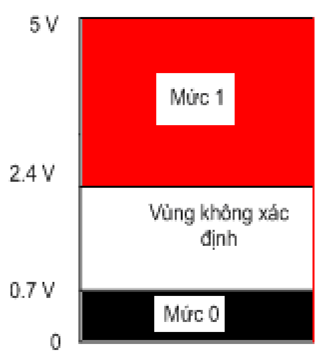
Bài 12: CỔNG LOGIC

Trong chương này chúng ta sẽ khảo sát qua các loại cổng logic thông dụng trong máy tính, các phương thức hoạt động và các ký hiệu của chúng. Đồng thời ta cũng nghiên cứu về đại số Boole, là đối tượng liên quan khá chặt chẽ đến các cổng. Đại số boole được phát minh bởi nhà toán học George Boole. Trong đại số này các biến chỉ mang một trong hai trạng thái: 0 và 1 (đúng hay sai) và cũng chính vì thế người ta còn gọi đại số boole là đại số lưỡng trạng. Riêng khả năng của nó chính là giải quyết rất tốt các bài toán về các mạch luận lý (logic) như đơn giản hoá mạch, mạch tương đương,…

Các mạch này còn được gọi là các cổng luận lý vì ta có thể dùng đại số boole để tính toán chúng.

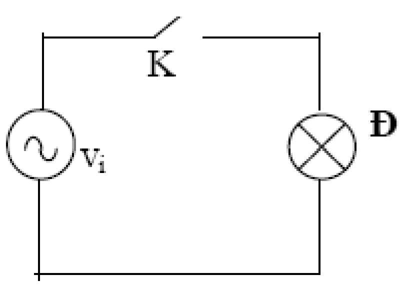
12.1 Biểu diễn các trạng thái logic 1 và 0

Trong hệ thống mạch logic, các trạng thái logic được biểu diễn bởi các mức điện thế. Với qui ước logic dương, điện thế cao biểu diễn logic "1", logic âm, điện thế thấp biểu diễn logic "0". Trong thực tế, mức "1" và "0" tương ứng với một khoảng điện thế xác định và có một khoảng chuyển tiếp giữa mức cao và thấp, ta gọi là khoảng không xác định. Khi điện áp của tín hiệu rơi vào khoảng này, mạch sẽ không nhận ra là mức "0" hay "1". Khoảng này tùy thuộc vào họ IC sử dụng và được cho trong bảng thông số kỹ thuật của linh kiện.



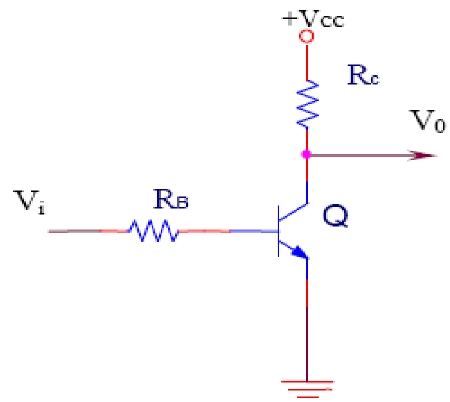
Đa số IC có:
Mức "0" : 0 -> 0.7v
Mức "1" : 2.4 v -> 5 v
Vùng không xác định: 2.4v -> 5v

Trạng thái "0" và "1" còn có thể được hiểu dưới dạng mạch điện như sau:



K mở ứng với logic "0" : đèn tắt
K đóng ứng với logic "1" : đèn sáng

Xét linh kiện điện tử Transistor lưỡng cực BJT có ba cực là: cực nền B – Base, cực thu C – Collector, cực phát E – Emitter. Điện trở R_B giới hạn dòng nền I_B, điện trở R_C giới hạn dòng thu I_C chạy qua tải ở ngõ ra.



