Thiết kế luận lý 1

Khoa KH & KTMT
Bộ môn Kỹ Thuật Máy Tính

Tài liệu tham khảo

- "*Digital Systems, Principles and Applications*", 11th Edition, Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss

dce
2014

Chương 2

Đại số Boole & các cổng luận lý

BK
TP.HCM

Nội dung

- Đại số Boole
- Đại số chuyển mạch
- Các cổng luận lý

Đại số Boole

- Đại số Boole được thế giới biết đến lần đầu tiên bởi **George Boole** qua tác phẩm "*An Investigation of the Laws of Thought*" vào năm 1854
- Các hằng và biến Boole chỉ được mang 2 giá trị 0 hoặc 1 (LOW / HIGH)
  - Các biến Boole biểu diễn cho một khoảng điện áp trên đường dây hoặc tại ngõ nhập/ngõ xuất của mạch
  - Giá trị 0 hoặc 1 được gọi là mức luận lý (*logic level*)

Đại số Boole

- Đại số Boole, cũng tương tự như các hệ đại số khác, được xây dựng thông qua việc xác định nghĩa một số những vấn đề cơ bản sau:
  - Miền (*domain*), là tập hợp (*set*) các phần tử (*element*) mà trên đó định nghĩa nên hệ đại số
  - Tập hợp các phép toán (*operation*) thực hiện được trên miền
  - Một tập hợp các định đề (*postulate*), hay tiên đề (*axiom*) được công nhận không qua chứng minh. Định đề phải đảm bảo tính nhất quán (*consistency*) và tính độc lập (*independence*)
  - Một tập hợp các hệ quả (*consequence*) được gọi là định lý (*theorem*), định luật (*law*) hay quy tắc (*rule*)

Định đề Huntington

- Phát biểu bởi nhà toán học Anh **E.V.Huntington** trên cơ sở hệ thống hóa các công trình của **G. Boole**
  - Sử dụng các phép toán trong luận lý mệnh đề (*propositional logic*)
- Tính đóng (*closure*)
  - Tồn tại miền ***B*** với ít nhất 2 phần tử phân biệt và 2 phép toán + và • sao cho:
    - Nếu *x* và *y* là các phần tử thuộc ***B*** thì *x* + *y* cũng là 1 phần tử thuộc ***B*** (phép cộng luận lý - *logical addition*)
    - Nếu *x* và *y* là các phần tử thuộc *B* thì *x* • *y* cũng là 1 phần tử thuộc ***B*** (phép nhân luận lý - *logical multiplication*)

Định đề Huntington …

- Tính đồng nhất (*identity*)

  Nếu x là một phần tử trong miền B thì

  – Tồn tại 1 phần tử 0 trong B, gọi là phần tử đồng nhất với phép toán $+$, thỏa mãn tính chất $x + 0 = x$

  – Tồn tại 1 phần tử 1 trong B, gọi là phần tử đồng nhất với phép toán $\bullet$, thỏa mãn tính chất $x \bullet 1 = x$

- Tính giao hoán (*commutative*)

  – Giao hoán của phép $+$:

  $$x + y = y + x$$

  – Giao hoán của phép $\bullet$:

  $$x \bullet y = y \bullet x$$

Định đề Huntington …

- Tính <u>phân phối</u> (*distributive*)
  - Phép • có tính phân phối trên phép +

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

  - Phép + có tính phân phối trên phép •

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

- <u>Bù</u> (*complementation*)

  Nếu x là 1 phần tử trong miền ***B*** thì sẽ tồn tại một phần tử khác gọi là x' (hay $\overline{x}$), là phần tử bù của x thỏa mãn:

  - $x + x' = 1$ và
  - $x \cdot x' = 0$

Tính đối ngẫu (*duality*)

- Quan sát các định đề Hungtinton, ta thấy chúng mang tính đối xứng (*symmetry*) tức là các định đề xuất hiện theo cặp
- Mỗi định đề trong 1 cặp có thể được xây dựng từ định đề còn lại bằng cách
  - Thay đổi các phép toán 2 ngôi (+ | •)
  - Thay đổi các phần tử đồng nhất (*0* | *1*)
- Có thể suy ra một kết quả nào đó từ các định đề bằng cách
  - Hoán đổi phép toán + với phép toán •
  - Hoán đổi phần tử đồng nhất *0* với phần tử đồng nhất *1*
- Điều này thể hiện tính đối ngẫu ở đại số Boole

