

# Introduction to Digital Designs

Huy Vu

Department of Telecommunications Engineering  
Faculty of Electrical and Electronics Engineering  
Ho Chi Minh City University of Technology

Ngày 10 tháng 8 năm 2024

# Mục lục

<b>1</b>	<b>HỆ THỐNG SỐ ĐẾM - SỐ NHỊ PHÂN</b>	<b>3</b>
1.1	Các hệ thống số đếm: . . . . .	3
1.1.1	Khái niệm: . . . . .	3
1.1.2	Chuyển đổi cơ số: . . . . .	3
1.2	Số nhị phân (Binary): . . . . .	4
1.2.1	Các tính chất của số nhị phân . . . . .	4
1.2.2	Các phép toán số học trên số nhị phân: . . . . .	5
1.2.3	Mã nhị phân: . . . . .	6
1.3	Số nhị phân có dấu: . . . . .	7
1.3.1	Biểu diễn số có dấu: . . . . .	7
1.3.2	Số bù 1 (1's Complement): . . . . .	7
1.3.3	Số bù 2 (2's Complement): . . . . .	8
1.3.4	Các phép toán cộng trừ số có dấu: . . . . .	9
1.4	Cộng trừ số BCD: . . . . .	10
<b>2</b>	<b>ĐẠI SỐ BOOLE - CỔNG LOGIC</b>	<b>11</b>
2.1	Cấu trúc đại số Boole: . . . . .	11
2.1.1	Các tiên đề (Axioms): . . . . .	11
2.1.2	Các định lý cơ bản (Basic Theorems): . . . . .	11
2.2	Hàm Boole (Boolean Function): . . . . .	12
2.2.1	Định nghĩa: . . . . .	12
2.2.2	Bù của 1 hàm: . . . . .	12
2.3	Dạng chính tắc và dạng chuẩn của hàm Boole: . . . . .	12
2.3.1	Các tích chuẩn (minterm) và tổng chuẩn (Maxterm): . . . . .	12
2.3.2	Dạng chính tắc (Canonical Form): . . . . .	13
2.3.3	Dạng chuẩn (Standard Form): . . . . .	13
2.4	Cổng logic . . . . .	14
2.4.1	Cổng NOT: . . . . .	14
2.4.2	Cổng AND: . . . . .	14
2.4.3	Cổng OR: . . . . .	15
2.4.4	Cổng NAND: . . . . .	15
2.4.5	Cổng NOR: . . . . .	15
2.4.6	Cổng XOR: . . . . .	16
2.4.7	Cổng XNOR: . . . . .	16
2.5	Rút gọn hàm Boole: . . . . .	16
2.5.1	Phương pháp đại số: . . . . .	17
2.5.2	Phương pháp bìa KARNAUGH: . . . . .	17
2.6	Thực hiện hàm Boole bằng cổng logic: . . . . .	21
2.6.1	Cấu trúc cổng AND_OR: . . . . .	21
2.6.2	Cấu trúc cổng OR_AND: . . . . .	21
2.6.3	Cấu trúc cổng AND_OR_INVERTER (AOI): . . . . .	22
2.6.4	Cấu trúc cổng OR_AND_INVERTER (OAI): . . . . .	22
2.6.5	Cấu trúc toàn cổng NAND: . . . . .	22
2.6.6	Cấu trúc toàn cổng NOR: . . . . .	23

<b>3</b>	<b>HỆ TỔ HỢP</b>	<b>25</b>
3.1	Giới thiệu - Cách thiết kế hệ tổ hợp:	25
3.2	Bộ cộng - trừ nhị phân:	25
3.2.1	Bộ cộng (Adder):	25
3.2.2	Bộ trừ (Subtractor):	25
3.2.3	Bộ cộng/trừ nhị phân song song:	25
3.3	Hệ chuyển mã (Code conversion):	25
3.4	Bộ giải mã (Decoder):	25
3.4.1	Giới thiệu:	25
3.4.2	IC giải mã:	25
3.4.3	Sử dụng bộ giải mã thực hiện hàm Boole:	25
3.5	Bộ mã hóa (Encoder):	25
3.5.1	Giới thiệu:	25
3.5.2	IC mã hóa ưu tiên 8 $\rightarrow$ 3 (74148):	25
3.6	Bộ dồn kênh (Multiplexer - MUX):	25
3.6.1	Giới thiệu:	25
3.6.2	IC dồn kênh:	25
3.6.3	Sử dụng bộ MUX thực hiện hàm Boole:	25
3.7	Bộ ohaan kênh (Demux):	25
3.7.1	Giới thiệu:	25
3.7.2	IC phân kênh 74LS155:	25
3.8	Bộ so sánh độ lớn (Comparator):	25
3.8.1	Giới thiệu:	25
3.8.2	IC so sánh 74LS85:	25

# 1 HỆ THỐNG SỐ ĐẾM - SỐ NHỊ PHÂN

## 1.1 Các hệ thống số đếm:

### 1.1.1 Khái niệm:

- Cơ số (r-radix): là số lượng ký tự chữ số (ký số - digit) sử dụng để biểu diễn trong hệ thống số đếm.
- Trọng số (weight): đại lượng biểu diễn cho vị trí của 1 con số trong chuỗi số.  

$$\text{Trọng số} = \text{Cơ số}^{\text{vị trí}}$$
- Giá trị (value): tính bằng tổng theo trọng số.  $\text{Giá trị} = \sum (\text{Ký số} \times \text{Trọng số})$ .

a. Số thập phân (Decimal): Cơ số  $r = 10$ .

4	0	7	.	6	2	5
$10^2$	$10^1$	$10^0$	.	$10^{-1}$	$10^{-2}$	$10^{-3}$
$4 \times 10^2$	$0 \times 10^1$	$7 \times 10^0$	.	$6 \times 10^{-1}$	$2 \times 10^{-2}$	$5 \times 10^{-3}$
400	0	7	.	0.6	0.02	0.005

$$400 + 0 + 7 + 0.6 + 0.02 + 0.005 = 407.625$$

b. Số nhị phân (Binary): Cơ số  $r = 2$ .

1	0	1	.	0	1	1
$2^2$	$2^1$	$2^0$	.	$2^{-1}$	$2^{-2}$	$2^{-3}$
$1 \times 2^2$	$0 \times 2^1$	$1 \times 2^0$	.	$0 \times 2^{-1}$	$1 \times 2^{-2}$	$1 \times 2^{-3}$
4	0	1	.	0	0.25	0.125

$$4 + 0 + 1 + 0 + 0.25 + 0.125 = 5.375$$

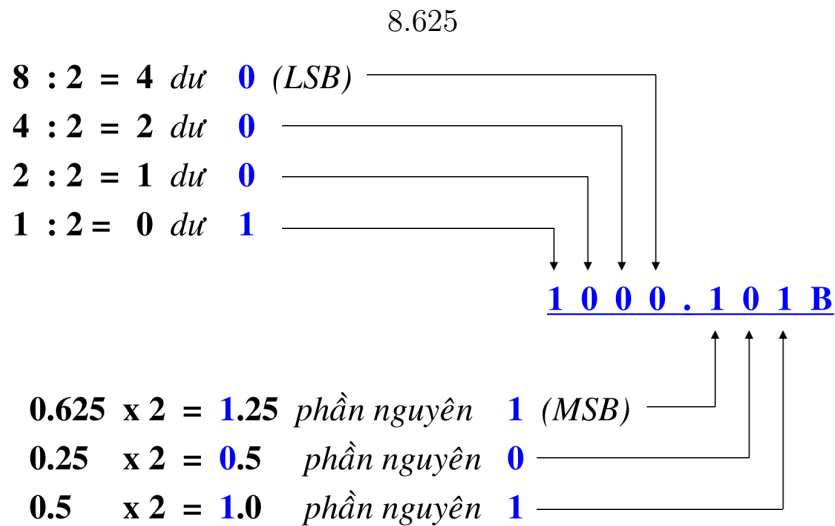
c. Số thập lục phân (Hexadecimal): Cơ số  $r = 16$

