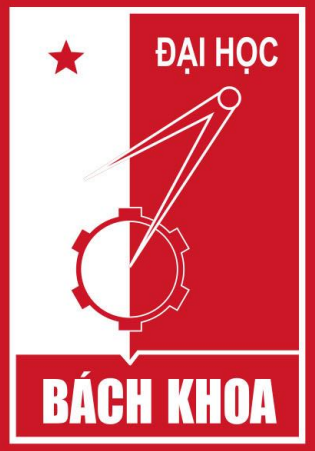


ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ĐIỆN TỬ CHO CÔNG NGHỆ THÔNG TIN

IT3420

Điện tử cho CNTT

- Chương 1: RLC
- Chương 2: Diode
- Chương 3: Transistor
- Chương 4: Khuếch đại thuật toán
- Chương 5: Cơ sở lý thuyết mạch số
- Chương 6: Các cổng logic cơ bản
- Chương 7: Mạch tổ hợp
- Chương 8: Mạch tuần tự

Chương 8 Mạch tuần tự

1. Khái niệm mạch tuần tự
2. Flip Flop
3. Phân loại Flip flop
4. Mô hình của mạch tuần tự
5. Một số ứng dụng mạch tuần tự

Bài giảng có sử dụng hình vẽ, text từ các tài liệu tham khảo:

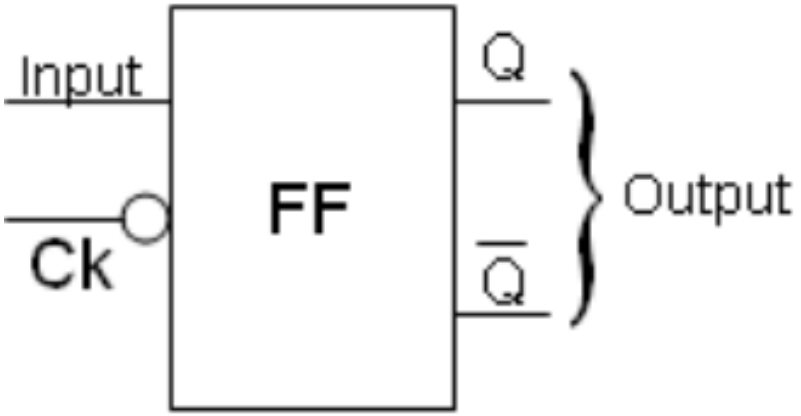
- ***Digital electronics: Principles, Devices, and Applications, Anil Kumar Maini 2007 John Wiley & Sons***

8.1 Khái niệm mạch tuần tự

- Mạch logic tuần tự là mạch có tín hiệu ra không chỉ phụ thuộc vào tín hiệu vào tại thời điểm hiện tại mà còn phụ thuộc vào quá khứ của tín hiệu vào.
- Một mạch có *n* biến trạng thái nhị phân sẽ có 2^n trạng thái xảy ra, và 2^n luôn là giá trị giới hạn, còn gọi là máy trạng thái giới hạn (Finite-state machines).
- Mạch logic tuần tự còn được gọi là hệ có nhớ.
- Để thực hiện được mạch tuần tự, nhất thiết phải có phần tử nhớ. Ngoài ra, còn có thể có các phần tử logic cơ bản.

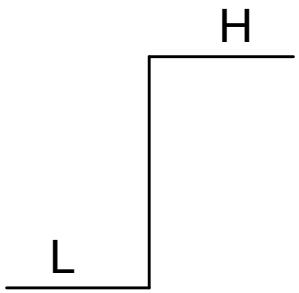
8.2 Flip Flop

- Là phần tử cơ bản của hệ tuần tự.
- Đầu ra của FF chính là trạng thái của nó
- Một FF có thể làm việc theo 2 kiểu:
  - Không đồng bộ: đầu ra của FF thay đổi chỉ phụ thuộc vào tín hiệu đầu vào
  - Đồng bộ: đầu ra của FF thay đổi phụ thuộc tín hiệu vào và tín hiệu đồng bộ.
    - Đồng bộ theo mức
    - Đồng bộ theo sườn
    - Đồng bộ theo xung



Đồng bộ theo mức

- Mức cao:
  - Khi tín hiệu đồng bộ có giá trị logic = 0 thì hệ nghỉ (giữ nguyên trạng thái)
  - Khi tín hiệu đồng bộ có giá trị logic = 1 thì hệ làm việc bình thường.

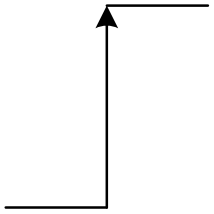


Đồng bộ theo mức

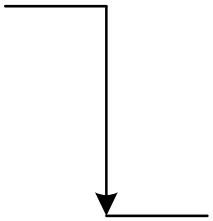
- Mức thấp
  - Khi tín hiệu đồng bộ có giá trị logic = 1 thì hệ nghỉ (giữ nguyên trạng thái)
  - Khi tín hiệu đồng bộ có giá trị logic = 0 thì hệ làm việc bình thường.

Đồng bộ theo sườn

- Sườn dương:
  - Khi tín hiệu đồng bộ xuất hiện sườn dương (sườn đi lên, từ 0 → 1) thì hệ làm việc bình thường
  - Trong các trường hợp còn lại, hệ nghỉ (giữ nguyên trạng thái).
- Sườn âm:
  - Khi tín hiệu đồng bộ xuất hiện sườn âm (sườn đi xuống, từ 1 → 0), hệ làm việc bình thường
  - Trong các trường hợp còn lại, hệ nghỉ (giữ nguyên trạng thái).



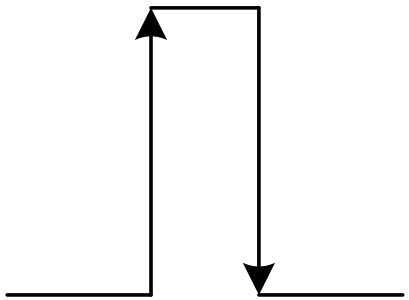
Đồng bộ theo sườn dương



Đồng bộ theo sườn âm

Đồng bộ kiểu xung

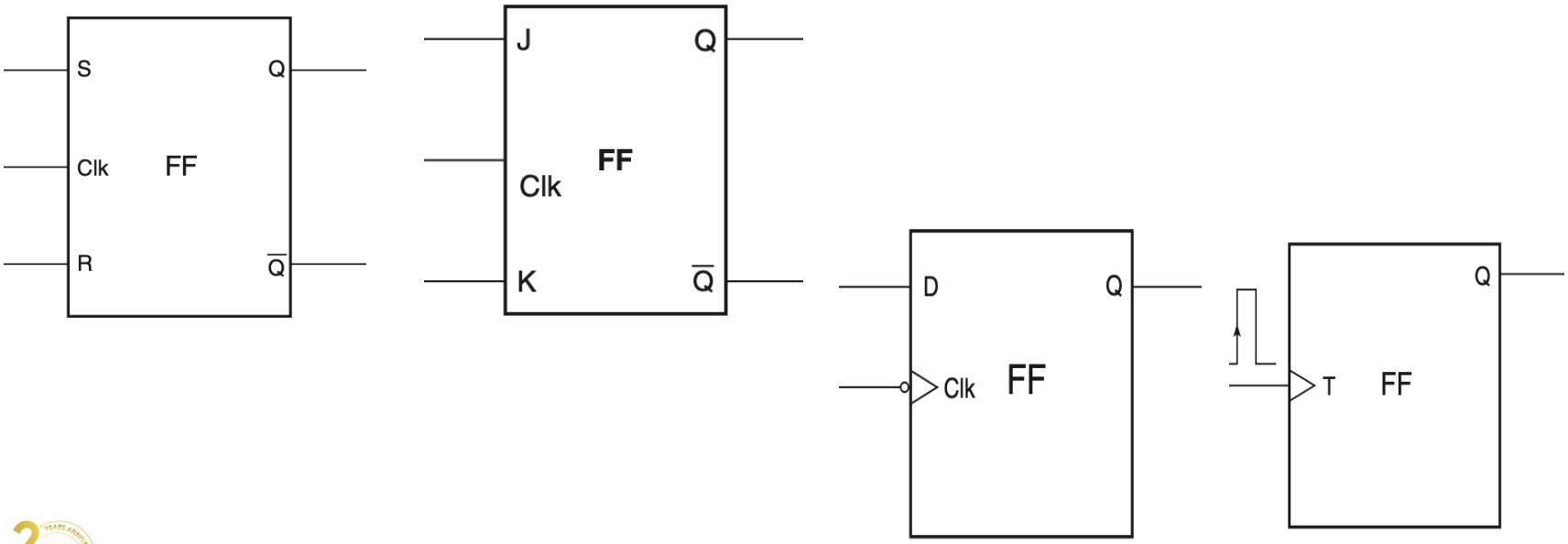
- Khi có xung thì hệ làm việc bình thường
- Khi không có xung thì hệ nghỉ (giữ nguyên trạng thái).



Đồng bộ kiểu xung

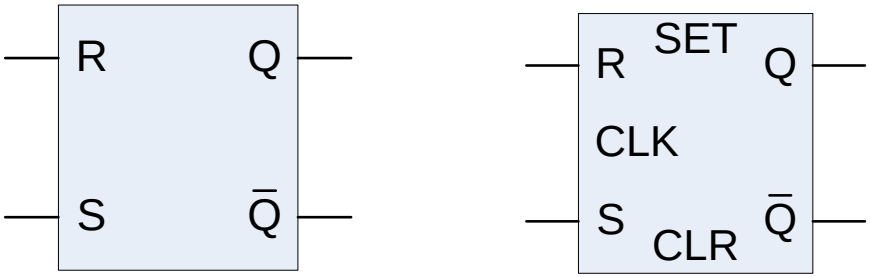
8.3 Phân loại Flip Flop

- Có 4 loại FF:
  - RS Reset - Set Xóa - Thiết lập
  - JK Jordan và Kelly Tên nhà phát minh
  - D Delay Trễ
  - T Toggle Đảo

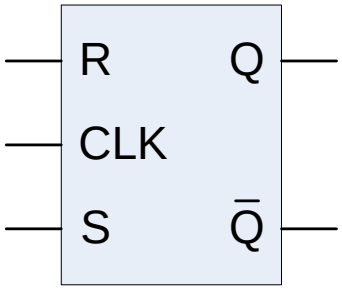


FF RS

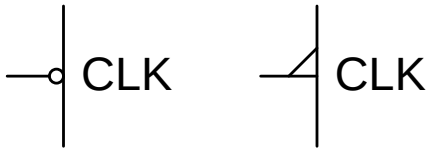
- Sơ đồ khối:



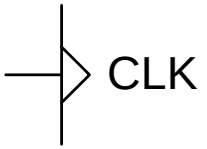
- FF RS hoạt động được ở cả 2 chế độ đồng bộ và không đồng bộ



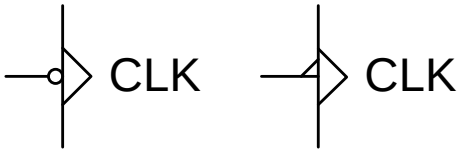
Đồng bộ mức cao



Đồng bộ mức thấp



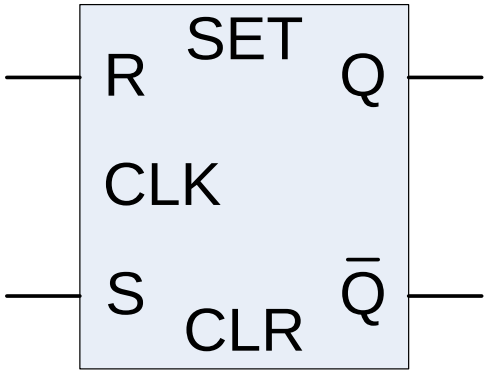
Đồng bộ sườn dương



Đồng bộ sườn âm

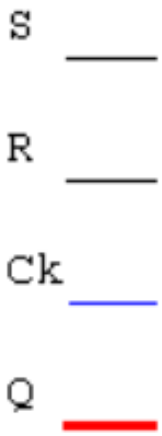
FF RS

- Xung tín hiệu đầu ra:



RS / q	00	01	11	10
0	0	1	-	0
1	1	1	-	0
	nhớ	thiết lập	không xác định	xóa

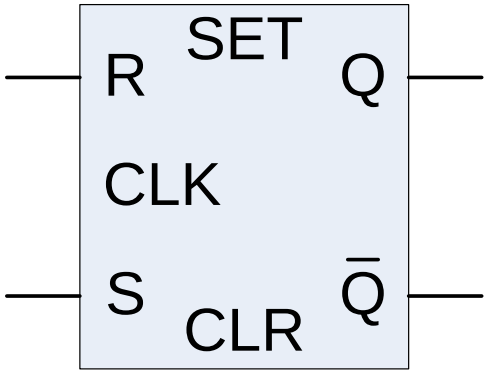
$$Q = S + q\bar{R}$$



S	R	Ck	Q	$\bar{Q}$	Trạng thái
0	0	↑	Q_0	$\bar{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	!	!	Cấm

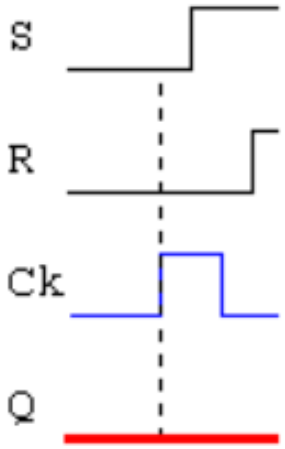
FF RS

- Xung tín hiệu đầu ra:



RS / q	00	01	11	10
0	0	1	-	0
1	1	1	-	0
	nhớ	thiết lập	không xác định	xóa

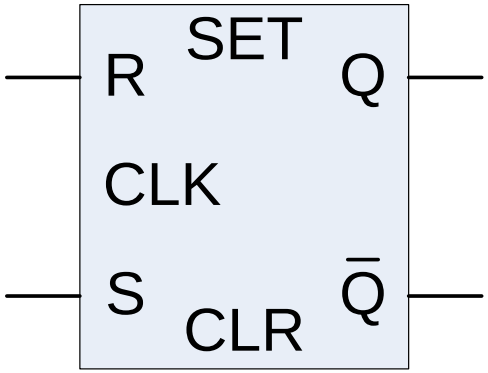
$$Q = S + q\overline{R}$$



S	R	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_0	$\overline{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	!	!	Cấm

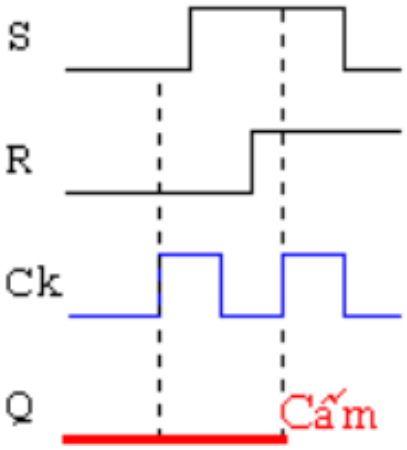
FF RS

- Xung tín hiệu đầu ra:



RS / q	00	01	11	10
0	0	1	-	0
1	1	1	-	0
	nhớ	thiết lập	không xác định	xóa

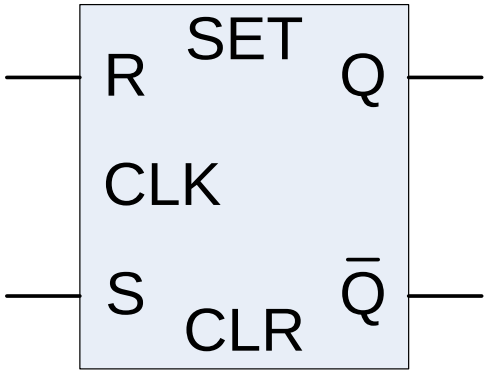
$$Q = S + q\bar{R}$$



S	R	Ck	Q	$\bar{Q}$	Trạng thái
0	0	↑	Q_0	$\bar{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	!	!	Cấm

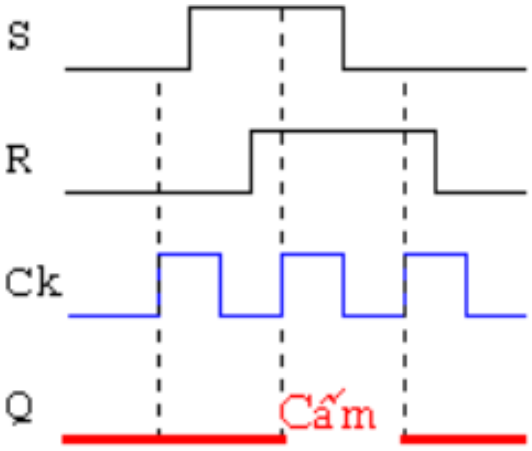
FF RS

- Xung tín hiệu đầu ra:



q \ RS	00	01	11	10
0	0	1	-	0
1	1	1	-	0
	nhớ	thiết lập	không xác định	xóa

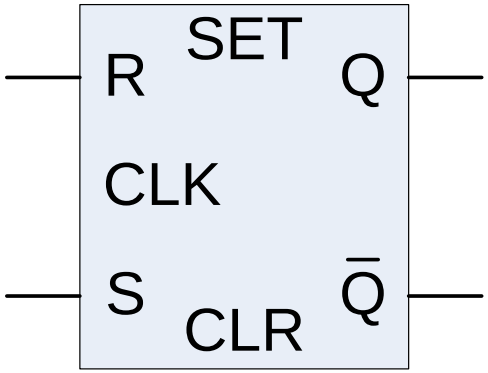
$$Q = S + q\bar{R}$$



S	R	Ck	Q	$\bar{Q}$	Trạng thái
0	0	↑	Q_0	$\bar{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	!	!	Cấm

FF RS

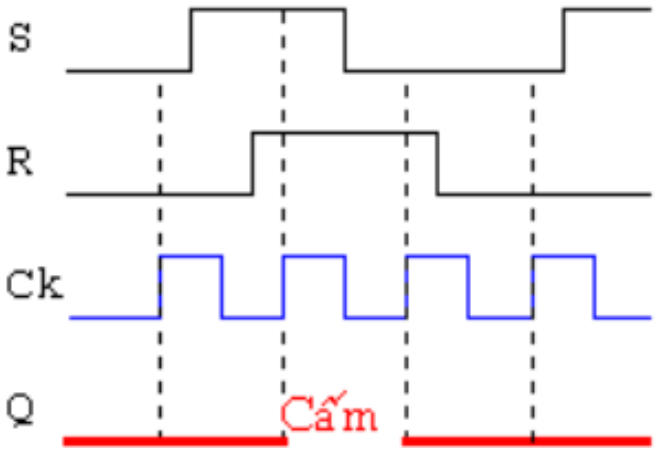
- Xung tín hiệu đầu ra:



RS / q	00	01	11	10
0	0	1	-	0
1	1	1	-	0

nhớ | thiết lập | không xác định | xóa

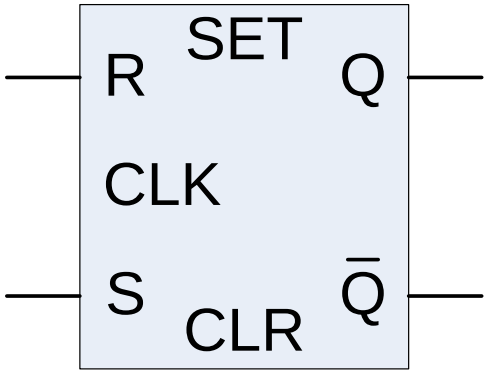
$$Q = S + q\bar{R}$$



S	R	Ck	Q	$\bar{Q}$	Trạng thái
0	0	↑	Q_0	$\bar{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	!	!	Cấm

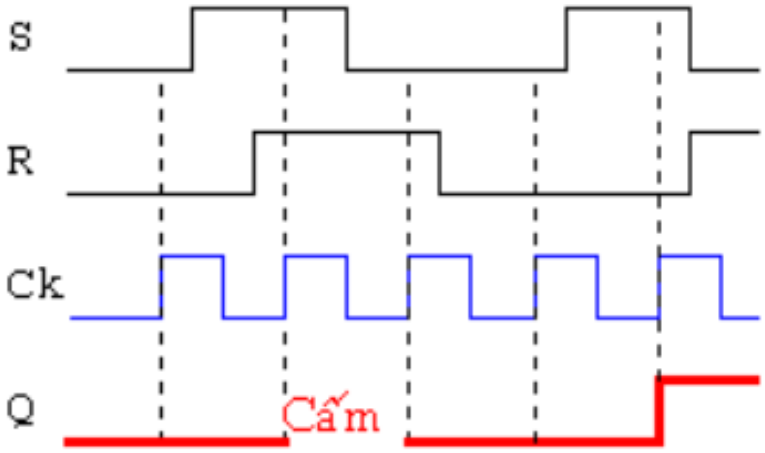
FF RS

- Xung tín hiệu đầu ra:



RS / q	00	01	11	10
0	0	1	-	0
1	1	1	-	0
	nhớ	thiết lập	không xác định	xóa

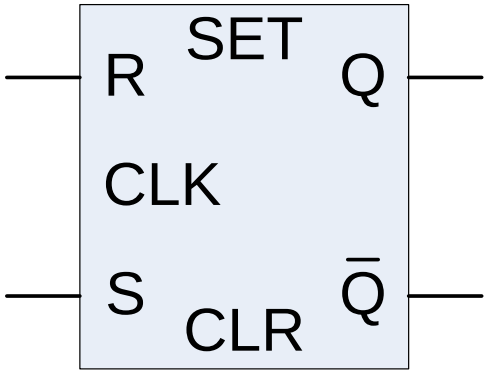
$$Q = S + q\bar{R}$$



S	R	Ck	Q	$\bar{Q}$	Trạng thái
0	0	↑	Q_0	$\bar{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	!	!	Cấm

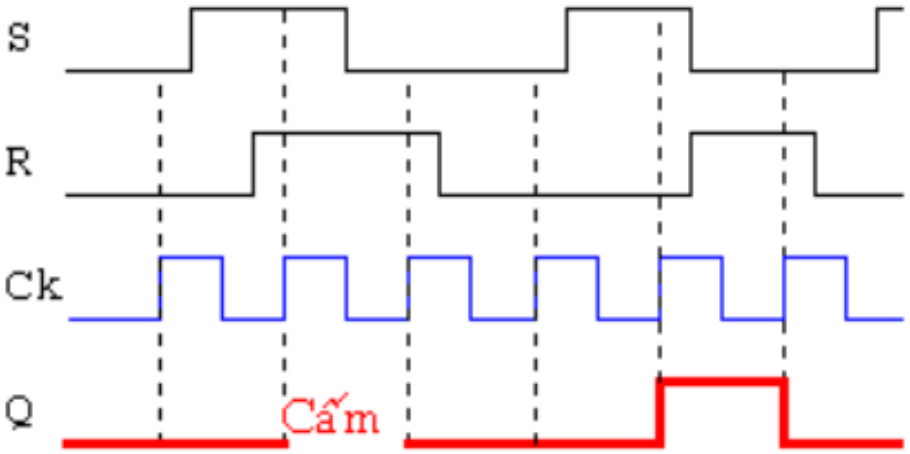
FF RS

- Xung tín hiệu đầu ra:



