



# **MẠCH LOGIC**


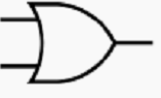

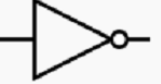



**Môn học: Kiến trúc máy tính & Hợp ngữ**

# Mạch số

- Là thiết bị điện tử hoạt động với **2 mức điện áp**:
  - **Cao**: thể hiện bằng giá trị luận lý (quy ước) là **1**
  - **Thấp**: thể hiện bằng giá trị luận lý (quy ước) là **0**
- Được xây dựng từ những thành phần cơ bản là **cổng luận lý (logic gate)**
  - Cổng luận lý là thiết bị điện tử gồm 1 / nhiều tín hiệu đầu vào (input) - 1 tín hiệu đầu ra (output)
  - **$output = F(input_1, input_2, \dots, input_n)$**
  - Tùy thuộc vào cách xử lý của hàm F sẽ tạo ra nhiều loại cổng luận lý
- Hiện nay linh kiện cơ bản để tạo ra mạch số là **transistor**

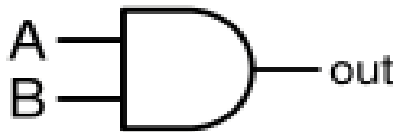


# Cổng luận lý (Logic gate)

Tên cổng	Hình vẽ đại diện	Hàm đại số Bun
AND		$x.y$ hay $xy$
OR		$x + y$
XOR		$x \oplus y$
NOT		$x'$ hay $\bar{x}$
NAND		$(x.y)'$ hay $\overline{x.y}$
NOR		$(x + y)'$ hay $\overline{x + y}$
NXOR		$(x \oplus y)'$ hay $\overline{x \oplus y}$

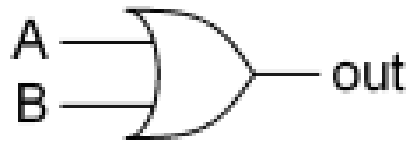
# Bảng chân trị

## AND



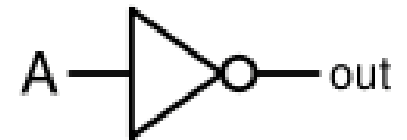
A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

## OR



A	B	out
0	0	0
0	1	1
1	0	1
1	1	1

## NOT



A	out
0	1
1	0

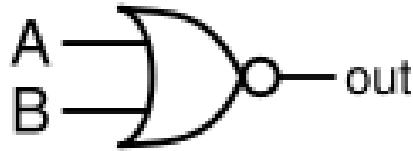
# Bảng chân trị

## NAND



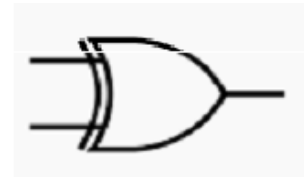
A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

