ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG CAO ĐẲNG LÝ TỰ TRỌNG THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ

★★★

GIÁO TRÌNH

MÔN HỌC: KỸ THUẬT SỐ

NGÀNH: ĐIỆN CÔNG NGHIỆP; CÔNG NGHỆ KỸ THUẬT ĐIỆN, ĐIỆN TỬ; CÔNG NGHỆ KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA; CƠ ĐIỆN TỬ

TRÌNH ĐỘ: CAO ĐẲNG

Ban hành kèm theo Quyết định số: 1676 /QĐ-LTT-ĐT ngày 31 tháng 12 năm 2020 của Hiệu trưởng Trường Cao đẳng Lý Tự Trọng Thành phố Hồ Chí Minh

Thành phố Hồ Chí Minh - Năm 2020

TUYÊN BỐ BẢN QUYỀN

LỜI GIỚI THIỆU

Hiện nay, chúng ta đang ở trong giai đoạn đầu của cuộc cách mạng công nghiệp lần thứ 4.0 (Industry 4.0) – nền công nghiệp tập trung vào công nghệ kỹ thuật số. Với sự trợ giúp kết nối thông qua Internet vạn vật, truy cập dữ liệu thời gian thực và các hệ thống vật lý không gian mạng, nền công nghiệp 4.0 cung cấp một cách tiếp cận liên kết và toàn diện hơn cho các nền tảng sản xuất. Công nghiệp 4.0 trao quyền cho các chủ doanh nghiệp kiểm soát và hiểu rõ hơn mọi khía cạnh hoạt động của họ và cho phép họ tận dụng dữ liệu tức thời để tăng năng suất, cải thiện quy trình và thúc đẩy tăng trưởng. Công nghiệp 4.0 cho phép các nhà máy thông minh, sản phẩm thông minh và chuỗi cung ứng cũng thông minh, và làm cho các hệ thống sản xuất và dịch vụ trở nên linh hoạt và đáp ứng khách hàng hơn. Tất cả các lợi ích trên của nền công nghiệp 4.0 đều dựa vào công nghệ kỹ thuật số.

Kỹ thuật số trở thành nền tảng cơ bản cho mọi công nghệ phát triển trong tương lai.

Giáo trình Kỹ thuật số này được biên soạn nhằm cung cấp các kiến thức cơ sở cho sinh viên tất cả các chuyên ngành của khoa Điện - Điện tử, trường Cao đẳng Lý tự Trọng Tp.HCM để từ đó, sinh viên có thể tiếp cận được các kiến thức, máy móc kỹ thuật số hiện đại một cách dễ dàng nhất.

Nội dung giáo trình gồm 5 chương:

Bài mở đầu: *Tổng quan về kỹ thuật số*

Chương 1*: Các hệ thống số*

Chương 2: *Đại số Boolean và thiết kế mạch logic*

Chương 3: *Mạch tổ hợp*

Chương 4: *Mạch tuần tự*

Chương 5: *Các mạch số khác*

Trong quá trình biên soạn giáo trình cũng không tránh khỏi những sai sót. Tác giả rất mong nhận được ý kiến đóng góp của các bạn đọc. Các ý kiến đóng góp xin gởi về:

Trần Nguyên Bảo Trân

Khoa Điện-Điện tử

Trường Cao đẳng Lý Tự Trọng TP.HCM

ĐT: 0908.225.475 - Địa chỉ Email: trannguyenbaotran@lttc.edu.vn

Ngày 22 tháng 11 năm 2020

MỤC LỤC

GIÁO TRÌNH MÔN HỌC

Tên môn học: KỸ THUẬT SỐ

Mã môn học: DDC103

Thời gian thực hiện học phần: 30 giờ; (Lý thuyết: 28 giờ; Thực hnh, thí nghiệm, thảo luận, bi tập: .... giờ; Kiểm tra: 2 giờ)

Vị trí, tính chất, ý nghĩa và vai trò của môn học:

- Vị trí: Môn học này phải học sau các môn học Mạch điện, Điện tử cơ bản.

- Tính chất: Đây là môn học chuyên ngành, thuộc các môn học đào tạo nghề bắt buộc, là môn học nền tảng để học các môn Vi xử lý, Lập trình PLC.

- Ý nghĩa và vai trò của môn học: Kỹ thuật số là môn học lý thuyết chuyên ngành, thuộc các môn học đào tạo nghề bắt buộc của ngành Điện công nghiệp; Điện – Điện Tử, Cơ Điện Tử, Tự động hóa ... Người học có khả năng thiết lập được các mạch logic để điều khiển các thiết bị điện tử. Người học có khả năng đọc, phân tích và chẩn đón hư hỏng của các mạch điện tử số.

Mục tiêu môn học:

- Về kiến thức: + Phân tích nguyên lý hoạt động của mạch số.
  + Thiết kế được mạch số tổ hợp và mạch đếm tuần tự.
- Về kỹ năng: + Tính toán, thiết kế mạch số tổ hợp, mạch đếm tuần tự
- Về năng lực tự chủ và trách nhiệm:

+ Sinh viên nghiêm túc, tích cực tham gia đóng góp xây dựng bài trên lớp. Tự giác tìm hiểu bài trước khi đến lớp.

+ Rèn luyện tính cẩn thận, tỉ mỉ trong thiết kế, phân tích mạch điện.

+ Phối hợp với tinh thần thi đua trong việc học nhóm, tự giác làm bài tập cuối mỗi chương.

Nội dung của môn học/mô đun:

BÀI MỞ ĐẦU: TỔNG QUAN VỀ KỸ THUẬT SỐ

Giới thiệu: *Trong chương này giới thiệu các khái niệm cơ bản về tín hiệu, tín hiệu xung, tín hiệu số.*

Mục tiêu:

- *Hiểu các khái niệm cơ bản về tín hiệu và các dạng tín hiệu.;*

Nội dung chính:

1. Tín hiệu:

Tín hiệu là một đại lượng vật lý chứa đựng thông tin hay dữ liệu có thể truyền đi xa và tách thông tin ra được. Hầu hết các tín hiệu đáng quan tâm đều ở dạng các hàm số, các phân bố hay các quá trình thay đổi ngẫu nhiên của thời gian hoặc vị trí. Trong các lĩnh vực kỹ thuật, các loại tín hiệu thường dùng là: quang, điện, khí nén, thủy lực và âm thanh.

Các tham số sau đây thường được dùng trực tiếp, gián tiếp hay kết hợp để biểu thị nội dung thông tin:

+ Biên độ (điện áp, dòng..)

+ Tần số, nhịp xung, độ rộng xung, sườn xung

+ Pha, vị trí xung

Có thể phân loại tín hiệu dựa theo tập hợp giá trị của tham số thông tin hoặc dựa theo diễn biến thời gian thành những dạng sau:

+ Tín hiệu tương tự: Tham số thông tin có thể có một giá trị bất kỳ trong một khoảng nào đó.

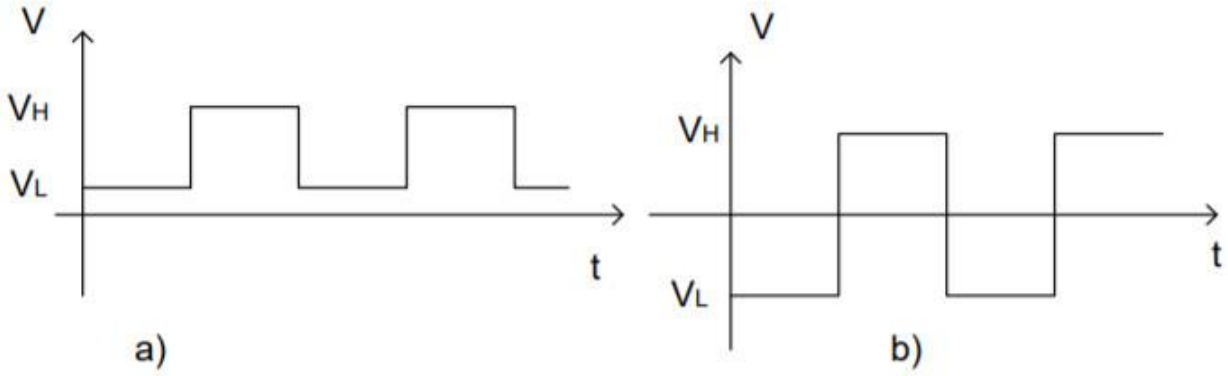
+ Tín hiệu rời rạc: Tham số thông tin chỉ có thể có một số giá trị (rời rạc) nhất định.

+ Tín hiệu liên tục: Tín hiệu có ý nghĩa tại bất kỳ thời điểm nào trong một khoảng thời gian quan tâm. Nói theo nghĩa toán học, một tín hiệu liên tục là một hàm liên tục của biến thời gian trong một khoảng xác định.

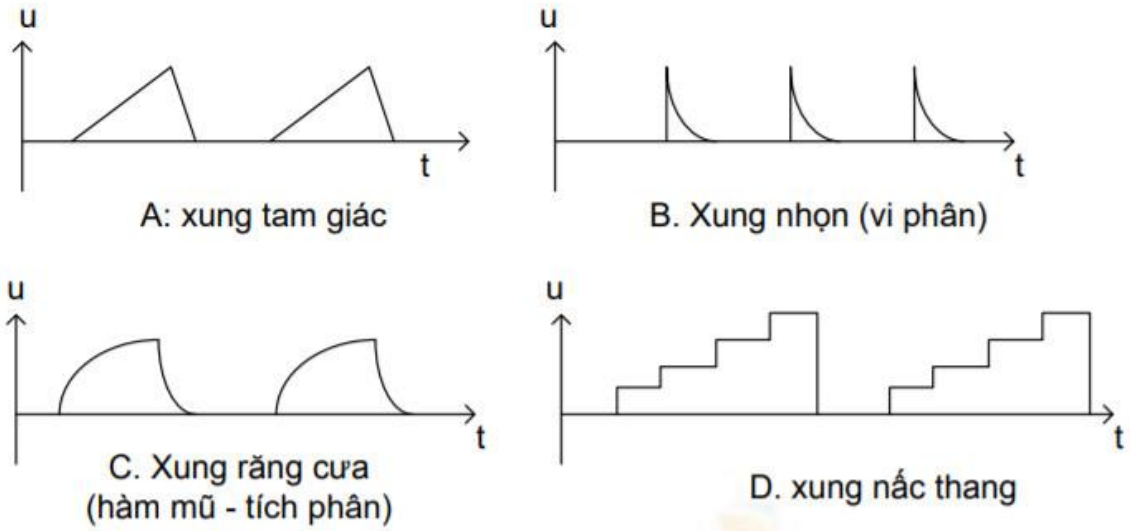
2. Tín hiệu xung:

Tiêu biểu cho tín hiệu rời rạc là tín hiệu vuông, dạng tín hiệu như hình 2, biên độ của tín hiệu chỉ có 2 giá trị mức cao V_H và mức thấp V_L, thời gian chuyển mức tín hiệu từ mức cao sang mức thấp và ngược là rất ngắn coi như bằng 0.

Tín hiệu xung không chỉ có tín hiệu xung vuông mà còn có mốt số dạng tín hiệu khác như xung tam giác, răng cưa, xung nhọn, xung nấc thang có chu kỳ tuần hoàn theo thời gian với chu kỳ lặp lại T.



Hình 1.2: a, xung vuông điện áp > 0. b, xung vuông điện áp đều nhau



Hình 1.3: Các dạng tín hiệu xung:

Trong nhiều trọờng hợp xung tam giác có thể coi là xung răng cưa.

Các dạng xung cơ bản trên rất khác nhau về dạng sóng, nhưng có điểm chung là

thời gian tồn tại xung rất ngắn, sự biến thiên biên độ từ thấp lên cao (xung nhọn) và từ

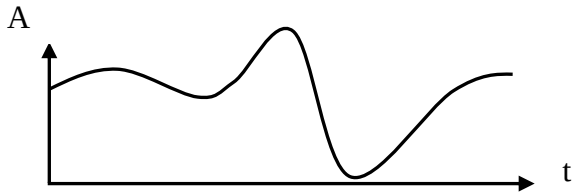
cao xuống thấp (nấc thang, tam giác) xảy ra rất nhanh.

Định nghĩa: Tín hiệu xung điện áp hay xung dòng điên là những tín hiệu có thời gian tồn tại rất ngắn, có thể so sánh với quá trình quá độ trong mạch điện mà chúng tác

dụng.

3. Tín hiệu tương tự, số:

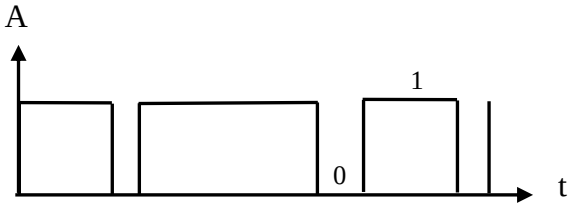
❖ *Tín hiệu tương tự:* Tín hiệu tương tự là tín hiệu có biên độ biến thiên liên tục theo thời gian. Trong thực tế các đại lượng vật lý như vận tốc , nhiệt độ môi trường biên độ tín hiệu đầu ra của loa trong máy thu vô tuyến vv…đều là các tín hiệu tương tự



Trong kỹ thuật điện tử mạch tương tự là mạch xử lý các tín hiệu có dạng như hình vẽ có nghĩa là mạch phải xử lý n mức tín hiệu khác nhau trong một khoảng thời gian xác định.

❖ *Tín hiệu số*

Tín hiệu số là tín hiệu có biên độ gián đoạn theo thời gian. Biên dộ chỉ có hai mức như hình vẽ . Mức (1) đặc trưng cho giá trị biên độ cao, mức (0) đặc trưng cho giá trị biên độ thấp



Như vậy khác với mạch tương tự mạch số chỉ xử lý các tín hiệu có dạng như hình vẽ có nghĩa là mạch chỉ phải xử lý 2 mức tín hiệu khác nhau theo thời gian mà thôi

CHƯƠNG 1: CÁC HỆ THỐNG SỐ

Giới thiệu: *Trong chương này, giới thiệu các khái niệm cơ bản về hệ thống tín hiệu số và các hệ thống đếm, phương pháp chuyển đổi giữa các hệ đếm.*

Mục tiêu:

- Hiểu các khái niệm về hệ thống đếm và làm được các bài tập về chuyển đổi số giữa các hệ đếm.

- Thực hiện thành thạo các phép toán chuyển đổi số, các phép toán cộng trừ ở hệ đếm 2, hệ đếm 8, hệ đếm 16.

- Rèn luyện tính tư duy logic và sáng tạo

Nội dung chính:

1.1. Các hệ thống số :

Hệ thống số thường sử dụng là hệ thống số có vị trí. Trong một hệ thống như vậy một số biểu diễn bằng một chuỗi các ký tự số (digit); Ở đó mỗi vị trí của ký tự số sẽ có một trọng số nhất định.

Trọng số ở đây chính là *cơ lũy thừa vị trí* của ký tự số trong chuỗi.

Cơ số chính là *số ký tự số* được dùng để biểu diễn trong một hệ thống.

Các hệ thống số thường gặp là hệ thống số thập phân (*Decimal system*), hệ thống số nhị phân (*Binary system*), hệ thống số bát phân (*Octal system*), hệ thống

$$G = \sum_{t=0}^{n} C^{t} \times A_{t} + \sum_{t'=-1}^{m} C^{t'} \times A_{t'}$$

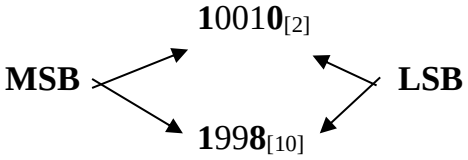
số thập lục phân (*Hexa-decimal*) v.v…Giá trị thập phân của một số được tính theo công thức sau :

Trong đó :

- G : *là giá trị.*
- t : *vị trí của ký tự số đứng trước dấu ngăn cách thập phân (0, 1, 2, 3, ...).*
- n : *số ký tự số đứng trước dấu ngăn cách thập phân của số trừ đi 1.*
- C : *cơ số.*
- A : *ký tự số.*
- t’ : *vị trí của ký tự số đứng sau dấu ngăn cách thập phân (-1, -2, -3, ...).*
- m : *số ký tự số đứng sau dấu ngăn cách thập phân của1 số.*

Trong các hệ thống số người ta thường quan tâm đến số có ý nghĩa cao nhất *(số có trọng số lớn nhất)* ký hiệu là MSB (**M**ost **S**ignificant **B**it) và số có ý nghĩa thấp nhất *(số có trọng số nhỏ nhất)* ký hiệu là LSB (**L**ess **S**ignificant **B**it)

Ví dụ :



MSD : **M**ost **S**ignificant **B**it

LSD : **L**ess **S**ignificant **B**it

1.1.1. Hệ thống số thập phân (Decimal) :

- Ký tự số : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
- Cơ số : **10**

Ví dụ : Vị trí

3 2 1 0

1 9 9 9$_{[10]}$ = **1**.10^3 + **9**.10^2 + **9**.10^1 + **9**.10^0

= 1000 + 900 + 90 + 9

0 -1 -2

1 , 2 5$_{[10]}$ = **1.**10^0 + **2.**10^{-1} + **5.**10^{-2}

= 1,00 + 0,2 + 0,05

1.1.2. Hệ thống số nhị phân (Binary) :

- Ký tự số : **0, 1**
- Cơ số : **2**

Mỗi con số trong số nhị phân (0 hoặc 1) đượ̣c gọi là một **bit** (viết tắt của **bi**nary digi**t**).

Các đơn vị đo khác :

Tên gọi	Viết tắt	Giá trị
Byte	B	8 bit
Kilo Byte	KB	1024 byte = 2^{10} B
Mega Byte	MB	1024 KB = 2^{20} B
Giga	GB	1024 MB = 2^{30} B

Ví du : ***Vị trí***

4 3 2 1 0

1 0 1 0 1$_{[2]}$ = **1**.2^4 + **0**.2^3 + **1**.2^2 + **0**.2^1 + **1**.2^0

= 16 + 0 + 4 + 0 + 1 = $21_{[10]}$

(Số nhị phân trên có 5 bit)

1 0 -1 -2 -3

1 1 , 1 0 1$_{[2]}$ = **1**.2^1 + **1**.2^0 + **1**.2^{-1} + **0**.2^{-2} + **1**.2^{-3}

= 2 + 1 + 0,5 + 0 + 0,125 = $3,625_{[10]}$

(Số nhị phân trên có 5 bit)

Nhận xét : - Nếu bit cuối cùng là 0 ⇒ số nhị phân đó là số chẳn.
- Nếu bit cuối cùng là 1 ⇒ số nhị phân đó là số lẻ.

1.1.3. Hệ thống số bát phân (Octal) :

- Ký tự số : **0, 1, 2, 3, 4, 5, 6, 7**
- Cơ số : **8**

Ví du : ***Vị trí***

1 0

4 6$_{[8]}$ = **4**.8^1 + **6**.8^0 = 32 + 6 = $38_{[10]}$

0 -1 -2

2 , 3 7$_{[8]}$ = **2**.8^0 + **3**.8^{-1} + **7**.8^{-2}

= 2 + 3.0,125 + 7.0,02 = $2,515_{[10]}$

1.1.4. Hệ thống số thập lục phân (Hexa-decimal) :

- Ký tự số : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**
- Cơ số : **16**

Ví du :

Vị trí

1 0

2 $\mathbf{E}_{[16]} = \mathbf{2}.16^1 + \mathbf{14}.16^0 = 46_{[10]}$

3 2 1 0 -1

0 1 2 C , D$_{[16]}$ = $\mathbf{0}.16^3 + \mathbf{1}.16^2 + \mathbf{2}.16^1 + \mathbf{12}.16^0 + \mathbf{13}.16^{-1}$

= 0 + 256 + 32 + 12 + 0,0625

= $300{,}0625_{[10]}$

Ghi chu : *Nếu số haxa-decimal bắt đầu bằng chữ thì khi viết phải thêm số 0 vào trước (Vd :* EF $\rightarrow$ 0EF).

1.2. Chuyển đổi giữa các hệ thống số:

1.2.1. Chuyển đổi số thập phân sang các hệ thống số khác:

Nguyên tắc : *lấy mỗi số hạng trong chuỗi số nhân với cơ số lũy thừa vị trí của nó sau đó lấy tổng tất cả* $\Rightarrow$ *kết quả (các ví dụ trên).*

1.2.2. Chuyển đổi số nhị phân sang hệ thống số bát phân và thập lục phân:

Nguyên tắc : *Nhóm từ phải qua trái* ***đủ bốn số*** *(bốn bit); nhóm cuối cùng nếu thiếu thì ta cứ thêm các số 0 vào. Thay thế các nhóm 4 bit thành các mã thập lục phân tương ứng.*

Ví du :

0010 1101$_{[2]}$ **= 2D**$_{[16]}$ 2 D

1100 1010 1111 1110 1101 1010$_{[2]}$ **= 0CAFEDA**$_{[16]}$ C A F E D A

Bảng mã thập lục phân :

Thập phân	Nhị phân	Thập lục phân
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A

11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1.2.3. Chuyển đổi số thập lục phân, số bát phân sang hệ thống số nhị phân:

Nguyên tắc : *Nhóm từ phải qua trái **đủ ba số** (ba bit); nhóm cuối cùng nếu thiếu thì ta cứ thêm các số 0 vào. Thay thế các nhóm ba bit thành các mã thập lục phân tương ứng.*

Ví dụ :

100 111 010$_{[2]}$ = **472**$_{[8]}$ (4 7 2)

001 000$_{[2]}$ = **10**$_{[8]}$ (1 0)

Chuyển đổi cơ số từ bát phân sang nhị phân :
Nguyên tắc : *Thay thế một ký tự số bằng một số nhị phân ba bit tương ứng theo bảng sau.*

Bát phân	0	1	2	3	4	5	6	7
Nhị phân	000	001	010	011	100	101	110	111

Ví dụ :

3 4 5$_{[8]}$ = **11100101**$_{[2]}$ (011 100 101)

1 3 7$_{[8]}$ = **1011111**$_{[2]}$ (001 011 111)

1.2.4. Chuyển đổi phần nguyên và phần thập phân của một số:

Nguyên tắc : *Thay thế một ký tự số bằng một số nhị phân bốn bit tương ứng.*

Ví dụ :

2 F E (H) = **1011111110**$_{[2]}$

0010 1111 1110

Chuyển đổi cơ số thập phân sang cơ số nhị phân, bát phân, thập lục phân :
Chia làm hai phần : phần nguyên (phần N) và phần thập phân (phần L).

* Phần nguyên **N** :

- Lấy N chia cho cơ số (2 hoặc 8 hoặc 16), thương số là N_0, số dư là n_0.

- Lấy N_0 chia cho cơ số (2 hoặc 8 hoặc 16), thương số là N_1, số dư là n_1.

- Lấy N_1 chia cho cơ số (2 hoặc 8 hoặc 16), thương số là N_2, số dư là n_2.

. . . . .

- Tiếp tục chia cho đến khi thương số $N_i = 0$, số dư là n_i .Khi đó số N biểu diễn dạng nhị phân là :

$$\mathbf{N_{[2]} = n_i\ n_{i-1}\ \ldots\ n_2\ n_1\ n_0}$$

(Các số dư được lấy theo thứ tự từ dưới lên)

Ví dụ 1 :

64$_{[10]}$ = **?**$_{[2]}$

64	2	
0	32	2
	0	16	2
		0	8	2
			0	4	2
				0	2	2
					0	1	2
						1	**0**

= **1000000**$_{[2]}$

35$_{[10]}$ = **?**$_{[2]}$

35	2	
1	17	2
	1	8	2
		0	4	2
			0	2	2
				0	1	2
					1	**0**

= **100011**$_{[2]}$

Ví dụ 2 :

1997$_{[10]}$ = **7CD**$_{[16]}$

1997	16	
13	124	16
	12	7	16
		7	0

423$_{[10]}$ = **1A7**$_{[16]}$

423	16	
7	26	16
	10	1	16
		1	0

Ví dụ 3 :

266$_{[10]}$ = **412**$_{[8]}$

266	8	
2	33	8
	1	4	8
		4	0

1999$_{[10]}$ = 3717$_{[8]}$

1999	8	
7	249	8
	1	31	8
		7	3	8

3 0

* Phần thập phân L :

- *Lấy phần L nhân cơ số thành là L' có phần nguyên là d_1, phần thập phân là L_1.*
- *Lấy phần L_1 nhân cơ số thành là L_1' có phần nguyên là d_2, phần thập phân là L_2.*
- *Lấy phần L_2 nhân cơ số thành là L_2' có phần nguyên là d_3, phần thập phân là L_3.*

. . . . . .

- *Tiếp tục cho đến khi phần thập phân của tích số bằng 0 hay đạt được số lẻ cần thi*ết.

Khi đó phần lẻ sẽ là :

$$\mathbf{L_{[2]} = d_1\, d_2\, d_3\, d_4 \ldots d_k}$$

Ví dụ 1 : $L_{[10]} = 0.6875 \Rightarrow L[2]$

_ 0.6875 x 2 = 1.3750 (L') $\Rightarrow$ d1 = 1; L1 = 0.3750

_ 0.3750 x 2 = 0.750 (L_1') $\Rightarrow$ d2 = 0; L2 = 0.750

_ 0.750 x 2 = 1.50 (L_2') $\Rightarrow$ d3 = 1; L3 = 0.50

_ 0.50 x 2 = 1.0 (L_3') $\Rightarrow$ d4 = 1; L4 = 0

$\Rightarrow$ L[2] = 0.1011

Ví dụ 2 : $L_{[10]} = 0.6875 \Rightarrow L[8]$

_ 0.6875 x 8 = 5.5 (L') $\Rightarrow$ d1 = 5; L1 = 0.5

_ 0.5 x 8 = 4.0(L_1') $\Rightarrow$ d2 = 4; L2 = 0

$\Rightarrow$ L[8] = 0.54

Ví dụ 3 : $L_{[10]} = 0.6875 \Rightarrow L[16]$

_ 0.6875 x 16 = 11 (L') $\Rightarrow$ d1 = B; L1 = 0

$\Rightarrow$ L[16] = 0.B

BÀI TẬP CHƯƠNG 1

1.Chuyển đổi từ số Binary sang Decimal

a.10110 b.10001101 c.100100001001

d.1111010111 e.10111111

2. Chuyển đổi từ số Decimal sang Binary

a.37 b.14 c.189 d.205 e.2313

3.Một số nhị phân 8 bit có giá trị thập phân tương ứng lớn nhất là bao nhiêu?

