

GIỚI THIỆU MÔN HỌC

Tên môn học : VI XỬ LÝ

Phân phối giờ : 75 tiết (45 LT, 30 TN)

Website để tải bài giảng : <http://doe.dee.hcmut.edu.vn/>

Sách & giáo trình chính:

1. Vi xử lý - Hồ Trung Mỹ

Tài liệu tham khảo :

1. **Họ vi điều khiển 8051** - Tống Văn On.

2. **The 8051 Microcontroller** - I.Scott Mackenzie - 2nd Ed

3. **The 8051 Microcontroller: Architecture, Programming, and Applications** - Kenneth J. Ayala - West Publishing Company

Lê Thị Kim Anh

Bộ môn Điện Tử – Khoa Điện Điện Tử.

Đại học Bách Khoa TP. HCM

Email: kimanhlebk@hcmut.edu.vn

cuu duong than cong . com

cuu duong than cong . com



TÓM TẮT MÔN HỌC

Nội dung kiến thức:

Chương 1: Khái niệm chung

Chương 2: Giới thiệu họ vi điều khiển MCS51

Chương 3: Tập lệnh

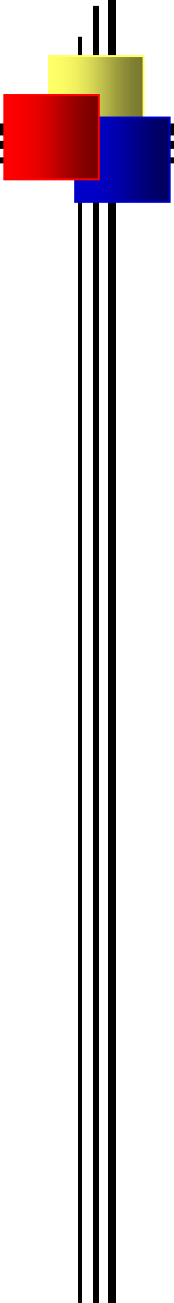
Chương 4: Bộ định thời (Timer)

Chương 5: Thu/phát nối tiếp (Serial port)

Chương 6: Ngắt (Interrupt)

Kết quả đạt được:

Kiến thức căn bản về hệ vi xử lý và CPU tổng quát -Thiết kế hệ vi xử lý 8 bit.



ĐIỂM VÀ CÁCH ĐÁNH GIÁ

Bài tập trên lớp :

- Cộng vào điểm thi cuối kỳ tùy thuộc vào số lượng bài làm đúng.
(Tối đa 1 điểm)

Bài tập nộp tại lớp :10%

Kiểm tra giữa kỳ:

- Thời gian: 45 ÷ 60 phút
- Hình thức: tự luận
- Đánh giá: 15%.

Thí nghiệm: 20%

Thi cuối kỳ:

- Thời gian: 90 ÷ 120 phút
- Hình thức: tự luận
- Đánh giá: 55%.

KHÁI NIỆM CHUNG

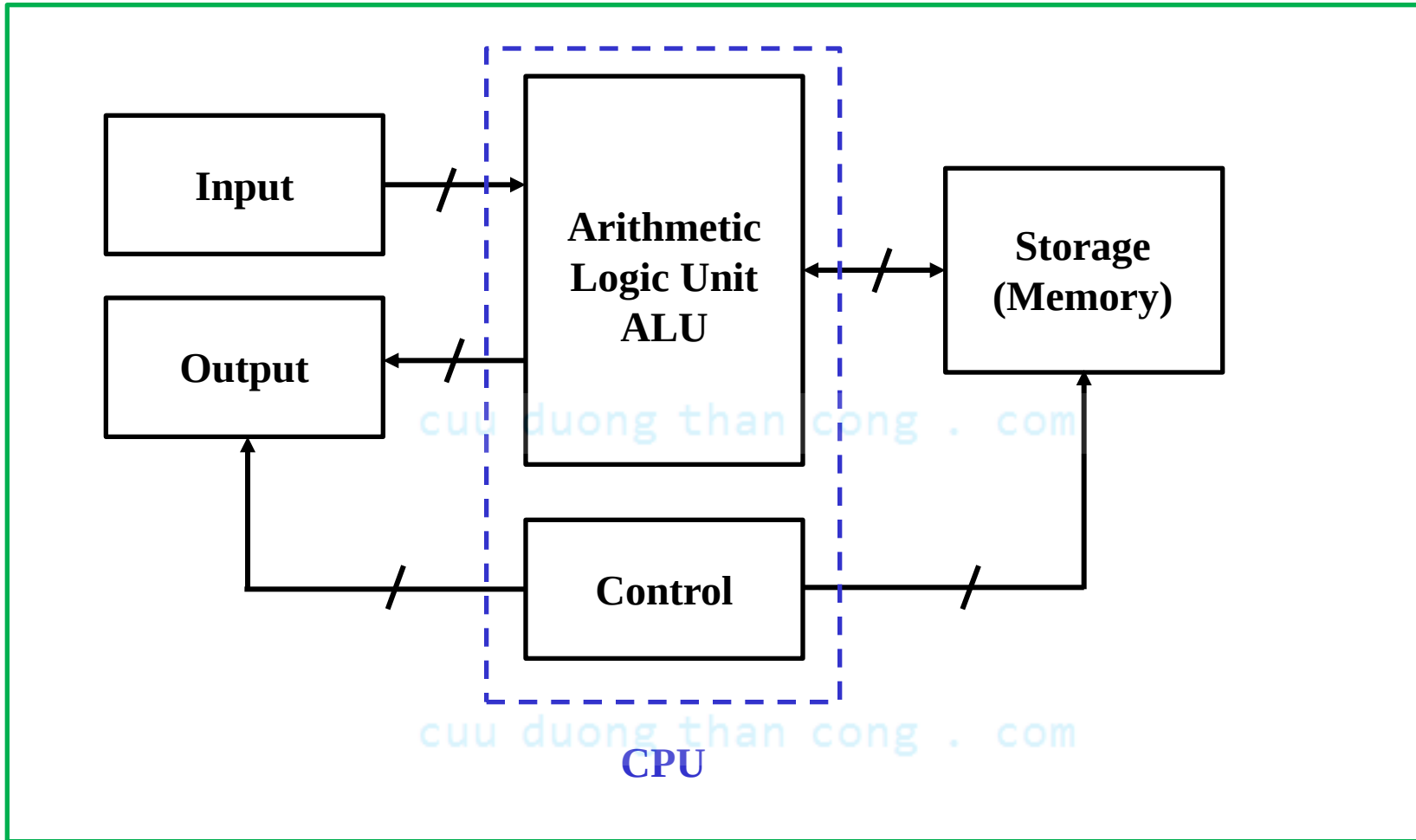
I. KHÁI NIỆM

Hardware (Phần cứng): bao gồm các đối tượng hữu hình – các mạch tích hợp, các board mạch in, cáp, nguồn cấp điện, bộ nhớ, đầu đọc thẻ, máy in và các thiết bị đầu cuối khác.

Software (Phần mềm): bao gồm các giải thuật và các biểu diễn trên máy tính của giải thuật này gọi là các chương trình.

Firmware (Phần dẻo): bao gồm các phần mềm được đặt vào bên trong các mạch điện tử trong thời gian sản xuất chúng.

Máy tính - Máy vi tính và Vi xử lý



Sơ đồ khối máy tính cổ điển

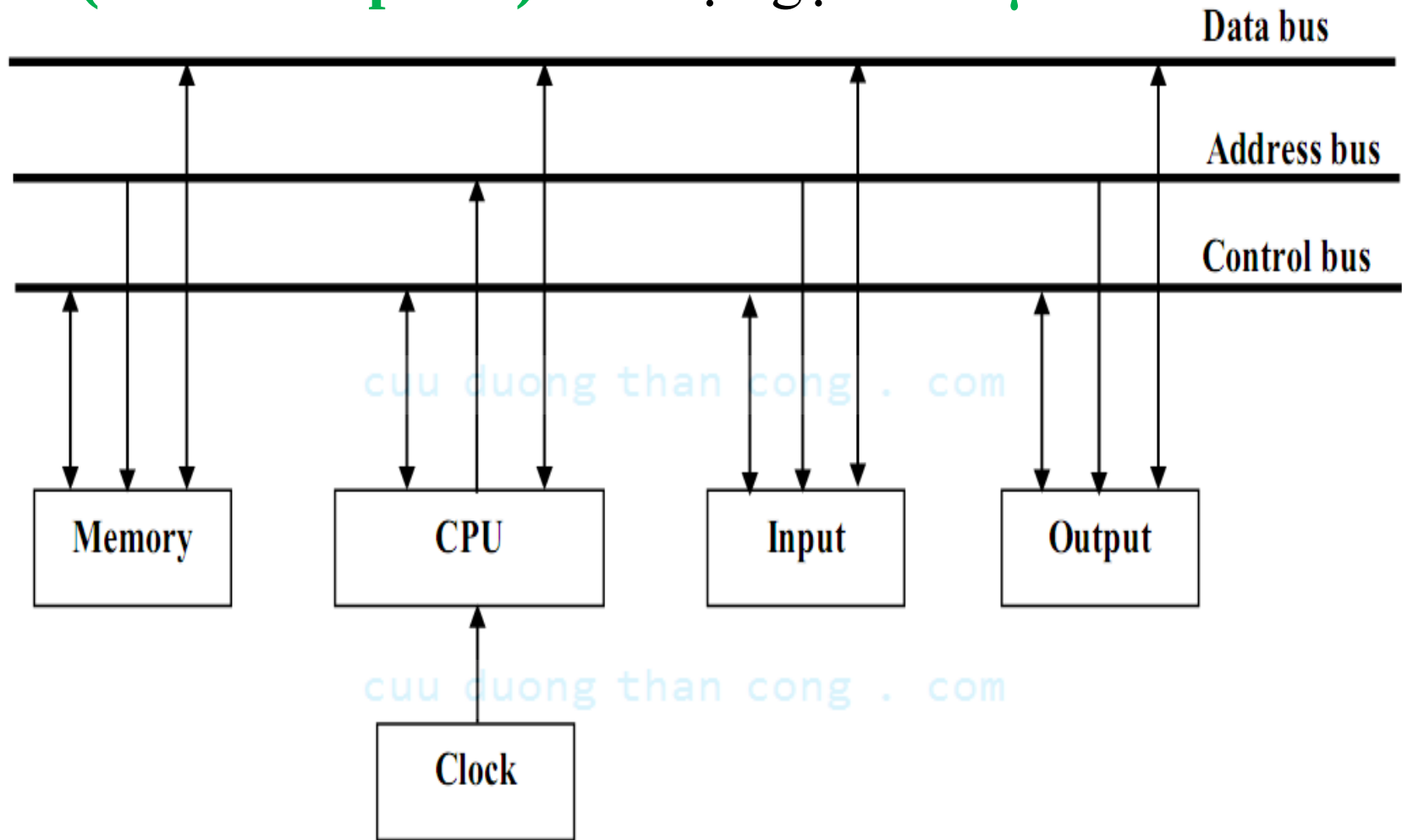
Một **Máy tính** cổ điển gồm 5 phần chính:

- **ALU**: đơn vị số học và logic. Thực hiện các phép tính +, -, *, /, và các phép toán logic(AND, OR, NOT,...).
- **Control**: đơn vị điều khiển và kiểm soát các luồng dữ liệu khác nhau.
- **Input**: thiết bị nhập.
- **Output**: thiết bị xuất.
- **Storage**: nơi lưu trữ chương trình và các kết quả trung gian.

