

TRƯỜNG ĐẠI HỌC PHẠM VĂN ĐỒNG
KHOA KỸ THUẬT CÔNG NGHỆ

BÀI GIẢNG ĐIỆN TỬ SỐ

Bậc học: ĐẠI HỌC
(Dành cho Sinh viên Khoa CNTT)

GV: Nguyễn Đình Hoàng
Bộ môn: Điện - Điện tử
Khoa: Kỹ thuật Công nghệ

Quảng Ngãi, năm 2018

TRƯỜNG ĐẠI HỌC PHẠM VĂN ĐỒNG
KHOA KỸ THUẬT CÔNG NGHỆ

BÀI GIẢNG ĐIỆN TỬ SỐ

Bậc học: ĐẠI HỌC
SỐ TÍN CHỈ: 3

GV: Nguyễn Đình Hoàng
Bộ môn: Điện - Điện tử
Khoa: Kỹ thuật Công nghệ

Quảng Ngãi, năm 2018

Lời nói đầu

Nhằm đáp ứng cho việc giảng dạy môn Điện tử số bậc Đại học, tác giả đã biên soạn bài giảng này nhằm làm tài liệu học tập cho các lớp chuyên ngành Công nghệ thông tin tại Đại học Phạm Văn Đồng. Tài liệu này được sử dụng cho sinh viên các lớp ĐH CNTT với thời lượng 45 tiết (3TC). Tác giả hy vọng rằng đây sẽ là tài liệu thiết thực cho các bạn sinh viên.

Trong quá trình biên soạn, chắc chắn tài liệu không tránh khỏi có những sai sót.

Mọi góp ý xin gửi về địa chỉ Nguyễn Đình Hoàng - Khoa Kỹ Thuật Công Nghệ - Trường Đại học Phạm Văn Đồng. Xin chân thành cảm ơn.

Tác giả

MUC LUC

Chương 1: CÁC HỆ THỐNG SỐ VÀ MÃ	1
Chương 2: ĐẠI SỐ BOOLE VÀ CÁC CỔNG LOGIC.....	8
Chương 3: MẠCH TỔ HỢP	25
Chương 4: MẠCH TUẦN TỰ.....	54
Chương 5: BỘ NHỚ BÁN DẪN	68
Tài liệu tham khảo:	76

TaiLieu.vn

Chương 1: CÁC HỆ THỐNG SỐ VÀ MÃ

1.1. Khái niệm

Hệ thống số chúng ta thường sử dụng là hệ thống số có vị trí. Trong một hệ thống như vậy một số biểu diễn bằng một chuỗi các ký tự số (digit); ở đó mỗi vị trí của ký tự số sẽ có một trọng số (weight) nhất định. Trọng số ở đây chính là cơ số lũy thừa vị trí của ký tự số trong chuỗi. Cơ số (radix) chính là số ký tự số được dùng để biểu diễn trong một hệ thống.

Các hệ thống số thường gặp là hệ thống số thập phân (Decimal system), hệ thống số nhị phân (Binary system), hệ thống số bát phân (Octal system), hệ thống số thập lục phân (Hexa-decimal) ...

Giá trị thập phân của một số được tính theo công thức sau:

$$\text{Giá trị} = \sum(\text{Ký số} \times \text{trọng số})$$

1.2. Hệ thống số thập phân (Decimal)

Hệ thập phân dùng 10 chữ số 0 1 2 3 4 5 6 7 8 9 để biểu diễn các số (cơ số 10).

Ví dụ: Tính giá trị của 1234567 trong hệ thập phân.

Biểu diễn theo công thức tổng quát:

$$1234567 = 1 \cdot 10^6 + 2 \cdot 10^5 + 3 \cdot 10^4 + 4 \cdot 10^3 + 5 \cdot 10^2 + 6 \cdot 10^1 + 7 \cdot 10^0 = 1.000.000 + 200.000 + 30.000 + 4.000 + 500 + 60 + 7 = 1234567_{10}$$

1.3. Hệ thống nhị phân (Binary)

Hệ nhị phân dùng 2 chữ số 0 và 1 để biểu diễn các số (cơ số 2).

Ví dụ: Tính giá trị của số 100111 trong hệ nhị phân.