Khi điện thế ngõ vào $V_I = 0$ thì dòng nền $I_B = 0$ khiến dòng thu $I_C = 0$ và điện thế ở ngõ ra là:

$V_o = V_{CC} - I_C R_C = V_{CC}$

(mức cao – mức “1”)

Khi điện thế ngõ vào $V_I = V_{CC}$ thì I_B lớn, Transistor dẫn bão hòa, khi đó điện thế ngõ ra là:

$V_o = V_{CESat} = 0{,}2$ V (mức thấp – mức “0”)

12.2 Các cổng logic cơ cản

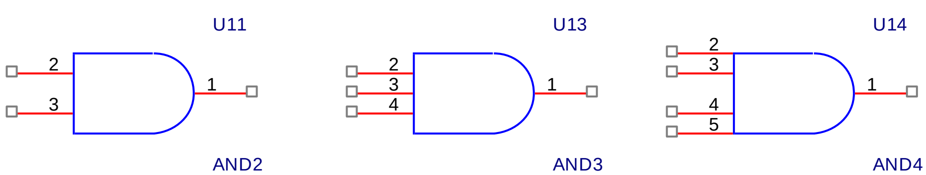
12.2.1 Cổng AND

Khái niệm: Là cổng có 2 hay nhiều ngõ vào nhưng chỉ có một ngõ ra. Giá trị ngõ ra phụ thuộc vào nhóm các giá trị vào. Cổng AND thực hiện phép toán nhân Logic với tín hiệu vào.

Phương Trình Mô Tả Hoạt Động của Cổng AND n Lối vào

$Y = X_1X_{2\ldots}X_n$

Ký Hiệu:



Hình: Cổng AND có 2, 3, 4 ngõ vào

Đặc Điểm:

Nếu tất cả ngõ vào có giá trị 1 thì ngõ ra có giá trị 1, ngõ ra bằng 0 khi có ít nhất một ngõ vào bằng 0.

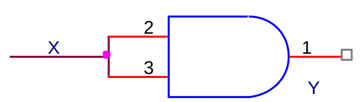
Bảng Trạng Thái:

Xét bảng chân trị của cổng AND có 2 và 3 ngõ vào

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	C	A AND B AND C
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

Sử dụng cổng AND để tạo ra cổng logic khác như sơ đồ sau:



Bảng trạng thái:

X	Y
0	0
1	1

Đây chính là cổng đệm giá trị ngõ ra: Y = X

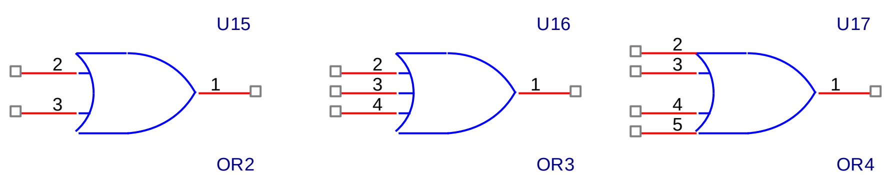
12.2.2 Cổng OR

Khái Niệm: Là cổng có 2 hay nhiều ngõ vào nhưng chỉ có một ngõ ra, thực hiện chức năng của phép cộng logic tín hiệu vào

Phương Trình Mô Tả Hoạt Động của Cổng OR n Lối vào

$Y = X_1 + X_2 + _{\ldots} + X_n$

Ký Hiệu:



Cổng OR có 2, 3, 4 ngõ vào

Đặc Điểm:

Nếu chỉ một trong các ngõ vào có giá trị 1 thì ngõ ra có giá trị 1 và nếu tất cả ngõ vào có giá trị 0 thì ngõ ra có giá trị 0.

Bảng Trạng Thái:

Xét bảng chân trị của cổng OR có 2 và 3 ngõ vào

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	C	A OR B OR C
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

Tương tự như cổng AND thì cổng OR cũng có khả năng làm cổng đệm khi ta nối các ngõ tín hiệu đầu vào lại với nhau.