4.Chuyển đổi từ số Octal sang decimal

a.743 b.36 c.3777 d.257 e.1204

5. Chuyển đổi từ số Decimal sang Octal

a.59 b.372 c.919 d.65536 e.255

6. Chuyển đổi sang số Binary các số từ bài 2 đến bài 4

7.Chuyển đổi từ số Hex sang Decimal

a.92 b.1A6 c.37FD d.2CO e.7FF

8.Chuyển đổi từ Decimal sang Hex

a.75 b.314 c.2048 d.25619

9.Mã hóa những số decimal sau sang mã BCD

a.47 b.962 c.187 d.1204

10.Chuyển đổi từ mã BCD sang Decimal

a.1001011101010010 b.000110000100

c.0111011101110101 d.010010010010

11.Dịch sang mã ASCII các ký tự sau: CDDTK6=2005

CHƯƠNG 2:
ĐẠI SỐ BOOLE VÀ THIẾT KẾ MẠCH LOGIC

Giới thiệu: *Trong chương này, giới thiệu các khái niệm cơ bản về đại số Boole (đại số logic), cổng logic và các phương pháp để thành lập bảng giá trị, xây dựng hàm logic, rút gọn hàm logic để từ một bài toán có thể thiết kế thành một mạch số logic.*

Mục tiêu:

- Hiểu các khái niệm về biến và hàm logic và thực hiện được các phép tính toán cơ bản trong đại số logic (đại số Boole).

- Hiểu được nguyên lý làm việc của các cổng logic.

- Biết biểu diễn một hàm logic dưới nhiều dạng.

- Có khả năng phân tích, tính toán, thiết kế được các mạch tổ hợp.

- Rèn luyện tính kiên trì, cẩn thận và tư duy logic.

Nội dung chính:

2.1. Đại số Boole :

2.1.1. Định nghĩa :

Đại số Boole (hay còn gọi là **đại số logic** do George Boole, nhà toán học người Anh, sáng tạo vào thế kỷ XIX) là một cấu trúc đại số được xây dựng trên tập các phần tử nhị phân (Binary) cùng với 2 phép toán cộng và nhân thỏa các điều kiện sau :

a. Kín với các phép toán cộng (+) và nhân (.).

Tức là ∀ A,B € X thì: A+B € X và A.B € X.

b. Tồn tại các phần tử trung hòa:

i. Đối với phép cộng sẽ có phần tử trung hòa 0 (đồng nhất):

$$x + 0 = x$$

ii. Đối với phép toán nhân sẽ có phần tử trung hòa 1 (đồng nhất):

$$x \cdot 1 = x$$

c. Giao hoán :

i. $x + y = y + x$

ii. $x \cdot y = y \cdot x$

d. Phân bố và kết hợp :

i. $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

ii. $a + (b \cdot c) = (a + b) \cdot (a + c)$

e. Luôn luôn tồn tại một phần tử nghịch (bù) sao cho :

i. $x + \bar{x} = 1$

ii. $x \cdot \bar{x} = 0$

+ Ghi chú: *Trong chương này và các chương sau các ký hiệu 0 và 1 là ký hiệu cho 2 mức Logic 0 và 1 chứ không phải là ký hiệu của số nhị phân. Do đó các phép toán phải tuân thủ theo nguyên tắt riêng của nó.*

2.1.2. Các phép toán:

a. Phép cộng **(OR):**

a	**b**	**a + b**
0	0	0
0	1	1
1	0	1
1	1	1

b. Phép nhân **(AND)**:

a	**b**	**a . b**
0	0	0
0	1	0
1	0	0
1	1	1

c. Phép bù **(NOT) :**

a	$\bar{a}$
0	1
1	0

2.1.3. Các công thức và định lý:

a. Quan hệ giữa các hằng số:

Những quan hệ dưới đây giữa hai hằng số (0, 1) làm tiên đề của đại số Boole. Đó là các quy tắc phép toán cơ bản đối với tư duy logic.

Công thức 1-1: $0 \cdot 0 = 0$

Công thức 1-2: $1 + 1 = 1$

Công thức 2-1: $0 \cdot 1 = 0$

Công thức 2-2: 1 + 0 = 1

Công thức 3-1: 0 + 0 = 0

Công thức 3-2: 1 . 1 = 1

Công thức 4-1: $\bar{0} = 1$

Công thức 4-2: $\bar{1} = 0$

b. Quan hệ giữa biến số và hằng số:

Công thức 5-1: x . 1 = x

Công thức 5-2: x + 0 = x

Công thức 6-1: x . 0 = 0

Công thức 6-2: x + 1= 1

Công thức 7-1: $x.\bar{x} = 0$

Công thức 7-2: $x + \bar{x} = 1$

Biến số ở đây đặt là x, hai hằng số Logic là 0 và 1.

c. Luật giao hoán :

Công thức 8-1: x + y = y+ x

Công thức 8-2: x . y = y. x

d. Luật kết hợp :

Công thức 9-1: (x . y).z = x.(y. z)

Công thức 9-2: (x + y) + z = x + (y+ z)

Công thức 10-1: x . (y + z) = x.y + x.z

Công thức 10-2: x + y . z = (x+y) . (x +z)

e. Luật phân phối :

Công thức 11-1: x + x = x

Công thức 11-2: x . x = x

f. Luật đồng nhất :

Công thức 12: $\bar{\bar{x}} = x$

g. Định lý De_Morgan :

Công thức 13-1: $\overline{x + y} = \bar{x}.\bar{y}$

Công thức 13-2: $\overline{x.y} = \bar{x} + \bar{y}$

h. Định lý hấp thu :

Công thức 14-1: x + x.y = x

Công thức 14-2: x . (x+y) = x

2.1.4. Hàm Boole:

Một biến nhị phân (x, y, z, …) có thể lấy giá trị 0 hoặc 1. Hàm Boole là một biểu thức tạo bởi các biến nhị phân, các phép toán cộng "+"; nhân "."; phép bù (đảo); các dấu bằng "="; dấu ngoặc "( )".

2.1.5. Biểu diễn hàm Boole

Một hàm Boole có thể được biểu diễn bằng các phương pháp khác nhau tùy theo đặc điểm của từng hàm. Thường dùng bốn phương pháp. Đó là:

2.1.5.1. Bảng trạng thái

(hay còn gọi là bảng sự thật, bảng chân lý - Truth table)

Bảng giá trị là bảng miêu tả quan hệ giữa các giá trị của hàm số tương ứng với mọi giá trị có thể có của các biến số.

Khi lập bảng ta cho biến số giá trị 0 và 1 để tạo thành các tổ hợp biến (không trùng nhau) rồi tính giá trị hàm. Đặc điểm của phương pháp này tương đối rõ ràng, trực quan nhưng sẽ rắc rối nếu biến số nhiều, không áp dụng được các công thức và định lý logic để tính toán.

Ví dụ :

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

2.1.5.2. Giải tích

a. Dạng chuẩn 1 – Minterm – Tổng các tích:

(tổng các Mintern - tích chuẩn)

i) Khái niệm Minterm :

- **Các mintern có được là khi ta kết hợp n biến bằng phép toán AND.**
- *Nếu có n biến ta sẽ có 2^n tổ hợp biến $\Rightarrow$ có 2^n mintern.*

- Nếu biến có giá trị "1" ta sử dụng dạng nguyên biến số, ngược lại, nếu biến có giá trị "0" ta sử dụng dạng bù biến số.
- *Ký hiệu của mintern là m_i ; với i là giá trị thập phân của tổ hợp các biến.*

ii) Dạng chuẩn 1 :

Dạng chuẩn 1 là biểu thức đại số dùng phép toán cộng (OR) để cộng tất cả các minterm làm cho hàm số logic bằng "1".

iii) Ví dụ :

x	**y**	**z**	**m_i**	**F_1**	**F_2**
0	0	0	$\bar{x}\bar{y}\bar{z} = m_0$	0	0
0	0	1	$\bar{x}\bar{y}z = m_1$	1	1
0	1	0	$\bar{x}y\bar{z} = m_2$	1	0
0	1	1	$\bar{x}yz = m_3$	1	0
1	0	0	$x\bar{y}\bar{z} = m_4$	0	1
1	0	1	$x\bar{y}z = m_5$	0	1
1	1	0	$xy\bar{z} = m_6$	0	0
1	1	1	$xyz = m_7$	0	1

* Lưu ý :

_ Các biến x, y, z có dấu bù hoặc không bù là tùy thuộc vào giá trị "0" hoặc "1".

_ Giá trị của F_1 hoặc F_2 là giá trị tự cho và ta có thể chọn giá trị khác.

Căn cứ vào bảng trên ta có dạng chuẩn 1 (cả ba cách viết đều được) của hai hàm F_1 và F_2.

$F_1 = \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz$

$= m_1 + m_2 + m_3$

$= \sum (1, 2, 3)$

$F_2 = \bar{x}\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + xyz$

$= m_1 + m_4 + m_5 + m_7$

$= \sum (1, 4, 5, 7)$

b. Dạng chuẩn 2 – Maxterm – Tích các tổng: (tích các Maxtern – tổng chuẩn)

i) Khái niệm Maxterm :

- **Các maxtern có được là khi ta kết hợp n biến bằng phép toán OR.**
- *Nếu có n biến ta sẽ có 2^n tổ hợp biến $\Rightarrow$ có 2^n maxtern.*
- Nếu biến có giá trị "1" ta sử dụng dạng bù biến số, ngược lại, nếu biến có giá trị "0" ta sử dụng dạng nguyên biến số.
- *Ký hiệu của maxtern là M_i ; với i là giá trị thập phân của tổ hợp các biến.*

ii) Dạng chuẩn 2 :

Dạng chuẩn 2 là biểu thức đại số dùng phép toán nhân (AND) để nhân tất cả các maxterm làm cho hàm số logic bằng "0".

iii) Ví dụ :

x	y	z	M_i	F_1	F_2
0	0	0	$x+y+z = M_0$	0	0
0	0	1	$x + y + \bar{z} = M_1$	1	1
0	1	0	$x+ \bar{y} +z = M_2$	1	0
0	1	1	$x+ \bar{y} + \bar{z} = M_3$	1	0
1	0	0	$\bar{x} + y+z = M_4$	0	1
1	0	1	$\bar{x} +y+ \bar{z} = M_5$	0	1
1	1	0	$\bar{x} + \bar{y} +z = M_6$	0	0
1	1	1	$\bar{x} + \bar{y} + \bar{z} = M_7$	0	1

*** Lưu ý :**

_ Các biến x, y, z có dấu bù hoặc không bù là tùy thuộc vào giá trị "1" hoặc "0".

_ Giá trị của F_1 hoặc F_2 là giá trị tự cho và ta có thể chọn giá trị khác.

Căn cứ vào bảng trên ta có dạng chuẩn 2 (cả ba cách viết đều được) của hai hàm F_1 và F_2.

$F_1 = (x + y + z)(\bar{x} + y + z).(\bar{x} + \bar{y} + \bar{z}).(\bar{x} + y + \bar{z}).(\bar{x} + y + z)$

$= M_0 . M_4 . M_5 . M_6 . M_7$

$= \Pi (0, 4, 5, 6, 7).$

$F_2 = (x + y + z).(x + \bar{y} + z).(\bar{x} + \bar{y} + \bar{z}).(\bar{x} + y + z)$

$= M_0 . M_2 . M_3 . M_6$

$= \Pi (0, 2, 3, 6).$

Quan hệ giữa dạng chuẩn 1 va dạng chuẩn 2 :

Giữa dạng chuẩn 1 và dạng chuẩn 2 cụ thể là giữa minterm và maxterm quan hệ với nhau bởi công thức sau :

$$m_i = \overline{M_i} \quad \text{với} \quad i = 0 \ldots 2^n\text{-}1)$$

Chứng minh :

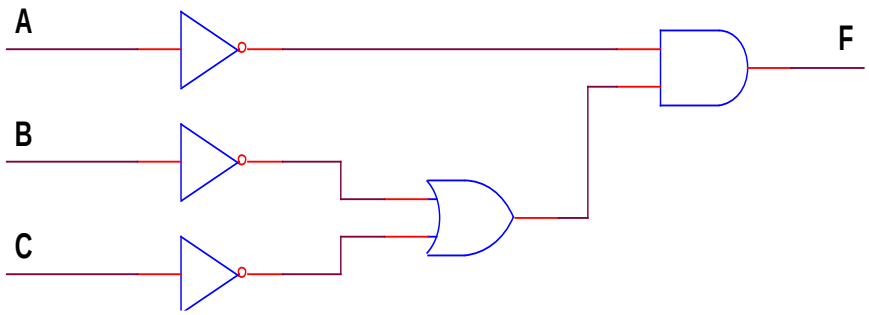
Giả sử cho minterm $m_0 = \bar{x}.\bar{y}.z$. Biến đổi m_0 ta có :

$$\xrightarrow{\text{Laáy buø } m_0} \overline{m}_0 = \overline{x.y.z} \xrightarrow{\text{Ñònh lyù De\_Morgan}} \overline{m}_0 = \overline{x} + \overline{y} + \overline{z}$$

$$\xrightarrow{\text{Luaät hoaøn nguyeân}} \overline{m}_0 = x + y + z = M_0 \text{ Hay } m_0 = M_0$$

2.1.5.3. Sơ đồ mạch logic :

Sơ đồ logic có được khi ta dùng các ký hiệu logic (ký hiệu các cổng logic) biểu thị hàm số.



Biểu thức hàm số :

Biểu thức hàm số dạng đại số logic dùng các phép toán nhân (AND), cộng (OR), bù (NOT) biểu thị quan hệ giữa các biến trong hàm.

Có hai dạng để biểu diễn hàm số, đó là dạng chuẩn 1 (tổng các tích hay tích chuẩn - Minterm) và dạng chuẩn 2 (tích các tổng hay tổng chuẩn – Maxterm).

Ví dụ :

$$F_1(w,x,y,z) = \sum m_i = \sum (0,1,2,4,5,6,8,9,12,13,14)$$
$$= \overline{w}\overline{z} + x\overline{z} + \overline{y} \qquad \textit{(Daïng chuaån 1)}$$

$$F_2(A,B,C) = \Pi M_i = \Pi (0,4,6,7)$$
$$= (\overline{A} + \overline{B})(B + C)$$

(Daïng chuaån 2)

2.5.1.4. Bìa Karnaugh :

Bìa Karnaugh là phương pháp hình vẽ biểu thị hàm logic (sẽ nói kỹ ở phần sau)

Ví dụ :

F	xy			
z	00	01	11	10
0			**1**	**1**
1		**1**	**1**	

Trong cấu trúc đại số Boole, một mệnh đề được gọi là đối ngẫu với mệnh đề khác nếu ta thay thế 0 thành 1 và 1 thành 0, dấu cộng (+) thành dấu nhân (.) và dấu nhân (.) thành dấu cộng (+).

Khi đã chứng minh một mệnh đề là đúng thì mệnh đề đối ngẫu của nó cũng đúng.

Ví dụ: 2 mệnh đề A+1=1 và A.0 = 0 là 2 mệnh đề đối ngẫu.

2.1.6. Đơn giản hóa hàm Boole:

2.1.6.1. Sử dụng các công thức, định lý:

Trong bất kỳ đẳng thức nào, nếu thay thế một biến nào đó bằng một hàm số (nhiều biến) thì đẳng thức vẫn thiết lập.

Quy tắc này được ứng dụng rất nhiều trong việc biến đổi các công thức đã biết để cho ra một công thức mới hay để rút gọn một hàm Boole nào đó.

Ví dụ :

+ Theo luật hoàn nguyên ta có: $\bar{\bar{x}} = x$

+ Cho một hàm Boole $F_1 = (A + B) . C$

+ Thay thế $(A + B) . C = x$

Nên $F_1 = \overline{\overline{(A + B)C}}$

2.1.6.2. Sử dụng bìa Karnaugh:

$\bar{Z}$ là bù của hàm số Z sẽ có được bằng cách đổi dấu "." thành dấu "+"; dấu "+" thành dấu "."; "0" thành "1"; "1" thành "0"; biến số thành đảo của biến số đó; đảo biến số thành nguyên biến số.

Ví dụ:

$$Z_1 = \bar{x}.\bar{y} + z.w + 0$$

Tìm $\overline{Z_1}$ là bù của Z: $\overline{Z_1} = (x + y).(\bar{z} + \bar{w}).1$

Khi tìm đảo của một hàm số, những gạch ngang nào (biểu thị phép toán đảo) ở trên nhiều biến thì vẫn giữ nguyên.

Ví dụ 2 :

$Z_2 = x + \overline{y + \bar{z} + \overline{w + i}} \xrightarrow{\text{aùp duïng quy taéc tìm haøm ñaûo}} \overline{Z}_2 = \bar{x} \cdot \overline{\bar{y} \cdot z \cdot \overline{w} \cdot \bar{i}}$

Chú ý thứ tự ưu tiên như sau : dấu móc "(,)"; dấu nhân "." ; dấu cộng "+".

Ví dụ 3 :

$Z_3 = \overline{A} \cdot \overline{B} + C \cdot D \xrightarrow{\text{aùp duïng quy taéc tìm haøm ñaûo vaø theo thöù töï öu tieân}} \overline{Z}_3 = (A + B) \cdot (\overline{C} + D) \rightarrow$ **ÑUÙNG**

$\overline{Z}_3 = A + B \cdot C + D \rightarrow$ **SAI**

c) Quy tắc đối ngẫu :

Hàm Z và Z' được gọi là đối ngẫu khi các dấu cộng "+" và dấu "." ; các giá trị "0" và "1" đổi chỗ cho nhau một cách tương ứng.

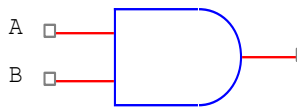
Ví dụ :

* $Z_1 = (A + B).(A + C.1) \xrightarrow{\text{(Ñoái ngaãu)}} Z'_1 = A.B + A.(C + 0)$

* $Z_2 = \overline{\overline{A} + \overline{B + \overline{C}}} \xrightarrow{\text{(Ñoái ngaãu)}} Z'_2 = \overline{\overline{A} \cdot \overline{B \cdot \overline{C}}}$

2.2. Các cổng logic cơ bản:

2.2.1. Cổng AND (AND gate)



X=A.B

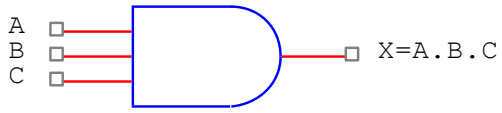
Bảng giá trị (Truth table)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

X=0 khi ∃ 1 ngõ vào =0

X=1 khi ∀ ngõ vào =1

Cổng AND 3 ngõ vào:

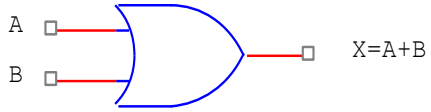


Bảng giá trị (Truth table)

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

2.2.2. Cổng OR (OR gate)

Cổng OR 2 ngõ vào:



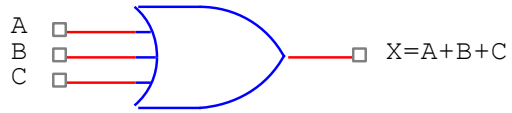
Bảng giá trị (Truth table)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

X=1 khi ∃ 1 ngõ vào = 1

X=0 khi ∀ ngõ vào = 0

Cổng OR 3 ngõ vào:

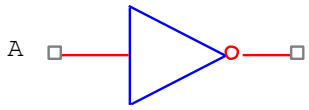


Bảng giá trị (Truth table)

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

2.2.3. Cổng đảo (NOT gate)

Ký hiệu

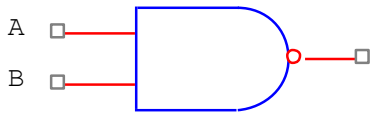


Bảng giá trị (Truth table)

A	X= $\overline{A}$
0	1
1	0

2.2.4. Cổng NAND (AND +NOT)

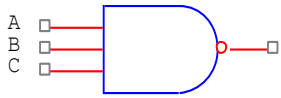
Cổng NAND 2 ngõ vào :



Bảng giá trị (Truth table)

A	B	X= $\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

Cổng NAND 3 ngõ vào :

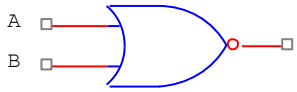


Bảng giá trị (Truth table)

A	B	C	X= $\overline{ABC}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

2.2.5. Cổng NOR (OR +NOT)

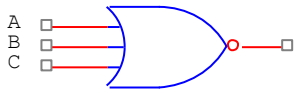
Hàm NOR 2 ngõ vào



Bảng giá trị (Truth table)

A	B	X= $\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

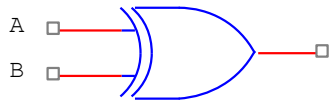
Hàm NOR 3 ngõ vào:



Bảng giá trị (Truth table)

A	B	C	X= $\overline{A+B+C}$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

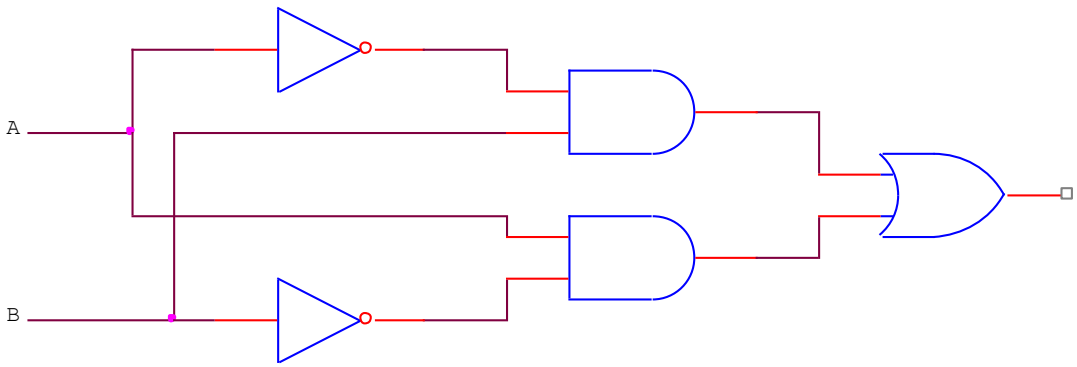
2.2.6. Cổng EXOR (EX-OR gate)



Bảng giá trị (Truth table)

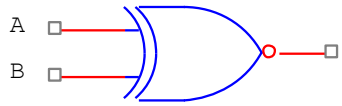
A	B	X= $\overline{A}B + A\overline{B} = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Hay sơ đồ được tương đương như sau:



Ghi chú cổng EX-OR không có nhiều hơn hai ngõ vào.

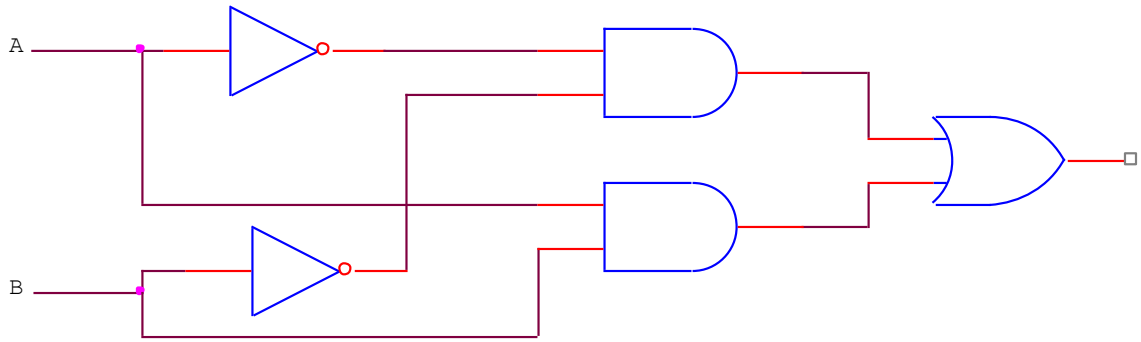
Cổng EXNOR (EX_NOR gate)



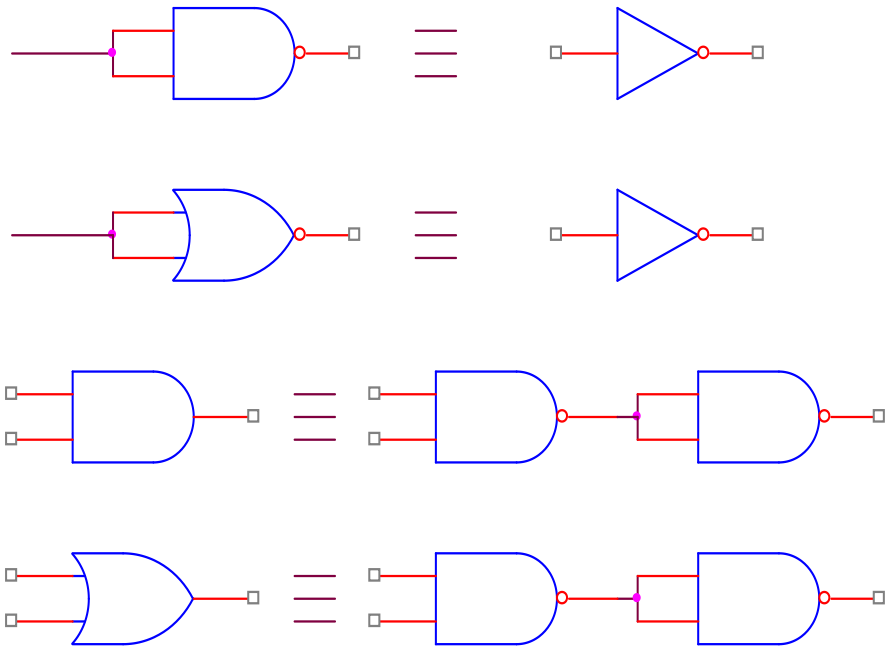
Bảng giá trị (Truth table)

A	B	X= $AB + \overline{A}\overline{B} = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

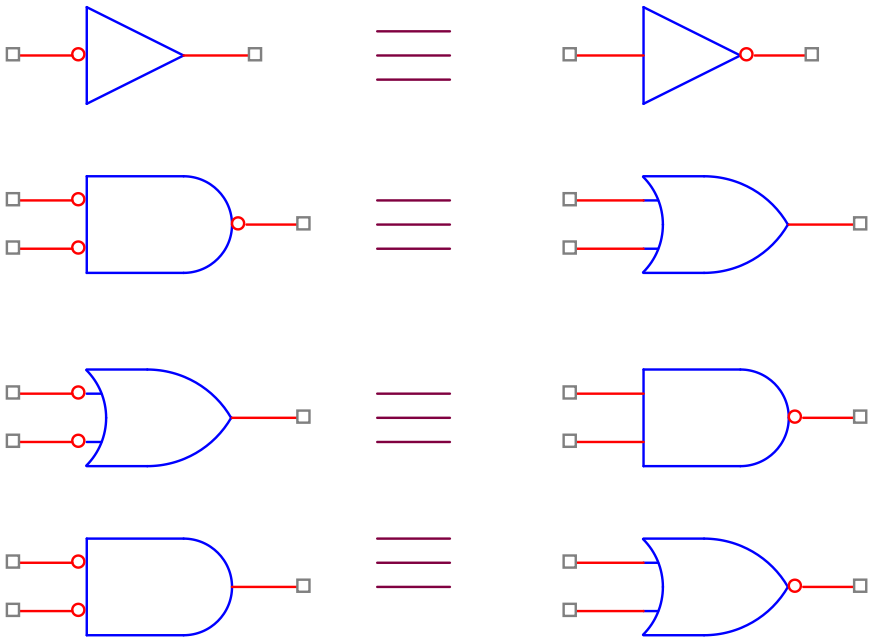
Hay sơ đồ được tương đương như sau:



2.2.7. Dùng cổng NAND tạo ra các cổng Logic khác



2.2.8. Dùng cổng NOR tạo ra các cổng Logic khác



2.3. Thiết kế mạch logic:

2.3.1. Lập bảng trạng thái:

Khi lập bảng ta cho biến số giá trị 0 và 1 để tạo thành các tổ hợp biến (không trùng nhau) rồi tính giá trị hàm. Đặc điểm của phương pháp này tương đối rõ ràng, trực quan nhưng sẽ rắc rối nếu biến số nhiều, không áp dụng được các công thức và định lý logic để tính toán.

