

CHƯƠNG II: ĐẠI SỐ LOGIC

1. Khái niệm cơ bản của hệ thống số

1.1. Khái niệm tín hiệu số

Về cơ bản có hai cách biểu diễn giá trị của đại lượng, đó là tương tự (analog) và số (digital).

- Biểu diễn dạng tương tự: trong cách biểu diễn dạng tương tự, một đại lượng được biểu diễn bằng hiệu điện thế, cường độ dòng điện, hay số đo chuyển động tương quan với giá trị của đại lượng đó.

Ví dụ: Đồng hồ đo vận tốc trong xe ôtô, kim đo phai lệch tương ứng với tốc độ hiện tại của xe và độ lệch này phải thay đổi tức thì khi vận tốc xe tăng hay giảm.

Một ví dụ khác về đại lượng tương tự là chiếc micrô. Trong thiết bị này, biên độ hiệu điện thế đầu ra luôn tỉ lệ với cường độ sóng âm tác động vào màng rung của micrô ở đầu vào.

Các đại lượng tương tự có một đặc điểm rất quan trọng đó là: Đại lượng tương tự có thể thay đổi theo một khoảng giá trị liên tục.

- Biểu diễn dạng số: Trong cách biểu diễn dạng số, đại lượng được biểu diễn bằng các biểu tượng gọi là ký số (digit).

Ví dụ như đồng hồ hiện số, hiển thị thời gian trong ngày như giờ, phút, giây dưới dạng số thập phân. Tuy thời gian trong ngày thay đổi liên tục, nhưng số hiện của đồng hồ số lại thay đổi từng bước, mỗi bước là một phút hay một giây.

Nói cách khác, các đại lượng số có đặc điểm là giá trị của nó thay đổi theo từng bước rời rạc.

Vì tính rời rạc trong biểu diễn dạng số nên khi đọc giá trị của đại lượng số, không hề có sự mơ hồ.

a. *Ưu điểm của kỹ thuật số so với kỹ thuật tương tự:*

- Do sử dụng chuyển mạch nên nhìn chung thiết bị số dễ thiết kế hơn.
- Thông tin được lưu trữ dễ dàng
- Tính chính xác và độ tin cậy cao hơn
- Có thể lập trình để điều khiển hệ thống số.
- ít ảnh hưởng bởi nhiễu
- Nhiều mạch số có thể được tích hợp trên một chip IC

b. *Giới hạn của kỹ thuật số:*

Mặc dù hệ thống số có rất nhiều ưu điểm, nhưng bên cạnh đó vẫn có một số hạn chế. Do hầu hết các đại lượng vật lý đều có bản chất là tương tự, nên muốn tận dụng được hệ thống kỹ thuật số thì chúng ta phải thực hiện các bước sau:

- Biến đổi đầu vào dạng tương tự thành dạng số (A/D)
- Xử lý tín hiệu số
- Biến đổi đầu ra dạng số thành dạng tương tự (D/A)
- Tuy nhiên, quá trình trên được coi là quá trình tắt yếu đối với hệ thống số.

ở một số hệ thống, để tận dụng cả ưu điểm của kỹ thuật số và kỹ thuật tương tự người ta dùng cả hai hệ thống. Trong các hệ thống lai ghép này thì việc quan trọng là phải xác định được phần nào của hệ thống nên sử dụng kỹ thuật số và phần nào nên sử dụng kỹ thuật tương tự.

1.2. Trạng thái nhị phân và mức logic

Trong hệ thống kỹ thuật số, thông tin được xử lý đều biểu diễn dưới dạng nhị phân. Bất kỳ thiết bị nào chỉ có hai trạng thái hoạt động đều có thể biểu diễn được các đại lượng dưới dạng nhị phân.

Ví dụ một công tắc chỉ có hai trạng thái hoạt động là đóng hoặc mở. Ta có thể quy ước công tắc mở biểu diễn nhị phân 0 và công tắc đóng biểu diễn nhị phân 1. Với quy ước này ta có thể biểu diễn số nhị phân bất kỳ.

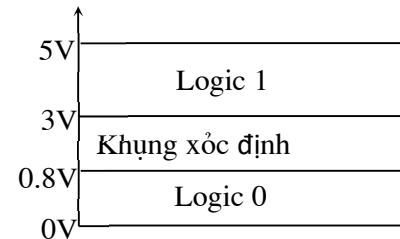
Có vô số thiết bị chỉ có hai trạng thái hoạt động hay vận hành ở hai điều kiện đối lập nhau như: bóng đèn (sáng/tối), đĩa (dẫn/không dẫn), role (ngắt/dòng),...

Trong thiết bị điện tử số, thông tin nhị phân được biểu diễn bằng hiệu điện thế (hay dòng điện) tại đầu vào hay đầu ra của mạch. Thông thường, số nhị phân 0 và 1 được biểu diễn bằng hai mức điện thế danh định. Ví dụ: 0V có thể biểu diễn bằng nhị phân 0 và +5V biểu diễn bằng nhị phân 1. Trên thực tế, các số 0 hoặc 1 được biểu diễn bằng một khoảng điện thế quy định nào đó.

Ví dụ:

Điện thế từ 0V đến 0.8V biểu thị nhị phân 0 và điện thế từ 3V đến 5V biểu diễn nhị phân 1.

Đối với hệ thống kỹ thuật số giá trị chính xác của hiệu điện thế hay dòng điện là không quan trọng, chỉ cần nó nằm trong khoảng uy định mức logic 0 hay 1.



2. Các phép toán logic và công logic cơ bản

Đại số logic còn được gọi là đại số Boole. Lý thuyết này do George Boole nhà toán học người Anh đưa ra năm 1847.

2.1. Cơ sở của đại số logic

Ta đã biết mạch số hoạt động ở chế độ nhị phân, nơi mỗi điện thế vào và ra sẽ có giá trị 0 hoặc 1; việc chỉ định giá trị 0 và 1 biểu thị khoảng điện thế định sẵn. Đặc điểm này của mạch logic cho phép sử dụng đại số logic làm công cụ phân tích và thiết kế các hệ thống kỹ thuật số.

Đại số logic dùng để phân tích hay thiết kế những mạch điện có quan hệ giữa biến và hàm. Trong đó biến và hàm chỉ nhận một trong hai giá trị là 0 và 1, hai giá trị này không biểu thị số lượng to nhỏ cụ thể mà chủ yếu là để biểu thị hai trạng thái logic khác nhau (đúng và sai, cao và thấp, mở và đóng,...).

Đại số logic là phương tiện biểu diễn mối quan hệ giữa đầu ra và đầu vào của mạch logic dưới dạng phương trình đại số. Đầu vào sẽ được xem là các biến logic có mức logic quyết định mức logic của đầu ra (hàm logic) tại thời điểm bất kỳ. Biến logic và hàm logic thường được ký hiệu bằng chữ cái.

Tóm lại ta có:

x_i là biến logic khi x_i chỉ lấy một trong hai giá trị là 0 và 1 ($x_i \approx \{0,1\}$).

Tập hợp n biến logic có 2^n tổ hợp giá trị khác nhau. Giá trị thập phân tương ứng biểu diễn các tổ hợp này là: $0 < 2^n - 1$.

$F(x_1, x_2, \dots, x_n)$ là hàm logic khi các biến của hàm là biến logic và F chỉ lấy một trong hai giá trị 0 hoặc 1.

Trong thực tế, đại số logic chỉ có ba phép toán cơ bản: OR, AND và NOT. Các phép toán cơ bản này được gọi là phép toán logic.

2.2. Các phép toán logic và các cỗng logic cơ bản

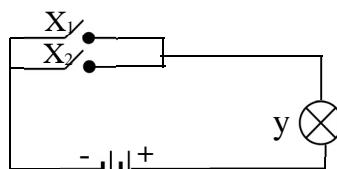
2.2.1. Phép toán OR và cỗng OR

a. Phép toán OR hay còn được gọi là phép cộng logic.

