

CHƯƠNG 2: MẠCH LOGIC TỔ HỢP

TÊN BÀI GIẢNG:

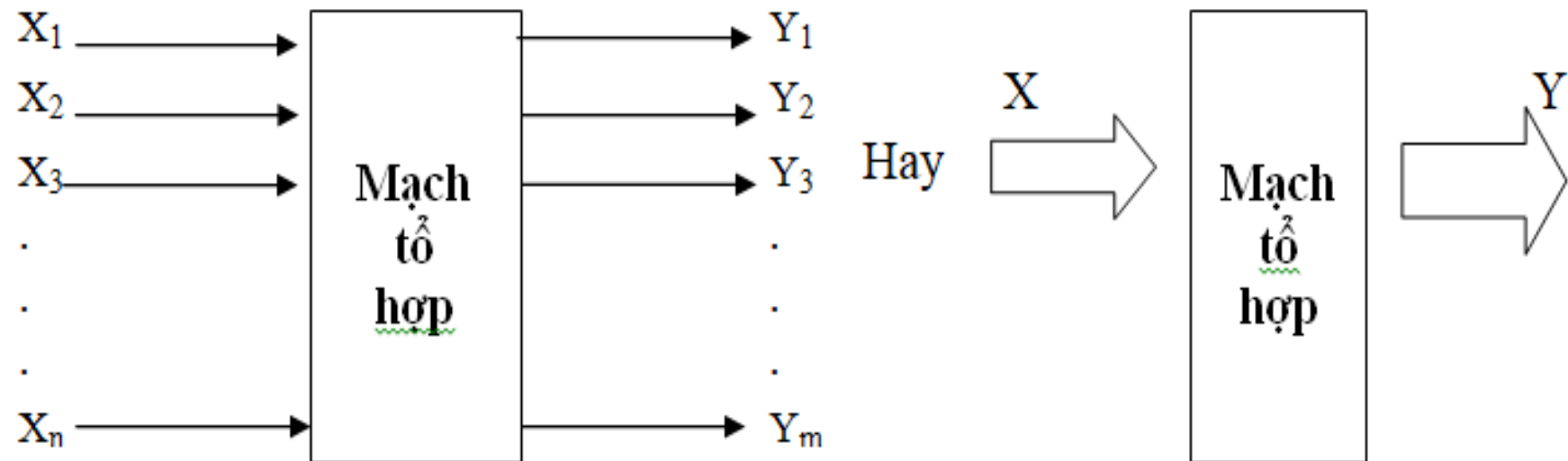
MẠCH LOGIC TỔ HỢP VÀ PHƯƠNG PHÁP THIẾT KẾ

I. Khái niệm

Mạch tổ hợp là mạch mà trị số của tín hiệu đầu ra ở thời điểm bất kỳ chỉ phụ thuộc vào tổ hợp giá trị biến đầu vào tại thời điểm đó

Ghi chú: Các trạng thái của mạch tổ hợp tại các thời điểm trước không làm ảnh hưởng đến tín hiệu ra của mạch tại thời điểm sau

II. Mô hình quan hệ của mạch logic tổ hợp



Mạch có:

n đầu vào (x); $X = \{X_1, X_2, \dots, X_n\}$

m đầu ra (y); $Y = \{Y_1, Y_2, \dots, Y_m\}$

Phương trình quan hệ mô tả

$$Y_j = f(x_1, x_2, \dots, x_n) \text{ với } \forall j \in \{1 \div m\}$$

Sơ đồ mạch logic tổ hợp tổng quát

III. Thiết kế mạch logic tổ hợp

- Phân tích yêu cầu và xác định tín hiệu vào - ra.
- Xác định bảng trạng thái hay bảng chân lý.
- Tối thiểu hóa.
- Đưa ra hàm tối thiểu: tổng của tích hoặc tích của tổng.
- Vẽ sơ đồ logic bằng các cổng AND - OR hoặc OR - AND

Biến đổi sang sơ đồ vi mạch thông dụng

Chọn linh kiện và ráp mạch.

a) Phân tích yêu cầu:

- + Đặt bài toán dưới dạng bài toán logic.
- + Xác định các biến số đầu vào và hàm đầu ra.
- + Xác định quan hệ giữa hàm và biến.

