

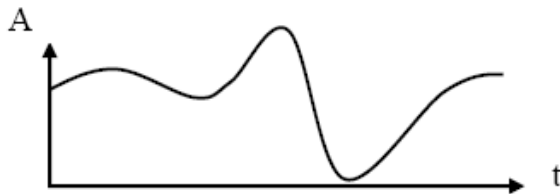
CHƯƠNG 1: CƠ SỞ KỸ THUẬT SỐ

1. Tổng quan về mạch tương tự và số

1.1 Định nghĩa

a) Mạch tương tự

- Mạch tương tự là mạch xử lý các tín hiệu dạng tương tự, xử lý ở đây là khuếch đại, phát sinh và biến đổi. Tín hiệu tương tự được biểu diễn theo thời gian như sau (time chart).



- Tín hiệu Analog thường do các hiện tượng tương tự nhiên phát sinh ra và nó có đặc điểm:

- + Liên tục về biên độ (có bất cứ trị số nào trong quá trình biến thiên của nó).
- + Liên tục về thời gian trong suốt thời gian có tín hiệu.

b) Mạch số

- Mạch số còn gọi là mạch logic “logic circuit” xử lý các tín hiệu số, tín hiệu số nhị phân là tín hiệu mà biên độ chỉ có 2 mức là mức cao (H) và mức thấp (L), còn gọi là mức (1) và mức (0)

- Tín hiệu số được biểu diễn bởi các mức logic như sau:



- Thông thường : Mức H $\approx$ 5 vol

Mức L $<$ 0.4 vol

- Mức cao biến thiên trong khoảng từ : 2,4 – 5 (vol)

Mức logic cao (H = mức 1)

Mức logic thấp (L = mức 0)

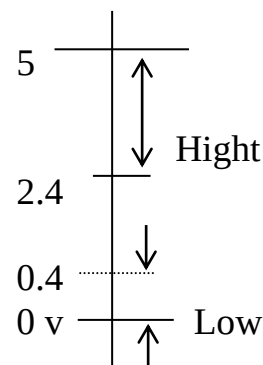
- **Mức logic âm và dương (- & +)**

- Nếu mức logic 1 chọn $>$ mức 0 ta có mức logic (+)

- Nếu mức logic 1 chọn $<$ mức 0 ta có mức logic (-)

- **Đặc điểm của các mạch logic**

- Chỉ phát sinh bởi các mạch thích hợp.



- Gián đoạn về biên độ, sự chuyển tiếp giữa 2 mức xảy ra nhanh chóng (không đáng kể về thời gian).
- Gián đoạn về thời gian

c) Ưu nhược điểm của kỹ thuật số so với kỹ thuật tương tự

*** Ưu điểm:**

- *Nhìn chung thiết bị số dễ thiết kế hơn.* Đó là do mạch được sử dụng các vi mạch chuyên dùng được thiết kế với các chức năng định trước.
- Việc lưu trữ và truy xuất thông tin nhanh chóng và dễ dàng thông qua việc sử dụng các vi mạch nhớ, thẻ nhớ vv...
- *Chính xác và độ tin cậy cao hơn.* Do đặc thù của hệ thống là xử lý các tín hiệu 1 và 0 và thời gian chuyển tiếp giữa chúng rất nhanh nên khả năng chống nhiễu rất cao và lại biên độ của tín hiệu nhiễu không có đủ khả năng thay đổi mức tín hiệu từ 1 sang 0 và ngược lại. Trong khi đó ở thiết bị tương tự độ chính xác thường bị giới hạn vì mạch phải xử lý các tín hiệu liên tục theo thời gian, hơn nữa các linh kiện sử dụng có đặc tính không thật sự tuyến tính. Các tín hiệu không mong muốn do môi trường hay phát sinh trong quá trình xử lý làm suy giảm hoặc ảnh hưởng trầm trọng đến chất lượng và tính ổn định của hệ thống.
- *Có thể lập trình hoạt động của hệ thống kỹ thuật số.* Hoạt động của hệ thống kỹ thuật số có được điều khiển theo các qui luật định trước bằng một tập hợp lệnh gọi là chương trình. Với việc ra đời của các vi xử lý và vi điều khiển việc điều khiển tự động hệ thống trở nên dễ dàng hơn. Hệ thống tương tự cũng có thể lập trình, nhưng rất đơn giản, không đa dạng.
- *Mật độ tích hợp cao.* Kỹ thuật chế tạo linh kiện ngày càng phát triển cho phép tạo ra được nhiều khối chức năng trên một đơn vị diện tích hẹp dẫn đến việc giảm đáng kể kích thước và giá thành của các thiết bị điện tử, thêm vào đó năng lượng tiêu hao trong quá trình vận hành cũng thấp hơn. Thật ra sự phát triển vượt bậc của công nghệ IC cũng mang lại nhiều thuận lợi cho mạch tương tự. Nhưng do mạch tương tự khá phức tạp và lại sử dụng những thiết bị không có hiệu quả kinh tế nên không thể đạt được hiệu quả tích hợp cao như mạch số.

*** Nhược điểm:**

Hầu hết đại lượng vật lý đều có bản chất tương tự và chính những đại lượng này thường là đầu vào và đầu ra của một hệ thống theo dõi điều khiển. Ví dụ như các đại lượng nhiệt độ, áp suất, vị trí, vận tốc, độ ẩm, thể tích, âm thanh, ánh sáng... Phân tích giá trị của các đại lượng này theo cội thời gian đó chính là các đại lượng tương tự. Các đại lượng này tồn tại trong thực tế cuộc sống quanh ta. Nói cách khác thế giới quanh ta là "tương tự".

Trong kỹ thuật người ta thường phải biến đổi thông tin qua lại giữa dạng tương tự và dạng số có thể xem là khuyết điểm vì nó làm cho thiết bị thêm phức tạp và giá thành cao hơn. Một nhân tố quan trọng khác là cần thêm thời gian để thực hiện các biến đổi này. Thực tế những bất lợi như trên thường bị lấn lướt bởi nhiều ưu điểm khác của kỹ thuật số và do vậy việc biến đổi qua lại giữa các đại lượng tương tự và số là công việc cần thiết và thông dụng trong công nghệ hiện nay.