+ Hàm OR (hàm hoặc): $y = x_1 + x_2$

+ Bảng chân lý:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1



Hình 2.1.

+ Mạch điện minh họa quan hệ logic OR (hình 2.1):

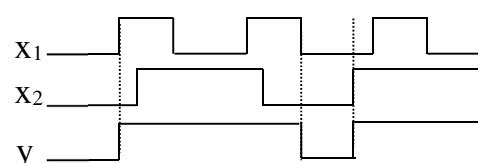
+ Mở rộng cho trường hợp tổng quát có n biến: $y = x_1 + x_2 + \dots + x_n$.

Mạch điện thực hiện quan hệ logic OR được gọi là cỗng OR.

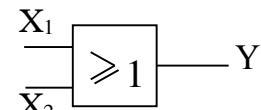
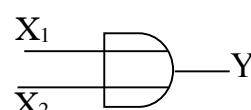
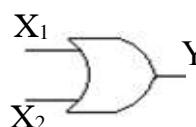
b. Cỗng OR:

+ Định nghĩa: Là mạch có từ hai đầu vào trở lên và có đầu ra bằng tổ hợp or các biến đầu vào.

+ Giản đồ thời gian:



+ Ký hiệu logic:



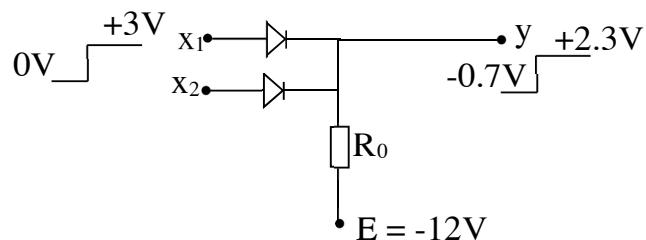
+ Mạch điện dùng diốt bán dẫn:

Điện áp sụt trên diốt khi phân cực thuận là 0.7V.

Khi $V_{x1} = V_{x2} = 0V$ thì

$V_y = 0V - 0.7V = -0.7V$.

Khi $V_{x1} = 0V, V_{x2} = 3V$ hoặc $V_{x1} = 3V, V_{x2} = 0V$ thì $V_y = 3V - 0.7V$



= 2.3V (do 2 diode có катод nối chung nên anode nào có điện thế cao hơn sẽ dẫn điện mạnh hơn làm cho diode kia chịu phân cực ngược và ở trạng thái ngắn hở mạch).

Khi $V_{x1} = V_{x2} = 3V$ thì $V_y = 3V - 0.7V = 2.3V$.

Nếu có n đầu vào thì mắc n diode tương tự như trên.

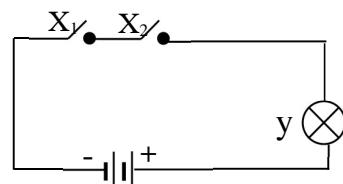
2.2.2. Phép toán AND và cỗng AND

a. Phép toán AND hay còn được gọi là phép nhân logic.

+ Hàm AND (hàm và): $y = x_1 \cdot x_2$

+ Bảng chân lý:

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



Hình 2.2

+ Mạch điện minh họa quan hệ logic AND (Hình 2.2):

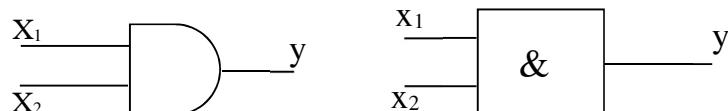
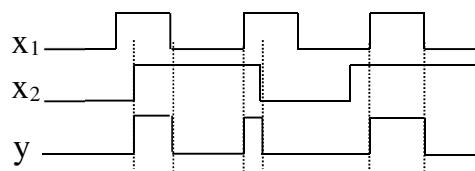
+ Mở rộng cho trường hợp tổng quát có n biến: $y = x_1 \cdot x_2 \dots \cdot x_n$.

Mạch điện thực hiện quan hệ logic AND được gọi là cỗng AND.

b. Cỗng AND

+ Định nghĩa: Là mạch có từ hai đầu vào trở lên và một đầu ra bằng tổ hợp AND các biến đầu vào.

+ Giản đồ thời gian:



+ Mạch điện dùng diode bán dẫn:

Điện áp sụt trên diode khi phân cực thuận là 0.7V.

Khi $V_{x1} = V_{x2} = 0V$ thì

$$V_y = 0V + 0.7V = 0.7V.$$

Khi $V_{x1} = 0V, V_{x2} = 3V$

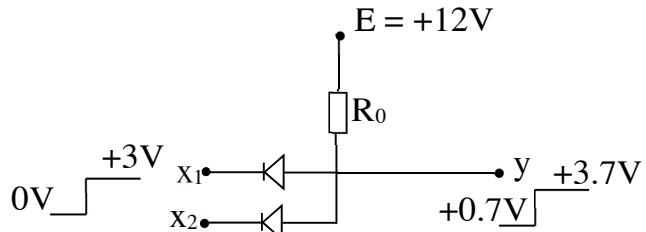
hoặc $V_{x1} = 3V, V_{x2} = 0V$ thì

$$V_y = 0V + 0.7V = 0.7V \text{ (do 2 diốt có anot nối chung nên katôt)}$$

nào có điện thế thấp hơn sẽ dẫn

điện mạnh hơn làm cho diốt kia chịu phân cực ngược và ở trạng thái ngắt hở mạch). Khi $V_{x1} = V_{x2} = 3V$ thì $V_y = 3V + 0.7V = 3.7V$.

Nếu có n đầu vào thì mắc n diốt tương tự.



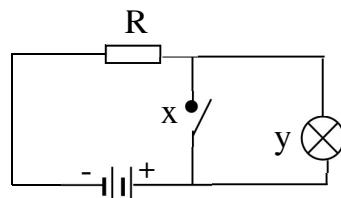
2.2.3. Phép toán NOT và cỗng NOT

a. Phép toán NOT hay còn được gọi phép đảo hay phép phủ định

+ Hàm NOT (hàm đảo): $y = \bar{x}$

+ Bảng chân lý:

x	y
0	1
1	0



Hình 2.3.

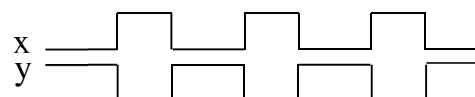
+ Mạch điện minh họa quan hệ logic NOT (Hình 2.3.):

Mạch điện thực hiện quan hệ logic NOT được gọi là cỗng NOT.

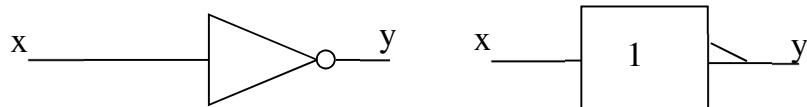
b. Cỗng NOT

+ Định nghĩa: Là mạch có duy nhất một đầu vào và mức logic ở đầu ra luôn ngược với mức logic ở đầu vào.

+ Giản đồ thời gian:

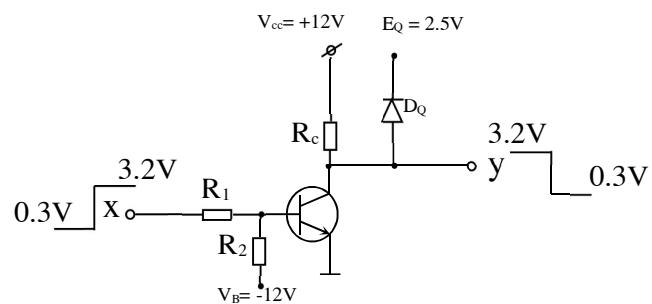


+ Ký hiệu logic:



+ Mạch điện:

Trong cổng NOT, tranzito làm việc ở chế độ đóng mở. Khi x ở mức thấp thì T ngắt hở mạch, y ở mức cao. Khi x ở mức cao thì T thông bão hòa, y ở mức thấp. Tác dụng của nguồn âm E_B là đảm bảo T ngắt hở mạch tin cậy khi x ở mức thấp. E_Q và D_Q có tác dụng giữ mức cao đầu ra ở giá trị quy định.