5	A	0	.	4	D	1
$16^2$	$16^1$	$16^0$	.	$16^{-1}$	$16^{-2}$	$16^{-3}$
$5 \times 16^2$	$10 \times 16^1$	$0 \times 16^0$	.	$4 \times 16^{-1}$	$13 \times 16^{-2}$	$1 \times 16^{-3}$
1280	160	0	.	0.25	0.0508	0.002

$$1280 + 160 + 0 + 0.25 + 0.0508 + 0.0002 = 1440.301$$

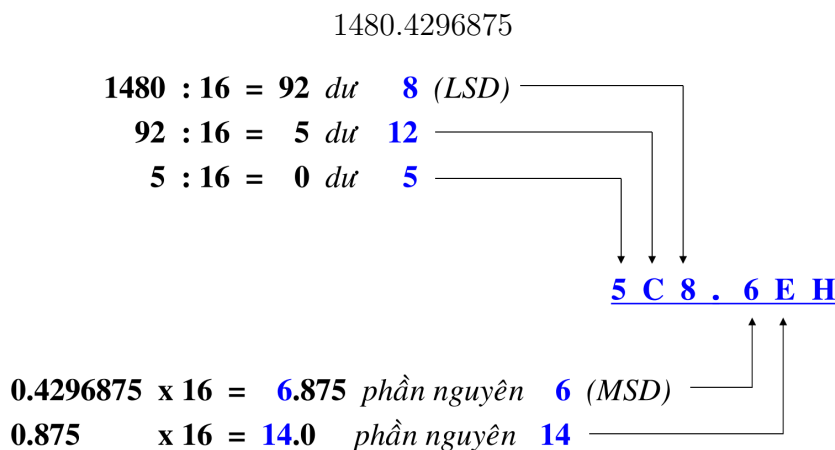
### 1.1.2 Chuyển đổi cơ số:

a. Từ thập phân sang nhị phân:

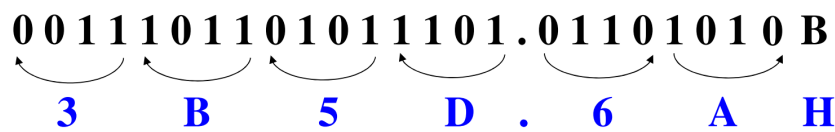


Ta lấy phần nguyên chia cho 2 và dừng khi ta được 0 dư 1, phần không nguyên nhân với cơ số 2 và giữ phần nguyên, tiếp tục cho tới khi có phần không nguyên là 0.

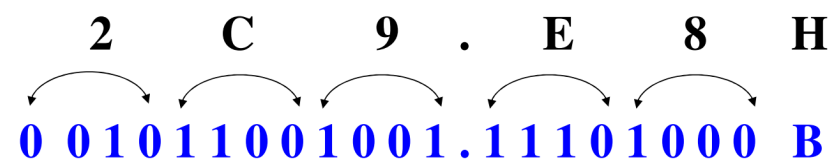
b. Từ thập phân sang thập lục phân:



c. Từ nhị phân sang thập lục phân



d. Từ thập lục phân sang nhị phân



## 1.2 Số nhị phân (Binary):

### 1.2.1 Các tính chất của số nhị phân

- Số nhị phân  $n$  bit có  $2^n$  giá trị từ 0 đến  $2^n - 1$ .

- Số nhị phân có giá trị  $2^n - 1$ :  $1 \dots 1$  (n bit 1) và giá trị  $2^n$ :  $1 0 \dots 0$  (n bit 0).
- Số nhị phân có giá trị lẻ là số có LSB = 1; ngược lại giá trị chẵn là số có LSB = 0.
- Các bội số của bit:

$$1 \text{ B (Byte)} = 8 \text{ bit}$$

$$1 \text{ KB} = 2^{10} \text{ B}$$

$$1 \text{ MB} = 2^{10} \text{ KB}$$

$$1 \text{ GB} = 2^{10} \text{ MB}$$

### 1.2.2 Các phép toán số học trên số nhị phân:

#### a. Phép cộng

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 0 \text{ nhớ } 1$

$$\begin{array}{r}
 \phantom{000}1 \phantom{00}1 \phantom{00}1 \\
 + \phantom{00}1 \phantom{00}0 \phantom{00}1 \phantom{00}1 \phantom{00}1 \\
 \hline
 1 \phantom{00}1 \phantom{00}1 \phantom{00}0 \phantom{00}0
 \end{array}$$

#### b. Phép trừ

$0 - 0 = 0$
$0 - 1 = 1 \text{ mượn } 1$
$1 - 0 = 1$
$1 - 1 = 0$

$$\begin{array}{r}
 \phantom{000}1 \phantom{00}1 \phantom{00}0 \phantom{00}1 \phantom{00}0 \\
 - \phantom{00}1 \phantom{00}1 \phantom{00}1 \phantom{00}1 \\
 \hline
 1 \phantom{00}0 \phantom{00}0 \phantom{00}1 \phantom{00}1
 \end{array}$$

#### c. Phép nhân

$$\begin{array}{r}
 \phantom{000}1 \phantom{00}0 \phantom{00}1 \phantom{00}1 \\
 \times \phantom{00}1 \phantom{00}0 \phantom{00}0 \phantom{00}1 \\
 \hline
 \phantom{000}1 \phantom{00}0 \phantom{00}1 \phantom{00}1 \\
 + \phantom{00}0 \phantom{00}0 \phantom{00}0 \phantom{00}0 \\
 + \phantom{00}0 \phantom{00}0 \phantom{00}0 \phantom{00}0 \\
 + \phantom{00}1 \phantom{00}0 \phantom{00}1 \phantom{00}1 \\
 \hline
 1 \phantom{00}1 \phantom{00}0 \phantom{00}0 \phantom{00}0 \phantom{00}1 \phantom{00}1
 \end{array}$$

#### d. Phép chia

$$\begin{array}{r}
 1 \phantom{00}0 \phantom{00}0 \phantom{00}1 \phantom{00}0 \phantom{00}0 \phantom{00}0 \phantom{00}1 \\
 - 1 \phantom{00}0 \phantom{00}1 \phantom{00}1 \\
 \hline
 1 \phantom{00}1 \phantom{00}1 \phantom{00}0 \\
 - 1 \phantom{00}0 \phantom{00}1 \phantom{00}1 \\
 \hline
 1 \phantom{00}1 \phantom{00}0 \phantom{00}1 \\
 - 1 \phantom{00}0 \phantom{00}1 \phantom{00}1 \\
 \hline
 1 \phantom{00}0
 \end{array}
 \quad \Bigg| \quad
 \begin{array}{r}
 1 \phantom{00}0 \phantom{00}1 \phantom{00}1 \\
 \hline
 1 \phantom{00}1 \phantom{00}0 \phantom{00}1
 \end{array}$$

### 1.2.3 Mã nhị phân:

**Từ mã:** là các tổ hợp nhị phân được sử dụng trong loại mã nhị phân.

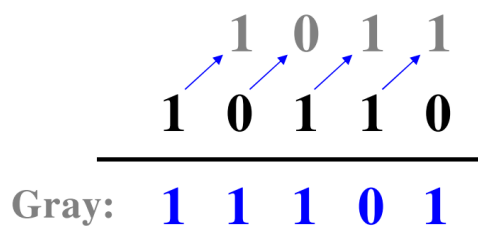
#### a. Mã nhị phân cho số thập phân (BCD - Binary Coded Decimal)

Số thập phân	BCD (8 4 2 1)	BCD (2 4 2 1)	BCD quá 3	Mã 1 trong 10
0	0000	0000	0011	0000000001
1	0001	0001	0100	0000000010
2	0010	0010	0101	0000000100
3	0011	0011	0110	0000001000
4	0100	0100	0111	0000010000
5	0101	1011	1000	0000100000
6	0110	1100	1001	0001000000
7	0111	1101	1010	0010000000
8	1000	1110	1011	0100000000
9	1001	1111	1100	1000000000

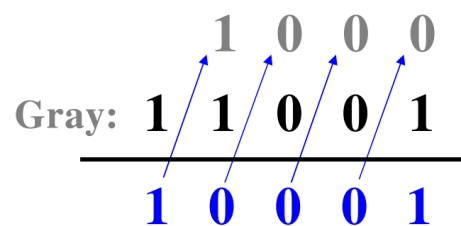
**b. Mã Gray:** là mã nhị phân mà 2 giá trị liên tiếp nhau có tổ hợp bit biểu diễn chỉ khác nhau 1 bit.