b) Lập bảng chân lý

- + Bước đầu tiên để lập bảng chân lý là ta phải phân tích logic và đặt ra các quy ước, các giả thiết.
- + Phân tích các trường hợp có thể xảy ra và giải bài toán cho từng trường hợp

Chú ý:

Mọi trạng thái tín hiệu vào luôn được liệt kê đầy đủ. Nếu có tổ hợp nào của biến đầu vào mà không sử dụng đến hoặc không có tín hiệu đầu ra cần thì tại đầu ra ta đánh dấu “X”; chúng được gọi là trạng thái không xác định

c) Tối thiểu hóa

Nếu lượng biến số không nhiều thì ta nên sử dụng phương pháp tối thiểu hoá bằng bảng Karnaugh.

Nếu lượng biến số nhiều thì ta nên tối thiểu hóa bằng phương pháp Quine - Mc.Cluskey hoặc bằng phương pháp biến đổi đại số

d) Sơ đồ logic

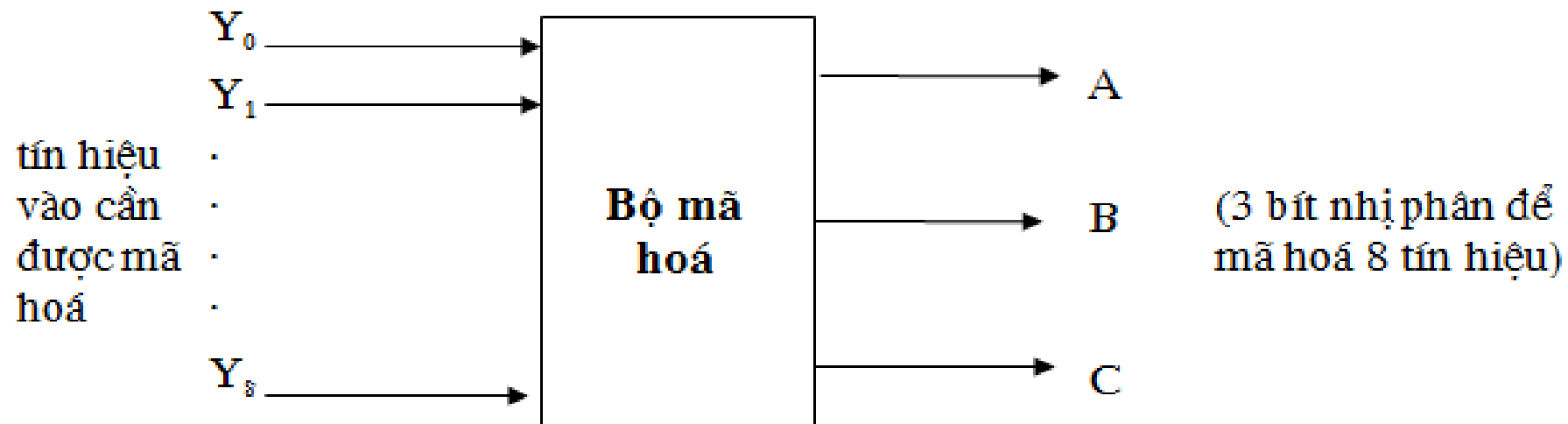
Sử dụng các linh kiện logic cơ bản để vẽ sơ đồ logic

Việc thực hiện sơ đồ logic phải căn cứ vào linh kiện được cung cấp của đầu bài và yêu cầu từ thực tế

IV. Các bộ mã hoá

Trong máy tính, các hệ thống số chỉ sử lý dữ liệu nhị phân; do đó vấn đề đặt ra là làm sao để tạo ra một giao diện dễ dàng giữa người và máy tính, nghĩa là máy tính phải hiểu các ký tự và thực hiện được những bài toán do con người đặt ra.

a) Bộ mã hóa nhị phân



Sử dụng bộ mã hoá nhị phân thì ta chỉ cần n بیت để mã hoá 2^n tín hiệu

Ví dụ:

Thiết kế bộ mã hóa 3 bit để mã hoá 8 tín hiệu vào
Vẽ mạch chỉ sử dụng phần tử NAND

Giải:

- Gọi 8 tín hiệu vào là Y_0 đến Y_7
- Gọi 3 bit đầu ra là A, B, C

Vậy trên bài toán có:

