

Digital System Design

MULTIPLEXERS, DECODERS, AND PROGRAMMABLE LOGIC DEVICES

**Bộ dồn kênh, mạch giải mã và thiết
bị logic khả trình**



Outline

- ❖ **9.1 Giới thiệu**
 - ❖ **9.2 Bộ dồn kênh MUX**
 - ❖ **9.3 Cổng đếm ba trạng thái**
 - ❖ **9.4 Mạch giải mã và mạch mã hóa**
 - ❖ **9.5 Bộ nhớ chỉ đọc(ROM)**
 - ❖ **9.6 Thiết bị logic khả trình**
 - ❖ **9.7 Mạch logic khả trình phức hợp**
 - ❖ **9.8 Mạng các cổng khả trình**
-



Mạch tích hợp(IC)

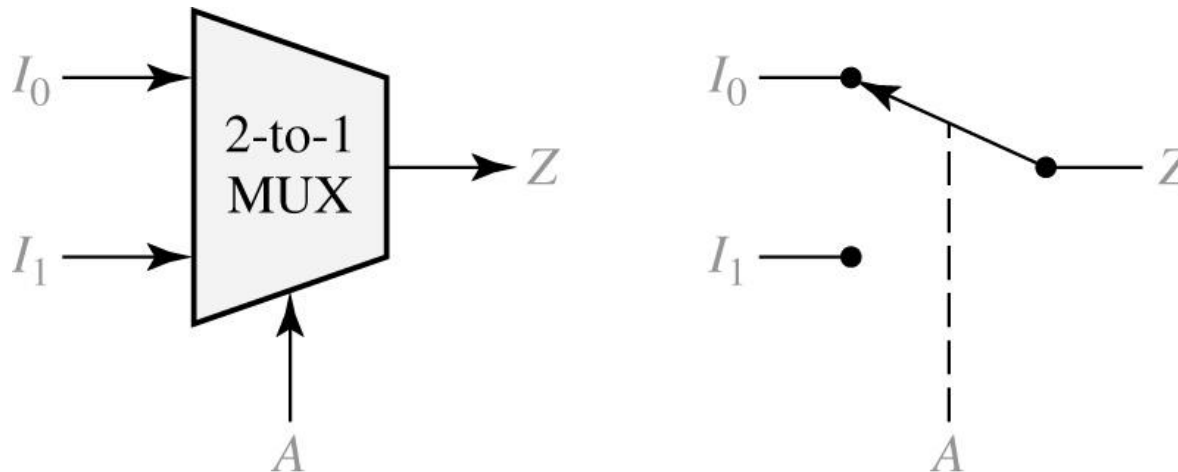
- ❖ Vi mạch có mức độ tích hợp nhỏ(SSI):
 - NOR
 - NAND
 - NOT
 - XOR
- ❖ Vi mạch với mức độ tích hợp trung bình(MSI)
 - Multiplexer
 - Decoder
- ❖ Vi mạch với mức độ tích hợp lớn(LSI) : Arithmetic-Logic Unit (ALU)
- ❖ Vi mạch có độ tích hợp rất lớn(VLSI)

9.1 Giới thiệu

- **Multiplexer, Decoder, encoder. Three-state Buffer(Bộ dồn kênh,mạch giải mã,mạch mã hóa,cổng đệm 3 trạng thái)**
- **ROMs**
- **PLD (Thiết bị logic khả trình-programable logic devices =FPD)**
- **PLA (Mảng logic khả trình-programable logic array)**
- **CPLD (Mạch logic khả trình phức hợp-Complex PLD)**
- **FPGA (Mảng các cổng khả trình)**

9.2 Bộ dồn kênh

Fig 9-1. 2-to-1 Bộ dồn kênh và khóa chuyển mạch tương tự

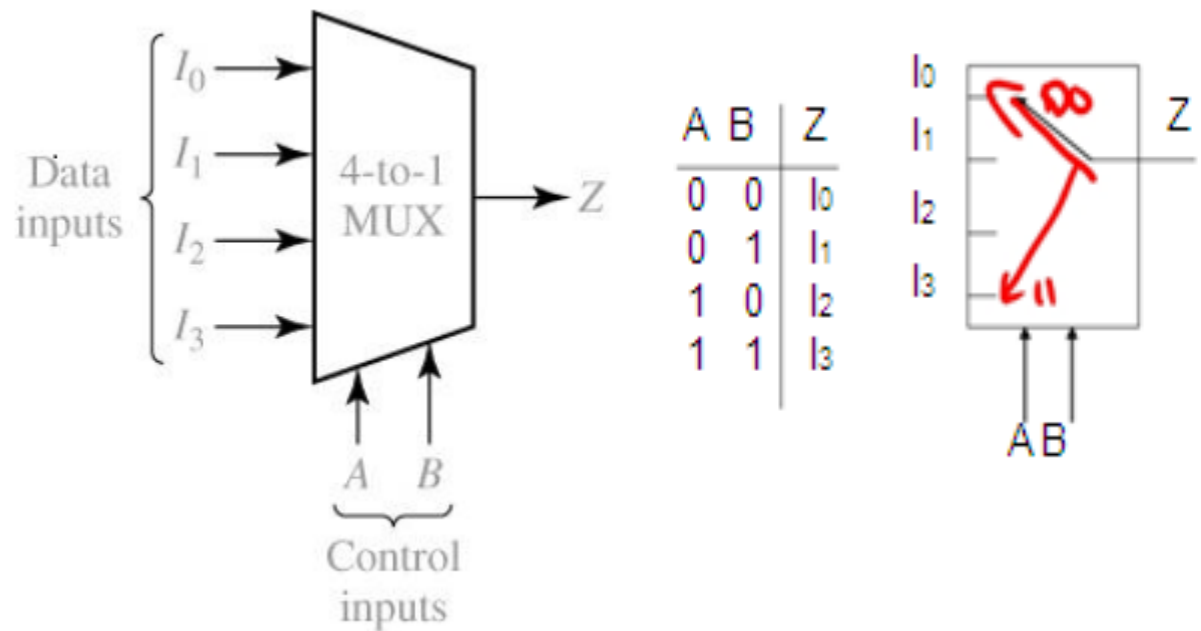


Phương trình logic(pt Boolean) của bộ dồn kênh 2:1

$$Z = A' I_0 + A I_1$$

9.2 Bộ dồn kênh

Fig 9-2. Bộ dồn kênh (1)



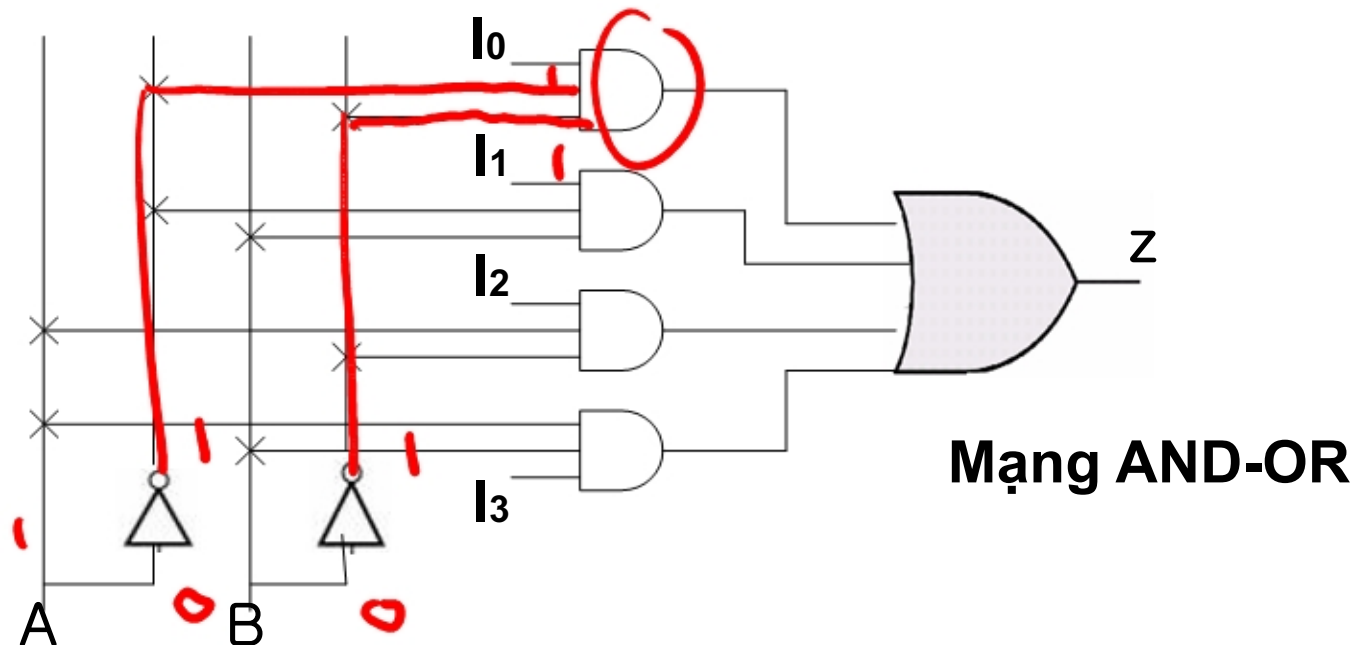
Phương trình logic(pt Boole) của bộ dồn kênh 4:1

$$Z = A' B' I_0 + A' B I_1 + A B' I_2 + A B I_3$$



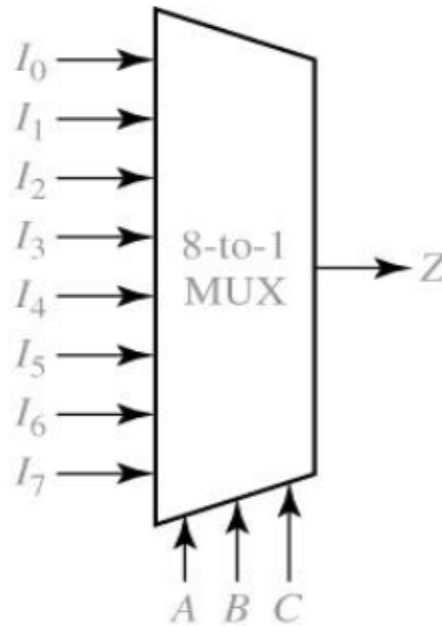
Cách thực hiện bộ dồn kênh

$$Z = \underline{A'B'}l_0 + \underline{A'B}l_1 + \underline{AB'}l_2 + \underline{AB}l_3$$
$$= m_0l_0 + m_1l_1 + m_2l_2 + m_3l_3 \quad (m_i : \text{ith Minterm})$$



9.2 Bộ dồn kênh

Fig 9-2. Bộ dồn kênh (2)

