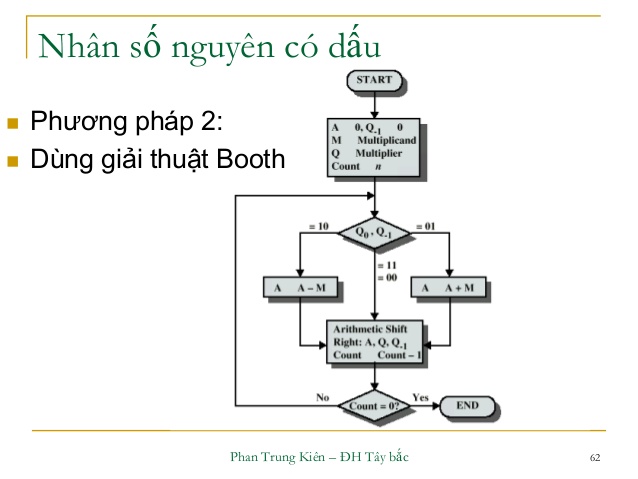
• Ghi lâu hơn đọc

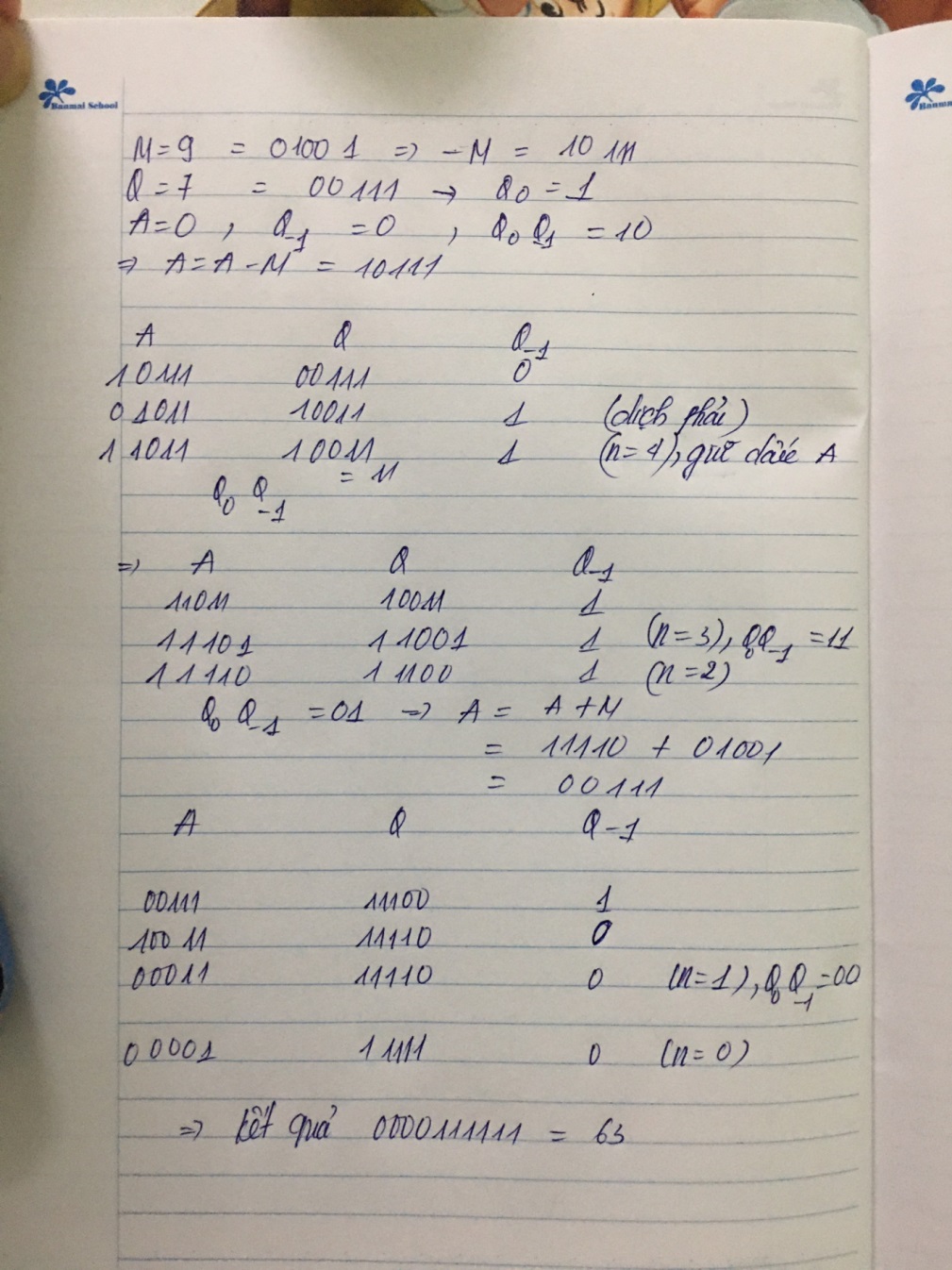
Flash memory (bộ nhớ cực nhanh)

• Ghi theo khối

• Xóa bằng điện

# **Câu 18: Vẽ sơ đồ thuật toán nhân nhanh Booth? Áp dụng tính 7 x 9?**





# **Câu 19: Hãy trình bày phương pháp biểu diễn số thực? Cho ví dụ?**

**Trả lời: (Chưa check kỹ)**

* **Phương pháp biểu diễn số thực:** Một số thực X được biểu diễn theo kiểu số dấu phẩy động như sau:

**X = M \* RE**

* + **M :** Là phần định vị (Mantissa )
  + **R :** Là cơ số ( Radix )
  + **E :** Là phần mũ ( Exponent )
* **Chuẩn IEEE 754/85**
  + Cơ số R = 2
  + Các dạng 32, 44, 64, 80 bit
* **Ví dụ dạng 32 bit**

|  |  |  |
| --- | --- | --- |
| **1 bit** | **8 bit** | **23 bit** |
| **S** | **e** | **m** |

* S là bit dấu ( số dương thì S = 0 )
* e là mã excess của phần mũ E ( e = E + 127 , số 127 là độ lệch bias )
* m là phần lẻ của phần định vị M ( M = 1.m )
* **Do vậy công thức xác định số thực biểu diễn như sau**

**X = ( -1 )S \* 1.m \* 2e – 127**

* Ví dụ biểu diễn số **-2345.12510**
  + B1: Chuyển đổi sang hệ 2 bình thường

-2345.12510 = - 1001 0010 1001.001

* + B2: Chuẩn hóa theo IEEE 754/85 32 bit

- 1001 0010 1001.001

=> - 1.001 0010 1001 001 \* 211

* + B3: Xác định các thông số biểu diễn S, e, m

Số âm nên **S = 1**

E = e - 127 = 11 → e = 127 + 11 = 13810 = 100010102

m = 001 0010 1001 0010 0000 0000 ( 23 bit )

