

# NHẬP MÔN MẠCH SỐ

## CHƯƠNG 6 – PHẦN 2

### Mạch tuần tự: Bộ đếm (Sequential circuit: Counters)



cuu duong than cong . com

# Nội dung

- **Bộ đếm bất đồng bộ (Asynchronous counters)**
  - Hệ số của bộ đếm (MOD number)
  - Bộ đếm lên/xuống (Up/ Down counters)
  - Phân tích và thiết kế bộ đếm bất đồng bộ
  - Delay của mạch (Propagation delay)
- **Bộ đếm đồng bộ (Synchronous counters)**
  - Phân tích bộ đếm đồng bộ (Analyze synchronous counters)
  - Thiết kế bộ đếm đồng bộ (Design synchronous counter)
- **Thanh ghi (Register)**

# Nội dung

- **Bộ đếm bất đồng bộ (Asynchronous counters)**
  - Hệ số của bộ đếm (MOD number)
  - Bộ đếm lên/xuống (Up/ Down counters)
  - Phân tích và thiết kế bộ đếm bất đồng bộ
  - Delay của mạch (Propagation delay)
- **Bộ đếm đồng bộ (Synchronous counters)**
  - Phân tích bộ đếm đồng bộ (Analyze synchronous counters)
  - Thiết kế bộ đếm đồng bộ (Design synchronous counter)
- **Thanh ghi (Register)**



# Bộ đếm bất đồng bộ

(Asynchronous counters)





cuu duong than cong . com

cuu duong than cong . com

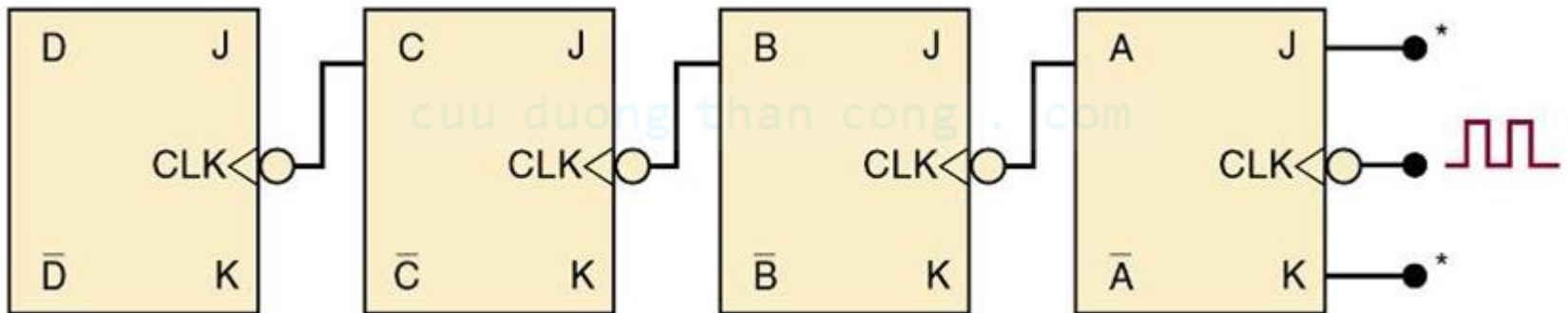
# Bộ đếm bất đồng bộ

Xem xét hoạt động của bộ đếm 4-bit bên dưới

- Clock chỉ được kết nối đến chân CLK của **FF A**
- *J và K của tất cả FF đều bằng 1*
- Ngõ ra Q của FF A kết nối với chân CLK của FF B, tiếp tục kết nối như vậy với FF C, D.
- Ngõ ra của các FF **D, C, B và A** tạo thành bộ đếm 4-bit binary với D có trọng số cao nhất (MSB)

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

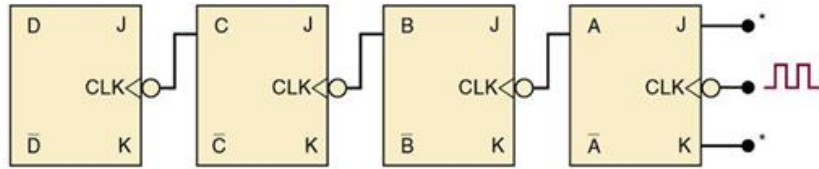
Bảng sự thật FF-J\_K



**Note:** \* tất cả ngõ vào J và K của các FF được đưa vào mức 1

# Bộ đếm bất đồng bộ

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

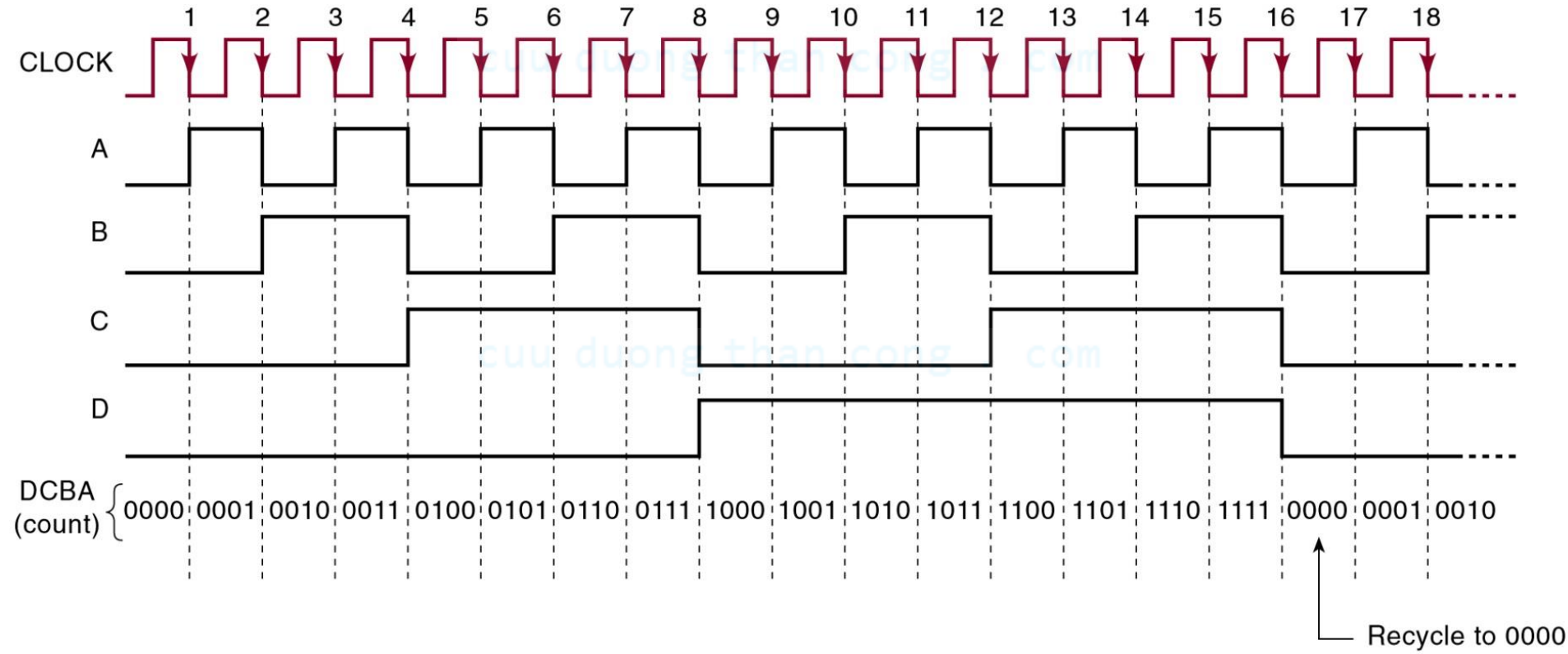


\* tất cả ngõ vào J và K của các FF được đưa vào mức 1

Sau cạnh xuống của xung *CLK* thứ 16, bộ đếm sẽ quay trở lại trạng thái ban đầu

DCBA = 0000

Bảng sự thật FF-J\_K



# Bộ đếm bất đồng bộ

- Các FFs không thay đổi trạng thái đồng bộ với xung Clock  
Trong ví dụ ở slide trước,  
Chỉ FF A mới thay đổi tại cạnh xuống của xung Clock ,  
FF B phải đợi FF A thay đổi trạng thái trước khi nó có thể lật,  
FF C phải đợi FF B thay đổi, tương tự với FF D phải đợi FF C  
→ Có trì hoãn (delay) giữa các FF liên tiếp nhau
- Chỉ FF có trọng số thấp nhất mới kết nối với xung Clock
- Bộ đếm trên còn được gọi là *bộ đếm tích lũy trì hoãn* (**ripple counter**)

# Ví dụ 1

- Giả sử bộ đếm ở Slide trước bắt đầu ở trạng thái **DCBA = 0000**, sau đó xung Clock được đưa vào
- Sau một khoảng thời gian, ta ngắt xung Clock với mạch và đọc được giá trị của bộ đếm **DCBA = 0011**
- Hỏi bao nhiêu xung Clock đã được đưa vào bộ đếm?

cuu duong than cong . com



# Duty cycle của một tín hiệu (xung)

**Duty cycle** của một xung là tỉ lệ phần trăm của thời gian xung tích cực với chu kì của xung

$$\text{duty cycle} = \frac{T_{\text{ON}}}{T_{\text{ON}} + T_{\text{OFF}}} \times 100\%$$

**Ví dụ**: giá trị duty cycle (mức 1) của xung

50% duty cycle



75% duty cycle

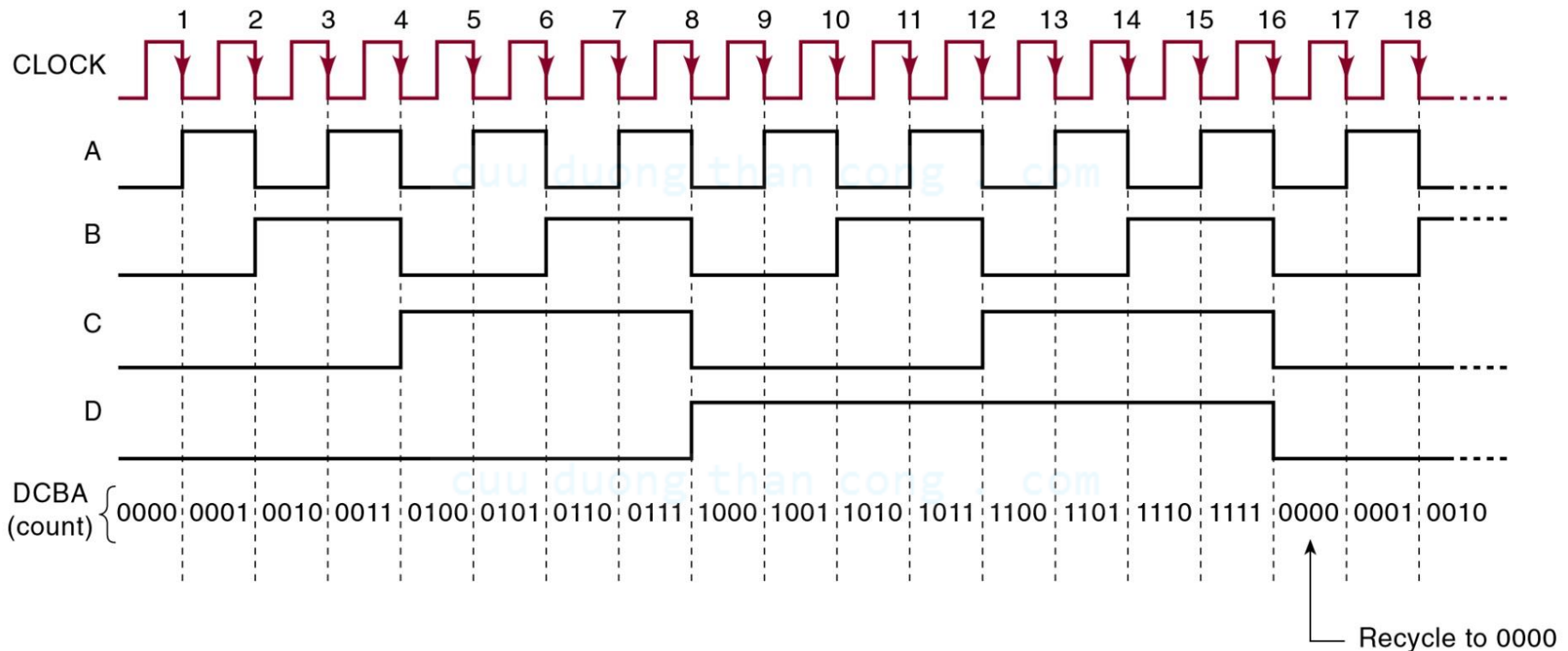


25% duty cycle



# Hệ số của bộ đếm (MOD number)

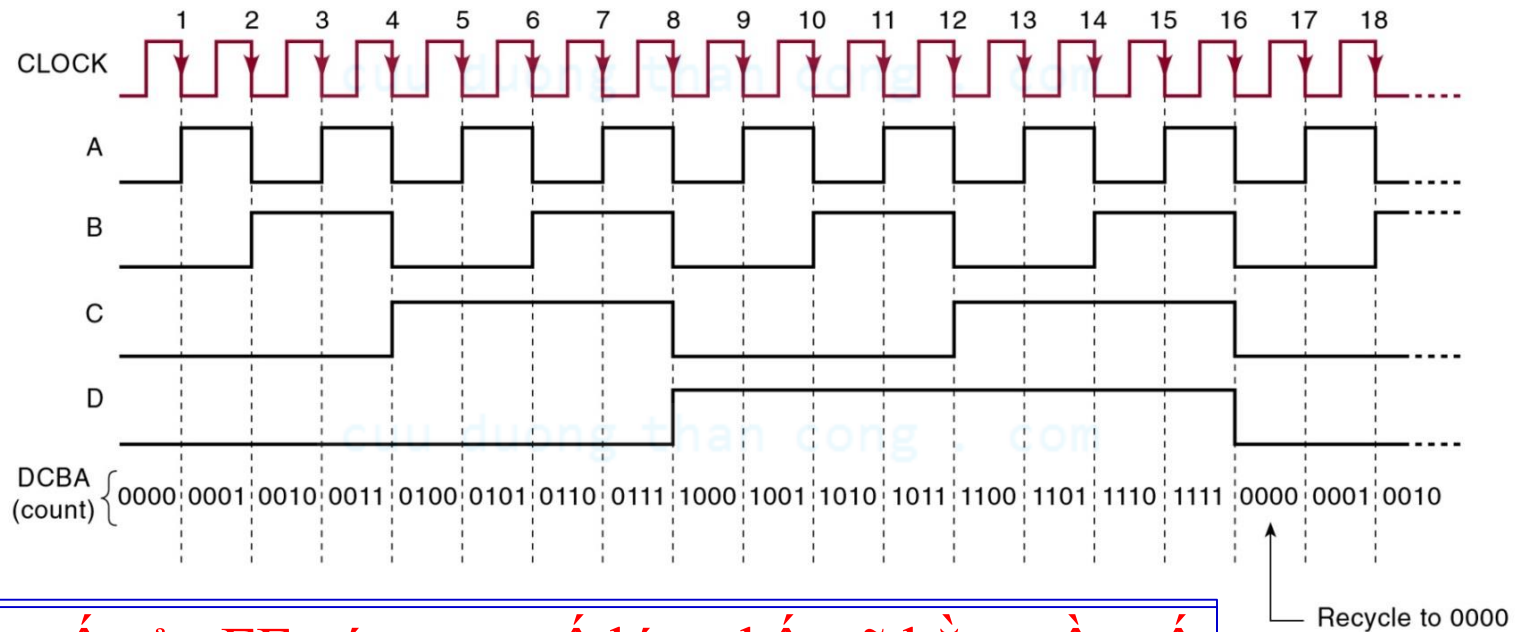
- **Hệ số của bộ đếm** là *số trạng thái khác nhau* của bộ đếm trước khi bộ đếm lặp lại chu trình đếm



Thêm vào Flip-flop sẽ tăng hệ số của bộ đếm

# Hệ số của bộ đếm (MOD number) (tt)

- **Chia tần số** – mỗi FF sẽ có tần số ngõ ra bằng  $\frac{1}{2}$  tần số của xung đưa vào chân Clock của FF đó  
Giả sử tần số của xung Clock đưa vào bộ đếm trong ví dụ 1 là 16 kHz  
→ Tần số của ngõ ra FF A, B, C, D lần lượt là 8, 4, 2, 1 kHz



Tần số của FF có trọng số lớn nhất sẽ bằng tần số xung Clock chia cho hệ số của bộ đếm

# Ví dụ 2

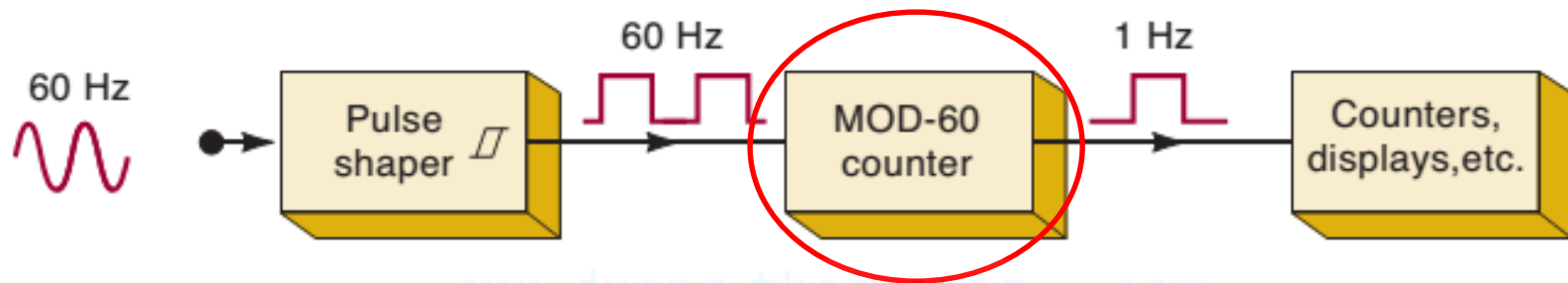
- Cần bao nhiêu FF cho bộ đếm 1000 sản phẩm?

cuu duong than cong . com

cuu duong than cong . com

# Ví dụ 3

- Các bước để làm một đồng hồ số



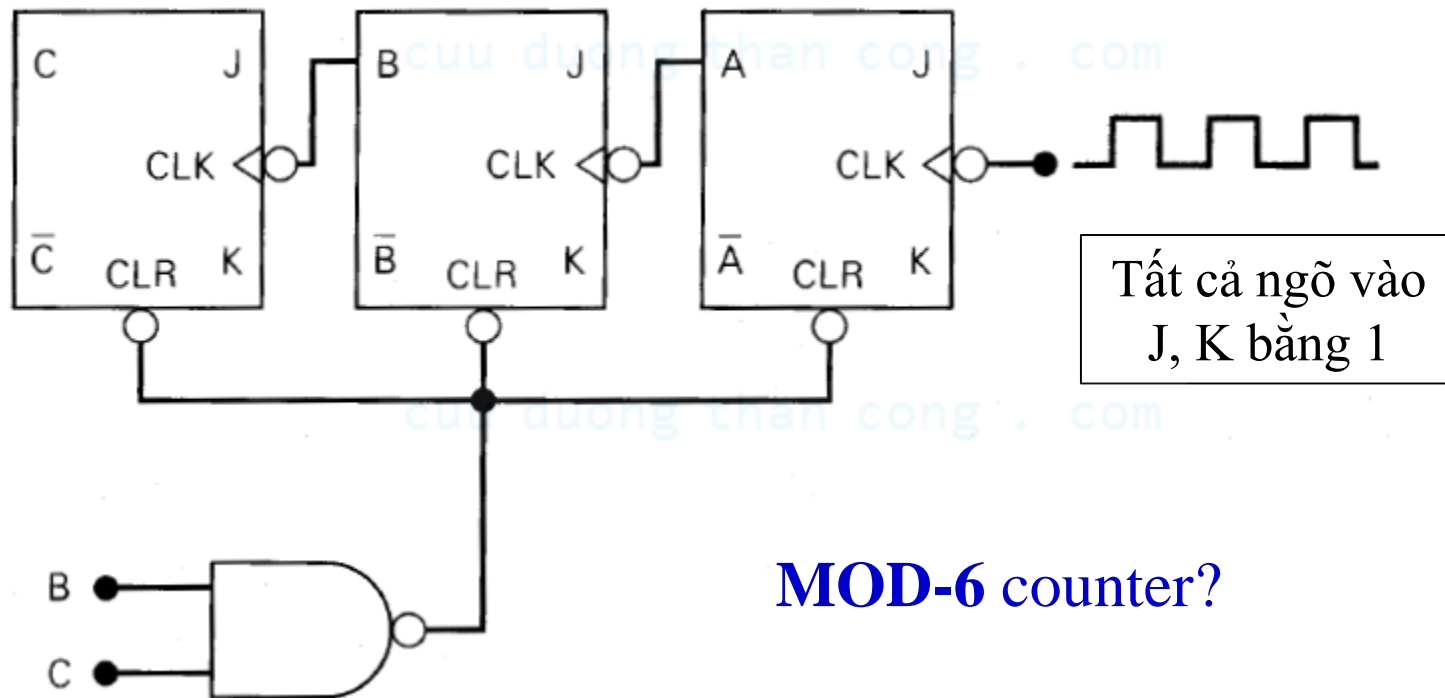
- Cần bao nhiêu FF cho bộ đếm có hệ số đếm 60 (MOD-60)?

# Câu hỏi thảo luận

1. Đúng hay sai? Trong một bộ đếm bất đồng bộ, tất cả các FF thay đổi trạng thái tại cùng một thời điểm
2. Giả sử bộ đếm trong ví dụ 1 đang có giá trị DCBA = 0101. Giá trị bộ đếm sẽ bằng bao nhiêu sau 27 xung clock tiếp theo?
3. Hệ số bộ đếm trong ví dụ 1 bằng bao nhiêu nếu 3 FF được thêm vào bộ đếm?

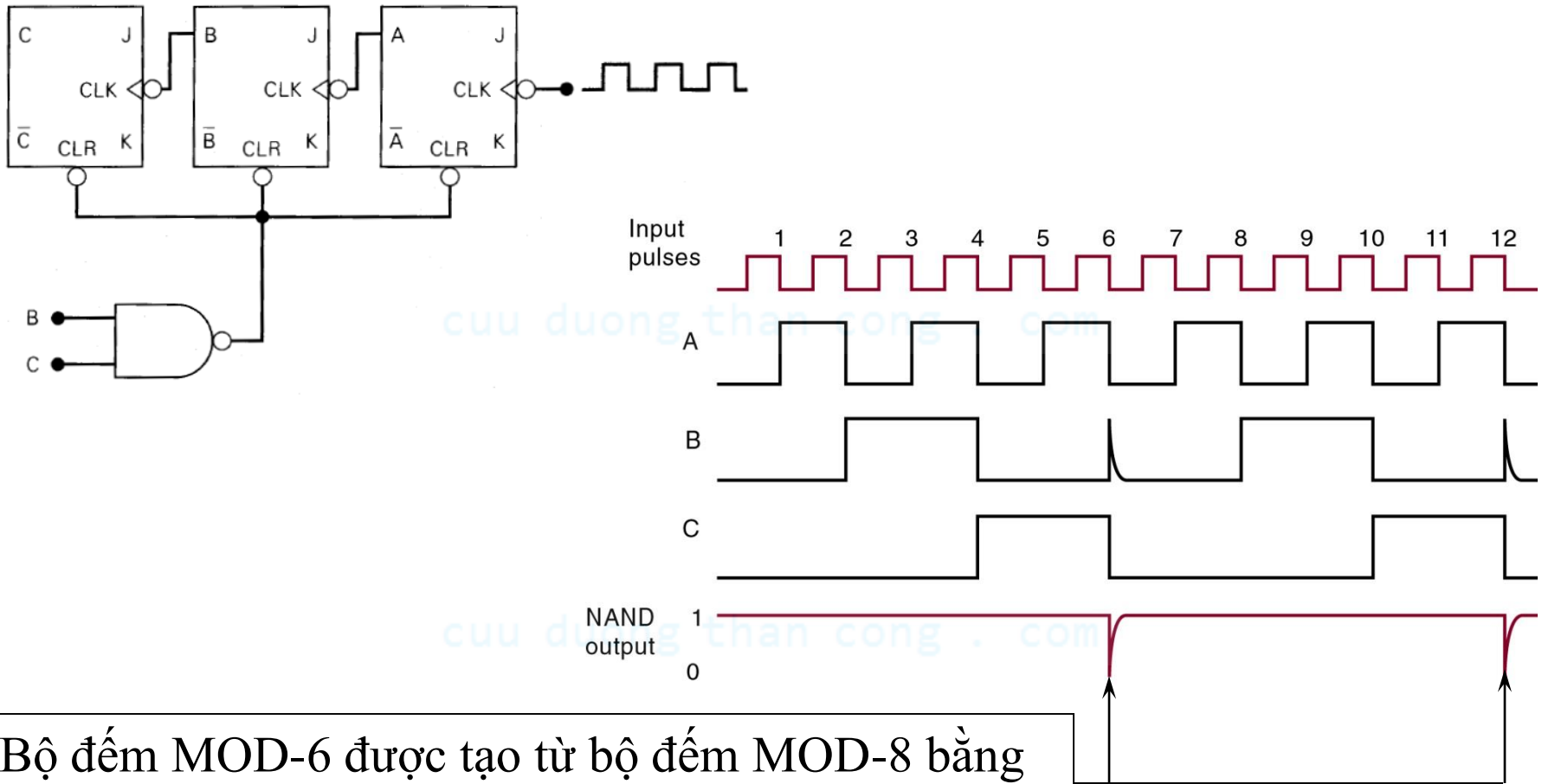
# Bộ đếm có **Hệ số bộ đếm** $< 2^N$

- Bộ đếm bất đồng bộ thông thường giới hạn hệ số bộ đếm bằng  $2^N$  (Hệ số đếm lớn nhất với  $N$  flip-flop được sử dụng)
- Xét bộ đếm với mạch cho bên dưới



**MOD-6 counter?**

# Bộ đếm có Hệ số bộ đếm $< 2^N$ (tt)

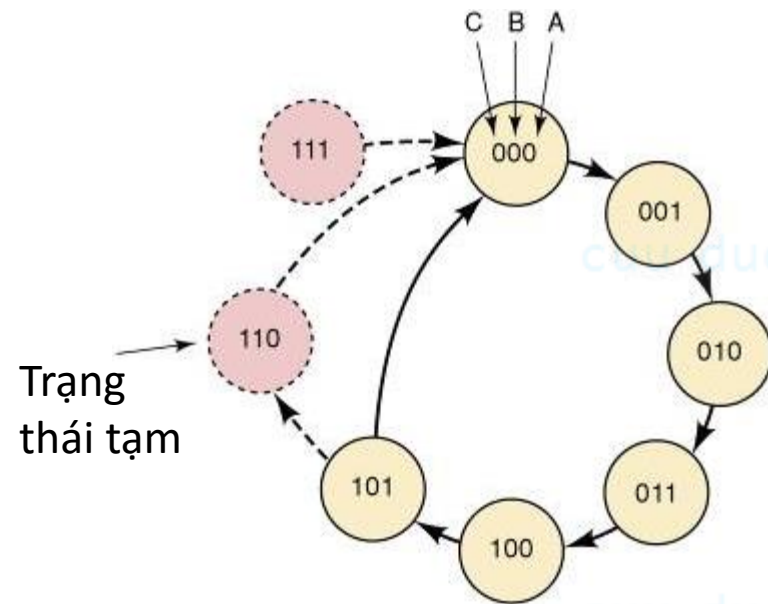


Bộ đếm MOD-6 được tạo từ bộ đếm MOD-8 bằng cách **clear bộ đếm** khi trạng thái **110** xuất hiện



# Bộ đếm có **Hệ số bộ đếm $< 2^N$** (tt)

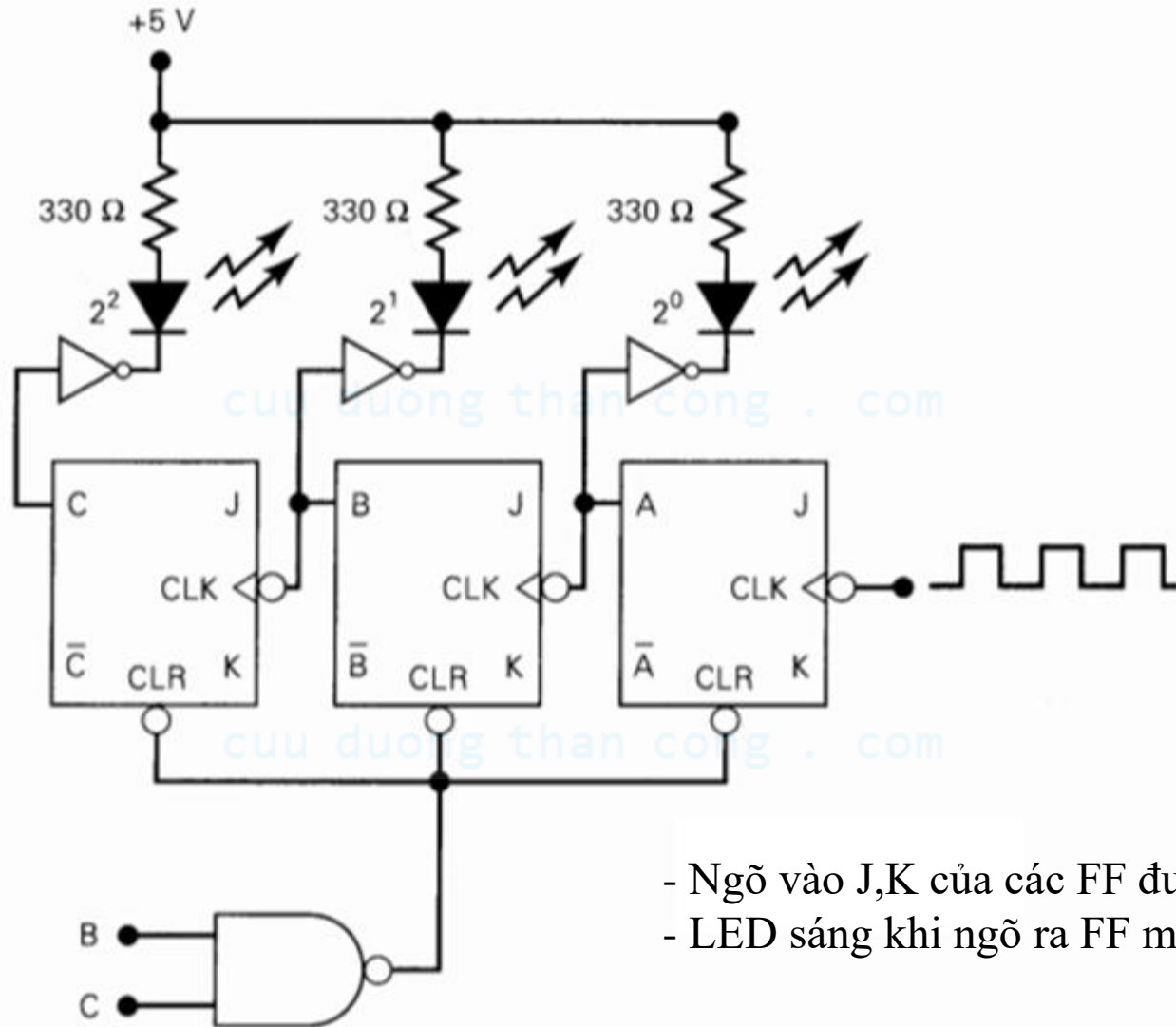
## Giản đồ chuyển trạng thái của bộ đếm MOD-6



- **Mỗi vòng tròn nét liền** chỉ một trạng thái thực sự của bộ đếm
- **Mỗi vòng tròn nét đứt** chỉ một trạng thái tạm của bộ đếm
- **Mũi tên nét liền** chỉ sự chuyển trạng thái giữa 2 trạng thái thực
- **Mũi tên nét đứt** chỉ sự chuyển từ trạng thái thực sang trạng thái tạm hoặc ngược lại

- Không có mũi tên chỉ đến trạng thái **111** vì trong chu trình của bộ đếm không có trạng thái nào chuyển đến trạng thái này
- Trạng thái **111** có thể xuất hiện khi bật nguồn (power-up)

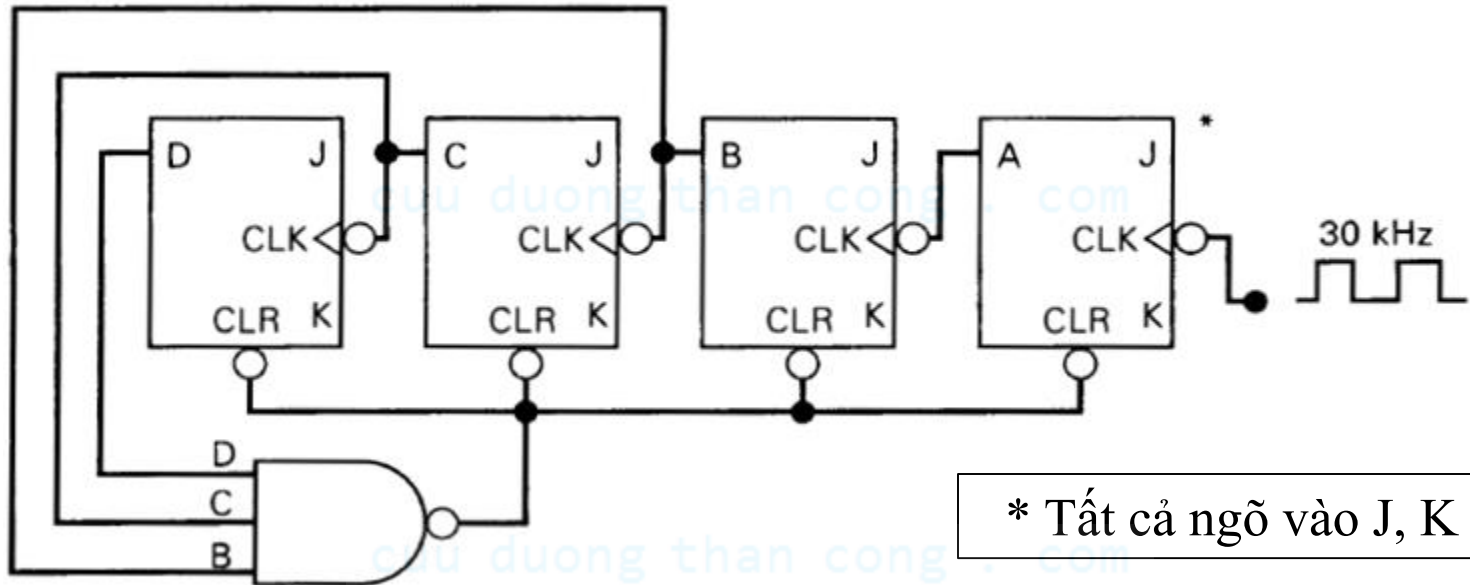
# Bộ đếm có **Hệ số bộ đếm $< 2^N$** (tt)



- Ngõ vào J,K của các FF được nối mức 1
- LED sáng khi ngõ ra FF mức cao

# Ví dụ 4

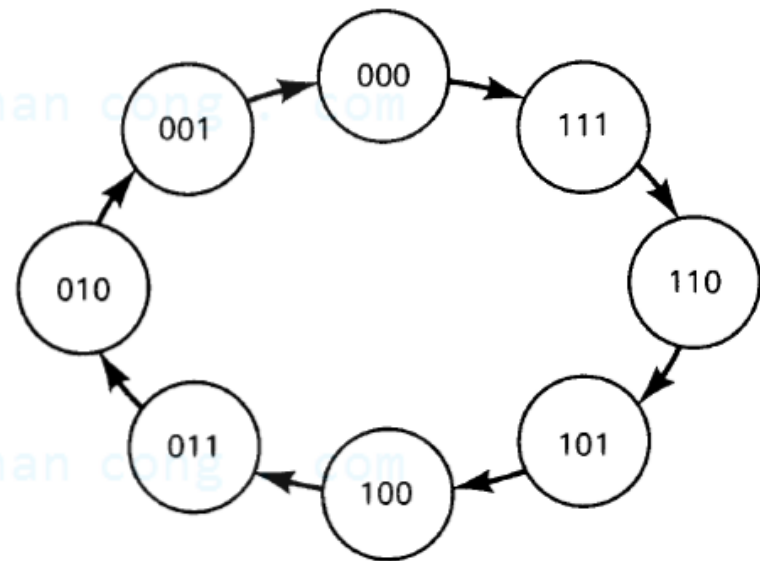
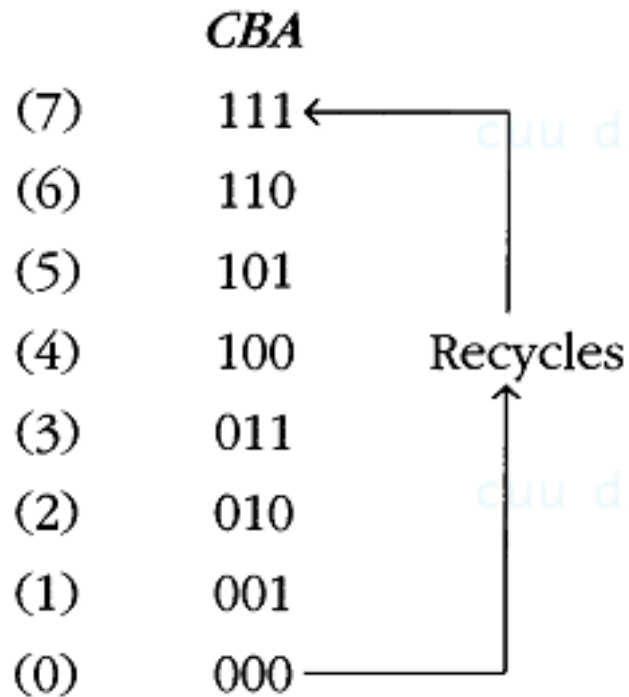
- Xác định hệ số bộ đếm (MOD number) của mạch đếm bên dưới?
- Xác định tần số tại ngõ ra D?



\* Tất cả ngõ vào J, K bằng 1

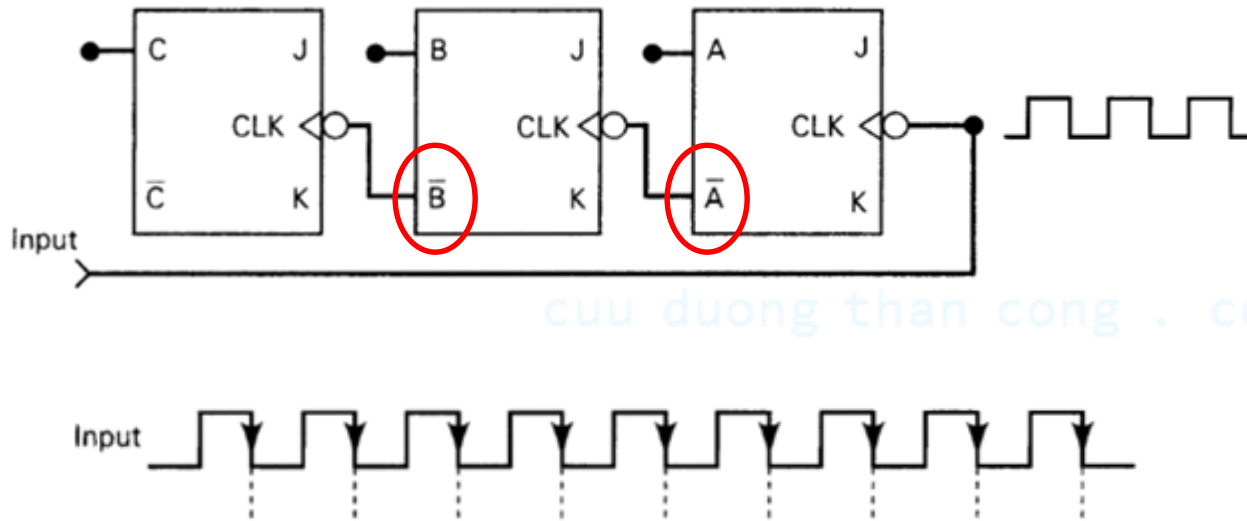
# Bộ đếm bất đồng bộ - Đếm xuống

- Bộ đếm xuống bất đồng bộ được xây dựng gần giống với bộ đếm lên bất đồng bộ



Lưu đồ chuyển trạng thái của bộ đếm xuống **MOD-8**

# Bộ đếm bất đồng bộ - Đếm xuống

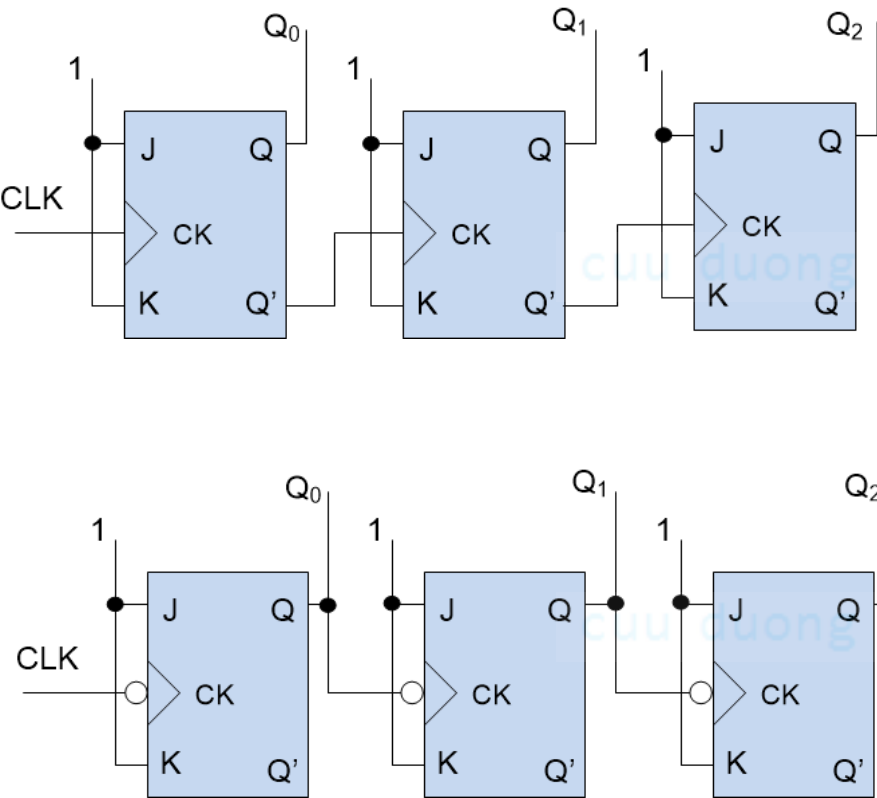


\* Tất cả ngõ vào J, K bằng 1

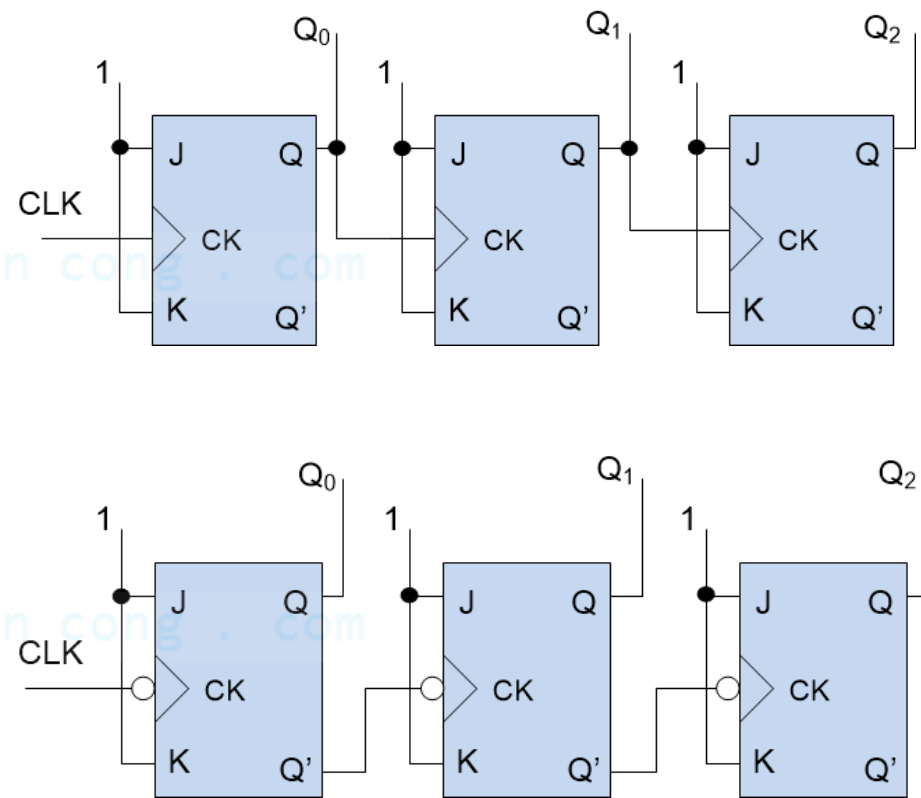
Bộ đếm xuống bất đồng bộ ít được sử dụng trong thực tế .

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

## Đếm lên



## Đếm xuống



**Chú ý:**  $Q_0$  có trọng số nhỏ nhất (LSB)  
 $Q_2$  có trọng số lớn nhất (MSB)

# Thiết kế bộ đếm bất đồng bộ MOD-X

**Ví dụ:** Thiết kế bộ đếm lên bất đồng bộ **MOD-5** dùng **FF-T** có xung clock kích cạnh xuống, ngõ vào *Preset* và *Clear* tích cực cao. Biết rằng trạng thái ban đầu của bộ đếm là 5.

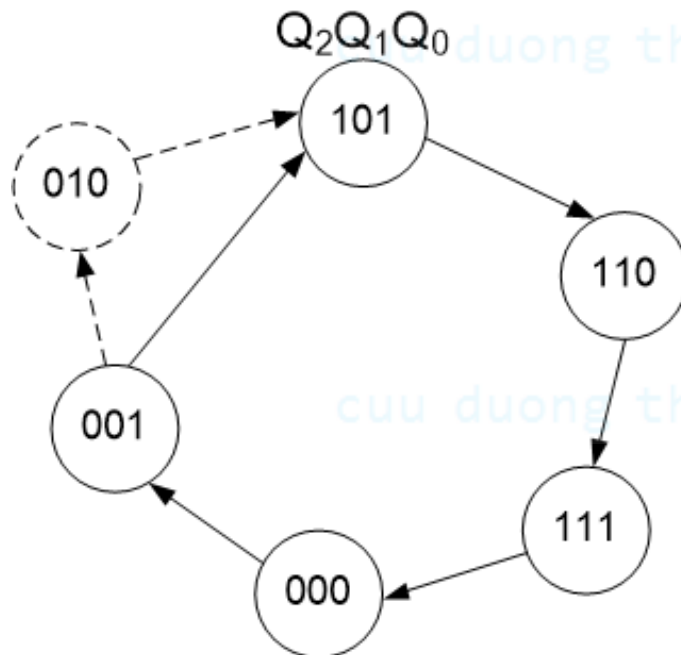
Bước 1: Tìm số flip-flop cần dùng nhỏ nhất thỏa yêu cầu bài toán  
( $2^N \geq X$ )

Ta có:  $2^3 \geq 5$  (MOD-5)  $\rightarrow$  Sử dụng **3 FF**

# Thiết kế bộ đếm bất đồng bộ MOD-X (tt)

**Ví dụ:** Thiết kế bộ đếm lên bất đồng bộ **MOD-5** dùng **FF-T** có xung clock kích cạnh xuống, ngõ vào *Preset* và *Clear* tích cực cao. Biết rằng trạng thái ban đầu của bộ đếm là 5.

Bước 2: Vẽ lưu đồ chuyển trạng thái của bộ đếm



- Trạng thái Reset của bộ đếm:  
 $Q_2Q_1Q_0 = 010$
- Trạng thái không có trong chu trình đếm  $Q_2Q_1Q_0 = 011, 100$



# Thiết kế bộ đếm bất đồng bộ MOD-X (tt)

**Ví dụ:** Thiết kế bộ đếm lên bất đồng bộ **MOD-5** dùng FF-T có xung clock *kích cạnh xuống*, ngõ vào *Preset* và *Clear* tích cực cao. Biết rằng *trạng thái ban đầu của bộ đếm là 5*.

## Bước 3: Thiết kế mạch Reset của bộ đếm

- Trường hợp 1:  $2^N = X \rightarrow$  Mạch không bị Reset  $\rightarrow$  bỏ qua bước 3
- Trường hợp 2:  $2^N \geq X$ 
  - **Nếu số FF sử dụng từ 6 trở lên:**
    - Sử dụng cổng AND/NAND nếu PR và CLR tích cực cao/thấp
    - Kết nối các giá trị ngõ ra tương ứng của các FF tại trạng thái Reset của bộ đếm với ngõ vào của cổng AND/NAND ở trên
    - Kết nối ngõ ra cổng AND/NAND tới chân PR và CLR thích hợp tại các FF
  - Nếu số FF sử dụng *nhỏ hơn 6*:

# Thiết kế bộ đếm bất đồng bộ MOD-X (tt)

## Bước 3: Thiết kế mạch Reset của bộ đếm (tt)

### ➤ Trường hợp 2: $2^N \geq X$

- Nếu số FF sử dụng *từ 6 trở lên*:

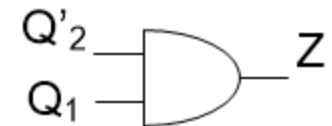
...

- Nếu số FF sử dụng *nhỏ hơn 6*:

Sử dụng bìa Karnaugh để rút gọn:

- Tùy thuộc vào chân **PR** và **CLR** của FF sử dụng tích cực *cao hay thấp*, ta sẽ điền giá trị *1 hay 0* tương ứng tại trạng thái Reset
- Những trạng thái không có trong chu trình đếm, để giá trị *x* (tùy định)

Z Q <sub>0</sub>	Q <sub>2</sub> Q <sub>1</sub>		00	01	11	10
	0	1		1		x
1	0	1		x		



Mạch Reset  
của bộ đếm

# Thiết kế bộ đếm bất đồng bộ MOD-X (tt)

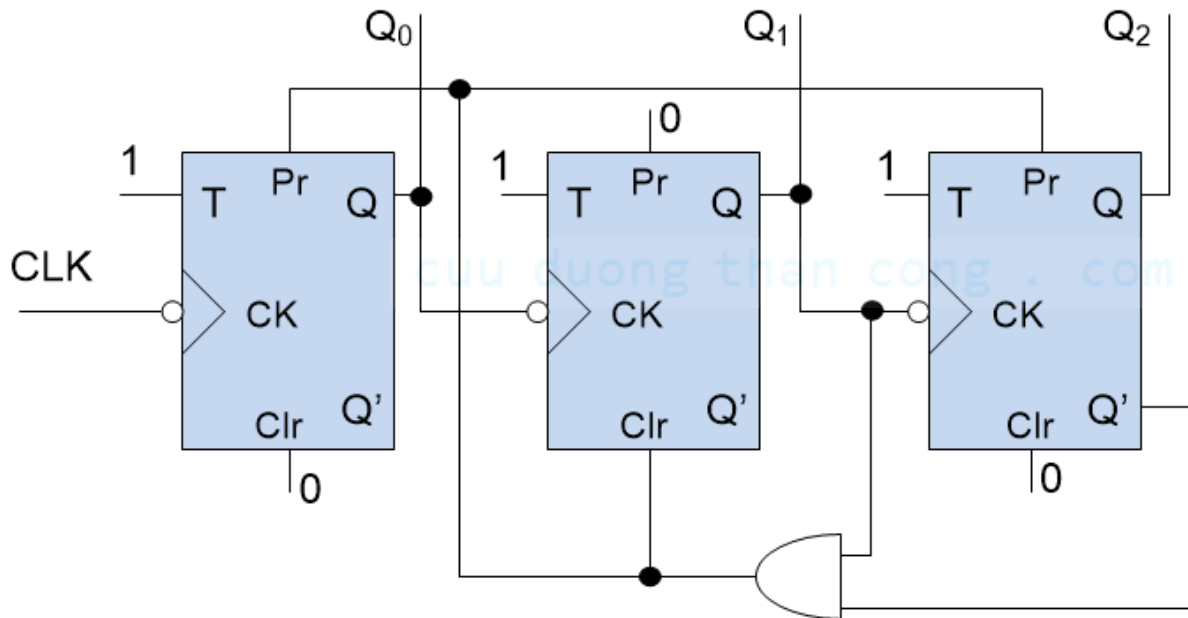
**Ví dụ:** Thiết kế bộ đếm lên bất đồng bộ **MOD-5** dùng FF-T có xung clock *kích cạnh xuống*, ngõ vào *Preset* và *Clear* tích cực cao. Biết rằng trạng thái ban đầu của bộ đếm là 5.

### Bước 4: Vẽ mạch cần thiết kế

**(Lưu ý: - FF kích cạnh lên/xuống; mạch đếm lên/xuống**

- Pr và Clr tích cực cao/thấp

- Trạng thái Reset và trạng thái của bộ đếm sau khi mạch được Reset)

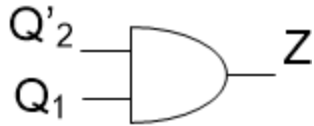


# Thiết kế bộ đếm bất đồng bộ MOD-X (tt)

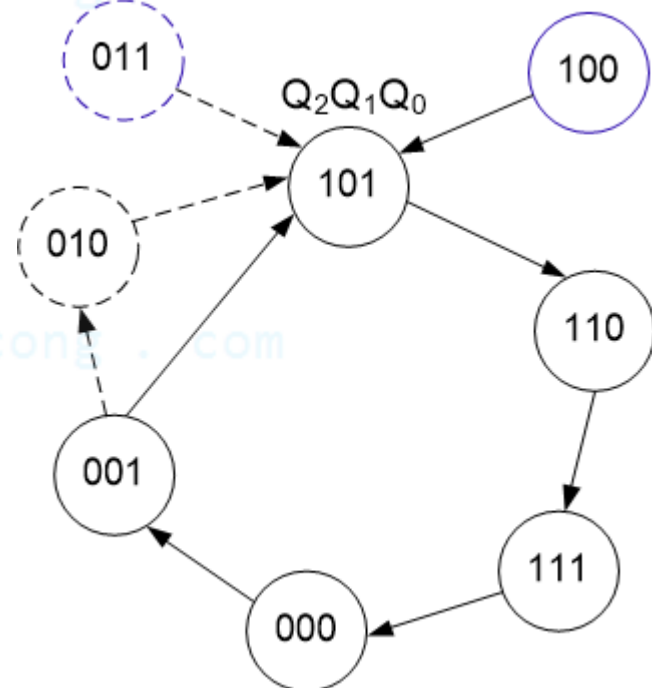
**Ví dụ:** Thiết kế bộ đếm lên bất đồng bộ **MOD-5** dùng FF-T có xung clock *kích cạnh xuống*, ngõ vào *Preset* và *Clear* tích cực cao. Biết rằng *trạng thái ban đầu của bộ đếm là 5*.

**Bước 5:** Vẽ lưu đồ trạng thái đầy đủ của bộ đếm  
(bao gồm các trạng thái không có trong chu trình đếm)

- Mạch Reset của bộ đếm

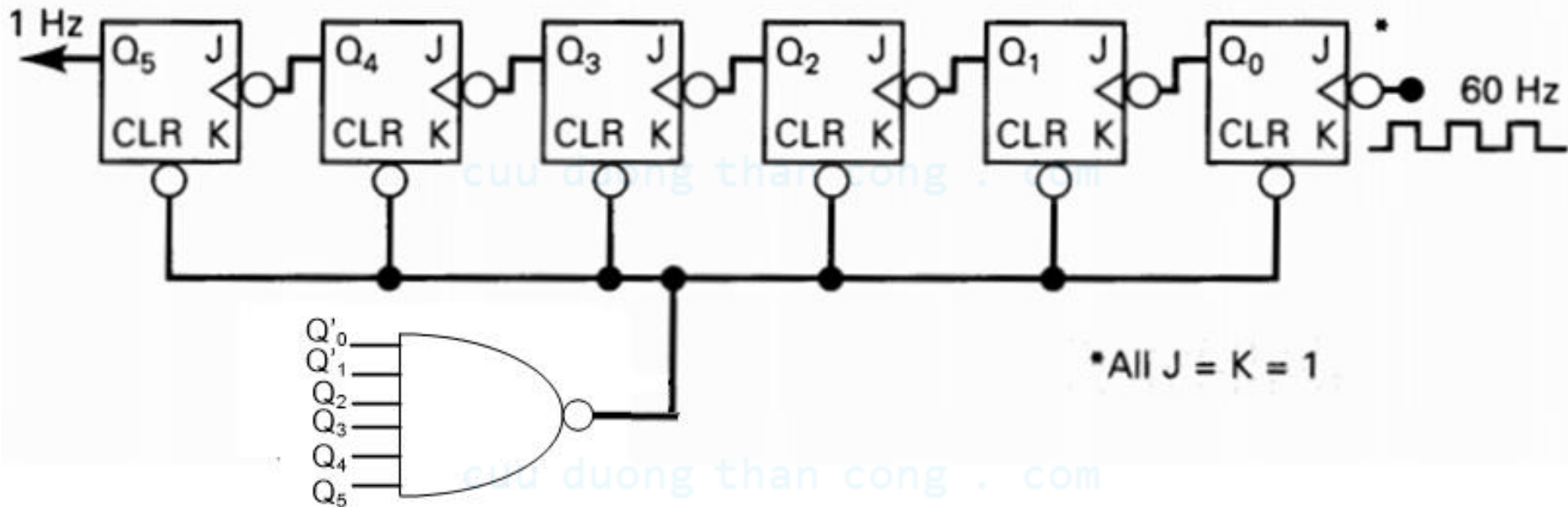


- Trạng thái không có trong chu trình đếm  $Q_2Q_1Q_0 = 011, 100$ 
  - Với  $Q_2Q_1Q_0 = 011 \rightarrow Z = 1$   
 $\rightarrow$  mạch bị Reset
  - Với  $Q_2Q_1Q_0 = 100 \rightarrow Z = 0$   
 $\rightarrow$  mạch không bị Reset



Response	Percentage
Yes, the current government is responsible	85%

## Thiết kế bộ đếm MOD-60 trong ví dụ 3



# Ví dụ 6

Sử dụng **FF-T** để thiết kế bộ đếm bất đồng bộ MOD-10 đếm từ giá trị 0 đến 9.

Biết rằng FF sử dụng kích cạnh xuống, ngõ vào Pr và Clr tích cực mức thấp.

# Delay của bộ đếm tích lũy trì hoãn (tt)

- Bộ đếm tích lũy trì hoãn có thiết kế đơn giản. Tuy nhiên, hạn chế của bộ đếm là delay của FF trước được tích lũy đến FF sau
  - Delay của toàn mạch lớn
  - Bộ đếm này không phù hợp cho các thiết kế hoạt động ở tần số cao
- Để mạch hoạt động đúng thì **chu kì** của xung Clock phải lớn hơn tổng Delay của mạch

$$T_{\text{clock}} \geq N \times t_{\text{pd}}$$

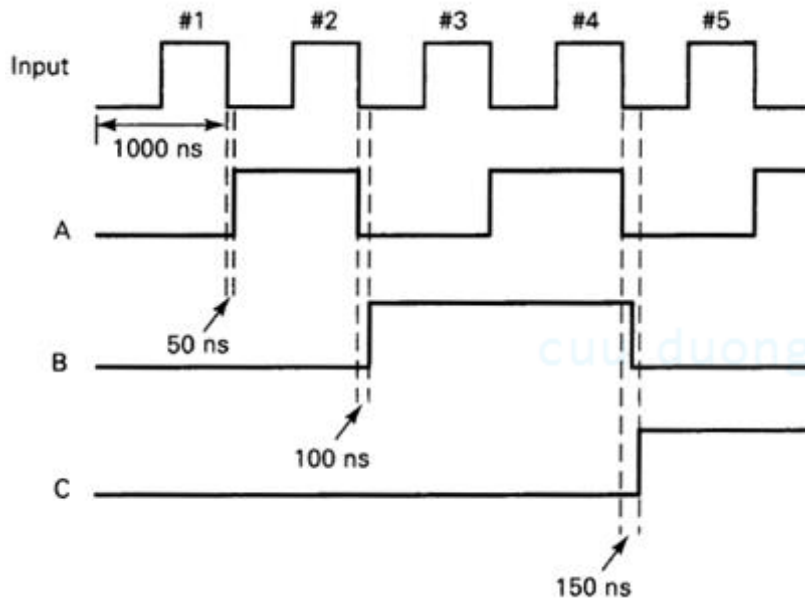
$T_{\text{clock}}$ : chu kì xung Clock

$N$ : số FF của mạch

$T_{\text{pd}}$ : delay của một FF

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

# Delay của bộ đếm tích lũy trì hoãn (tt)

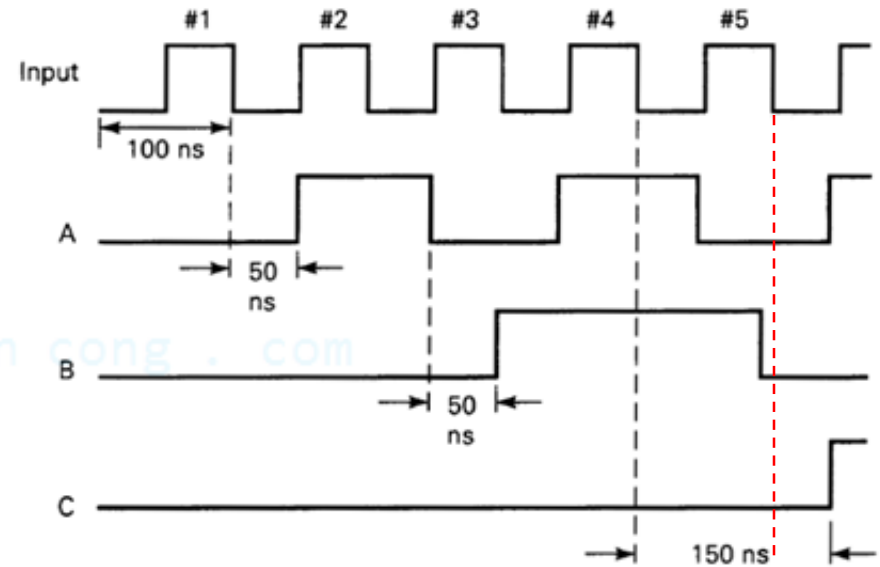


- $T=1000\text{ns}$

- $t_{pd}=50\text{ns}$

$\Rightarrow T \geq 3 \times t_{pd}$

**Bộ đếm hoạt động đúng**



Trạng thái CBA = 100  
không xuất hiện

- $T=100\text{ns}$

- $t_{pd}=50\text{ns}$

$\Rightarrow T < 3 \times t_{pd}$

**Bộ đếm hoạt động sai**



# Delay của bộ đếm tích lũy trì hoãn (tt)



- Bộ đếm bất đồng bộ sẽ không hữu ích khi hoạt động ở tần số cao, đặc biệt khi bộ đếm sử dụng nhiều flip-flop.
- Tuy nhiên, vì tính đơn giản trong thiết kế, bộ đếm bất đồng bộ vẫn được sử dụng trong các mạch không đòi hỏi tần số cao.

cuu duong than cong . com

# Nội dung

- Bộ đếm bất đồng bộ (Asynchronous counters)
  - Hệ số của bộ đếm (MOD number)
  - Bộ đếm lên/xuống (Up/ Down counters)
  - Phân tích và thiết kế bộ đếm bất đồng bộ
  - Delay của mạch (Propagation delay)
- **Bộ đếm đồng bộ (Synchronous counters)**
  - Phân tích bộ đếm đồng bộ (Analyze synchronous counters)
  - Thiết kế bộ đếm đồng bộ (Design synchronous counter)
- Thanh ghi (Register)



## Bộ đếm đồng bộ (Synchronous counters)

cuu duong than cong . com

cuu duong than cong . com

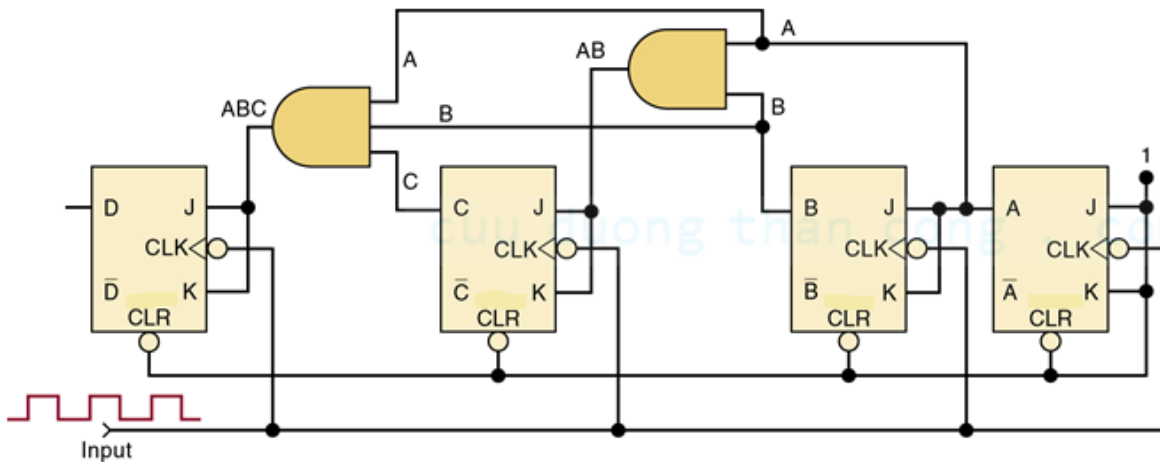
# Bộ đếm đồng bộ

## (Synchronous Counters)

- **Bộ đếm đồng bộ** hay **bộ đếm song song** là bộ đếm trong đó các FF được kích đồng thời bởi một xung Clock

Tín hiệu Clock được kết nối tới ngõ vào CLK của tất cả các FF trong mạch  $\rightarrow$  Delay của mạch sẽ bằng với 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* theo mong muốn của người thiết kế



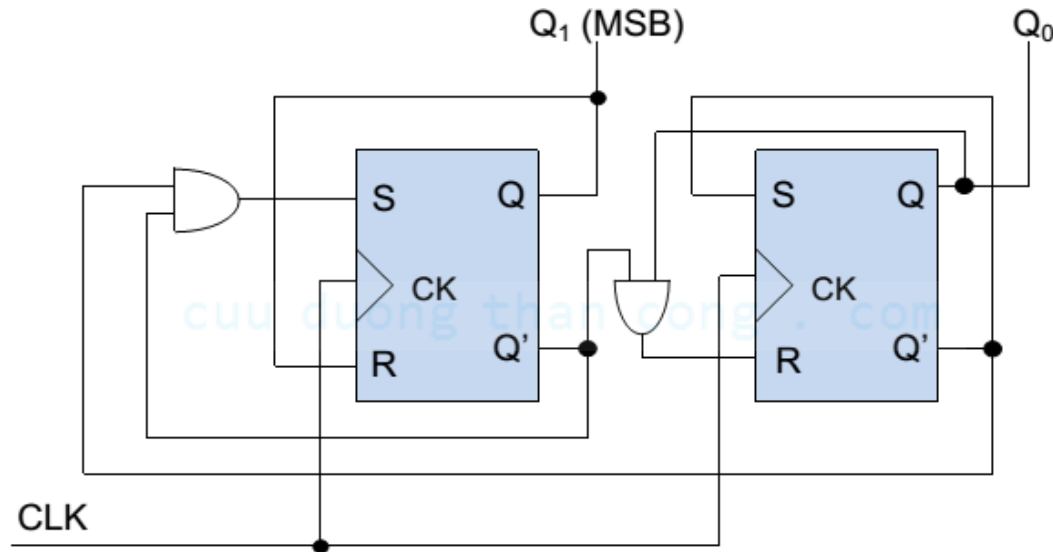
Bộ đếm đồng bộ thường sử dụng  
trong các mạch có tần số cao

Count	D	C	B	A
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
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0

# Phân tích bộ đếm đồng bộ

## (Analyze Synchronous Counters)

**Ví dụ:** Phân tích mạch đếm ở hình bên dưới



Bước 1: Tìm phương trình ngõ vào của các FF

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

$$S_0 = Q'_0$$

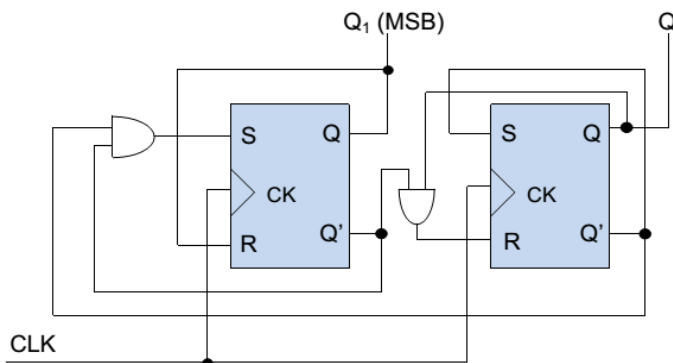
$$R_1 = Q_1$$

$$R_0 = Q'_1 Q_0$$

# Phân tích bộ đếm đồng bộ

## (Analyze Synchronous Counters)

**Ví dụ:** Phân tích mạch đếm ở hình bên



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

TTHT Q <sub>1</sub> Q <sub>0</sub>	Ngõ vào S <sub>1</sub> R <sub>1</sub> S <sub>0</sub> R <sub>0</sub>				TTKT Q <sup>+</sup> <sub>1</sub> Q <sup>+</sup> <sub>0</sub>
0 0	-	-	-	-	- -
0 1	-	-	-	-	- -
1 0	-	-	-	-	- -
1 1	-	-	-	-	- -

Bảng chuyển trạng thái

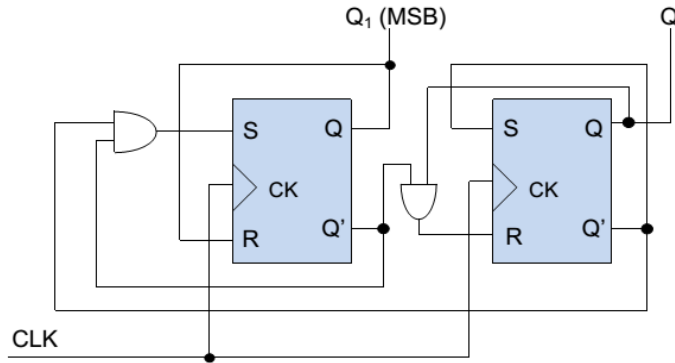
**TTHT:** Trạng thái hiện tại (Current State)

**TTKT:** Trạng thái kế tiếp (Next State)

# Phân tích bộ đếm đồng bộ

## (Analyze Synchronous Counters)

**Ví dụ:** Phân tích mạch đếm ở hình bên dưới



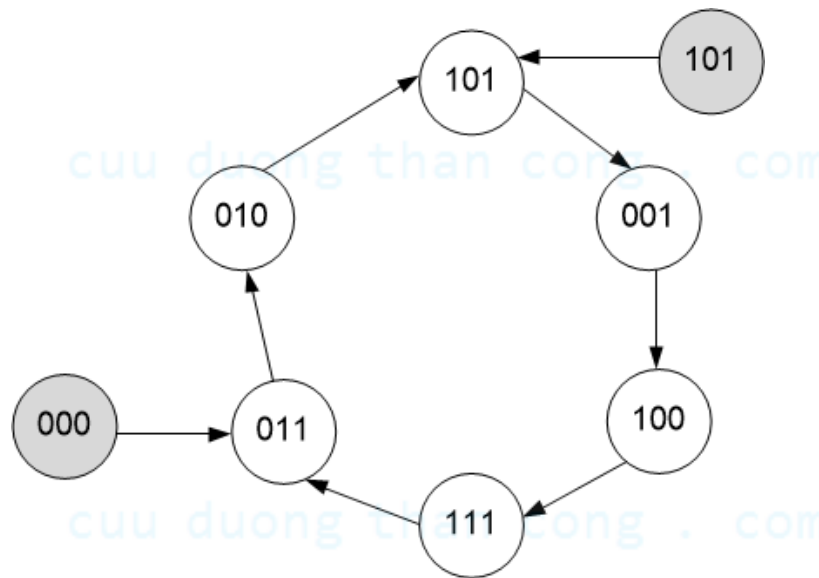
Bước 3: Vẽ lưu đồ chuyển trạng thái của bộ đếm

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

# Thiết kế bộ đếm đồng bộ

## (Design Synchronous Counter)

- Bộ đếm đồng bộ có thể được thiết kế để tạo ra chuỗi đếm bất kì theo mong muốn của người thiết kế



- Thiết kế bộ đếm đồng bộ?



# Mô tả đầy đủ của một Flip-flop

Có 4 dạng FF cơ bản: **D, T, S\_R, J\_K**

FF có thể được mô tả bằng *ký hiệu hình học*, *bảng sự thật*, *bảng đặc tính*, *phương trình đặc tính* hoặc *bảng kích thích*

✓ **Bảng đặc tính** (*Characteristic table*)

Một bảng chỉ ra *trạng thái kế tiếp* như một hàm của *trạng thái hiện tại* và *ngõ vào* của của mỗi FF

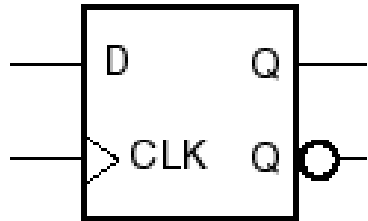
**Phương trình đặc tính** (*Characteristic equation*)

Một *biểu thức* chỉ ra quan hệ của *trạng thái kế tiếp* theo *trạng thái hiện tại* và *ngõ vào* của mỗi FF

✓ **Bảng kích thích** (*Excitation table*)

Một bảng liệt kê các *yêu cầu ngõ vào (input)* để FF chuyển từ *trạng thái hiện tại* đến *trạng thái kế tiếp*

# Mô tả đầy đủ của FF-D



Ký hiệu

D	CLK	Q	QN
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

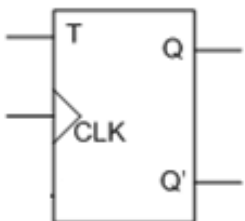
Bảng kích thích

D	Q	$Q^+$
0	0	0
0	1	0
1	0	1
1	1	1

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

Phương trình đặc tính

# Mô tả đầy đủ của FF-T



Ký hiệu

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

Bảng sự thật

T	$Q_{(next)}$
0	Q
1	Q'

Bảng đặc tính

Q	$Q_{(next)}$	T
0	0	0
0	1	1
1	0	1
1	1	0

Bảng kích thích

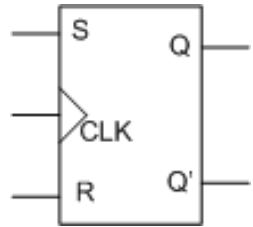
T	Q	$Q^+$
0	0	0
0	1	1
1	0	1
1	1	0

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

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

Phương trình đặc tính

# Mô tả đầy đủ của FF-S\_R



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 <sub>(next)</sub>
0	0	Q
0	1	0
1	0	1
1	1	?

Bảng đặc tính

Q	Q <sub>(next)</sub>	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

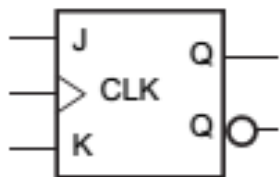
Bảng kích thích

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

Phương trình đặc tính

S	R	Q	Q <sup>+</sup>
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

# Mô tả đầy đủ của FF-J\_K



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

J	K	$Q_{(next)}$
0	0	Q
0	1	0
1	0	1
1	1	$Q'$

Bảng đặc tính

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	$Q^+$
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

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

Phương trình đặc tính

# Thiết kế bộ đếm đồng bộ

**Ví dụ:** Sử dụng FF-J\_K để thiết kế một bộ đếm 3-bit có chuỗi đếm như bảng bên cạnh

**Lưu ý:** *Thuộc tính* (đếm lên/xuống) của **bộ đếm đồng bộ** chỉ phụ thuộc vào trạng thái hiện tại và trạng thái kế tiếp mà *không quan tâm* đến tính chất của FF (kích cạnh lên/xuống)

→ Khác với bộ đếm bất đồng bộ

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

**Bước 1:** Tìm số FF nhỏ nhất thỏa yêu cầu bài toán

Ví dụ này đã chỉ ra sử dụng 3 FF ngay trong đề

# Thiết kế bộ đếm đồng bộ

## Bước 2: Vẽ biểu đồ chuyển trạng thái (state diagram) của bộ đếm

**Lưu ý:** - 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 có 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.		

# Thiết kế bộ đếm đồng bộ

## Bước 3: Lập bảng trạng thái (state table)

- sử dụng 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 <sup>+</sup>	B <sup>+</sup>	A <sup>+</sup>
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



# Thiết kế bộ đếm đồng bộ

## Bước 4: Lập bảng kích thích của mạch (circuit excitation table)

- 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 <sup>+</sup>	B <sup>+</sup>	A <sup>+</sup>		J <sub>C</sub>	K <sub>C</sub>	J <sub>B</sub>	K <sub>B</sub>	J <sub>A</sub>	K <sub>A</sub>
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

# Thiết kế bộ đếm đồng bộ

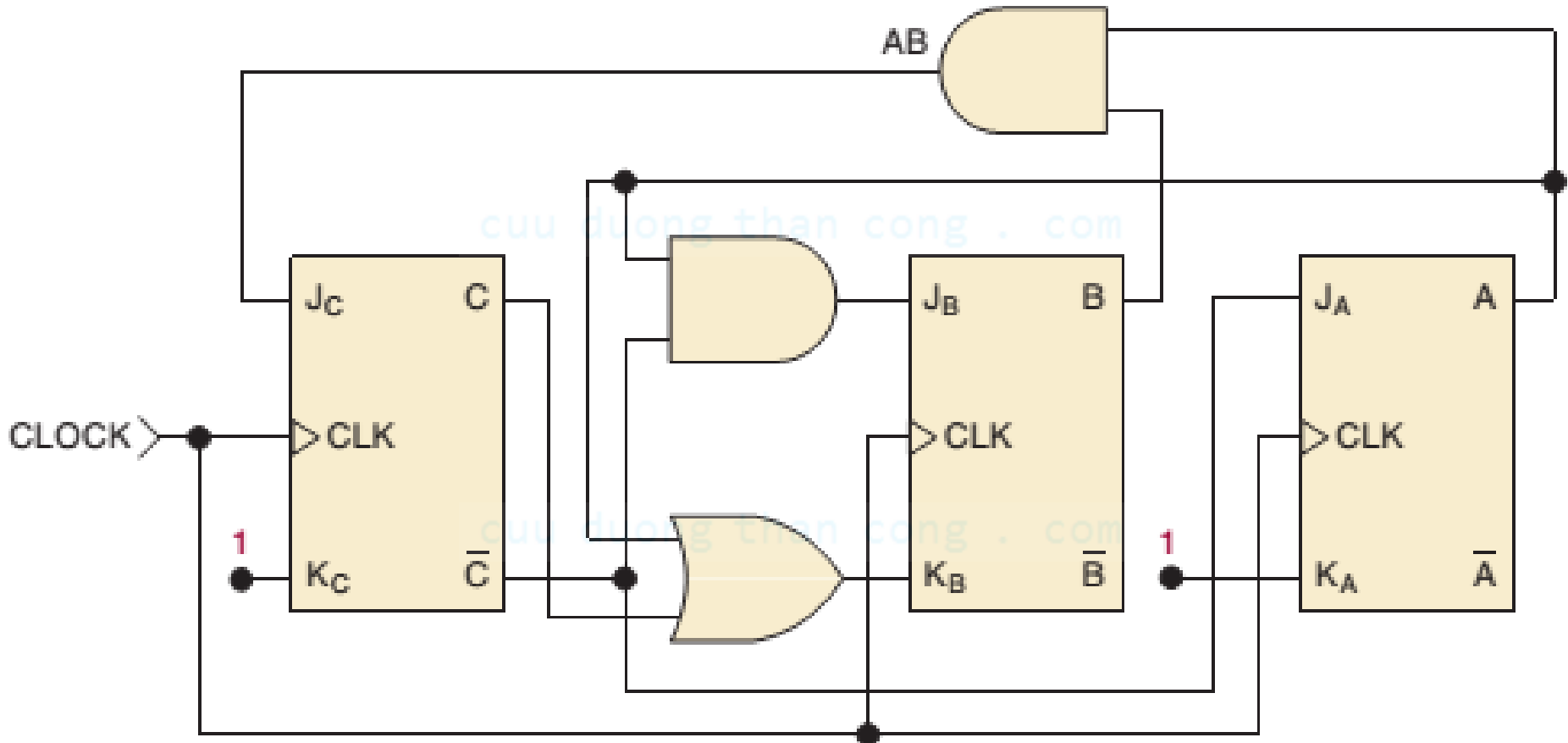
**Bước 5:** Sử dụng bìa Karnaugh (bìa K) để tìm  
*phương trình ngõ vào của các FF được sử dụng*

cuu duong than cong . com

cuu duong than cong . com

# Thiết kế bộ đếm đồng bộ

## Bước 6: Vẽ mạch cần thiết kế



# Câu hỏi thảo luận?

## Đúng hay Sai?

1. Thiết kế bộ đếm đồng bộ để thực hiện chuỗi đếm sau:  
**0010, 0011, 0100, 0111, 1010, 1111, và lặp lại.**

cuu duong than cong . com

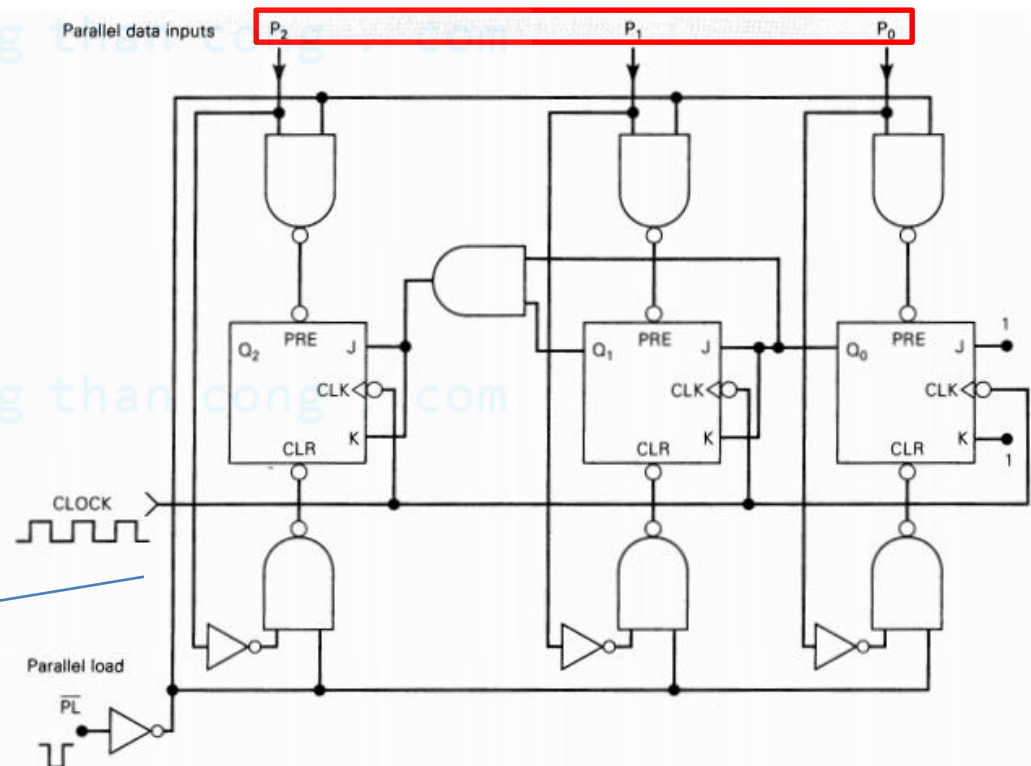
2. Thiết kế bộ đếm đồng bộ để thực hiện chuỗi đếm sau:  
**0010, 0011, 0100, 0111, 1010, 0100, 1111 và lặp lại**

cuu duong than cong . com

# Bộ đếm có khả năng định giá trị ban đầu (Presettable Counters)

- **Bộ đếm có khả năng định giá trị ban đầu** là bộ đếm có thể định giá trị ban đầu trước khi bộ đếm hoạt động.
  - Việc định giá trị ban đầu có thể thực hiện đồng bộ hoặc bất đồng bộ
- Thao tác định giá trị ban đầu cho bộ đếm còn được gọi là nạp dữ liệu song song (parallel loading) cho bộ đếm

1. Đưa giá trị dữ liệu mong muốn vào các ngõ vào song song ( $P_2P_1P_0$ )
2. Điều khiển  $\overline{PL} = 0$  để nạp dữ liệu ban đầu vào bộ đếm



**Bộ đếm lên đồng bộ**

*nạp dữ liệu song song bất đồng bộ*

# Câu hỏi thảo luận?

- Thế nào là bộ đếm có khả năng định giá trị ban đầu?
- Mô tả sự khác nhau giữa định giá trị theo kiểu đồng bộ (synchronous presetting) và theo kiểu bất đồng bộ (asynchronous presetting)?

cuu duong than cong . com

# Nội dung

- Bộ đếm bất đồng bộ (Asynchronous counters)
  - Hệ số của bộ đếm (MOD number)
  - Bộ đếm lên/xuống (Up/ Down counters)
  - Phân tích và thiết kế bộ đếm bất đồng bộ
  - Delay của mạch (Propagation delay)
- Bộ đếm đồng bộ (Synchronous counters)
  - Phân tích bộ đếm đồng bộ (Analyze synchronous counters)
  - Thiết kế bộ đếm đồng bộ (Design synchronous counter)
- **Thanh ghi (Register)**



# Thanh ghi (Registers)

cuu duong than cong . com

cuu duong than cong . com



# Truyền dữ liệu thanh ghi

## (Register Data Transfer)

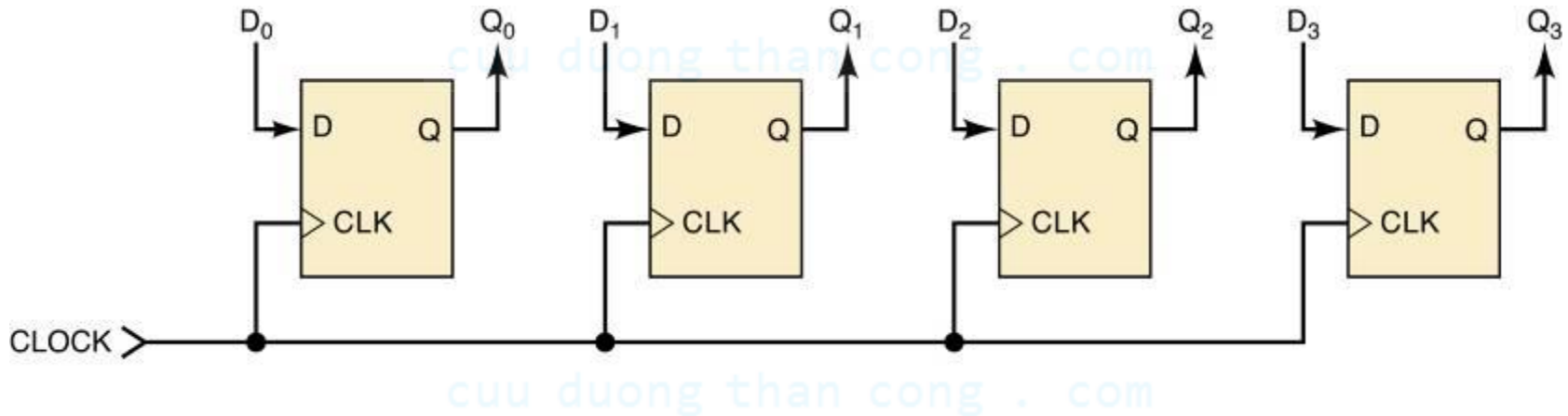
Sự phân loại thanh ghi dựa vào 2 đặc điểm:

- Cách dữ liệu được đưa vào thanh ghi để lưu trữ
  - Cách dữ liệu được lấy ra từ thanh ghi
- **Thanh ghi nối tiếp (Serial register):** dữ liệu được nạp vào thanh ghi theo dạng nối tiếp từ phải sang trái hoặc từ trái sang phải
    - Thanh ghi nối tiếp có dữ liệu ngõ ra được nối đến ngõ vào (feedback) được gọi là *thanh ghi quay vòng (rotate register)*
    - Thanh ghi nối tiếp có dữ liệu ngõ ra *không* nối đến ngõ vào được gọi là *thanh ghi dịch (shift register)*
  - **Thanh ghi song song (Parallel register):** dữ liệu được nạp vào thanh ghi theo dạng song.  
Thanh ghi này còn được gọi là *thanh ghi nạp (load register)*

# Truyền dữ liệu thanh ghi

## (Register Data Transfer)

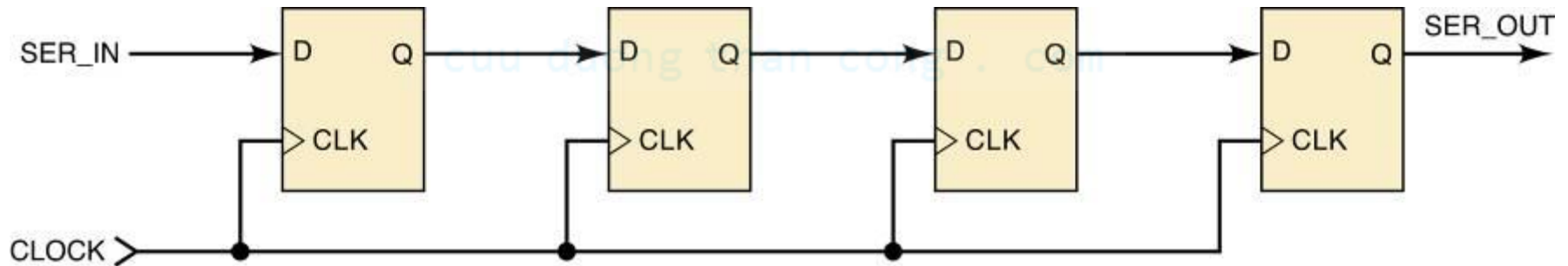
### Ngõ vào song song - ngõ ra song song (PIPO) (Parallel in/parallel out)



# Truyền dữ liệu thanh ghi

## (Register Data Transfer)

### Ngõ vào nối tiếp - ngõ ra nối tiếp (SISO) (serial in/serial out)

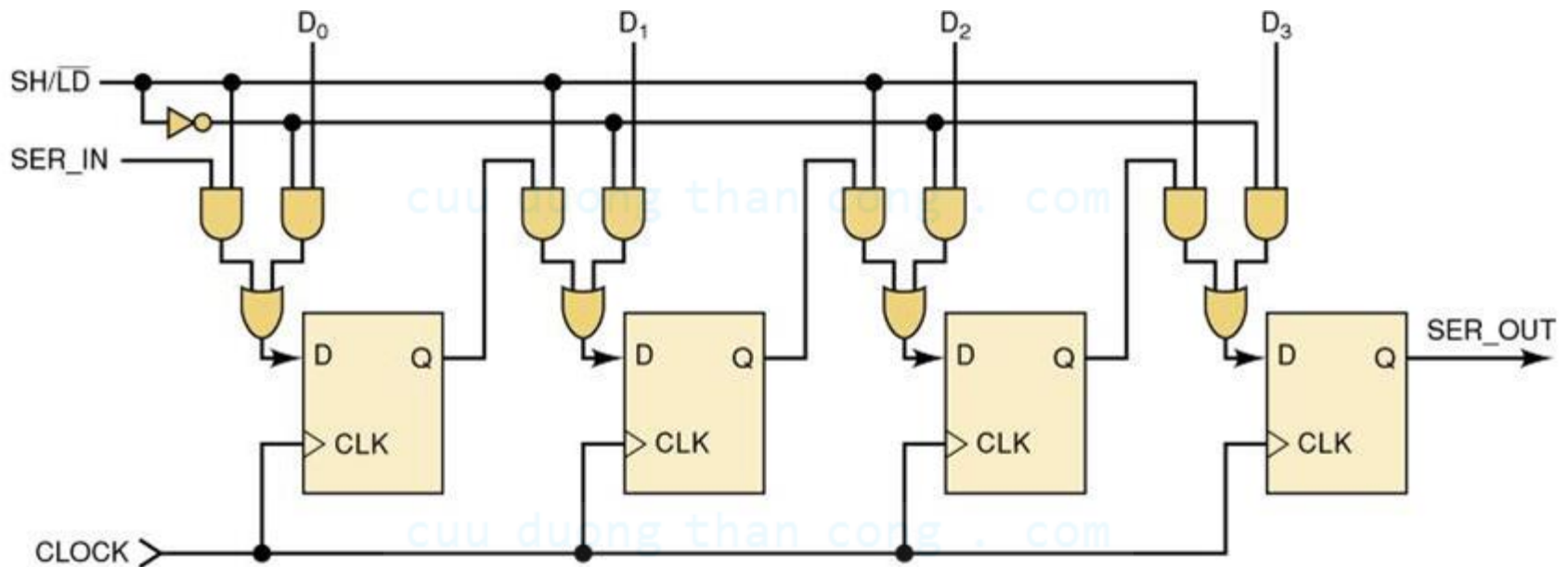


# Truyền dữ liệu thanh ghi

## (Register Data Transfer)

### Ngõ vào song song - ngõ ra nối tiếp (PISO)

(Parallel in/serial out)



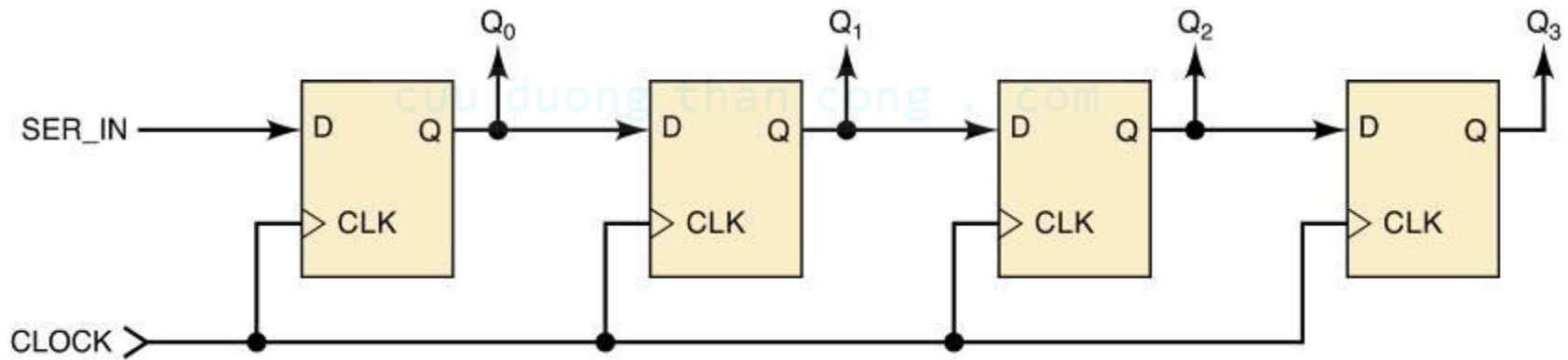
$\overline{\text{SH/LD}} = 0 \rightarrow$  parallel in/serial out

$\overline{\text{SH/LD}} = 1 \rightarrow$  serial in/serial out

# Truyền dữ liệu thanh ghi

## (Register Data Transfer)

### Ngõ vào nối tiếp - ngõ ra song song (SIPO) (serial in/parallel out)



# Bộ đếm thanh ghi dịch

## (Shift Register Counter)

- **Bộ đếm thanh ghi dịch** sử dụng **feedback**—dữ liệu *ngõ ra của FF cuối* được kết nối ngược lại *ngõ vào của FF đầu tiên*

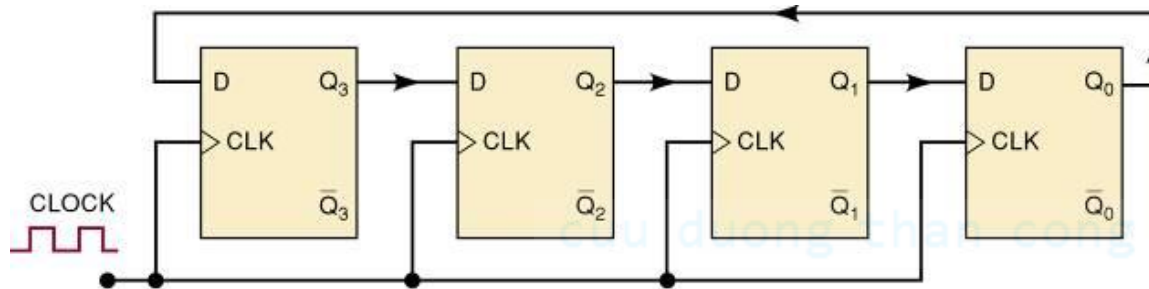
cuu duong than cong . com

cuu duong than cong . com

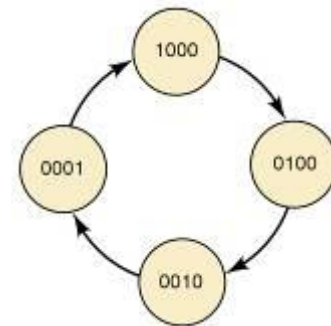
# Bộ đếm thanh ghi dịch

## Bộ đếm vòng tròn (Ring counter)

- Bộ đếm vòng tròn** là bộ đếm trong đó ngõ ra của FF sau cùng kết nối đến ngõ vào của FF đầu tiên



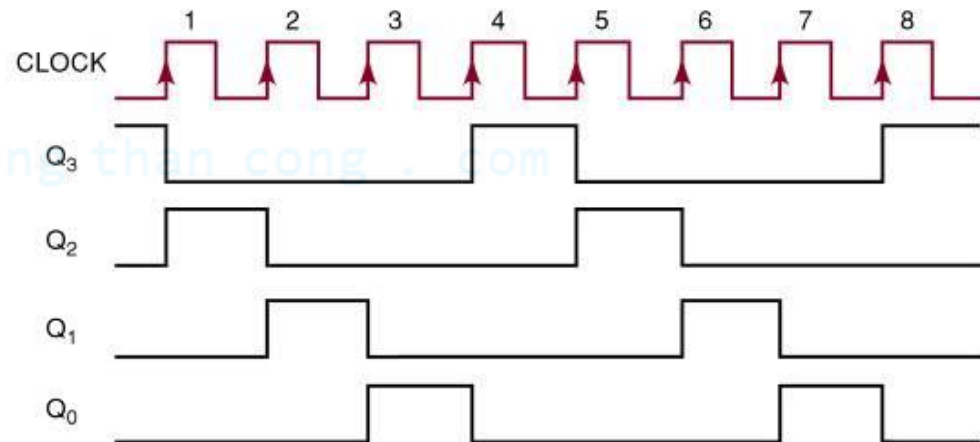
Bộ đếm vòng tròn 4-bit (MOD-4)  
( $Q_0$ : MSB,  $Q_3$ : LSB)



Biểu đồ chuyển trạng thái

$Q_3$	$Q_2$	$Q_1$	$Q_0$	CLOCK pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7
.	.	.	.	.

Bảng tuần tự



Dạng sóng của bộ đếm vòng tròn

# Bộ đếm thanh ghi dịch

## Bộ đếm vòng tròn (Ring counter)

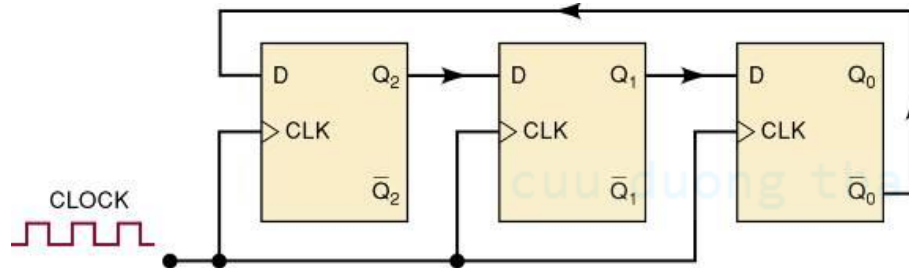
- **Tần số tại ngõ ra của mỗi FF bằng  $1/N$  tần số xung Clock đối với bộ đếm vòng tròn MOD-N**
  - Bộ đếm vòng tròn MOD-N  $\rightarrow$  cần  $N$  flip-flop
  - Bộ đếm vòng tròn yêu cầu nhiều FF hơn bộ đếm Binary thông thường có cùng hệ số đếm  
(ví dụ: MOD-8 cần 8 FF so với 3 FF trong bộ đếm thông thường)
  - Sự giải mã cho mỗi trạng thái đạt được bằng cách lấy giá trị ngõ ra tương ứng của mỗi FF mà *không cần dùng đến mạch giải mã*.
- **Để hoạt động chính xác, bộ đếm vòng tròn phải bắt đầu với chỉ một FF có ngõ ra bằng 1 và các FF còn lại có ngõ ra bằng 0.**
  - Khi mới bật nguồn, giá trị của các FF sẽ không dự đoán được, bộ đếm sẽ sử dụng chân **Preset** để định giá trị cho một FF và chân **Clear** để xóa các FF còn lại trước khi xung Clock được đưa vào



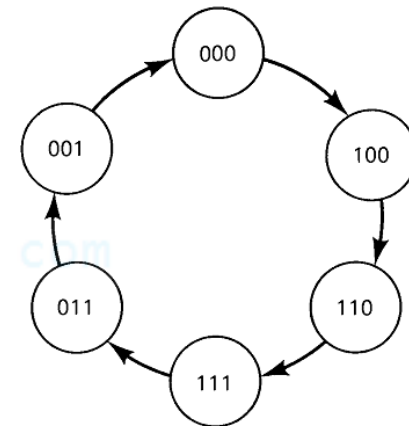
# Bộ đếm thanh ghi dịch

## Bộ đếm Johnson (Johnson counter)

- Trong bộ đếm Johnson hay bộ đếm vòng xoắn (twisted-ring counter) ngõ ra bù ( $\bar{Q}$ -bù) của FF cuối cùng sẽ kết nối với ngõ vào của FF đầu tiên.



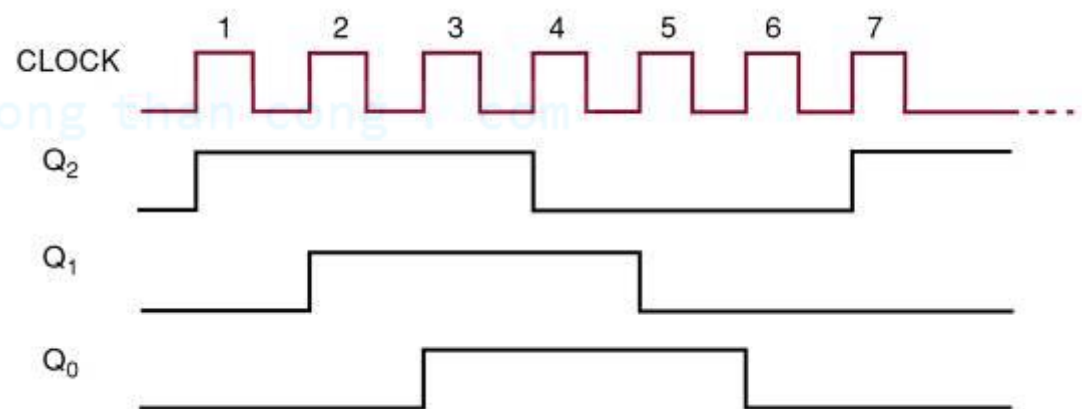
Bộ đếm Johnson 3-bit (MOD-6)  
( $Q_0$ : MSB,  $Q_2$ : LSB)



Biểu đồ chuyển trạng thái

$Q_2$	$Q_1$	$Q_0$	CLOCK pulse
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5
0	0	0	6
1	0	0	7
1	1	0	8
.	.	.	.

Bảng tuần tự

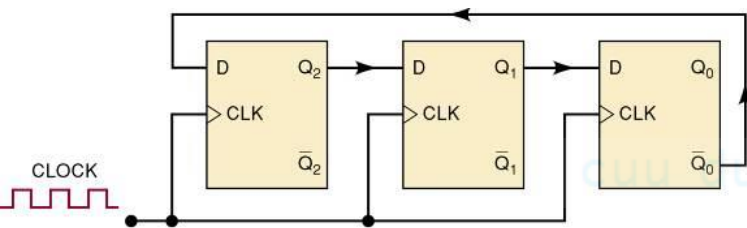


Dạng sóng của bộ đếm Johnson

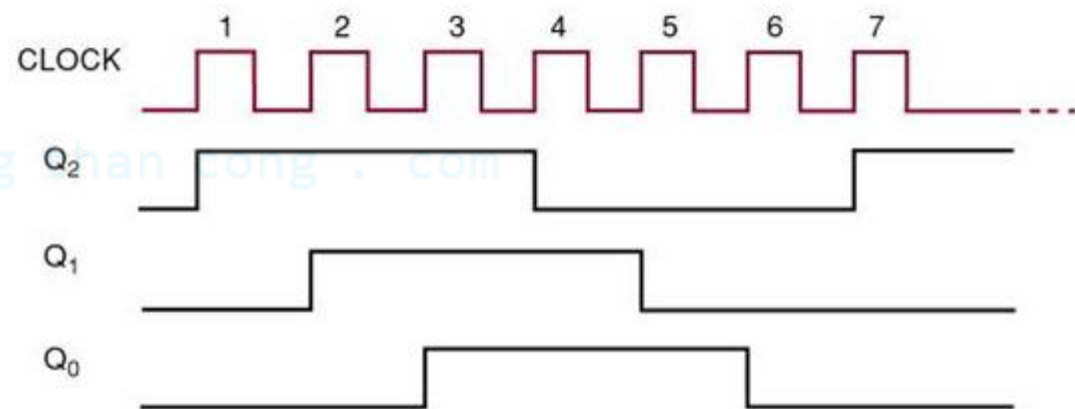
# Bộ đếm thanh ghi dịch

## Bộ đếm Johnson (Johnson counter)

- Với hệ số bộ đếm là  $N$  ( $N$  là số chẵn), bộ đếm Johnson chỉ cần  $N/2$  flip-flop
- Dạng sóng ở ngõ ra của mỗi FF là một xung vuông (50% duty cycle) và tần số bằng  $1/N$  tần số của xung Clock
- Dạng sóng ở ngõ ra của mỗi FF sẽ bị dịch đi một chu kỳ so với dạng sóng ở ngõ ra của FF trước nó (giống bộ đếm vòng tròn)



Bộ đếm Johnson 3-bit (MOD-6)  
( $Q_0$ : MSB,  $Q_2$ : LSB)

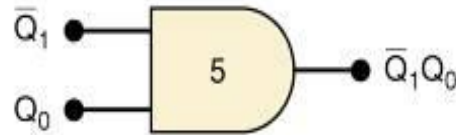
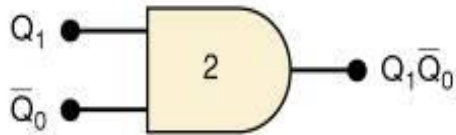
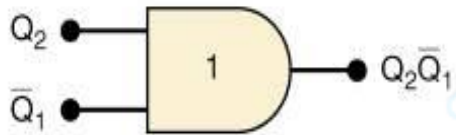
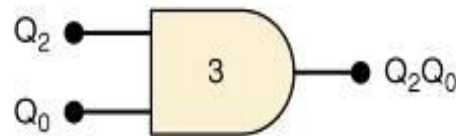
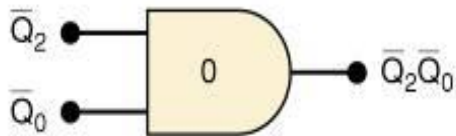


Dạng sóng của bộ đếm Johnson

# Bộ đếm thanh ghi dịch

## Bộ đếm Johnson (Johnson counter)

- Bộ đếm Johnson cần cổng logic bên ngoài để giải mã cho trạng thái.
- Cổng **AND-2** được dùng để giải mã cho bộ đếm Johnson mà *không quan tâm số FF* được sử dụng.



$Q_2$	$Q_1$	$Q_0$	Active gate
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5

# Câu hỏi thảo luận?

1. Bộ đếm thanh ghi dịch cần nhiều FF hơn bộ đếm Binary thông thường với cùng hệ số bộ đếm (MOD number)?
2. Bộ đếm thanh ghi dịch cần mạch giải mã phức tạp hơn bộ đếm Binary thông thường?
3. Làm sao để chuyển đổi bộ đếm vòng tròn sang bộ đếm Johnson?
4. Đúng hay Sai?
  - a) Ngõ ra của bộ đếm vòng tròn luôn luôn là xung vuông
  - b) Mạch giải mã cho bộ đếm Johnson đơn giản hơn bộ đếm Binary thông thường?
  - c) Bộ đếm vòng tròn và Johnson là bộ đếm đồng bộ?
5. Cần bao nhiêu FF để thiết kế bộ đếm vòng tròn MOD-16? Bộ đếm Johnson MOD-16?



# Thảo luận?

cuu duong than cong . com

cuu duong than cong . com