CHƯƠNG VIII

ĐẠI SỐ BOOLE

Các mạch điện trong máy tính và các dụng cụ điện tử khác đều có các đầu vào, mỗi đầu vào là số 0 hoặc số 1, và tạo ra các đầu ra cũng là các số 0 và 1. Các mạch điện đó đều có thể được xây dựng bằng cách dùng bất kỳ một phần tử cơ bản nào có hai trạng thái khác nhau. Chúng bao gồm các chuyển mạch có thể ở hai vị trí mở hoặc đóng và các dụng cụ quang học có thể là sáng hoặc tối. Năm 1938 Claude Shannon chứng tỏ rằng có thể dùng các quy tắc cơ bản của lôgic do George Boole đưa ra vào năm 1854 trong cuốn "Các quy luật của tư duy" của ông để thiết kế các mạch điện. Các quy tắc này đã tạo nên cơ sở của đại số Boole. Sự hoạt động của một mạch điện được xác định bởi một hàm Boole chỉ rõ giá trị của đầu ra đối với mỗi tập đầu vào. Bước đầu tiên trong việc xây dựng một mạch điện là biểu diễn hàm Boole của nó bằng một biểu thức được lập bằng cách dùng các phép toán cơ bản của đại số Boole. Biểu thức mà ta sẽ nhận được có thể chứa nhiều phép toán hơn mức cần thiết để biểu diễn hàm đó. Ở cuối chương này, ta sẽ có các phương pháp tìm một biểu thức với số tối thiểu các phép tổng và tích được dùng để biểu diễn một hàm Boole. Các thủ tục được mô tả là bản đồ Karnaugh và phương pháp Quine-McCluskey, chúng đóng vai trò quan trọng trong việc thiết kế các mạch điện có hiệu quả cao.

8.1. KHÁI NIỆM ĐẠI SỐ BOOLE.

8.1.1. Định nghĩa: Tập hợp khác rỗng S cùng với các phép toán ký hiệu nhân (**.**), cộng (+), lấy bù (') được gọi là một đại số Boole nếu các tiên đề sau đây được thoả mãn với mọi a, b, c $\in$ S.

1. Tính giao hoán: **a)** a.b = b.a,
b) a+b = b+a.

2. Tính kết hợp: **a)** (a.b).c = a.(b.c),
b) (a+b)+c = a+(b+c).

3. Tính phân phối: **a)** a.(b+c) = (a.b)+(a.c),
b) a+(b.c) = (a+b).(a+c).

4. Tồn tại phần tử trung hoà: Tồn tại hai phần tử khác nhau của S, ký hiệu là 1 và 0 sao cho: **a)** a.1 = 1.a = a,
b) a+0 = 0+a = a.

1 gọi là phần tử trung hoà của phép **.** và 0 gọi là phần tử trung hoà của phép +.

5. Tồn tại phần tử bù: Với mọi a $\in$ S, tồn tại duy nhất phần tử a' $\in$ S sao cho:
a) a.a' = a'.a = 0,
b) a+a' = a'+a = 1.

a’ gọi là phần tử bù của a.

Thí dụ 1:

1) Đại số lôgic là một đại số Boole, trong đó S là tập hợp các mệnh đề, các phép toán $\wedge$ (hội), $\vee$ (tuyển), − (phủ định) tương ứng với **.** , +, ’, các hằng đ (đúng), s (sai) tương ứng với các phần tử trung hoà 1, 0.

2) Đại số tập hợp là một đại số Boole, trong đó S là tập hợp $\mathcal{P}$(X) gồm các tập con của tập khác rỗng X, các phép toán $\cap$ (giao), $\cup$ (hợp), − (bù) tương ứng với **.** , +, ’, các tập X, Ø tương ứng với các phần tử trung hoà 1, 0.

3) Cho B = {0,1}, các phép toán **.** , +, ’ trên B được định nghĩa như sau:

1.1 = 1,	1+1 = 1,	1’ = 0,	
1.0 = 0,	1+0 = 1,	0’ = 1.	(1)
0.1 = 0,	0+1 = 1,		
0.0 = 0,	0+0 = 0,		

Khi đó B là một đại số Boole. Đây cũng chính là đại số lôgic, trong đó 1, 0 tương ứng với đ (đúng), s (sai). Mỗi phần tử 0,1 của B gọi là một bit. Ta thường viết $\overline{x}$ thay cho x'.

Tổng quát, gọi B^n là tập hợp các xâu n bit (xâu nhị phân độ dài n). Ta định nghĩa tích, tổng của hai chuỗi và bù của một chuỗi theo từng bit một như trong Bảng 1, mà thường được gọi là các phép toán AND-bit, OR-bit, NOT-bit. B^n với các phép toán này tạo thành một đại số Boole.

4) Cho M là tập hợp các số thực có cận trên p, cận dưới q và tâm đối xứng O. Các phép toán **.** , +, ’ trên M được định nghĩa như sau:

a.b = min(a, b), a+b = max(a, b), a’ là điểm đối xứng của a qua O.

Khi đó M là một đại số Boole, trong đó q, p tương ứng với các phần tử trung hoà 1, 0.

8.1.2. Chú ý: Trước hết cần lưu ý điều quan trọng sau đây: các tiên đề của đại số Boole được xếp theo từng cặp a) và b). Từ mỗi tiên đề a), nếu ta thay **.** bởi +, thay + bởi **.**, thay 1 bởi 0 và thay 0 bởi 1 thì ta được tiên đề b) tương ứng.

Ta gọi cặp tiên đề a), b) là đối ngẫu của nhau. Do đó nếu ta chứng minh được một định lý trong đại số Boole thì ta có ngay một định lý khác, đối ngẫu của nó, bằng cách thay **.** và 1 tương ứng bởi + và 0 (và ngược lại). Ta có:

Quy tắc đối ngẫu: Đối ngẫu của một định lý là một định lý.

8.1.3. Định lý:

6. (Tính nuốt)

a) a.0 = 0,

b) a+1 = 1

7. (Tính luỹ đẳng)

a) a.a = a,

b) a+a = a.

8. (Hệ thức De Morgan)

a) $(a.b)' = a'+b'$,

b) $(a+b)' = a'.b'$.

9. (Hệ thức bù kép)

$(a')' = a$.

10. **a)** $1' = 0$,

b) $0' = 1$.

11. (Tính hút)

a) $a.(a+b) = a$,

b) $a+(a.b) = a$.

Chứng minh:

6. $0 = a.a$ (tiên đề 5a))

$= a.(a'+0)$ (tiên đề 4b))

$= (a.a')+(a.0)$ (tiên đề 3a))

$= 0+(a.0)$ (tiên đề 5a))

$= a.0$ (tiên đề 4b)).

7. $a = a.1$ (tiên đề 4a))

$= a.(a+a')$ (tiên đề 5b))

$= (a.a)+(a.a')$ (tiên đề 3a))

$= (a.a)+0$ (tiên đề 5a))

$= a.a$ (tiên đề 4b))

8. Ta chứng minh rằng $a'+b'$ là bù của $a.b$ bằng cách chứng minh rằng:

$(a.b).(a'+b') = 0$ (theo 5a)) và $(a.b)+(a'+b') = 1$ (theo 5b)).

Thật vậy, $(a.b).(a'+b') = (a.b.a')+(a.b.b') = (a.a'.b)+(a.b.b') = (0.b)+(a.0) = 0+0 = 0$,

$(a.b)+(a'+b') = (a'+b')+(a.b) = (a'+b'+a).(a'+b'+b) = (1+b').(a'+1) = 1.1 = 1$.

Vì $a.b$ chỉ có một phần tử bù duy nhất nên $(a.b)' = a'+b'$.

9. Có ngay từ tiên đề 5.

10. Có từ các hệ thức $1.0 = 0$ và $1+0 = 1$.

11. $a.(a+b) = (a+0).(a+b) = a+(0.b) = a+0 = a$.