Với các máy thế hệ cũ thì CPU có thể bao gồm nhiều IC (thậm chí nhiều mạch) và kích thước lớn.

Từ khi phát triển công nghệ IC thì cả CPU chỉ chứa trong 1 chip và lúc này người ta gọi nó là **Vi xử lý** (Microprocessor-MPU).

Một máy tính dựa trên vi xử lý thì được gọi là **máy vi tính (microcomputer)** và được gọi tắt là **μC**.



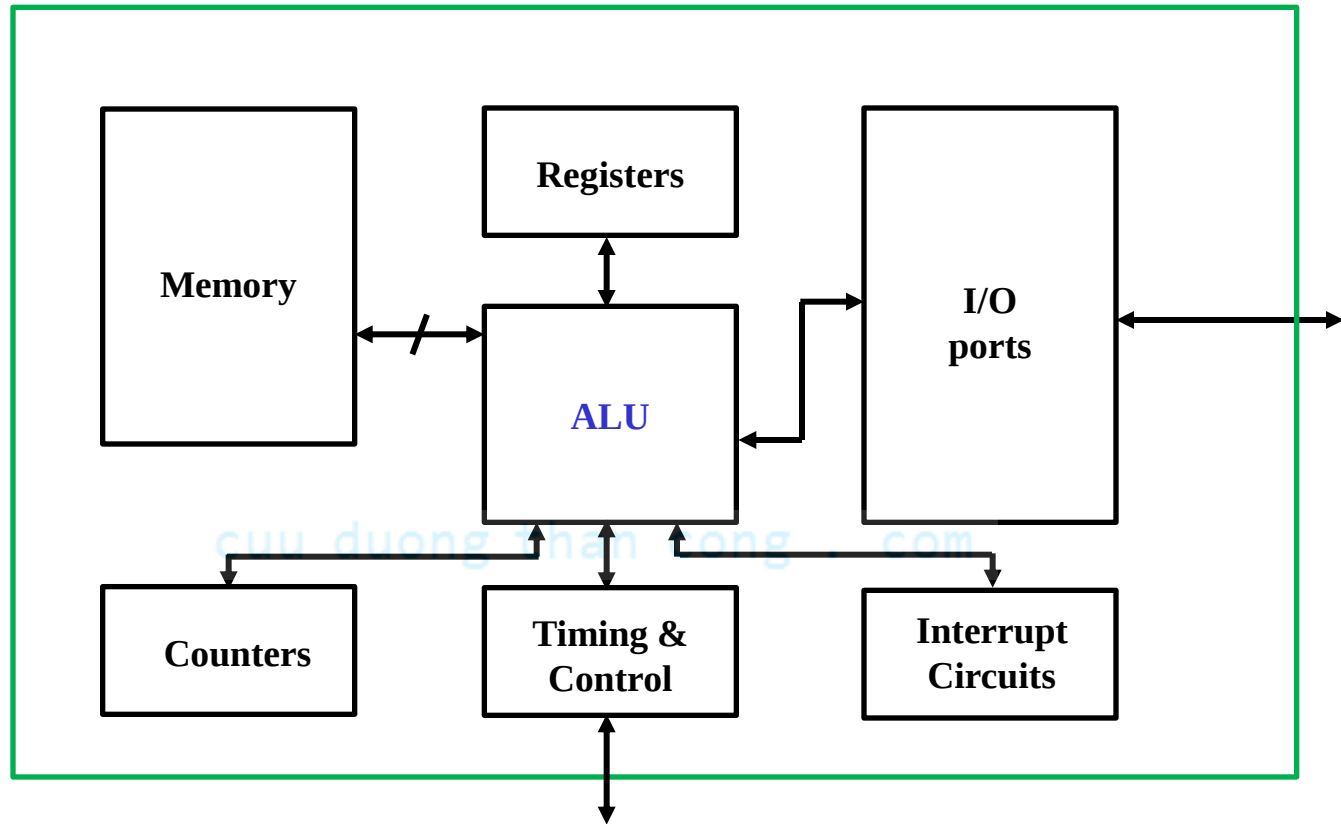
Sơ đồ khối máy vi tính

Bài giảng môn Vi xử lý
GV: Lê Thị Kim Anh

Một hệ thống máy vi tính bao gồm các khối sau:

- **Bộ nhớ (memory):** gồm bộ nhớ chính (ROM hoặc RAM để chứa dữ liệu và các lệnh) và có thể có thêm bộ nhớ phụ (ổ đĩa mềm, ổ đĩa cứng, CD-ROM,...).
- **Thiết bị nhập (input) và xuất (output):** dùng trao đổi thông tin giữa CPU và bên ngoài. Ví dụ như bàn phím, chuột,... là thiết bị nhập, màn hình, máy in,... là các thiết bị xuất.
- **Vi xử lý:** là CPU của hệ thống bao gồm ALU, các thanh ghi và mạch điều khiển.
- **Bus:** dùng để kết nối các khối cơ bản trong máy vi tính với nhau. Có 3 bus cơ bản là *bus địa chỉ*, *bus dữ liệu* và *bus điều khiển*.
- **Xung nhịp (Clock):** dùng để đồng bộ các khối trong hoạt động của hệ thống máy vi tính.

Vì điều khiển (MicroController-MCU)



Sơ đồ khối một MCU

So với MPU thì MCU có thêm 1 số mạch làm cho MCU hoạt động như một thiết bị duy nhất. MCU làm việc với các dữ liệu được đưa vào qua cổng nhập nối tiếp hoặc song song được điều khiển bởi phần mềm được cất trong bộ nhớ chương trình.

So sánh giữa Vi xử lý và Vi điều khiển

MicroProcessor - μP (Vi xử lý):

- Được thiết kế để thực hiện chức năng CPU trong hệ máy vi tính.
- Tập lệnh được sắp xếp để cho phép mã và một lượng lớn dữ liệu được chuyển đi giữa vi xử lý với bộ nhớ và thanh ghi ngoài.
- Các tác vụ tác động với nhóm bit không nhỏ hơn 4 bit.

MicroController - μC (Vi điều khiển)

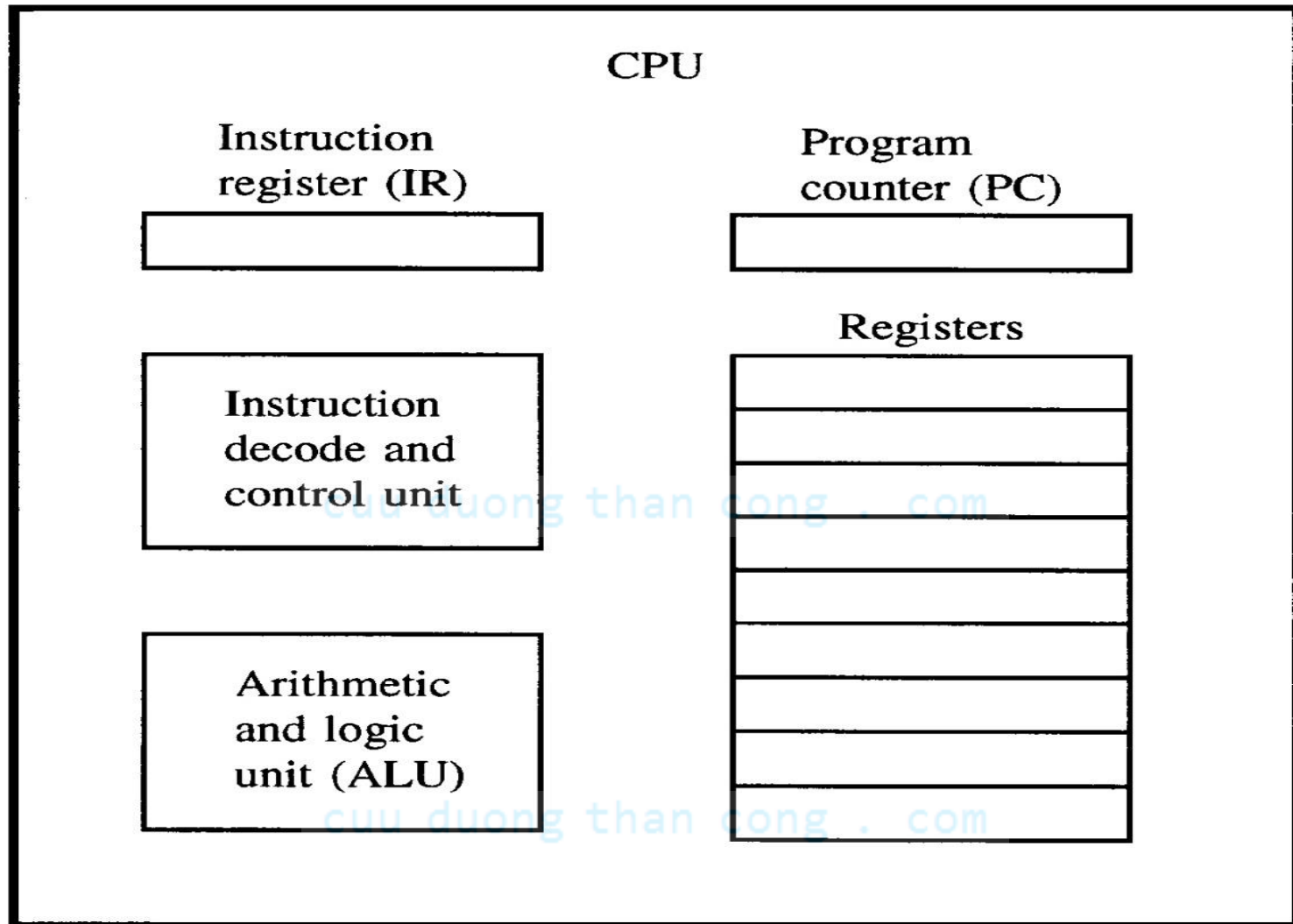
- Được thiết kế để làm việc với mạch ngoài tối thiểu.
- Tập lệnh đơn giản (khoảng 255 lệnh).
- Các tác vụ có thể tác động lên từng bit.

MCU = CPU + Bộ nhớ + Giao tiếp I/O

CPU(Central processor unit)

Chia làm 3 loại chính:

- **Multichip CPU**: gồm 2 hay nhiều chip LSI (Large Scale Integration) để cài đặt ALU (Arithmetic Logic Unit) và phần điều khiển.
- **Microprocessor**: ứng dụng nhiều trong công việc điều khiển, có kết hợp cả bộ nhớ trong cùng một IC.
- **Single chip processor** (còn gọi là Microcomputer hay Microcontroller): là 1 chip LSI/VLSI chứa toàn bộ một máy tính.



Đơn vị xử lý trung tâm CPU

CPU: quản lý tất cả các hoạt động của hệ thống và thực hiện tất cả các thao tác trên dữ liệu.