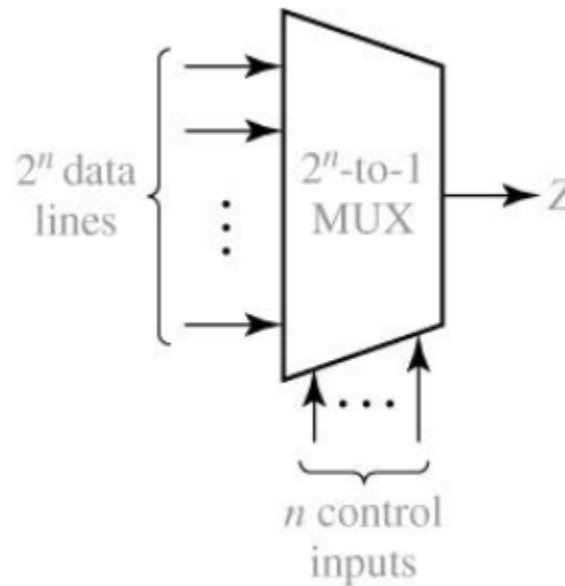


Phương trình logic(pt Boole) của bộ dồn kênh 8:1

$$Z = A'B'C'I_0 + A'B'CI_1 + A'BC'I_2 + A'BCI_3 \\ + AB'C'I_4 + AB'CI_5 + ABC'I_6 + ABCI_7$$

9.2 Bộ dồn kênh

Fig 9-2. Bộ dồn kênh tổng quát $2^n:1$

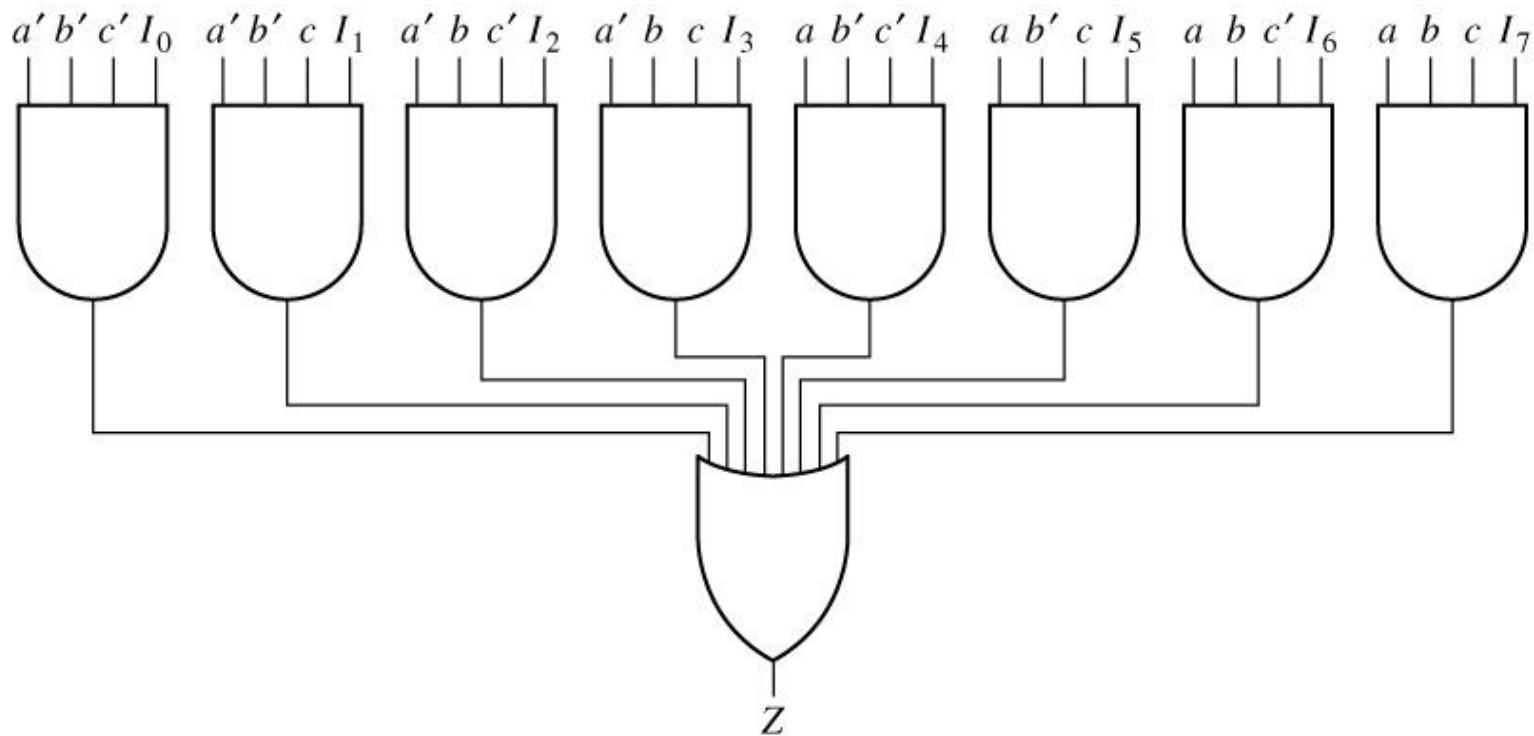


Phương trình logic(pt Boole) của bộ dồn kênh $2^n:1$

$$Z = \sum_{k=0}^{2^n-1} m_k I_k$$

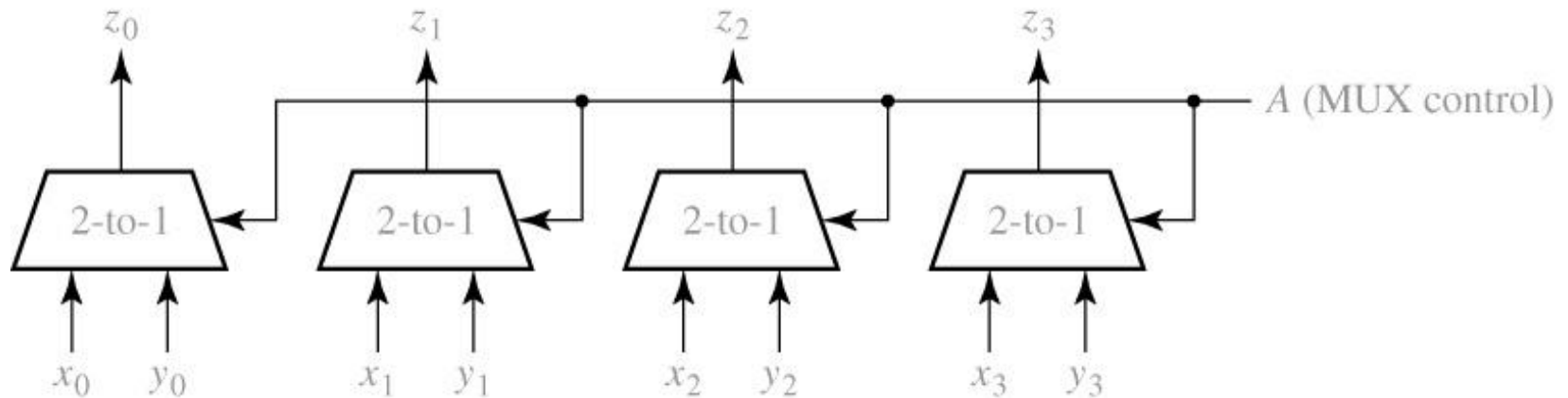
9.2 Bộ dồn kênh

Fig 9-3. Sơ đồ Logic cho bộ dồn kênh 8:1



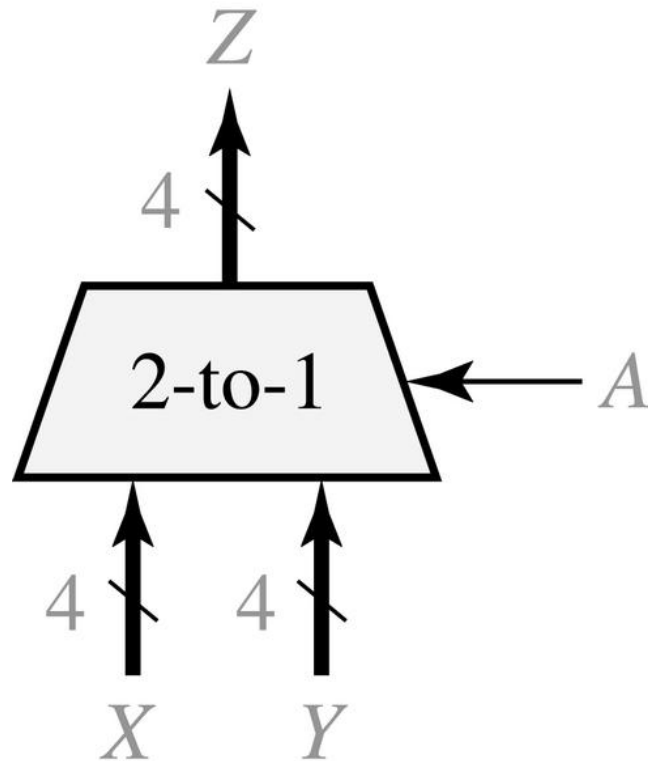
9.2 Bộ dồn kênh

Fig 9-4. Quad Multiplexer Used to Select Data



9.2 Bộ dồn kênh

Fig 9-5. Quad Multiplexer with Bus Inputs and Output



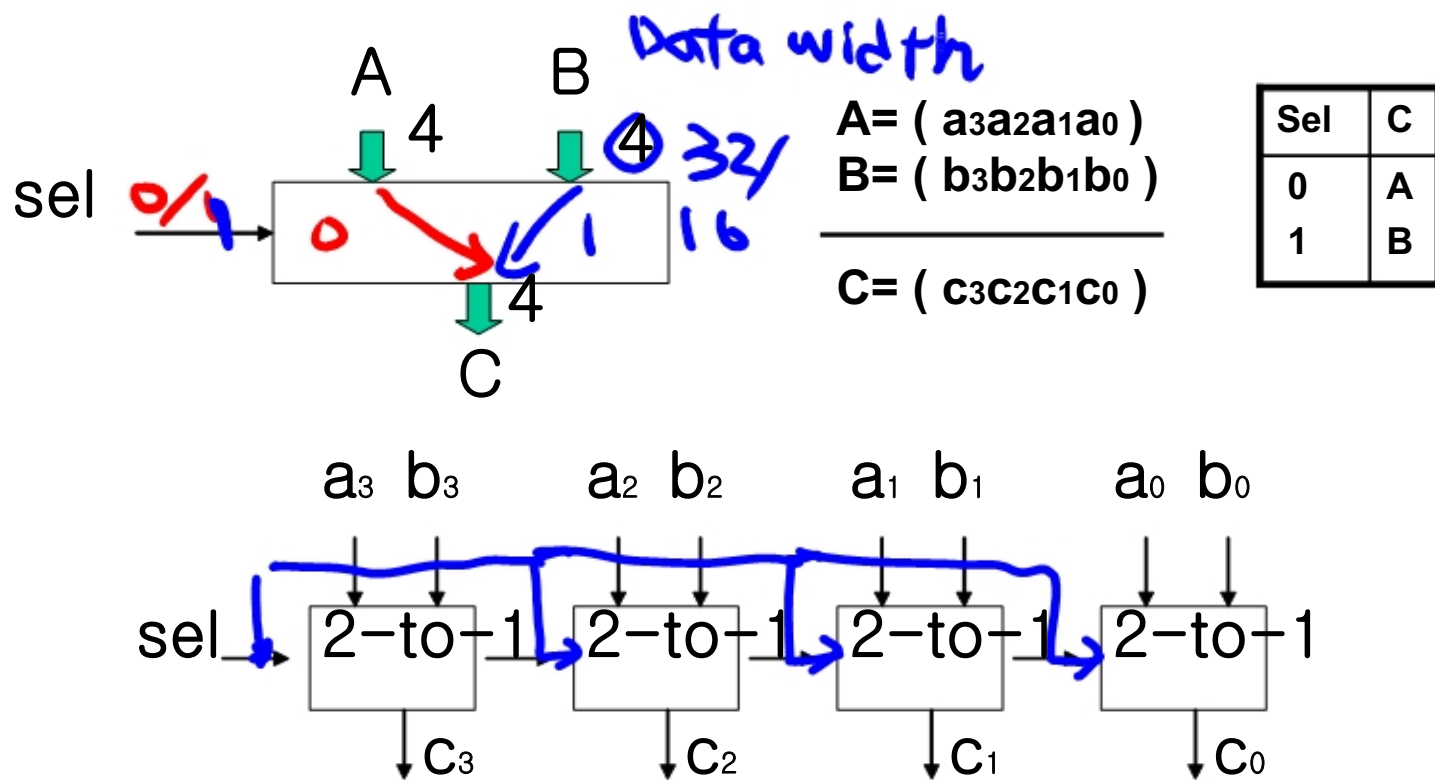
Ứng dụng của bộ dồn kênh và bộ phân kênh

APPLICATION OF MUX AND DEMUX



Ứng dụng: Bộ dồn kênh

(4-bit word selector): $\mathbf{A}=(a_3 a_2 a_1 a_0)$

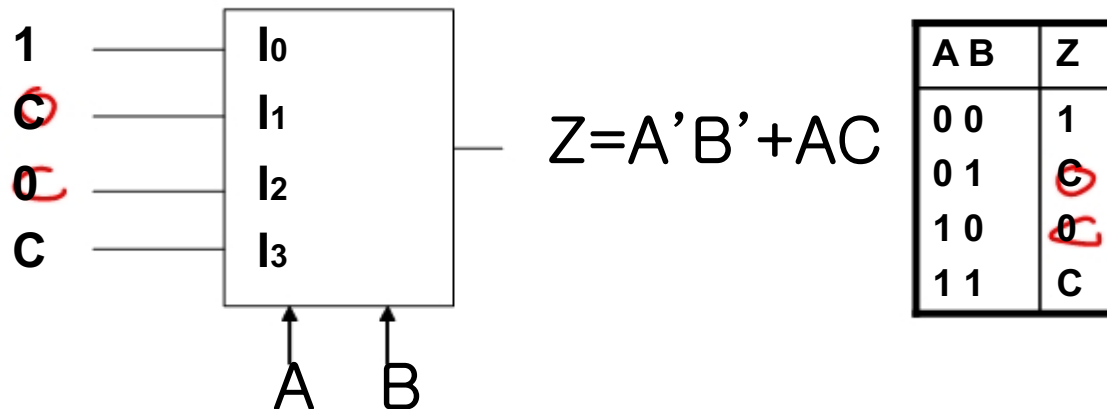




Thực hiện hàm logic tổ hợp

$$\begin{aligned} F(A, B, C) &= A'B' + AC = A'B' + AC(B + B') \\ &= 1 \times \underbrace{A'B'}_{00} + \underbrace{AB'C}_{10} + \underbrace{ABC}_{11} + \underbrace{(A'B) \times 0}_{01} \end{aligned}$$

$m_0 \quad m_2 \quad m_3 \quad m_1$





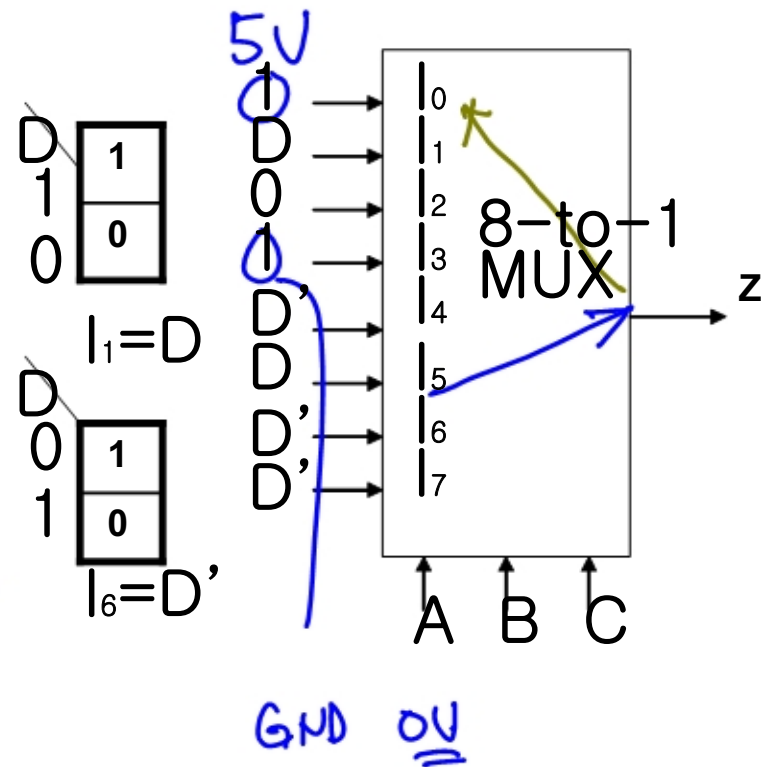
Ví dụ: Sử dụng bộ dồn kênh 8-to-1 thực hiện hàm $F(A,B,C)$

Truth table for $F(A,B,C)$ using an 8-to-1 MUX:

AB	00	01	11	10
CD	00	01	11	10
00	1	0	1	1
01	1	0	0	0
11	1	1	0	1
10	0	1	1	0

Handwritten annotations:

- Green circle: $ABC = 000$
- Red text: $ABC = 001$
- Blue text: $ABC = 101$
- Blue text: I_6





Ví dụ: Sử dụng bộ dồn kênh 8-to-1 thực hiện hàm $F(A,B,C)$

	A B C	Z
0	0 0 0	1
1	0 0 1	D
2	0 1 0	0
3	0 1 1	1
4	1 0 0	D'
5	1 0 1	D
6	1 1 0	D'
7	1 1 1	D'



$$F = A'B'C' + B'CD + A'BC + A'BC + AC'D'$$

(Từ bản đồ K-map)