RS / q	00	01	11	10
0	0	1	-	0
1	1	1	-	0
	nhớ	thiết lập	không xác định	xóa

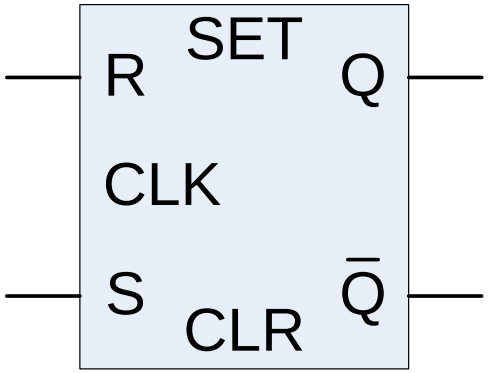
$$Q = S + q\bar{R}$$



S	R	Ck	Q	$\bar{Q}$	Trạng thái
0	0	↑	Q_0	$\bar{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	!	!	Cấm

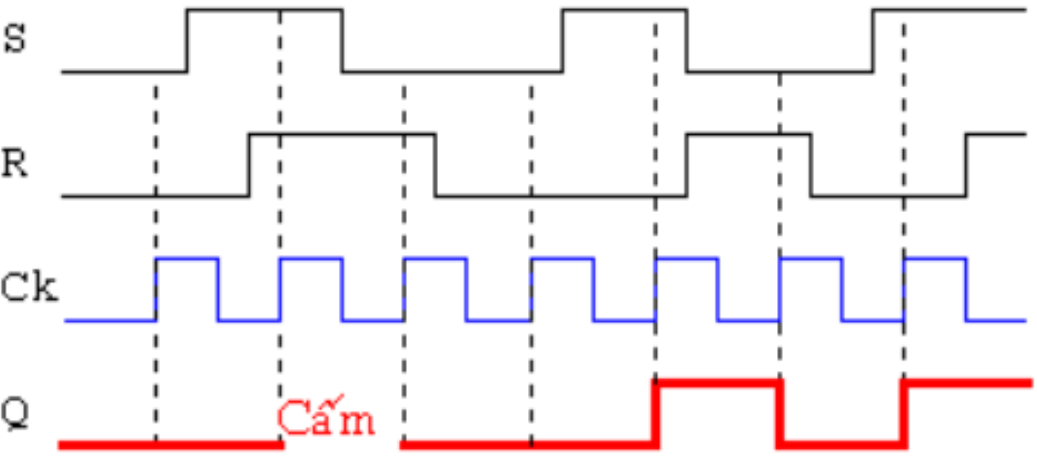
FF RS

- Xung tín hiệu đầu ra:



RS / q	00	01	11	10
0	0	1	-	0
1	1	1	-	0
	nhớ	thiết lập	không xác định	xóa

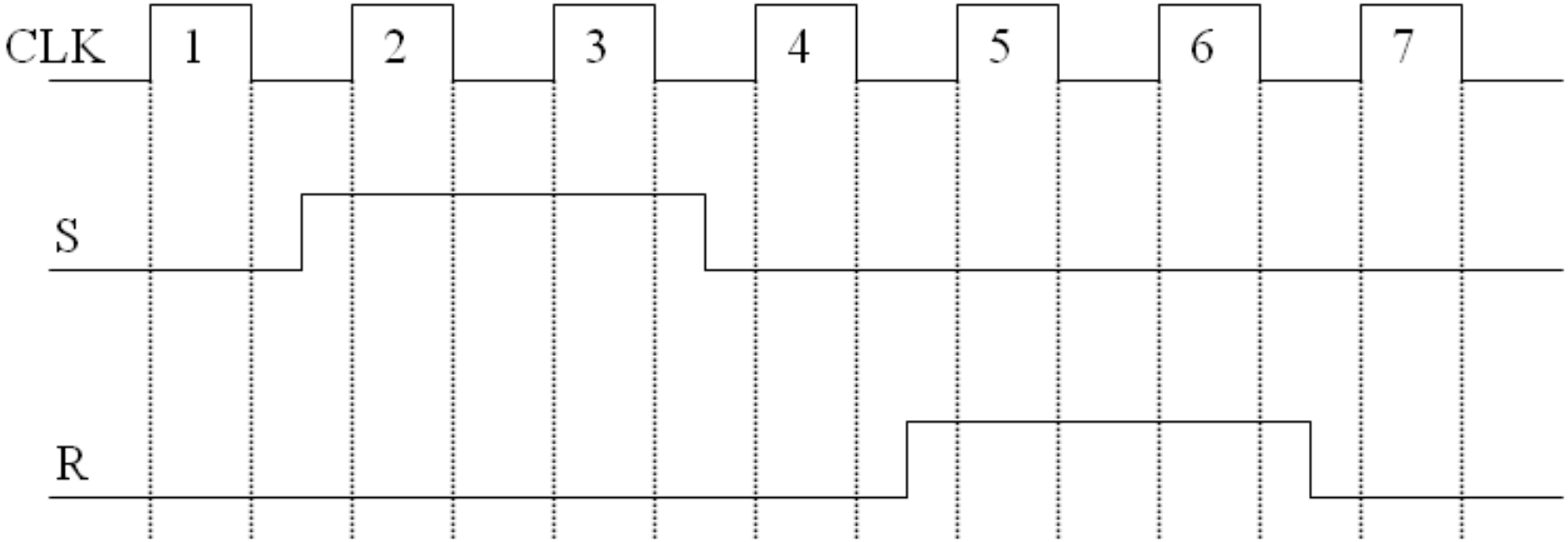
$$Q = S + q\bar{R}$$



S	R	Ck	Q	$\bar{Q}$	Trạng thái
0	0	↑	Q_0	$\bar{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	!	!	Cấm

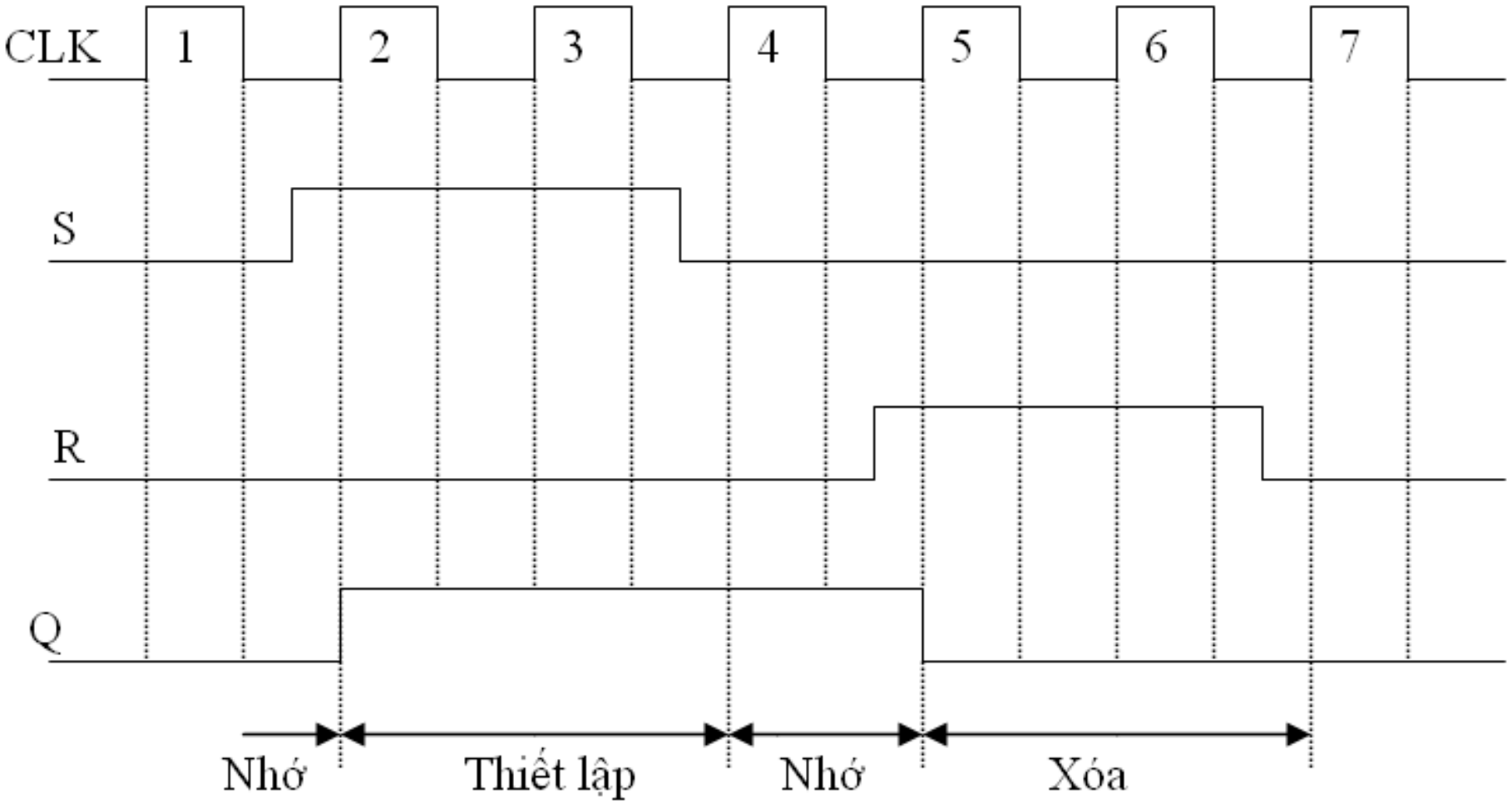
Ví dụ 8.1

- Cho FF RS đồng bộ mức cao và đồ thị các tín hiệu R, S như hình vẽ. Hãy vẽ đồ thị tín hiệu ra Q.



Q

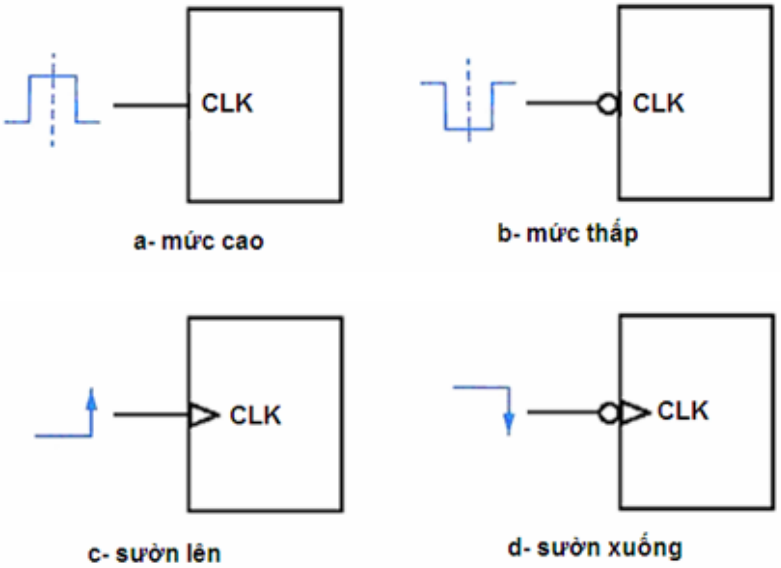
Ví dụ 8.1



RS FF với đầu vào có tín hiệu đồng bộ

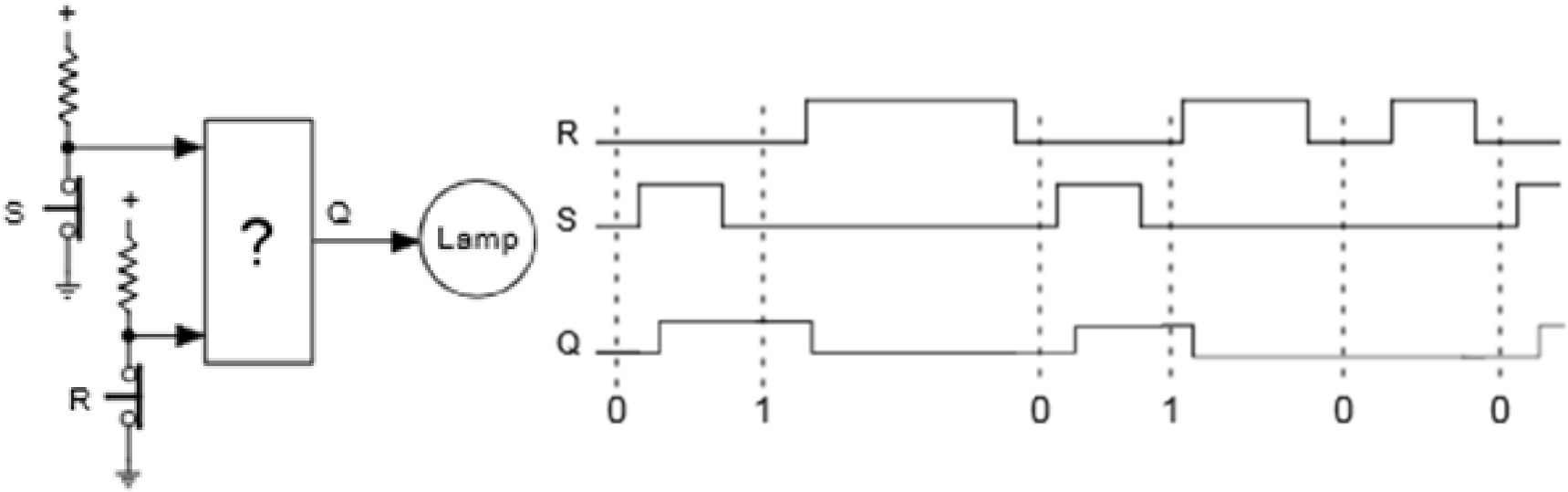
- Các FF thường được kích hoạt để nhận thông tin nhờ một tín hiệu đồng bộ.
- Tín hiệu đồng bộ có thể tích cực:
  - Theo mức (cao, thấp)
  - Theo sườn (lên, xuống)
- FF chỉ có thể trao đổi thông tin khi tín hiệu đồng bộ tích cực và ngược lại.

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



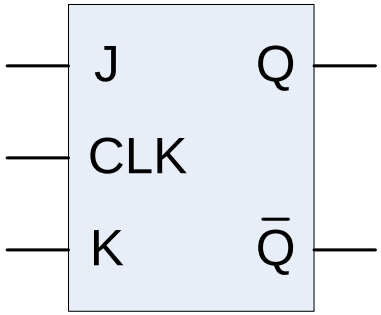
Ví dụ 8.2

- Thiết kế một mạch điều khiển đèn từ 2 nút bấm kí hiệu là S và R. Nếu bấm vào S thì đèn sáng, nhả S đèn vẫn sáng. Nếu bấm vào R thì đèn tắt và nhả R thì đèn vẫn tắt.
- Giả thiết S và R không được bấm đồng thời.

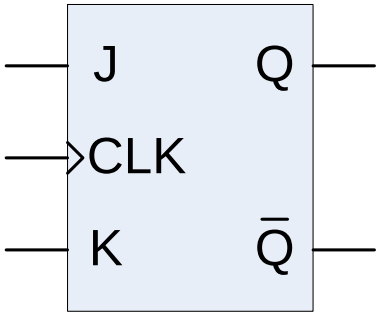


FF JK

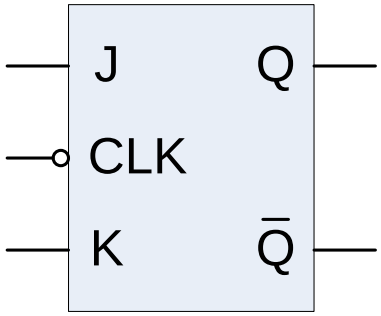
- JK FF chỉ hoạt động ở chế độ đồng bộ
- Sơ đồ khối:



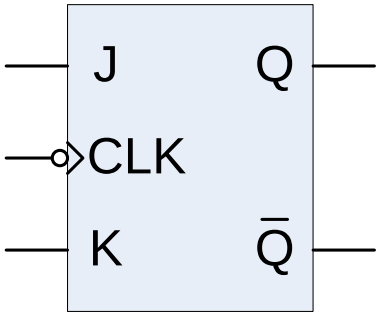
Tích cực mức cao



Tích cực sườn dương



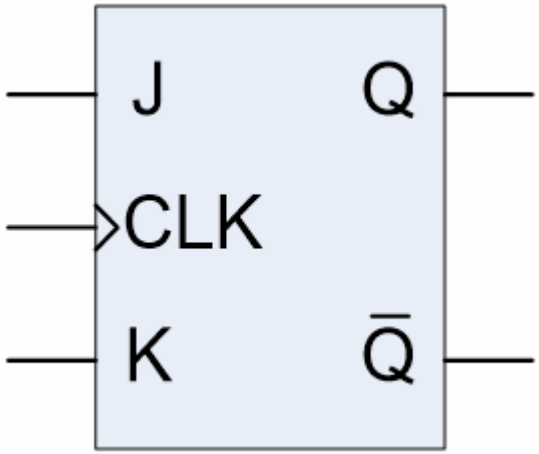
Tích cực mức thấp



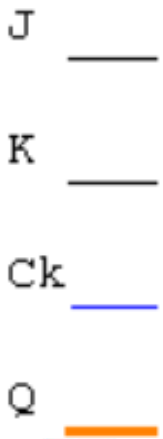
Tích cực sườn âm

FF JK

- Xung tín hiệu ra:



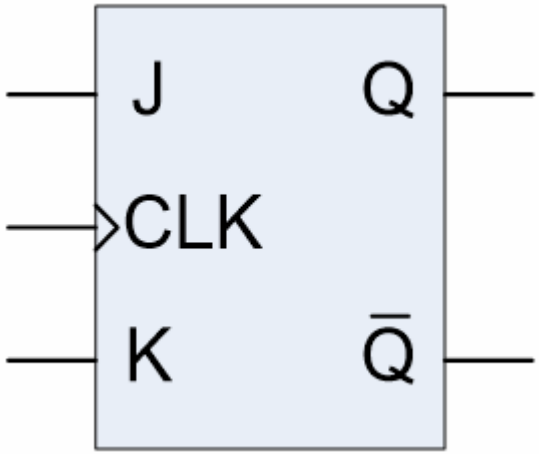
Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$



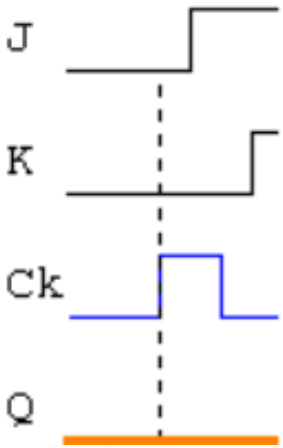
J	K	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_0	$\overline{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	$\overline{Q_0}$	Q_0	Lật

FF JK

- Xung tín hiệu ra:



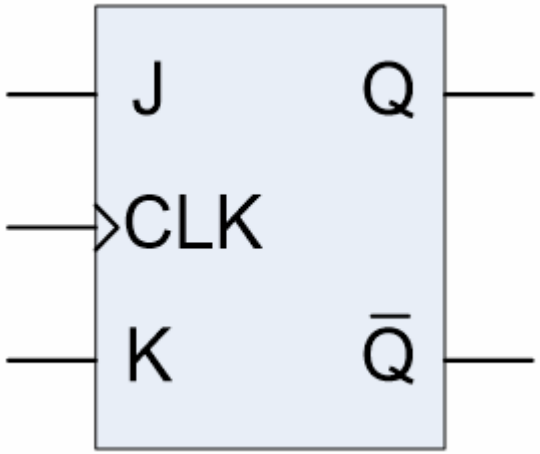
Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$



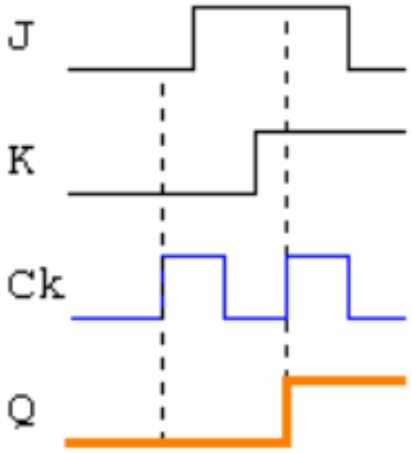
J	K	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_0	$\overline{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	$\overline{Q_0}$	Q_0	Lật

FF JK

- Xung tín hiệu ra:



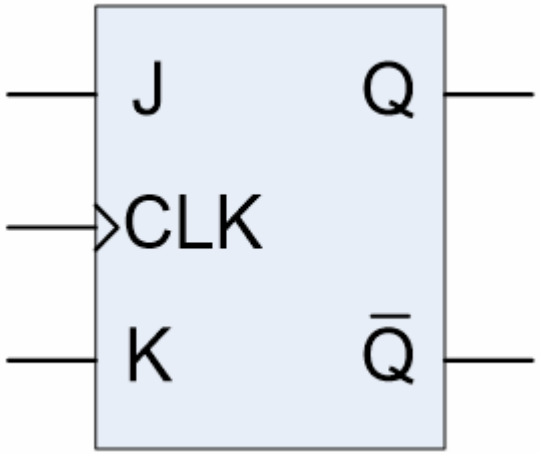
Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$



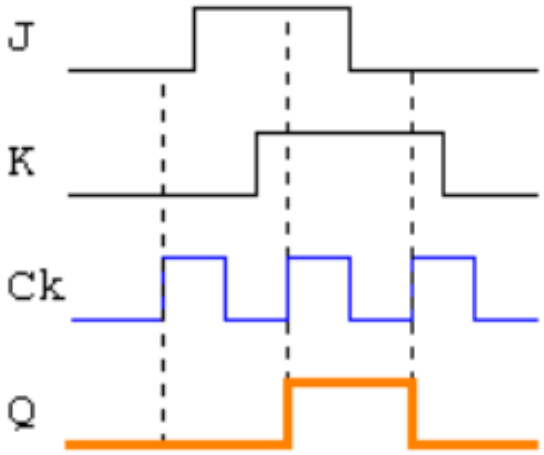
J	K	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_0	$\overline{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	$\overline{Q_0}$	Q_0	Lật

FF JK

- Xung tín hiệu ra:



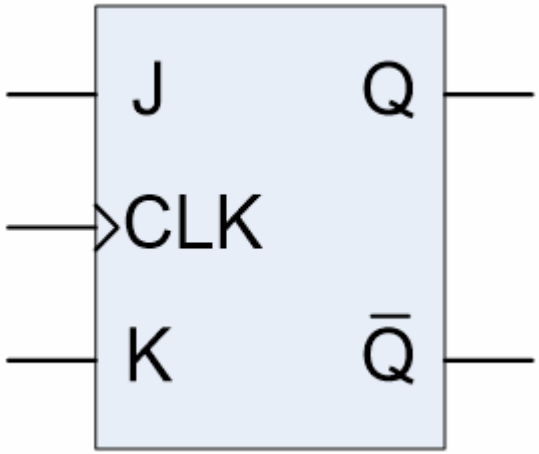
Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$

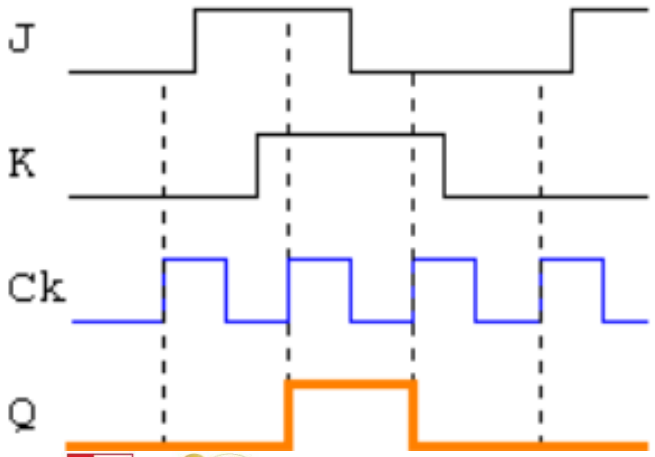


J	K	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_0	$\overline{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	$\overline{Q_0}$	Q_0	Lật

