

CHƯƠNG 2: MẠCH LOGIC TỔ HỢP

MỤC TIÊU: Sinh viên sau khi học xong có thể thu được các kiến thức

- Hiểu phương pháp xây dựng một số dạng mạch logic tổ hợp cơ bản
- Biết được phương pháp thiết kế 1 số mạch logic tổ hợp có tính ứng dụng

TÊN BÀI GIẢNG

MỘT SỐ MẠCH LOGIC TỔ HỢP CƠ BẢN

I. Quy trình thực hiện

Khi làm một bài toán thiết kế logic tổ hợp ta cần thực hiện đúng các bước sau đây

Bước 1: Dựa vào yêu cầu của bài toán đặt ra, chúng ta đặt các biến cho ngõ vào và các hàm của ngõ ra tương ứng

Bước 2: Thiết lập bảng sự thật cho ngõ ra và ngõ vào theo yêu cầu của bài toán

Bước 3: Từ bảng sự thật viết ra biểu thức mô tả sự liên hệ logic giữa ngõ ra và các ngõ vào

Có hai cách viết biểu thức logic cho ngõ ra, hoặc cho trường hợp logic ra bằng 1, hoặc cho trường hợp logic bằng 0 (*hai trường hợp này là tương đương nhau*)

Cách viết biểu thức thường ở dạng tổng-các-tích và tích-các-tổng.

Bước 4: Áp dụng các định lý của đại số boole để rút gọn biểu thức logic ngõ ra. Sau đó chuyển sang dạng logic khác để thuận lợi hơn cho việc thực hiện mạch logic

Bước 5: Từ biểu thức logic rút gọn được ta chuyển sang mạch logic tương ứng

II. Một số mạch logic tổ hợp ứng dụng cơ bản

II.1. Mạch chuyển mã

Mạch chuyển mã là một bộ giải mã, thực hiện chức năng chuyển từ 1 loại mã thành một loại mã khác tương ứng



Thiết kế

Ví dụ: Thiết kế bộ chuyển đổi từ mã nhị phân sang mã GRAY (4 bit)

Mạch thiết kế sẽ có 4 tín hiệu vào là 4 bit mã nhị phân và 4 hàm đầu ra là 4 bit của mã Gray

Gọi 4 bit của mã nhị phân là $X_1; X_2; X_3; X_4$;

Gọi 4 bit của mã Gray là $Y_1; Y_2; Y_3; Y_4$

Bảng chân lý

	Mã nhị phân				Mã Gray			
TT	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1