Tuy nhiên cũng có một số trường hợp chỉ nên sử dụng kỹ thuật tương tự, vì đơn giản hơn có hiệu quả kinh tế hơn.

Để tận dụng ưu thế của cả hai loại, người ta thường sử dụng cả kỹ thuật số và kỹ thuật tương tự trong cùng hệ thống. Ở những hệ thống này khâu quan trọng nhất trong giai đoạn thiết kế là quyết định cho phần nào của hệ thống kỹ thuật tương tự và phần nào dùng kỹ thuật số.

2. Hệ thống số và mã số

2.1 Hệ thống số thập phân (Decimal system_D)

- Hệ thống số thập phân dựa trên cơ sở 10 đầu ngón tay, dùng các con số (digital) từ 0 ÷ 9 để biểu diễn 1 số nào đó, khi số biểu diễn có giá trị lớn hơn 9 thì người ta quy ước hàng chục, hàng trăm, hàng đơn vị theo 1 qui luật như sau:

$$X = a_{n-1} \cdot 10^{n-1} + \dots + a_0 \cdot 10^0$$

$$\begin{aligned} \text{Ví dụ : } 270 &= 2 \times 10^2 + 7 \times 10^1 + 0 \times 10^0 \\ &= 200 + 70 + 0 = 270 \\ 2151 &= 2 \times 10^3 + 1 \times 10^2 + 5 \times 10^1 + 1 \times 10^0 \\ &= 2000 + 100 + 50 + 1 = 2151 \end{aligned}$$

2.2 Hệ thống số nhị phân (Binary system_B)

- Hệ nhị phân hay hệ đếm cơ số 2 chỉ dùng 2 con số 0 và 1 để biểu diễn 1 giá trị nào đó.

- Mỗi một số nhị phân như vậy được gọi là 1 bit. "Bit" là đơn vị thông tin nhỏ nhất nó có thể lấy giá trị 0 hay 1.

- Hệ nhị phân được biểu diễn theo quy luật sau:

$$X = a_{n-1} \cdot 2^{n-1} + \dots + a_0 \cdot 2^0$$

$$\begin{aligned} \text{Ví dụ: } 1001_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 9_{10} \text{ hay } (9_D) \\ 11011_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 27_{10} \end{aligned}$$

- Nếu 1 số nhị phân biểu diễn 1 giá trị nào đó có 8 bit thì 8 bit được gọi là 1 byte. 16 bit = 1 từ (2 byte), 1024 bit (2^{10}) = 1Kbit, 1048576 bit (2^{20}) = 1Mbit.

2.3 Hệ thống bát phân (Octo system_O)

- Hệ thống này người ta dùng các con số từ 0 tới 7 và cơ số 8 để biểu diễn một hệ thống số như bảng sau :

MÃ BÁT PHÂN	MÃ THẬP PHÂN	MÃ BCD
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
10	8	1000
11	9	1001
12	10	1010
13	11	1011
14	12	1100
15	13	1101
16	14	1110
17	15	1111

- Mã bát phân biểu diễn các con số theo qui luật sau đây:

$$X = a_{n-1}.8^{n-1} + \dots\dots\dots a_0.8^0$$

Ví dụ: $57_8 = 5 \times 8^1 + 7 \times 8^0 = 40 + 7 = 47_{10}$

2.4 Hệ thống thập lục phân (Hecxa system_D)

- Mã này người ta dùng các con số từ 0 tới 9 và các chữ cái từ A tới F và cơ số 16 để biểu diễn một hệ thống số như bảng sau:

MÃ THẬP LỤC	MÃ THẬP PHÂN	MÃ BCD
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000

9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

- Mã thập lục phân biểu diễn các con số theo qui luật sau đây:

$$X = a_{n-1}.16^{n-1} + \dots + a_0.16^0$$

Ví dụ: $5B_{16} = 5 \times 16^1 + B \times 16^0 = 80 + 11 = 91_{10}$

$$2AF_{16} = 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = 256 + 160 + 15 = 687_{10}$$

2.5 Mã BCD (còn gọi là mã 8421)

- Hệ thống số nhị phân dùng các phương pháp tập hợp các bit 0 và 1 theo một qui luật nào đó thì hình thành cho ta các loại mã, để tiện lợi sử dụng người ta thường dùng các loại mã sau: BCD, ASCII “American national standard information interchanger”, mã thừa 3, mã Gray....

- Mã BCD này dùng 4 bit để biểu diễn 1 số nào đó.

Ví dụ : $9_{10} = 1001_{BCD}$; $13_{10} = 1101_{BCD}$; $15_{10} = 1111_{BCD}$; $0101_{BCD} = 5_{10}$; $0011_{BCD} = 3_{10}$

- Như vậy mỗi nhóm bit chỉ biểu diễn được 1 số tương ứng lớn nhất là 15 của hệ 10.

- Nếu số thập phân lớn hơn 15 người ta dùng nhiều nhóm mã BCD để biểu diễn.

2.6 Mã ASCII

Ngoài dữ liệu dạng số, máy tính còn phải có khả năng thao tác thông tin khác số. Nói cách khác máy tính phải nhận ra được mã biểu thị mẫu tự abc, dấu chấm câu, những ký tự đặc biệt, cũng như ký tự số.

Mã chữ số được sử dụng rộng rãi nhất hiện nay là mã ASCII (American Standard Code for Information Interchange). Mã ASCII (đọc là ”aski”) là mã 7 bit, nên có $2^7 = 128$ nhóm mã, quá đủ biểu thị các ký tự của một bàn phím chuẩn cũng như các chức năng điều khiển.