FF JK

- Xung tín hiệu ra:



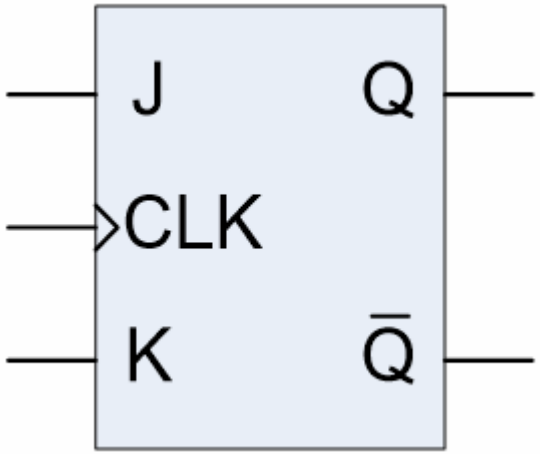


Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$

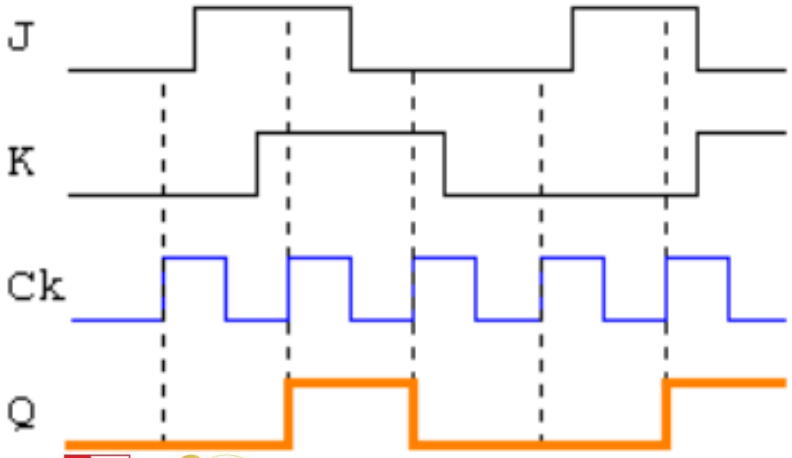
J	K	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_o	$\overline{Q_o}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	$\overline{Q_o}$	Q_o	Lật

FF JK

- Xung tín hiệu ra:



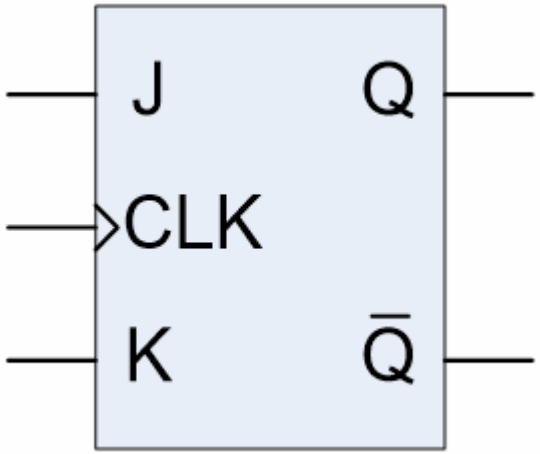
Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$



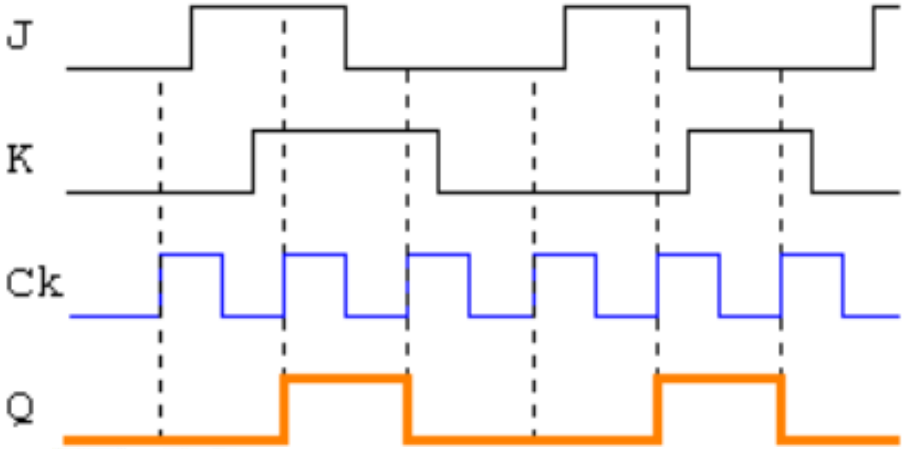
J	K	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_o	$\overline{Q_o}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	$\overline{Q_o}$	Q_o	Lật

FF JK

- Xung tín hiệu ra:



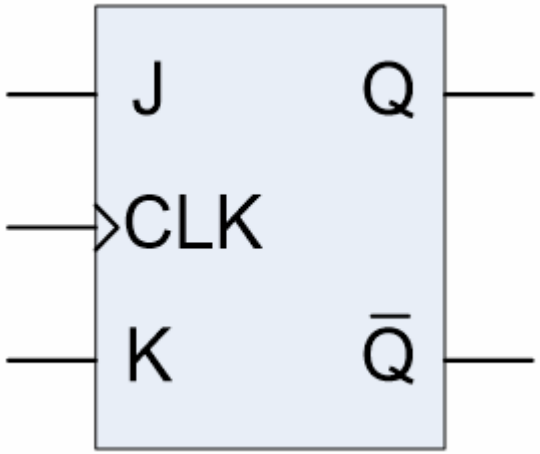
Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$



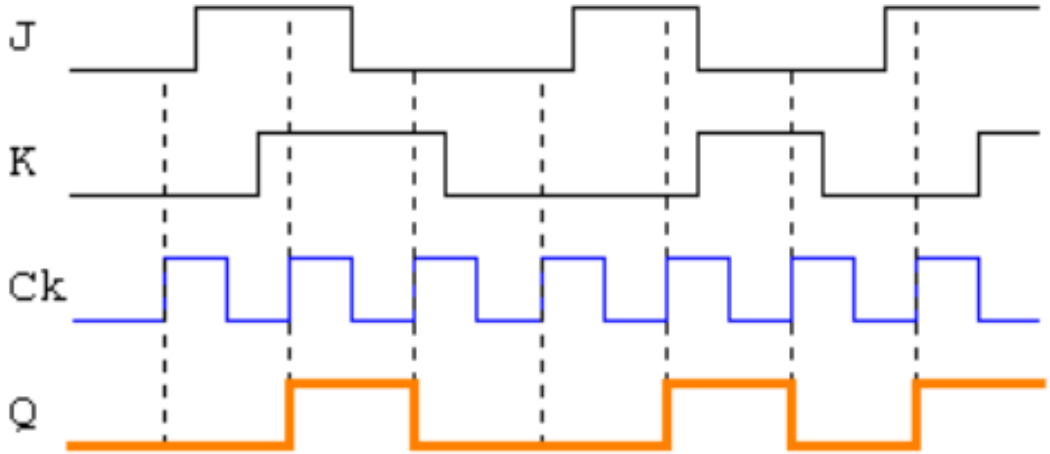
J	K	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_0	$\overline{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	$\overline{Q_0}$	Q_0	Lật

FF JK

- Xung tín hiệu ra:



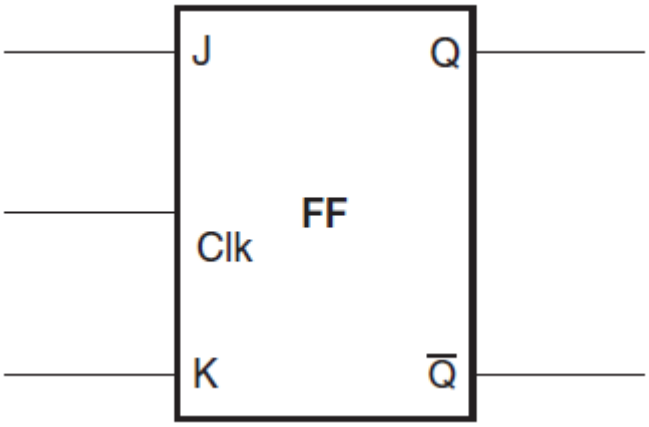
Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$



J	K	Ck	Q	$\overline{Q}$	Trạng thái
0	0	↑	Q_0	$\overline{Q_0}$	Không đổi
0	1	↑	0	1	Xoá
1	0	↑	1	0	Đặt
1	1	↑	$\overline{Q_0}$	Q_0	Lật

FF JK với đầu vào có tín hiệu đồng bộ

- JK FF với đầu vào tích cực ở mức CAO

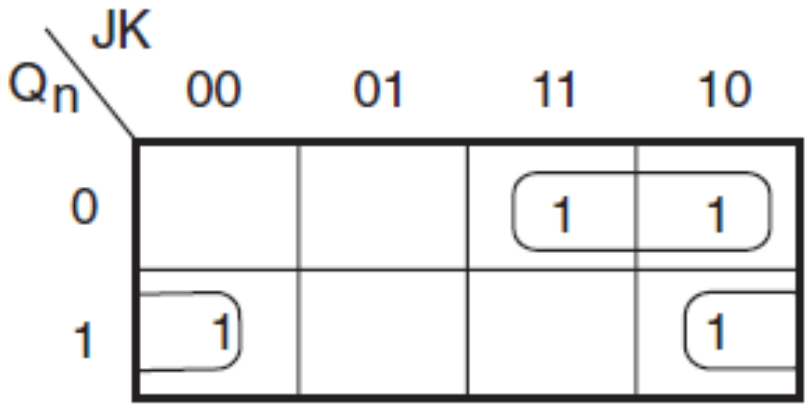


Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q_n}$

FF JK với đầu vào có tín hiệu đồng bộ

- JK FF với đầu vào tích cực ở mức CAO

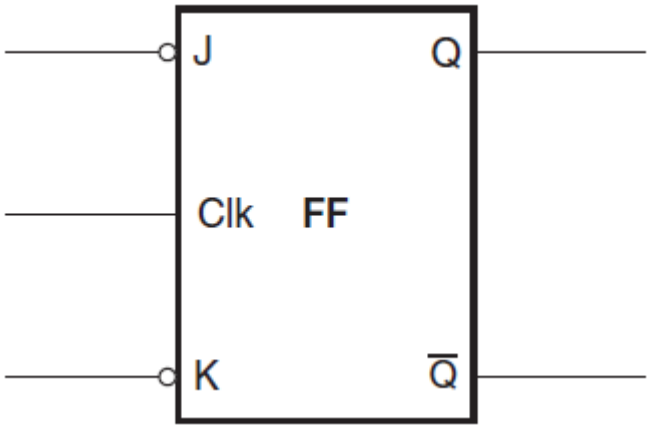
Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



$$Q_{n+1} = J.\overline{Q_n} + \overline{K}.Q_n$$

JK Flip Flop với đầu vào có tín hiệu đồng bộ

- JK FF với đầu vào tích cực ở mức THẤP

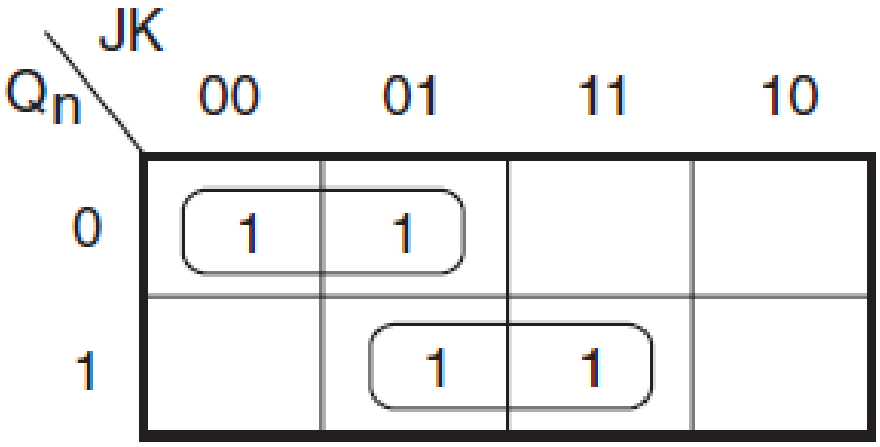


Operation Mode	J	K	Clk	Q_{n+1}
SET	0	1	1	1
RESET	1	0	1	0
NO CHANGE	1	1	1	Q_n
TOGGLE	0	0	1	$\overline{Q}_n$

FF JK với đầu vào có tín hiệu đồng bộ

- JK FF với đầu vào tích cực ở mức THẤP

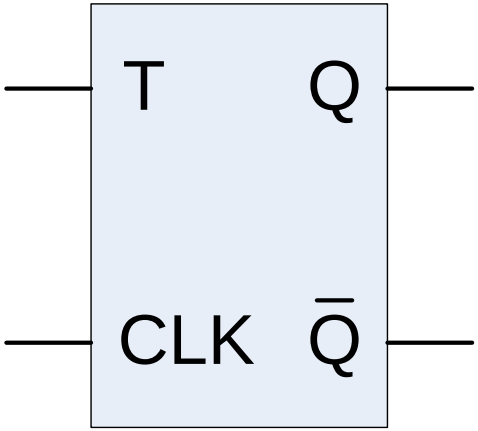
Q_n	J	K	Q_{n+1}
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



$$Q_{n+1} = \overline{J}.\overline{Q_n} + K.Q_n$$

FF T

- FF T (FF đảo) chỉ hoạt động ở chế độ đồng bộ.
- FF T thay đổi trạng thái mỗi khi được kích thích tại đầu vào T (đầu vào đảo).
- Sơ đồ khối:



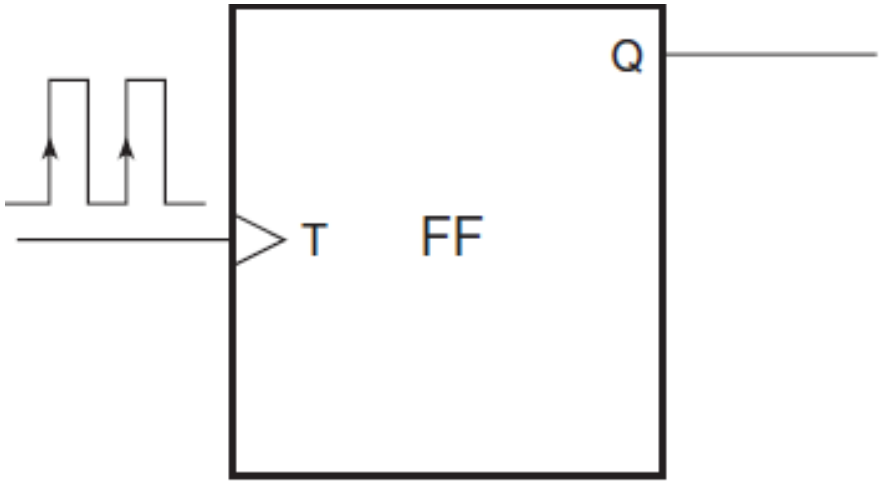
Đầu vào đảo kích thích cạnh dương

- Bảng thật và sơ đồ khối:

Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Q_n \ T	0	1
0		1
1	1	

$$Q_{n+1} = T.\overline{Q_n} + \overline{T}.Q_n$$



T	Q_n	Q_{n+1}
↑	0	1
↑	1	0

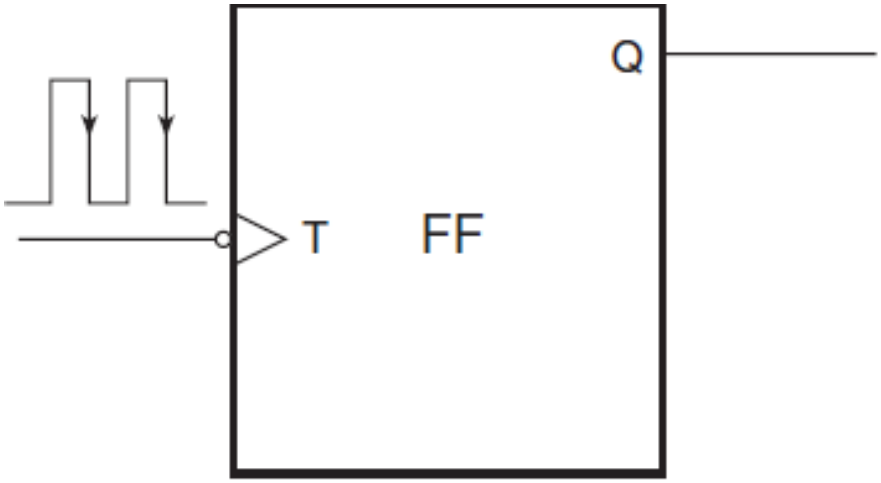
Đầu vào đảo kích thích cạnh âm

- Bảng thật và sơ đồ khối:

Q_n	T	Q_{n+1}
0	0	1
0	1	0
1	0	0
1	1	1

Q_n \ T	0	1
0	1	
1		1

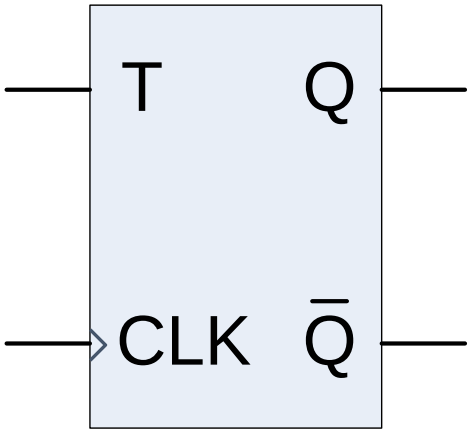
$$Q_{n+1} = \overline{T}.\overline{Q_n} + T.Q_n$$



T	Q_n	Q_{n+1}
↓	0	1
↓	1	0

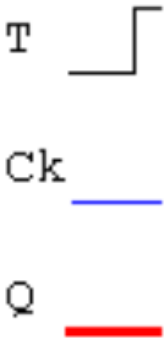
FF T

- Xung tín hiệu ra:



q \ T	0	1
0	0	1
1	1	0
	nhớ	lật

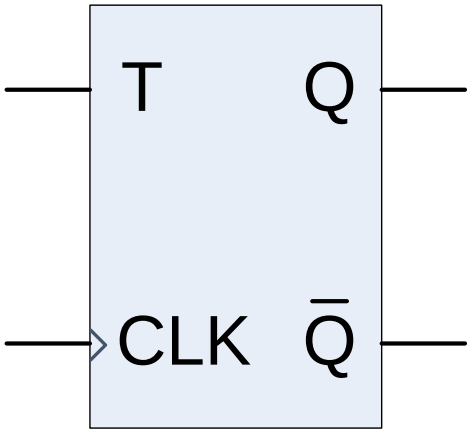
$$Q = \overline{q}T + q\overline{T} = q \oplus T$$



T	Ck	Q	$\overline{Q}$	Trạng thái
0	↑	Q_0	$\overline{Q_0}$	Không đổi
1	↑	$\overline{Q_0}$	Q_0	lật

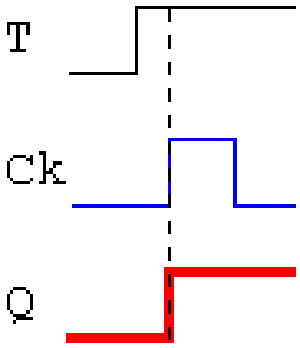
FF T

- Xung tín hiệu ra:



q \ T	0	1
0	0	1
1	1	0
	nhớ	lật

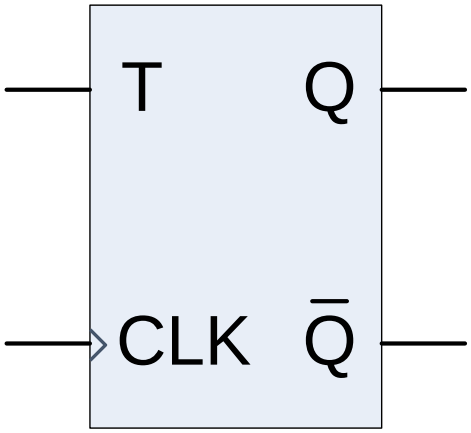
$$Q = \overline{q}T + q\overline{T} = q \oplus T$$



T	Ck	Q	$\overline{Q}$	Trạng thái
0	↑	Q_0	$\overline{Q_0}$	Không đổi
1	↑	$\overline{Q_0}$	Q_0	lật

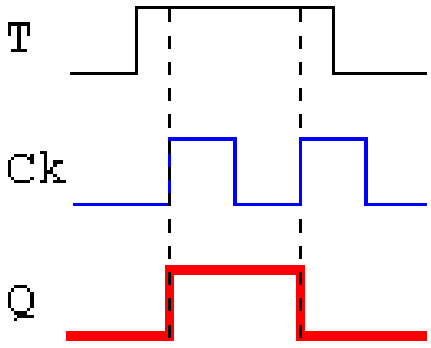
FF T

- Xung tín hiệu ra:



q \ T	0	1
0	0	1
1	1	0
	nhớ	lật

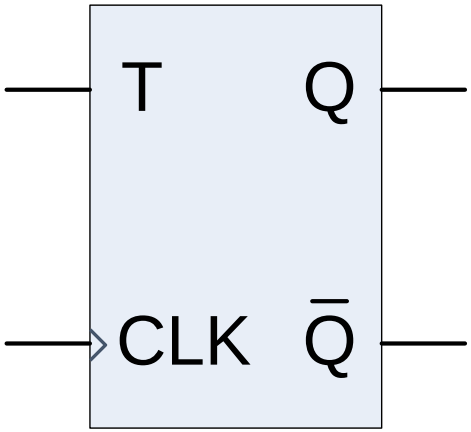
$$Q = \bar{q}T + q\bar{T} = q \oplus T$$



T	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	Q_0	$\bar{Q_0}$	Không đổi
1	↑	$\bar{Q_0}$	Q_0	lật

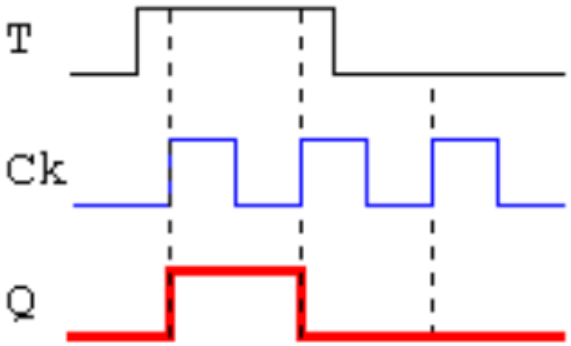
FF T

- Xung tín hiệu ra:



q \ T	0	1
0	0	1
1	1	0
	nhớ	lật

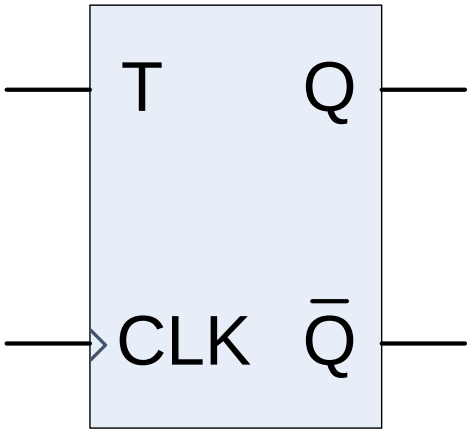
$$Q = \overline{q}T + q\overline{T} = q \oplus T$$