Các định lý cơ bản (*fundamental theorem*)

- Các định lý được chứng minh từ các định đề Huntington và các định đề đối ngẫu theo 2 cách
  - Chứng minh bằng phản chứng (*contradiction*)
  - Chứng minh bằng quy nạp (*induction*)
- Định lý 1 (*Null Law*)
  - $x + 1 = 1$
  - $x \cdot 0 = 0$
- Định lý 2 (*Involution*)
  - $(x')' = x$
- Định lý 3 (*Idempotency*)
  - $x + x = x$
  - $x \cdot x = x$
- Định lý 4 (*Absorption*)
  - $x + x \cdot y = x$
  - $x \cdot (x + y) = x$

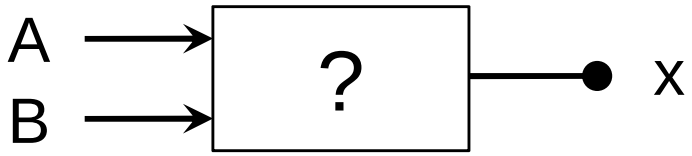
Các định lý cơ bản …

- Định lý 5 (*Simplification*)
  - $x + x'y = x + y$
  - $x(x' + y) = xy$
- Định lý 6 (*Associative Law*)
  - $x + (y + z) = (x + y) + z = x + y + z$
  - $x(yz) = (xy)z = xyz$
- Định lý 7 (*Consensus*)
  - $xy + x'z + yz = xy + x'z$
  - $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$
- **Định lý 8** (*De Morgan's Law*)
  - $(x + y)' = x'y'$
  - $(xy)' = x' + y'$

Bảng sự thật (Truth table)

- Phương tiện mô tả sự phụ thuộc của ngõ xuất vào mức luận lý (logic level) tại các ngõ nhập của mạch
  - Liệt kê tất cả các tổ hợp có thể của mức luận lý tại các ngõ nhập và kết quả mức luận lý tương ứng tại ngõ xuất của mạch
  - Số tổ hợp của bảng N-ngõ nhập: 2^N

A	B	x
0	0	1
0	1	0
1	0	1
1	1	0



A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Đại số chuyển mạch (*switching algebra*)

- Đối với đại số Boole, miền không bị hạn chế (không có giới hạn đặt ra đối với số lượng các phần tử trong miền)
- Các định đề Huntington giới hạn xem xét đại số Boole với 2 phần tử đồng nhất mà thôi

  ⇨ **Đại số Boole 2 phần tử**

- Năm 1937, **Claude Shannon** hiện thực đại số Boole 2 phần tử bằng mạch điện với các chuyển mạch (*switch*)
  - Chuyển mạch là thiết bị có 2 vị trí bền: tắt (*off*) hay mở (*on*)
  - 2 vị trí này phù hợp để biểu diễn cho *0* hay *1*

⇨ **Đại số Boole 2 phần tử còn được gọi là đại số chuyển mạch**

- Các phần tử đồng nhất được gọi là các hằng chuyển mạch (*switching constant*)
- Các biến (*variable*) biểu diễn các hằng chuyển mạch được gọi là các biến chuyển mạch (*switching variable*) ⇔ **tín hiệu**

Các phép toán chuyển mạch

- Đại số chuyển mạch sử dụng các phép toán trong luận lý mệnh đề với tên gọi khác
- Phép toán **AND**
  - Phép toán 2 ngôi tương đương với phép nhân luận lý
- Phép toán **OR**
  - Phép toán 2 ngôi tương đương với phép cộng luận lý

x	y	$x \cdot y$	$x + y$	x'
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Bảng sự thật các phép chuyển mạch

- Phép toán **NOT**
  - Phép toán 1 ngôi tương đương với phép bù luận lý

Các phép toán chuyển mạch …

- Các phép toán chuyển mạch có thể được hiện thực bởi mạch phần cứng
- Bảng sự thật có thể sử dụng như 1 công cụ dùng để xác minh quan hệ giữa các phép toán chuyển mạch
- Sử dụng bảng sự thật để chứng minh định lý De Morgan

$$(x + y)' = x' y'$$

x	y	x'	y'	$x + y$	$(x + y)'$	$x' y'$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Biểu thức (*expression*) chuyển mạch