8.1.4. Chú ý: Hệ tiên đề của đại số Boole nêu ra ở đây không phải là một hệ tối thiểu. Chẳng hạn, các tiên đề về tính kết hợp có thể suy ra từ các tiên đề khác. Thật vậy, với $A=(a.b).c$ và $B=a.(b.c)$, ta có: $a+A = a+((a.b).c) = (a+(a.b)).(a+c) = a.(a+c) = a$, $a+B = a+(a.(b.c)) = (a+a).(a+(b.c)) = a.(a+(b.c)) = a$, $a'+A = a'+((a.b).c) = (a'+(a.b)).(a'+c) = ((a'+a).(a'+b)).(a'+c) = (1.(a'+b)).(a'+c) = (a'+b).(a'+c) = a'+(b.c)$, $a'+B = a'+(a.(b.c)) = (a'+a).(a'+(b.c)) = 1.(a'+(b.c)) = a'+(b.c)$.

Do đó $a+A = a+B$ và $a'+A = a'+B$. Từ đó suy ra rằng:

A = A+0 = A+(a.a') = (A+a).(A+a') = (a+A).(a'+A) = (a+B).(a'+B)=(a.a')+B=0+B= B hay ta có 2a) và đối ngẫu ta có 2b). Ngoài ra, tính duy nhất của phần tử bù cũng được suy ra từ các tiên đề khác.

Tương tự trong đại số lôgic, trong đại số Boole ta cũng xét các công thức, được thành lập từ các biến a, b, c, … nhờ các phép toán **.** , +, '. Trong công thức, ta quy ước thực hiện các phép toán theo thứ tự: ', **.**, +; a**.**b được viết là ab, gọi là tích của a và b còn a+b gọi là tổng của a và b. Ta có thể biến đổi công thức, rút gọn công thức tương tự trong đại số lôgic. Ta cũng xét các tích sơ cấp và tổng sơ cấp tương tự "hội sơ cấp" và "tuyển sơ cấp". Mọi công thức đều có thể đưa về dạng tích chuẩn tắc hoàn toàn hoặc về dạng tổng chuẩn tắc hoàn toàn tương tự dạng "hội và tuyển chuẩn tắc hoàn toàn". Mỗi công thức trong đại số Boole cũng được gọi là biểu diễn một hàm Boole.

8.2. HÀM BOOLE.

8.2.1. Định nghĩa: Ký hiệu B = {0, 1} và $B^n = \{(x_1, x_2, \ldots, x_n) \mid x_i \in B, 1 \leq i \leq n\}$, ở đây B và B^n là các đại số Boole (xem 2) và 3) của Thí dụ 1). Biến x được gọi là một biến Boole nếu nó nhận các giá trị chỉ từ B. Một hàm từ B^n vào B được gọi là một hàm Boole (hay hàm đại số lôgic) bậc n.

Các hàm Boole cũng có thể được biểu diễn bằng cách dùng các biểu thức được tạo bởi các biến và các phép toán Boole (xem Bảng 1 trong Thí dụ 1). Các biểu thức Boole với các biến $x_1, x_2, \ldots, x_n$ được định nghĩa bằng đệ quy như sau:

- 0, 1, $x_1, x_2, \ldots, x_n$ là các biểu thức Boole.
- Nếu *P* và *Q* là các biểu thức Boole thì $\overline{P}$, *PQ* và *P*+*Q* cũng là các biểu thức Boole.

Mỗi một biểu thức Boole biểu diễn một hàm Boole. Các giá trị của hàm này nhận được bằng cách thay 0 và 1 cho các biến trong biểu thức đó.

Hai hàm n biến F và G được gọi là bằng nhau nếu $F(a_1, a_2, \ldots, a_n)=G(a_1, a_2, \ldots,a_n)$ với mọi $a_1, a_2, \ldots, a_n \in B$. Hai biểu thức Boole khác nhau biểu diễn cùng một hàm Boole được gọi là tương đương. Phần bù của hàm Boole *F* là hàm $\overline{F}$ với $\overline{F}(x_1, x_2, \ldots, x_n) = \overline{F(x_1, x_2, \ldots, x_n)}$. Giả sử F và G là các hàm Boole bậc n. Tổng Boole F+G và tích Boole FG được định nghĩa bởi:

$(F+G)(x_1, x_2, \ldots, x_n) = F(x_1, x_2, \ldots, x_n)+G(x_1, x_2, \ldots, x_n)$,

$(FG)(x_1, x_2, \ldots, x_n) = F(x_1, x_2, \ldots, x_n)G(x_1, x_2, \ldots, x_n)$.

Thí dụ 2:

Theo quy tắc nhân của phép đếm ta suy ra rằng có 2^n bộ n phần tử khác nhau gồm các số 0 và 1. Vì hàm Boole là việc gán 0 hoặc 1 cho mỗi bộ trong số 2^n bộ n phần tử đó, nên lại theo quy tắc nhân sẽ có 2^{2^n} các hàm Boole khác nhau.

Bậc	Số các hàm Boole
1	4
2	16
3	256
4	65.536
5	4.294.967.296
6	18.446.744.073.709.551.616

Bảng sau cho giá trị của 16 hàm Boole bậc 2 phân biệt:

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
0	0	0	1	0	0	0	1	1	1	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	1
1	0	0	1	0	1	1	0	0	0	1	1	1	0	1	1	0	0
1	1	0	1	1	1	0	1	1	0	0	1	1	1	0	0	0	0

trong đó có một số hàm thông dụng như sau:

- Hàm F_1 là hàm hằng 0,
- Hàm F_2 là hàm hằng 1,
- Hàm F_3 là hàm hội, $F_3(x,y)$ được viết là xy (hay $x \wedge y$),
- Hàm F_4 là hàm tuyển, $F_4(x,y)$ được viết là x+y (hay $x \vee y$),
- Hàm F_5 là hàm tuyển loại, $F_5(x,y)$ được viết là $x \oplus y$,
- Hàm F_6 là hàm kéo theo, $F_6(x,y)$ được viết là $x \Rightarrow y$,
- Hàm F_7 là hàm tương đương, $F_7(x,y)$ được viết là $x \Leftrightarrow y$,
- Hàm F_8 là hàm Vebb, $F_8(x,y)$ được viết là $x \downarrow y$,
- Hàm F_9 là hàm Sheffer, $F_9(x,y)$ được viết là $x \uparrow y$.

Thí dụ 3: Các giá trị của hàm Boole bậc 3 $F(x, y, z) = xy + \overline{z}$ được cho bởi bảng sau:

x	y	z	xy	$\overline{z}$	$F(x, y, z) = xy + \overline{z}$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

8.2.2. Định nghĩa: Cho x là một biến Boole và $\sigma \in$ B. Ký hiệu:

$$x^{\sigma} = \begin{cases} x \; khi \; \sigma = 1, \\ \overline{x} \; khi \; \sigma = 0. \end{cases}$$

Dễ thấy rằng $x^{\sigma} = 1 \Leftrightarrow x = \sigma$. Với mỗi hàm Boole *F* bậc n, ký hiệu:

$$T_F = \{(x_1, x_2, \ldots, x_n) \in B^n \mid F(x_1, x_2, \ldots, x_n) = 1\}$$

Và gọi nó là tập đặc trưng của hàm *F*. Khi đó ta có:

$$T_{\overline{F}} = \overline{T_F}, \; T_{F+G} = T_F \cup T_G, \; T_{FG} = T_F \cap T_G.$$

Cho n biến Boole $x_1, x_2, \ldots, x_n$. Một biểu thức dạng:

$$x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} \ldots x_{i_k}^{\sigma_k}$$

trong đó $\sigma_1, \sigma_2, \ldots, \sigma_k \in$ B, $1 \le i_1 < i_2 < \cdots < i_k \le n$ được gọi là một hội sơ cấp của n biến $x_1, x_2, \ldots, x_n$. Số các biến xuất hiện trong một hội sơ cấp đựoc gọi là hạng của của hội sơ cấp đó.

Cho F là một hàm Boole bậc n. Nếu F được biểu diễn dưới dạng tổng (tuyển) của một số hội sơ cấp khác nhau của n biến thì biểu diễn đó được gọi là dạng tổng (tuyển) chuẩn tắc của F. Dạng tổng (tuyển) chuẩn tắc hoàn toàn là dạng chuẩn tắc duy nhất của F mà trong đó các hội sơ cấp đều có hạng n.

Thí dụ 4: $\overline{x}y + x\overline{y}$ là một dạng tổng chuẩn tắc của hàm $x \oplus y$.