T	Ck	Q	$\overline{Q}$	Trạng thái
0	↑	Q_0	$\overline{Q_0}$	Không đổi
1	↑	$\overline{Q_0}$	Q_0	lật

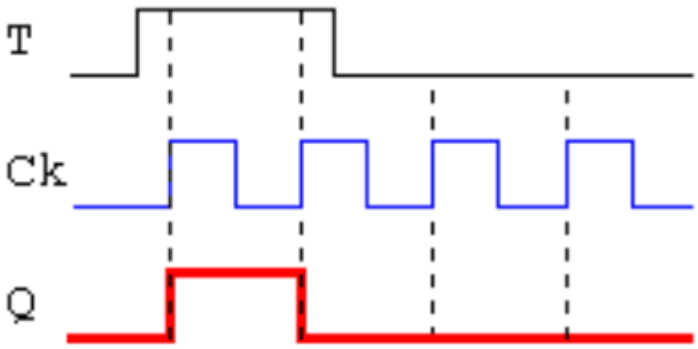
FF T

- Xung tín hiệu ra:



q \ T	0	1
0	0	1
1	1	0
	nhớ	lật

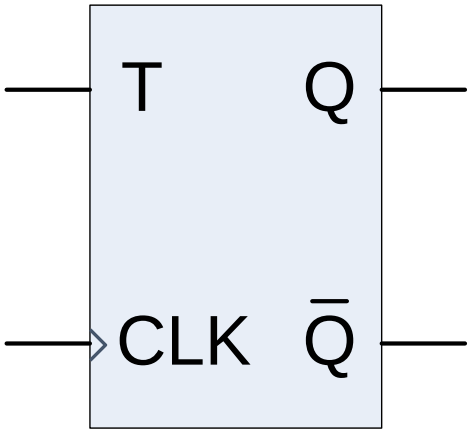
$$Q = \overline{q}T + q\overline{T} = q \oplus T$$



T	Ck	Q	$\overline{Q}$	Trạng thái
0	↑	Q_0	$\overline{Q_0}$	Không đổi
1	↑	$\overline{Q_0}$	Q_0	lật

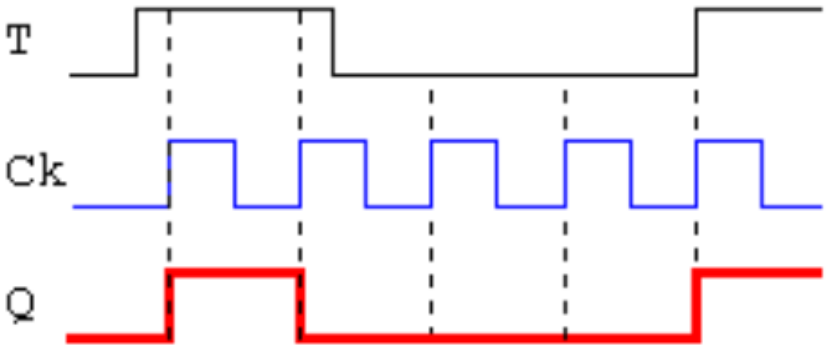
FF T

- Xung tín hiệu ra:



q \ T	0	1
0	0	1
1	1	0
	nhớ	lật

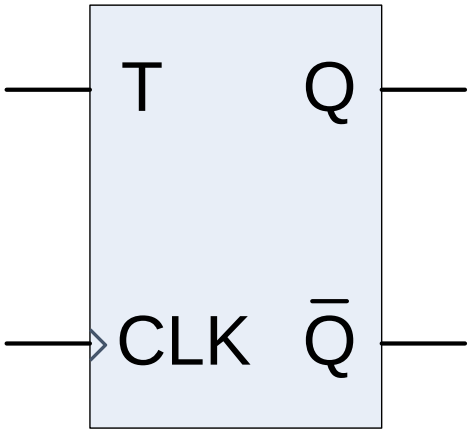
$$Q = \bar{q}T + q\bar{T} = q \oplus T$$



T	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	Q_0	$\bar{Q_0}$	Không đổi
1	↑	$\bar{Q_0}$	Q_0	lật

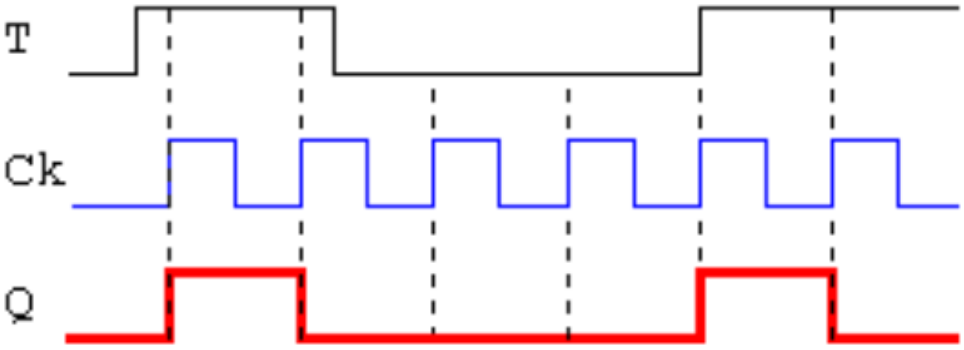
FF T

- Xung tín hiệu ra:



q \ T	0	1
0	0	1
1	1	0
	nhớ	lật

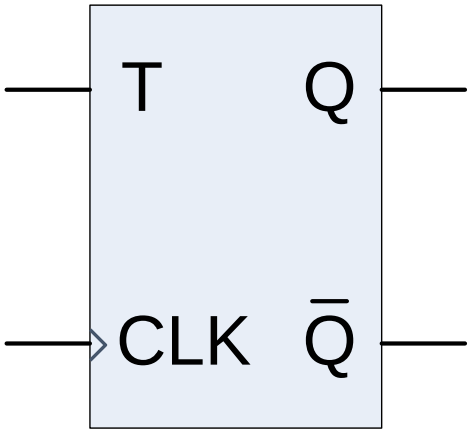
$$Q = \overline{q}T + q\overline{T} = q \oplus T$$



T	Ck	Q	$\overline{Q}$	Trạng thái
0	↑	Q_0	$\overline{Q_0}$	Không đổi
1	↑	$\overline{Q_0}$	Q_0	lật

FF T

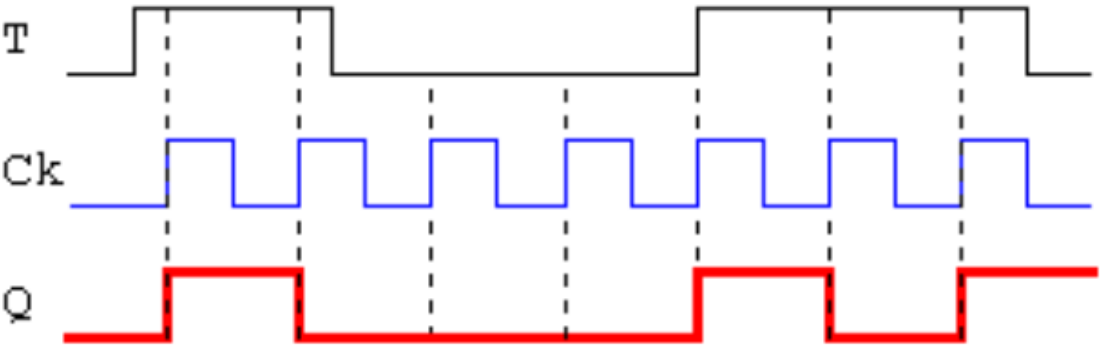
- Xung tín hiệu ra:



q \ T	0	1
0	0	1
1	1	0

nhớ lật

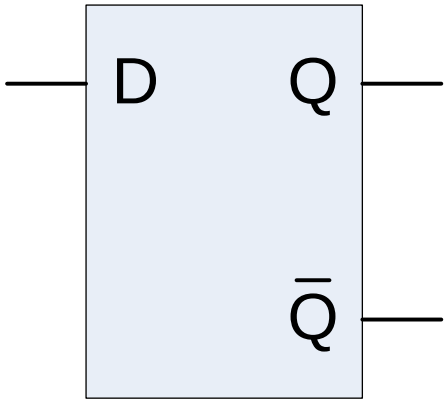
$$Q = \overline{q}T + q\overline{T} = q \oplus T$$



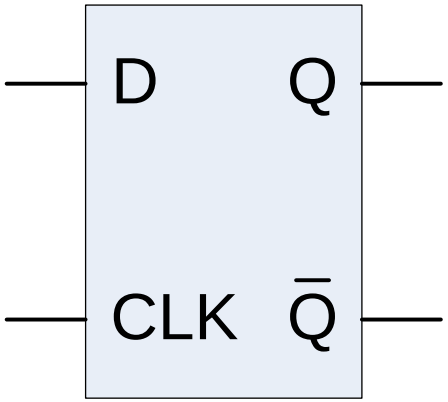
T	Ck	Q	$\overline{Q}$	Trạng thái
0	↑	Q_0	$\overline{Q_0}$	Không đổi
1	↑	$\overline{Q_0}$	Q_0	lật

FF D

- FF D còn gọi là FF trễ, có thể được sử dụng để lưu trữ thông tin tạm thời của 1 bit.
- FF D có 1 đầu vào là D và hoạt động ở 2 chế độ đồng bộ và không đồng bộ.
- Chỉ xét FF D hoạt động ở chế độ đồng bộ.



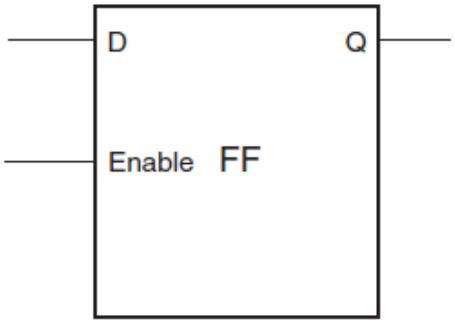
Không đồng bộ



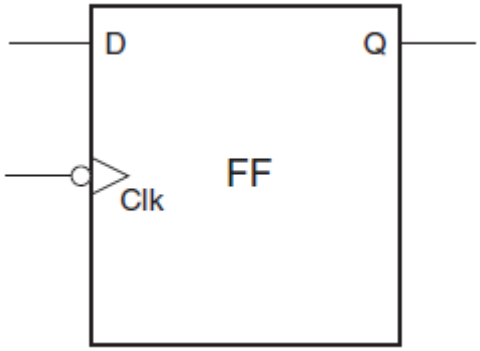
Đồng bộ

FF D

- D FF đồng bộ theo mức gọi là chốt D (Latch)

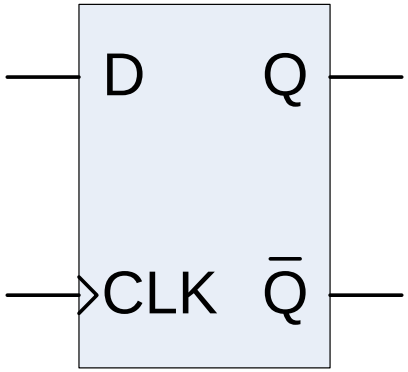


- D FF đồng bộ theo sườn được gọi là xuất phát sườn (Edge trigged)

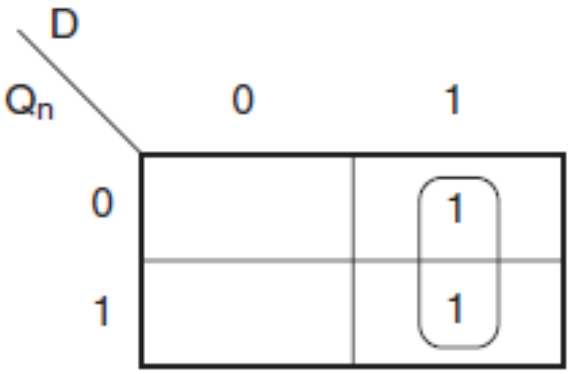


FF D

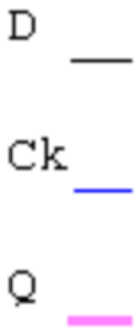
- Xung tín hiệu đầu ra:



Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1



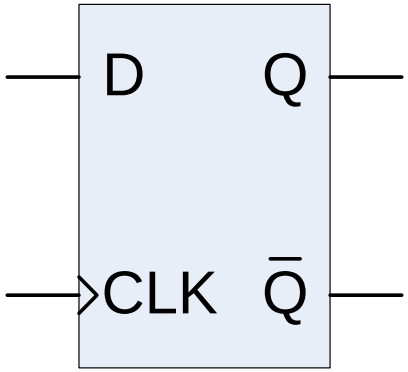
$$Q_{n+1} = D$$



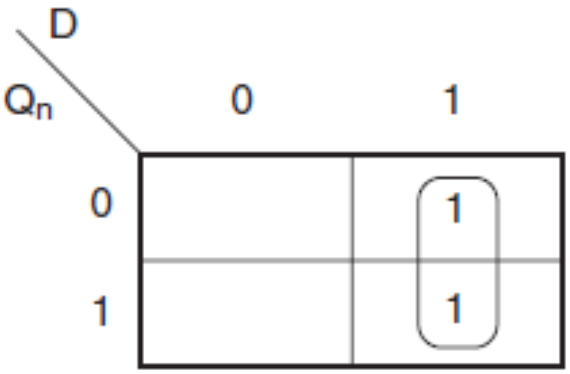
D	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	0	1	Xoá
1	↑	1	0	Đặt

FF D

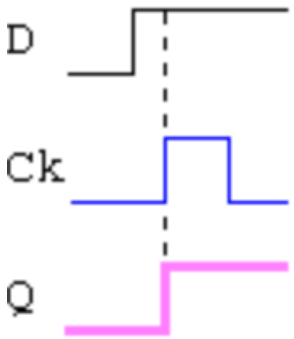
- Xung tín hiệu đầu ra:



Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1



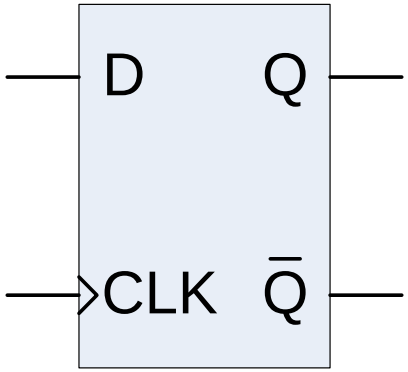
$$Q_{n+1} = D$$



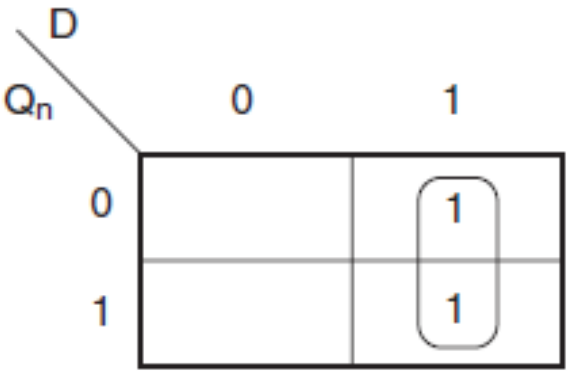
D	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	0	1	Xoá
1	↑	1	0	Đặt

FF D

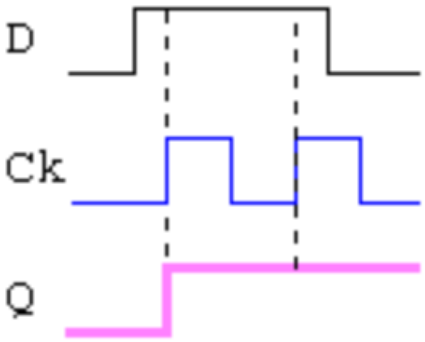
- Xung tín hiệu đầu ra:



Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1



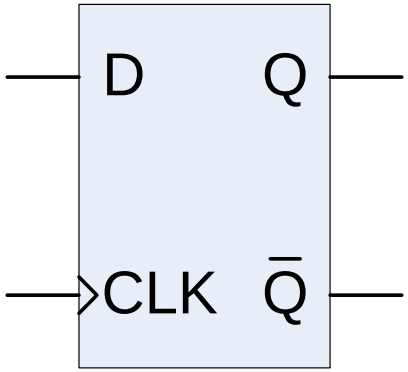
$$Q_{n+1} = D$$



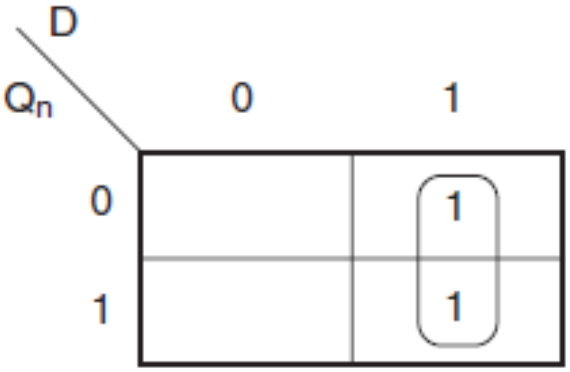
D	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	0	1	Xoá
1	↑	1	0	Đặt

FF D

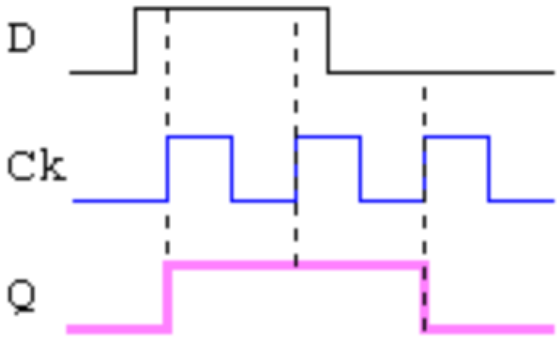
- Xung tín hiệu đầu ra:



Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1



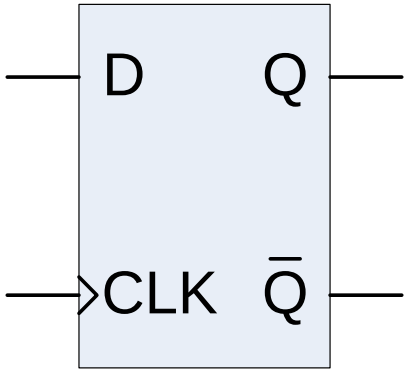
$$Q_{n+1} = D$$



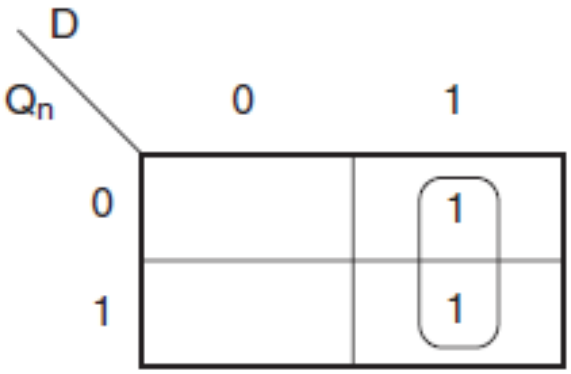
D	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	0	1	Xoá
1	↑	1	0	Đặt

FF D

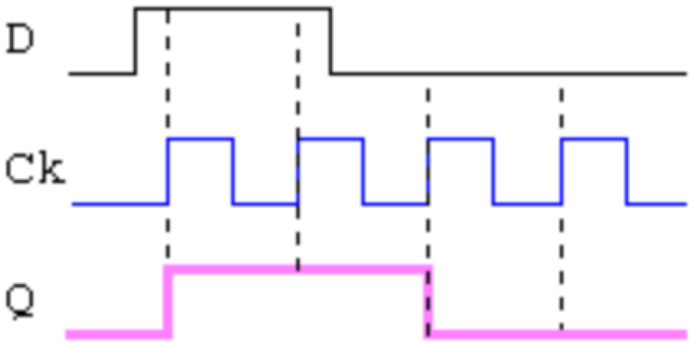
- Xung tín hiệu đầu ra:



Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1



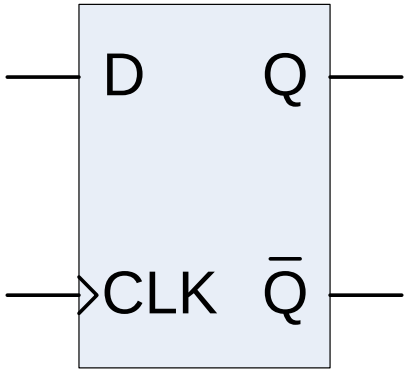
$$Q_{n+1} = D$$



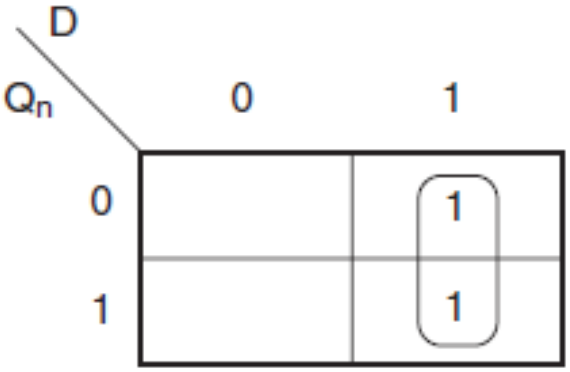
D	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	0	1	Xoá
1	↑	1	0	Đặt

FF D

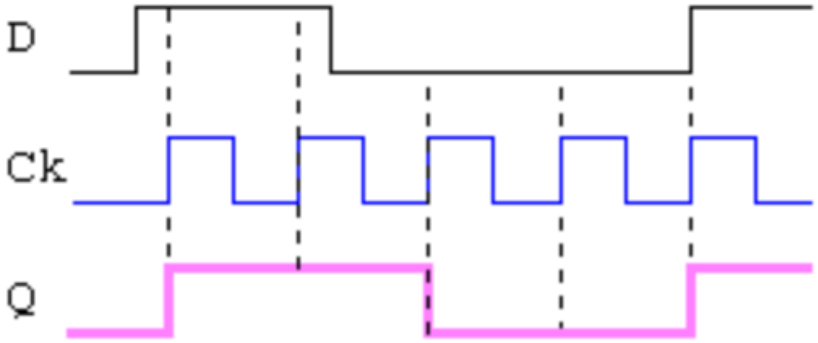
- Xung tín hiệu đầu ra:



Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1



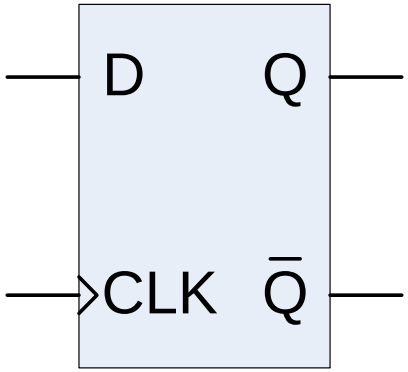
$$Q_{n+1} = D$$



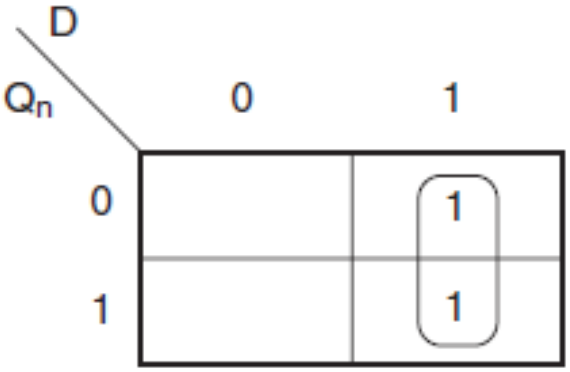
D	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	0	1	Xoá
1	↑	1	0	Đặt