Đưa vào bảng Karnaugh

Y₁ X_3X_4

X_1X_2	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

Y₂ X_3X_4

X_1X_2	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

Y₃ X_3X_4

X_1X_2	00	01	11	10
00			1	1
01	1	1		
11	1	1		
10			1	1

Y₄ X_3X_4

X_1X_2	00	01	11	10
00		1		1
01		1		1
11		1		1
10		1		1

Tối thiểu hóa và đưa ra hàm

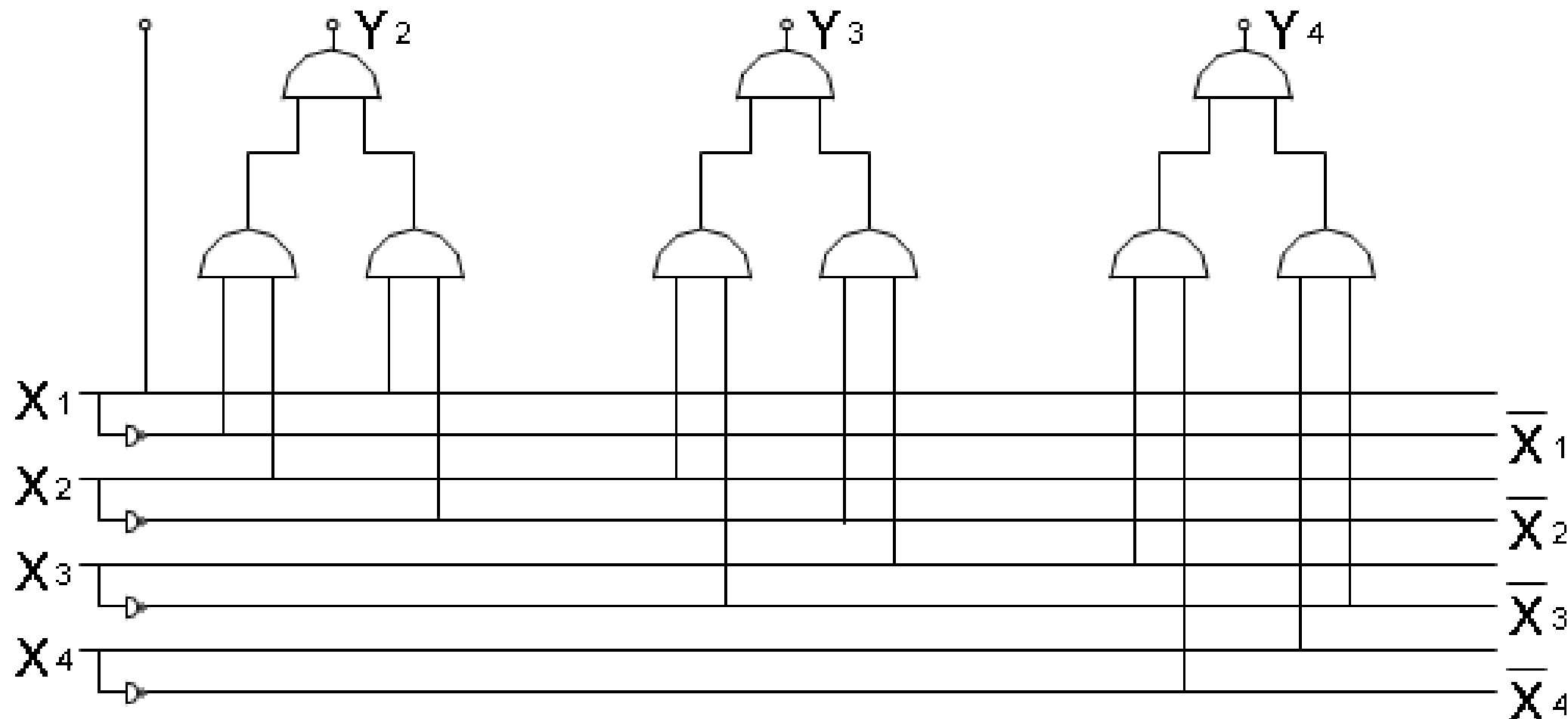
$$Y_1 = X_1$$

$$Y_2 = \overline{X_1}X_2 + \overline{X_2}X_1$$

$$Y_3 = \overline{X_3}X_2 + \overline{X_2}X_3$$

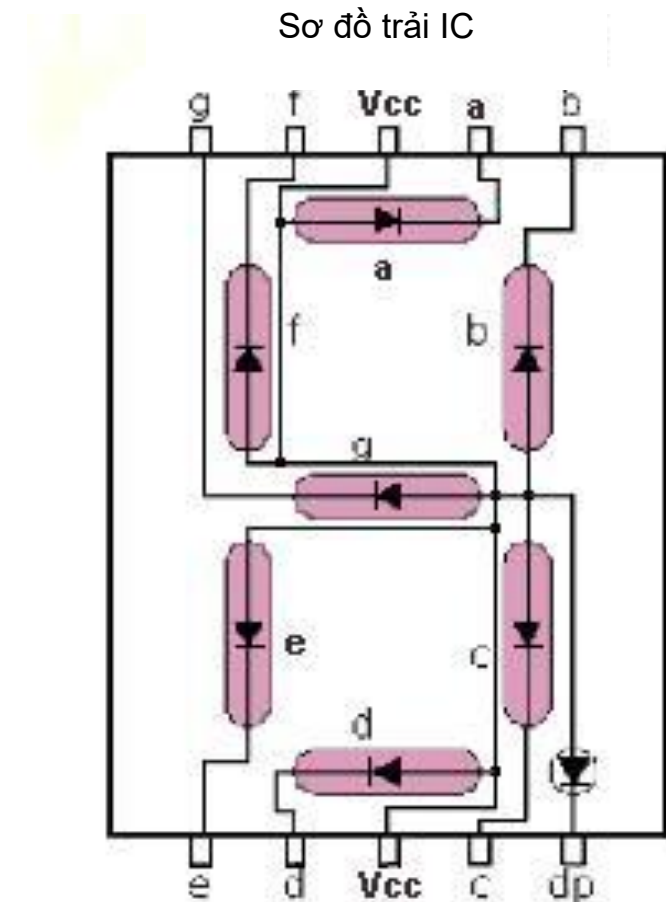
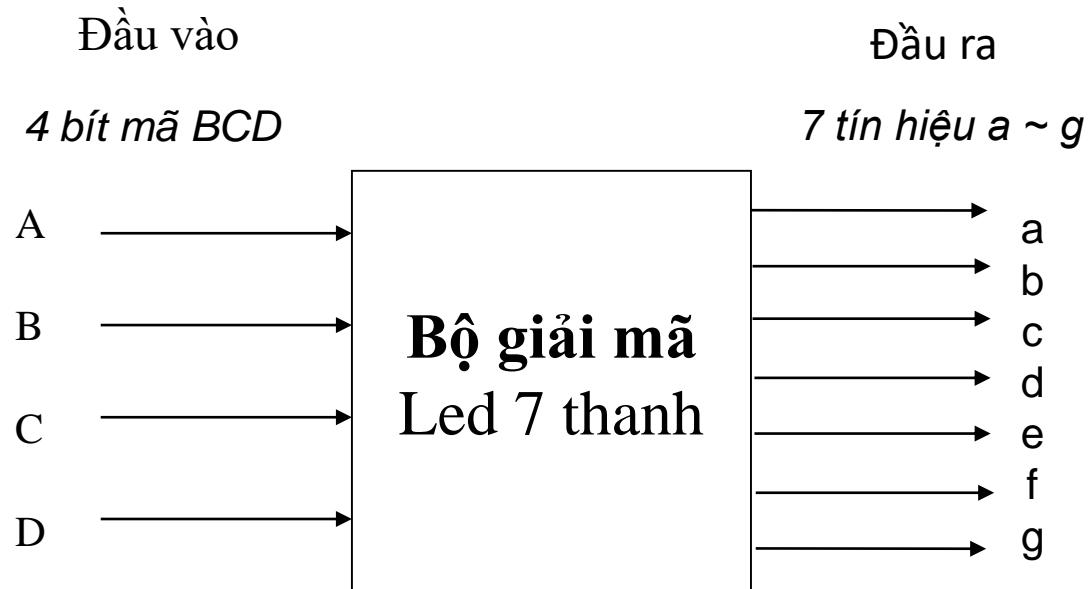
$$Y_4 = \overline{X_4}X_3 + \overline{X_3}X_4$$

Mạch (Sơ đồ logic)



II.2. Bộ giải mã hiển thị ký tự (hiển thị mã 7 thanh)

- Đây là bộ giải mã BCD sang mã 7 thanh sang mã BCD
- Bộ mã sử dụng 7 thanh LED. Sắp xếp theo một trật tự nhất định nhằm hiển thị ký tự số



Có 2 dạng hoạt động của LED 7 thanh

- Hoạt động ở mức tích cực dương (*LED 7 thanh Cathode chung*)
- Hoạt động ở mức tích cực âm (*LED 7 thanh Anode chung*)

Xây dựng bảng trạng thái (sử dụng mức logic âm)

- Với việc sử dụng mức logic âm thì khi nhận tín hiệu cấp bằng 1, thanh LED sẽ tắt và khi mức tín hiệu cấp bằng 0 thanh LED sẽ sáng
- 4 bit đầu vào thì sẽ tạo ra được $2^4 = 16$ tổ hợp trạng thái khác nhau, ta chỉ sử dụng hết 10 trạng thái → còn 06 trạng thái không sử dụng
- 06 trạng thái không sử dụng đến được gọi là trạng thái không xác định (*trên bảng Karnaugh ta sẽ ký hiệu chữ X*)