Giá trị	Binary	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110

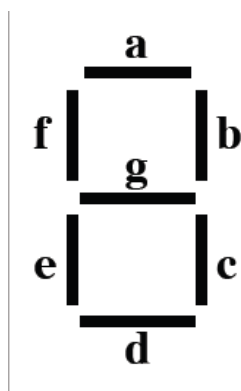
#### Đổi từ Binary sang Gray



#### Đổi từ Gray sang Binary



#### c. Mã LED 7 đoạn



Giá trị	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

**d. Mã 1 trong n:** là mã nhị phân n bit có mỗi từ mã chỉ có 1 bit là 1 (hoặc 0) và n-1 bit còn lại là 0 (hoặc 1).

$$\text{Mã 1 trong 4:} \quad \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \quad \text{hoặc} \quad \begin{matrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{matrix}$$

**e. Mã ký tự ASCII:**

		(Cột) $b_6b_5b_4$							
(Hàng)		000	001	010	011	100	101	110	111
$b_3b_2b_1b_0$	Hex	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	'	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	)	8	H	X	h	x
1001	9	HT	EM	(	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	ES	,	<	L	\	l	
1101	D	CR	GS	-	=	M	]	m	}
1110	E	SO	RS	.	>	N	^	n	~
1111	F	SI	US	/	?	O	_	o	DEL

### 1.3 Số nhị phân có dấu:

#### 1.3.1 Biểu diễn số có dấu:

**Số có dấu theo biên độ (Signed Magnitude):**

- Bit MSB là bit dấu: 0 là số dương và 1 là số âm, các bit còn lại biểu diễn giá trị độ lớn.

$$+13 : \quad 01101$$

$$-13 : \quad 11101$$

- Phạm vi biểu diễn:

$$-(2^{n-1} - 1) \div +(2^{n-1} - 1)$$

#### 1.3.2 Số bù 1 (1's Complement):

- Số bù 1 của 1 số nhị phân N có chiều dài n bit

$$\text{Bù 1 (N)} = 2^n - 1 - N$$



VD: Bù 1 (1001) =  $2^4 - 1 - 1001 = 1111 - 1001 = 0110$

- Có thể lấy Bù 1 của 1 số nhị phân bằng cách lấy đảo từng bit của nó (0 thành 1 và 1 thành 0).

### Biểu diễn số có dấu bù 1:

- \* Số có giá trị dương: bit dấu = 0, các bit còn lại biểu diễn độ lớn.
- \* Số có giá trị âm: lấy bù 1 của số dương có cùng độ lớn.

Phạm vi biểu diễn:

$$-(2^{n-1} - 1) \div +(2^{n-1} - 1)$$

### 1.3.3 Số bù 2 (2's Complement):

- Số bù 2 của 1 số nhị phân N có chiều dài n bit cũng có n bit

$$\text{Bù 2 (N)} = 2^n - N = \text{Bù 1 (N)} + 1$$

VD: Bù 2 (1001) =  $2^4 - 1001 = 10000 - 1001 = 0111$ . Hoặc Bù 2(1001) = Bù 1 (1001) + 1 = 0110 + 1 = 0111.

### Biểu diễn số có dấu bù 2:

- \* Số có giá trị dương: bit dấu = 0, các bit còn lại biểu diễn độ lớn.
- \* Số có giá trị âm: lấy bù 2 của số dương có cùng độ lớn.

Phạm vi biểu diễn:

$$-(2^{n-1}) \div +(2^{n-1} - 1)$$

Giá trị dương	Giá trị âm
000 = 0	100 = -4
001 = +1	101 = -3
010 = +2	110 = -2
011 = +3	111 = -1

- Để tìm được giá trị của số âm: ta lấy bù 2 của nó; sẽ nhận được số dương có cùng biên độ.
- Mở rộng chiều dài bit số có dấu: số dương thêm các bit 0 và số âm thêm các bit 1 vào trước.
- Lấy bù 2 hai lần một số thì bằng chính số đó.
- Giá trị -1 được biểu diễn là 1/cdots11 (n bit 1).
- Giá trị  $-2^n$  được biểu diễn là 100/cdots00 (n bit 0).

$$-32 = -2^5 = 100000$$

### 1.3.4 Các phép toán cộng trừ số có dấu:

- Thực hiện giống như số không dấu.

Thực hiện trên toán hạng có cùng chiều dài bit, và kết quả cũng có cùng số bit.

Kết quả đúng nếu nằm trong phạm vi biểu diễn số có dấu (nếu kết quả sai thì cần mở rộng chiều dài bit).

$$\begin{array}{rcl}
 + \begin{array}{r} -6 : 1010 \\ +3 : 0011 \\ \hline -3 : 1101 \end{array} & & + \begin{array}{r} -2 : 1110 \\ -5 : 1011 \\ \hline -7 : 1001 \end{array} \\
 \\
 + \begin{array}{r} +4 : 0100 \longrightarrow 00100 \\ +5 : 0101 \longrightarrow 00101 \\ \hline -7 : 1001 \text{ (Kq sai)} \end{array} & & \begin{array}{r} 01001 : +9 \text{ (Kq đúng)} \\ \hline \end{array} \\
 \\
 - \begin{array}{r} -6 : 1010 \\ -2 : 1110 \\ \hline -4 : 1100 \end{array} & & - \begin{array}{r} +2 : 0010 \\ -5 : 1011 \\ \hline +7 : 0111 \end{array} \\
 \\
 - \begin{array}{r} -7 : 1001 \longrightarrow 11001 \\ +5 : 0101 \longrightarrow 00101 \\ \hline +4 : 0100 \text{ (Kq sai)} \end{array} & & \begin{array}{r} 10100 : -12 \text{ (Kq đúng)} \\ \hline \end{array}
 \end{array}$$

Trừ với số bù 2:  $A - B = A + \text{Bù 2 (B)}$

- Trừ với số không có dấu:

$$\begin{array}{rcl}
 - \begin{array}{r} 6 : 0110 \longrightarrow \\ 13 : 1101 \xrightarrow{\text{bù}_2} \\ \hline -7 : \end{array} & & + \begin{array}{r} 0110 \\ 0011 \\ \hline 1001 \end{array}
 \end{array}$$

- Trừ với số có dấu:

$$\begin{array}{rcl}
 - \begin{array}{r} -6 : 1010 \longrightarrow \\ -3 : 1101 \xrightarrow{\text{bù}_2} \\ \hline -3 : \end{array} & & + \begin{array}{r} 1010 \\ 0011 \\ \hline 1101 \end{array}
 \end{array}$$

## 1.4 Cộng trừ số BCD:

$A + B$	$S = A + B$		Nếu tổng $S_i \geq 10$ hoặc có bit nhớ $C_i = 1$ , thì hiệu chỉnh $S_i$ : $S_i = S_i + 6$ và $S_{i+1} = S_{i+1} + C_i$
$A - B$	$D = A - B$ $= A +$ $B_{\text{bù}_2(B)}$ (Kết quả bỏ bit $C_n$ )	$C_n = 1$ : kết quả là số dương ( $A \geq B$ )	Nếu $C_i = 1$ thì không hiệu chỉnh
		$C_n = 0$ : kết quả là số âm ( $A < B$ ) Lấy bù kết quả	Nếu $C_i = 0$ thì hiệu chỉnh $D_i$ : $D_i = D_i + 10$
			Nếu $C_i = 1$ thì hiệu chỉnh $D_i$ : $D_i = D_i + 6$
			Nếu $C_i = 0$ thì không hiệu chỉnh

$$\begin{array}{r}
 \begin{array}{r}
 29 : 0010 \ 1001 \\
 + 55 : 0101 \ 0101 \\
 \hline
 0111 \ 1110 \\
 0110 \\
 \hline
 84 : 1000 \ 0100
 \end{array}
 \qquad
 \begin{array}{r}
 \begin{array}{r}
 28 : 0010 \ 1000 \\
 + 19 : 0001 \ 1001 \\
 \hline
 0100 \ 0001 \\
 0110 \\
 \hline
 47 : 0100 \ 0111
 \end{array}
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 \begin{array}{r}
 29 : 0010 \ 1001 \\
 - 55 : 0101 \ 0101 \\
 \hline
 \end{array}
 \longrightarrow
 \begin{array}{r}
 \begin{array}{r}
 0010 \ 1001 \\
 + 1010 \ 1011 \\
 \hline
 1101 \ 0100 \\
 0110 \\
 \hline
 -26 : 1101 \ 1010 \\
 0010 \ 0110
 \end{array}
 \end{array}
 \end{array}$$

## 2 ĐẠI SỐ BOOLE - CỔNG LOGIC

### 2.1 Cấu trúc đại số Boole:

Là cấu trúc đại số được định nghĩa trên tập 1 phần tử nhị phân  $B = \{0, 1\}$  và các phép toán nhị phân: AND(.), OR(+), NOT(').

x	y	x.y (x AND y)
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x+y (x OR y)
0	0	0
0	1	1
1	0	1
1	1	1

x	x' (NOT x, $\bar{x}$ )
0	1
1	0

**\*Thứ tự phép toán:** theo thứ tự dấu ngoặc (), NOT, AND, OR.

#### 2.1.1 Các tiên đề (Axioms):

- Tính kín (Closure Property)
- Phần tử đồng nhất (Identity Element):

$$x.1 = 1.x = x$$

$$x + 0 = 0 + x = x$$

- Tính giao hoán (Commutative Property):

$$x.y = y.x$$

$$x + y = y + x$$

- Tính phân bố (Complement Element):

$$x.(y + z) = x.y + x.z$$

$$x + (y.z) = (x + y)(x + z)$$

- Phần tử bù (Complement Element):

$$x + \bar{x} = 1 \quad x.\bar{x} = 0$$

#### 2.1.2 Các định lý cơ bản (Basic Theorems):

- Định lý 1:  $\bar{\bar{x}} = x$

- Định lý 2:  $x + x = x \quad x.x = x$

- Định lý 3:  $x + 1 = 1 \quad x.0 = 0$

- Định lý 4: Định lý hấp thụ (Absorption)

$$x + x.y = x \quad x.(x + y) = x$$

- Định lý 5: Định lý kết hợp (Associative)

$$x + (y + z) = (x + y) + z \quad x.(y.z) = (x.y).z$$

- Định lý 6: Định lý De Morgan

$$\overline{x + y} = \bar{x}.\bar{y} \quad \overline{x.y} = \bar{x} + \bar{y}$$

## 2.2 Hàm Boole (Boolean Function):

### 2.2.1 Định nghĩa:

- Hàm Boole là 1 biểu thức được tạo bởi các biến nhị phân và các phép toán nhị phân NOT, AND, OR.