$\overline{x} + \overline{y}$ và $\overline{x}\,\overline{y} + \overline{x}y + x\overline{y}$ là các dạng tổng chuẩn tắc của hàm Sheffer $x \uparrow y$.

8.2.3. Mệnh đề: Mọi hàm Boole *F* bậc n đều có thể biểu diễn dưới dạng:

$$F(x_1, x_2, \ldots, x_n) = \sum_{(\sigma_1, \ldots, \sigma_n) \in B^i} x_1^{\sigma_1} \ldots x_i^{\sigma_i} F(\sigma_1, \ldots, \sigma_i, x_{i+1}, \ldots, x_n) \quad (1),$$

trong đó i là số tự nhiên bất kỳ, $1 \le i \le n$.

Chứng minh: Gọi *G* là hàm Boole ở vế phải của (1). Cho $(x_1, x_2, \ldots, x_n) \in T_F$. Khi đó số hạng ứng với bộ giá trị $\sigma_1 = x_1, \ldots, \sigma_i = x_i$ trong tổng ở vế phải của (1) bằng 1, do đó $(x_1, x_2, \ldots, x_n) \in T_G$. Đảo lại, nếu $(x_1, x_2, \ldots, x_n) \in T_G$ tức là vế phải bằng 1 thì phải xảy ra bằng 1 tại một số hạng nào đó, chẳng hạn tại số hạng ứng với bộ giá trị $(\sigma_1, \ldots, \sigma_i)$, khi đó $x_1 = \sigma_1, \ldots, x_i = \sigma_i$ và $f(\sigma_1, \ldots, \sigma_i, x_{i+1}, \ldots, x_n) = 1$ hay $(x_1, x_2, \ldots, x_n) \in T_F$. Vậy $T_F = T_G$ hay $F = G$.

Cho i=1 trong mệnh đề trên và nhận xét rằng vai trò của các biến x_i là như nhau, ta được hệ quả sau.

8.2.4. Hệ quả: Mọi hàm Boole *F* bậc n đều có thể được khai triển theo một biến x_i:

$$F(x_1, \ldots, x_n) = \overline{x_i} F(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) + x_i F(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n).$$

Cho i=n trong mệnh đề trên và bỏ đi các nhân tử bằng 1 trong tích, các số hạng bằng 0 trong tổng, ta được hệ quả sau.

8.2.5. Hệ quả: Mọi hàm Boole *F* bậc n đều có thể được khai triển dưới dạng:

$$F(x_1, \ldots, x_n) = \sum_{(\sigma_1, \ldots, \sigma_n) \in T_F} x_1^{\sigma_1} \ldots x_n^{\sigma_n}.$$

8.2.6. Chú ý: Từ Hệ quả 8.2.5, ta suy ra rằng mọi hàm Boole đều có thể biểu diễn dưới dạng tổng (tuyển) chuẩn tắc hoàn toàn. Như vậy mọi hàm Boole đều có thể biểu diễn bằng một biểu thức Boole chỉ chứa ba phép tích (hội), tổng (tuyển), bù (phủ định). Ta nói rằng hệ {tích, tổng, bù} là đầy đủ.

Bằng đối ngẫu, ta có thể chứng minh một kết quả tương tự bằng việc thay tích bởi tổng và ngược lại, từ đó dẫn tới việc biểu diễn *F* qua một tích các tổng. Biểu diễn này được gọi là dạng tích (hội) chuẩn tắc hoàn toàn của *F*:

$$F(x_1,\ldots,x_n) = \prod_{(\sigma_1,\ldots,\sigma_n)\in\overline{T_F}} (x_1^{\overline{\sigma_1}} + \ldots + x_n^{\overline{\sigma_n}})$$

Thí dụ 5: Dạng tổng chuẩn tắc hoàn toàn của hàm F cho trong Thí dụ 3 là:

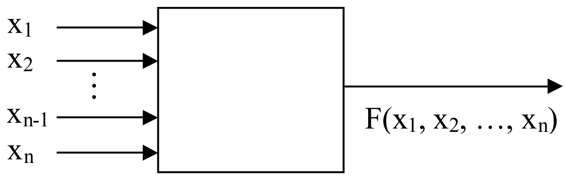
$$F(x,y,z) = \bar{x}\bar{y}\bar{z} + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xy\bar{z} + xyz,$$

và dạng tích chuẩn tắc hoàn toàn của nó là:

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

8.3. MẠCH LÔGIC.

8.3.1. Cổng lôgic:



Xét một thiết bị như hình trên, có một số đường vào (dẫn tín hiệu vào) và chỉ có một đường ra (phát tín hiệu ra). Giả sử các tín hiệu vào $x_1, x_2, \ldots, x_n$ (ta gọi là đầu vào hay input) cũng như tín hiệu ra F (đầu ra hay output) đều chỉ có hai trạng thái khác nhau, tức là mang một bit thông tin, mà ta ký hiệu là 0 và 1.

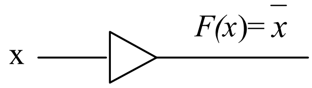
Ta gọi một thiết bị với các đầu vào và đầu ra mang giá trị 0, 1 như vậy là một mạch lôgic.

Đầu ra của một mạch lôgic là một hàm Boole F của các đầu vào $x_1, x_2, \ldots, x_n$. Ta nói mạch lôgic trong hình trên thực hiện hàm F.

Các mạch lôgic được tạo thành từ một số mạch cơ sở, gọi là cổng lôgic. Các cổng lôgic sau đây thực hiện các hàm phủ định, hội và tuyển.

1. Cổng NOT: Cổng NOT thực hiện hàm phủ định. Cổng chỉ có một đầu vào. Đầu ra *F(x)* là phủ định của đầu vào *x*.

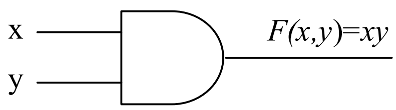
$$F(x) = \bar{x} = \begin{cases} 0 \ khi = 1, \\ 1 \ khi \ x = 0. \end{cases}$$

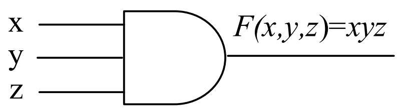


Chẳng hạn, xâu bit 100101011 qua cổng NOT cho xâu bit 011010100.

2. Cổng AND: Cổng AND thực hiện hàm hội. Đầu ra *F(x,y)* là hội (tích) của các đầu vào.

$$F(x,y) = xy = \begin{cases} 1 \ khi \ x = y = 1, \\ 0 \ \ trong\ các\ trường\ hợp\ khác. \end{cases}$$

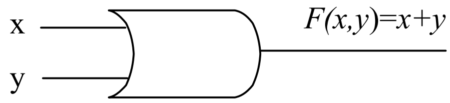


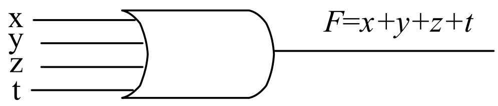


Chẳng hạn, hai xâu bit 101001101 và 111010110 qua cổng AND cho 101000100.

3. Cổng OR: Cổng OR thực hiện hàm tuyển (tổng). Đầu ra *F(x,y)* là tuyển (tổng) của các đầu vào.

$$F(x,y) = x + y = \begin{cases} 1 \; khi \; x = 1 \; hay \; y = 1, \\ 0 \; khi \; x = y = 0. \end{cases}$$





Chẳng hạn, hai xâu bit 101001101 và 111010100 qua cổng OR cho 111011101.

8.3.2. Mạch lôgic:

1. Tổ hợp các cổng: Các cổng lôgic có thể lắp ghép để được những mạch lôgic thực hiện các hàm Boole phức tạp hơn. Như ta đã biết rằng một hàm Boole bất kỳ có thể biểu diễn bằng một biểu thức chỉ chứa các phép −, ., +. Từ đó suy ra rằng có thể lắp ghép thích hợp các cổng NOT, AND, OR để được một mạch lôgic thực hiện một hàm Boole bất kỳ.