Ký tự	Mã ASCII 7bit	Octal	Hexa
A	100 0001	101	41
B	100 0010	102	42
C	100 0011	103	43
D	100 0100	104	44
E	100 0101	105	45
F	100 0110	106	46
G	100 0111	107	47
H	100 1000	110	48
I	100 1001	111	49
J	100 1010	112	4A
K	100 1011	113	4B
L	100 1100	114	4C
M	100 1101	115	4D
N	100 1110	116	4E
O	100 1111	117	4F
P	101 0000	120	50
Q	101 0001	121	51
R	101 0010	122	52
S	101 0011	123	53
T	101 0100	124	54
U	101 0101	125	55
V	101 0110	126	56
W	101 0111	127	57
X	101 1000	130	58
Y	101 1001	131	59
Z	101 1010	132	5A

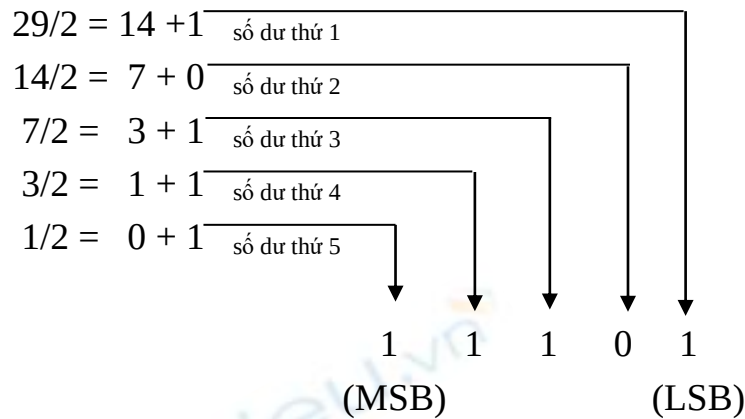
0	011 0000	060	30
1	011 0001	061	31
2	011 0010	062	32
3	011 0011	063	33
4	011 0100	064	34
5	011 0101	065	35
6	011 0110	066	36
7	011 0111	067	37
8	011 1000	070	38
9	011 1001	071	39
<Ký tự trắng>	010 0000	040	20
.	010 1110	056	2E
(	010 1000	050	28
+	010 1011	053	2B
\$	010 0100	044	24
*	010 1010	052	2A
)	010 1001	051	29
_	010 1101	055	2D
/	010 1111	057	2F
,	010 1100	054	2C
=	011 1101	075	3D
<RETURN>	000 1101	015	0D
<LINEFEED>	000 1010	012	0A

2.7 Chuyển đổi giữa các hệ thống số

* Đổi hệ D sang hệ B

- Để đổi 1 số từ hệ thập phân sang hệ nhị phân ta tiến hành chia cặp số thập phân cho 2, các số dư 0 và 1 là các con số của hệ nhị phân.

Ví dụ 1: Đổi 29_{10} sang hệ nhị phân.



MSB : Most Significant Bit (Bit số nghĩa lớn nhất)

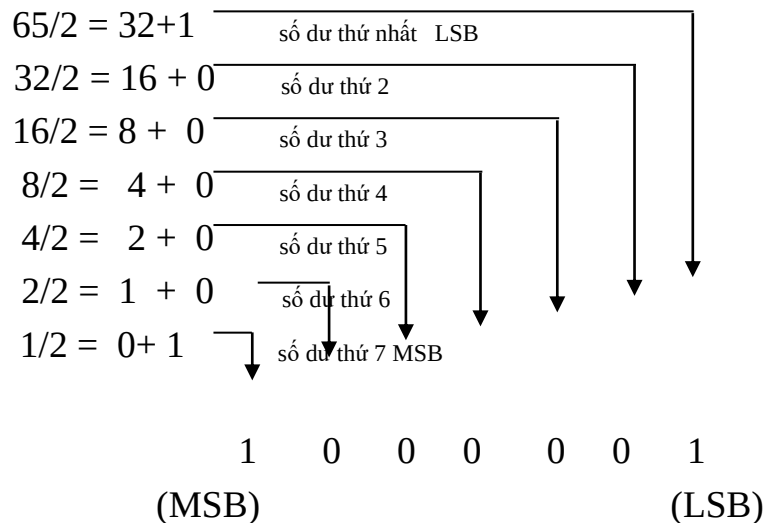
LSB : Least significant Bit (Bit có nghĩa nhỏ nhất)

Ta được số : $11101_{(2)} = 29_{(10)}$

Với số dư thứ nhất gọi là bit có nghĩa nhỏ nhất LSB

Với số dư cuối cùng ta gọi là bit có nghĩa lớn nhất MSB.

Ví dụ 2: Đổi 65_{10} sang hệ nhị phân.



Vậy $65_{10} = 100\ 0001_2$

* Đổi hệ B sang hệ D

- Để đổi 1 số từ hệ nhị phân sang hệ thập phân ta chỉ việc tính tổng của tổng đại số các bit trong số nhị phân sau khi nhân với hệ số tương ứng.

Ví dụ 1: Đổi $100110_{(2)}$ sang thập phân.

$$= 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 32 + 0 + 0 + 4 + 2 + 0 = 38_{(10)}$$

Ví dụ 2: Đổi 11011 sang thập phân

$$= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$= 16 + 8 + 0 + 2 + 1 = 27_{(10)} \text{ Vậy } 11011_2 = 27_{10}$$

* Đổi hệ D sang hệ H

- Ta lấy số thập phân chia lặp lần lượt cho 16, số dư của các phép chia là các chữ số của số ở hệ thập lục phân.