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

- Với giá trị cho trước của các biến, hàm Boole sẽ có giá trị là 0 hoặc 1.
- Bảng giá trị:

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

### 2.2.2 Bù của 1 hàm:

- Sử dụng định lý De Morgan:

$$F = x.y + \bar{x}.\bar{y}.z$$

$$\bar{F} = \overline{x.y + \bar{x}.\bar{y}.z} = \overline{(x.y)} \cdot \overline{(\bar{x}.\bar{y}.z)}$$

$$\bar{F} = (\bar{x} + \bar{y}) \cdot (x + y + \bar{z})$$

- Lấy biểu thức đối ngẫu và lấy bù các biến: Tính đối ngẫu (Duality): Hai biểu thức được gọi là đối ngẫu của nhau khi thay phép toán AND bằng OR, phép toán OR bằng AND, 0 thành 1 và 1 thành 0.

$$F = x.y + \bar{x}.\bar{y}.z$$

Lấy đối ngẫu:  $(x.y)(\bar{x}.\bar{y}.z)$

Bù các biến:  $\bar{F} = (\bar{x} + \bar{y}) \cdot (x + y + \bar{z})$

## 2.3 Dạng chính tắc và dạng chuẩn của hàm Boole:

### 2.3.1 Các tích chuẩn (minterm) và tổng chuẩn (Maxterm):

- Tích chuẩn (minterm):  $m_i$  ( $0 \leq i \leq 2^n - 1$ ) là các số hạng tích (AND) của n biến mà hàm Boole phụ thuộc với quy ước biến đó có bù nếu nó là 0 và không bù nếu nó là 1.
- Tổng chuẩn (Maxterm):  $M_i$  ( $0 \leq i \leq 2^n - 1$ ) là các số hạng tổng (OR) của n biến mà hàm Boole phụ thuộc với quy ước biến đó có bù nếu nó là 1 và không bù nếu nó là 0.

x	y	z	<i>minterm</i>	<i>Maxterm</i>
0	0	0	$m_0 = \bar{x} \bar{y} \bar{z}$	$M_0 = x + y + z$
0	0	1	$m_1 = \bar{x} \bar{y} z$	$M_1 = x + y + \bar{z}$
0	1	0	$m_2 = \bar{x} y \bar{z}$	$M_2 = x + \bar{y} + z$
0	1	1	$m_3 = \bar{x} y z$	$M_3 = x + \bar{y} + \bar{z}$
1	0	0	$m_4 = x \bar{y} \bar{z}$	$M_4 = \bar{x} + y + z$
1	0	1	$m_5 = x \bar{y} z$	$M_5 = \bar{x} + y + \bar{z}$
1	1	0	$m_6 = x y \bar{z}$	$M_6 = \bar{x} + \bar{y} + z$
1	1	1	$m_7 = x y z$	$M_7 = \bar{x} + \bar{y} + \bar{z}$

$$m_i = \overline{M_i}$$

### 2.3.2 Dạng chính tắc (Canonical Form):

**a. Dạng chính tắc 1:** là dạng tổng của các tích chuẩn (minterm) làm cho hàm Boole có giá trị 1.

x	y	z	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	1

$$\begin{aligned}
 F(x, y, z) &= \bar{x} \bar{y} z + \bar{x} y \bar{z} + x \bar{y} z + xy \bar{z} + xyz \\
 &= m_1 + m_2 + m_5 + m_6 + m_7 \\
 &= \Sigma m(1, 2, 5, 6, 7) = \Sigma(1, 2, 5, 6, 7) \\
 F(x, y, z) &= (x + y + z)(x + \bar{y} + \bar{z})(\bar{x} + y + z) \\
 &= M_0 \cdot M_3 \cdot M_4 \\
 &= \Pi M(0, 3, 4) = \Pi(0, 3, 4)
 \end{aligned}$$

**b. Dạng chính tắc 2:** là dạng tích của các tổng chuẩn (Maxterm) làm cho hàm Boole có giá trị 0.

**TRƯỜNG HỢP HÀM BOOLE TÙY ĐỊNH (don't care):** Hàm Boole n biến có thể không được định nghĩa hết tất cả  $2^n$  tổ hợp của n biến phụ thuộc. Khi đó tại các tổ hợp không sử dụng này, hàm Boole sẽ nhận giá trị tùy định (don't care), nghĩa là hàm Boole có thể nhận giá trị 0 hoặc 1.

x	y	z	F
0	0	0	X
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	X

$$\begin{aligned}
 F(x, y, z) &= \Sigma(1, 2, 5, 6) + d(0, 7) \\
 &= \Pi(3, 4) \cdot D(0, 7)
 \end{aligned}$$

### 2.3.3 Dạng chuẩn (Standard Form):

**a. Dạng chuẩn 1:** là dạng tổng các tích (S.O.P - Sum of Product)

$$F(x, y, z) = xy + z$$

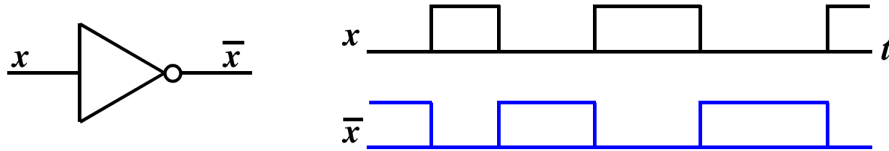
$$\begin{aligned}
F(x, y, z) &= xy + z \\
&= xy(\bar{z} + z) + (\bar{x} + x)(\bar{y} + y)z \\
&= xy\bar{z} + xyz + \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xyz \\
&= m_6 + m_7 + m_1 + m_5 + m_3 \\
&= \Sigma(1, 3, 5, 6, 7) \\
F(x, y, z) &= xy + z \\
&= (x + z)(y + z) \\
&= (x + \bar{y}y + z)(\bar{x}x + y + z) \\
&= (x + \bar{y} + z)(x + y + z)(\bar{x} + y + z)(x + y + z) \\
&= M_2 \cdot M_0 \cdot M_4 \\
&= \Pi(0, 2, 4)
\end{aligned}$$

**b. Dạng chuẩn 2:** là dạng tích các tổng (P.O.S - Product of Sum)

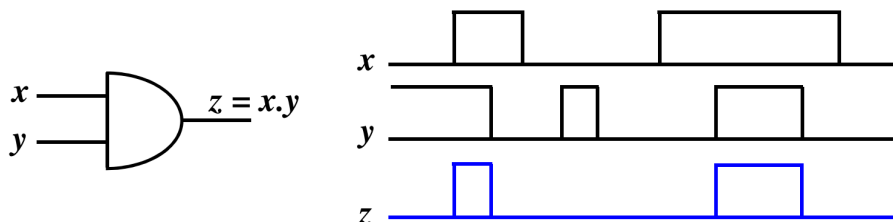
$$\begin{aligned}
F(x, y, z) &= (x + \bar{z})\bar{y} \\
F(x, y, z) &= (x + \bar{z})\bar{y} = x\bar{y} + \bar{y}\bar{z} \\
&= x\bar{y}(\bar{z} + z) + (\bar{x} + x)\bar{y}\bar{z} \\
&= x\bar{y}\bar{z} + x\bar{y}z + \bar{x}\bar{y}\bar{z} + x\bar{y}\bar{z} \\
&= m_4 + m_5 + m_0 \\
&= \Sigma(0, 4, 5) \\
F(x, y, z) &= (x + \bar{z})\bar{y} \\
&= (x + \bar{y}y + \bar{z})(\bar{x}x + \bar{y} + \bar{z}z) \\
&= (x + \bar{y} + \bar{z})(x + y + \bar{z}) \\
&= (\bar{x} + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)(x + \bar{y} + \bar{z})(x + \bar{y} + z) \\
&= M_3 \cdot M_1 \cdot M_7 \cdot M_6 \cdot M_2 \\
&= \Pi(1, 2, 3, 6, 7)
\end{aligned}$$