- Biểu thức chuyển mạch là một quan hệ hữu hạn các hằng, biến, biểu thức chuyển mạch liên kết với nhau bởi các phép toán **AND**, **OR** và **NOT**
- Ví dụ

  $y + 1$, $x\,x' + x$, $z\,(x + y')'$

  $E = (x + yz)(x + y') + (x + y)'$
- *literal* được sử dụng để ám chỉ <u>biến</u> hay <u>bù của biến</u>

Biểu thức (*expression*) chuyển mạch...

- Một biểu thức có thể được chuyển thành nhiều dạng tương đương bằng cách sử dụng các luật Boole

  $E = (x + y\,z)\,(x + y') + (x + y)'$

  $E_1 = x\,x + x\,y' + x\,y\,z + y\,y'\,z + x'\,y'$

  $E_2 = x + x\,(y' + y\,z) + x'\,y'$

  $E_3 = x + x'\,y'$

  $E_4 = x + y'$

- Tại sao phải chuyển đổi dạng của các biểu thức ?
- Các thành phần thừa (*redundant*) trong biểu thức
  - *literal* lặp ($x\,x$ hay $x + x$)
  - biến và bù ($x\,x'$ hay $x + x'$)
  - hằng (0 hay 1)
- **Không hiện thực các thành phần thừa của biểu thức vào mạch**

Hàm (*function*) chuyển mạch

- Hàm chuyển mạch (*switching function*) là một phép gán xác định và duy nhất của những giá trị *0* và *1* cho tất cả các tổ hợp giá trị của các biến thành phần
- Hàm được xác định bởi danh sách các trị hàm tại mỗi tổ hợp giá trị của biến (*bảng sự thật*)
  - Tồn tại nhiều biểu thức biểu diễn cho 1 hàm
- Số lượng hàm chuyển mạch với *n* biến là 2 luỹ thừa 2^n

x	y	x'	y'	$x'y'$	$E_1 = x + x'y'$	$E_2 = x + y'$
0	0	1	1	1	1	1
0	1	1	0	0	0	0
1	0	0	1	0	1	1
1	1	0	0	0	1	1

Các phép toán chuyển mạch khác

- Phép toán **NAND**
  - Phép toán 2 ngôi tương đương với (**NOT AND**)
- Phép toán **NOR**
  - Phép toán 2 ngôi tương đương với (**NOT OR**)
- Phép toán **Exclusive OR**
  - $E = x \oplus y = x'y + xy'$
- Phép toán **XNOR (Ex. NOR)**
  - $E = (x \oplus y)' = xy + x'y'$

Biến		NAND	NOR	Ex. OR	XNOR
x	y	$(x \cdot y)'$	$(x + y)'$	$x \oplus y$	$(x \oplus y)'$
0	0	1	1	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	0	0	1

Cổng luận lý

- Để đại số chuyển mạch có thể thực hiện các công việc trong đời thật, cần phải có
  - Thiết bị vật lý thực hiện các phép toán chuyển mạch
  - Tín hiệu vật lý (điện áp, …) thay thế cho các biến chuyển mạch
- Cổng (*gate*) hay cổng luận lý (*logic gate*) là tên chung dùng để gọi các thiết bị vật lý thực hiện các phép toán chuyển mạch với độ chính xác (*accuracy*) và thời gian trễ (*delay*) chấp nhận được

Cổng luận lý

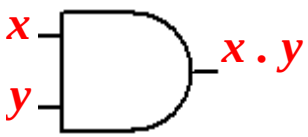
- Mỗi cổng được biểu diễn bởi 1 biểu tượng (*schematic symbol*) đặc trưng cùng với 1 số chân (*pin*, *terminal*) tượng trưng cho các biến chuyển mạch

Một biểu thức chuyển mạch bất kỳ luôn có thể được hiện thực trong đời thật bằng cách kết nối các cổng luận lý lại với nhau

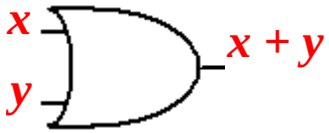
⇨ *Mạch luận lý (logic circuit) hay **mạch chuyển mạch** (switching circuit)*