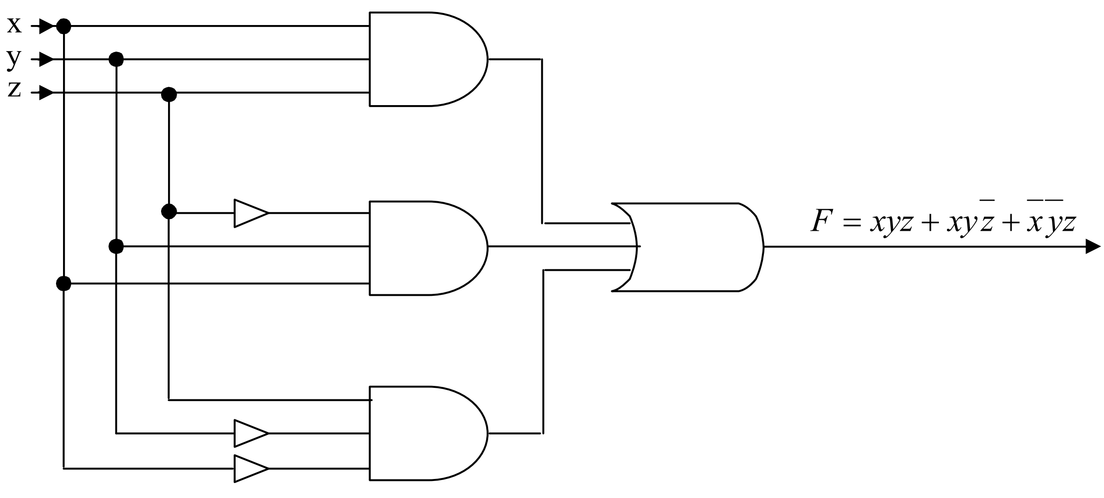
Thí dụ 6: Xây dựng một mạch lôgic thực hiện hàm Boole cho bởi bảng sau.

x	y	z	F(x,y,z)
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

Theo bảng này, hàm F có dạng tổng (tuyển) chuẩn tắc hoàn toàn là:

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

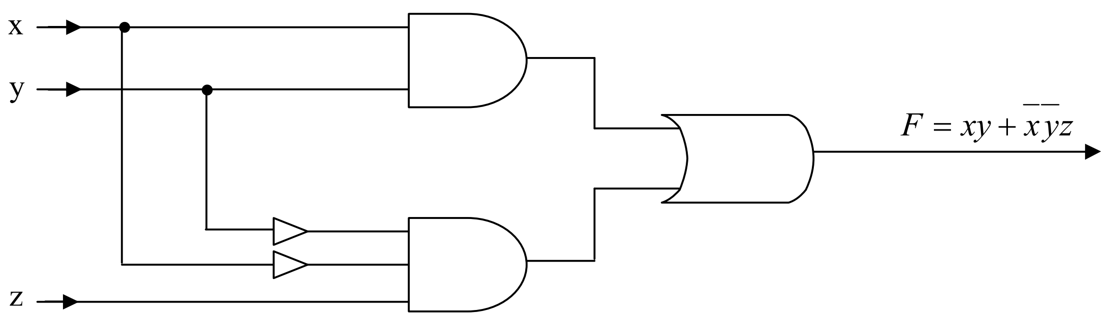
Hình dưới đây vẽ mạch lôgic thực hiện hàm F đã cho.



Biểu thức của *F(x, y, z)* có thể rút gọn:

$xyz + xy\bar{z} + \bar{x}\bar{y}z = xy(z+\bar{z}) + \bar{x}\bar{y}z = xy + \bar{x}\bar{y}z$.

Hình dưới đây cho ta mạch lôgic thực hiện hàm $xy + \bar{x}\bar{y}z$.



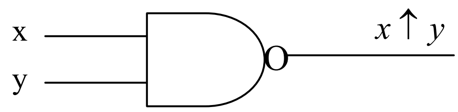
Hai mạch lôgic trong hai hình trên thực hiện cùng một hàm Boole, ta nói đó là hai mạch lôgic tương đương, nhưng mạch lôgic thứ hai đơn giản hơn.

Vấn đề tìm mạch lôgic đơn giản thực hiện một hàm Boole F cho trước gắn liền với vấn đề tìm biểu thức đơn giản nhất biểu diễn hàm ấy. Đây là vấn đề khó và lý thú, tuy ý nghĩa thực tiễn của nó không còn như mấy chục năm về trước.

Ta vừa xét việc thực hiện một hàm Boole bất kỳ bằng một mạch lôgic chỉ gồm các cổng NOT, AND, OR.

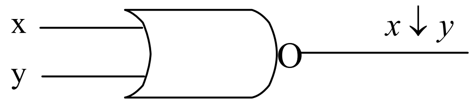
Dựa vào đẳng thức $x + y = \overline{\bar{x}.\bar{y}}$ cũng như $xy = \overline{\bar{x} + \bar{y}}$, cho ta biết hệ {**.**, −} và hệ {+, −} cũng là các hệ đầy đủ. Do đó có thể thực hiện một hàm Boole bất kỳ bằng một mạch lôgic chỉ gồm có các cổng NOT, AND hoặc NOT, OR.

Xét hàm Sheffer $F(x,y) = x \uparrow y = \begin{cases} 0 \ khi \ x = y = 1, \\ 1 \ khi \ x = 0 \ hay \ y = 0. \end{cases}$ Mạch lôgic thực hiện hàm $\uparrow$ gọi là cổng NAND, được vẽ như hình dưới đây.



Dựa vào các đẳng thức $\bar{x} = x \uparrow x, xy = (x \uparrow y) \uparrow (x \uparrow y), x + y = (x \uparrow x) \uparrow (y \uparrow y)$, cho ta biết hệ $\{\uparrow\}$ là đầy đủ, nên bất kỳ một hàm Boole nào cũng có thể thực hiện được bằng một mạch lôgic chỉ gồm có cổng NAND.

Xét hàm Vebb $F(x,y) = x \downarrow y = \begin{cases} 0 \ khi \ x = 1 \ hay \ y = 1, \\ 1 \ khi \ x = y = 0. \end{cases}$ Mạch lôgic thực hiện hàm $\downarrow$ gọi là cổng NOR, được vẽ như hình dưới đây.

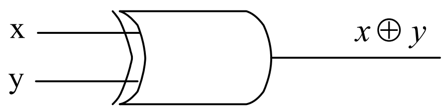


Tương tự hệ $\{\downarrow\}$ là đầy đủ nên bất kỳ hàm Boole nào cũng có thể thực hiện được bằng một mạch lôgic chỉ gồm có cổng NOR.

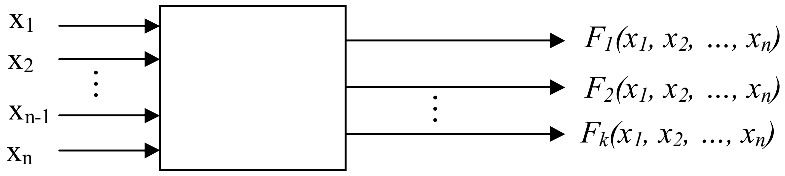
Một phép toán lôgic quan trọng khác là phép tuyển loại:

$$F(x,y) = x \oplus y = \begin{cases} 0 \; khi \; x = y, \\ 1 \; khi \; x \neq y. \end{cases}$$

Mạch lôgic này là một cổng lôgic, gọi là cổng XOR, được vẽ như hình dưới đây.



2. Mạch cộng: Nhiều bài toán đòi hỏi phải xây dựng những mạch lôgic có nhiều đường ra, cho các đầu ra $F_1, F_2, \ldots, F_k$ là các hàm Boole của các đầu vào $x_1, x_2, \ldots, x_n$.

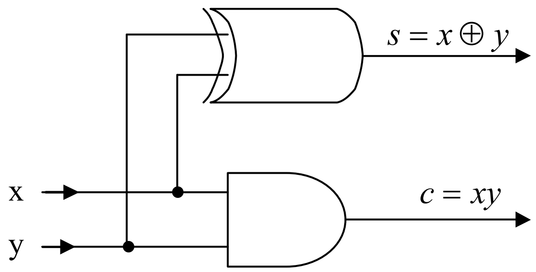


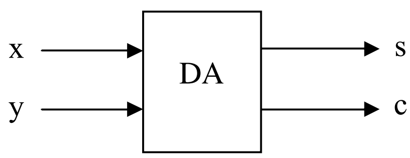
Chẳng hạn, ta xét phép cộng hai số tự nhiên từ các khai triển nhị phân của chúng. Trước hết, ta sẽ xây dựng một mạch có thể duợc dùng để tìm x+y với x, y là hai số 1-bit. Đầu vào mạch này sẽ là x và y. Đầu ra sẽ là một số 2-bit $\overline{cs}$, trong đó s là bit tổng và c là bit nhớ.

