

Ban học tập
Khoa Kỹ thuật máy tính

Training cuối học kỳ II năm học 2021-2022

NHẬP MÔN MẠCH SỐ



Email : bht.ktmt@gmail.com

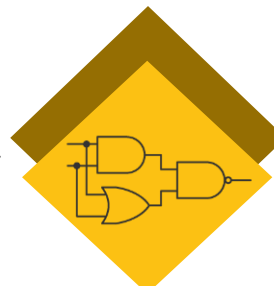
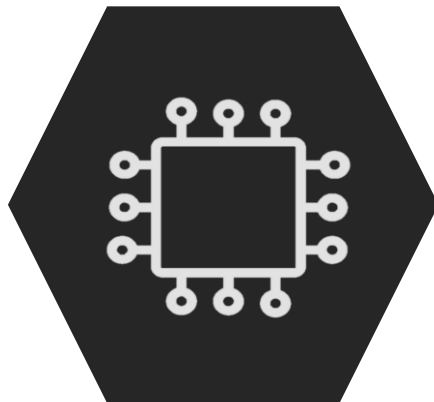
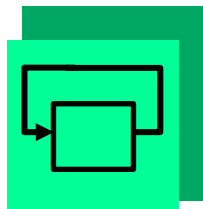


Fanpage : www.facebook.com/bht.ktmt



BAN HỌC TẬP
KHOA KỸ THUẬT MÁY TÍNH

Mạch tuần tự



Mạch tổ hợp

MỤC LỤC

I. MẠCH TỔ HỢP



Mạch
cộng



Mạch
trừ




Decoder


NỘI DUNG



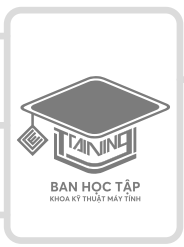
Mux



Biểu thức
Shannon



Mạch so
sánh



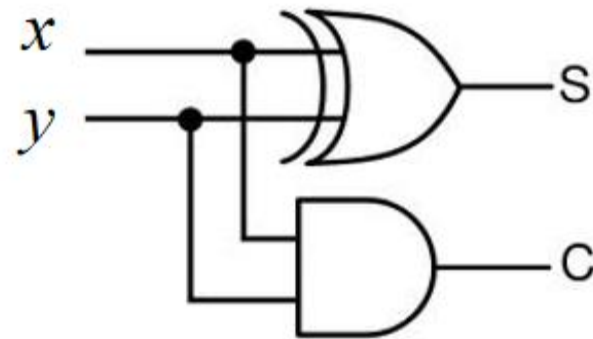
1. Mạch cộng (ADDER)

a. Mạch cộng bán phần

Là **mạch** tổ hợp số học **cộng** hai số tạo ra một bit tổng (S) và bit nhớ (C) làm đầu ra.

Biểu thức tính: $S = x \oplus y$
 $\rightarrow c = x.y$

Sơ đồ mạch:



Bảng chân trị:

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



1. Mạch cộng (ADDER)

b. Mạch cộng toàn phần

Bảng chân trị

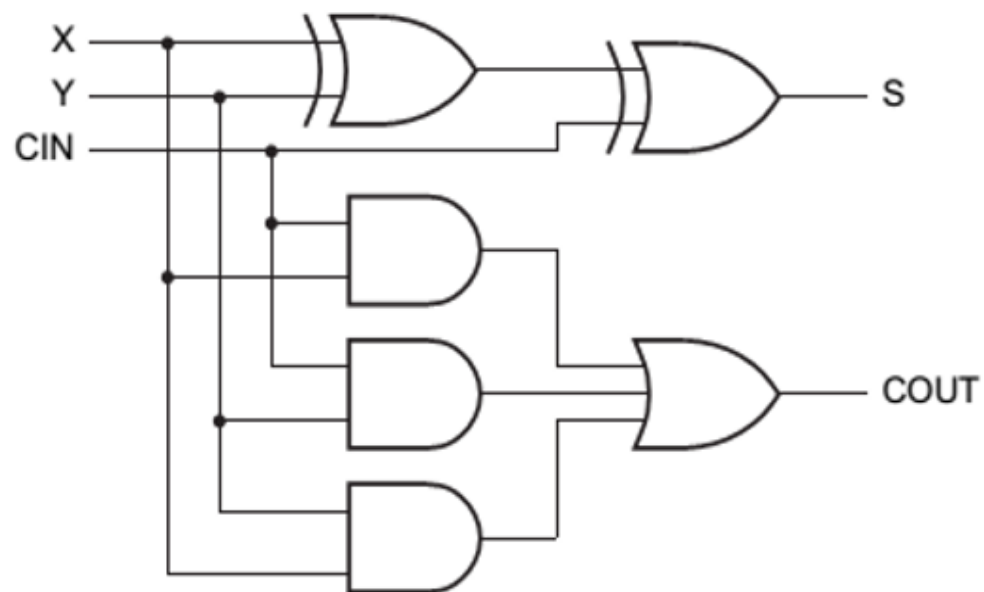
X	Y	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

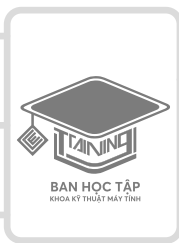
Biểu thức tính:

$$S = x \oplus y \oplus cin$$

$$C_{out} = x.y + x.c_{in} + y.c_{in}$$

Sơ đồ mạch:





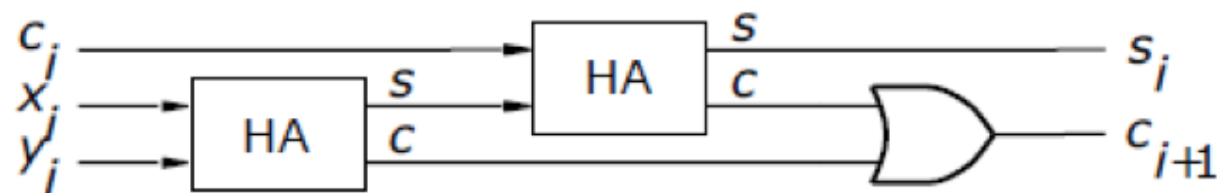
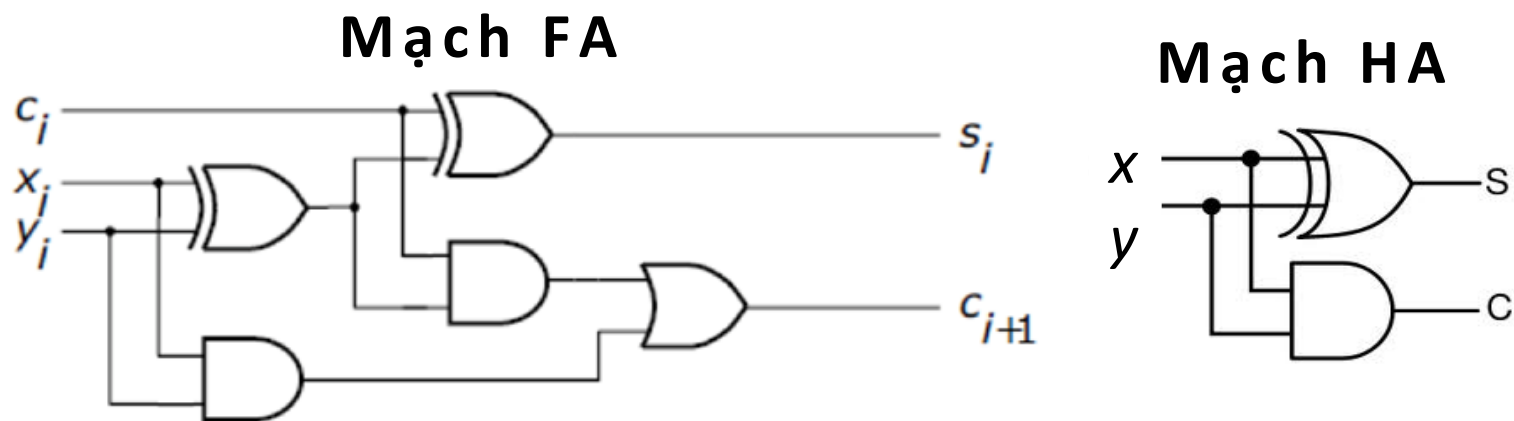
1. Mạch cộng (ADDER)

b. Mạch cộng toàn phần

Có thể sử dụng lại mạch cộng bán phần để tạo ra mạch cộng toàn phần

$$S_i = x_i \oplus y_i \oplus c_i$$

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



Mạch FA sử dụng lại HA



1. Mạch cộng (ADDER)

c. Mạch cộng Carry Ripple (có nhớ)

carry xuất hiện ở bit i

=> cộng thêm ở bit thứ $i+1$

Yếu điểm: Tốc độ bị giới hạn bởi quá trình truyền số nhớ.

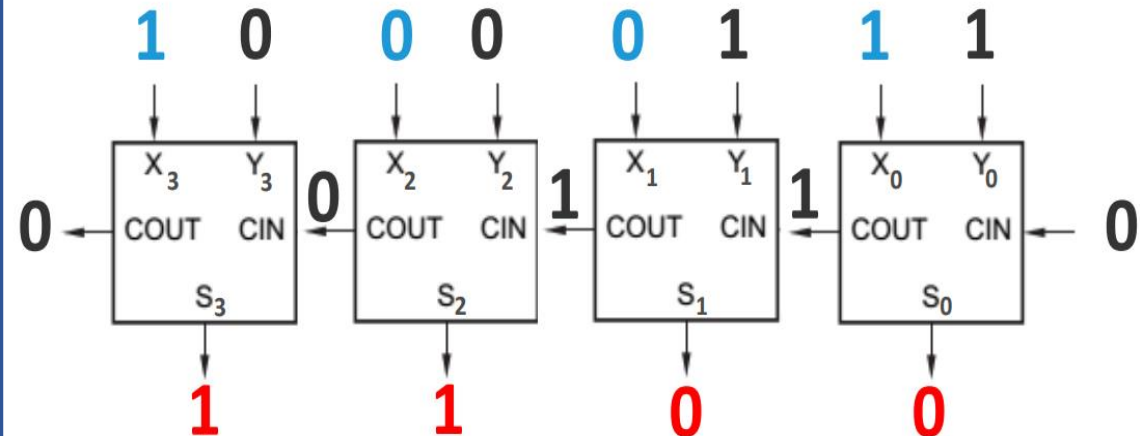
Độ trễ phụ thuộc vào số lượng bit:

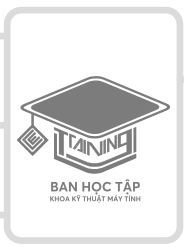
$$C_n = n \cdot \Delta t$$

=> Sử dụng mô hình carry look ahead (CLA) để cải thiện tốc độ

VD: Mạch cộng song song 4 bit

$$\begin{array}{l} X = 1001 \\ Y = 0011 \end{array} \Rightarrow X + Y = S = 1100$$





2. Mạch trừ

Với X, Y là 2 số không dấu n -bit

Phép cộng: $S = X + Y$

Phép trừ: $D = X - Y$

$$= X + (-Y)$$

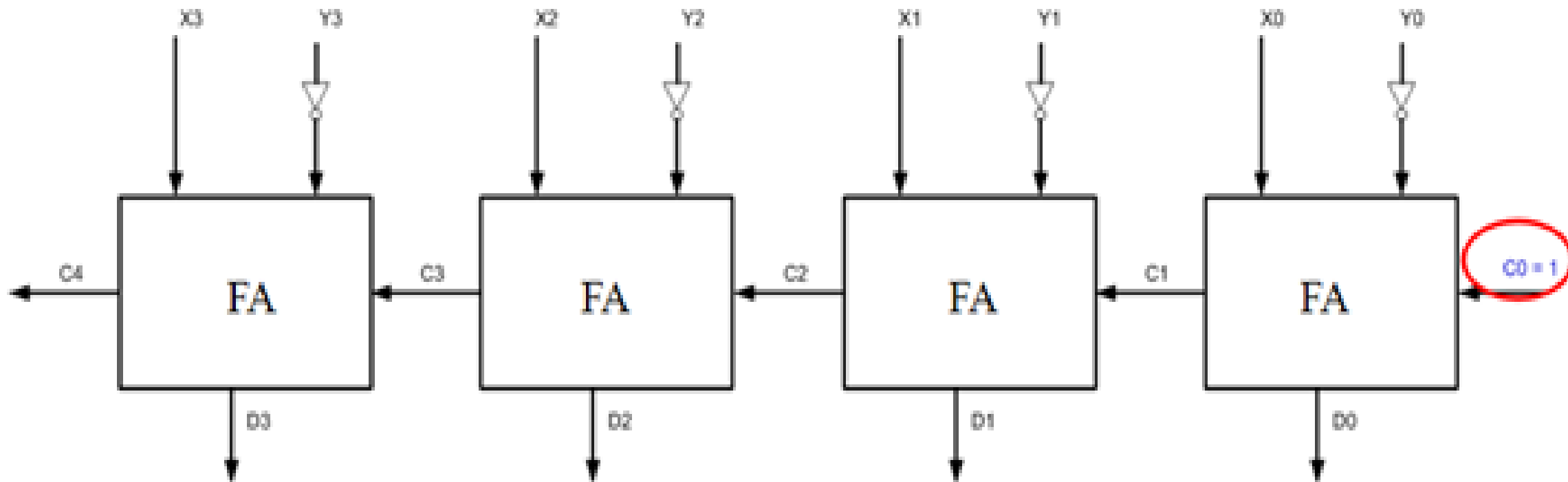
$$= X + (\text{Bù 2 của } Y)$$

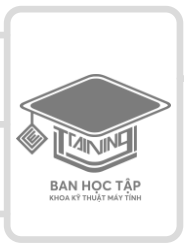
$$= X + (\text{Bù 1 của } Y) + 1$$

$$= X + Y' + 1$$

- Mạch cộng Carry Ripple có thể được dùng để xây dựng mạch trừ Carry Ripple bằng cách đảo Y và đặt số nhớ đầu tiên là 1

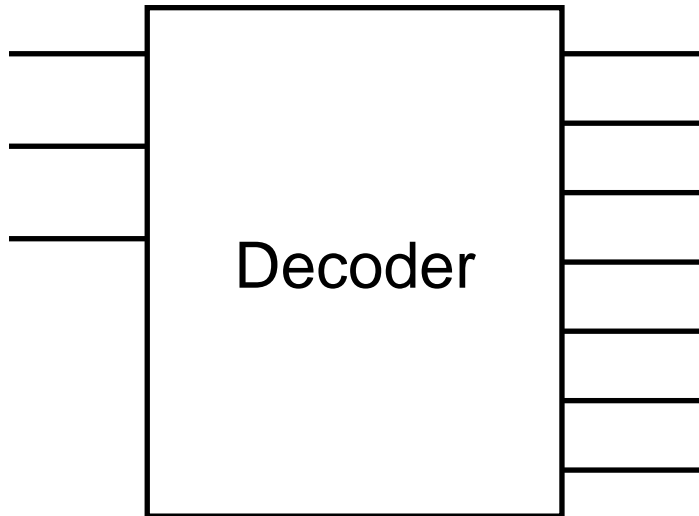
2. Mạch trừ



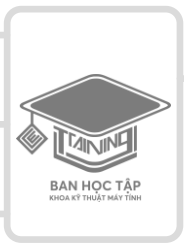


3. Decoder

Mạch giải mã nhị phân



- n -ra- 2^n
- Input code: Mã nhị phân
- Output code: 1-trong- 2^n



3. Decoder

Mạch giải mã (Decoder)

Nhiều inputs / nhiều outputs

Inputs < outputs

1 mã input chỉ tạo ra **1** mã output

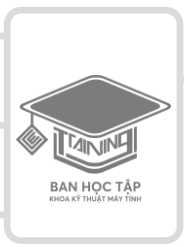
Các mã ngõ vào: binary, Code

Các mã ngõ ra: 1-trong-m, Gray Code, BCD Code

Mạch giải mã **n -ra- 2^n** : n ngõ vào và 2^n ngõ ra

- input: n bit nhị phân

- output: 1-trong- 2^n



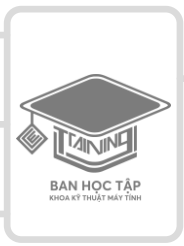
3. Decoder

Mạch giải mã (Decoder)

Ví dụ: $n=2$, mạch giải mã 2-ra-4

<i>Inputs</i>			<i>Outputs</i>			
EN	I1	I0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Chú ý: “x” (kí hiệu ngõ vào don’t care)