TT	A	B	C	D	a	b	c	d	e	f	g	Số hiển thị
1	0	0	0	0	0	0	0	0	0	0	1	0
2	0	0	0	1	1	0	0	1	1	1	0	1
3	0	0	1	0	0	0	1	0	0	1	0	2
4	0	0	1	1	0	0	0	0	1	1	0	3
5	0	1	0	0	1	1	0	1	1	0	0	4
6	0	1	0	1	0	1	0	0	1	0	0	5
7	0	1	1	0	0	0	0	0	0	0	0	6
8	0	1	1	1	0	0	0	1	1	1	1	7
9	1	0	0	0	0	0	0	0	0	0	0	8
10	1	0	0	1	0	0	0	0	1	0	0	9
11	1	0	1	0	Trạng thái không xác định							
12	1	0	1	1								
13	1	1	0	0								
14	1	1	0	1								
15	1	1	1	0								
16	1	1	1	1								

Đưa vào bìa Karnaugh hàm của từng thanh

a

CD				
AB	00	01	11	10
00		1		
01	1			1
11	x	x	x	x
10			x	x

$$a = B\bar{D} + \bar{A}\bar{B}\bar{C}D$$

b

CD				
AB	00	01	11	10
00				
01		1		1
11	x	x	x	x
10			x	x

$$b = B\bar{C}D + BC\bar{D}$$

c

CD				
AB	00	01	11	10
00				1
01				
11	x	x	x	x
10			x	x

$$c = \bar{B}C\bar{D}$$

d

CD \ AB	00	01	11	10
00		1		
01	1		1	
11	x	x	x	x
10		1	x	x

$$d = \overline{B}\overline{C}\overline{D} + \overline{B}\overline{C}D + BCD$$

e

CD \ AB	00	01	11	10
00		1	1	
01	1	1	1	
11	x	x	x	x
10		1	x	x

$$e = \overline{B}\overline{C} + D$$

f

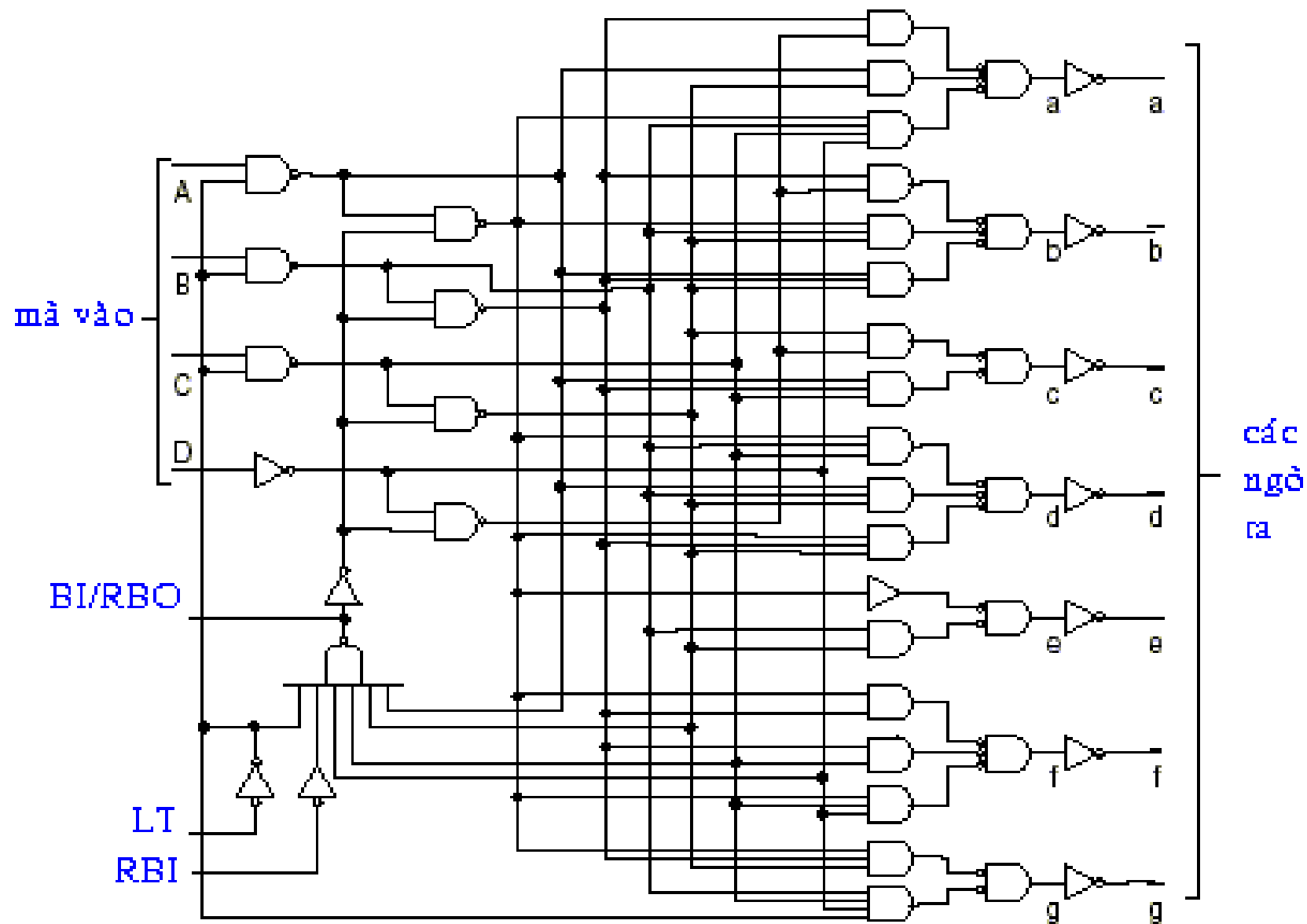
CD \ AB	00	01	11	10
00		1	1	1
01			1	
11	x	x	x	x
10			x	x

$$f = CD + \overline{B}C + \overline{A}\overline{B}D$$

g

CD \ AB	00	01	11	10
00	1	1		
01			1	
11	x	x	x	x
10			x	x

$$g = BCD + \overline{A}\overline{B}\overline{C}$$



số thấy

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

II.3 Bộ kiểm tra so sánh

Bộ kiểm tra so sánh là một mạch tổ hợp, có chức năng so sánh hai số nhị phân và chỉ ra mối quan hệ giữa chúng

