

Đại số Boole

Đại số logic

Nguyễn Quốc Cường – 31

1

Nội dung

- Giới thiệu
- Các tiên đề trong đại số logic
- Các định lý
- Nguyên lý của tính đối ngẫu (duality)
- Cách biểu diễn hàm logic

2

Tài liệu tham khảo

- Digital Design: Principles & Practices – John F Wakerly – Printice Hall

3

Giới thiệu

- 1854 nhà toán học Anh, George Boole (1815-1864) phát minh ra hệ thống đại số chỉ có hai giá trị
- Năm 1938, tại Bell Lab, Claude E. Shannon đã chỉ ra cách áp dụng đại số Boole vào phân tích và mô tả các mạch sử dụng role (còn gọi là switching algebra), và cũng được áp dụng cho các phân tích mạch số hiện nay.

4

Tiên đề

- Tiên đề 1:

$$(A1) X = 0 \text{ if } X \neq 1$$

$$(A1') X = 1 \text{ if } X \neq 0$$

- Tiên đề 2: (định nghĩa toán tử đảo)

$$(A2): \text{If } X = 0 \text{ then } X' = 1$$

$$(A2'): \text{If } X = 1 \text{ then } X' = 0$$

Toán tử “ ’ ” là toán tử đảo hay bù

(một số ký hiệu khác của toán tử đảo: $\sim X$, \bar{X})

Tuy nhiên việc sử dụng ‘ thường được sử dụng trong các ngôn ngữ lập trình HDLs)

5

- Tiên đề 3 , 4 và 5 :Định nghĩa các toán VÀ và HOẶC logic:

$$(A3) \quad 0 \cdot 0 = 0$$

$$(A3') \quad 1 + 1 = 1$$

$$(A4) \quad 1 \cdot 1 = 1$$

$$(A4') \quad 0 + 0 = 0$$

$$(A5) \quad 0 \cdot 1 = 1 \cdot 0 = 0$$

$$(A5') \quad 1 + 0 = 0 + 1 = 1$$

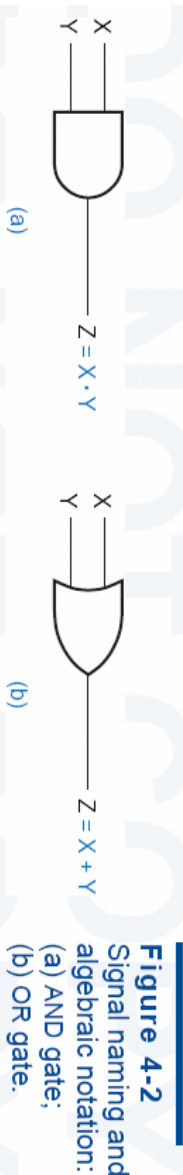
Toán tử AND sử dụng ký hiệu \cdot
Toán tử OR sử dụng ký hiệu $+$

Tất cả các hệ thống logic đều có thể mô tả và phân tích dựa trên 5 tiên đề trên

6

Ký hiệu các phần tử logic trên sơ đồ

Figure 4-1
Signal naming and algebraic notation for an inverter.



Định lý cho một biến

Table 4-1
Switching-algebra theorems with one variable.

(T1)	$X + 0 = X$	(T1')	$X \cdot 1 = X$	(Identities)
(T2)	$X + 1 = 1$	(T2')	$X \cdot 0 = 0$	(Null elements)
(T3)	$X + X = X$	(T3')	$X \cdot X = X$	(Idempotency)
(T4)	$(X')' = X$			(Involution)
(T5)	$X + X' = 1$	(T5')	$X \cdot X' = 0$	(Complements)

Việc chứng minh các định lý này có thể sử dụng phương pháp quy nạp hoàn toàn (vì số giá trị của các biến chỉ có 0 và 1 nên rất dễ áp dụng phương pháp quy nạp)

cho 2 và ba biến

Table 4-2 Switching-algebra theorems with two or three variables.