Biểu diễn theo công thức tổng quát:

$$100111_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 32 + 0 + 0 + 4 + 2 + 1 = 39_{10}$$

1.4. Hệ thống thập lục phân (Hexadecimal)

Hệ thập lục phân dùng 16 chữ số 0 1 2 3 4 5 6 7 8 9 A B C D E F để biểu diễn các số (cơ số 16).

Ví dụ: Tính giá trị của số 4B trong hệ thập lục phân.

Biểu diễn theo công thức tổng quát:

$$4B_{\text{Hex}} = 4 \cdot 16^1 + B \cdot 16^0 = 64 + 10 = 74_{10}$$

1.5. Biến đổi qua lại giữa các cơ số

1.5.1. Đổi một số từ nhị phân, thập lục phân sang thập phân

Nguyên tắc: lấy mỗi số hạng trong chuỗi số nhân với cơ số lũy thừa vị trí của nó sau đó lấy tổng tất cả.

- Đổi từ nhị phân sang thập phân.

Ví dụ: Đổi số 1010,11 ở cơ số 2 sang cơ số 10

$$1011,11_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 8 + 0 + 4 + 1 + 0,5 + 0,25 = 13,75_{10}$$

- Đổi từ thập lục phân sang thập phân.

Ví dụ: Đổi giá trị của số 4B,8F trong hệ thập lục phân sang hệ thập phân.

$$4B,8F_{16} = 4 \times 16^1 + B \times 16^0 + 8 \times 16^{-1} + 15 \times 16^{-2} = 64 + 11 + 0,5 + 0,05859375 = 75,55859375_{10}$$

1.5.2. Đổi một số từ hệ thập phân sang hệ nhị phân

Chia làm hai phần : phần nguyên và phần thập phân.

Phần nguyên: chia liên tiếp cho 2 đến khi thương số bằng 0, số nhị phân là các số dư lấy từ dưới lên.

Phần thập phân: nhân liên tiếp cho 2 cho đến khi bằng 0 hoặc đạt được số lẻ cần thiết, số nhị phân phần phân sẽ là phần nguyên của phép nhân 2.

Ví dụ: Đổi số thập phân 8.25 sang nhị phân.

Phần nguyên:

$$\begin{array}{l} 8:2 = 4 \text{ dư } 0 \\ 4:2 = 2 \text{ dư } 0 \\ 2:2 = 1 \text{ dư } 0 \\ 1:2 = 0 \text{ dư } 1 \end{array} \quad \begin{array}{l} \uparrow \text{MSB} \\ \downarrow \text{LSB} \end{array}$$

.Phần thập phân:

$$\begin{array}{l} 0.25 \times 2 = 0.5 \text{ phần nguyên } 0 \\ 0.5 \times 2 = 1.0 \text{ phần nguyên } 1 \end{array} \quad \begin{array}{l} \uparrow \text{MSB} \\ \downarrow \text{LSB} \end{array}$$

$$\text{Kết quả chuyển đổi: } 8.25_{10} = 1000.01_2$$

1.5.3. Đổi một số từ thập phân sang thập lục phân

Tương tự như cách đổi từ thập phân sang nhị phân, với số chia là 16 (phần nguyên) và số nhân 16 (phần phân).

1.5.4. Đổi một số từ nhị phân sang thập lục phân

Nguyên tắc : Nhóm từ phải qua trái đủ bốn bit; nhóm cuối cùng nếu thiếu thì ta cứ thêm các số 0 vào. Thay thế các nhóm 4 bit thành các mã thập lục phân tương ứng.

Ví dụ : 1010 0101₂

A 5

$$1010 \ 0101_2 = A5_{16}$$

1.5.5. Đổi một số từ thập lục phân sang nhị phân

Ta thay một số thập lục phân bằng một số nhị 4 bit tương ứng

Ví dụ: $2B6_{16} = \underline{0010} \ \underline{1011} \ \underline{0110}_2$

Bảng mã tương quan giữa thập phân, nhị phân và thập lục phân.

Thập phân	Nhi phân	Thập lục phân
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1.6. Số có dấu

1.6.1 Biểu diễn số có dấu

Khi biểu diễn số có dấu thông thường ta sử dụng thêm 1 bit gọi là bit dấu (thường đặt ở vị trí số có trọng số cao nhất MSB) với qui ước bit 0 để chỉ số dương; bit 1 để chỉ số âm.