3. Các phương pháp biểu diễn hàm logic

3.1. Các định luật cơ bản của Đại số logic

Các mệnh đề cơ sở:

$$x + 0 = x$$

$$x + 1 = 1$$

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

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

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

1. Định luật đồng nhất:

$$x + x = x$$

$$x \cdot x = x$$

2. Định luật phủ định của phủ định:

$$\overline{\overline{x}} = x$$

3. Định luật kết hợp:

$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$$

$$x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3$$

4. Định luật giao hoán:

$$x_1 + x_2 = x_2 + x_1$$

$$x_1 \cdot x_2 = x_2 \cdot x_1$$

5. Định luật phân phối:

$$x_1(x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$$

$$(x_1 + x_2)(x_1 + x_3) = x_1 \cdot x_1 + x_1 \cdot x_3 + x_2 \cdot x_1 + x_2 \cdot x_3$$

$$= x_1(1 + x_2 + x_3) + x_2 \cdot x_3 = x_1 + x_2 \cdot x_3$$

6. Định lý Demorgan

$$\begin{aligned}\overline{x_1 \cdot x_2} &= \overline{x_1} + \overline{x_2} \\ \overline{x_1 + x_2} &= \overline{x_1} \cdot \overline{x_2}\end{aligned}$$

Định lý này có thể mở rộng cho hàm nhiều biến:

$$\begin{aligned}\overline{x_1 \cdot x_2 \dots x_n} &= \overline{x_1} + \overline{x_2} + \dots + \overline{x_n} \\ \overline{x_1 + x_2 + \dots + x_n} &= \overline{x_1} \cdot \overline{x_2} \dots \overline{x_n}\end{aligned}$$

Định lý này giúp ta chuyển phép cộng logic thành phép nhân logic và ngược lại. Vận dụng định lý De Morgan chúng ta có thể giải các bài toán thiết kế mạch logic tổ hợp theo các cửa logic cơ bản cho sẵn.

Chú ý: Trong các định luật trên x_i có thể là biến đơn hoặc biểu thức.

❖ Bài tập

1. Chứng minh các đẳng thức sau:

$$1) AB + A\bar{B} = A$$

$$2) A + AB = A$$

$$3) A + \bar{A}B = A + B$$

$$4) AB + \bar{A}C + BC = AB + \bar{A}C$$

2. Hãy tìm hàm đảo của các hàm logic dưới đây (dùng định lý De Morgan và các định luật):

$$a./ F = (\bar{A}\bar{B} + \bar{B}\bar{D})(AC + BD);$$

$$b./ F = AB + B\bar{D} + \bar{B}C + \bar{C}D$$

3.2. Các phương pháp biểu diễn hàm logic

Trước hết ta xét khái niệm hàm xác định đầy đủ và không xác định đầy đủ.

Hàm xác định đầy đủ là hàm có trị số xác định với mọi tổ hợp biến. Hàm không thoả mãn điều kiện trên là hàm không xác định đầy đủ. Tại những tổ hợp biến mà trị số của hàm không xác định (có thể là “0” hoặc “1”) giá trị của hàm sẽ được ký hiệu bằng dấu “x”. Những tổ hợp biến này cũng có thể không bao giờ xảy ra.

3.2.1 Biểu diễn bằng bảng chân lý

Tương tự như trong đại số thông thường, một hàm logic có thể được biểu diễn bởi bảng giá trị của của hàm số đó. Là bảng miêu tả quan hệ giữa các giá trị của hàm số tương ứng với mọi giá trị có thể của biến số.

Một hàm có n biến bảng sẽ có $(n+1)$ cột (trong đó n cột là giá trị của biến và một cột là giá trị của hàm) và 2^n hàng tương ứng với 2^n tổ hợp giá trị khác nhau của n biến vào. Ứng với mỗi tổ hợp giá trị biến ghi giá trị hàm tương ứng. Để khỏi bỏ sót hoặc trùng lặp ta nên sắp xếp các tổ hợp biến lối vào tuân tự theo số đếm nhị phân.

Ví dụ: Đèn báo hiệu của một hội đồng giám khảo gồm 3 thành viên sẽ sáng nếu đa số trong các thành viên đều đóng công tắc bỏ phiếu thuận. Lập bảng chân lý của hàm số logic đó.

Giải: Gọi A, B, C là ba công tắc, công tắc đóng thì các biến A, B, C lấy giá trị 1, công tắc ngắt thì các biến lấy giá trị 0. Gọi F là trạng thái của đèn được điều khiển, đèn sáng F = 1, đèn tắt F = 0. Ta được bảng chân lý:

Lối vào			Lối ra
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	1
1	1	0	1
1	1	1	1

Dùng bảng chân lý để biểu diễn hàm tuy có ưu điểm là rõ ràng trực quan, nhưng có nhược điểm là cách biểu diễn này sẽ trở nên rối rắm khi hàm có nhiều biến

❖ Bài tập

- Cho hàm F có ba biến A, B, C; ba biến này không bao giờ cùng ở mức cao hay cùng ở mức thấp. Hàm có mức logic cao khi có hai đầu vào có mức logic cao, trong trường hợp còn lại hàm có mức logic thấp. Hãy lập bảng chân lý biểu diễn hàm.
- Một bóng đèn đường cần đóng, ngắt độc lập ở 4 nơi khác nhau. Lập bảng chân lý của hàm logic đó.

3.2.2. Biểu diễn bằng phương trình logic

Trước hết ta xét khái niệm về minterm (số hạng tối thiểu) và maxterm (số hạng tối đa):

Một hàm logic có n biến, mỗi biến có thể nhận một trong hai giá trị 0 hoặc 1 như vậy ta sẽ có 2^n tổ hợp biến. Mỗi tổ hợp biến ta có thể tạo thành một số hạng là tích tất cả các biến có trong cùng một tổ hợp biến. Trong các số hạng đó biến bằng 1 được giữ nguyên biến còn biến bằng 0 được viết đảo biến, các số hạng này được gọi là minterm (số hạng tối thiểu). Gọi là số hạng tối thiểu vì minterm là tích các biến có trong một tổ hợp biến, tích này chỉ bằng 1 khi tất cả các biến đều bằng 1. Như vậy ứng với mỗi một minterm ta chỉ tìm được một tổ hợp giá trị biến tương ứng để nó bằng 1 và chỉ có một tổ hợp biến mà thôi.

Mỗi tổ hợp biến ta cũng có thể tạo thành một số hạng là tổng tất cả các biến có trong cùng một tổ hợp biến. Trong các số hạng đó biến bằng 0 được giữ nguyên biến còn biến bằng 1 được viết đảo biến, các số hạng này được gọi là maxterm (số hạng tối đa). Maxterm là tổng tất cả các biến có trong tổ hợp biến nên chỉ cần trong các biến bằng 1 thì maxterm bằng 1, maxterm bằng 0 chỉ trong một trường hợp duy nhất ứng với tất cả các biến trong tổ hợp biến đều bằng 0. Như vậy, các trường hợp maxterm bằng 1 là tối đa, trường hợp minterm bằng 1 là tối thiểu. Một hàm có n biến ta có 2^n maxterm và 2^n minterm.