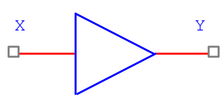
12.2.3 Cổng NOT

Khái Niệm: Là cổng có một ngõ (tín hiệu) vào và chỉ có một ngõ (tín hiệu) ra. Cổng đảo có chức năng như một cổng đệm (đã trình bày trên) nhưng ngõ ra luôn đảo so với ngõ vào.

Phương Trình Ngõ Ra Cổng Đảo

Y = $\overline{X}$

Ký Hiệu và Bảng Trạng Thái



Hình: Ký hiệu cổng đảo

X	Y
0	1
1	0

Đặc điểm:

Trạng thái ngõ ra trái ngược trạng thái ngõ vào

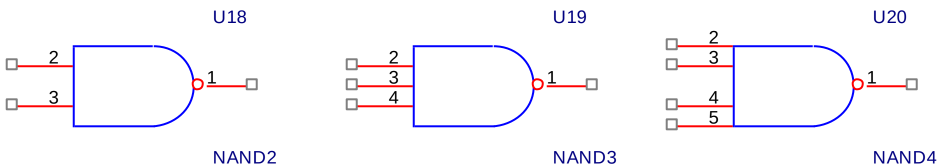
12.2.4 Cổng NAND

Khái Niệm: Là cổng thực hiện phép toán nhân đảo.

Phương Trình Ngõ Ra Cổng NAND:

Y = $\overline{X_1 X_2 ... X_n}$

Ký Hiệu:



Hình: Cổng NAND 2,3,4 lối vào

Đặc Điểm:

Tín hiệu ngõ ra chỉ bằng 0 khi tất cả các tín hiệu ngõ vào đề bằng 1, và tín hiệu ngõ ra sẽ bằng 1 chỉ cần ít nhất một ngõ vào bằng 0.

Bảng Trạng Thái:

Bảng Trạng Thái của NAND 2 và 3 lối vào

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

A	B	C	A NAND B NAND C
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

Cổng NAND thực sự chính là:

NAND = AND + NOT

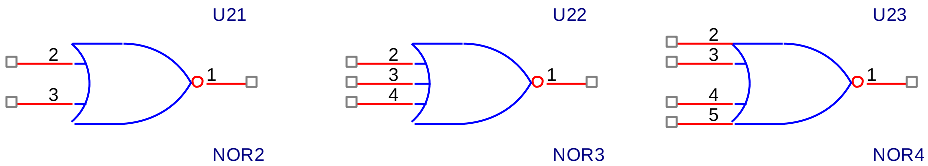
12.2.5 Cổng NOR

Khái Niệm: Là cổng thực hiện chức năng phép toán cộng đảo logic.

Phương Trình Ngõ Ra Cổng NOR:

$Y = \overline{X_1 + X_2 + ... + X_n}$

Ký Hiệu:



Hình: Cổng NOR 2,3,4 lối vào

Đặc Điểm:

Tín hiệu ngõ ra chỉ bằng 1 khi tất cả các ngõ vào bằng 0, tín hiệu ngõ ra sẽ bằng 0 khi có ít nhất một ngõ vào bằng 1.

Bảng Trạng Thái:

Xét bảng trạng thái NOR 2 và 3 lối vào:

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

A	B	C	A NOR B NOR 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

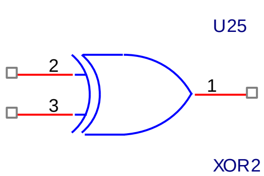
12.2.6 Cổng XOR

Khái Niệm: Là cổng Logic thực hiện chức năng mạch cộng không nhớ.

Phương Trình Ngõ Ra Cổng XOR:

Y = $X_1\overline{X_2} + \overline{X_1}X_2 = X_1 \oplus X_2$

Ký Hiệu:



Hình: Cổng XOR 2 lối vào

Đặc Điểm: Dùng để so sánh 2 tín hiệu. Nếu 2 tín hiệu bằng nhau thì ngõ ra bằng 0 và ngược lại 2 ngõ tín hiệu vào không bằng nhau thì ngõ ra bằng 1.