Biểu tượng của các cổng luận lý

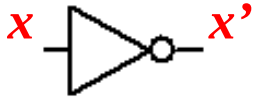
- Cổng **AND**



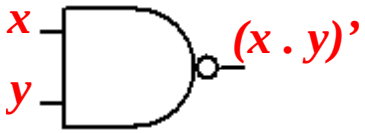
- Cổng **OR**



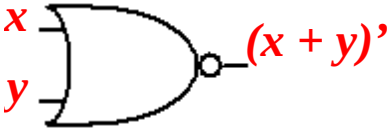
- Cổng **NOT**
  (cổng đảo - *inverter*)



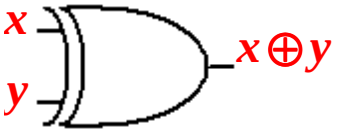
- Cổng **NAND**



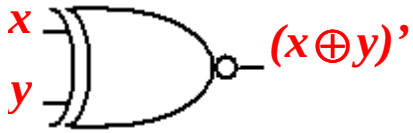
- Cổng **NOR**



- Cổng **XOR**

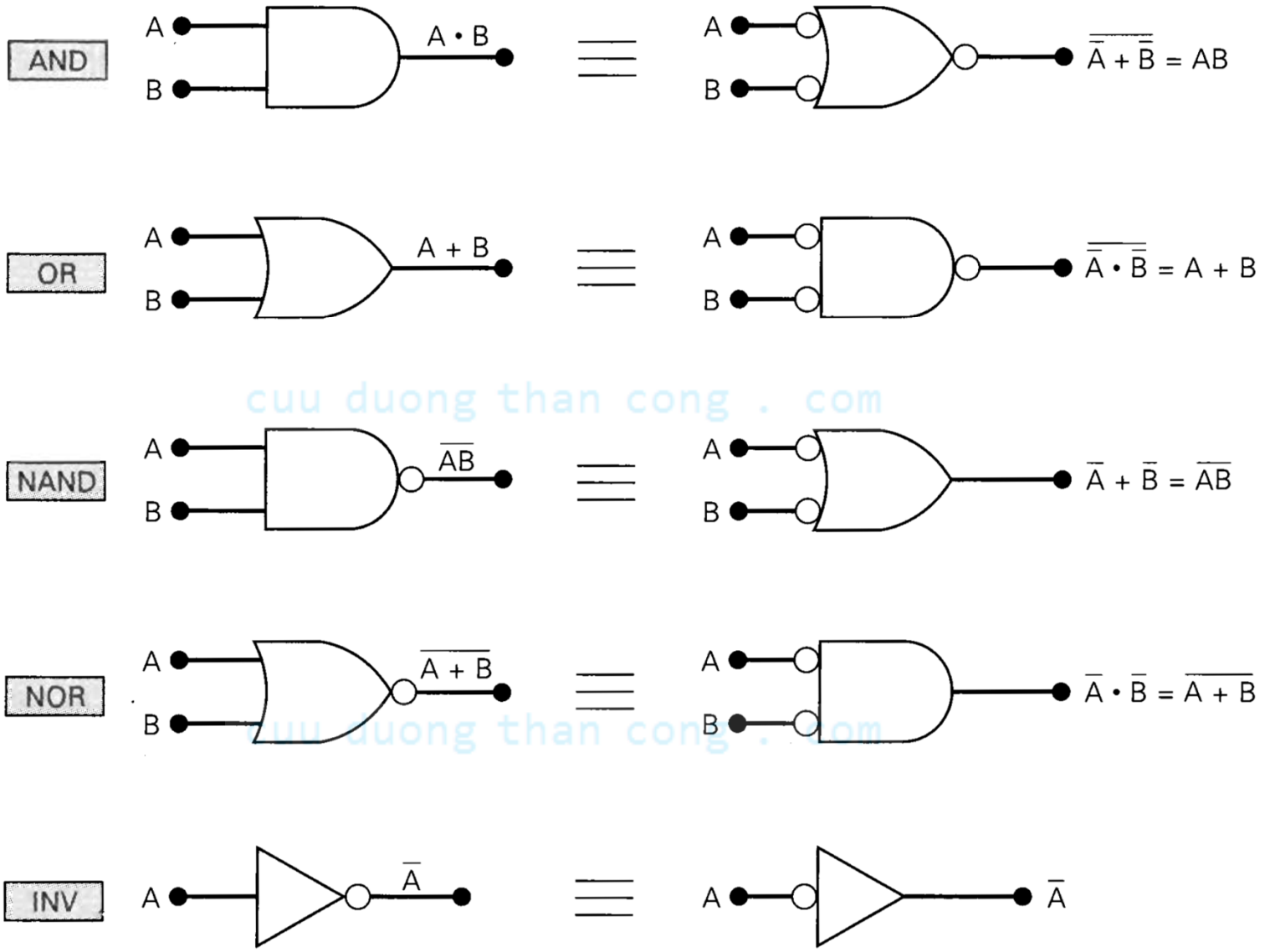


- Cổng **XNOR**



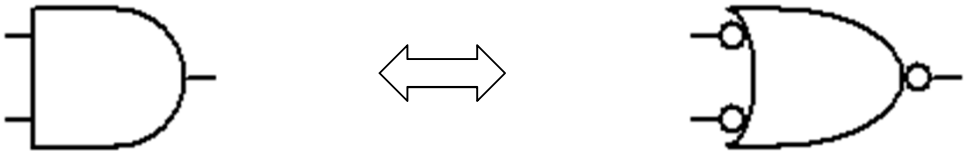
- Các cổng nhiều hơn 2 ngõ nhập

Dạng tương đương



Diễn dịch biểu tượng cổng luận lý

- Dạng tương đương của cổng **AND**