	A B D	Z
0	0 0 0	C'
1	0 0 1	1
2	0 1 0	C
3	0 1 1	C
4	1 0 0	C'
5	1 0 1	C
6	1 1 0	1
7	1 1 1	0

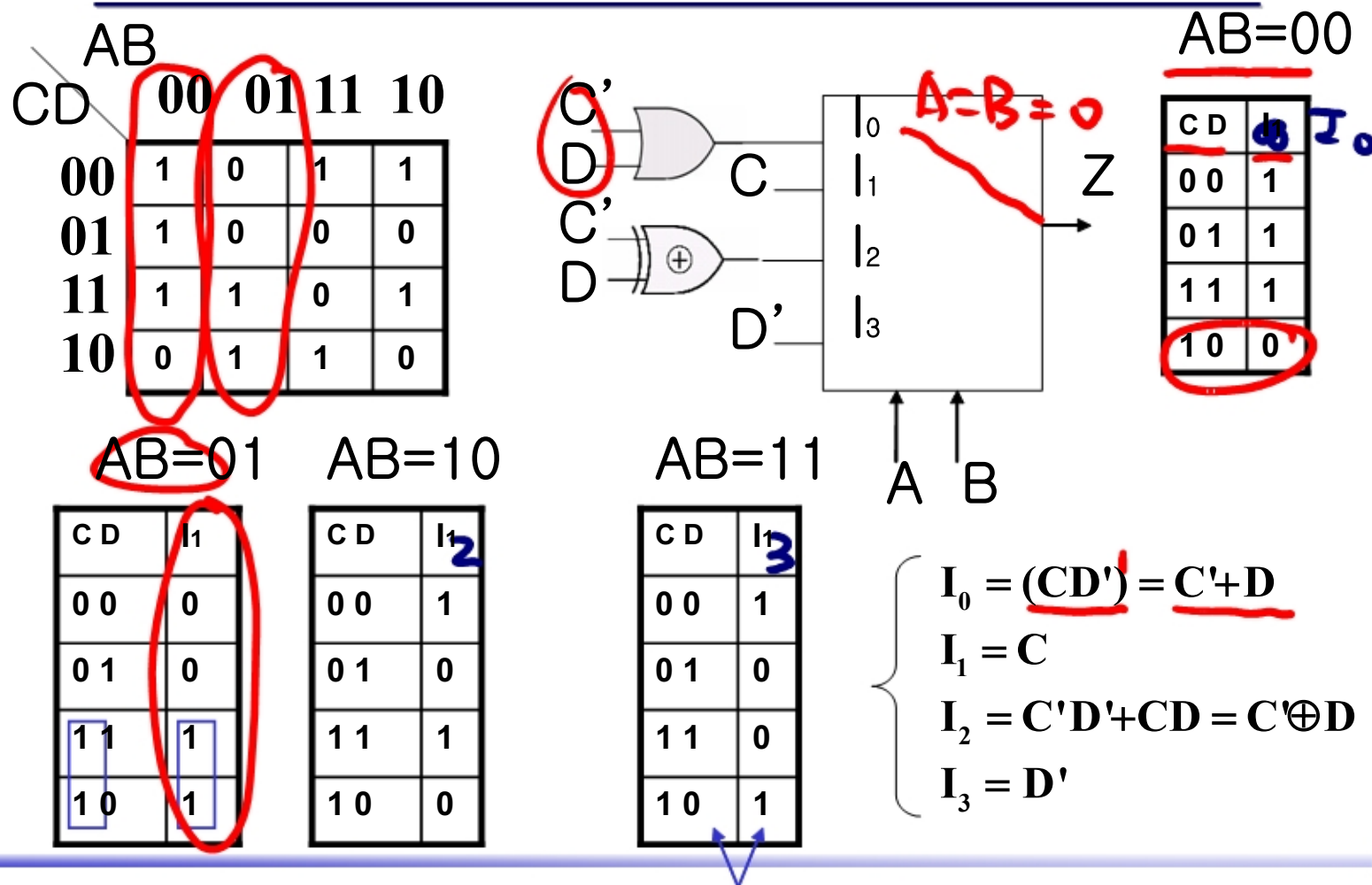


		AB			
		00	01	11	10
CD	00	1	0	1	1
	01	1	0	0	0
	11	1	1	0	1
	10	0	1	1	0

$$\begin{aligned} I_0 &= C' & I_1 &= 1 \\ I_2 &= C & I_3 &= C \\ I_4 &= C' & I_5 &= C \\ I_6 &= 1 & I_7 &= 0 \end{aligned}$$

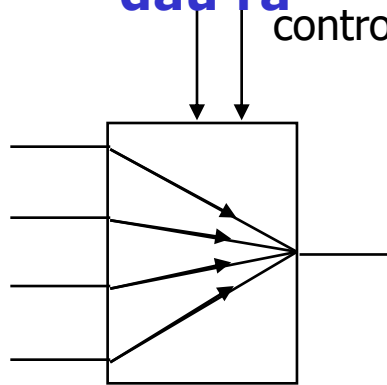


Ví dụ: Sử dụng bộ dồn kênh 4-to-1 thực hiện hàm $F(A,B,C)$

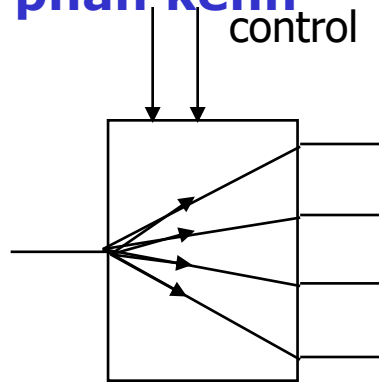


Tạo các kết nối

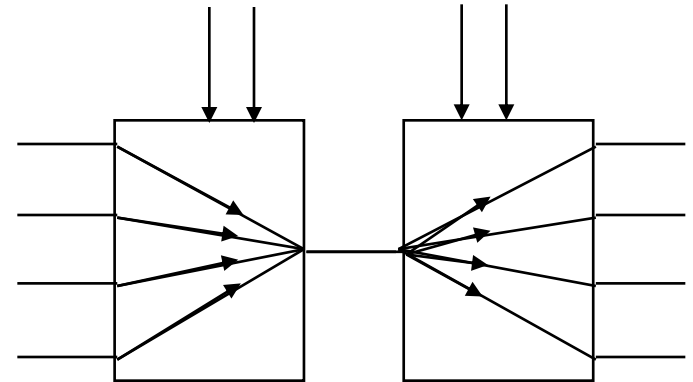
- **Kết nối trực tiếp điểm-điểm giữa các cửa**
 - ✓ **wires we've seen so far**
- **Điều khiển một trong các đầu vào đến một đầu ra duy nhất--- Bộ dồn kênh**
- **Điều khiển một đầu vào duy nhất đến một trong số các đầu ra --- Bộ phân kênh**



multiplexer



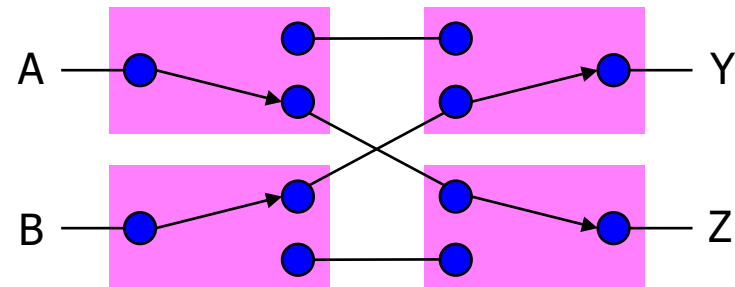
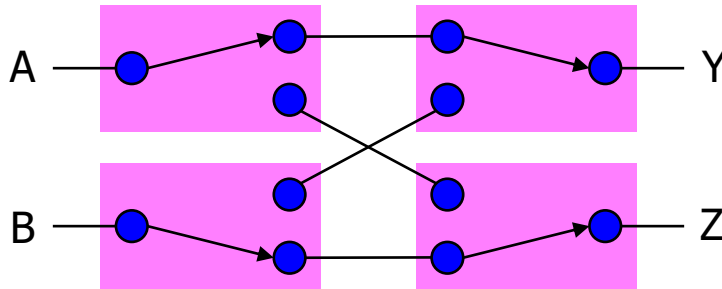
demultiplexer



4x4 switch

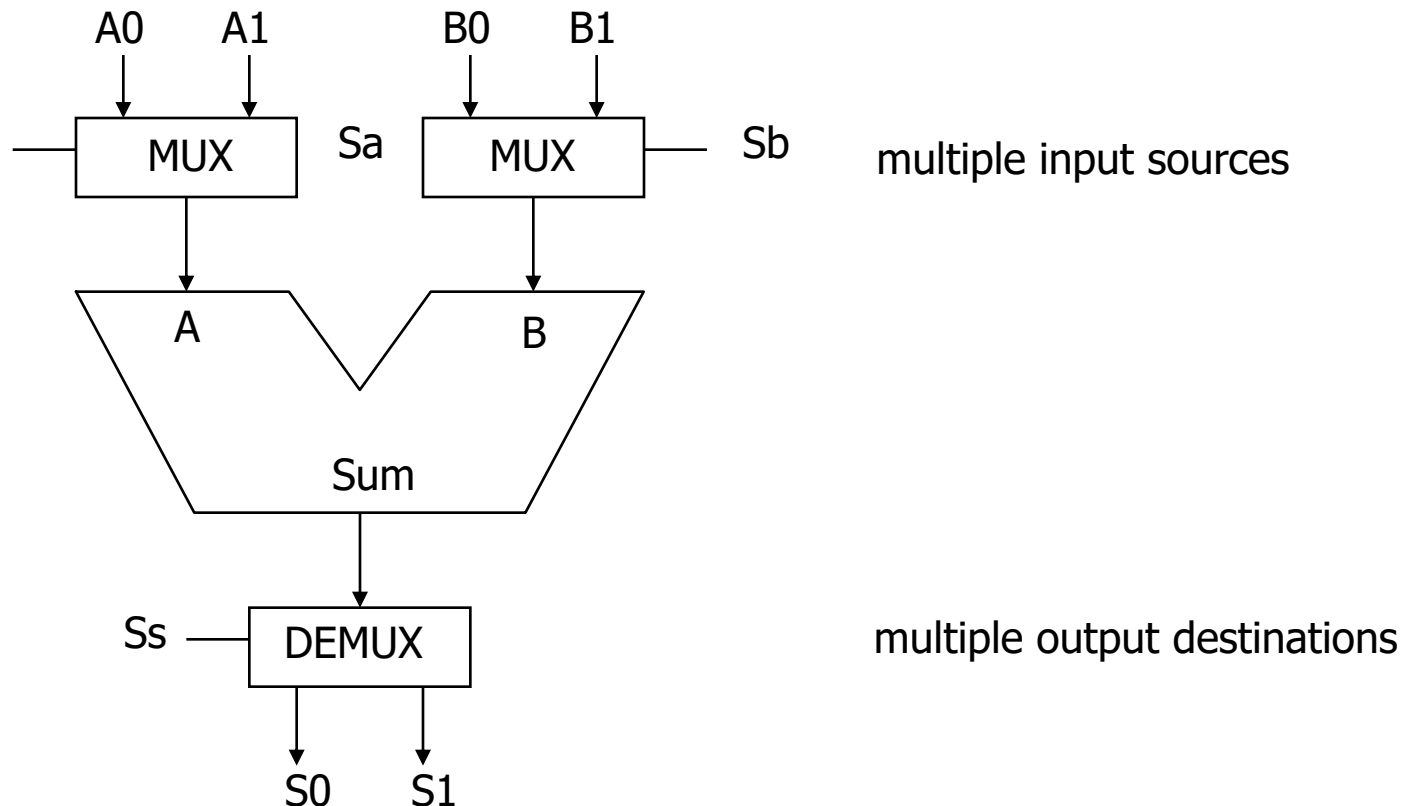
Bộ dồn kênh và bộ phân kênh

- **Chuyển đổi thực hiện bộ dồn kênh và bộ phân kênh**
 - ✓ Có thể được đưa vào các mạng chuyển đổi có kích thước tùy ý
 - ✓ Được sử dụng để thực hiện mỗi liên kết multiple-source/multiple-destination



Bộ dồn kênh và bộ phân kênh

- **Sử dụng bộ dồn kênh/bộ phân kênh trong các kết nối đa điểm**



Bộ dồn kênh/Bộ chọn

- **Bộ dồn kênh/bộ chọn: khái niệm chung**

- ✓ **2ⁿ đầu vào dữ liệu, n đầu vào điều khiển (gọi là chân chọn) và một đầu ra dữ liệu**
- ✓ **Dùng để kết nối 2ⁿ các điểm đến một điểm duy nhất**
- ✓ **Những giá trị ở các đầu vào điều khiển hình thành chỉ số nhị phân của đầu vào dữ liệu được kết nối với đầu ra**

$$Z = A' I_0 + A I_1$$

A	Z
0	I ₀
1	I ₁

I ₁	I ₀	A	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Dạng chức năng

Dạng logic

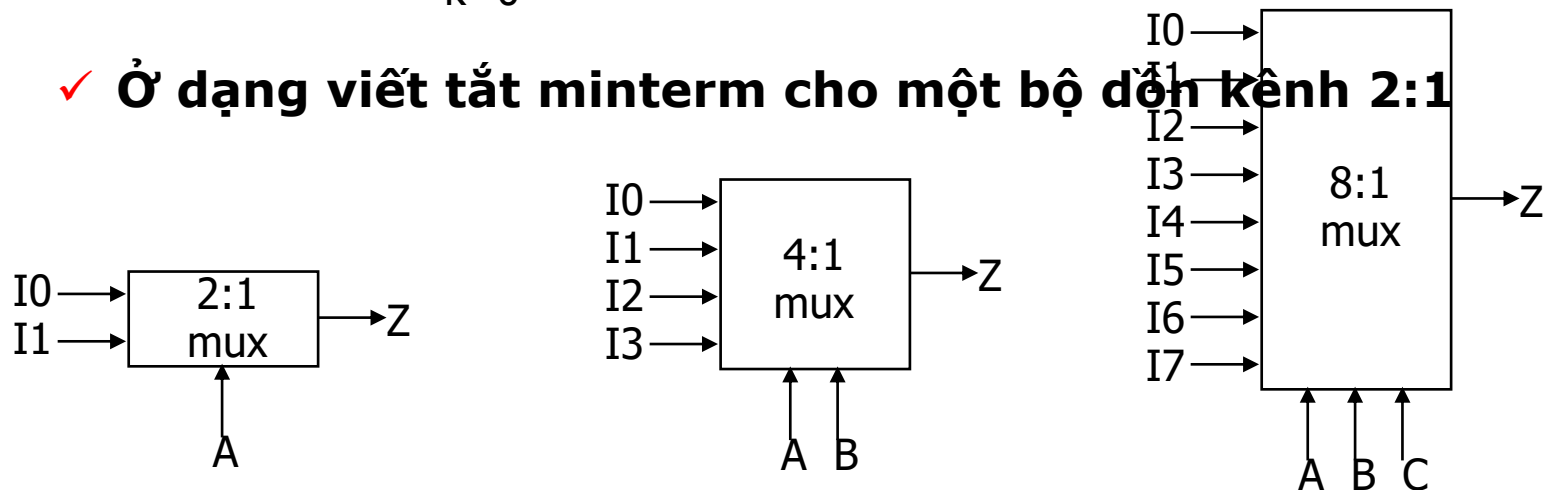
Hai dạng thay thế cho bảng
sự thật bộ dồn kênh 2:1

Bộ dồn kênh/Bộ chọn

- Bộ dồn kênh 2:1: $Z = A'I_0 + AI_1$
- Bộ dồn kênh 4:1: $Z = A'B'I_0 + A'BI_1 + AB'I_2 + ABI_3$
- Bộ dồn kênh 8:1: $Z = A'B'C'I_0 + A'B'CI_1 + A'BC'I_2 + A'BCI_3 +$
 $AB'C'I_4 + AB'CI_5 + ABC'I_6 + ABCI_7$

- Tổng quát: $Z = \sum_{k=0}^{2^n-1} (m_k I_k)$

✓ Ở dạng viết tắt minterm cho một bộ dồn kênh 2:1



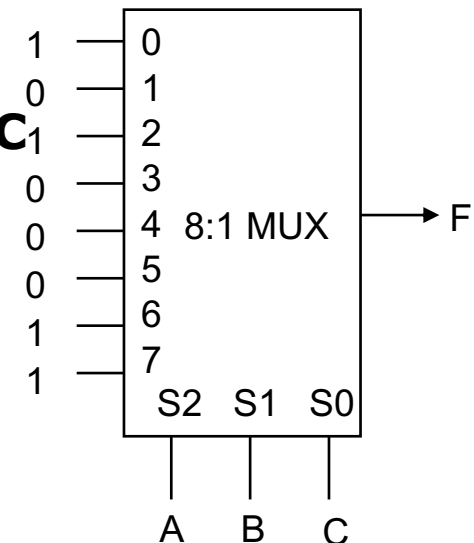
Bộ dồn kênh như một khối logic mục đích chung

- **Một bộ dồn kênh 2:1 có thể thực hiện bất kì hàm n biến nào**
 - ✓ Với các biến được sử dụng như các đầu vào điều khiển
 - ✓ Các dữ liệu đầu vào gán với 0 hoặc 1
 - ✓ Bảng tra

