

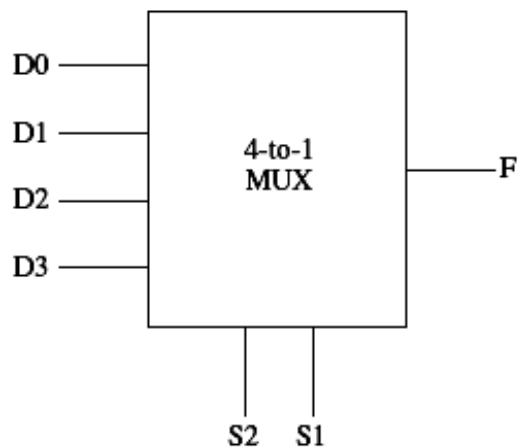
Mạch tổ hợp

➤ Khái niệm

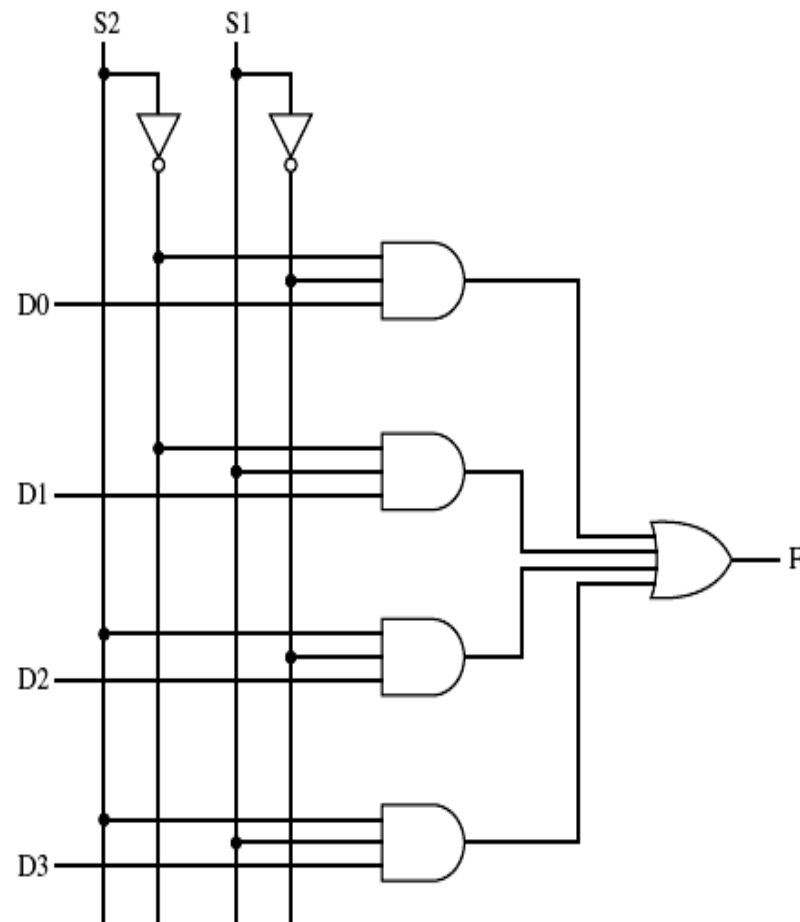
- Mạch tổ hợp (combinational circuit) là mạch logic trong đó tín hiệu ra chỉ phụ thuộc tín hiệu vào ở thời điểm hiện tại.
- Là mạch không nhớ (memoryless) và được thực hiện bằng các cổng logic cơ bản
- Mạch tổ hợp được cài đặt từ 1 hàm hoặc bảng chân trị cho trước
- Được ứng dụng nhiều trong thiết kế mạch máy tính

Bộ dồn kênh (Multiplexer)

- 2^n đầu vào dữ liệu D
- n đầu vào lựa chọn S
- 1 đầu ra F
- (S) xác định đầu vào (D) nào sẽ được nối với đầu ra (F)

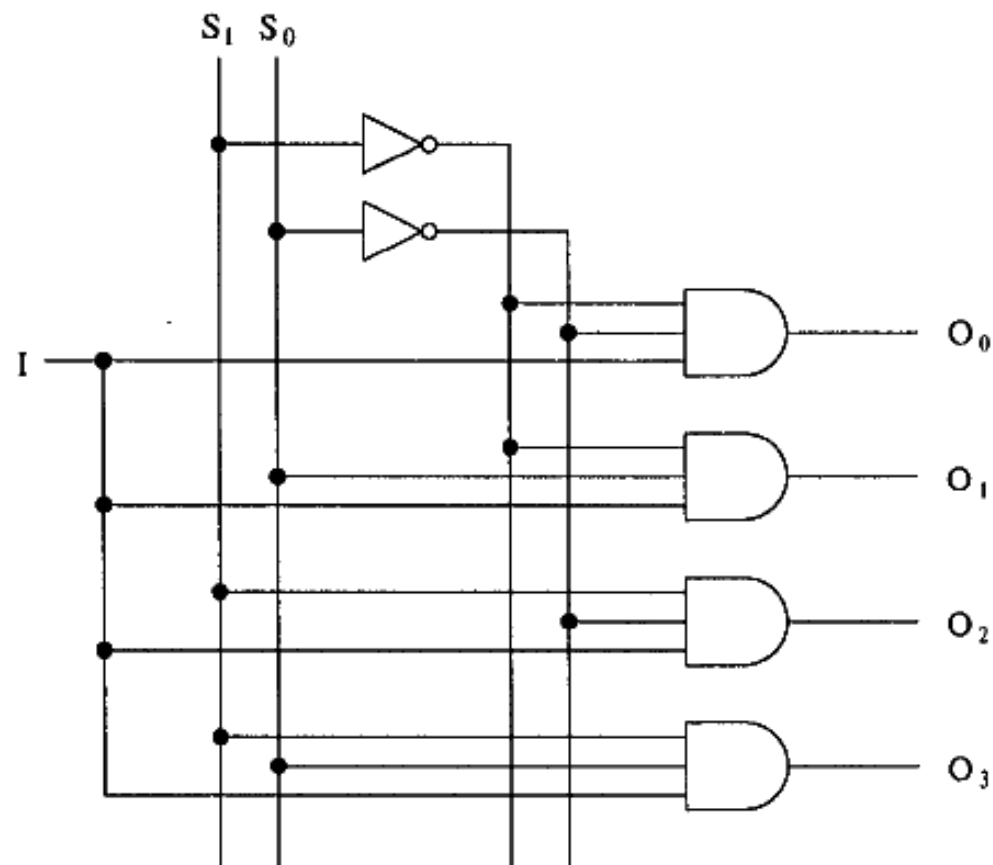
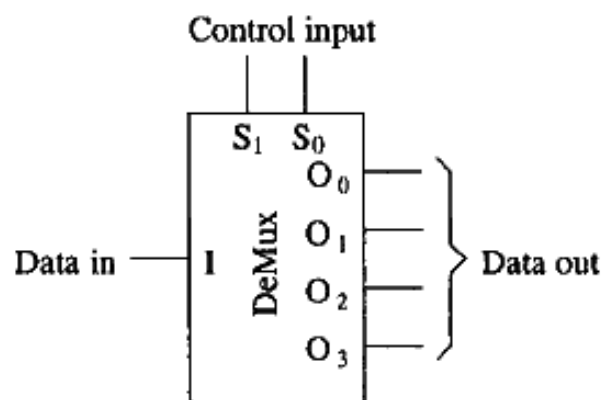


S2	S1	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3



Bộ phân kênh (Demultiplexer)

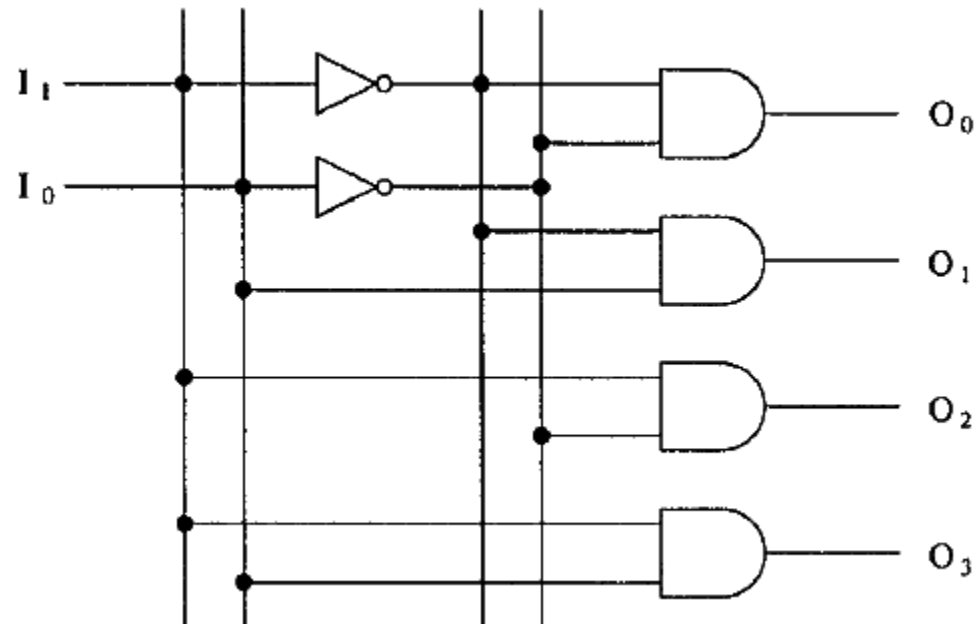
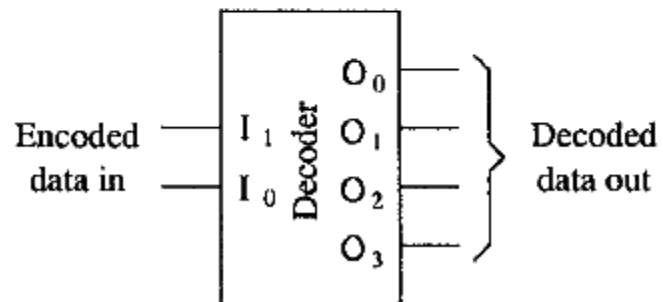
- Ngược với bộ dồn kênh
- Tín hiệu điều khiển (S) sẽ chọn đầu ra nào kết nối với đầu vào (I)
- Ví dụ: Demux 1-to-4



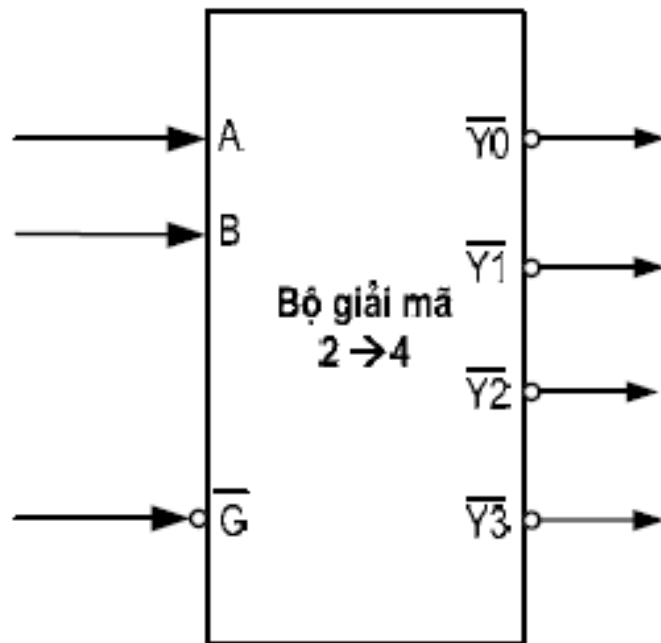
Bộ giải mã (Decoder)

- Bộ giải mã chọn một trong 2^n đầu ra (O) tương ứng với một tổ hợp của n đầu vào (I)
- Ví dụ : Mạch giải mã 2 ra 4