- 8 biến số vào Y_0, Y_1, \dots, Y_7
- 3 hàm đầu ra A, B, C

Ta có bảng chân lý

Bảng chân lý



Hàm

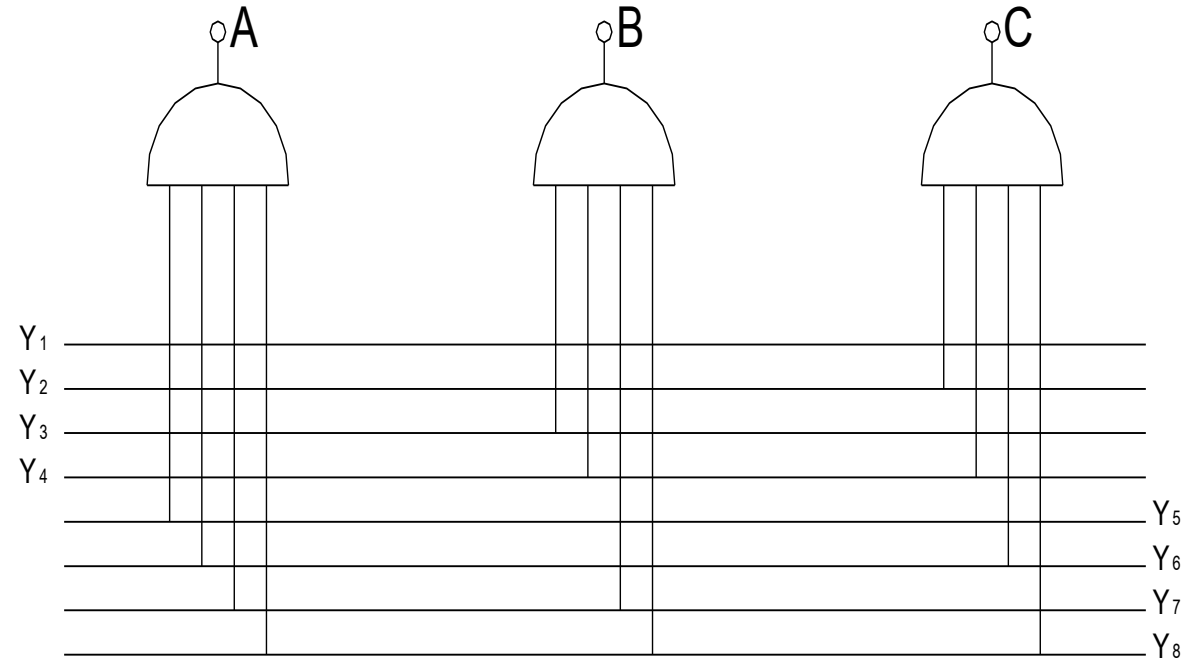
$$A = Y_5 + Y_6 + Y_7 + Y_8$$

$$B = Y_3 + Y_4 + Y_7 + Y_8$$

$$C = Y_2 + Y_4 + Y_6 + Y_8$$

Y	A	B	C
Y_1	0	0	0
Y_2	0	0	1
Y_3	0	1	0
Y_4	0	1	1
Y_5	1	0	0
Y_6	1	0	1
Y_7	1	1	0
Y_8	1	1	1

Ta có sơ đồ mạch



Một phương thức biểu diễn hàm qua dạng minterm, đó là ta có thể chuyển đổi các cổng logic AND, OR thành một dạng duy nhất, đó là cổng NAND