- **Ví dụ:**

- ✓ $F(A,B,C) = m_0 + m_2 + m_6 + m_7$
 $= A'B'C' + A'BC' + ABC' + ABC$

$$\begin{aligned} &= A'B'C'(1) + A'B'C(0) \\ &\quad + A'BC'(1) + A'BC(0) \\ &\quad + AB'C'(0) + AB'C(0) \\ &\quad + ABC'(1) + ABC(1) \end{aligned}$$



$$Z = A'B'C'I_0 + A'B'CI_1 + A'BC'I_2 + A'BCI_3 +$$

$$AB'C'I_4 + AB'CI_5 + ABC'I_6 + ABCI_7$$

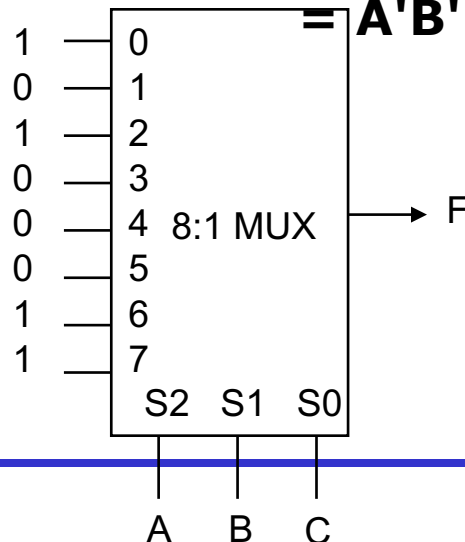
Bộ dồn kênh như một khối logic mục đích chung

- Một bộ dồn kênh $2^{n-1}:1$ có thể thực hiện bất kì hàm n biến nào

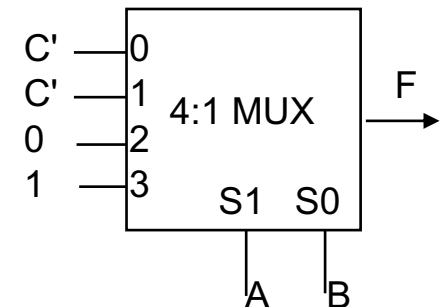
- ✓ Với $n-1$ biến được sử dụng như đầu vào điều khiển
- ✓ Các đầu vào dữ liệu gắn với biến cuối cùng hoặc bù của nó

Ví dụ:

- ✓
$$\begin{aligned} F(A,B,C) &= m_0 + m_2 + m_6 + m_7 \\ &= A'B'C' + A'BC' + ABC' + ABC \\ &= A'B'(C') + A'B(C') + AB'(0) + AB(1) \end{aligned}$$



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

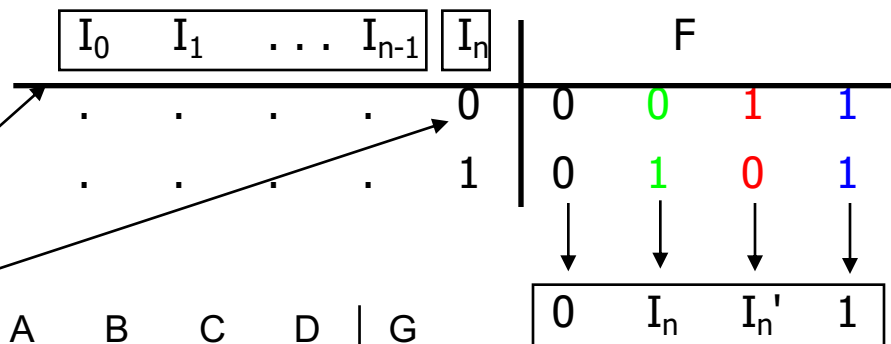


Bộ dồn kênh như một khối logic mục đích chung

- Tổng quát hóa**

n-1 mux control variables

single mux data variable

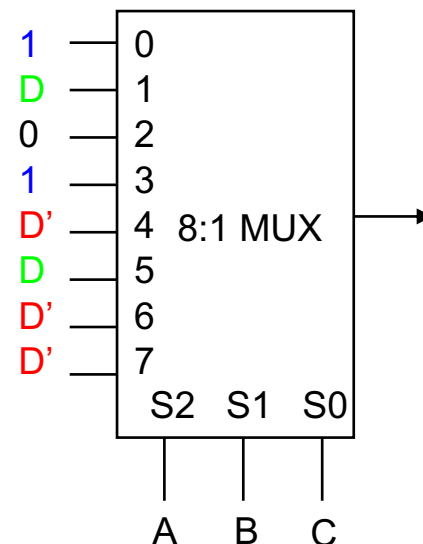


4 khả năng của giá trị hàm ứng với 2 hàng của bảng sự thật

- Ví dụ: hàm $G(A,B,C,D)$ có thể thực hiện bằng bộ dồn kênh 8:1**

Chọn A,B,C là biến điều khiển

A	B	C	D	G
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



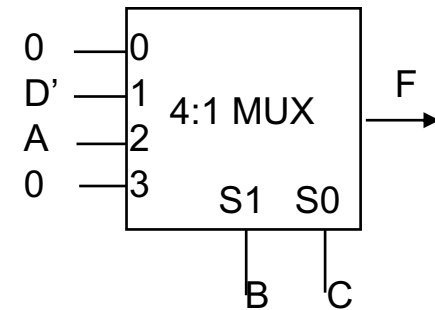
Hoạt động

- Thực hiện hàm $F = B'CD' + ABC'$ với một bộ dồn kênh 4:1 và sử dụng tối thiểu các cổng khác

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Annotations for the truth table:

- 0 when $B'C'$ (points to rows 1 and 2)
- D' when $B'C$ (points to rows 3 and 4)
- A when BC' (points to rows 9 and 10)
- 0 when BC (points to rows 11 and 12)



$$Z = B'C'(0) + B'C(D') + BC'(A) + BC(0)$$

Bộ phân kênh/Bộ giải mã

- **Bộ giải mã/Bộ phân kênh: khái niệm chung**

- ✓ **Một đầu vào dữ liệu duy nhất, n đầu vào điều khiển, 2^n đầu ra**
- ✓ **Các đầu vào điều khiển(gọi là“selects” (S)) đại diện cho chỉ số nhị phân của đầu ra mà đầu vào được kết nối với**
- ✓ **Đầu vào dữ liệu thường được gọi là “enable/cho phép” (G)**

1:2 Decoder:

$$O0 = G \bullet S'$$

$$O1 = G \bullet S$$

2:4 Decoder:

$$O0 = G \bullet S1' \bullet S0'$$

$$O1 = G \bullet S1' \bullet S0$$

$$O2 = G \bullet S1 \bullet S0'$$

$$O3 = G \bullet S1 \bullet S0$$

3:8 Decoder:

$$O0 = G \bullet S2' \bullet S1' \bullet S0'$$

$$O1 = G \bullet S2' \bullet S1' \bullet S0$$

$$O2 = G \bullet S2' \bullet S1 \bullet S0'$$

$$O3 = G \bullet S2' \bullet S1 \bullet S0$$

$$O4 = G \bullet S2 \bullet S1' \bullet S0'$$

$$O5 = G \bullet S2 \bullet S1' \bullet S0$$

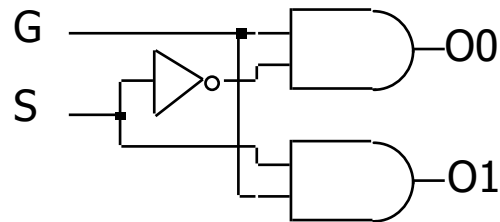
$$O6 = G \bullet S2 \bullet S1 \bullet S0'$$

$$O7 = G \bullet S2 \bullet S1 \bullet S0$$

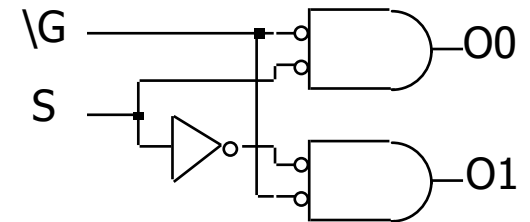
Thực hiện bằng cổng logic của bộ phân kênh

- **Bộ giải mã 1:2**

Cho phép tích cực
cao

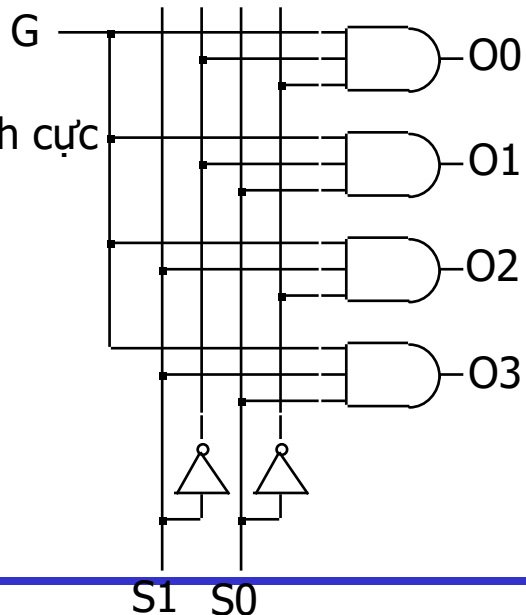


Cho phép tích cực
thấp

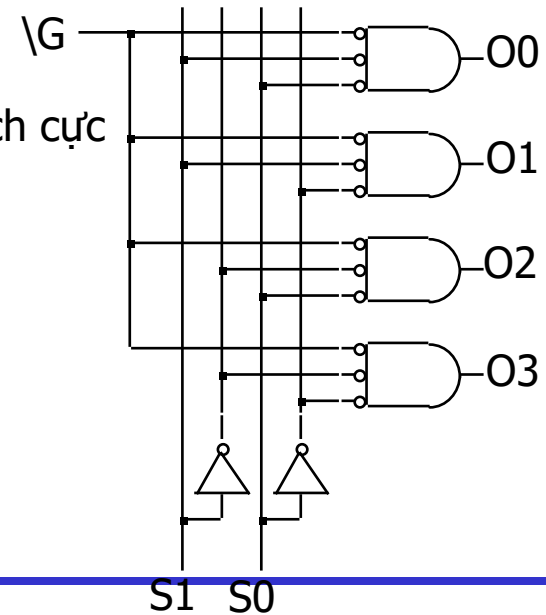


- **Bộ giải mã 2:4**

Cho phép tích cực
cao

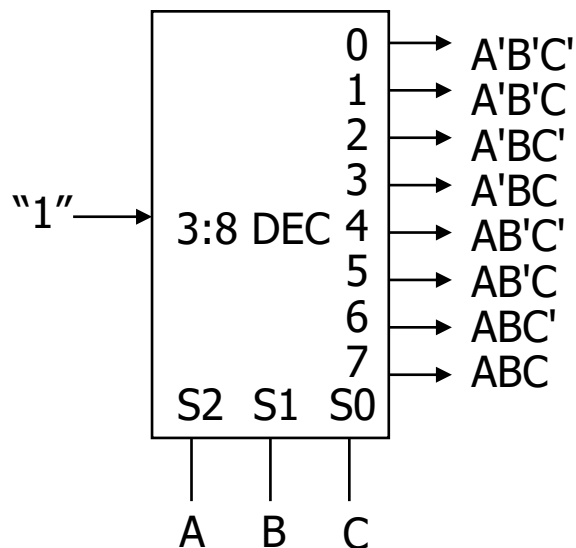


Cho phép tích cực
thấp



Bộ phân kênh như một khối logic mục đích chung

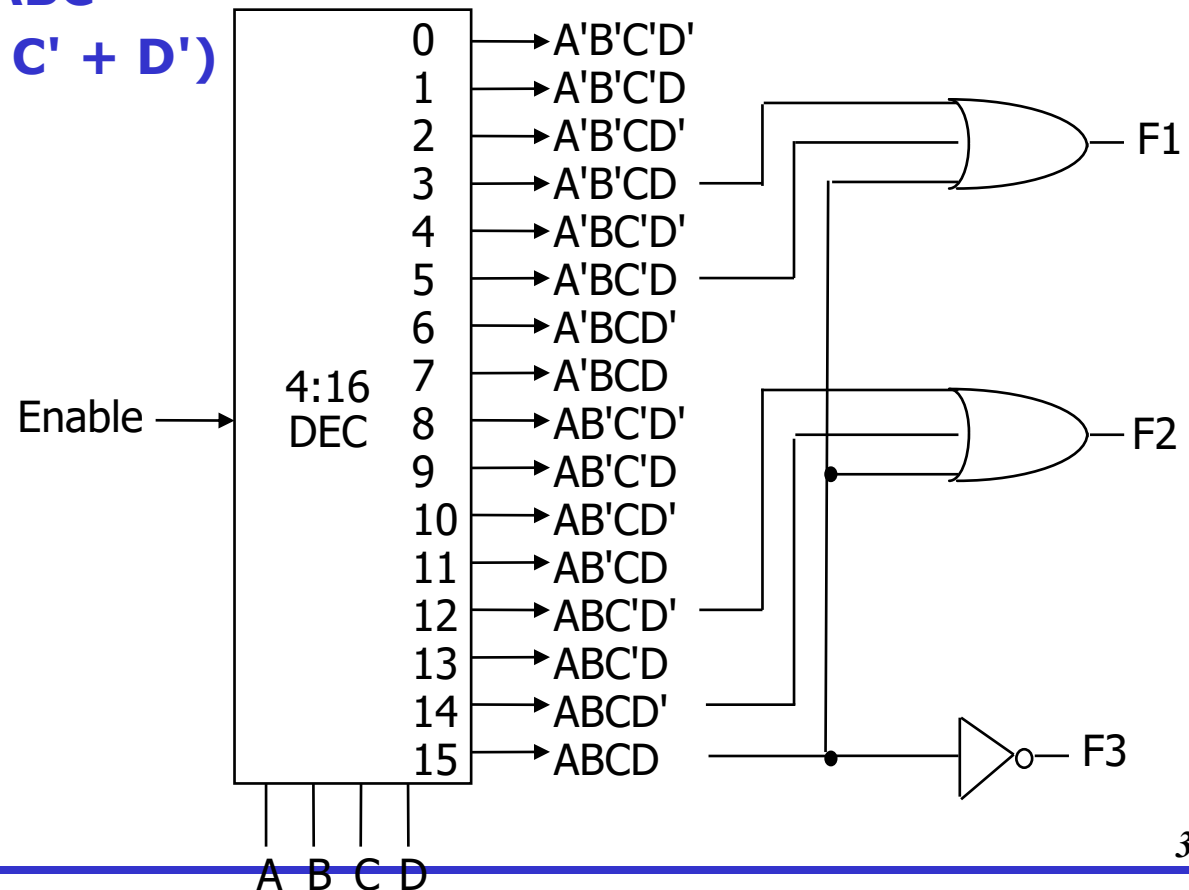
- **Một bộ giải mã $n:2^n$ có thể thực hiện bất kì hàm n biến nào**
 - ✓ với các biến được sử dụng như là đầu vào điều khiển
 - ✓ đầu vào cho phép gắn với 1
 - ✓ các minterms thích hợp được tổng hợp để thiết lập hàm



Bộ phân kênh tạo ra các minterm thích hợp dựa trên tín hiệu điều khiển (nó "giải mã" tín hiệu điều khiển)