0+0 = 00
0+1 = 01
1+0 = 01
1+1 = 10

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Từ bảng trên, ta thấy ngay $s = x \oplus y, c = xy$. Ta vẽ được mạch thực hiện hai hàm $s = x \oplus y$ và $c = xy$ như hình dưới đây. Mạch này gọi là mạch cộng hai số 1-bit hay mạch cộng bán phần, ký hiệu là DA.





Xét phép cộng hai số 2-bit $\overline{a_2 a_1}$ và $\overline{b_2 b_1}$,

$$\begin{array}{c} a_2 \; a_1 \\ \underline{b_2 \; b_1} \end{array}$$

Thực hiện phép cộng theo từng cột, ở cột thứ nhất (từ phải sang trái) ta tính $a_1 + b_1$ được bit tổng s_1 và bit nhớ c_1; ở cột thứ hai, ta tính $a_2 + b_2 + c_1$, tức là phải cộng ba số 1-bit.

Cho *x, y, z* là ba số 1-bit. Tổng *x+y+z* là một số 2-bit $\overline{cs}$, trong đó *s* là bit tổng của *x+y+z* và *c* là bit nhớ của *x+y+z*. Các hàm Boole *s* và *c* theo các biến *x, y, z* được xác định bằng bảng sau:

x	y	z	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Từ bảng này, dễ dàng thấy rằng:

$$s = x \oplus y \oplus z .$$

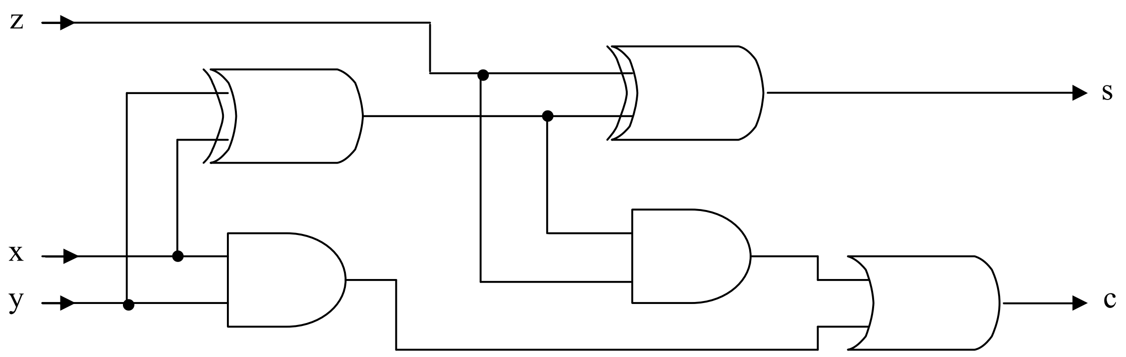
Hàm c có thể viết dưới dạng tổng chuẩn tắc hoàn toàn là:

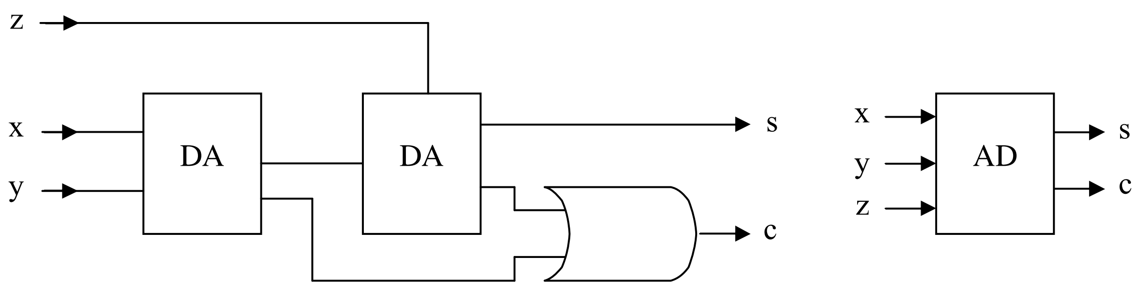
$$c = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz .$$

Công thức của *c* có thể rút gọn:

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

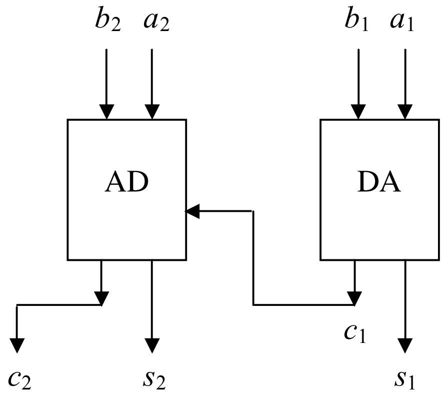
Ta vẽ được mạch thực hiện hai hàm Boole $s = x \oplus y \oplus z$ và $c = z(x \oplus y) + xy$ như hình dưới đây, mạch này là ghép nối của hai mạch cộng bán phần (DA) và một cổng OR. Đây là mạch cộng ba số 1-bit hay mạch cộng toàn phần, ký hiệu là AD.



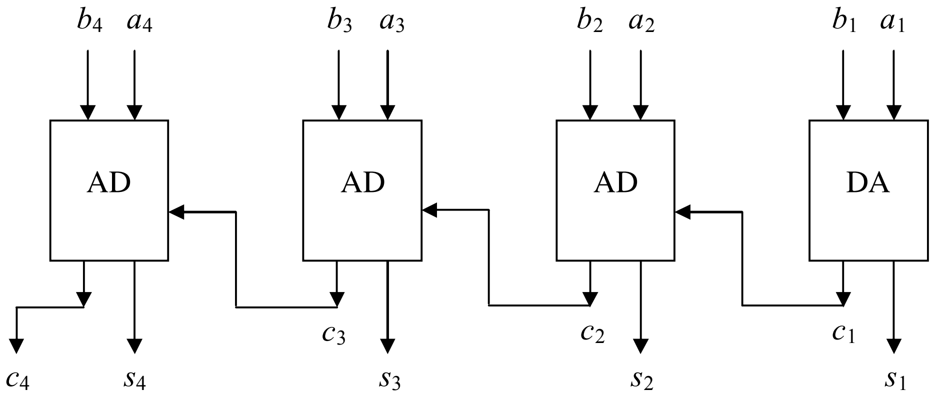


Trở lại phép cộng hai số 2-bit $\overline{a_2a_1}$ và $\overline{b_2b_1}$. Tổng $\overline{a_2a_1}+\overline{b_2b_1}$ là một số 3-bit $\overline{c_2s_2s_1}$, trong đó s_1 là bit tổng của a_1+b_1: $s_1 = a_1 \oplus b_1$, s_2 là bit tổng của a_2+b_2+c_1, với c_1 là bit nhớ của a_1+b_1: $s_2 = a_2 \oplus b_2 \oplus c_1$ và c_2 là bit nhớ của a_2+b_2+c_1.

Ta có được mạch thực hiện ba hàm Boole s_1, s_2, c_2 như hình dưới đây.



Dễ dàng suy ra mạch cộng hai số n-bit, với n là một số nguyên dương bất kỳ. Hình sau cho một mạch cộng hai số 4-bit.



8.4. CỰC TIỂU HOÁ CÁC MẠCH LÔGIC.

Hiệu quả của một mạch tổ hợp phụ thuộc vào số các cổng và sự bố trí các cổng đó. Quá trình thiết kế một mạch tổ hợp được bắt đầu bằng một bảng chỉ rõ các giá trị đầu ra đối với mỗi một tổ hợp các giá trị đầu vào. Ta luôn luôn có thể sử dụng khai triển tổng các tích của mạch để tìm tập các cổng lôgic thực hiện mạch đó. Tuy nhiên,khai triển tổng các tích có thể chứa các số hạng nhiều hơn mức cần thiết. Các số hạng trong khai triển tổng các tích chỉ khác nhau ở một biến, sao cho trong số hạng này xuất hiện biến đó và trong số hạng kia xuất hiện phần bù của nó, đều có thể được tổ hợp lại. Chẳng hạn, xét mạch có đầu ra bằng 1 khi và chỉ khi $x = y = z = 1$ hoặc $x = z = 1$ và $y = 0$. Khai triển tổng các tích của mạch này là $xyz + x\overline{y}z$. Hai tích trong khai triển này chỉ khác nhau ở một biến, đó là biến y. Ta có thể tổ hợp lại như sau:

$$xyz + x\overline{y}z = (y + \overline{y})xz = 1xz = xz .$$

Do đó xz là biểu thức với ít phép toán hơn biểu diễn mạch đã cho. Mạch thứ hai chỉ dùng một cổng, trong khi mạch thứ nhất phải dùng ba cổng và một bộ đảo (cổng NOT).