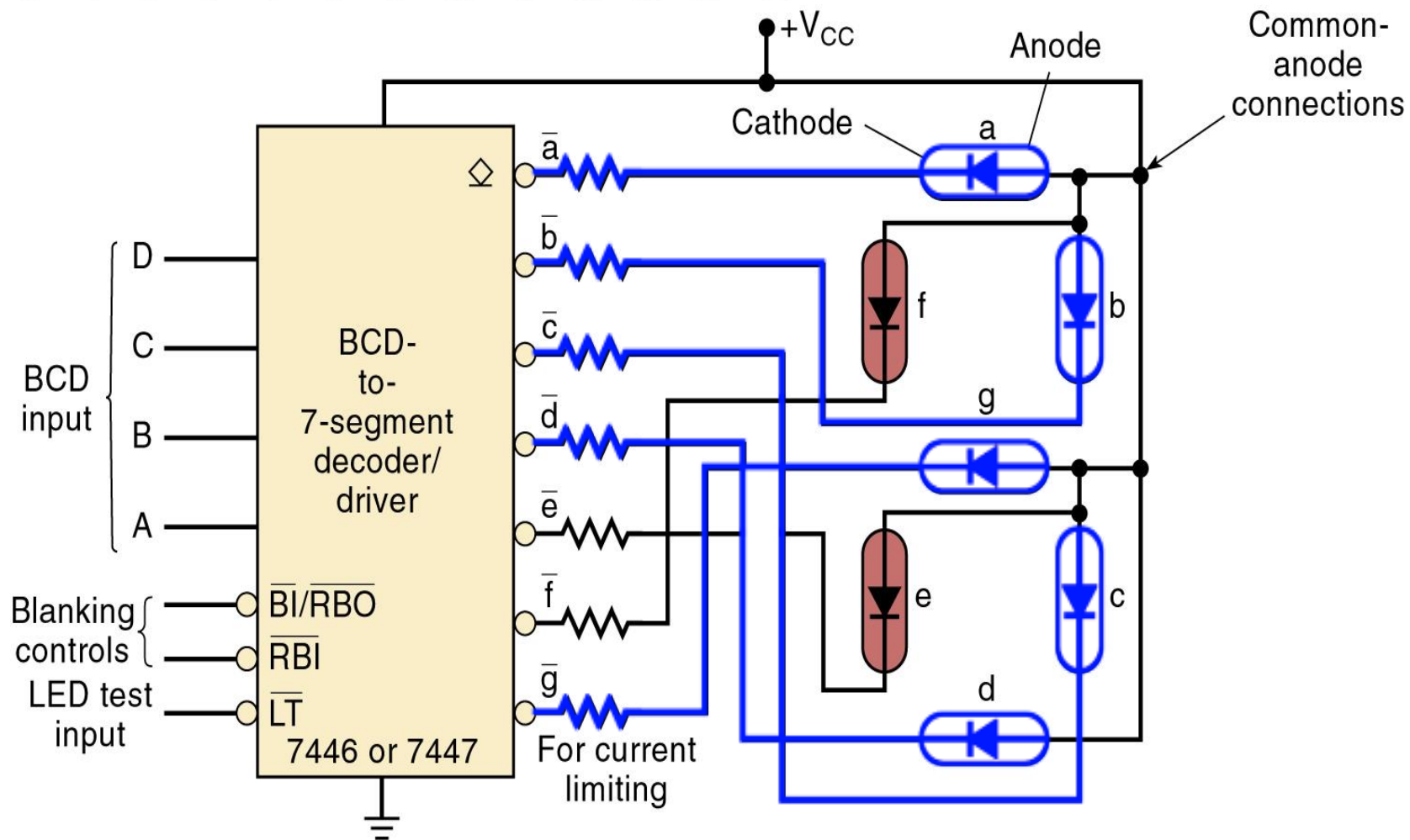
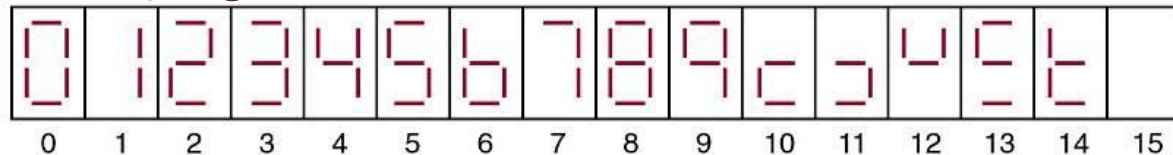
3. Decoder

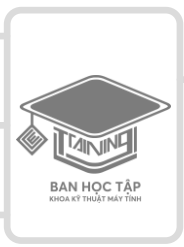
Ứng dụng Decoder

- ✓ Giải mã địa chỉ cho các chip nhớ
- ✓ Dùng trong LED 7 đoạn, LED 7 đoạn là cách phổ biến để hiển thị số thập phân hoặc số thập lục phân
 - Sử dụng LED cho mỗi đoạn
 - Điều khiển dòng điện qua mỗi LED => một số đoạn sẽ sáng và một số tắt => hiển thị số mong muốn

3. Decoder

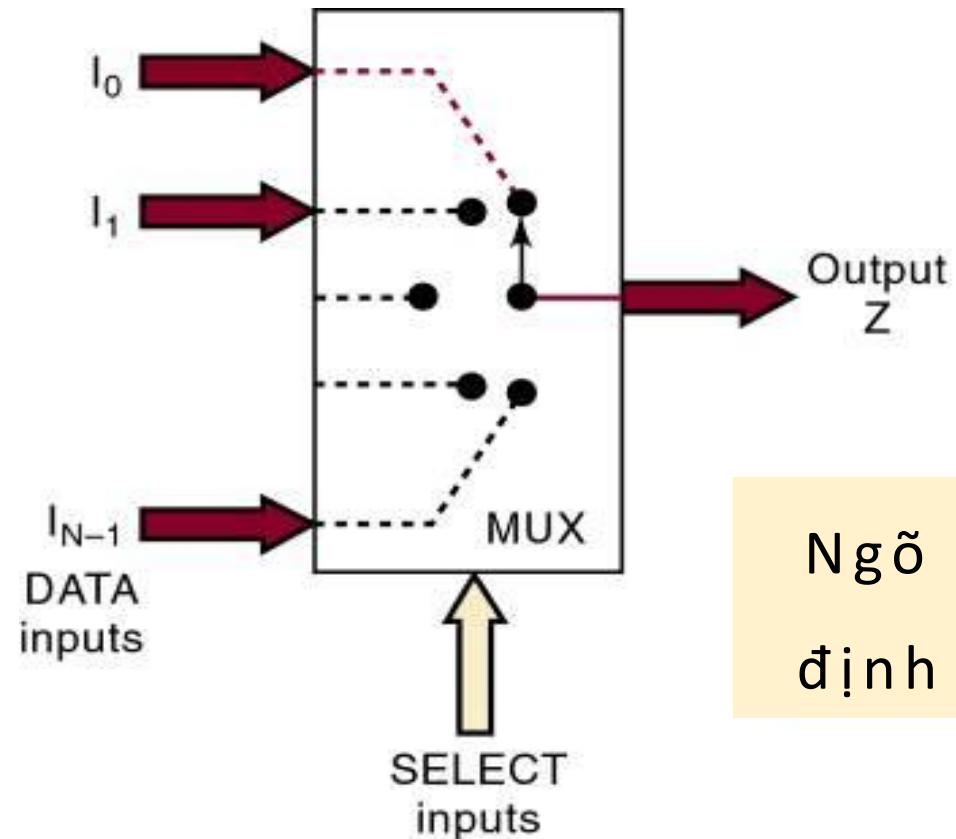
Ví dụ: giải mã BCD ra LED 7 đoạn





4. Mạch dồn kênh (MUX)

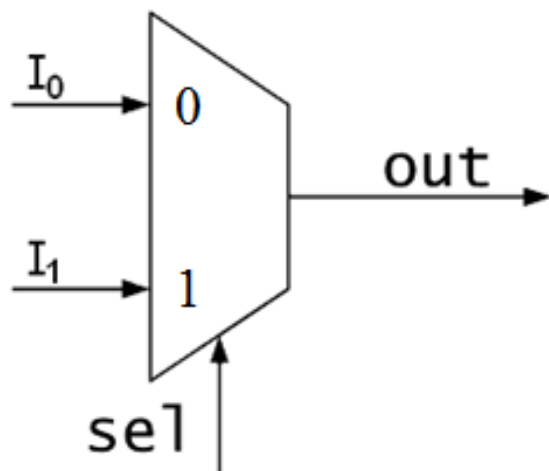
MUX truyền một trong những inputs tới output dựa trên tín hiệu Select



Ngõ vào SELECT sẽ xác định lấy input nào ra Z

MUX 2-to-1

Ký hiệu

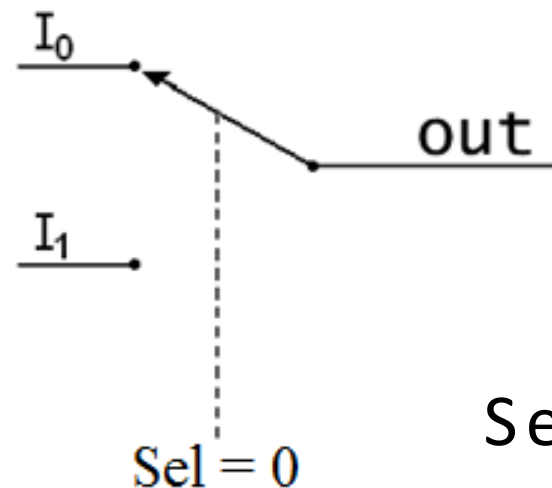


Biểu thức đại số

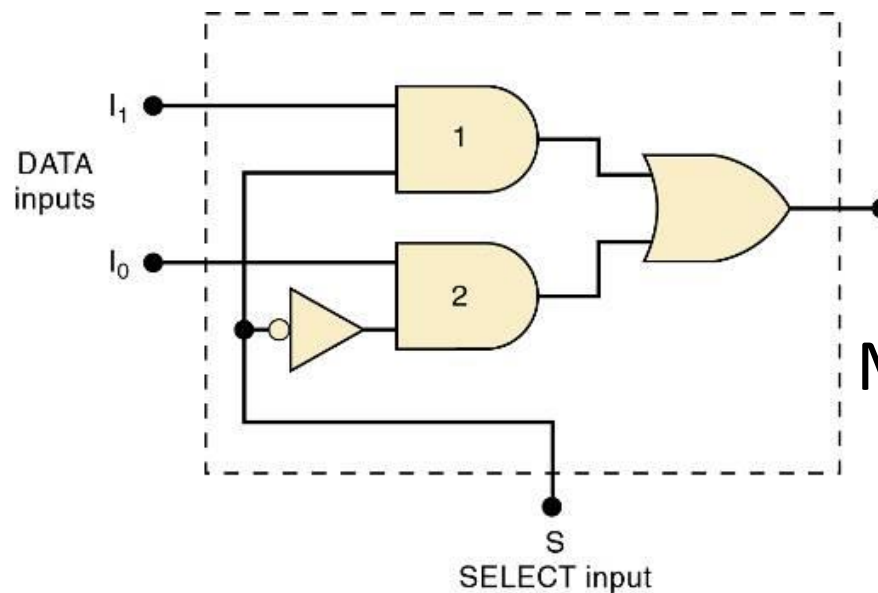
Sel	Out
0	I_0
1	I_1

$$\text{Out} = \overline{\text{Sel}} * I_0 + \text{Sel} * I_1$$

4. Mạch dồn kênh (MUX)



Sel = 0



Mạch logic

4. Mạch dồn kênh (MUX)

MUX 4-to-1

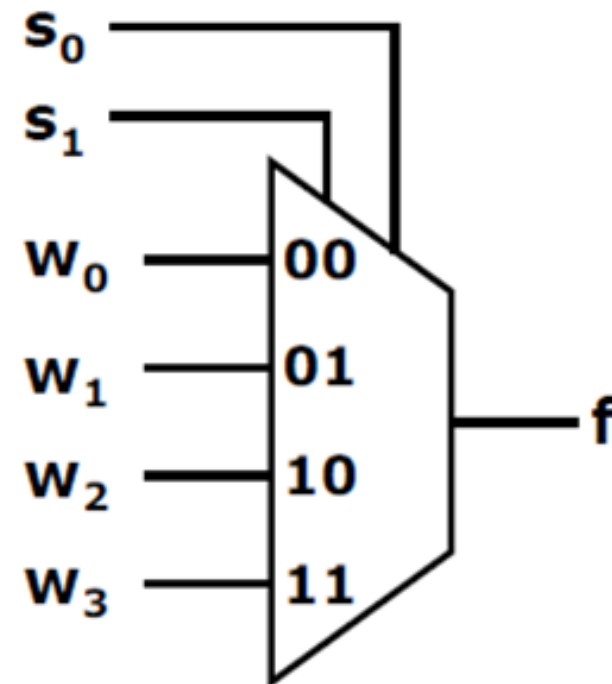
Từ 4 inputs, xuất ra 1 output dựa vào 2 tín hiệu select

s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

Bảng sự thật

$$f = s_1' s_0' w_0 + s_1' s_0 w_1 + s_1 s_0' w_2 + s_1 s_0 w_3$$

Biểu thức đại số



Ký hiệu

4. Mạch dồn kênh (MUX)

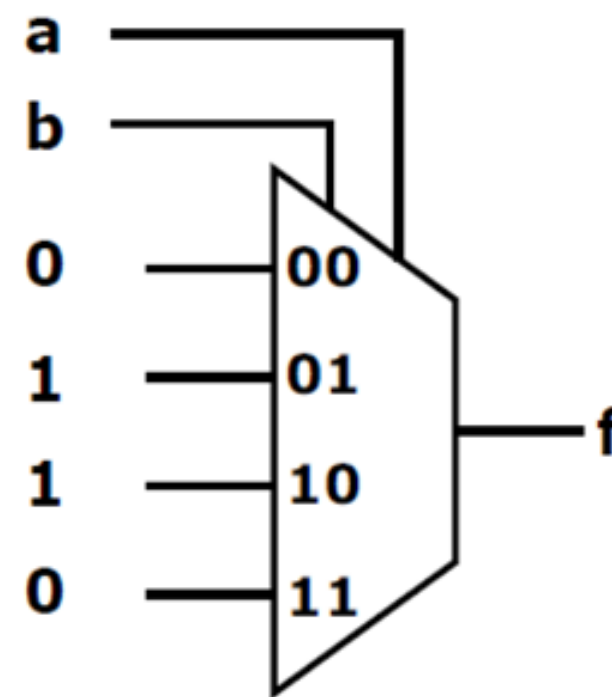
Thiết kế mạch logic sử dụng MUX

■ Cách hiện thực LUT (Look-up table)

- Sử dụng MUX để chọn một giá trị (**hằng số**) từ 1 LUT

Ví dụ: Thiết kế mạch XOR sử dụng MUX

<i>a</i>	<i>b</i>	<i>f</i>
0	0	0
0	1	1
1	0	1
1	1	0



4. Mạch dồn kênh (MUX)

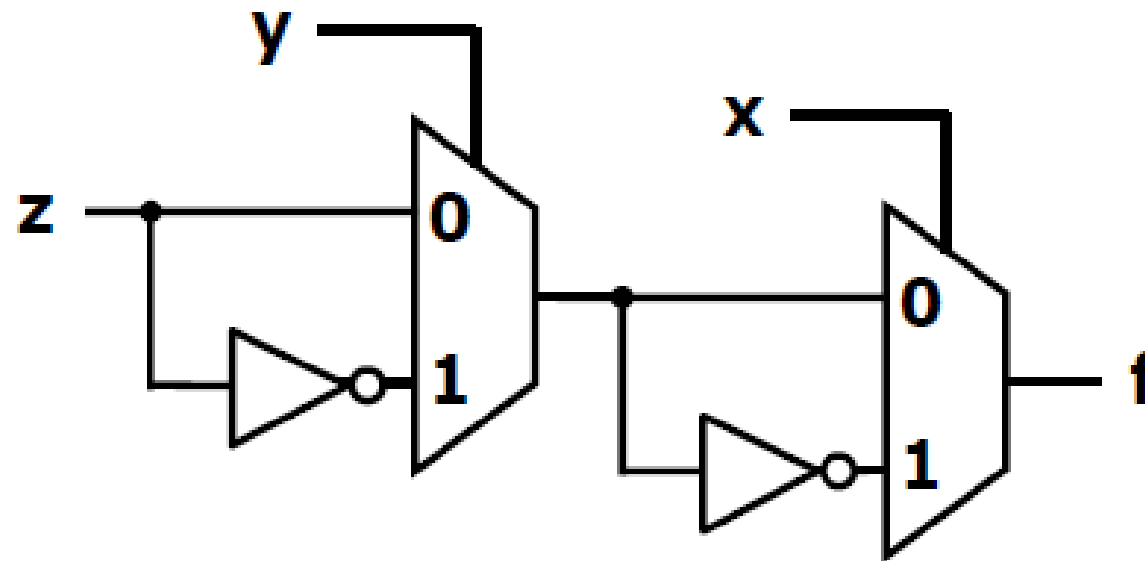
Thiết kế mạch logic sử dụng MUX

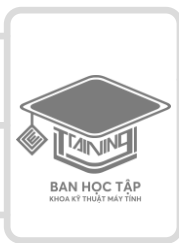
➤ XOR 3 ngõ vào có thể hiện thực bằng 2 MUX 2-to-1

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

$y \oplus z$

$(y \oplus z)'$





5. Biểu thức Shannon

Bất kì hàm Boolean $f(w_1, w_2, \dots, w_n)$ đều có thể được viết:

$$f(w_1, w_2, \dots, w_n) = \overline{w_1} * f(0, w_2, \dots, w_n) + w_1 * f(1, w_2, \dots, w_n)$$

VD1:

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

- Phân tích hàm này theo biến w_1 :

$$f(w_1, w_2, w_3) = \underbrace{w_1 (w_2 + w_3)}_{\text{f khi } w_1=1} + \underbrace{\overline{w_1} (w_2 w_3)}_{\text{f khi } w_1=0}$$

f khi $w_1=1$

f khi $w_1=0$

5. Biểu thức Shannon



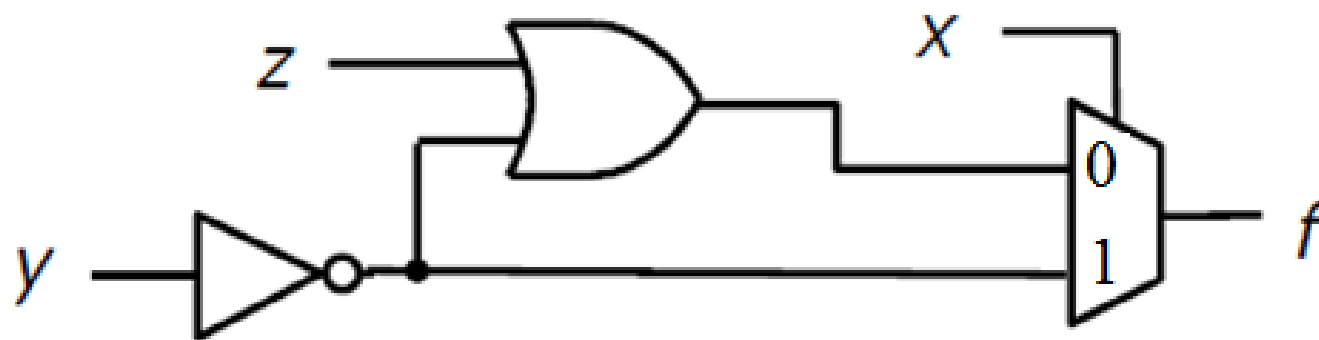
x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

VD2: $f = x'y'z' + x'y'z + x'yz + xy'z' + xy'z$

Chọn x làm biến mở rộng

$$f = x'(y'z' + y'z + yz) + x(y'z' + y'z)$$

$$f = x'(y' + z) + x(y')$$



5. Biểu thức Shannon



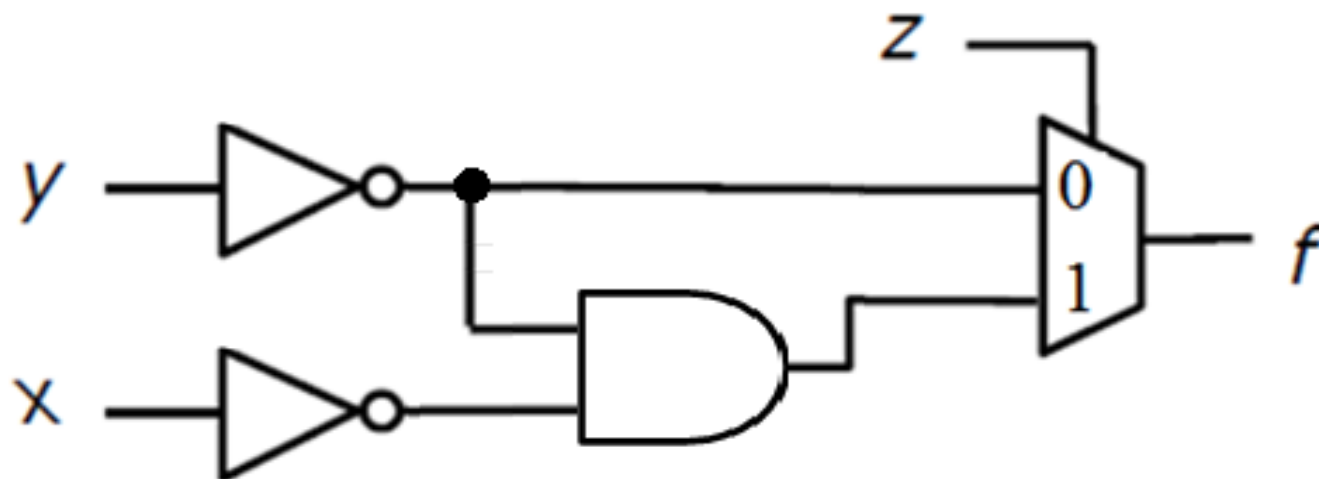
$$\text{VD2: } f = x'y'z' + x'y'z + x'yz + xy'z' + xy'z$$

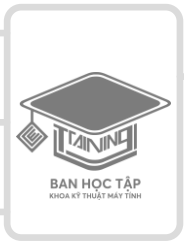
Chọn z làm biến mở rộng

$$f = z' \cdot (x'y' + xy') + z \cdot (x'y' + x'y + xy')$$

$$= z' \cdot y' + z \cdot (x' + xy')$$

$$= z' \cdot y' + z \cdot (x' + y')$$

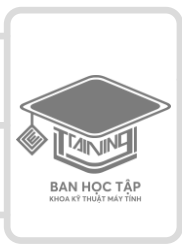




6. Mạch so sánh

Mạch so sánh 2 số A và B

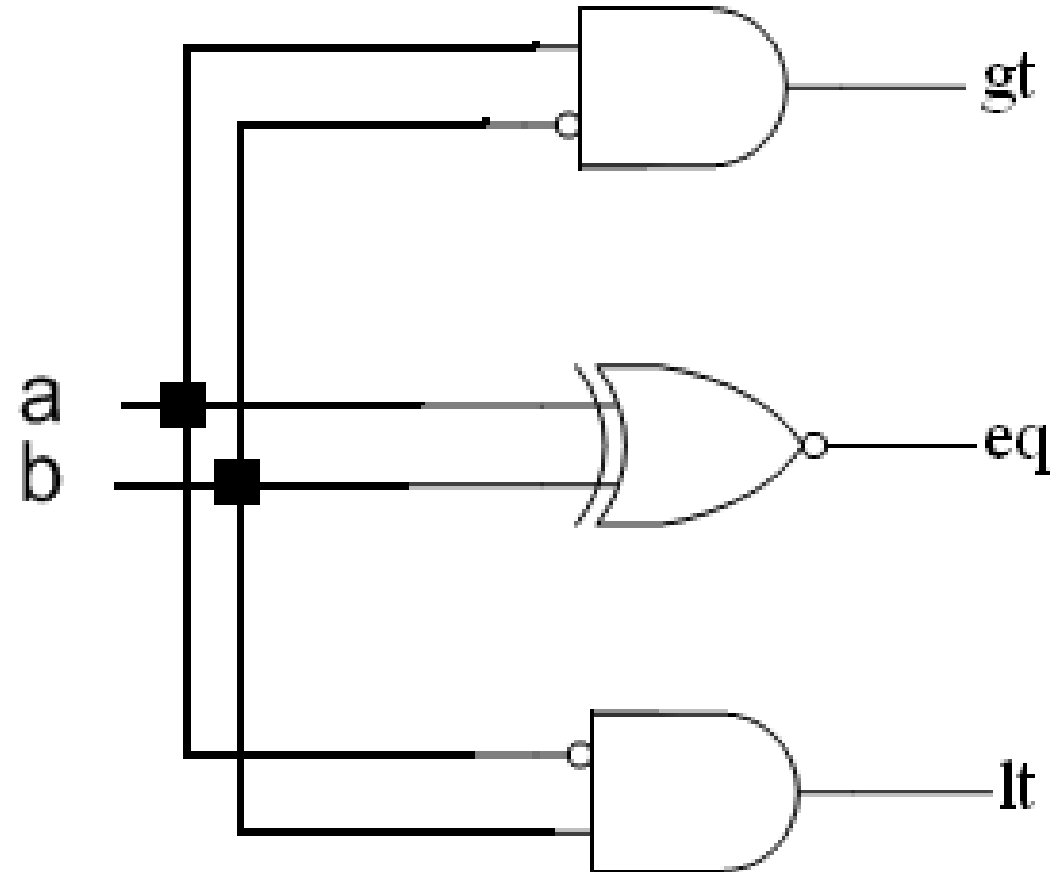
- Xuất 1 nếu $A = B$
- Xuất 0 nếu $A \neq B$
- Dùng cổng **XOR** (= 0 nếu ngõ vào giống nhau và =1 nếu khác nhau)
- Dùng cổng **XNOR** (=1 nếu ngõ vào giống nhau và =0 nếu khác nhau)



6. Mạch so sánh

Mạch so sánh 1 bit

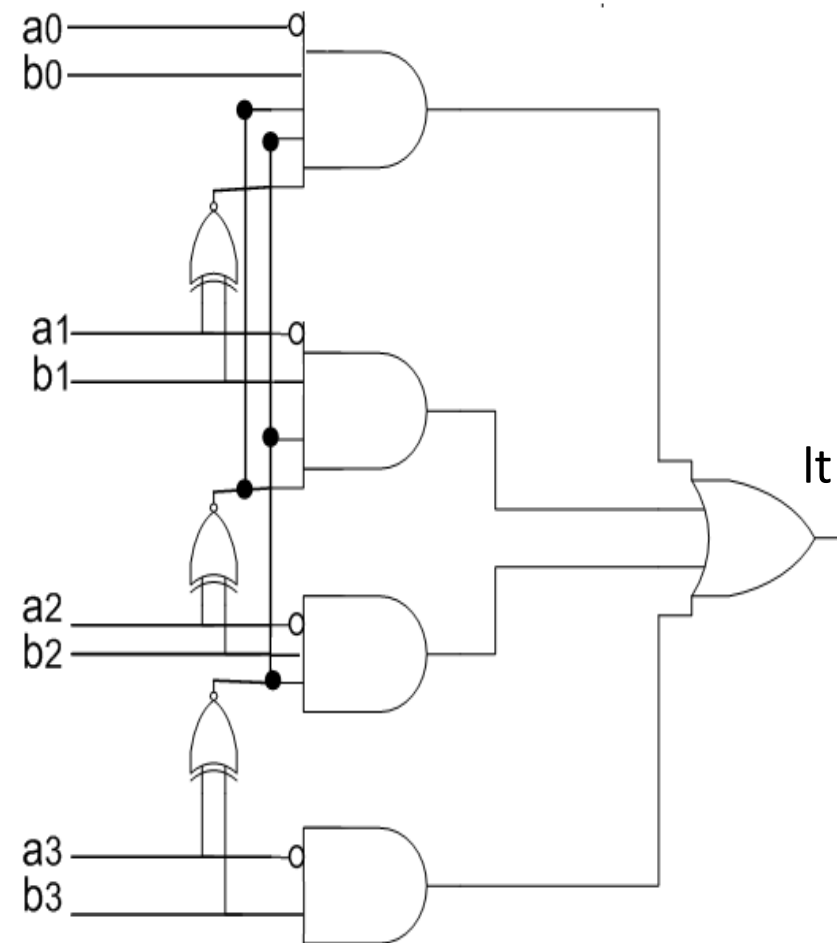
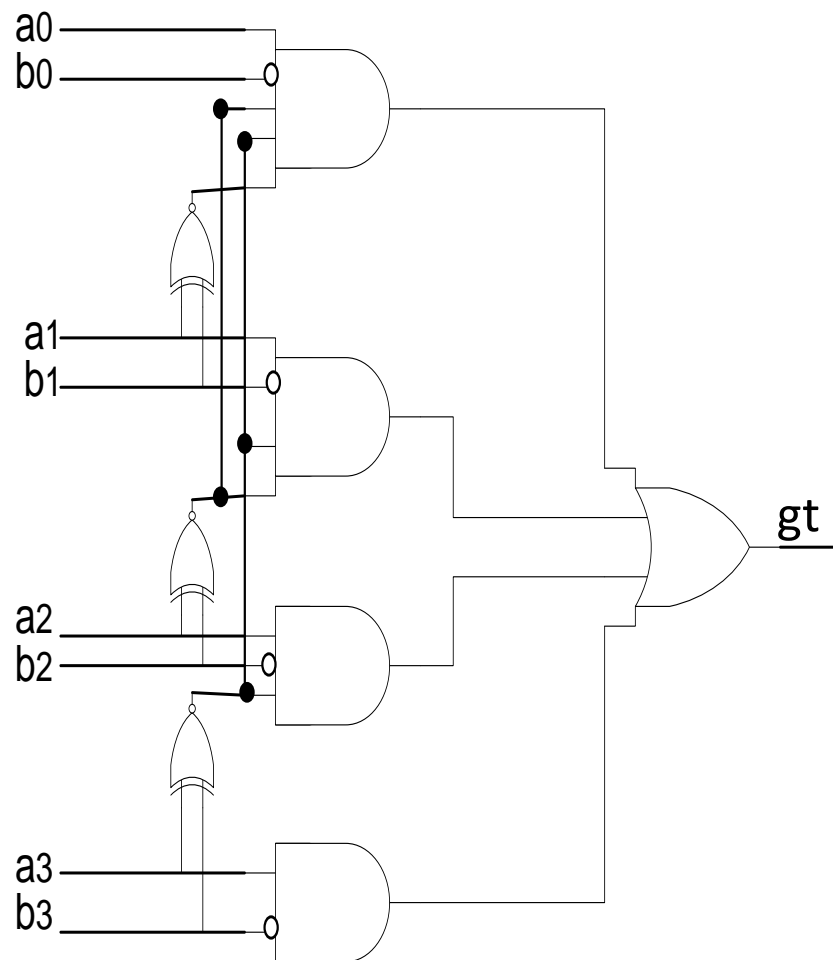
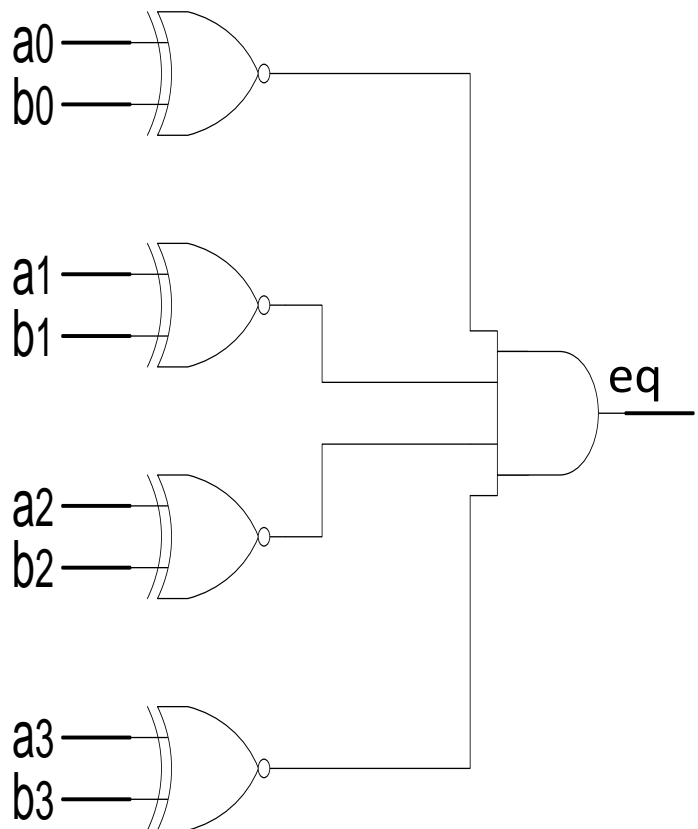
a	b	gt	eq	lt
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

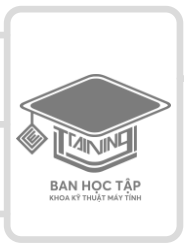


6. Mạch so sánh



Mạch so sánh 4 bit





7. Bài tập

Câu 1: Cần ít nhất bao nhiêu mạch cộng toàn phần để thực hiện thao tác cộng 5 bit

A. 2

B. 3

☒ C. 4 (ít nhất 4 mạch cộng toàn phần với 1 mạch cộng bán phần)

D. 5

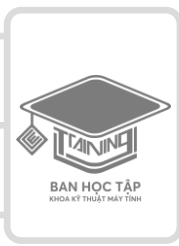
Câu 2: Mạch giải mã nhị phân (Binary Decoders) với n ngõ vào sẽ có bao nhiêu ngõ ra

A. 2^{n-1}

☒ B. 2^n

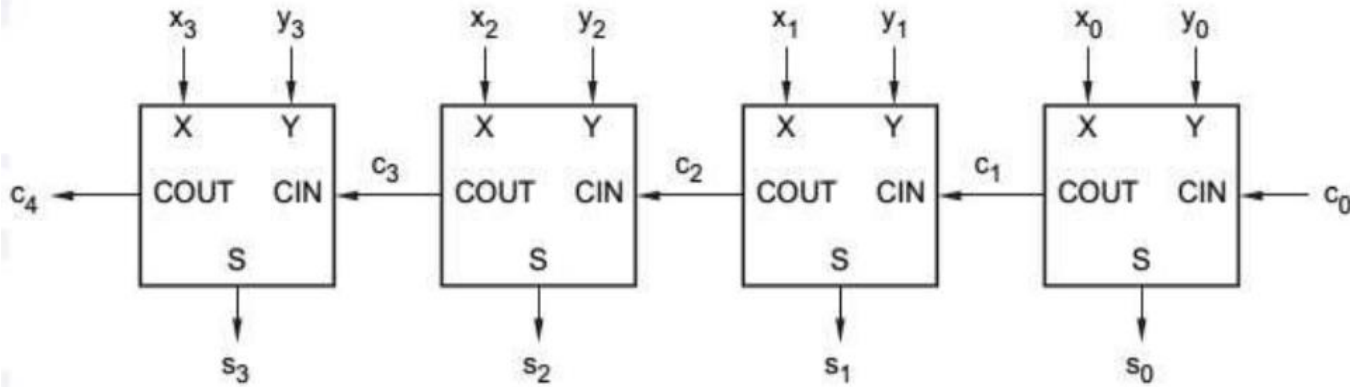
C. n^2

D. $2^n - 1$



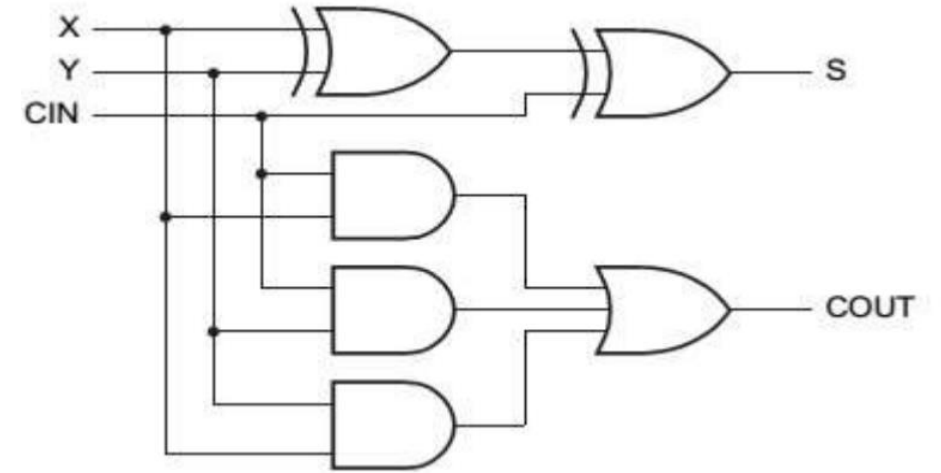
7. Bài tập

Câu 3: Mạch sau thực hiện thao tác gì

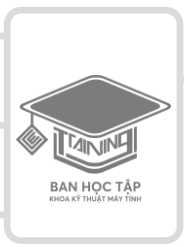


- A. cộng 5-bit
- B. trừ 5-bit
- ☒ C. cộng 4-bit
- D. trừ 4-bit

Câu 4 : Sơ đồ sau là mạch gì ?



- ☒ A. FA
- B. HA
- C. MUX 2-1
- D. Decoder



7. Bài tập

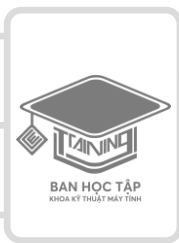
Câu 5: Mạch mã hoá nhị phân (Binary Encoder) với 16 ngõ vào sẽ có bao nhiêu ngõ ra ?

- A, 8
- ☒ B. 4
- C. 2
- D. 6

2^n ngõ vào có n ngõ ra
 $\Rightarrow 24$ ngõ vào $\rightarrow 4$ ngõ ra.

Câu 6: MUX 16:1 cần ít nhất bao nhiêu ngõ vào điều khiển?

- ☒ A. 4
- B. 2
- C. 3
- D. 5



7. Bài tập

Câu 7: cho hàm $F(A,B,C) = \text{SOP}(1,2,6,7)$. Chỉ dùng MUX và cổng NOT. Thiết kế mạch F?

Bảng chân trị

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

=> A là tín hiệu điều khiển

* Xét hàm I0: B là tín hiệu điều khiển

C \ B	0	1
0	0	1
1	1	0

Ta thấy B = 0 thì I0 = C

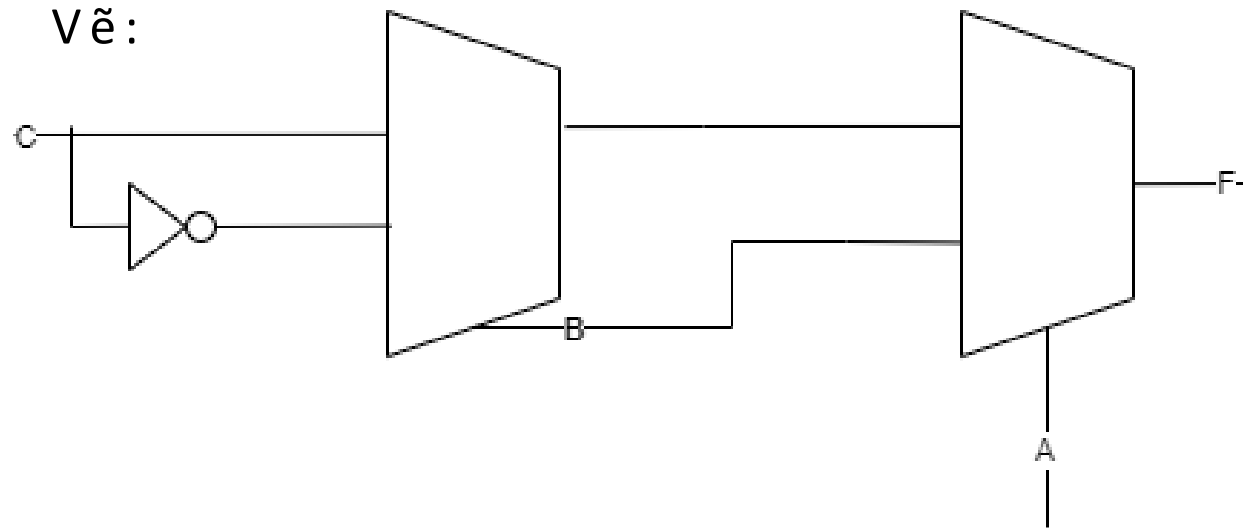
B = 1 thì I0 = C'

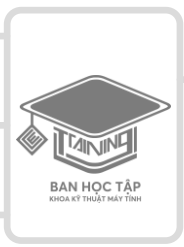
* Xét hàm I1: Lấy B là tín hiệu điều khiển

C \ B	0	1
0	0	1
1	0	1

Ta thấy với B = 0 | B = 1 thì I1 = B

Vẽ:

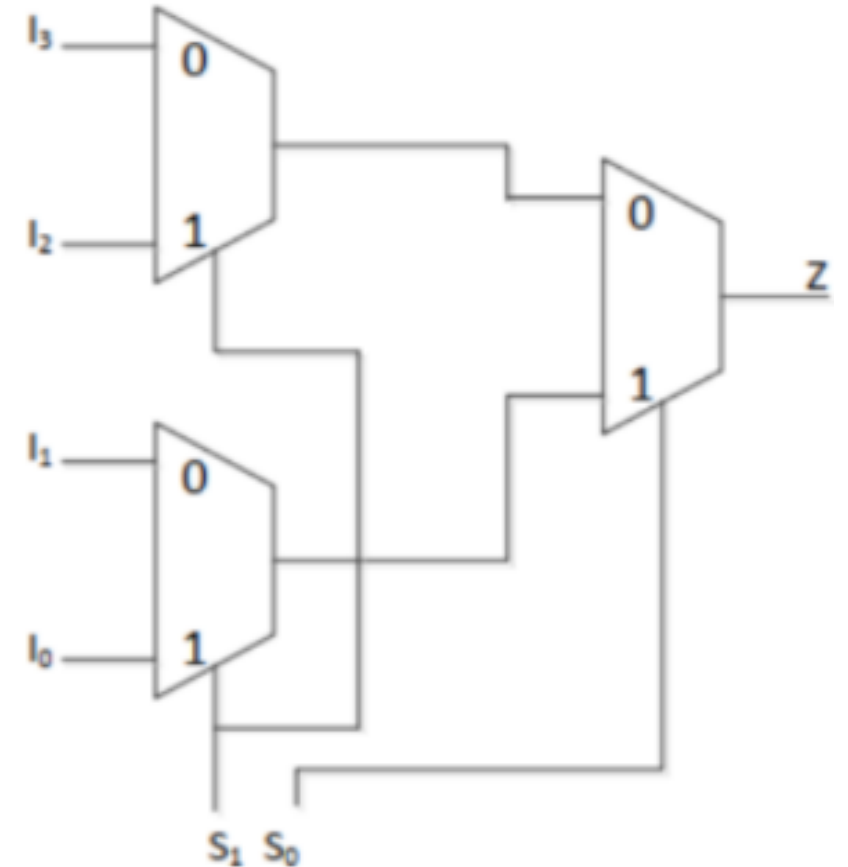




7. Bài tập

Câu 8 : Cho mạch như hình bên dưới, ngõ ra $Z = 0$ khi ngõ vào $S_1S_0 = 10$ và $I_3I_2I_1I_0$ là

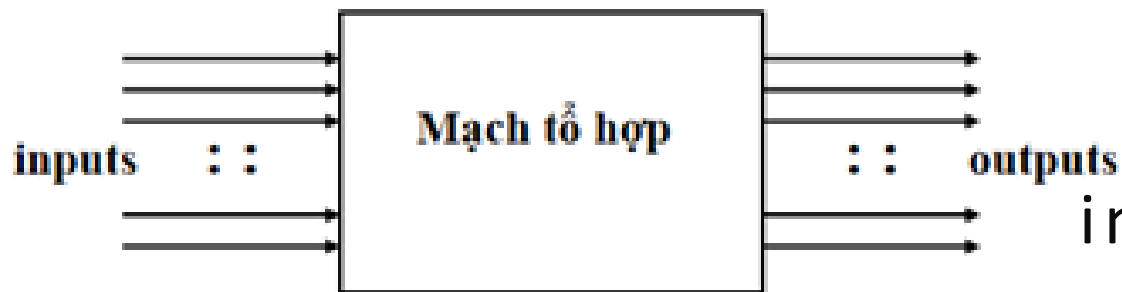
- A. 1101
- B. 0110
- ☒ C. 1010
- D. 1000



II. MẠCH TUẦN TỰ

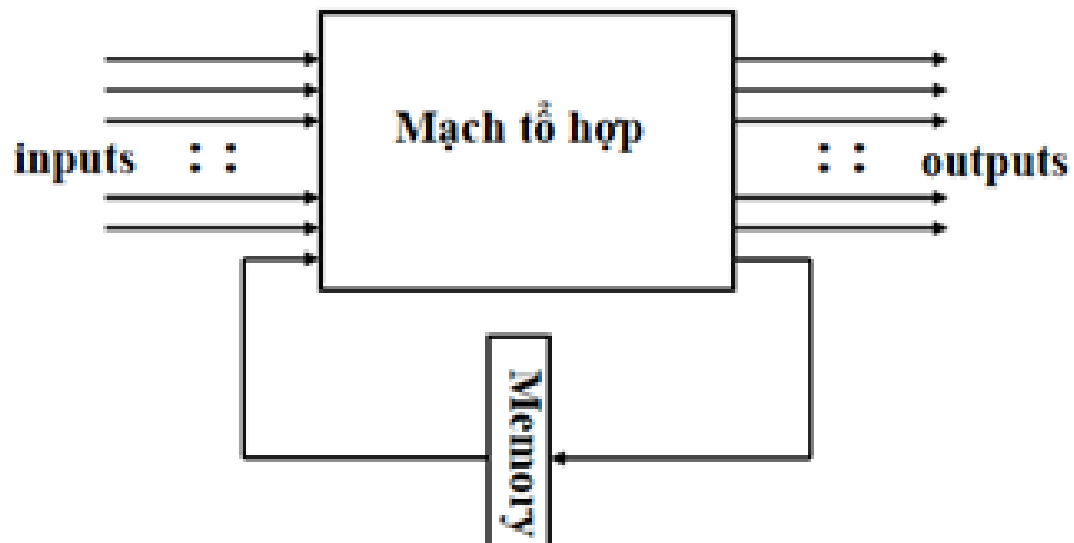


0. Phân biệt mạch tổ hợp và mạch tuần tự



MẠCH TỔ HỢP

inputs thay đổi => outputs thay đổi



MẠCH TUẦN TỰ

inputs
outputs trước đó } => Outputs thay đổi

Mạch có tính chất nhớ

NỘI DUNG

Latch

S-R latch

D latch

Bộ đếm bất đồng bộ

Bộ đếm đồng bộ

Flip-flop

D flip-flop

S-R flip-flop

T flip=flop

J-K flip-flop

Chuyển đổi giữa các FF



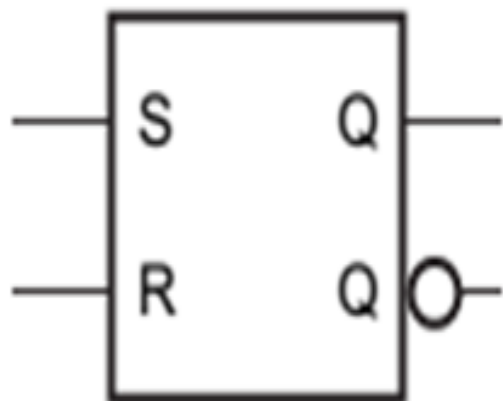
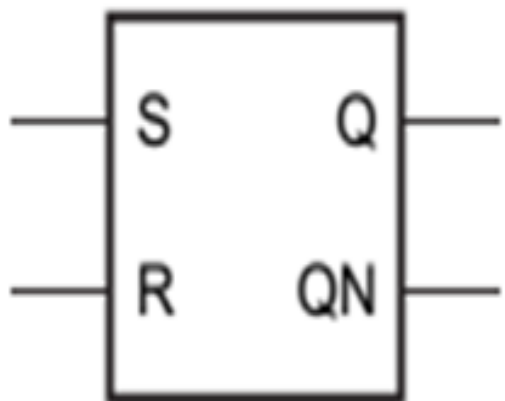
1. Latch (Chốt)

Bảng sự thật

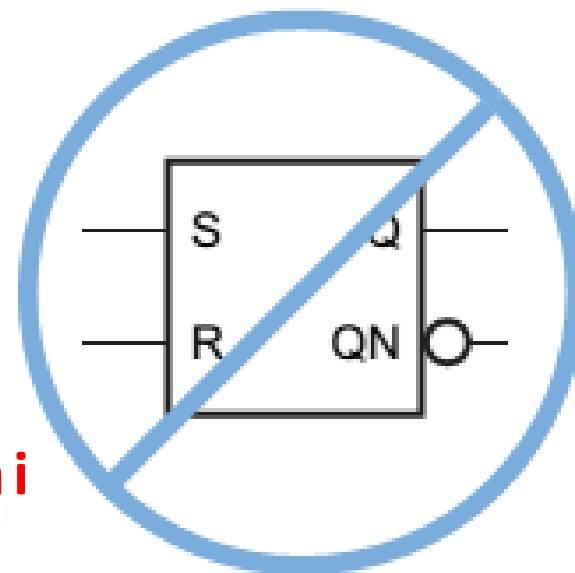
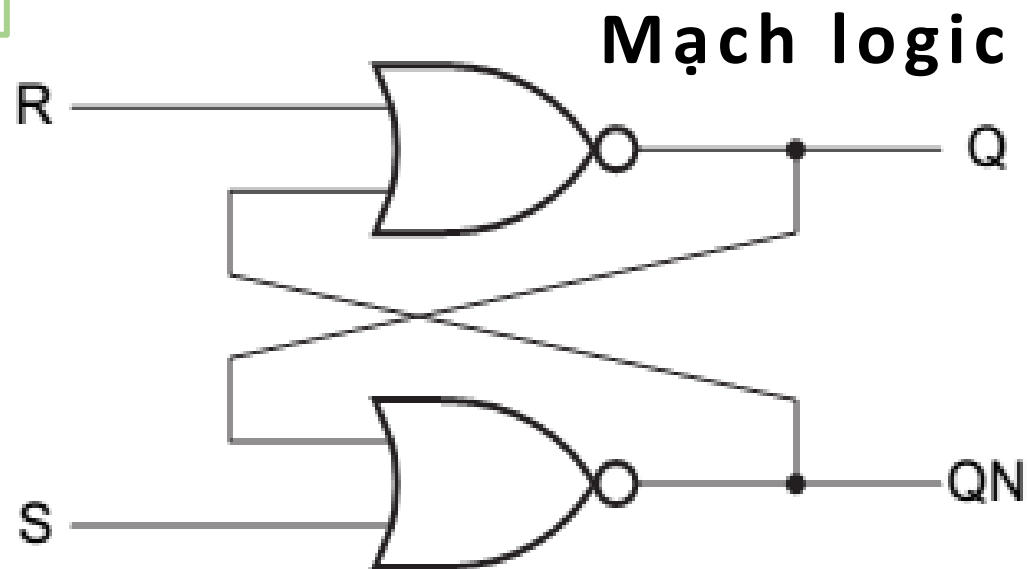
Chốt S - R

S	R	Q	QN
0	0	last Q	last QN
0	1	0	1
1	0	1	0
1	1	0	0

Cấm sử dụng



Ký hiệu



Ký hiệu sai



1. Latch (Chốt)

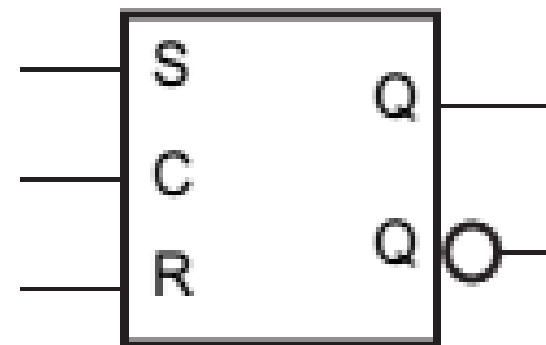
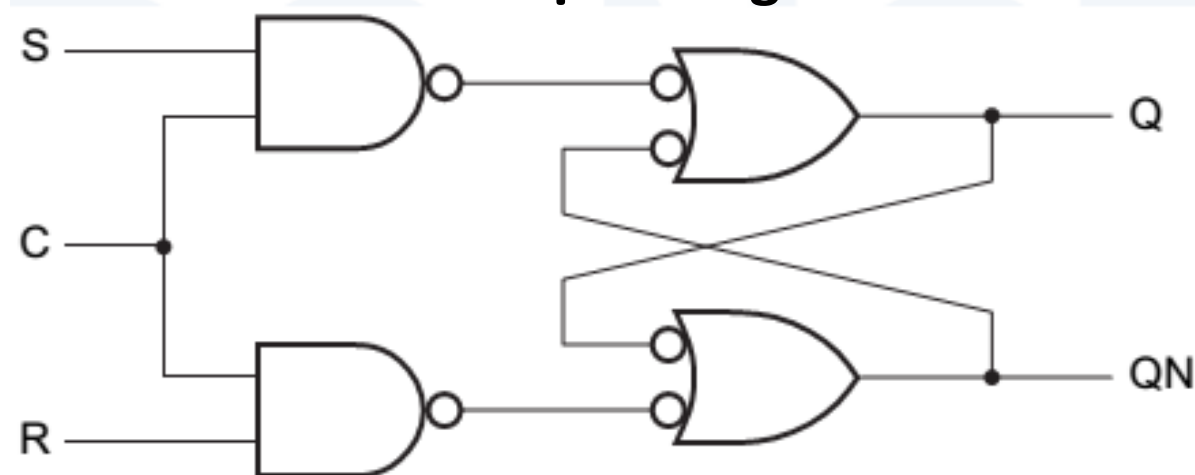
Chốt S-R với ngõ vào cho phép

Bảng sự thật

S	R	C	Q	QN
0	0	1	last Q	last QN
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1
x	x	0	last Q	last QN

Cấm sử dụng

Mạch logic



Ký hiệu

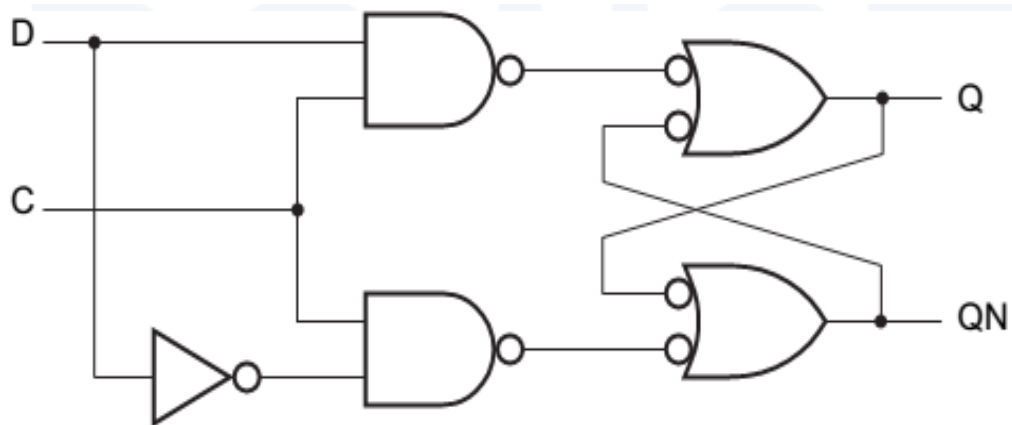
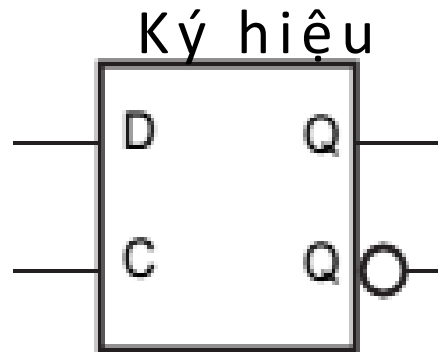


1. Latch (Chốt)

Chốt D

C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN

Bảng sự thật



Mạch logic

Loại bỏ những hạn chế trong chốt S-R khi **S** và **R** cùng chuyển từ 1 xuống 0

Ngõ vào điều khiển **C** giống với ngõ vào cho phép (enable)

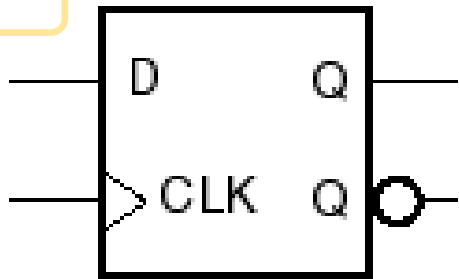
C tích cực, $Q = D \rightarrow$ chốt mở

C không tích cực, Q giữ giá trị trước đó \rightarrow chốt đóng

2. Flip-Flop



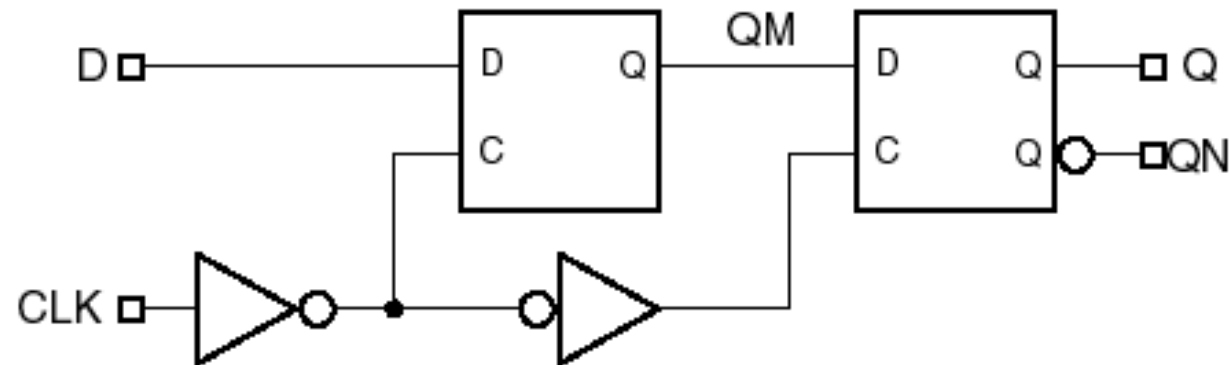
Flip-flop D kích cạnh lên



Ký hiệu

D	CLK	Q	QN
0		0	1
1		1	0

Bảng sự thật



Mạch logic

Gồm **2 D-latch** kết nối sao cho có dữ liệu truyền từ ngõ D ra ngõ Q mỗi khi CLK tích cực cạnh lên

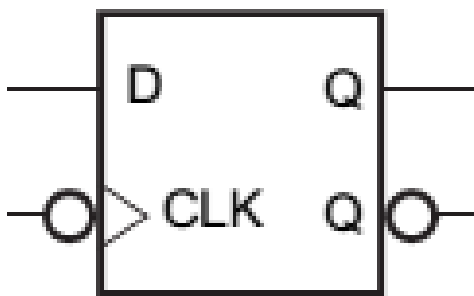
D-latch thứ 1 là Chủ (master), hoạt động khi CLK ở mức 0

D-latch thứ 2 là Tớ (slave), hoạt động khi CLK ở mức 1

2. Flip-Flop



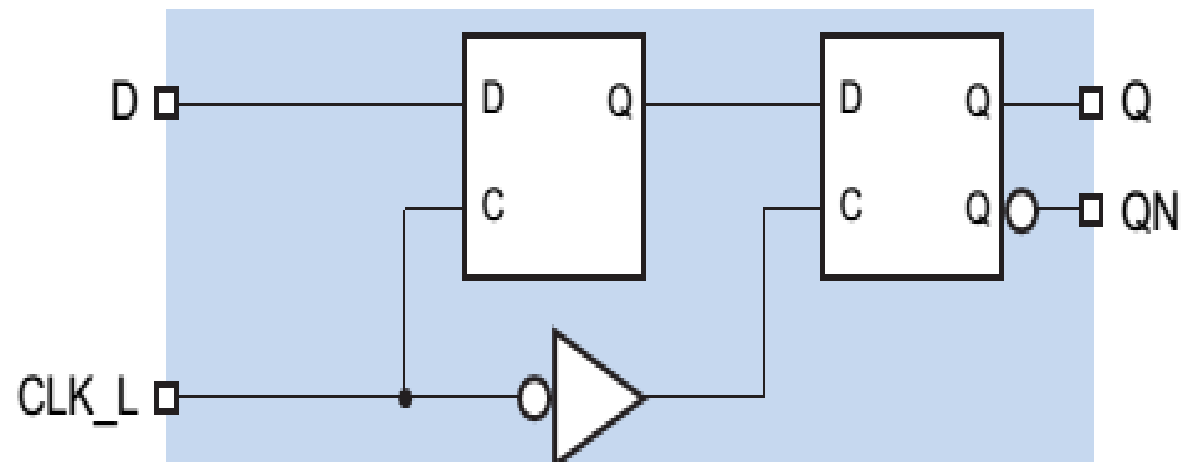
Flip-flop D kích cạnh xuống



Ký hiệu

D	CLK_L	Q	QN
0		0	1
1		1	0

Bảng sự thật



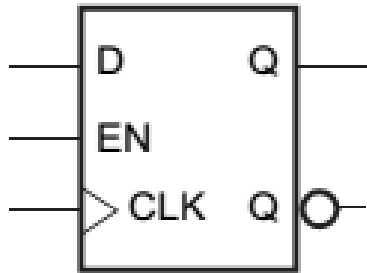
Mạch logic

FF-D kích cạnh xuống thiết kế giống FF-D kích cạnh lên, nhưng **đảo** ngõ vào xung Clock của 2 chốt D

2. Flip-Flop



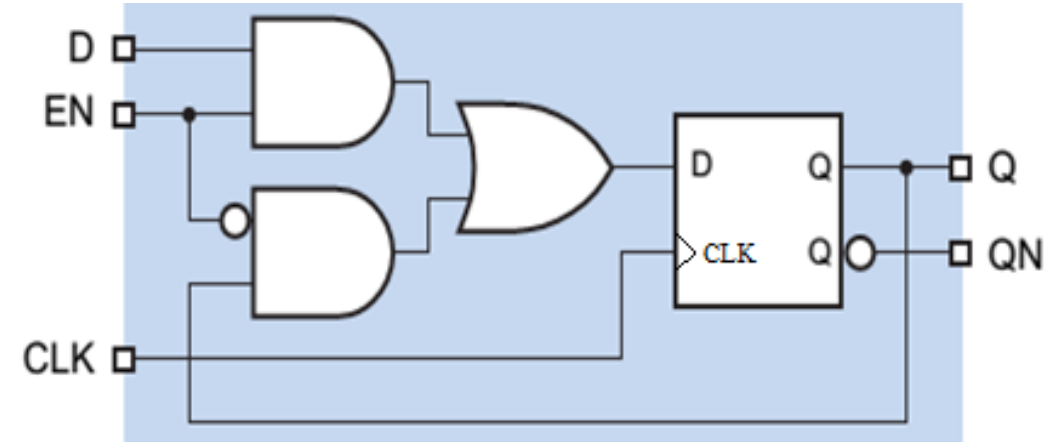
Flip-flop D với ngõ vào điều khiển



Ký hiệu

D	EN	CLK	Q	QN
0	1		0	1
1	1		1	0
x	0		last Q	last QN

Bảng sự thật



Mạch logic

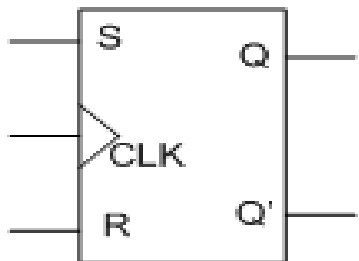
Thêm vào ngõ vào Enable cho mỗi FF (ký hiệu **EN** hoặc **CE**)

=> Giúp FF-D có lưu giữ dữ liệu sau cùng tốt hơn là nạp vào dữ liệu mới.



2. Flip-Flop

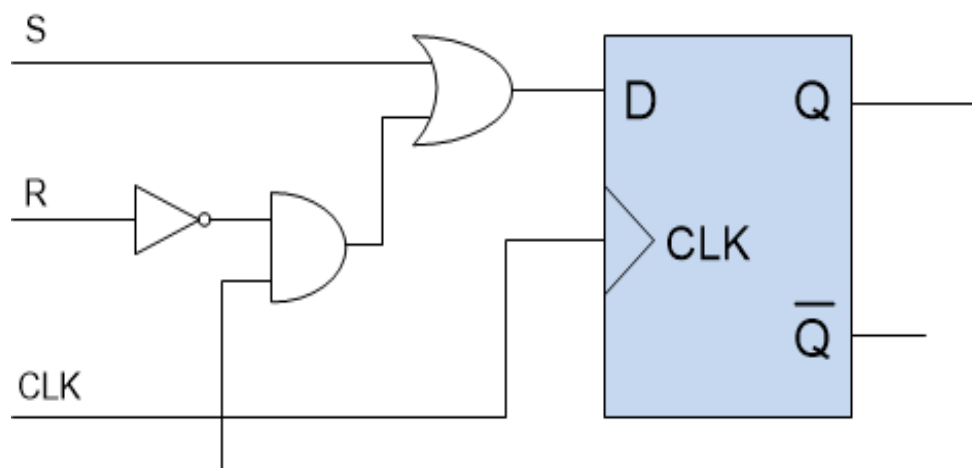
Flip-flop S-R kích cạnh lên



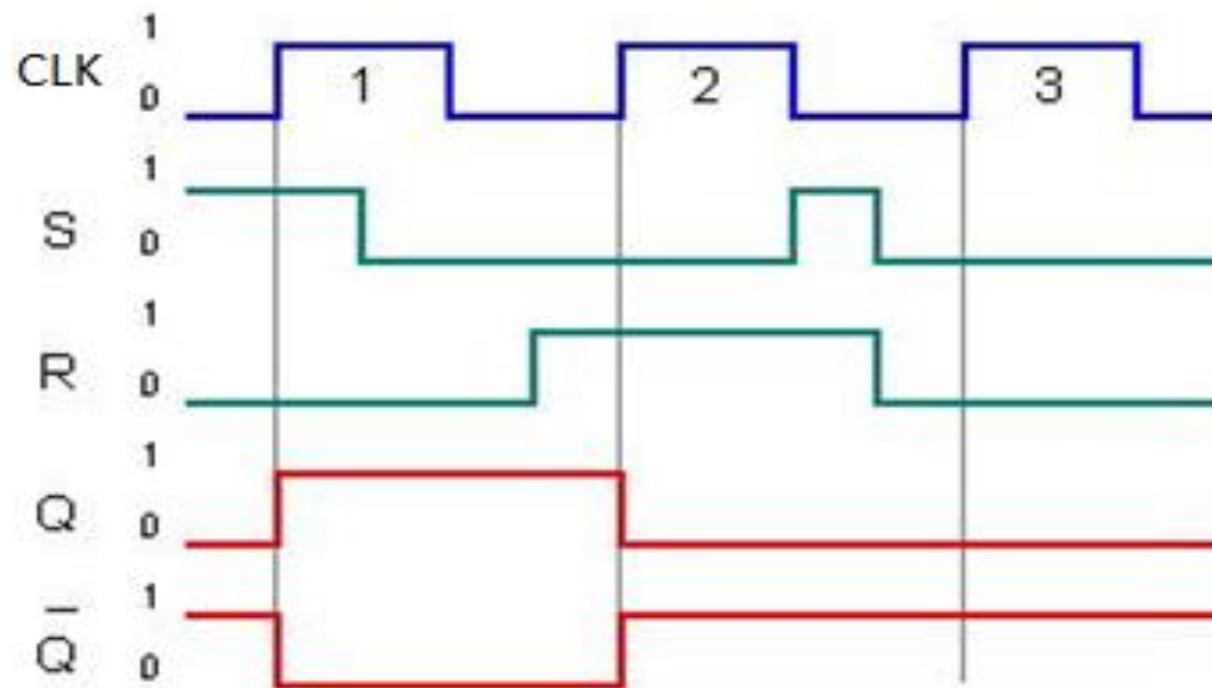
Ký hiệu

S	R	CLK	Q	Q'
0	0	f	last Q	last Q'
0	1	f	0	1
1	0	f	1	0
1	1	f	x	x

Bảng sự thật



FF S-R kích cạnh lên

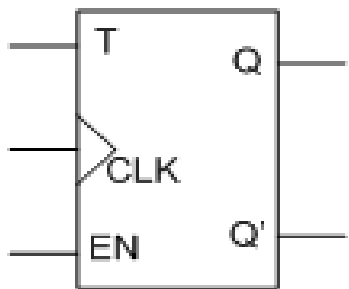


Hoạt động của **FF-S_R** kích cạnh lên

2. Flip-Flop



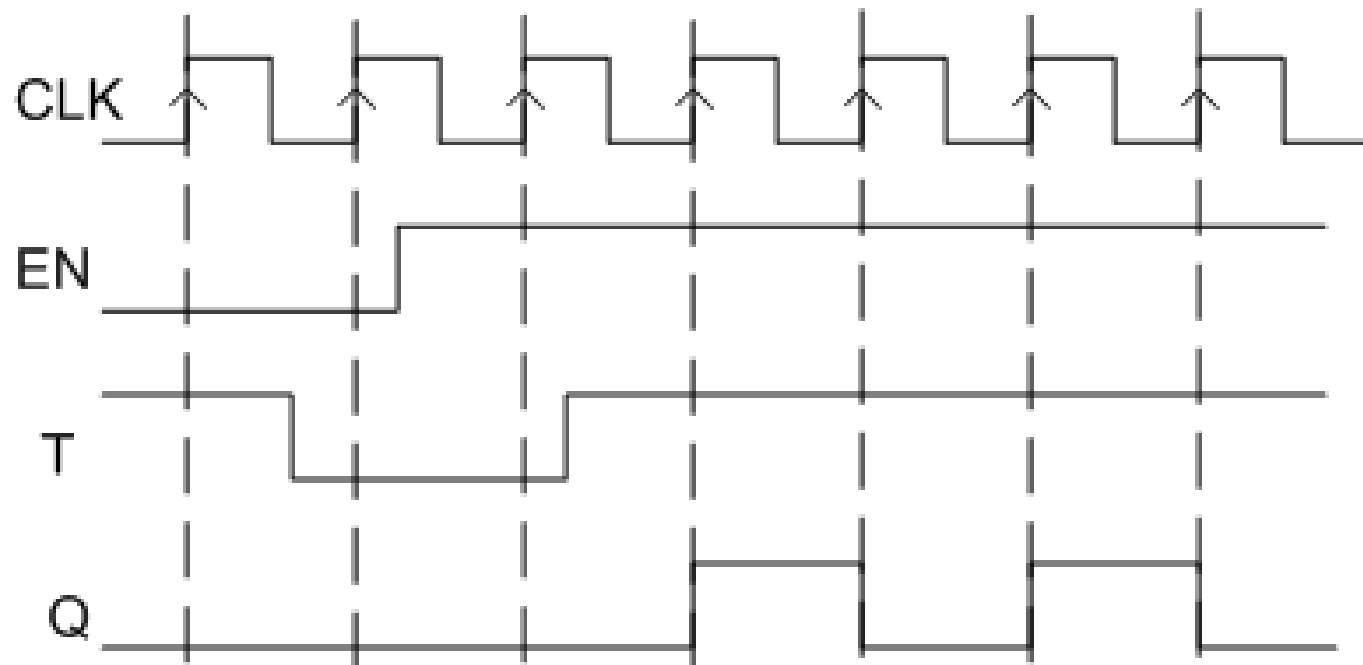
Flip-flop T



Ký hiệu

EN	T	CLK	Q	Q'
0	x	x	last Q	last Q'
1	0	f	last Q	last Q'
1	1	f	Q'	Q

Bảng sự thật

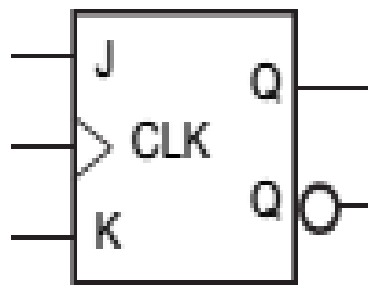


Hoạt động của **FF-T** tích cực cạnh lên

Flip-flop đảo trạng thái tại cạnh lên của xung Clock chỉ khi ngõ vào **EN** và **T** tích cực.

2. Flip-Flop

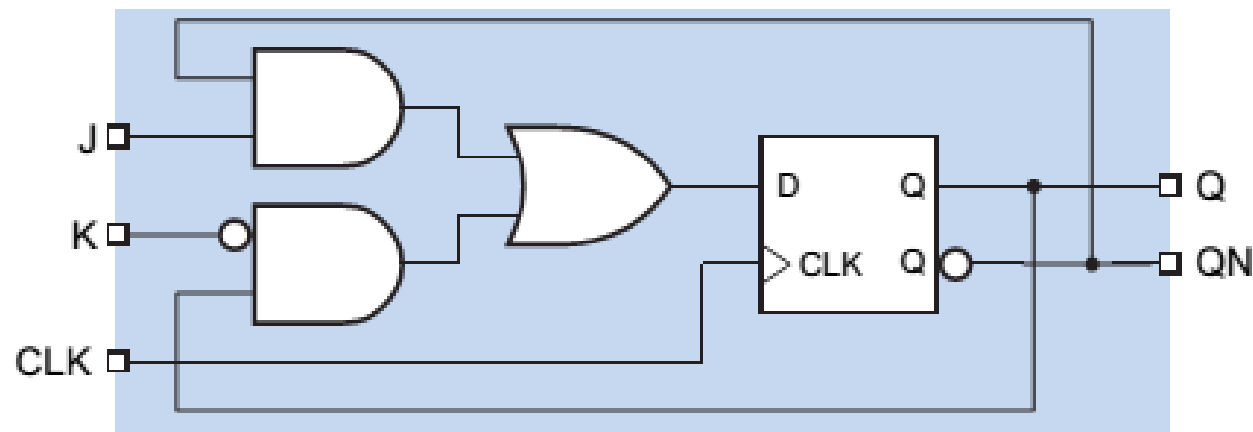
Flip-flop J-K kích cạnh lên



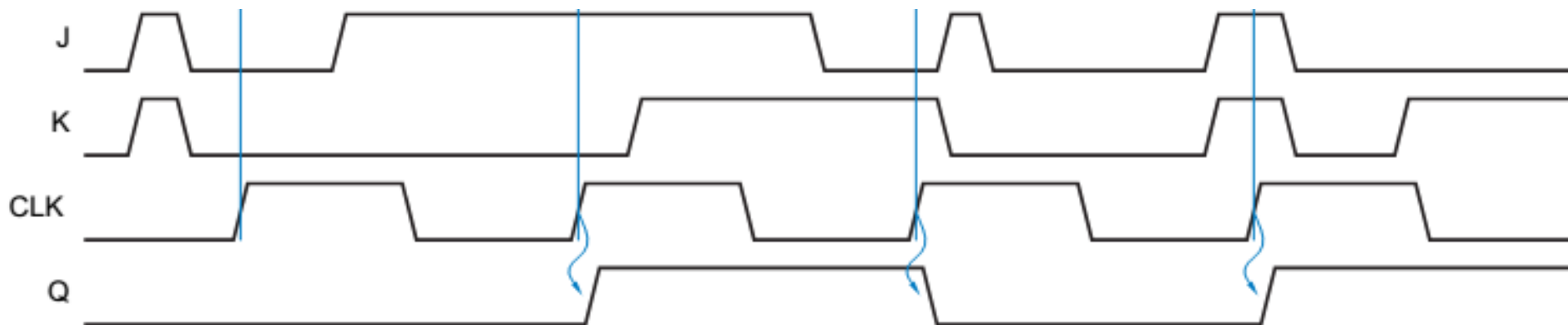
Ký hiệu

J	K	CLK	Q	QN
0	0		last Q	last QN
0	1		0	1
1	0		1	0
1	1		last QN	last Q

Bảng sự thật



Thiết kế từ FF-D kích cạnh lên



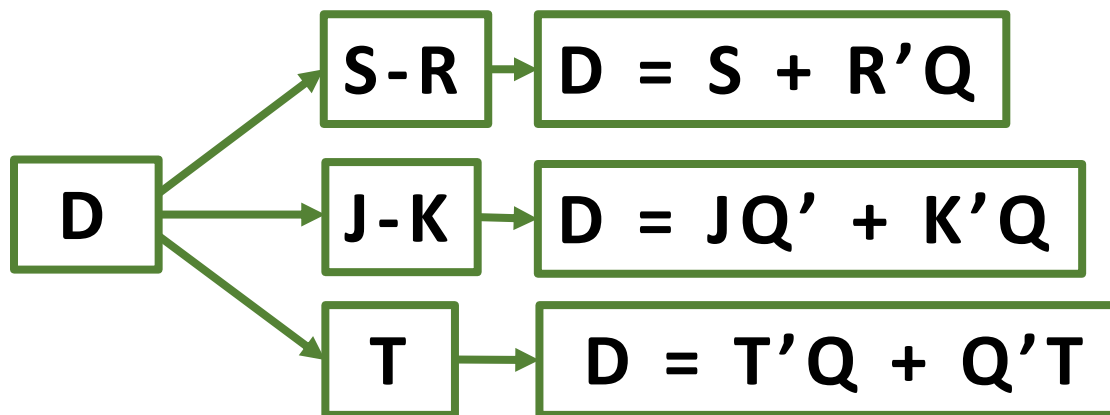
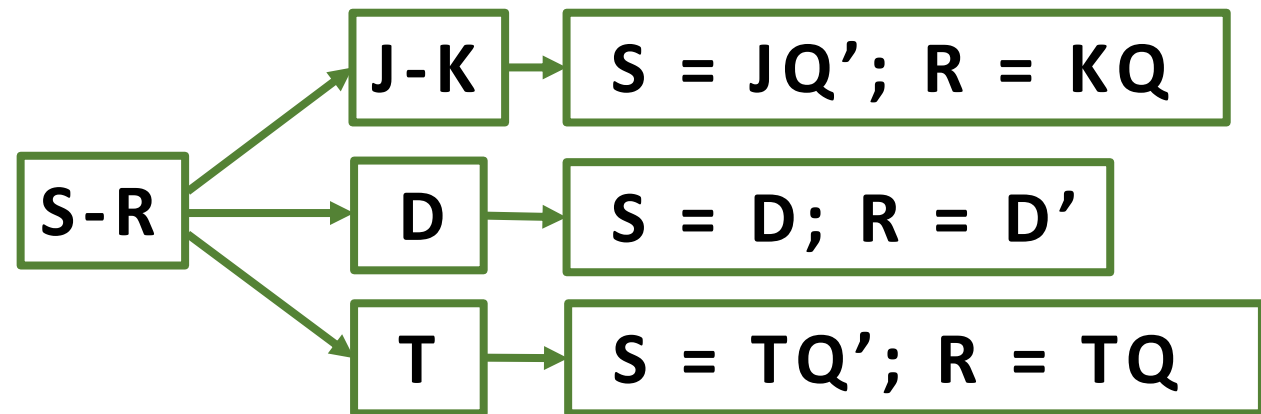
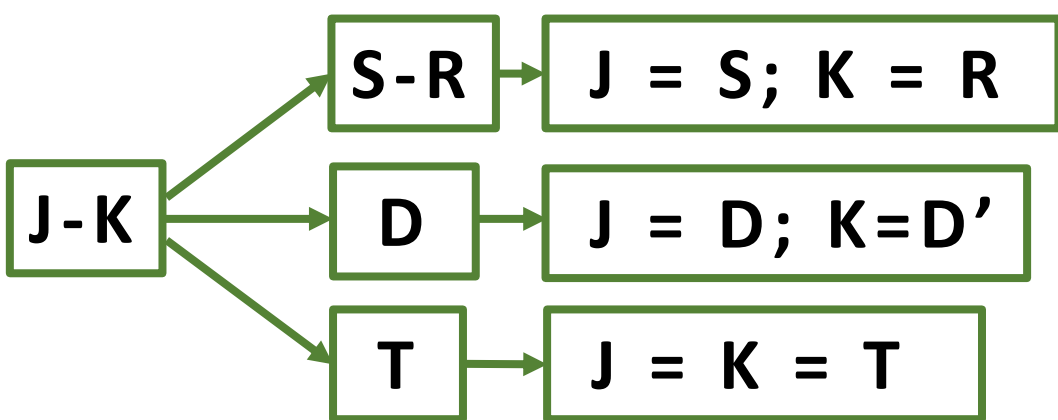
Hoạt động



2. Flip-Flop

Thiết kế chuyển đổi giữa các FF

Tổng quan





2. Flip-Flop

Thiết kế chuyển đổi giữa các FF

D-ff sang T-ff

T	Q	Q+	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0



$$D = T'Q + TQ'$$

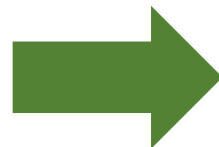


2. Flip-Flop

Thiết kế chuyển đổi giữa các FF

D-ff sang SR-ff

S	R	Q	Q+	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	X	X
1	1	1	X	X



RQ S	00	01	11	10
0	0	1	0	0
1	1	1	X	X



$$D = S + R'Q$$

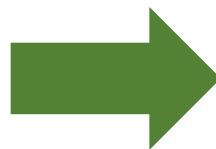
2. Flip-Flop



Thiết kế chuyển đổi giữa các FF

D-ff sang JK-ff

J	K	Q	Q+	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0



KQ					
J		00	01	11	10
0		0	1	0	0
1		1	1	0	1



$$D = JQ' + K'Q$$



2. Flip-Flop

Thiết kế chuyển đổi giữa các FF

JK-ff sang SR-ff

S	R	Q	Q+	J	K
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	X	1
1	0	0	1	1	X
1	0	1	1	X	0
1	1	0	X	X	X
1	1	1	X	X	X

RQ

S	00	01	11	10
0	0	X	X	0
1	1	X	X	X

$J = S$

RQ

S	00	01	11	10
0	X	0	1	X
1	X	0	X	X

$K = R$

2. Flip-Flop



Thiết kế chuyển đổi giữa các FF

JK-ff sang T-ff

T	Q	Q+	J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

T \ Q	0	1
0	0	X
1	1	X

$$J = T$$

T \ Q	0	1
0	X	0
1	X	1

$$K = T$$



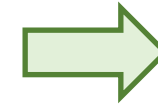
2. Flip-Flop

Thiết kế chuyển đổi giữa các FF

JK-ff sang D-ff

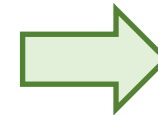
D	Q	Q+	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	0	X	0

D \ Q	0	1
0	0	X
1	1	X



$$J = D$$

D \ Q	0	1
0	X	1
1	X	0



$$K = D'$$



2. Flip-Flop

Thiết kế chuyển đổi giữa các FF

SR-ff sang JK-ff

J	K	Q	Q+	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	X
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

J \ KQ				
	00	01	11	10
0	0	X	0	0
1	1	X	0	1

$$S = J'Q$$

J \ KQ				
	00	01	11	10
0	X	0	1	X
1	0	0	1	0

$$R = KQ$$



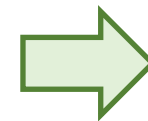
2. Flip-Flop

Thiết kế chuyển đổi giữa các FF

SR-ff sang D-ff

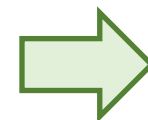
D	Q	Q+	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	0	X	0

D \ Q	0	1
0	0	0
1	1	X



$$S = D$$

D \ Q	0	1
0	X	1
1	0	0



$$S = D'$$



3. Bộ đếm bất đồng bộ

Là bộ đếm mà các FF không chung ngõ CLK

FF có trọng số thấp nhất: cổng CLK nối trực tiếp với xung

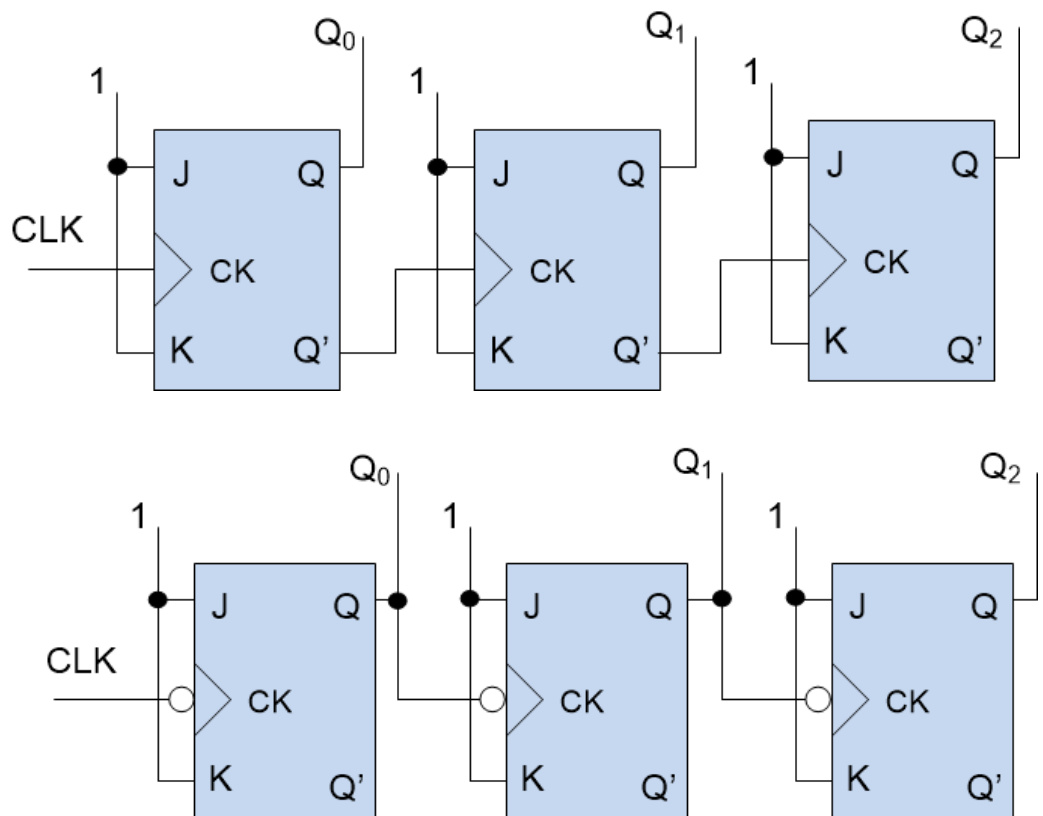
MOD (hệ số bộ đếm)

- ✓ Là số trạng thái trong một chu kì của bộ đếm
- ✓ Bộ đếm 4 bit có 16 trạng thái khác nhau $\Rightarrow \text{mod } 16$
- ✓ Số FF thay đổi sẽ làm MOD thay đổi
- ✓ $\text{MOD} \leq 2^N$ (N: số FF)

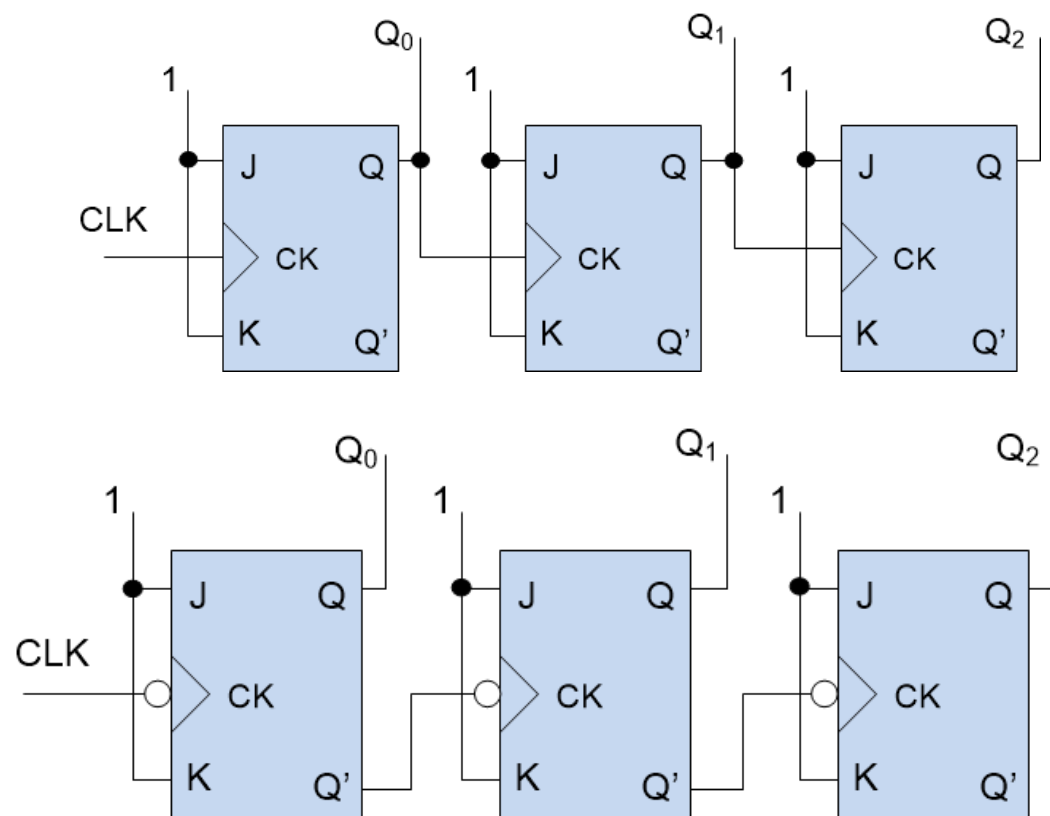
3. Bộ đếm bất đồng bộ

Một vài ví dụ bộ đếm lên/đếm xuống bất đồng bộ

Đếm lên



Đếm xuống

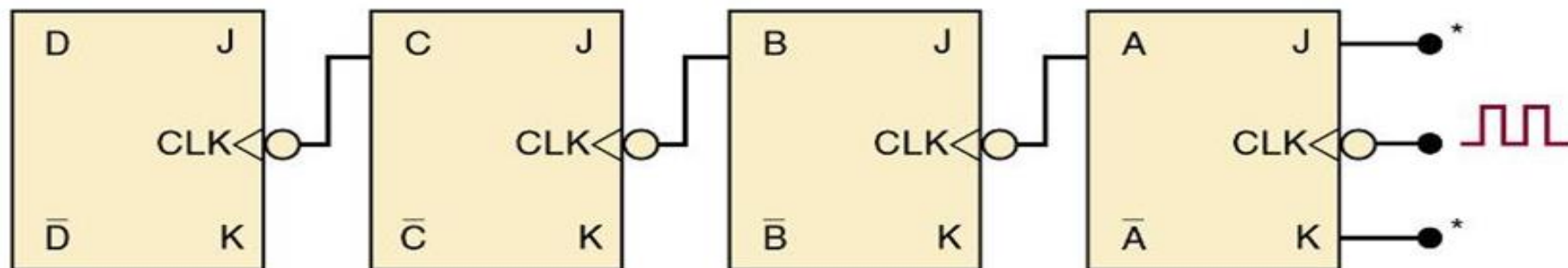


Chú ý: Q₀ có trọng số nhỏ nhất (LSB)
Q₂ có trọng số lớn nhất (MSB)

3. Bộ đếm bất đồng bộ



VD bộ đếm 4-bit



✓ Clock nối chân CLK của FF A \Rightarrow A trọng số thấp nhất

✓ J và K của tất cả FF đều bằng 1

✓ Q của FF A nối CLK của B, tương tự với B, C, D

✓ Ngõ ra Q của các FF là kết quả đếm 4-bit

J	K	CLK	Q	QN
0	0		last Q	last QN
0	1		0	1
1	0		1	0
1	1		last QN	last Q

Bảng sự thật FF-J_K

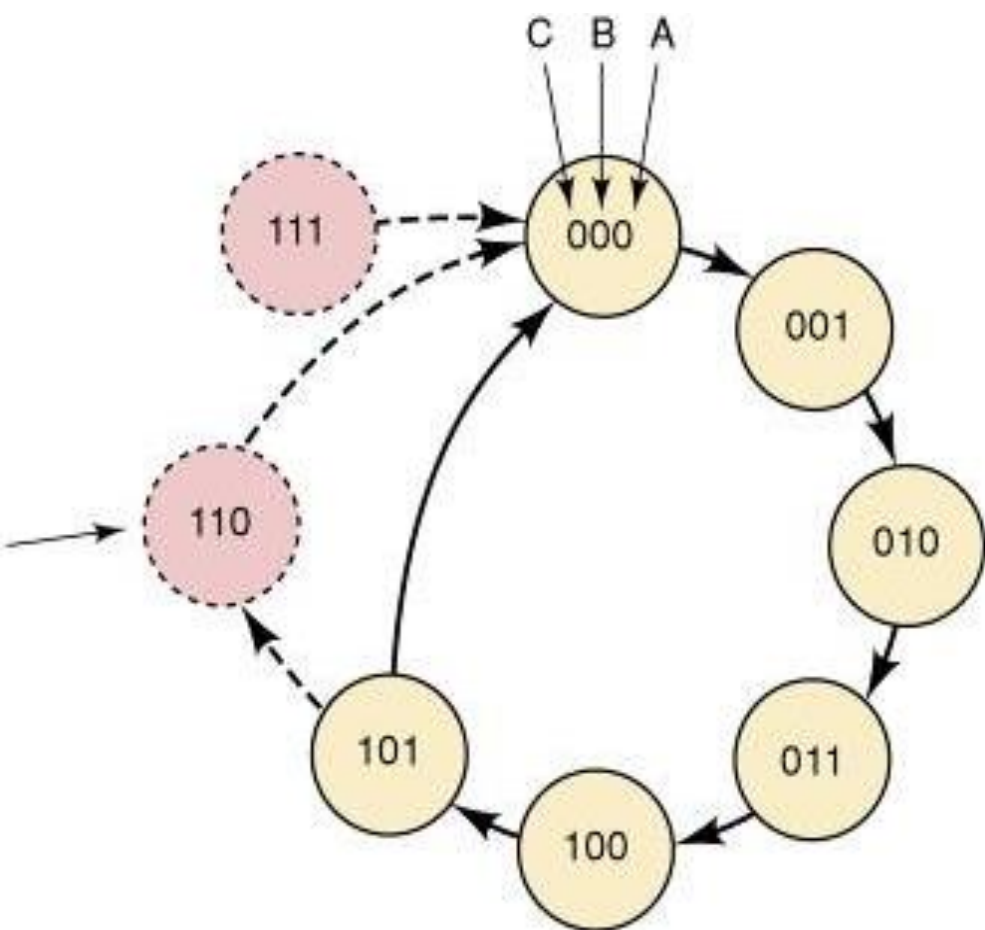
Nghĩa là, nếu ta có kết quả đếm là 10 thì biểu diễn: $Q_D Q_C Q_B Q_A = 1010_2$

3. Bộ đếm bất đồng bộ



Đối với bộ đếm có $MOD < 2^N$

VD: vẽ giản đồ chuyển trạng thái của bộ đếm MOD-6 (đếm từ 0 đến 5)



Vòng tròn nét liền: trạng thái thực sự

Vòng tròn nét đứt: trạng thái tạm

Mũi tên nét liền: chuyển trạng thái giữa 2 trạng thái thực

Mũi tên nét đứt: chuyển từ trạng thái thực -> trạng thái tạm hoặc ngược lại



3. Bộ đếm bất đồng bộ

Delay

Delay của FF trước được tích lũy đến FF sau

→ Delay của toàn mạch khá lớn

→ không phù hợp cho các mạch hoạt động ở tần số cao hoặc khi dùng nhiều FF

Để mạch hoạt động đúng thì $T_{\text{clock}} \geq N \times t_{\text{pd}}$

T_{clock} : chu kì xung Clock

N : số FF của mạch

T_{pd} : delay của một FF

→ Tần số tối đa của mạch: $F_{\text{max}} = 1 / (N \times t_{\text{pd}})$

→ Tần số ngõ ra: : $F_{\text{raqp}} = \text{Tần số xung} / \text{Số MOD}$

Tần số xung = Tần số ngõ vào tại FF đầu



4. Bộ đếm đồng bộ (Bộ đếm song song)

Là bộ đếm mà tín hiệu clock được kết nối tới ngõ vào CLK của tất cả các FF \rightarrow delay của mạch = delay của mỗi FF

Khác với bộ đếm bất đồng bộ, **bộ đếm đồng bộ** có thể được thiết kế để tạo ra ***chuỗi đếm bất kì***. *Thuộc tính* đếm lên/xuống ***chỉ phụ thuộc*** vào **trạng thái hiện tại** và **trạng thái kế tiếp** mà *không quan tâm* đến FF kích cạnh lên/xuống

4. Bộ đếm đồng bộ (Bộ đếm song song)

Phân tích mạch đếm

Ví dụ: Phân tích mạch đếm ở hình bên dưới

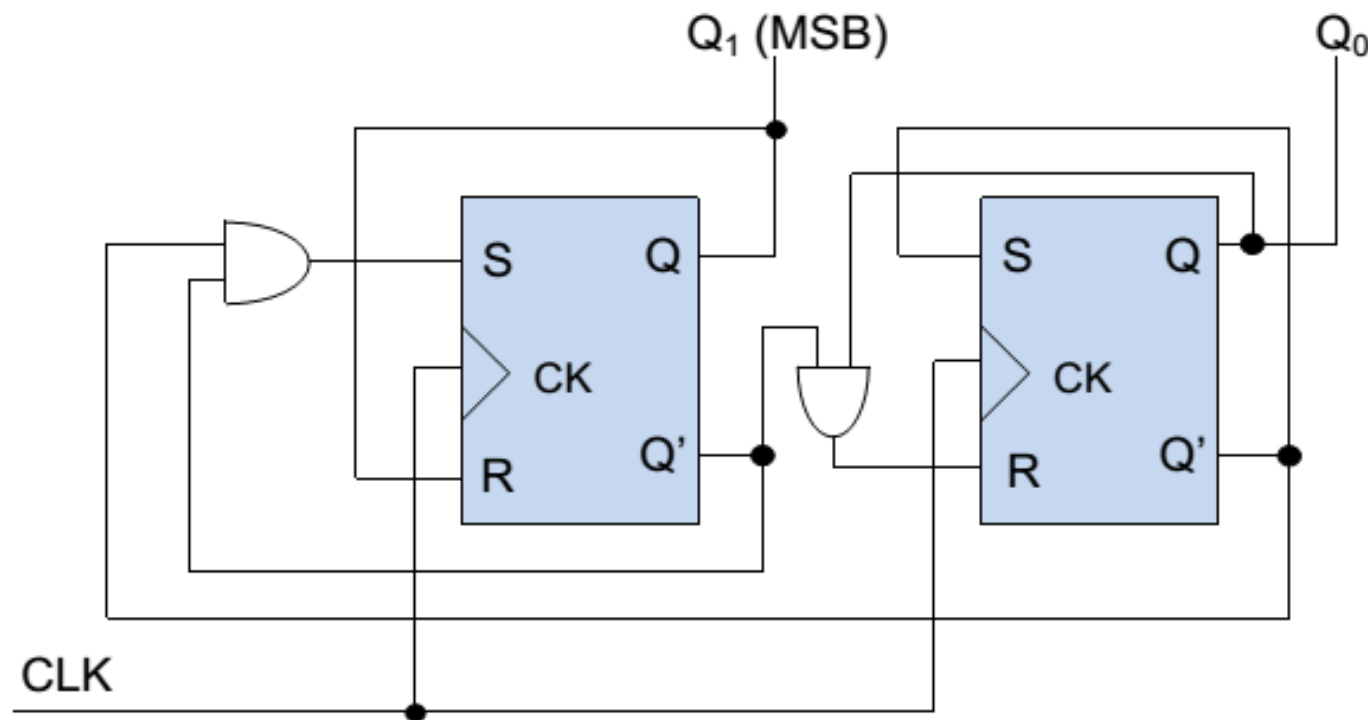
Bước 1: Tìm phương trình ngõ vào của các FF

$$S_1 = Q'_1 Q'_0$$

$$S_0 = Q'_0$$

$$R_1 = Q_1$$

$$R_0 = Q'_1 Q_0$$



4. Bộ đếm đồng bộ (Bộ đếm song song)

Phân tích mạch đếm

Bước 2: Lập bảng chuyển trạng thái

$$S_1 = Q'_1 Q'_0$$

$$R_1 = Q_1$$

$$S_0 = Q'_0$$

$$R_0 = Q'_1 Q_0$$

S	R	CLK	Q	Q'
0	0	f	last Q	last Q'
0	1	f	0	1
1	0	f	1	0
1	1	f	x	x

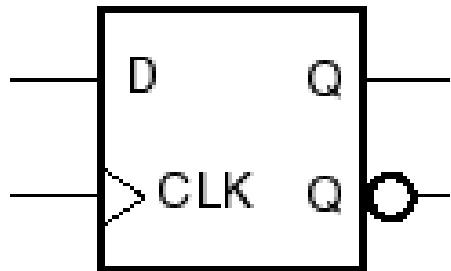
Bảng sự thật FF-S_R

Bảng chuyển trạng thái

TTHT $Q_1 Q_0$	Ngõ vào $S_1 R_1 \quad S_0 R_0$	TTKT $Q_1^+ Q_0^+$
0 0		
0 1		
1 0		
1 1		

4. Bộ đếm đồng bộ (Bộ đếm song song)

Các kiểu mô tả của FF-D



Ký hiệu

D	CLK	Q	Q _N
0		0	1
1		1	0

Bảng sự thật

D	Q _(next)
0	0
1	1

Bảng đặc tính

Q	Q _(next)	D
0	0	0
0	1	1
1	0	0
1	1	1

Bảng kích thích

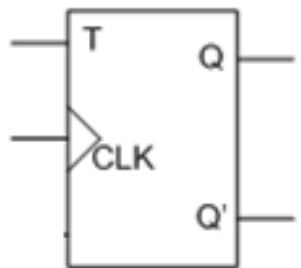
$$Q_{(next)} = D$$

D	Q	Q ⁺
0	0	0
0	1	0
1	0	1
1	1	1

Phương trình đặc tính

4. Bộ đếm đồng bộ (Bộ đếm song song)

Các kiểu mô tả của FF-T



Ký hiệu

T	CLK	Q	Q'
0	f	last Q	last Q'
1	f	Q'	Q

Bảng sự thật

Q	Q _(next)	T
0	0	0
0	1	1
1	0	1
1	1	0

Bảng kích thích

T	Q _(next)
0	Q
1	Q'

Bảng đặc tính

T	Q	Q ⁺
0	0	0
0	1	1
1	0	1
1	1	0

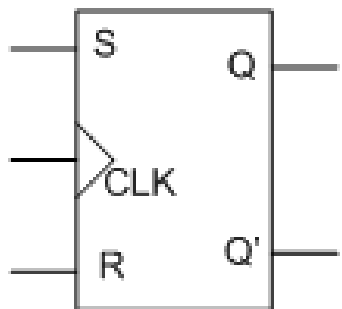
$$Q_{(next)} = TQ' + T'Q$$

$$Q^+ = T \oplus Q$$

Phương trình đặc tính

4. Bộ đếm đồng bộ (Bộ đếm song song)

Các kiểu mô tả của FF-SR



Ký hiệu

S	R	CLK	Q	Q'
0	0	f	last Q	last Q'
0	1	f	0	1
1	0	f	1	0
1	1	f	x	x

Bảng sự thật

S	R	Q _(next)
0	0	Q
0	1	0
1	0	1
1	1	?

Q	Q _(next)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Bảng kích thích

S	R	Q	Q ⁺
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

Bảng đặc tính

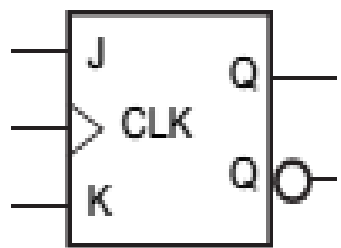
$$Q_{(next)} = S + R'Q$$

Phương trình đặc tính



4. Bộ đếm đồng bộ (Bộ đếm song song)

Các kiểu mô tả của FF-JK



Ký hiệu

J	K	CLK	Q	Q _N
0	0		last Q	last Q _N
0	1		0	1
1	0		1	0
1	1		last Q _N	last Q

Bảng sự thật

Q	Q _(next)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Bảng kích thích

J	K	Q _(next)
0	0	Q
0	1	0
1	0	1
1	1	Q'

J	K	Q	Q ⁺
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Bảng đặc tính

$$Q_{(next)} = JQ' + K'Q$$

Phương trình
đặc tính

4. Bộ đếm đồng bộ (Bộ đếm song song)

Thiết kế bộ đếm đồng bộ

VD: Sử dụng FF-J_K để thiết kế một bộ đếm có chuỗi đếm như bảng

□ Bước 1: Tìm số FF nhỏ nhất

Chu trình đếm 0-1-2-3-4-0-... $\Rightarrow \text{MOD} = 5$

Mà $\text{MOD} = < 2^N \Rightarrow N = 3$

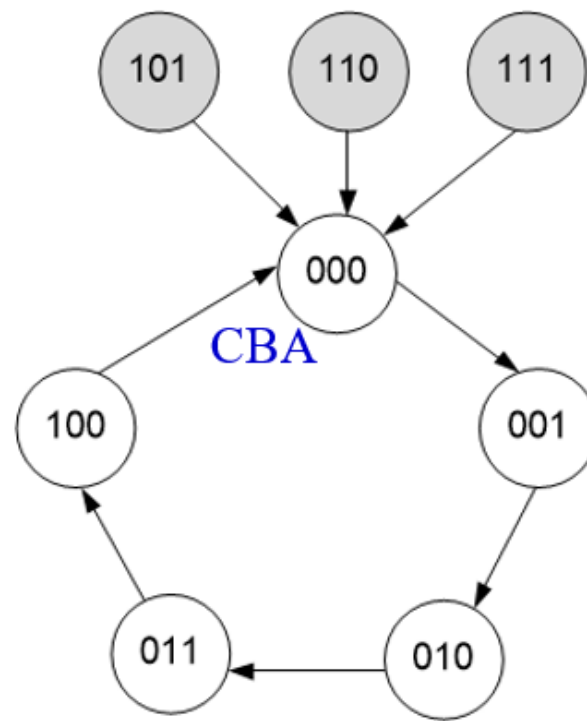
C	B	A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
0	0	0
etc.		

4. Bộ đếm đồng bộ (Bộ đếm song song)

Thiết kế bộ đếm đồng bộ

❑ Bước 2: Vẽ biểu đồ chuyển trạng thái

- ✓ Vẽ tất cả các trạng thái có thể
- ✓ Những trạng thái không có trong chu trình đếm, có thể cho chuyển đến một trạng thái trong chu trình đếm



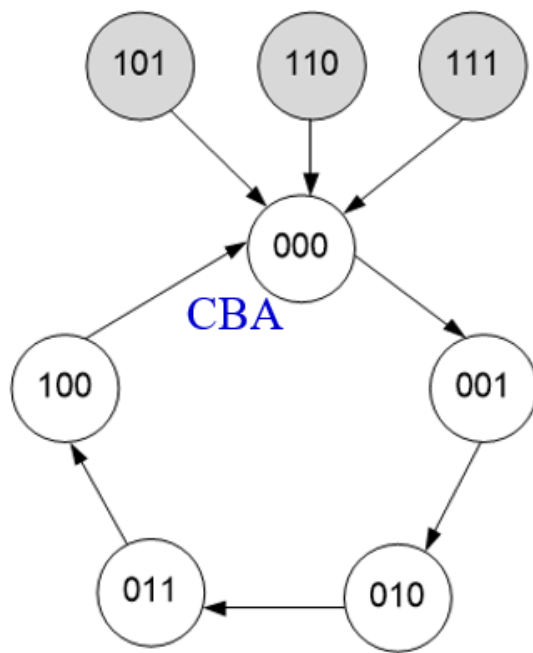
C	B	A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
0	0	0
etc.		

4. Bộ đếm đồng bộ (Bộ đếm song song)

Thiết kế bộ đếm đồng bộ

❑ Bước 3: Lập bảng trạng thái

Dùng biểu đồ chuyển trạng thái để lập một bảng bao gồm các trạng thái hiện tại và trạng thái kế



CLK	TTHT				TTKT		
	C	B	A		C ⁺	B ⁺	A ⁺
1	0	0	0				
2	0	0	1				
3	0	1	0				
4	0	1	1				
5	1	0	0				
6	1	0	1				
7	1	1	0				
8	1	1	1				

Bảng trạng thái của mạch

4. Bộ đếm đồng bộ (Bộ đếm song song)

Thiết kế bộ đếm đồng bộ

❑ Bước 4: Lập bảng kích thích của mạch

Dựa vào trạng thái hiện tại và trạng thái kế tiếp, thêm các cột giá trị ngõ vào mỗi FF vào bên phải bảng chuyển trạng thái

CLK	TTHT				TTKT				Ngõ vào các FF					
	C	B	A		C ⁺	B ⁺	A ⁺		J _C	K _C	J _B	K _B	J _A	K _A
1	0	0	0											
2	0	0	1											
3	0	1	0											
4	0	1	1											
5	1	0	0											
6	1	0	1											
7	1	1	0											
8	1	1	1											

Bảng kích thích của mạch

4. Bộ đếm đồng bộ (Bộ đếm song song)

Thiết kế bộ đếm đồng bộ

- ❑ **Bước 5:** Sử dụng bìa K để tìm *phương trình ngõ vào* của các FF

J_C

CB	00	01	11	10
A				
0			x	x
1		1	x	x

BA

$$J_C = BA$$

J_B

CB	00	01	11	10
A				
0		x	x	
1		1	x	

$C'A$

$$J_B = C'A$$

J_A

CB	00	01	11	10
A				
0	1	1		
1	x	x	x	x

C'

$$J_A = C'$$

K_C

CB	00	01	11	10
A				
0	x	x	1	1
1	x	x	1	1

1

$$K_C = 1$$

K_B

CB	00	01	11	10
A				
0	x		1	x
1	x	1	1	x

A C

$$K_B = A + C$$

K_A

CB	00	01	11	10
A				
0	x	x	x	x
1	1	1	1	1

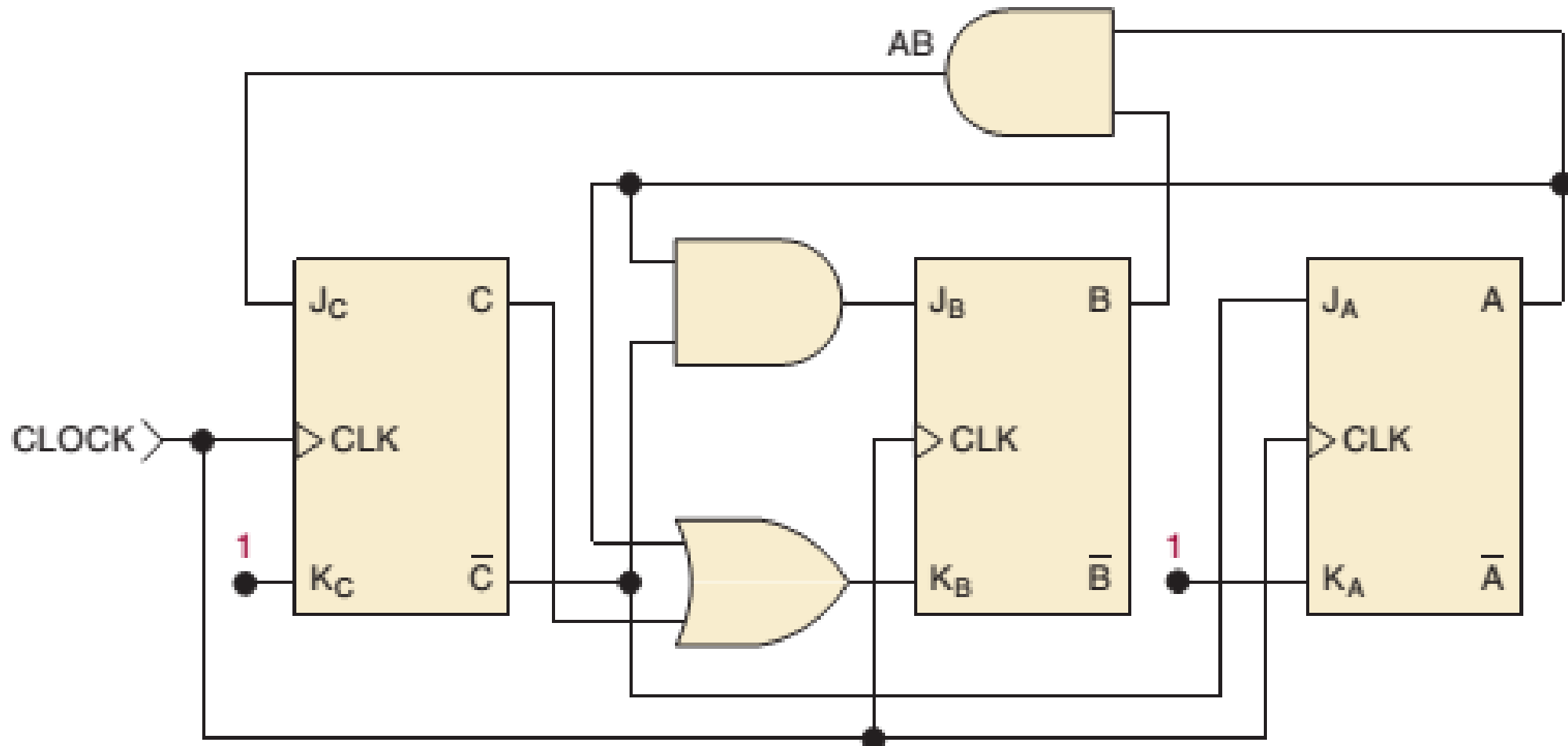
1

$$K_A = 1$$

4. Bộ đếm đồng bộ (Bộ đếm song song)

Thiết kế bộ đếm đồng bộ

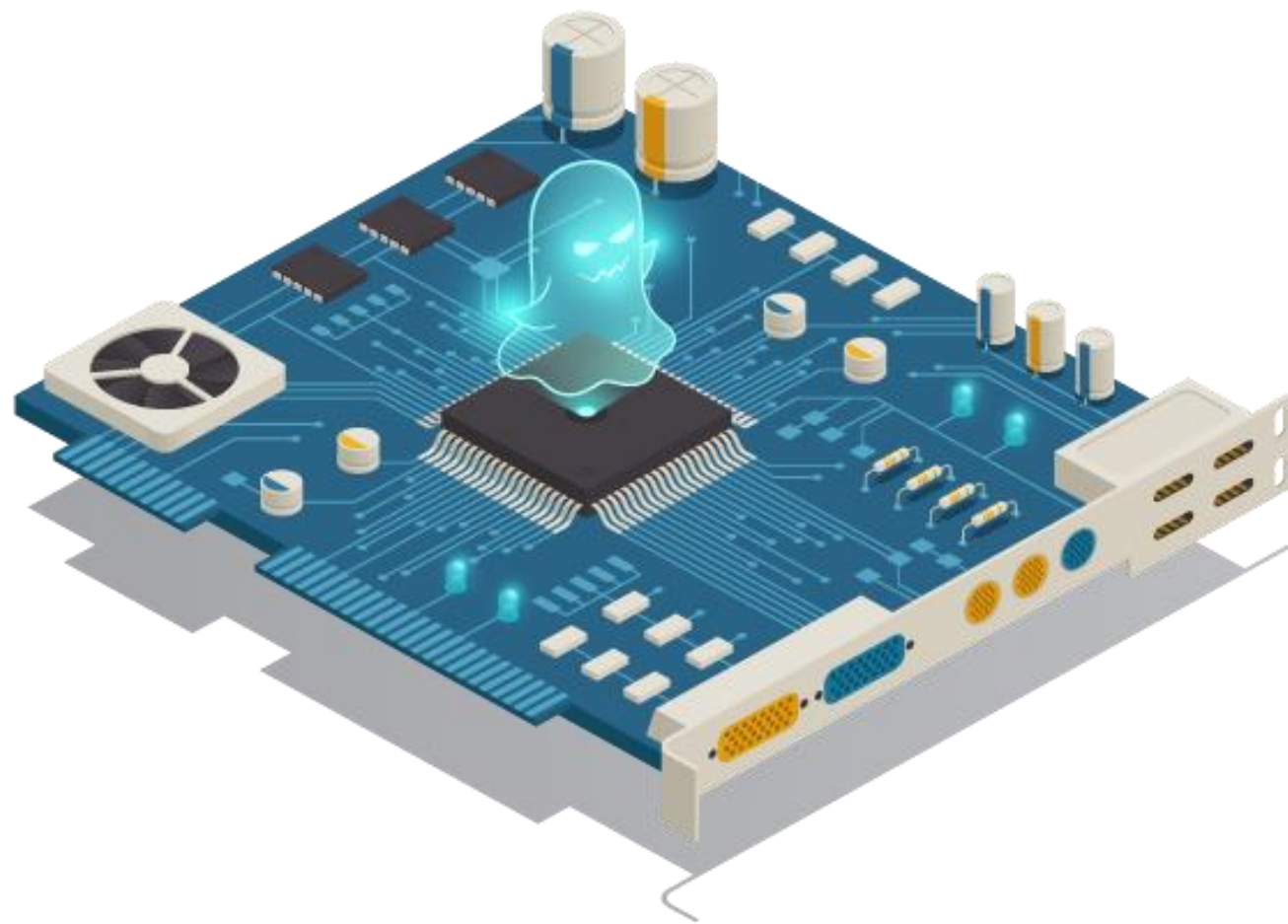
❑ Bước 6: Vẽ mạch



4. Bộ đếm đồng bộ (Bộ đếm song song)

So sánh hai bộ đếm

Bộ đếm bất đồng bộ	Bộ đếm đồng bộ
Kích lần lượt các xung CLK	Kích các xung CLK cùng 1 lúc
Đếm theo thứ tự lên/xuống	Đếm bất kì không theo thứ tự
Các trạng thái của bộ đếm không được thay đổi đồng thời	Cùng lúc các trạng thái của bộ đếm đều thay đổi



CỦNG CỐ KIẾN THỨC



Bài tập

Câu 1

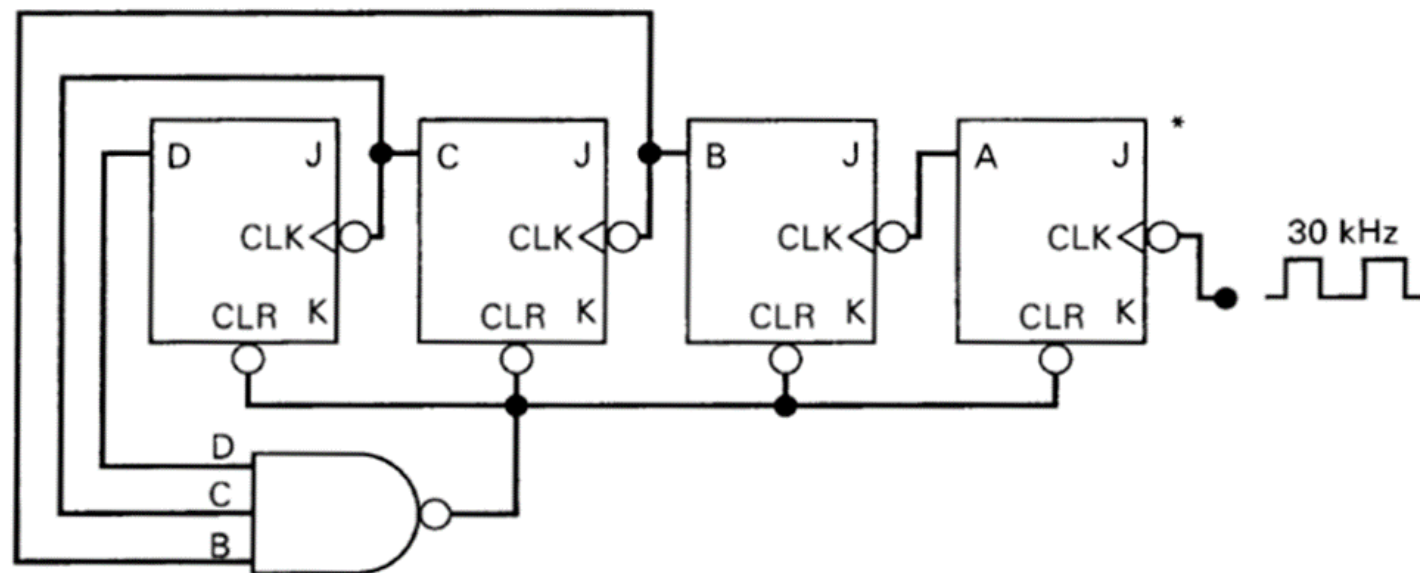
Xác định hệ số bộ đếm (MOD number) của mạch dưới:

A. 16

B. 12

C. 14

D. 10



* Khi D C B bằng 1 (đếm tới 1110 -14) thì clear về giá trị đầu.

* Bộ đếm đếm từ 0-13 => có 14 MOD



Bài tập

Câu 2

Một bộ đếm bất đồng bộ Ripple Counter được xây dựng từ các Flip Flop JK có thời gian trì hoãn của mỗi FF $t_{pd}=12ns$, hệ số MOD ít nhất của bộ đếm mà bộ đếm vẫn hoạt động được ở tần số 10MHz là:

A. 1024

☒ B. 256

C. 512

D. 64

** Để mạch hoạt động đúng thì chu kỳ của xung clock phải lớn hơn tổng Delay của mạch: $T_{clock} \geq N * t_{pd}$*

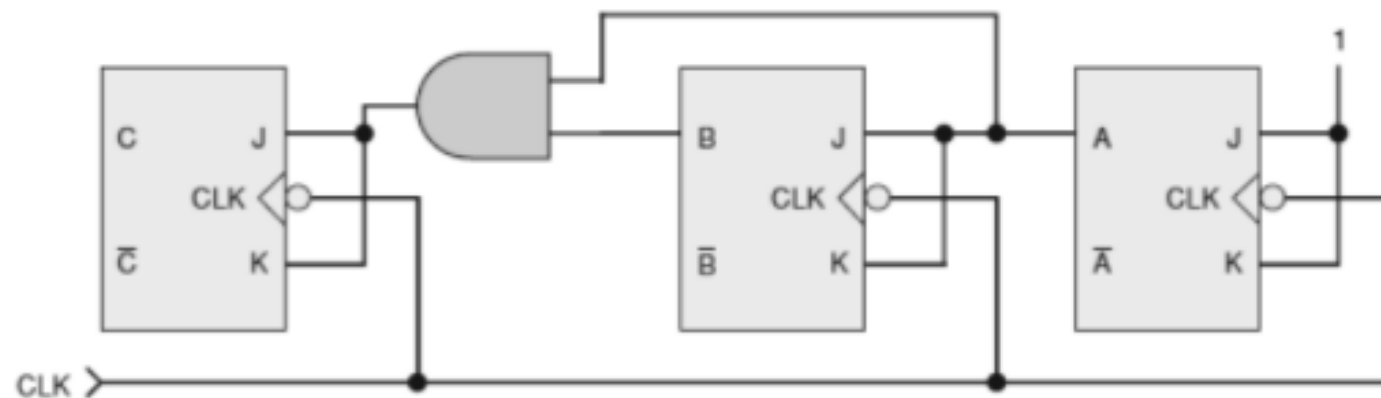


Bài tập

Câu 3

Cho bộ đếm đồng bộ như hình bên dưới, chuỗi đếm lặp vòng của bộ đếm là:

- A. 0,1,2,3,4,5,6
- B. 6,5,4,3,2,1,0
- C. 0,1,2,3,4,5,6,7**
- D. 7,6,5,4,3,2,1,0



***Phương trình đầu vào :**

$$J_0 = 1 \quad J_1 = A \quad J_2 = A.B$$

$$K_0 = 1 \quad K_1 = A \quad K_2 = A.B$$

***chuyển trạng thái:**

1 → 2 → 3 → 4 → 5 → 6 → 7 → 0 → 1 →

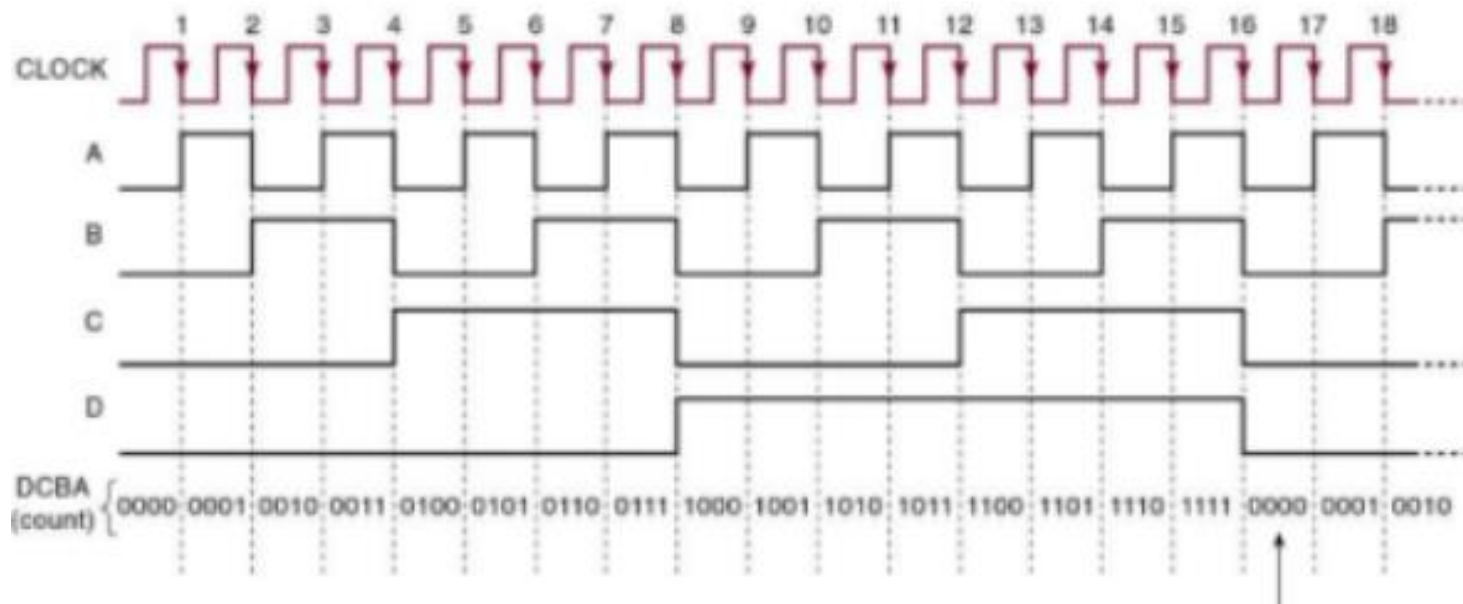


Bài tập

Câu 4

Cho giản đồ xung như hình bên dưới, Giá trị của DCBA ngay sau cạnh xuống xung clock thứ 2022 là:

- ☒ A. 0110
- ☐ B. 0011
- ☐ C. 0010
- ☐ D. 1100



- ✓ bộ đếm có 16 trạng thái $\Rightarrow \text{MOD } 16$
- ✓ $2022/16 = 126 \text{ dư } 6 \Rightarrow \text{trạng thái } 0110$



Bài tập

Câu 5

Phát biểu nào sau đây về mạch tuần tự và tổ hợp là sai :

- A. Mạch tổ hợp là mạch có ngõ ra thay đổi ngay lập tức khi ngõ vào thay đổi
- ☒ B. Mạch tuần tự là mạch có ngõ ra thay đổi ngay lập tức khi ngõ vào thay đổi
- C. Mạch tuần tự là mạch có thể chứa các thành phần là mạch tổ hợp
- D. Chốt là một phần tử có tính chất nhớ



Bài tập

Câu 6

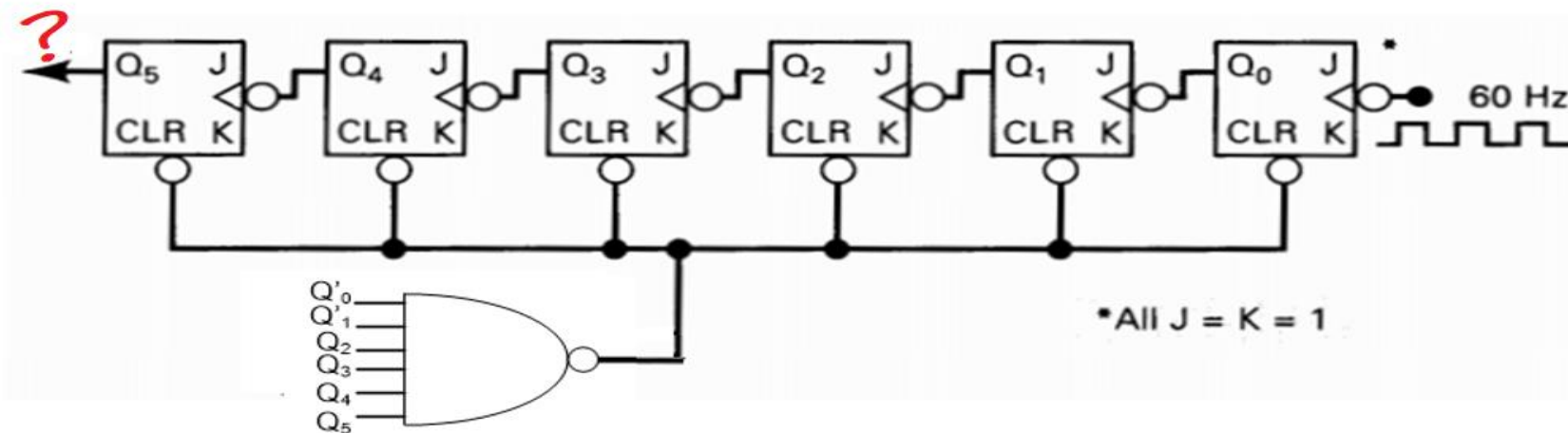
Tính tần số ngõ ra của bộ đếm dưới đây:

A. 4 Hz

B. 1 Hz

C. 5 Hz

D. 3.5 Hz





ASK & ANSWER

THANK
THANK YOU