Ví dụ :

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

2.3.2. Viết hàm boole từ bảng trạng thái:

Biểu thức hàm số :

Biểu thức hàm số dạng đại số logic dùng các phép toán nhân (AND), cộng (OR), bù (NOT) biểu thị quan hệ giữa các biến trong hàm.

Có hai dạng để biểu diễn hàm số, đó là dạng chuẩn 1 (tổng các tích hay tích chuẩn - Minterm) và dạng chuẩn 2 (tích các tổng hay tổng chuẩn – Maxterm).

Ví dụ :

$$F_1(w,x,y,z) = \sum m_i = \sum(0,1,2,4,5,6,8,9,12,13,14)$$
$$= \overline{w}\overline{z} + x\overline{z} + \overline{y}$$ *(Daïng chuaån 1)*

$$F_2(A,B,C) = \Pi M_i = \Pi(0,4,6,7)$$
$$= (\overline{A} + \overline{B})(B + C)$$ (Daïng chuaån 2)

2.3.3. Rút gọn hàm boole

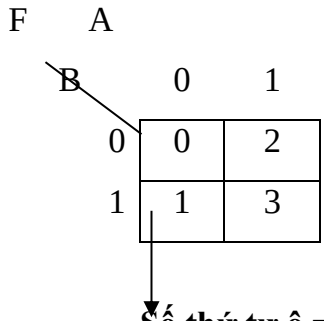
Bìa Karnaugh là bìa có số ô bằng 2^n ,với n là số biến của hàm Boole, một ô sẽ tương đương với một tổ hợp của các biến đã cho.

Hai ô được gọi là liên tiếp nhau(kế cận nhau)khi nó chỉ khác nhau 1 biến.

Các biến phải được sắp xếp với nhau sau cho 2 ô kế cận nhau chỉ khác nhau 1 bit. Nếu không tuân theo nguyên tắt này thì không còn là bìa karnaugh nữa.

a. Bìa K 2 biến

Số ô cần biểu diễn hàm là 2^2= 4 ô (Có n biến sẽ cần 2^n ô)



Số thứ tự ô = giá trị thập phân của tổ hợp nhị phân tương ứng.

VD: theo hình trên thì khi A=1;B=0 thì tổ hợp nhị phân là $10_{[2]}=2_{[D]}$.Do đó ô này có số thứ tự là 2

VD:Biểu diễn hàm sau bằng Bìa K :F(A,B)=Σ(0,2).Đây là dạng chuẩn 1 .Nếu biểu diễn dưới dạng bảng giá trị thì ta có như sau:

Hàng	A	B	F
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	0

Từ bảng giá trị trên ta thấy Số thứ tự ô=Số thứ tự hàng.Lúc đó ta biểu diễn hàm Boole bằng bìa K như sau:

F \ A / B	0	1
0	1	1
1	0	0

+ Ô số 0 và ô 2 có giá trị là 1.Các ô còn lại có giá trị là 0

Tuy nhiên ta có thể biểu diễn hàm trên như sau:

F \ A / B	0	1
0	1	1
1		

F \ A / B	0	1
0		
1	0	0

b. Bìa K 3 biến

Số ô = 2^3= 8 ô

F \ AB / C	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Số thứ tự ô = giá trị thập phân của tổ hợp nhị phân tương ứng.

VD: Cho hàm Boole F(A,B,C)=Σ(1,2,4,7).Ta biểu diễn dạng bìa K như sau:

F A F A

B	00	01	10	11
0		1		1
1	1		1	

B	00	01	10	11
0	0		0	
1		0		0

c. Bìa K 4 biến:

F AB / CD	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Số thứ tự ô = giá trị thập phân của tổ hợp nhị phân tương ứng.

VD : Cho hàm Boole F(A,B,C,D)=Σ(0,1,2,4,7,10,14,15).Biểu diễn bằng bìa K

F AB / CD	00	01	11	10
00	1	1		
01	1			
11		1		
10	1		1	1

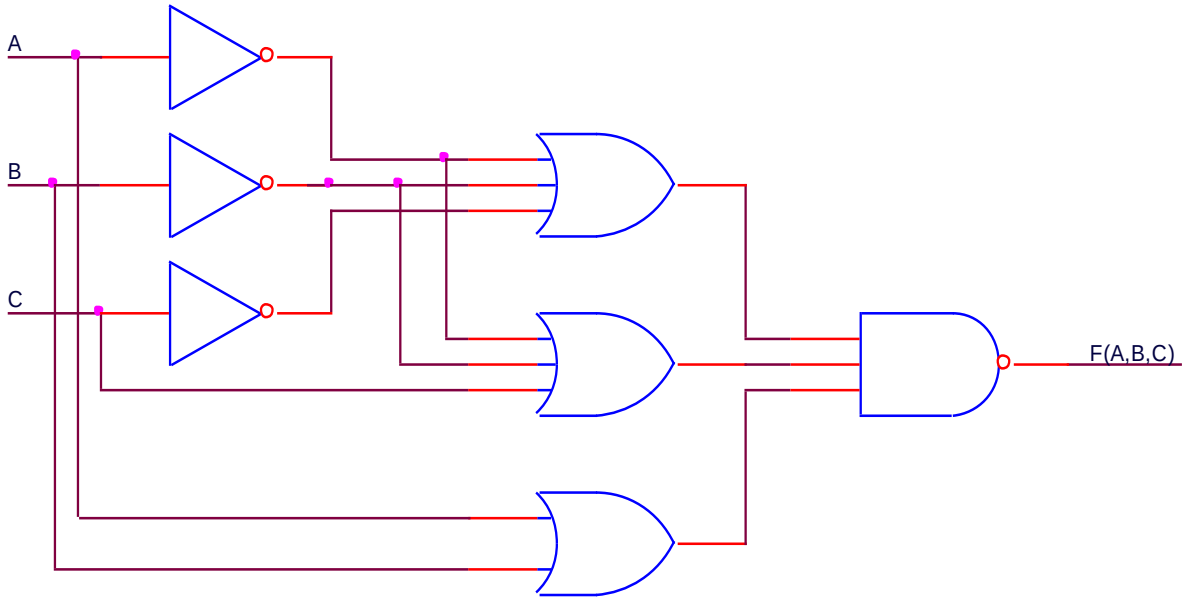
F AB / CD	00	01	11	10
00			0	0
01		0	0	0
11	0		0	0
10		0		

d. Bìa K 5 biến:

F ABC / DE	0 00	0 01	0 11	0 10	1 10	1 11	1 01	1 00
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

2.3.4. Vẽ sơ đồ mạch logic

Vẽ sơ đồ thực hiện mạch bằng các cổng Logic

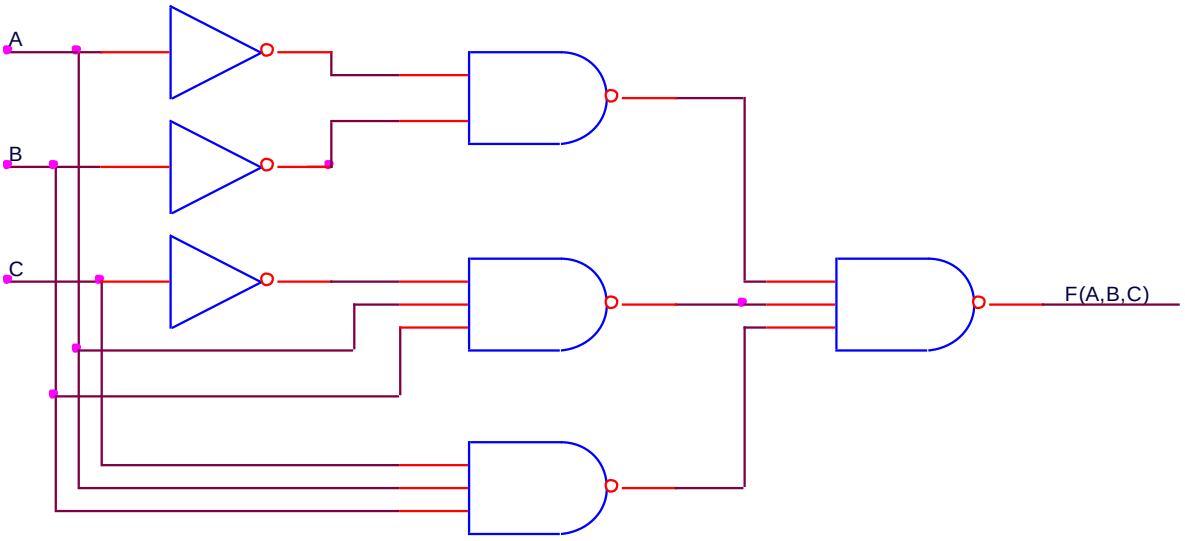


Cũng cho hàm như trên .Thực hiện bằng cấu trúc NAND-NAND

F (A,B,C)= $ABC+AB\overline{C}+\overline{A}\,\overline{B}=\overline{\overline{ABC}+\overline{AB\overline{C}}+\overline{\overline{A}\,\overline{B}}}=\overline{\overline{ABC}}.\overline{\overline{AB\overline{C}}}\;\overline{\overline{\overline{A}\,\overline{B}}}$

Đến đây ta thấy đã xuất hiện cấu trúc mong muốn nên không áp dụng tiếp định lý Demorgan.

Sơ đồ thực hiện mạch



2.3.5. Tối ưu mạch logic

Đơn giản hóa hàm Boole:

a. Phương pháp đại số :Sử dụng các công thức ,các tiên đề và định lý để rút gọn

VD: Rút gọn hàm sau

+F(A,B,C)=ABC+AB+C =AB(C+1) + C =1+C =1

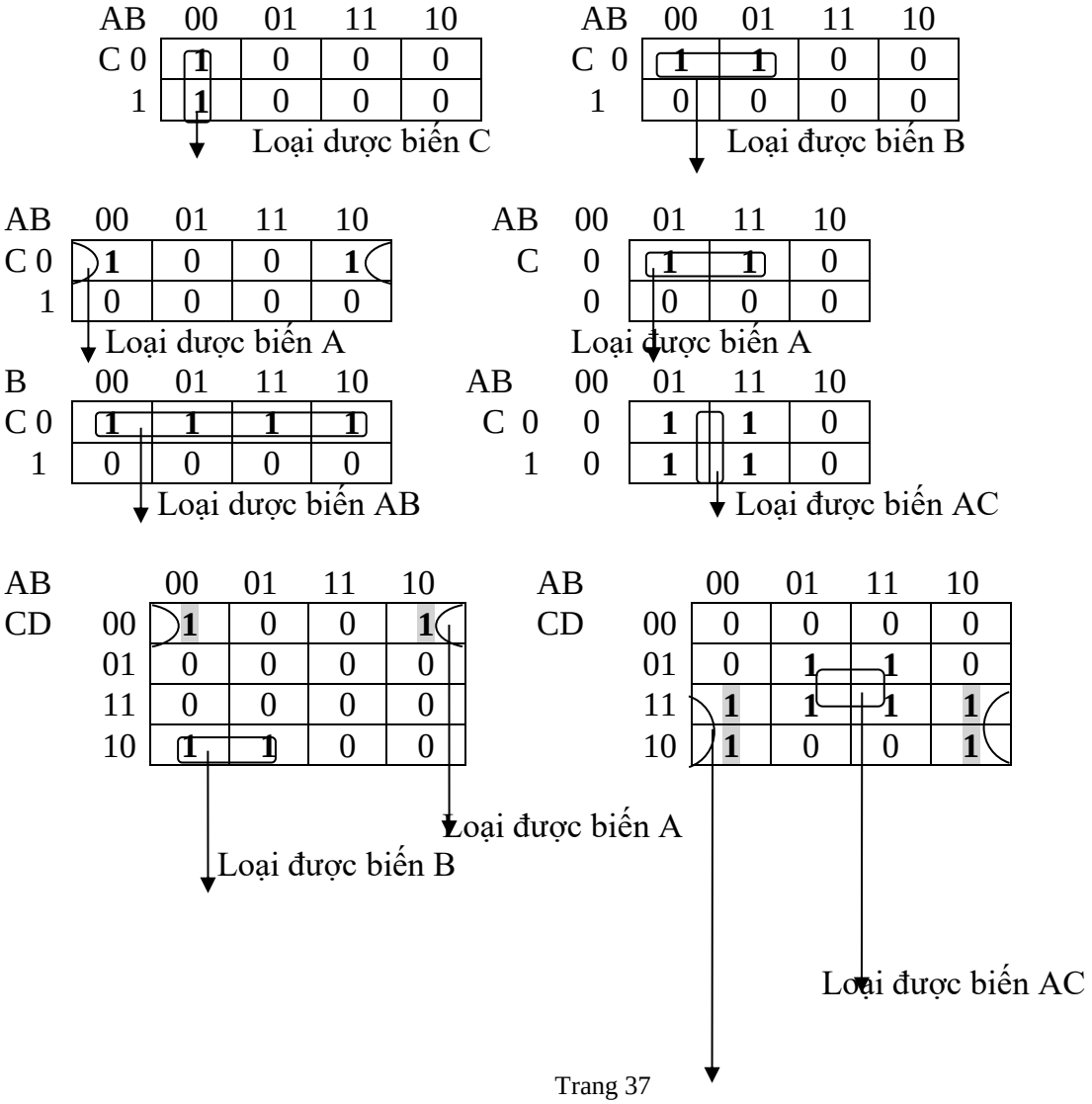
+F(x,y) = x(x+y) = xx+xy = xy

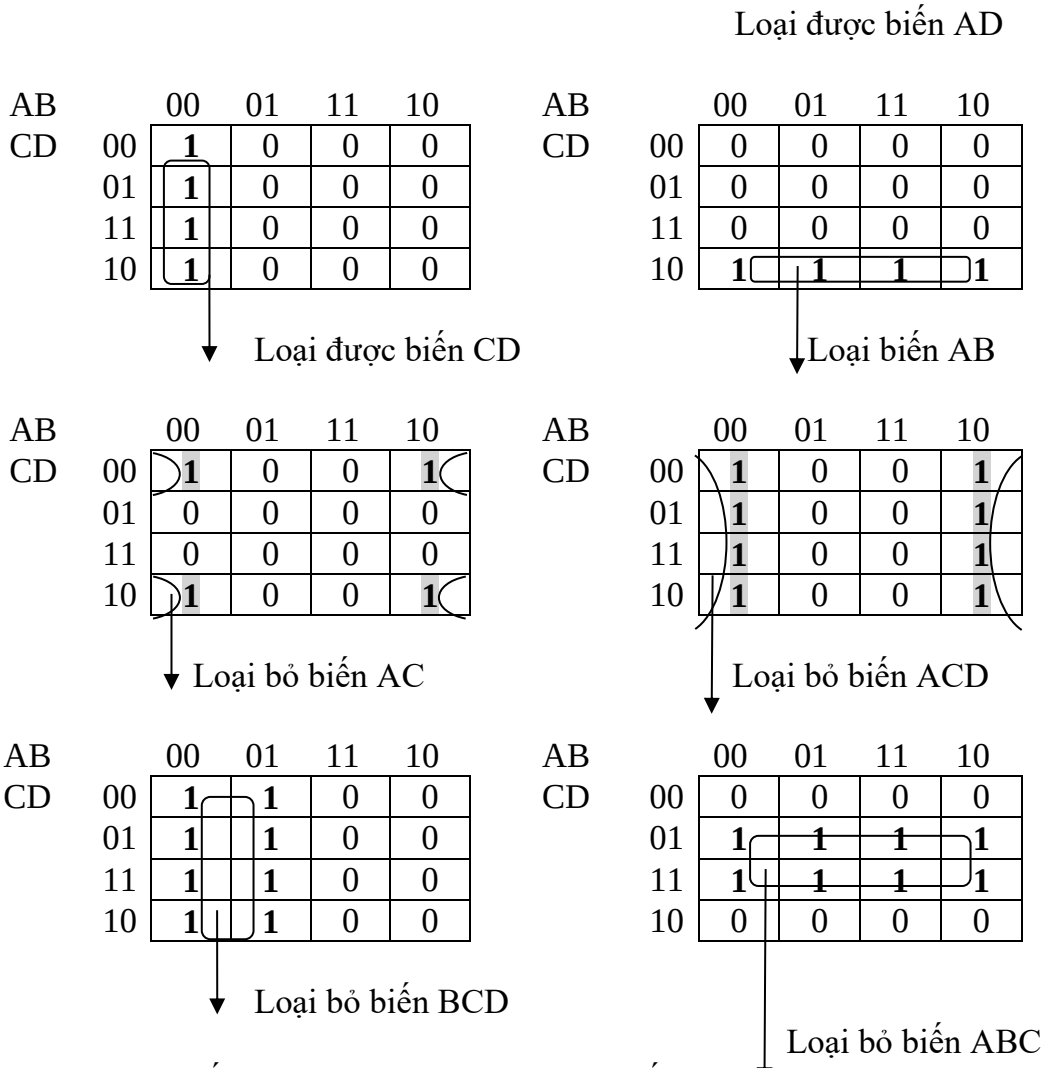
+F(x,y,z) = xyz+xyz+xy = xz(y+y)+xy = xz+xy

+F(x,y,z) = xy+xz+yz = xy+xz+yz(x+x) = xy+xz+yzx+yzx = xy(1+z)+xz(1+y) = xy+xz.

Phương pháp đại số rút gọn hàm Boole bắt buộc ta phải nhớ các công thức,các quy tắt,các định lý …Kết quả cuối cùng ta cũng không biết là đã tối ưu chưa.Ta có một phương pháp khác có thể khắc phục được những vấn đề trên là phương pháp rút gọn bằng bìa K

Nguyên tắc: Khi gom 2 ô liên tiếp với nhau thì ta sẽ loại đi được 1 biến. Biến bị loại chính là biến khác nhau trong 2 ô liên tiếp.Ta có thể gom cùng lúc 2 ô,4 ô,8 ô,16 ô tức là gom 2^n ô kế cận nhau.Khi gom 2^n ô kế cận nhau ta loại bỏ được n biến. Vị trí các ô kế cận cho phép như sau:





Khi gom các ô kế cận nhau ta loại bỏ những biến khác nhau,chỉ giữ lại những biến giống nhau.Khi ta gom những ô kế cận có giá trị là 1 thì biến giữ lại là chính nó nếu biến đó mang giá trị là 1 và sẽ có giá trị bù nếu biến đó là 0

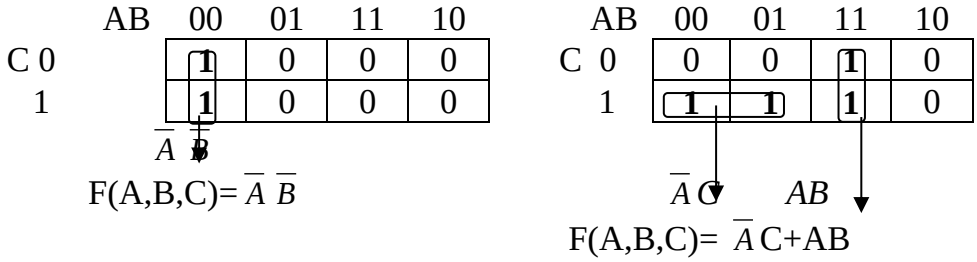
VD:Có 2 tổ hợp được gom có giá trị là

Tổ hợp 1:ABC 010

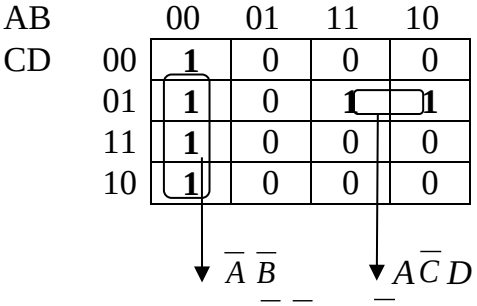
Tổ hợp 2:ABC 011

Khi gom 2 ô này ta loại bỏ biến C và giữ lại biến AB.Vì A có giá trị là 0 và B có giá trị là 1 nên tổ hợp này sẽ được biểu diễn là $\overline{A}B$

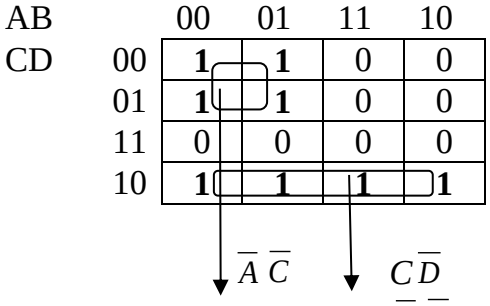
VD 1: Cho hàm Boole có bảng giá trị như sau.Rút gọn bằng bìa K



VD 2: Cho hàm Boole có bảng giá trị như sau.Rút gọn bằng bìa K

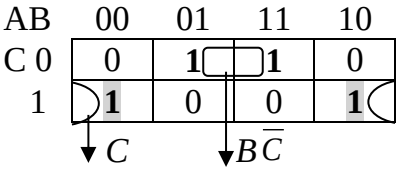


F(A,B,C,D)= $\overline{A}\,\overline{B} + A\overline{C}\,D$

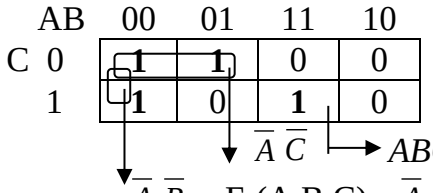


F(A,B,C,D)= $\overline{A}\,\overline{C} + C\overline{D}$

VD3:Rút gọn hàm Boole $F_1(A,B,C)=\sum(1, 2, 5,6)$, $F_2(A,B,C)=\sum(0,1, 2, 7)$



$F_1(A,B,C)=C + B\overline{C}$



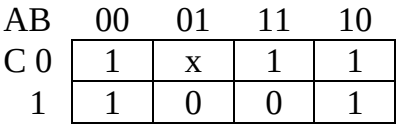
$F_2(A,B,C)= \overline{A}\,\overline{C} + \overline{A}\,\overline{B} + ABC$

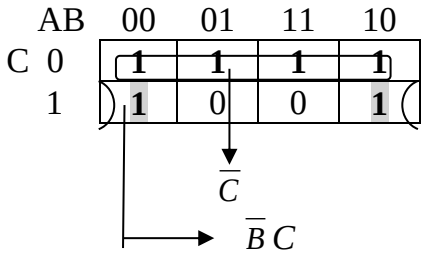
b. Tùy định (don't care).Thường ký hiệu là d(vị trí của ô)

VD:Cho hàm Boole $F(A,B,C)=\Sigma(0,1,4,5,6)+d_2$ hoặc là $F(A,B,C)=)=\Sigma(0,1,4,5,6)+d(2)$

Có nghĩa là khi biểu diễn bằng bìa K ta có thể cho ô thứ 2 là 0 hoặc 1 tùy ý sao cho có lợi nhất khi rút gọn.

Trong bìa K ta có thể dùng dấu x cho ô tùy định.Nhìn vào bìa K ta thấy nếu chọn tùy định là 1 thì rút gọn sẽ tối ưu.



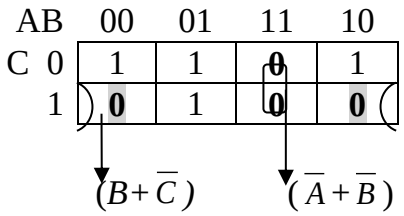


$F_2(A,B,C)= \overline{C} + \overline{B}\,C$

c. Đơn giản hóa theo dạng chuẩn 2

Phương pháp:Vẫn thực hiện tương tự như dạng chuẩn 1 nhưng khi gom các ô kế cận ta gom những ô có ký hiệu là 0 .Mỗi số hạng là một tổng.Kết quả cuối cùng là tích của các tổng đó.Khi liên kết thì ta chú ý các biến có giá trị là 0 thì là chính nó và có giá trị là 1 thì sẽ lấy bù (đảo).

VD:Rút gọn dạng chuẩn 2 hàm $F(A,B,C)= \Pi(0, 2, 3, 6)$.