I_1	I_0	O_3	O_2	O_1	O_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



Bộ giải mã 2- \rightarrow 4

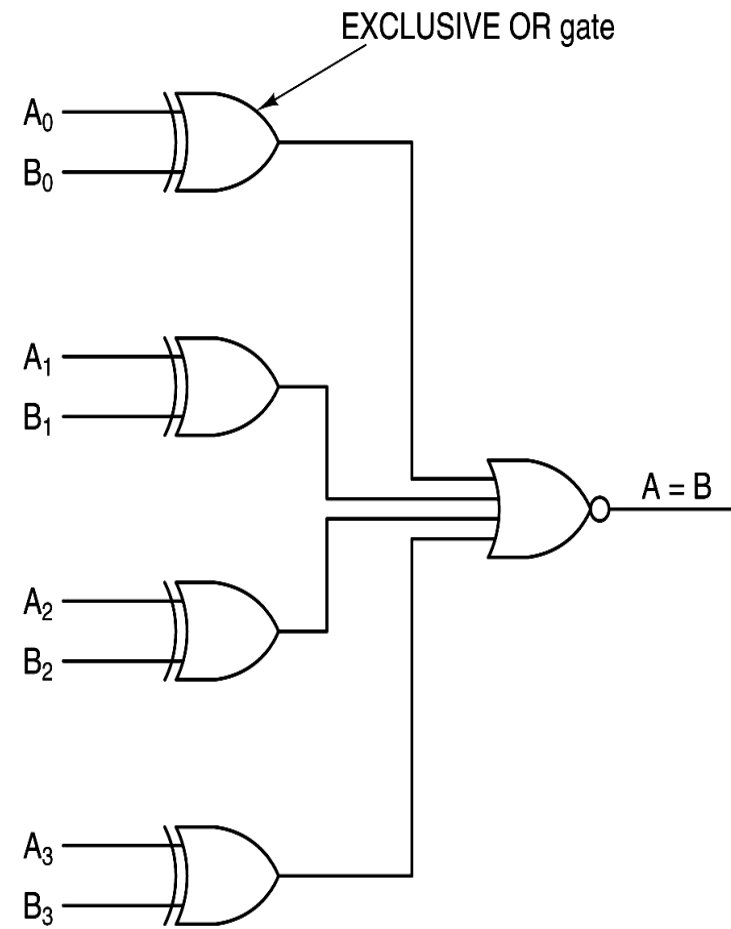


\overline{G}	B	A	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	x	x	1	1	1	1

Mạch so sánh (Comparator)

- So sánh các bit của 2 ngõ vào và xuất kết quả 1 nếu bằng nhau.
- Ví dụ : Mạch so sánh 4 bit dùng các cổng XOR

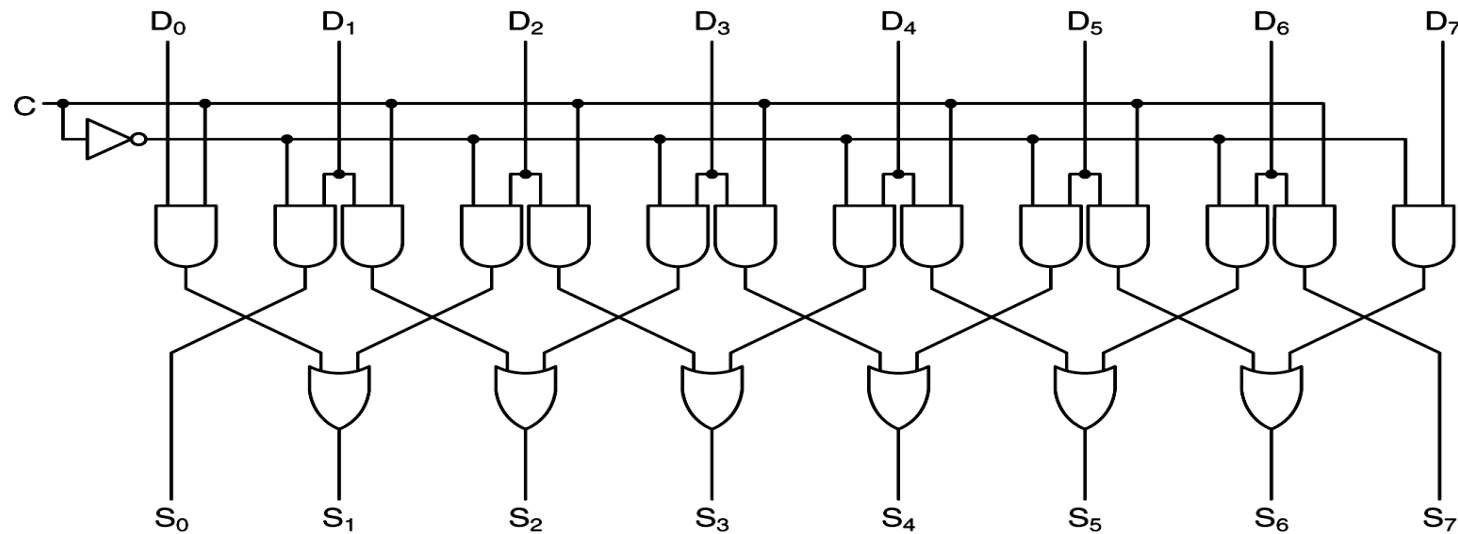
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



Mạch tính toán

➤ Mạch dịch (Shifter)

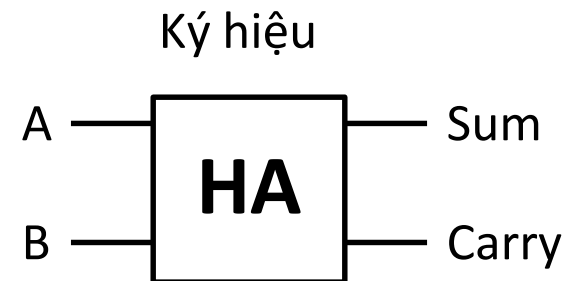
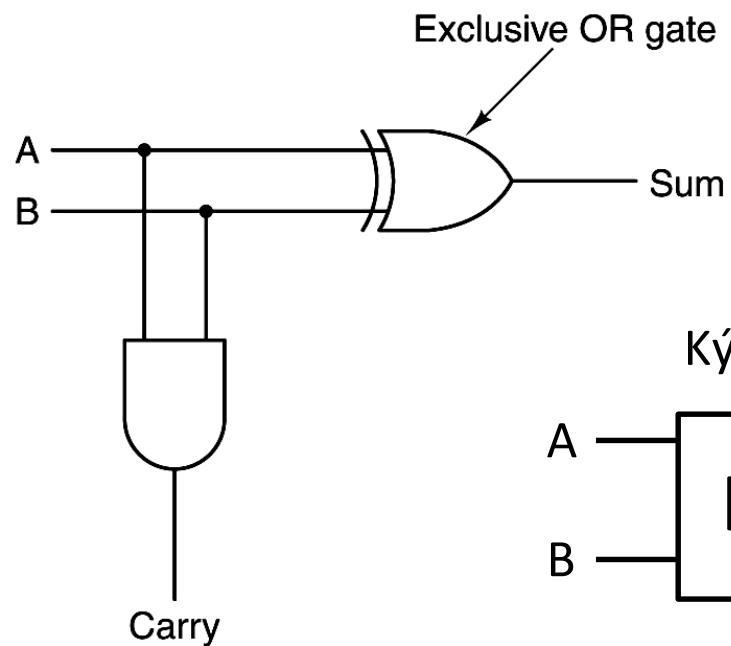
- Dịch các tín hiệu sang trái hoặc phải 1 vị trí. Ứng dụng cho phép nhân/chia cho 2.
- Ví dụ : mạch dịch 8 bit với tín hiệu điều khiển chiều dịch trái ($C=0$) hay phải ($C=1$)



➤ Mạch cộng bán phần (Half adder)

- Cộng 2 bit đầu vào thành 1 bit đầu ra và 1 bit nhớ

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

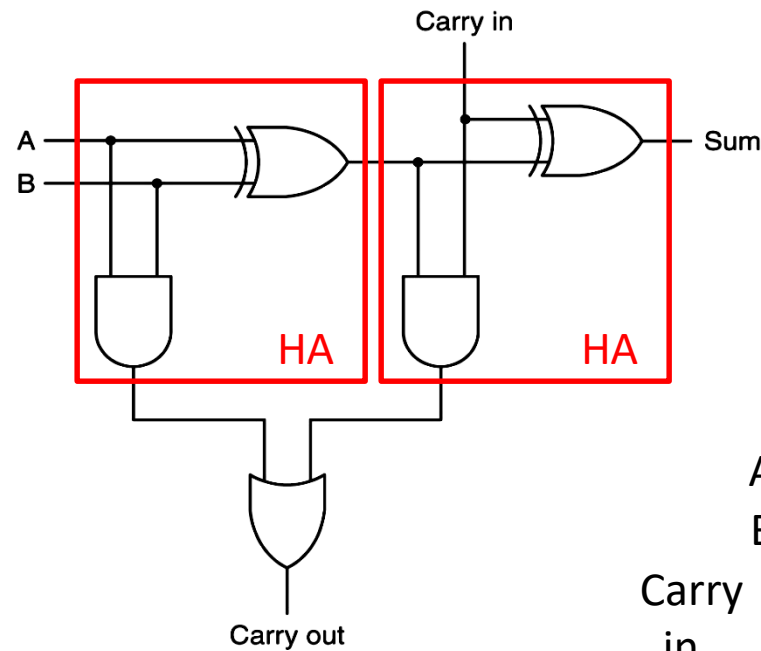


➤ Mạch cộng toàn phần (Full adder)

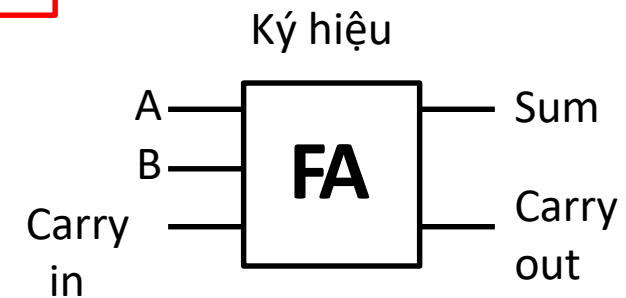
- Cộng 3 bit đầu vào thành 1 bit đầu ra và 1 bit nhớ
- Cho phép xây dựng bộ cộng nhiều bit

A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a)

