

Chương 2: Mạch tổ hợp 2 mức



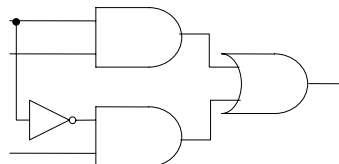
- Giới thiệu
- Mạng logic lập trình được
- Các tính chất căn bản và dư thừa
- Các phép toán Bool trên các hàm logic
- Biến đổi trên Cube và Cover
- Độ phức tạp của mạch 2 mức

Giới thiệu

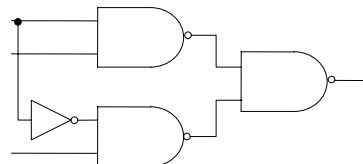
- Mạch tổ hợp 2 mức có thể được biểu diễn ở nhiều dạng khác nhau:
 - Sum-of-Product
 - Product-of-Sum
 - NAND-NAND
 - NOR-NOR
 - ...
- Sum-of-Product
 - Mức thứ nhất của các cổng là AND
 - Mức thứ hai của các cổng là OR
 - Có thể xem là mạch tổ hợp 3 mức nếu tính cả cổng not



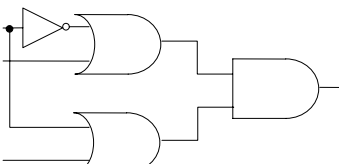
Giới thiệu



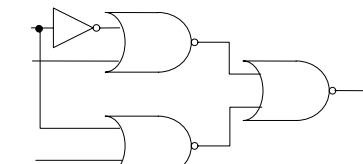
Sum-of-Product



NAND-NAND



Product-of-Sum

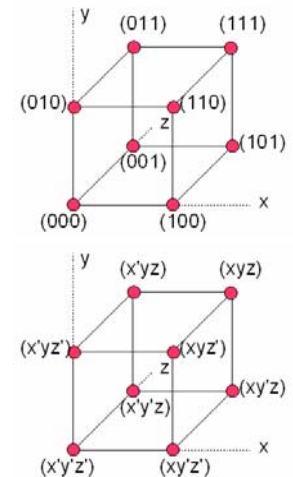


NOR-NOR



Thuật ngữ-Căn bản

- **Đại số Boole** gồm một tập giá trị $B = \{0, 1\}$ và hai phép toán "+" và "."
- Mỗi **biến Boole** nhận một trong hai giá trị **0** hoặc **1**
- Mỗi **biến Boole a** có **phần bù** kí hiệu a'
- Một không gian nhiều chiều được bao phủ bởi một tập hợp n biến Boole được biểu diễn bằng B^n
- Mỗi điểm trong không gian B^n được gọi là **đỉnh** và được biểu diễn bởi một vector nhị phân n chiều



Thuật ngữ-Bool

- Một **biến Boole** được biểu diễn bằng một **ký tự**
– Ví dụ: a, b, c' ...
- Một **biểu thức Boole** được biểu diễn bằng một **chuỗi** các biến và các **phép toán** Boole
– Ví dụ: $a \cdot b \cdot c', a + b' \cdot c$...
- Một **cube** được biểu diễn bằng tích các biến
– Ví dụ: $abc', a'b$...
- Một cube chứa một hay nhiều đỉnh



Thuật ngữ-Hàm Bool đầy đủ

- Hàm số **Bool đầy đủ** (completely specified Boolean function f)
- Một hàm số Bool đầy đủ f của N biến nhập $x_1, \dots, x_N$ và M biến xuất $f_1, \dots, f_M$ là một ánh xạ
 $f: B^N \rightarrow B^M$
với $B^N = \{0, 1\}^N$ và $B^M = \{0, 1\}^M$
Nếu $M = 1$: Hàm ngõ xuất đơn (single-output function)
- Tập **On-Set** là tập các giá trị nhập của biến x sao cho $f_i(x) = 1$
- Tập **OFF-Set** là tập các giá trị nhập của biến x sao cho $f_i(x) = 0$



Thuật ngữ-Hàm Bool không đầy đủ

- Hàm số **Bool không đầy đủ** (incompletely specified Boolean function f)
- Một hàm số Bool không đầy đủ f của N biến nhập $x_1, \dots, x_N$ và M biến xuất $f_1, \dots, f_M$ là một ánh xạ
 $f: B^N \rightarrow Y^M$
với $B^N = \{0, 1\}^N$ và $Y^M = \{0, 1, X\}^M$
 $x \in B^N$ mà $f_i(x) = X$, với X là giá trị don't care (DC-không quan tâm)
- Tập **DC-Set** là tập các giá trị nhập của biến x sao cho $f_i(x) = X$