  - Ngõ xuất ở mức cao khi tất cả các ngõ nhập ở mức cao
  - Ngõ xuất ở mức thấp khi một trong các ngõ nhập ở mức thấp

- Một số cấu trúc của cổng **XOR**
  - $E = x \oplus y = x y' + x' y = (x y + x' y')'$

Tích cực cao – Tích cực thấp

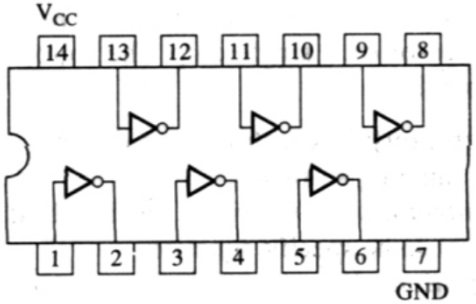
- Hai trạng thái hoạt động của thiết bị là tích cực (*activity*) và không tích cực (*inactivity*)
  - Xét các thí dụ đối với điện thoại, đèn, động cơ, v.v…
- Tích cực cao (*active high*)

  tích cực → luận lý **1** → mức điện áp cao **H**
- Tích cực thấp (*active low*)

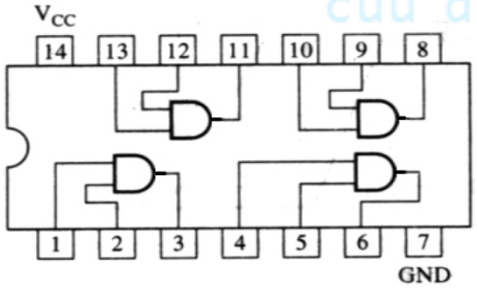
  tích cực → luận lý **0** → mức điện áp thấp **L**