FF D

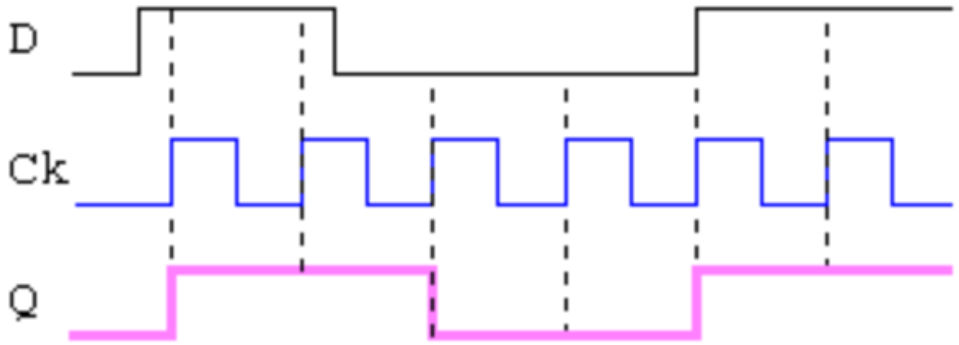
- Xung tín hiệu đầu ra:



Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1



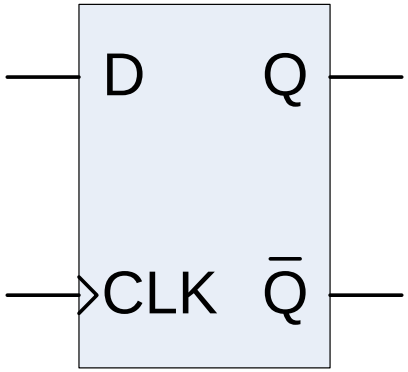
$$Q_{n+1} = D$$



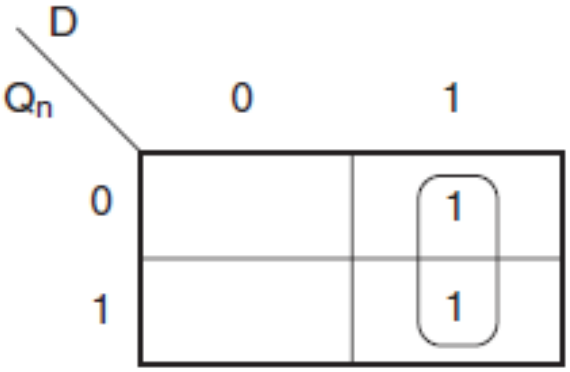
D	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	0	1	Xoá
1	↑	1	0	Đặt

FF D

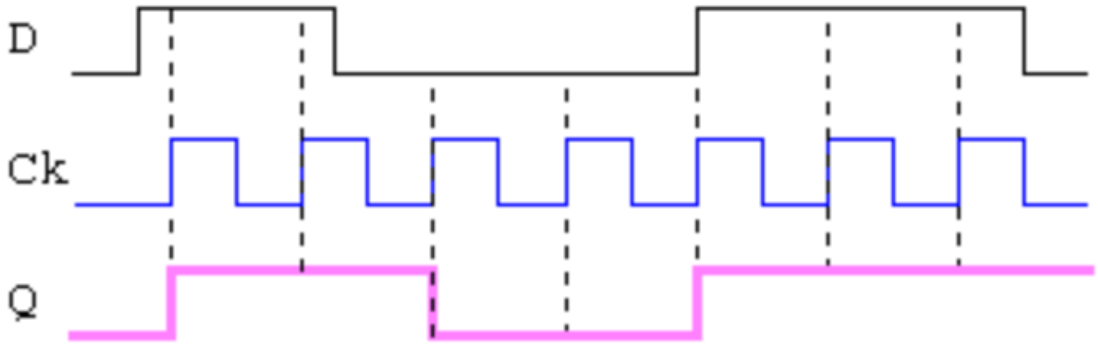
- Xung tín hiệu đầu ra:



Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1



$$Q_{n+1} = D$$



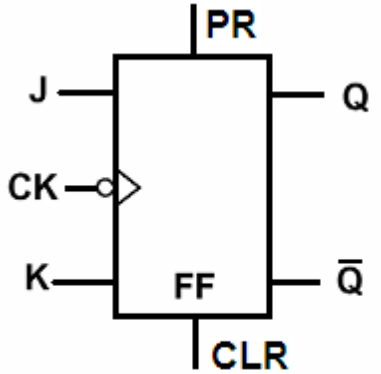
D	Ck	Q	$\bar{Q}$	Trạng thái
0	↑	0	1	Xoá
1	↑	1	0	Đặt

FF D

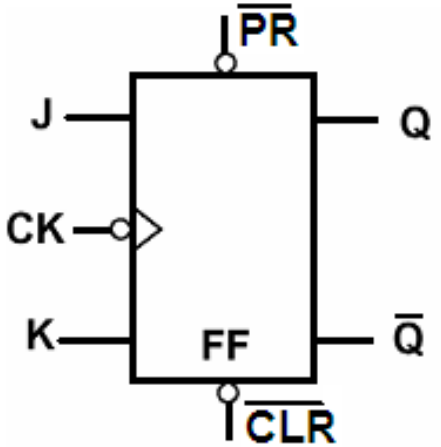
- FF D thường là nơi để chuyển dữ liệu từ tín hiệu vào D đến tín hiệu ra Q, cung cấp cho mạch sau như mạch cộng, ghi dịch.
- FF D phải chờ một khoảng thời gian khi xung kích thích kích thì mới đưa ra tín hiệu ra Q.
- Do đó, FF D còn được xem như mạch trì hoãn hay còn gọi là FF trễ.

FF với đầu vào Preset và Clear

- Mỗi FF đều có các tín hiệu sau:
  - Tín hiệu vào, ví dụ J, K
  - Tín hiệu đồng bộ clock
  - Tín hiệu ra Q
- Ngoài ra, nhiều FF còn có thêm các tín hiệu trực tiếp có tác dụng điều khiển cưỡng bức trạng thái ra của FF:
  - Clear (CLR), có tác dụng điều khiển để Q = 0
  - Preset (PR), làm cho Q = 1



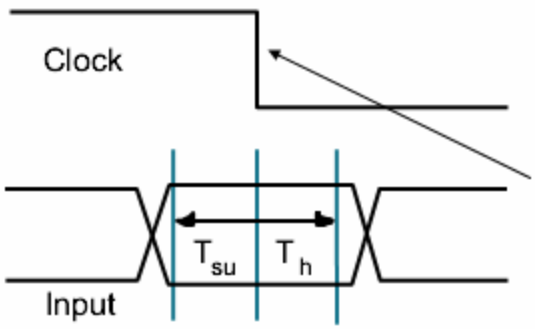
Tín hiệu điều khiển trực tiếp tích cực cao



Tín hiệu điều khiển trực tiếp tích cực thấp

Quan hệ thời gian ở Flip Flop

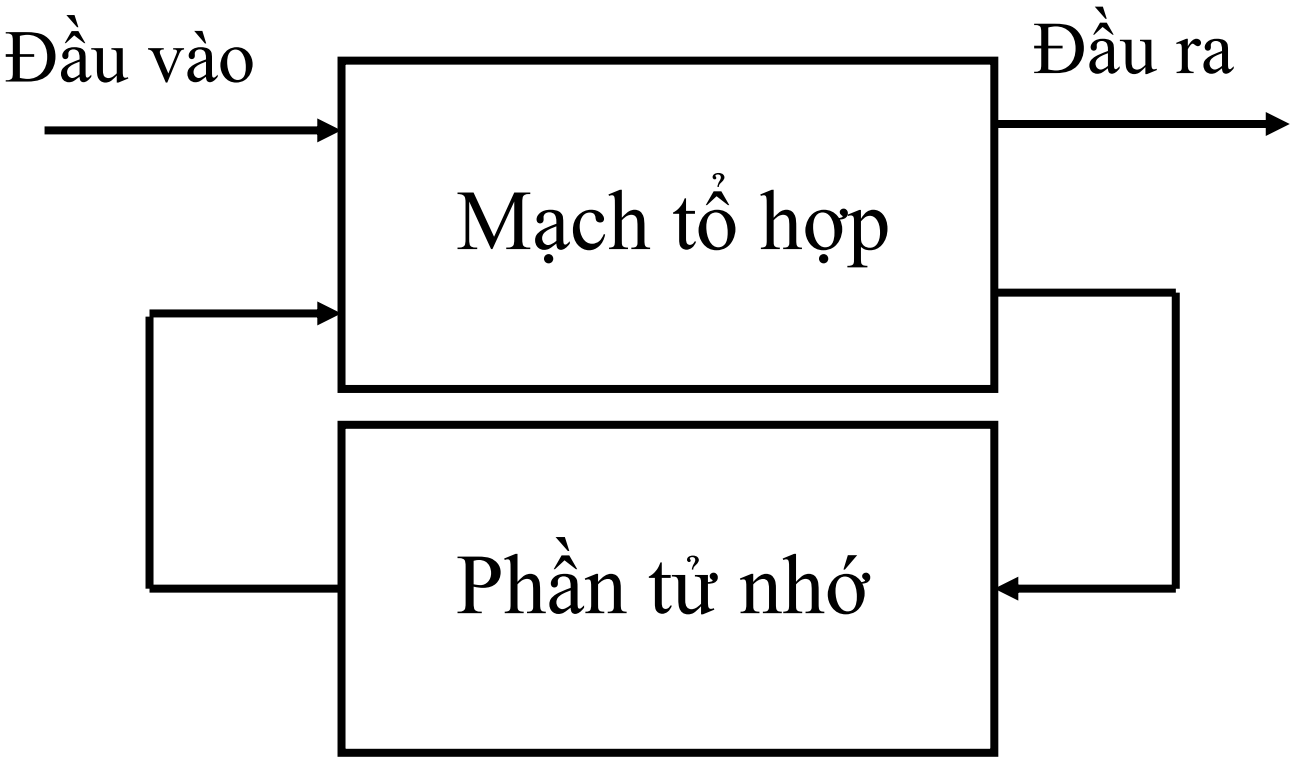
- "Cửa sổ" thời gian của FF được xác định bởi:
  - t_{su}: thời gian chuẩn bị (Setup) – tín hiệu vào cần phải xác lập ổn định ở một khoảng thời gian ≥ t_{su}, trước khi có '***sự kiện***' clock
  - t_h: thời gian duy trì (Hold) – tín hiệu vào cần phải duy trì ổn định thêm một khoảng thời gian ≥ t_h, sau khi kết thúc '***sự kiện***' clock
- Đây là một trong những yếu tố hạn chế tần số của mạch logic dãy



Có một vùng "cửa sổ" thời gian bao quanh sườn lấy mẫu của tín hiệu clock, ở đó tín hiệu vào cần phải được duy trì ổn định và không thay đổi để cho việc truyền tín hiệu qua Flip Flop được đảm bảo tin cậy.

8.4 Mô hình của mạch tuần tự

- Mô hình của mạch tuần tự được sử dụng để mô tả hệ thống qua tín hiệu vào, ra và trạng thái mà không quan tâm đến cấu trúc bên trong.



8.4 Mô hình của mạch tuần tự

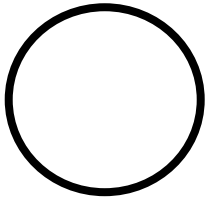
- Hệ tuần tự đồng bộ: khi làm việc cần có 1 tín hiệu đồng bộ để giữ nhịp cho toàn bộ hệ hoạt động.
- Hệ tuần tự không đồng bộ: không cần tín hiệu này để giữ nhịp chung cho toàn bộ hệ hoạt động.
- Hệ tuần tự đồng bộ nhanh hơn hệ tuần tự không đồng bộ tuy nhiên lại có thiết kế phức tạp hơn.

Mô hình trạng thái

- Tập hợp hữu hạn các trạng thái
- Một trạng thái khởi tạo, nằm trong tập hữu hạn các trạng thái
- Tập hợp hữu hạn các đầu vào
- Tập hợp hữu hạn các đầu ra
- Một hàm chuyển trạng thái (của một trạng thái và một tín hiệu đầu vào với trạng thái tương ứng)
- Một hàm đầu ra (của một trạng thái và một tín hiệu đầu vào với đầu ra tương ứng)

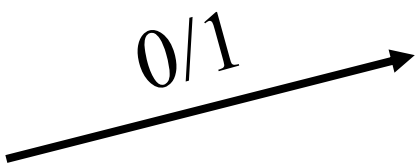
Sơ đồ trạng thái

- Các trạng thái được biểu diễn bằng hình tròn



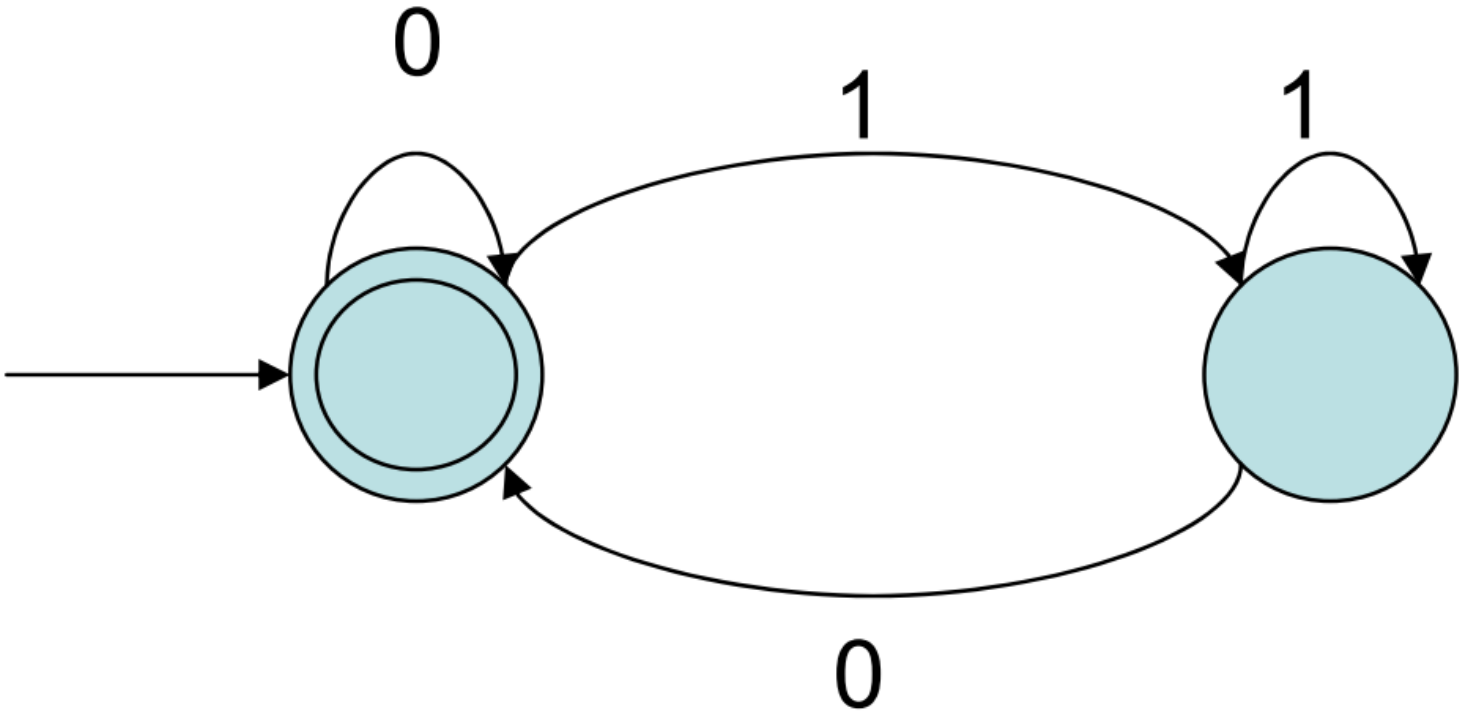
- Mũi tên chỉ hàm chuyển trạng thái

- Mũi tên được đánh dấu bằng các tín hiệu đầu vào



Ví dụ 8.3

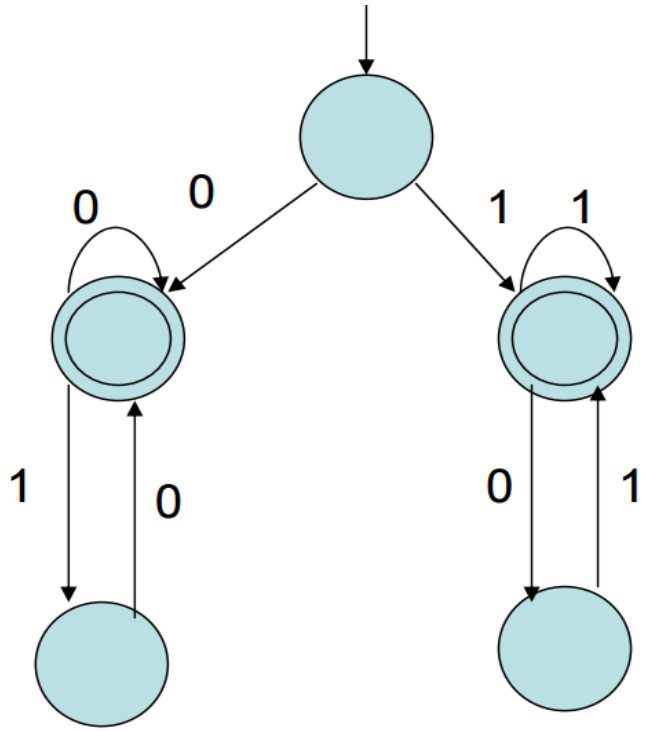
- Xét đồ hình trạng thái sau:



- Chấp nhận chuỗi rỗng hoặc bất kỳ chuỗi nào kết thúc bởi bit ‘0’

Ví dụ 8.4

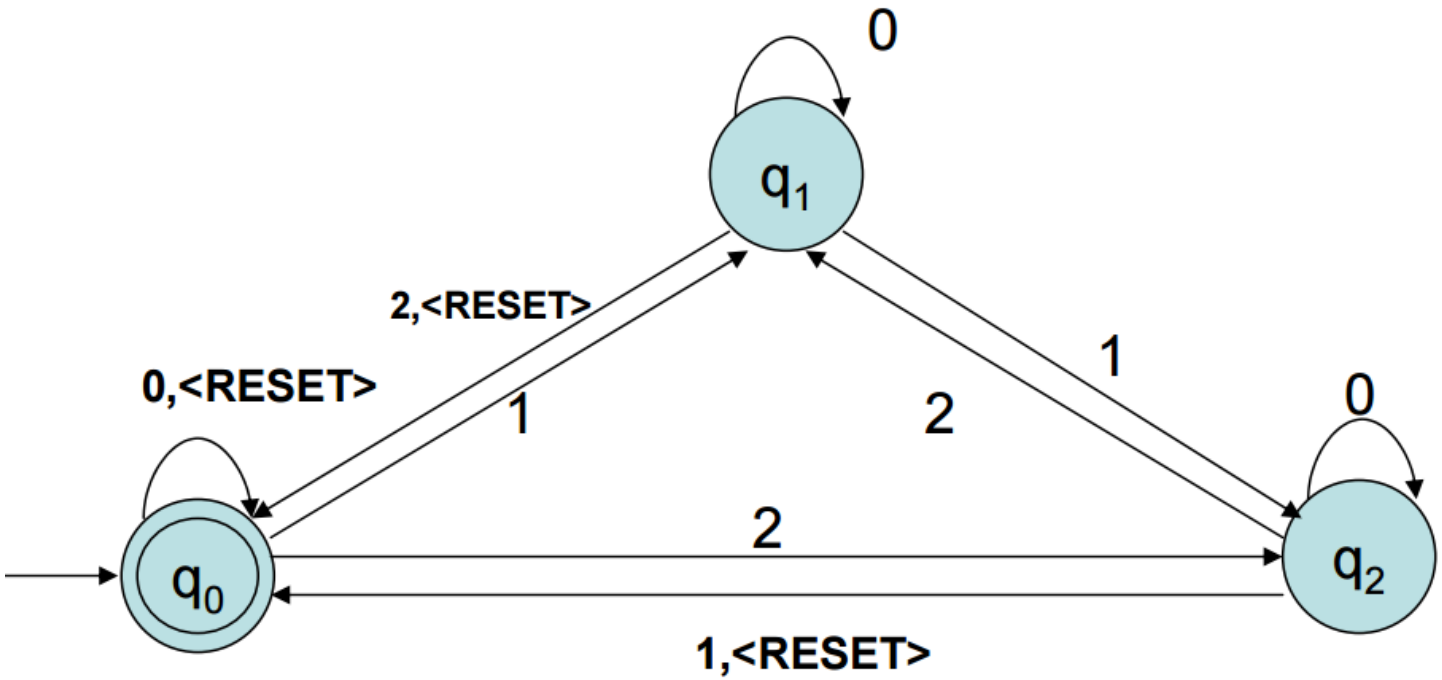
- Xét đồ hình trạng thái sau:



- Chấp nhận chuỗi bắt đầu và kết thúc bởi cùng 1 bit

Ví dụ 8.5

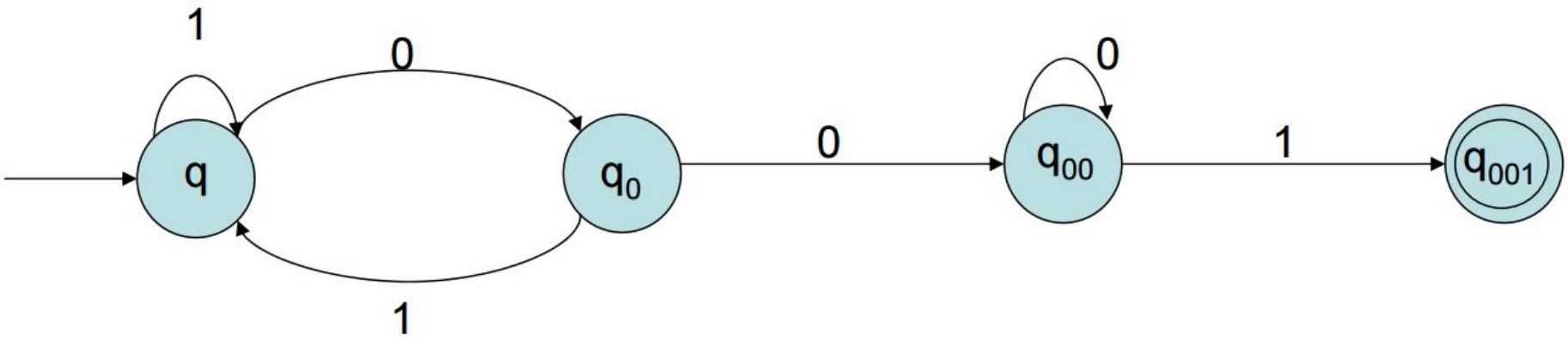
- Xét đồ hình trạng thái sau:



- Chấp nhận nếu tổng chuỗi đầu vào chia hết cho 3
- Khởi tạo lại tổng bằng 0

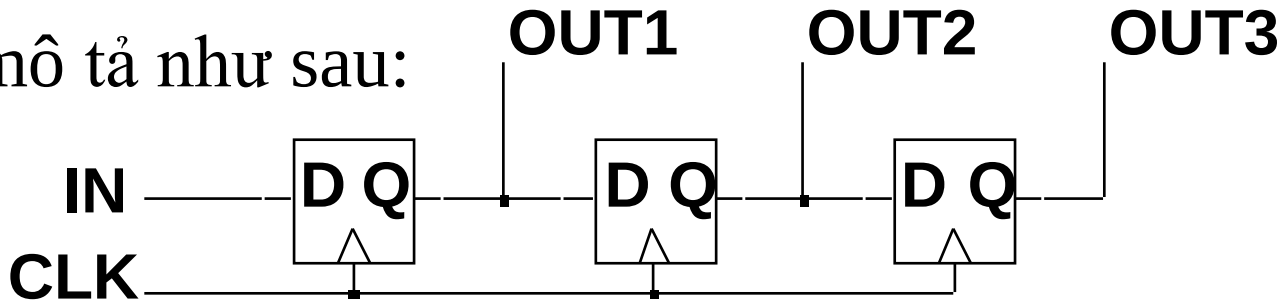
Ví dụ 8.6

- Thiết kế đồ hình trạng thái chấp nhận chuỗi 001.
- Có 4 khả năng:
  - Không có chuỗi nào
  - Nhìn thấy chuỗi 0
  - Nhìn thấy chuỗi 00
  - Nhìn thấy chuỗi 001

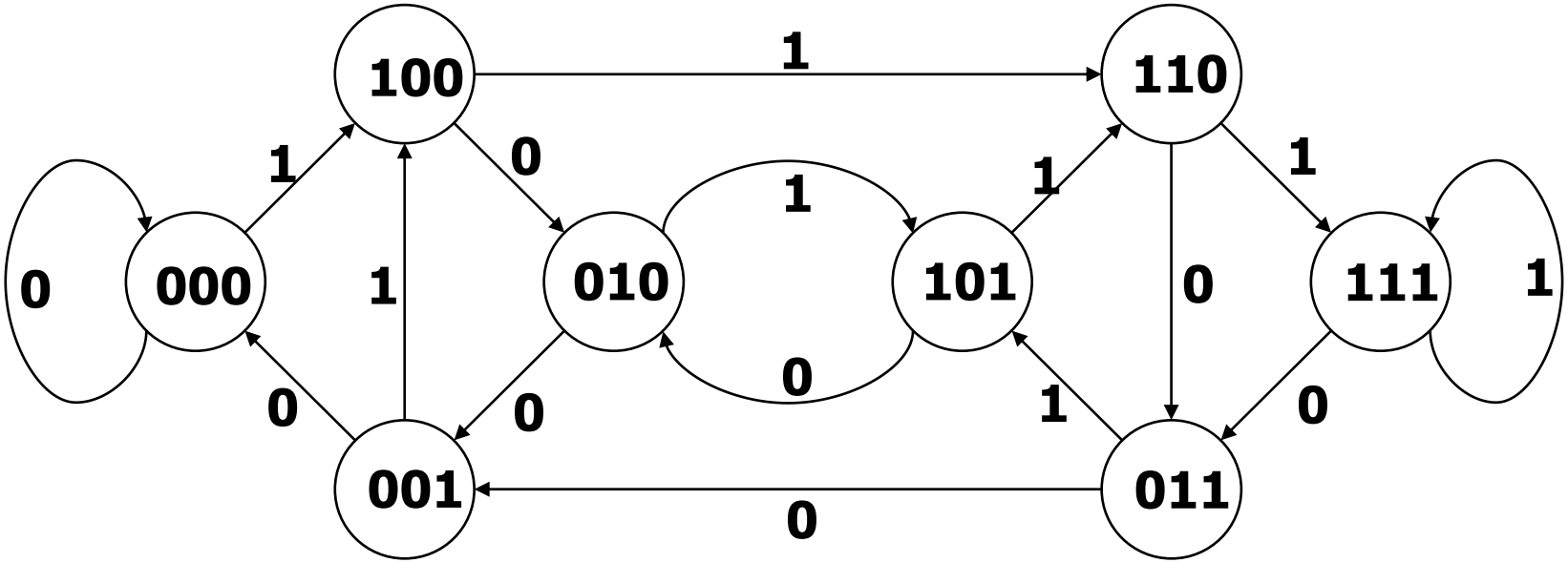


Ví dụ 8.7

- Cho mạch mô tả như sau:

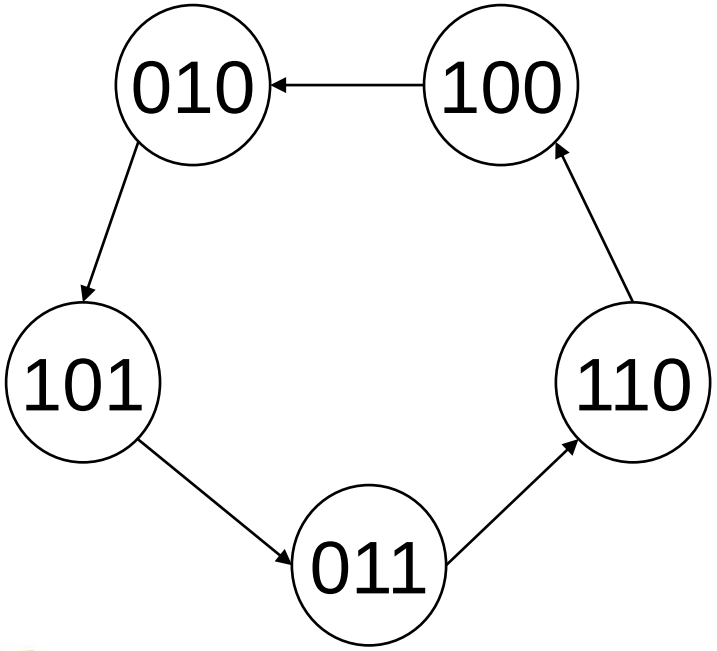


- Vẽ đồ hình trạng thái của mạch trên:



Ví dụ 8.8

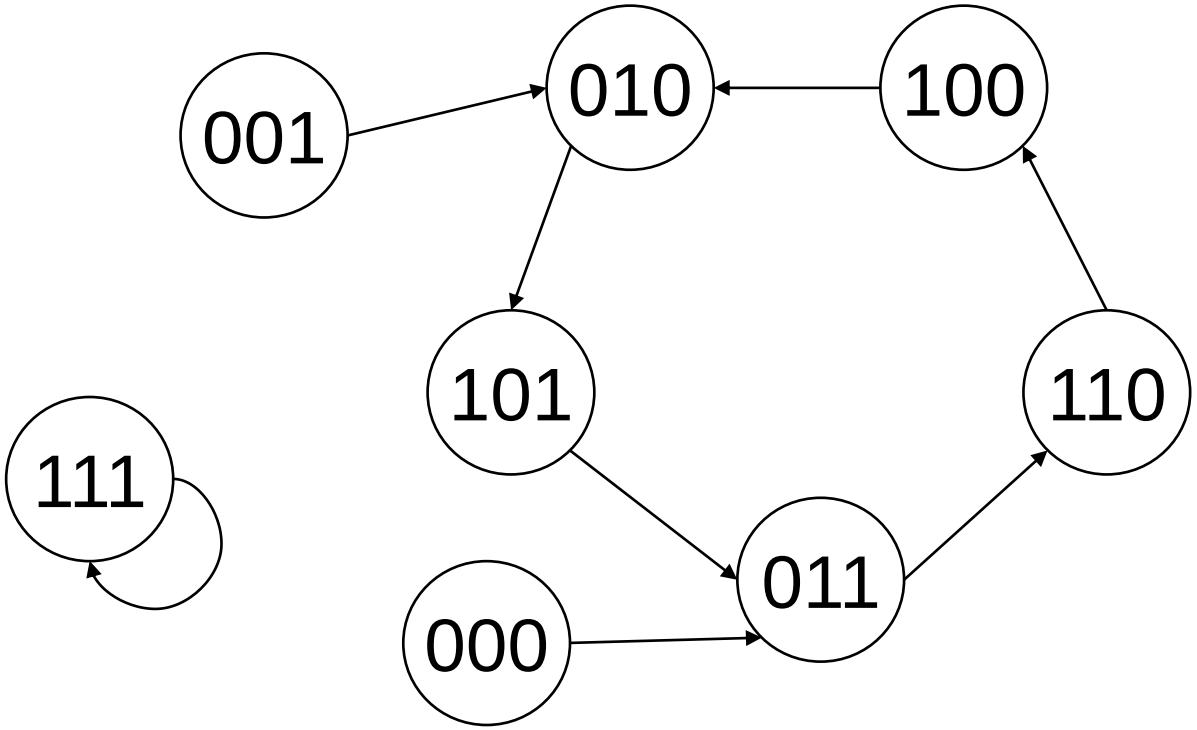
- Thiết kế một mạch đếm tuần tự đơn giản: 100, 010, 101, 011, 110
  - Bỏ qua trạng thái 000, 001 và 111
- Vẽ đồ hình trạng thái và bảng chuyển trạng thái:



Present State			Next State		
C	B	A	C+	B+	A+
0	0	0	x	x	x
0	0	1	x	x	x
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	x	x	x

Ví dụ 8.8

- Ở trạng thái khởi tạo, hệ tuần tự có thể rơi vào các trạng thái không xác định.
- Khi thiết kế phải đảm bảo không rơi vào trường hợp đó.

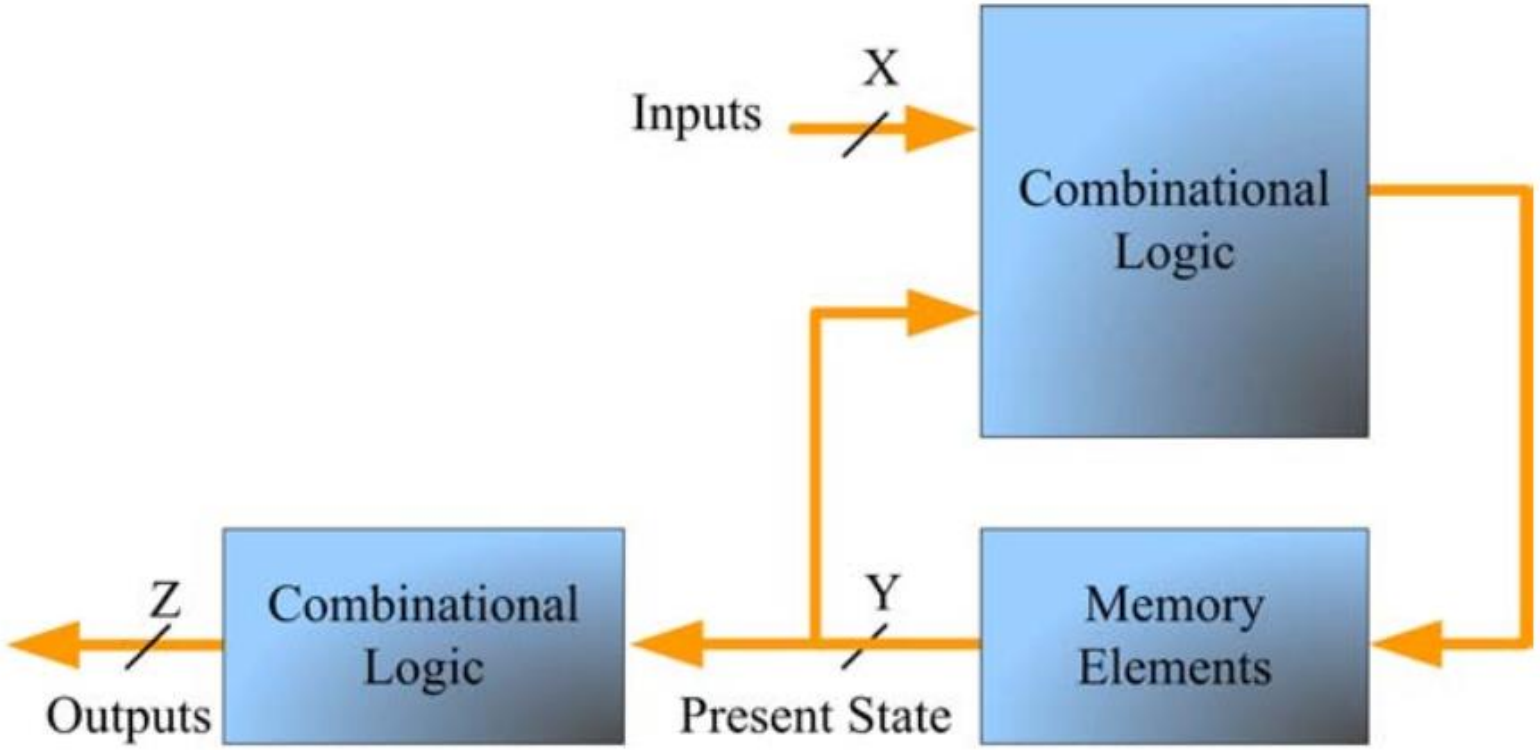


Mô hình trạng thái

- Tập hợp hữu hạn các trạng thái
- Một trạng thái khởi tạo, nằm trong tập hữu hạn các trạng thái
- Tập hợp hữu hạn các đầu vào
- Tập hợp hữu hạn các đầu ra
- Một hàm chuyển trạng thái (của một trạng thái và một tín hiệu đầu vào với trạng thái tương ứng)
- Một hàm đầu ra (của một trạng thái và một tín hiệu đầu vào với đầu ra tương ứng)
  - Mô hình Moore: Hàm của trạng thái
  - Mô hình Mealy: Hàm của trạng thái và tín hiệu đầu vào.

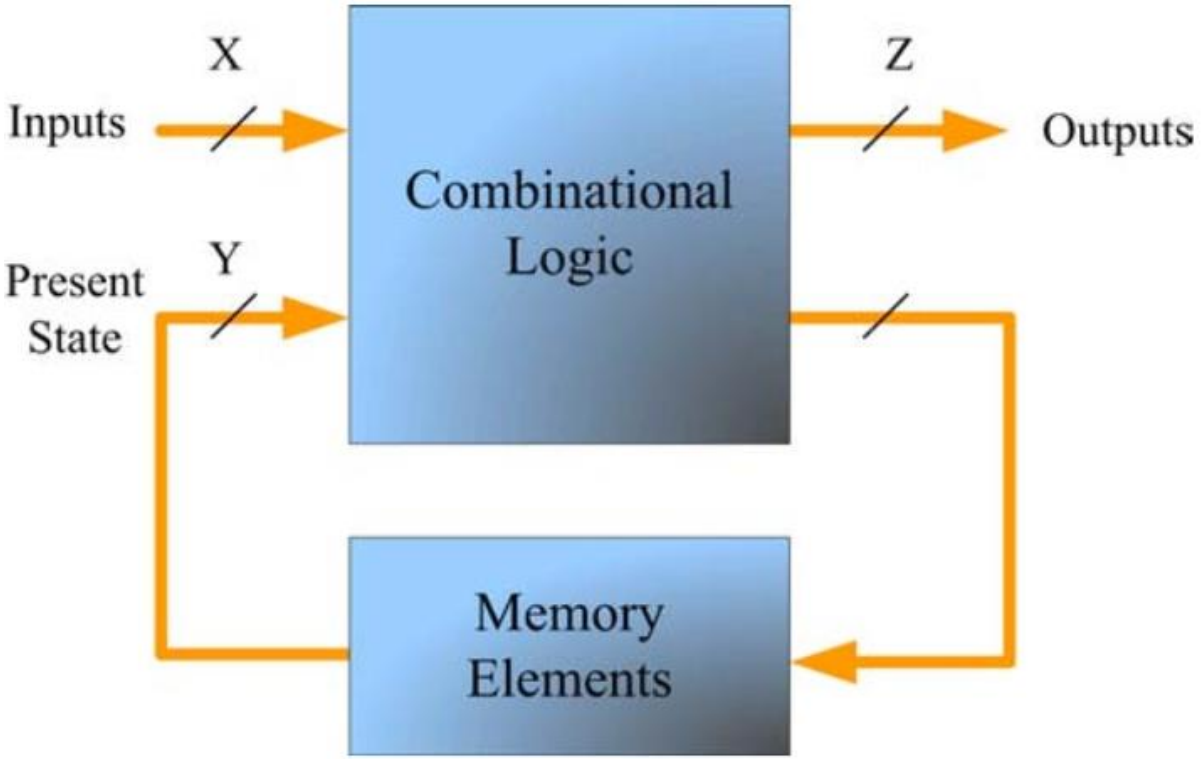
Mô hình Moore – Sơ đồ trạng thái

- Tín hiệu đầu ra chỉ phụ thuộc vào trạng thái hiện tại, không phụ thuộc trực tiếp vào tín hiệu đầu vào.



Mô hình Mealy – Sơ đồ trạng thái

- Tín hiệu đầu ra phụ thuộc vào trạng thái hiện tại và tín hiệu đầu vào

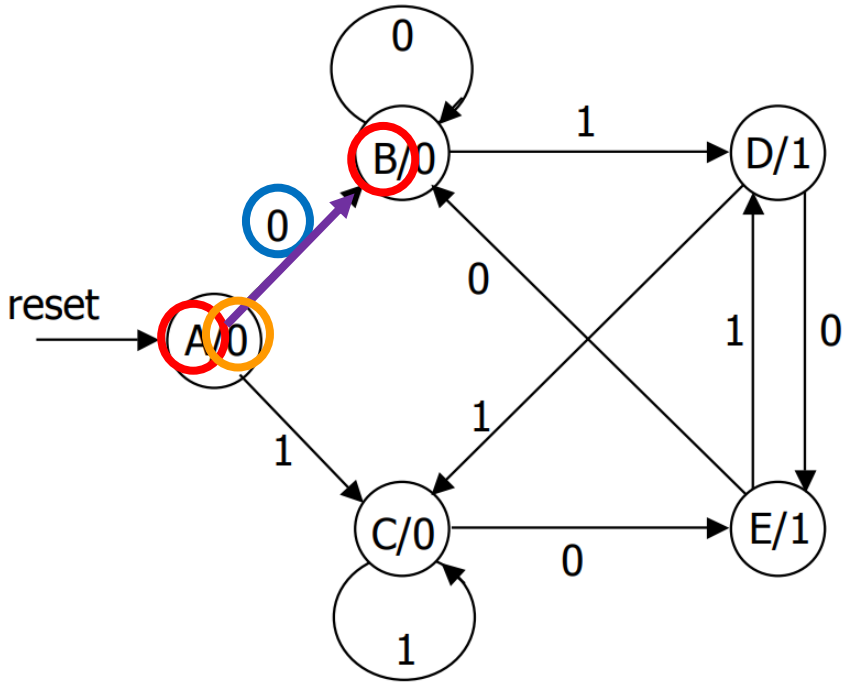


Mô hình trạng thái – Biểu diễn bằng sơ đồ trạng thái

- Các trạng thái được biểu diễn bằng hình tròn
- Mũi tên chỉ hàm chuyển trạng thái
- Mũi tên được đánh dấu bằng các tín hiệu đầu vào
- Đầu ra:
  - Mô hình Mealy: tín hiệu đầu ra phụ thuộc vào trạng thái hiện tại và đầu vào
    - Mũi tên được ký hiệu bởi **đầu vào/đầu ra** khi chuyển trạng thái.
  - Mô hình Moore: tín hiệu đầu ra chỉ phụ thuộc trạng thái trước đó:
    - Mũi tên được ký hiệu bởi **đầu vào** khi chuyển trạng thái.
    - Vòng tròn trạng thái được ký hiệu bởi **trạng thái k/đầu ra**.

Xác định đầu ra với mô hình Moore

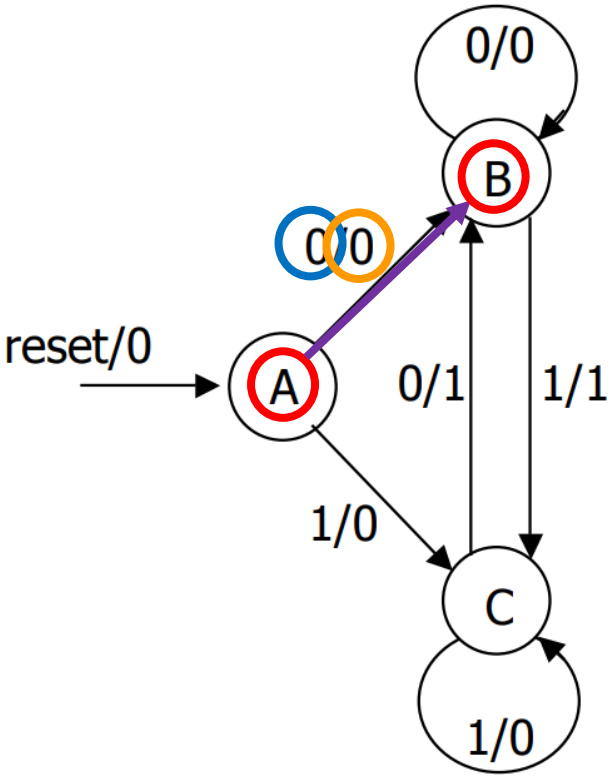
- Đầu ra là 1 hàm của trạng thái



reset	input	current state	next state	output
1	–	–	A	
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	D	0
0	0	C	E	0
0	1	C	C	0
0	0	D	E	1
0	1	D	C	1
0	0	E	B	1
0	1	E	D	1

Xác định đầu ra với mô hình Mealy

- Đầu ra là 1 hàm của trạng thái và đầu vào



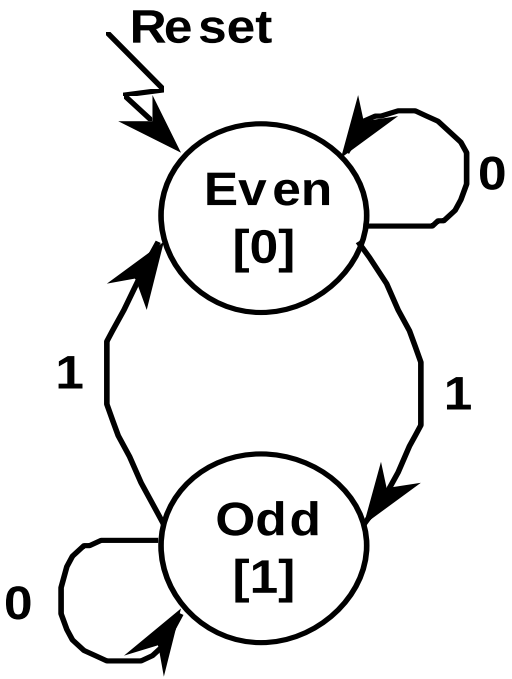
reset	input	current state	next state	output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	C	1
0	0	C	B	1
0	1	C	C	0

Các bước thiết kế hệ tuần tự

1. Mô tả hoạt động của mạch tuần tự cần thiết kế (đầu vào, đầu ra, các trạng thái, biểu đồ trạng thái, biểu đồ thời gian, hoặc các thông tin thích hợp khác)
2. Lập bảng chuyển trạng thái (state table)
3. Mã hoá trạng thái, đầu vào (nếu có)
4. Xác định số FF cần dùng và đánh ký hiệu cho mỗi FF
5. Lựa chọn kiểu FF cần dùng
6. Từ bảng chuyển trạng thái, xác định đầu vào cho mỗi FF (hàm trạng thái của tín hiệu đầu vào với trạng thái tương ứng) và biểu thức của mỗi biến ra
7. Tối thiểu hóa cho đầu vào cho mỗi FF và đầu ra
8. Lập sơ đồ mạch logic từ các phần tử cơ bản

Ví dụ 8.9

- Thiết kế bộ kiểm tra bit lẻ: đầu ra bằng 1 khi đầu vào có số bit 1 lẻ.
- Vẽ đồ hình trạng thái và bảng chuyển trạng thái:



Trạng thái hiện tại	Đầu vào	Trạng thái tiếp theo	Đầu ra
Chẵn	0	Chẵn	0
Chẵn	1	Lẻ	1
Lẻ	0	Lẻ	1
Lẻ	1	Chẵn	0

Ví dụ 8.9

- Gán giá trị nhị phân cho mỗi trạng thái:

Trạng thái hiện tại	Đầu vào	Trạng thái tiếp theo	Đầu ra
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

- Xác định số lượng FF cần sử dụng:
  - Cần 1 FF, vì chỉ cần 1 bit để thể hiện 2 trạng thái.