## NOR



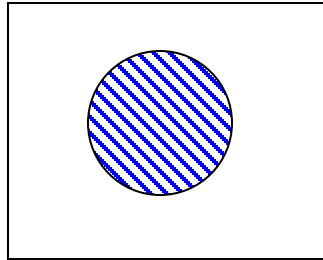
A	B	out
0	0	1
0	1	0
1	0	0
1	1	0

## XOR

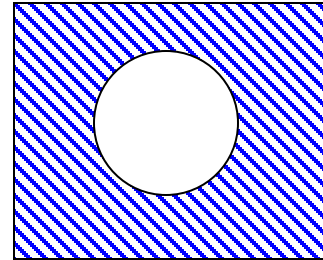


A	B	out
0	0	0
0	1	1
1	0	1
1	1	0

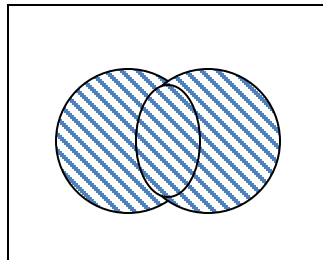
# Lược đồ Venn



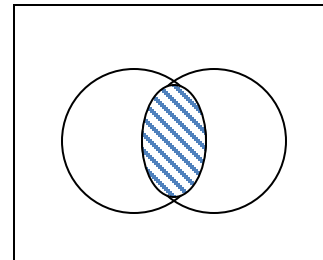
$A$



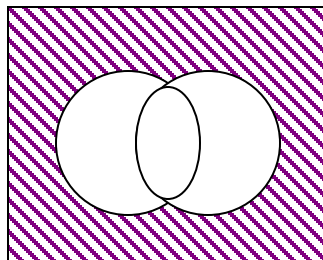
$\overline{A}$



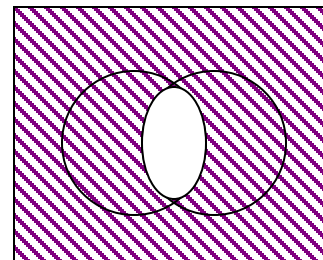
$A+B$



$A.B$

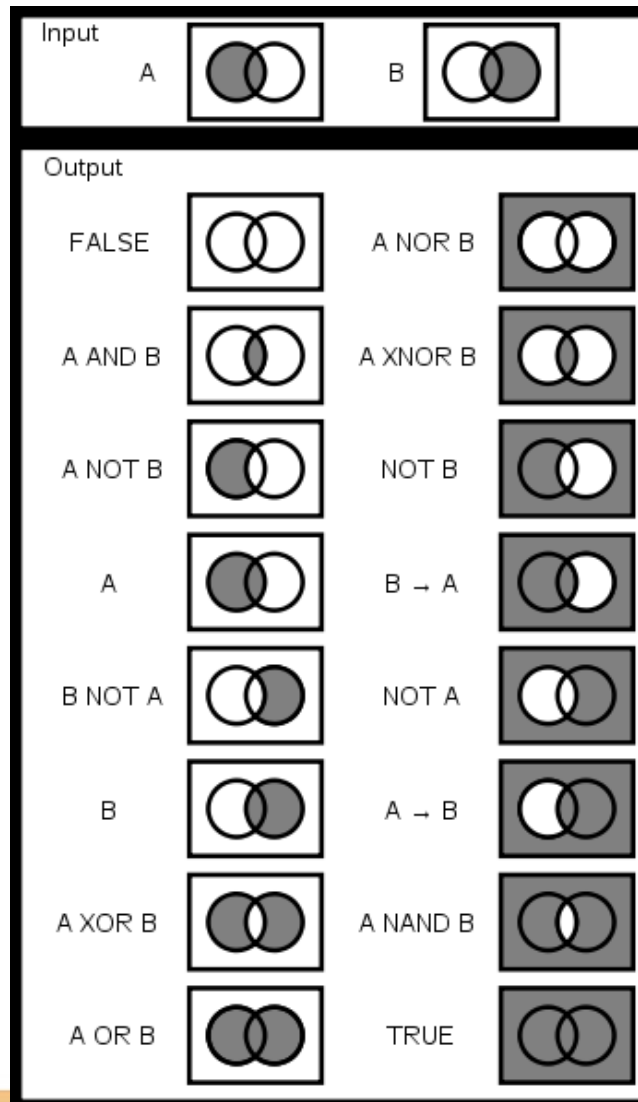


$\overline{A.B}$

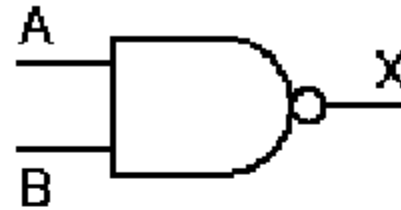
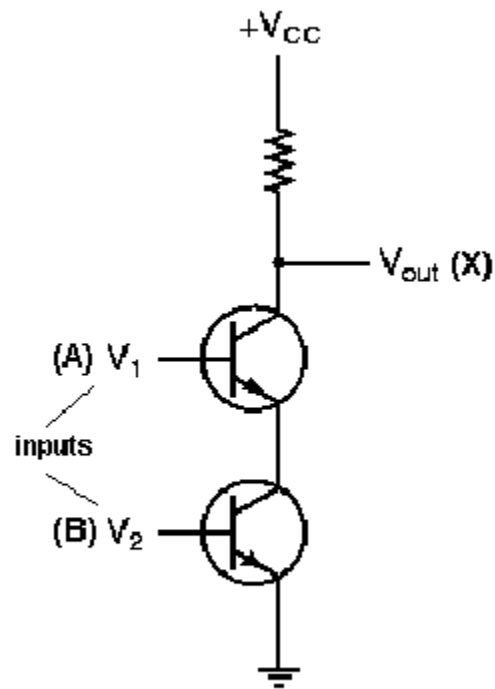


$\overline{A+B}$

# Lược đồ Venn



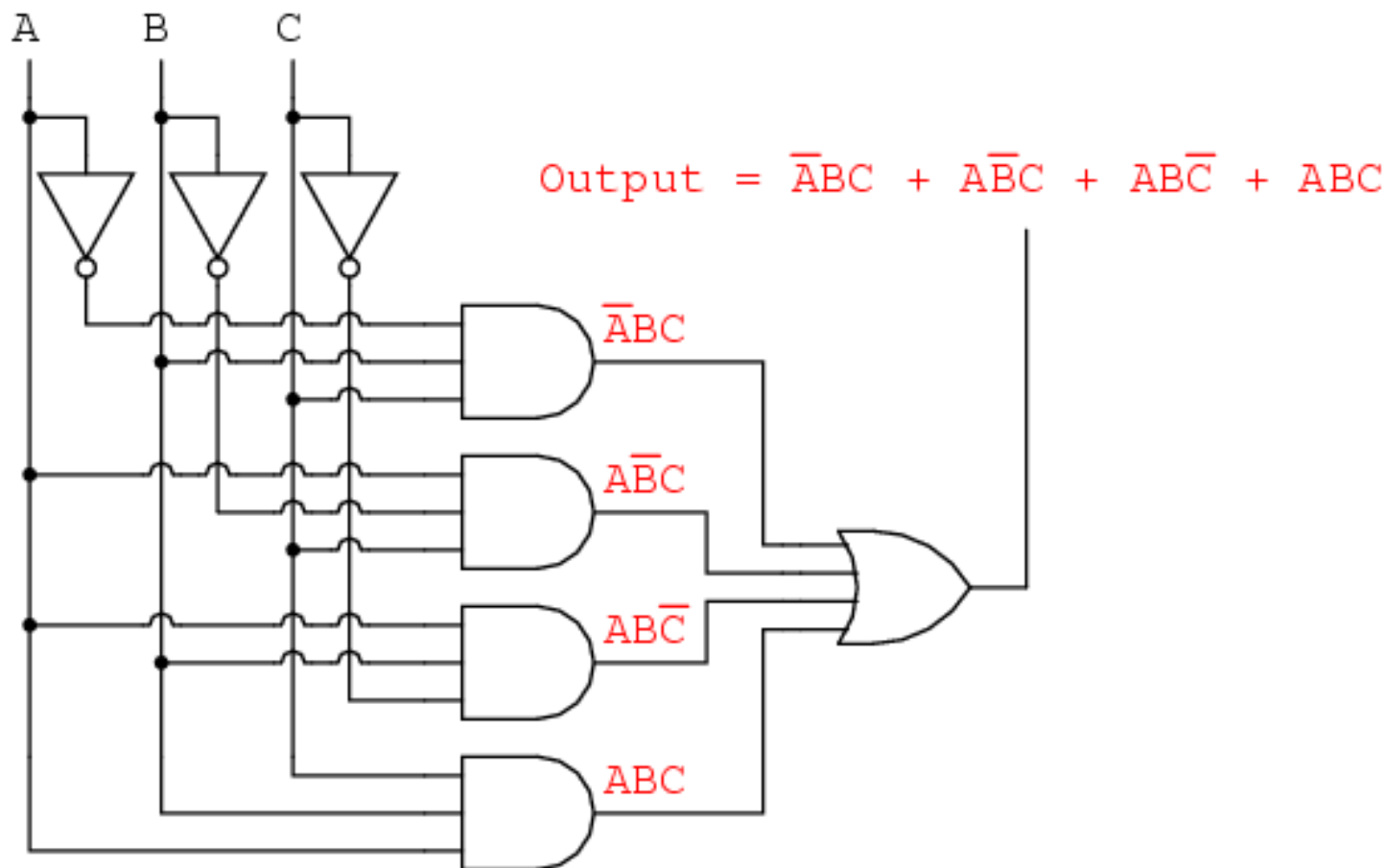
# Ví dụ công luận lý



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



# Ví dụ mạch số

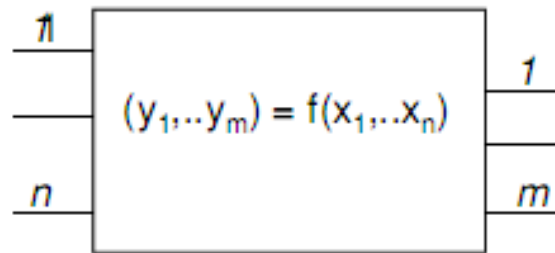


# Một số đẳng thức cơ bản

$x + 0 = x$	$x \cdot 0 = 0$
$x + 1 = 1$	$x \cdot 1 = x$
$x + x = x$	$x \cdot x = x$
$x + x' = 1$	$x \cdot x' = 0$
$x + y = y + x$	$xy = yx$
$x + (y + z) = (x + y) + z$	$x(yz) = (xy)z$
$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
$(x + y)' = x' \cdot y'$ (De Morgan)	$(xy)' = x' + y'$ (De Morgan)
$(x')' = x$	