(T6)	$X + Y = Y + X$	(T6')	$X \cdot Y = Y \cdot X$	(Commutativity)
(T7)	$(X + Y) + Z = X + (Y + Z)$	(T7')	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	(Associativity)
(T8)	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	(T8')	$(X + Y) \cdot (X + Z) = X + Y \cdot Z$	(Distributivity)
(T9)	$X + X \cdot Y = X$	(T9')	$X \cdot (X + Y) = X$	(Covering)
(T10)	$X \cdot Y + X \cdot Y' = X$	(T10')	$(X + Y) \cdot (X + Y') = X$	(Combining)
(T11)	$X \cdot Y + X' \cdot Z + Y \cdot Z = X \cdot Y + X' \cdot Z$			(Consensus)
(T11')	$(X + Y) \cdot (X' + Z) \cdot (Y + Z) = (X + Y) \cdot (X' + Z)$			

Chú ý: để thuận tiện thường viết $X \cdot Y$ thay cho $(X \cdot Y)$

Cho n biến

Table 4-3 Switching-algebra theorems with n variables.

(T12)	$X + X + \dots + X = X$	(Generalized idempotency)
(T12')	$X \cdot X \cdot \dots \cdot X = X$	
(T13)	$(X_1 \cdot X_2 \cdot \dots \cdot X_n)' = X_1' + X_2' + \dots + X_n'$	(DeMorgan's theorems)
(T13')	$(X_1 + X_2 + \dots + X_n)' = X_1' \cdot X_2' \cdot \dots \cdot X_n'$	(Generalized DeMorgan's theorem)
(T14)	$[F(X_1, X_2, \dots, X_n, +, \cdot)]' = F(X_1', X_2', \dots, X_n', \cdot, +)$	(Generalized DeMorgan's theorem)
(T15)	$F(X_1, X_2, \dots, X_n) = X_1 \cdot F(1, X_2, \dots, X_n) + X_1' \cdot F(0, X_2, \dots, X_n)$	(Shannon's expansion theorem)
(T15')	$F(X_1, X_2, \dots, X_n) = [X_1 + F(0, X_2, \dots, X_n)] \cdot [X_1' + F(1, X_2, \dots, X_n)]$	

Để chứng minh sử dụng phương pháp quy nạp hữu hạn:

- chứng minh đúng với $n = 2$
- giả thiết đúng với $n = i$, chúng minh đúng với $n = i + 1$

Nguyên lý đối ngẫu

- Các định lý hay đồng nhất thức trong đại số logic sẽ luôn đúng nếu thay 0 và 1 tráo đổi cho nhau và đồng thời \cdot và $+$ cũng được tráo đổi cho nhau.

- Hàm đối ngẫu:

– Cho hàm logic $F(X_1, X_2, \dots, X_n, +, \cdot, ')$

– Hàm đối ngẫu của F được định nghĩa là hàm có cùng dạng biểu thức với các toán tử \cdot và $+$ được đổi chỗ cho nhau

$$F^D(X_1, X_2, \dots, X_n, +, \cdot, ') = F(X_1, X_2, \dots, X_n, \cdot, +, ')$$

$+$ và \cdot đổi chỗ

11

Nguyên lý đối ngẫu và định lý DeMorgan

$$[F(X_1, X_2, \dots, X_n)]' = F^D(X_1', X_2', \dots, X_n')$$

$$F(X_1, X_2, \dots, X_n) = [F^D(X_1', X_2', \dots, X_n')]'$$

(định lý DeMorgan)

12

Biểu diễn hàm logic thông qua bảng

Table 4-4t
General truth table
structure for a
3-variable logic
function, $F(X,Y,Z)$.

Bảng sự thực (không bao
gồm hàng ROW), tuy nhiên
thường được sử dụng để
chỉ giá trị tổ hợp của các
biến

Row	X	Y	Z	F
0	0	0	0	$F(0,0,0)$
1	0	0	1	$F(0,0,1)$
2	0	1	0	$F(0,1,0)$
3	0	1	1	$F(0,1,1)$
4	1	0	0	$F(1,0,0)$
5	1	0	1	$F(1,0,1)$
6	1	1	0	$F(1,1,0)$
7	1	1	1	$F(1,1,1)$

Table 4-5
Truth table for a
particular 3-variable
logic function, $F(X,Y,Z)$.

Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Một số khái niệm

- Hệ số chữ (literal): là một biến đơn , hoặc phần bù của nó. Ví dụ: X, Y, X', \dots
- Số hạng tích (product term): là một literal hoặc tích logic của nhiều literal
Ví dụ: $Z', X \oplus Y, X' \oplus Y \oplus Z'$
- Biểu thức tổng của các tích: là một tổng logic của các số hạng tích
- Số hạng tổng (sum term): là một literal hoặc tổng logic của nhiều literal
Ví dụ: $X', X+Y+Z'$
- Biểu thức tích của các tổng: là tích logic của các số hạng tổng

15

- Một số hạng chuẩn (normal term): là một số hạng tích hoặc tổng mà trong đó không có biến nào xuất hiện hơn một lần
- Ví dụ các số hạng **không chuẩn**:
 - $X + Y + X', Y \oplus X \oplus X' \oplus Z$
 - Ví dụ các số hạng chuẩn:
 - $X + Y, X \oplus Y \oplus Z$
 - minterm n biến: là một số hạng tích chuẩn của n literal
 - maxterm n biến: là số hạng tổng chuẩn của n literal

16

The information contained in a truth table can also be conveyed algebraically. To do so, we first need some definitions:

- A *literal* is a variable or the complement of a variable. Examples: X , Y , X' , Y' . *literal*
- A *product term* is a single literal or a logical product of two or more literals. Examples: Z' , $W \cdot X \cdot Y$, $X \cdot Y' \cdot Z$, $W' \cdot Y' \cdot Z$. *product term*
- A *sum-of-products expression* is a logical sum of product terms. Example: $Z' + W \cdot X \cdot Y + X \cdot Y' \cdot Z + W' \cdot Y' \cdot Z$. *sum-of-products expression*
- A *sum term* is a single literal or a logical sum of two or more literals. Examples: Z' , $W + X + Y$, $X + Y' + Z$, $W' + Y' + Z$. *sum term*
- A *product-of-sums expression* is a logical product of sum terms. Example: $Z' \cdot (W + X + Y) \cdot (X + Y' + Z) \cdot (W' + Y' + Z)$. *product-of-sums expression*
- A *normal term* is a product or sum term in which no variable appears more than once. A nonnormal term can always be simplified to a constant or a normal term using one of theorems T3, T3', T5, or T5'. Examples of non-normal terms: $W \cdot X \cdot X \cdot Y'$, $W + W + X' + Y$, $X \cdot X' \cdot Y$. Examples of normal terms: $W \cdot X \cdot Y'$, $W + X' + Y$. *normal term*
- An *n-variable minterm* is a normal product term with n literals. There are 2^n such product terms. Examples of 4-variable minterms: $W' \cdot X' \cdot Y' \cdot Z'$, $W \cdot X \cdot Y' \cdot Z$, $W' \cdot X' \cdot Y \cdot Z'$. *minterm*
- An *n-variable maxterm* is a normal sum term with n literals. There are 2^n such sum terms. Examples of 4-variable maxterms: $W' + X' + Y' + Z'$, $W + X' + Y' + Z$, $W' + X' + Y + Z'$. *maxterm*

17

- Minterm: có thể được định nghĩa là số hạng tích ứng với một hàng của bảng chân lý sao cho tích đó bằng 1
- Maxterm: có thể được định nghĩa là số hạng tổng ứng với một hàng của bảng chân lý sao cho tổng đó bằng 0

18

Table 4-6
Minterms and maxterms
for a 3-variable logic
function, $F(X,Y,Z)$.

Row	X	Y	Z	F	Minterm	Maxterm
0	0	0	0	$F(0,0,0)$	$X' \cdot Y' \cdot Z'$	$X + Y + Z$
1	0	0	1	$F(0,0,1)$	$X' \cdot Y' \cdot Z$	$X + Y + Z'$
2	0	1	0	$F(0,1,0)$	$X' \cdot Y \cdot Z'$	$X + Y' + Z$
3	0	1	1	$F(0,1,1)$	$X' \cdot Y \cdot Z$	$X + Y' + Z'$
4	1	0	0	$F(1,0,0)$	$X \cdot Y' \cdot Z'$	$X' + Y + Z$
5	1	0	1	$F(1,0,1)$	$X \cdot Y' \cdot Z$	$X' + Y + Z'$
6	1	1	0	$F(1,1,0)$	$X \cdot Y \cdot Z'$	$X' + Y' + Z$
7	1	1	1	$F(1,1,1)$	$X \cdot Y \cdot Z$	$X' + Y' + Z'$

19

Biểu diễn hàm qua minterm và maxterm

- Hàm logic có thể biểu diễn dưới dạng:
 - canonical sum: tổng của các minterm ứng với các hàng của bảng chân lý mà tại đó giá trị hàm bằng 1
 - canonical product: tích của các maxterm ứng với các hàng của bảng chân lý mà tại đó giá trị hàm bằng 0

20

Table 4-5
Truth table for a
particular 3-variable
logic function, $F(X,Y,Z)$.

Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$$F = (X' \cdot Y' \cdot Z') + (X' \cdot Y \cdot Z) + (X \cdot Y' \cdot Z') + (X \cdot Y \cdot Z')$$

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

21

- Để đơn giản trong ký hiệu, người ta thường sử dụng dạng viết rút gọn sau:

$$\begin{aligned} F &= (X' \cdot Y' \cdot Z') + (X' \cdot Y \cdot Z) + (X \cdot Y' \cdot Z') \\ &\quad + (X \cdot Y \cdot Z') + (X \cdot Y \cdot Z) \\ &= \Sigma_{X,Y,Z}(0, 3, 4, 6, 7) \end{aligned}$$

$$\begin{aligned} F &= (X + Y + Z')(X + Y' + Z)(X' + Y + Z') \\ &= \prod_{Z,Y,Z}(1, 2, 5) \end{aligned}$$

X, Y, Z là các biến, đi kèm với chỉ số các hàng tương ứng của các minterm hoặc maxterm

22

Tối thiểu hóa hàm logic

- Hàm logic có thể biểu diễn thông qua:
 - canonical sum
 - canonical product

Tuy nhiên đó là các dạng chưa được tối thiểu.

- Để giảm số input hay số gate sử dụng trong mạch cần phải tối thiểu hóa mạch.

23

Bìa Karnaugh

- Là cách biểu diễn đồ họa của bảng chân lý

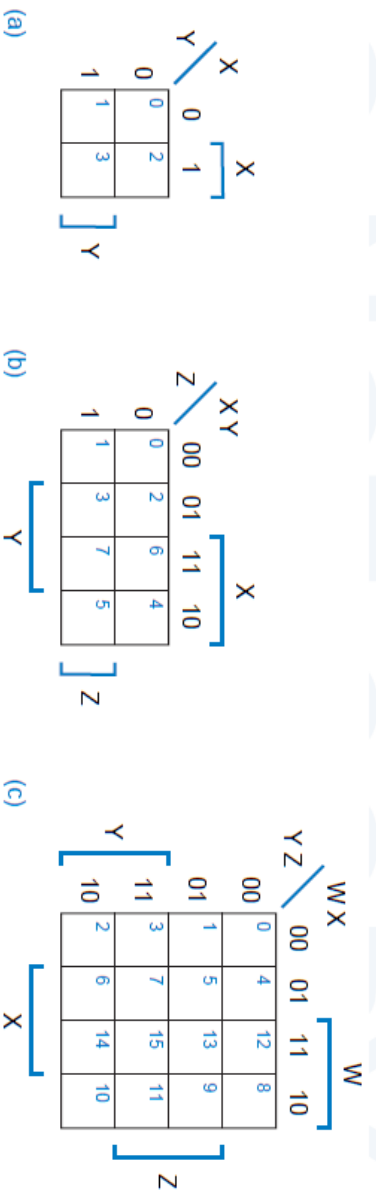


Figure 4-26 Karnaugh maps: (a) 2-variable; (b) 3-variable; (c) 4-variable.

24

- K-map : n biến sẽ có 2^n ô
- Mỗi một ô trong K-map ứng với một hàng trong bảng chân lý.
- Quy ước các ô kề nhau thì tổ hợp các biến chỉ được khác nhau một giá trị
- K-map chỉ thuận tiện sử dụng cho hàm logic có 6 biến trở xuống
- Từ K-map có thể viết được các canonical sum hoặc canonical product tương tự như bảng chân lý