Vậy: X = 1 10001010 001 0010 1001 0010 0000 0000

# **Câu 20: Trình bày khái quát thuật toán thay thế và chính sách ghi trong Cache?**

* Có 4 giải thuật chủ yếu dùng để chọn khối thay thế trong cache:
* Thay thế ngẫu nhiên (RS: Random Selection): để phân bố đồng đều việc thay thế, các khối cần thay thế trong cache được chọn ngẫu nhiên.
* Khối xưa nhất (LRU: Least Recently Used): Các khối đã được thâm nhập sẽ được đánh dấu và khối bị thay thế là khối không được dùng từ lâu nhất.
* Vào trước ra trước (FIFO: First In First Out): Khối được đưa vào cache đầu tiên, nếu bị thay thế, khối đó sẽ được thay thế trước nhất.
* Tần số sử dụng ít nhất (LFU: Least Frequently Used): Khối trong cache được tham chiếu đến ít nhất. Nguyên tắc này sử dụng hệ quả của nguyên tắc sử dụng ô nhớ theo thời gian: nếu các khối mới được dùng có khả năng sẽ được dùng trong tương lai gần, khối thay thế là khối không dùng trong thời gian lâu nhất.
* Chính sách ghi trong cache:
* Ghi đồng thời hay ghi xuyên qua (write-through):
  + Ghi cả cache và bộ nhớ chính
  + Tốc độ chậm
* Ghi lại (Write back):
  + Chỉ ghi ra cache
  + Tốc độ nhanh

*// Có thời gian thì trình bày chi tiết thêm đoạn sau (nhưng để thời gian mà học câu khác)//*

* Ghi đồng thời (Write-Through): Khi hệ thống ghi cho một vị trí bộ nhớ hiện đang được tổ chức trong bộ nhớ cache, nó ghi ra các thông tin mới cho cả các dòng bộ nhớ cache bộ nhớ vị trí thích hợp và chính bản thân cùng 1 lúc. Đây là loại bộ nhớ đệm cung cấp hiệu suất kém hơn ghi lại, nhưng đơn giản để thực hiện và có lợi thế là thống nhất nội bộ, bởi vì bộ nhớ cache là không bao giờ ra khỏi đồng bộ và bộ nhớ cách đó là một bộ nhớ cache ghi lại.
* Ghi lại (Write-back): Khi hệ thống viết cho một vị trí bộ nhớ hiện đang được tổ chức trong bộ nhớ cache, nó chỉ ghi các thông tin mới cho dòng bộ nhớ cache thích hợp. Khi dòng bộ nhớ cache là cuối cùng cần thiết đối với một số địa chỉ bộ nhớ khác, các dữ liệu thay đổi là “ghi lại“ với bộ nhớ hệ thống. Đây là loại bộ nhớ cache cung cấp hiệu suất tốt hơn so với một ghi thông qua bộ nhớ cache, bởi vì nó tiết kiệm (tốn thời gian) ghi chu kỳ bộ nhớ.

# **Câu 21: Các loại bộ nhớ trong của máy tính? Đặc trưng của từng loại? Ngày nay khi mua máy tính người ta thường quan tâm tới bộ nhớ trong nào và có những tham số gì đáng quan tâm nhất?**

**Bộ nhớ máy tính** có thể chia thành hai dạng: Bộ nhớ trong và bộ nhớ ngoài.

* **Bộ nhớ trong:**

+ Bộ nhớ đệm nhanh (*cache memory*): Tốc độ truy xuất nhanh. Thường nằm trong [CPU](https://vi.wikipedia.org/wiki/CPU), một số cache cũ có thể nằm ngoài CPU: như các cache trên đế cắm kiểu slot 1, hoặc cache dạng thanh, có thể tháo rời giống như các thanh [RAM](https://vi.wikipedia.org/wiki/RAM) ngày nay. Bao gồm Cache L1 và Cache L2, Cache L3 (L3 chỉ có ở một số CPU) có tốc độ truy xuất gần bằng tốc độ truyền dữ liệu trong [CPU](https://vi.wikipedia.org/wiki/CPU);

+ Bộ nhớ chính (*main memory*);

Bộ nhớ [RAM](https://vi.wikipedia.org/wiki/RAM) (*Random access memory*), hay [Bộ nhớ truy cập ngẫu nhiên](https://vi.wikipedia.org/wiki/RAM): Tốc độ truy cập nhanh, lưu trữ dữ liệu tạm thời, dữ liệu sẽ bị mất đi khi bị cắt nguồn điện;

Bộ nhớ [ROM](https://vi.wikipedia.org/wiki/B%E1%BB%99_nh%E1%BB%9B_ch%E1%BB%89_%C4%91%E1%BB%8Dc) (*Read Only Memory*), hay [Bộ nhớ chỉ đọc](https://vi.wikipedia.org/wiki/B%E1%BB%99_nh%E1%BB%9B_ch%E1%BB%89_%C4%91%E1%BB%8Dc): Lưu trữ các chương trình mà khi mất nguồn điện cung cấp sẽ không bị (xóa) mất. Ngày nay còn có công nghệ FlashROM tức bộ nhớ ROM không những chỉ đọc mà còn có thể ghi lại được, nhờ có công nghệ này BIOS được cải tiến thành FlashBIOS.

* **Bộ nhớ ngoài:**

Bộ nhớ ngoài được hiểu là bộ nhớ máy tính gắn bên ngoài thùng máy, có thể dùng để mang đi lại được. Bao gồm:

* Bộ nhớ từ: đĩa cứng, Đĩa mềm,
* Bộ nhớ quang: CD, DVD,...
* Bộ nhớ bán dẫn: flash disk, thẻ nhớ
* **Bộ nhớ ảo:**

Ngày nay khi mua máy tính người ta thường quan tâm đến bộ nhớ: [RAM](https://vi.wikipedia.org/wiki/RAM) , *cache*

Ví dụ thông số cơ bản của 1 máy tính: Bộ vi xử lý (CPU): Core i5-3317U (1,7 GHz, 4GB RAM, 3MB smart cache)**.**

**Cho ta biết:**

* **CPU là** Core i5-3317U
* Tốc độ xung nhịp: 1,7Ghz.
* RAM: dung lượng 4GB.
* **Cache: dung lượng** 3MB

# **Câu 22: Trình bày hiệu quả thực hiện lệnh sử dụng đường ống (Pipeline) trong kiến trúc RISC với đường ống 2 giai đoạn, 3 giai đoạn, 4 giai đoạn so với thực thi tuần tự?**

**Trả lời: //** *(Không kịp thì bỏ qua các hình vẽ)*

+ Với đường ống hai giai đoạn, gồm Nhận lệnh (Fetch Instruction -FI) và Thực hiện lệnh (Execute Instruction -EI), 7 lệnh được thực hiện trong 8 chu kì xung nhịp. Với thực thi tuần tự thông thường, 7 lệnh được thực hiện trong 14 chu kì xung nhịp.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Lệnh 1 | FI | EI |  |  |  |  |  |  |
| Lệnh 2 |  | FI | EI |  |  |  |  |  |
| Lệnh 3 |  |  | FI | EI |  |  |  |  |
| Lệnh 4 |  |  |  | FI | EI |  |  |  |
| Lệnh 5 |  |  |  |  | FI | EI |  |  |
| Lệnh 6 |  |  |  |  |  | FI | EI |  |
| Lệnh 7 |  |  |  |  |  |  | FI | EI |