Thuật ngữ-Literal

- Bù (complement)
- Bù của một hàm luận lý đầy đủ f là $\bar{f}$ sao cho:
 - Tập On-Set của $\bar{f}$ bằng tập OFF-Set của f
 - Tập OFF-Set của $\bar{f}$ bằng tập On-Set của f
- Literal**: là một **biến** Bool hoặc là **phản bù** của nó.
- Một **cube** là một sự kết hợp của nhiều **literal**
– Ví dụ: $abc, abc'd, \dots$



Thuật ngữ-Cube

- Một **cube** cũng có thể biểu diễn theo một vector nhị phân.
- Một **cube** của một hàm Bool f với N ngõ nhập và M ngõ xuất được biểu diễn là:

$$c = [c_1, \dots, c_N, c_{N+1}, \dots, c_{N+M}]$$

với $1 \leq i \leq N$:

c_i là 0 nếu biến x_i là bù trong c

c_i là 1 nếu biến x_i không bù trong c

c_i là _ hoặc x nếu x_i không xuất hiện trong c

với $N+1 \leq i \leq N+M$

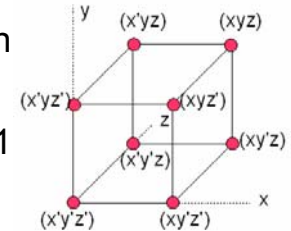
c_i là 0 nếu c thuộc tập OFF-set của ngõ xuất $i-N$ của f

c_i là 1 nếu c thuộc tập ON-set của ngõ xuất $i-N$ của f

c_i là _ nếu c thuộc tập DC-set của ngõ xuất $i-N$ của f

Thuật ngữ-Minterm

- Minterm**: là một cube mà trong đó tất cả các biến đều xuất hiện
- Minterm** có thể được biểu diễn bởi một đỉnh trong N -cube
- Minterm** m_1 là lớn hơn (**dominate**) m_2 ($m_1 > m_2$) nếu mỗi vị trí mà m_2 có 1 thì m_1 cũng có 1



Thuật ngữ-Cover

- Cover**: được định nghĩa là một tập các cube và được biểu diễn dưới dạng tổng các tích
 - Ví dụ: tập 2 cube $\{a, b, c\}, \{d, e, f\}$ là một cover đơn ngõ nhập được biểu diễn: $abc+def$
- Contains** (chứa): cube q là chứa cube r khi và chỉ khi:
 - Tập các literal của q là tập con của tập literal của r
 - Các output của q là tập cha của các output của r
 - Ví dụ:
 - cube $[0-, 1]$ contains (chứa) cube $[00, 1]$
 - Cube $[0-, 11]$ contains cube $[0-, 10]$

Thuật ngữ-Implicant

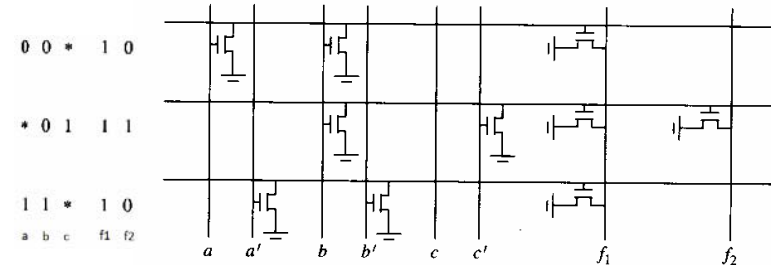
- Implicant**: 1 cube chỉ chứa các tập đỉnh ON-set và DC-set của hàm Bool f thì được gọi là một Implicant
- Prime implicant/prime**: là một implicant mà
 - không chứa bất kỳ một implicant khác nào của f
 - và cũng không hoàn toàn chứa trong DC-set của f .
- Essential Prime implicant/essential prime** là một prime mà chứa 1 hoặc nhiều tập ON-set mà không chứa trong bất kỳ một prime implicant nào khác
- Optional prime implicant** là một prime implicant mà tất cả các đỉnh được chứa trong các prime implicant khác

Thuật ngữ-Minimal cover

- **Minimal cover** cho một hàm f được sinh ra bởi việc lựa chọn tất cả những essential prime implicants và một tập nhỏ nhất của optional prime implicant để mà tất cả các đỉnh trong tập ON-set của f đều được chứa trong đó
- **Relatively essential vertex** của một cube q trong một cover C là một đỉnh trong tập ON-set có trong q và không có trong bất kỳ một cube nào khác trong C .