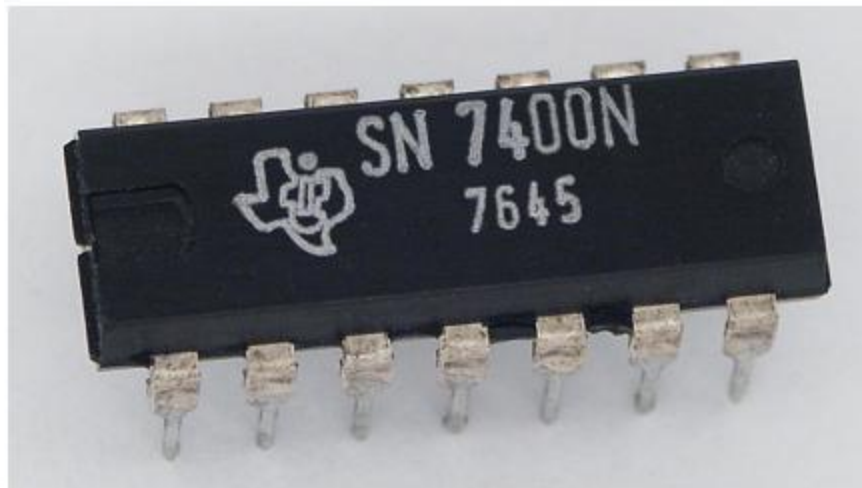
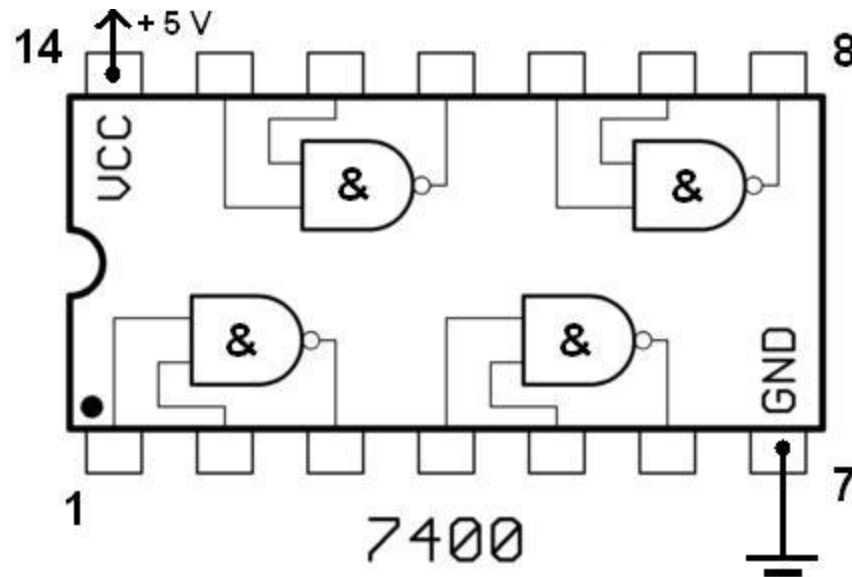
# Mạch tổ hợp (tích hợp)

- Gồm  $n$  ngõ vào (input);  $m$  ngõ ra (output)
  - Mỗi ngõ ra là 1 hàm luận lý của các ngõ vào



- Mạch tổ hợp không mang tính ghi nhớ:  
Ngõ ra chỉ phụ thuộc vào Ngõ vào hiện tại, không xét những giá trị trong quá khứ

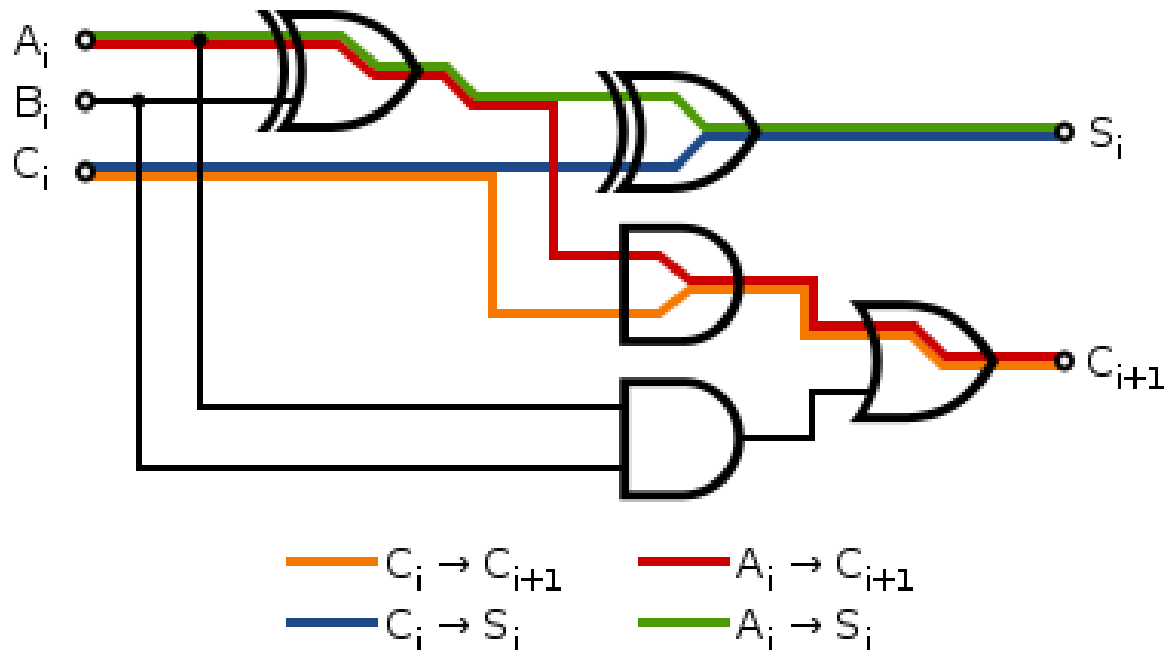
# Ví dụ mạch tổ hợp



- Chip 7400 có 4 cổng NAND.
- Hai chân bổ sung cung cấp nguồn (+5 V) và nối đất (GND).