F(A,B,C)= $(B+\overline{C}).(\overline{A}+\overline{B})$

*** Một số phương pháp thực hiện hàm Boole bằng các cấu trúc cho trước**

T 2 T 1	AND	OR	NAND	NOR
AND	x	Chuẩn 1	x	-Chuẩn 2 -Lấy bù hàm F 2 lần.Ap dụng Demorgan
OR	Chuẩn 2	x	-Chuẩn 1 -Lấy bù F 2 lần.Ap dụng D	x
NAND	-Chuẩn 2 -Lấy bù các thành phần.Ap dụng D	x	-Chuẩn 1 -Lấy bù F 2 lần.Ap dụng D	x
NOR	x	-Chuẩn 1 -Lấy bù các thành phần.Ap dụng D	x	-Chuẩn 2 -Lấy bù hàm F 2 lần.Ap dụng Demorgan

Dựa vào bảng trên ta áp dụng sẽ giải quyết được các bài toán.Các ô đánh dấu x sẽ không thực hiện được cấu trúc dạng đó.

VD:Cho hàm F(A,B,C)= $ABC+AB\overline{C}+\overline{A}\,\overline{B}$.Dùng cấu trúc OR -NAND thực hiện hàm trên

Thực hiện:

Bước 1: Đưa về dạng chuẩn 1 (bài toán đã cho sẵn).

Bước 2:Lấy bù 2 lần

F(A,B,C)= $ABC+AB\overline{C}+\overline{A}\,\overline{B}=\overline{\overline{ABC+AB\overline{C}+\overline{A}\,\overline{B}}}=\overline{\overline{ABC}.\overline{AB\overline{C}}\;\overline{\overline{A}\,\overline{B}}}$

$=\overline{(\overline{A}+\overline{B}+\overline{C}).(\overline{A}+\overline{B}+C).(A+B)}$

BÀI TẬP CHƯƠNG 2

1. Rút gọn hàm Boole bằng phương pháp đại số

a. y= $A\overline{B}D + A\overline{B}\overline{D}$

b. z= $(\overline{A} + B)(A + B)$

c. y= $ACD + \overline{A}BCD$

d. y= $A\overline{C} + AB\overline{C}$

e. y= $\overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D}$

f. y= $\overline{(\overline{A} + C)(B + \overline{\overline{D}})}$

2. Rút gọn hàm sau bằng bìa K

a. F= $ABC + \overline{ABC} + \overline{A}B$

b. F= $ABCD + \overline{ABC} + \overline{A}BD + \overline{AB}$

c. F(A,B,C,D)= $\sum(1,4,5,7,10,15)$

d. F(A,B,C,D)= $\sum(1,4,5,7,10,15) + d_0$

e. F(A,B,C,D)= $\sum(0,1,2,4,5,7,10,11)$

f. F(A,B,C)= $\prod(0,3,4,7)$

g. F(A,B,C,D)= $\prod(0,3,4,7,10,13) + d_5$

h. F(A,B,C,D= $(A + B + C + D)(\overline{A} + B + \overline{C} + D)(A + C + \overline{D})(\overline{A} + \overline{C})$

3. Cho hàm Boole F(A,B,C,D)= $\sum(1,4,5,7,10,15)$

a.Thực hiện hàm dùng cấu trúc NAND-NOR

b.Thực hiện hàm dùng cấu trúc NOR-NOR

c.Thực hiện hàm dùng cấu trúc NAND-NAND

d.Thực hiện hàm dùng cấu trúc OR-NOR

e.Thực hiện hàm dùng cấu trúc AND-OR

f. Thực hiện hàm dùng cấu trúc AND-NOR

g.Thực hiện hàm dùng toàn cổng NAND

h.Thực hiện hàm dùng toàn cổng NOR

i.Thực hiện hàm dùng toàn cổng NAND 2 ngõ vào

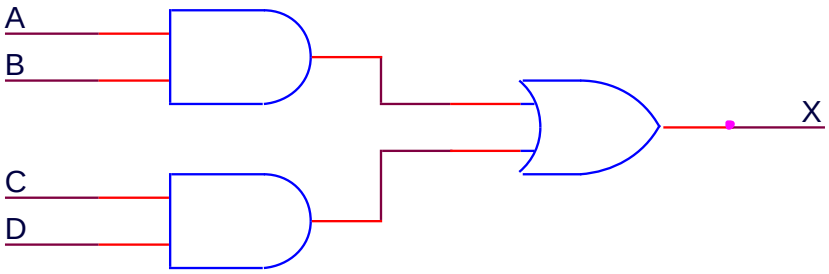
j.Thực hiện hàm dùng toàn cổng NOR 2 ngõ vào

4. Cho Z= $\overline{A} + \overline{B} + C$ Dùng cổng NAND và cổng đảo biểu diễn hàm trên

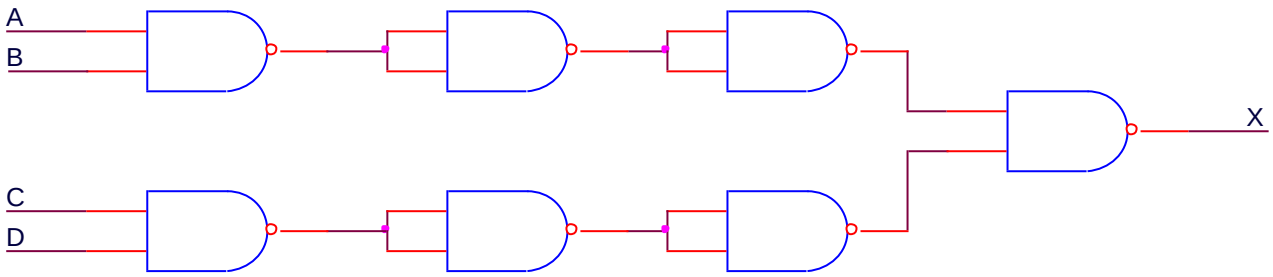
5. Cho Z= $\overline{ABC}$ Dùng cổng NOR và cổng đảo biểu diễn hàm trên

6. Xác định biểu thức ở ngõ ra của các mạch sau

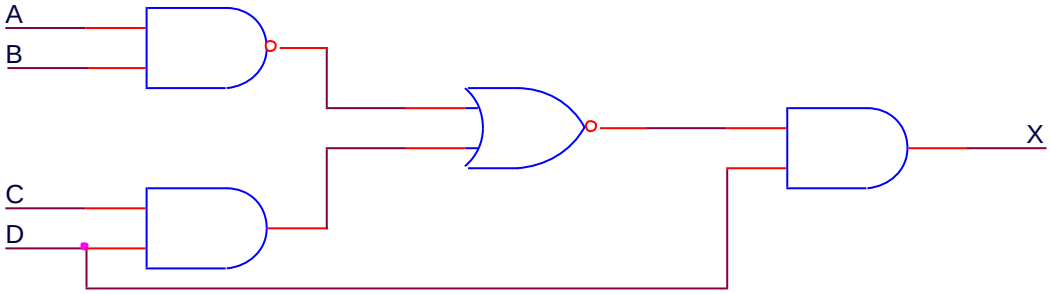
a.



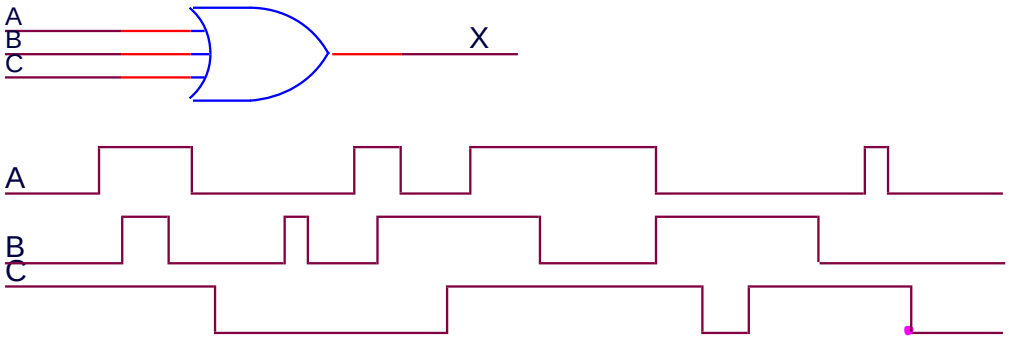
b.



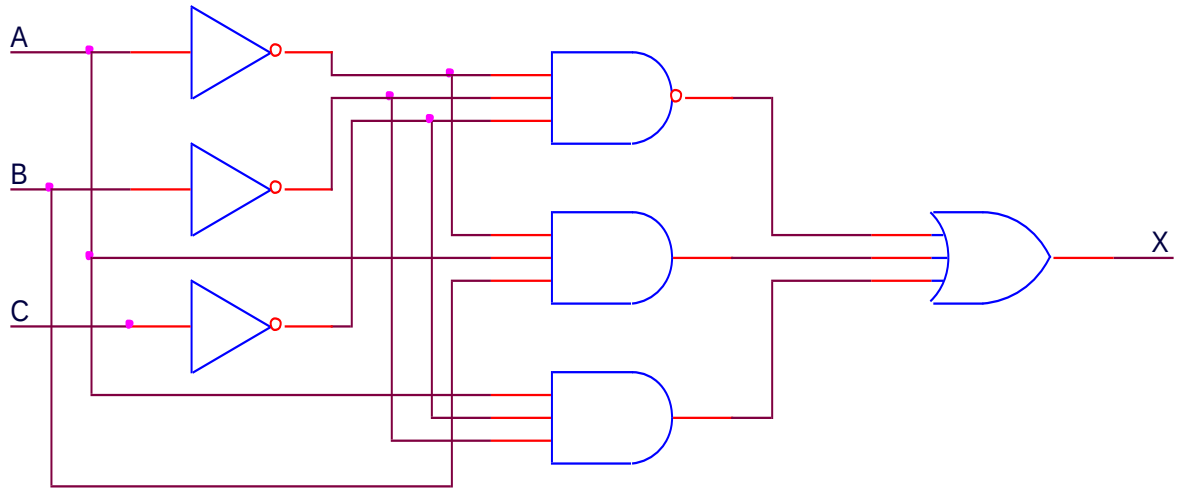
d.



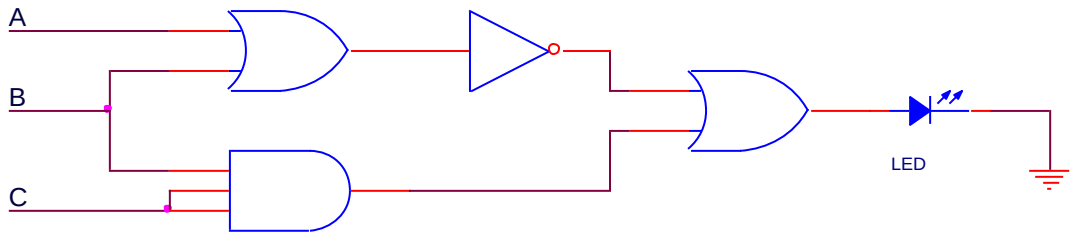
7. Vẽ dạng sóng ngõ ra X theo tín hiệu vào A,B,C



8. Xác định ngõ ra X



9. Với những giá trị nào của A,B,C thì LED sáng



CHƯƠNG 3: MẠCH TỔ HỢP

Giới thiệu: *Trong chương này, giới thiệu về các loại mạch tổ hợp.*

Mục tiêu:

- Hiểu các mạch mã hóa và mạch giải mã; mạch phân kên – dồn kênh.

- Hiểu được nguyên lý làm việc của các mạch mã hóa và mạch giải mã; mạch phân kên – dồn kênh.

- Rèn luyện tính kiên trì, cẩn thận và tư duy logic.

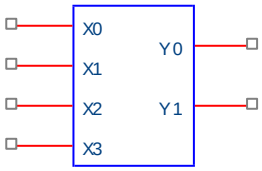
Nội dung chính:

3.1. Mạch mã hóa (Encoder):

3.1.1. Mạch mã hóa 8 sang 3

Là mạch có 2^n ngõ vào và n ngõ ra

Xét Encoder 4→ 2



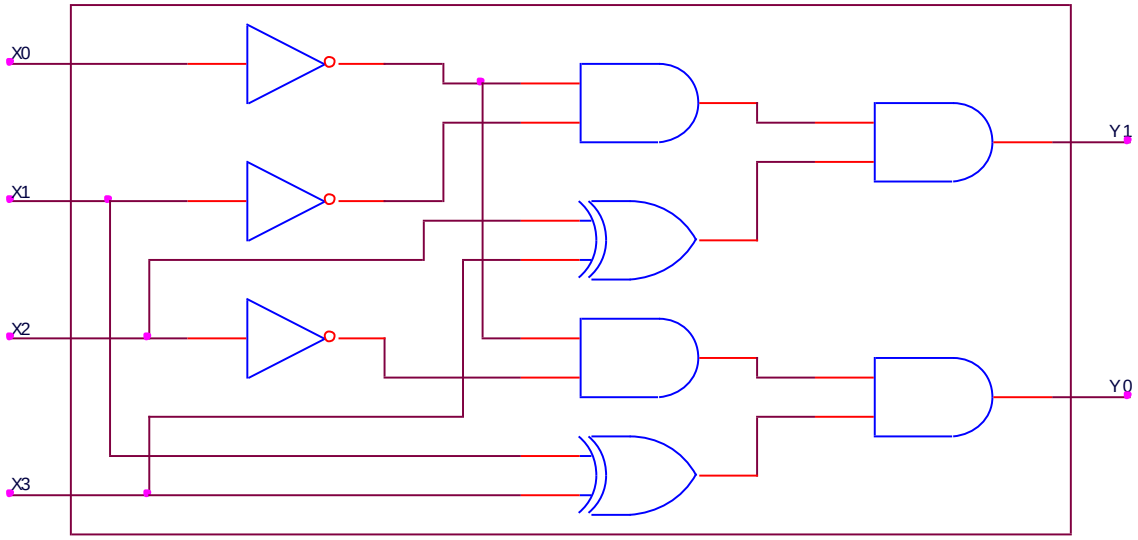
X3	X2	X1	X0	Y1	Y0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

3.1.2. Mạch mã hóa thập phân sang BCD (10 sang 4)

Từ bảng giá trị trên ta thấy nếu đưa tín hiệu từ mạch giải mã đến mạch mã hóa thì dữ liệu sẽ đúng như trạng thái ban đầu.Từ đó người ta có thể mã hóa dữ liệu trước khi đưa lên đường tín hiệu.

Sơ đồ tương đương của Encoder 4→ 2

Từ bảng giá trị trên ta có:Y0= $\overline{X2}\,\overline{X0}(X3 \oplus X1)$;Y1= $\overline{X1}\,\overline{X0}(X3 \oplus X2)$



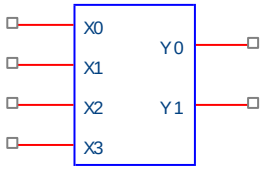
Cũng tương tự như Decoder ta cũng có Encoder có ngõ vào,ngõ ra tích cực mức cao hoặc thấp.Tùy từng trường hợp cụ thể mà ta chọn cho phù hợp.

3.1.3. Mạch mã hóa có ưu tiên:

Khi có 2 hay nhiều ngõ vào cùng ở mức tích cực thì ngõ ra có thể có thể sẽ có 2 giá trị khác nhau , chính vì vậy người ta qui định bộ mã hóa có ưu tiên .

Khi đó nếu có 2 tín hiệu cùng ở mức tích cực thì chỉ có ngõ vào có độ ưu tiên cao hơn mới cho tác động tại ngõ ra

Cho Encoder như hình vẽ



X3	X2	X1	X0	Y1	Y0
x	x	x	1	0	0
x	x	1	0	0	1
x	1	0	0	1	0
1	0	0	0	1	1

Độ ưu tiên giảm dần từ X0 đến X3.

3.1.4. Khảo sát IC 74147

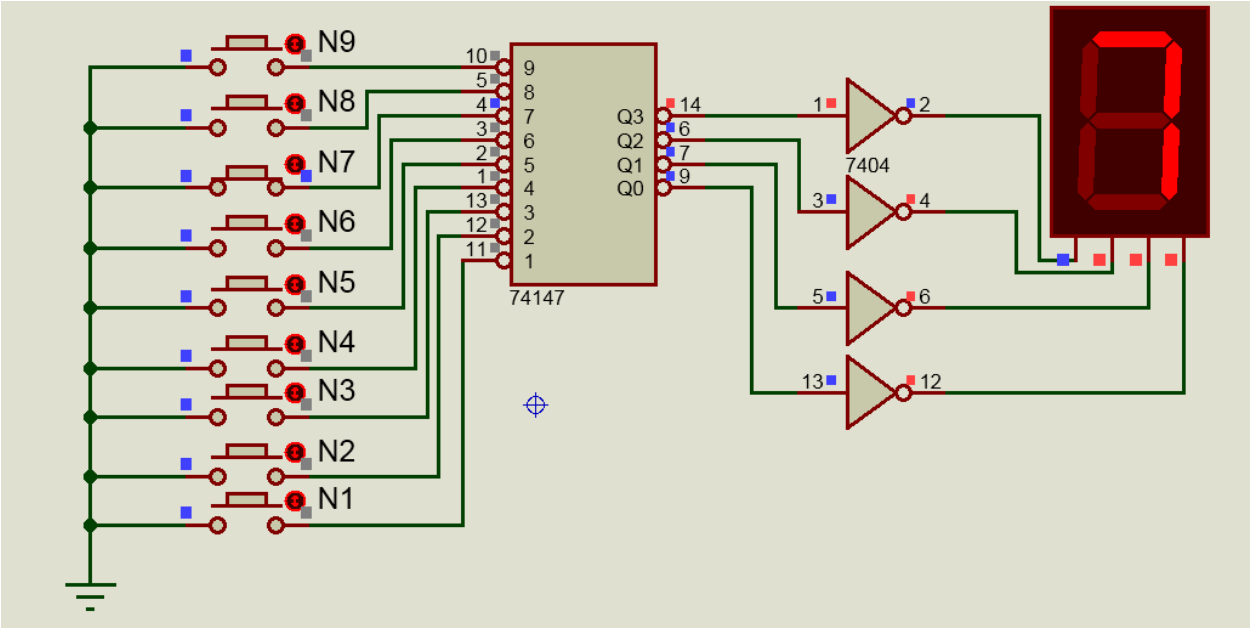
FUNCTION TABLE – '147, 'LS147

INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

H = high logic level, L = low logic level, X = irrelevant

Hình 5.1 : sơ đồ chân và bảng thực trị của mạch

mã hóa các chữ số từ 0 đến 9 (74147)



Hình: Mạch mã hóa 10 → 4 .

3.2. Mạch giải mã (Decoder)

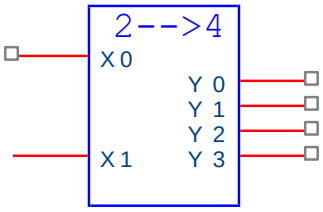
Decoder là mạch chuyển đổi N bit ở đầu vào thành M ngõ ra .Mỗi ngõ ra được chọn (tích cực) tương ứng với một tổ hợp ở đầu vào.

Nếu có N ngõ vào tức có 2^N tổ hợp.ứng với mỗi tổ hợp ở đầu vào sẽ có một ngọ ra ở mức Logic cao còn tất cả các ngõ ra khác sẽ ở mức Logic thấp

Tuy nhiên có những Decoder được thiết kế ngược lại tức ngõ nào tích cực thì ngõ đó có mức logic thấp còn các ngõ còn lại ở mức cao.

3.2.1. Mạch giải mã từ 2 sang 4; 3 sang 8; 4 sang 16

Bảng giá trị



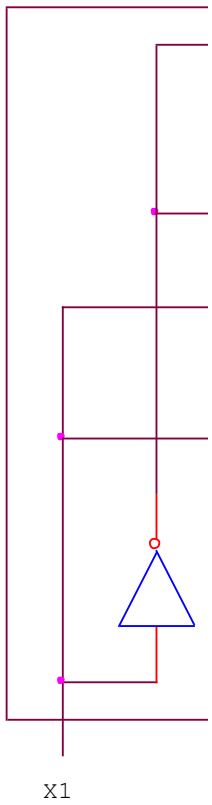
Từ bảng giá trị trên ta có

Y0= $\overline{X0}\ \overline{X1}$; Y1= $X0\ \overline{X1}$;

X0	X1	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

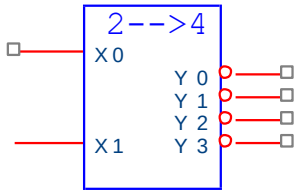
Y2= $\overline{X0}\ X1$; Y0= $X0\ X1$;

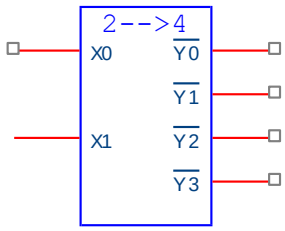
Sơ đồ tương đương



Decoder 2→ 4 có ngõ ra tích cực mức cao

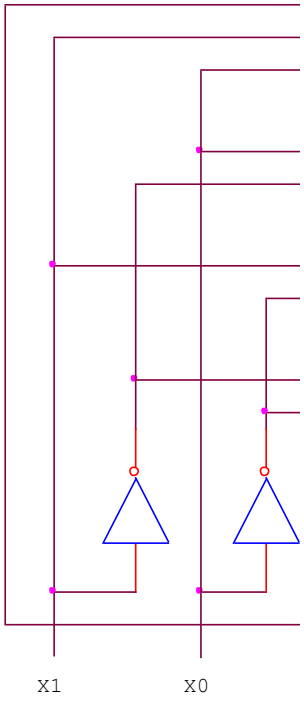
3.2.2. Mạch giải mã từ BCD sang thập phân





Từ bảng giá trị trên ta có

Y0=X0+X1;Y1=X0+ $\overline{X1}$;Y2= $\overline{X0}$ +X1;Y3= $\overline{X0}$ + $\overline{X1}$



X0	X1	Y3	Y2	Y1	Y0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

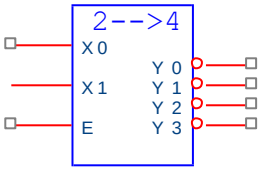
Decoder 2→ 4 có ngõ ra tích cực mức thấp

Nhìn vào bảng giá trị của Decoder ta thấy có một điểm bất lợi là tại một thời điểm phải có một ngõ ra được chọn.Nếu như ta không muốn chọn ngõ nào thì Decoder loại này

không thực hiện được.Xuất phát từ nguyên nhân đó người ta còn tích hợp một loại Decoder có ngõ vào cho phép gọi là Enable (E).

Xét Decoder 2→ 4 có ngõ ra tích cực mức thấp có ngõ vào Enable tích cực mức cao

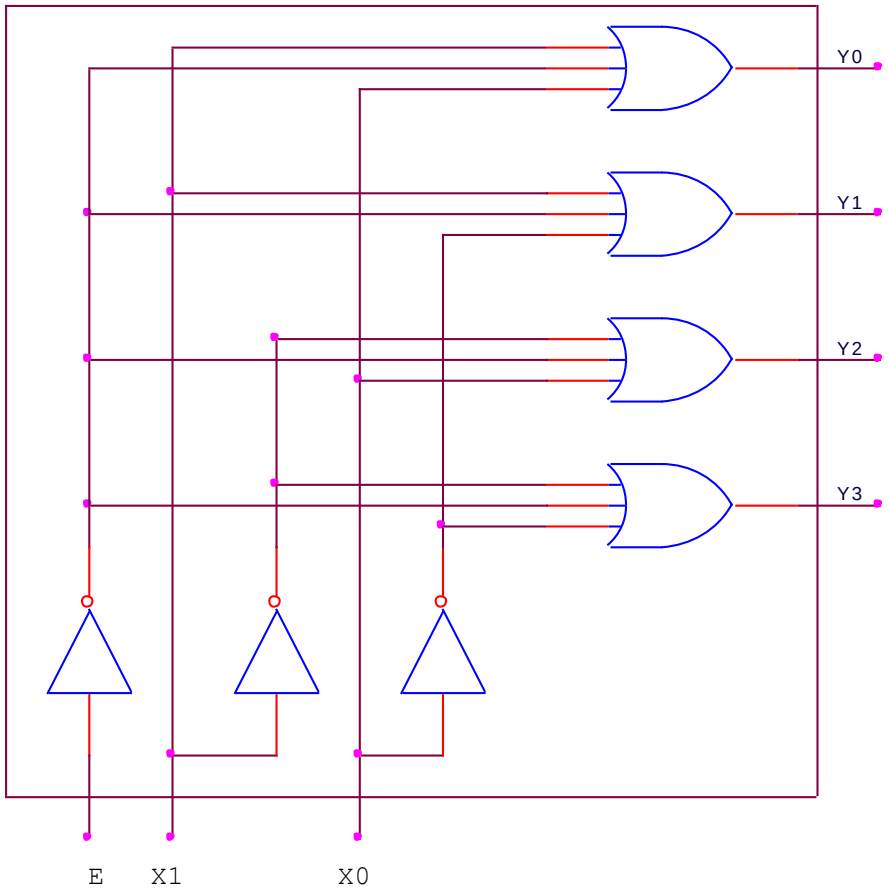
Bảng giá trị



E	X1	X0	Y3	Y2	Y1	Y0
1	0	0	1	1	1	0
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1
0	x	x	1	1	1	1

3.2.3. Mạch giải mã từ BCD sang mã 7 đoạn

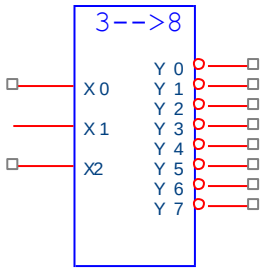
Tương tự như các cách làm trước ta cũng sẽ tìm được sơ đồ tương đương của Decoder trong trường hợp có Enable.



***Nhận xét**:tại một thời điểm có một ngõ ra tích cực .Ngõ ra đó tương ứng với số thập phân của tổ hợp nhị phân đó

Ví dụ:Nếu ngõ vào có tổ hợp là 10 thì ngõ ra được chọn là Y2 vì $10_{[B]}=2_{[D]}$.