## 2.4 Cổng logic

### 2.4.1 Cổng NOT:



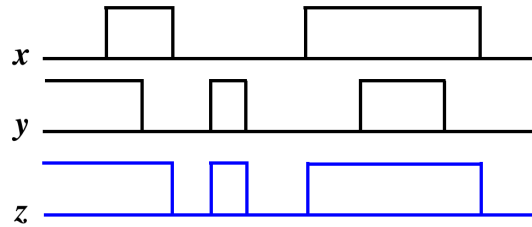
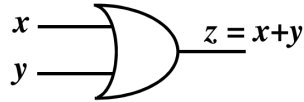
### 2.4.2 Cổng AND:



x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

Với cổng AND có nhiều ngõ vào,  
ngõ ra sẽ là 1 nếu tất cả các ngõ vào đều là 1

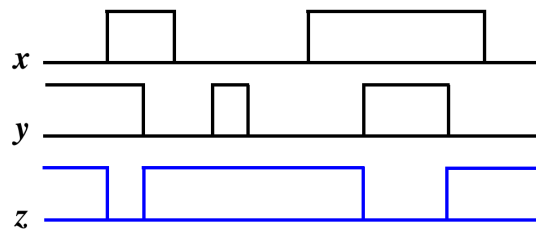
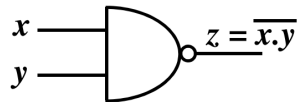
### 2.4.3 Cổng OR:



x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

Với cổng OR có nhiều ngõ vào,  
ngõ ra sẽ là 0 nếu tất cả các ngõ vào đều là 0

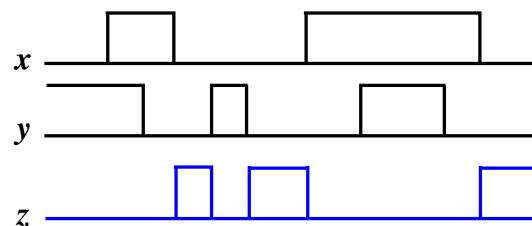
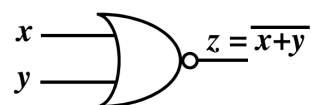
### 2.4.4 Cổng NAND:



x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

Với cổng NAND có nhiều ngõ vào,  
ngõ ra sẽ là 0 nếu tất cả các ngõ vào đều là 1

### 2.4.5 Cổng NOR:

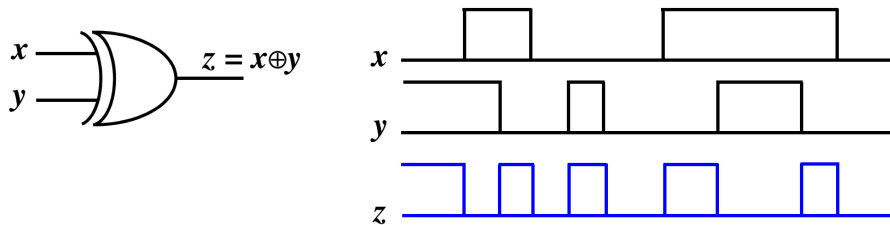




x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

Với cổng NOR có nhiều ngõ vào,  
ngõ ra sẽ là 1 nếu tất cả các ngõ vào đều là 0

#### 2.4.6 Cổng XOR:

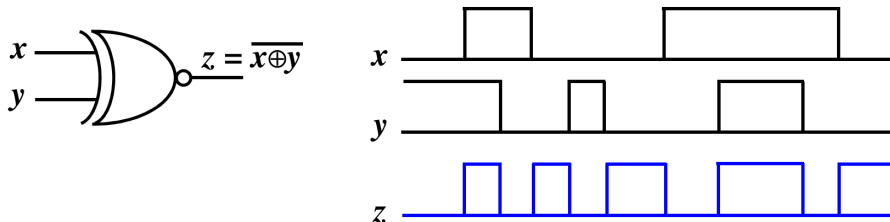


x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

Với cổng XOR có nhiều ngõ vào, ngõ ra sẽ là 1  
nếu tổng số bit 1 ở các ngõ vào là số lẻ

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

#### 2.4.7 Cổng XNOR:



x	y	z
0	0	1
0	1	0
1	0	0
1	1	1

Với cổng XNOR có nhiều ngõ vào, ngõ ra sẽ là 1  
nếu tổng số bit 1 ở các ngõ vào là số chẵn

$$z = \overline{x \oplus y} = \bar{x}\bar{y} + xy = (x + \bar{y})(\bar{x} + y)$$

### 2.5 Rút gọn hàm Boole:

Rút gọn (tối thiểu hóa) hàm Boole nghĩa là đưa hàm Boole về dạng biểu diễn đơn giản nhất, sao cho:

- Biểu thức có chứa ít nhất các thừa số và mỗi thừa số chứa ít nhất các biến.
- Mạch logic thực hiện có chứa ít nhất các vi mạch số.

### 2.5.1 Phương pháp đại số:

Dùng các định lý và tiên đề để rút gọn hàm.

$$\begin{aligned}
 F(A, B, C) &= \Sigma(2, 3, 5, 6, 7) \\
 &= \overline{A}\overline{B}\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + AB\overline{C} + ABC \\
 &= \overline{A}B(\overline{C} + C) + AC(\overline{B} + B) + AB(\overline{C} + C) \\
 &= \overline{A}B + AC + AB \\
 &= (\overline{A} + A)B + AC \\
 &= B + AC
 \end{aligned}$$

### 2.5.2 Phương pháp bìa KARNAUGH:

#### a. Cách biểu diễn:

- Bìa K gồm các ô vuông, mỗi ô vuông biểu diễn cho tổ hợp n biến. Như vậy bìa K cho n biến sẽ có  $2^n$  ô.
- Hai ô được gọi là kề nhau khi tổ hợp biến mà chúng biểu diễn chỉ khác nhau 1 biến.
- Trong ô sẽ ghi giá trị tương ứng của hàm Boole tại tổ hợp đó. Ở dạng chính tắc 1 thì đưa các giá trị 1 và X lên các ô, không đưa các giá trị 0. Ngược lại, dạng chính tắc 2 thì chỉ đưa giá trị 0 và X.

\* Bìa 2 biến:  $F(A, B) = \Sigma(0, 2) + d(3) = \Pi(1).D(3)$

F	A	0	1
B	0	0	2
	1	1	3

F	A	0	1
B	0	1	1
	1		X

F	A	0	1
B	0		
	1	0	X

\* Bìa 3 biến:

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

$$F(A, B, C) = \Sigma(2, 4, 7) + d(0, 1) = \Pi(3, 5, 6).D(0, 1)$$

F	AB	00	01	11	10
C	0	X	1		1
	1	X		1	

F	AB	00	01	11	10
C	0	X		0	
	1	X	0		0

\* Bìa 4 biến:

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

\* Bìa 5 biến:

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

b. Rút gọn bìa Karnaugh:

\* Nguyên tắc:

- Liên kết đôi: Khi liên kết (OR) hai ô có giá trị 1 ( $\bar{0}_1$ ) kề cận với nhau trên bìa K, ta sẽ được 1 số hạng tích mất đi 1 biến so với tích chuẩn (biến mất đi là biến khác nhau giữa 2 ô) Hoặc khi liên kết (AND) hai ô có giá trị 0 ( $\bar{0}_0$ ) kề cận với nhau trên bìa K, ta sẽ được 1 số hạng tổng mất đi 1 biến so với tổng chuẩn (biến mất đi là biến khác nhau giữa 2 ô).

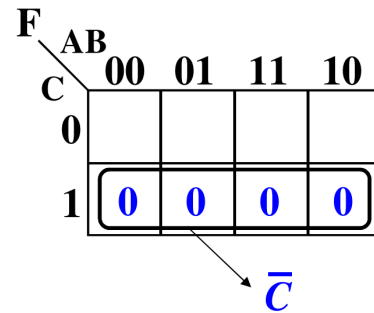
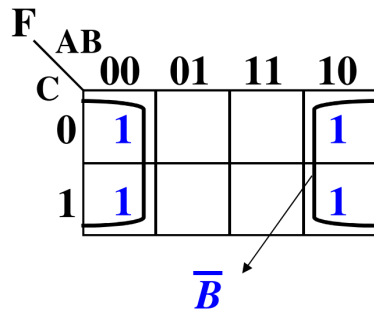
F \ AB	C			
	00	01	11	10
0		1	1	
1				

$B\bar{C}$

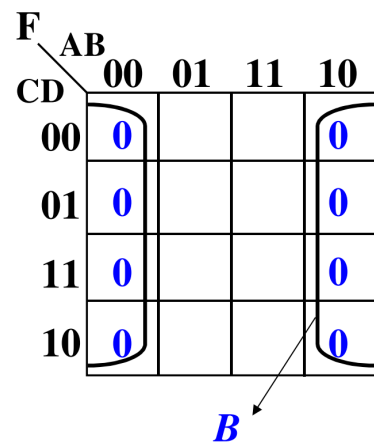
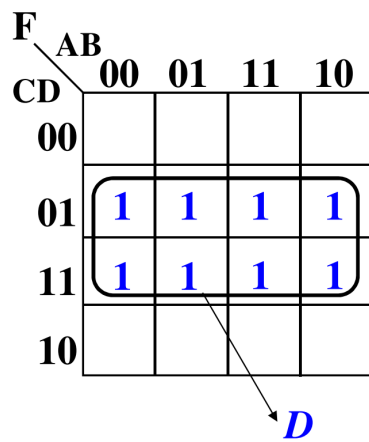
F \ AB	C			
	00	01	11	10
0				0
1				0

$\bar{A} + B$

- Liên kết 4: Tương tự như liên kết đôi khi liên kết 4  $\bar{O}_1$  hoặc 4  $\bar{O}_0$  kề cận với nhau, ta sẽ loại đi 2 biến (2 biến khác nhau giữa 4 ô).