Mạch tích hợp

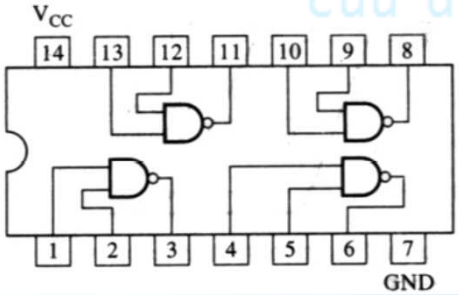
- Cổng **NOT** **7404**



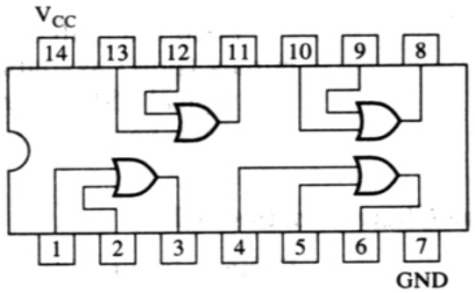
- Cổng **AND** **7408**



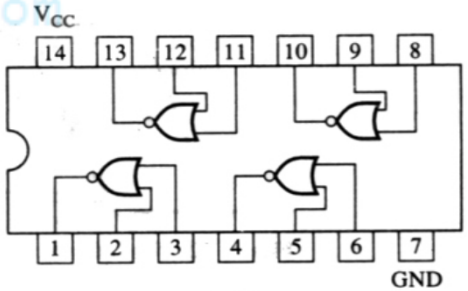
- Cổng **NAND** **7400**



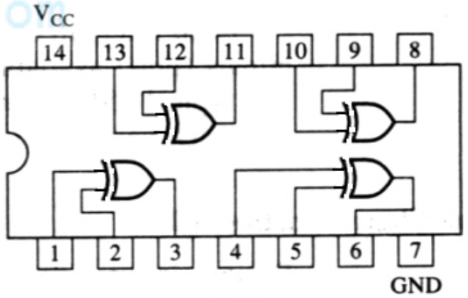
- Cổng **OR** **7432**



- Cổng **NOR** **7402**



- Cổng **Ex-OR** **7486**



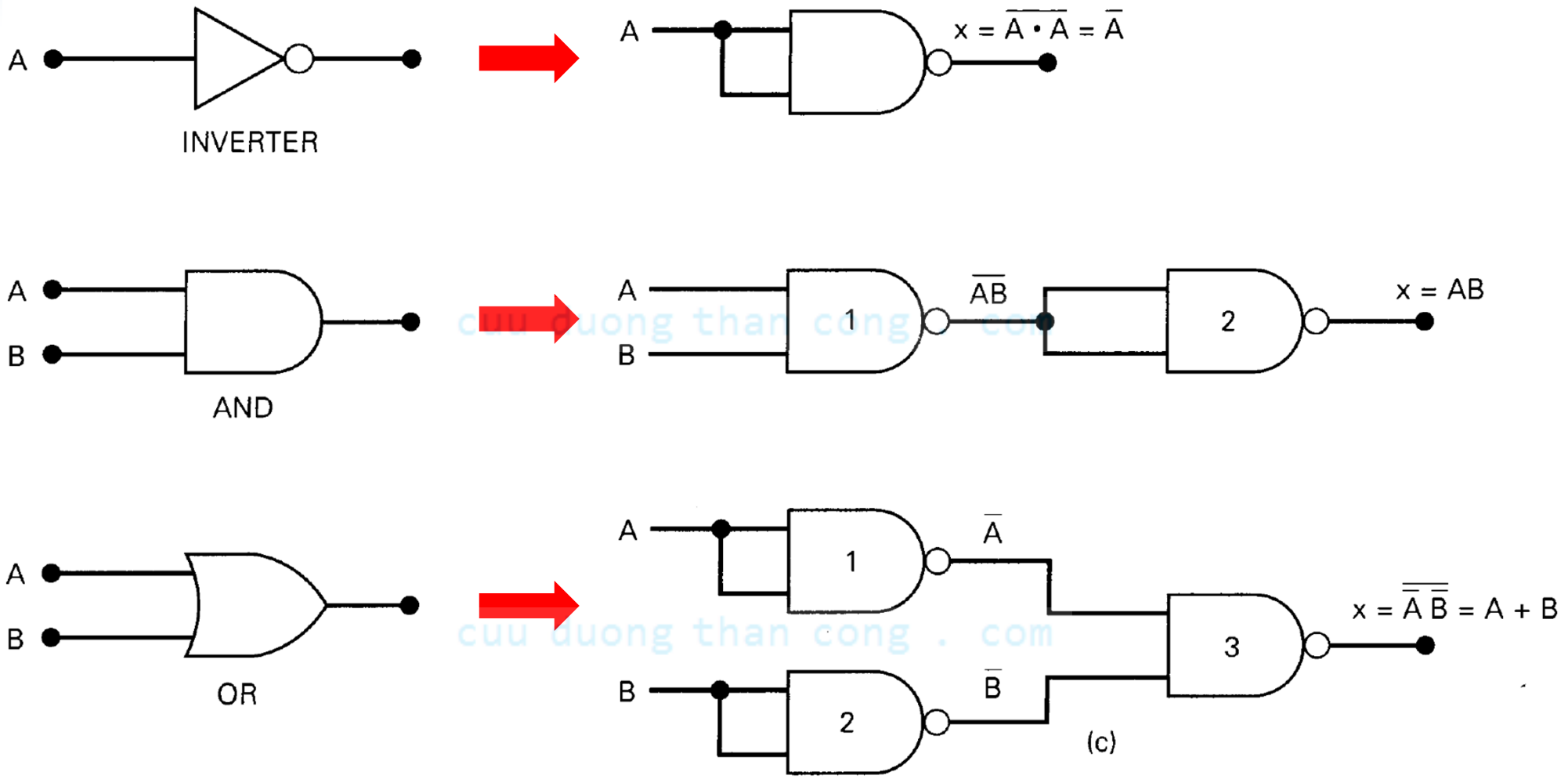
Tập phổ biến của các phép toán