Xét Decoder 3→ 8 có ngõ ra tích cực mức thấp



X2	X1	X0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1
1	1	0	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1

3.2.4. Ghép các mạch giải mã

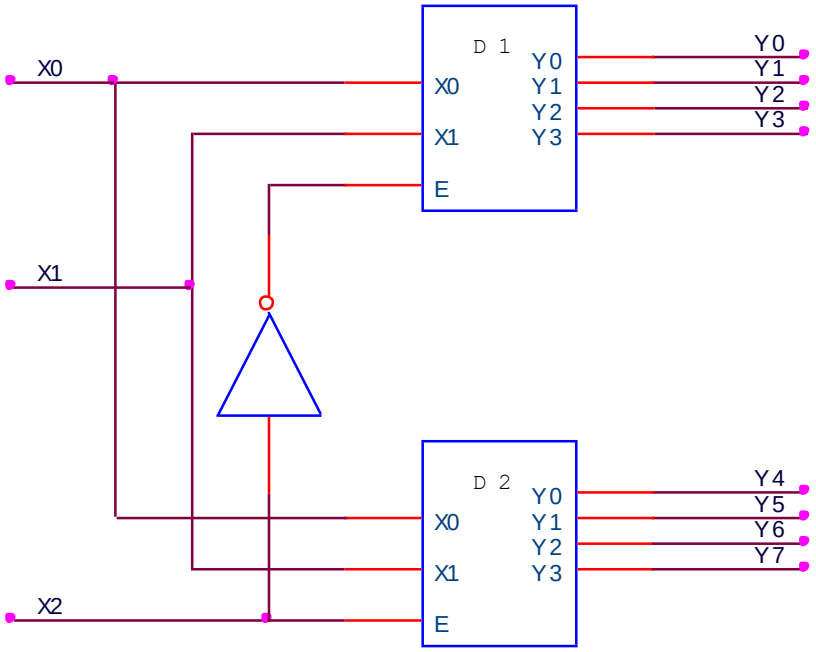
Để ghép các Decoder với nhau thì các decoder đó phải có ngõ vào E.

Khi ghép 2 Decoder 2→ 4 ta sẽ được 1 Decoder 3→ 8

Khi ghép 2 Decoder 3→ 8 ta sẽ được 1 Decoder 4→ 16

Khi ghép 2 Decoder n→ 2^n ta sẽ được 1 Decoder n+1→ 2^{n+1}

Ví dụ:ghép 2 Decoder 2→ 4 thành 1 Decoder 3→ 8



Bảng giá trị khi ghép 2 Decoder

X2 X1 X0	Y7 Y6 Y5 Y4	Y3 Y2 Y1 Y0

0	0	0	0	0	0	0	0	0	0	1	
0	0	1	0	0	0	0	0	0	1	0	D 1 hoạt động
0	1	0	0	0	0	0	0	1	0	0	D 2 bị cấm
0	1	1	0	0	0	0	1	0	0	0	
1	0	0	0	0	0	1	0	0	0	0	
1	0	1	0	0	1	0	0	0	0	0	D 2 hoạt động
1	1	0	0	1	0	0	0	0	0	0	D 1 bị cấm
1	1	1	1	0	0	0	0	0	0	0	

Nhìn vào bảng giá trị trên ta thấy nếu ghép như trên thì 2 Decoder này hoàn toàn tương đương như 1 decoder $3\rightarrow 8$

Dùng Decoder thực hàm Boole

Ví dụ: Cho hàm Boole có F(A,B,C)=Σ(0,1,3,6).Dùng Decoder thực hiện hàm trên thay cho các cổng Logic khác.

Ta thấy Decoder thực sự là các cổng Logic được tích hợp thành.thực hiện hàm Boole dùng Decoder thực chất là ta tận dụng các cổng Logic có sẵn trong Decoder để thực hiện hàm Boole đó.

Để thực hiện hàm Boole dùng Decoder ta lần lượt theo các bước sau:

- Vì có 3 biến nên ta dùng Decoder $3\rightarrow 8$.Tổng quát nếu hàm có n biến thì dùng Decoder $n\rightarrow 2^n$

- Lập bảng giá trị hàm Boole.Theo như ví dụ trên ta có bảng giá trị như sau:

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Ta có thể biểu diễn hàm Boole trên bằng dạng chuẩn 1 như sau:

F(A,B,C)= Σ(0,1,3,6)= $\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}BC + AB\overline{C}$

Bảng giá trị của decoder 3→ 8 như sau:

X2	X1	X0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

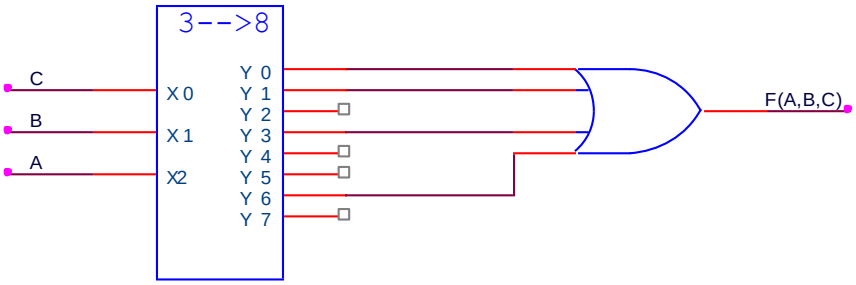
Ta có:

Y0= $\overline{A}\overline{B}\overline{C}$;Y1= $\overline{A}\overline{B}C$;Y2= $\overline{A}B\overline{C}$;Y3= $\overline{A}BC$;Y4= $A\overline{B}\overline{C}$;Y5= $A\overline{B}C$;Y6= $AB\overline{C}$;Y7= ABC

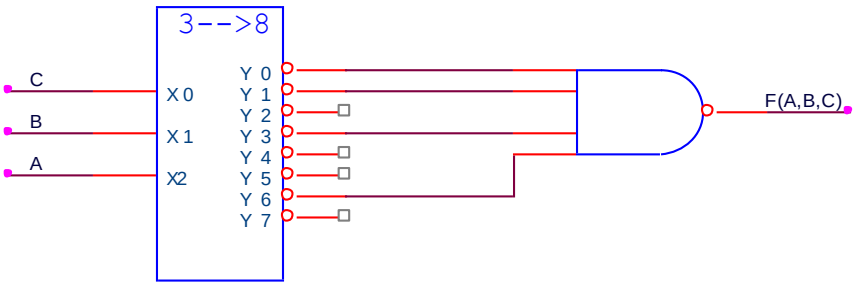
So sánh 2 kết quả này ta thấy có thể biểu diễn hàm Boole như sau:

F(A,B,C)= Σ(0,1,3,6)= $\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}BC + AB\overline{C}$ =Y0+Y1+Y3+Y6 (của Decoder).

Sơ đồ thực hiện dùng Decoder như sau:



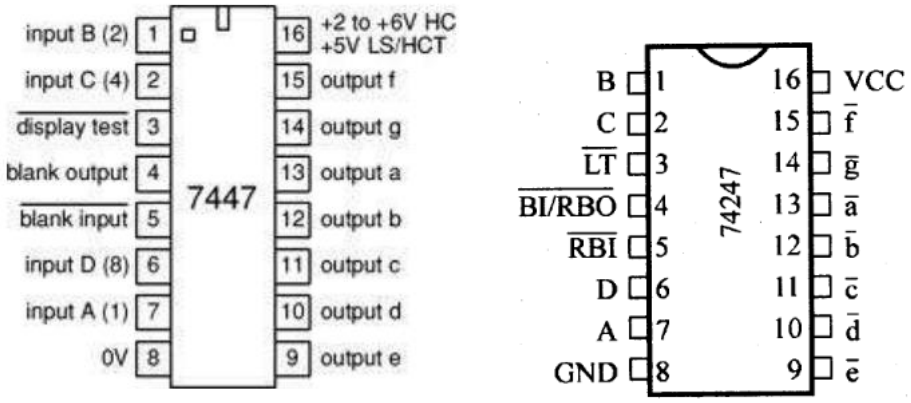
Nếu sử dụng Deoder có ngõ ra tích cực mức thấp thì cũng cách làm tương tự như trên ta sẽ có mạch như sau:



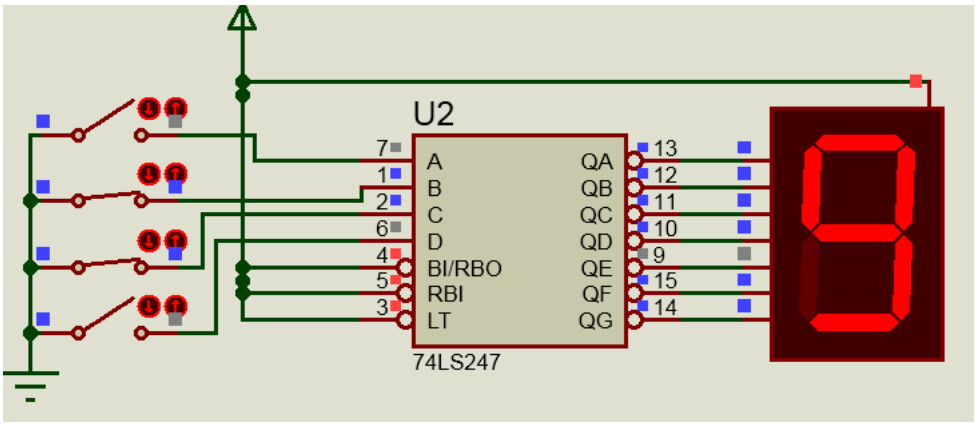
Ta có thể có nhiều cách thực hiện hàm Boole dùng Decoder nhưng tất cả đều cho cùng kết quả.

3.2.5. Khảo sát IC mã hóa – giải mã 74147, 74151, 74153, 7447, 74247, 4543, 4017

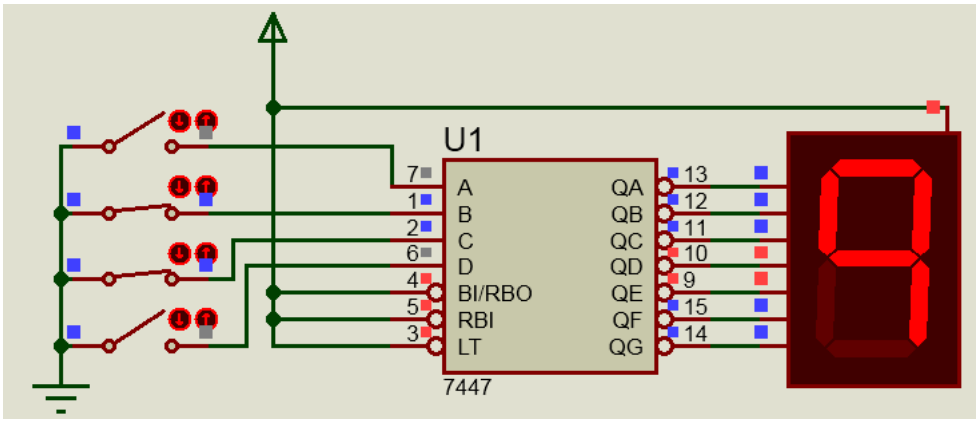
*** IC giải mã 7447, 4017**



Hình: Sơ đồ chân của IC7447 VÀ IC 74247



Hình: Mạch giải mã 7SEG 74247

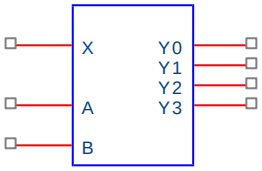


Hình: Mạch giải mã 7SEG 7447

3.3. Mạch phân kênh 1→4:

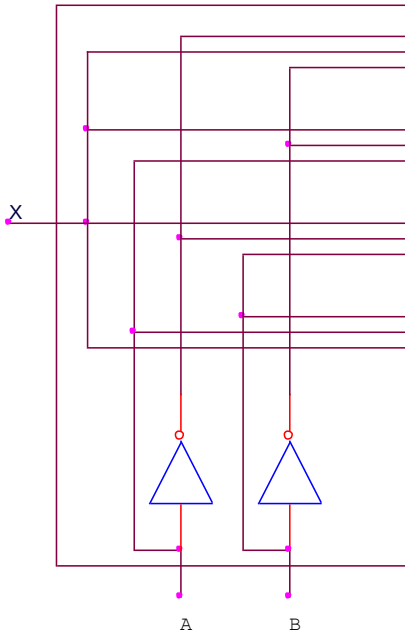
DEMUX là mạch tổ hợp có 1 ngõ vào n ngõ điều khiển và 2^n ngõ ra.Nếu dữ liệu từ MUX đưa đến DEMUX thì dữ liệu sẽ được phục hồi đúng trạng thái ban đầu.

Xét DEMUX 1 → 4



Từ bảng trên ta có $Y0 = X\overline{A}\overline{B}; Y1 = XA\overline{B}; Y2 = X\overline{A}B; Y3 = XAB$

Sơ đồ mạch



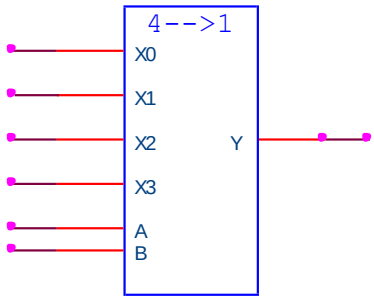
A	B	Y3	Y2	Y1	Y0
0	0	0	0	0	X
0	1	0	0	X	0
1	0	0	X	0	0
1	1	X	0	0	0

3.4. Mạch dồn kênh

Mạch dồn kênh hay còn gọi là MUX là mạch có 2^n dữ liệu (Data),n ngõ vào điều khiển (Selects) và có một ngõ ra

3.4.1. Mạch dồn kênh 4 → 1

A	B	Y
0	0	X0
0	1	X1

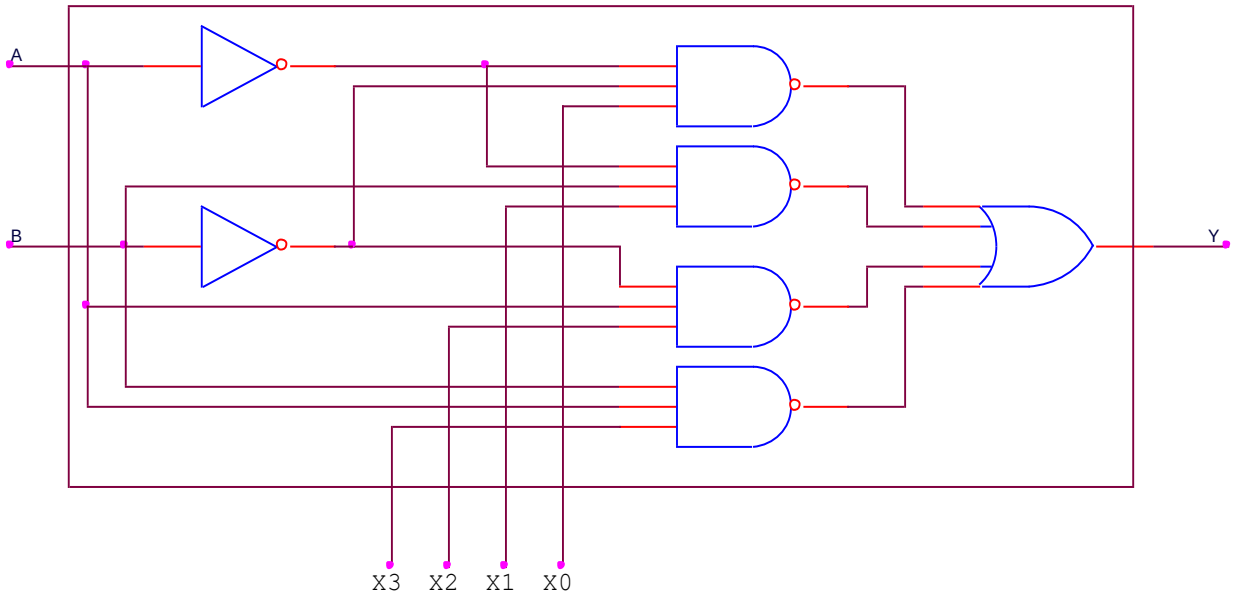


1	0	X2
1	1	X3

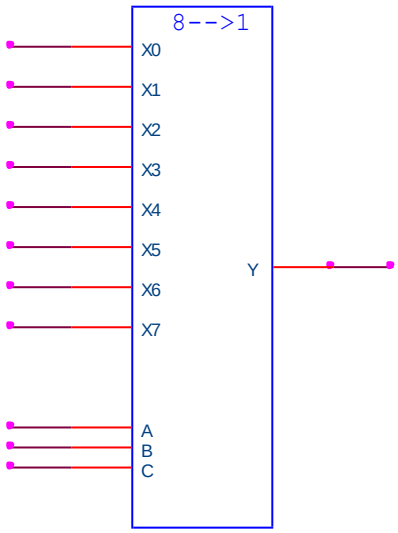
MUX có 4 ngõ vào dữ liệu là X0,X1,X2,X3 và có 2 ngõ vào điều khiển là A,B.

Từ bảng giá trị ta có Y= $X0\overline{A}\overline{B} + X1\overline{A}B + X2A\overline{B} + X3AB$

Sơ đồ mạch tương đương



Xét MUX 8→ 1 (2^3→ 1)



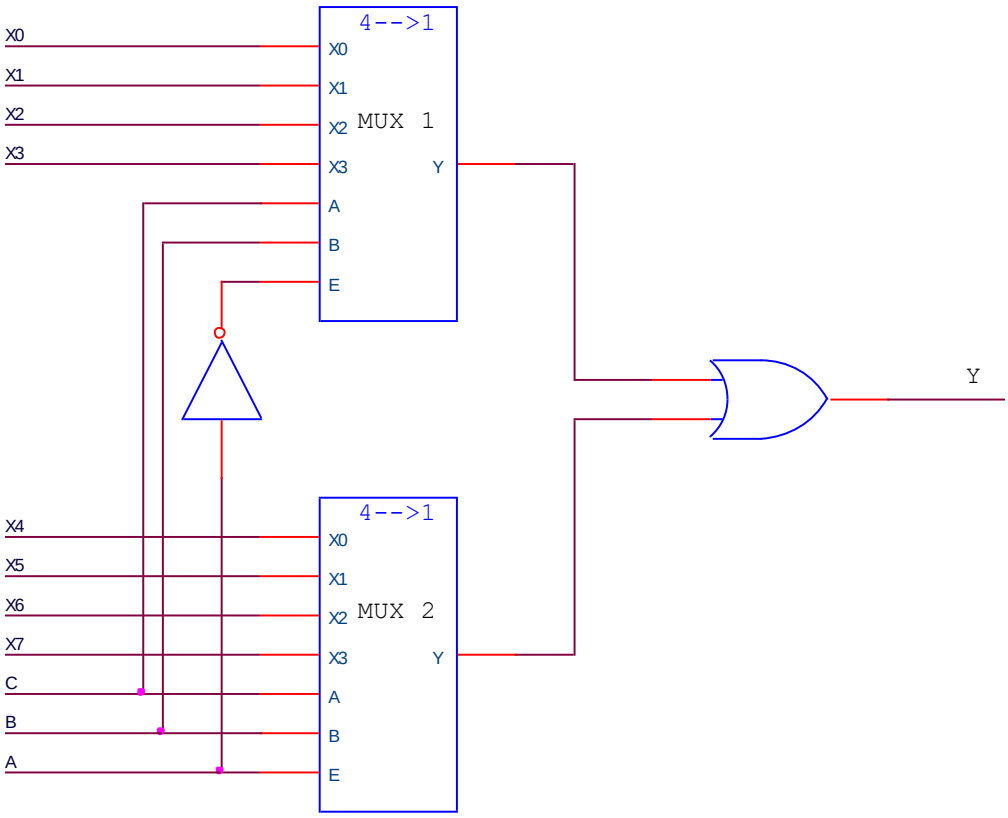
A	B	C	Y
0	0	0	X0
0	0	1	X1
0	1	0	X2
0	1	1	X3
1	0	0	X4
1	0	1	X5
1	1	0	X6
1	1	1	X7

Tương tự như trên ta có thể vẽ sơ đồ tương đương của MUX 8→ 1

3.4.2. Ghép các mạch dồn kênh

Để ghép 2 MUX với nhau thì 2 MUX phải có Enable

Ghép 2 MUX 4→ 1 thành 1 MUX 8→ 1.Thực hiện ghép như sau



*** Chú ý :Khi ghép như trên ta phải chú ý rằng A là MSB và B là LSB**

Bảng giá trị

A C B	Y	
0 0 0	X0	
0 0 0	X1	MUX 1 hoạt động
0 0 0	X2	MUX 2 bị cấm
0 0 0	X3	
0 0 0	X4	MUX 2 hoạt động
0 0 0	X5	MUX 1 bị cấm
0 0 0	X6	

0 0 0	X7	

Bảng giá trị này hoàn toàn tương đương như bảng giá trị của MUX $8 \rightarrow 1$

d. Dùng MUX thực hiện hàm Boole

+ Dùng MUX $2^n \rightarrow 1$ thực hiện hàm Boole n biến

Cho hàm Boole F(A,B,C)= Σ(1,2,5,7).Dùng MUX thực hiện hàm trên

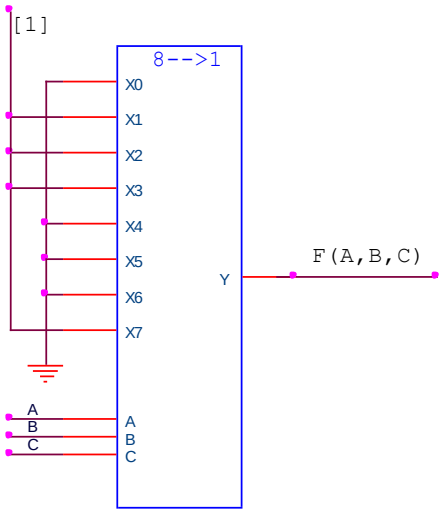
Hàm có 3 biến nên ta dùng MUX $8 \rightarrow 1$(có n biến thì dùng MUX $2^n \rightarrow 1$)

Bảng giá trị của hàm Boole Bảng giá trị của MUX

A B C	Y
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

A B C	Y
0 0 0	X0
0 0 1	X1
0 1 0	X2
0 1 1	X3
1 0 0	X4
1 0 1	X5
1 1 0	X6
1 1 1	X7

Nhìn vào 2 bảng giá trị trên ta thấy nếu cho X0=X3=X4=X6=0 và X1=X2=X3=X7=1 thì ngõ ra của MUX chính là hàm F(A,B,C).Sơ đồ mạch thực hiện như sau



+ Dùng MUX $2^n \rightarrow 1$ thực hiện hàm Boole n+1 biến

Thực hiện hàm F(A,B,C)= Σ(2,5,6,7) dùng MUX $4 \rightarrow 1$

Trước hết ta phải nhớ bảng sau:

2 giá trị liên tiếp của hàm Boole	**Bit LSB**

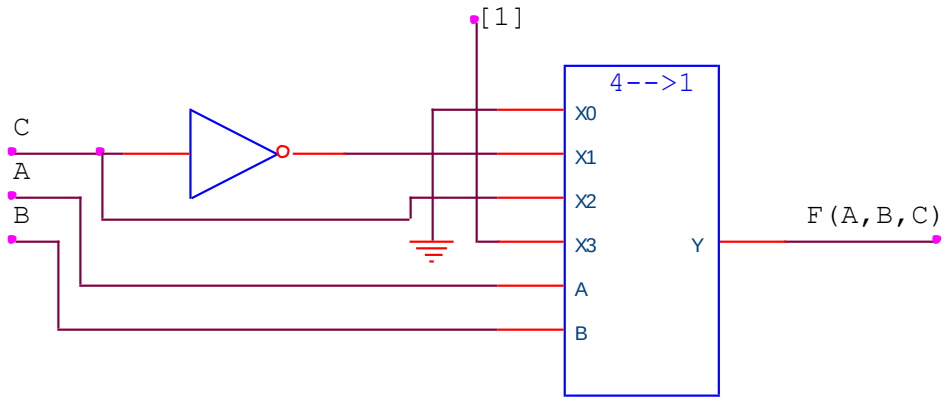
0 0	0
0 1	N_0
1 0	$\overline{N_0}$
1 1	1

Sau đó lập bảng giá trị của hàm Boole rồi dựa vào bảng trên ta ghi lại thành bảng như bên dưới.Trong đó X0,X1,X2,X3 là các ngõ data của MUX 4→1.Các biến cao đưa vào 2 ngõ điều khiển còn biến là LSB thì thực hiện từ bảng giá trị

Bảng giá trị

A B C	Y		
0 0 0	0	0	X0
0 0 1	0		
0 1 0	1	$\overline{C}$	X1
0 1 1	0		
1 0 0	0	C	X2
1 0 1	1		
1 1 0	1	1	X3
1 1 1	1		

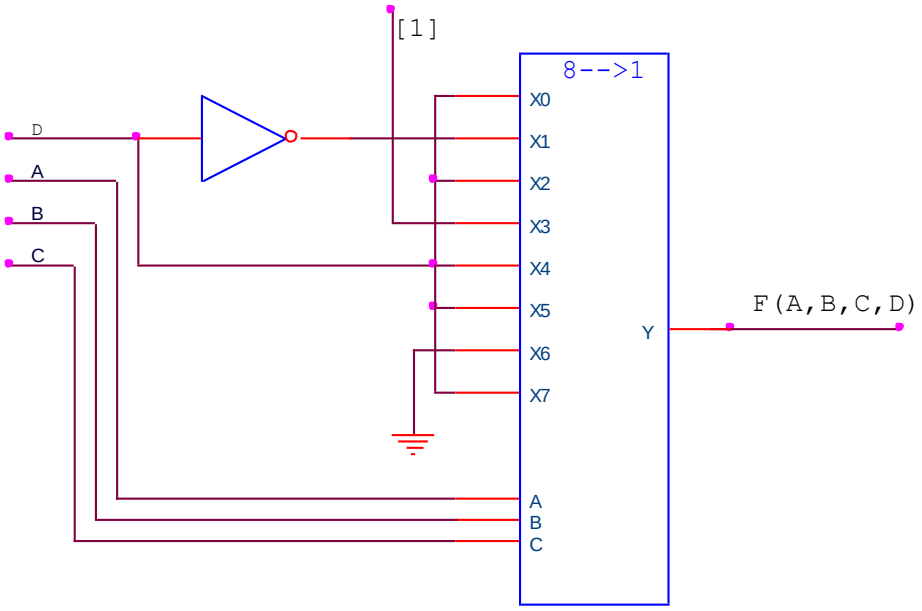
Sơ đồ mạch thực hiện



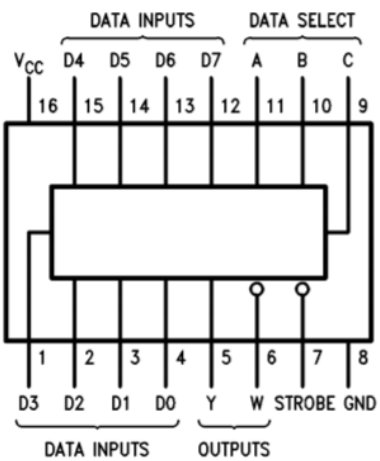
Xét một VD khác:Dùng MUX 8→1 thực hiện hàm Boole F(A,B,C,D)= Σ(1,2,5,6,7,9,11,15)