Giả thiết ta có hai số nhị phân A và B \rightarrow ta có thể thấy **A = B**, **A > B** hay **A < B**

Để thiết kế bộ kiểm tra so sánh 2 số nhị phân, ta bắt đầu xây dựng bộ so sánh 2 số 1 bit

a) Bộ so sánh 2 số nhị phân 1bit

Giả thiết có hai số nhị phân 1 bit là a_i và b_i



Bảng chân lý

F_1 : Là hàm tín hiệu ra khi $a_i = b_i$

F_2 : Là hàm tín hiệu ra khi $a_i > b_i$

F_3 : Là hàm tín hiệu ra khi $a_i < b_i$

a_i	b_i	F_1	F_2	F_3
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

Ta có hàm ra như sau



$$F_1 = \overline{a_i} \overline{b_i} + a_i b_i = \overline{a_i \oplus b_i}$$

$$F_2 = \overline{a_i} b_i$$

$$F_3 = a_i \overline{b_i}$$

b) Bộ so sánh hai số nhị phân nhiều bit

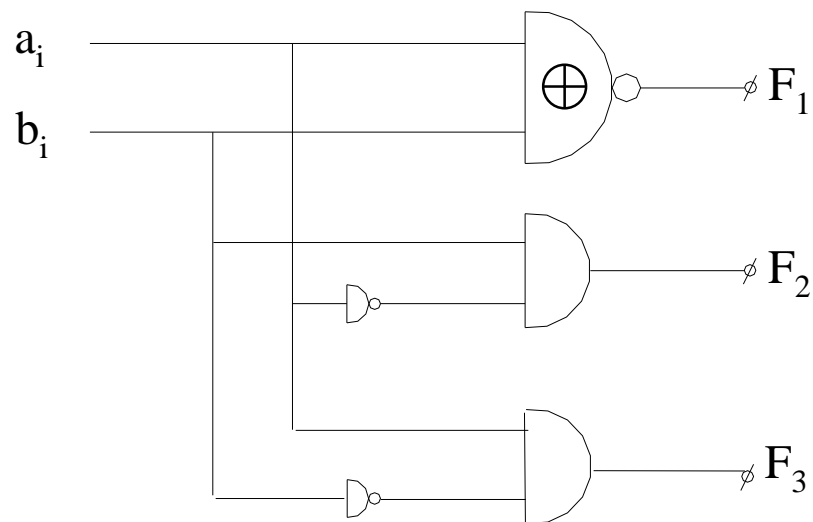
Giả thiết ta có 2 số nhị phân nhiều biết A và B

A: $a_n a_{n-1} \dots a_1$ **B:** $b_n b_{n-1} \dots b_1$ trong đó a_1 và b_1 là bit có trọng số thấp nhất

Phương pháp thiết kế:

- Biện luận đầy đủ các trường hợp có thể xảy ra
- Căn cứ từ biện luận, xây dựng hàm ra cho mạch so sánh

Sơ đồ mạch



Ví dụ : Xây dựng mạch so sánh cho 2 số nhị phân 3 bit A và B

Giải:

Giả thiết 2 số nhị phân A và B như sau: **A: $a_3 a_2 a_1$** và **B: $b_3 b_2 b_1$**

(a_1 và b_1 là hai bit có trọng số nhỏ nhất)

Biện luận (*phân tích các trường hợp có thể xảy ra*)

A = B khi và chỉ khi $a_3 = a_3$ và $a_2 = a_2$ và $a_1 = b_1$

A < B khi và chỉ khi $a_3 < b_3$ hoặc $a_3 = b_3$ và $a_2 < b_2$ hoặc $a_3 = b_3$ và $a_2 = b_2$ và $a_3 < b_3$

A > B khi và chỉ khi $a_3 > b_3$ hoặc $a_3 = b_3$ và $a_2 > b_2$ hoặc $a_3 = b_3$ và $a_2 = b_2$ và $a_3 > b_3$

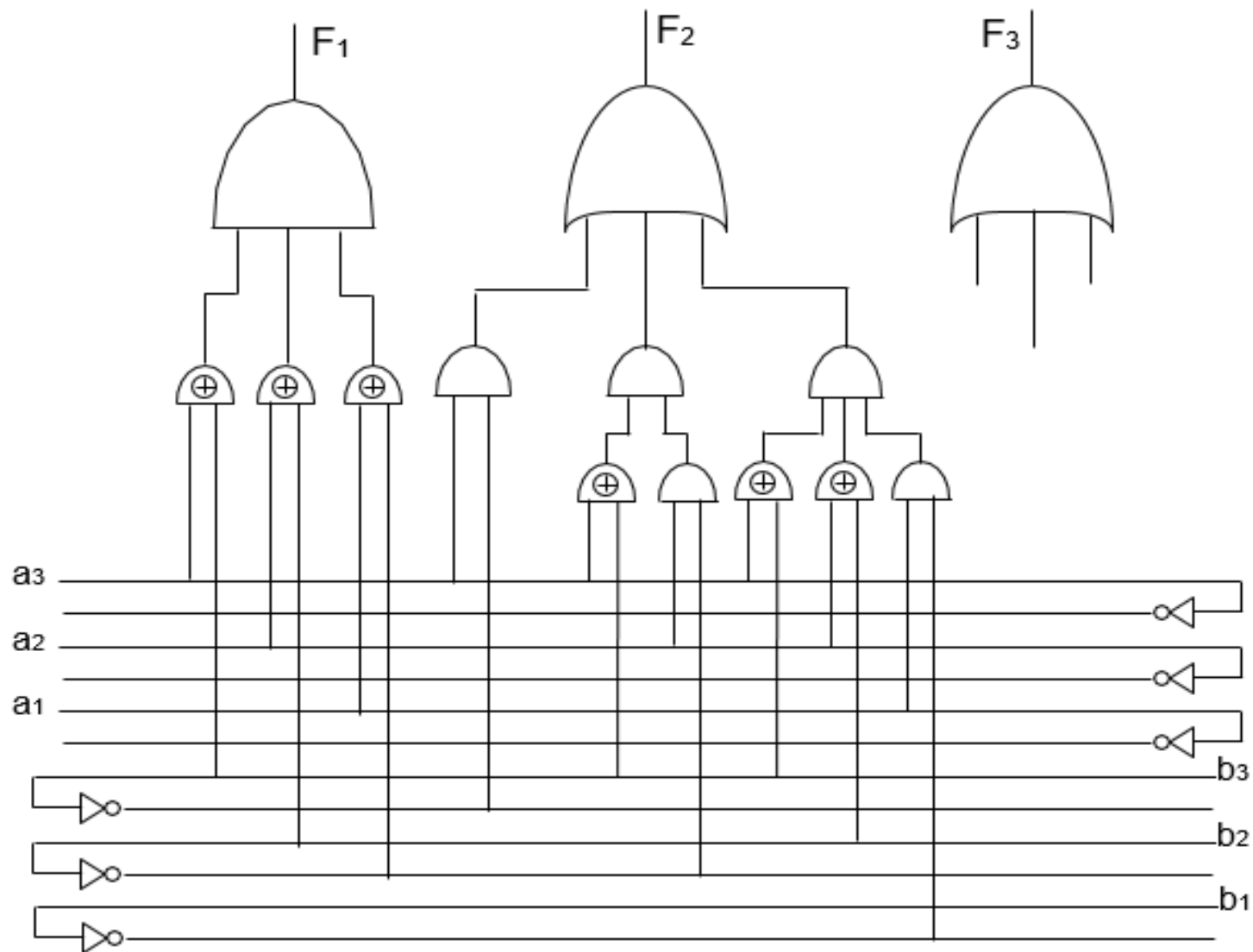
Từ biện luận trên ta sẽ đưa ra các hàm đầu ra