# Độ trễ mạch

- Độ trễ mạch (**Propagation delay / gate delay**) = Thời điểm tín hiệu ra ổn định - thời điểm tín hiệu vào ổn định
  - Mục tiêu thiết kế mạch: làm giảm thời gian độ trễ mạch



# Mô tả mạch tổ hợp

- Bảng ngôn ngữ
- Bảng bảng chân trị
  - $n$  input –  $m$  output
  - $2^n$  hàng –  $(n + m)$  cột
- Bảng công thức (hàm luận lý)
- Bảng sơ đồ



# Thiết kế

- Thường trải qua 3 bước:

- Lập bảng chân trị

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

- Viết hàm luận lý

$$F = (AB)'$$

- Vẽ sơ đồ mạch và thử nghiệm



# SOP – Sum of Products

- Giả sử đã có bảng chân trị cho mạch n đầu vào  $x_1, \dots, x_n$  và 1 đầu ra  $f$
- Ta dễ dàng thiết lập công thức (hàm) logic theo thuật toán sau:
  - Ứng với mỗi hàng của bảng chân trị có đầu ra = 1 ta tạo thành 1 tích có dạng  $u_1.u_2...u_n$  với:

$$u_i = \begin{cases} x_i & \text{nếu } x_i = 1 \\ (x_i)' & \text{nếu } x_i = 0 \end{cases}$$

- Cộng các tích tìm được lại thành tổng  $\rightarrow$  công thức của  $f$





# Ví dụ

$x_1$	$x_2$	$x_3$	$f$
0	0	0	0
0	0	1	1 $\rightarrow \bar{x}_1 \cdot \bar{x}_2 \cdot x_3$
0	1	0	1 $\rightarrow \bar{x}_1 \cdot x_2 \cdot \bar{x}_3$
0	1	1	0
1	0	0	0
1	0	1	1 $\rightarrow x_1 \cdot \bar{x}_2 \cdot x_3$
1	1	0	0
1	1	1	0

$$f = \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 x_3$$

# POS – Product of Sum

- Trường hợp số hàng có giá trị **đầu ra = 1**  
**nhiều hơn = 0**, ta có thể đặt  $g = (f)'$
- Viết công thức dạng **SOP** cho  $g$
- Lấy  $f = (g)' = (f')'$  để có công thức dạng POS (Tích các tổng) của  $f$



# Ví dụ

x	y	z	f	g
0	0	0	1	0
0	0	1	1	0
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	0

$$g = \bar{x}.y.\bar{z} + x.\bar{y}.\bar{z}$$

$$f = \bar{g} = (x + \bar{y} + z)(\bar{x} + y + z)$$

# Đơn giản hoá hàm logic

- Sau khi viết được hàm logic, ta có thể vẽ sơ đồ của mạch tổ hợp từ những công luận lý cơ bản
  - Ví dụ:  $f = xy + xz$
- Tuy nhiên ta có thể viết lại hàm logic sao cho sơ đồ mạch sử dụng ít cổng hơn
  - Ví dụ:  $f = xy + xz = x(y + z)$
- Cách đơn giản hoá hàm tổng quát? Một số cách phổ biến:
  - Dùng đại số Bun (Xem lại bảng 1 số đẳng thức cơ bản để áp dụng)
  - Dùng bản đồ Karnaugh (Cac-nô)



# Đại số Boole

- Dùng các phép biến đổi đại số Bun để lược giản hàm logic
- Khuyết điểm:
  - Không có cách làm tổng quát cho mọi bài toán
  - Không chắc kết quả cuối cùng đã tối giản chưa
- Ví dụ: Đơn giản hoá các hàm sau
  - $F(x,y,z) = xyz + x'yz + xy'z + xyz'$



# Bản đồ Karnaugh

- Mỗi tổ hợp biến trong bảng chân trị gọi là bộ trị (tạm hiểu là 1 dòng)
  - Biểu diễn hàm có  $n$  biến thì sẽ cho ra tương ứng  $2^n$  bộ trị, với vị trí các bộ trị được đánh số từ 0
  - Thông tin trong bảng chân trị có thể **cô đọng** bằng cách:
    - Liệt kê vị trí các bộ trị (**minterm**) với giá trị đầu ra = **1** (**SOP**)
    - Liệt kê vị trí các bộ trị (**maxterm**) với giá trị đầu ra = **0** (**POS**)



# Ví dụ

- $F(x,y,z) = m_1 + m_4 + m_5 + m_6 + m_7 = \Sigma(1,4,5,6,7)$
- $F(x,y,z) = M_0M_2M_3 = \Pi(0,2,3)$

Vị trí	x	y	z	minterm	maxterm	F
0	0	0	0	$m0 = x'y'z'$	$M0 = x + y + z$	0
1	0	0	1	$m1 = x'y'z$	$M1 = x + y + z'$	1
2	0	1	0	$m2 = x'yz'$	$M2 = x + y' + z$	0
3	0	1	1	$m3 = x'yz$	$M3 = x + y' + z'$	0
4	1	0	0	$m4 = xy'z'$	$M4 = x' + y + z$	1
5	1	0	1	$m5 = xy'z$	$M5 = x' + y + z'$	1
6	1	1	0	$m6 = xyz'$	$M6 = x' + y' + z$	1
7	1	1	1	$m7 = xyz$	$M7 = x' + y' + z'$	1

# Các dạng bản đồ Karnaugh cơ bản

24

Diagram 1: 2x2 Karnaugh Map

Variables: A (vertical), B (horizontal)

	B	0	1
A	0	0	1
1	1	2	3

Diagram 2: 2x4 Karnaugh Map

Variables: A (vertical), BC (horizontal)

	BC	00	01	11	10
A	0	0	1	3	2
1	1	4	5	7	6

Grouping: C (columns 11 and 10)

Diagram 3: 4x4 Karnaugh Map

Variables: A (vertical), B (horizontal), C (horizontal), D (horizontal)

	CD	00	01	11	10
AB	00	0	1	3	2
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

Grouping: B (rows 01, 11, 10)

Grouping: D (columns 01, 11)





# Ví dụ

- $F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$

		BC			
		00	01	11	10
A	0	0	1	0	0
	1	1	1	1	1

Diagram illustrating the Karnaugh map for the function  $F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$ . The map is a 2x4 grid with rows labeled A (0, 1) and columns labeled BC (00, 01, 11, 10). The values in the cells are: (0,00)=0, (0,01)=1, (0,11)=0, (0,10)=0; (1,00)=1, (1,01)=1, (1,11)=1, (1,10)=1. The map is annotated with a bracket labeled B above the columns 11 and 10, and a bracket labeled C below the columns 01, 11, and 10. The row A=1 is also bracketed.