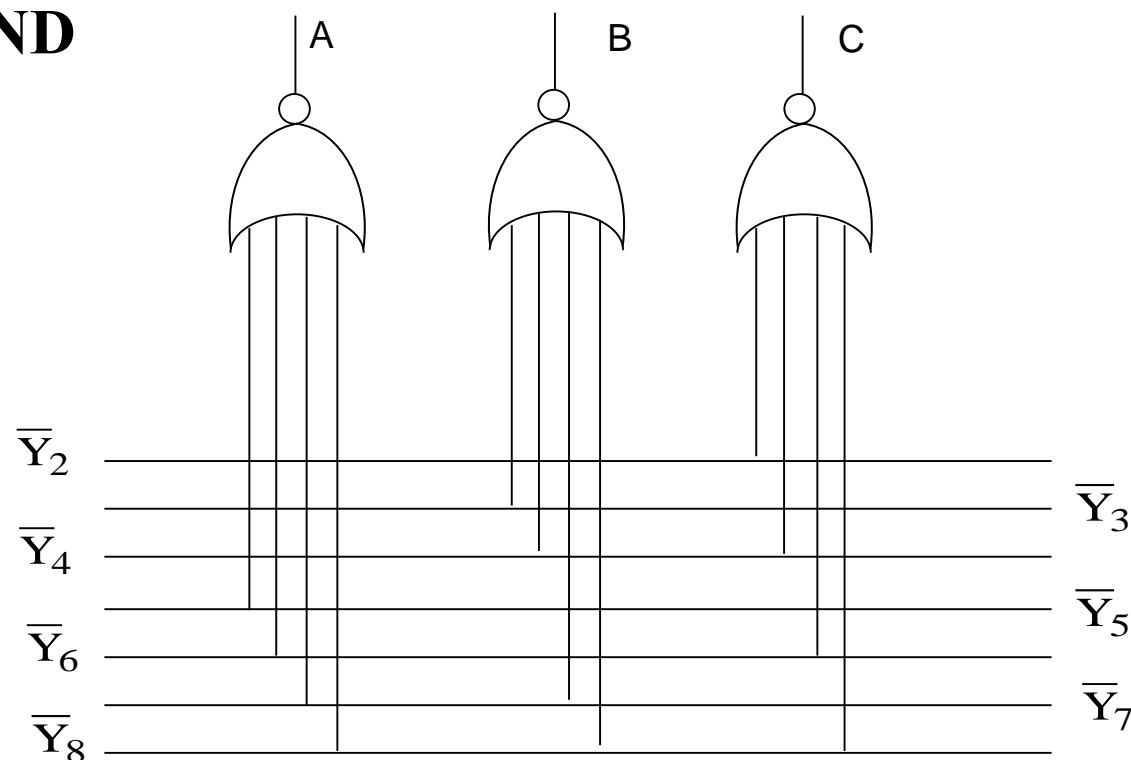
Bởi vì cổng NAND là cổng logic có đáp ứng nhanh nhất trong họ logic TTL, đây là một đặc tính quan trọng chúng ta cần lưu ý

Biến đổi hàm để mạch toàn phần tử NAND

$$A = \overline{\overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7} \cdot \overline{Y_8}}$$

$$B = \overline{\overline{Y_3} \cdot \overline{Y_4} \cdot \overline{Y_7} \cdot \overline{Y_8}}$$

$$C = \overline{\overline{Y_2} \cdot \overline{Y_4} \cdot \overline{Y_6} \cdot \overline{Y_8}}$$



b) Mã BCD (Binary Code Decimal) : Nhị phân mã hoá thập phân

Là bộ mã nhị phân, sử dụng để mã hóa 10 chữ số thập phân từ 0 → 9, các trạng thái còn lại không sử dụng đến.

Do bộ mã thập phân gồm 10 chữ số nên phải sử dụng ít nhất là 4 bit đầu ra.

Như vậy:

Tín hiệu vào Y là 10 chữ số trong hệ thập phân (0~9)

Giả thiết là tín hiệu ra là 4 bit A, B, C, D

Ta có bảng chân lý

Mã thập phân	Mã BCD			
Y	X ₁	X ₂	X ₃	X ₄
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Ta có các hàm ra như sau