Chu kỳ xung nhịp

+Với đường ống ba giai đoạn, gồm Nhận lệnh (Fetch Instruction -FI), Giải mã lệnh (Decode Instruction -DI) và Thực hiện lệnh (Execute Instruction -EI), 7 lệnh được thực hiện trong 9 chu kì xung nhịp. Thực thi tuần tự thông thường: 7 lệnh được thực hiện trong 21 chu kì xung nhịp

Chu kì xung nhịp

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Lệnh 1 | FI | DI | EI |  |  |  |  |  |  |
| Lệnh 2 |  | FI | DI | EI |  |  |  |  |  |
| Lệnh 3 |  |  | FI | DI | EI |  |  |  |  |
| Lệnh 4 |  |  |  | FI | DI | EI |  |  |  |
| Lệnh 5 |  |  |  |  | FI | DI | EI |  |  |
| Lệnh 6 |  |  |  |  |  | FI | DI | EI |  |
| Lệnh 7 |  |  |  |  |  |  | FI | DI | EI |

+Với đường ống 4 giai đoạn, gồm Nhận lệnh (Fetch Instruction -FI), Giải mã lệnh (Decode Instruction -DI), Thực hiện lệnh (Execute Instruction -EI) và Ghi toán hạng (Write Operands -WO), 7 lệnh được thực hiện trong 10 chu kì xung nhịp. Với thực thi tuần tự thông thường: 7 lệnh được thực hiện trong 28 chu kì xung nhịp

Chu kì xung nhịp

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Lệnh 1 | FI | DI | EI | WO |  |  |  |  |  |  |
| Lệnh 2 |  | FI | DI | EI | WO |  |  |  |  |  |
| Lệnh 3 |  |  | FI | DI | EI | WO |  |  |  |  |
| Lệnh 4 |  |  |  | FI | DI | EI | WO |  |  |  |
| Lệnh 5 |  |  |  |  | FI | DI | EI | WO |  |  |
| Lệnh 6 |  |  |  |  |  | FI | DI | EI | WO |  |
| Lệnh 7 |  |  |  |  |  |  | FI | DI | EI | WO |

=> Pipeline làm tăng tốc độ thực hiện các lệnh, tăng hiệu quả thực hiện lệnh so với thực thi tuần tự. (ý này phải kết luận được)

**Câu 23: Trình bày quá trình đọc/ ghi bộ nhớ trong máy tính?**

**Trả lời: (chưa check lại)**

Đơn vị bộ nhớ hỗ trợ hai hoạt động cơ bản: Đọc và ghi. Các hoạt động đọc dữ liệu được lưu trữ trước đó và hoạt động ghi lưu trữ một giá trị trong bộ nhớ.

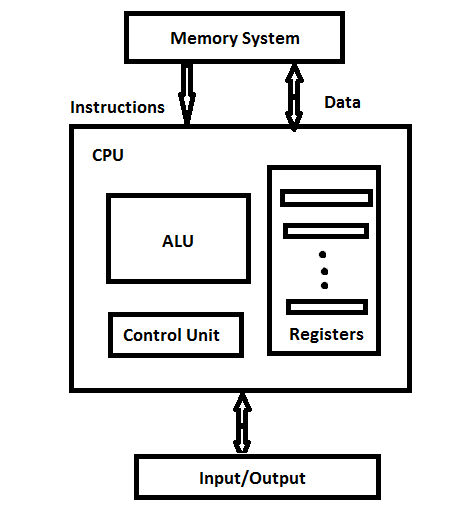
Các bước sau đây phải được tuân theo trong một chu trình đọc thông thường:

* Đặt địa chỉ của địa điểm sẽ được đọc trên bus địa chỉ.
* Kích hoạt tín hiệu điều khiển đọc bộ nhớ trên bus điều khiển.
* Đợi bộ nhớ lấy dữ liệu từ bộ nhớ địa chỉ vị trí.
* Đọc dữ liệu từ bus dữ liệu.
* Gửi tín hiệu điều khiển đọc bộ nhớ để chấm dứt chu kỳ đọc.

Các bước sau đây phải được tuân theo trong một chu trình ghi thông thường:

* Đặt địa chỉ của địa điểm sẽ được ghi trên bus địa chỉ.
* Đặt dữ liệu được ghi trên bus dữ liệu.
* Kích hoạt tín hiệu điều khiển ghi bộ nhớ trên bus điều khiển.
* Đợi bộ nhớ lưu trữ dữ liệu tại vị trí địa chỉ ..
* Gửi tín hiệu điều khiển ghi bộ nhớ để chấm dứt chu kỳ ghi.

# **Câu 24: Trình bày các khối trong bộ vi xử lý của máy tính (CPU)? Chức năng, nhiệm vụ của từng khối?**



CPU đơn giản cần bao gồm 3 thành phần quan trọng nhất: Tập các thanh ghi (Registers), bộ xử lý số học và logic (ALU) và bộ điều khiển (CU).

* Tập các thanh ghi của các loại máy tính là khác nhau và thường được chia thành hai loại: các thanh ghi được dùng với mục đích chung và các thanh ghi với mục đích đặc biệt. Các thanh ghi mục đích chung được sử dụng trong nhiều trường hợp như có thể dùng làm nới lưu trữ dữ liệu, dùng để lưu các toán hạng của 1 chỉ lệnh. Trong khi các thanh ghi với mục đích đặc biệt chỉ có một số chức năng bên trong CPU, như bộ đến chương trình là thanh ghi chỉ dùng với mục đích đặc biệt là chỉ dùng để lưu địa chỉ của chỉ lệnh tiếp theo.
* Bộ logic số học ALU cung cấp 1 mạch số cần thiết để thục hiện các phép toán số học (như công, trừ, nhân, chia) logic (như NOT, AND, OR) các phép toán dịch trong bộ chỉ lệnh.
* Bộ Điều khiển (CU) chị trách nhiệm điểu khiển mọi hoạt động của CPU như tìm nạp chỉ lệnh từ bộ nhớ chính, giải mã nó, xác định xem nó thuộc loại nào và cuối cùng là ra chỉ lệnh cho các bộ phận bên trong CPU thực hiện.

# **Câu 25: Các loại bộ nhớ RAM của máy tính? Đặc trưng của từng loại? Ngày nay máy tính thường sử dụng bộ nhớ RAM nào, các RAM của máy tính hay ghi DDR3 có ý nghĩa gì?**

Tùy theo công nghệ chế tạo, người ta phân biệt thành 2 loại:

SRAM (Static RAM): RAM tĩnh

DRAM (Dynamic RAM): RAM động

**RAM tĩnh**

RAM tĩnh được chế tạo theo công nghệ ECL (dùng trong CMOS và BiCMOS). Mỗi bit nhớ gồm có các cổng logic với 6 transistor MOS. SRAM là bộ nhớ nhanh, việc đọc không làm hủy nội dung của ô nhớ và thời gian thâm nhập bằng chu kỳ của bộ nhớ.