Gọi F_1, F_2, F_3 là các hàm
tín hiệu đầu ra của bộ so sánh

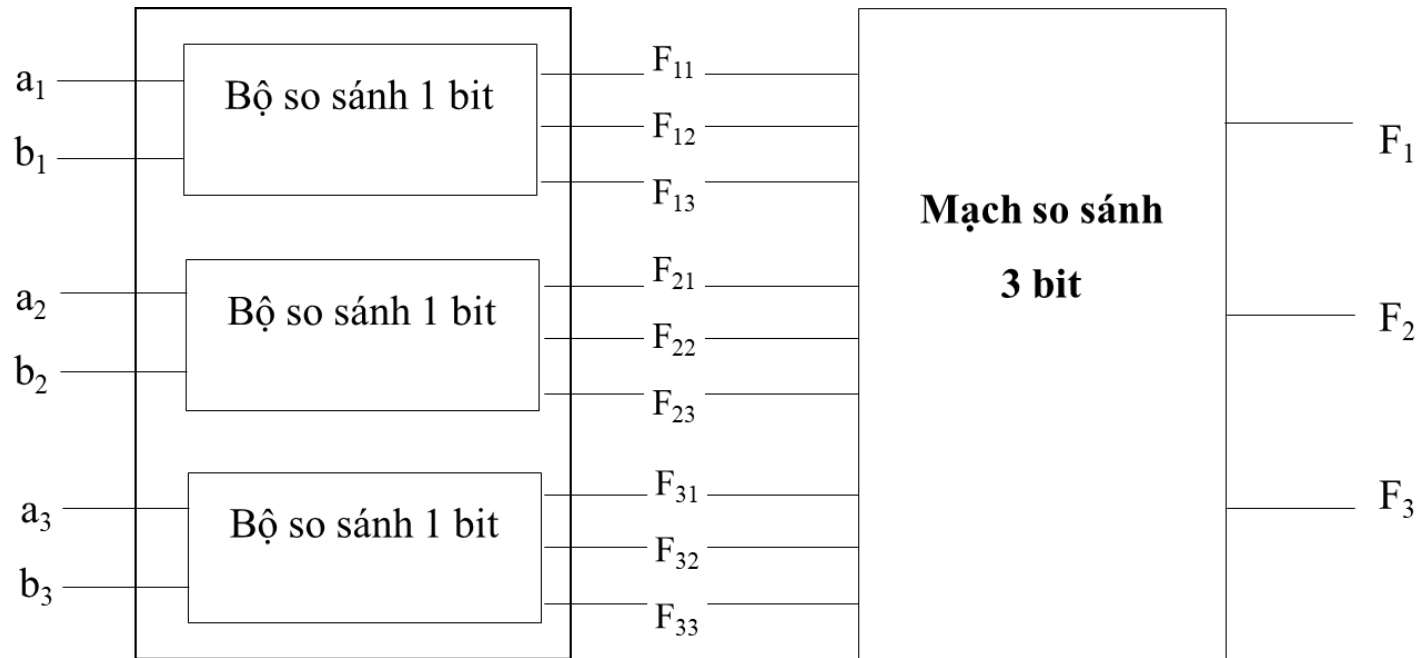
$$F_1 = \overline{a_3 \oplus b_3} * \overline{a_2 \oplus b_2} * \overline{a_1 \oplus b_1}$$

$$F_2 = \overline{a_3}b_3 + \overline{a_3 \oplus b_3}.\overline{a_2}b_2 \\ + \overline{a_3 \oplus b_3}.\overline{a_2 \oplus b_2}.\overline{a_1}b_1$$

$$F_3 = a_3\overline{b_3} + \overline{a_3 \oplus b_3}.a_2\overline{b_2} \\ + \overline{a_3 \oplus b_3}.\overline{a_2 \oplus b_2}.a_1\overline{b_1}$$