- Một tập các phép toán được gọi là phổ biến (*universal*) nếu mọi hàm chuyển mạch đều có thể được biểu diễn một cách tường minh chỉ bởi các phép toán của tập trên
- Đối với các phép toán chuyển mạch đã xét, ta có một số các tập phổ biến sau
  - Tập { **NOT** , **AND** , **OR** }
  - Tập { **NOT** , **AND** }
  - Tập { **NOT** , **OR** }
  - Tập { **NAND** }
  - Tập { **NOR** }
  - Tập **...**

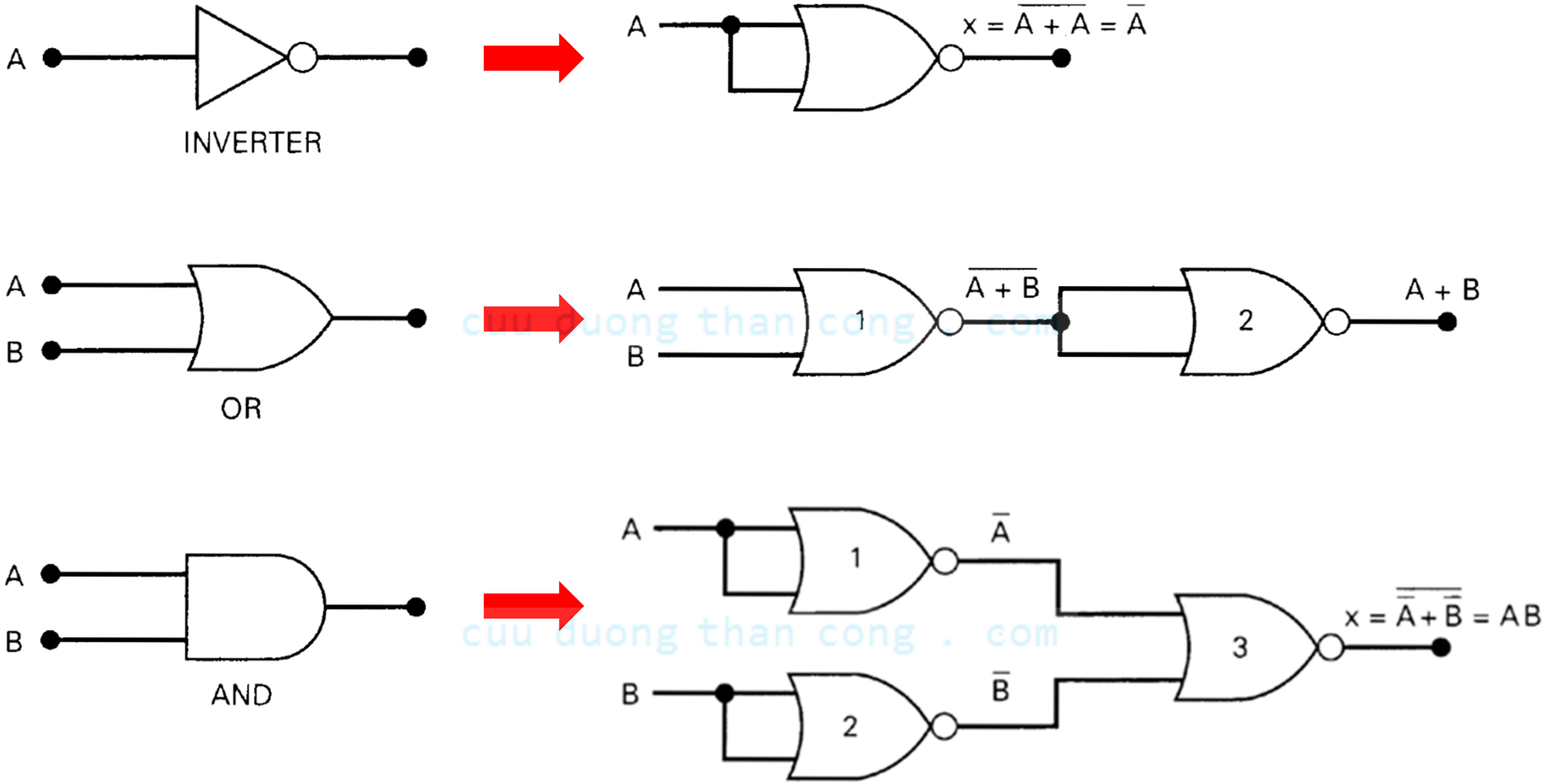
Bất kỳ hàm chuyển mạch nào cũng đều có thể được biểu diễn một cách tường minh chỉ bởi các phép toán ***NOT và AND***