- Liên kết 8: Liên kết 8 ô kề cận với nhau, ta sẽ loại đi được 3 biến (3 biến khác nhau giữa 8 ô).

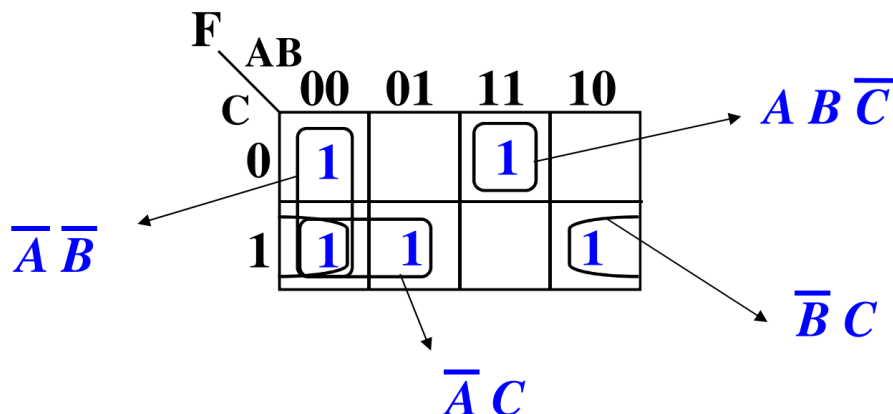


- Liên kết  $2^k$ : Khi ta liên kết  $2^k$   $\bar{O}_1$  hoặc  $2^k$   $\bar{O}_0$  kề cận với nhau ta sẽ loại đi được  $k$  biến ( $k$  biến khác nhau giữa  $2^k$  ô).

**\*Các bước thực hiện rút gọn theo dạng S.O.P:**

- Biểu diễn các  $\bar{O}_1$  lên bảng Karnaugh
- Thực hiện các liên kết có thể có sao cho các  $\bar{O}_1$  được liên kết ít nhất 1 lần; mỗi liên kết cho ta 1 số hạng tích. (Nếu  $\bar{O}_1$  không có kề cận với các  $\bar{O}_1$  khác thì ta có liên kết 1; số hạng tích chính bằng minterm của ô đó).
- Biểu thức rút gọn có được bằng cách lấy tổng (OR) của các số hạng tích liên kết trên.

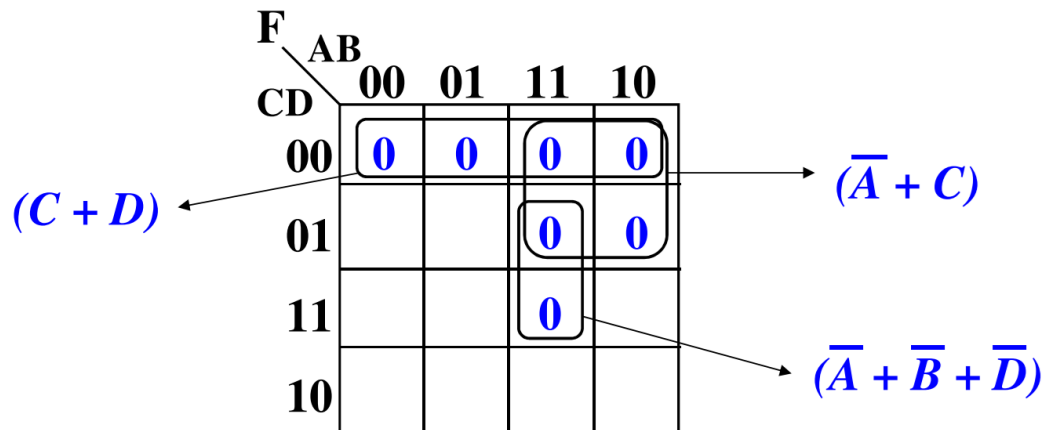
$$F(A, B, C, D) = \Sigma(0, 1, 3, 5, 6) = \bar{A} \bar{B} + \bar{A} C + \bar{B} C + A B \bar{C}$$



**\*Các bước thực hiện rút gọn theo dạng P.O.S:**

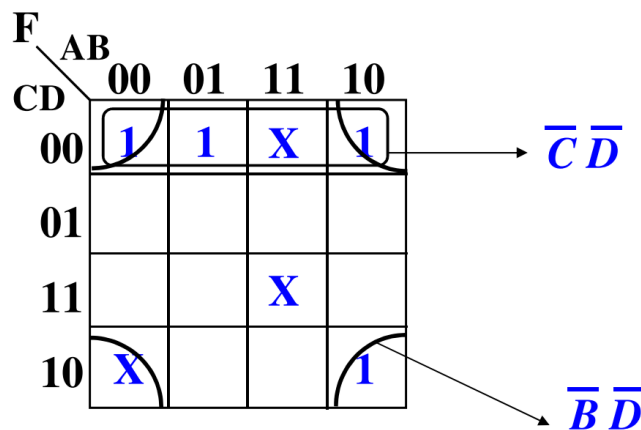
- Biểu diễn các Ô\_0 lên bìa Karnaugh
- Thực hiện các liên kết có thể có sao cho các Ô\_0 được liên kết ít nhất 1 lần; mỗi liên kết cho ta 1 số hạng tổng.
- Biểu thức rút gọn có được bằng cách lấy tích (AND) của các số hạng tổng liên kết trên.

$$F(A, B, C, D) = \Pi(0, 4, 8, 9, 12, 13, 15) = (C + D)(\bar{A} + C)(\bar{A} + \bar{B} + \bar{D})$$

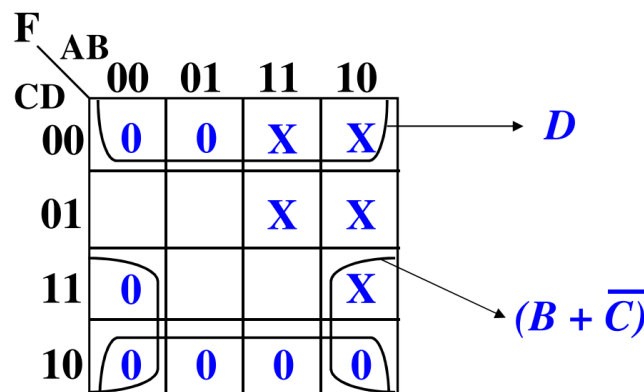


**\*Trường hợp rút gọn hàm Boole có tùy định:** thì ta có thể coi các Ô tùy định này là Ô\_1 hoặc Ô\_0 sao cho có lợi khi liên kết (nghĩa là có được liên kết nhiều Ô kề cận nhất).

$$F(A, B, C, D) = \Sigma(0, 4, 8, 10) + d(2, 12, 15) = \bar{C}\bar{D} + \bar{B}\bar{D}$$



$$F(A, B, C, D) = \Pi(0, 2, 3, 4, 6, 10, 14).D(8, 9, 11, 12, 13) = D(B + \bar{C})$$



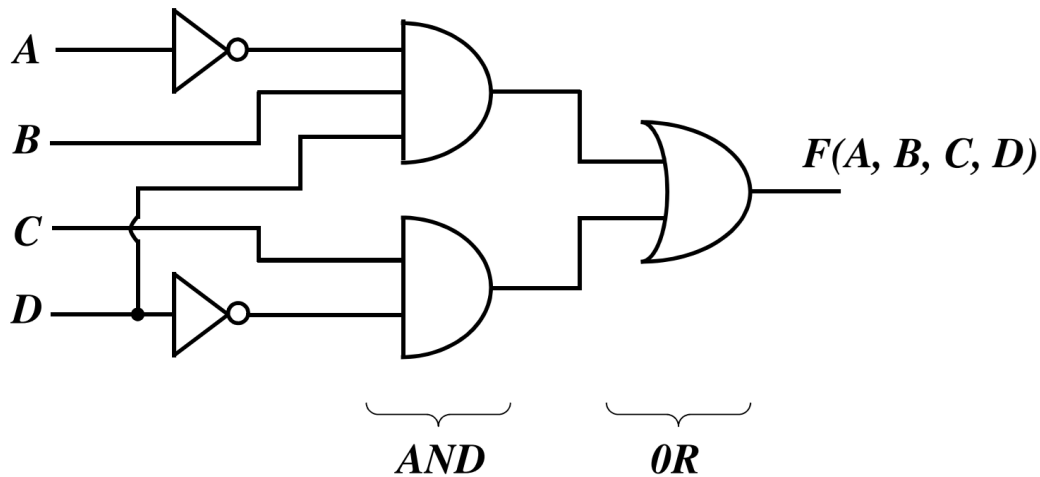
**\*Chú ý:**

- Ưu tiên liên kết cho các ô cử co 1 kiểu liên kết (phải là liên kết có nhiều ô nhất).
- Khi liên kết phải đảm bảo có chứa ít nhất 1 ô chưa được liên kết lần nào.
- Có thể có nhiều cách liên kết có kết quả tương đương nhau.
- Ta coi các tùy định như là những ô đã liên kết rồi.