Ví dụ 1: Đổi 125 sang hệ 16

$$\begin{array}{l} 125/16 = 7 + 13 \\ 7/16 = 0 + 7 \end{array}$$

$$\text{Vậy } 125_{10} = 7D_{16}$$

* Đổi hệ H sang hệ D

- Để đổi từ hệ thập lục phân sang hệ thập phân, ta chỉ việc tìm tổng đại số của tổng các số hạng trong hệ 16.

$$\text{Ví dụ 1: } 2C5_{16} = 2 \times 16^2 + 12 \times 16^1 + 5 \times 16^0$$

$$= 512 + 192 + 5 = 709_{10}$$

3. Các cổng logic cơ bản

3.1 Cổng AND (AND_gate)_Cổng và

* Định nghĩa: Cổng và thực hiện phép tính tích logic với hàm số: $Y = A.B....$

* Ký hiệu :



* Bảng trạng thái (Bảng chân lý)

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

A	B	C	Y
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

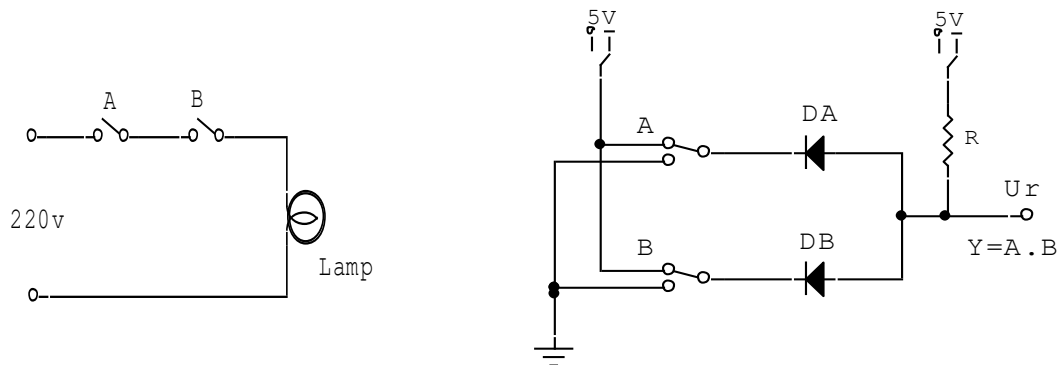
* ý nghĩa vật lý : Mạch điện có dạng sau đây diễn tả hàm AND

+ Rõ ràng đèn L chỉ sáng khi:

$$(L = Y = 1) \text{ khi và chỉ khi: } A = B = 1$$

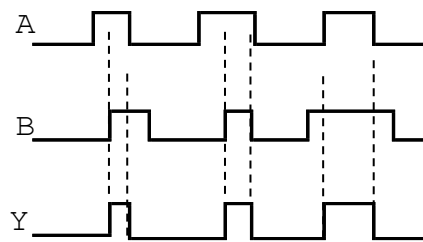
* Mạch điện tử diễn tả sự hoạt động của hàm AND

- + Các đầu vào A,B có : mức 1 = 5v ; mức 0 = 0v
- + Đầu ra Y có: mức 1 = 5 v ; mức 0 = 0,7v



Rõ ràng chỉ khi nào $A = B = 1$ cả 2 diode đều dẫn tiếp, đầu ra mới lên cao tức $Y = 1 = 5 \text{ vol}$

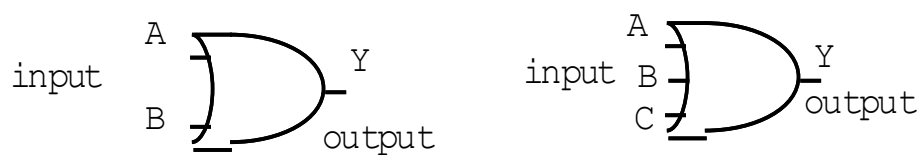
* Dạng sóng của các ngõ vào và ngõ ra



3.2 Cổng OR (OR_gate)_ Cổng hoặc

* Định nghĩa: Cổng OR thực hiện phép tính tổng logic theo hàm: $Y = A + B + \dots$

* Ký hiệu:

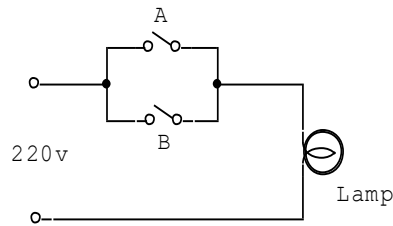


* Bảng trạng thái:

input		out put
A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

* ý nghĩa vật lý: Ta có 1 mạch điện thể hiện hàm OR như sau:

- Đèn L sẽ sáng khi 1 trong khoá A hoặc khoá B đóng hoặc là $A = B = 1$ thì L= 1



- Một mạch điện tử như sau thể hiện sự hoạt động của hàm OR.

+ Các đầu vào A,B,C có :

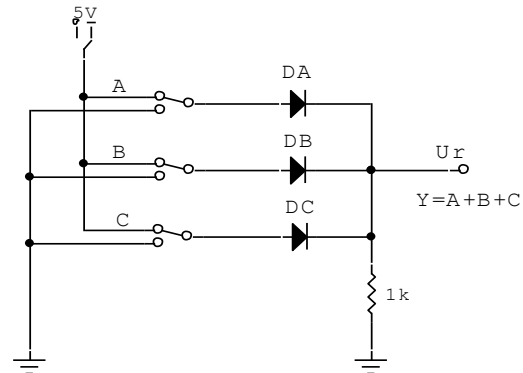
mức (1) = 5v

mức (0) = 0v

+ Đầu ra có :

mức (1) = 4,3v

mức (0) = 0v

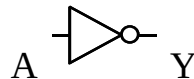


- Rõ ràng ta thấy chỉ cần 1 trong các đầu vào có mức (H) là đầu ra có mức H, hoặc tất cả các đầu vào ở mức cao thì đầu ra ở mức cao ($Y = 1 = H$).

3.3 Cổng NOT (NOT_gate)_Cổng đảo

* Định nghĩa: Cổng đảo thực hiện phép tính đảo logic hoặc phép phủ định logic .