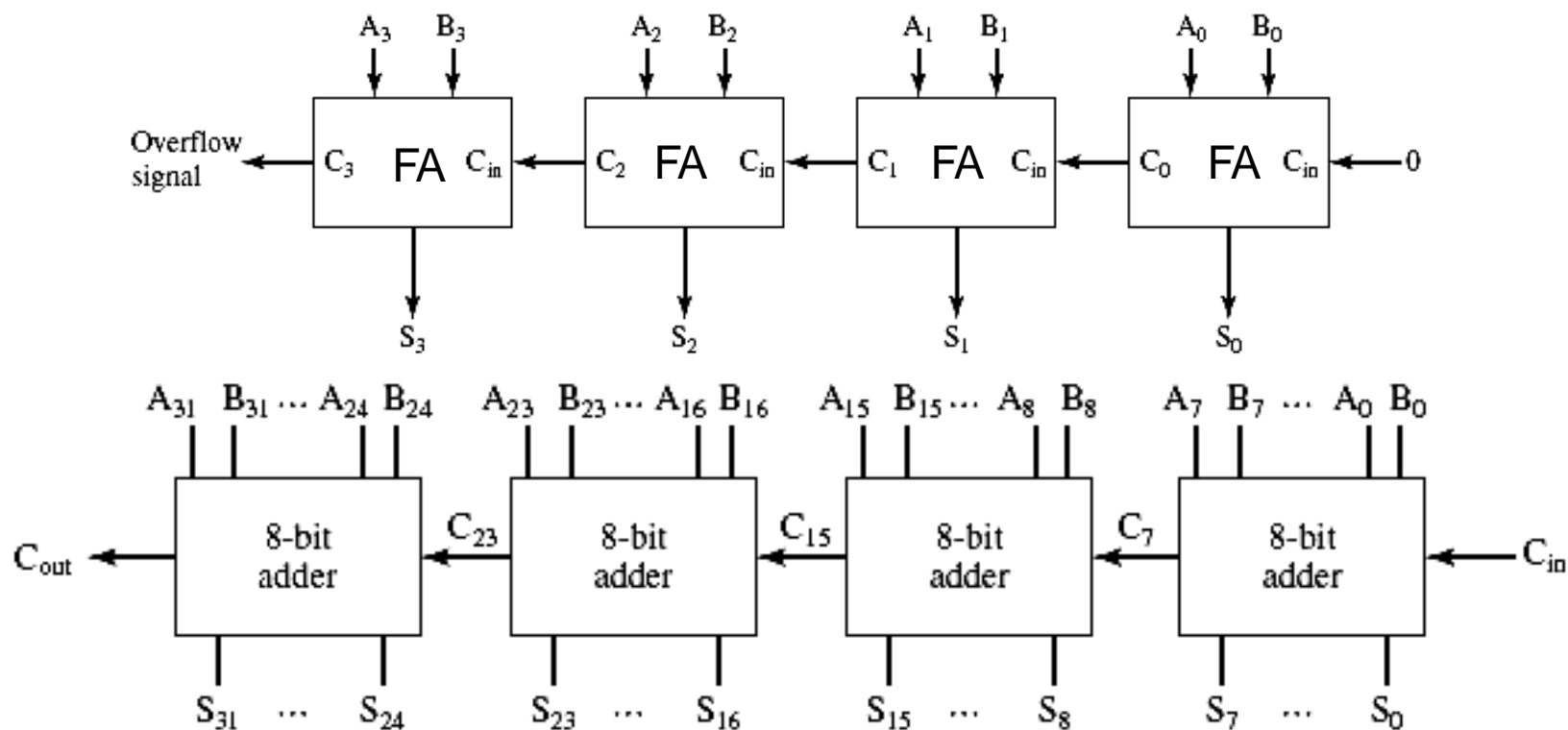


(b)



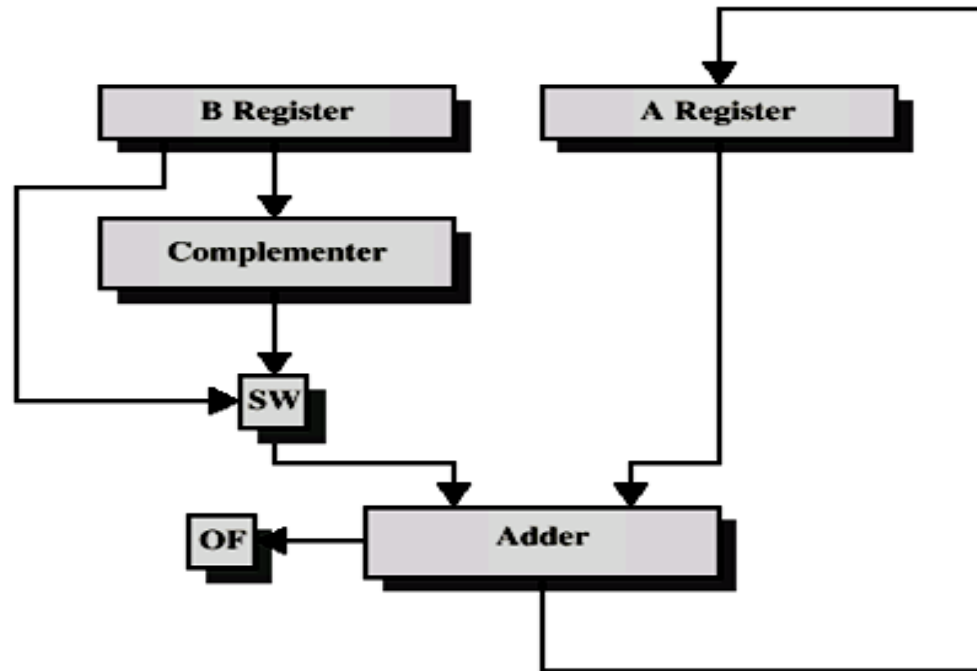
➤ Mạch cộng nhiều bit

- Ghép từ nhiều bộ cộng toàn phần



➤ Mạch cộng và trừ

- Mạch trừ: Đổi sang số bù 2 rồi cộng
- Có thể dùng chung mạch cộng để thực hiện phép trừ



OF = overflow bit
SW = Switch (select addition or subtraction)

- Ví dụ ALU 1 bit

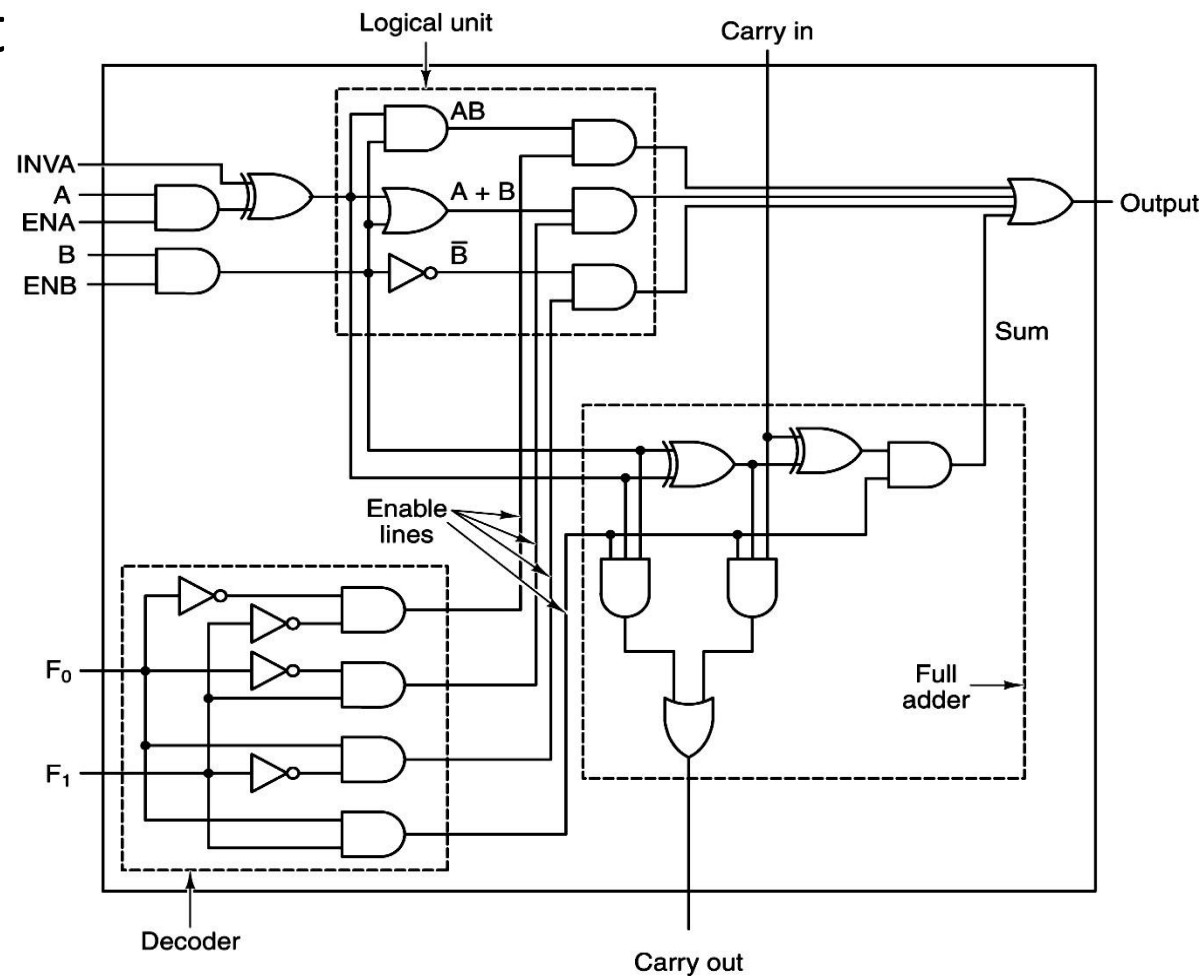
F_0F_1	Functions
00	A AND B
01	A OR B
10	
11	A + B

Điều kiện
bình thường

ENA=1

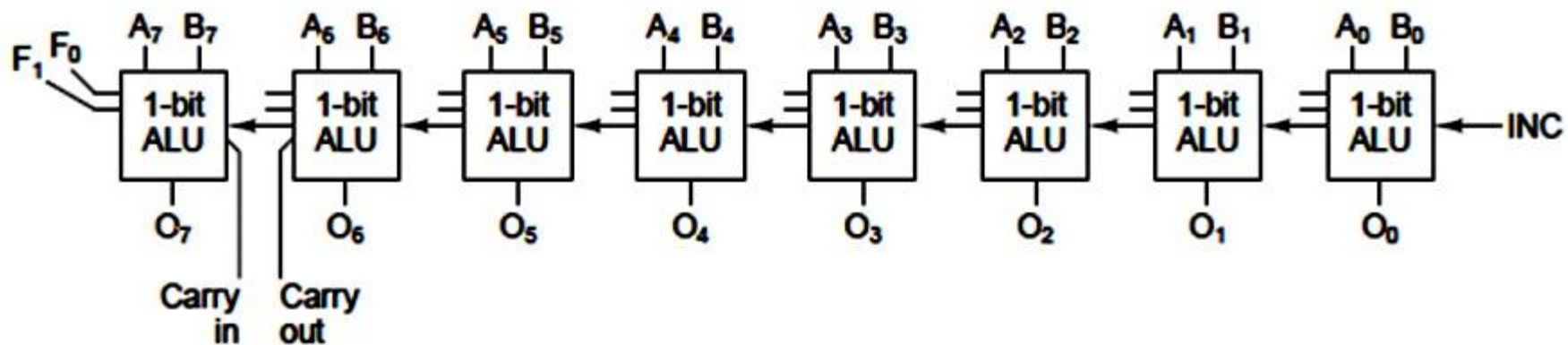
ENB=1

INVA=0



➤ ALU 8 bit

- Ví dụ tạo 1 mạch ALU 8 bit bằng cách ghép 8 bộ ALU 1 bit ở ví dụ trước



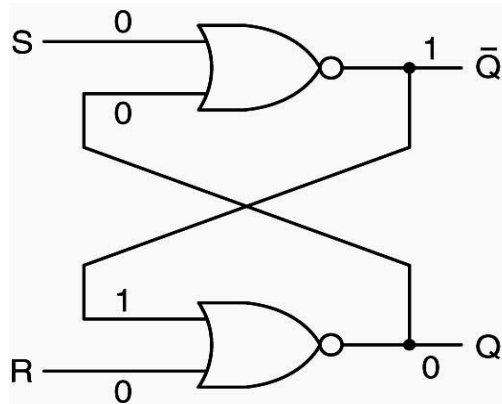
Mạch tuần tự

➤ Khái niệm

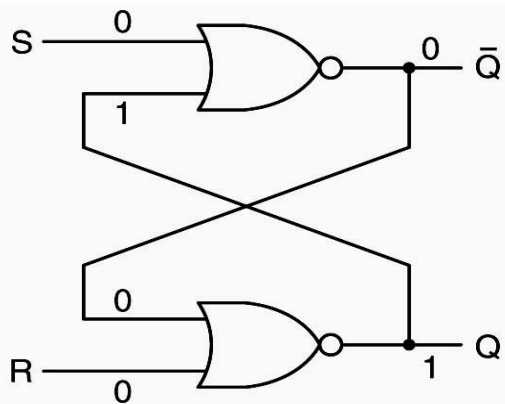
- Mạch tuần tự (sequential circuit) là mạch logic trong đó tín hiệu ra phụ thuộc tín hiệu vào ở hiện tại và quá khứ
- Là mạch có nhớ, được thực hiện bằng phần tử nhớ (Latch, Flip-Flop) và có thể kết hợp với các cổng logic cơ bản
- Ứng dụng làm bộ nhớ, thanh ghi, mạch đếm,... trong máy tính

➤ Mạch chốt (Latch)