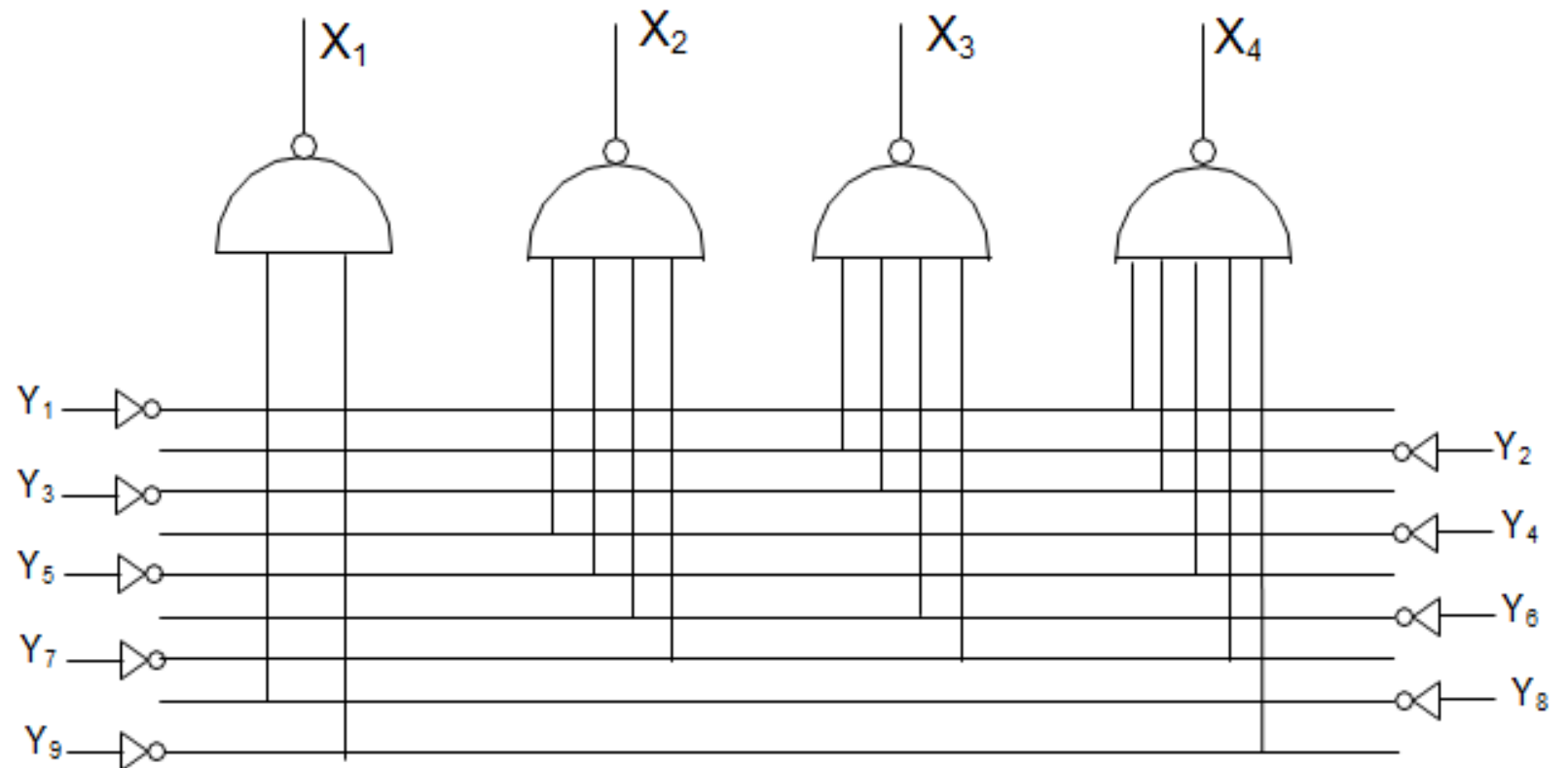
$$X_1 = Y_8 + Y_9 = \overline{\overline{Y_8 \cdot Y_9}}$$

$$X_2 = Y_4 + Y_5 + Y_6 + Y_7 = \overline{\overline{Y_4 \cdot Y_5 \cdot Y_6 \cdot Y_7}}$$

$$X_3 = Y_2 + Y_3 + Y_6 + Y_7 = \overline{\overline{Y_2 \cdot Y_3 \cdot Y_6 \cdot Y_7}}$$

$$X_4 = Y_1 + Y_3 + Y_6 + Y_7 + Y_9 = \overline{\overline{Y_1 \cdot Y_3 \cdot Y_5 \cdot Y_7 \cdot Y_9}}$$

Ta có mạch như hình vẽ:
(chỉ sử dụng phân tử NAND)



c) Mã GRAY

Là loại mã không có trọng số, các từ mã kế cận nhau chỉ khác nhau ở một biến số

Mã GRAY dùng để biểu diễn bảng Karnaugh

Bảng bên là phương thức xây dựng mã Gray 4 bit

Số thứ tự	Mã Gray 4 bit			
Y	X ₁	X ₂	X ₃	X ₄
1	0	0	0	0
2	0	0	0	1
3	0	0	1	1
4	0	0	1	0
5	0	1	1	0
6	0	1	1	1
7	0	1	0	1
8	0	1	0	0
9	1	1	0	0
10	1	1	0	1
11	1	1	1	1
12	1	1	1	0
13	1	0	1	0
14	1	0	1	1
15	1	0	0	1
16	1	0	0	0

d) Mã dư 3

Được tạo thành bằng cách cộng thêm 3 đơn vị vào mã BCD

Các mã dư 3 được sử dụng rộng rãi trong các thiết bị tính toán số học

(của hệ thống xử lý hoặc gia công các tín hiệu số)