* Ký hiệu cổng NOT :



* Hàm số logic: $Y = \bar{A}$

* Bảng trạng thái:

A	Y
0	1
1	0

* ý nghĩa vật lý:

- Hàm NOT được diễn tả bằng mạch điện sau:

Rõ ràng ta thấy nếu:

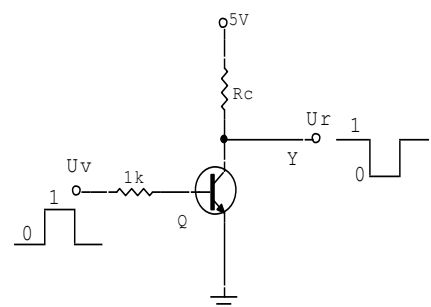
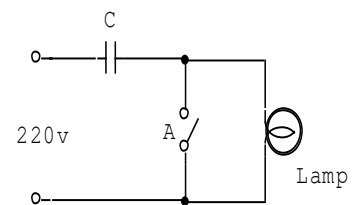
+ A đóng ($A = 1$) thì đèn tắt $Y = 0$

+ A mở ($A = 0$) thì đèn sáng $Y = 1$

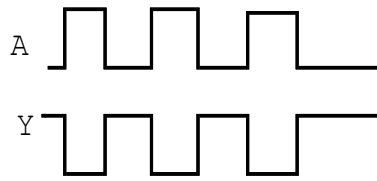
- Mạch điện tử sau đây diễn tả hàm Not:

Nếu $A = 1$ thì $Y = 0 = 0,2v$

Nếu $A = 0$ thì $Y = 1 = +5v$

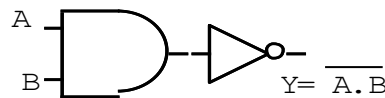


* Dạng sóng ở đầu vào và đầu ra của cổng NOT

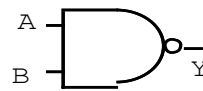


3.4 Cổng NAND (NAND_gate)_Cổng và đảo

* Cổng và đảo được xây dựng trên cơ sở kết hợp giữa cổng AND và 1 cổng NOT theo sau. Nó thực hiện phép tính phủ định tích.



* Hàm số đặc trưng: $Y = \overline{A.B}$



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

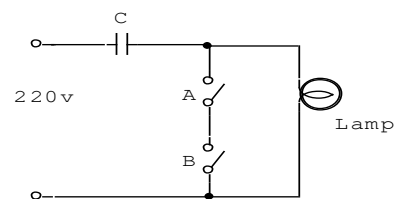
* Ký hiệu của cổng NAND

* Bảng chân lý - bảng sự thật

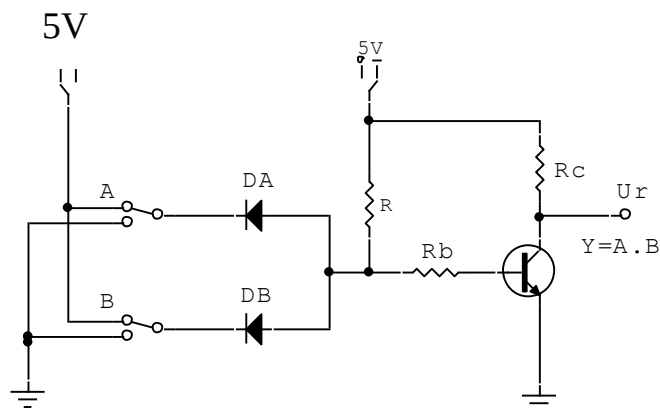
Đầu ra ở mức thấp khi và chỉ khi tất cả các đầu vào ở mức cao.

* Mạch điện biểu diễn hàm NAND

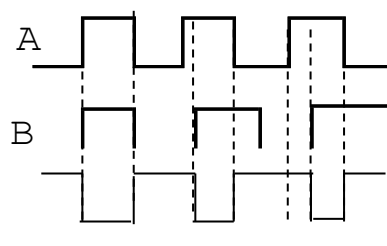
Rõ ràng chỉ khi nào $A=B=1$ thì $Y = 0$.



* Mạch điện tử biểu diễn hàm NAND.



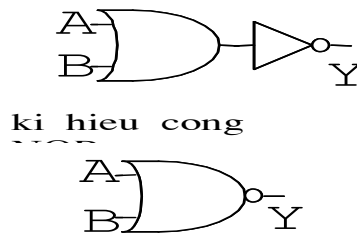
* Dạng sóng ở đầu vào và đầu:



3.5 Cổng NOR (NOR_gate)_Cổng hoặc đảo

* Cổng NOR được xây dựng trên cơ sở kết hợp giữa cổng OR và cổng NOT theo sau. Cổng NOR thực hiện phủ định của 1 tổng logic. $Y = \overline{A+B+...}$

* Kí hiệu :



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

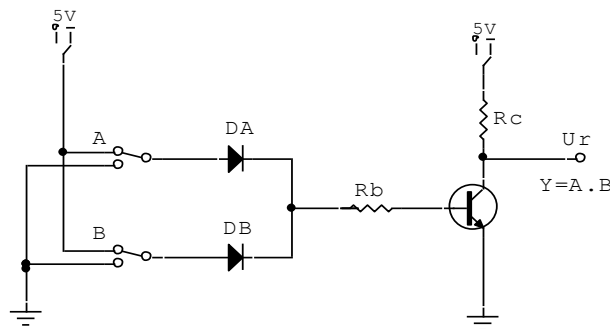
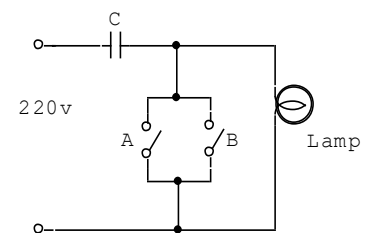
* Bảng chân lý (truth table)

Rõ ràng là chỉ và chỉ khi $A = B = 0$ thì $Y = 1$

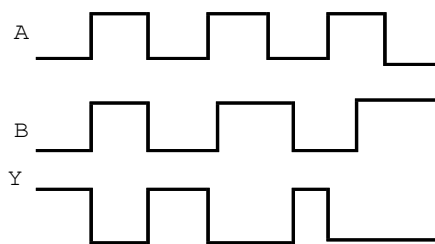
* Mạch điện diễn tả hàm số NOR.