...

Tính phổ biến của cổng NAND



Tính phổ biến của cổng NOR

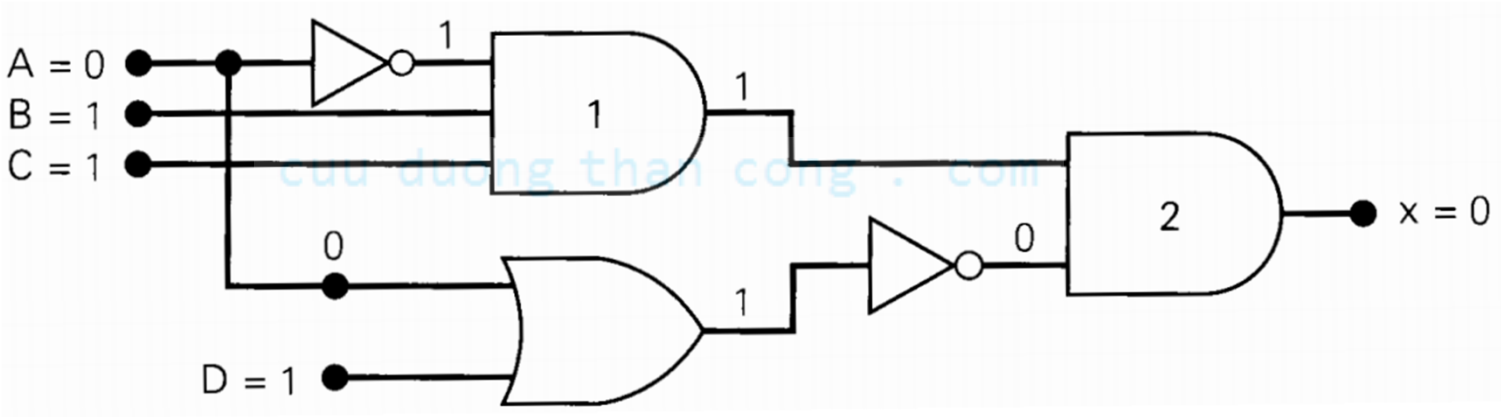


Xác định giá trị ngõ xuất mạch luận lý

- Sử dụng biểu thức Boole cho ngõ xuất của mạch luận lý
  - Với $A = 0, B = 1, C = 1, D = 1$

$$x = A' B C (A + D)'$$
$$= 0'. 1 . 1 . (0 + 1)'$$
$$= 1 . 1 . 1 . 1' = 1 . 1 . 1 . 0 = 0$$

- Sử dụng trực tiếp sơ đồ mạch luận lý mà không cần sử dụng biểu thức Boolean



Giản đồ xung theo thời gian (Timing Waveform)