- Dùng 2 cổng NOR mắc hồi tiếp với nhau. S, R là ngõ vào, Q và \bar{Q} là ngõ ra.
- Đây là mạch chốt SR. Nó có thể ở 1 trong 2 trạng thái $Q=1$ hoặc $Q=0$ khi $S=R=0$.
- Khi $S=1 \rightarrow Q=1$ bất kể trạng thái trước đó (set)
- Khi $R=1 \rightarrow Q=0$ bất kể trạng thái trước đó (reset)



(a)



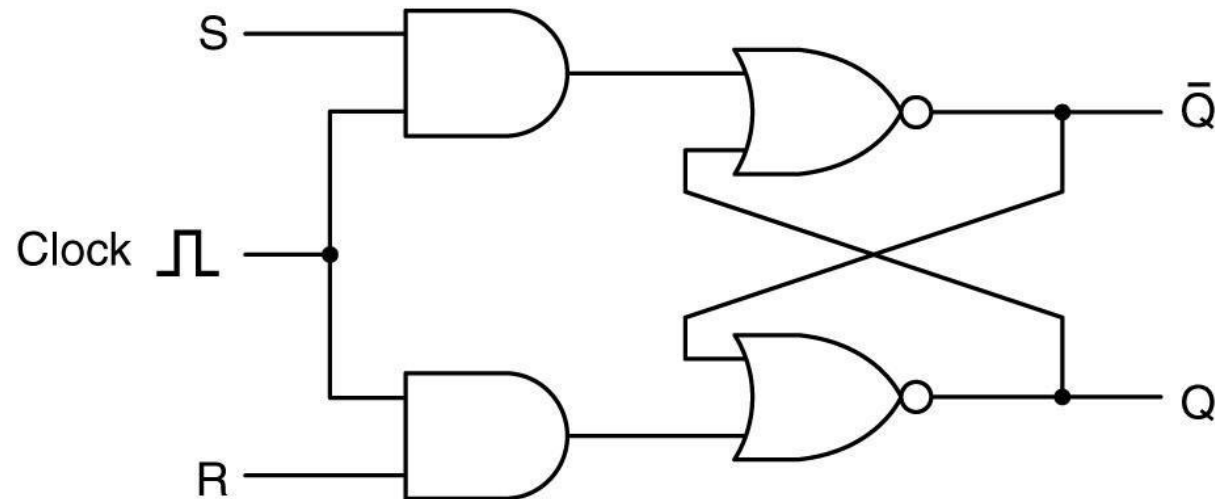
(b)

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

(c)

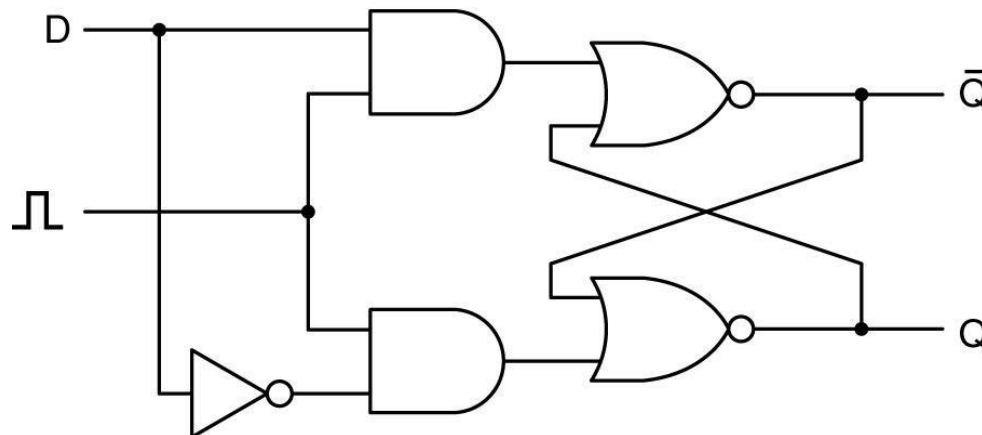
➤ Mạch chốt SR có xung Clock

- Thêm vào mạch chốt SR 2 cổng AND nối với xung đồng hồ để điều khiển trạng thái mạch chốt tại thời điểm xác định
- Tín hiệu vào chỉ có tác dụng khi xung clock=1 (mức cao)



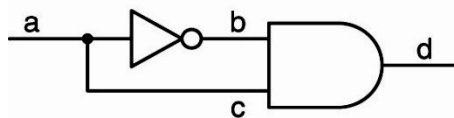
➤ Mạch chốt D có xung Clock

- Mạch chốt SR sẽ ở trạng thái không xác định khi $S=R=1$
- Khắc phục bằng cách chỉ dùng 1 tín hiệu vào và đấu nối R với S qua cổng NOT
- Đây chính là mạch bộ nhớ 1 bit với D là ngõ vào, Q là ngõ ra

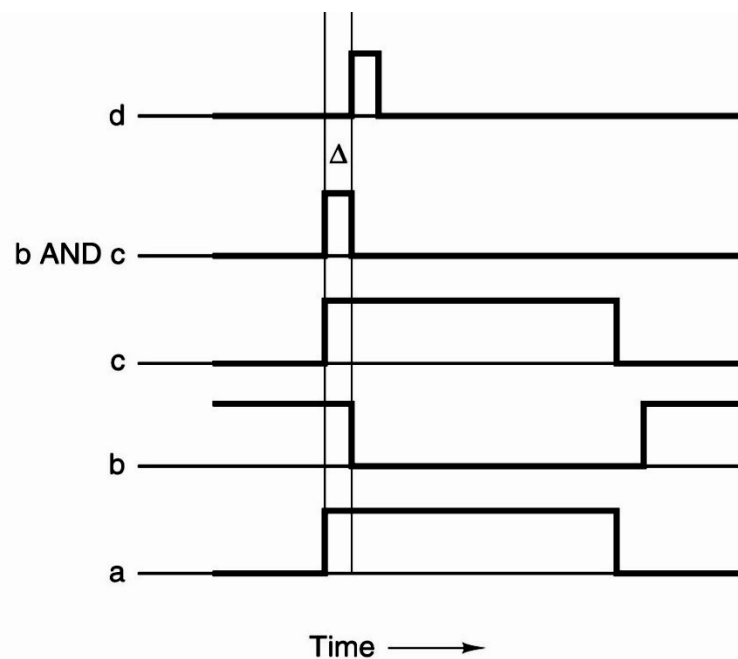


➤ Flip-Flop

- Trong thực tế ta muốn bộ nhớ chỉ được ghi trong 1 khoảng thời gian nhất định → cần thiết kế mạch xung Clock tác dụng theo cạnh (lên hoặc xuống)



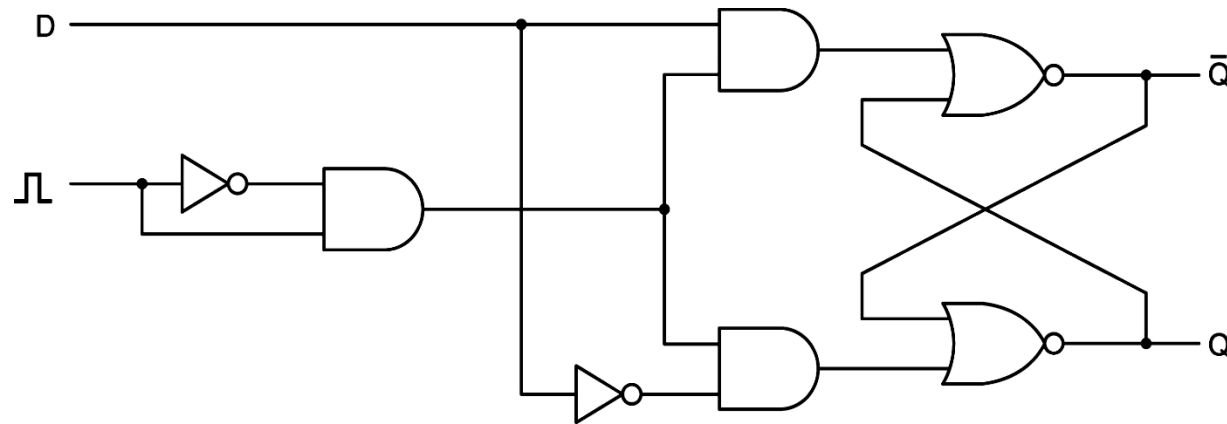
(a)



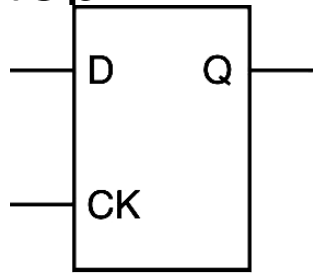
(b)

➤ D Flip-Flop

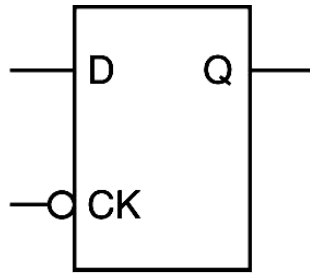
- Là mạch chốt D có xung Clock điều khiển bằng Flip-flop
- Phân biệt:
 - Flip-flop: edge triggered
 - Latch: level triggered



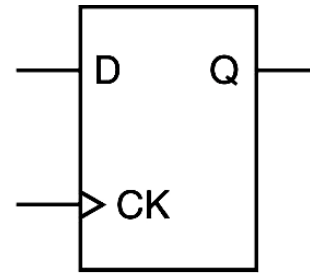
➤ Ký hiệu mạch chốt và Flip-Flop



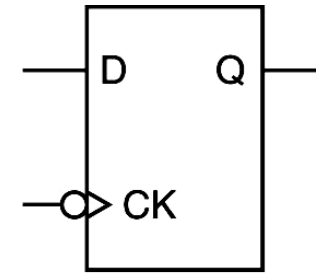
(a)



(b)



(c)

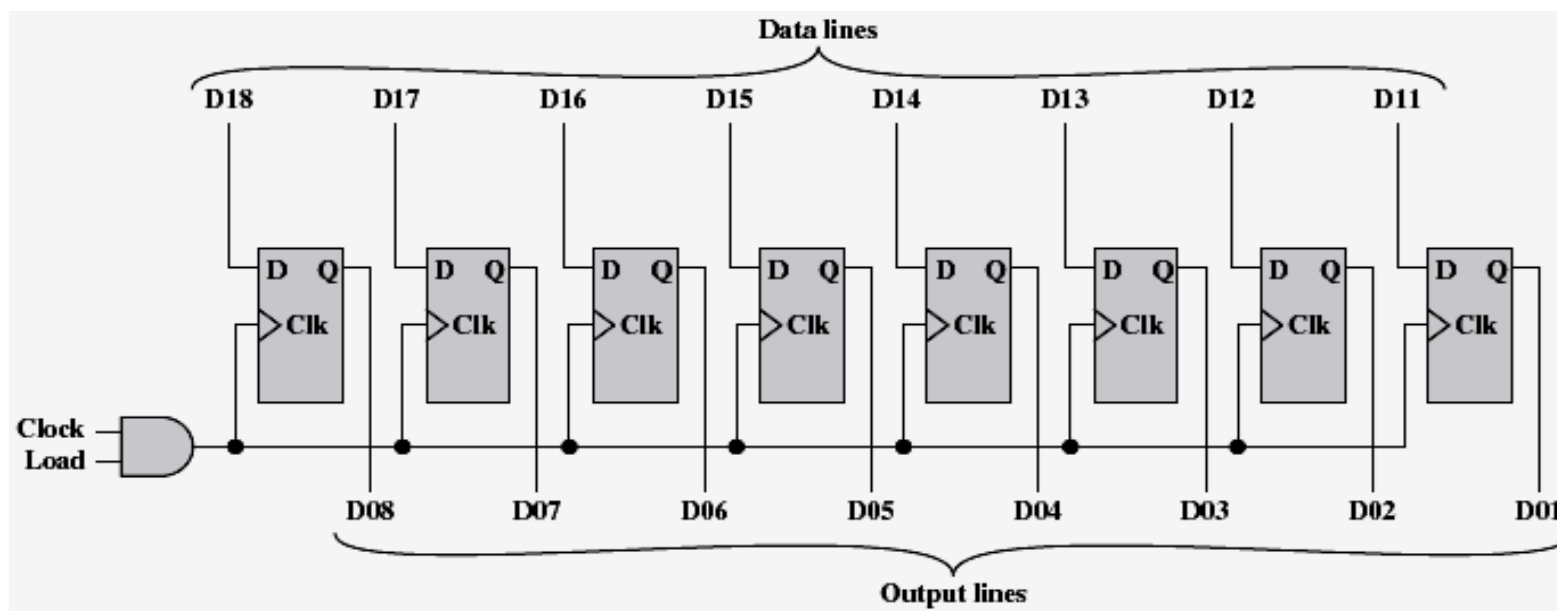
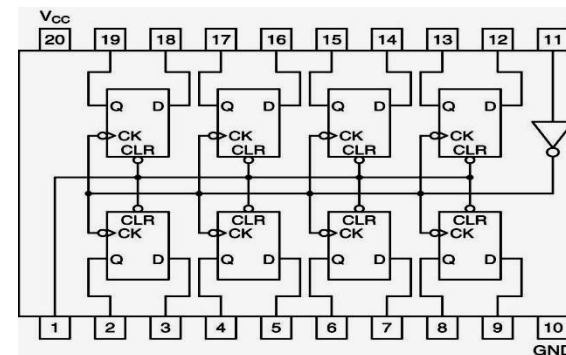