❖ Xây dựng mạch qua các bộ so sánh 2 số nhị phân 1 bit



+ Gọi F_{11} là Là hàm tín hiệu ra khi $a_1 = b_1$

+ Gọi F_{12} là Là hàm tín hiệu ra khi $a_1 < b_1$

+ Gọi F_{13} là Là hàm tín hiệu ra khi $a_1 > b_1$

+ Gọi F_{21} là Là hàm tín hiệu ra khi $a_2 = b_2$

+ Gọi F_{22} là Là hàm tín hiệu ra khi $a_2 < b_2$

+ Gọi F_{23} là Là hàm tín hiệu ra khi $a_2 > b_2$

+ Gọi F_{31} là Là hàm tín hiệu ra khi $a_3 = b_3$

+ Gọi F_{32} là Là hàm tín hiệu ra khi $a_3 < b_3$

+ Gọi F_{33} là Là hàm tín hiệu ra khi $a_3 > b_3$

$$F_1 = F_{31} \cdot F_{21} \cdot F_{11}$$

$$F_2 = F_{32} + F_{31} \cdot F_{22} + F_{31} \cdot F_{21} \cdot F_{12}$$

$$F_3 = F_{33} + F_{31} \cdot F_{23} + F_{31} \cdot F_{21} \cdot F_{13}$$

Với

F_1 là tín hiệu ra khi $A = B$

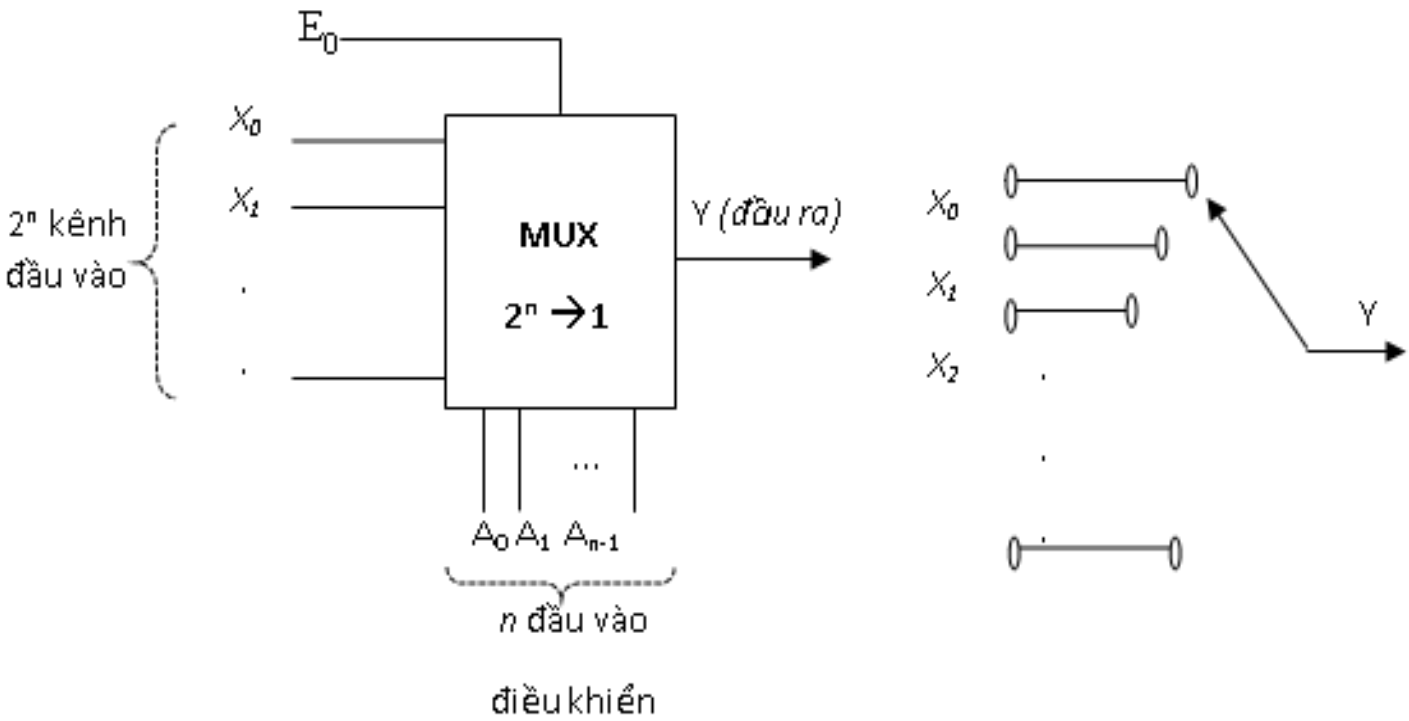
F_2 là tín hiệu ra khi $A < B$

F_3 là tín hiệu ra khi $A > B$

II.4. Bộ tách ghép kênh

II.4.1. Bộ ghép kênh (MUX - Multiplexer)

Bộ ghép kênh còn được gọi là bộ dồn kênh (MUX) là một mạch logic tổ hợp sử dụng n bit điều khiển để chọn 1 trong 2^n đầu tín hiệu vào



$\{X_0; X_1; \dots\}$: $2n$ kênh đầu vào bộ ghép kênh

Y : Tín hiệu ra bộ ghép kênh (đường trung kế)

$\{A_0; A_1; \dots; A_{n-1}\}$: n bit điều khiển

E_0 : Tín hiệu ngắt DEMUX

❖ *Nguyên lý hoạt động của MUX:*

Nếu giá trị tổ hợp bit $(A_0A_1 \dots A_n)_2 = (J)_{10} \rightarrow Y = X_j$

Như vậy ta có phương trình tín hiệu ra của MUX

$$Y = X_0(\bar{A}_0\bar{A}_1 \dots \bar{A}_{n-1})E_0 + X_1(\bar{A}_0\bar{A}_1 \dots \bar{A}_{n-2}A_{n-1})E_0 + X_{n-1}(A_0A_1 \dots A_{n-1})E_0$$

Ví dụ: Hãy thiết kế MUX sử dụng 2 bit điều khiển để ghép 4 kênh tín hiệu vào

Giải:

Gọi 4 kênh tín hiệu vào là $\{X_0, X_1, X_2, X_3\}$

Y là trung kế đầu ra

AB là 2 bit điều khiển kết nối của MUX

E_0 là tín hiệu ngắt MUX

➡ $Y = X_0(\overline{A} \overline{B})E_0 + X_1(\overline{A} B)E_0 + X_2(A \overline{B})E_0 + X_3(A B)E_0$

Ta có mạch của MUX 4 kênh

$E = 0 \rightarrow Y = 0$ (MUX ngắt không làm việc)

$E = 1 \rightarrow Y = 1$ (MUX ở chế độ làm việc)