A B C D	F(A,B,C,D)		
0 0 0 0	0	D	X0
0 0 0 1	1		

0 0 1 0 0 0 1 1	1 0	$\overline{D}$	X1
0 1 0 0 0 1 0 1	0 1	D	X2
0 1 1 0 0 1 1 1	1 1	1	X3
1 0 0 0 1 0 0 1	0 1	D	X4
1 0 1 0 1 0 1 1	0 1	D	X5
1 1 0 0 1 1 0 1	0 0	0	X6
1 1 1 0 1 1 1 1	0 1	D	X7



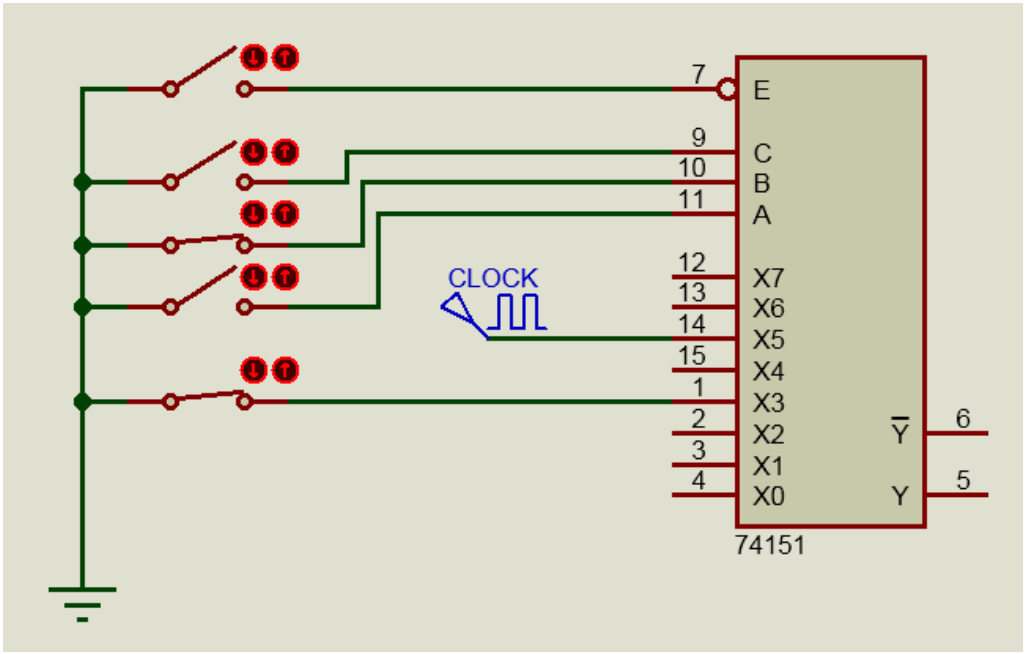
*** IC dồn kênh 74151**



Hình 6.1 Sơ đồ chân , các đầu vào đầu ra và bảng thực trị IC74151

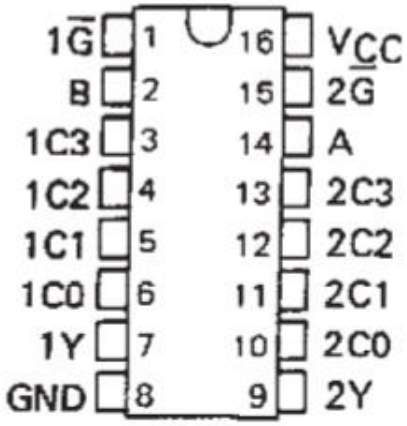
Inputs				Outputs	
Select			Strobe		
C	B	A	S	Y	W
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = High Level, L = Low Level, X = Don't Care
D0, D1 . . . D7 = the level of the respective D input



Hình: Mạch chọn 1 trong 8 nguồn tín hiệu số X0 đến X7 để xuất đi từ đầu ra Y và /Y

*** IC dồn kênh 74153**

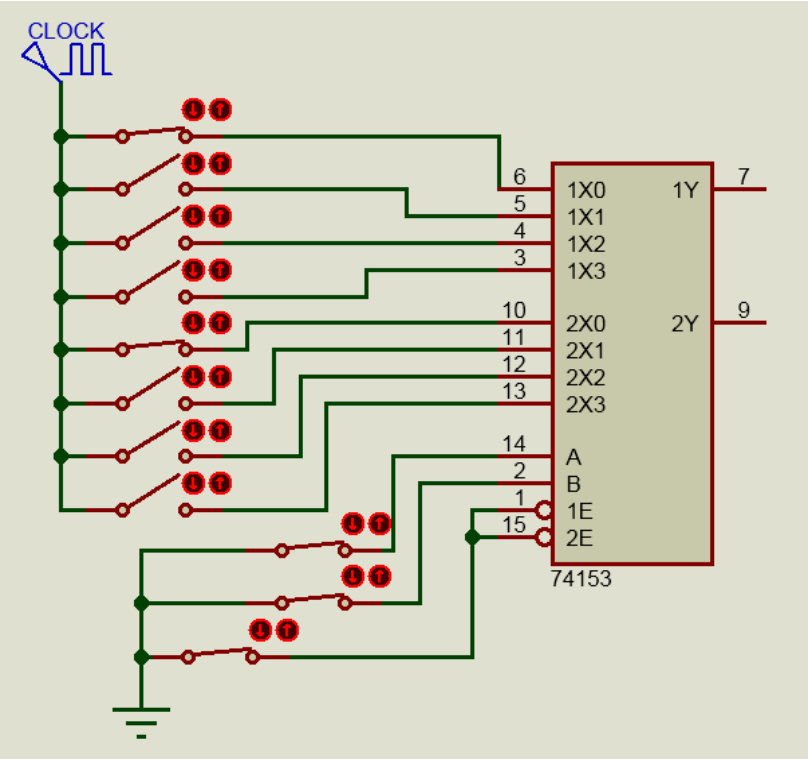


FUNCTION TABLE

SELECT INPUTS		DATA INPUTS				STROBE	OUTPUT
B	A	C0	C1	C2	C3	$\bar{G}$	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

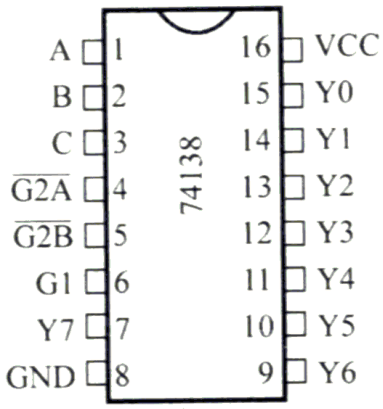
Select inputs A and B are common to both sections.
H = high level, L = low level, X = irrelevant

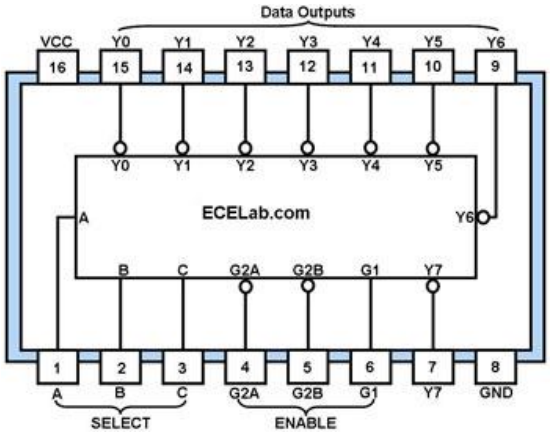
Hình: Sơ đồ chân và bảng sự thật của IC74153



Hình: Mạch đa hợp 4 → 1 hai kênh ra, 8 kênh vào

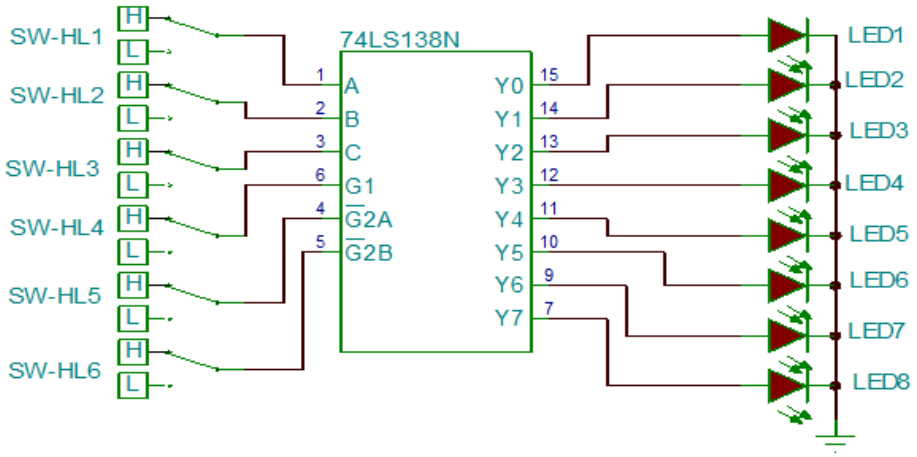
*** IC phân kênh 74138:**





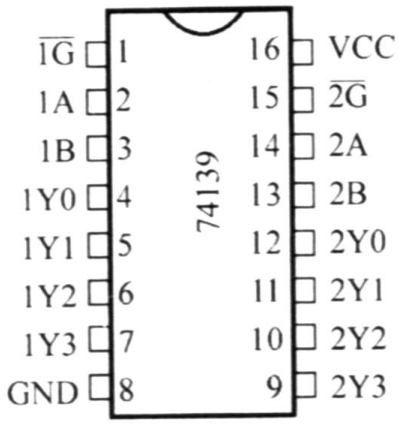
Đầu vào						Đầu ra							
G2A	**G2B**	**G1**	**A**	**B**	**C**	**Y_0**	**Y_1**	**Y_2**	**Y_3**	**Y_4**	**Y_5**	**Y_6**	**Y_7**
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	0	1	1	1	0	1	1	1	1
0	0	1	0	0	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

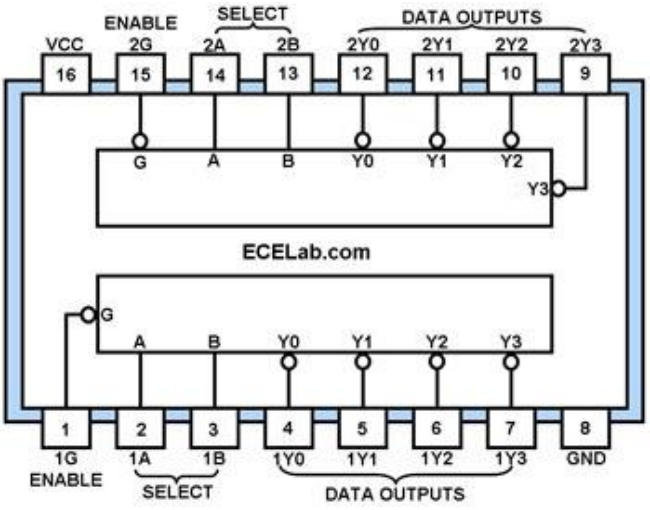
Hình: Sơ đồ chân, sơ đồ logic, bảng trạng thái IC 74LS138.



Hình 6.2 :Mạch kiểm tra IC 74LS138.

*** IC phân kênh 74139:**

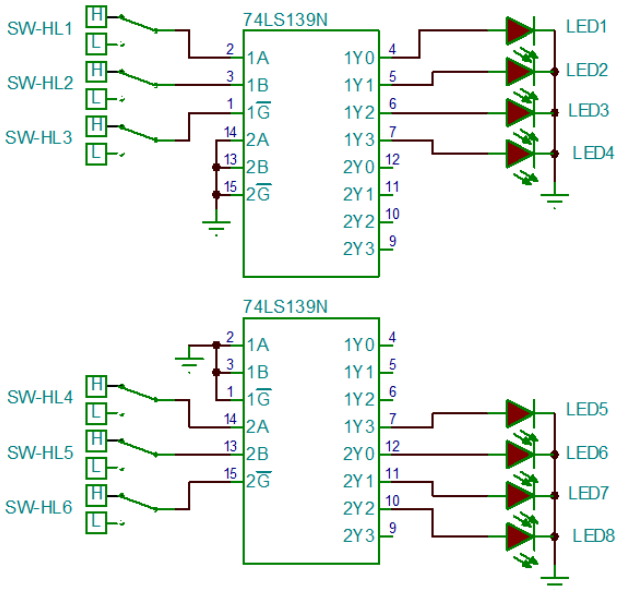




INPUTS			OUTPUTS			
G	**A**	**B**	**Y0**	**Y1**	**Y2**	**Y3**
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

Hình: Sơ đồ chân, sơ đồ logic, bảng trạng thái IC 74LS139.

6



Hình 6.7: kiểm tra IC giải mã 74LS139.

3.5. Thiết kế mạch tổ hợp

a. Phương pháp

+Xác định số ngõ vào-ra từ bài toán yêu cầu

+Lập bảng giá trị theo bài toán đặt ra

+Rút gọn các hàm Boole

+Dùng cổng Logic hoặc các mạch tổ hợp thực hiện hàm Boole

b. Một số VD về thiết kế mạch tổ hợp

VD1:Thiết kế một mạch tổ hợp thực hiện việc cộng 2 số nhị phân 1 bit có nhớ.Còn gọi là mạch cộng bán phần=Haft-Adder=HA

-Trước hết ta xác định số ngõ vào-ra của mạch tổ hợp:Theo đề bài thì mạch có 2 ngõ vào là 2 số nhị phân và 2 ngõ ra là tổng 2 số nhị phân và số nhớ

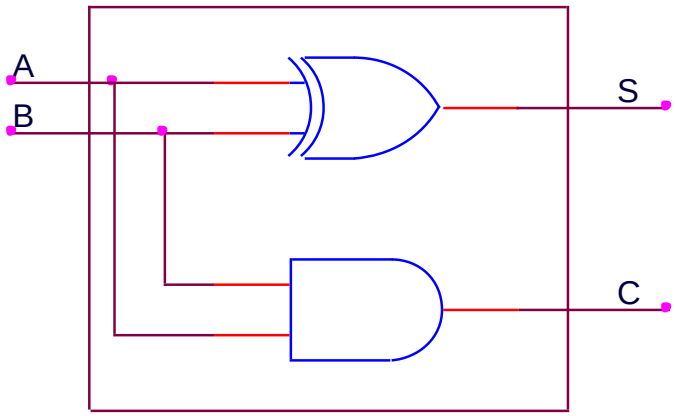
-Bảng giá trị

A B	S C
0 0	0 0
0 1	1 0
1 0	1 0
1 1	0 1

Trong đó S(sum) là tổng 2 số nhị phân và C(carry) là số nhớ

-Rút gọn ta được:S= $A \oplus B$;C=AB

-Thực hiện mạch



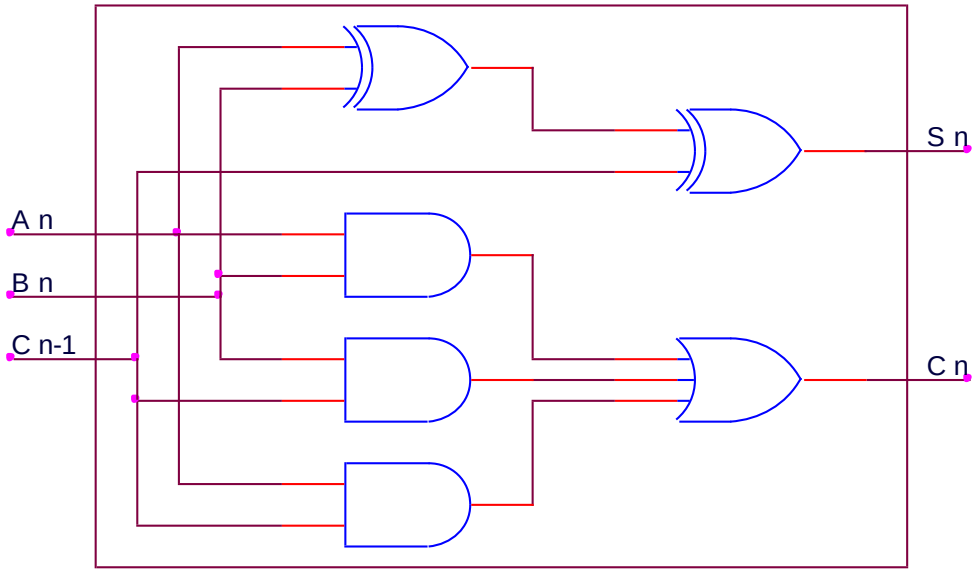
Mạch cộng 1 bit có thành phần nhớ trước đó gọi là mạch cộng toàn phần hay Full-Adder=FA

A_n B_n C_{n-1}	S_n C_n

0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S_n = C_{n-1} \oplus (A_n \oplus B_n); C_n = A_n B_n + B_n C_{n-1} + A_n C_{n-1}$

Sơ đồ mạch



VD 2:Thiết kế mạch tổ hợp thực hiện việc nhân 2 số nhị phân 2 bit

-Mạch có 4 ngõ vào 4 ngõ ra vì tích của 2 số nhị phân khi nhân nhau lớn nhất là 1001.

- Bảng giá trị

A1	B1	A2	B2	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0

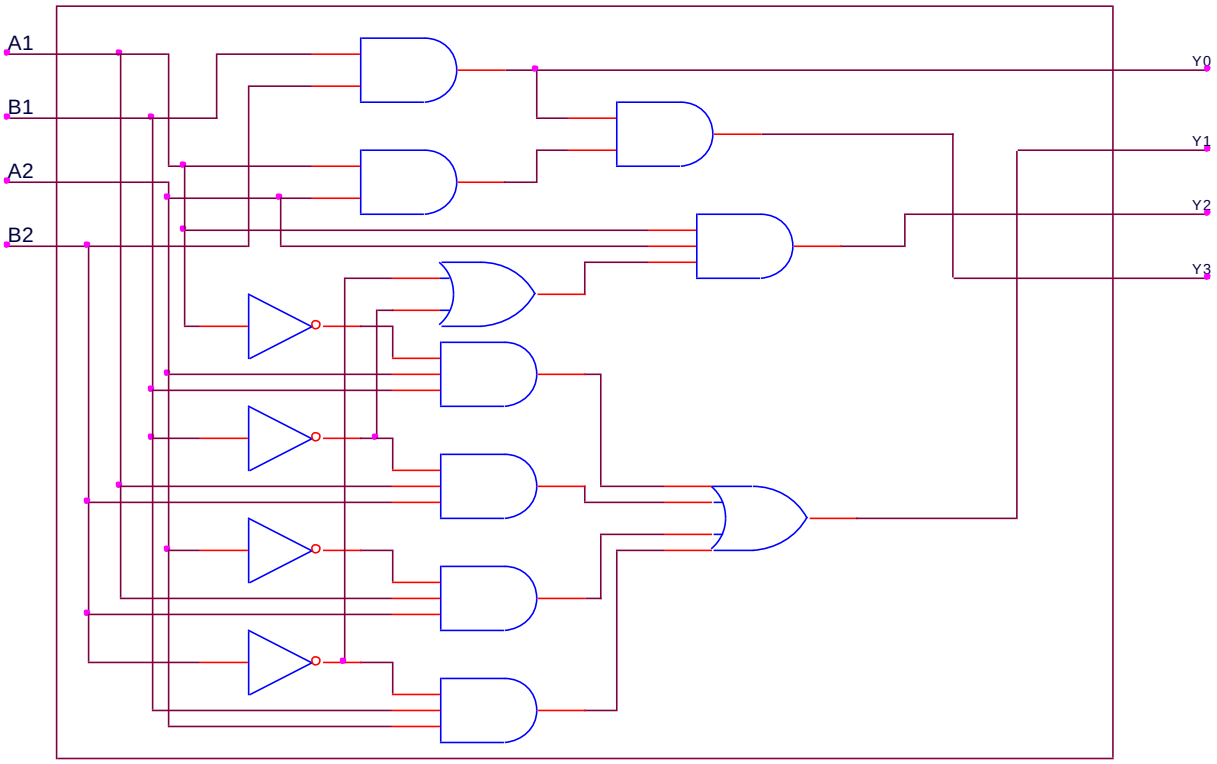
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Rút gọn ta được

Y3=A1B1A2B2;Y2=A1A2($\overline{B1}+\overline{B2}$);Y1= $\overline{A1}B1A2+A1\overline{B1}B2+A1\overline{A2}B2+B1A2\overline{B2}$

Y0=B1B2.

-Thực hiện mạch



VD3:So sánh 2 số nhị phân 2 bit

-Số ngõ vào là 4 và ngõ ra là 3 ;Y0 (2 số bằng nhau);Y1 (số thứ 1 lớn hơn);Y2 (số thứ nhất nhỏ hơn)

-Bảng giá trị

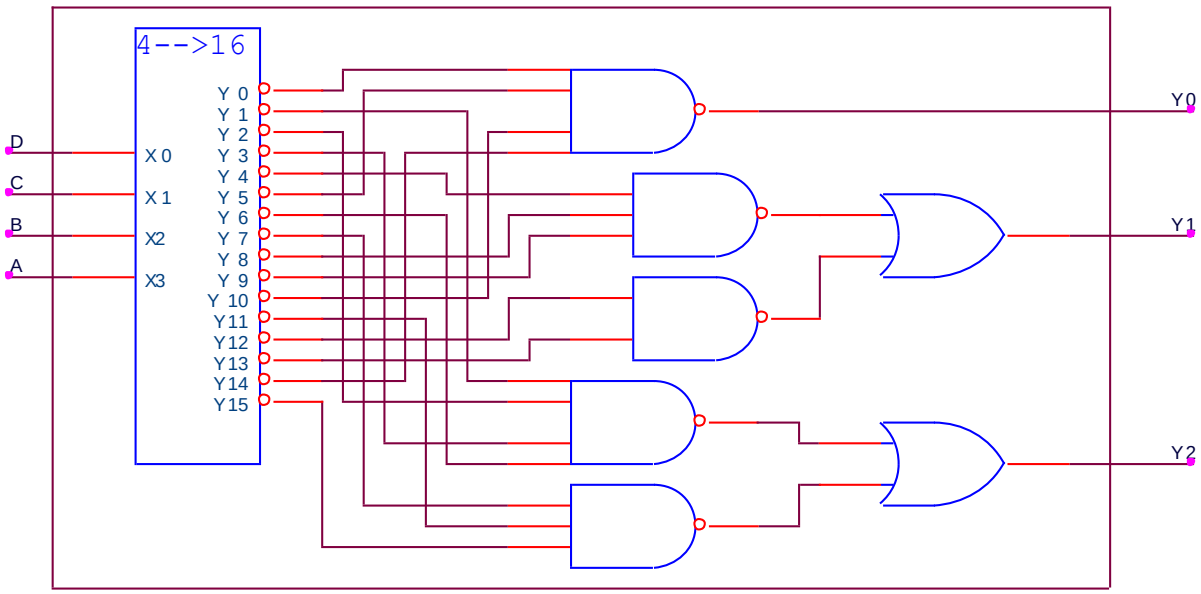
A1	B1	A2	B2	Y2	Y1	Y0
0	0	0	0	0	0	1
0	0	0	1	1	0	0

0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	0	0	1
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

-Rút gọn:Từ bảng giá trị trên ta có

Y0=Σ(0,5,10,15);Y1=Σ(4,8,9,12,13);Y2=Σ(1,2,3,6,7,11,14)

-Thực hiện mạch.Có nhiều cách thực hiện ở đây ta sử dụng Decoder thực hiện



BÀI TẬP CHƯƠNG 3

1.Thực hiện ghép 2 Decoder 3→8 thành 4→16

2.Thực hiện ghép 4 Decoder 2→4 thành 4→16

3.Cho hàm F(A,B,C,D)= $\sum(0,3,4,7,13,14,15)$

a.Thực hiện hàm Boole bằng cổng Logic

b.Thực hiện hàm Boole bằng Decoder

c.Thực hiện hàm Boole bằng MUX

d.Thực hiện hàm Boole bằng MUX 8→1

4.Cho hàm F(A,B,C,D)= $\prod(0,3,4,6,9,10,15)$

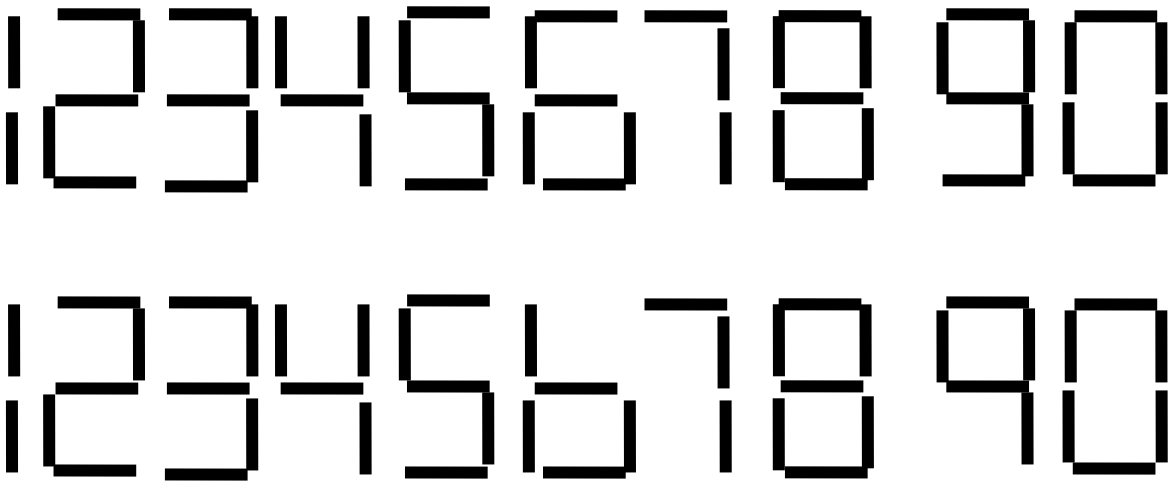
a.Thực hiện hàm Boole bằng cổng Logic

b.Thực hiện hàm Boole bằng Decoder

c.Thực hiện hàm Boole bằng MUX

d.Thực hiện hàm Boole bằng MUX 8→1

5.Thiết kế mạch tổ hợp sao cho mã vào là BCD và mã ra là LED 7 đoạn



6.Thiết kế mạch tổ hợp sao cho khi cho vào mã nhị phân 4 bit .Ngõ ra sẽ ở mức 0 khi số thập phân tương ứng của ngõ vào nhỏ hơn 10

7.Thiết kế mạch so sánh 2 số nhị phân 3 bit.Dùng LED báo hiệu.