Ví dụ:

$$\underline{1} 0101 = -5$$

$$\underline{0} 0101 = +5$$

1.6.2. Số bù 1 (dùng biểu diễn số âm)

Số bù 1 được định nghĩa cho một số N có n số sẽ bằng: $r^n - 1 - N$ (với r là cơ số).

Ví dụ: Tìm số bù của số nhị phân $N = 1010_2$

$$r^n = 2^4 = 10000, r^n - 1 - N = 10000 - 1 - 1010 = 0101 \text{ (bù 1)}$$

hoặc có thể tìm bù 1 bằng cách đảo các bit của số N.

1.6.3. Số bù 2 (biểu diễn số âm)

Số bù 2 có được bằng cách cộng số bù 1 với 1

Ví dụ: Tìm số bù 2 của số 1010

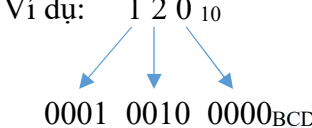
Bù 1 của số 1010 là 0101

Bù 2: $0101 + 1 = 0110$ (bù 2 của số nhị phân 1010)

1.7. Các loại mã

1.7.1. Mã BCD

Mã số BCD là số thập phân mã hóa theo nhị phân. Mã số này dùng nhóm bốn bit để biểu thị số thập phân từ 0 đến 9.

Ví dụ: $1\ 2\ 0_{10}$

0001 0010 0000_{BCD}

1.7.2. Mã thừa 3

Mã quá 3 (thừa 3) là mã có được khi tăng 3 đơn vị từ mã Binary. Nghĩa là cộng thêm 011_2 .

Ví dụ: $0101 \rightarrow 1000$ (thừa 3)

1.7.2. Mã Gray

Mã Gray hay còn gọi là mã vòng suy ra từ mã nhị phân. Giả sử cho mã nhị phân có bốn bit $B_3\ B_2\ B_1\ B_0$, mã Gray tương ứng là $G_3\ G_2\ G_1\ G_0$ thì có thể tính theo công thức sau :

$$G_i = B_{i+1} \oplus B_i$$

Đặc điểm của mã Gray là giữa hai số nhị phân liên tiếp chỉ được phép thay đổi 1 bit.

1.7.3. Mã ký tự (ASCII)

Mã ASCII là mã mà hầu hết các máy tính đều dùng (mã chuẩn của Mỹ: American Standard Code for Information Interchange). Mỗi ký tự (chữ cái, chữ số, dấu, ký hiệu đặt biệt ...) tương ứng với một mã 8 bit (là dãy liên tiếp các chữ số 0 và 1)

Bảng mã ký tự ASCII:

		(Cột) $b_6 b_5 b_4$							
(Hàng)		000	001	010	011	100	101	110	111
$b_3 b_2 b_1 b_0$	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	FS	,	<	L	\	l	
1101	D	CR	GS	-	=	M	]	m	}
1110	E	SO	RS	.	>	N	^	n	~
1111	F	SI	US	/	?	O	_	o	DEL

1.8. Các phép toán trên số nhị phân

Cũng như số học thập phân, số học nhị phân cũng có bốn phép tính cơ bản là: Cộng (+), Trừ (-), Nhân (*), Chia (/)

1.8.1. Phép cộng

Nguyên tắc:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ Nhớ 1 (carry)}$$

Ví dụ:

$$\begin{array}{r} 10011 \\ + \quad 011 \\ \hline 10110 \end{array}$$

1.8.2. Phép trừ

Nguyên tắc:

$$0 - 0 = 0$$

$$0 - 1 = 1 \text{ mượn 1 (borrow)}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Ví dụ:

$$\begin{array}{r} 10011 \\ - \quad 111 \\ \hline 01100 \end{array}$$

1.8.3. Phép nhân

Nguyên tắc:

$$\begin{aligned} 0 \times 0 &= 0 \\ 0 \times 1 &= 0 \\ 1 \times 0 &= 0 \\ 1 \times 1 &= 1 \end{aligned}$$