Mã BCD + 3 đơn vị (011) = Mã dư 3

Mã thập phân	X ₁	X ₂	X ₃	X ₄	Mã dư 3
0	0	0	0	0	Bỏ
1	0	0	0	1	Bỏ
2	0	0	1	0	Bỏ
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	2
6	0	1	1	0	3
7	0	1	1	1	4
8	0	0	0	0	5
9	1	0	0	1	6
	1	0	1	0	7
	1	0	0	1	8
	1	1	0	0	9

e) Bộ giải mã

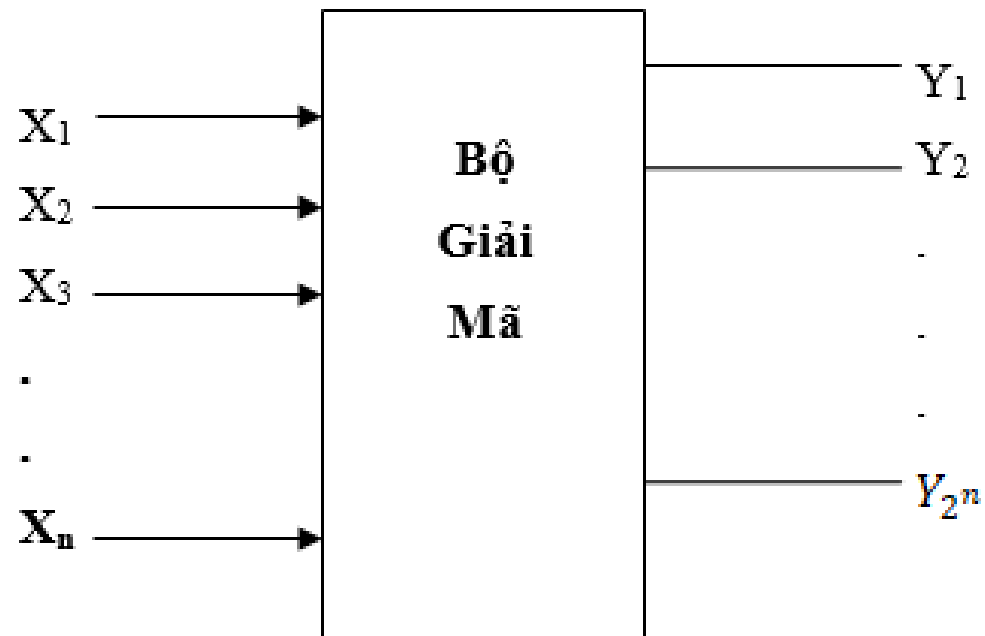
Là bộ có chức năng phiên dịch từ các mã trở lại thành tín hiệu đầu ra

Nguyên lý bộ giải mã như hình vẽ

Trong đó

X_1, X_2, \dots, X_n : Tín hiệu mã đầu vào

Y_1, Y_2, \dots, Y_{2^n} : Tín hiệu mã đầu ra



Ghi chú: Nếu đầu vào là bộ mã với n bit (n đầu vào) thì tín hiệu đầu ra bộ giải mã không được phép vượt quá 2^n tín hiệu.

Ví dụ: Thiết kế bộ giải mã nhị phân 3 bit thành 8 tín hiệu đầu ra

Giải

Bảng chân lý

Phân tích yêu cầu bài toán:

Giả thiết ta cần thiết kế
**bộ giải mã 3 bit nhị
phân thành 8 tín hiệu
đầu ra**

Từ yêu cầu đầu bài, ta gọi
+ 3 bit đầu vào là $a; b; c$
+ 8 tín hiệu ra là Y_1 đến Y_8