=

		BC			
		00	01	11	10
A	0		1		
	1	1	1	1	1

Diagram illustrating the simplified Karnaugh map for the function  $F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$ . The map is a 2x4 grid with rows labeled A (0, 1) and columns labeled BC (00, 01, 11, 10). The values in the cells are: (0,00)=, (0,01)=1, (0,11)=, (0,10)=; (1,00)=1, (1,01)=1, (1,11)=1, (1,10)=1. The map is annotated with a bracket labeled B above the columns 11 and 10, and a bracket labeled C below the columns 01, 11, and 10. The row A=1 is also bracketed.

# Nhận xét

- Bộ trị giữa 2 ô liền kề trong bản đồ chỉ khác nhau 1 biến
  - Biến đó bù 1 ô, không bù ở ô kế hoặc ngược lại
- Các ô đầu / cuối của các dòng / cột là các ô liền kề
- 4 ô nằm ở 4 góc bản đồ cũng coi là ô liền kề



# Đơn giản hàm theo dạng SOP

- Hàm logic F biểu diễn bảng chân trị được đưa vào bản đồ bằng các trị 1 tương ứng
- Các ô liên kề có giá trị 1 được gom thành nhóm sao cho mỗi nhóm sau khi gom có tổng số ô là lũy thừa của 2 (2, 4, 8,...)
- Các nhóm có thể dùng chung ô có giá trị 1 để tạo thành nhóm lớn hơn. Cố gắng tạo những nhóm lớn nhất có thể
- Nhóm 2/4/8 ô sẽ đơn giản bớt 1/2/3 biến trong số hạng
- Mỗi nhóm biểu diễn 1 số hạng nhân (Product), Cộng (Sum – OR) các số hạng này ta sẽ được biểu thức tối giản của hàm logic F



# Ví dụ 1

- $F(A, B, C) = \Sigma(3, 4, 6, 7)$

		B			
		00	01	11	10
A	0			1	
	1	1		1	1

Diagram illustrating the Karnaugh map for the function  $F(A, B, C) = \Sigma(3, 4, 6, 7)$ . The map is a 2x4 grid with rows labeled A (0, 1) and columns labeled BC (00, 01, 11, 10). The function is 1 for minterms 3, 4, 6, and 7, which are marked with red '1's. The map is grouped into two groups: a group of four cells (A=1, B=0, C=0, 1) and a group of two cells (A=0, B=1, C=0, 1).

		B			
		00	01	11	10
A	0			1	
	1	1		1	1

Diagram illustrating the Karnaugh map for the function  $F(A, B, C) = \Sigma(3, 4, 6, 7)$ . The map is a 2x4 grid with rows labeled A (0, 1) and columns labeled BC (00, 01, 11, 10). The function is 1 for minterms 3, 4, 6, and 7, which are marked with red '1's. The map is grouped into two groups: a group of four cells (A=1, B=0, C=0, 1) and a group of two cells (A=0, B=1, C=0, 1). The groups are highlighted with blue and yellow boxes.

$$F(A, B, C) = BC + AC'$$

# Ví dụ 2

- $F(A, B, C) = \Sigma(0, 2, 4, 5, 6)$

		B					
		BC		00	01	11	10
A	0	1					1
	1	1	1				1

Diagram illustrating the Karnaugh map for the function  $F(A, B, C) = \Sigma(0, 2, 4, 5, 6)$ . The map is a 2x4 grid with rows labeled A (0, 1) and columns labeled BC (00, 01, 11, 10). The values in the cells are 1 for (A=0, BC=00), (A=0, BC=10), (A=1, BC=00), (A=1, BC=01), and (A=1, BC=10). Brackets indicate the grouping of cells for the prime implicants: a horizontal bracket for C' (covering A=0) and a vertical bracket for AB' (covering A=1).

		B					
		BC		00	01	11	10
A	0	1					1
	1	1	1				1

Diagram illustrating the Karnaugh map for the function  $F(A, B, C) = \Sigma(0, 2, 4, 5, 6)$  with prime implicants highlighted. The map is a 2x4 grid with rows labeled A (0, 1) and columns labeled BC (00, 01, 11, 10). The values in the cells are 1 for (A=0, BC=00), (A=0, BC=10), (A=1, BC=00), (A=1, BC=01), and (A=1, BC=10). Brackets indicate the grouping of cells for the prime implicants: a horizontal bracket for C' (covering A=0) and a vertical bracket for AB' (covering A=1). The prime implicants are highlighted: a yellow box for C' (covering A=0) and a green box for AB' (covering A=1).

$$F(A, B, C) = C' + AB'$$

# Ví dụ

- $F(A, B, C, D) = \Sigma(0, 1, 2, 6, 8, 9, 10)$

		C			
		00	01	11	10
A	00	1	1		1
	01				1
	11				
	10	1	1		1
		D			

Diagram illustrating the Karnaugh map for the function  $F(A, B, C, D) = \Sigma(0, 1, 2, 6, 8, 9, 10)$ . The map is a 4x4 grid with rows labeled AB (00, 01, 11, 10) and columns labeled CD (00, 01, 11, 10). The function is represented by 1s in the following cells: (00,00), (00,01), (00,10), (01,10), (10,00), (10,01), and (10,10). Brackets indicate the groups used for simplification: A (rows 00, 01, 11, 10), B (columns 00, 01, 11, 10), C (columns 00, 01, 11, 10), and D (columns 00, 01, 11, 10).

		C			
		00	01	11	10
A	00	1	1		1
	01				1
	11				
	10	1	1		1
		D			

Diagram illustrating the Karnaugh map for the function  $F(A, B, C, D) = \Sigma(0, 1, 2, 6, 8, 9, 10)$ . The map is a 4x4 grid with rows labeled AB (00, 01, 11, 10) and columns labeled CD (00, 01, 11, 10). The function is represented by 1s in the following cells: (00,00), (00,01), (00,10), (01,10), (10,00), (10,01), and (10,10). Brackets indicate the groups used for simplification: A (rows 00, 01, 11, 10), B (columns 00, 01, 11, 10), C (columns 00, 01, 11, 10), and D (columns 00, 01, 11, 10). The 1s are highlighted with colored boxes: yellow for (00,00), (00,01), (10,00), and (10,01); green for (00,10) and (01,10); and blue for (10,10).

$$F(A, B, C, D) = B'D' + B'C' + A'CD'$$

# Đơn giản hàm theo dạng POS

- Đôi khi biểu diễn dạng tổng các tích (SOP) sẽ khó làm khi **số bộ trị có đầu ra = 1 < số bộ trị có đầu ra = 0**