Ví dụ: Một hàm F(A,B,C) có ba biến là A, B, C ta có 8 tổ hợp biến được sắp xếp một cách trình tự theo nhị phân là: 000, 001, 010, 011, 100, 101, 110, 111. Tương ứng với 8 tổ hợp biến này ta có 8 số hạng tối thiểu minterm ký hiệu là m_0, m_1, \dots, m_7 , và 8 số hạng tối đa maxterm ký hiệu là M_0, M_1, \dots, M_7 .

Biến			Minterm	Maxterm
A	B	C		
0	0	0	$m_0 = \overline{A} \overline{B} \overline{C}$	$M_0 = A + B + C$
0	0	1	$m_1 = \overline{A} \overline{B} C$	$M_1 = A + B + \overline{C}$
0	1	0	$m_2 = \overline{A} B \overline{C}$	$M_2 = A + \overline{B} + C$
0	1	1	$m_3 = \overline{A} BC$	$M_3 = A + \overline{B} + \overline{C}$
1	0	0	$m_4 = A \overline{B} \overline{C}$	$M_4 = \overline{A} + B + C$
1	0	1	$m_5 = A \overline{B} C$	$M_5 = \overline{A} + B + \overline{C}$
1	1	0	$m_6 = A B \overline{C}$	$M_6 = \overline{A} + \overline{B} + C$
1	1	1	$m_7 = A B C$	$M_7 = \overline{A} + \overline{B} + \overline{C}$

Trong một minterm và maxterm có mặt tất cả các biến số có trong tổ hợp biến của hàm, các biến số này chỉ xuất hiện một lần dưới dạng trực tiếp hoặc dạng đảo. Hàm logic có thể được biểu diễn dưới dạng là tổng các minterm hoặc tích các maxterm.

+ Các tính chất của maxterm và minterm:

- Hai minterm và maxterm của số hạng có cùng chỉ số là phủ định của nhau.

Ví dụ: $m_0 = \overline{M}_0$

- Tổng logic của tất cả các minterm = 1.
- Tích logic của tất cả các maxterm = 0.
- Tích hai minterm khác nhau bất kỳ = 0.
- Tổng hai maxterm khác nhau bất kỳ = 1.

+ Phương pháp biểu diễn:

Biểu diễn hàm logic bằng các phương trình logic cho thấy rõ mối quan hệ giữa hàm và biến thông qua các phép toán logic cơ bản là phương pháp biểu diễn thích hợp trong mọi trường hợp kể cả các quan hệ logic phức tạp, hàm có nhiều biến. Dùng phương trình logic biểu diễn hàm sẽ đơn giản gọn ghẽ hơn là dùng bảng chân lý và rất tiện để thực hiện các phép toán logic và tối thiểu hóa hàm bằng phương pháp đại số.

Phương trình logic có thể được xác lập theo các cách sau:

Cách 1: Biểu diễn hàm dưới dạng chuẩn tắc tuyển (CTT) - lấy tổng của các tích túc là lấy

tổng các minterm : $F = \sum_0^{2^n-1} f_i m_i$ (f_i là giá trị của hàm tương ứng với tổ hợp thứ i)

Như vậy ta chỉ lấy tổng các minterm tương ứng với $f_i = 1$.

Cách 2: Biểu diễn hàm dưới dạng chuẩn tắc hội (CTH) - lấy tích của các tổng túc là lấy tích của các maxterm: $F = \prod_i (f_i + M_i)$. Như vậy chỉ lấy tích của các maxterm tương ứng với $f_i=0$.

Ví dụ 1: Một hàm ba biến có bảng chân lý như sau:

i	A	B	C	F	minterm	Maxterm
0	0	0	0	1	m_0	M_0
1	0	0	1	0	m_1	M_1
2	0	1	0	0	m_2	M_2
3	0	1	1	1	m_3	M_3
4	1	0	0	1	m_4	M_4
5	1	0	1	0	m_5	M_5
6	1	1	0	0	m_6	M_6
7	1	1	1	1	m_7	M_7

Ta có thể xác định hàm logic theo hai cách nói trên:

Cách 1: Lấy tổng chuẩn các minterm ứng với $f_i = 1$ ta được:

$$F = m_0 + m_3 + m_4 + m_7$$

$$F = \overline{A} \overline{B} \overline{C} + \overline{A} BC + A \overline{B} \overline{C} + ABC$$

$$\text{Ký hiệu: } F(A,B,C) = \sum_m 0,3,4,7$$

0, 3, 4, 7 là các giá trị thập phân tương ứng với các tổ hợp nhị phân mà hàm có giá trị bằng 1.

Cách 2: Lấy tích chuẩn các maxterm ứng với $f_i = 0$ ta được:

$$F = M_1.M_2.M_5M_6$$

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

$$\text{Ký hiệu: } F(A,B,C) = \prod(1,2,5,6)$$

1, 2, 5, 6 là các giá trị thập phân tương ứng với các tổ hợp nhị phân mà hàm có giá trị bằng 0.

Hàm logic F xác định theo hai cách trên là như nhau.

Ví dụ 2: Cho bảng chân lý của hàm không xác định đầy đủ có ba biến như sau:

i	A	B	C	F	minterm	Maxterm
0	0	0	0	X	m_0	M_0
1	0	0	1	1	m_1	M_1
2	0	1	0	0	m_2	M_2
3	0	1	1	1	m_3	M_3
4	1	0	0	1	m_4	M_4
5	1	0	1	0	m_5	M_5
6	1	1	0	0	m_6	M_6
7	1	1	1	X	m_7	M_7

Ta có: $F(A, B, C) = \sum_m (1, 3, 4)$ với $N = 0, 7$ hoặc $F(A, B, C) = \prod (2, 5, 6)$ với $N = 0, 7$.

ở đây $N = 0, 7$ để chỉ rằng các tổ hợp ứng với các giá trị thập phân đó hàm có giá trị không xác định.

❖ Bài tập

1. Cho hàm F có bảng chân lý như sau:

A	B	C	F
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	1
1	1	1	0

Biểu diễn hàm bằng phương trình logic dưới dạng CTT và CTH.

2. Thành lập bảng chân lý cho hàm số sau: $F = ABD + \bar{B}CD + \bar{A}\bar{C}$

3.2.3 Biểu diễn bằng bảng Karnaugh

Khi một hàm logic có số lượng biến tương đối nhỏ ($k \leq 6$) người ta thường biểu diễn chúng dưới dạng một bảng gọi là bảng Karnaugh (Các nô). Theo phương pháp này một

hàm có n biến được biểu diễn trên một bảng gồm 2^n ô vuông. Mỗi ô vuông tương ứng với 1 hàng trong bảng chân lý. Lưu ý rằng các tổ hợp biến ở đây được xếp theo thứ tự của mã Gray tức là hai ô liền kề các minterm chỉ khác nhau có một bit.

Trong các ô của bảng K ghi giá trị của hàm tương ứng.

Lưu ý: các tổ hợp biến hàm có giá trị 0 thì có thể bỏ trống hoặc ghi 0.

Trên bảng 2.4 là bảng Karnaugh của một số hàm logic có 2,3,4,5 biến, ở dưới mỗi bảng là phương trình logic tương ứng của các hàm này.

❖ Bài tập

1. Cho hàm F có bảng chân lý, hãy biểu diễn hàm bằng bảng Karnaugh.

A	B	C	F
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	1
1	1	1	0

2. Cho các hàm logic có phương trình như sau:

a./ $F_1(A, B, C) = \Sigma(0, 2, 4, 6)$ với $N = 1, 3$

b./ $F_2(A, B, C, D) = \Pi(1, 2, 3, 6, 8, 9, 11, 12)$

Biểu diễn hàm bằng bảng Karnaugh.