Bảng Trạng Thái:

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

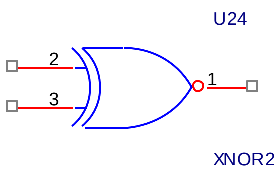
12.2.7 Cổng XNOR:

Khái Niệm: Là cổng thực hiện chức năng cộng đảo không nhớ.

Phương Trình Ngõ Ra:

Y = $\overline{X_1\overline{X_2} + \overline{X_1}X_2} = \overline{X_1 \oplus X_2}$

Ký Hiệu:



Hình: XNOR 2 lối vào

Đặc Điểm:

Trái với đặc điểm của XOR

Bảng Trạng Thái:

X_1	X_2	Y
0	0	1
0	1	0
1	0	0
1	1	1

12.3 Đại số Boole

Đại số Boole còn được gọi là đại số lưỡng trạng thái do tính chất hai trạng thái của nó. Khi phát minh ra môn đại số này, nó không được sử dụng cho một ứng dụng nào. Sau này, khi Shannon đã áp dụng vào việc giải quyết các mạch điện thoại, và từ đó nó được phát triển không ngừng. Ngày nay, đại số boole được áp dụng rộng rãi trong việc thiết kế và phân tích các mạch điện trong máy tính.

Bởi vì các đại lượng chỉ có hai trạng thái nên đại số Boole khác nhiều so với đại số thường và thao tác khá dễ dàng. Đại số Boole không có phân số, số âm, số thập phân, số ảo, số phức, căn số,… Đại số Boole chỉ có 3 phép toán là:

1. Phép cộng thể hiện qua hàm OR
2. Phép nhân thể hiện qua hàm AND
3. Phép phủ định thể hiện qua hàm NOT

Nhắc lại:

OR

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 1$

AND

$0.0 = 0$

$0.1 = 0$

$1.0 = 0$

$1.1 = 1$

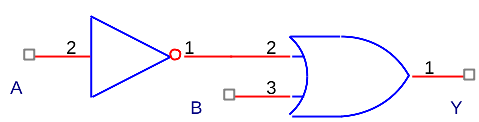
NOT

$\overline{0} = 1$

$\overline{1} = 0$

Ví dụ 1:

Tìm phương trình boole và thiết lập bảng chân trị của mạch sau:



Theo mạch trên ta có phương trình luận lý như sau:

Y = NOT(A) OR B

Và phương trình boole:

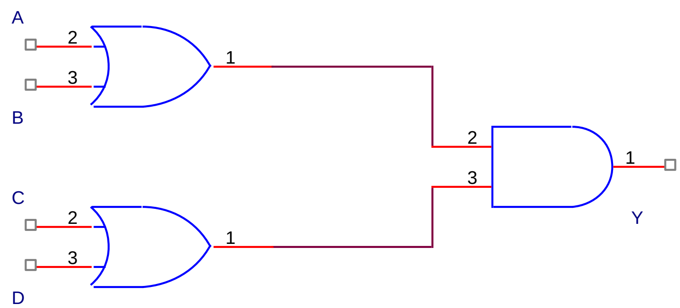
$$Y = \overline{A} + B$$

Bảng trạng thái:

A	B	$\overline{A}$	Y
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Ví dụ 2:

Tìm phương trình boole và thiết lập bảng chân trị cho mạch sau:



Từ mạch trên ta có phương trình luận lý như sau:

Y = (A OR B) AND (C OR D)

Và phương trình boole:

$$Y = (A + B)(C + D)$$

Bảng trạng thái:

A	B	C	D	A OR B	C OR D	Y
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	1	1	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	0	0
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	0	0
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

Các Định lý:

Một biến số:

1. $\overline{\overline{A}} = A$
2. A.0 = 0
3. A.1 = A
4. A.A = A
5. $A.\overline{A} = 0$
6. A + 0 = A
7. A + 1 = 1
8. A + A = A
9. $A + \overline{A} = 1$

Luật giao hoán:

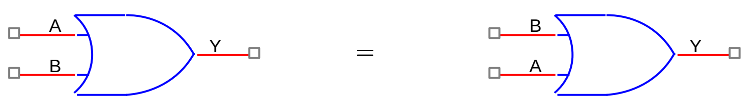
$$A + B = B + A$$
$$A \,.\, B = B \,.\, A \text{ (hay } AB = BA)$$

Chứng minh:

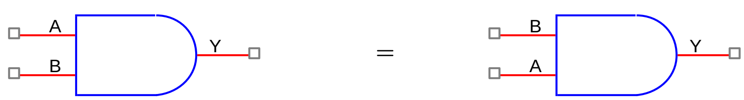
Để chứng minh ta xét cổng AND có 2 ngõ vào. Rõ ràng việc hoán đổi vị trí không ảnh hưởng đến giá trị của ngõ ra. Ta cũng có thể dùng bảng chân trị để chứng minh.

Hình vẽ minh hoạ:

- Trường hợp cổng OR:



- Trường hợp cổng AND:



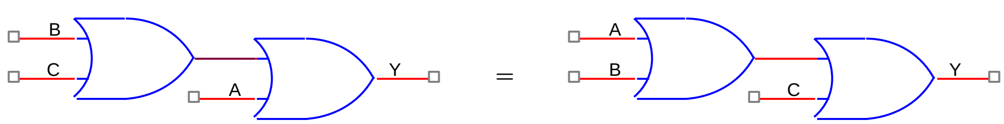
Luật phối hợp:

$$A + (B + C) = (A + B) + C$$
$$A\,(B\,C) = (A\,B)\,C$$

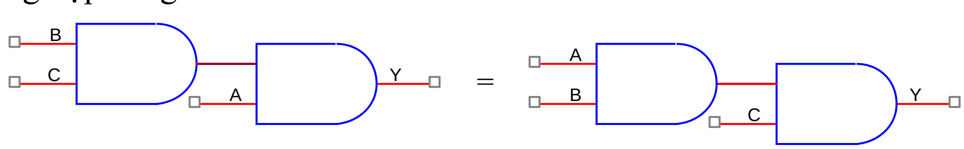
Chứng minh tương tự như trên

Hình vẽ minh hoạ:

- Trường hợp cổng OR:



- Trường hợp cổng AND:



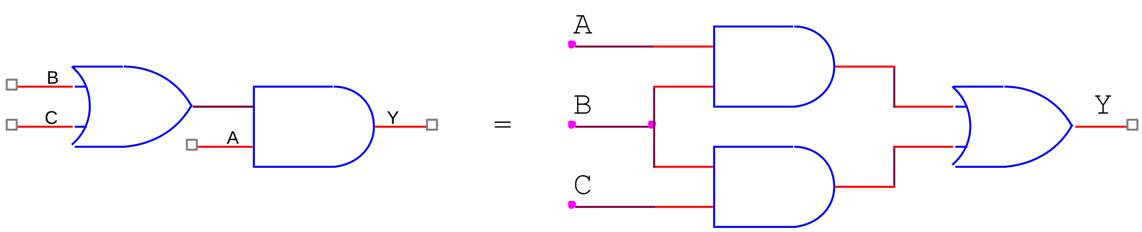
Luật Phân Bố:

$$A (B + C) = AB + AC$$

Chứng minh: Ta dùng bảng chân trị để chứng minh

A	B	C	B + C	A(B+C)	AB+ AC
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Hình vẽ minh hoạ:



Một số đẳng thức thông dụng:

1. $A(A+B) = A$
2. $A + AB = A$
3. $AB + A\overline{B} = A$
4. $A + \overline{A}B = A + B$
5. $A(\overline{A} + B) = AB$
6. $(A + B)(A + \overline{B}) = A$
7. $(A + B)(A + C) = A + BC$
8. $AB + \overline{A}C + BC = AB + \overline{A}C$
9. $(A + B)(\overline{A} + C)(B + C) = (A + B))(\overline{A} + C)$

Định lý De Morgan 1:

Định lý này dựa trên cơ sở đại số boole để xác định phương trình boole.