(d)

- a) Mạch chốt D tác động theo mức 1 (clock=1)
- b) Mạch chốt D tác động theo mức 0 (clock=0)
- c) Flip-flop D tác động theo cạnh lên (clock= $0 \rightarrow 1$)
- d) Flip-flop D tác động theo cạnh xuống (clock= $1 \rightarrow 0$)

➤ Thanh ghi (Register)

- Việc ghép nối nhiều ô nhớ 1 bit tạo thành các ô nhớ lớn hơn
- Ví dụ : Vi mạch 74273 gồm 8 D flip-flop ghép nối lại tạo thành 1 thanh ghi 8 bit



➤ Ví dụ : mạch bộ nhớ 4 ô x 3 bit

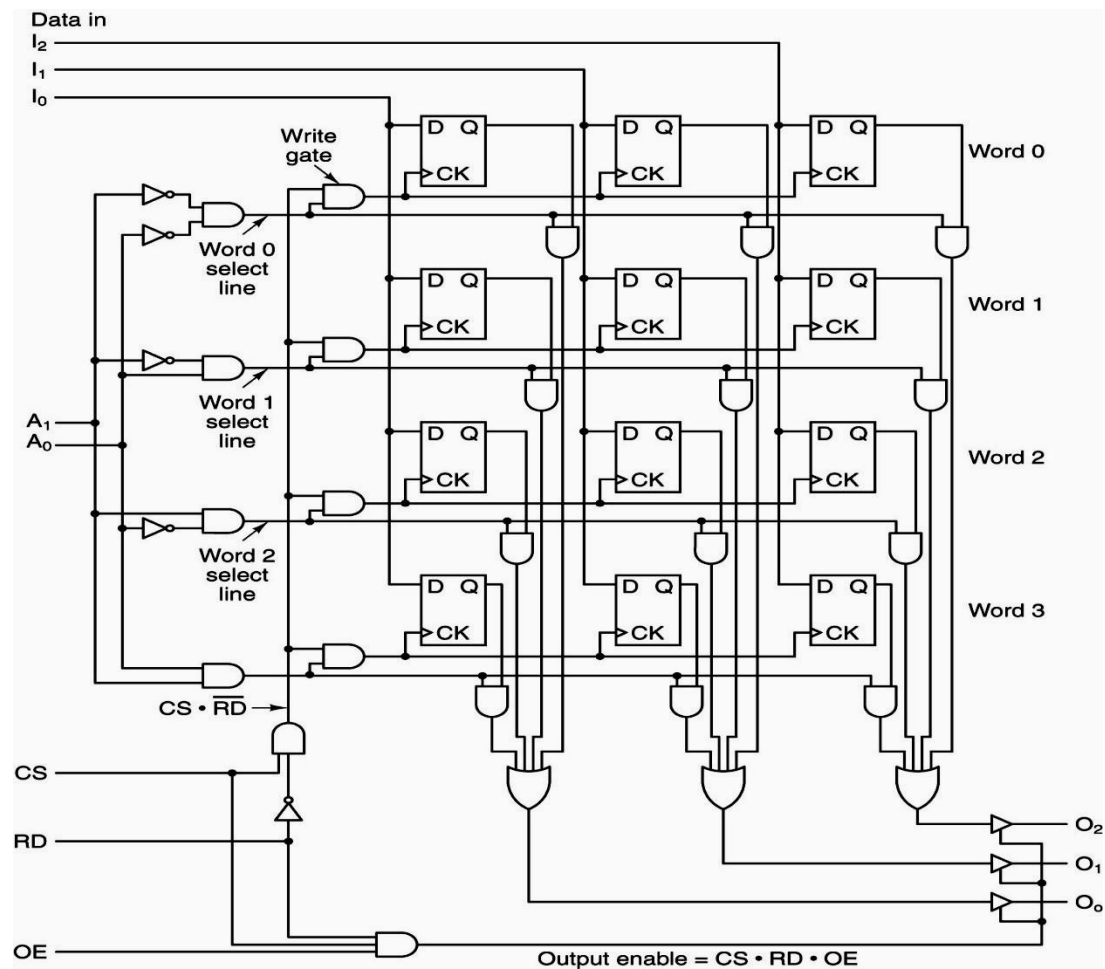
- A: Address
- I: Input data
- O: Output data
- CS: Chip select
- RD: Read/write
- OE: Output enable

Write:

CS=1, RD=0, OE=0

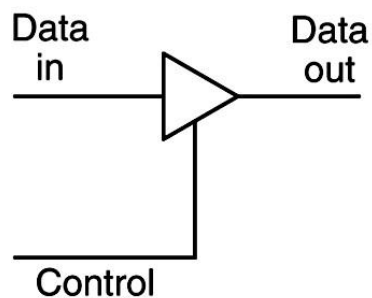
Read:

CS=1, RD=1, OE=1



➤ Mạch đệm (Buffer)

- Dùng để đọc dữ liệu đồng bộ trên nhiều đường tín hiệu bằng 1 đường điều khiển riêng.
- Sử dụng các cổng 3 trạng thái (tri-state devices)



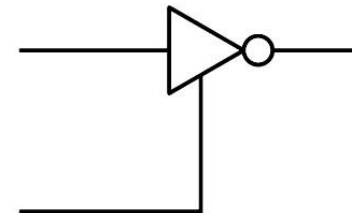
(a)



(b)



(c)

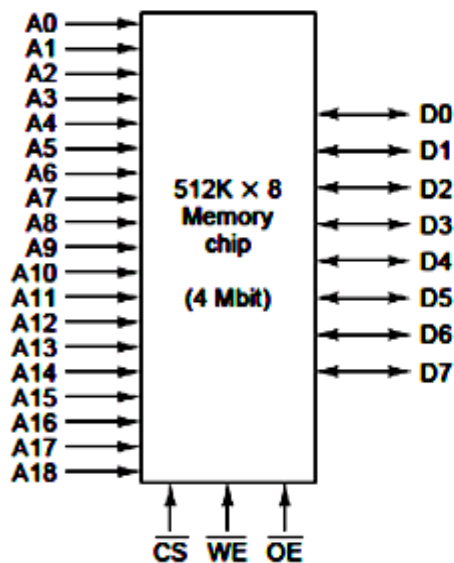


(d)

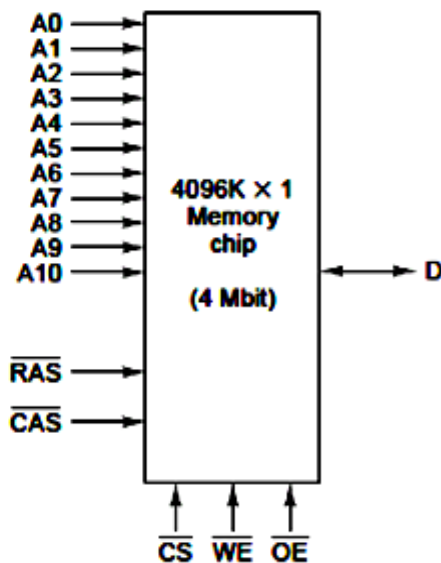
- a. Buffer không đảo.
- b. Khi control ở mức cao (=1).
- c. Khi control ở mức thấp (=0).
- d. Buffer đảo.

➤ Chip bộ nhớ

- Bộ nhớ thường gồm nhiều ô nhớ ghép lại
- Ví dụ 1: Chip bộ nhớ 4Mbit có thể tạo thành từ 512K ô 8 bit hoặc ma trận 2048x2048 ô 1 bit



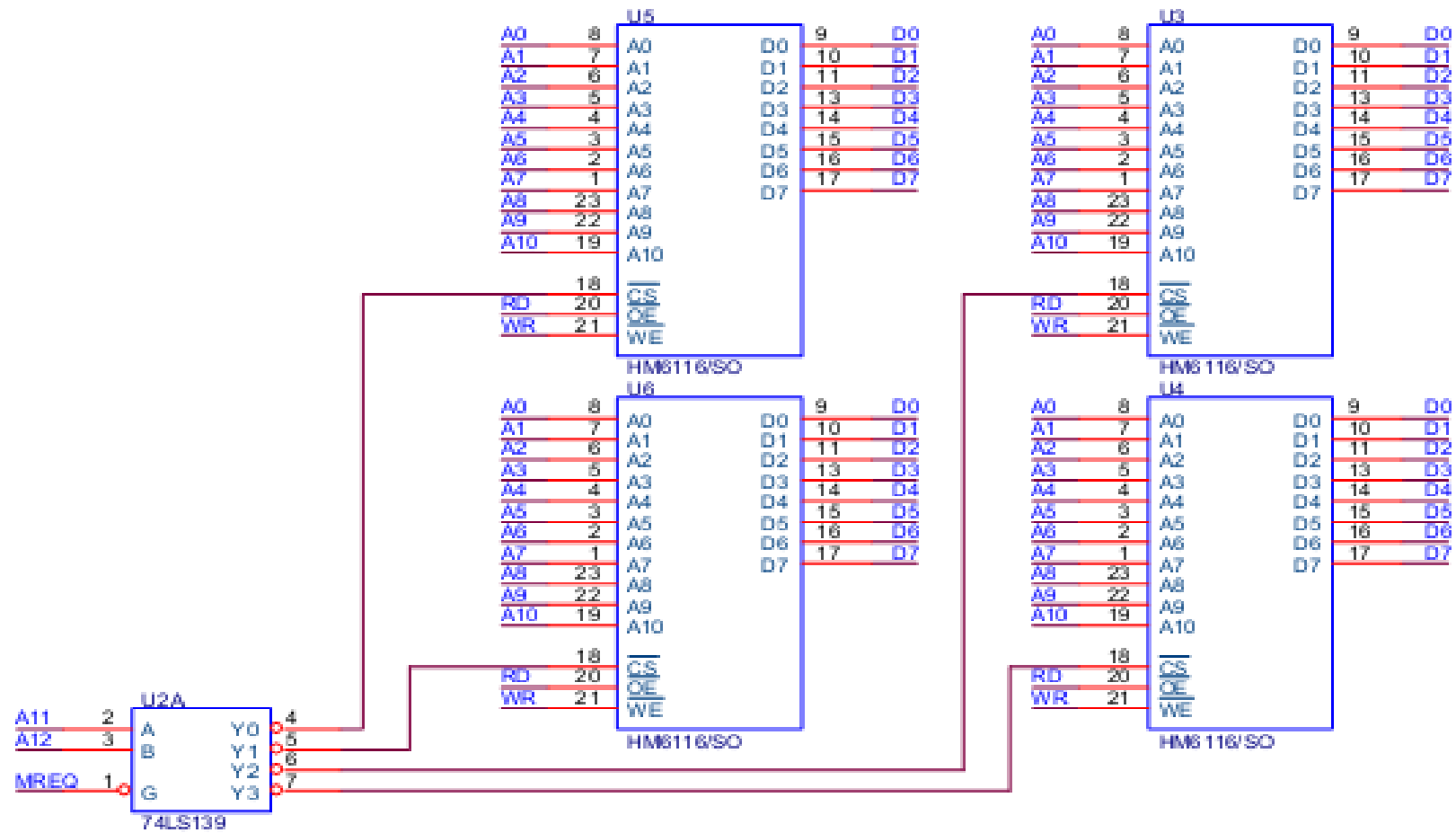
(a)