Bộ phân kênh như một khối logic mục đích chung

- $F1 = A'BC'D + A'B'CD + ABCD$
- $F2 = ABC'D' + ABC$
- $F3 = (A' + B' + C' + D')$

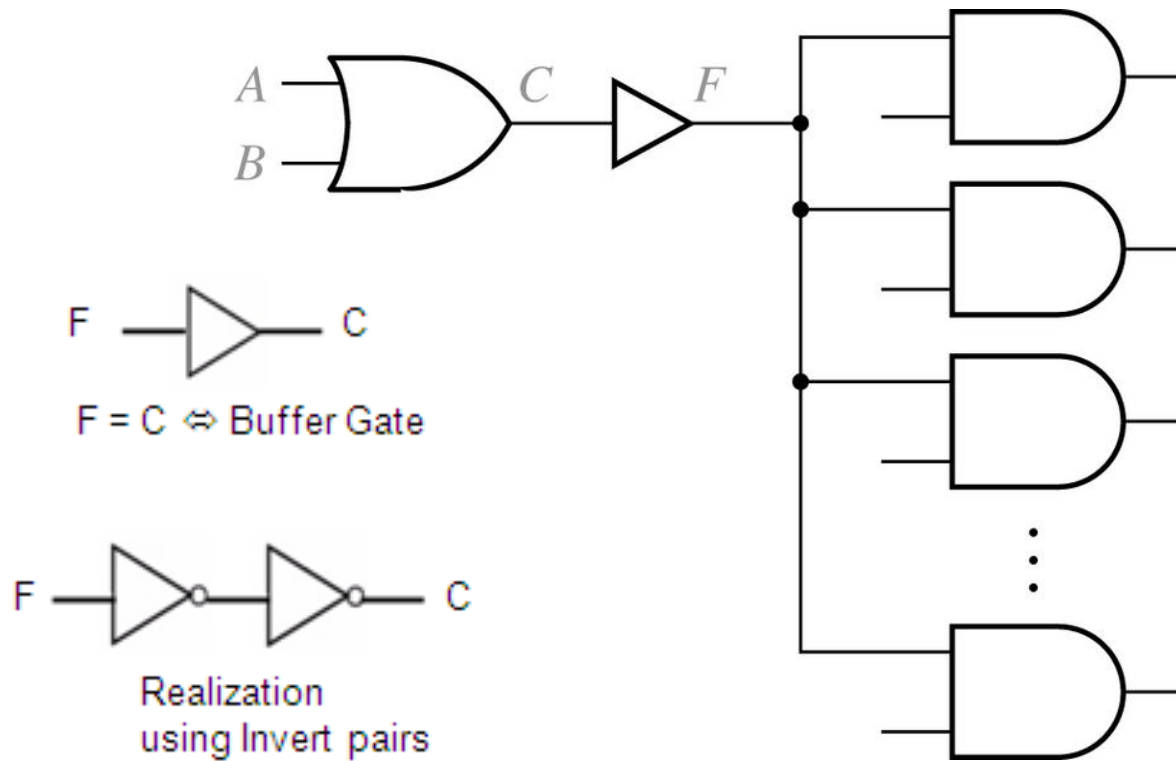


Cổng đệm ba trạng thái

Three-State Buffers

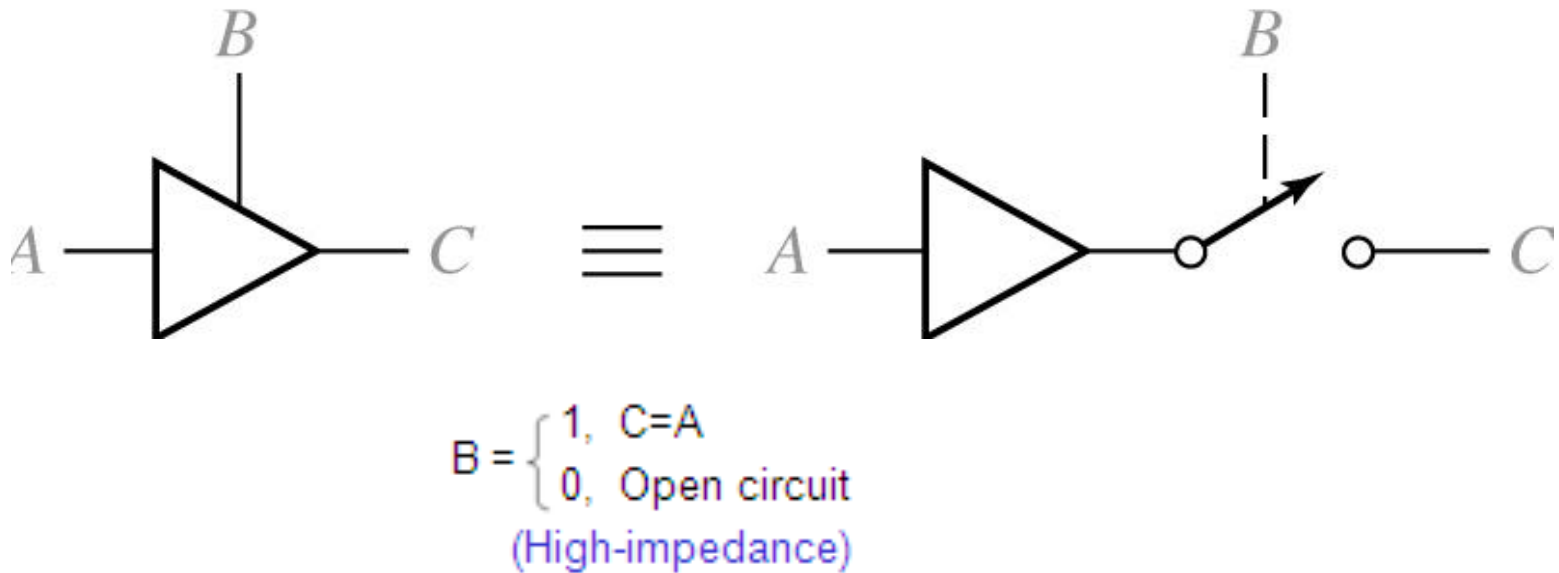
9.3 Cổng đếm ba trạng thái

Fig 9-6. Mạch cổng logic có đếm



9.3 Cổng đệm ba trạng thái

Fig 9-7. Cổng đệm ba trạng thái

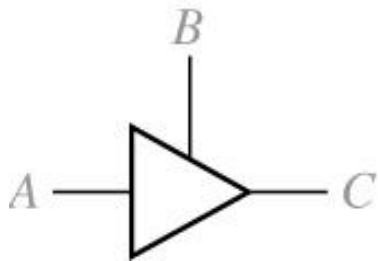


9.3 Cổng đệm ba trạng thái

Operations of
tri-state buffers

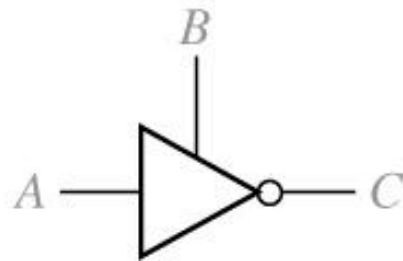
Fig 9-8. Bốn kiểu kí hiệu cổng đệm ba trạng thái

- Active High
- Active Low



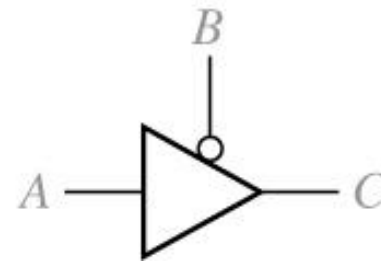
B	A	C
0	0	Z
0	1	Z
1	0	0
1	1	1

(a)



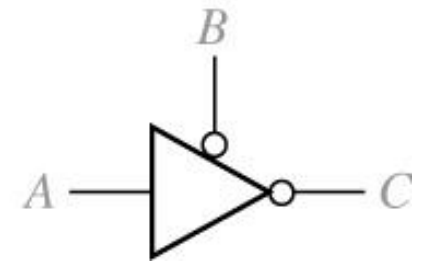
B	A	C
0	0	Z
0	1	Z
1	0	1
1	1	0

(b)



B	A	C
0	0	0
0	1	1
1	0	Z
1	1	Z

(c)

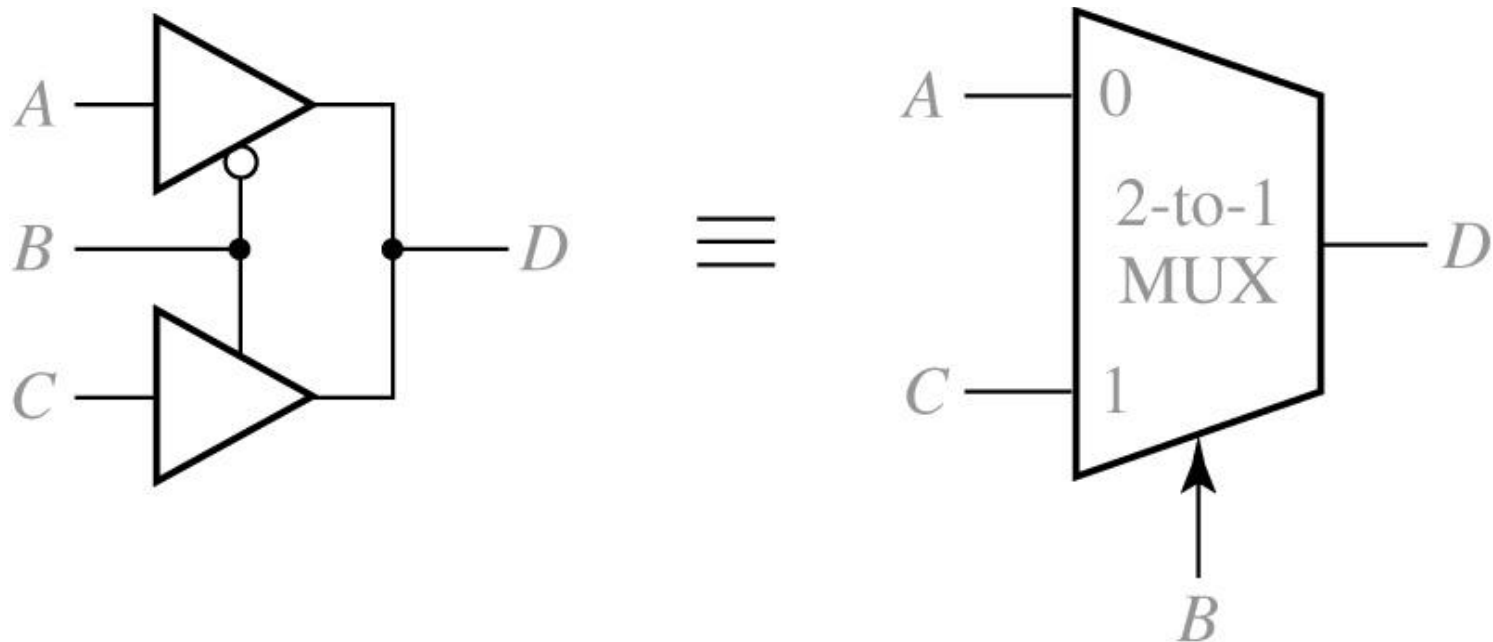


B	A	C
0	0	1
0	1	0
1	0	Z
1	1	Z

(d)

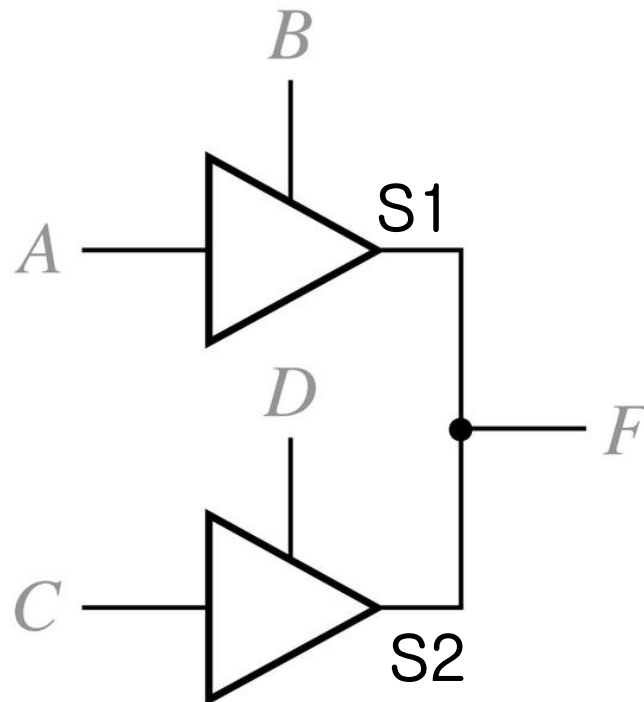
9.3 Cổng đếm ba trạng thái

Fig 9-9. Lựa chọn dữ liệu sử dụng cổng đếm ba trạng thái



9.3 Cổng đếm ba trạng thái

Fig 9-10. Mạch với hai cổng đếm ba trạng thái



$X = \text{Unknown}$

$S1$	X	$S2$	0	1	Z
X	X	X	X	X	X
0	X	0	0	X	0
1	X	X	1	1	1
Z	X	0	1	1	Z

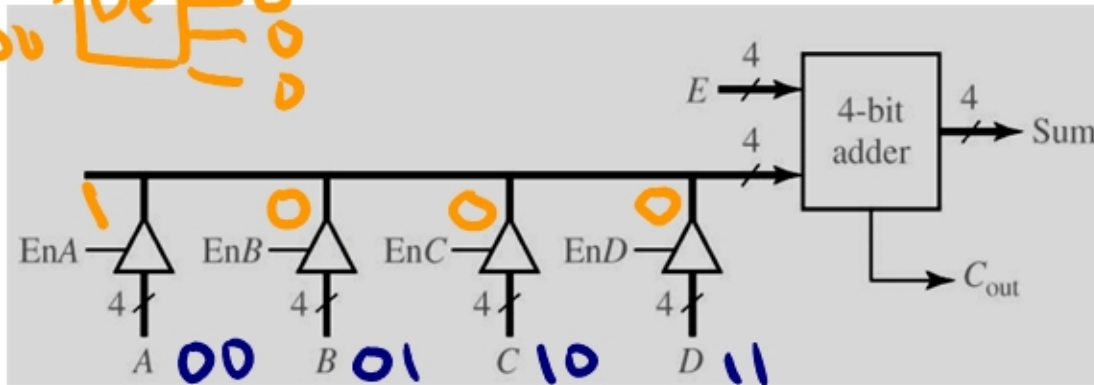
$S1$	$S2$	x	0	1	z
x	x	x	x	x	x
0	x	0	x	0	0
1	x	x	1	1	1
z	x	0	1	z	z

← Outputs of both Tri-state Buffers

(B, D are independent)



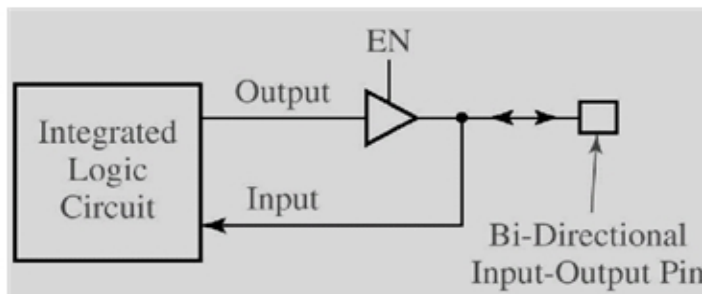
Ứng dụng của cổng đếm ba trạng thái



4-bit adder with 4 sources for one operand:

→ **Bus structure:**
Multiple I/O on a Bus for communication

{EnA, EnB, EnC, EnD} should be exclusive (Only 1 active)

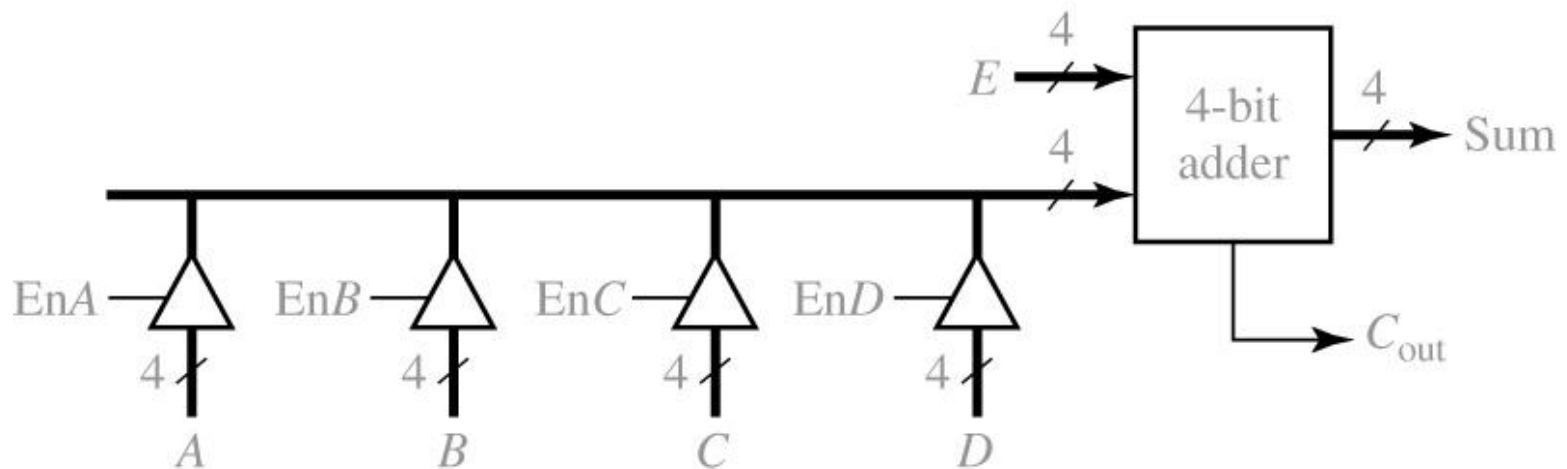


Bi-directional I/O pin

Bi-directional means that the same pin can be used as input pin and as an output pin, but not both at the same time

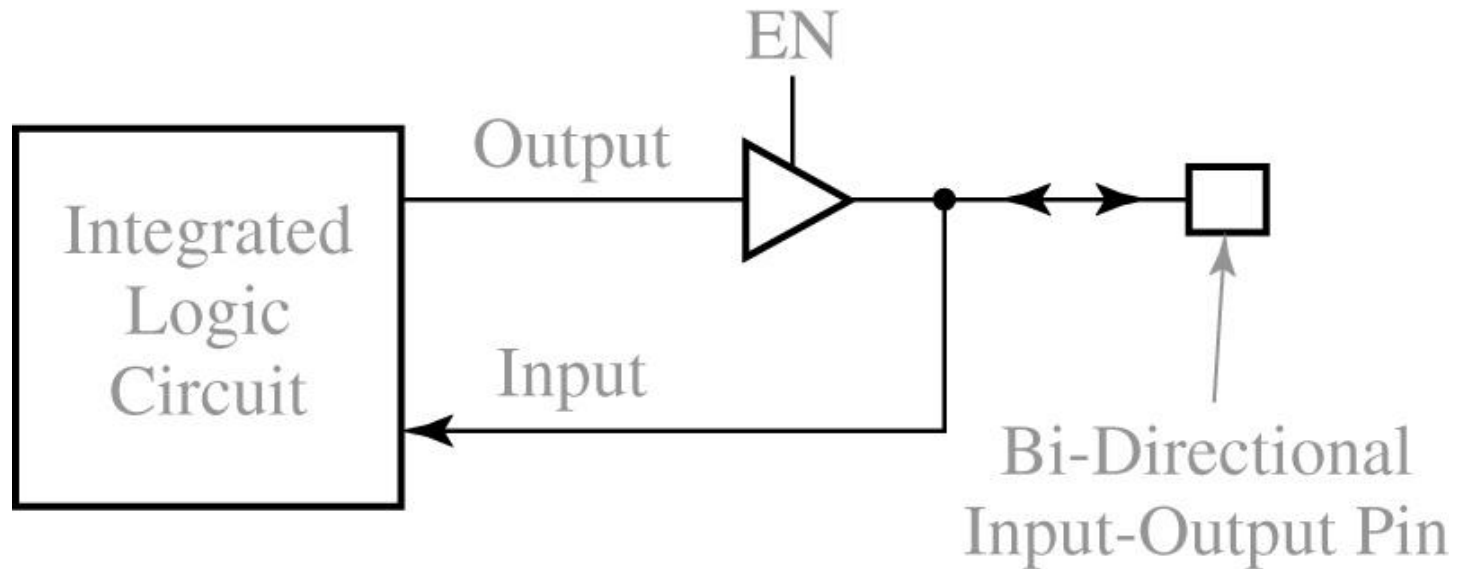
9.3 Cổng đệm ba trạng thái

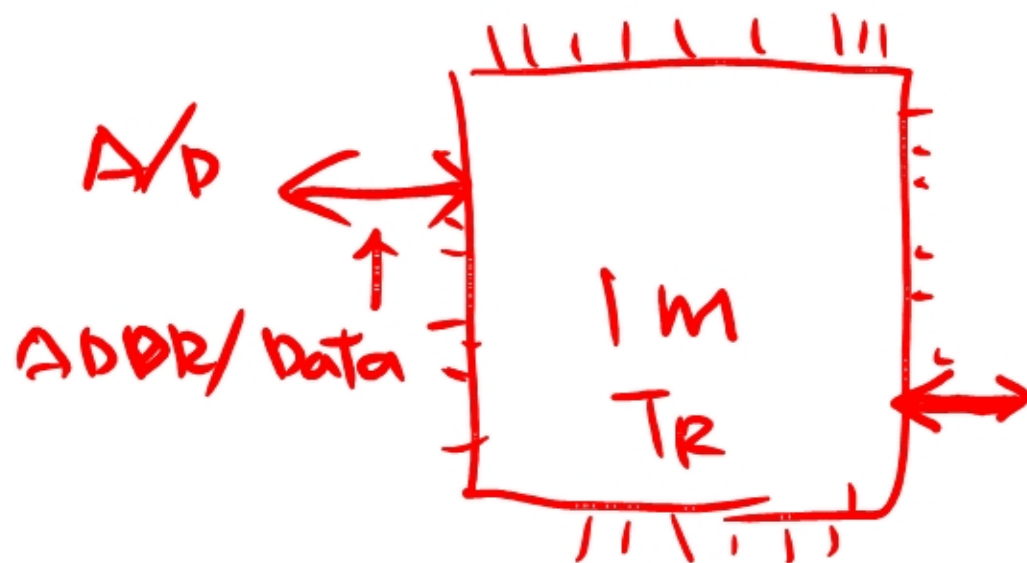
Fig 9-11. Bộ công 4bit với 4 nguồn cho một toán hạng



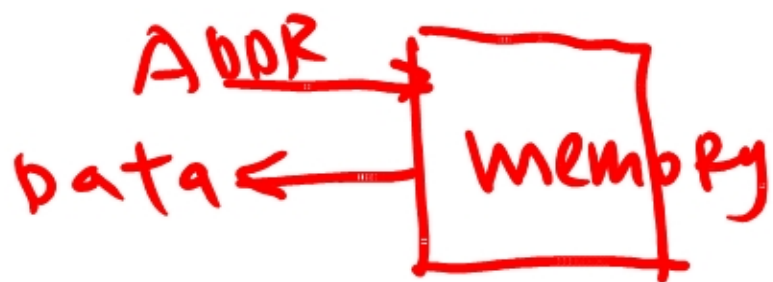
9.3 Cổng đếm ba trạng thái

Fig 9-12. Mạch tích hợp với chân I/O hai hướng



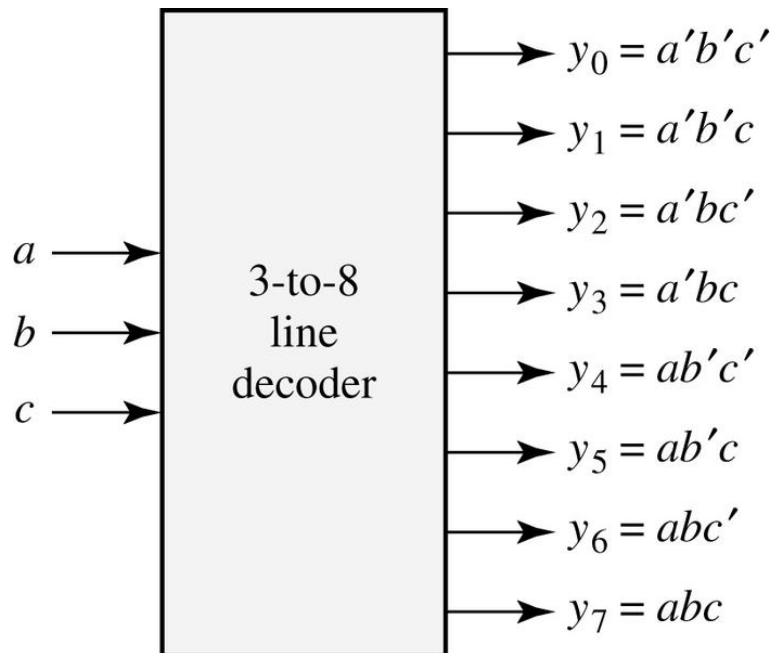


200-500
I/O



9.4 Mạch giải mã và mạch mã hóa

Fig 9-13. Vi mạch giải mã 3-ra-8



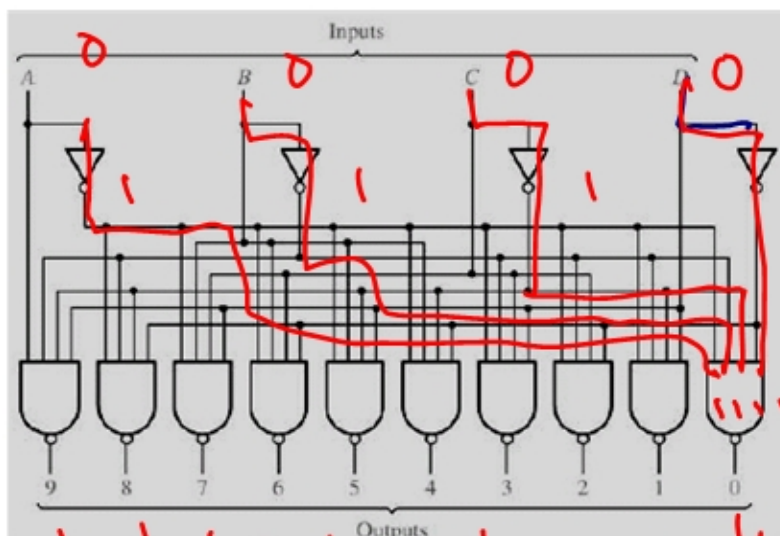
$a\ b\ c$	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	1	0
111	0	0	0	0	0	0	0	1

Master
Binary number
(a2a1a0)

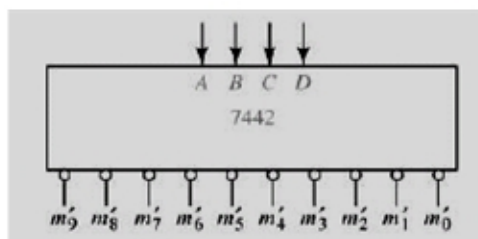
Decode 0
Decode 1
Decode 2
Decode 3
Decode 4
Decode 5
Decode 6
Decode 7



Thực hiện một bộ giải mã



(a) Logic Diagram



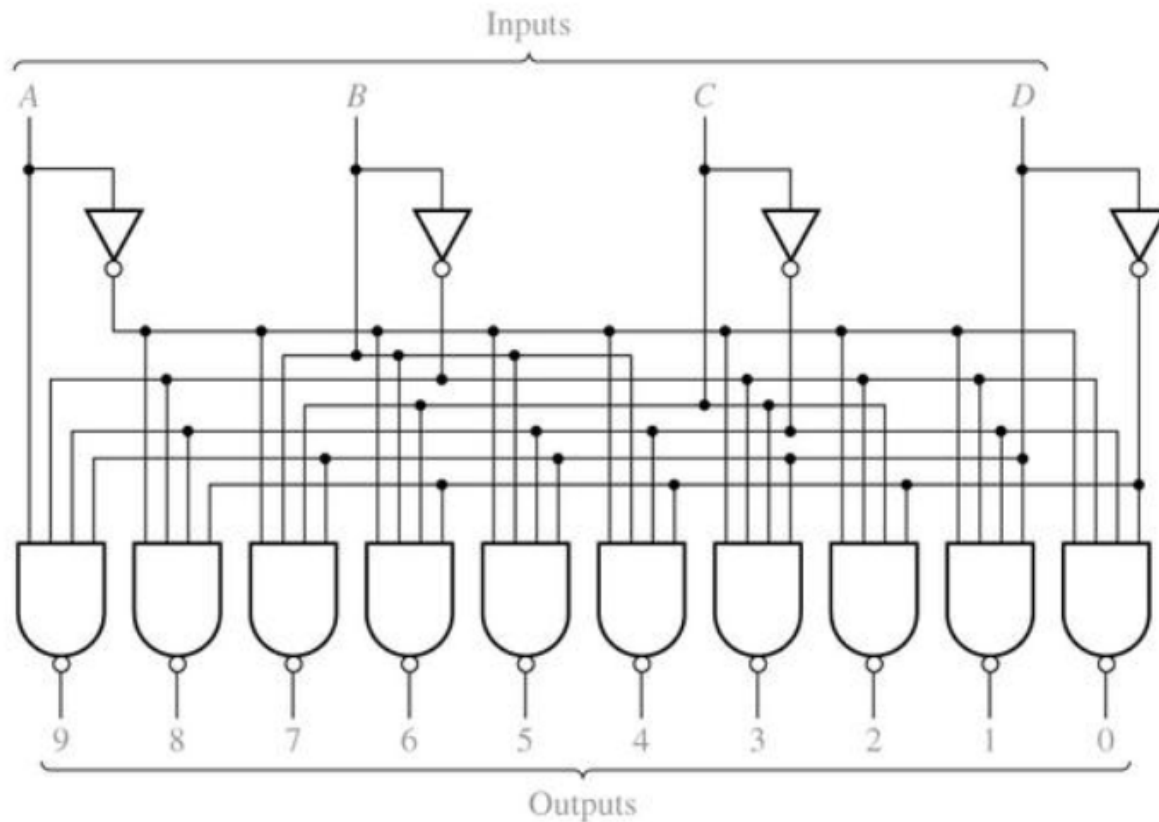
(b) Block Diagram (with active-low output)