Rõ ràng chỉ khi nào $A = B = 0$ thì đèn L mới sáng

* Mạch điện tử diễn tả hàm NOR:



* Dạng sóng ở đầu vào và đầu ra:



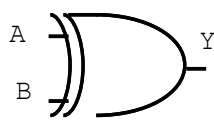
3.6 Cổng EX-OR (EX-OR_gate)_Cổng hoặc loại trừ (Exclusive-or)

* Cổng hoặc loại trừ được xây dựng trên cơ sở đầu ra bằng tổng khác dấu của các ngõ vào. Tức là: $Y = \overline{A}B + A\overline{B} = A \oplus B$

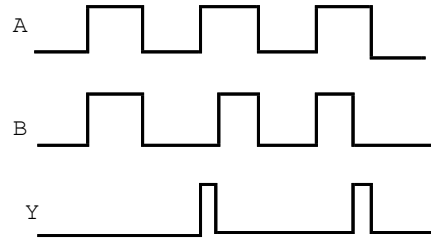
* Cổng hoặc loại trừ hoạt động theo bảng sự thật sau:

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

* Kí hiệu:



* Dạng sóng ngõ vào và ngõ ra:



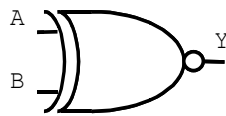
3.7 Cổng EX-NOR (EX-NOR_gate)_Cổng hoặc đảo loại trừ

* Cổng EX-NOR được xây dựng trên cơ sở đầu ra bằng phủ định tổng khác dấu của các tín hiệu ở đầu vào, tức là: $Y = \overline{A \oplus B}$

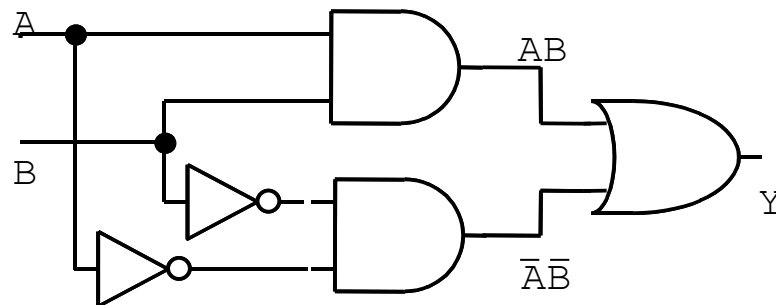
* Cổng EX-NOR hoạt động theo bảng chân lý sau:

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

* Ký hiệu:



* Để chứng minh cho bảng chân lý ta dùng mạch điện sau :

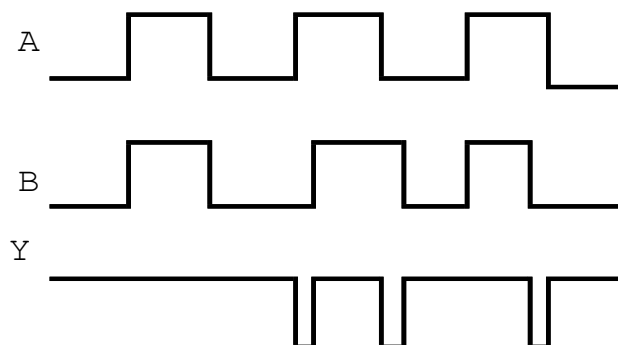


$$Y = \overline{A} \overline{B} + AB$$

Rõ ràng hàm số $Y = \overline{A} \overline{B} + AB$ nghiệm đúng bảng sự thật vì vậy ta có:

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

* Dạng sóng ở đầu vào và đầu ra:

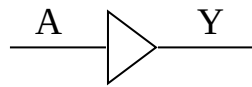


3.8 Cổng Buffer (Buffer_gate)_Cổng đệm

* Cổng đệm có tác dụng cho qua tín hiệu mà không hề làm thay đổi dạng sóng của tín hiệu truyền qua nó. Tức là nó thực hiện thuật toán logic : $Y = A$

- Cổng đệm được dùng trong những trường hợp khi ta cần một dòng điện cung cấp cho tải tương đối lớn, trị số của nó vượt qua khả năng tải dòng của IC logic thì ta cần phải lắp thêm một cổng đệm nằm trung gian giữa IC và tải.

* Ký hiệu

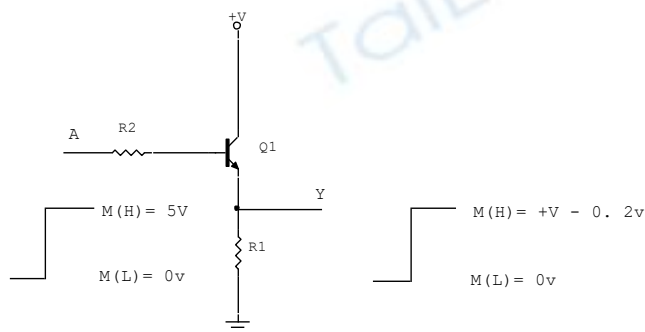


* Bảng trạng thái

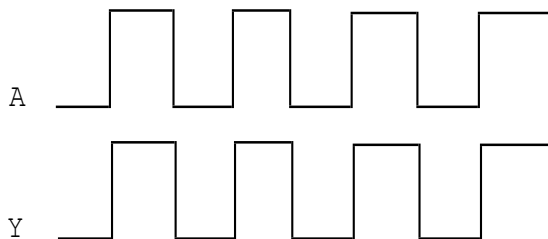
A	Y
0	0
1	1

* Ý nghĩa vật lý

Ta dùng một mạch TZT đơn giản để biểu diễn sự hoạt động của cổng đệm như sau:



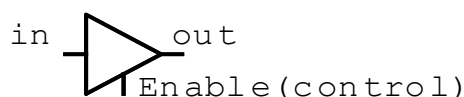
* Dạng sóng của cổng đệm



Dạng sóng đầu vào “A” và ngõ ra “Y” luôn luôn đồng pha với nhau

3.9. Transistor 3 trạng thái (Three state transistor)

* Transistor 3 trạng thái “three state hay còn gọi là Tri-state” thường gọi là đệm 3 trạng thái “Tri state bus buffer” thường chế tạo ở dạng IC – gọi là logic 3 trạng thái nó có ký hiệu như sau:

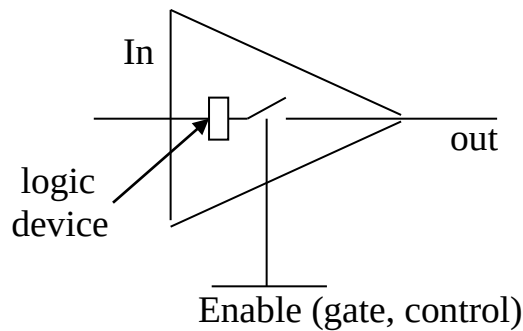


* 3 Trạng thái của Three state là:

1. Trạng thái logic 0
2. Trạng thái logic 1
3. Trạng thái trở kháng cao

Vì vậy nó có thể truyền từ đầu vào ra đầu ra các mức logic 0 và 1 hoặc là nó cắt mạch “coi như không có mặt trong mạch điện” khi ở trở kháng cao.