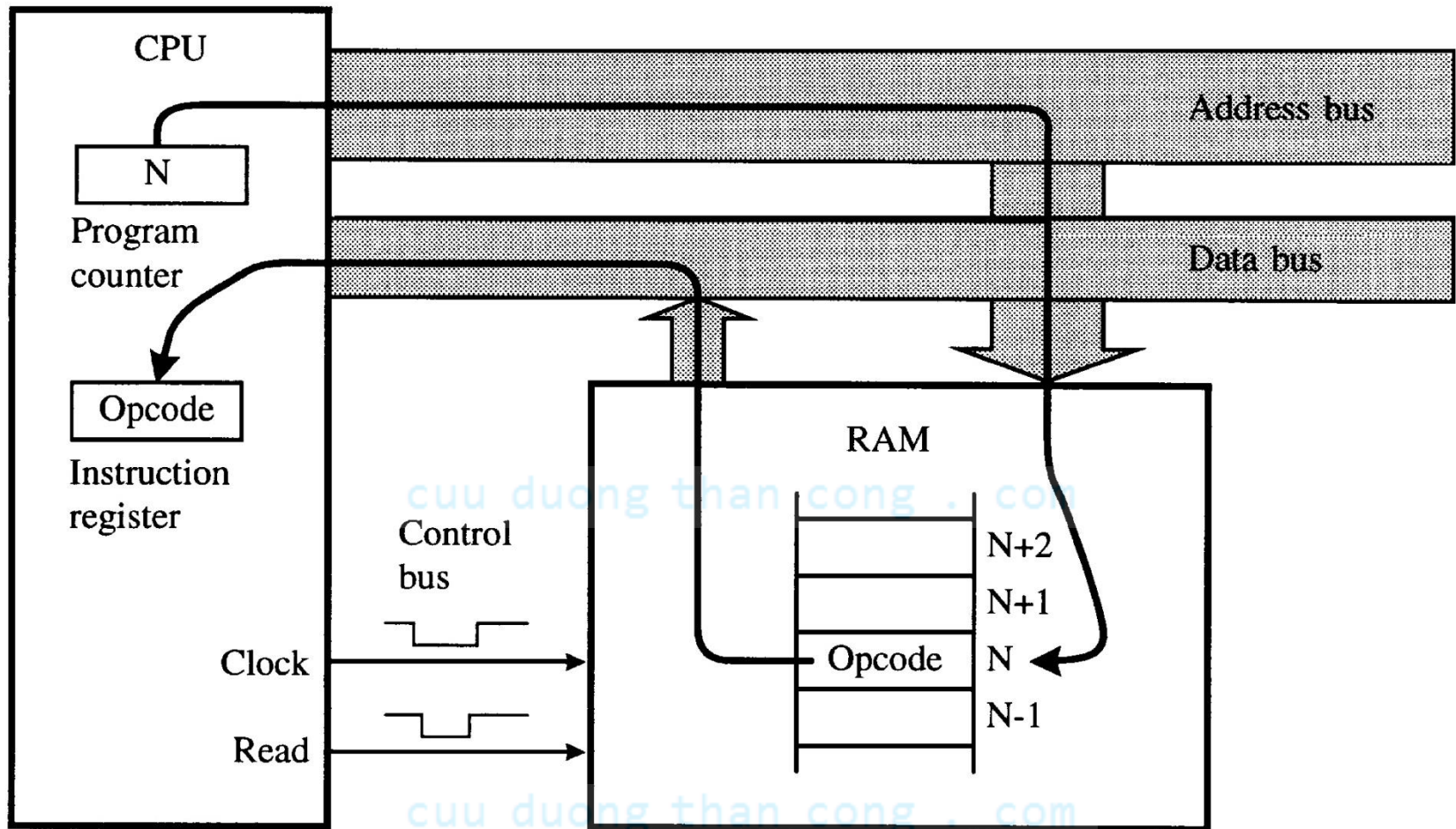
Registers: tập các thanh ghi, lưu giữ tạm thời các thông tin.

ALU: thực hiện các thao tác trên các thông tin chứa trong các thanh ghi.

Intrusion decode and control unit: đơn vị giải mã lệnh và điều khiển có nhiệm vụ xác định thao tác cần thực hiện và thiết lập các hoạt động cần thiết để thực hiện thao tác.

IR: thanh ghi lệnh lưu giữ mã nhị phân của lệnh để được thực thi.

PC: thanh ghi bộ đếm chương trình lưu giữ địa chỉ của lệnh kế tiếp trong bộ nhớ cần được thực thi.

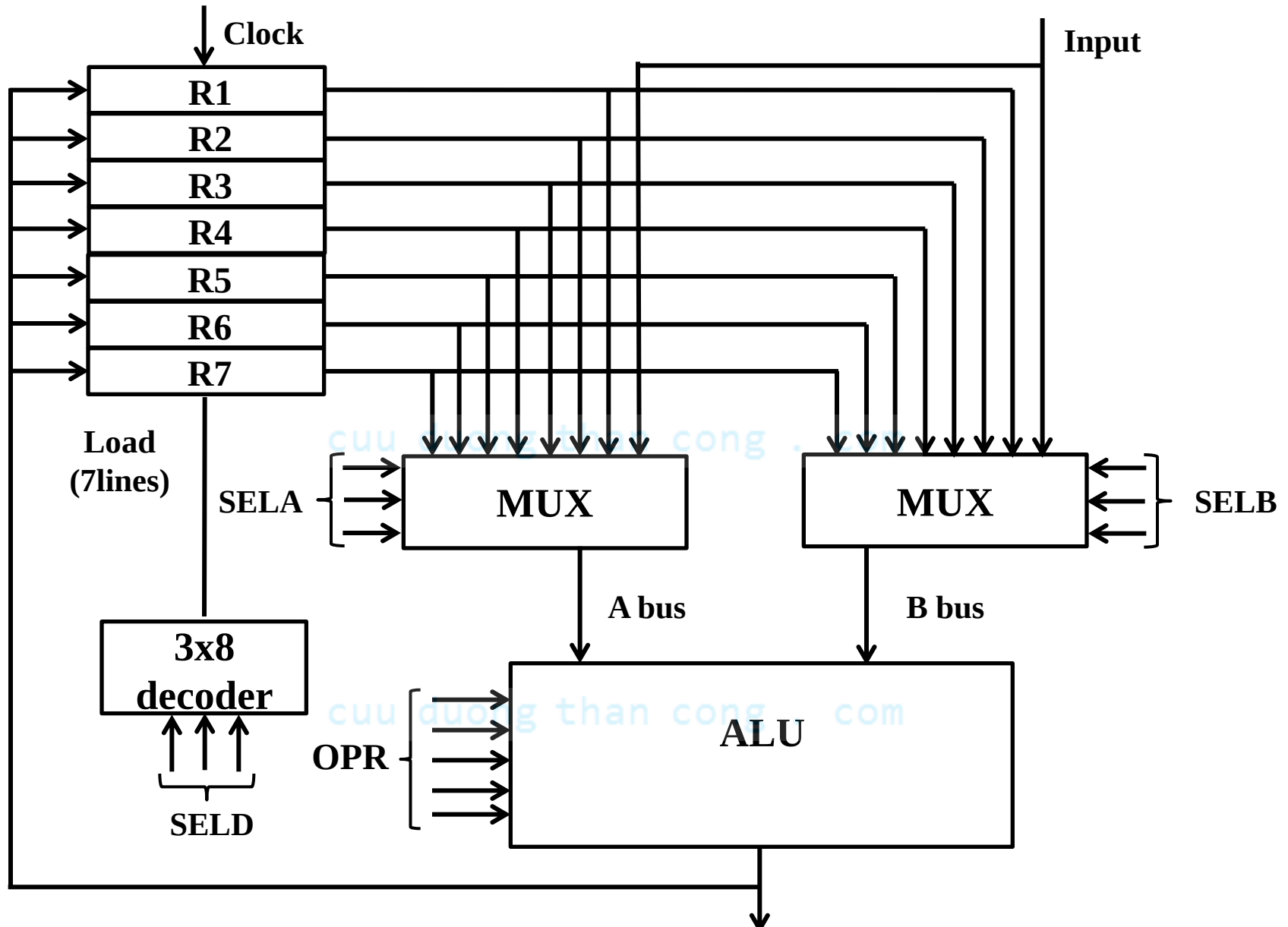


Hoạt động của bus cho chu kỳ tìm nạp lệnh

Việc tìm nạp lệnh được thực hiện theo các bước sau:

- Nội dung của **PC** được đặt lên bus địa chỉ.
- Tín hiệu điều khiển **READ** được xác lập (chuyển sang trạng thái tích cực).
- Dữ liệu (**OPCODE** của lệnh) được đọc từ RAM và đưa lên bus dữ liệu.
- **Opcode** được chốt vào thanh ghi lệnh **IR** bên trong CPU.
 - + **Giải mã lệnh**: xác định các tác động(phép toán) cần có và kích thước lệnh(bao nhiêu địa chỉ).
 - + **Nhận toán hạng**: định vị trí để lấy dữ liệu toán hạng.
 - + **Thực thi**: tính giá trị kết quả hoặc trạng thái.
 - + **Lưu kết quả**: cất kết quả vào bộ nhớ hay thanh ghi để sử dụng sau.
- **PC** được tăng để chuẩn bị tìm nạp lệnh kế từ bộ nhớ.

Ví dụ minh họa cho một tổ chức các thành ghi của 1CPU đơn giản



Mô tả

CPU được thiết kế có 7 thanh ghi R1-R7, bao gồm một số phép toán được mã hóa theo bảng qui ước sau:

Mã	SELA	SELB	SELD
000	Input	Input	None
001	R1	R1	R1
010	R2	R2	R2
011	R3	R3	R3
100	R4	R4	R4
101	R5	R5	R5
110	R6	R6	R6
111	R7	R7	R7

Mã	Phép toán	Ký hiệu
00000	Di chuyển A	TSFA
00001	$A + 1$	INCA
00010	$A + B$	ADD
00101	$A - B$	SUB
00110	$A - 1$	DECA
01000	A and B	AND
01010	A or B	OR
01100	A xor B	XOR
01110	$/A$	COMA
10000	Dịch phải A	SHRA
11000	Dịch trái A	SHLA

Ví dụ

Giả sử muốn thực hiện phép toán $A + B$ và kết quả được lưu vào thanh ghi R1. Với nội dung của A và B được cất trong 2 thanh ghi R2 và R3.

Lệnh được viết:

ADD R1,R2,R3 (phụ thuộc vào tập lệnh đưa ra)

Mã lệnh được đưa đến CPU sẽ là:

- **SELA** = 010 : nội dung của R2 sẽ đưa lên A bus.
- **SELB** = 011: nội dung của R3 sẽ đưa lên B bus.
- **OPR** = 00010: ALU sẽ thực hiện phép toán cộng.
- **SELD** = 001: kết quả lưu vào R1.

⇒ Từ điều khiển “control word” có dạng:

SELA	SELB	SELD	OPR
010	011	001	00010

Quá trình thực thi của CPU:

- PC xuất các địa chỉ chứa mã lệnh (opcode) lên bus địa chỉ.
- Nội dung của opcode(**010 011 001 00010**) được đọc từ bộ nhớ chương trình đưa lên bus dữ liệu và chứa vào thanh ghi lệnh.
- Đơn vị giải mã lệnh sẽ thực hiện giải mã lệnh cần thực thi, kết hợp với đơn vị điều khiển để xác định vị trí để lấy dữ liệu các toán hạng và nơi lưu giữ kết quả.
 - + Mã lệnh cộng (**00010**) sẽ đưa vào các tín hiệu **OPR** của **ALU**.
 - + Nội dung của **R2** sẽ đưa lên **A bus** bằng cách tác động vào tín hiệu **SELA** của MUX tương ứng là **010**. Tương tự, nội dung của **R3** cũng được đọc từ **B bus** bằng cách tác động giá trị **011** vào tín hiệu **SELB**.
 - + Giá trị **001** được đưa đến các tín hiệu **SELD** của decoder để lưu kết quả cuối cùng vào R1.
- **ALU** thực thi việc tính toán kết quả và lưu vào vị trí mong muốn.

cuu duong than cong . com

Định dạng lệnh:

Mỗi một lệnh trong tập lệnh sẽ có các phần định dạng như sau:

- Vùng mã lệnh (opcode): cho biết tác vụ nào sẽ được thực hiện.
- Vùng địa chỉ: chỉ địa chỉ bộ nhớ hoặc thanh ghi của CPU.
- Vùng cách định địa chỉ: chỉ cách xác định giá trị của toán hạng hoặc địa chỉ của toán hạng.