Ví dụ:

$$\begin{array}{r} 10011 \\ \times \quad 011 \\ \hline 10011 \\ 10011 \\ 00000 \\ \hline 0111001 \end{array}$$

1.8.4. Phép chia

Thực hiện giống như phép chia thập phân

Ví dụ:

$$\begin{array}{r|l} 1011 & 101 \\ - & \\ \hline 101 & 10 \text{ (thương số)} \\ 0001 & \\ \hline & \text{(số dư)} \end{array}$$

Bài tập chương 1

1. Biểu diễn các số sau trong hệ nhị phân (binary)
 - a. 23
 - b. 14
 - c. 27
 - d. 34
2. Biểu diễn các số sau trong hệ nhị phân (binary)
 - a. 23Hex 23Decimal
 - b. C06AH
 - c. 5DEFH
3. Biểu diễn các số sau trong hệ thập phân (decimal)
 - a. 01101001B
 - b. 01111111B
 - c. 10000000B
 - d. 11111111B
4. Biểu diễn các số sau trong hệ thập phân (decimal)
 - a. 1FH
 - b. 10H
 - c. FFH
 - d. 03H
5. Biểu diễn các số sau trong hệ thập lục phân (hex)
 - a. 100D
 - b. 128
 - c. 127
 - d. 256
6. Cho các số nhị phân sau, hãy xác định giá trị của chúng nếu chúng là (i) số nhị phân không dấu; (ii) số nhị phân có dấu
 - a. 0000B
 - b. 0001B
 - c. 0111B
 - d. 1000B
 - e. 1001B
 - f. 1110B
 - g. 1111B

Chương 2: ĐẠI SỐ BOOLE VÀ CÁC CÔNG LOGIC

2.1. Đại số Boole

2.1.1. Các định nghĩa

Biến logic: là 1 đại lượng có thể biểu diễn bằng 1 ký hiệu nào đó, về mặt giá trị chỉ lấy giá trị 0 hoặc 1.

Hàm logic: biểu diễn của nhóm các biến logic, liên hệ với nhau thông qua các phép toán logic, về mặt giá trị cũng lấy giá trị 0 hoặc 1.

2.1.2. Phép toán logic

Có 3 loại phép toán logic cơ bản:

- Phép Và - "AND"
- Phép Hoặc - "OR"
- Phép Phủ định - "NOT"

2.2. Hàm và các tính chất của hàm logic cơ bản

2.2.1. Các hàm logic cơ bản

- Hàm Hoặc (OR)
 $F(A, B) = A + B$

Bảng sự thật:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

- Hàm Và (AND)
 $F(A, B) = A.B$

Bảng sự thật:

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

- Hàm phủ định (NOT)
 $F(A) = \overline{A}$

Bảng sự thật:

A	F
0	1
1	0

2.2.2. Các tính chất của các hàm logic cơ bản

- a. Tồn tại phần tử trung tính duy nhất trong phép toán "AND" và "OR"
- Phần tử trung tính của một phép toán là phần tử mà khi ta thực hiện phép toán giữa phần tử này và 1 đại lượng bất kỳ nào đó thì kết quả thu được chính là bằng đại lượng đó.
 - Phần tử trung tính duy nhất của phép "AND" là 1.
 - Phần tử trung tính duy nhất của phép "OR" là 0.

- b. Tính giao hoán

$$A.B = B.A$$

$$A + B = B + A$$

- c. Tính kết hợp

$$(A.B).C = A.(B.C) = A.B.C$$

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

- d. Tính phân phối

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

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

- e. Phép bù

$$\overline{\overline{A}} = A$$

$$A + \overline{A} = 1$$

$$A.\overline{A} = 0$$

- f. Định lý De Morgan

- Phủ định của một "tổng" bằng "tích" các phủ định thành phần.

$$\overline{(a+b)} = \overline{a}.\overline{b}$$

- Phủ định của một "tích" bằng "tổng" các phủ định thành phần.

$$\overline{(a.b)} = \overline{a} + \overline{b}$$

- g. Nguyên lý đối ngẫu

Cộng đối ngẫu với nhân: $+ \sim \times$; 0 đối ngẫu với 1: $0 \sim 1$

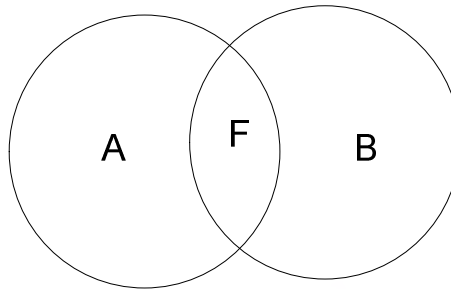
2.3. Các phương pháp biểu diễn hàm và biến logic

2.3.1. Biểu đồ Ven (Ole)

- Mỗi biến logic chia không gian thành 2 không gian con.