8.4.1. Bản đồ Karnaugh:

Để làm giảm số các số hạng trong một biểu thức Boole biểu diễn một mạch, ta cần phải tìm các số hạng để tổ hợp lại. Có một phương pháp đồ thị, gọi là bản đồ Karnaugh, được dùng để tìm các số hạng tổ hợp được đối với các hàm Boole có số biến tương đối nhỏ. Phương pháp mà ta mô tả dưới đây đã được Maurice Karnaugh đưa ra vào năm 1953. Phương pháp này dựa trên một công trình trước đó của E.W. Veitch. Các bản đồ Karnaugh cho ta một phương pháp trực quan để rút gọn các khai triển tổng các tích, nhưng chúng không thích hợp với việc cơ khí hoá quá trình này. Trước hết, ta sẽ minh hoạ cách dùng các bản đồ Karnaugh để rút gọn biểu thức của các hàm Boole hai biến.

Có bốn hội sơ cấp khác nhau trong khai triển tổng các tích của một hàm Boole có hai biến x và y. Một bản đồ Karnaugh đối với một hàm Boole hai biến này gồm bốn ô vuông, trong đó hình vuông biểu diễn hội sơ cấp có mặt trong khai triển được ghi số 1. Các hình ô được gọi là kề nhau nếu các hội sơ cấp mà chúng biểu diễn chỉ khác nhau một biến.

	y	$\overline{y}$
x	xy	$x\overline{y}$
$\overline{x}$	$\overline{x}y$	$\overline{x}\,\overline{y}$

Thí dụ 7: Tìm các bản đồ Karnaugh cho các biểu thức:

a) $xy+\overline{x}y$ b) $x\overline{y}+\overline{x}y$ c) $x\overline{y}+\overline{x}y+\overline{x}\,\overline{y}$

và rút gọn chúng.

Ta ghi số 1 vào ô vuông khi hội sơ cấp được biểu diễn bởi ô đó có mặt trong khai triển tổng các tích. Ba bản đồ Karnaugh được cho trên hình sau.

	y	
x	1	
$\overline{x}$	1	

	1
1	

	y	$\overline{y}$
x		1
$\overline{x}$	1	1

Việc nhóm các hội sơ cấp được chỉ ra trong hình trên bằng cách sử dụng bản đồ Karnaugh cho các khai triển đó. Khai triển cực tiểu của tổng các tích này tương ứng là:

a) y, b) $x\overline{y}+\overline{x}y$, c) $\overline{x}+\overline{y}$.

Bản đồ Karnaugh ba biến là một hình chữ nhật được chia thành tám ô. Các ô đó biểu diễn tám hội sơ cấp có được. Hai ô được gọi là kề nhau nếu các hội sơ cấp mà chúng biểu diễn chỉ khác nhau một biến. Một trong các cách để lập bản đồ Karnaugh ba biến được cho trong hình bên.

	yz	$y\overline{z}$	$\overline{y}\,\overline{z}$	$\overline{y}z$
x	xyz	$xy\overline{z}$	$x\overline{y}\,\overline{z}$	$x\overline{y}z$
$\overline{x}$	$\overline{x}yz$	$\overline{x}y\overline{z}$	$\overline{x}\,\overline{y}\,\overline{z}$	$\overline{x}\,\overline{y}z$

Để rút gọn khai triển tổng các tích ba biến, ta sẽ dùng bản đồ Karnaugh để nhận dạng các hội sơ cấp có thể tổ hợp lại. Các khối gồm hai ô kề nhau biểu diễn cặp các hội sơ cấp có thể được tổ hợp lại thành một tích của hai biến; các khối 2 x 2 và 4 x 1 biểu diễn các hội sơ cấp có thể tổ hợp lại thành một biến duy nhất; còn khối gồm tất cả tám ô biểu diễn một tích không có một biến nào, cụ thể đây là biểu thức 1.

Thí dụ 8: Dùng các bản đồ Karnaugh ba biến để rút gọn các khai triển tổng các tích sau:

a) $xy\bar{z} + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}\bar{z}$,

b) $x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$,

c) $xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$.

Bản đồ Karnaugh cho những khai triển tổng các tích này được cho trong hình sau:

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1	1	
$\bar{x}$	1		1	

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x			1	1
$\bar{x}$	1		1	1

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	1	1	1	1
$\bar{x}$	1		1	1

Việc nhóm thành các khối cho thấy rằng các khai triển cực tiểu thành các tổng Boole của các tích Boole là:

a) $x\bar{z} + \bar{y}\bar{z} + \bar{x}yz$, b) $\bar{y} + \bar{x}z$, c) $x + \bar{y} + z$.

Bản đồ Karnaugh bốn biến là một hình vuông được chia làm 16 ô. Các ô này biểu diễn 16 hội sơ cấp có được. Một trong những cách lập bản đồ Karnaugh bốn biến được cho trong hình dưới đây.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	$wxyz$	$wxy\bar{z}$	$wx\bar{y}\bar{z}$	$wx\bar{y}z$
$w\bar{x}$	$w\bar{x}yz$	$w\bar{x}y\bar{z}$	$w\bar{x}\bar{y}\bar{z}$	$w\bar{x}\bar{y}z$
$\bar{w}\bar{x}$	$\bar{w}\bar{x}yz$	$\bar{w}\bar{x}y\bar{z}$	$\bar{w}\bar{x}\bar{y}\bar{z}$	$\bar{w}\bar{x}\bar{y}z$
$\bar{w}x$	$\bar{w}xyz$	$\bar{w}xy\bar{z}$	$\bar{w}x\bar{y}\bar{z}$	$\bar{w}x\bar{y}z$

Hai ô được gọi là kề nhau nếu các hội sơ cấp mà chúng biểu diễn chỉ khác nhau một biến. Do đó, mỗi một ô kề với bốn ô khác. Sự rút gọn một khai triển tổng các tích bốn biến được thực hiện bằng cách nhận dạng các khối gồm 2, 4, 8 hoặc 16 ô biểu diễn các hội sơ cấp có thể tổ hợp lại được. Mỗi ô biểu diễn một hội sơ cấp hoặc được dùng để lập một tích có ít biến hơn hoặc được đưa vào trong khai triển. Cũng như trong trường hợp bản đồ Karnaugh hai và ba biến, mục tiêu là cần phải nhận dạng các khối lớn nhất có chứa các số 1 bằng cách dùng một số ít nhất các khối, mà trước hết là các khối lớn nhất.

8.4.2. Phương pháp Quine-McCluskey:

8.4.2.1. Mở đầu: Ta đã thấy rằng các bản đồ Karnaugh có thể được dùng để tạo biểu thức cực tiểu của các hàm Boole như tổng của các tích Boole. Tuy nhiên, các bản đồ Karnaugh sẽ rất khó dùng khi số biến lớn hơn bốn. Hơn nữa, việc dùng các bản đồ Karnaugh lại dựa trên việc rà soát trực quan để nhận dạng các số hạng cần được nhóm lại. Vì những nguyên nhân đó, cần phải có một thủ tục rút gọn những khai triển tổng các tích có thể cơ khí hoá được. Phương pháp Quine-McCluskey là một thủ tục như vậy. Nó có thể được dùng cho các hàm Boole có số biến bất kỳ. Phương pháp này được W.V. Quine và E.J. McCluskey phát triển vào những năm 1950. Về cơ bản, phương pháp Quine-McCluskey có hai phần. Phần đầu là tìm các số hạng là ứng viên để đưa vào khai triển cực tiểu như một tổng các tích Boole mà ta gọi là các nguyên nhân nguyên tố. Phần thứ hai là xác định xem trong số các ứng viên đó, các số hạng nào là thực sự dùng được.