Kích thích cho các flip flop

- Khi thiết kế mạch tuần tự, cần phải xác định điều kiện kích thích cho các FF tuỳ theo đáp ứng cần có của chúng.
- Với hai giá trị logic ‘0’ và ‘1’ cho mỗi biến, mỗi FF có thể có một trong bốn đáp ứng là: ‘S0’, ‘S1’, ‘T0’, và ‘T1’
- Bảng dưới đây mô tả các điều kiện kích thích cho các loại FF khác nhau

Đáp ứng		Kích thích					
Ký hiệu	$Q \rightarrow Q^+$	S	R	J	K	T	D
S0	0 → 0	0	x	0	x	0	0
T1	0 → 1	1	0	1	x	1	1
T0	1 → 0	0	1	x	1	1	0
S1	1 → 1	x	0	x	0	0	1

Ví dụ 8.9

- Lựa chọn FF: Giả sử chọn FF D
- Giả thiết:
  - Q = Trạng thái hiện tại
  - Q^+ = Trạng thái tiếp theo
  - X = Đầu vào

Đáp ứng		Kích thích					
Ký hiệu	Q → Q^+	S	R	J	K	T	D
S0	0 → 0	0	x	0	x	0	0
T1	0 → 1	1	0	1	x	1	1
T0	1 → 0	0	1	x	1	1	0
S1	1 → 1	x	0	x	0	0	1

Q	X	Q^+ Đầu ra	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

D = Q+

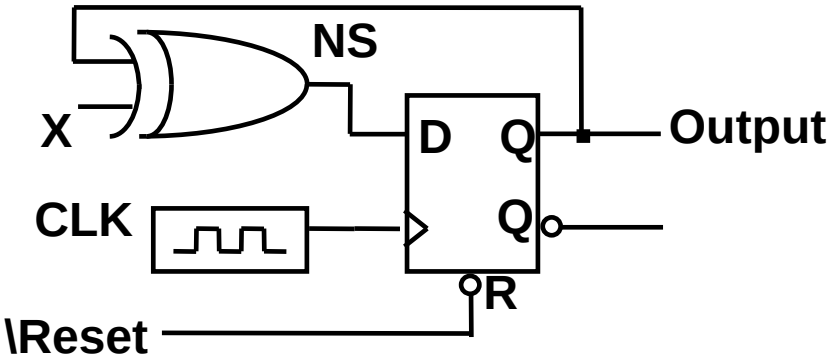
Ví dụ 8.9

- Tối thiểu hóa bằng bảng Karnaugh:

D \ X / Q	0	1
0	0	1
1	1	0

$$D = \overline{Q}X + Q\overline{X} = Q \oplus X$$

- Thực hiện mạch:

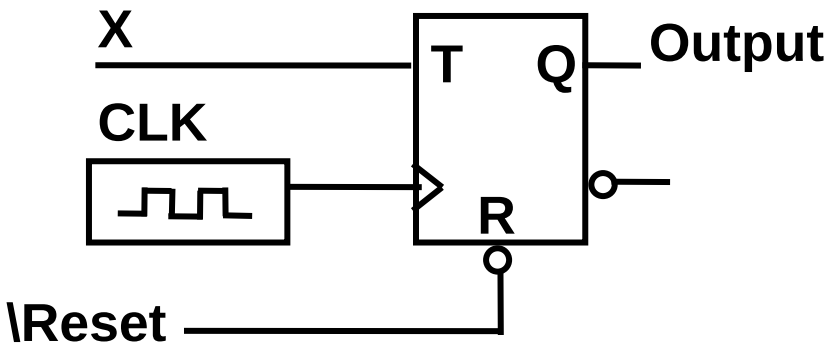


Ví dụ 8.9

- Lựa chọn FF: Giả sử chọn FF T
- Giả thiết:
  - Q = Trạng thái hiện tại
  - Q^+ = Trạng thái tiếp theo
  - X = Đầu vào
- Thực hiện mạch

Q	X	Q^+	Đầu ra	T
0	0	0	0	0
0	1	1	1	1
1	0	1	1	0
1	1	0	0	1

T = X



Các bước thiết kế hệ tuần tự

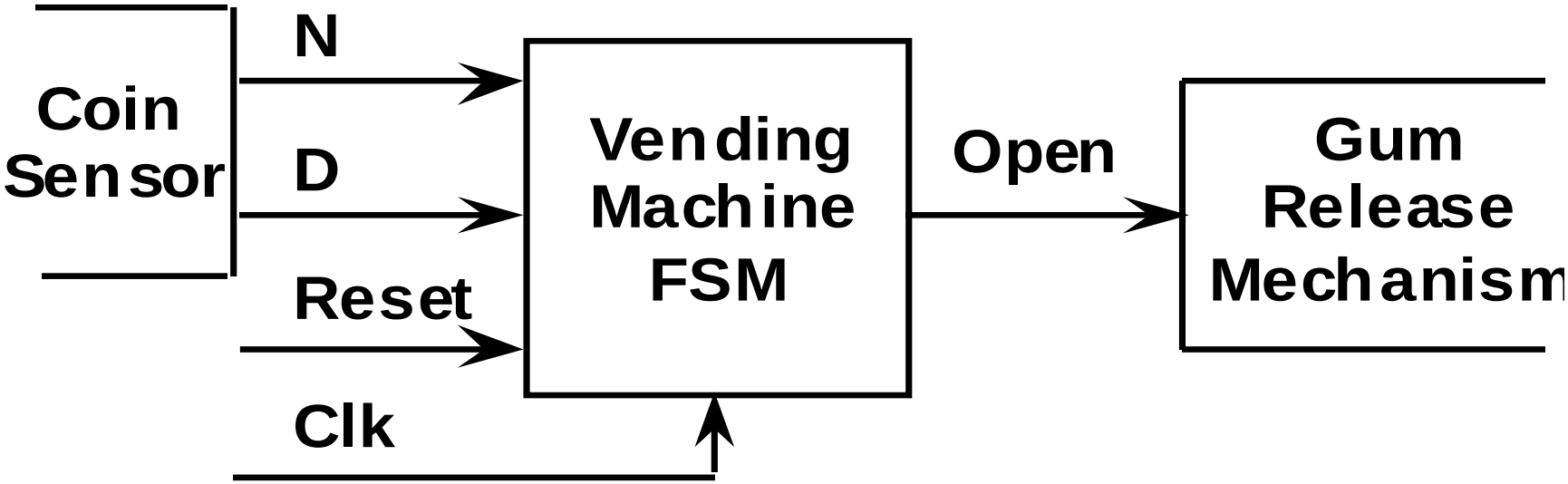
1. Mô tả hoạt động của mạch tuần tự cần thiết kế (đầu vào, đầu ra, các trạng thái, biểu đồ trạng thái, biểu đồ thời gian, hoặc các thông tin thích hợp khác)
2. Lập bảng chuyển trạng thái (state table)
3. Mã hoá trạng thái, đầu vào (nếu có)
4. Xác định số FF cần dùng và đánh ký hiệu cho mỗi FF
5. Lựa chọn kiểu FF cần dùng
6. Từ bảng chuyển trạng thái, xác định đầu vào cho mỗi FF (hàm trạng thái của tín hiệu đầu vào với trạng thái tương ứng) và biểu thức của mỗi biến ra
7. Tối thiểu hóa cho đầu vào cho mỗi FF và đầu ra
8. Lập sơ đồ mạch logic từ các phần tử cơ bản

Ví dụ 8.10

- Thiết kế máy bán hàng tự động:
  - Nhả một gói kẹo sau khi nhận được 15 xu
  - Khe để nhét tiền xu nhận loại 5 xu, 10 xu
  - Không trả lại tiền thừa

Ví dụ 8.10 – Bước 1: Mô tả bài toán

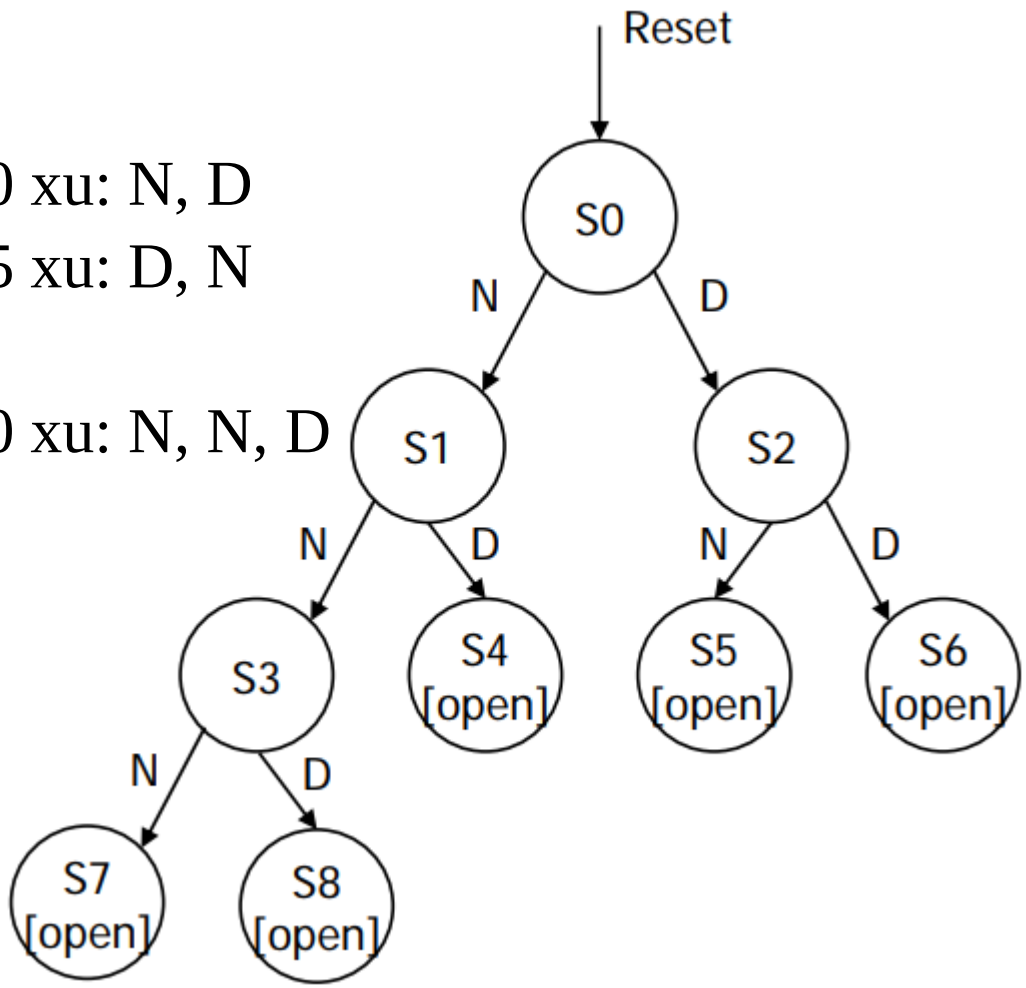
- Mô tả bài toán:



- Mô tả đầu vào và ra:
  - Đầu vào: N, D, reset
  - Đầu ra: open

Ví dụ 8.10 – Bước 2: Lập bảng chuyển trạng thái

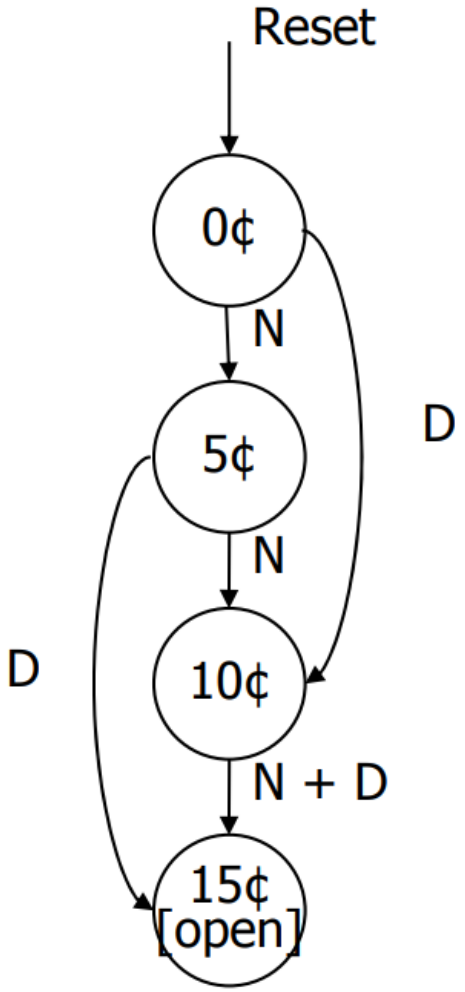
- Vẽ đồ hình trạng thái:
  - 3 đồng 5 xu: N, N, N
  - 1 đồng 5 xu, 1 đồng 10 xu: N, D
  - 1 đồng 10 xu, 1 đồng 5 xu: D, N
  - 2 đồng 10 xu: D, D
  - 2 đồng 5 xu, 1 đồng 10 xu: N, N, D



Ví dụ 8.10 – Bước 2: Lập bảng chuyển trạng thái

- Moore machine

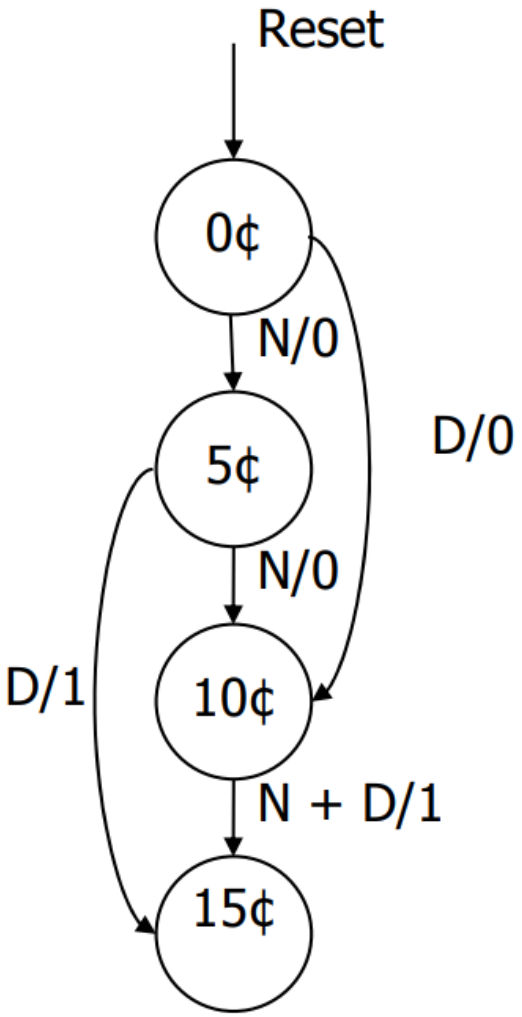
present state	inputs D	N	next state	present output
0¢	0	0	0¢	0
	0	1	5¢	0
	1	0	10¢	0
	1	1	–	–
5¢	0	0	5¢	0
	0	1	10¢	0
	1	0	15¢	0
	1	1	–	–
10¢	0	0	10¢	0
	0	1	15¢	0
	1	0	15¢	0
	1	1	–	–
15¢	–	–	15¢	1



Ví dụ 8.10 – Bước 2: Lập bảng chuyển trạng thái

- Mealy machine

present state	inputs D	N	next state	present output
0¢	0	0	0¢	0
	0	1	5¢	0
	1	0	10¢	0
	1	1	–	–
5¢	0	0	5¢	0
	0	1	10¢	0
	1	0	15¢	1
	1	1	–	–
10¢	0	0	10¢	0
	0	1	15¢	1
	1	0	15¢	1
	1	1	–	–
15¢	–	–	15¢	1



Ví dụ 8.10 – Bước 2, 3, 4, 5

- Mã hóa trạng thái, xác định số lượng FF: 02, sử dụng FF D

Moore

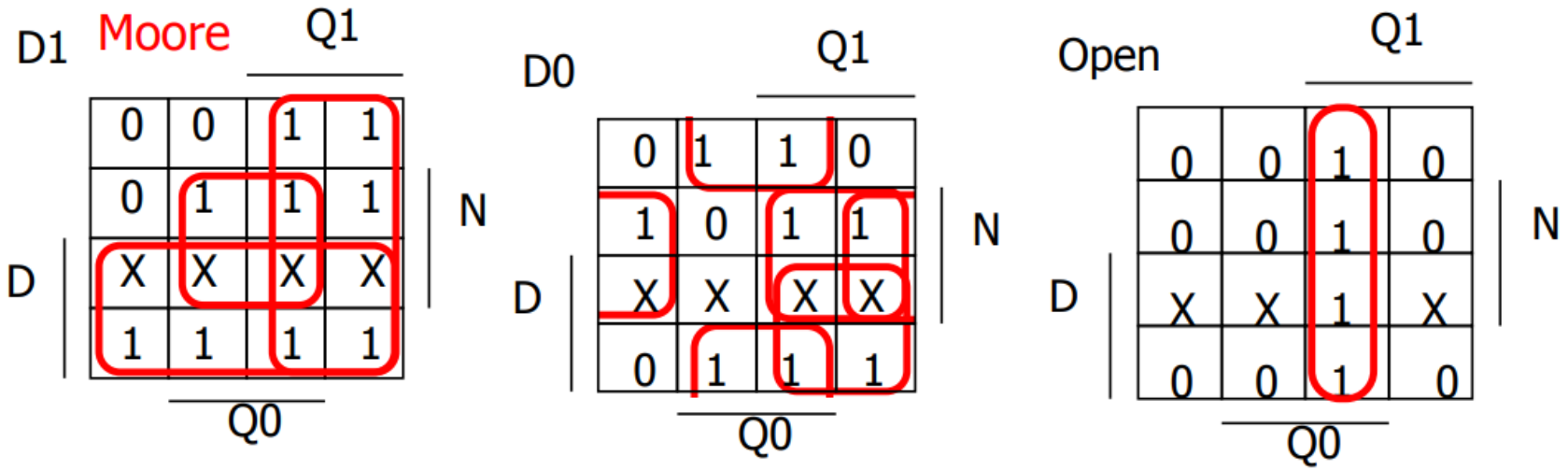
present state Q1	Q0	inputs D	N	next state D1	D0	present output
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	–	–	–
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	0
		1	1	–	–	–
1	0	0	0	1	0	0
		0	1	1	1	0
		1	0	1	1	0
		1	1	–	–	–
1	1	–	–	1	1	1

Mealy

present state Q1	Q0	inputs D	N	next state D1	D0	present output
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	–	–	–
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	1
		1	1	–	–	–
1	0	0	0	1	0	0
		0	1	1	1	1
		1	0	1	1	1
		1	1	–	–	–
1	1	–	–	1	1	1