Ví dụ:

MOV	R1, R2
ADD	R1,R2,R3
MUL	X
PUSH	A



Vùng opcode

Vùng địa chỉ hoặc cách định địa chỉ

Các phương pháp định địa chỉ trong lệnh sẽ phụ thuộc vào tổ chức nội của CPU dẫn đến mỗi lệnh có nhiều hay ít hoặc không có địa chỉ. (Xem các ví dụ trong giáo trình)

Các phương pháp định địa chỉ

- Định địa chỉ hiệu ngầm(implied addressing mode) hay hàm ý.
- Định địa chỉ tức thời.
- Định địa chỉ thanh ghi.
- Định địa chỉ gián tiếp qua thanh ghi.
- Định địa chỉ trực tiếp.
- Định địa chỉ gián tiếp.
- Định địa chỉ tương đối.

Với mỗi phương pháp khác nhau sẽ ảnh hưởng đến độ dài ngắn của chương trình, số bit cần sử dụng và thời gian thực thi của từng lệnh.(Tham khảo chi tiết trong giáo trình)

cuu duong than cong . com

Bộ nhớ

ROM (read only memory)

Là bộ nhớ chỉ đọc, không thể đưa dữ liệu thêm vào.

ROM được chia làm nhiều loại:

- PROM (Programmable ROM): được lập trình tại nhà máy.
- EPROM (Erasable PROM): ROM lập trình (thiết bị lập trình) và xóa (tia cực tím) được.
- EEPROM (Electrical EPROM): ROM lập trình và xóa được bằng điện.

RAM (Random Access Memory)

Là bộ nhớ truy xuất ngẫu nhiên.

RAM chỉ lưu dữ liệu khi được cung cấp điện.

RAM được chia làm 2 loại cơ bản là SRAM và DRAM.

- SRAM (Static RAM): RAM tĩnh được lưu trữ bằng các FF.
- DRAM (Dynamic RAM): RAM động, được lưu trữ bằng điện tích trong các tụ điện, cơ chế hoạt động phức tạp hơn SRAM và cần có cơ chế làm tươi (refresh).

Thiết bị ngoại vi (Input/Output – IO)

Các thiết bị nhập và xuất được gọi chung là thiết bị ngoại vi hay I/O.

Các thiết bị nhập	Các thiết bị xuất
Bàn phím	Máy đọc phiếu, máy đọc băng giấy
Các thiết bị nhập loại quang học -Máy đọc thẻ -Máy đọc băng giấy -Máy đọc mã vạch (bar code) -Máy số hóa dữ liệu (Digitizer)	Màn hình CRT Máy in Máy vẽ Thiết bị xuất analog Tiếng nói
Máy nhập loại từ - Máy đọc thẻ từ	
Thiết bị nhập theo màn hình -Màn hình nhạy với chạm -Bút quang (light pen) -Chuột (mouse)	
Thiết bị nhập analog	

BUS

Các khối cơ bản trong hệ vi xử lý được nối kết lại với nhau qua nhóm các dây dẫn điện được gọi là bus. Trong hệ vi xử lý thường có 3 bus: data bus, address bus và control bus.

- **Data bus:** là đường truyền 2 chiều để trao đổi thông tin giữa CPU và các thành phần khác của hệ thống.

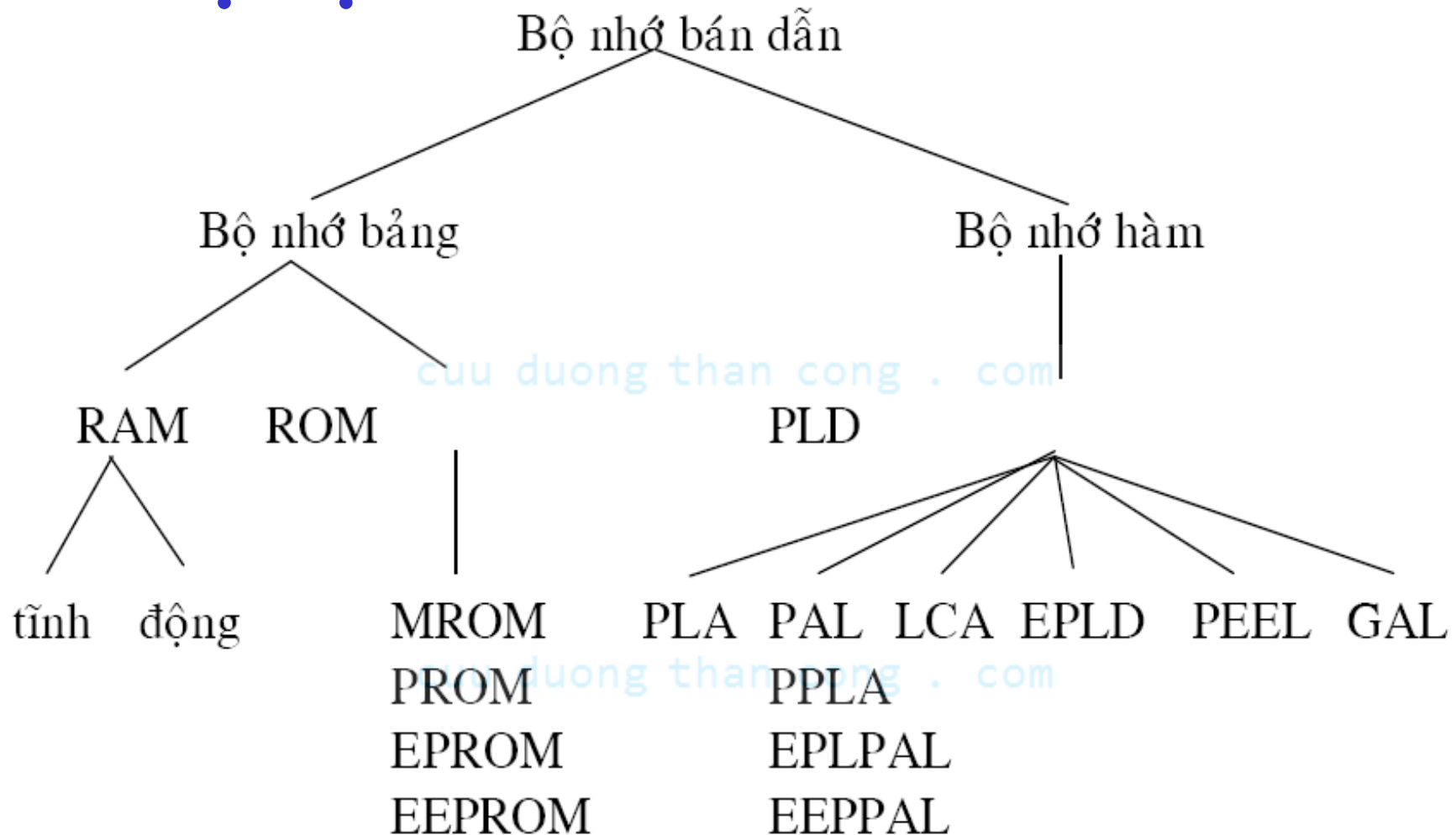
- **Address bus:** dùng để chỉ vị trí ô nhớ, địa chỉ của thiết bị ngoại vi mà CPU muốn truyền thông tin (dữ liệu hay lệnh).

- **Control bus:** là các đường truyền tín hiệu mà CPU điều khiển các hoạt động của các đơn vị khác nhau trong hệ thống.

cuu duong than cong . com

II. NHẮC LẠI VỀ BỘ NHỚ

Các loại bộ nhớ



Các loại ROM, RAM thông dụng

1. EPROM: thường được ký hiệu là 27XX hay 28XX (XX là dung lượng tính theo Kbit), mỗi ô nhớ chứa 8 bit dữ liệu.

Ví dụ: - 2716 là EPROM 16Kbit tương đương với 2KB.

- 2764 là EPROM 64Kbit tương đương với 8KB.

- 27512 là EPROM 512Kbit tương đương với 64KB.

Đối với EEPROM thường được ký hiệu là 28XX.

2. SRAM thường được ký hiệu là 61XX hay 62XX.

Ví dụ: - 6116 là SRAM 2KB.

- 6264 là SRAM 8KB.

Một số thuật ngữ khi nói đến bộ nhớ

Tế bào nhớ (memory cell)

Bit là đơn vị cơ bản của thông tin nhị phân (bit=**b**inary **d**igit). Nó lấy giá trị 0 hoặc 1. Trong các máy tính số thì bit được truyền đi qua kết nối điện và được lưu trữ trong tế bào nhớ.

Byte là đơn vị lớn hơn gồm 8 bit.

Word là nhóm gồm nhiều byte (tùy theo quy ước có thể số byte là 1, 2, 4, 8, ..).

Theo quy ước thông thường thì word gồm 2 byte (hay 16 bit) và word dài gồm 4 byte (32 bit).

Dung lượng (kích thước) : là đại lượng cho biết khả năng bao nhiêu bit bộ nhớ có thể lưu trữ được. ($2^n \times m$)

Thời gian truy xuất (Access Time) : là thời gian từ khi bộ nhớ nhận địa chỉ cho đến khi dữ liệu được xuất ra ổn định.

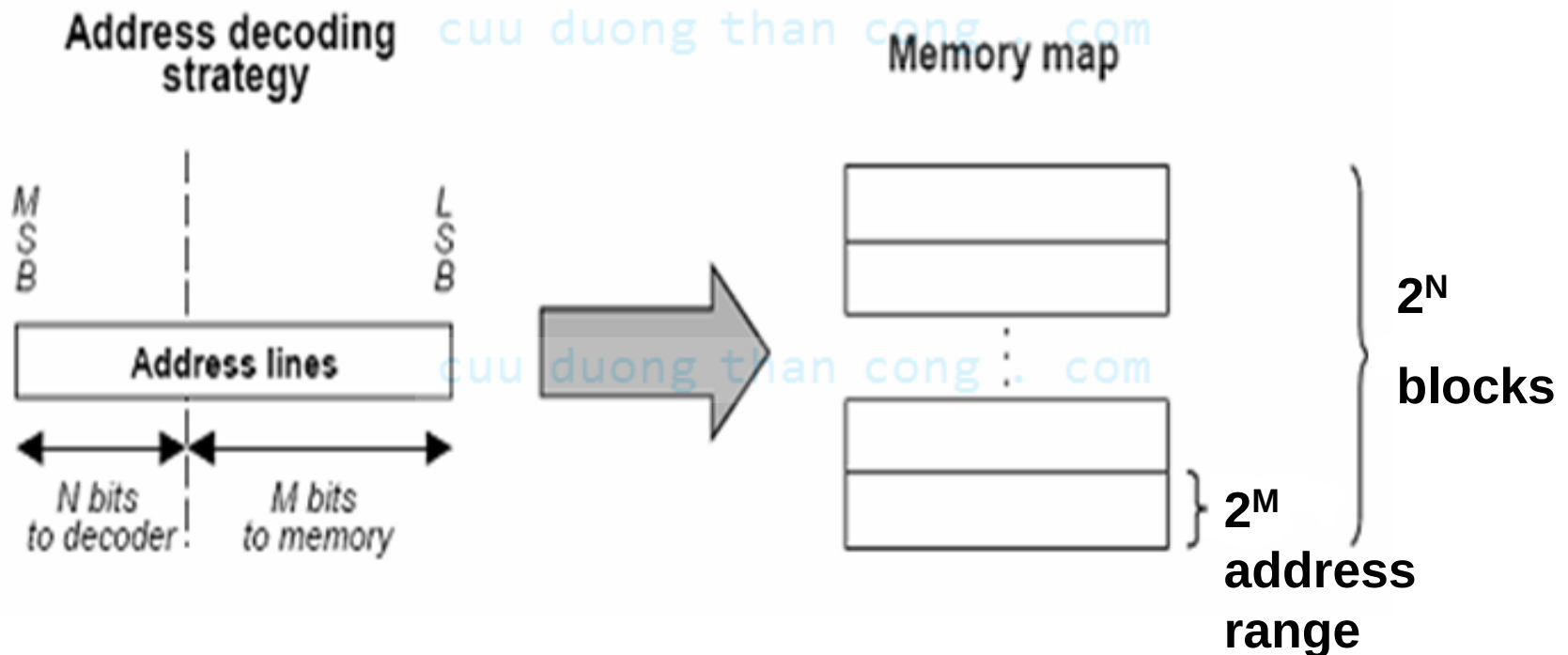
Chế độ đọc/ghi (Read/Write)

Read: ở chế độ này, từ nhị phân vốn được lưu trữ tại địa chỉ nào đó của bộ nhớ được đưa ra ở bus dữ liệu.