BCD input	Decimal Output
A B C D	0 1 2 3 4 5 6 7 8 9
0 0 0 0	0 1 1 1 1 1 1 1 1 1
0 0 0 1	1 0 1 1 1 1 1 1 1 1
0 0 1 0	1 1 0 1 1 1 1 1 1 1
0 0 1 1	1 1 1 0 1 1 1 1 1 1
0 1 0 0	1 1 1 1 0 1 1 1 1 1
0 1 0 1	1 1 1 1 1 0 1 1 1 1
0 1 1 0	1 1 1 1 1 1 0 1 1 1
0 1 1 1	1 1 1 1 1 1 1 0 1 1
1 0 0 0	1 1 1 1 1 1 1 1 0 1
1 0 0 1	1 1 1 1 1 1 1 1 1 0
1 0 1 0	1 1 1 1 1 1 1 1 1 1
1 0 1 1	1 1 1 1 1 1 1 1 1 1
1 1 0 0	1 1 1 1 1 1 1 1 1 1
1 1 0 1	1 1 1 1 1 1 1 1 1 1
1 1 1 0	1 1 1 1 1 1 1 1 1 1
1 1 1 1	1 1 1 1 1 1 1 1 1 1

$a'b'c'd'$ (c) Truth table

9.4 Mạch giải mã và mạch mã hóa

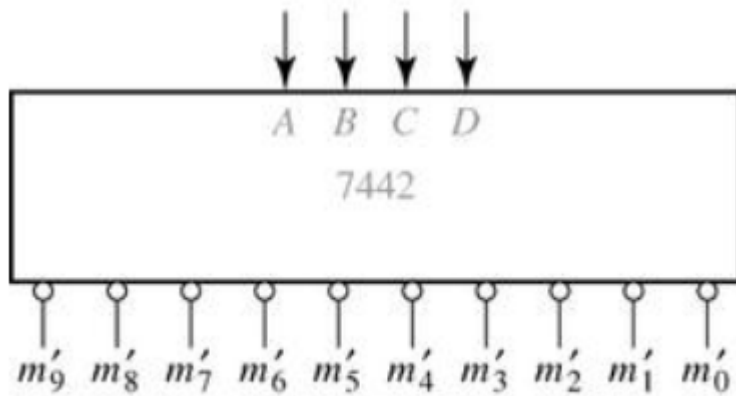
Fig 9-14. Vi mạch giải mã 4-ra-10(1)



(a) Logic diagram

9.4 Mạch giải mã và mạch mã hóa

Fig 9-14. Vi mạch giải mã 4-ra-10(2)



(b) Block diagram

BCD Input				Decimal Output									
A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	11
1	1	1	1	1	1	1	1	1	1	1	1	1	

(c) Truth Table

9.4 Mạch giải mã và mạch mã hóa

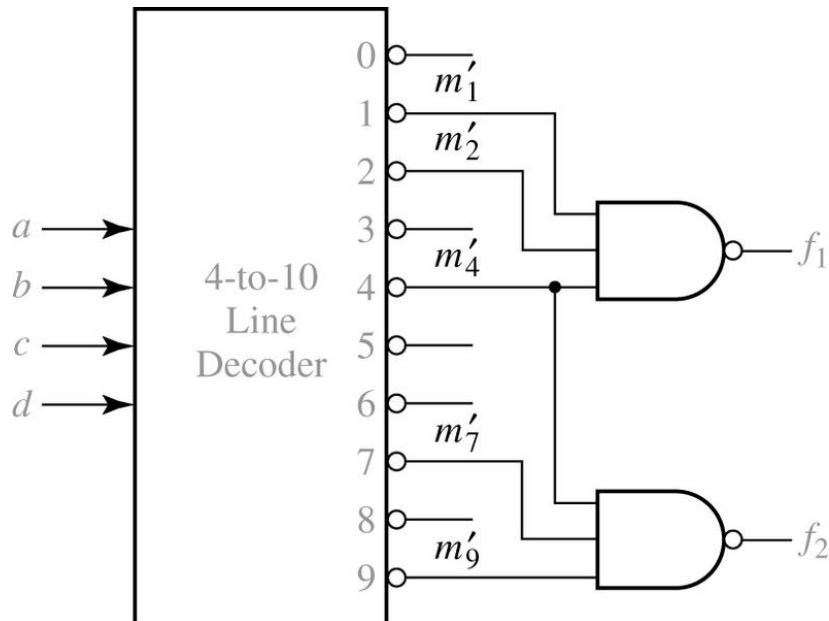
Fig 9-15. Thực hiện một mạch nhiều đầu ra sử dụng một bộ giải mã

Một bộ giải mã n -to- 2^n tạo ra 2^n minterms

$$y_i = m_i, \quad i = 0 \text{ to } 2^n - 1 \quad (\text{noninverted outputs})$$

or

$$y_i = m_i = M_i, \quad i = 0 \text{ to } 2^n - 1 \quad (\text{inverted outputs})$$



VD:

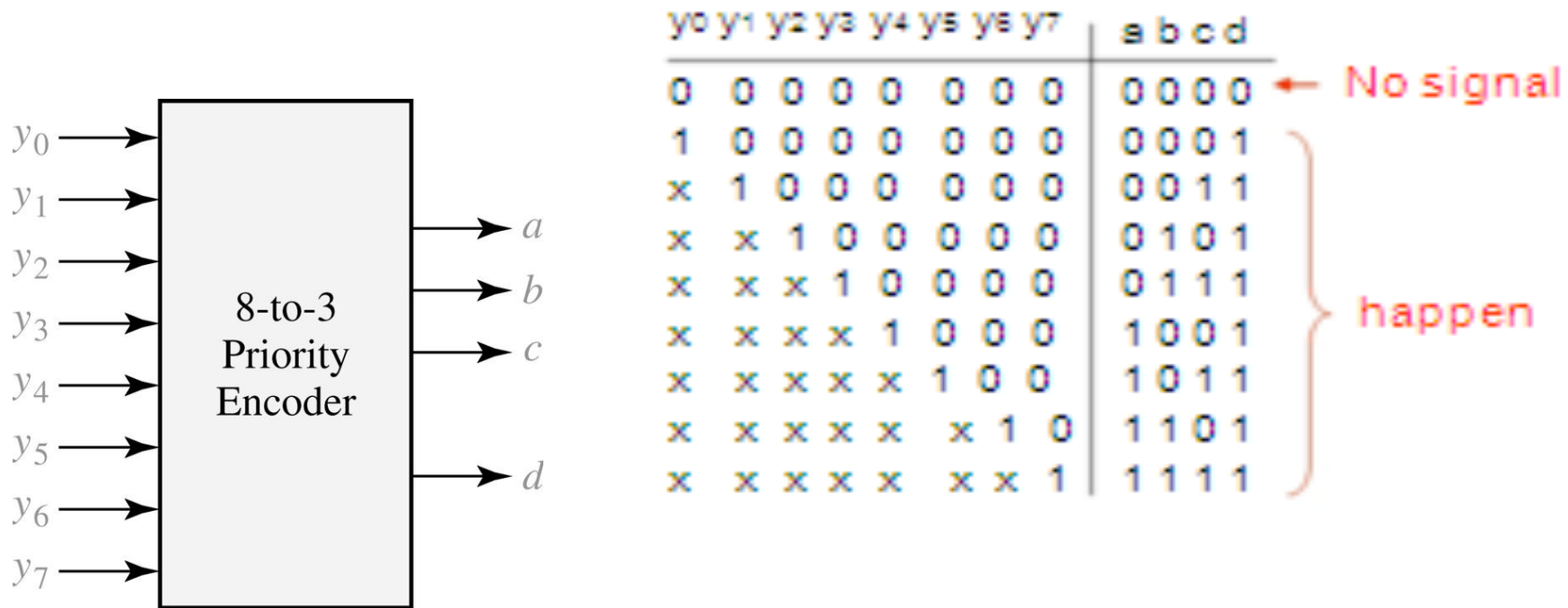
$$\begin{aligned} f_1(a, b, c, d) &= m_1 + m_2 + m_4 \\ &= (m_1' m_2' m_4')' \quad (\text{NAND}) \end{aligned}$$

$$\begin{aligned} f_2(a, b, c, d) &= m_4 + m_7 + m_9 \quad (\text{NAND}) \\ &= (m_4' m_7' m_9')' \end{aligned}$$

$M > 10$ là không được phép khi dùng MSI 7442 (BCD input decoder)

9.4 Mạch giải mã và mạch mã hóa

Fig 9-16. Mạch mã hóa ưu tiên 8-ra-3



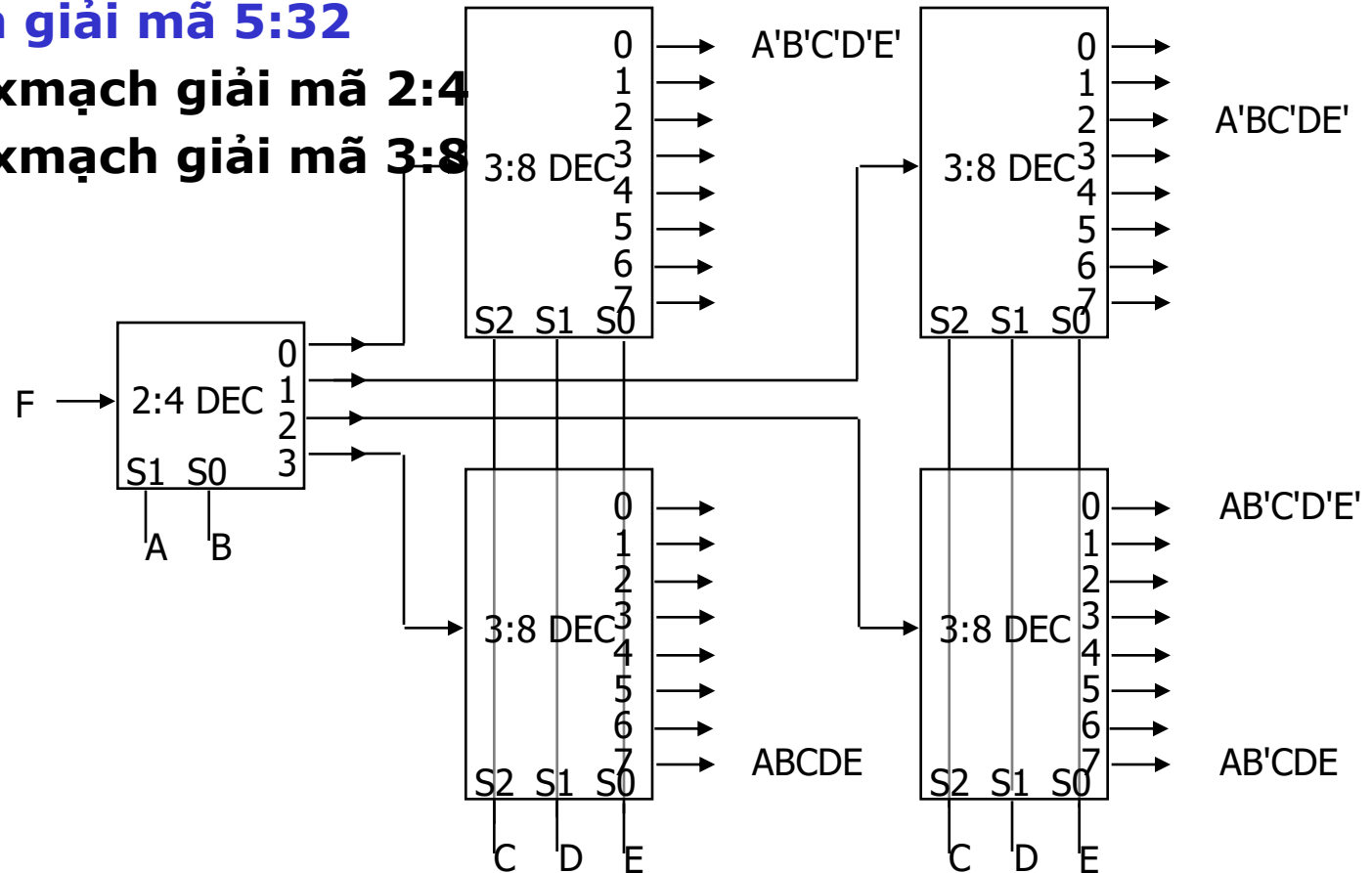
Phát hiện sự kiện có ưu tiên!
 $\Rightarrow y_7 > y_6 > \dots > y_0$

Mạch giải mã tầng

- **Mạch giải mã 5:32**

- ✓ **1x mạch giải mã 2:4**

- ✓ **4x mạch giải mã 3:8**



Read-Only Memories (ROM)

Bộ nhớ chỉ đọc ROM

9.5 Bộ nhớ ROM

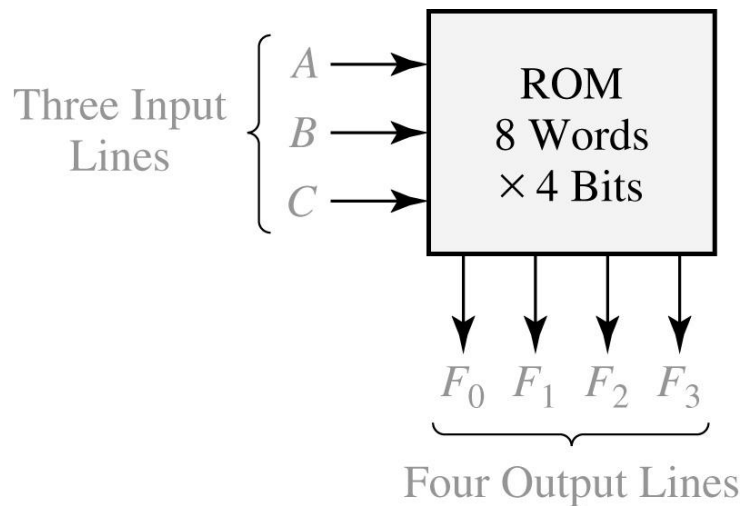
Fig 9-17. An 8-Word x 4-Bit ROM

Mỗi word là 4-bit, tổng cộng có 8 words trong ROM đầu vào (0~7 địa chỉ)

❖ Đầu vào $(ABC)=2^3$ giá trị

❖ Output:

$(F_0 F_1 F_2 F_3)=(\text{word})$



(a) Block diagram

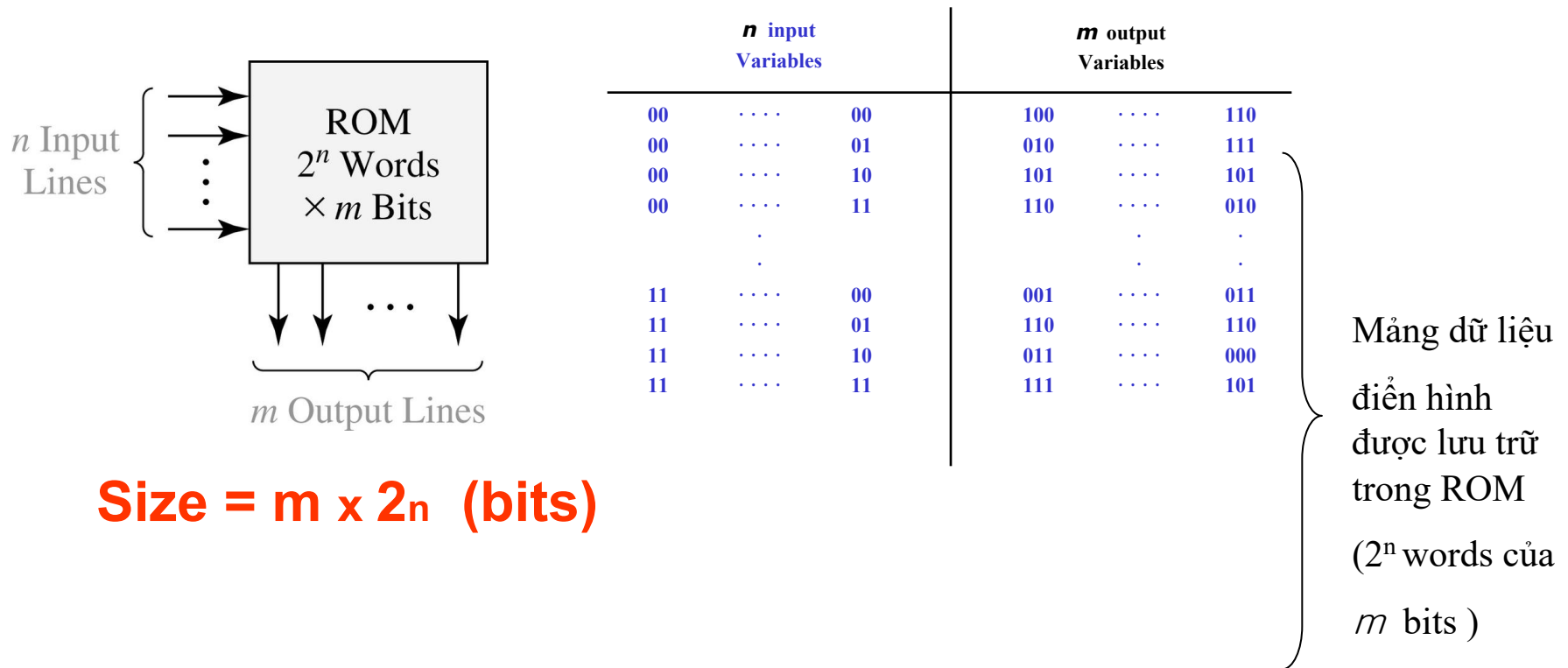
ABC	F_0	F_1	F_2	F_3
000	1	0	1	0
001	1	0	1	0
010	0	1	1	1
011	0	1	0	1
100	1	1	0	0
101	0	0	0	1
110	1	1	1	1
111	0	1	0	1

Typical data stored in ROM (2ⁿ words of 4 bits each)

(b) Truth table for ROM

9.5 Bộ nhớ ROM

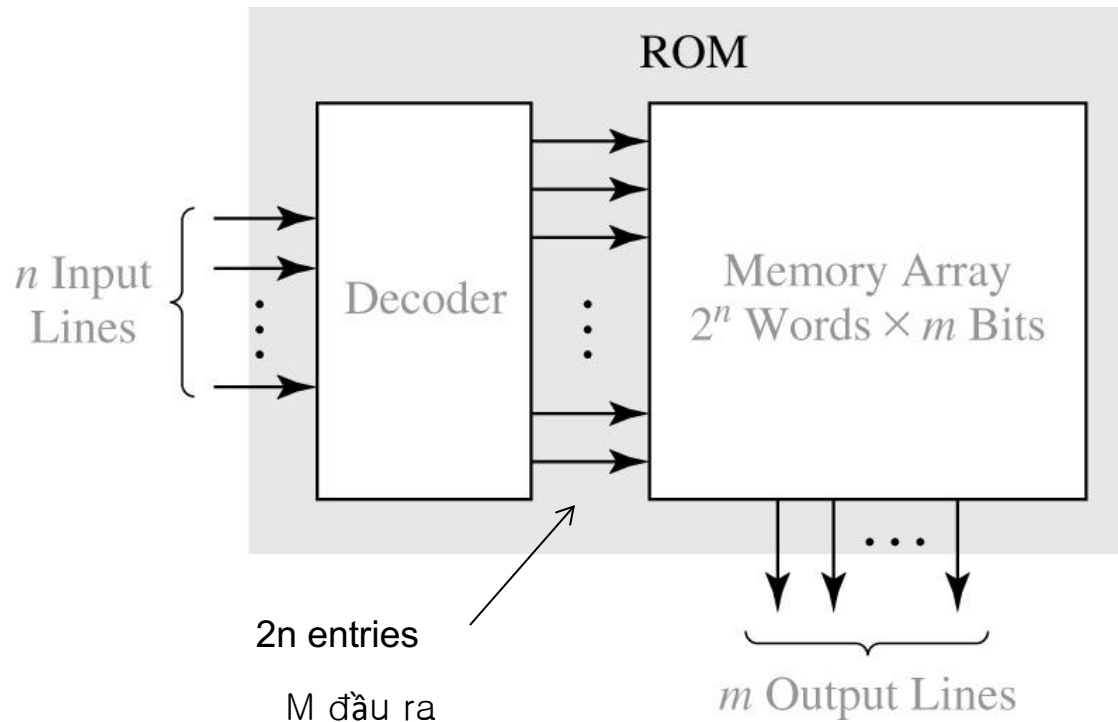
Fig 9-18. Bộ nhớ ROM với n đầu vào và m đầu ra



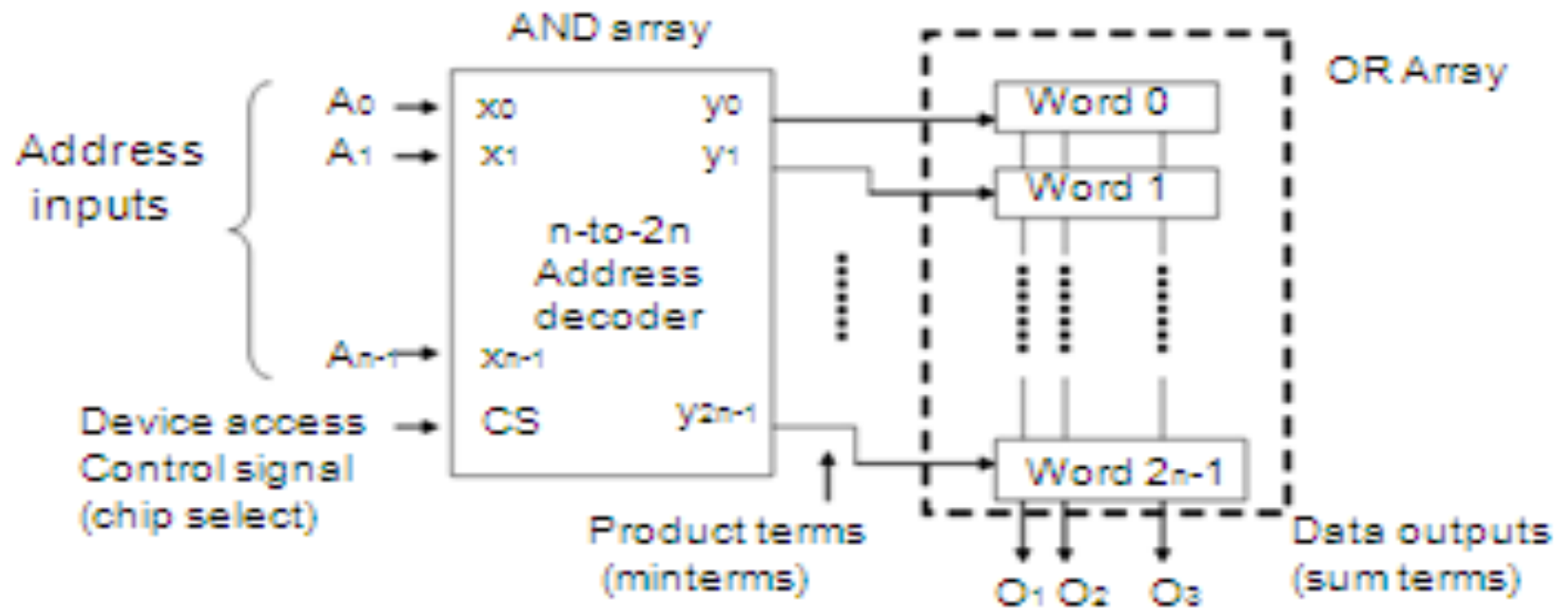
Size = $m \times 2^n$ (bits)

9.5 Bộ nhớ ROM

Fig 9-19. Cấu trúc cơ bản của bộ nhớ ROM



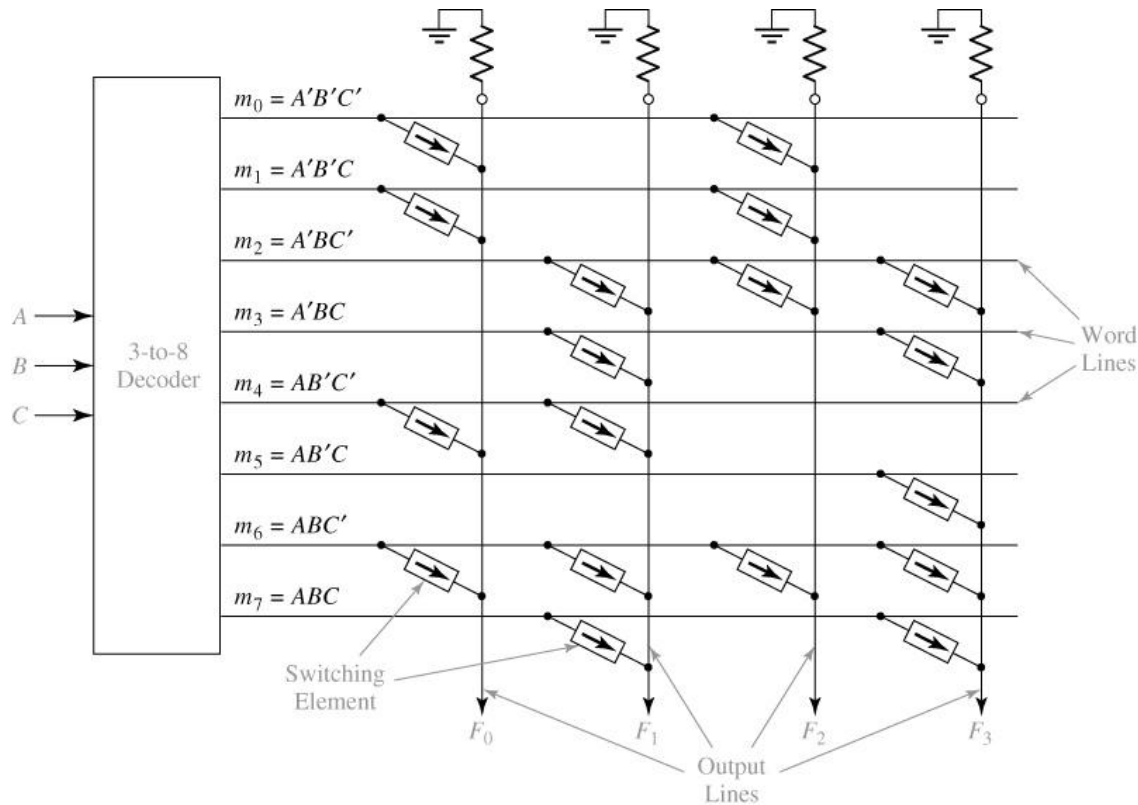
ROM



- Types: {
- Mask-programmed ROM
 - Erasable Programmable ROM (EPROM)
 - Electrically Erasable Programmable ROM (EEPROM)

9.5 Bộ nhớ ROM

Fig 9-20. An 8-Word x 4-Bit ROM



$$F_0 = \sum m(0,1,4,6) = A' B' + AC'$$

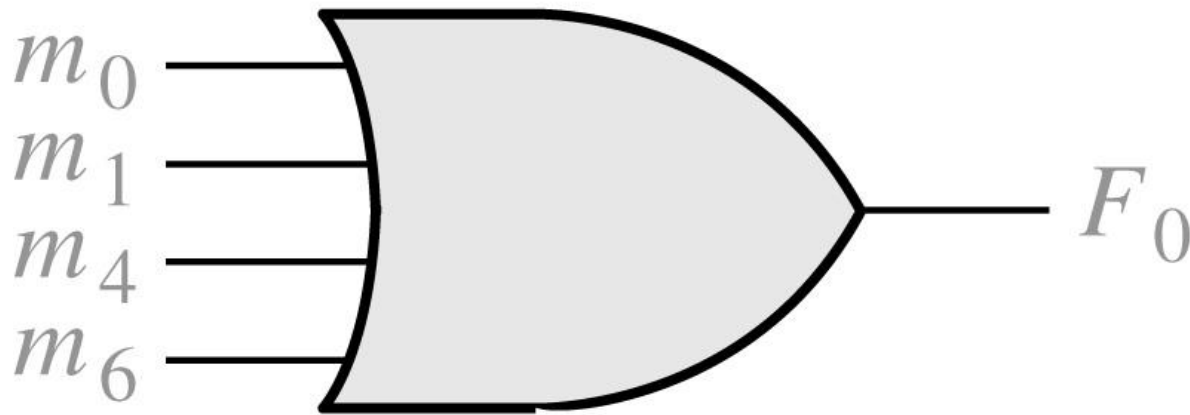
$$F_1 = \sum m(2,3,4,6,7) = B + AC'$$

$$F_2 = \sum m(0,1,2,6) = A' B' + BC'$$

$$F_3 = \sum m(2,3,5,6,7) = AC + B$$

9.5 Bộ nhớ ROM

Fig 9-21. Tương đương cổng OR cho F_0

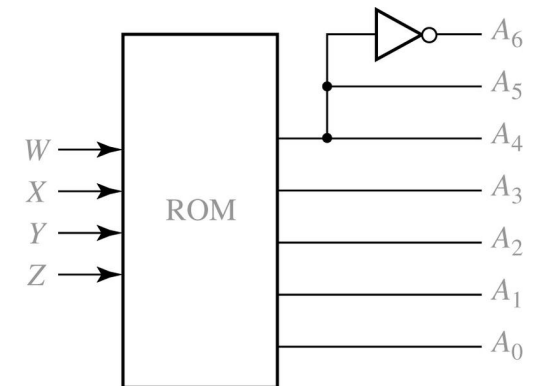


$$F_0 = \sum m(0,1,4,6) = A'B' + AC'$$

9.5 Bộ nhớ ROM

Fig 9-22. Mã chuyển đổi hệ thập lục phân sang ASCII

Input				Hex Digit	ASCII Code for Hex Digit						
W	X	Y	Z		A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	1	1	0	0	0	1
0	0	1	0	2	0	1	1	0	0	1	0
0	0	1	1	3	0	1	1	0	0	1	1
0	1	0	0	4	0	1	1	0	1	0	0
0	1	0	1	5	0	1	1	0	1	0	1
0	1	1	0	6	0	1	1	0	1	1	0
0	1	1	1	7	0	1	1	0	1	1	1
1	0	0	0	8	0	1	1	1	0	0	0
1	0	0	1	9	0	1	1	1	0	0	1
1	0	1	0	A	1	0	0	0	0	0	1
1	0	1	1	B	1	0	0	0	0	1	0
1	1	0	0	C	1	0	0	0	0	1	1
1	1	0	1	D	1	0	0	0	1	0	0
1	1	1	0	E	1	0	0	0	1	0	1
1	1	1	1	F	1	0	0	0	1	1	0



9.5 Bộ nhớ ROM

Fig 9-23. ROM thực hiện mã chuyển đổi