TT	a	b	c	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8
1	0	0	0	1	0	0	0	0	0	0	0
2	0	0	1	0	1	0	0	0	0	0	0
3	0	1	0	0	0	1	0	0	0	0	0
4	0	1	1	0	0	0	1	0	0	0	0
5	1	0	0	0	0	0	0	1	0	0	0
6	1	0	1	0	0	0	0	0	1	0	0
7	1	1	0	0	0	0	0	0	0	1	0
8	1	1	1	0	0	0	0	0	0	0	1

Đưa ra hàm (tối thiểu hoá nếu có thể)

$Y_1 = \bar{a}\bar{b}\bar{c}$

$Y_2 = \bar{a}\bar{b}c$

$Y_3 = \bar{a}b\bar{c}$

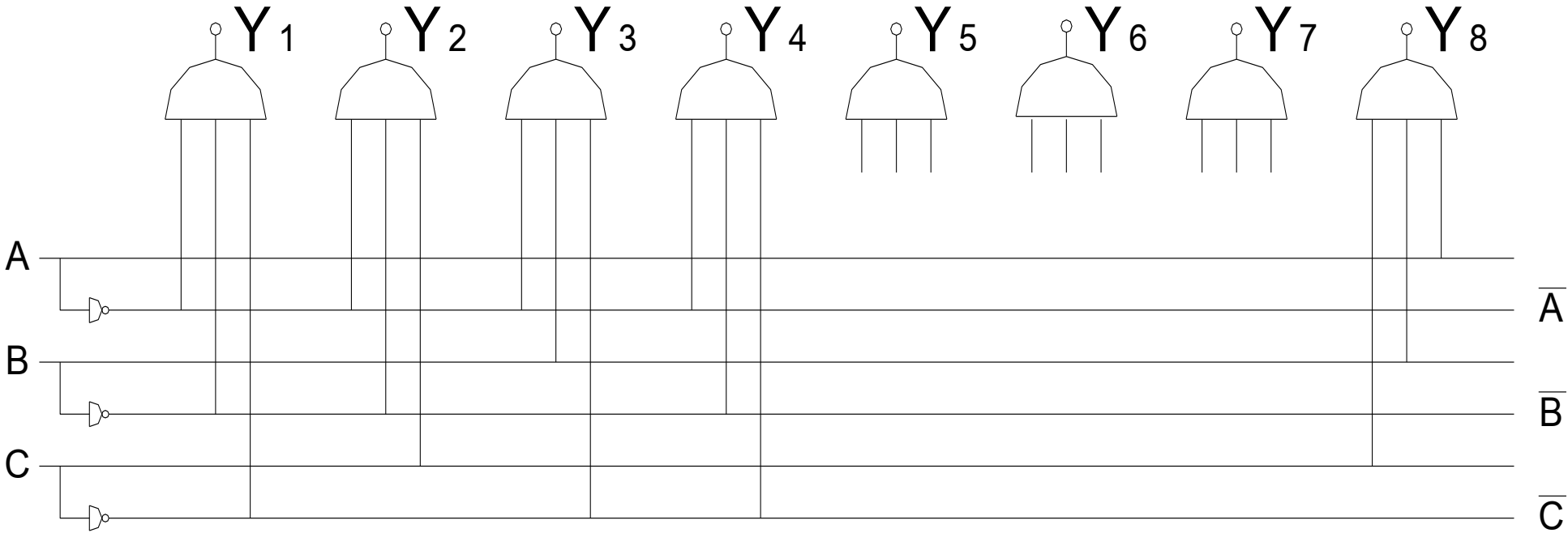
$Y_4 = \bar{a}bc$

$Y_5 = a\bar{b}\bar{c}$

$Y_6 = a\bar{b}c$

$Y_7 = ab\bar{c}$

$Y_8 = abc$



TỔNG KẾT

Xây dựng mạch logic tổ hợp cần xác định biến đầu vào, hàm đầu ra, quan hệ giữa hàm số và biến số

Xây dựng mạch (sơ đồ logic) cần chú trọng yêu cầu về linh kiện cho trước của đầu bài (*Ví dụ: Xây dựng mạch bằng các phần tử NAND, NOR*) để thực hiện chính xác các yêu cầu đặt ra

BÀI TẬP

Thiết kế mạch điện kiểm tra bit lẻ (chức năng như sau)

Nếu các tín hiệu đầu vào có số lẻ ở mức cao thì tín hiệu đầu ra ở mức cao, còn lại đầu ra ở mức thấp

Tín hiệu vào là 5 bit