Write: từ đặt ở bus dữ liệu sẽ được đưa vào trong một vị trí nào đó của ô nhớ.

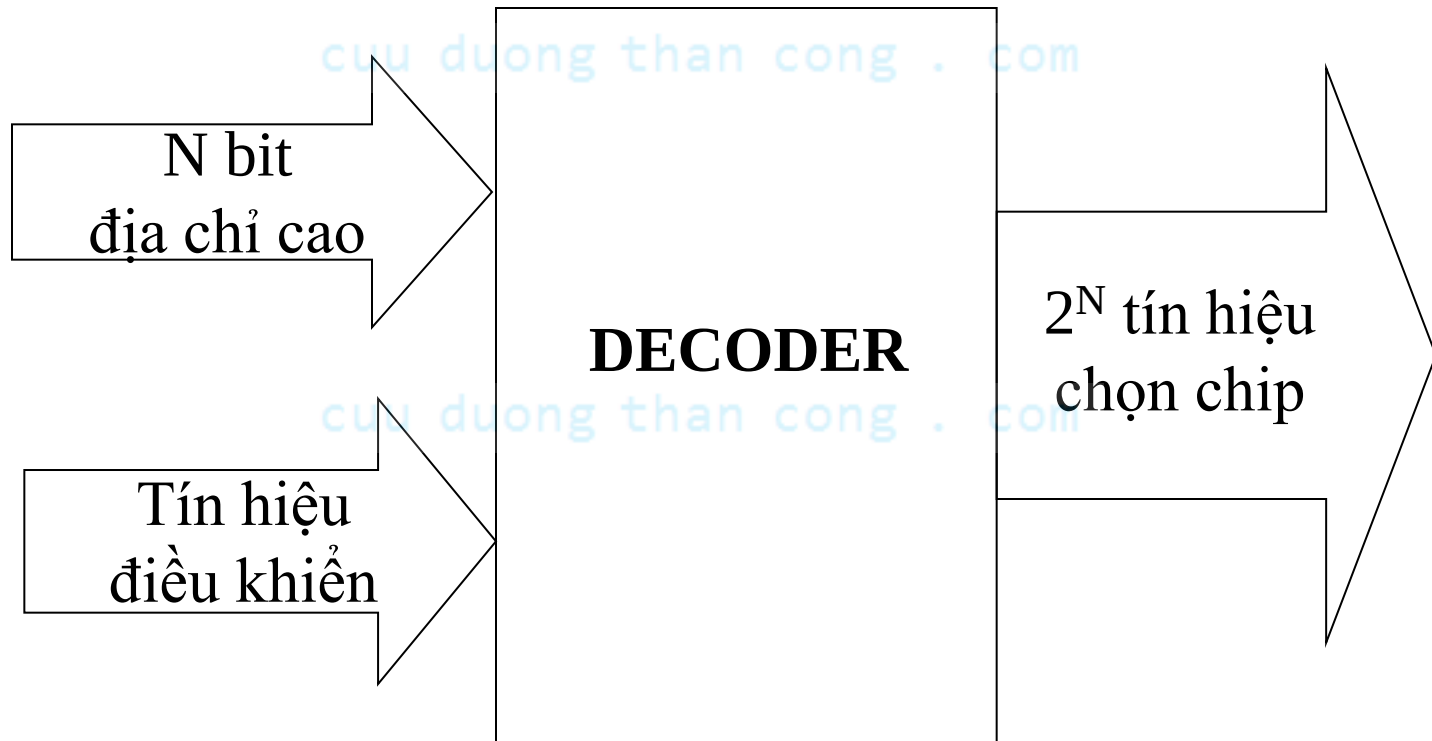
III. GIẢI MÃ ĐỊA CHỈ

- Thông thường việc quản lý bộ nhớ được chia thành từng vùng xác định nhờ địa chỉ đầu và địa chỉ cuối của mỗi vùng. Sơ đồ biểu diễn địa chỉ của các vùng nhớ được gọi là bản đồ bộ nhớ.
- Bus địa chỉ từ CPU có thể được chia làm 2 phần: N bit cao quản lý 2^N vùng nhớ, M bit thấp quản lý 2^M ô nhớ trong mỗi vùng nhớ.



Nguyên tắc giải mã

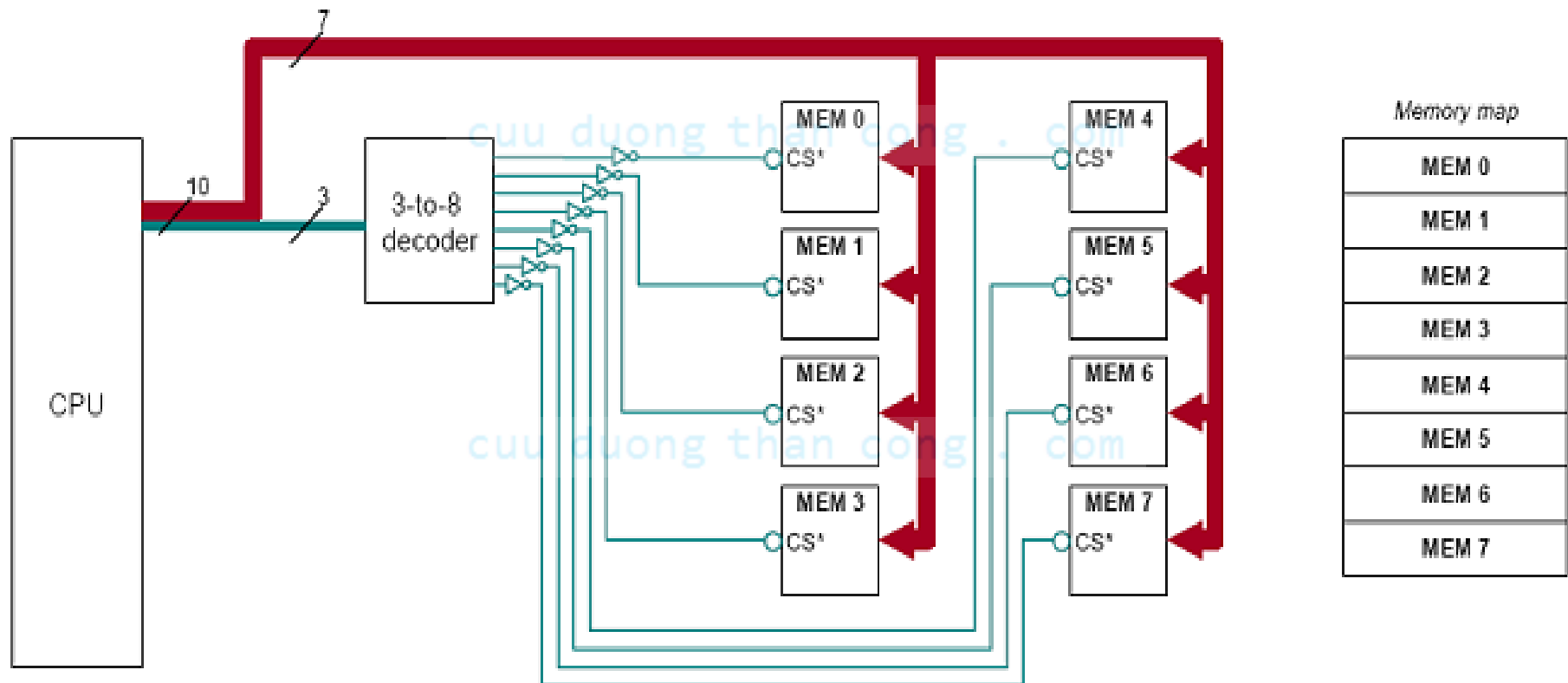
Thông thường N bit cao được đưa qua mạch Decoder $N \rightarrow 2^N$ để tạo ra 2^N tín hiệu cho phép 2^N vùng nhớ. Tín hiệu cho phép vùng nhớ hoạt động còn gọi là 2^N tín hiệu chọn chip.



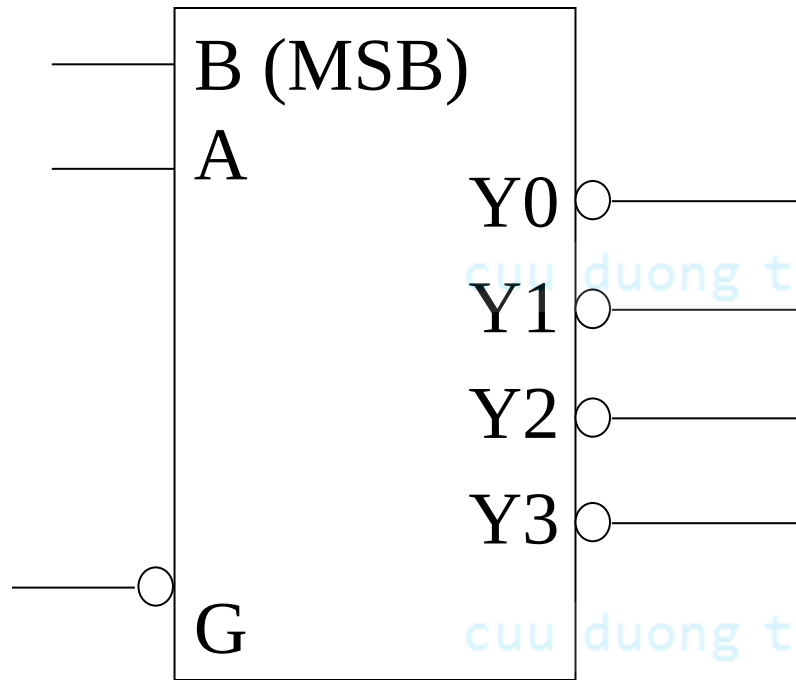
Ví dụ:

Một CPU có 10 đường địa chỉ (A0-A9) sẽ cấp phát được 2^{10} địa chỉ tương đương với 2^{10} (1024) ô nhớ hay nói cách khác là có thể quản lý tối đa 1KB bộ nhớ.