→ Dùng phương pháp tích các tổng (POS)

- Hoàn toàn giống phương pháp đơn giản hàm theo dạng SOP, chỉ khác ta **nhóm các ô liền kề = 0** thay vì 1

→ Tìm được  $F'$

→  $F = (F')'$



# Ví dụ 3

- $F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10)$

		C			
		00	01	11	10
A	00	1	1	0	1
	01	0	1	0	0
	11	0	0	0	0
	10	1	1	0	1
	D				B

		C			
		00	01	11	10
A	00	1	1	0	1
	01	0	1	0	0
	11	0	0	0	0
	10	1	1	0	1
	D				B

$$F'(A, B, C) = CD + BD' + AB$$

$$F = (F')' = (A' + B')(C' + D')(B' + D)$$



# Điều kiện không cần / tùy chọn

- Trong 1 số trường hợp ta **không cần quan tâm** đến giá trị ngõ ra của 1 số bộ trị nào đó (**1 hay 0 đều được**)
- Trong bản đồ ta sẽ ghi tương ứng những ô đó là x (gọi là giá trị tùy chọn /không cần)
- **x có thể dùng để gom nhóm với các ô liền kề nhằm đơn giản hàm**
- Lưu ý: Không được gom nhóm bao gồm toàn những ô có giá trị x



# Ví dụ

- $F(A, B, C) = \Sigma(0, 2, 6)$
- $d(A, B, C) = \Sigma(1, 3, 5)$

Vị trí	A	B	C	F
0	0	0	0	1
1	0	0	1	x
2	0	1	0	1
3	0	1	1	x
4	1	0	0	0
5	1	0	1	x
6	1	1	0	1
7	1	1	1	0

# Ví dụ

- $F(A, B, C) = \Sigma(0, 2, 6)$
- $d(A, B, C) = \Sigma(1, 3, 5)$

BC		B			
		00	01	11	10
A	0	1	x	x	1
	1		x		1

Groupings:  $A=0$  (row 0),  $A=1$  (row 1),  $C=0$  (columns 00, 01),  $C=1$  (columns 11, 10).

BC		B			
		00	01	11	10
A	0	1	x	x	1
	1		x		1

Groupings:  $A=0$  (row 0),  $A=1$  (row 1),  $C=0$  (columns 00, 01),  $C=1$  (columns 11, 10).

$$F(A, B, C) = A' + BC'$$

# Bài tập thiết kế mạch tổ hợp

- Yêu cầu: Thiết kế mạch tổ hợp 3 ngõ vào, 1 ngõ ra, sao cho giá trị logic ở ngõ ra là giá trị nào chiếm đa số trong các ngõ vào



# Bước 1: Lập bảng chân trị

- Gọi các ngõ vào là  $x, y, z$  - ngõ ra là  $f$

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$f(x, y, z) = \Sigma(3, 5, 6, 7)$$



# Bước 2: Viết hàm logic

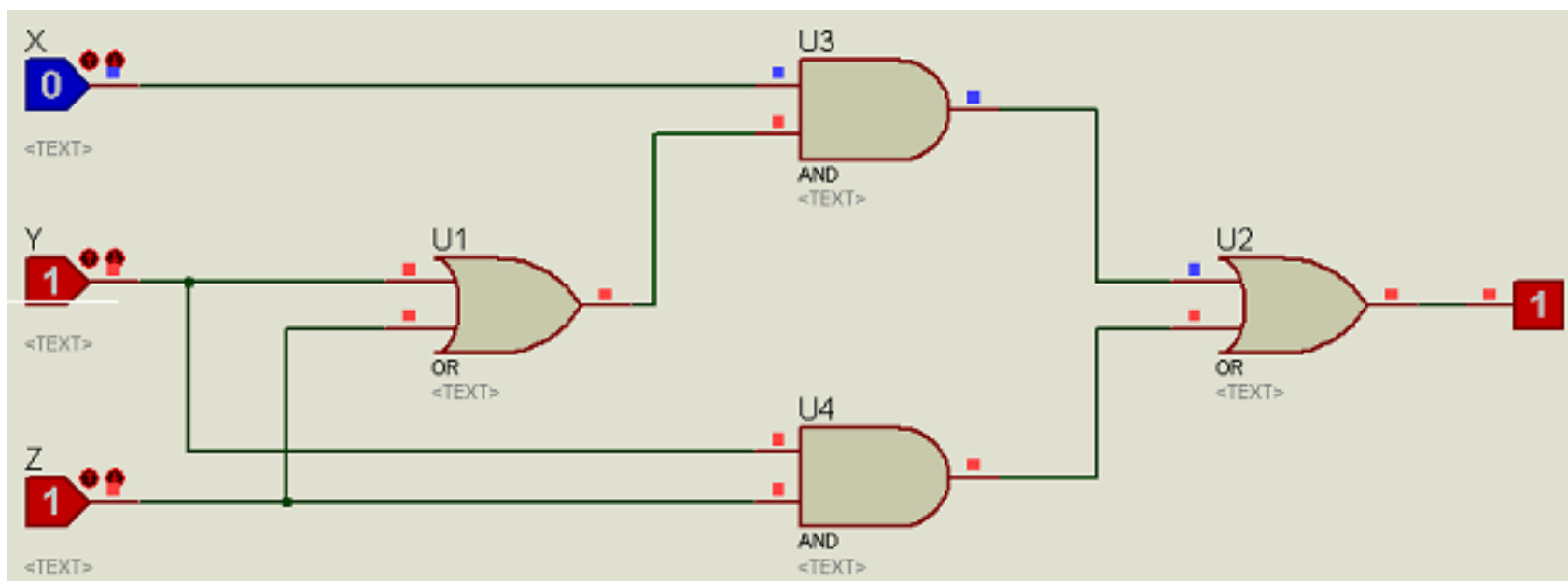
□  $f(x, y, z) = \Sigma(3, 5, 6, 7)$

		y z			
		00	01	11	10
x	0			1	
	1		1	1	1

		y z			
		00	01	11	10
x	0			1	
	1		1	1	1

$$f(x, y, z) = xz + xy + yz = x.(y+z) + yz$$

## Bước 3: Vẽ sơ đồ mạch và Thử nghiệm



## Phần 2: Một số mạch tổ hợp cơ bản

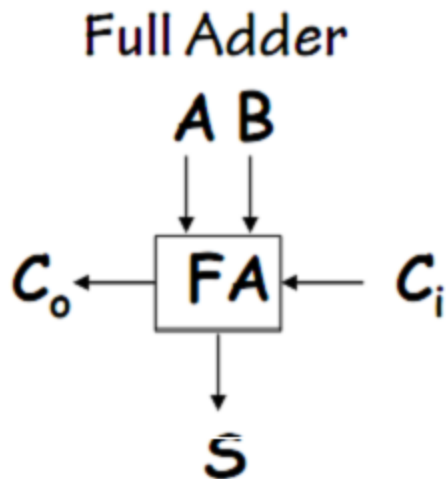
- Mạch toàn cộng (Full adder)
- Mạch giải mã (Decoder)
- Mạch mã hoá (Encoder)