Mảng luận lý lập trình được

- Programmable Logic Arrays (**PLAs**)
- Mạch tổ hợp 2 lớp có thể được hiện thực bởi **PLAs**



Các tính chất căn bản và dư thừa

- **Irredundant** (không dư thừa): một implicant trong một cover là không dư thừa nếu nó chứa một essential hoặc một relatively essential vertex. Ngược lại thì nó là **redundant** (dư thừa)
- Một cover là một prime nếu mỗi một implicant trong nó đều là prime

Đại số Bool trên các hàm luận lý

- Phép nhân/giao (intersection/product):
 - $h = f.g$
 - $\text{ON-set}_h = \text{ON-set}_f \wedge \text{ON-set}_g$
 - $\text{OFF-set}_h = \text{OFF-set}_f \vee \text{OFF-set}_g$
 - DC-set???
- Phép trừ (difference):
 - $h = f - g = f.g'$
- Phép hội/tổng (union/sum)
 - $h = f \cup g = f + g$
 - $\text{ON-set}_h = \text{ON-set}_f \vee \text{ON-set}_g$
 - $\text{OFF-set}_h = \text{OFF-set}_f \wedge \text{OFF-set}_g$
- Tautology: ngõ ra luôn là 1 với mọi ngõ vào
 - $F \cong 1$
 - $\text{OFF-set} = \text{empty}$

Biến đổi trên Cube và Cover

- Phép giao trên cube
 - $q = c \cap d$
 - Tập literal là hội của literal trong c và d
 - $q_i = c_i$ nếu $(c_i = d_i \vee d_i = X)$
 - $q_i = d_i$ nếu $c_i = X$
 - Nếu $(c_i = 1 \wedge d_i = 0) \vee (c_i = 0 \wedge d_i = 1)$ thì $q = \phi$
 - Ví dụ:
 - $11X \cap X10 = 110$
 - $11X \cap X0X = \phi$
- Phép giao trên **cover**: thực hiện phép giao trên từng cặp cube của các cover đó

Biến đổi trên Cube và Cover

- Giải thuật Disjoint SHARP
- Ký hiệu: Θ
- Được sử dụng để tính $C - D$ hay $C \cap \bar{D}$ với C, D là các cover.

Biến đổi trên Cube và Cover

- Giải thuật Disjoint SHARP
- Thực hiện phép trừ một cube c cho một cube d ($c \Theta d$)
 - Nếu $c \cap d = \phi$ thì $c \Theta d = c$
 - Ngược lại:
 1. $Q = \phi$, $q = c$
 2. $i = 0$
 3. $i = i + 1$, nếu $i > N$ (số literal) đến bước 7
 4. Nếu $(q_i = d_i) \vee (d_i \text{ là } x)$ thì đến bước 3
 5. Nếu $q_i = x$, gán $p = q$, loại trừ $p_i = \bar{d}_i$, $Q = Q \vee p$
 6. Gán q_i vào d_i trở lại bước 3
 7. $c \Theta d$ đã được tính trong Q

Biến đổi trên Cube và Cover

- Ví dụ: cho d là __ 11, c là _ _ _ _
- $c \Theta d = ?$

$$\bar{d}_i$$

Biến đổi trên Cube và Cover

- Giải thuật Disjoint SHARP
- Thực hiện phép trừ một cover C cho một cube d ($C \ominus d$)

1. $Q = \emptyset$, $C_1 = C$
2. Lấy 1 cube (mới) $c \in C_1$
3. Tính $I = c \ominus d$
4. $Q = Q \cup I$. Loại c khỏi C_1
5. Nếu $C_1 = \emptyset$, thì $C \ominus d$ có kết quả là Q , ngược lại vào bước 2

Ví dụ: thực hiện phép trừ của $_001$ với $\{_ _ 0 _, _ _ 10\}$

Biến đổi trên Cube và Cover

- Giải thuật Disjoint SHARP
- Thực hiện phép trừ một cover C cho cover D ($C \ominus D$)

1. $Q = C$, $D_1 = D$
2. Lấy 1 cube (mới) $d \in D_1$
3. Tính $I = Q \ominus d$
4. $Q = I$. Loại c khỏi C_1
5. Nếu $D_1 = \emptyset$, thì $C \ominus D$ có kết quả là Q , ngược lại vào bước 2

- Giải pháp để hạn chế khối lượng tính toán, cần thiết phải hạn chế kích thước của các cover. Vì vậy cần phải hạn loại trừ trường hợp một cube c chứa một cube d nào đó trong cover (**single cube containment minimality**)

Độ phức tạp của mạch 2 mức

- Mạch 2 mức có thể khó kiểm soát
 - Ví dụ: $a_1 \oplus a_2 \dots \oplus a_n$. Có độ phức tạp là 2^n phép nhân nếu ở dạng tổng các tích
 - $(a_1+b_1)(a_2+b_2)\dots(a_n+b_n)$ có độ phức tạp 2^n ở dạng tổng các tích
- Các phép toán nhân nguyên trên các toán hạng n -bit, so sánh trên 2 toán hạng n -bit, cộng trừ... tất cả đều ở dạng tổng các tích và tích các tổng.

Bài tập