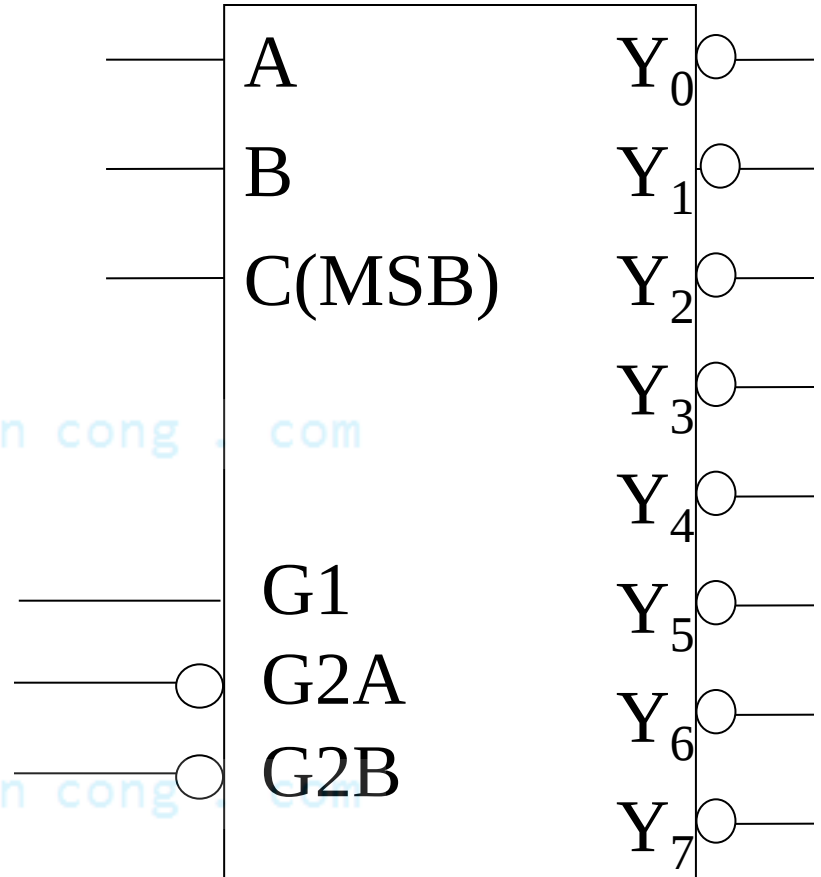
Nếu có 8 bộ nhớ, mỗi bộ nhớ có dung lượng là 128 byte (2^7) thì mạch giải mã địa chỉ sẽ được thực hiện như sau:



Giới thiệu một số IC giải mã thường gặp

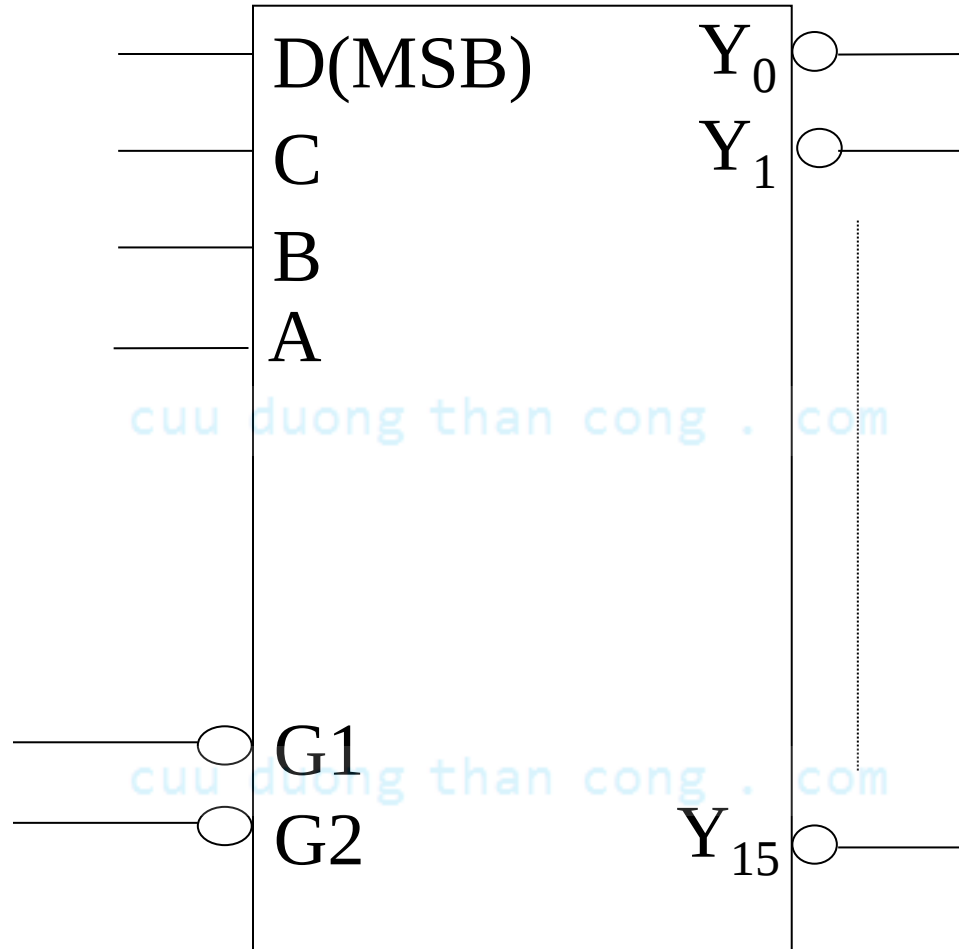


IC74139: DECODER 2→4



IC 74138: DECODER 3→8

IC 74154: DECODER 4→16



Ví dụ

*Thiết kế mạch giải mã địa chỉ cho một hệ vi xử lý gồm 2ROM 2764 và 2RAM 6264 (giả sử CPU có 16 đường địa chỉ).
Biết rằng các tín hiệu chọn chip đều tích cực ở mức thấp.*

Hướng dẫn

ROM 2764 có dung lượng: 8K

RAM 6264 có dung lượng: 8K

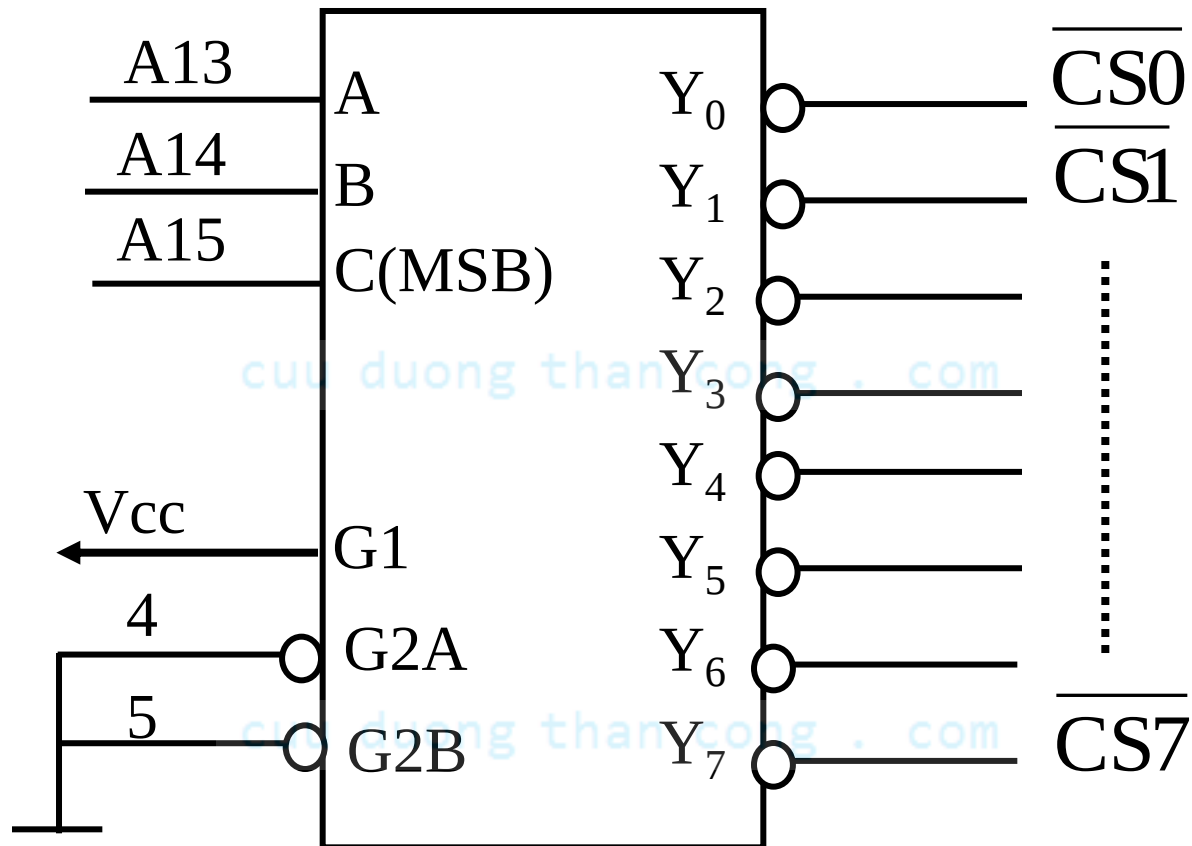
⇒ Cần phân vùng địa chỉ $8K = 2^{13}$

⇒ Cần 13 đường địa chỉ thấp A0 - A12

⇒ Còn lại 3 đường địa chỉ cao A13-A15 có nhiệm vụ chọn 8 vùng nhớ.

⇒ Dùng Decoder 3 → 8 (74138) để giải mã.

Mạch giải mã



Bảng địa chỉ

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	$\overline{CS}$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\overline{CS}_0$
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	$\overline{CS}_1$
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	$\overline{CS}_7$
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Tầm địa chỉ

$\overline{CS}_0$: 0000H – 1FFFFH

$\overline{CS}_2$: 4000H – 5FFFFH

$\overline{CS}_1$: 2000H – 3FFFFH

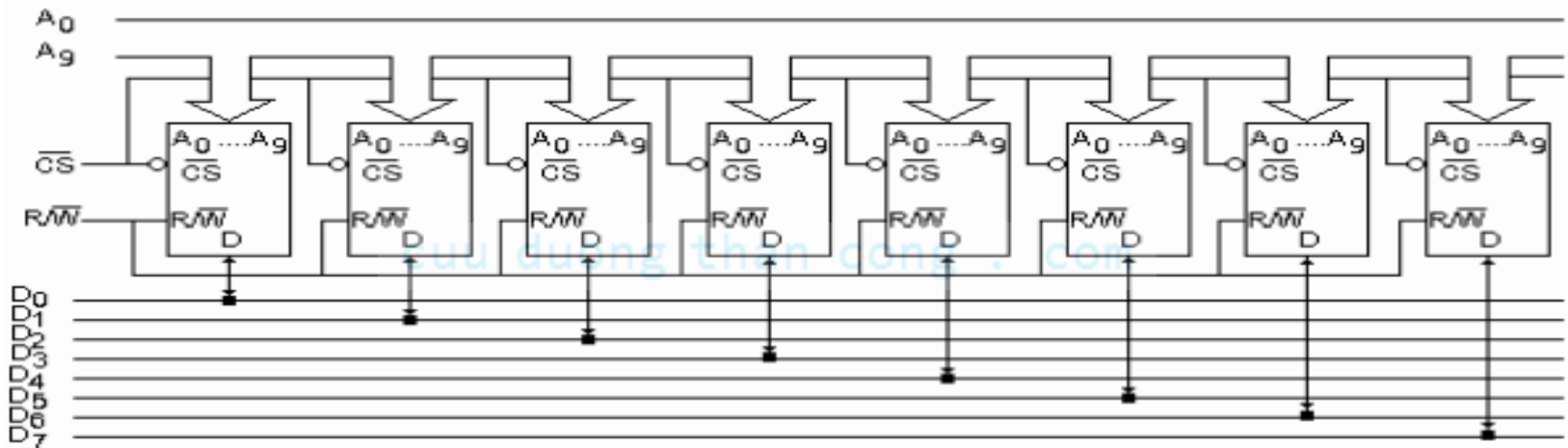
$\overline{CS}_7$: E000H – FFFFFH

Mở rộng dung lượng bộ nhớ

* **Mở rộng độ dài bit:** trong trường hợp số vị trí nhớ đủ cho yêu cầu nhưng số bit dữ liệu cho mỗi ô nhớ thì không đủ thì ta cần mở rộng thêm số bit.