- Không gian con thứ nhất, biến nhận giá trị đúng (=1), không gian con thứ còn lại, biến nhận giá trị sai (=0).

Ví dụ: $F = A \text{ AND } B$



2.3.2. Biểu thức đại số

- Ký hiệu phép Và (AND): $\cdot$
 - Ký hiệu phép Hoặc (OR): $+$
 - Ký hiệu phép Đảo (NOT): $\bar{}$
- Ví dụ: $F = A \text{ AND } B$ hay $F = A \cdot B$

2.3.3. Bảng sự thật (Truth Table)

Bảng thật biểu diễn 1 hàm logic n biến có: $(n+1)$ cột và 2^n hàng. Trong đó,

$(n+1)$ cột có:

- o n cột đầu tương ứng với n biến
- o cột còn lại tương ứng với giá trị của hàm

2^n hàng tương ứng với 2^n giá trị của tổ hợp biến.

Bảng sự thật của hàm AND: $F = A \text{ AND } B$, hay $F = A \cdot B$

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Hàm OR: $F = A \text{ OR } B$, hay $F = A + B$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

2.3.4. Bìa Các-nô

Đây là cách biểu diễn tương đương của bảng sự thật. Trong đó, mỗi ô trên bìa tương ứng với 1 dòng của bảng sự thật. Tọa độ của ô xác định giá trị của tổ hợp biến. Giá trị của hàm được ghi vào ô tương ứng.

Ví dụ: $F = A \text{ AND } B$

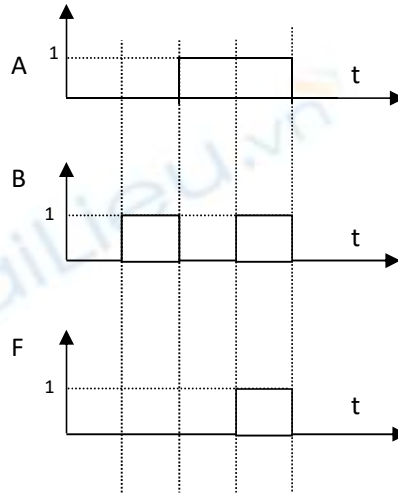
A \ B	0	1
	0	1
0	0	0
1	0	1

2.3.5. Biểu đồ thời gian

Là đồ thị biểu diễn sự biến đổi theo thời gian của biến và hàm logic.

Ví dụ: $F = A \text{ AND } B$

Ta có biểu đồ thời gian như sau:



2.4. Biểu diễn hàm logic dưới dạng chính tắc

Một hàm logic thông thường được biểu diễn dưới 2 dạng:

Tuyến: dạng tổng các tích

Ví dụ: $f(a,b,c)=ab+acb+cb$

Hội: dạng tích các tổng

Ví dụ: $f(a,b,c)=(a+b)(a+c+b)$

Một hàm logic được gọi là biểu diễn dưới dạng chính tắc nếu mỗi số hạng của nó đều có đầy đủ các biến.

Tuyến chính tắc:

Ví dụ: $f(a,b,c)=abc+\bar{a}\bar{b}c$

Hội chính tắc:

Ví dụ: $f(a,b,c)=(a+b+c)(\bar{a}+\bar{b}+c)$

Một hàm logic được gọi là biểu diễn dưới dạng không chính tắc nếu như có ít nhất một biến vắng mặt trong ít nhất một số hạng. Lúc này hàm được gọi là biểu diễn dưới dạng đơn giản hóa.