(b)

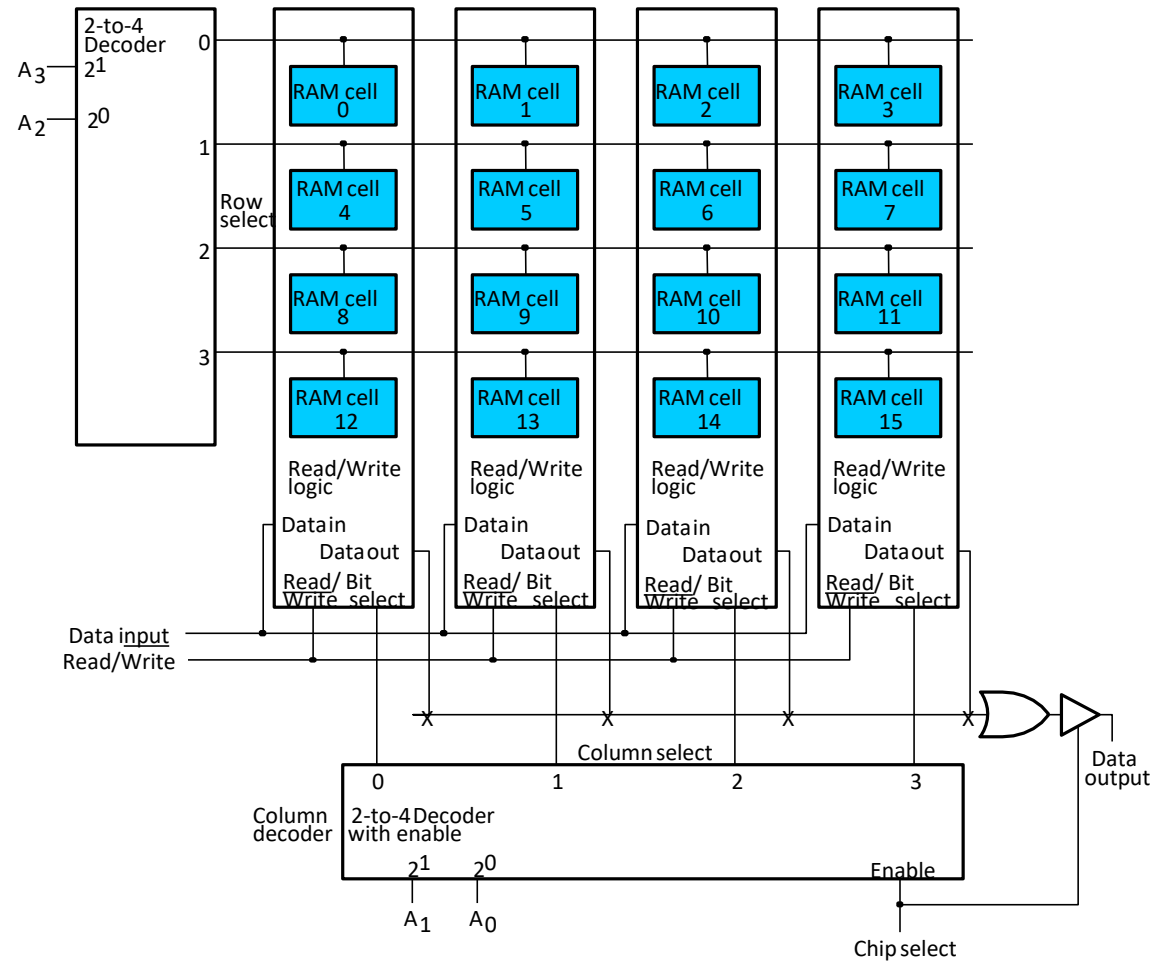
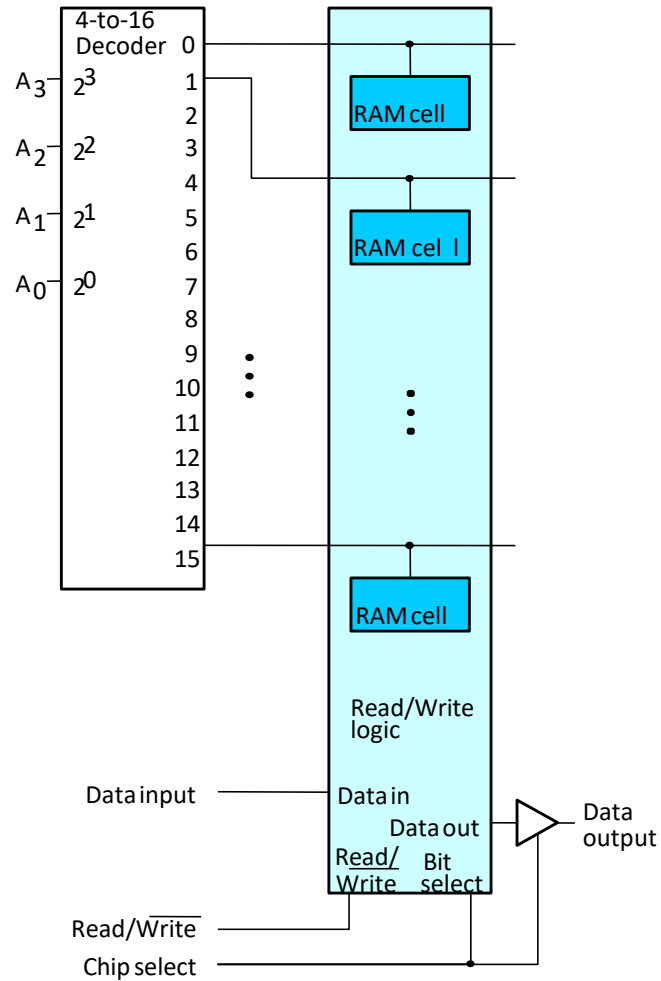
Ghi chú:

RAS: Row Address Strobe
CAS: Column Address Strobe
CS: Chip select
WE: Write enable
OE: Output enable
D: Data
A: Address



➤ Chip bộ nhớ (tiếp)

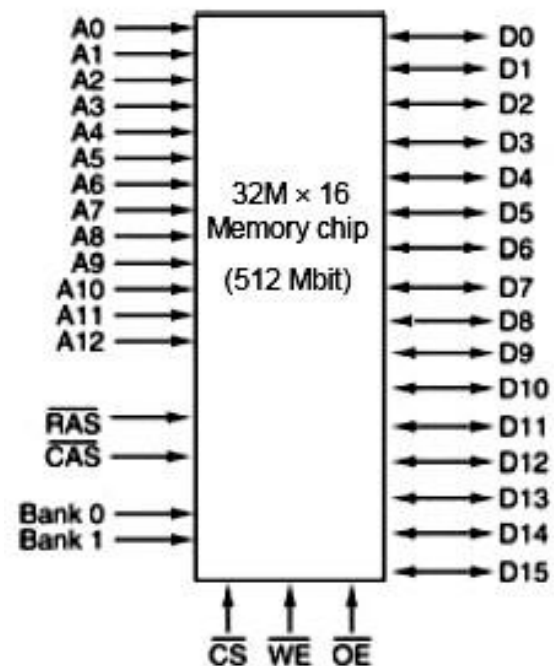
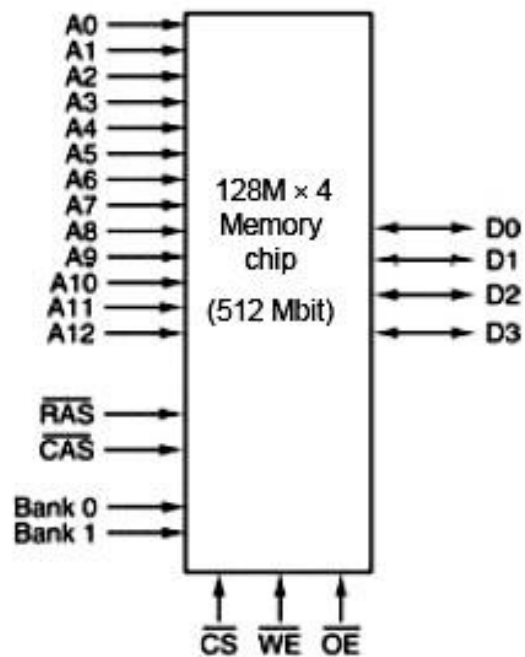
- Mạch giải mã địa chỉ n bit có thể giải mã cho 2^n ô nhớ
→ cần n chân tín hiệu địa chỉ
- Có thể giảm kích thước bộ giải mã còn bằng cách tổ chức thành ma trận các ô nhớ → sử dụng 2 bộ giải mã cho hàng và cột riêng
- Ví dụ: bộ nhớ 16 ô cần 4 bit địa chỉ có thể tổ chức thành ma trận 4×4 → chỉ cần giải mã 2 bit cho hàng và 2 bit cho cột.
- Có thể ghép địa chỉ hàng và cột chung 1 chân tín hiệu → giảm số chân kết nối bus địa chỉ
- Nhược điểm: cần gấp đôi thời gian truy cập bộ nhớ



➤ Chip bộ nhớ (tiếp)

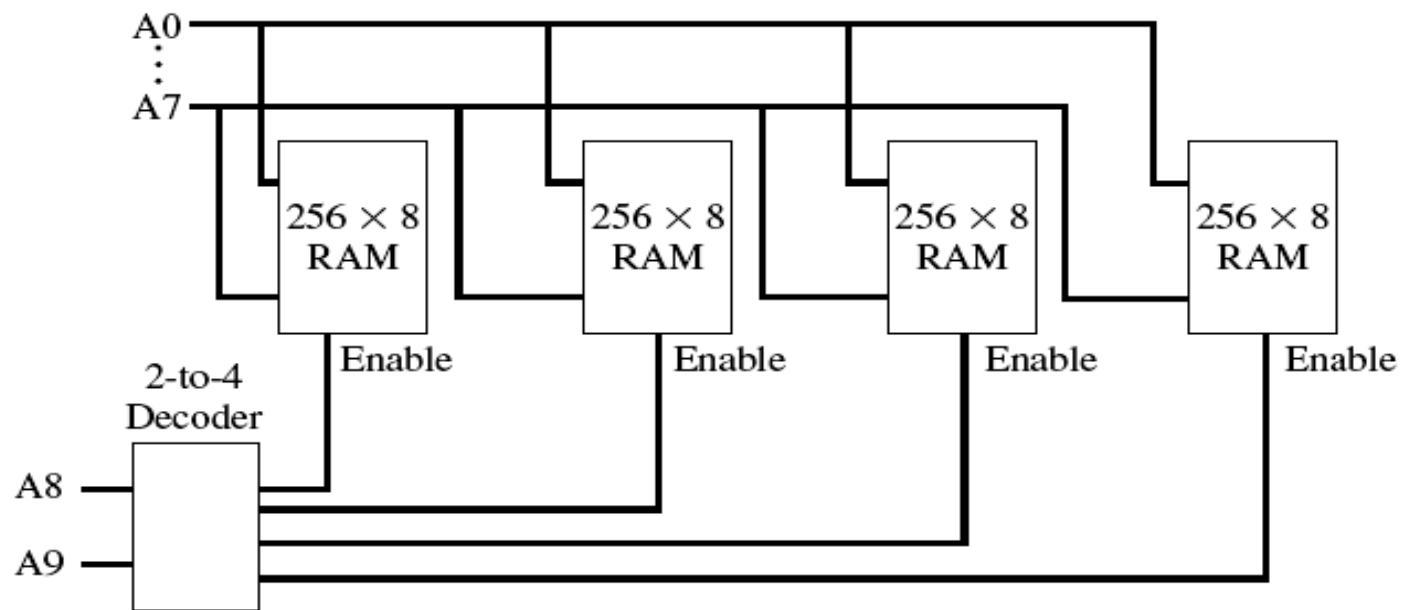
○ Ví dụ 2: Chip bộ nhớ 512Mbit = 4 bank 128Mbit