Ví dụ: Mở rộng bộ nhớ ROM có kích thước 1Kx1 lên 1Kx8.

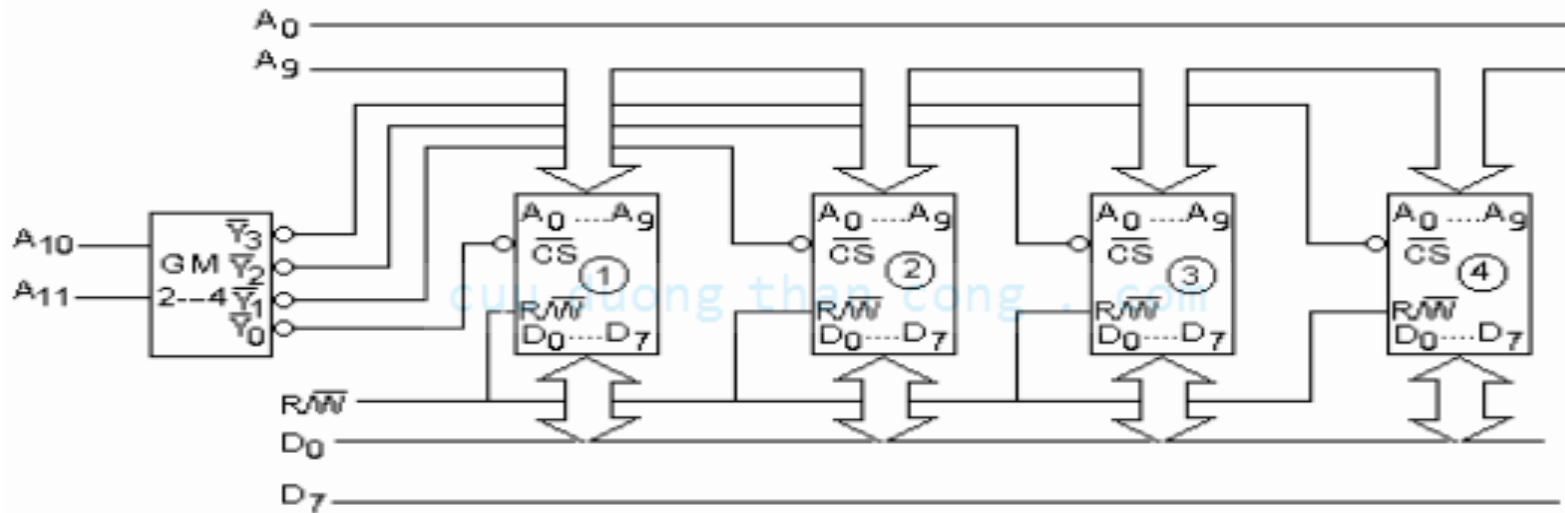
HD: Sẽ cần dùng 8 ROM 1Kx1, các ROM này sẽ được nối chung 1 bus địa chỉ và các đường tín hiệu điều khiển, mỗi IC sẽ quản lý 1 bit ra vận hành cùng 1 lúc để cho ra 1 từ nhớ 8bit.



*** Mở rộng vị trí nhớ:** trong trường hợp số bit cho mỗi ô nhớ đủ theo yêu cầu nhưng không đủ số ô nhớ.

Ví dụ: Mở rộng bộ nhớ ROM có kích thước 1Kx8 lên 4Kx8.

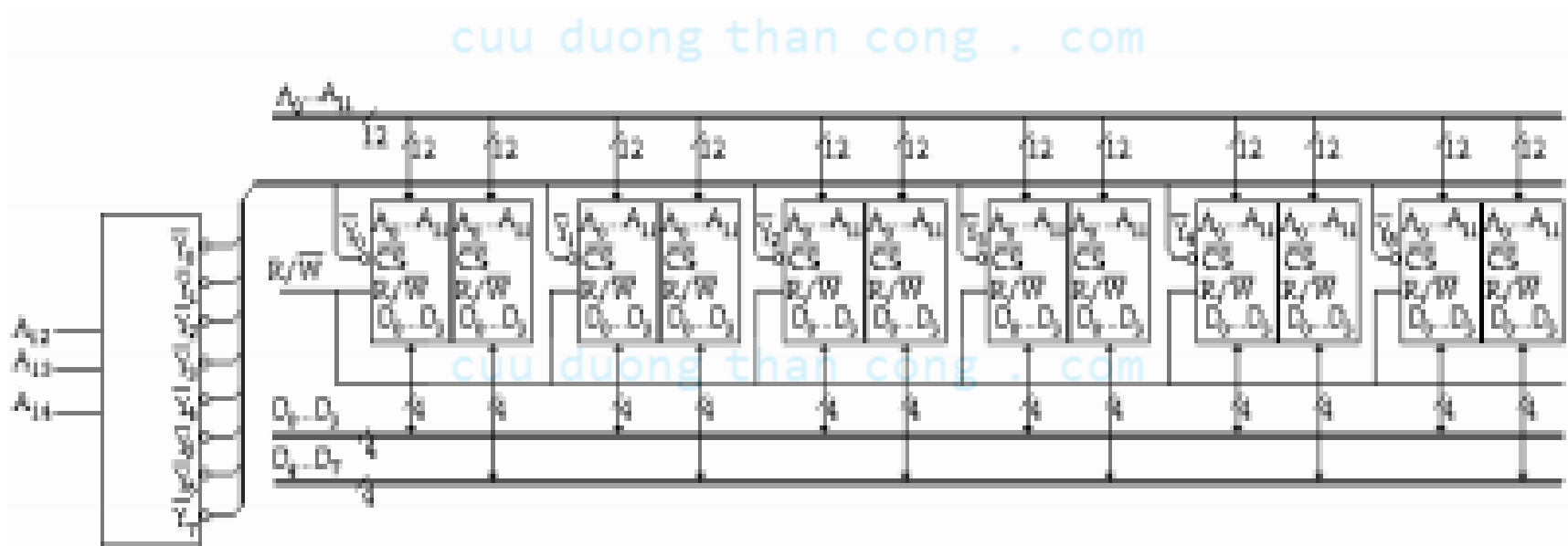
HD: Sẽ cần dùng 4 ROM 1Kx8 để tạo ra 1 vùng nhớ 4K. Cần 1 mạch giải mã 2 sang 4 đưa đến các chân CS của các ROM để truy xuất 4K dữ liệu, do đó địa chỉ của các ROM này sẽ khác nhau.



*** Mở rộng dung lượng nhớ:** trong trường hợp cả vị trí nhớ và độ dài từ của các ROM đều không đủ cho yêu cầu thiết kế.

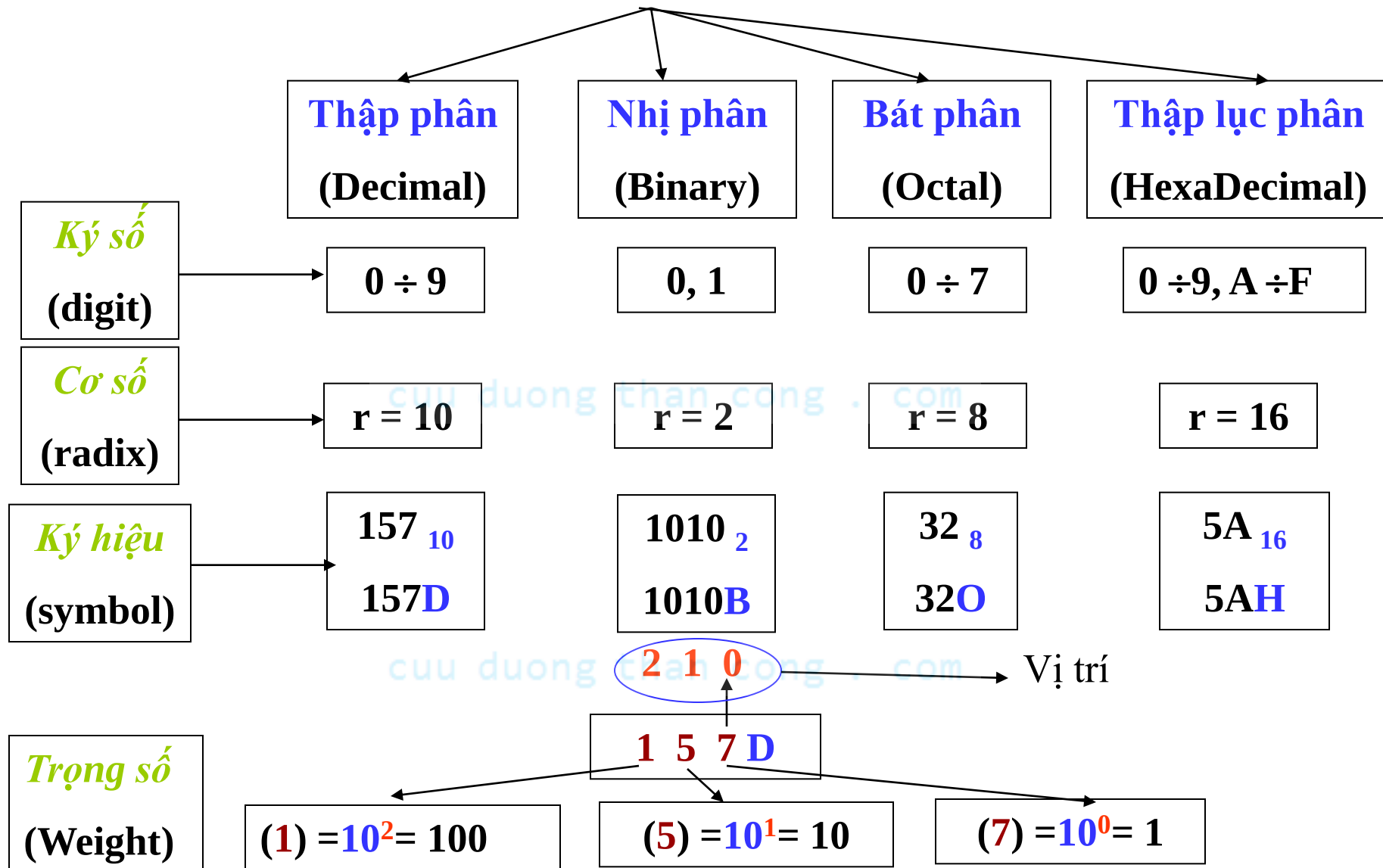
Ví dụ: Mở rộng bộ nhớ ROM có kích thước 4Kx4 lên 24Kx8.

HD: Sẽ cần dùng 6 ROM 4Kx4 mắc song song, mỗi cặp có chung bus địa chỉ và được chọn bởi một mạch giải mã 3 sang 8, trường hợp này chỉ cần dùng 6 ngõ ra của mạch giải mã.