8.4.2.2. Định nghĩa: Cho hai hàm Boole F và G bậc n. Ta nói G là một nguyên nhân của F nếu $T_G \subset T_F$, nghĩa là $G \Rightarrow F$ là một hằng đúng.

Dễ thấy rằng mỗi hội sơ cấp trong một dạng tổng chuẩn tắc của F là một nguyên nhân của F. Hội sơ cấp A của F được gọi là một nguyên nhân nguyên tố của F nếu trong A xoá đi một biến thì hội nhận được không còn là nguyên nhân của F.

Nếu $F_1, \ldots, F_k$ là các nguyên nhân của F thì $T_{F_i} \subset T_F$, $1 \le i \le k$. Khi đó $T_{\sum_{i=1}^{k} F_i} = \bigcup_{i=1}^{k} T_{F_i} \subset T_F$. Do đó $\sum_{i=1}^{k} F_i$ là một nguyên nhân của F.

Cho S là một hệ các nguyên nhân của F. Ta nói rằng hệ S là đầy đủ đối với F nếu $F = \sum_{G \in S} G$, nghĩa là $T_F = \bigcup_{G \in S} T_G$.

8.4.2.3. Mệnh đề: Hệ các nguyên nhân nguyên tố của hàm F là một hệ đầy đủ.

Chứng minh: Gọi S là hệ các nguyên nhân nguyên tố của F. Ta có $T_G \subset T_F, \forall g \in S$, Nên $T_{\sum_{G \in S} G} = \bigcup_{G \in S} T_G \subset T_F$. Giả sử dạng tổng chuẩn tắc hoàn toàn của F là $F = \sum_{G' \in S'} G'$ nên $T_F = \bigcup_{G' \in S'} T_{G'}$.

Xét $G' \in S'$, nếu G' không phải là nguyên nhân nguyên tố của F thì bằng cách xoá bớt một số biến trong G' ta thu được nguyên nhân nguyên tố G của F. Khi đó $T_{G'} \subset T_G$ và $\bigcup_{G' \in S'} T_{G'} \subset \bigcup_{G \in S} T_G$ hay $T_F \subset \bigcup_{G \in S} T_G$. Vì vậy $T_F = \bigcup_{G \in S} T_G$ hay $F = \sum_{G \in S} G$.

Dạng tổng chuẩn tắc $F = \sum_{G \in S} G$ được gọi là dạng tổng chuẩn tắc thu gọn của F.

8.4.2.4. Phương pháp Quine-McCluskey tìm dạng tổng chuẩn tắc thu gọn:

Giả sử F là một hàm Boole n biến $x_1, x_2, \ldots, x_n$. Mỗi hội sơ cấp của n biến đó được biểu diễn bằng một dãy n ký hiệu trong bảng {0, 1, −} theo quy ước: ký tự thứ i là 1 hay 0 nếu x_i có mặt trong hội sơ cấp là bình thường hay với dấu phủ định, còn nếu x_i không có mặt thì ký tự này là −. Chẳng hạn, hội sơ cấp của 6 biến $x_1, \ldots, x_6$ là $\overline{x_1}x_3x_4\overline{x_6}$ được biểu diễn bởi 0−11−0. Hai hội sơ cấp được gọi là kề nhau nếu các biểu diễn nói trên của chúng chỉ khác nhau ở một vị trí 0, 1. Rõ ràng các hội sơ cấp chỉ có thể dán được với nhau bằng phép dán $\mathrm{A}x + \mathrm{A}\overline{x} = A$ nếu chúng là kề nhau.

Thuật toán được tiến hành như sau: Lập một bảng gồm nhiều cột để ghi các kết quả dán. Sau đó lần lượt thực hiện các bước sau:

Bước 1: Viết vào cột thứ nhất các biểu diễn của các nguyên nhân hạng n của hàm Boole F. Các biểu diễn được chia thành từng nhóm, các biểu diễn trong mỗi nhóm có số các ký hiệu 1 bằng nhau và các nhóm xếp theo thứ tự số các ký hiệu 1 tăng dần.

Bước 2: Lần lượt thực hiện tất cả các phép dán các biểu diễn trong nhóm i với các biểu diễn trong nhóm i+1 (i=1, 2, …). Biểu diễn nào tham gia ít nhất một phép dán sẽ được ghi nhận một dấu * bên cạnh. Kết quả dán được ghi vào cột tiếp theo.

Bước 3: Lặp lại Bước 2 cho cột kế tiếp cho đến khi không thu thêm được cột nào mới. Khi đó tất cả các biểu diễn không có dấu * sẽ cho ta tất cả các nguyên nhân nguyên tố của F.

Thí dụ 9: Tìm dạng tổng chuẩn tắc thu gọn của các hàm Boole:

$$F_1 = \overline{w}\,\overline{x}\,\overline{y}z + \overline{w}x\overline{y}z + \overline{w}\,\overline{x}yz + wx\overline{y}\,\overline{z} + w\overline{x}yz + \overline{w}xyz + wxyz\,,$$

$$F_2 = \overline{w}\,\overline{x}y\overline{z} + \overline{w}\,\overline{x}yz + w\overline{x}yz + wx\overline{y}\,\overline{z} + wx\overline{y}z + wxy\overline{z} + wxyz\,.$$

0 0 0 1 *	0 − 0 1 *	0 − − 1
0 1 0 1 *	0 0 − 1 *	− 0 − 1
0 0 1 1 *	− 0 0 1 *	− − 1 1
1 0 0 1 *	− 0 1 1 *	
1 0 1 1 *	1 0 − 1 *	
0 1 1 1 *	0 1 − 1 *	
1 1 1 1 *	0 − 1 1 *	
	1 − 1 1 *	
	− 1 1 1 *	

0 0 1 0 *	0 0 1 −	1 1 − −
0 0 1 1 *	− 0 1 1	
1 1 0 0 *	1 1 0 − *	
1 0 1 1 *	1 1 − 0 *	
1 1 0 1 *	1 − 1 1	
1 1 1 0 *	1 1 − 1 *	
1 1 1 1 *	1 1 1 − *	

Từ các bảng trên ta có dạng tổng chuẩn tắc thu gọn của F_1 và F_2 là:

$$F_1 = \overline{w}z + \overline{x}z + yz\,,$$

$$F_2 = \overline{w}\,\overline{x}y + \overline{x}yz + wyz + wx\,.$$

8.4.2.5. Phương pháp Quine-McCluskey tìm dạng tổng chuẩn tắc tối thiểu:

Sau khi tìm được dạng tổng chuẩn tắc thu gọn của hàm Boole *F*, nghĩa là tìm được tất cả các nguyên nhân nguyên tố của nó, ta tiếp tục phương pháp Quine-McCluskey tìm dạng tổng chuẩn tắc tối thiểu (cực tiểu) của *F* như sau.

Lập một bảng chữ nhật, mỗi cột ứng với một cấu tạo đơn vị của *F* (mỗi cấu tạo đơn vị là một hội sơ cấp hạng n trong dạng tổng chuẩn tắc hoàn toàn của *F*) và mỗi dòng ứng với một nguyên nhân nguyên tố của *F*. Tại ô (i, j), ta đánh dấu cộng (+) nếu nguyên nhân nguyên tố ở dòng i là một phần con của cấu tạo đơn vị ở cột j. Ta cũng nói rằng khi đó nguyên nhân nguyên tố i là phủ cấu tạo đơn vị j. Một hệ S các nguyên nhân nguyên tố của *F* được gọi là phủ hàm *F* nếu mọi cấu tạo đơn vị của *F* đều được phủ ít nhất bởi một thành viên của hệ. Dễ thấy rằng nếu hệ S là phủ hàm *F* thì nó là đầy đủ, nghĩa là tổng của các thành viên trong S là bằng *F*.

Một nguyên nhân nguyên tố được gọi là cốt yếu nếu thiếu nó thì một hệ các nguyên nhân nguyên tố không thể phủ hàm *F*. Các nguyên nhân nguyên tố cốt yếu được tìm như sau: tại những cột chỉ có duy nhất một dấu +, xem dấu + đó thuộc dòng nào thì dòng đó ứng với một nguyên nhân nguyên tố cốt yếu.