25

Tối thiểu hóa dạng tổng các tích

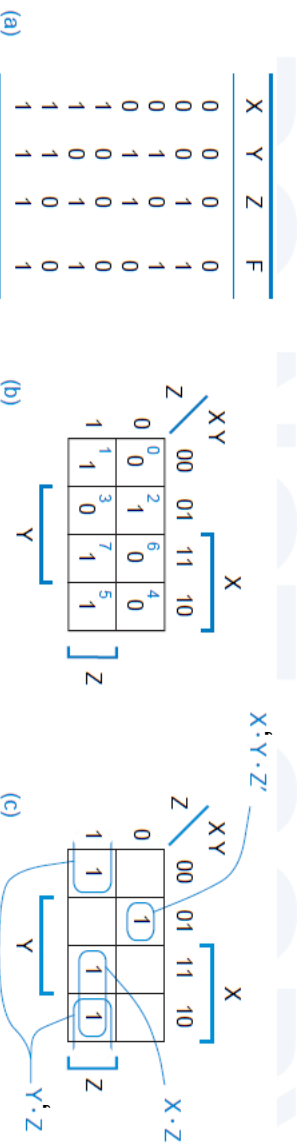


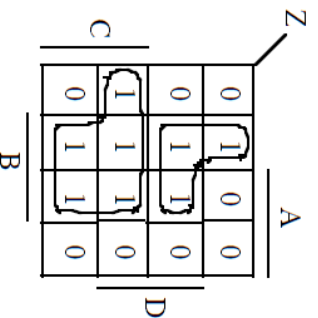
Figure 4-27 $F = \Sigma_{X,Y,Z}(1,2,5,7)$: (a) truth table; (b) Karnaugh map; (c) combining adjacent 1-cells.

26

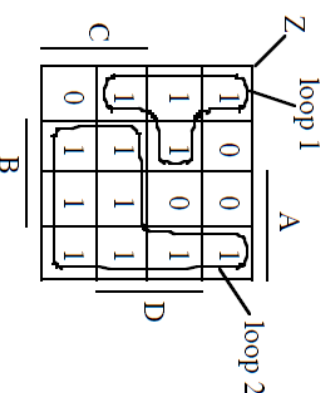
- Quy tắc nhóm các ô của K-map:
 - Nhóm 2^k các ô có giá trị 1 kề nhau sao cho k là max ($1 \leq k \leq n$, với n là số biến)
 - Có chính xác $(n-k)$ biến có giá trị không đổi trong số các ô được nhóm
- Dạng tích:
 - nếu biến có giá trị là 1 trong 2^k ô được nhóm thì product term sẽ chứa biến đó
 - nếu biến có giá trị 0 trong 2^k ô được nhóm thì product term sẽ chứa bù của biến đó
 - nếu biến có cả giá trị 1 và 0 trong 2^k ô được nhóm thì nó sẽ không xuất hiện trong product term