2.4.1. Tuyến chính tắc

a. Định lý Shannon

Một hàm logic bất kỳ có thể được triển khai theo 1 trong các biến dưới dạng tổng của 2 tích logic như sau:

$$F(A_1, A_2, \dots, A_n) = A_1 F(1, A_2, \dots, A_n) + \bar{A}_1 F(0, A_2, \dots, A_n)$$

$$\begin{aligned}
 \text{Ví dụ: } F(A,B) &= A F(1,B) + \overline{A} F(0,B) \\
 &= A(BF(1,1) + \overline{B} F(1,0)) + \overline{A} (BF(0,1) + \overline{B} F(0,0)) \\
 &= ABF(1,1) + A\overline{B} F(1,0) + \overline{A} BF(0,1) + \overline{A}\overline{B} F(0,0)
 \end{aligned}$$

Kết luận: 1 hàm logic bất kỳ đều có thể chuyển về dạng tuyển chính tắc nhờ áp dụng định lý Shannon.

b. Cách áp dụng

Cách áp dụng nhanh định lý Shannon: Từ bảng sự thật, ta chỉ quan tâm tới giá trị của hàm bằng 1. Với mỗi giá trị bằng 1, ta thành lập biểu thức tổ hợp tích các biến theo quy tắc giá trị biến bằng 1 thì giữ nguyên, giá trị biến bằng 0 thì đảo. Biểu thức cuối cùng là tổng của các tổ hợp biến nói trên.

Ví dụ:

A	B	C	F ₁	F ₂
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

$$\begin{aligned}
 F_1 &= \overline{A} \overline{B} C + \overline{A} B \overline{C} + \overline{A} BC + A \overline{B} \overline{C} + A \overline{B} C + AB \overline{C} + ABC \\
 F_2 &= \overline{A} BC + A \overline{B} C + ABC
 \end{aligned}$$

2.4.2. Hội chính tắc

a. Định lý Shanon

Một hàm logic bất kỳ có thể được khai triển theo một trong các biến dưới dạng tích của hai tổng logic như sau:

$$F(A_1, \dots, A_n) = [A_1 + F(0, \dots, A_n)][\overline{A}_1 + F(1, \dots, A_n)]$$

Ví dụ:

$$\begin{aligned}
 F(A,B) &= [A + F(0,B)][\overline{A} + F(1,B)] \\
 &= \{A + [B + F(0,0)][\overline{B} + F(0,1)]\} \{ \overline{A} + [B + F(0,1)][\overline{B} + F(1,1)] \} \\
 &= [A + B + F(0,0)][A + \overline{B} + F(0,1)][\overline{A} + B + F(1,0)][\overline{A} + \overline{B} + F(1,1)]
 \end{aligned}$$

Ví dụ:

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

Kết luận: 1 hàm logic bất kỳ đều có thể chuyển về dạng hội chính tắc nhờ áp dụng định lý Shannon.

b. Cách áp dụng

Cách áp dụng nhanh định lý Shannon: Từ bảng sự thật, ta chỉ quan tâm tới giá trị của hàm bằng 0. Với mỗi giá trị bằng 0, ta thành lập biểu thức tổ hợp tổng các biến theo quy tắc giá trị biến bằng 1 thì đảo, giá trị biến bằng 0 thì giữ nguyên. Biểu thức cuối cùng là tích của các tổ hợp biến nói trên.

Ví dụ:

A	B	C	F
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

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

2.5. Biểu diễn hàm logic dưới dạng số

2.5.1. Tuyển chính tắc

Dạng tuyển chính tắc quan tâm tới những tổ hợp biến mà tại đó hàm nhận giá trị bằng 1. Việc biểu diễn hàm tuyển chính tắc dưới dạng số liệt kê các tổ hợp biến mà tại đó hàm có giá trị bằng 1.

Ví dụ: $F(A,B) = \Sigma(3)$

Trong đó, 3 tương ứng với tổ hợp biến $AB = 11$.

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

$$F = AB$$

Ví dụ: $F(A,B) = \Sigma(1,3)$

Trong đó, 1, 3 tương ứng với tổ hợp biến $AB = 01, 11$.

A	B	F
0	0	0
0	1	1
1	0	0
1	1	1

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

Ví dụ: $F(A,B,C) = \Sigma(1,2,4,6)$

Trong đó, 1, 2, 4, 6 tương ứng với tổ hợp biến $ABC = 001, 010, 100, 110$.