CHƯƠNG 4: MẠCH TUẦN TỰ

Giới thiệu: *Trong chương này, giới thiệu các loại mạch chốt và Flip Flop cơ bản, các dạng mạch đếm và mạch ghi dịch.*

Mục tiêu:

- Hiểu các mạch chốt, các Flip Flop cơ bản.

- Hiểu được nguyên lý làm việc của các Flip Flop, các dạng mạch đếm và mạch ghi dịch.

- Rèn luyện tính kiên trì, cẩn thận và tư duy logic.

Nội dung chính:

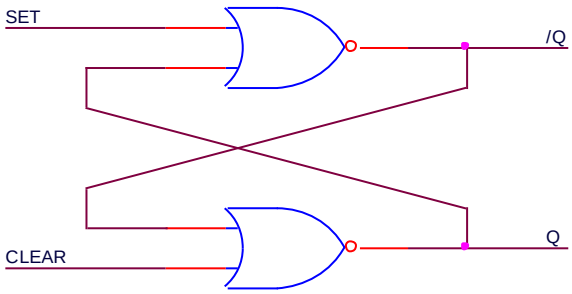
4.1. Các mạch chốt và Flip Flop:

Mạch tuần tự nhận thông tin từ các ngõ vào và kết hợp với các phần tư nhớ để xác định trạng thái của ngõ ra. Như vậy mạch tuần tự có các đầu ra không chỉ phụ thuộc vào đầu vào hiện hành mà còn phụ thuộc vào một chuỗi các giá trị đầu vào ở quá khứ.

Các mạch chốt (Latch):

Mạch chốt là mạch tuần tự sẽ xét các ngõ vào một cách liên tục và thay đổi các ngõ ra của nó ở bất cứ lúc nào.

4.1.1. Mạch chốt cổng NOR:

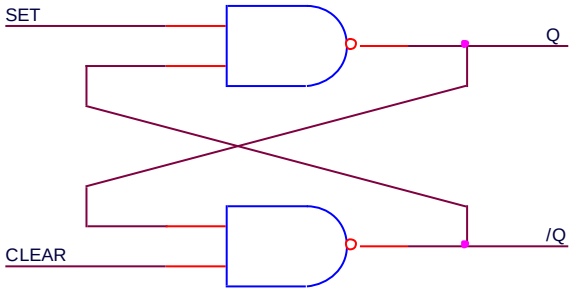


Bảng giá trị

SET	CLEAR	OUTPUT
0	0	Không đổi
0	1	Q=0
1	0	Q=1
1	1	Cấm

***Cấm : là trạng thái Q=/Q=0 (Trạng thái này không dùng)**

4.1.2. Chốt cổng NAND:



Bảng giá trị

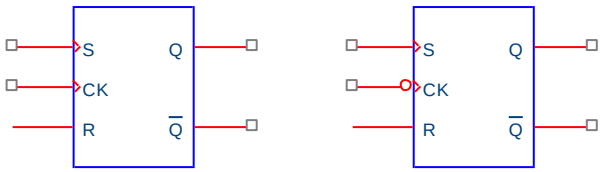
SET	CLEAR	OUTPUT
0	0	Cấm
0	1	Q=1
1	0	Q=0
1	1	Không đổi

Các loại Flip-Flop:

Flip-Flop là một mạch tuần tự thông thường lấy mẫu các ngõ vào và thay đổi các ngõ ra của chúng ở những thời điểm xác định bởi tín hiệu đồng hồ (xung clock).

Người ta phân loại FF dựa vào tín hiệu điều khiển :FF điều khiển (tích cực) cạnh lên và FF điều khiển (tích cực) cạnh xuống.

4.1.3. S-R Flip Flop

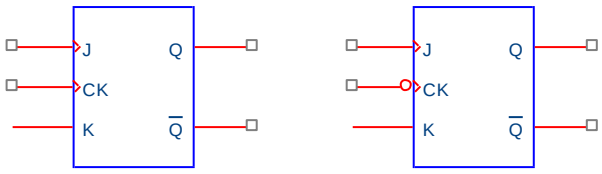


Bảng giá trị

S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	x

x: Trạng thái cấm không dùng

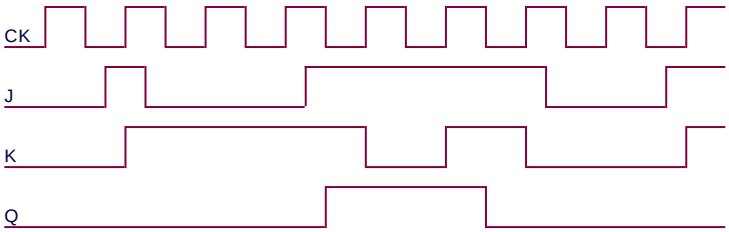
4.1.4. J-K Flip Flop



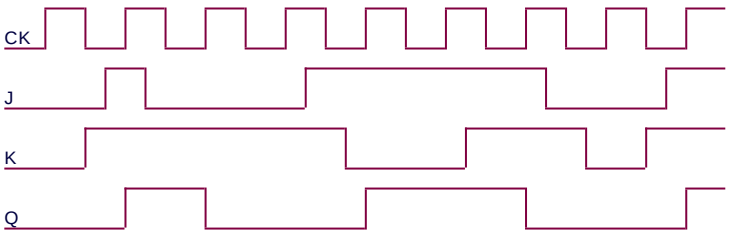
Bảng giá trị

J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Biểu diễn dạng tín hiệu ngõ ra theo tín hiệu vào và xung Clock.Giả sử ban đầu Q_0=0.FF tích cực cạnh xuống.

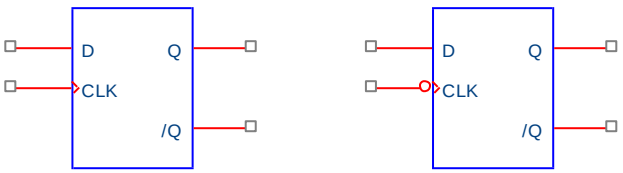


Biểu diễn dạng tín hiệu ngõ ra theo tín hiệu vào và xung Clock.Giả sử ban đầu Q_0=0.FF tích cực cạnh lên.



4.1.5. D Flip Flop

Ký hiệu



(FF tích cực cạnh lên) (FF tích cực cạnh xuống)

Bảng giá trị :

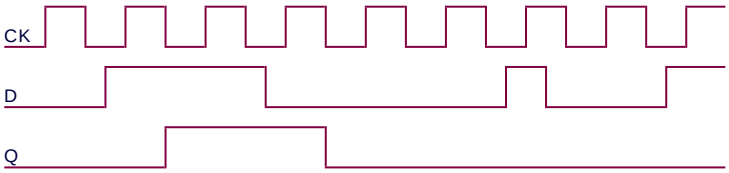
D_n	Q_{n+1}
0	0
1	1

D_n là trạng thái của tín hiệu vào ở xung thứ n

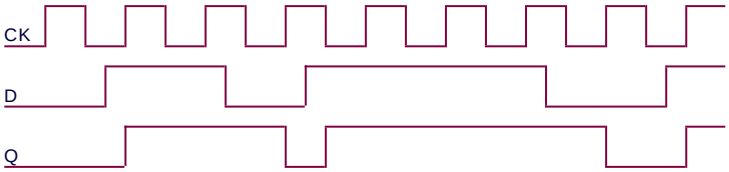
Q_{n+1} là trạng thái của tín hiệu ra ở xung thứ n+1

Một cách tổng quát: $Q_{n+1}=D_n$

Biểu diễn dạng tín hiệu ngõ ra theo tín hiệu vào và xung Clock.Giả sử ban đầu Q_0=0.FF tích cực cạnh xuống.



Biểu diễn dạng tín hiệu ngõ ra theo tín hiệu vào và xung Clock.Giả sử ban đầu Q_0=0.FF tích cực cạnh lên



4.1.6. T Flip Flop

(FF tích cực cạnh lên) (FF tích cực cạnh xuống)

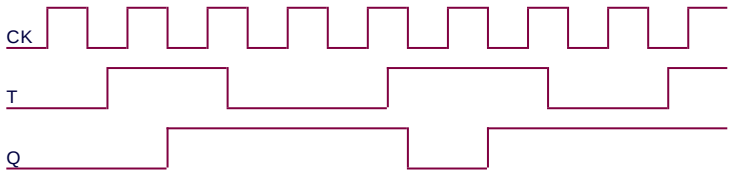
Bảng giá trị :

T_n	Q_{n+1}
0	Q_n
1	$\overline{Q_n}$

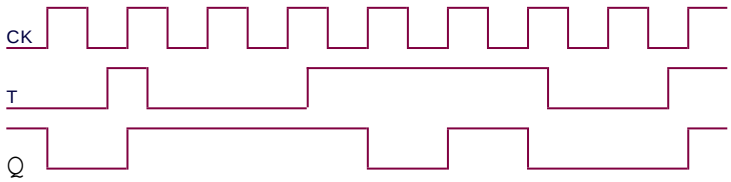
T_n là trạng thái của tín hiệu vào ở xung thứ n

Q_{n+1} là trạng thái của tín hiệu ra ở xung thứ n+1

Biểu diễn dạng tín hiệu ngõ ra theo tín hiệu vào và xung Clock.Giả sử ban đầu Q_0=0.FF tích cực cạnh xuống.

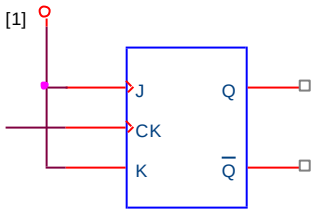


Biểu diễn dạng tín hiệu ngõ ra theo tín hiệu vào và xung Clock.Giả sử ban đầu Q_0=1.FF tích cực cạnh lên.

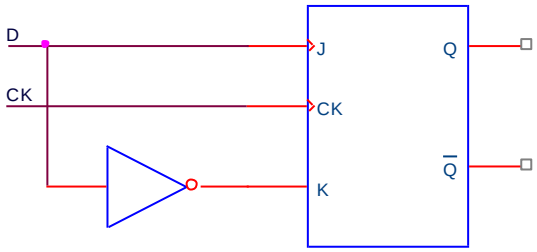


4.1.7. Chuyển đổi giữa các Flip Flop:

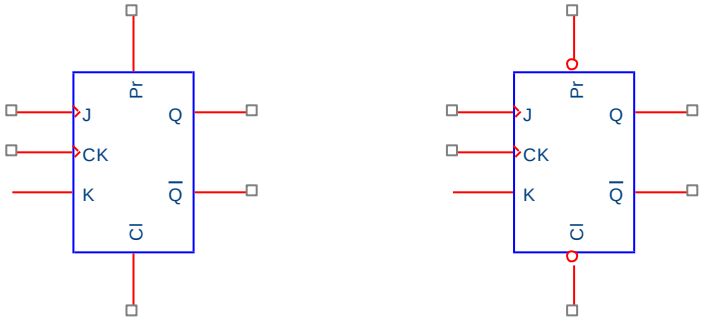
- J_K FF sẽ tương đương T- FF khi hai ngõ vào J= K = 1



- J_K FF sẽ tương đương D-FF khi hai ngõ vào J=và K nối với nhau qua cổng đảo.



Ngoài ra FF còn có các ngõ vào điều khiển là Preset(Pr) và Clear (Cl).Ký hiệu thường như sau



(FF có Pr và Cl tích cực mức cao) (FF có Pr và Cl tích cực mức thấp)

Khi Pr và Cl tác động thì ngõ ra sẽ ở mức 0 hoặc 1 mà không phụ thuộc vào tín hiệu ở ngõ vào

+Trường hợp Pr và Cl tích cực mức cao

-Khi Pr=Cl=0 thì FF có ngõ ra phụ thuộc vào ngõ vào và xung CK

-Khi Pr=1 và Cl=0 thì FF có ngõ ra Q=1

-Khi Pr=0 và Cl=1 thì FF có ngõ ra Q=0

-Khi Pr=Cl=1 thì FF có ngõ ra không xác định được

+Trường hợp Pr và Cl tích cực mức thấp

-Khi Pr=Cl=1 thì FF có ngõ ra phụ thuộc vào ngõ vào và xung CK

-Khi Pr=0 và Cl=1 thì FF có ngõ ra Q=1

-Khi Pr=1 và Cl=0 thì FF có ngõ ra Q=0

-Khi Pr=Cl=1 thì FF có ngõ ra không xác định được

4.2. Mạch đếm nối tiếp (không đồng bộ):

4.2.1. Mạch đếm nối tiếp MOD chẵn 2n

Phương pháp thiết kế

-MOD 2^n sẽ dùng n JKFF

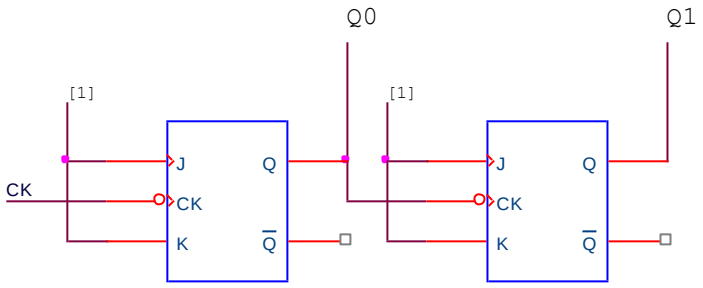
-Đếm nối tiếp thì xung CK của FF sau được lấy từ ngõ ra Q của FF trước

-Các ngõ J&K nối chung và nối với mức Logic 1.Số đếm được lấy từ các ngõ ra Q của các JKFF (dạng BCD).

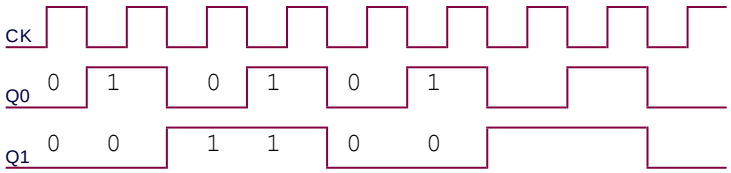
Ví dụ1: Thiết kế mạch đếm có MOD=4 dùng JKFF tích cực cạnh xuống.Giả sử ban đầu các ngõ ra bằng 0.

-MOD=4=2^2 do đó dùng 2 JKFF

-Sơ đồ thực hiện như hình vẽ



Tín hiệu ra được biểu diễn như sau

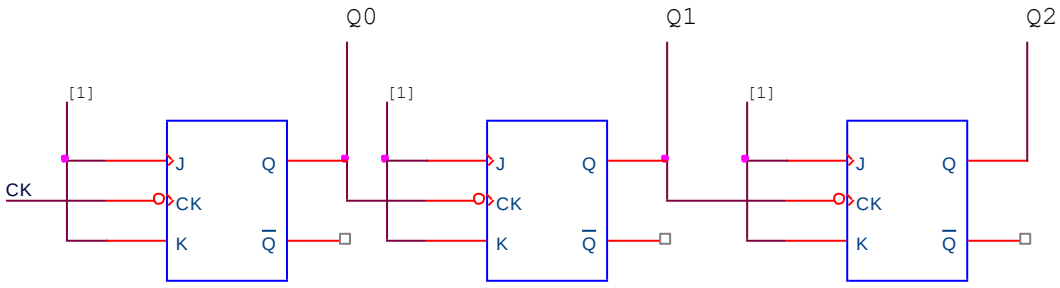


Nhìn vào kết quả trên ta thấy dữ liệu ra là :00,01,10,11,00,01…Nếu ta ta sử dụng mạch giải mã chuyển đổi từ BCD sang LED 7 đoạn ta sẽ thấy số hiển thị lần lượt là 0,1,2,3,0,1…

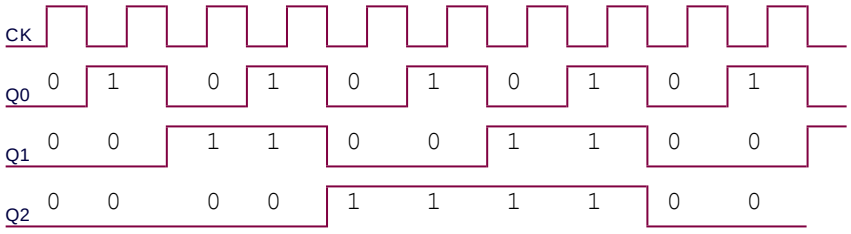
Ví dụ2: Thiết kế mạch đếm có MOD=8 dùng JKFF tích cực cạnh xuống.Giả sử ban đầu các ngõ ra bằng 0.

-MOD=8=2^3 do đó dùng 3 JKFF

-Sơ đồ thực hiện như hình vẽ



Tín hiệu ra được biểu diễn như sau



Trường hợp mạch đếm nối tiếp nhưng đếm giảm dần thì cách thiết kế tương tự như trên nhưng xung CK của FF sau được lấy từ ngõ $\overline{Q}$ của FF trước

Tóm lại **nếu CK tác động cạnh xuống**

Q_n=CK_{n+1} thì mạch đếm là đếm nối tiếp và đếm lên

$\overline{Q_n}$ =CK_{n+1} thì mạch đếm là đếm nối tiếp và đếm xuống

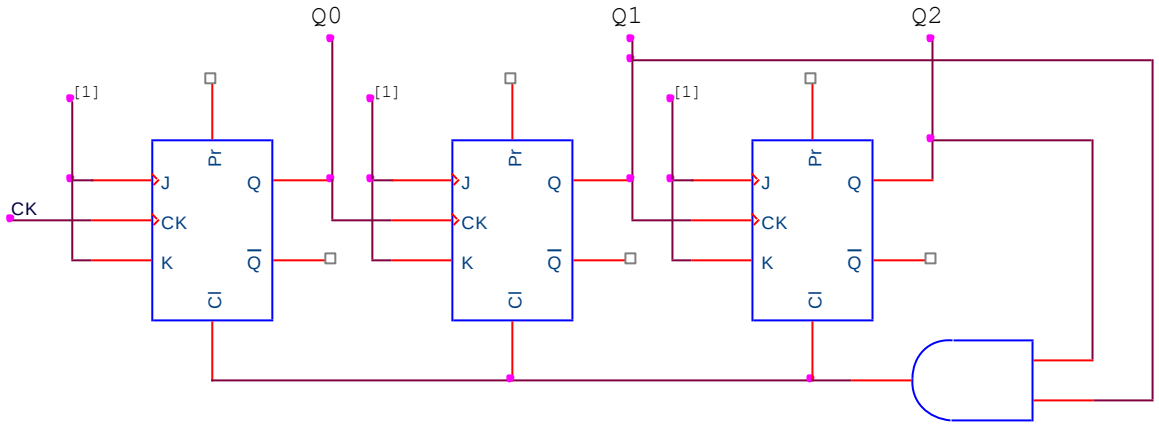
4.2.2. Mạch đếm nối tiếp MOD lẻ

Để thiết kế mạch đếm loại này bắt buột ta phải dùng loại JKFF có Pr và Cl

Ví dụ1: Thiết kế mạch đếm đếm lên dùng JKFF tích cực cạnh lên có Pr và Cl tích cực mức cao đếm chuỗi số sau:0,1,2,3,4,5,0,1,2,.. và cứ thế lặp lại

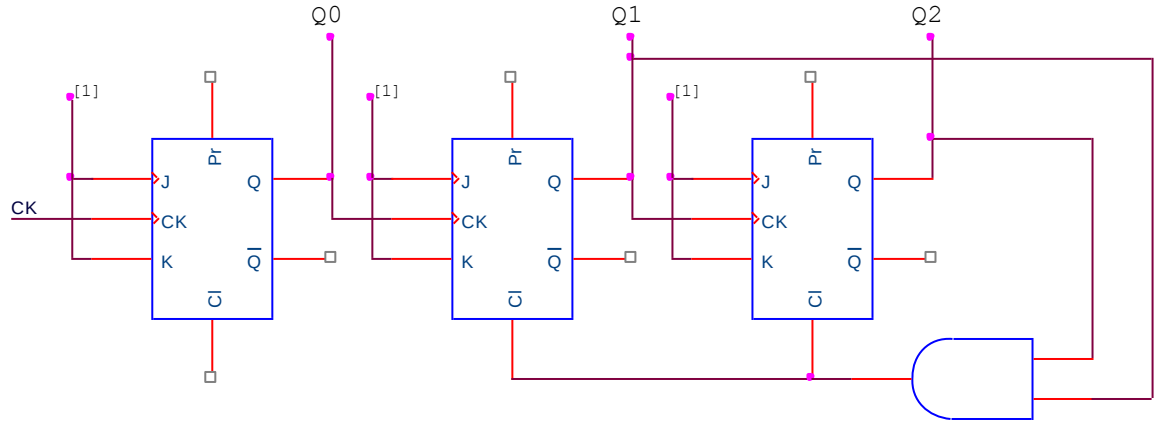
Ta thấy rằng trong một chu kỳ đếm xuất hiện 6 trạng thái khác nhau.Vì vậy mạch đếm có MOD=6 tức là MOD lẻ.Do $2^2<6<2^3$ nên ta dùng 3JKFF.Nếu như mắc theo phương pháp trên thì mạch sẽ đếm từ 0 đến 7 sau đó sẽ quay trở lại chu kỳ mới.Trong trường hợp này ta lại không muốn điều đó xảy ra mà ta muốn khi đếm đến 5 mạch tự động quay về 0.Trong JKFF có một ngõ làm cho FF trở về 0 lập tức đó là chân Cl.Như vậy thay vì khi đếm đến 5 mạch sẽ tiếp tục đếm lên 6 ,tại thời điểm này ta sẽ tác động vào chân Cl để mạch tự động trở về 0.Phương pháp tốt nhất là ta dùng cổng AND hoặc cổng NAND

Từ đó ta có thể đưa ra sơ đồ mạch cho ví dụ trên như sau:



* Chân Pr ta không tác động đến nên khi thực hiện mạch ta nên nối các chân này xuống mức logic 0 (GND)

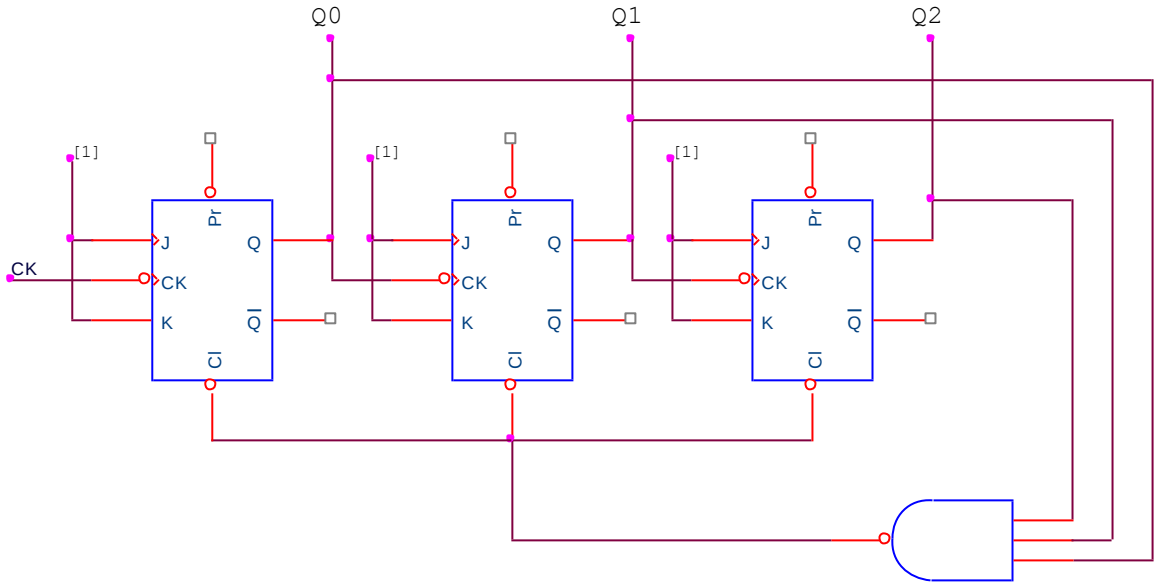
Tuy nhiên ta có thể đưa ra sơ đồ khác để thực hiện mạch trên như sau



Ta có được sơ đồ này vì lúc đó ngõ ra Q_0 =0 nên ta cũng không cần tác động vào chân Cl của FF đầu tiên

Ví dụ 2: Thiết kế mạch đếm đếm lên dùng JKFF tích cực cạnh xuống có Pr và Cl tích cực mức thấp đếm chuỗi số sau:0,1,2,3,4,5,6,0,1,2,.. và cứ thế lặp lại

Tương tự như trên ta có sơ đồ thực mạch như sau:



4.3. Mạch ghi dịch

4.3.1. Dữ liệu vào nối tiếp

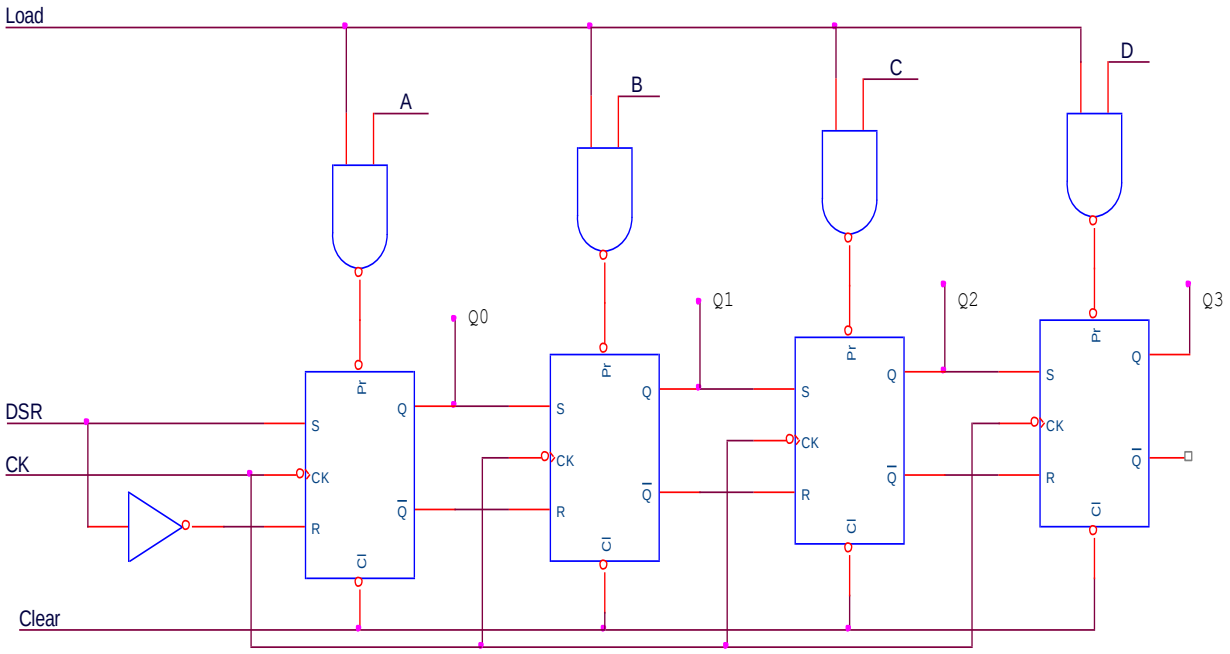
Xét hệ ghi dịch 4 bit dịch phải

-Load:Xung nhập data

-Clear:xung xóa

-DSR:Data Shift Right

-Các SR được nối thành DFF



+Khi tác động vào Cl thì $Q_3Q_2Q_1Q_0$=0000

+Khi tác động vào Load thì $Q_3Q_2Q_1Q_0$=DCBA

+Thực hiện dời phải

Xung	$Q_3Q_2Q_1Q_0$ (hiện tại)	$Q_3Q_2Q_1Q_0$ (kế tiếp)
1	CDBA	DSR_1CDB
2	DSR_1CDB	DSR_2DSR_1CD
3	DSR_2DSR_1CD	$DSR_3DSR_2DSR_1C$
4	…….	
	……..	

*Xét trường hợp Q_3 nối với DSR

Xung	$Q_3Q_2Q_1Q_0$ (hiện tại)	$Q_3Q_2Q_1Q_0$ (kế tiếp)
1	CDBA	ACDB
2	ACDB	BACD
3	BACD	DBAC
4	DBAC	CDBA
5	CDBA	ACDB

Trong trường hợp này ta có mạch đếm vòng

*Xét trường hợp $\overline{Q}_3$ nối với DSR

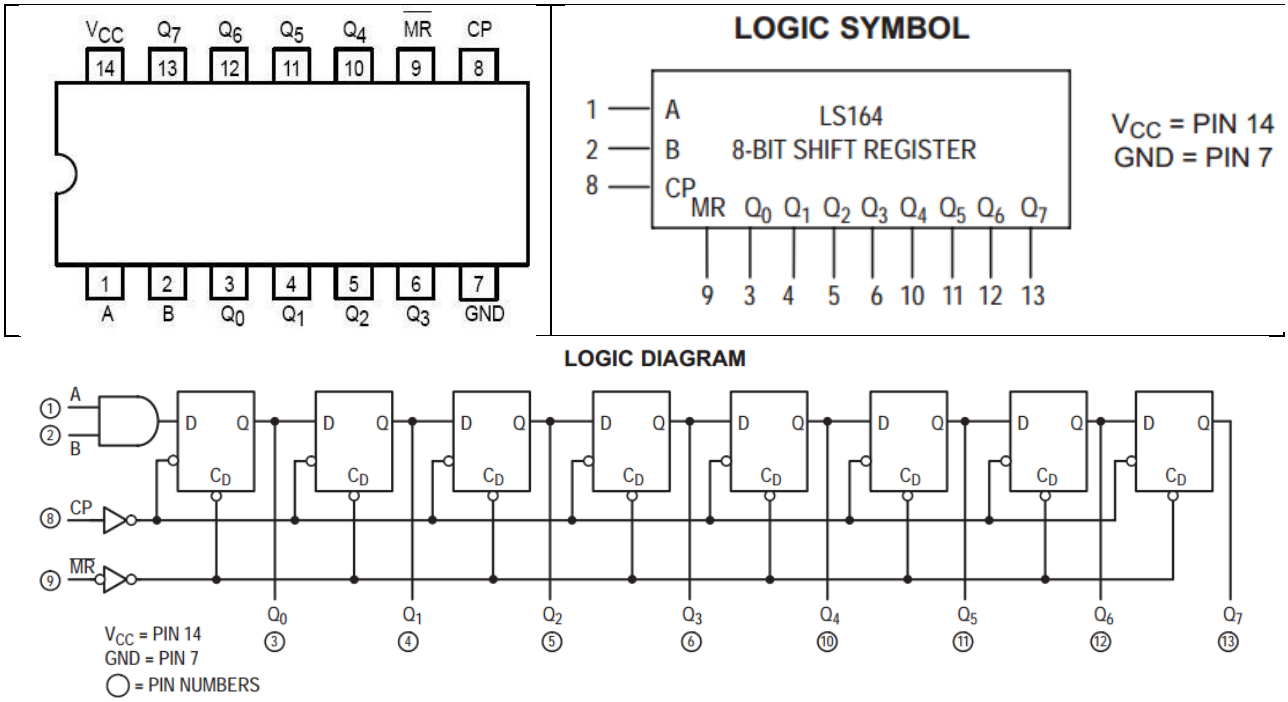
Gọi là mạch đếm vòng xoắn.Giả sử DCBA=0000

Xung	$Q_3Q_2Q_1Q_0$
1	0000
2	1000
3	1100
4	1110
5	1111
6	0111
7	0011
8	0001
9	0000

Ta có thể dùng mạch này cho hệ thống đèn nhấp nháy thường thấy ở bên ngoài.Tốc độ nhanh chậm phụ thuộc vào xung CK.

4.3.2. Dữ liệu vào song song

*** Khảo sát IC 74164**



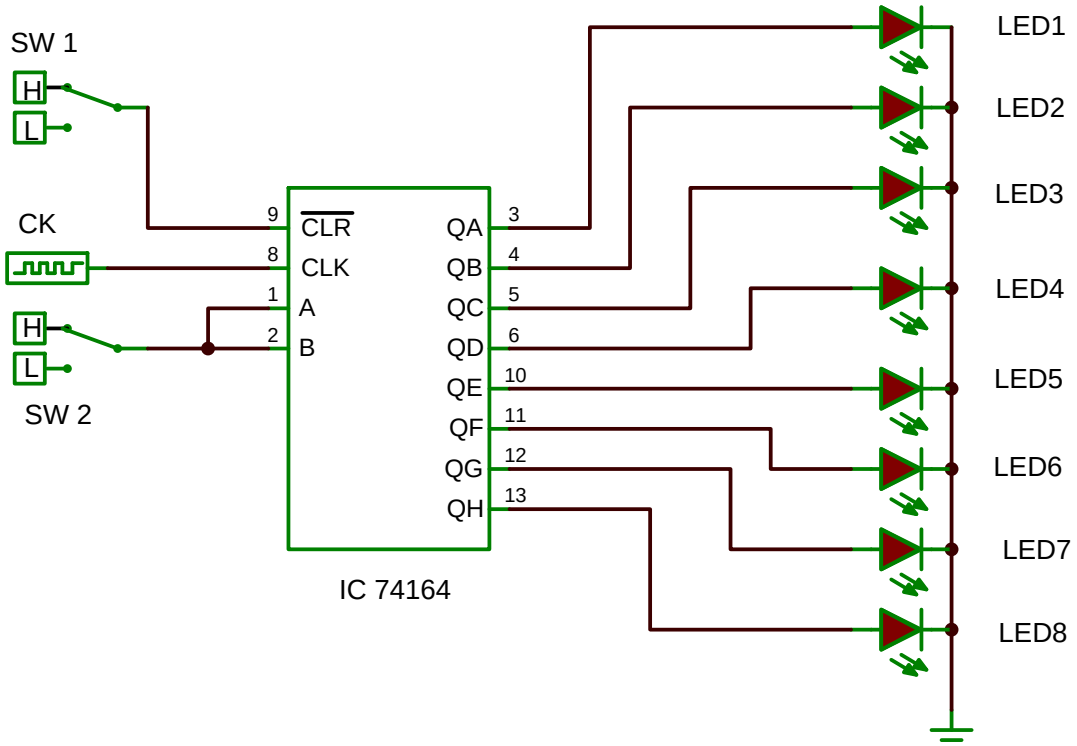
Hình: Sơ đồ chân và cấu tạo của IC74164

MODE SELECT — TRUTH TABLE

OPERATING MODE	INPUTS			OUTPUTS	
	$\overline{MR}$	A	B	Q_0	Q_1–Q_7
Reset (Clear)	L	X	X	L	L – L
Shift	H	l	l	L	$q_0 - q_6$
	H	l	h	L	$q_0 - q_6$
	H	h	l	L	$q_0 - q_6$
	H	h	h	H	$q_0 - q_6$

Hình: Bảng sự thật của IC74164

Mạch kiểm tra IC74164



Hình: Mạch kiểm tra IC 74164

BÀI TẬP CHƯƠNG 4

1.Thiết kế mạch đếm, đếm lên MOD 13 theo 2 phương pháp song song và nối tiếp.

2.Thiết kế mạch đồng hồ số hiển thị giờ phút giây.

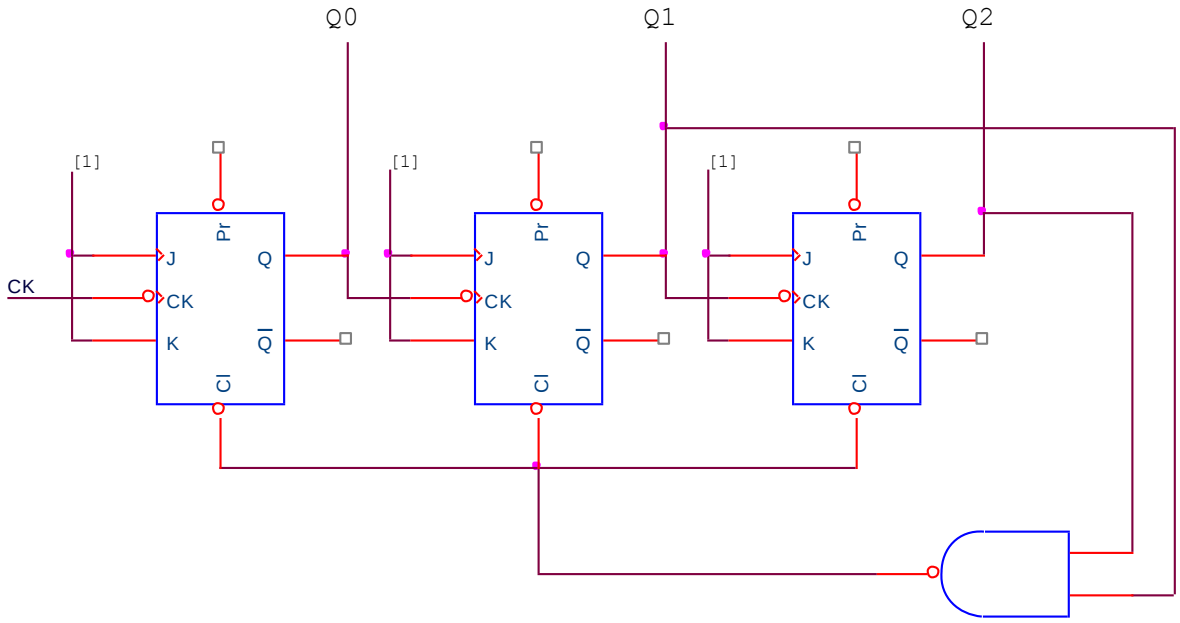
3.Thiết kế mạch sao cho tần số vào là 32.768Hz và tần số ra là 1Hz.

4.Thiết kế mạch đo chu kỳ –tần số .

5.Thiết kế mạch có chuỗi số đếm như sau:1,3,5,7,9,1,3…Nếu số ban đầu không nằm trong chuỗi số đếm thì số tiếp theo bằng 0.

6.Thiết kế mạch đếm, đếm xuống MOD 6 theo 2 phương pháp song song và nối tiếp.

7.Cho mạch như hình vẽ.Xác định mạch đếm loại gì,MOD là bao nhiêu



CHƯƠNG 5: CÁC MẠCH SỐ KHÁC

Giới thiệu: *Trong chương này, giới thiệu về các mạch chuyển đổi A/D, D/A và bộ nhớ.*

Mục tiêu:

- Hiểu các mạch chuyển đổi A/D, D/A và bộ nhớ.

- Hiểu được nguyên lý làm việc của các mạch chuyển đổi A/D, D/A và bộ nhớ..

- Rèn luyện tính kiên trì, cẩn thận và tư duy logic.

Nội dung chính:

5.1. Mạch chuyển đổi A/D, D/A:

5.1.1. Mạch chuyển đổi ADC:

Mạch chuyển đổi tương tự ra số hay **ADC** (viết tắt từ tiếng Anh *Analog-to-Digital Converter*) là hệ thống mạch thực hiện chuyển đổi một tín hiệu tương tự liên tục, ví dụ như tín hiệu âm thanh thanh micro, hay tín hiệu ánh sáng trong máy ảnh kỹ thuật số, thành tín hiệu số .

Một hệ thống ADC có thể bao gồm một bộ phận phần cứng (như một bộ tính toán độc lập) làm nhiệm vụ chuyển đổi tín hiệu analog (dưới dạng điện áp hay dòng điện) thành các giá trị số (digital) đại diện cho cường độ điện áp hay tín hiệu đó. Thông thường, tín hiệu số ngõ ra (digital output) mang dạng nhị phân bù 2 tỉ lệ với giá trị ngõ vào, nhưng cũng có một số khả năng khác.

Có một số kiến trúc ADC đang được sử dụng. Do sự phức tạp của kiến trúc và yêu cầu về độ chính xác, phần lớn các hệ thống ADC đều được sản xuất bên trong mạch tích hợp (IC). Tại ngõ vào chính của ADC trong chip có thể có phần tử Multiplexer, cho ra ADC đa ngõ vào hay ADC đa kênh. Trước đây giá thành ADC cao, nên đã bố trí 8 đến 64 ngõ vào. Hiện nay xuất hiện các chip chỉ bố trí 1, 2 hoặc 4 ngõ vào.

Nguyên lý hoạt động:

Để thực hiện việc chuyển đổi một tín hiệu analog thực tế (như nhiệt độ, độ ẩm, âm thanh,...) thành tín hiệu số, thì tín hiệu analog thực tế này phải được chuyển đổi thành dạng điện áp. Bộ ADC sau đó sẽ đọc các giá trị điện áp này và chuyển đổi thành tín hiệu số tương ứng.

Do quá trình chuyển đổi này liên quan đến việc lượng tử hóa tín hiệu ngõ vào, do đó nhất thiết mắc một lượng lỗi hoặc bị ảnh hưởng bởi nhiễu tín hiệu. Thay vì liên tục thực hiện việc chuyển đổi, bộ ADC thực hiện việc chuyển đổi theo chu kì, lấy mẫu (sampling) tín hiệu ngõ vào, giới hạn băng thông cho phép của tín hiệu.

Hoạt động của một bộ ADC được đặc trưng bởi băng thông và tỉ số tín hiệu trên nhiễu (SNR signal-to-noise ratio). Băng thông của ADC được đặc trưng bởi tốc độ lấy mẫu (sampling rate). Tỉ số SNR của bộ ADC bị ảnh hưởng bởi nhiều yếu tố bao gồm: độ phân giải (resolution), độ tuyến tín (linearity) và độ chính xác (accuracy) (đánh giá tính hiệu

quả của quá trình lượng tử hoá tín hiệu từ tính hiệu analog thực tế), aliasing và jitter. Tỉ số SNR của bộ ADC thể hiện số bit trung bình trả về trong mỗi tính toán mà không bị nhiễu, được gọi là số bit hiệu quả (ENOB effective number of bits). Một bộ ADC lý tưởng có số ENOB bằng với độ phân giải của nó.

Độ phân giải

Bậc số hóa là số bit xác định số mức số hóa cho dải giá trị điện áp danh định. Hệ M bit có 2^M mức cho tín hiệu đơn cực, chỉ dương hoặc chỉ âm. Nếu là tín hiệu song cực, phải dành 1 bit dấu, và do mức 0 bị dính nên hệ cho ra 2^{M-1}-1 mức.

Dải giá trị điện áp danh định này được gọi là dải động. Điện áp lớn hơn thì gây tràn (overflow).

Alias

Điểm chú ý là tác động của hiện tượng Aliasing đến đặc trưng số hóa, và nó dẫn đến đòi hỏi tần số số hóa phải lớn hơn trên gấp đôi tần cực đại của băng tần tín hiệu trong các nhu cầu thông thường, còn trong nhu cầu kỹ thuật thì là gấp 4, ví dụ phải dùng 1 KHz để số hóa tín hiệu có băng tần 10–250 Hz.

Tốc độ lấy mẫu

Vì tín hiệu analog là liên tục theo thời gian nên cần thiết để chuyển đổi tín hiệu này thành một dãy các giá trị kỹ thuật số. Do đó cần định nghĩa một đại lượng tốc độ đặc trưng cho thời gian mà các giá trị kỹ thuật số (digital values) được lấy mẫu từ tín hiệu analog. Tốc độ này được gọi là tốc độ lấy mẫu hay tần số lấy mẫu. Một tín hiệu analog liên tục có thể được lấy mẫu và sau đó được khôi phục lại dạng tín hiệu gốc ban đầu từ các giá trị mẫu rời rạc theo thời gian bởi bộ lọc khôi phục (reconstruction filter).

Định lý lấy mẫu Nyquist–Shannon chỉ ra rằng tín hiệu gốc chỉ có thể được khôi phục lại như ban đầu nếu tốc độ lấy mẫu lớn hơn hoặc bằng 2 lần tần số lớn nhất của tín hiệu gốc.

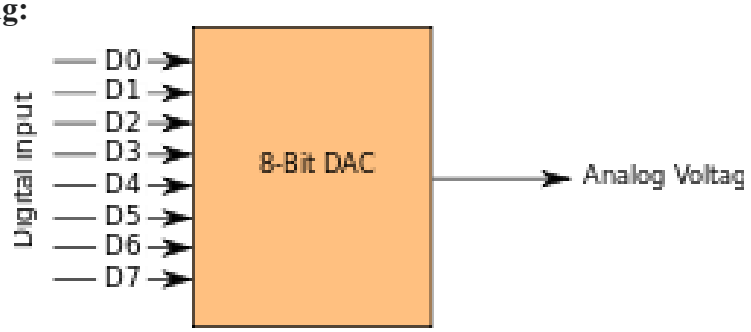
Do bộ ADC không thể thực hiện việc chuyển đổi tín hiệu tức thời, giá trị đầu vào phải được lưu như hằng số trong thời điểm thực hiện tính toán-chuyển đổi (gọi là thời gian chuyển đổi, conversion time). Khối mạch lấy giá trị lấy mẫu và thực hiện quá trình tính toán-chuyển đổi này trong phần lớn các trường hợp dùng tụ điện để lưu các giá trị analog điện áp đầu vào và sử dụng mạch switch hoặc gate để ngắt kết nối tụ với ngõ vào. Nhiều IC ADC ngày nay đều có thành phần là các khối xử lý đó.

5.1.2. Mạch chuyển đổi DAC:

DAC haymạch chuyển đổi số ra tương tự hay *Digital-to-Analog converter* là linh kiện dán dẫn thực hiện chuyển đổi *dữ liệu kỹ thuật số* (thường là nhị phân) thành tín hiệu *tương tự* thường là điện áp.

Nó hoàn nguyên tín hiệu tương tự từng được số hóa bởi ADC.

Nguyên lý hoạt động:

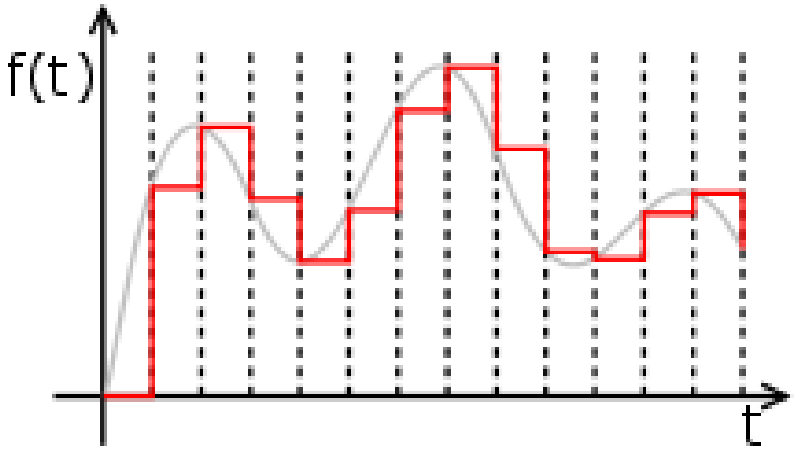


Biến đổi DAC 8 bit

Thông thường thì chuyển đổi DAC thực hiện bằng mảng điện trở, với số ngõ vào tương ứng với số bit của số liệu, và một ngõ ra. Giá trị điện trở được chọn tương ứng với bậc của bit số, để tạo ra *trọng số biến đổi*, sao cho khi điện áp vào ở mức logic quy định thì phần của bit đó góp vào điện áp tổng đúng như bậc của bit đó. Kết quả là ở ngõ ra có mức điện áp tương ứng với giá trị số của ngõ vào.

Do có sơ đồ đơn giản, chuyển đổi DAC bằng mảng điện trở có thể làm việc ở tần số tương đối cao.

Dẫu vậy trong thực tế thì chế tạo được mảng điện trở chính xác và ổn định cao là khó khăn về công nghệ. Nó dẫn đến giá thành mảng cao. Vì thế chip DAC thường bố trí ghép kênh, mảng điện trở thực thi biến đổi cho nhiều đường tín hiệu. Ví dụ chip Cirrus Logic CS4382 là DAC 8 kênh, và trong ứng dụng soundcard thì sử dụng 2 đường.



Hoàn nguyên DAC, cần đến bộ lọc làm trơn để cho ra tín hiệu (đường màu xám)

Kết quả DAC là tín hiệu có từng bậc, nên cần có bộ lọc làm trơn để lọc bỏ các hài tần cao.

5.2. Bộ nhớ:

Bộ nhớ (*memory*), là một thiết bị công nghệ bao gồm các phần tử máy tính và lưu trữ dữ liệu, được dùng để duy trì dữ liệu số. Nó là một linh kiện cơ bản có chức năng cốt lõi của các máy tính.

Bộ nhớ máy tính bao gồm các bộ nhớ điện tĩnh (*non-volatile memory*) để lưu trữ được dữ liệu của máy tính một cách lâu dài (khi kết thúc một phiên làm việc của máy thì dữ liệu không bị mất đi), hoặc bộ nhờ điện động (*volatile memory*) để lưu dữ liệu tạm thời trong quá trình làm việc của máy tính (khi kết thúc một phiên làm việc của máy tính thì bộ nhớ này bị mất hết dữ liệu).

Các thiết bị lưu trữ dữ liệu cho bộ nhớ lâu dài bao gồm:

+ Đĩa cứng

+ Đĩa mềm

+ Đĩa quang

+ Băng từ

+ ROM

….

Các thiết bị lưu trữ dữ liệu tạm thời trong quá trình làm việc:

+ RAM máy tính

+ Cache ...

Hầu hết các bộ nhớ nêu trên thuộc loại bộ nhớ có thể truy cập dữ liệu ngẫu nhiên, riêng băng từ là loại bộ nhớ truy cập tuần tự.

Bộ nhớ máy tính có thể chia thành hai dạng:

Bộ nhớ trong (main memory) và bộ nhớ ngoài (external storage).

5.2.1. RAM:

Bộ nhớ trong được hiểu là các loại bộ nhớ nằm nội bộ bên trong thùng máy. Còn có tên gọi khác là bộ nhớ chính (*Main Memory*)

- Bộ nhớ đệm nhanh (cache memory):
  - Tốc độ truy xuất nhanh;
  - Thường nằm trong 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 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;

- Bộ nhớ chính (M*ain Memory*):
  - Bộ nhớ RAM (*Random Access Memory*), hay Bộ nhớ truy cập ngẫu nhiên: 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 (*Read Only Memory*), hay Bộ nhớ chỉ đọc: 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 ảo (Virtual Memory);

5.2.2. ROM:

Bộ nhớ ngoài được hiểu là bộ nhớ máy tính gắn bên ngoài, có thể dùng để mang đi lại được giữa các máy tính.

Bao gồm:

- Bộ nhớ từ: đĩa cứng, Đĩa mềm,...
- Bộ nhớ quang: CD, DVD,...
- Bộ nhớ bán dẫn: flash disk, thẻ nhớ...
- Các loại bộ nhớ dựa trên công nghệ **Flash ROM**: Kết hợp với chuẩn giao tiếp máy tính USB (*Universal Serial Bus*) tạo ra các bộ nhớ máy tính di động thuận tiện và đa năng như: Các thiết bị giao tiếp USB lưu trữ dữ liệu, thiết bị giao tiếp **USB** chơi nhạc số, chơi video số; khóa bảo mật qua giao tiếp USB; thẻ nhớ... Dung lượng thiết bị lưu trữ **Flash ROM** đã lên tới 32**GB** (Samsung, Intel công bố năm 2005), trong tương lai, có thể **Flash ROM** sẽ dần thay thế các ổ đĩa cứng, các loại đĩa **CD**, **DVD**...
- Cách phân biệt trong và ngoài như trên chỉ mang tính tương đối. Ví dụ các loại ổ cứng, ổ đĩa CD có thể gắn ngoài (qua giao tiếp USB, DATA)*tốc độ truy cập nhanh. Ổ đĩa mềm có thể đặt vào máy, lấy ra khỏi máy dễ dàng. dung lượng nhỏ tốc độ quay chậm, tốc độ truy cập chậm. Đĩa **CD** và **USB** là những thiết bị nhớ có dung lượng tương đối cao đến hàng trăm **MB** hoặc vài **GB**.*

TÀI LIỆU THAM KHẢO

[1]. Sơ đồ chân linh kiện bán dẫn, Dương Minh Trí, NXB KHKT, 2007.

[2]. Charles H.Roth, Fundamentals of Logic Design, 4th Ed., Prentice Hall, 1991.

[3]. John F.Wakerly, Digital Design Principles and Practices, Prentice Hall, 1990