IV. NHẮC LẠI MỘT SỐ KIẾN THỨC VỀ ĐẠI SỐ BOOLE

1. Các hệ thống số đếm



Giá trị (Value) = $\sum$ Ký số * trọng số

157₁₀
157D

$$= 1 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0$$
$$= 157$$

1010₂
1010B

$$= 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$
$$= 10$$

32₈
32O

$$= 3 \cdot 8^1 + 2 \cdot 8^0$$
$$= 26$$

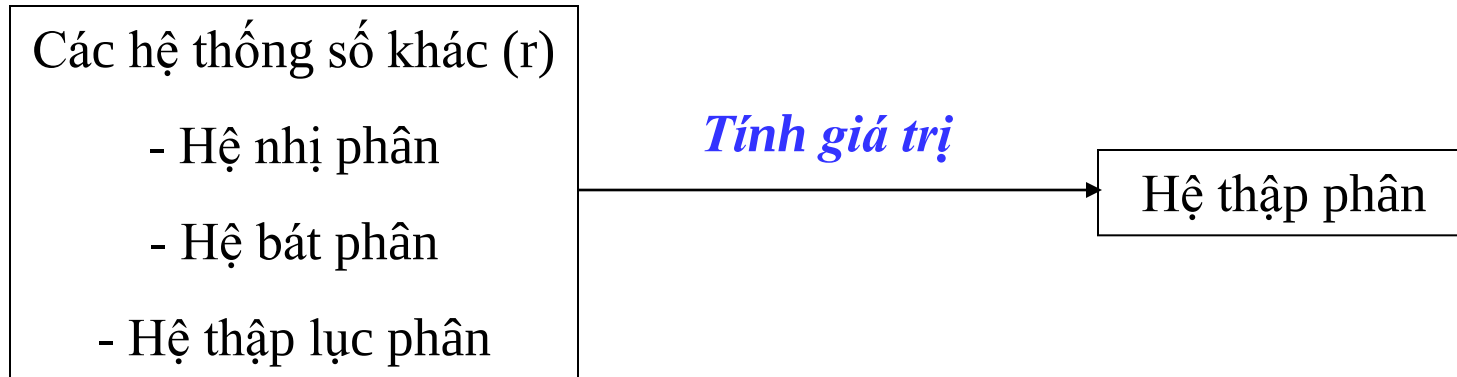
5A₁₆
5AH

$$= 5 \cdot 16^1 + 10 \cdot 16^0$$
$$= 90$$

Giá trị (Value) của một số chính là chuyển đổi số đó về hệ thập phân.

CHUYỂN ĐỔI CƠ SỐ

(a)



(b)

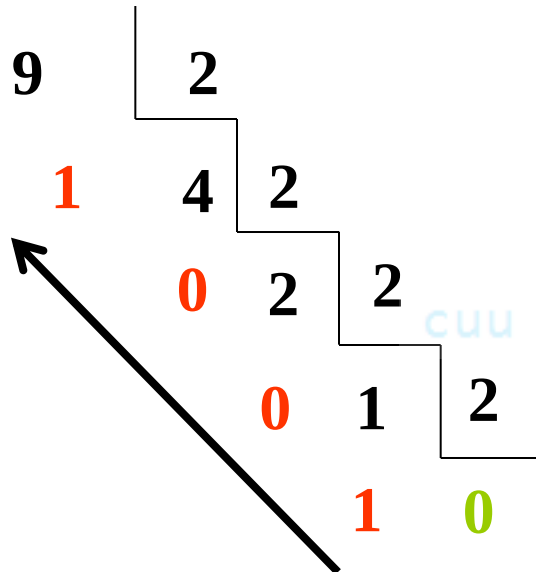


+ ***Phần nguyên***: chia liên tiếp cho r đến khi có kết quả của phép chia là 0 rồi lấy các số dư theo thứ tự từ dưới lên.

+ ***Phần lẻ***: nhân liên tiếp với r , sau mỗi lần nhân lấy đi số phần nguyên, tiếp tục cho đến khi kết quả là 0 hoặc đến khi đạt độ chính xác cần thiết. Kết quả là lấy các số nguyên đi theo thứ tự từ trên xuống.

Ví dụ : đổi số 9.25D sang hệ nhị phân

Phần nguyên



Phần lẻ

$0,25 \times 2 = 0,5 \rightarrow$ lấy bit **0**

$0,5 \times 2 = 1,0 \rightarrow$ lấy bit **1**

Kết quả:

9.25 D = 1001.01 B

(c) Chuyển đổi giữa các hệ thống cơ số khác

- Một số bát phân biểu diễn bởi 3 bit nhị phân
- Một số thập lục phân biểu diễn bởi 4 bit nhị phân

Oct	Bin
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Hex	Bin
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

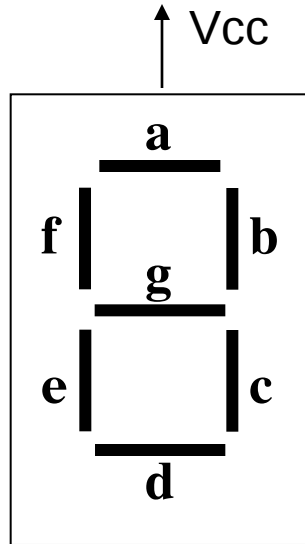
Hex	Bin	Oct
8	1000	10
9	1001	11
A	1010	12
B	1011	13
C	1100	14
D	1101	15
E	1110	16
F	1111	17

2. Các mã nhị phân

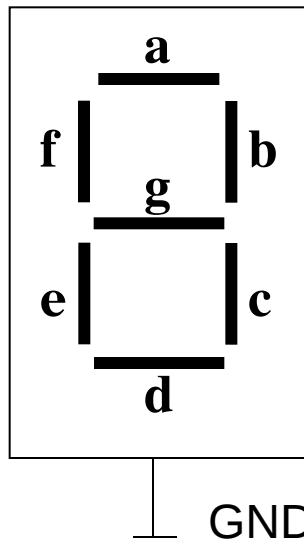
Mã nhị phân cho số thập phân (BCD)

Số thập phân	BCD (8 4 2 1)	BCD (2 4 2 1)	BCD quá 3	Mã 1 trong 10
0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0 0 0 0 0 0 0 1
1	0 0 0 1	0 0 0 1	0 1 0 0	0 0 0 0 0 0 0 0 0 1 0
2	0 0 1 0	0 0 1 0	0 1 0 1	0 0 0 0 0 0 0 0 1 0 0
3	0 0 1 1	0 0 1 1	0 1 1 0	0 0 0 0 0 0 0 1 0 0 0
4	0 1 0 0	0 1 0 0	0 1 1 1	0 0 0 0 0 1 0 0 0 0 0
5	0 1 0 1	1 0 1 1	1 0 0 0	0 0 0 0 1 0 0 0 0 0 0
6	0 1 1 0	1 1 0 0	1 0 0 1	0 0 0 1 0 0 0 0 0 0 0
7	0 1 1 1	1 1 0 1	1 0 1 0	0 0 1 0 0 0 0 0 0 0 0
8	1 0 0 0	1 1 1 0	1 0 1 1	0 1 0 0 0 0 0 0 0 0 0
9	1 0 0 1	1 1 1 1	1 1 0 0	1 0 0 0 0 0 0 0 0 0 0

Mã led 7 đoạn



Anode chung



Cathode chung

Giá trị	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Bảng mã led 7 đoạn
loại cathode chung

Mã ký tự ASCII:

		(Cột) $b_6 b_5 b_4$							
(Hàng)		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
$b_3 b_2 b_1 b_0$	Hex	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	”	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	\$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	,	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	A	LF	SUB	*	:	J	Z	j	z
1 0 1 1	B	VT	ESC	+	;	K	[	k	{
1 1 0 0	C	FF	FS	,	<	L	\	l	
1 1 0 1	D	CR	GS	-	=	M	]	m	}
1 1 1 0	E	SO	RS	.	>	N	^	n	~
1 1 1 1	F	SI	US	/	?	O	_	o	DEL ⁴⁷

3. Các phép toán luận lý

a	NOT a
0	1
1	0

a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1

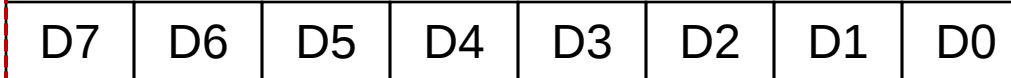
a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

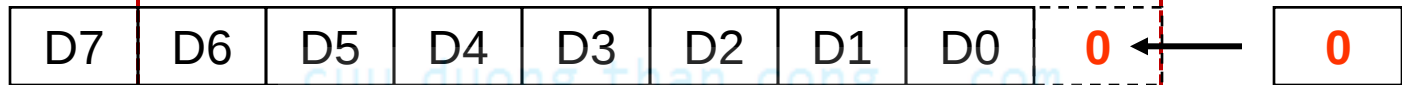
Phép Dịch - Shifting

Thực hiện di chuyển các bit trong 1 byte dữ liệu sang trái hoặc phải 1 bit

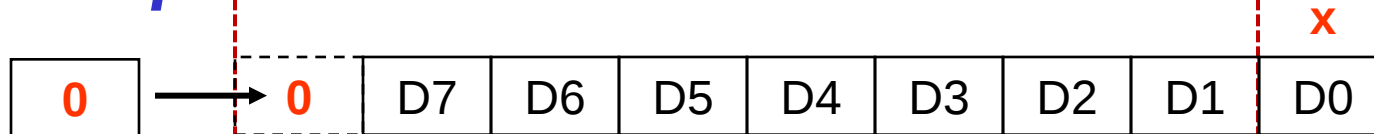
Data



Left Shift:



Right Shift:



Ví dụ:

11011100 **Left Shift** → 10111000

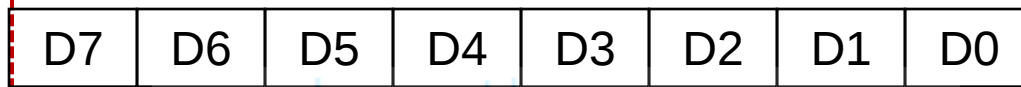
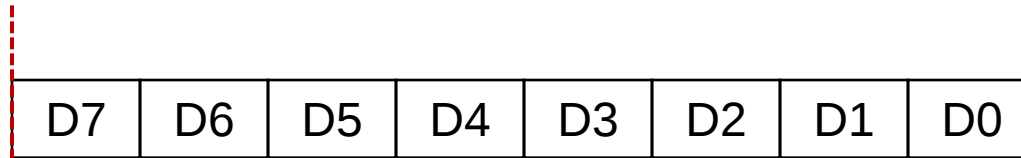
11011100 **Right Shift** → 01101110

Phép Xoay - Rotating

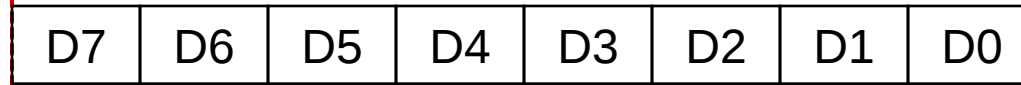
Thực hiện xoay các bit trong 1 byte dữ liệu 1 vòng sang trái hoặc phải 1 bit

Data

Left rotate



Right rotate



Ví dụ:

11011100 *Left rotate* → 10111001

11011100 *Right rotate* → 01101110

Nhận xét:

- **Phép AND và OR có thể dùng để xóa (clear) bit về 0 hoặc đặt (set) bit lên 1.**
- **Phép XOR có thể dùng để lấy bù của dữ liệu hoặc là chính nó.**
- **Phép dịch trái 1bit của dữ liệu đồng nghĩa với việc nhân với 2.**
- **Phép dịch phải 1 bit của dữ liệu tương tự với việc chia cho 2.**