A	B	C	F ₂
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

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

2.5.2. Hội chính tắc

Dạng hội chính tắc quan tâm tới những tổ hợp biến mà tại đó hàm nhận giá trị bằng 0. Việc biểu diễn hàm logic hội chính tắc dưới dạng số liệt kê các tổ hợp biến mà tại đó hàm có giá trị bằng 0.

Ví dụ: $F(A, B, C) = \Pi(0, 3, 5, 7)$

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

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

2.6. Tối thiểu hóa các hàm logic

Một hàm logic được gọi là tối thiểu hoá nếu như nó có số lượng số hạng ít nhất và số lượng biến ít nhất.

Mục đích của việc tối thiểu hoá: Mỗi hàm logic có thể được biểu diễn bằng các biểu thức logic khác nhau. Mỗi 1 biểu thức logic có một mạch thực hiện tương ứng với nó. Biểu thức logic càng đơn giản thì mạch thực hiện càng đơn giản.

Có hai phương pháp để tối thiểu hoá hàm logic:

- Phương pháp đại số
- Phương pháp bìa Các-nô

2.6.1. Phương pháp đại số

a. Sử dụng các tính chất của đại số Boole

$$- \quad AB + A\overline{B} = A \quad \leftrightarrow \quad (A+B)(A+\overline{B}) = A$$

$$\begin{array}{llll}
 - & A + AB & = A & \leftrightarrow A(A+B) = A \\
 - & A + \overline{A}B & = A+B & \leftrightarrow A(\overline{A}+B) = AB
 \end{array}$$

CM:

$$\begin{array}{llll}
 AB + A\overline{B} & = A(B + \overline{B}) & = A.1 = A \\
 A + AB & = A(1+B) & = A.1 = A \\
 A + \overline{A}B & = (A + \overline{A})(A+B) \\
 & = 1(A+B) \\
 & = A+B
 \end{array}$$

b. Quy tắc tối thiểu hoá

- Sử dụng phương pháp nhóm số hạng

Ví dụ:

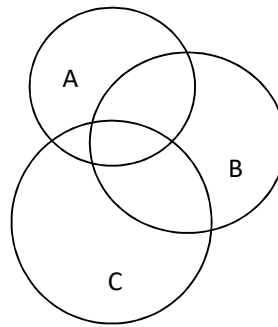
$$\begin{aligned}
 F(A,B,C,D) &= ABC + AB\overline{C} + A\overline{B}CD \\
 &= AB(C + \overline{C}) + A\overline{B}CD \\
 &= AB + A\overline{B}CD \\
 &= A(B + \overline{B}CD) \\
 &= A(B + CD) \\
 &= AB + ACD
 \end{aligned}$$

- Thêm 1 số hạng đã có vào biểu thức:

$$\begin{aligned}
 F(x,y,z) &= xyz + \overline{x}yz + xy\overline{z} + x\overline{y}z \\
 &= xyz + \overline{x}yz + xyz + xy\overline{z} + xyz + x\overline{y}z \\
 &= yz + xy + xz
 \end{aligned}$$

- Loại bỏ đi số hạng thừa:

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



→ AC là số hạng thừa

Ta có:

$$\begin{aligned}
 F &= AB + \overline{B}C + AC(B + \overline{B}) \\
 &= AB + \overline{B}C + ACB + AC\overline{B} \\
 &= AB + ABC + \overline{B}C + A\overline{B}C \\
 &= AB(1+C) + \overline{B}C(1+A) \\
 &= AB + \overline{B}C
 \end{aligned}$$

2.6.2. Phương pháp sử dụng Bìa Các-nô

a. Quy tắc lập bìa Các-nô

Bìa Các-nô có số ô bằng 2^n , với n là số biến của hàm Boole, một ô sẽ tương đương với một tổ hợp của các biến đã cho. Hai ô liền kề nhau chỉ sai khác nhau 1 giá trị của 1 biến (tương ứng với tổ hợp biến khác nhau 1 giá trị). Bìa Các-nô có tính không gian

- Bìa Các-nô cho 2 biến:

A \ B	0	1
B		
0		
1		

- Bìa Các-nô cho 3 biến:

AB \ C	00	01	11	10
C				
0				
1				

- Bìa Các-nô cho 4 biến:

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