**2.6 Thực hiện hàm Boole bằng cổng logic:****2.6.1 Cấu trúc cổng AND\_OR:**

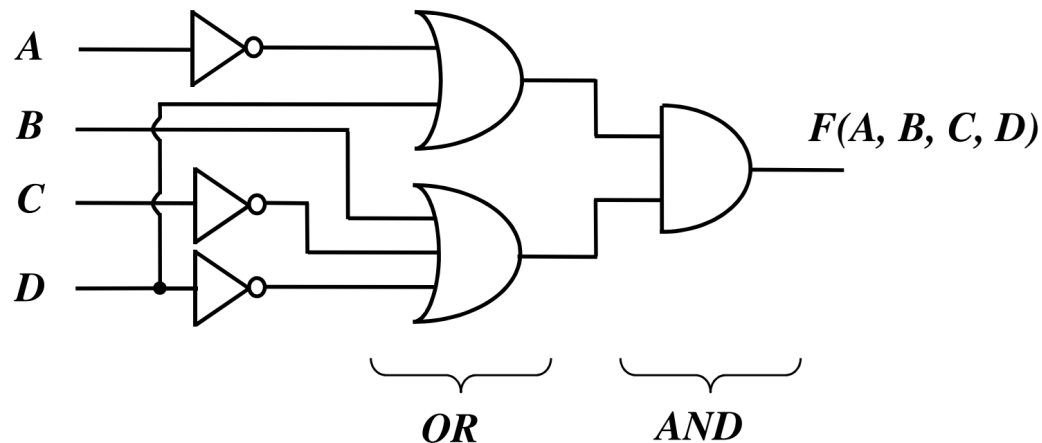
Cấu trúc cổng AND\_OR là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng tổng các tích (S.O.P).

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

**2.6.2 Cấu trúc cổng OR\_AND:**

Cấu trúc cổng OR\_AND là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng tích các tổng (P.O.S).

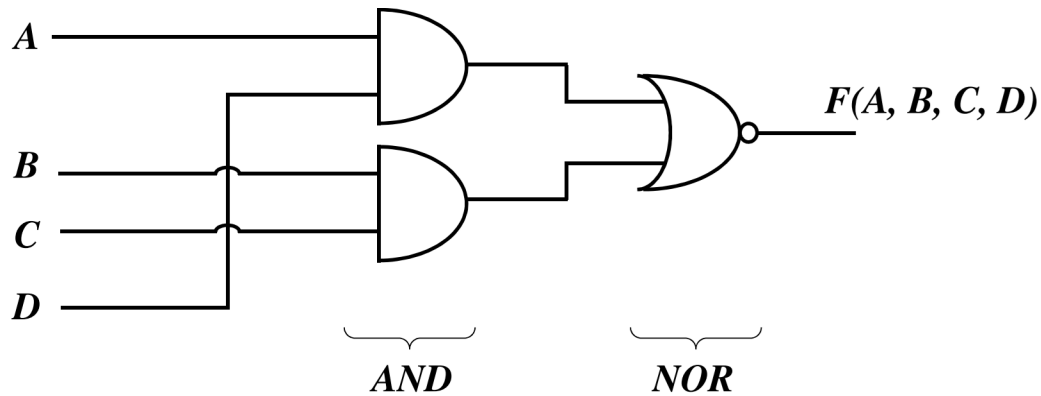
$$F(A, B, C, D) = (\bar{A} + D)(B + \bar{C} + \bar{D})$$



### 2.6.3 Cấu trúc cổng AND\_OR\_INVERTER (AOI):

Cấu trúc cổng AOI là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng bù (INVERTER = NOT) của tổng các tích.

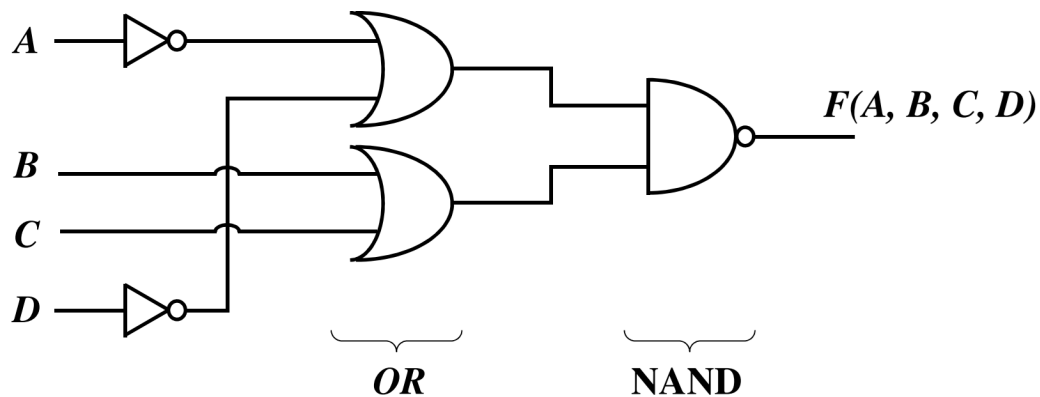
$$F(A, B, C, D) = \overline{AD + BC}$$



### 2.6.4 Cấu trúc cổng OR\_AND\_INVERTER (OAI):

Cấu trúc cổng OAI là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng bù của tích các tổng.

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

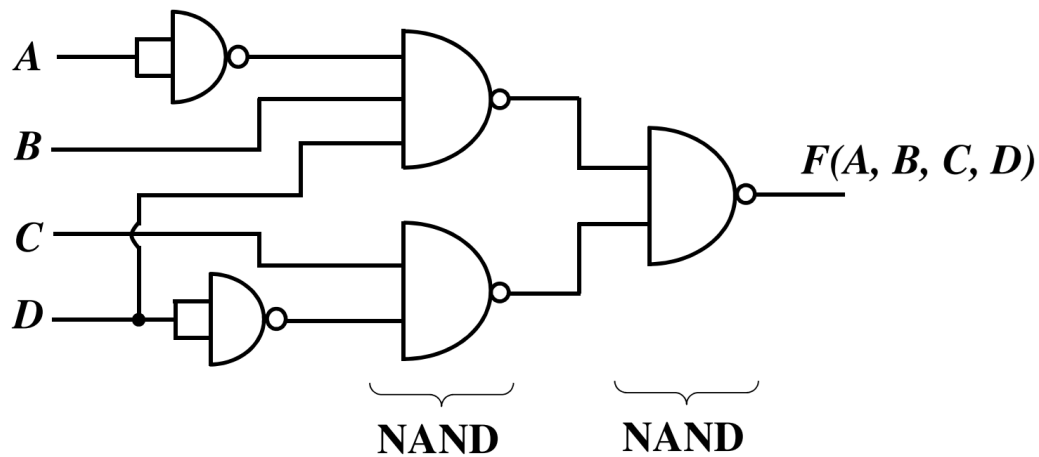


### 2.6.5 Cấu trúc toàn cổng NAND:

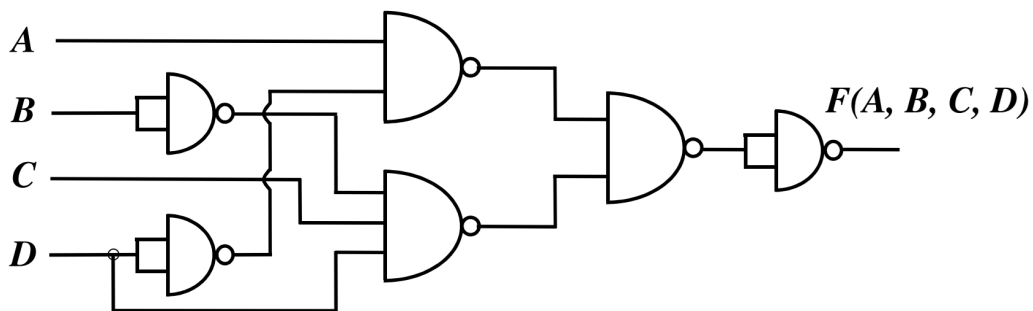
Cấu trúc NAND là sơ đồ logic thực hiện cho hàm Boole có biểu thức là dạng bù của 1 số hạng tích.