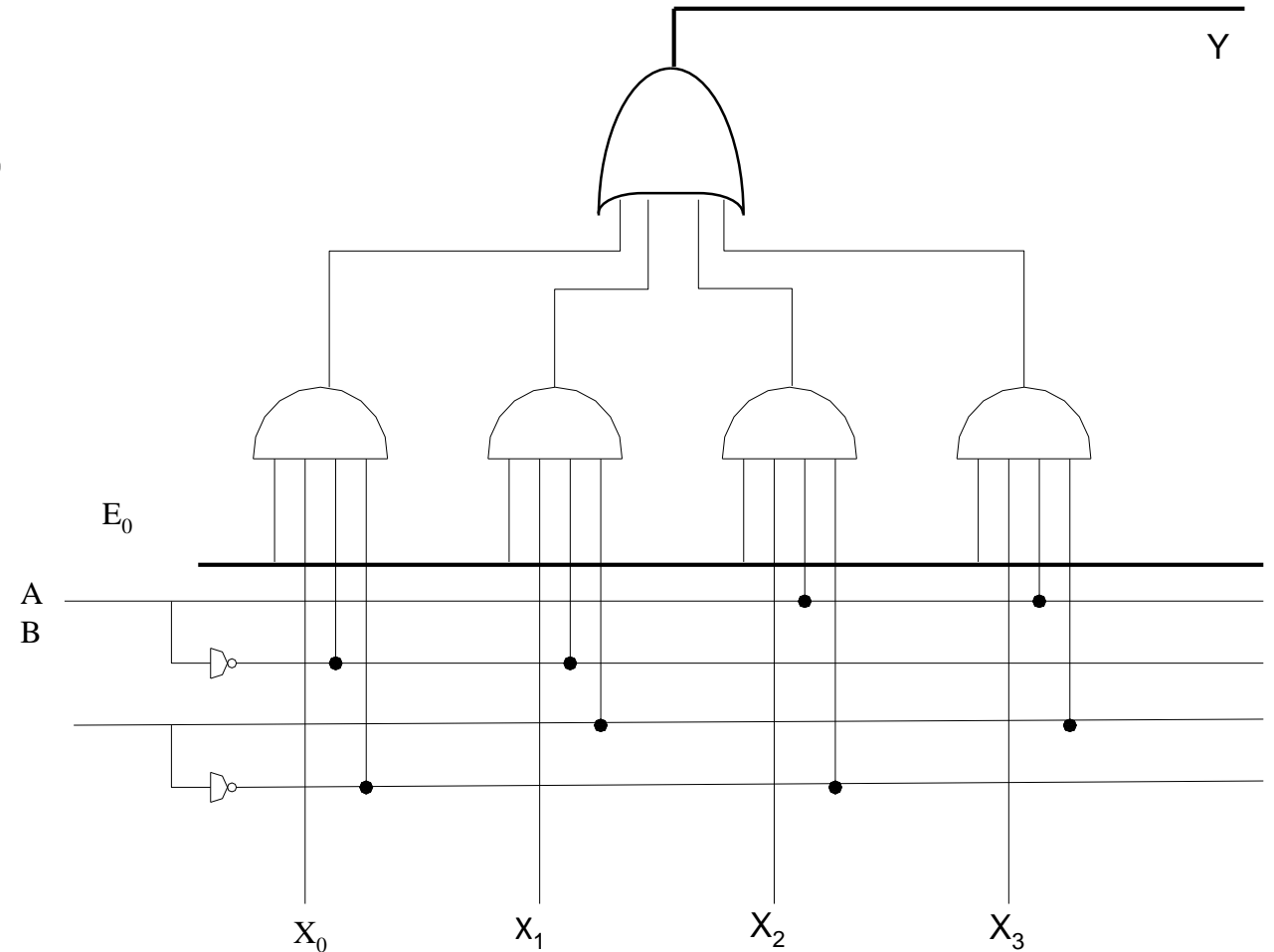
Tổ hợp bit AB và các kết nối

$AB = 00 \rightarrow$ điều khiển kết nối Y với X_0

$AB = 01 \rightarrow$ điều khiển kết nối Y với X_1

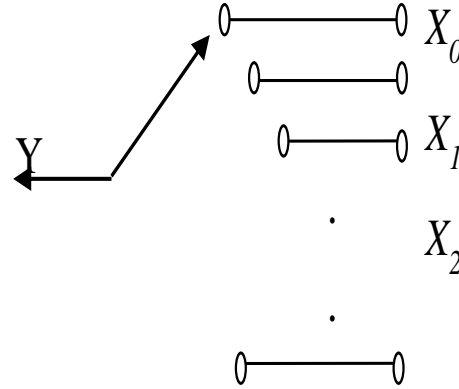
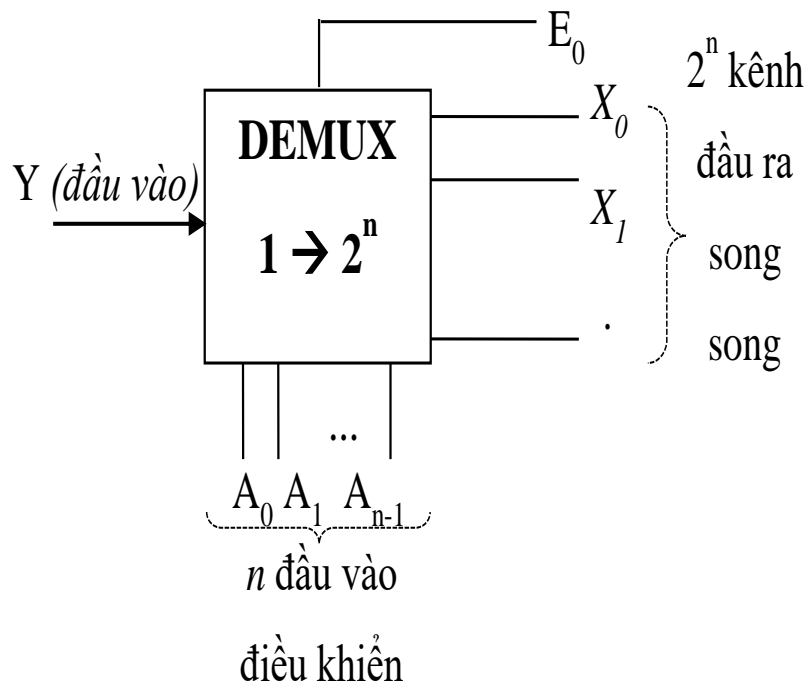
$AB = 10 \rightarrow$ điều khiển kết nối Y với X_2

$AB = 11 \rightarrow$ điều khiển kết nối Y với X_3



II.4.2. Bộ phân kênh (DEMUX - Demultiplexer)

Bộ chuyển mạch phân kênh hay còn gọi là tách kênh (DEMUX) có chức năng ngược lại với mạch dồn kênh tức là: tách kênh truyền thành 1 trong các kênh dữ liệu song song.