- Ma trận 13 bit hàng * 12 bit cột * ô nhớ 4 bit
- Ma trận 13 bit hàng * 10 bit cột * ô nhớ 16 bit



➤ Tổ chức bộ nhớ

- Bộ nhớ thường gồm nhiều chip nhớ dung lượng nhỏ ghép lại
- Dùng 1 mạch giải mã địa chỉ để chọn chip khi truy cập
- Ví dụ: Bộ nhớ 1KB gồm 4 chip 256B ghép lại



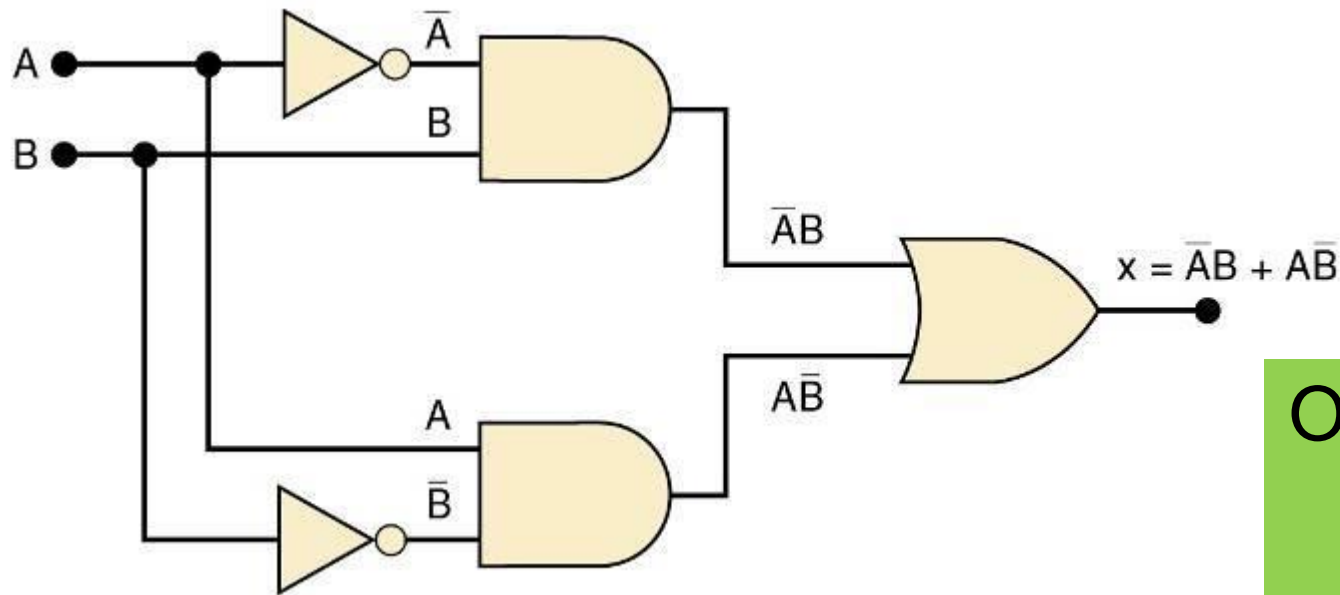
65

XOR & XNOR Gate

Exclusive OR

66

- Exclusive OR (XOR) cho ra kết quả HIGH khi hai đầu vào khác nhau

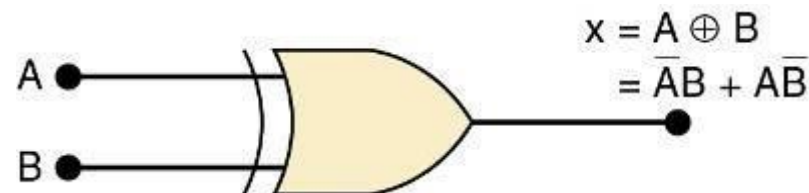


A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

Output expression:

$$x = \bar{A}B + A\bar{B}$$

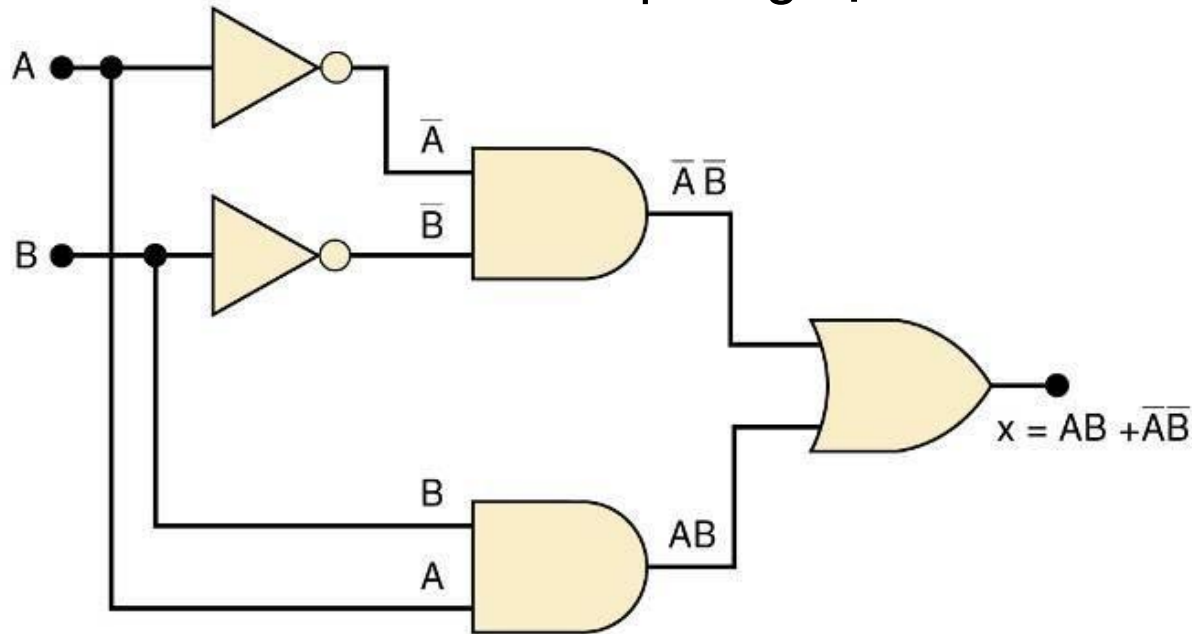
Ký hiệu XOR Gate



Exclusive NOR

67

- Exclusive NOR (XNOR) cho ra kết quả HIGH khi hai đầu vào giống nhau
 - ▣ XOR và XNOR cho ra kết quả ngược nhau

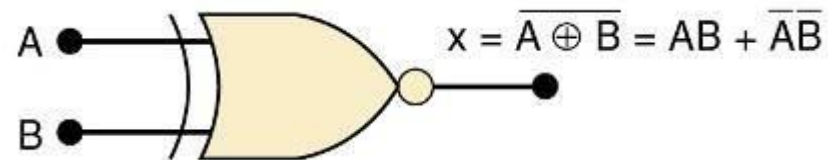


A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

Output

$$x = AB + \bar{A}\bar{B}$$

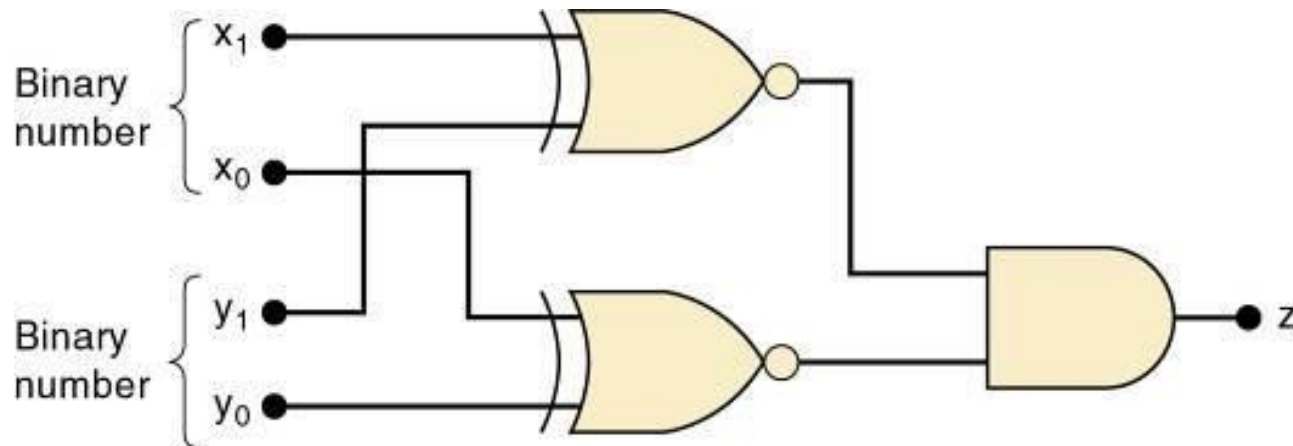
Ký hiệu XNOR



Ví dụ

68

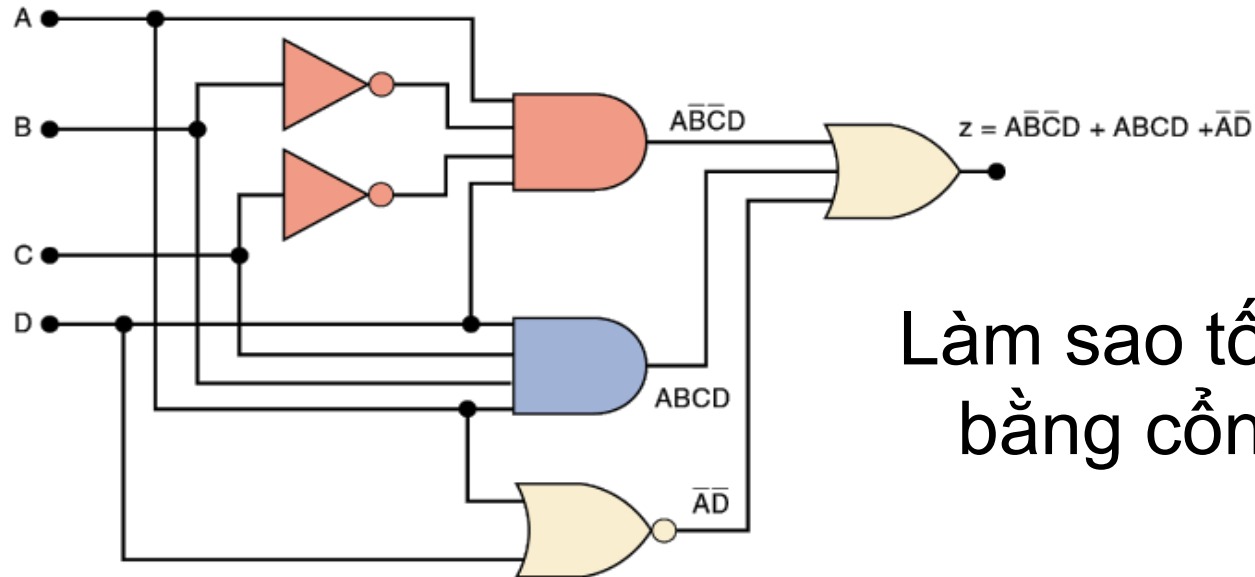
- Thiết kế một mạch để phát hiện ra 2 số nhị phân 2 bit có bằng nhau hay không



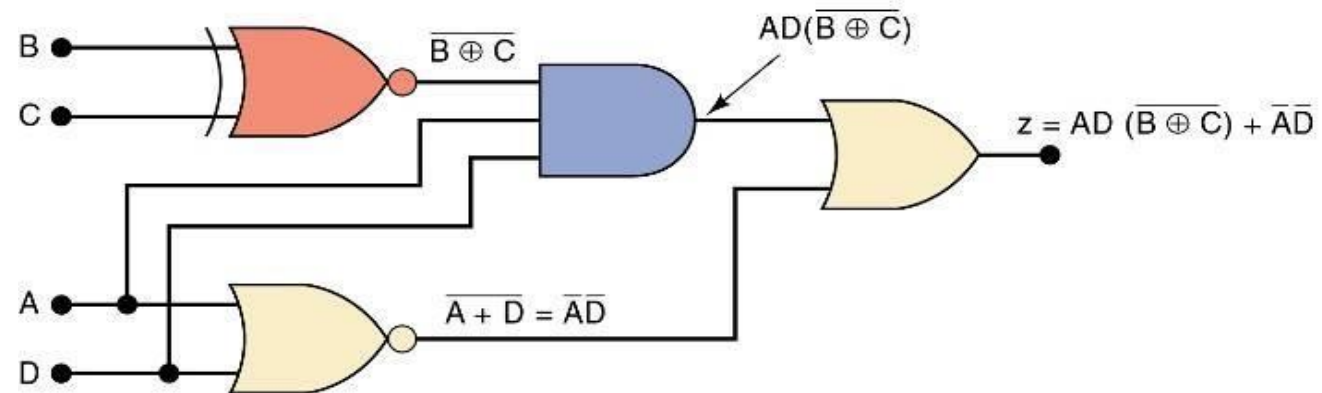
x_1	x_0	y_1	y_0	z (Output)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

XOR & XNOR

69



Làm sao tối ưu mạch
bằng cổng **XNOR**



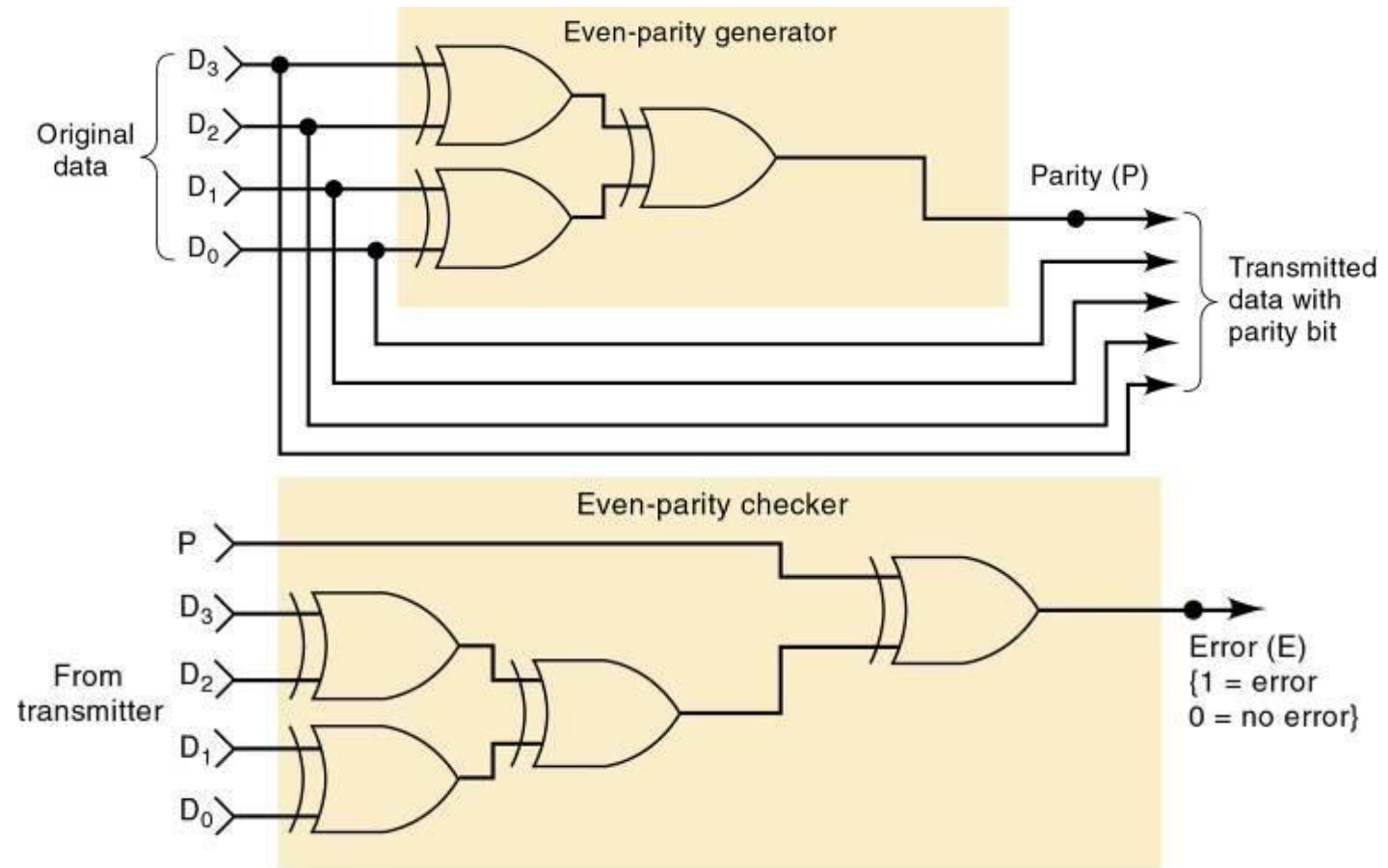
70

Một số mạch logic

Parity generator and checker

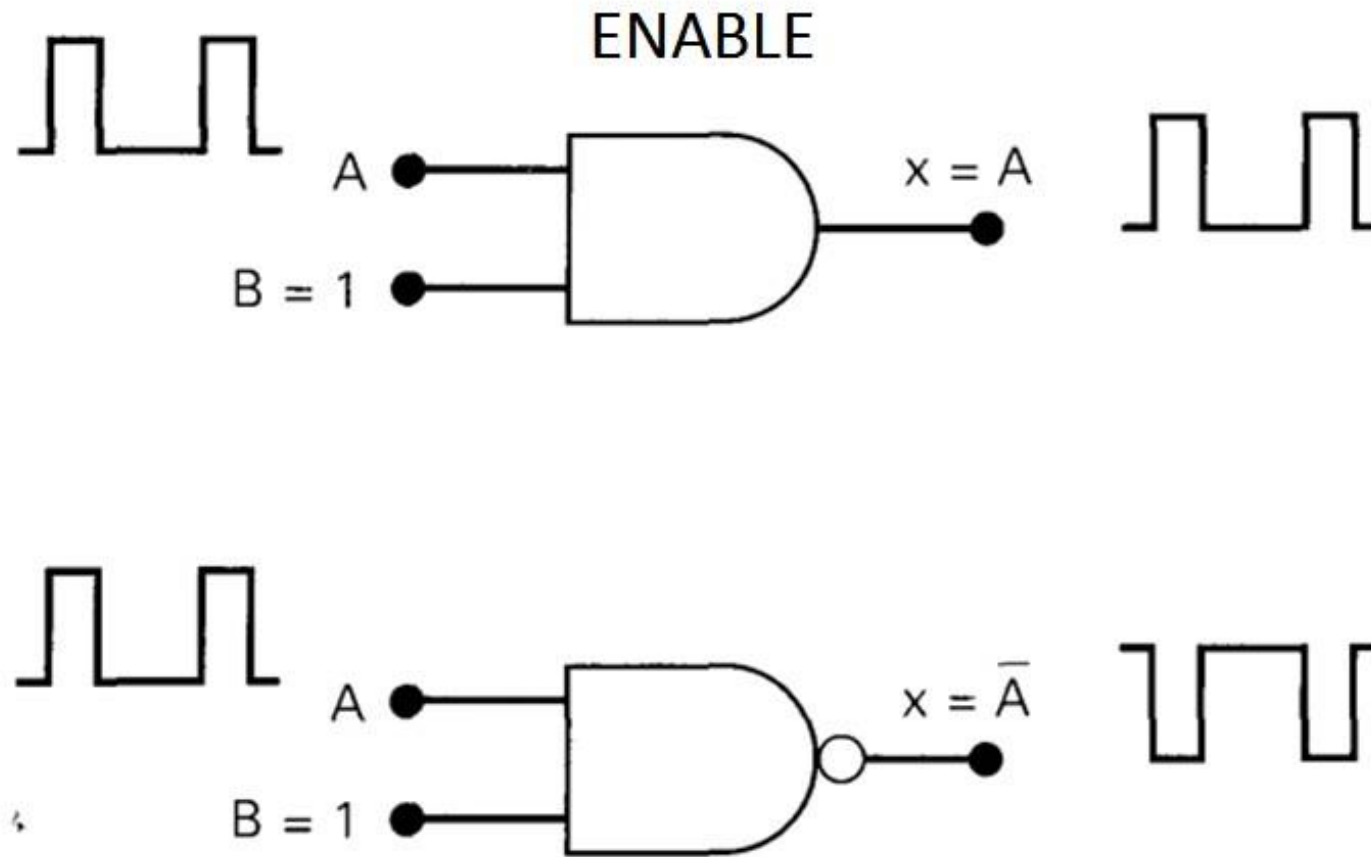
71

- Cổng XOR và XNOR rất hữu dụng trong các mạch với mục đích phát hiện và kiểm tra parity



Mạch enable

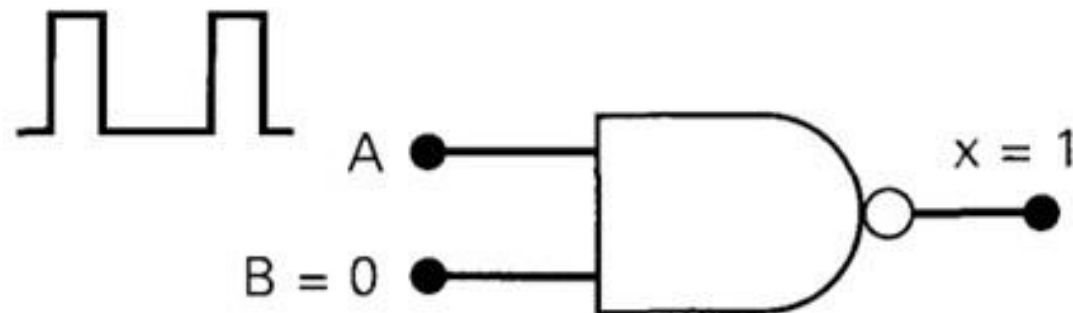
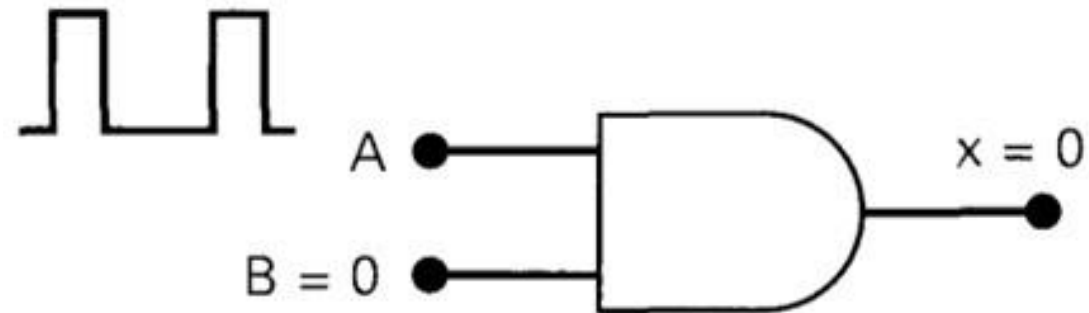
72



Mạch disable

73

DISABLE



Ví dụ

74

- Thiết kế mạch tổ hợp cho phép 1 tín hiệu truyền đến ngõ xuất khi một trong 2 tín hiệu điều khiển ở **mức 1** (không đồng thời). Các trường hợp khác ngõ xuất ở **mức 1** (HIGH).