- Dùng định lý De-Morgan để biến đổi số hạng tổng thành tích.
- Cổng NOT cũng được thay thế bằng cổng NAND.

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

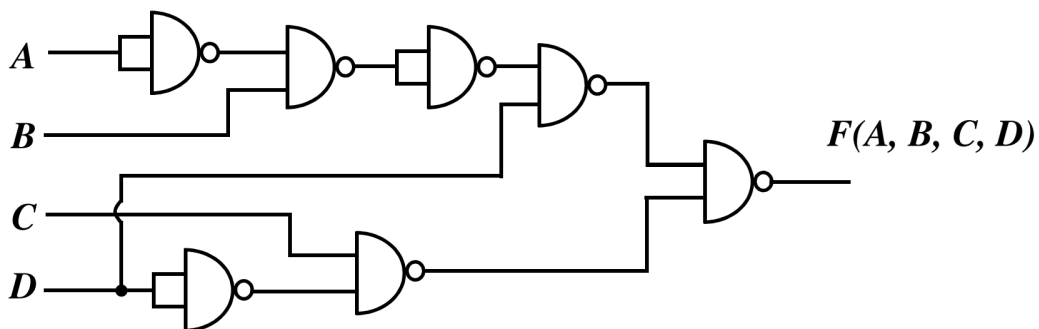


$$F(A, B, C, D) = \overline{\overline{A + D}} \overline{\overline{B + C + D}} = \overline{\overline{AD} \cdot \overline{BCD}}$$



Trong thực tế người ta chỉ sử dụng 1 loại cổng NAND 2 ngõ vào; khi đó ta phải biến đổi biểu thức sao cho chỉ có dạng bù trên 1 số hạng tích chỉ có 2 biến.

$$F(A, B, C, D) = \overline{\overline{ABD} \cdot \overline{CD}} = \overline{\overline{ABD} \cdot \overline{CD}}$$



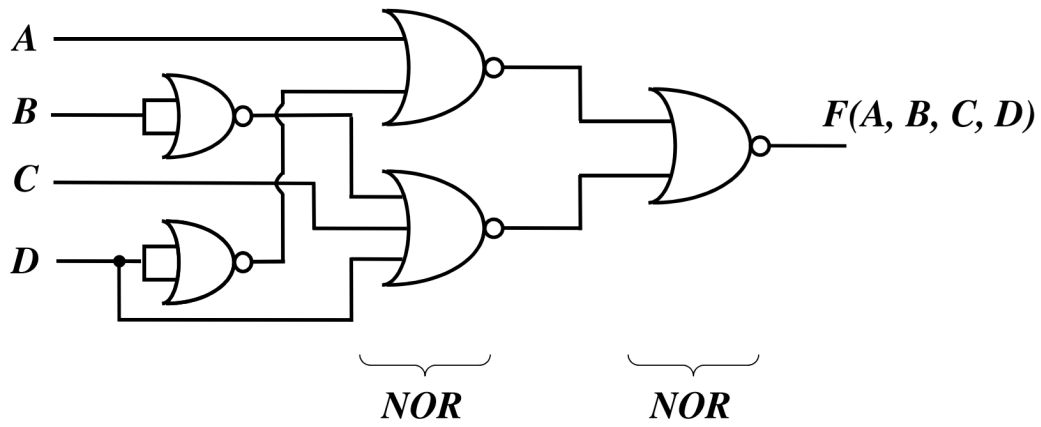
### 2.6.6 Cấu trúc toàn cổng NOR:

Cấu trúc NOR là sơ đồ logic thực hiện cho hàm Boole có biểu thức là dạng bù của 1 số hạng tổng.

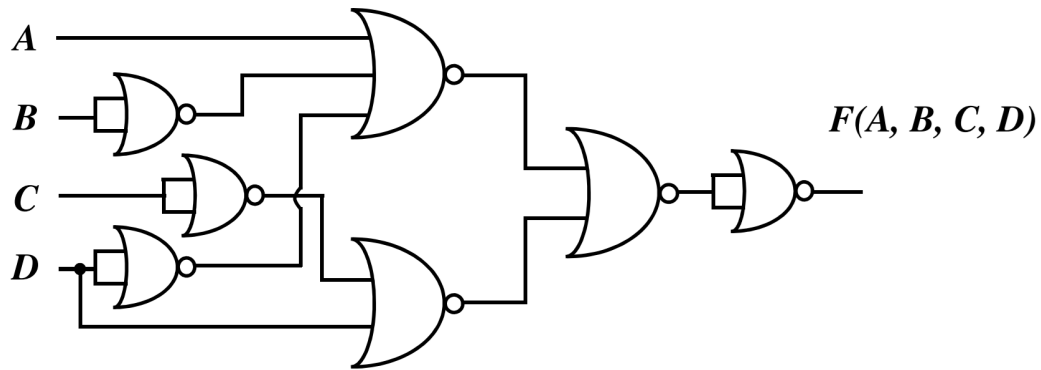
- Dùng định lý De-Morgan để biến đổi số hạng tích thành tổng.
- Cổng NOT cũng được thay thế bằng cổng NOR.

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

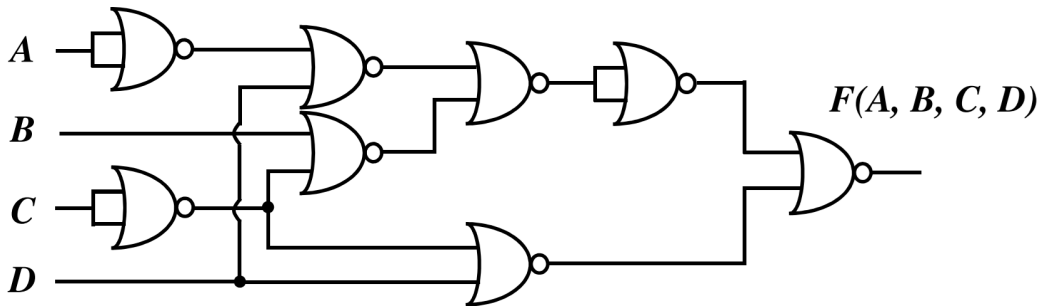




$$F(A, B, C, D) = \overline{\overline{A}BD} + \overline{\overline{C}D} = \overline{\overline{\overline{A} + \overline{B} + \overline{D}}} + \overline{\overline{\overline{C} + D}}$$



$$\begin{aligned} F(A, B, C, D) &= \overline{\overline{\overline{A} + D}(B + \overline{C})(\overline{C} + D)} = \overline{\overline{\overline{A} + D} + (B + \overline{C}) + (\overline{C} + D)} \\ &= \overline{\overline{\overline{A} + D} + \overline{\overline{B + \overline{C}}} + \overline{\overline{\overline{C} + D}}} \end{aligned}$$



## 3 HỆ TỔ HỢP

### 3.1 Giới thiệu - Cách thiết kế hệ tổ hợp:

### 3.2 Bộ cộng - trừ nhị phân:

#### 3.2.1 Bộ cộng (Adder):

#### 3.2.2 Bộ trừ (Subtractor):

#### 3.2.3 Bộ cộng/trừ nhị phân song song:

### 3.3 Hệ chuyển mã (Code conversion):

### 3.4 Bộ giải mã (Decoder):

#### 3.4.1 Giới thiệu:

#### 3.4.2 IC giải mã:

#### 3.4.3 Sử dụng bộ giải mã thực hiện hàm Boole:

### 3.5 Bộ mã hóa (Encoder):

#### 3.5.1 Giới thiệu:

#### 3.5.2 IC mã hóa ưu tiên $8 \rightarrow 3$ (74148):

### 3.6 Bộ dồn kênh (Multiplexer - MUX):

#### 3.6.1 Giới thiệu:

#### 3.6.2 IC dồn kênh:

#### 3.6.3 Sử dụng bộ MUX thực hiện hàm Boole:

### 3.7 Bộ ohaan kênh (Demux):

#### 3.7.1 Giới thiệu:

#### 3.7.2 IC phân kênh 74LS155:

### 3.8 Bộ so sánh độ lớn (Comparator):

#### 3.8.1 Giới thiệu:

#### 3.8.2 IC so sánh 74LS85: