NỘI DUNG MÔN HỌC

Chương 1: Hệ đếm

Chương 2: Hàm Boole và cổng logic

Chương 3: Mạch logic tổ hợp

Chương 4: Mạch logic tuần tự

Chương 5: Bộ nhớ bán dẫn

Chương 2 – HÀM BOOLE VÀ CỔNG LOGIC

2.1. Đại số Boole

2.2. Phương pháp biểu diễn hàm Boole

2.3. Phương pháp tối thiểu hàm Boole

2.4. Cổng logic

2.5. Tham số chính

Chương 2 – HÀM BOOLE VÀ CỔNG LOGIC

2.1. Đại số Boole

2.1.1. Các định lý cơ bản

2.1.2. Các định luật cơ bản

2.1.3. Ba quy tắc về đẳng thức

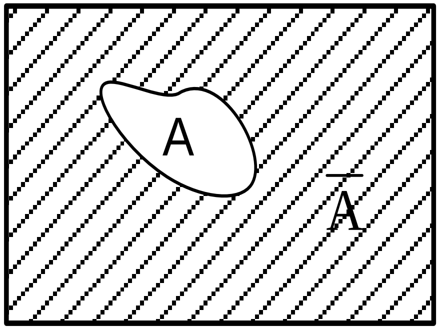
Đại số Boole:

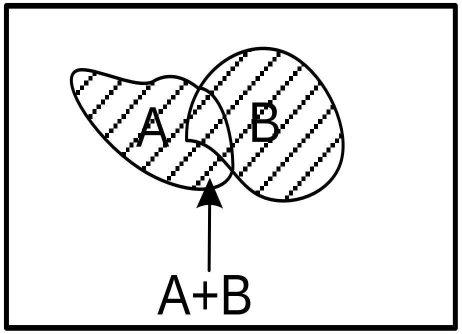
- Là một tập hợp các đối tượng có hai trạng thái: có hoặc không, đúng hoặc sai, được biểu diễn bằng biến logic với hai giá trị 1(A) và 0 ($\overline{A}$).

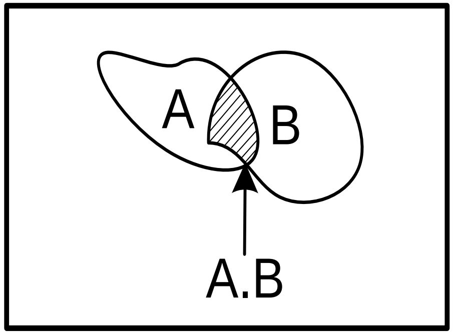
- Phép phủ định logic: $f(A) = \overline{A}$

- Phép cộng logic (phép hoặc): $f(A,B) = A + B$

- Phép nhân logic (phép và): $f(A,B) = A.B = AB$







2.1.1. Các định lý cơ bản

STT	Tên gọi	Dạng tích	Dạng tổng
1	Đồng nhất	X.1 = X	X + 0 = X
2	Phần tử 0, 1	X.0 = 0	X + 1 = 1
3	Bù	$X.\overline{X} = 0$	$X + \overline{X} = 1$
4	Bất biến	X.X = X	X + X = X
5	Hấp thụ	X + X.Y = X	X.(X + Y) = X
6	Phủ định đúp	$\overline{\overline{X}} = X$	
7	Định lý DeMorgan	$\overline{(X.Y.Z...)} = \overline{X} + \overline{Y} + \overline{Z} + ...$	$\overline{(X + Y + Z + ...)} = \overline{X}.\overline{Y}.\overline{Z}...$

2.1.2. Các định luật cơ bản

- Hoán vị: $X.Y = Y.X$

  $X + Y = Y + X$

- Kết hợp: $X.(Y.Z) = (X.Y).Z$

  $X + (Y + Z) = (X + Y) + Z$

- Phân phối: $X.(Y + Z) = X.Y + X.Z$

  $(X + Y).(X + Z) = X + Y.Z$

- Nhất quán: nếu $X + Y = Y$ thì $X.Y = X$

2.1.3. Ba quy tắc về đẳng thức

- *Quy tắc thay thế:* trong bất kỳ đẳng thức logic nào, nếu thay một biến bằng một hàm thì đẳng thức vẫn được thiết lập.

  Ví dụ:

- *Quy tắc tìm hàm đảo:* Phép đảo của hàm số được thực hiện bằng cách đổi dấu nhân thành dấu cộng và ngược lại; đổi 0 thành 1 và ngược lại; đổi nguyên biến thành đảo biến và ngược lại. Giữ nguyên dấu đảo của hàm nhiều biến, tuân thủ nguyên tắc “nhân trước, cộng sau”.

  Ví dụ:

- *Quy tắc đối ngẫu:* Hàm F và F’ được gọi là đối ngẫu với nhau khi các dấu cộng và dấu nhân, các số 0 và số 1 được đổi chỗ cho nhau một cách tương ứng.

Chương 2 – HÀM BOOLE VÀ CỔNG LOGIC

2.1. Đại số Boole

2.2. Phương pháp biểu diễn hàm Boole

2.3. Phương pháp tối thiểu hàm Boole

2.4. Cổng logic

2.5. Tham số chính

2.6. Một số lưu ý khi sử dụng IC số

2.7. Các họ cổng logic (dạy bổ sung)

2.2. Các phương pháp biểu diễn hàm Boole

2.2.1. Bảng trạng thái

2.2.2. Biểu thức đại số

2.2.3. Bảng Karnaugh

2.2.1. Bảng trạng thái (1)

- Bảng trạng thái gồm các cột, liệt kê giá trị (trạng thái) mỗi biến theo từng cột và giá trị hàm theo một cột riêng (thường là bên phải bảng).

- Bảng trạng thái còn được gọi là ***bảng sự thật*** hay ***bảng chân lý***.

- Hàm n biến sẽ có 2^n bộ giá trị.

A	B	C	f1 = ABC	f2 = A+B+C	f3 = A+B.C
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1

2.2.1. Bảng trạng thái (2)

- Hạng tích (**minterm)**: ký hiệu m_i, với i = 0 đến 2^n -1, là các tổ hợp gồm tích các biến, trong đó:

Giá trị ‘1’ được biểu diễn bằng nguyên biến (biến trực tiếp)

Giá trị ‘0’ được biểu diễn bằng đảo biến (biến phủ định)

- Hạng tổng (**Maxterm**): ký hiệu M_i, với i = 0 đến 2^n -1, là các tổ hợp gồm tổng các biến, trong đó:

Giá trị ‘0’ được biểu diễn bằng nguyên biến (biến trực tiếp)

Giá trị ‘1’ được biểu diễn bằng đảo biến (biến phủ định)

- Mối quan hệ giữa minterm và Maxterm:

$$\overline{m_i} = M_i$$

$$m_i = \overline{M_i}$$

2.2.1. Bảng trạng thái (3)

A	B	C	f	m	M
0	0	0	0	$m_0 = \overline{A}\,\overline{B}\,\overline{C}$	$M_0 = A + B + C$
0	0	1	0	$m_1 = \overline{A}\,\overline{B}C$	$M_1 = A + B + \overline{C}$
0	1	0	0	$m_2 = \overline{A}B\overline{C}$	$M_2 = A + \overline{B} + C$
0	1	1	0	$m_3 = \overline{A}BC$	$M_3 = A + \overline{B} + \overline{C}$
1	0	0	0	$m_4 = A\overline{B}\,\overline{C}$	$M_4 = \overline{A} + B + C$
1	0	1	0	$m_5 = A\overline{B}C$	$M_5 = \overline{A} + B + \overline{C}$
1	1	0	0	$m_6 = AB\overline{C}$	$M_6 = \overline{A} + \overline{B} + C$
1	1	1	1	$m_7 = ABC$	$M_7 = \overline{A} + \overline{B} + \overline{C}$

2.2.2. Biểu thức đại số (1)

- Có 2 dạng biểu diễn:

Dạng tuyển (tổng các tích): Mỗi số hạng là một *hạng tích* hay *minterm* (m_i).

Dạng hội (tích các tổng): Mỗi thừa số là một *hạng tổng* hay *Maxterm* (M_i).

- Nếu trong tất cả các hạng tích hay hạng tổng có đủ mặt các biến thì dạng tổng các tích hay dạng tích các tổng tương ứng được gọi là dạng chuẩn. Dạng chuẩn là duy nhất.

- Biểu diễn hàm dưới dạng tổng các tích: $f\left(X_{n-1},...,X_0\right)=\sum_{i=0}^{2^n-1} a_i m_i$

- Biểu diễn hàm dưới dạng tích các tổng: $f\left(X_{n-1},...,X_0\right)=\prod_{i=0}^{2^n-1}\left(a_i+M_i\right)$

Với a_i chỉ nhận hai giá trị 0 và 1.

2.2.2. Biểu thức đại số (2) – Dạng tổng các tích

$$f_1(A,B,C) = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

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

$$= 011 + 101 + 110 + 111$$

$$= m_3 + m_5 + m_6 + m_7$$

$$= \sum(3,5,6,7)$$

$$f_2(A,B,C) = A + BC$$

2.2.2. Biểu thức đại số (3) – Dạng tổng các tích

** Chuẩn hoá hàm về dạng chuẩn tổng các tích (chuẩn tắc tuyển):*

- Thêm các biến còn thiếu vào các hạng tích mà không làm ảnh hưởng đến kết quả bằng cách nhân hạng tích đó với ‘1’ (tổng của nguyên biến và đảo biến còn thiếu).
- Loại bỏ các hạng tích lặp lại (hạng tích thừa).

Ví dụ: $f_2(A,B,C) = A + BC$

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

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

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

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

$$= \sum(3,4,5,6,7)$$

2.2.2. Biểu thức đại số (4) – Dạng tích các tổng

$$f_1(A,B,C) = (A+B+C)(A+B+\overline{C})(A+\overline{B}+C)$$
$$= (0+0+0)(0+0+1)(0+1+0)$$
$$= M_0 \;.\; M_1 \;.\; M_2$$
$$= \prod(0,1,2)$$

$$f_2(A,B,C) = (A+B)(A+C)$$

2.2.2. Biểu thức đại số (5) – Dạng tích các tổng

** Chuẩn hoá hàm về dạng chuẩn tích các tổng (chuẩn tắc hội):*

- Thêm các biến còn thiếu vào các hạng tổng mà không làm ảnh hưởng đến kết quả bằng cách cộng hạng tổng đó với ‘0’ (tích của nguyên biến và đảo biến còn thiếu).
- Loại bỏ các hạng tổng lặp lại (hạng tổng thừa).

Ví dụ:

$$
\begin{aligned}
f_3(A,B,C) &= (A+B)(A+\overline{C}) \\
&= (A+B+C.\overline{C})(A+B.\overline{B}+\overline{C}) \\
&= (A+B+C)(A+B+\overline{C})(A+B+\overline{C})(A+\overline{B}+\overline{C}) \\
&= (A+B+C)(A+B+\overline{C})(A+\overline{B}+\overline{C}) \\
&= \prod(0,1,3)
\end{aligned}
$$

2.2.3. Bảng Karnaugh (1)

- *Tổ chức của bảng Karnaugh:*
  - Các tổ hợp biến được viết theo một dòng (thường là phía trên) và một cột (thường là bên trái).
  - Hàm logic n biến có 2^n ô.
  - Mỗi ô thể hiện một hạng tích hay một hạng tổng, các hạng tích trong **hai *ô kế cận* chỉ khác nhau một biến**.
- *Tính tuần hoàn của bảng Các nô:*
  - Không những các *ô kế cận khác nhau một biến* mà *các ô đầu dòng* và *cuối dòng*, *đầu cột* và *cuối cột* cũng chỉ *khác nhau một biến* (kể cả 4 góc vuông của bảng). Bởi vậy *các ô này cũng gọi là kế cận*.

A \ BC	00	01	11	10
0				
1				

AB \ CD	00	01	11	10
00				
01				
11				
10				

2.2.3. Bảng Karnaugh (2)

➢ *Thiết lập bảng Các nô của một hàm:*

- Dưới dạng tổng các tích: ghi giá trị 1 vào các ô ứng với hạng tích có mặt trong biểu diễn, các ô còn lại lấy giá trị 0.

- Dưới dạng tích các tổng: ghi giá trị 0 vào các ô ứng với hạng tổng có mặt trong biểu diễn, các ô còn lại lấy giá trị 1.

$$f_1(A,B,C) = \sum(3,4,5,6,7)$$

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

$$f_2(A,B,C) = \prod(0,1,3)$$

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

2.2.3. Bảng Karnaugh (3)

➢*Hàm chuyển mạch không đầy đủ:*

- Có thể tồn tại một số hạng tích hay hạng tổng mà tại đó hàm không xác định (có thể lấy giá trị ‘1’ hoặc ‘0’). Các giá trị này gọi là các giá trị tuỳ chọn (x).

$$f = \sum(m_i) + \sum_d(m_j) = \sum(m_i) + \sum_K(m_j)$$

$$f = \prod(M_i).\prod_d(M_j) = \prod(M_i).\prod_K(M_j)$$

$$f_3(A,B,C) = \sum(1,4,5) + \sum_d(6,7)$$

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

$$f_4(A,B,C) = \prod(2,3,6) + \prod_d(0,4)$$

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

Chương 2 – HÀM BOOLE VÀ CỔNG LOGIC

2.1. Đại số Boole

2.2. Phương pháp biểu diễn hàm Boole

2.3. Phương pháp tối thiểu hàm Boole

2.4. Cổng logic

2.5. Tham số chính

2.3. Các phương pháp rút gọn hàm Boole

2.3.1. Phương pháp đại số

2.3.2. Phương pháp bảng Karnaugh

2.3.3. Phương pháp Quine Mc. Cluskey

Các phương pháp rút gọn hàm Boole

- Rút gọn mạch logic tổ hợp có một vai trò quan trọng trong việc tối giản các thiết kế mạch logic tổ hợp.
- Có 3 phương pháp phổ biến được sử dụng để tối giản mạch logic tổ hợp:
  - Phương pháp đại số,
  - Phương pháp bảng Kanaugh,
  - Phương pháp Quine Mc. Cluskey.
- Phương pháp đại số và bảng Kanaugh: rút gọn mạch logic với số lượng biến không lớn (thường < 6), thực hiện thủ công.
- Phương pháp Quine Mc. Cluskey: rút gọn được hàm (mạch) nhiều biến và có thể thực hiện nhờ máy tính.

2.3.1. Phương pháp đại số (1)

- Dựa vào các định lý đã học để đưa biểu thức về dạng tối giản.
- Một số cách rút gọn bằng phương pháp đại số
  1. Loại bỏ tổ hợp thừa
  2. Áp dụng định lí De Morgan
  3. Triển khai từ thành phần nhiều biến
  4. Triển khai từ thành phần ít biến, đặt nhân tử chung

2.3.1. Phương pháp đại số (2)

1. Loại bỏ tổ hợp thừa

Ví dụ: Hãy đưa hàm logic về dạng tối giản:

$$f = AB + \bar{A}C + BC$$

Áp dụng định lý $A + \bar{A} = 1$, $X + XY = X$, ta có:

$$f = AB + \bar{A}C + BC\left(A + \bar{A}\right)$$

$$= AB + ABC + \bar{A}C + \bar{A}BC$$

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

=> nếu trong tổng các tích, xuất hiện một biến và đảo của biến đó trong hai số hạng khác nhau, các thừa số còn lại trong hai số hạng đó tạo thành thừa số của một số hạng thứ ba thì số hạng thứ ba đó là thừa và có thể bỏ đi.

2.3.1. Phương pháp đại số (3)

2. Áp dụng định lí DeMorgan

Ví dụ: Rút gọn biểu thức: $f = \overline{CD + \bar{C}\bar{D}}.\overline{\bar{A}C + \bar{D}}$

Áp dụng linh hoạt định lý DeMorgan, ta có:

$$f = \overline{CD + \bar{C}\bar{D} + \bar{A}C + \bar{D}}\ (Chuyển\ tích\ về\ tổng)$$

$$= \overline{CD + \bar{D} + \bar{A}C}$$

$$= \overline{C + \bar{D} + \bar{A}C}$$

$$= \overline{C + \bar{D}}$$

$$= \bar{C}D$$

Chú ý quan trọng:

$$X\bar{Y} + Y = X\bar{Y} + Y(X+1)$$

$$= X\bar{Y} + XY + Y$$

$$= X + Y$$

2.3.1. Phương pháp đại số (4)

3. Triển khai từ thành phần nhiều biến

Trong biểu thức dạng tổng các tích, số hạng nào có chứa nhiều biến nhất (nhưng không chứa đầy đủ các biến), thì ta áp dụng định lí bù bổ sung các biến còn thiếu để số hạng đó trở thành chứa đầy đủ thành phần các biến, đặt thừa số chung (nếu có) với các số hạng khác để triệt tiêu và tiếp tục áp dụng các định lí khác để rút gọn.

Ví dụ: Hãy đưa hàm logic về dạng tối giản: $f = AB + BCD + \bar{A}C + \bar{B}C$

Lời giải: Vi du: $f_1 = A\bar{D} + \bar{B}D + B\bar{C}D + \bar{A}CD + ABC$

Nhận thấy trong biểu thức tổng các tích có số hạng BCD có chứa nhiều biến nhất nhưng không chứa biến A, Áp dụng định lý, $A + \bar{A} = 1$, nên ta lấy $BCD\,(A + \bar{A})$

Áp dụng định lý, $A + A.B = A$ ta có:

$$f = AB + BCD(A + \bar{A}) + \bar{A}C + \bar{B}C = (AB + ABCD) + (\bar{A}BCD + \bar{A}C) + \bar{B}C$$

$$= AB + \bar{A}C + \bar{B}C = AB + \overline{AB}.C = AB(1 + C) + \overline{AB}.C$$

$$= AB + C$$

2.3.1. Phương pháp đại số (5)

4. Triển khai từ thành phần ít biến, đặt nhân tử chung

Khi hai hay một vài số hạng có chứa một biến thành phần nào đó giống nhau, mà sau khi đặt thành phần biến giống nhau đó làm thừa số chung thì trong ngoặc sẽ xuất hiện một tổ hợp có chứa các thành phần mà có chứa biến giống với số hạng khác trong biểu thức, thì ta sẽ làm theo phương pháp đặt nhân tử chung đó rồi áp dụng các định lí vào rút gọn.

Ví dụ: Rút gọn biểu thức sau: $f = AB + BCD + \bar{A}C + \bar{B}C$

Nhận thấy trong biểu thức có $\bar{A}C$, $\bar{B}C$ có C chung, nếu sau khi đặt C ra làm thừa số chung thì trong ngoặc có $(\bar{A} + \bar{B}) = \overline{AB}$, có chứa thành phần biến giống AB. Do đó đặt C làm thừa số chung, áp dụng thêm định lí A+AB=A, $A + \bar{A}B = A + B$, ta có

$$
\begin{aligned}
f &= AB + BCD + \bar{A}C + \bar{B}C \\
&= AB + BCD + C(\bar{A} + \bar{B}) \\
&= AB + BCD + C\overline{AB} \\
&= (AB + C\overline{AB}) + BCD \\
&= AB + C + BCD \\
&= AB + C
\end{aligned}
$$

<u>Ví du</u>: $f_1 = A\bar{B} + BD + CDE + D\bar{A}$

2.3.2. Phương pháp bảng Karnaugh (1)

* Tối thiểu hóa hàm dạng tổng các tích:

- Gộp các ô kế cận có giá trị ‘1’ lại thành từng nhóm 2, 4, ..., 2^i các ô.
- Thay mỗi nhóm bằng một tích mới, trong đó giữ lại các biến giống nhau theo hàng và cột.
- Biến có giá trị ‘1’ được biểu diễn bằng nguyên biến, biến có giá trị ‘0’ được biểu diễn bằng đảo biến.
- Cộng các tích mới, được hàm đã tối giản.

AB \ CD	00	01	11	10
00			1	1
01			1	1
11	1	1	1	1
10			1	1

$f_2 = AB$ $\quad$ $f_1 = C$

2.3.2. Phương pháp bảng Karnaugh (2)

** Chú ý:*

- Số ô trong mỗi nhóm càng lớn kết quả thu được càng tối giản. Gộp 2^i các ô lân cận, rút gọn được i biến.

- Một ô có thể được gộp nhiều lần trong các nhóm khác nhau. Mỗi nhóm phải có ít nhất một ô mới (ô chưa được gộp trong các nhóm khác).

- Nếu gộp theo các ô có giá trị ‘0’ ta sẽ thu được biểu thức bù của hàm.

- Với các giá trị tuỳ chọn, chọn ‘x’ bằng ‘1’ hoặc ‘0’ để số ô gộp được là tối đa; số nhóm là tối thiểu.

** Ví dụ:*

2.3.2. Phương pháp bảng Karnaugh (3)

$$f_1(A,B,C) = \sum(0,1,3,4,5)$$

$$f_1(A,B,C) = \overline{B} + \overline{A}C$$

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

$\overline{B}$ $\overline{A}C$

$$f_2(A,B,C) = \sum(3,7) + \sum_d(1,2,6)$$

$$f_2(A,B,C) = B$$

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

B

2.3.2. Phương pháp bảng Karnaugh (4)

$$f_3(A,B,C,D)=\sum(3,6,8,9,11,12)+\sum_d(0,1,2,13,14,15)$$

AB \ CD	00	01	11	10
00	x	x	1	x
01	0	0	0	1
11	1	x	x	x
10	1	1	1	0

$A\overline{C}$ $\overline{B}D$ $BC\overline{D}$

$$f_3=A\overline{C}+\overline{B}D+BC\overline{D}$$

$$f_4(A,B,C,D)=\sum(0,1,2,3,5,7,8,9,10,13)$$

2.3.2. Phương pháp bảng Karnaugh (4)

***** Tối thiểu hóa hàm dạng tích các tổng:

- Gộp các ô kế cận có giá trị ‘0’ lại thành từng nhóm 2, 4, ..., 2^i các ô.
- Thay mỗi nhóm bằng một tổng mới, trong đó giữ lại các biến giống nhau theo hàng và cột.
- Biến có giá trị ‘1’ biểu diễn bằng đảo biến, biến có giá trị ‘0’ biểu diễn bằng nguyên biến.
- Nhân các tổng mới, được hàm đã tối giản.
- Gộp các ô ‘1’ sẽ thu được biểu thức bù của hàm.

CD / AB	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	1	1	1	1
10	0	0	1	1

$f_1 = A+C$

$f_2 = B+C$

2.3.2. Phương pháp bảng Karnaugh (5)

$$f(A,B,C,D)=\prod(0,2,3,4,5,8).\prod_d(10,11,12,13,14,15)$$

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	0	1	1
11	x	x	x	x
10	0	1	x	x

$B+\overline{C}$

$B+D$ $\quad \overline{B}+C$

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

2.3.3. Phương pháp Quine Mc. Cluskey (1)

* Quy tắc:

- Lập bảng liệt kê các hạng tích dưới dạng nhị phân theo từng nhóm với số lượng bit 1 bằng nhau và xếp chúng theo số bit 1 tăng dần.
- Gộp 2 hạng tích của mỗi cặp nhóm chỉ khác nhau 1 bit để tạo các nhóm mới. Trong mỗi nhóm mới, giữ lại các biến giống nhau, biến bỏ đi thay bằng một dấu ngang (-).

  Lặp lại cho đến khi trong các nhóm tạo thành không còn khả năng gộp nữa. Mỗi lần rút gọn, ta đánh dấu # vào các hạng ghép cặp được.
- Các hạng tích không đánh dấu trong mỗi lần rút gọn sẽ được tập hợp lại để lựa chọn biểu thức tối giản. Cách lập bảng lựa chọn hàm như sau:
  - –Ta thiết lập các số hạng có thể có trong biểu thức bằng cách thay dấu gạch ngang bằng các giá trị 0 và 1 sau đó đánh dấu ký hiệu “x” dưới vị trí mà nó chứa số hạng đó.
  - –Sau đó ta xem xét các cột chỉ chứa một dấu “x”. Biểu thức tối giản là tổng của các hạng tích tương ứng với các cột này. Trong các hạng tích đó, bỏ đi các biến ương ứng với dấu (-).

2.3.3. Phương pháp Quine Mc. Cluskey (2)

- Ví dụ: Rút gọn $f(A,B,C,D)=\sum(0,1,2,5,6,8,9,10,13,14)$

Bảng a		Bảng b	
Hạng tích đã sắp xếp	Nhị phân A B C D	Rút gọn lần thứ nhất. A B C D	Rút gọn lần thứ 2 A B C D
0	0 0 0 0	0 0 0 – # (0,1)	– 0 0 – # (0,1,8,9)
1	0 0 0 1	0 0 – 0 # (0,2)	– 0 – 0 # (0,2,8,10)
2	0 0 1 0	– 0 0 0 # (0,8)	– – 0 1 # (1,5,9,13)
8	1 0 0 0	0 – 0 1 # (1,5)	– – 1 0 # (2,6,10,14)
5	0 1 0 1	– 0 0 1 # (1,9)	
6	0 1 1 0	0 – 1 0 # (2,6)	
9	1 0 0 1	– 0 1 0 # (2,10)	
10	1 0 1 0	1 0 0 – # (8,9)	
13	1 1 0 1	1 0 – 0 # (8,10)	
14	1 1 1 0	– 1 0 1 # (5,13)	
		– 1 1 0 # (6,14)	
		1 – 0 1 # (9,13)	
		1 – 1 0 # (10,14)	

) Lập bảng, liệt kê các nhóm đại diện cho các phần tử tương ứng:

Nhóm	0	1	2	5	6	9	10	13	14	
– 0 – 0	X		X				X			
– 0 0 –	X	X				X				
– – 0 1		X		X		X		X		
– – 1 0			X		X		X		X	

2.3.3. Phương pháp Quine Mc. Cluskey (3)

***) Đánh dấu các phần tử là xuất hiện duy nhất 1 lần.**

- Khi đó nhóm tương ứng với phần tử đó là **nhóm bắt buộc** phải có trong biểu thức tối giản, Phần tử thuộc nhóm bắt buộc nào đó thì ta không cần xét đến nhóm cũng chứa nó khác.
- Các phần tử xuất hiện từ 2 lần trở lên (ứng với 2 nhóm khác nhau trở lên) thì nhóm đó có thể coi là **tùy chọn**.

Nhóm	0	1	2	5	6	9	10	13	14	
– 0 – 0	X		X				X			
– 0 0 –	X	X				X				
– – 0 1		X		⊗		X		⊗		5, 13 chỉ xuất hiện một lần
– – 1 0			X		⊗		X		⊗	6, 14 chỉ xuất hiện 1 lần

- Nhóm bắt buộc: – – 0 1; – – 1 0;

+ Đánh dấu các phần tử chỉ xuất hiện 1 lần để tìm nhóm bắt buộc.

+ Đánh dấu (tô xanh) các phần tử không xuất hiện một lần nhưng đã nằm trong nhóm bắt buộc.

- Nhóm tùy chọn: – 0 – 0 và – 0 0 –;

+ Sau khi tìm ra nhóm bắt buộc và các phần tử đã nằm trong nhóm bắt buộc, ta nhận thấy còn một phần tử chưa nằm trong nhóm bắt buộc là: 0.

+ Có hai nhóm có thể làm đại diện cho phần tử này là: – 0 – 0 và – 0 0 –; và ta có thể lấy nhóm nào làm đại diện cho hai phần tử này cũng được.

Kết quả tối giản cuối cùng phải là:

– – 0 1; – – 1 0; – 0 – 0; TƯƠNG ỨNG: F(A,B,C,D) = cD + Cd + bd

Hoặc:

– – 0 1; – – 1 0; – 0 0 –; TƯƠNG ỨNG: F(A,B,C,D) = cD + Cd + bc

Câu hỏi ôn tập phần 1

- Chuyển các hàm sang dạng chuẩn của minterm và maxterm:

a) $A.(B+\overline{A.\overline{C}}).\overline{(A+B.\overline{C})}$ b) $\overline{A.B.\overline{A.C.\overline{\overline{A.D}}}}+\overline{B.C.\overline{\overline{B.C}}}+\overline{C.D}$

- Rút gọn hàm sau theo phương pháp bảng Karnaugh:

a) F (A, B, C, D) = Σ (3, 7, 8, 9, 10, 12).

b) F (A, B, C, D) = Σ (0, 1, 4, 9, 12, 13) + Σd (2, 3, 6, 10, 11, 14).

- Rút gọn hàm sau theo phương pháp đại số:

a) $\overline{CD+\overline{C}\,\overline{D}}\,.\,\overline{\overline{A}C+\overline{D}}$ b) $\overline{\overline{A}}\,\overline{\overline{B}}\,\overline{\overline{C}}\,.\,\overline{\overline{A}\,\overline{B}+\overline{B}\,\overline{C}+\overline{C}\,\overline{A}}$

- Tối thiểu hóa bằng phương pháp Quine Mc. Cluskey:

a) F (A, B, C, D) = Σ (0, 2, 5, 6, 7, 10, 13, 14, 15)

Giảng viên: TS. Nguyễn Trung Hiếu
Bộ môn: Điện tử máy tính - Khoa KTĐT1

Chương 2 – HÀM BOOLE VÀ CỔNG LOGIC

2.1. Đại số Boole

2.2. Phương pháp biểu diễn hàm Boole

2.3. Phương pháp tối thiểu hàm Boole

2.4. Cổng logic

2.5. Tham số chính

Chương 2 – HÀM BOOLE VÀ CỔNG LOGIC

2.4. Cổng logic

2.4.1. Cổng logic cơ bản

2.4.2. Một số cổng ghép thông dụng

2.4.3. Tính đa chức năng của cổng NAND, NOR

2.4.4. Logic dương và logic âm

2.4.1. Cổng logic cơ bản

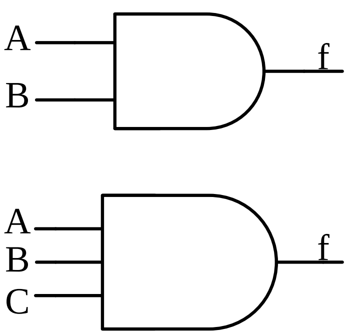
- Cổng AND
- Cổng OR
- Cổng NOT

2.4.1. Cổng logic cơ bản (1) – Cổng AND

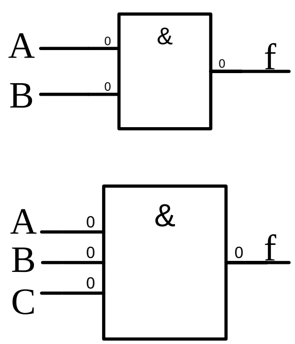
- Hàm ra của cổng AND 2 và nhiều biến vào như sau:

$$f = f(A,B) = AB; \qquad f = f(A,B,C,D,...) = A.B.C.D...$$

Ký hiệu cổng AND



Chuẩn ANSI



Chuẩn IEEE

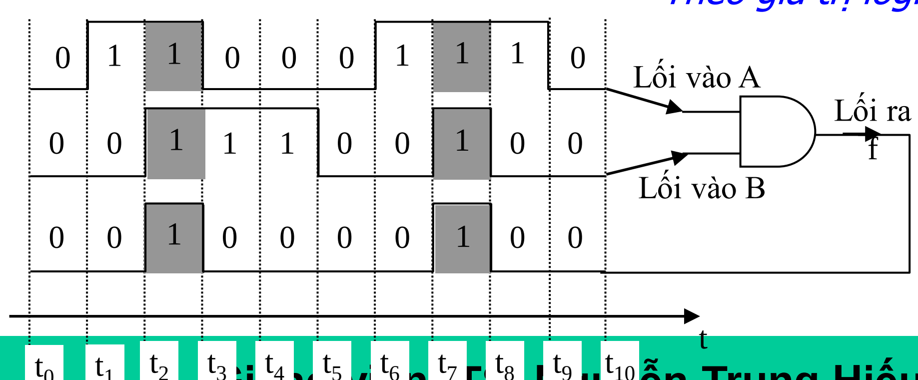
Bảng trạng thái cổng AND 2 lối vào

A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

Theo giá trị logic

A	B	f
L	L	L
L	H	L
H	L	L
H	H	H

Theo mức logic



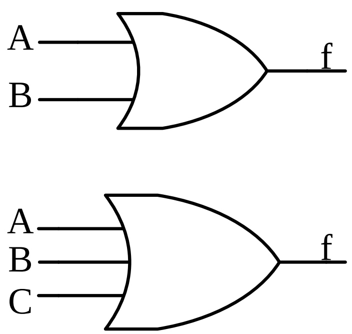
Đồ thị dạng xung vào, ra của cổng AND

2.4.1. Cổng logic cơ bản (2) – Cổng OR

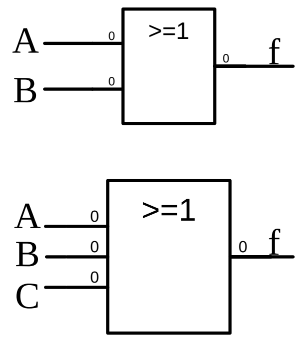
- Hàm ra của cổng OR 2 và nhiều biến vào như sau:

$$f = f(A,B) = A + B; \qquad f = f(A,B,C,D,...) = A + B + C + D + ...$$

Ký hiệu cổng OR



Chuẩn ANSI



Chuẩn IEEE

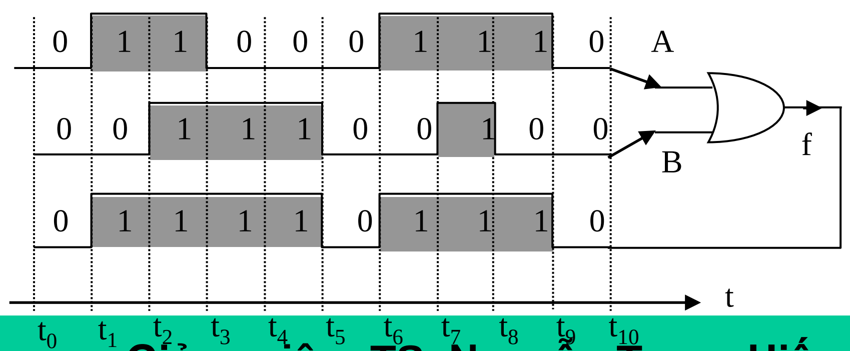
Bảng trạng thái cổng OR 2 lối vào

A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

Theo giá trị logic

A	B	f
L	L	L
L	H	H
H	L	H
H	H	H

Theo mức logic



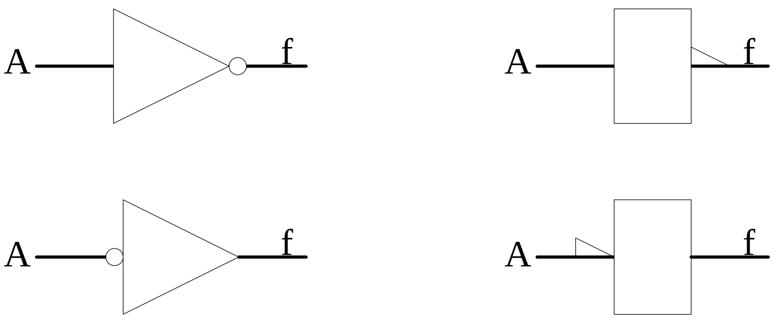
Đồ thị dạng xung của cổng OR.

2.4.1. Cổng logic cơ bản (3) – Cổng NOT

- Hàm ra của cổng NOT:

$$f = \overline{A}$$

Ký hiệu cổng NOT



Chuẩn ANSI   Chuẩn IEEE

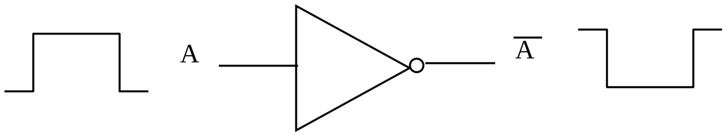
Bảng trạng thái cổng NOT

A	f
0	1
1	0

Theo giá trị logic

A	f
L	H
H	L

Theo mức logic



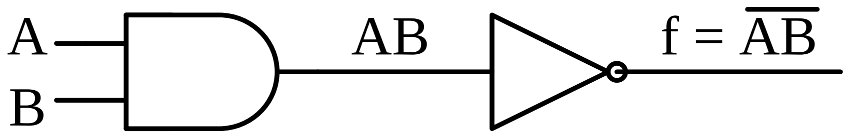
Dạng xung ra

2.4.2. Một số cổng ghép thông dụng

- Cổng NAND
- Cổng NOR
- Cổng khác dấu (XOR)
- Cổng đồng dấu (XNOR)

2.4.2. Một số cổng ghép thông dụng (1) – Cổng NAND

- Ghép nối tiếp một cổng AND với một cổng NOT ta được cổng NAND.

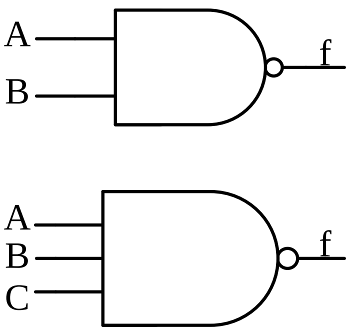


- Hàm ra của cổng NAND 2 và nhiều biến vào như sau:

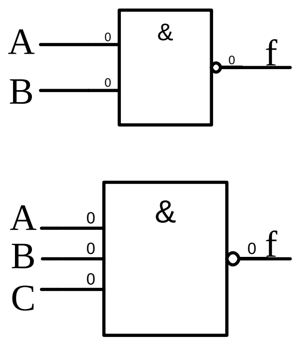
$$f = \overline{AB}$$

$$f = \overline{ABCD...}$$

Ký hiệu cổng NAND



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng NAND 2 lối vào

A	B	f
0	0	1
0	1	1
1	0	1
1	1	0

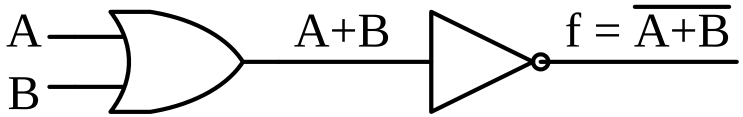
Theo giá trị logic

A	B	f
L	L	H
L	H	H
H	L	H
H	H	L

Theo mức logic

2.4.2. Một số cổng ghép thông dụng (2) – Cổng NOR

- Ghép nối tiếp một cổng OR với một cổng NOT ta được cổng NOR.

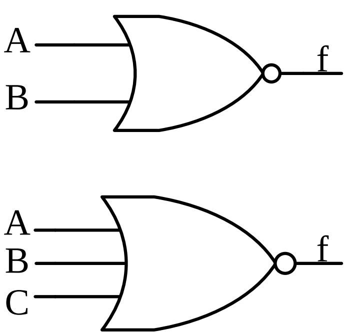


- Hàm ra của cổng NOR 2 và nhiều biến vào như sau:

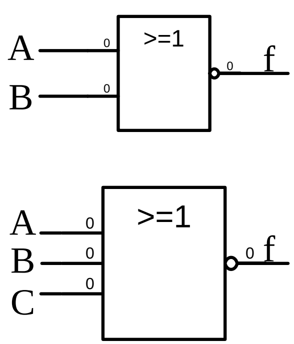
$$f = \overline{A + B}$$

$$f = \overline{A + B + C + D + \ldots}$$

Ký hiệu cổng NOR



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng NOR 2 lối vào

A	B	f
0	0	1
0	1	0
1	0	0
1	1	0

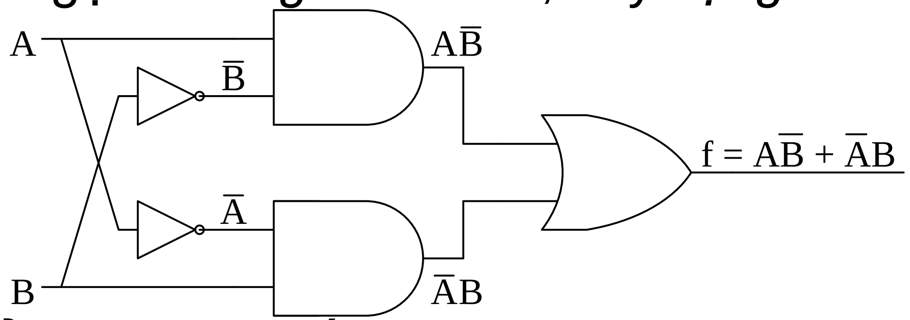
Theo giá trị logic

A	B	f
L	L	H
L	H	L
H	L	L
H	H	L

Theo mức logic

2.4.2. *Một số cổng ghép thông dụng (3) – Cổng XOR*

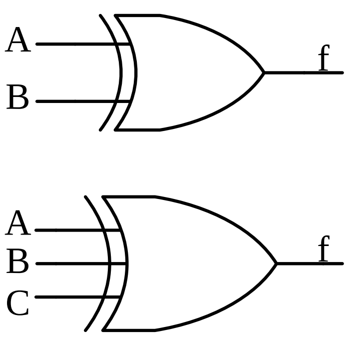
- *Cổng XOR* còn gọi là *cổng khác dấu*, hay *cộng modul 2*.



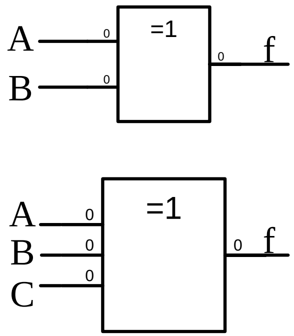
- Hàm ra của cổng XOR 2 biến vào như sau:

$$f = A\overline{B} + \overline{A}B \qquad \text{hay} \qquad f = A \oplus B$$

Ký hiệu cổng XOR



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng XOR 2 lối vào

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

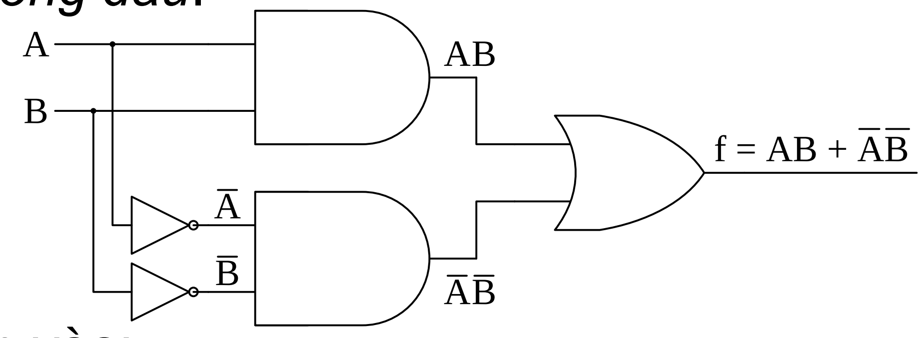
Theo giá trị logic

A	B	f
L	L	L
L	H	H
H	L	H
H	H	L

Theo mức logic

2.4.2. Một số cổng ghép thông dụng (4) – Cổng XNOR

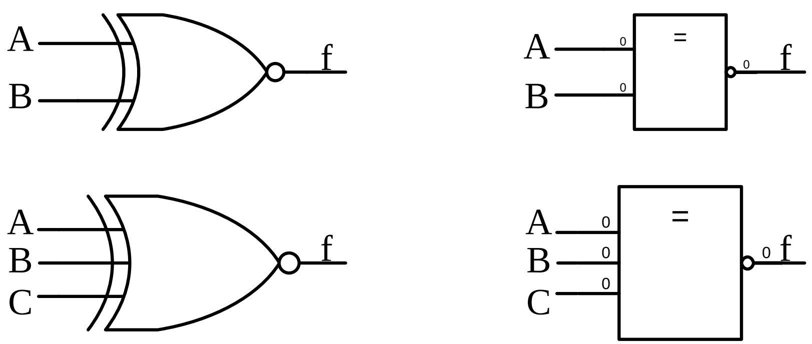
- *Cổng XNOR* còn gọi là *cổng đồng dấu*.



- Hàm ra của cổng XNOR 2 biến vào:

$$f = AB + \overline{A}\overline{B} \qquad \text{hay} \qquad f = \overline{A \oplus B} = A \sim B$$

Ký hiệu cổng XNOR



Chuẩn ANSI

Chuẩn IEEE

Bảng trạng thái cổng XNOR 2 lối vào

A	B	f
0	0	1
0	1	0
1	0	0
1	1	1

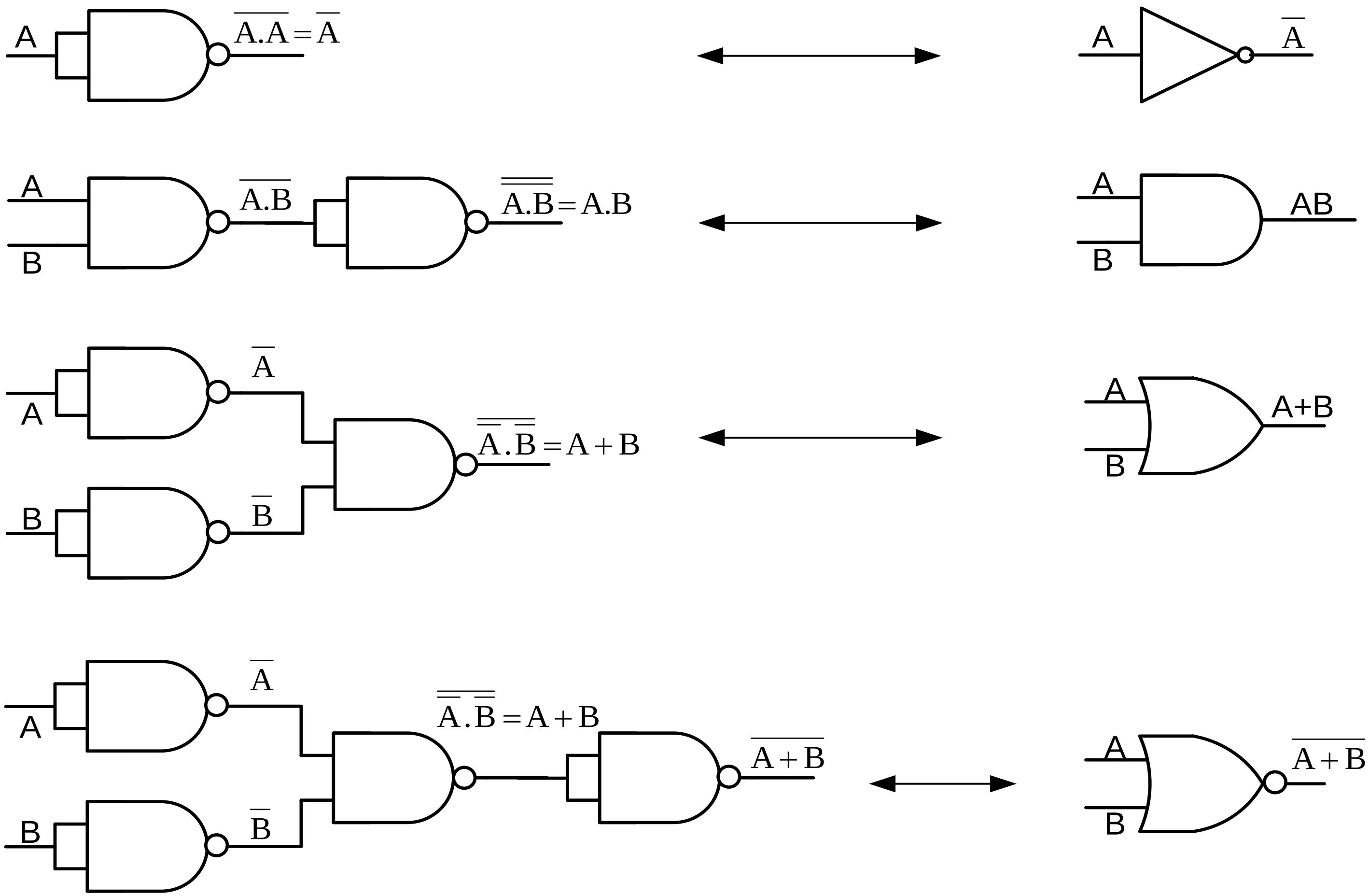
Theo giá trị logic

A	B	f
L	L	H
L	H	L
H	L	L
H	H	H

Theo mức logic

2.4.3. Tính đa chức năng của cổng NAND, NOR (1)

- Từ cổng NAND có thể tạo ra các cổng NOT, AND, OR và NOR.



2.4.3. Tính đa chức năng của cổng NAND, NOR (2)

- Cổng XOR dùng toàn cổng NAND.

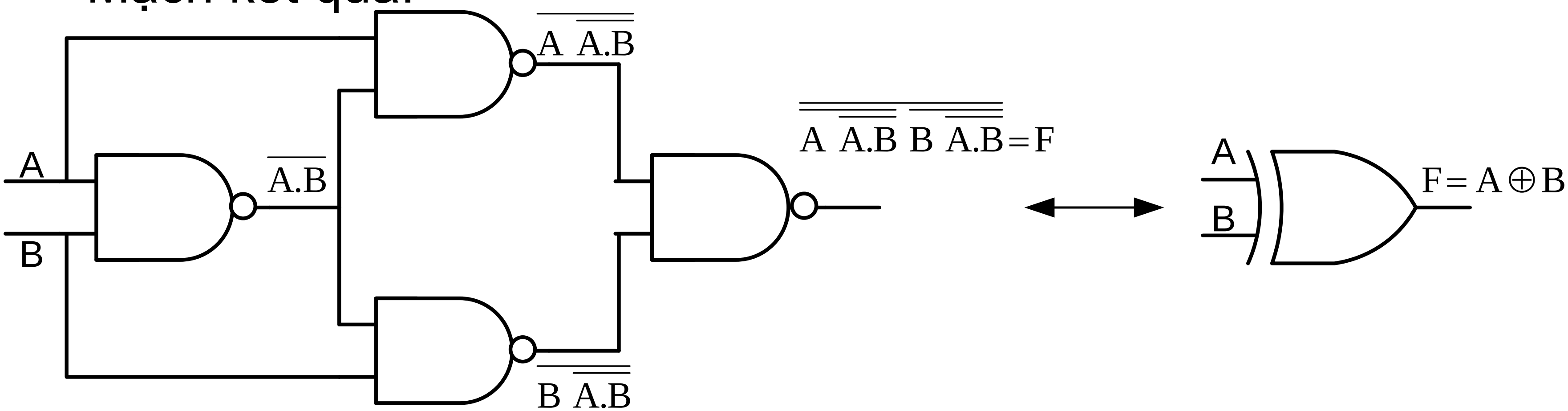
Hàm cổng XOR: $F = A\overline{B} + \overline{A}B$

Biến đổi:

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

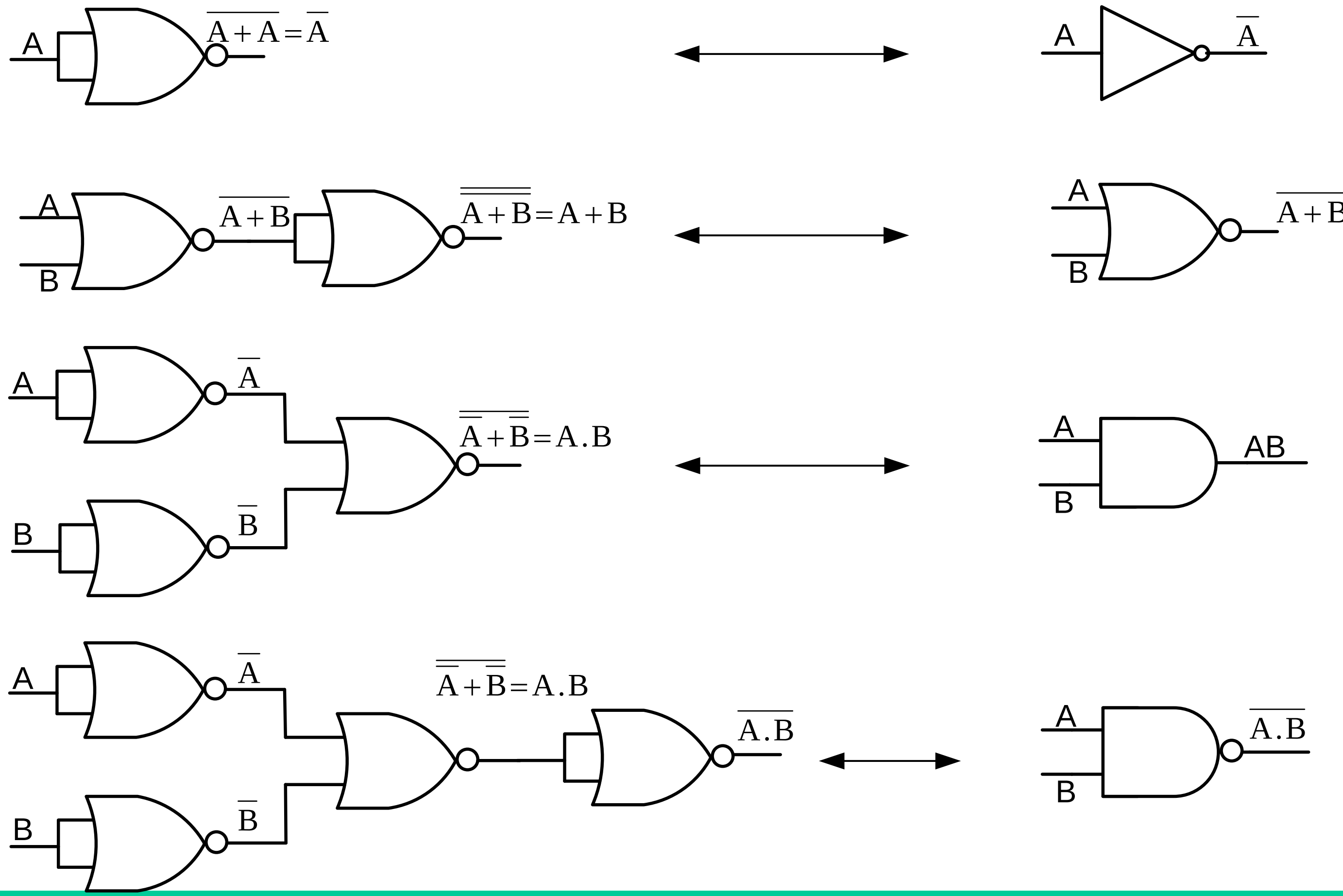
$$= A\overline{AB} + B\overline{AB} = \overline{\overline{A\overline{AB} + B\overline{AB}}} = \overline{\overline{A\overline{AB}}\,.\,\overline{B\overline{AB}}}$$

Mạch kết quả:



2.4.3. Tính đa chức năng của cổng NAND, NOR (3)

- Từ cổng NOR có thể tạo ra các cổng NOT, AND, OR và NAND.



2.4.3. Tính đa chức năng của cổng NAND, NOR (4)

- Cổng XNOR dùng toàn cổng NOR.

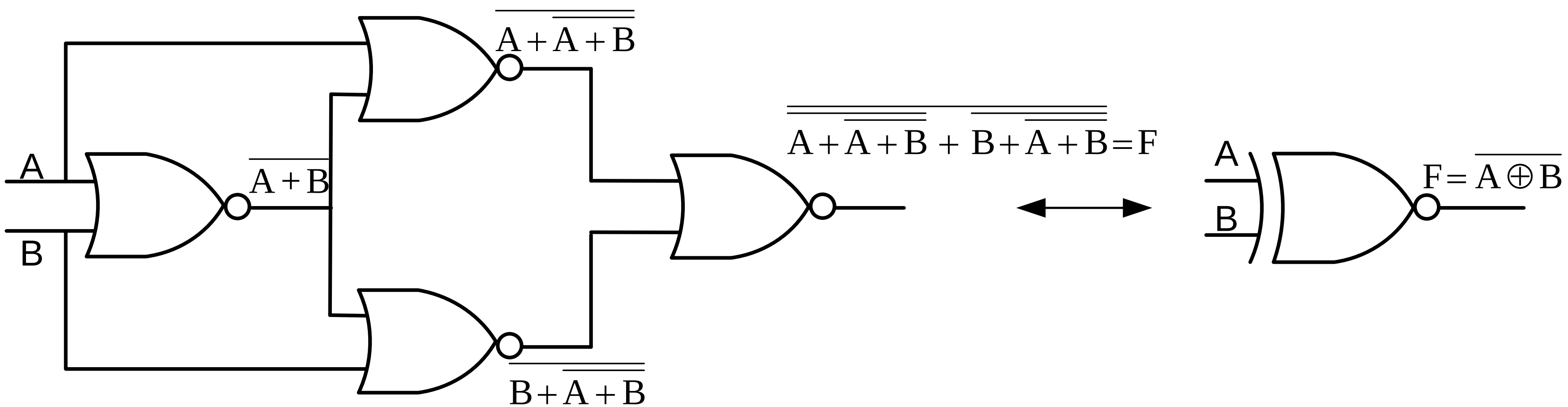
Hàm cổng XNOR: $F = AB + \bar{A}\,\bar{B}$

Biến đổi:

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

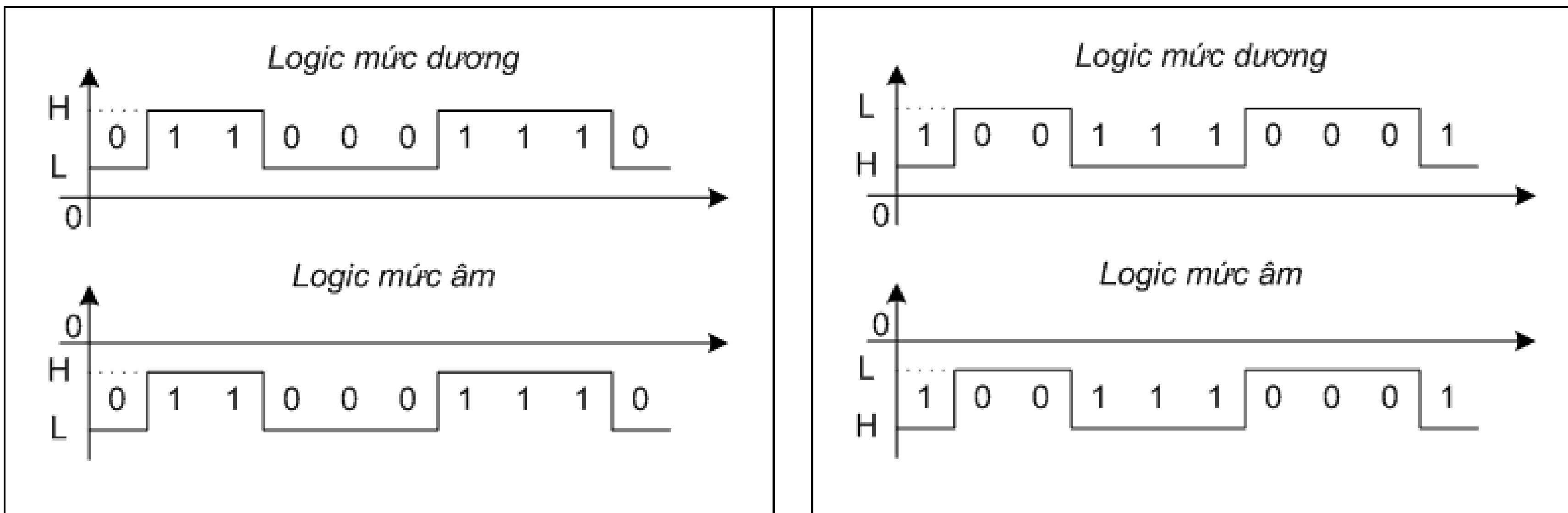
$$= \overline{\overline{\overline{\bar{A}(A+B)}} + \overline{\overline{\bar{B}(A+B)}}} = \overline{\overline{\bar{\bar{A}} + \overline{(A+B)}} + \overline{\bar{\bar{B}} + \overline{(A+B)}}} = \overline{\overline{A + \overline{(A+B)}} + \overline{B + \overline{(A+B)}}}$$

Mạch kết quả:



2.4.4. *Logic dương và logic âm*

- *Logic dương* là logic có điện thế mức cao H luôn lớn hơn điện thế mức thấp L ($V_H > V_L$).
- *Logic âm* là đảo của logic dương ($V_H < V_L$).
  - Khái niệm logic âm thường được dùng để biểu diễn trị các biến.
  - Logic âm và mức âm của logic là hoàn toàn khác nhau.



a) Logic dương | b) Logic âm

Chương 2 – HÀM BOOLE VÀ CỔNG LOGIC

2.1. Đại số Boole

2.2. Phương pháp biểu diễn hàm Boole

2.3. Phương pháp tối thiểu hàm Boole

2.4. Cổng logic

2.5. Tham số chính

2.5. Tham số chính

2.5.1. Mức logic

2.5.2. Độ chống nhiễu

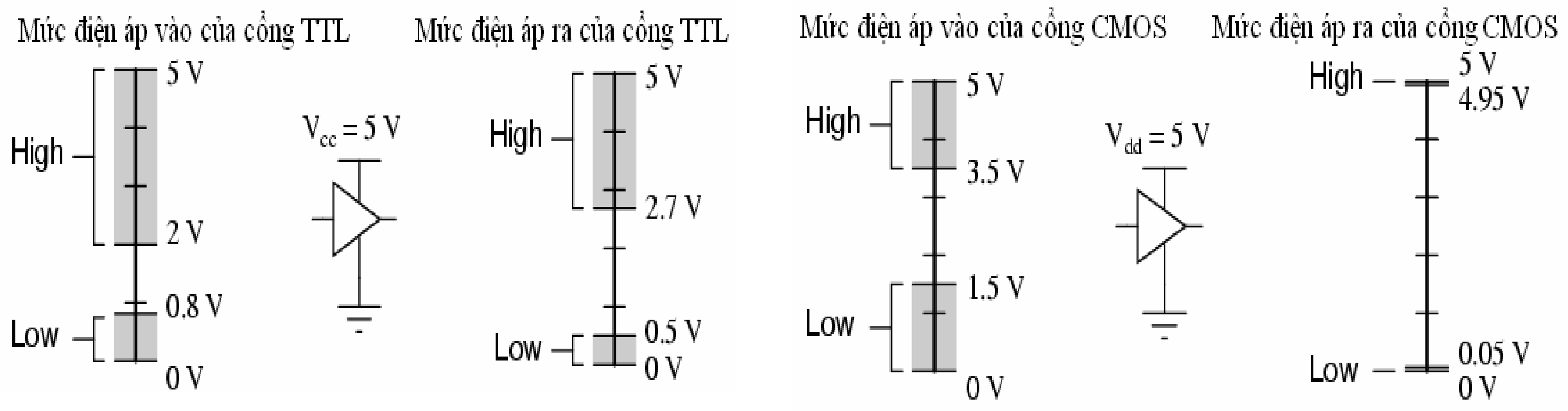
2.5.3. Hệ số ghép tải

2.5.4. Công suất tiêu thụ

2.5.5. Trễ truyền đạt

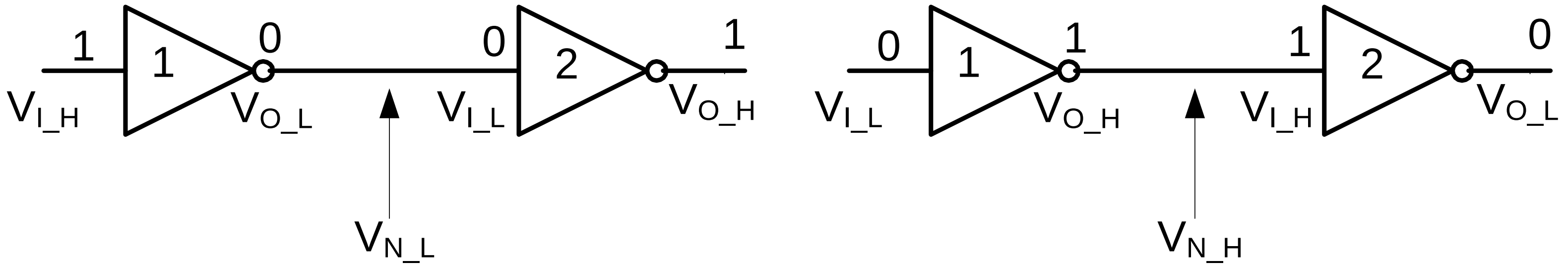
2.5.1. Mức logic

- *Mức logic:* là mức điện thế trên đầu vào và đầu ra của cổng tương ứng với logic "1" và logic "0“
- Mức logic phụ thuộc điện thế nguồn nuôi của cổng



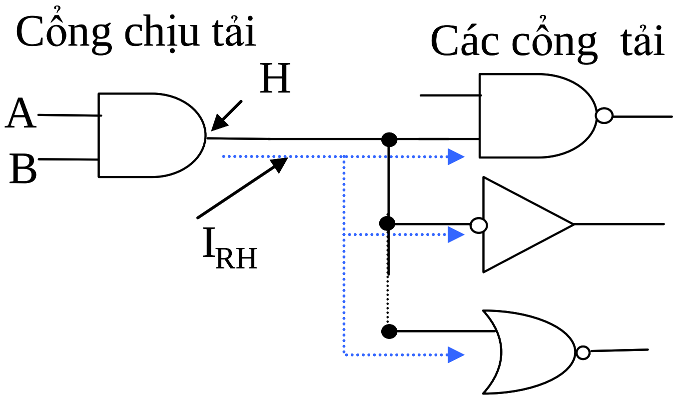
2.5.2. Độ chống nhiễu

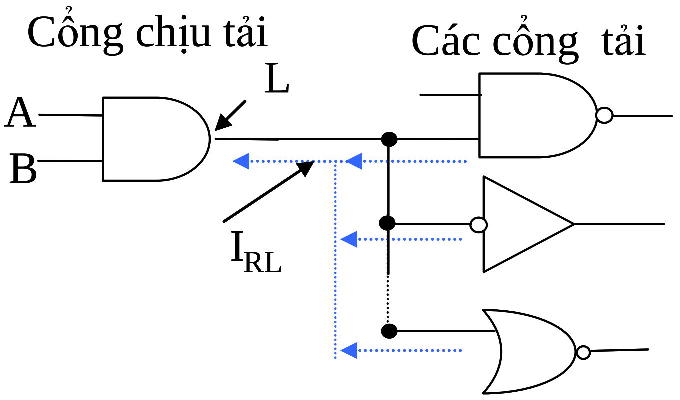
- *Độ chống nhiễu* (hay độ phòng vệ nhiễu) là mức nhiễu lớn nhất tác động tới đầu vào hoặc đầu ra của cổng mà chưa làm thay đổi trạng thái vốn có của nó.
- Là tiêu chuẩn đánh giá độ nhạy của mạch logic đối với tạp âm xung trên đầu vào vi mạch.
- Có thể chia thành hai trường hợp: nhiễu mức cao và nhiễu mức thấp.



2.5.3. Hệ số ghép tải

- *Hệ số mắc tải* cho biết khả năng nối được bao nhiêu đầu vào tới đầu ra của một cổng đã cho mà vẫn đảm bảo sự hoạt động tin cậy, đảm bảo tốc độ, giới hạn về nhiệt độ và các tham số khác.
- Hệ số mắc tải phụ thuộc dòng ra (hay dòng phun) của cổng chịu tải và dòng vào (hay dòng hút) của các cổng tải ở cả hai trạng thái H, L.





2.5.4. Công suất tiêu thụ

- *Công suất tiêu tán:*

  - Đây là tiêu chuẩn để đánh giá lượng công suất tiêu thụ (tổn hao) trên các phần tử trong vi mạch. Công suất tiêu hao thường cỡ vài mW đối với một vi mạch số và là giá trị trung bình giữa công suất tiêu tán.khi đầu ra ở mức 0, 1 (Các công suất này thường khác nhau).

  - Công suất tiêu tán càng nhỏ càng tốt và có ý nghĩa đặc biệt quan trọng trong các thiết bị xách tay hay các thiết bị dùng pin.

- *Công suất điều khiển:* là công suất của tín hiệu điều khiển ở đầu vào sao cho mạch vẫn hoạt động tốt. Công suất điều khiển càng nhỏ càng tốt.

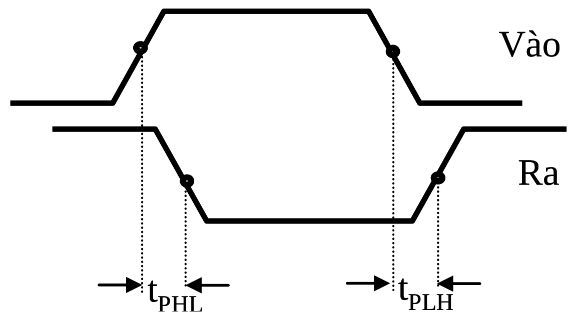
2.5.5. Trễ truyền đạt

- Trễ truyền đạt là khoảng thời gian để đầu ra của mạch có đáp ứng khi có sự thay đổi mức logic của đầu vào.

- Trễ truyền đạt là tiêu chuẩn để đánh giá tốc độ làm việc của mạch. Trễ truyền đạt càng nhỏ thì càng tốt tương ứng với tốc độ làm việc càng lớn càng tốt.

- Trễ truyền đạt thường được tính toán ở điểm 50% biên độ trên các sườn trước và sườn sau tương ứng giữa xung vào và xung ra.

- Trễ truyền đạt trung bình được tính theo công thức: $t_{pd} = \frac{t_{PHL} + t_{PLH}}{2}$



Một số lưu ý khi sử dụng IC số

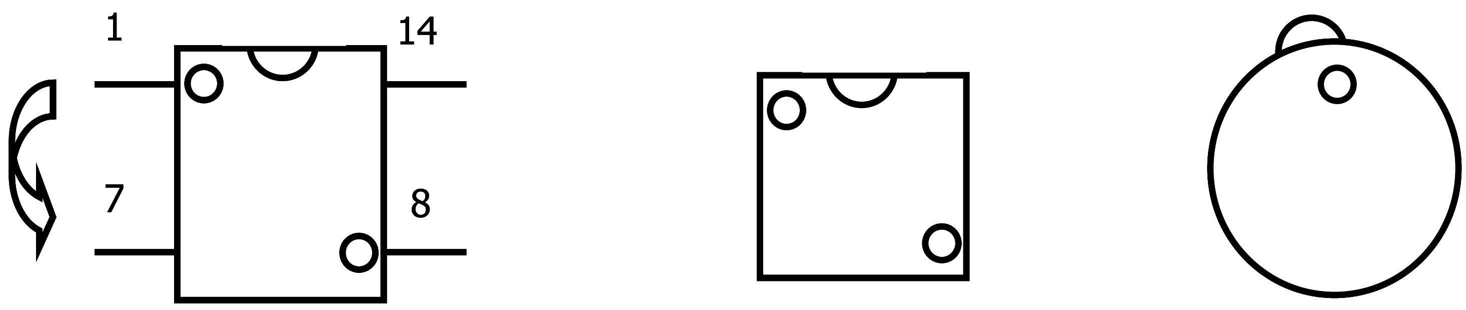
- **Ký hiệu vỏ IC**

TTL	CMOS
. . 74 . . x x x . : mục đích thương mại **. . 54 . . x x x . : mục đích quân sự**	**. . 14 . . x x x .** **. . 4 . . x x x .**

- Mỗi dấu chấm (.) thay cho một chữ cái:
  - Hai chữ cái đầu: tên hãng sản xuất
  - Hai chữ cái giữa: đặc điểm cấu trúc và tính năng
  - Chữ cái cuối: kết cấu vỏ
- Các dấu ‘x’: là tập hợp số từ 0 đến 999, cho biết chức năng logic của IC.
- VD: SN 74 LS 00 J

Một số lưu ý khi sử dụng IC số

- Đóng vỏ IC: có 3 phương pháp: T05, đóng vỏ dạng hộp, DIP (hai hàng chân song song)

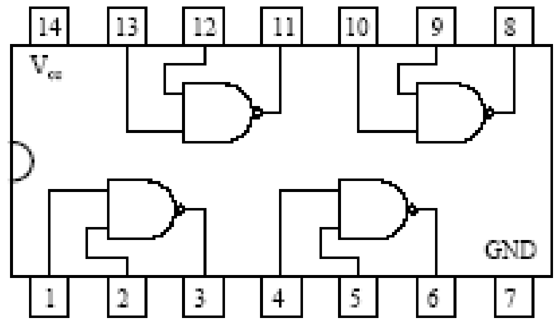


- DIP: phổ biến nhất, dễ lắp ráp và sử dụng. Thường gặp: SSI (8, 14, 16 chân), MSI (14, 16, 24 chân), LSI (24, 28, 40 chân).
- Loại IC phổ biến nhất là hình chữ nhật, hình vuông hoặc hình tròn.

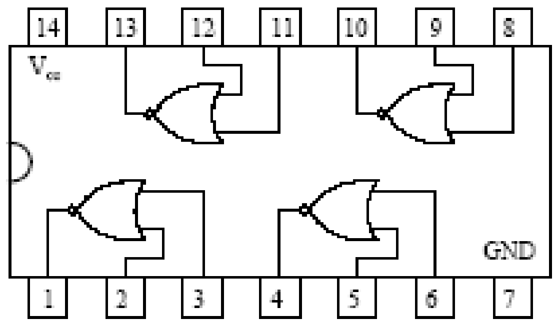
Một số lưu ý khi sử dụng IC số

➢ Sơ đồ chân một số IC TTL:

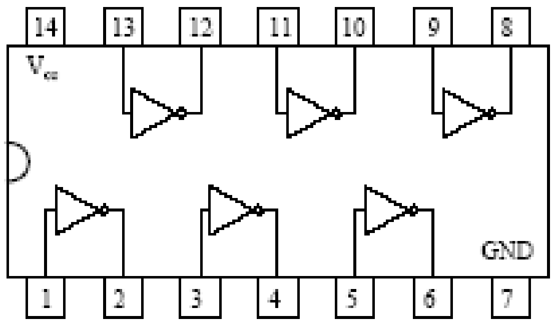
74LS00
NAND



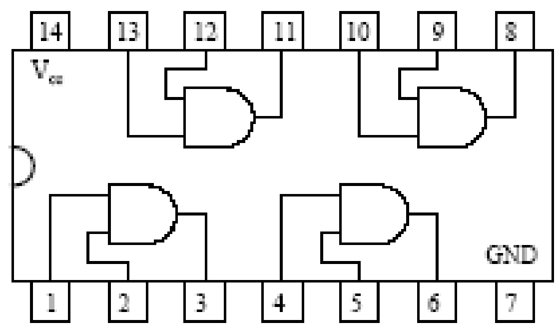
74LS02
NOR



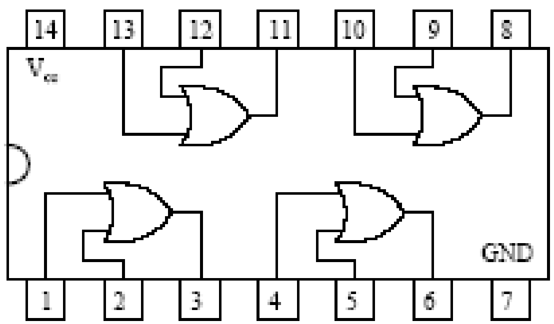
74LS04
NOT



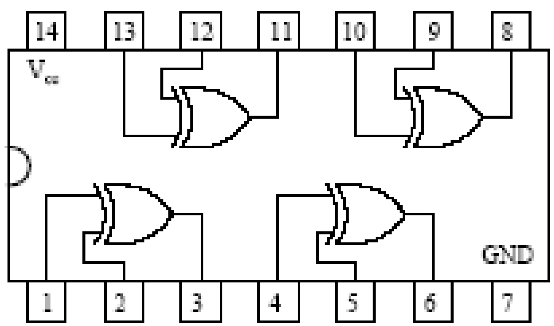
74LS08
AND



74LS32
OR



74LS86
XOR



Câu hỏi

- Làm các câu hỏi trong ngân hàng câu hỏi thi. Đầu buổi sau lên bảng làm bài tập.