* Sơ đồ khối có dạng như sau:

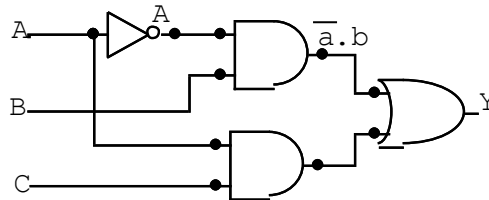


+ Cực điều khiển gọi là cực cho phép (Enable), cực cho phép có thể tác động ở mức cao hay mức thấp là do từng loại three state.

+ Three state được ứng dụng rộng rãi trong mạch số, nó làm chức năng nối mạch ở 2 trạng thái cao và thấp, khi không làm việc nó được xem như hở mạch.

4. Biểu thức logic và mạch điện

4.1. Mạch điện biểu diễn biểu thức logic

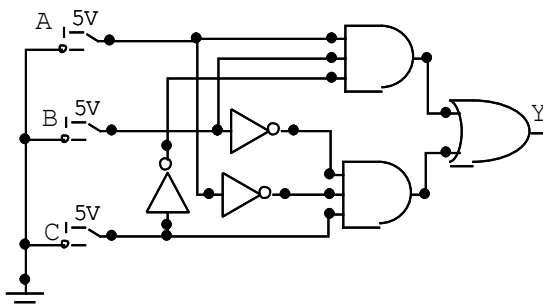


Từ mạch điện ta có hàm số: $Y = \bar{A} \cdot B + A \cdot C$

4.2 Xây dựng biểu thức logic theo mạch điện cho trước

Cho hàm số sau xây dựng mạch điện của hàm: $Y = A \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C$

Ta có mạch điện:



5. Đại số boole và định lý Demorgan

Đại số boolean là 1 môn toán học dùng để giải quyết các phép toán về hệ thống số nhị phân.

5.1. Hàm Boole một biến

$$\begin{aligned} \overline{\overline{A}} &= A & A \cdot \overline{A} &= 0 & A \cdot 1 &= A & \overline{A} + A &= 1 & A \cdot A &= A \\ A \cdot 0 &= 0 & A + 0 &= A & A + A &= A & 1 + A &= 1 \end{aligned}$$

Để chứng minh các biểu thức trên ta có thể dùng phương pháp lập bảng như sau :

A	0	A+0
1	0	1
0	0	0

A	A	A+A
0	0	0
1	1	1

5.2. Hàm nhiều biến

Các luật cơ bản (base laws)

a. Luật giao hoán (Commutative law)

$$A \cdot B = B \cdot A \qquad A + B = B + A$$

b. Luật phối hợp (associative laws)

$$A \cdot B \cdot C = A(B \cdot C)$$

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

c. Luật phân bố (Distributive law)

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

$$(A + B)(A + C) = A + BC$$

d. Một số đẳng thức tiện dụng

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

5.3. Định lý Demorgan (Demorgan law)

- $\overline{A \cdot B \cdot C} = \overline{A} + \overline{B} + \overline{C} + \dots$ Phủ định của 1 tích bằng tổng các phủ định.

- $\overline{A+B+C} = \overline{A} \cdot \overline{B} \cdot \overline{C} \dots$ Phủ định của 1 tổng bằng tích các phủ định

Áp dụng:

Xuất phát từ 1 bài toán logic về yêu cầu điều khiển trong công nghiệp, ta thiết lập các hệ thức logic liên quan đến các ngõ vào và ngõ ra gọi là các hệ thức boole, kể đến nhờ các định lý ở trên (và vài phương pháp khác) ta rút gọn tối đa các hệ thức boole, rồi dựa vào hệ thức đã rút gọn ta thiết lập mạch điện.

6. Đơn giản biểu thức logic

Mỗi một mạch logic được xây dựng trên cơ sở 1 hệ thức boole. Hệ thức boole được rút ra từ bảng chân lý thông thường là chưa tối giản, vì vậy mạch logic được xây dựng từ hệ thức logic này thường là rất phức tạp, rườm rà. Bởi vậy người ta nghĩ cách cực tiểu hoá hệ thức boole sao cho kết quả thu được vẫn tương đương với hệ thức cũ, nhưng mạch điện thì đơn giản đi rất nhiều.

- Có nhiều phương pháp cực tiểu hệ thức boole sau đây ta sẽ nghiên cứu :

6.1. Đơn giản biểu thức logic bằng phương pháp đại số

Đây là phương pháp dựa vào các định lý và đẳng thức trong đại số boole để ta tối thiểu hoá các hệ thức boole.

Ví dụ 1: $Y = \overline{A}.B.C + \overline{A}.B. \overline{C}$

Giải: ta nhóm các số hạng đồng dạng : $Y = \overline{A}.B (\overline{C} + C) = \overline{A}.B$

Ví dụ 2: $Y = A.B.C + A.B. \overline{C} + A. \overline{B}.C$

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

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

áp dụng đẳng thức (9) ta có $(B + \overline{B}.C) = B + C$

6.2. Rút gọn biểu thức logic bằng biểu đồ Karnaugh

Bản đồ Karnaugh là cách trình bày của hàm logic ở dạng bản đồ. Nó giúp việc đơn giản hoá các biểu thức logic, dựa vào bảng chân lý của hệ thức boole ta có thể thành lập được bản đồ Karnaugh, rồi ta đơn giản bản đồ mà không phải thông qua biểu thức logic.

a. Cách thành lập bản đồ Karnaugh.