A	0		1
	0	2	1
B	1	1	3
	1	1	

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

C	AB		00	01	11	10
	0	1	2	1	6	4
B	1	1	3		7	5
	1	1				1

$$F(A, B, C) = \sum_m(0, 1, 2, 5)$$

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

$$F = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D}$$

$$+ A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}CD + A\bar{B}C\bar{D}$$

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

$$F(A, B, C, D) = \sum_m (0, 2, 3, 8, 9, 10, 11, 13, 15)$$

		ABC	000	001	011	010	110	111	101	100
		DE	0 1	4	12	8 1	24 1	28	20	16 1
		00	1	5	13	9 1	25 x	29	21 1	17 1
		01	3 1	7	15	11 1	27 1	31	23 1	19 1
		11	2 1	6	14	10 1	26 1	30	22	18 1
		10								

$$F(A, B, C, D, E) = \sum_m (0, 2, 3, 8, 9, 10, 11, 16, 17, 18, 19, 21, 23, 24, 26, 27)$$

3. Cho hàm logic có phương trình sau: $F = ABC + \bar{A}BC + \bar{B}C + ABCD + \bar{C}\bar{D}$

Biểu diễn hàm bằng bảng Karnaugh.

4. Cho hàm logic có phương trình sau: $F = (A + B)(\bar{B} + C + D)(\bar{A} + \bar{C})$

Biểu diễn hàm bằng bảng Karnaugh.

3.2.4. Biểu diễn bằng sơ đồ logic

+ Cách vẽ sơ đồ logic của hàm logic:

Ta dùng ký hiệu logic của mạch điện tử thay thế phép tính logic có trong biểu thức hàm logic thì được sơ đồ logic của hàm.

Ví dụ: Hàm $F = AB + BC + AC$

Vẽ sơ đồ logic của hàm.

Giải:

Sơ đồ logic như hình 2.5. Thay phép toán OR bằng ký hiệu OR và phép toán AND bằng ký hiệu AND.

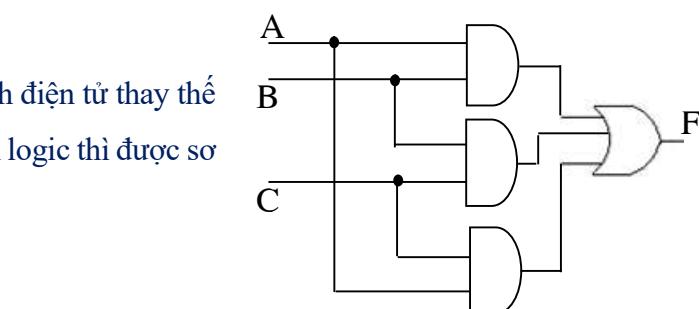
+ Cách xác định biểu thức từ sơ đồ logic:

Trên sơ đồ logic, từ đầu vào đến đầu ra, viết biểu thức hàm đầu ra của từng cấp, cuối cùng được biểu thức hàm logic toàn sơ đồ.

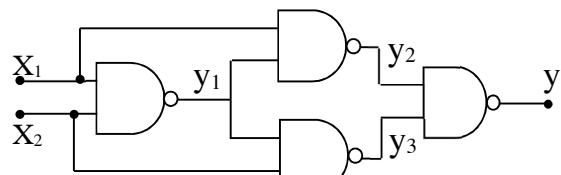
Ví dụ: Cho sơ đồ logic như hình 2.6a, hãy viết biểu thức hàm logic của sơ đồ.

Giải:

$$\begin{aligned}
 y_1 &= \overline{x_1 \cdot x_2} = (\overline{x_1} + \overline{x_2}) \\
 y_2 &= \overline{x_1 \cdot y_1} \\
 y_3 &= \overline{x_2 \cdot y_1} \\
 \text{Tacó: } y &= \overline{y_2 \cdot y_3} = \overline{\overline{x_1 \cdot y_1} \cdot \overline{x_2 \cdot y_1}} \\
 &= x_1 \cdot y_1 + x_2 \cdot y_1 = y_1(x_1 + x_2) \\
 &= (\overline{x_1} + \overline{x_2})(x_1 + x_2) = \overline{x_1} x_2 + x_1 \overline{x_2}
 \end{aligned}$$



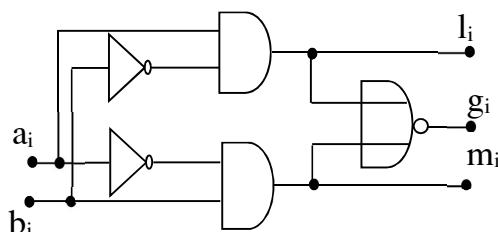
Hình 2.5



Hình 2.6a

❖ Bài tập

Cho sơ đồ logic như hình 2.6b, viết phương trình hàm logic F.



Hình:

4. Một số hàm Logic khác và phương pháp tối thiểu hóa hàm logic

4.1. Hàm NOR và hàm NAND

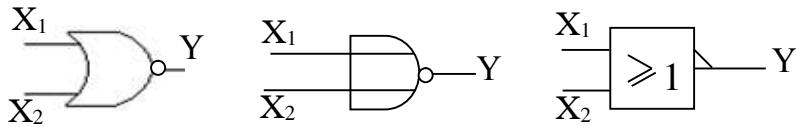
4.1.1. Hàm NOR (không hoặc: NOT - OR)

+ Hàm logic: $y = \overline{x_1 + x_2}$

+ Bảng chân lý:

x ₁	x ₂	y
0	0	1
0	1	0
1	0	0
1	1	0

+ Ký hiệu logic:



+ Trong trường hợp tổng quát nếu n biến ta cũng có:

$$y = \overline{x_1 + x_2 + \dots + x_n}$$

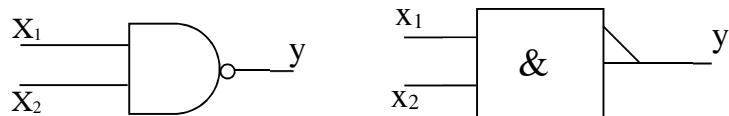
4.1.2. Hàm NAND (không và: NOT - AND)

+ Hàm logic: $y = \overline{x_1 \cdot x_2}$

+ Bảng chân lý:

x ₁	x ₂	y
0	0	1
0	1	1
1	0	1
1	1	0

+ Ký hiệu logic:



Tổng quát nếu có n biến ta cũng có:

$$y = \overline{x_1 \cdot x_2 \cdot \dots \cdot x_n}$$

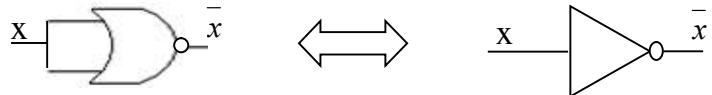
4.1.3. Tính đa dụng của cổng NOR và cổng NAND

Tất cả các biểu thức Boole đều kết hợp 3 phép toán cơ bản OR, AND và NOT. Do đó, bất kỳ biểu thức nào cũng đều có thể được thực hiện bằng cách dùng cổng OR, AND và

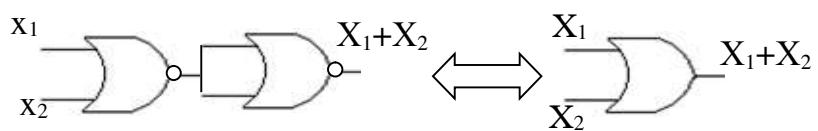
NOT. Tuy nhiên, có thể thực hiện biểu thức logic bất kỳ chỉ dùng cổng NOR hoặc NAND mà không cần thêm loại cổng nào khác.

a. Dùng cổng NOR thay cho ba cổng logic cơ bản:

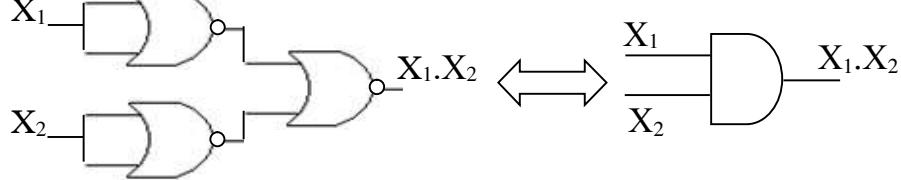
$$+ \overline{x+x} = \bar{x}$$



$$+ \overline{\overline{x_1+x_2}} = x_1 + x_2$$



$$+ \overline{\overline{x_1+x_2}} = x_1 \cdot x_2$$



❖ Bài tập

Thực hiện các hàm logic sau:

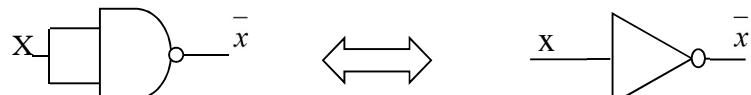
$$F_1 = AB + \overline{B}\overline{C} + BC; \quad F_2 = A(\overline{B} + CD);$$

a./ Bằng cổng NOR hai đầu vào

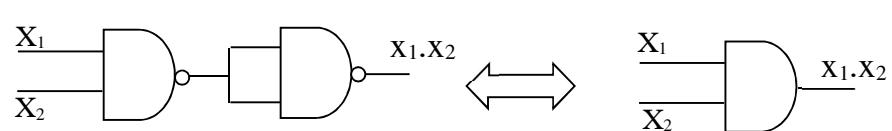
b./ Bằng cổng NOR có số đầu vào tùy ý.

b. Dùng cổng NAND thay cho ba cổng logic cơ bản:

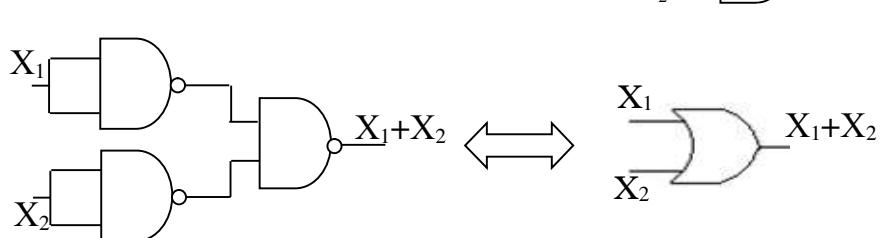
$$+ \overline{x \cdot x} = \bar{x}$$



$$+ \overline{\overline{x_1 \cdot x_2}} = x_1 \cdot x_2$$



$$+ \overline{\overline{x_1 \cdot x_2}} = x_1 + x_2$$



❖ Bài tập

Thực hiện các hàm logic sau:

$$F_1 = AB + \overline{B}\overline{C} + BC; \quad F_2 = A(\overline{B} + CD);$$

- a./ Bằng công NAND hai đầu vào
- b./ Bằng công NAND có số đầu vào tùy ý.

4.2. Hàm XOR và hàm XNOR

4.2.1. Hàm XOR (Exclusive - OR)

Hàm hoặc loại trừ hay còn được gọi là hàm hoặc tuyệt đối, hàm cộng modul 2, hàm không tương đương, hàm khác dấu,...

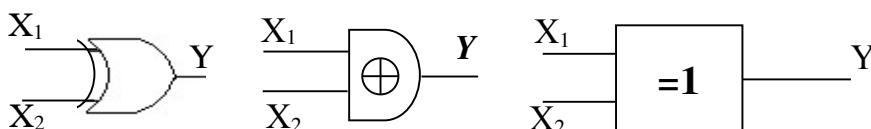
+ Hàm logic: $y = x_1 \cdot \overline{x}_2 + \overline{x}_1 \cdot x_2$

Được viết là: $y = x_1 \oplus x_2$

+ Bảng chân lý:

x ₁	x ₂	y
0	0	0
0	1	1
1	0	1
1	1	0

+ Ký hiệu logic:

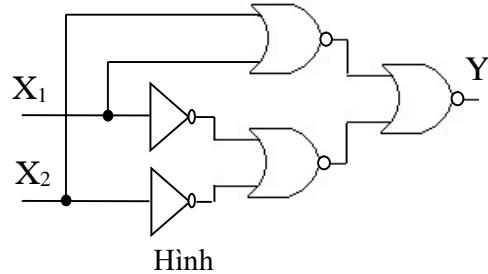
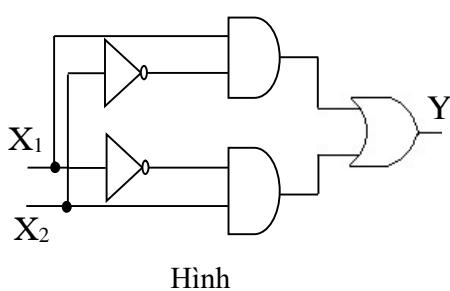


+ Sơ đồ logic tương ứng với phương trình trên được trình bày trên hình 2.7.

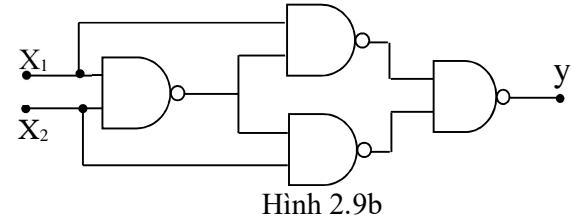
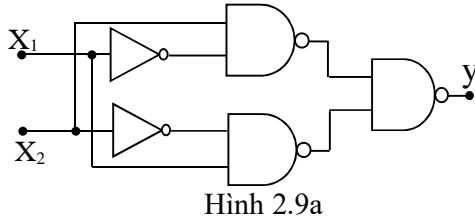
+ Ta có thể thiết kế sơ đồ mạch XOR bằng NOT và NOR:

$$y = \overline{\overline{x}_1 \cdot x_2} + \overline{x_1 \cdot \overline{x}_2} = \overline{x_1 + x_2} + \overline{\overline{x}_1 + \overline{x}_2}$$

Sơ đồ logic của mạch tương ứng với phương trình này được trình bày trên hình 2.8.



+ Ta cũng có thể thiết kế mạch XOR bằng các cổng NOT và NAND (hình 2.9a), hoặc chỉ bằng cổng NAND (hình 2.9b).



+ Các tính chất của hàm XOR:

- Tính giao hoán: $A \oplus B = B \oplus A$
- Tính kết hợp: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
- Tính phân phối: $A(B \oplus C) = AB \oplus AC$

Ngoài ra ta còn có một số tính chất sau:

$$A \oplus 0 = A; \quad A \oplus 1 = \overline{A}$$

$$A \oplus A = 0; \quad A \oplus \overline{A} = 1$$

$$\overline{A \oplus B} = \overline{A} \oplus \overline{B} = A \oplus \overline{B}$$

$$A \oplus B = \overline{A} \oplus \overline{B}$$

$$A \oplus B = C \Leftrightarrow A \oplus C = B \Leftrightarrow B \oplus C = A$$

+ Hàm XOR nhiều biến:

Dùng tính chất kết hợp ta cũng có thể xây dựng được các mạch XOR nhiều lõi vào từ mạch XOR hai lõi vào này. Lấy mạch XOR 3 lõi vào làm ví dụ:

- Hàm logic: $y = x_1 \oplus x_2 \oplus x_3$

- Bảng chân lý:

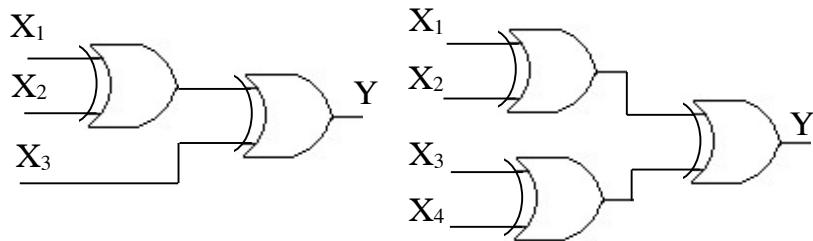
x_1	x_2	x_3	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- Nhận xét bảng chân lý:

- Nếu số mức logic 1 ở lõi vào là lẻ \square lõi ra là mức logic 1.
- Nếu số mức logic 1 ở lõi vào là chẵn \square lõi ra là mức logic 0.