Việc lựa chọn các nguyên nhân nguyên tố trên bảng đã đánh dấu, để được một dạng tổng chuẩn tắc tối thiểu, có thể tiến hành theo các bước sau.

Bước 1: Phát hiện tất cả các nguyên nhân nguyên tố cốt yếu.

Bước 2: Xoá tất cả các cột được phủ bởi các nguyên nhân nguyên tố cốt yếu.

Bước 3: Trong bảng còn lại, xoá nốt những dòng không còn dấu + và sau đó nếu có hai cột giống nhau thì xoá bớt một cột.

Bước 4: Sau các bước trên, tìm một hệ S các nguyên nhân nguyên tố với số biến ít nhất phủ các cột còn lại.

Tổng của các nguyên nhân nguyên tố cốt yếu và các nguyên nhân nguyên tố trong hệ S sẽ là dạng tổng chuẩn tắc tối thiểu của hàm *F*.

Các bước 1, 2, 3 có tác dụng rút gọn bảng trước khi lựa chọn. Độ phức tạp chủ yếu nằm ở Bước 4. Tình huống tốt nhất là mọi nguyên nhân nguyên tố đều là cốt yếu. Trường hợp này không phải lựa chọn gì và hàm *F* có duy nhất một dạng tổng chuẩn tắc tối thiểu cũng chính là dạng tổng chuẩn tắc thu gọn. Tình huống xấu nhất là không có nguyên nhân nguyên tố nào là cốt yếu. Trường hợp này ta phải lựa chọn toàn bộ bảng.

Thí dụ 10: Tìm dạng tổng chuẩn tắc tối thiểu của các hàm Boole cho trong Thí dụ 9.

	$\bar{w}\bar{x}\bar{y}z$	$\bar{w}x\bar{y}z$	$\bar{w}\bar{x}yz$	$w\bar{x}\bar{y}z$	$w\bar{x}yz$	$\bar{w}xyz$	$wxyz$
$\bar{w}z$	+	+	+				
$\bar{x}z$	+		+	+	+		
yz			+		+	+	+

Các nguyên nhân nguyên tố đều là cốt yếu nên dạng tổng chuẩn tắc tối thiểu của F_1 là:

$$F_1 = \overline{w}z + \overline{x}z + yz$$

	$\overline{w}\,\overline{x}y\overline{z}$	$\overline{w}\,\overline{x}yz$	$\overline{w}xyz$	$wx\overline{y}\,\overline{z}$	$wx\overline{y}z$	$wxy\overline{z}$	$wxyz$
wx				+	+	+	+
$\overline{w}\,\overline{x}y$	+	+					
$\overline{x}yz$		+	+				
wyz			+				+

Các nguyên nhân nguyên tố cốt yếu nằm ở dòng 1 và 2. Sau khi rút gọn, bảng còn dòng 3, 4 và một cột 3. Việc chọn S khá đơn giản: có thể chọn một trong hai nguyên nhân nguyên tố còn lại. Vì vậy ta được hai dạng tổng chuẩn tắc tối thiểu là:

$$F_2 = wx + \overline{w}\,\overline{x}y + \overline{x}yz,$$

$$F_2 = wx + \overline{w}\,\overline{x}y + wyz.$$

BÀI TẬP CHƯƠNG VIII:

1. Cho S là tập hợp các ước nguyên dương của 70, với các phép toán •, + và **’** được định nghĩa trên S như sau:

a • b = UCLN(a, b), a + b = BCNN(a, b), a**’** = 70/a.

Chứng tỏ rằng S cùng với các phép toán •, + và **’** lập thành một đại số Boole.

2. Chứng minh trực tiếp các định lý 6b, 7b, 8b (không dùng đối ngẫu để suy ra từ 6a, 7a, 8a).

3. Chứng minh rằng:

a) (a+b)**.**(a+b’) = a;

b) (a.b)+(a’.c) = (a+c)**.**(a’+b).

4. Cho các hàm Boole F_1, F_2, F_3 xác định bởi bảng sau:

x	y	z	F_1	F_2	F_3
0	0	0	1	1	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	1	1	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	0	1	1
1	1	1	1	1	1

Vẽ mạch thực hiện các hàm Boole này.

5. Hãy dùng các cổng NAND để xây dựng các mạch với các đầu ra như sau:

a) $\bar{x}$ **b)** xy **c)** $x+y$ **d)** $x \oplus y$.

6. Hãy dùng các cổng NOR để xây dựng các mạch với các đầu ra được cho trong Bài tập **5**.

7. Hãy dùng các cổng NAND để dựng mạch cộng bán phần.

8. Hãy dùng các cổng NOR để dựng mạch cộng bán phần.

9. Dùng các bản đồ Karnaugh, tìm dạng tổng chuẩn tắc tối thiểu (khai triển cực tiểu) của các hàm Boole ba biến sau:

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

b) $F = xyz + xy\bar{z} + +\bar{x}yz + \bar{x}y\bar{z}$.

c) $F = xy\bar{z} + +\bar{x}yz + \bar{x}\bar{y}z + \bar{x}yz + \bar{x}\bar{y}z$.

d) $F = xyz + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}y\bar{z} + \bar{x}\bar{y}\bar{z}$.

10. Dùng các bản đồ Karnaugh, tìm dạng tổng chuẩn tắc tối thiểu của các hàm Boole bốn biến sau:

a) $F = wxyz + wx\bar{y}z + wx\bar{y}\bar{z} + w\bar{x}y\bar{z} + w\bar{x}\bar{y}z$.

b) $F = wxy\bar{z} + wx\bar{y}z + w\bar{x}yz + \bar{w}x\bar{y}z + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}z$.

c) $F = wxyz + wxy\bar{z} + wx\bar{y}z + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}z$.

d) $F = wxyz + wxy\bar{z} + wx\bar{y}z + w\bar{x}yz + w\bar{x}y\bar{z} + \bar{w}xyz + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}z$.

11. Dùng phương pháp Quine-McCluskey, tìm dạng tổng chuẩn tắc tối thiểu của các hàm Boole ba biến cho trong Bài tập 9 và hãy vẽ mạch thực hiện các dạng tối thiểu tìm được.

12. Dùng phương pháp Quine-McCluskey, tìm dạng tổng chuẩn tắc tối thiểu của các hàm Boole bốn biến cho trong Bài tập 9 và hãy vẽ mạch thực hiện các dạng tối thiểu tìm được.

13. Hãy giải thích làm thế nào có thể dùng các bản đồ Karnaugh để rút gọn dạng tích chuẩn tắc (tích các tổng) hoàn toàn của một hàm Boole ba biến. (Gợi ý: Đánh dấu bằng số 0 tất cả các tuyển sơ cấp trong biểu diễn và tổ hợp các khối của các tuyển sơ cấp.)

14. Dùng phương pháp ở Bài tập 13, hãy rút gọn dạng tích chuẩn tắc hoàn toàn:

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

TÀI LIỆU THAM KHẢO

[1] *Nguyễn Cam-Chu Đức Khánh*, Lý thuyết đồ thị, NXB Thành phố Hồ Chí Minh, 1999.
[2] *Hoàng Chúng*, Đại cương về toán học hữu hạn, NXB Giáo dục, 1997.
[3] *Phan Đình Diệu*, Lý thuyết Ô-tô-mat và thuật toán, NXB Đại học và THCN, 1977.
[4] *Đỗ Đức Giáo*, Toán rời rạc, NXB Đại học Quốc Gia Hà Nội, 2000.
[5] *Nguyễn Xuân Quỳnh*, Cơ sở toán rời rạc và ứng dụng, NXB Giáo dục, 1995.
[6] *Đặng Huy Ruận*, Lý thuyết đồ thị và ứng dụng, NXB Khoa học và Kỹ thuật, 2000.
[7] *Nguyễn Tô Thành-Nguyễn Đức Nghĩa*, Toán rời rạc, NXB Giáo dục, 1997.
[8] *Claude Berge*, Théorie des graphes et ses applications, Dunod, Paris 1963.
[9] *Richard Johnsonbaugh*, Discrete Mathematics, Macmillan Publishing Company, New york 1992.