# Mạch toàn cộng (Full adder - FA)

- Mạch tổ hợp thực hiện phép cộng số học 3 bit
- Gồm 3 ngõ vào (A, B: bit cần cộng –  $C_i$ : bit nhớ) và 2 ngõ ra (kết quả có thể từ 0 đến 3 với giá trị 2 và 3 cần 2 bit biểu diễn – S: ngõ tổng,  $C_0$ : ngõ nhớ)



A	B	$C_i$	S	$C_0$
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

$$S = F(A, B, C_i) \\ = \Sigma(1, 2, 4, 7)$$

$$C_0 = F(A, B, C_i) \\ = \Sigma(3, 5, 6, 7)$$

# Bước 2: Viết hàm logic

		A			
		AB			
		00	01	11	10
Ci	0		1		1
	1	1		1	
		B			

$$S = F(A, B, Ci) = \Sigma(1, 2, 4, 7)$$

$$S = A'BCi' + AB'Ci' + A'B'Ci + ABCi$$

$$S = A \oplus B \oplus Ci$$

$$(\text{Lưu ý: } x \oplus y = x'y + xy')$$

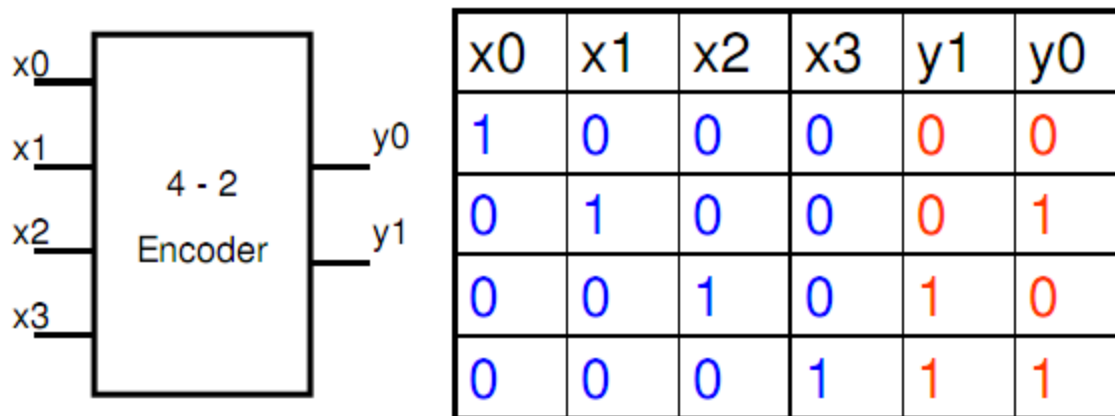
		A			
		AB			
		00	01	11	10
Ci	0			1	
	1		1	1	1
		B			

$$C_0 = F(A, B, Ci) = \Sigma(3, 5, 6, 7)$$

$$C_0 = AB + BCi + ACi$$

# Mạch mã hoá nhị phân (Binary Encoder)

- Có  $2^n$  (hoặc ít hơn) ngõ vào, n ngõ ra
- Quy định chỉ có **duy nhất một ngõ vào mang giá trị = 1** tại một thời điểm
- Nếu ngõ vào = 1 đó là ngõ thứ k thì các ngõ ra tạo thành số nhị phân có giá trị = k