Nguyên tắc này áp dụng cho các mạch XOR nhiều lối vào (n lối vào).

- + Ta có thể xây dựng cổng XOR 3 lối vào hoặc 4 lối vào từ các cổng XOR 2 lối vào như hình 2.10.



Hình 2.10.

4.2.2. Hàm XNOR (Exclusive-NOR)

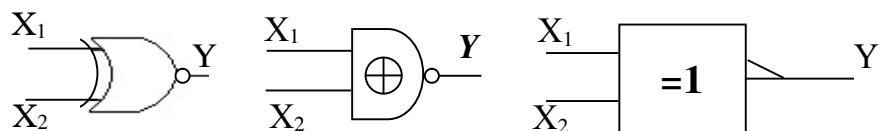
Hàm không hoặc loại trừ hay còn được gọi là hàm không hoặc tuyệt đối, hàm tương đương, hàm cùng dấu,...

- + Hàm logic: $y = x_1 x_2 + \overline{x_1} \overline{x_2} = \overline{x_1 \oplus x_2} = x_1 \sim x_2$

- + Bảng chân lý:

x ₁	x ₂	y
0	0	1
0	1	0
1	0	0
1	1	1

- + Ký hiệu logic:

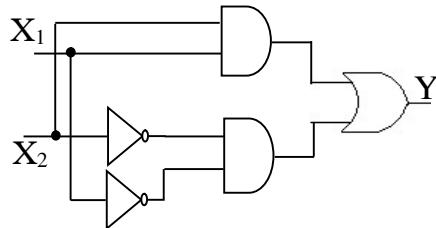


Nhận xét: Lối ra của XNOR 2 lối vào là đảo của XOR 2 lối vào.

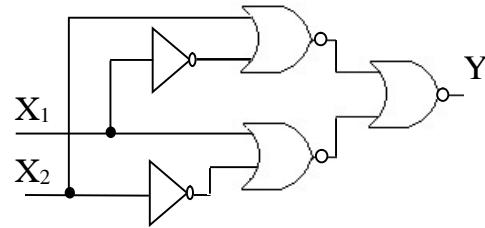
Ta cũng có thể xây dựng được các cổng XNOR nhiều lối vào bằng cách tương tự như xây dựng XOR nhiều lối vào. Từ các phần tử logic cơ bản AND, OR, NOT hoặc NAND và NOT hay NOR và NOT hoặc chỉ bằng NAND và chỉ bằng NOR ta có thể tạo được các cổng XNOR.

Trên hình 2.11a giới thiệu sơ đồ logic mạch XNOR hai lối vào được xây dựng từ các phần tử logic cơ bản NOT, AND và OR.

Trên hình 2.11b là sơ đồ logic mạch XNOR hai lối vào được xây dựng từ các phần tử logic NOT và NOR.



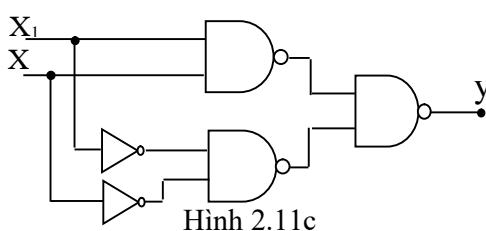
Hình 2.11a.



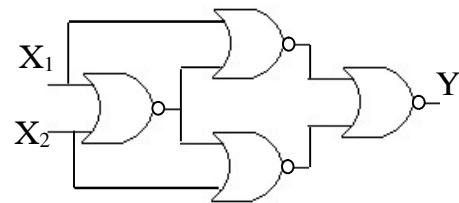
Hình 2.11b.

Trên hình 2.11c vẽ sơ đồ logic của mạch XNOR 2 lối vào được tạo nên từ cổng NOT và NAND.

Trên hình 2.11d vẽ sơ đồ logic mạch XNOR 2 lối vào chỉ dùng các cổng NOR.



Hình 2.11c



Hình 2.11d

+Các tính chất của hàm XNOR

$A \sim 0 = \bar{A}$	$A \sim \bar{A} = 0$	$\overline{A \sim B} = \bar{A} \sim B = A \sim \bar{B}$
$A \sim A = 1$	$A \sim B = B \sim A$	$A \sim B = \bar{A} \sim \bar{B}$
$A \sim 1 = A$	$A \sim (B \sim C) = (A \sim B) \sim C$	$A \sim B = C \Leftrightarrow A \sim C = B \Leftrightarrow B \sim C = A$

❖ Bài tập

Xây dựng sơ đồ mạch tổ hợp dùng các cổng XOR 2 đầu vào cho bởi hàm sau:

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

4.3. Các phương pháp tối thiểu hóa hàm logic

Trong việc thiết kế các khối chức năng logic, tìm ra được một sơ đồ logic đơn giản đáp ứng đầy đủ các yêu cầu của khối chức năng cần thiết kế, thì yêu cầu hàng đầu của công tác thiết kế các mạch điện tử là tính kinh tế và mạch phải có tính ổn định độ tin cậy cao. Để đảm bảo các yêu cầu này thì sơ đồ logic phải bao gồm số các phần tử logic cơ bản ít nhất,

các sơ đồ càng đơn giản càng có độ tin cậy và ổn định cao. Để xây dựng được một sơ đồ như vậy chúng ta phải tìm ra được một phương trình logic tối giản mô tả đúng chức năng logic của mạch điện tử cần thiết kế. Các hàm logic mà ta thường gặp thường không phải là dạng tối giản, nếu ta xây dựng mạch dựa trên phương trình này thì sẽ tốn kém vì phải dùng nhiều phần tử linh kiện logic, sơ đồ càng phức tạp độ ổn định, độ tin cậy càng kém xác suất hư hỏng càng tăng. Vì thế trước khi xây dựng mạch bao giờ cũng phải tìm cách rút gọn hàm đưa phương trình biểu diễn về dạng tối giản. Phương trình ở dạng tối giản khi các số hạng phải là ít nhất và số biến trong mỗi số hạng cũng phải là ít nhất.

4.3.1. Tối thiểu hóa hàm logic bằng phương pháp đại số

áp dụng các định luật của đại số logic để đơn giản hàm logic sao cho hàm cuối cùng là tối giản, thực hiện hàm cần ít phần tử logic cơ bản nhất. Vì trong thực tế các biểu thức logic rất đa dạng, từ một hàm logic cũng có thể biểu diễn theo nhiều cách khác nhau nên khó có thể tìm ra một quy trình tối ưu để tìm ra được một biểu thức logic tối giản một cách nhanh nhất. Tuy nhiên, nếu nắm chắc các định luật của đại số Boole và có kinh nghiệm chúng ta có thể thu được kết quả tốt.