- Các bit được lưu trữ bằng các Flip-Flop →thông tin ổn định

* Cấu trúc phức tạp
* Dung lượng chip nhỏ
* Tốc độ nhanh
* Đắt tiền
* Dùng làm bộ nhớ cache

**RAM động**

RAM động dùng kỹ thuật MOS. Mỗi bit nhớ gồm một transistor và một tụ điện. Việc ghi nhớ dữ liệu dựa vào việc duy trì điện tích nạp vào tụ điện và như vậy việc đọc một bit nhớ làm nội dung bit này bị hủy. Do vậy sau mỗi lần đọc một ô nhớ, bộ phận điều khiển bộ nhớ phải viết lại nội dung ô nhớ đó. Chu kỳ bộ nhớ cũng theo đó mà ít nhất là gấp đôi thời gian thâm nhập ô nhớ.

Bộ nhớ DRAM chậm nhưng rẻ tiền hơn SRAM.

* 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
* Rẻ tiền hơn
* Dùng làm bộ nhớ chính

Ngày nay các máy tính thường sử dụng RAM động (đặc biệt là DDR SDRAM).

Các Ram của máy tính hay ghi DDR3 có ý nghĩa:

DDR là viết tắt của Double Data Rate - Tốc độ dữ liệu gấp đôi.

Và DDR3 có thể truyền 8 khối dữ liệu trong một chu kỳ xung nhịp.

DDR3 có tốc độ bus 800/1066/1333/1600 Mhz, số bit dữ liệu là 64, điện thế là 1.5v, tổng số chân là 240.

# **Câu 26: Khái niệm Cache miss, thời gian truy cập cache, các nguyên nhân Cache miss? Cho ví dụ?**

**Khái niệm cache miss:** Cache dùng làm vùng đệm truy xuất nên nếu CPU truy xuất data mà có sẵn trong cache thì thời gian truy xuất nhanh hơn nhiều. Hiệu quả của cache ngoài việc cho tốc đọ truy xuất nhanh còn phụ thuộc vào cache hit hoặc miss.

* Cache hit: dữ liệu có sẵn trong cache.
* Cache miss: dữ liệu chưa có sẵn trong cache.

**Thời gian truy cập cache**

Gọi c là thời gian truy xuất của cache, m là trười gian truy xuất bộ nhớ, h là tỉ lệ thành công(là tỉ số giữa số lần tham chiếu cache với tổng số lần tham chiếu h = (k-1)/k ).

Tỉ lệ thất bại (1-h):

Thời gian truy xuất trung bình: = c + (1-h)/m

Khi h -> 1, tất cả truy xuất đều tham chiếu tới cache, thời gian truy xuất trung bình ->c

Khi h -> 0, cần phải tham chiếu bộ nhớ cính mọi lúc, thới gian truy xuất trung bình -> c+m

**Nguyên nhân:**

Khi bộ xử lý cần đọc hay viết một vị trí trong bộ nhớ chính, nó sẽ tìm entry tương ứng trong cache đầu tiên. Cache sẽ kiểm tra nội dung của vị trí dữ liệu yêu cầu trong bất cứ cache line nào có thể có địa chỉ. Nếu bộ xử lý tìm thấy vị trí dữ liệu trong cache, một cache hit đã xảy ra. Tuy nhiên, nếu bộ xử lý không tìm thấy được vị trí dữ liệu trong cache, thì một cache miss đã xảy ra

# **Câu 27: Các chế độ địa chỉ của bộ vi xử lý? Nêu rõ ưu nhược điểm của từng chế độ?**

**Trả lời:**

Có nhiều cách chia số lượng chế độ địa chỉ nhưng cơ bản nhất gồm những chế độ địa chỉ sau (*Cột thứ nhất dịch ra tiếng việt hoặc không kẻ bảng có thể trình bày theo kiểu liệt kê ra thành 3 câu: Chế độ địa chỉ- Ưu điểm- Nhược điểm trong bảng ra thành dòng*)

Địa chỉ tức thời – Immediate  
Địa chỉ trực tiếp – Direct  
Địa chỉ gián tiếp – Indirect  
Địa chỉ thanh ghi – Register  
Địa chỉ gián tiếp thanh ghi – Register Indirect  
Địa chỉ dịch chuyển – Displacement  
Địa chỉ ngăn xếp – Stack



A – Nội dung trong trường địa chỉ trong một lệnh.

R – Nội dung trong trường địa chỉ mà chỉ ra một thanh ghi nào đó.

EA – Địa chỉ thực của nơi chứa (memory hoặc register) toán hạng.

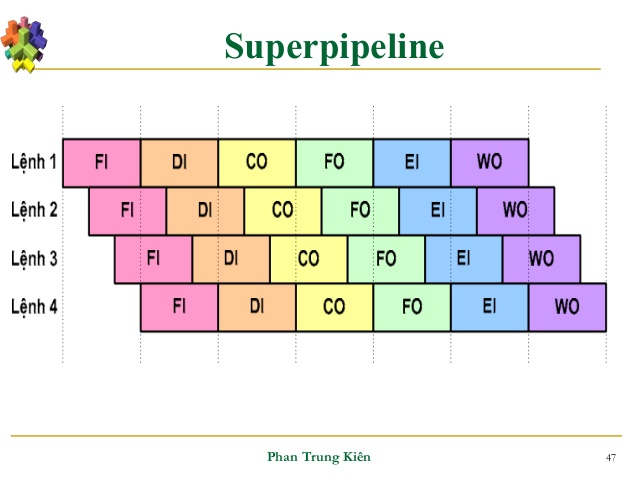
(X) – Nội dung của trị trí nhớ X hoặc là thanh ghi X

# **Câu 28: Hãy trình bày các kiểu hệ thống bộ vi xử lý song song và cách tổ chức song trong trong máy tính sử dụng bộ vi xử lý song song?**