- Bảng Karnaugh là 1 hình chữ nhật, nếu hệ thức chứa N biến số thì bản đồ sẽ có 2^n ô: **Ví dụ:** Hàm có 2 biến $n = 2$ do đó có $2^2 = 4$ ô.

- Giá trị của các biến được xếp theo mã vòng.

- Giá trị của từng tổ hợp biến được ghi vào từng ô.

- Ta có bảng chân lý (hình vẽ)

Căn cứ vào bảng chân lý ta có

1 bản đồ gồm $2^2 = 4$ ô.

Biến số		Hàm số
A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

	$\bar{A}$	A
$\bar{B}$	$\bar{A}\bar{B}$	$A\bar{B}$
B	$\bar{A}B$	AB

Tương ứng :

	0	1
0	00	10
1	01	11

	0	1
0	0	2
1	1	3

	$\bar{A}$	A
$\bar{B}$	0	0
B	1	1

Nếu hàm có 3 biến tức là $2^3 = 8$ ô, ta có bản đồ nh- sau:

BC	A	00	01	11	10
0					
1					

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	0
1	1	0	1
1	1	1	0

Nếu có bảng chân lý thì ta có bảng: điền các giá trị của tổ hợp các biến vào ô tương ứng :

BC	A	00	01	11	10
0		0	1	0	1
1		0	0	0	1

BC	A	00	01	11	10
0			1		2
1					6

Với hàm 4 biến ta có 16 ô nh- sau: (các biến là ABCD)

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

b. Cực tiểu hoá hệ thức trên bản đồ Karnaugh

- Trên bản đồ Karnaugh cứ 2 biến bỏ đi- ợc 1 biến liên quan, cứ 4 tổ hợp biến thì bỏ đi- ợc 2 biến, cứ 8 tổ hợp biến thì bỏ đi- ợc 3 biến, cứ 2^n tổ hợp thì bỏ đi- ợc n biến

Ví dụ 1: ta có hệ thức boole biểu diễn bảng sự thật sau:

ác tổ hợp biến có giá trị 1 kề nhau, đều có thể gán với nhau. Cứ 2 tổ hợp biến thì

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

BC	00	01	11	10
A				
0	1	1		1
1				1

- Hàm số rút ra từ bảng sự thật :

$$Y = \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + \bar{A} B \bar{C} + A B \bar{C}$$

- Từ bản đồ karnaugh:

$$\bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C = \bar{A} \bar{B}$$

$$\bar{A} B \bar{C} + A B \bar{C} = B \bar{C}$$

⇒ Ta có $Y = \bar{A} \bar{B} + B \bar{C}$

Ví dụ 2: Cho hàm boole : $Y = \Sigma (1,3,4,5,10,11,12,13)$

Ta vẽ bản đồ như sau: Vì tổ hợp biến có tới 13 giá trị vậy đây là hàm 4 biến số gọi 4 biến số là ABCD ta có bảng sau :

Ta có : $\Sigma (4,5,12,13) = B \bar{C}$

$$\Sigma (10,11) = A \cdot \bar{B} \cdot C$$

$$\Sigma (1,5) = \bar{A} \cdot C \cdot D$$

$$\Sigma (3,11) = \bar{B} \cdot C \cdot D$$

CD	00	01	11	10
AB				
00		1	3	
01	4	5		
11	12	13		
10			11	10

Chú ý:

Nguyên tắc chọn các số hạng:

- Phải bao gồm tất cả các số hạng nhỏ nhất của hàm (các số hạng nhận giá trị 1)
- Số các số hạng được chọn phải là ít nhất
- Số thừa số của mỗi số hạng phải là ít nhất

*** Tóm tắt:**

- Một hàm logic (hàm Boolean) có thể được biểu diễn dưới dạng một bảng chân lý, một bản đồ Karnaugh hoặc một mạch logic.
- Nếu giá trị của chính biến = 1 (A,B, ... = 1) thì đảo của biến = 0 (tức A, B ...= 0)
- Giá trị của hàm: nếu ta lấy trị của chính hàm = 1(thì đảo của hàm = 0) và ngược lại.
- Một hàm logic có thể được biểu diễn dưới 2 dạng:
 - Dạng tuyển chuẩn toàn phần
 - Dạng hội chuẩn toàn phần

* Người ta hay biểu diễn dưới dạng tuyển chuẩn toàn phần hơn. Hai dạng này là tương đương nhau.

* Dạng tuyển chuẩn toàn phần:

- Người ta lấy giá trị của hàm bằng tổng các giá trị của tích tổ hợp các biến số có giá trị 1.

Ví dụ: Nếu hàm $Y = f(A,B,C)$

$$Y = \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + ABC \text{ nếu như tổ hợp này có giá trị bằng 1}$$

* Dạng hội chuẩn toàn phần:

- Người ta lấy giá trị của hàm bằng cách lấy tích của tổng các tổ hợp biến số tại những ô mà tổ hợp biến nhận giá trị “0” với điều kiện là các biến được lấy giá trị đảo của nó.

Ví dụ: Cho hàm $Y = f(a,b)$ ta có bảng

- Y dạng hội sẽ là: $Y = (a + \bar{b}) . (\bar{a} + b)$
- Y dạng tuyển là: $Y = \bar{a} . \bar{b} + a . b$

		b	
		0	1
a	0	1	0
	1	1	0

7. Thiết kế mạch logic

*** Bài tập1:**

Trong nhà có 3 công tắc điện A,B,C chủ nhà muốn:

- + Đèn Y sáng khi 3 công tắc A,B,C đều hở.
- + Đèn Y sáng khi A và B đóng, còn C hở.

Hãy dùng các cổng NAND để thiết kế mạch điện theo yêu cầu trên.

Giải:

- Gọi các trạng thái đóng của các công tắc là 1, hở là 0.
Trạng thái sáng của bóng đèn là 1, tắt của bóng đèn là 0.
- Vậy ta có ta có hệ thức boole như sau:

$$Y = \bar{A} \bar{B} \bar{C} + AB\bar{C}$$