$\{X_0; X_1; \dots\}$: 2^n kênh đầu ra song song

Y : Tín hiệu vào bộ phân kênh (đường trung kế)

$\{A_0; A_1; \dots A_{n-1}\}$: n bit điều khiển

❖ Nguyên lý hoạt động của DEMUX

Nếu giá trị tổ hợp bit $(A_0A_1 \dots A_n)_2 = (J)_{10} \rightarrow Y = X_j$

Như vậy ta có phương trình tín hiệu ra của DEMUX

$$X_0 = Y(\bar{A}_0\bar{A}_1 \dots \bar{A}_{n-1})E_0 \quad X_1 = Y(\bar{A}_0\bar{A}_1 \dots \bar{A}_{n-2}A_{n-1})E_0 \quad X_{n-1} = Y(A_0A_1 \dots A_{n-1})E_0$$

Ví dụ: Hãy thiết kế DEMUX sử dụng 2 bit điều khiển để tách 1 kênh tín hiệu vào thành 4 kênh tín hiệu ra song song

Giải: Gọi Y là tín hiệu đầu vào (*trung kế*)

4 kênh tín hiệu ra là $\{X_0, X_1, X_2, X_3\}$

AB là 2 bit điều khiển kết nối của DEMUX

E_0 là tín hiệu ngắt DEMUX

Ta có các hàm ra:

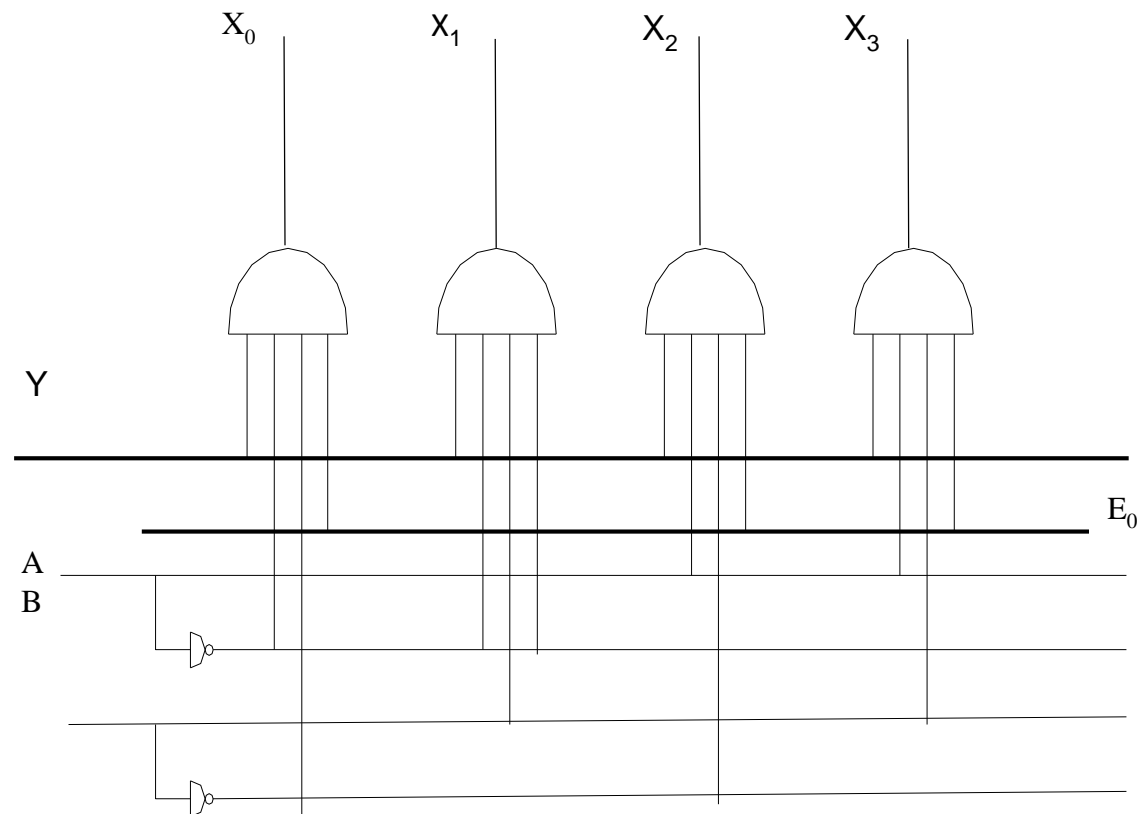
$$X_0 = Y(\overline{A} \overline{B})E_0$$

$$X_1 = Y(\overline{A} B)E_0$$

$$X_2 = Y(A \overline{B})E_0$$

$$X_3 = Y(A B)E_0$$

Ta có mạch DEMUX 4 kênh
như hình vẽ



BÀI TẬP

1) Thiết kế bộ ghép kênh – MUX thỏa mãn các yêu cầu sau

Sử dụng 3 bit điều khiển để ghép 6 kênh đầu vào

Bài 2: Thiết kế mô hình mạch cho khóa điện tử của két sắt thỏa mãn các yêu cầu sau

- Đầu vào nhập là các chữ số thập phân
- Mã khóa điện tử gồm 4 chữ số như hình vẽ
- Mã khóa của két đã đặt trước

Khi nhập mã vào, nếu mã nhập trùng với mã của két thì mạch choc ho phép mở khoa