+ Một số công thức thường dùng:

$$1) AB + A\bar{B} = A$$

$$2) A + AB = A$$

$$3) A + \bar{A}B = A + B$$

$$4) AB + \bar{A}C + BC = AB + \bar{A}C$$

Từ công thức (4) ta có hệ quả: $AB + \bar{A}C + BCD = AB + \bar{A}C$

Ví dụ: Tối thiểu hóa hàm logic sau:

$$a) F = A\bar{B}C + A\bar{B}\bar{C} = A\bar{B}(C + \bar{C}) = A\bar{B}$$

$$b) F = A(BC + \bar{B}\bar{C}) + A(\bar{B}C + B\bar{C}) = A(BC + \bar{B}\bar{C} + \overline{BC + \bar{B}\bar{C}}) = A$$

$$c) F = AB + \bar{A}C + \bar{B}C = AB + \bar{A}C + BC + \bar{B}C = AB + C(\bar{A} + B + \bar{B}) = AB + C$$

$$\text{Hoặc : } F = AB + (\bar{A} + \bar{B})C = AB + \bar{A}\bar{B}C = AB + C$$

$$d) F = A\bar{B} + B\bar{C} + \bar{B}C + \bar{A}B = A\bar{B} + B\bar{C} + A\bar{C} + \bar{B}C + \bar{A}B = \\ = (A\bar{B} + A\bar{C} + \bar{B}C) + (B\bar{C} + A\bar{C} + \bar{A}B) = A\bar{C} + \bar{B}C + \bar{A}B$$

❖ Bài tập

1. Tối thiểu hóa các hàm sau bằng phương pháp đại số:

$$a. \quad F(A, B, C) = \sum_m 0, 2, 3, 4, 6; \quad b. \quad F(A, B, C) = \prod 1, 2, 3, 6, 7$$

2. Tối thiểu hoá các hàm sau bằng phương pháp đại số:

$$a. \quad A(\bar{A}C + BD) + B(C + DE) + B\bar{C}$$

$$b. \quad \bar{A}\bar{B} + AC + \bar{C}D + \bar{B}\bar{C}\bar{D} + B\bar{C}E + \bar{B}CG + \bar{B}CF$$

4.3.2. *Tối thiểu hoá bằng bảng Karnaugh*

- Cho hàm dạng CTT:

Phương pháp này được tiến hành theo các bước sau:

1. Biểu diễn hàm đã cho trên bảng Karnaugh.
2. Kết hợp thành từng nhóm 2^n ô gồm các ô có giá trị bằng “1” hoặc “x” kế cận hoặc đối xứng nhau tạo thành một vòng kín trên bảng Karnaugh. Khi kết hợp các ô cần tuân theo quy tắc sau:
 - Các ô kế cận hoặc đối xứng nhau là các ô chỉ khác nhau 1 bit.
 - Số ô chứa trong 1 nhóm phải là tối đa (2^n ô với n là tối đa).
 - Trong mỗi nhóm phải có ít nhất một ô chứa giá trị “1” không nằm trong nhóm khác, nhóm nào bao gồm các ô chứa giá trị “1” đều đã có trong nhóm khác thì nhóm đó là thừa, mặt khác mỗi ô chứa giá trị “1” có thể được sử dụng để kết hợp nhiều lần.
 - Phải đảm bảo tất cả các ô chứa giá trị “1” đều được kết hợp và số nhóm kết hợp phải là tối thiểu.
3. Nhóm 2^n ô sẽ bỏ đi được n biến đó là những biến vừa xuất hiện ở cả dạng trực tiếp lẫn dạng đảo, số hạng tạo thành là tích các biến còn lại (gọi là tích cực tiểu). Kết quả là tổng các số hạng tạo thành từ các nhóm được kết hợp (dạng tổng các tích).
4. Trong một số trường hợp, có thể có nhiều cách kết hợp, nghĩa là có thể có nhiều hàm tối thiểu. Những hàm tối thiểu này cần được so sánh, kiểm tra để chọn ra hàm tối thiểu thực sự.

Ví dụ 1: Cho hàm $F(A, B, C) = \sum_m (0, 1, 2, 5)$ hãy tối thiểu hóa hàm bằng bảng Karnaugh.

Giải: Kết hợp các ô như bảng bên ta được ít nhất 2 nhóm phủ hết các ô chứa giá trị “1” của hàm, các ô đó ở kề nhau hoặc đối xứng nhau.

Ví dụ

$$F = \overline{A} \overline{C} + \overline{B} C$$

Ví dụ 1

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

CD	AB	00	01	11	10
00	0 1	4	12	8 1	
01	1	5	13	9 1	
11	3 1	7	15	11 1	
10	2 1	6	14	16 1	

Ví dụ 2: Cho hàm $F = \overline{A} \overline{B} \overline{C} \overline{D} + \overline{A} \overline{B} C D + \overline{A} \overline{B} C \overline{D} + A \overline{B}$

Hãy tối thiểu hóa hàm bằng bảng Karnaugh.

Giải: Kết hợp các ô như bảng bên ta được ít nhất 3 nhóm phủ hết các ô chứa giá trị “1” của hàm.

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

Ví dụ 3: Cho hàm $F(A,B,C,D) = \sum_m (0,2,3,8,9,10,11,13,15)$ với $N = 1$

Hãy tối thiểu hóa hàm bằng bảng Karnaugh.

Giải:

$$F = \overline{B} + AD$$

Vớ dụ 3

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

Vớ dụ 4

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

- Cho hàm dạng CTH:

Phương pháp tương tự như hàm ở dạng CTT, chỉ khác là thay các ô chứa giá trị “1” bằng các ô chứa giá trị “0” và thay tổng các tích bằng tích các tổng khi biểu diễn hàm.

Ví dụ 4:

Cho hàm $F(A, B, C, D) = \Pi(4, 5, 6, 7, 12, 14)$ với $N = 1$

Tối thiểu hóa hàm bằng bảng Karnaugh.

Giải:

$$F = (A + \bar{B})(\bar{B} + D)$$

Tùy thuộc vào yêu cầu thiết kế (loại phần tử sử dụng) mà ta lựa chọn phương pháp tối thiểu cho phù hợp. Chẳng hạn, chỉ dùng công NAND thì phương trình viết dưới dạng tổng các tích, chỉ dùng công NOR thì phương trình viết dưới dạng tích các tổng sau đó dùng luật phủ định của phủ định và định lý Morgan để biến đổi phương trình.

❖ Bài tập

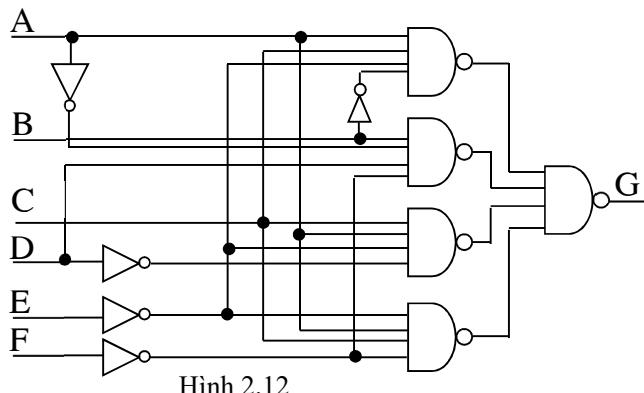
1. Tối thiểu hóa các hàm sau bằng bảng Cacnô:

$$F(A, B, C) = \sum_m 0, 1, 2, 3, 4, 5, 6; \quad F(A, B, C, D) = \sum_m 0, 1, 2, 3, 4, 9, 10, 12, 13, 14, 15$$
$$F = \bar{A}\bar{B}C + AD + B\bar{D} + C\bar{D} + A\bar{C} + \bar{A}\bar{D}$$

2. Vẽ sơ đồ mạch cho biểu thức sau với số công sử dụng là ít nhất:

$$F = ABCD + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{D}$$

4.15. Hãy đơn giản hóa mạch logic cho ở hình 2.12.



4.18. Thiết kế mạch logic tổ hợp có 4 đầu vào A, B, C, D và 2 đầu ra F₁, F₂ như sau:

$$F_1(A, B, C, D) = \sum_m (3, 4, 5, 7, 13, 15)$$

$$F_2(A, B, C, D) = \sum_m (3, 8, 9, 11, 13, 15)$$

a./ Chỉ dùng các công logic cơ bản.

b./ Chỉ dùng các công NAND.

c./ Chỉ dùng các cổng NOR.