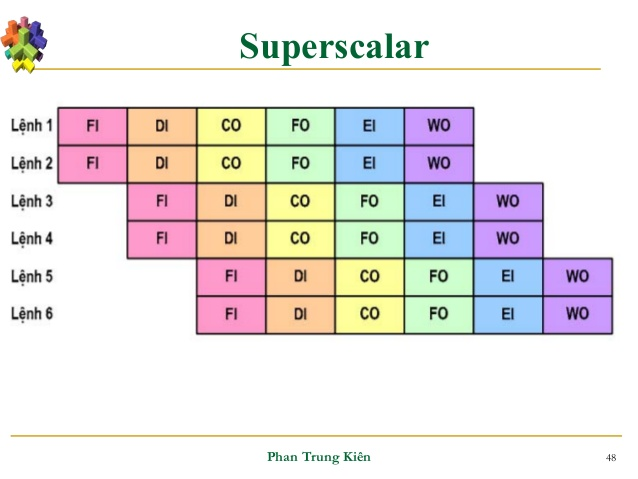
**Trả lời:**

Các kiểu hệ thống bộ vi xử lý song song

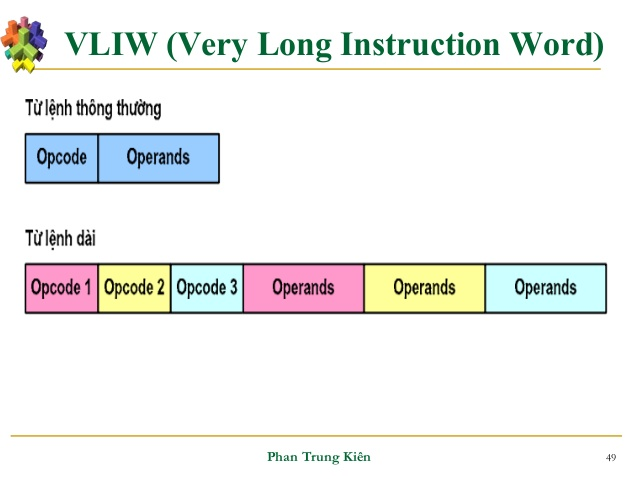
* Siêu đường ống (Superpipeline & Hyperpipeline)



* Siêu vô hướng (Superscalar)



* VLIW (Very Long Instruction Word)



# **Câu 29: Trình bày chi tiết các bước thực hiện lệnh trong máy tính?**

**Trả lời:**

*(Trình bày chi tiết nên phải đủ, không còn thời gian thì ghi 6 bước bôi đậm)*

Các bước thực hiện lệnh:

**+Nhận lệnh:**

• CPU đưa địa chỉ của lệnh cần nhận từ bộ đếm chương trình PC ra bus địa chỉ

•CPU phát tín hiệu điều khiển đọc bộ nhớ

•Lệnh từ bộ nhớ được đặt lên bus dữ liệu và được CPU copy vào thanh ghi lệnh IR

•CPU tăng nội dung PC để trỏ sang lệnh kế

**+Giải mã lệnh:**

•Lệnh từ thanh ghi lệnh IR được đưa đến đơn vị điều khiển

•Đơn vị điều khiển tiến hành giải mã lệnh để xác định thao tác phải thực hiện

•Giải mã lệnh xảy ra bên trong CPU

**+Nhận dữ liệu:**

•CPU đưa địa chỉ của toán hạng ra bus địa chỉ

•CPU phát tín hiệu điều khiển đọc

•Toán hạng được đọc vào CPU

•Tương tự như nhận lệnh

**+Thực hiện lệnh:**

•Có nhiều dạng tuỳ thuộc vào lệnh

•Có thể là:

-Đọc/Ghi bộ nhớ

-Vào/Ra

-Chuyển giữ liệu giữa các thanh ghi

-Thao tác số học/logic

-Chuyển điều khiển (rẽ nhánh)

**+Ghi toán hạng:**

•CPU đưa địa chỉ ra bus địa chỉ

•CPU đưa dữ liệu cần ghi ra bus dữ liệu

•CPU phát tín hiệu điều khiển ghi

•Dữ liệu trên bus dữ liệu được copy đến vị trí xác định

**+Ngắt:**

•Nội dung của bộ đếm chương trình PC (địa chỉ trở về sau khi ngắt) được đưa ra bus dữ liệu.

•CPU đưa địa chỉ (thường được lấy từ con trỏ ngăn xếp SP) ra bus địa chỉ

•CPU phát tín hiệu điều khiển ghi bộ nhớ

•Địa chỉ trở về trên bus dữ liệu được ghi ra vị trí xác định (ở ngăn xếp)

•Địa chỉ lệnh đầu tiên của chương trình con điều khiển ngắt được nạp vào PC

# **Câu 30: Các loại bus trong máy tính? Chức năng của từng loại? ( giống câu 6)**

* Các loại bus trong máy tính: Bus dữ liệu, Bus địa chỉ, Bus điều khiển
* Chức năng của từng loại:
  + **Bus dữ liệu:** cung cấp đường dẫn để truyền dữ liệu giữa các module hệ thống. Các đường này gọi là bus dữ liệu. Bus dữ liệu có thể là 4, 16, 32, 64, 128,… thậm chí nhiều đường riêng biệt, số lượng các đường này gọi là độ rộng của bus dữ liệu. Vì mỗi đường chỉ có thể truyền tải một bit tín hiệu tại 1 thời điểm, số lượng các đường cho biết bao nhiêu bit có thể truyền cùng 1 lúc. Độ rộng của bus dữ liệu là nhân tố quan trọng trong việc xác định hiệu suất hoạt động của hệ thống.
  + **Bus địa chỉ:** được sử dụng để xác minh nguồn hoặc đích của dữ liệu trên bus dữ liệu. Độ rộng của bus địa chỉ định rõ khả năng xác định bộ nhớ lớn nhất của hệ thống. Ngoài ra, các đường địa chỉ được sử dụng để địa chỉ hóa các cổng vào ra.
  + **Bus điều khiển:** được sử dụng đểđiều khiển truy cập đến và sử dụng các đường dữ liệu và địa chỉ. Bởi vì các đường địa chỉ và dữ liệu được sử dụng chung bởi tất cả các thiết bị, phải có một phương tiện kiểm soát việc sử dụng chúng. Các tín hiệu điều khiển truyền cả lệnh và thông tin thời gian giữa các module hệ thống. Các tín hiệu thời gian xác định tính hợp lệ của dữ liệu và thông tin địa chỉ. Các tín hiệu lệnh xác định các phép toán được thực hiện.
  + Ngoài ra còn phân loại BUS theo đường truyền: BUS đồng bộ và BUS không đồng bộ.

# **Câu 31: Cấu trúc lệnh? Số địa chỉ trong lệnh? Cho ví dụ minh họa?**

* Cấu trúc của lệnh gồm 2 phần:

+Mã thao tác: mã hóa cho thao tác mà bộ xử lý phải thực hiện

+Địa chỉ các toán hạng chỉ ra nơi chứa các toán hạng mà thao tác sẽ tác động

• Toán hạng nguồn: dữ liệu vào của thao tác

• Toán hạng đích: dữ liệu ra của thao tác

* Số lượng địa chỉ toán hạng trong lệnh

+Ba địa chỉ toán hạng:

• 2 toán hạng nguồn, 1 toán hạng đích

• c = a + b

+Hai địa chỉ toán hạng:

• Một toán hạng vừa là toán hạng nguồn vừa là toán hạng đích; toán hạng còn lại là toán hạng nguồn

• a = a + b

+Một địa chỉ toán hạng:

• Một toán hạng được chỉ ra trong lệnh: a++

+0 địa chỉ toán hạng:

• Các toán hạng đều được ngầm định

• Sử dụng Stack

• Ví dụ:

push a

push b

add

pop c

có nghĩa là : c = a+b

# **Câu 32: Trình bày giải pháp đường ống lệnh (Pipeline) và Sơ đồ thời gian thực hiện lệnh bằng đường ống lệnh? Sơ đồ đường ống lệnh CPU 6 giai đoạn?**