Định lý này phát biểu như sau:

$$\overline{A+B} = \overline{A} . \overline{B}$$

Chứng minh:

Để chứng minh định lý này ta xét 2 phương trình Boole sau:

$$Y_1 = \overline{A + B} \text{ và } Y_2 = \overline{A} \, . \, \overline{B}$$

Đơn giản chúng ta dùng bảng trạng thái để chứng minh

A	B	Y_1	Y_2
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Định lý De Morgan 2:

Định lý này phát biểu như sau:

$$\overline{AB} = \overline{A} + \overline{B}$$

Việc chứng minh định lý này tương tự như chứng minh định lý 1 (có thể dùng bảng chân trị để chứng minh)

Mở rộng:

Đối với các cổng có 3 hay 4 ngõ vào. Định lý De Morgan được biểu thị như sau:

Mạch có 3 cổng vào A, B, C:

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

Mạch có 4 cổng vào A, B, C, D:

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

Đơn giản hoá phương trình Boole:

Sau khi thiết lập phương trình boole, điều cần thiết chính là đơn giản nó. Vì mạch càng đơn giản thì càng dễ thực hiện và hiệu quả kinh tế cao.

Phương pháp đại số:

Áp dụng các định lý Boole để rút gọn các biểu thức logic.

Ví dụ 1:

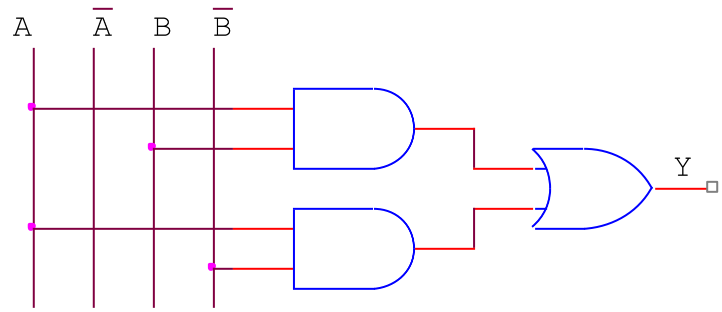
Cho phương trình Boole $Y = AB + A\overline{B}$

Áp dụng quan hệ boole ta có:

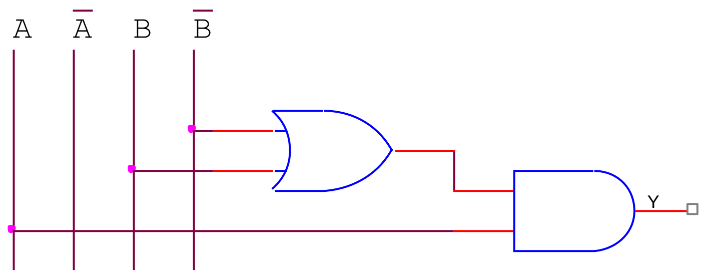
$$Y = A(B + \overline{B}) = A.1 = A$$

Bước đơn giản trên được mô tả bằng hình vẽ như sau:

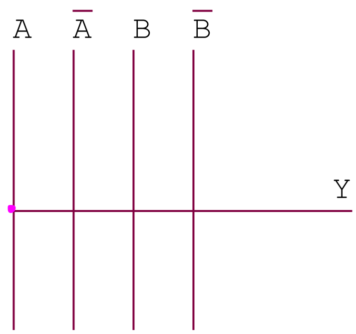
Ban đầu: $Y = AB + A\overline{B}$



Lúc Y = A (B + $\overline{B}$)



Cuối cùng:



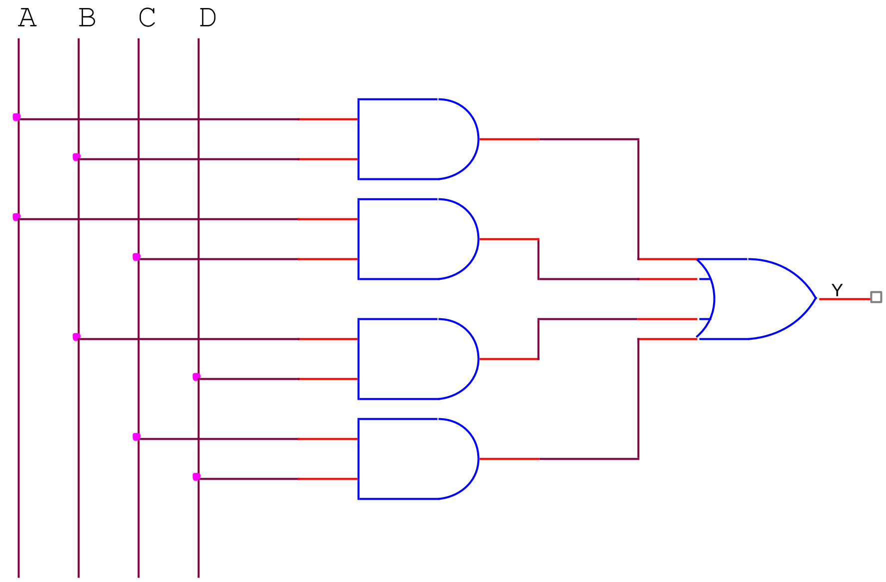
Như vậy: Mạch cuối cùng có phương trình Y = A (có 1 ngõ vào và 1 ngõ ra)

Ví dụ 2:

Cho phương trình như sau Y = AB + AC + DB + DC

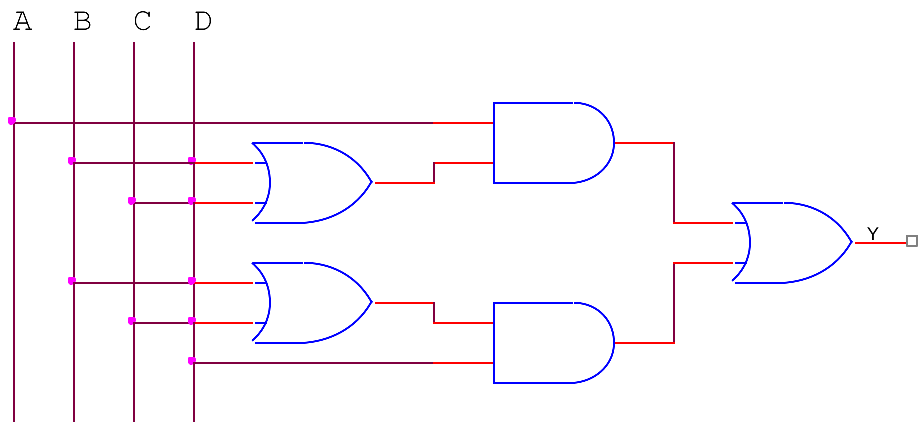
Tìm phương trình đơn giản nhất.

Hình vẽ cho mạch này như sau:



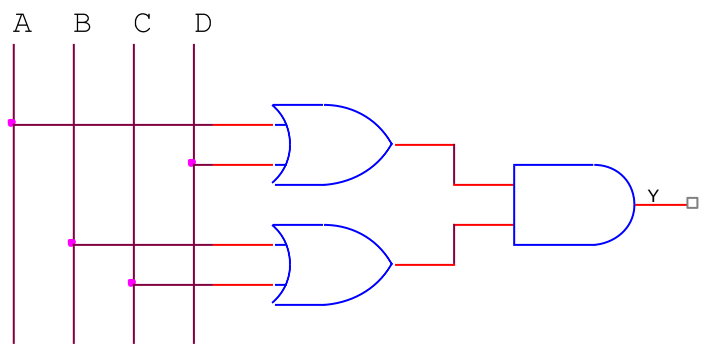
Đơn giản phương trình trên:

$$Y = A(B + C) + D(B + C)$$



Bước đơn giản cuối cùng:

$$Y = (A + D)(B + C)$$



Ví dụ 3: Cho biểu thức sau:

Y = $\overline{A}C(\overline{\overline{A}BD}) + \overline{A}B\overline{C}\overline{D} + A\overline{B}C$

Áp dụng định lý De Morgan 2 ta được:

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

$= \overline{A}CA + \overline{A}C\overline{B} + \overline{A}C\overline{D}) + \overline{A}B\overline{C}\overline{D} + A\overline{B}C$

$= 0 + \overline{A}\overline{B}C + \overline{A}C\overline{D} + \overline{A}B\overline{C}\overline{D} + A\overline{B}C$

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

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

Như vậy khi sử dụng các định lý đại số Boole ta đã có thể rút gọn từ một biểu thức phức tạp sang biểu thức đơn giản và như thế sẽ tạo điều kiện thuận lợi trong quá trình thi công mạch điện.

Bài tập

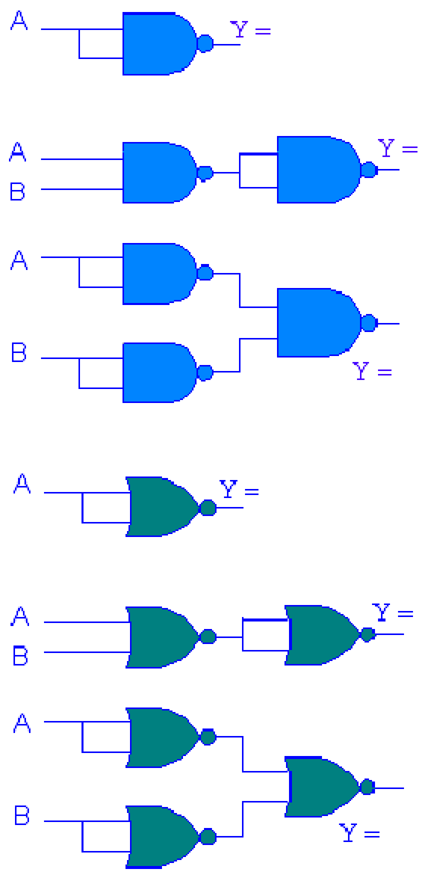
Câu 1. Thiết kế mạch dùng hai cổng logic thỏa bảng sự thật sau đây:

Vào		Ra
A	B	Y
0	0	1
0	1	0
1	0	1
1	1	1

Câu 2: Chứng tỏ

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

Câu 3: Cho các mạch logic sau:



a)Hãy viết biểu thức ở ngõ ra-Y=?

b)Từ các sơ đồ mạch trên chuyển sang 1 cổng logic tương ứng.

Câu 4: Áp dụng các định lý đại số Boole để rút gọn biểu thức logic sau:

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

(b) $ABC + ABD + AB$

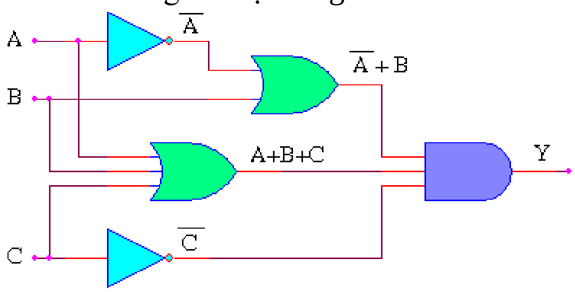
(c) $AB(\overline{A} + C)$

(d) $\overline{\overline{A} + \overline{BC}}\,.\overline{A}$

Câu 5: Đơn giản biểu thức logic sau:

$$Y = ABC + AB\overline{C} + A\overline{B}C$$

Câu 6: Đơn giản mạch logic ở hình sau:



Câu 7: Hãy sử dụng các toán tử NOR để thực hiện các toán tử logic NOT, AND, OR, NAND, NOR, XOR, và XNOR.