27

các nhóm không đúng



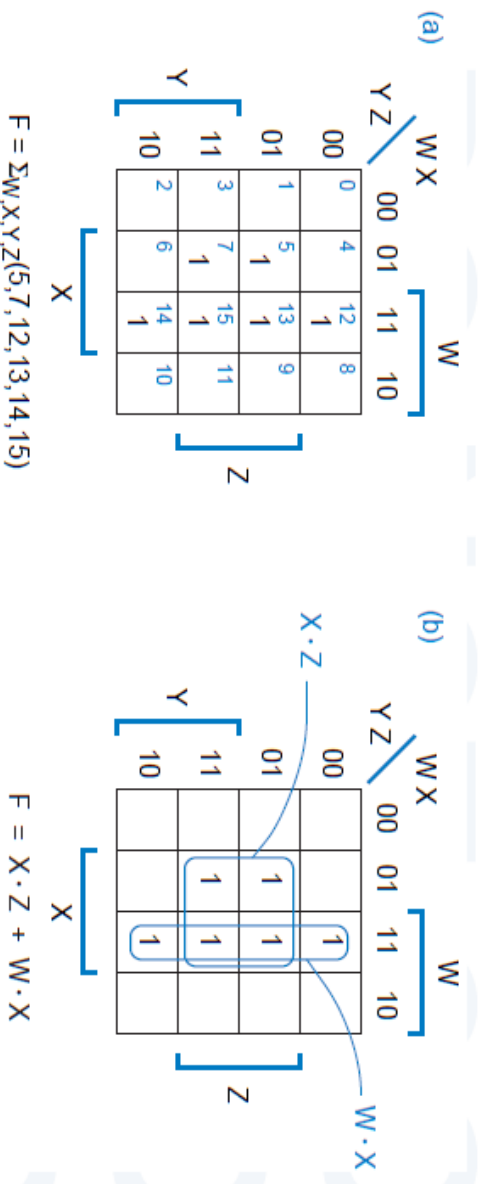
Violates Rule 1



Violates Rule 2

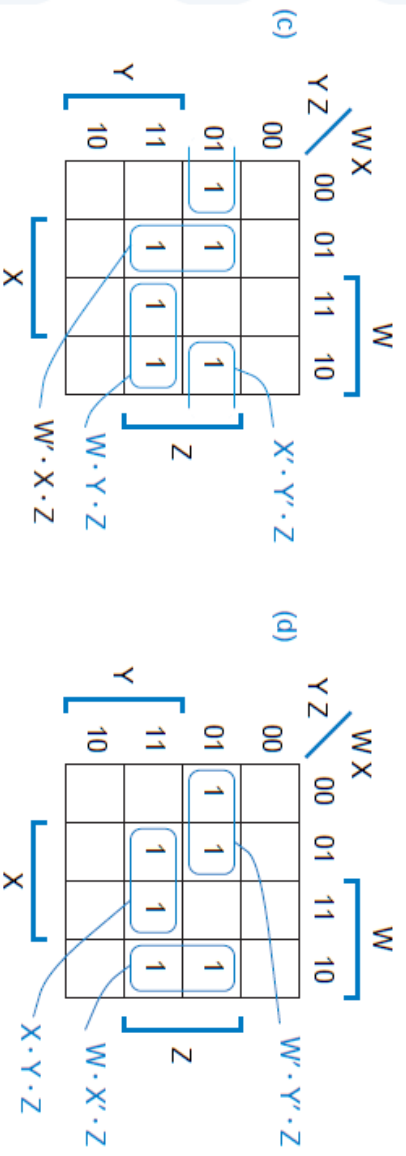
28

Ví dụ



29

- Dạng tối giản sử dụng K-map không phải là duy nhất



30

Tối thiểu hóa dạng tích các tổng

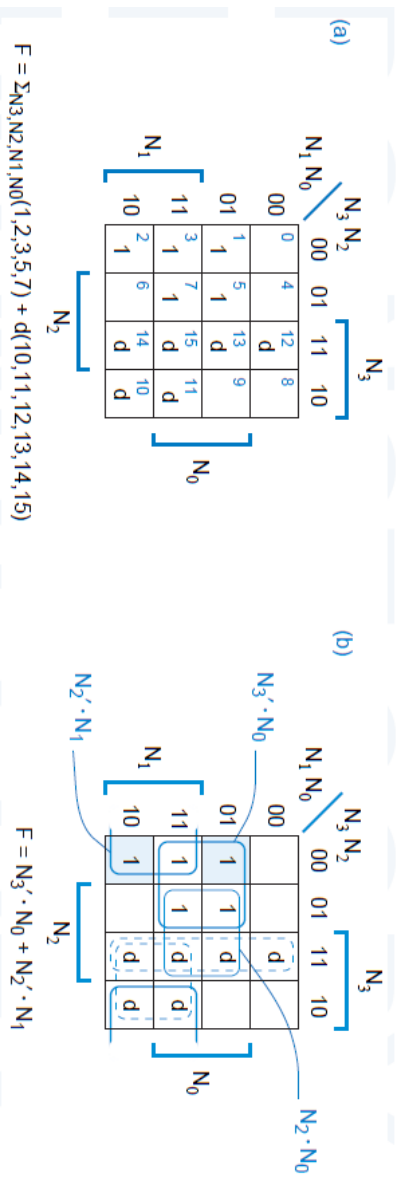
- Nhóm 2^k các ô có giá trị 0 kề nhau sao cho k là max:
 - nếu biến có giá trị là 1 trong 2^k ô được nhóm thì sum-term sẽ chứa bù của biến đó
 - nếu biến có giá trị 0 trong 2^k ô được nhóm thì sum-term sẽ chứa biến đó
 - nếu biến có cả giá trị 1 và 0 trong 2^k ô được nhóm thì nó sẽ không xuất hiện trong sum term

31

Các tổ hợp đầu vào “Don’t-Care”

- Trong trường hợp ứng với một số tổ hợp giá trị các inputs giá trị hàm logic có thể tùy ý (bằng 0 hoặc bằng 1) → các tổ hợp “don’t-care”
- Sử dụng các tổ hợp “don’t-care” trong tối giản hàm:
 - Cho phép tổ hợp don’t-care tham gia vào các ô sao cho số ô 2^k là lớn nhất
 - *Không* nhóm các ô chỉ toàn don’t-care

32



33

Các phương pháp tối giản sử dụng chương trình

- Khi số biến lớn, sử dụng thuật toán:
 - Queen-McCluskey (tham khảo)
 - Espresso II, Espresso-MV (tham khảo)

34