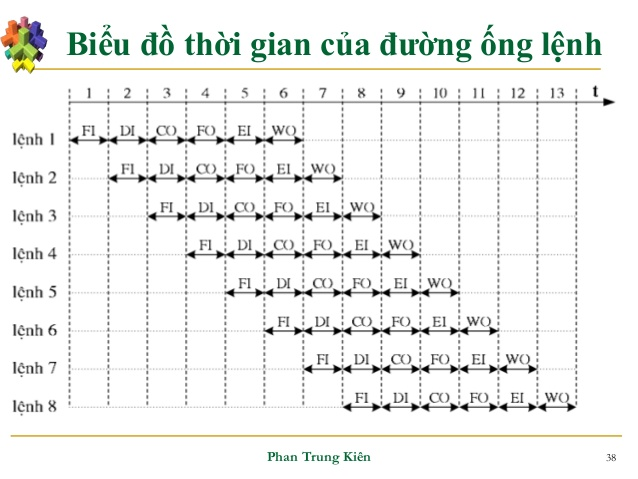
**Trả lời:**

+Giải pháp đường ống lệnh (Instruction pipeling): Chia chu trình lệnh thành các công đoạn và cho phép thực hiện gối lên nhau (như dây chuyền lắp ráp).

Chẳng hạn 6 giai đoạn: (*Không nhớ được tiếng anh thì nhớ từ viết tắt để còn dùng trong sơ đồ sau*)

* Nhận lệnh (Fetch Instruction – FI)
* Giải mã lệnh (Decode Instruction – DI)
* Tính địa chỉ toán hạng (Calculate Operand Address – CO)
* Nhận toán hạng (Fetch Operands – FO)
* Thực hiện lệnh (Execute Instruction – EI)
* Ghi toán hạng (Write Operands – WO)

+Biểu đồ thời gian của đường ống lệnh:



# **BÀI TẬP**

**Có 3 dạng chính trong đó dạng 3 thường hay thi nhất. dạng 2 nếu thi thường là bài 4 điểm.**

**Chú ý chung: Xem IC nhớ chỉ đọc hay đọc ghi. Các cổng Logic tùy ý hay cho 2 đầu vào 1 đầu ra.**

**Cách giải các dạng như sau:**

**Bài tập dạng 1: Mở rộng từ nhớ (thay đổi dữ liệu)**

***Chú ý: Xem bộ nhớ chỉ đọc hay bộ nhớ đọc/ghi dữ liệu***

VD: Thiết kế module nhớ đọc ghi 8K x 16 bit từ các chip nhớ 8K x 8 bit.

(Dạng này đơn giản nhất nhưng phục vụ cho làm dạng 3)

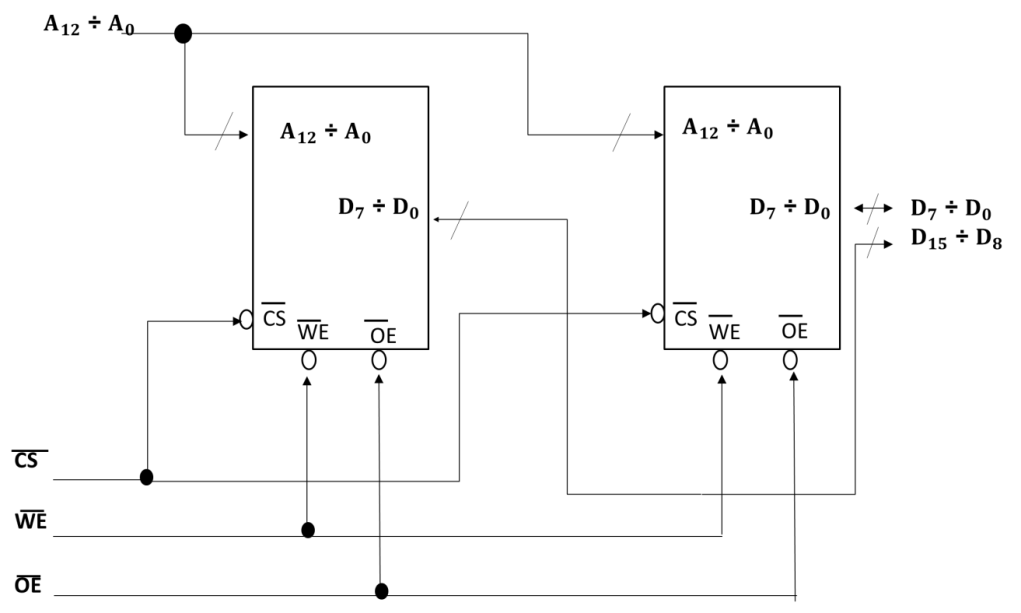
Giải:

+ Chip nhớ: 8KBx8bit = 213 x 8 bit: có 13 đường địa chỉ A12-A0, 8 đường dữ liệu D7-D0

+ Module nhớ: 8KBx16bit = 213x 16 bit: có 13 đường địa chỉ A13-A0, 16 đường dữ liệu D15-D0

+ Số chip nhớ cần dùng là : = 2 chip.

+ Sơ đồ nguyên lý mở rộng bộ nhớ:

**Bài tập dạng 2: Mở rộng số lượng từ nhớ (thay đổi địa chỉ)**

**VD1: Cho IC nhớ cơ bản 4KBx8bit, mạch cơ bản AND, OR, XOR, NAND… hai đầu vào 1 đầu ra (số lượng không hạn chế). Tổ chức bộ nhớ chỉ đọc có dung lượng 24KBx8bit dữ liệu. (Lập bảng chân lý và vẽ mạch giải mã chọn IC nhớ, vẽ sơ đồ nguyên lý mở rộng bộ nhớ).**

**Giải:**

+ Chip nhớ: 4KBx8bit = 212 x 8 bit: có 12 đường địa chỉ A11-A0, 8 đường dữ liệu D7-D0

+ Module nhớ: 24KBx8bit = 24 x 210 x 8 bit: có 15 đường địa chỉ A14-A0, 8 đường dữ liệu D7-D0

+ Số chip nhớ cần dùng là : = 6 chip.

+ Số đường địa chỉ đưa vào bộ giải mã: 14-11=3 đường.

+ Cần dùng bộ giải mã: 3 đầu vào (A,B,C nối với A12, A13, A14) 8 đầu ra (,,….) để chọn Chip nhớ. Chân chọn bộ giải mã là (Chip select) mức tích cực thấp.

Có 8 đầu ra nhưng số chip nhớ chỉ có 6 (IC0 đến IC5) vậy ta chỉ lấy từ đến (bảng giải mã vẫn nên vẽ đủ)

+ Ta có bảng chân lý:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A14 | A13 | A12 | IC0 | IC1 | IC2 | IC3 | IC4 | IC5 |  |  |
|  | **C** | **B** | **A** |  |  |  |  |  |  |  |  |
| 0 | **0** | **0** | **0** | **0** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | **0** | **0** | **1** | 1 | **0** | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | **0** | **1** | **0** | 1 | 1 | **0** | 1 | 1 | 1 | 1 | 1 |
| 0 | **0** | **1** | **1** | 1 | 1 | 1 | **0** | 1 | 1 | 1 | 1 |
| 0 | **1** | **0** | **0** | 1 | 1 | 1 | 1 | **0** | 1 | 1 | 1 |
| 0 | **1** | **0** | **1** | 1 | 1 | 1 | 1 | 1 | **0** | 1 | 1 |
| 0 | **1** | **1** | **0** | 1 | 1 | 1 | 1 | 1 | 1 | **0** | 1 |
| 0 | **1** | **1** | **1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **0** |
| 1 | x | x | x | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

(Chú ý: Đầu ra là OR của các cổng đầu vào tránh nhầm lẫn thành phép XOR như trong giáo trình. Tốt nhất không nên ghi vào chỉ ghi như bảng trên là đủ)+Vẽ mạch giải mã chọn IC nhớ: (Do số cổng tùy ý nên các bạn có thể có cách vẽ đến 16 cổng OR cũng đúng)



+Sơ đồ nguyên lý mở rộng bộ nhớ:



Chú ý: Với bài bộ nhớ đọc/ghi thì vẽ thêm WE trong IC nhớ. Và chú ý đầu vào và giống các bài dưới.

Ví dụ 2: **Áp dụng cho bài thường có 4 đường địa chỉ đưa vào bộ giải mã. Bài này nếu có thì 4 điểm.**

Cho IC nhớ cơ bản 2MBx8bit, mạch cơ bản AND, OR, XOR, NAND, NOT… (số lượng không hạn chế), mạch giải mã địa chỉ 3 đầu vào, 8 đầu ra.

Tổ chức bộ nhớ đọc ghi có dung lượng 30MBx8bit dữ liệu. (Lập bảng chân lý và vẽ mạch giải mã chọn IC nhớ, vẽ sơ đồ nguyên lý mở rộng bộ nhớ).

Giải:

Nhận dạng đề: Bài thiết kế module nhớ chỉ thay đổi địa chỉ, giữ nguyên dữ liệu

+ Chip nhớ: 2MBx8bit = 221 x 8 bit: có 21 đường địa chỉ A20-A0, 8 đường dữ liệu D7-D0

+ Module nhớ: 30MBx8bit = 30 x 220 x 8 bit: có 25 đường địa chỉ A24-A0, 8 đường dữ liệu D7-D0

+ Số chip nhớ cần dùng là : = 15 chip.

+ Số đường địa chỉ đưa vào bộ giải mã là : 25-21=4 đường

Cần dùng bộ giải mã: 4 đầu vào (A,B,C,D nối với A21….A24) 16 đầu ra (,,….). Ta sử dụng ghép 2 bộ giải mã 3 đầu vào, 8 đầu ra. A24 là chân chọn Chip mức tích cực thấp.

Bảng chân lý:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A24 | A23 | A22 | A21 | IC0 | IC1 |  |  |  |  |  |  |  |  |  |  |  |  | IC14 |  |
| D | C | B | A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 1 |  | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 1 | 0 |  |  | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 1 | 1 |  |  |  | 0 |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 0 | 0 |  |  |  |  | 0 |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 0 | 1 |  |  |  |  |  | 0 |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 1 | 0 |  |  |  |  |  |  | 0 |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 1 | 1 |  |  |  |  |  |  |  | 0 |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 | 0 |  |  |  |  |  |  |  |  | 0 |  |  |  |  |  |  |  |
| 1 | 0 | 0 | 1 |  |  |  |  |  |  |  |  |  | 0 |  |  |  |  |  |  |
| 1 | 0 | 1 | 0 |  |  |  |  |  |  |  |  |  |  | 0 |  |  |  |  |  |
| 1 | 0 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  | 0 |  |  |  |  |
| 1 | 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  | 0 |  |  |  |
| 1 | 1 | 0 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |  |  |
| 1 | 1 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |  |
| 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |

Chú ý: Số chip nhớ cần dùng bao nhiêu thì chỉ ghi đến đó, như bài trên 15 chip thì ghi đến IC14 thôi. Tránh trường hợp ghi full đến IC15Mạch nguyên lý mở rộng bộ nhớ. (Bài này dài nên vẽ luôn không cần vẽ riêng mạch giải mã địa chỉ)

**Bài tập dạng 3: Mở rộng cả địa chỉ và dữ liệu (Thi thường vào dạng này)**

VD: Cho IC nhớ cơ bản 8MBx8bit, mạch cơ bản AND, OR, XOR, NAND… hai đầu vào 1 đầu ra (số lượng không hạn chế). Tổ chức bộ nhớ đọc ghi có dung lượng 32MBx16bit dữ liệu.(Lập bảng chân lý và vẽ mạch giải mã chọn IC nhớ, vẽ sơ đồ nguyên lý mở rộng bộ nhớ).

(Các bài khác tương tự chỉ khác nhau chỗ màu đỏ. Thường 8 chíp gồm 4 bộ 2 chip nên ghép 2 lần)

Giải:

+ Chip nhớ: 8MBx8bit = 223 x 8 bit: có 23 đường địa chỉ A22-A0, 8 đường dữ liệu D7-D0

+ Module nhớ: 32MBx16bit = 225 x 16 bit: có 25 đường địa chỉ A24-A0, 16 đường dữ liệu D15-D0

+ Số chip nhớ cần dùng là : = 8 chip gồm 4 bộ 2 chip.

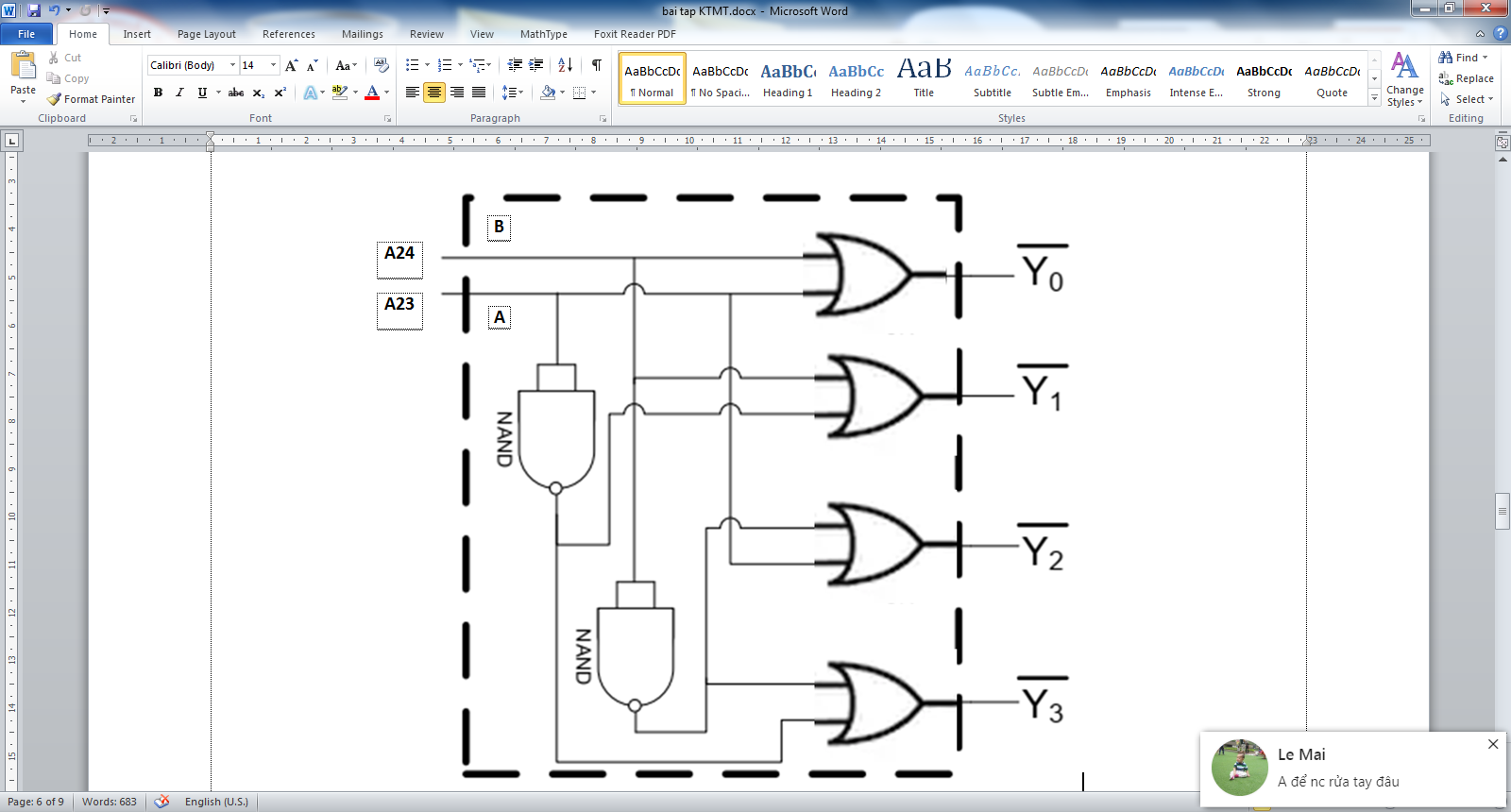
+ Số đường địa chỉ đưa vào bộ giải mã địa chỉ là: 25-23=2 đường (A24, A23)

+ Cần dùng bộ giải mã: 2 đầu vào (A,B nối với A23, A24) 4 đầu ra (,,). Chân chọn bộ giải mã là (Chip select) mức tích cực thấp.

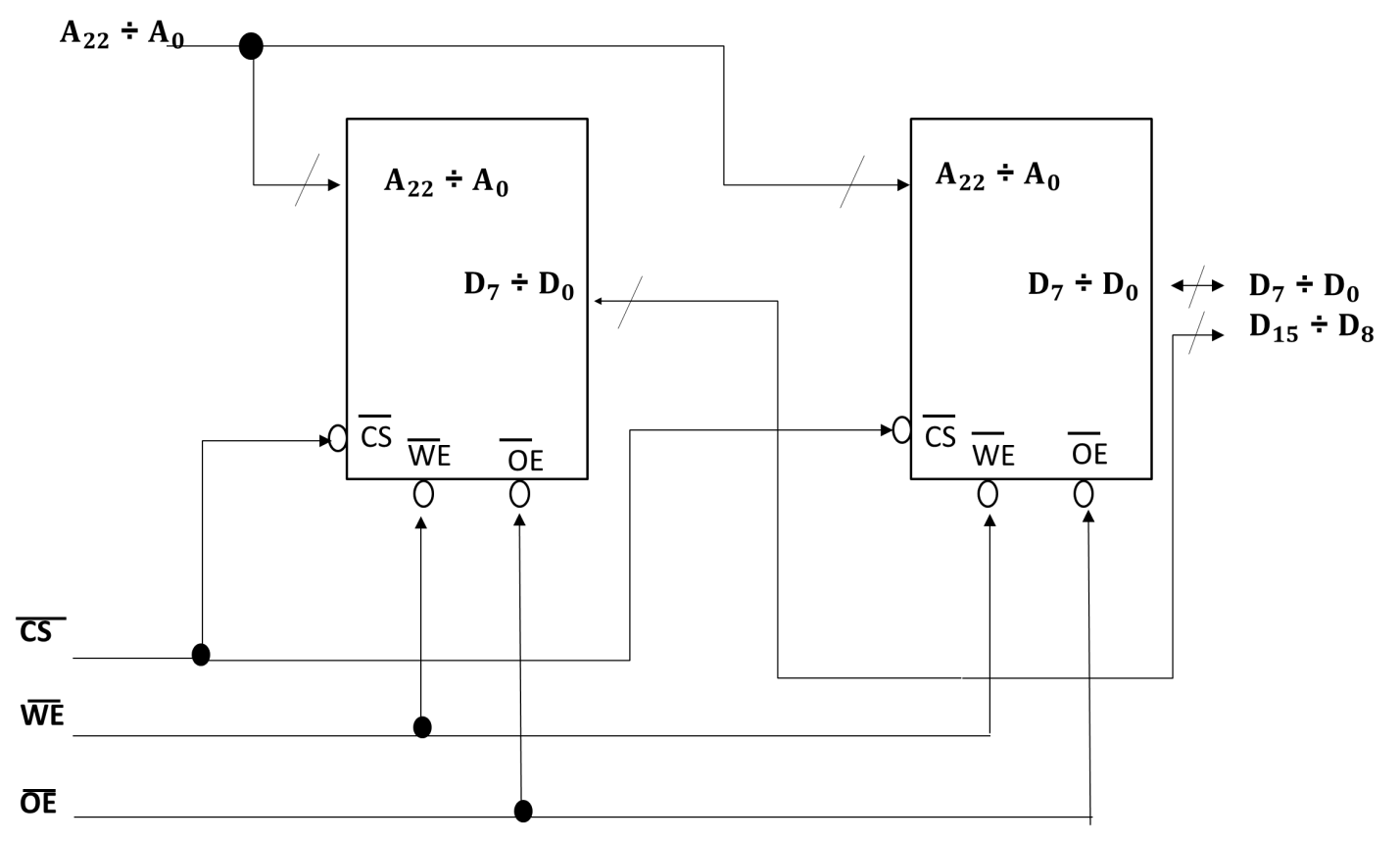
+ Ta có bảng chân lý:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A24 | A23 |  |  |  |  |
|  | B | A |  |  |  |  |
| 0 | 0 | 0 | **0** | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | **0** | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | **0** | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | **0** |
| 1 | x | x | 1 | 1 | 1 | 1 |

+Vẽ mạch giải mã chọn IC nhớ



+Sơ đồ nguyên lý mở rộng bộ nhớ **8MBx8 bit** thành **8MBx16 bit**:



+Sơ đồ nguyên lý mở rộng bộ nhớ 8MBx16 bit thành 32MBx16 bit:



**(Nhớ vẽ mũi tên các đường dữ liệu)**