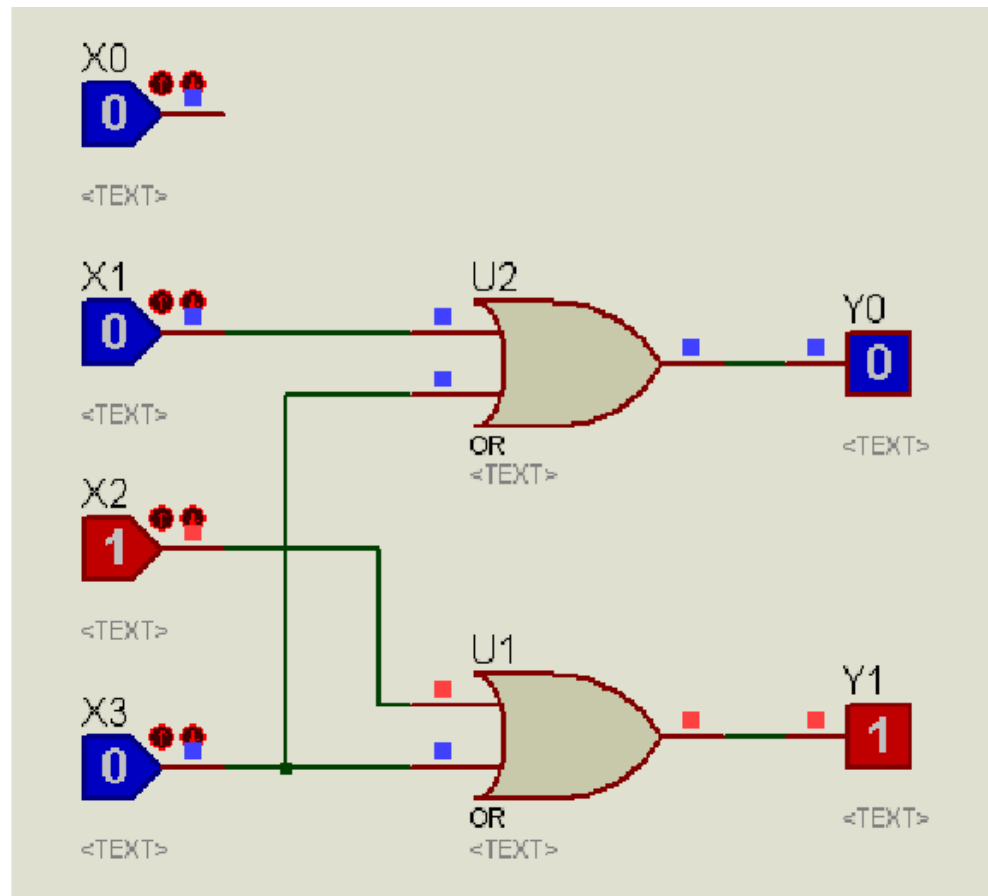


# Sơ đồ mạch 4-2 Binary Encoder

- Ngõ vào:  $X_0, X_1, X_2, X_3$
- Ngõ ra:  $Y_0, Y_1$

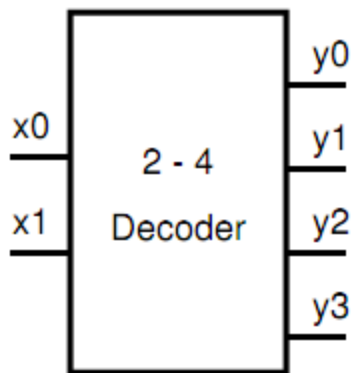
$$Y_0 = X_1 + X_3$$

$$Y_1 = X_2 + X_3$$



# Mạch giải mã (Decoder)

- Có  $n$  ngõ vào,  $2^n$  (hoặc ít hơn) ngõ ra
- Quy định chỉ có **duy nhất một ngõ ra mang giá trị = 1** tại một thời điểm
- Nếu các ngõ vào tạo thành số nhị phân có giá trị =  $k$  thì ngõ ra = 1 đó là ngõ thứ  $k$



$x_1$	$x_0$	$y_0$	$y_1$	$y_2$	$y_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

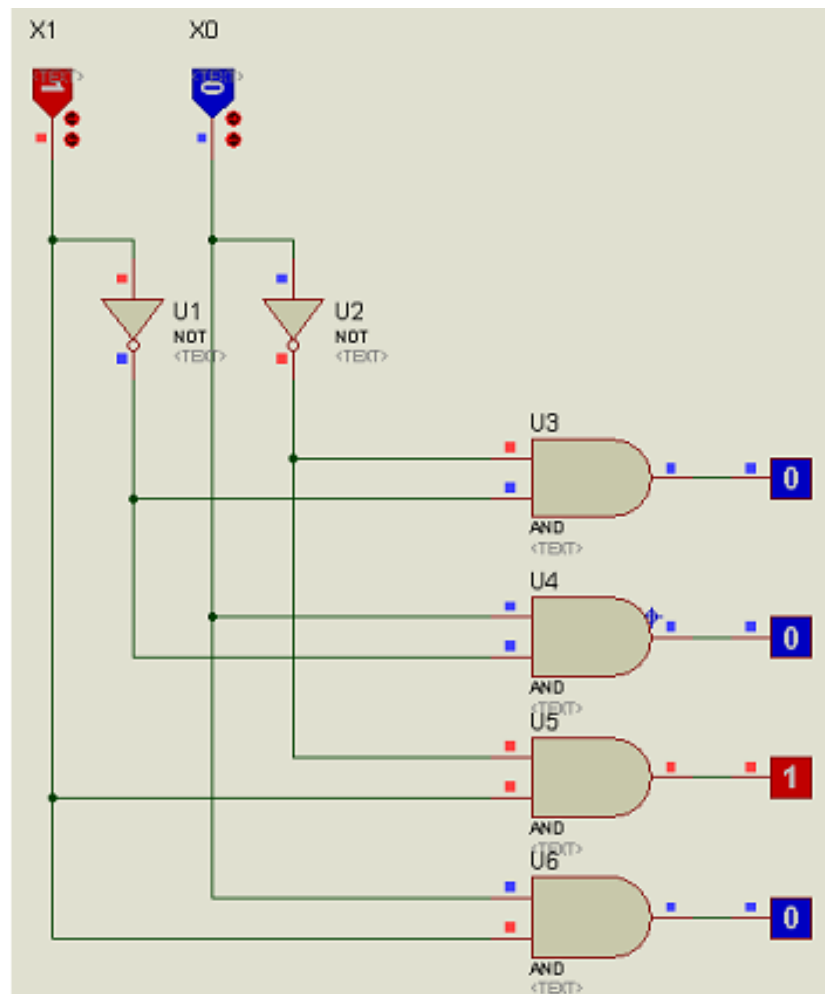
$$y_0 = \overline{x_1} \cdot \overline{x_0}$$

$$y_1 = \overline{x_1} \cdot x_0$$

$$y_2 = x_1 \cdot \overline{x_0}$$

$$y_3 = x_1 \cdot x_0$$

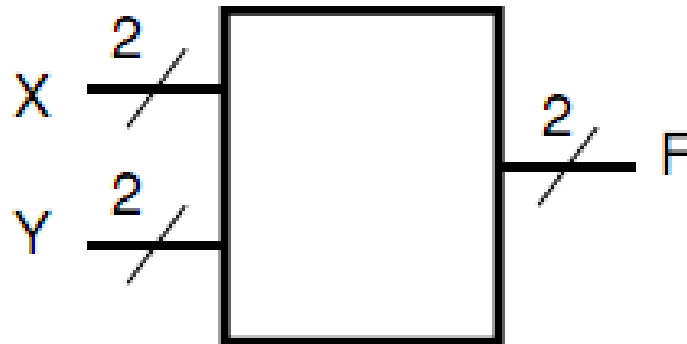
# Sơ đồ mạch 2-4 Decoder



# Bài tập: Thiết kế mạch ALU

- $F = (5X + 2Y) \% 4$
- Input: X (2 bit), Y (2 bit)
- Output: F (2 bit)

→ Có 4 ngõ vào, 2 ngõ ra (mỗi ngõ có 1 tín hiệu biểu diễn cho 1 bit)



# Bước 1: Lập bảng chân trị

X	Y	F
0 (00)	0 (00)	0 (00)
0 (00)	1 (01)	2 (10)
0 (00)	2 (10)	0 (00)
0 (00)	3 (11)	2 (10)
1 (01)	0 (00)	1 (01)
1 (01)	1 (01)	3 (11)
1 (01)	2 (10)	1 (01)
1 (01)	3 (11)	3 (11)
2 (10)	0 (00)	2 (10)
2 (10)	1 (01)	0 (00)
2 (10)	2 (10)	2 (10)
2 (10)	3 (11)	0 (00)
3 (11)	0 (00)	3 (11)
3 (11)	1 (01)	1 (01)
3 (11)	2 (10)	3 (11)
3 (11)	3 (11)	1 (01)





## Bước 2: Xác định hàm

		Y1 Y0			
		00	01	11	10
X1 X0	00	0	1	1	0
	01	0	1	1	0
	11	1	0	0	1
	10	1	0	0	1

$$F1 = X1.Y0' + X1'.Y0$$

$$F0 = X0$$

# Bước 3: Vẽ mạch