Ví dụ 8.10 - Bước 6, 7

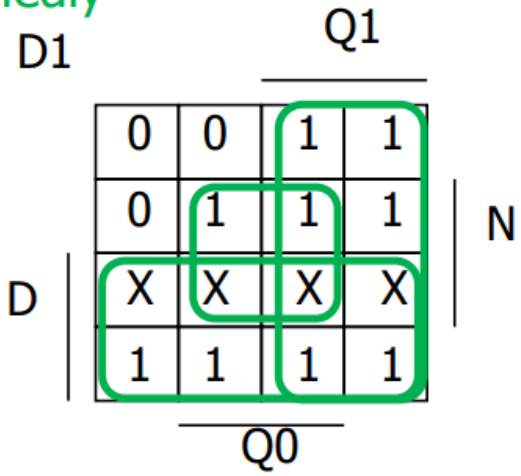
- Tối thiểu hóa

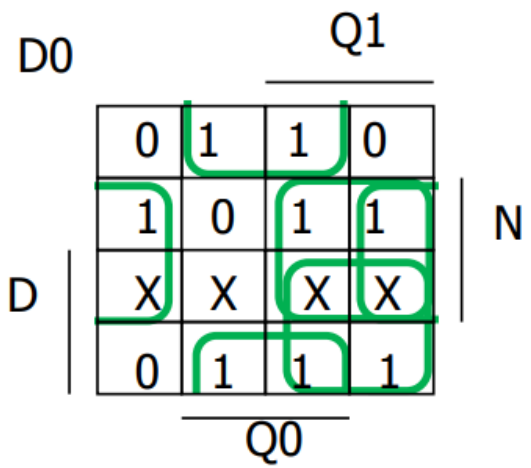


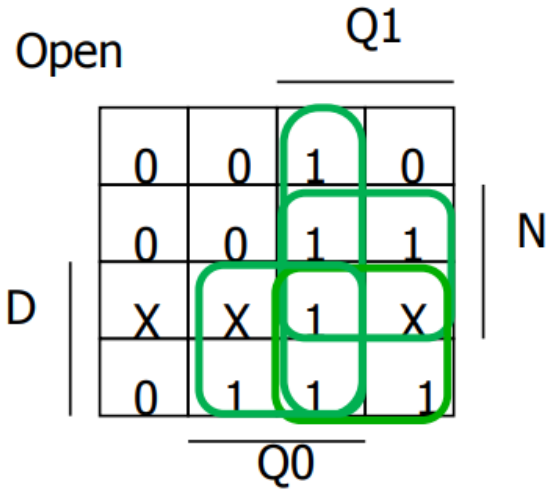
Ví dụ 8.10 - Bước 6, 7

- Tối thiểu hóa

Mealy



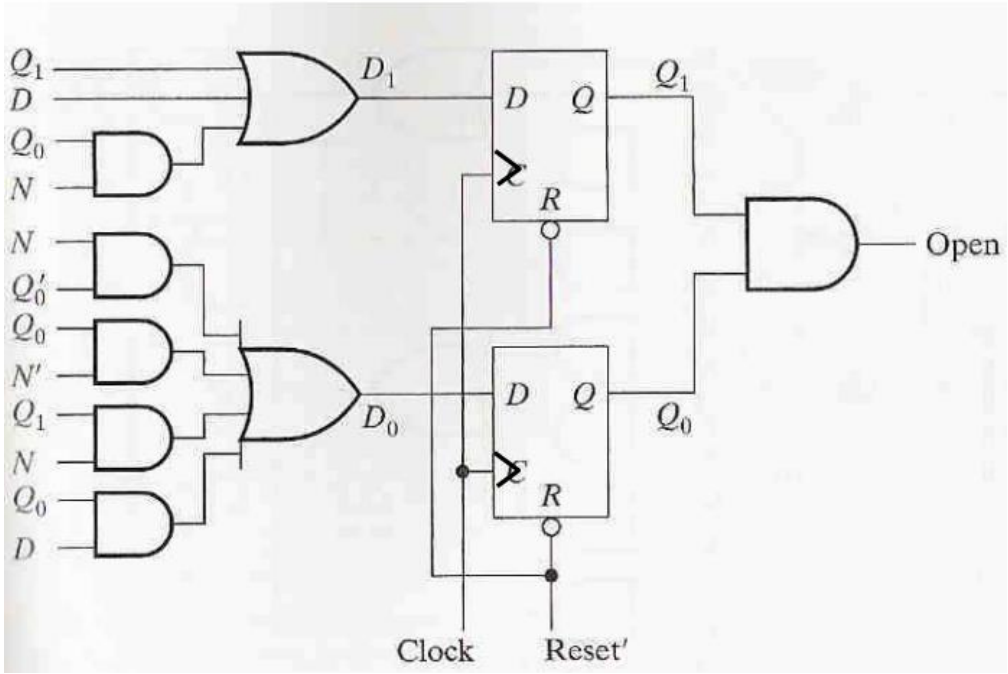




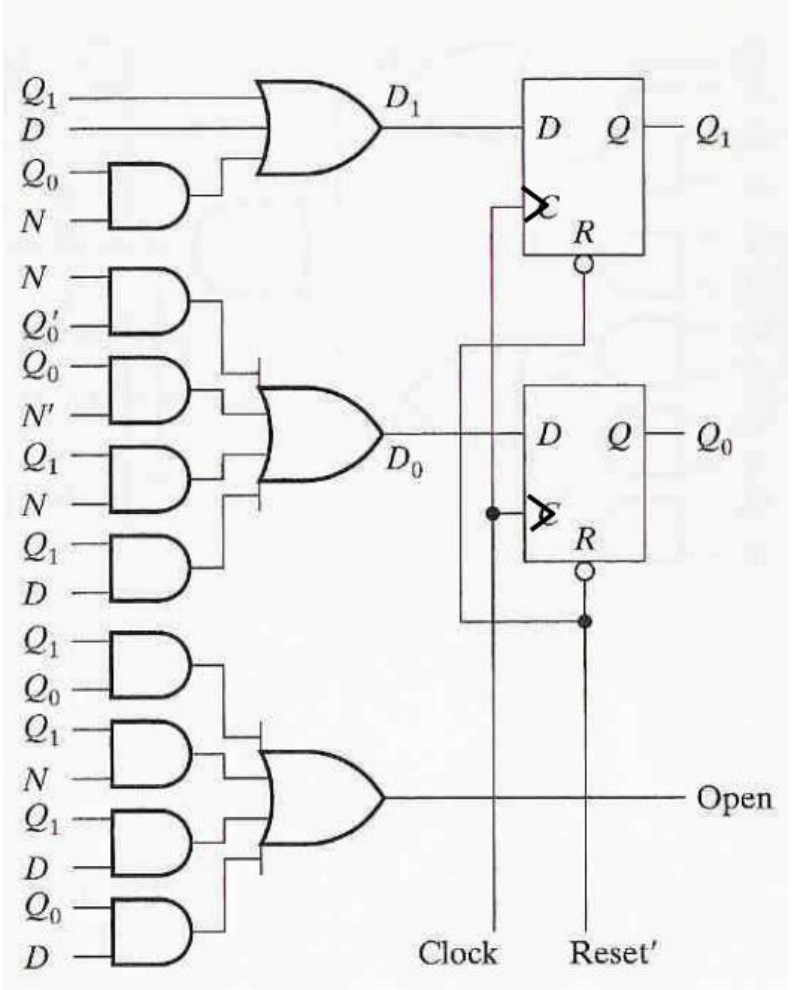
Ví dụ 8.10 – Bước 8

- Sơ đồ mạch

Moore



Mealy



So sánh mô hình Mealy và Moore

Mô hình Moore	Mô hình Mealy
Đầu ra chỉ phụ thuộc vào trạng thái hiện tại mà không phụ thuộc vào đầu vào	Đầu ra phụ thuộc vào trạng thái hiện tại và đầu vào
Có xu hướng cần ít trạng thái hơn, và do đó cần ít phần cứng hơn để giải quyết bài toán	Có xu hướng cần nhiều trạng thái hơn, và do đó cần nhiều phần cứng hơn để giải quyết bài toán
An toàn hơn khi sử dụng: - Đầu ra thay đổi theo xung đồng hồ (1 chu kỳ muộn hơn) - Trong mô hình Mealy, đầu vào có thể thay đổi đầu ra ngay khi hàm logic được thực hiện, là 1 vấn đề khi kết hợp 2 mô hình với nhau.	Đáp ứng nhanh hơn với đầu vào: - Đáp ứng trong cùng 1 chu kỳ, không cần đợi xung đồng bộ. - Trong mô hình Moore, có thể cần mã hóa nhiều trạng thái hơn nên sẽ cần nhiều cổng hơn, dẫn đến trễ lớn hơn.

Bài tập 8.1

- Thiết kế bộ nhận diện 3 bit ‘1’ liên tục theo mô hình Mealy/Moore.
  - Đầu ra bằng ‘1’ khi đầu vào là 3 bit ‘1’ liên tục
  - Các trường hợp khác đầu ra bằng ‘0’

8.5 Một số ứng dụng của Mạch tuần tự

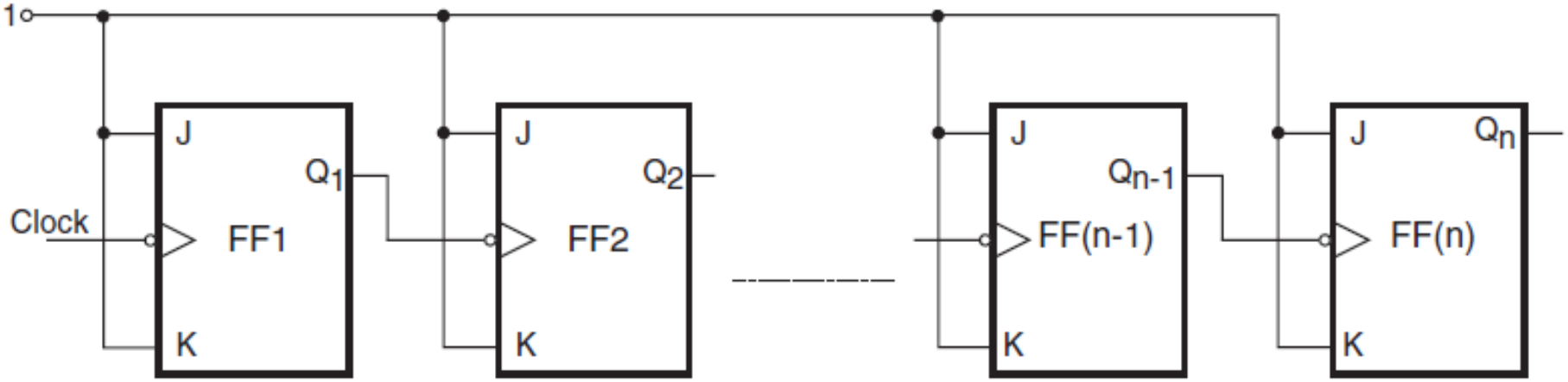
- Bộ đếm và chia tần
- Thanh ghi
- …

8.5.1 Bộ đếm và chia tần số

- Bộ đếm được dùng để đếm xung
- Bộ đếm được gọi là module n nếu nó có thể đếm được n xung: từ 0 đến n-1
- Có 2 loại bộ đếm:
  - Bộ đếm không đồng bộ: không đồng thời đưa tín hiệu đếm vào các đầu vào của các trigger
  - Bộ đếm đồng bộ: có xung đếm đồng thời là xung đồng hồ clock đưa vào tất cả các trigger của bộ đếm

Bộ đếm không đồng bộ

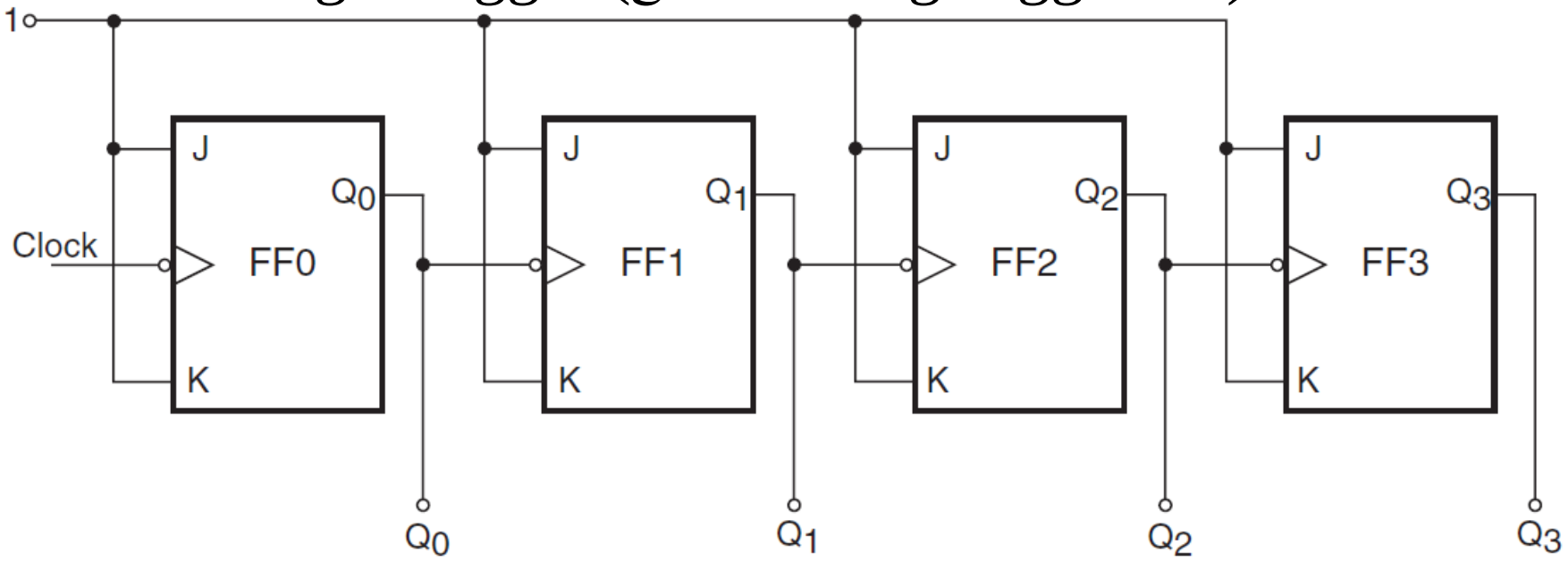
- Mạch thực hiện bộ đếm không đồng bộ:



- Trễ lan truyền: sau mỗi xung đồng hồ đầu vào, bộ đếm phải đợi một khoảng thời gian tương đương với tổng trễ của tất cả FF trước khi có thể thực hiện xung tiếp theo.

Bộ đếm không đồng bộ module 2^N

- Xét bộ đếm không đồng bộ module 16
- Có 16 trạng thái
- Mã hóa thành 4 bit tương ứng với Q_3, Q_2, Q_1, Q_0
- Cần dùng 4 trigger (giả sử dùng trigger JK)



Ví dụ 8.11

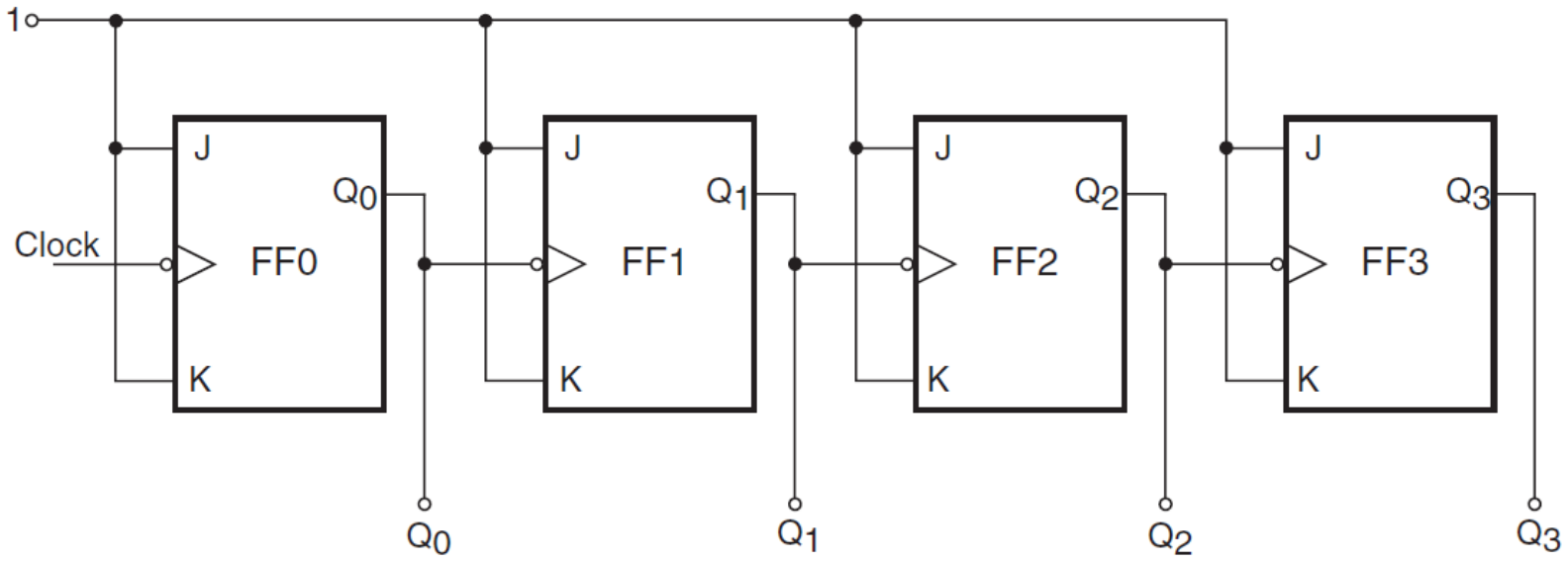
- Cần thiết kế một bộ đếm nhị phân không đồng bộ, có khả năng đếm sản phẩm trên băng chuyền.
- Mỗi lần 1 sản phẩm đi qua 1 điểm cố định, 1 xung sẽ được tạo ra đưa vào xung đồng hồ.
- Nếu số lượng sản phẩm cần đếm tối đa là 6000 thì cần bao nhiêu FF?

Solution

- The counter should be able to count a maximum of 6000 items.
- An N-flip-flop would be able to count up to a maximum of $2^N - 1$ counts.
- On the 2^Nth clock pulse, it will get reset to all 0s.
- Now, $2^N - 1$ should be greater than or equal to 6000.
- That is, $2^N - 1 \geq 6000$, which gives $N \geq \log 6001/\log 2 \geq 3.778/0.3010 \geq 12.55$.
- The smallest integer that satisfies this condition is 13.
- Therefore, the minimum number of flip-flops required = 13

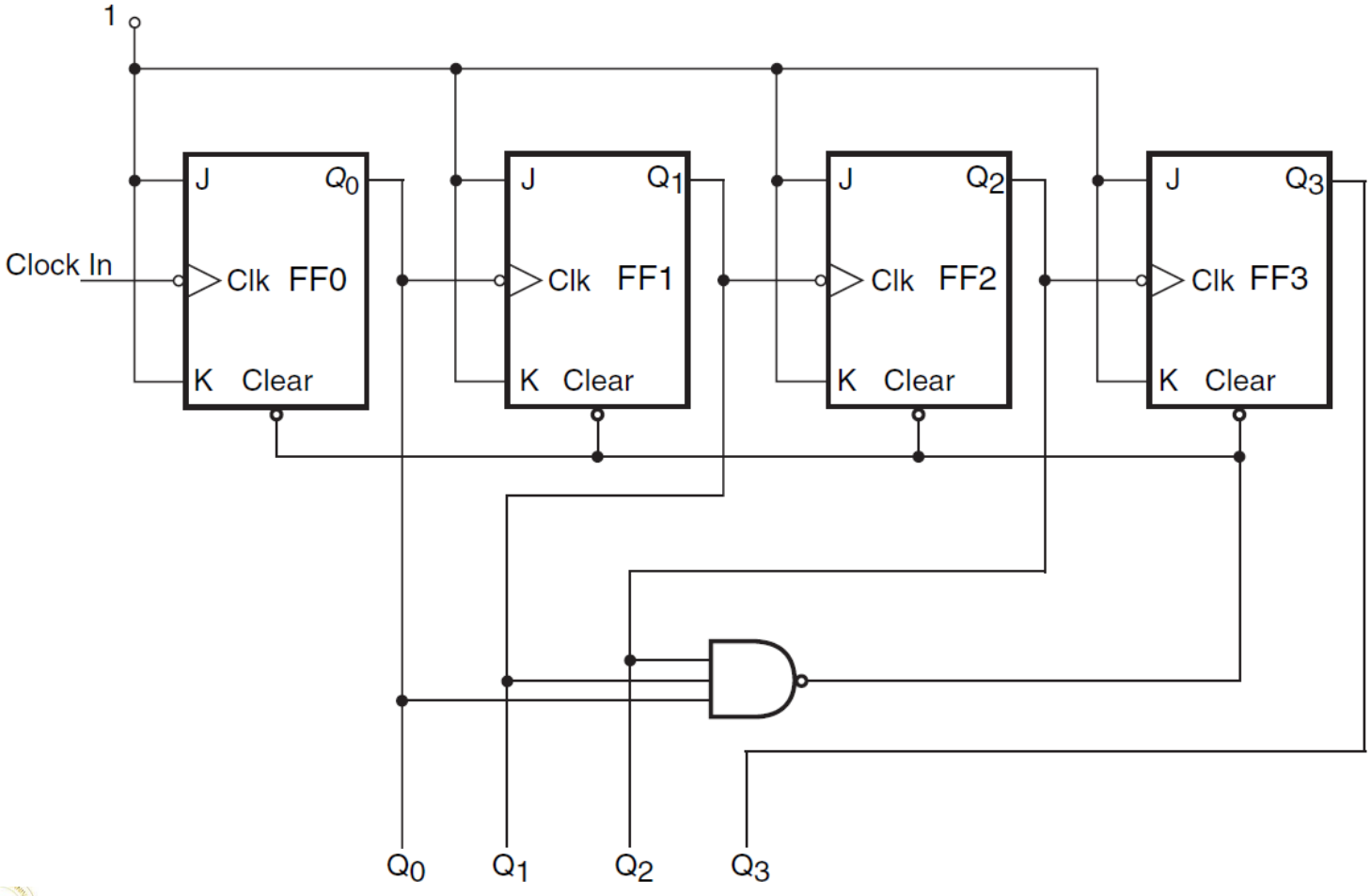
Bộ đếm không đồng bộ module < 2^N

- Xét bộ đếm module 7 dựa trên bộ đếm như hình dưới.
- Giả sử dùng Trigger JK có đầu vào CLR (CLEAR) tích cực ở mức thấp.
- Cứ mỗi khi đếm đến xung thứ 6 thì tất cả các FF bị xóa về 0.



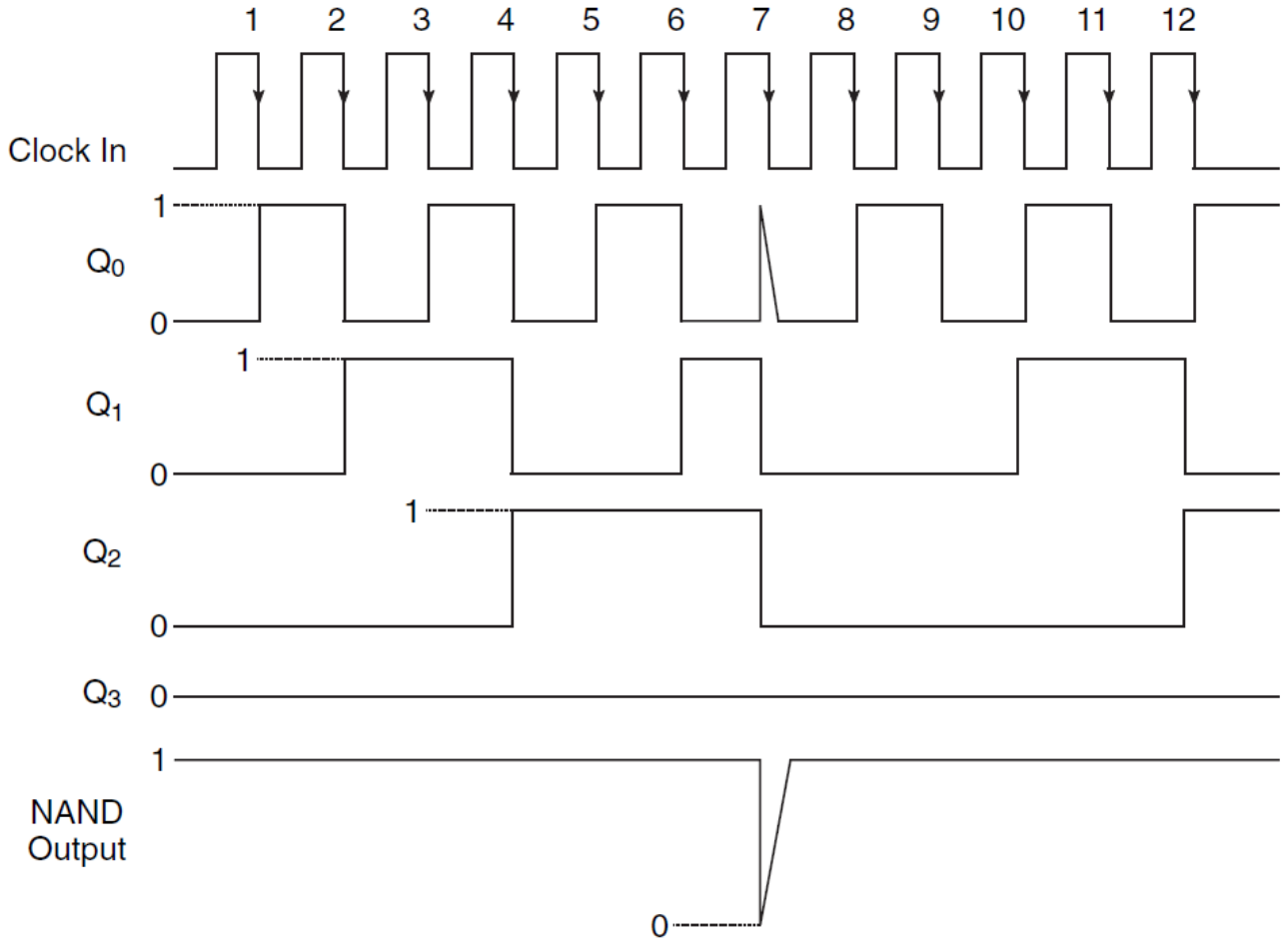
Bộ đếm không đồng bộ module < 2^N

- Sơ đồ mạch:



Bộ đếm không đồng bộ module < 2^N

- Sơ đồ mạch:

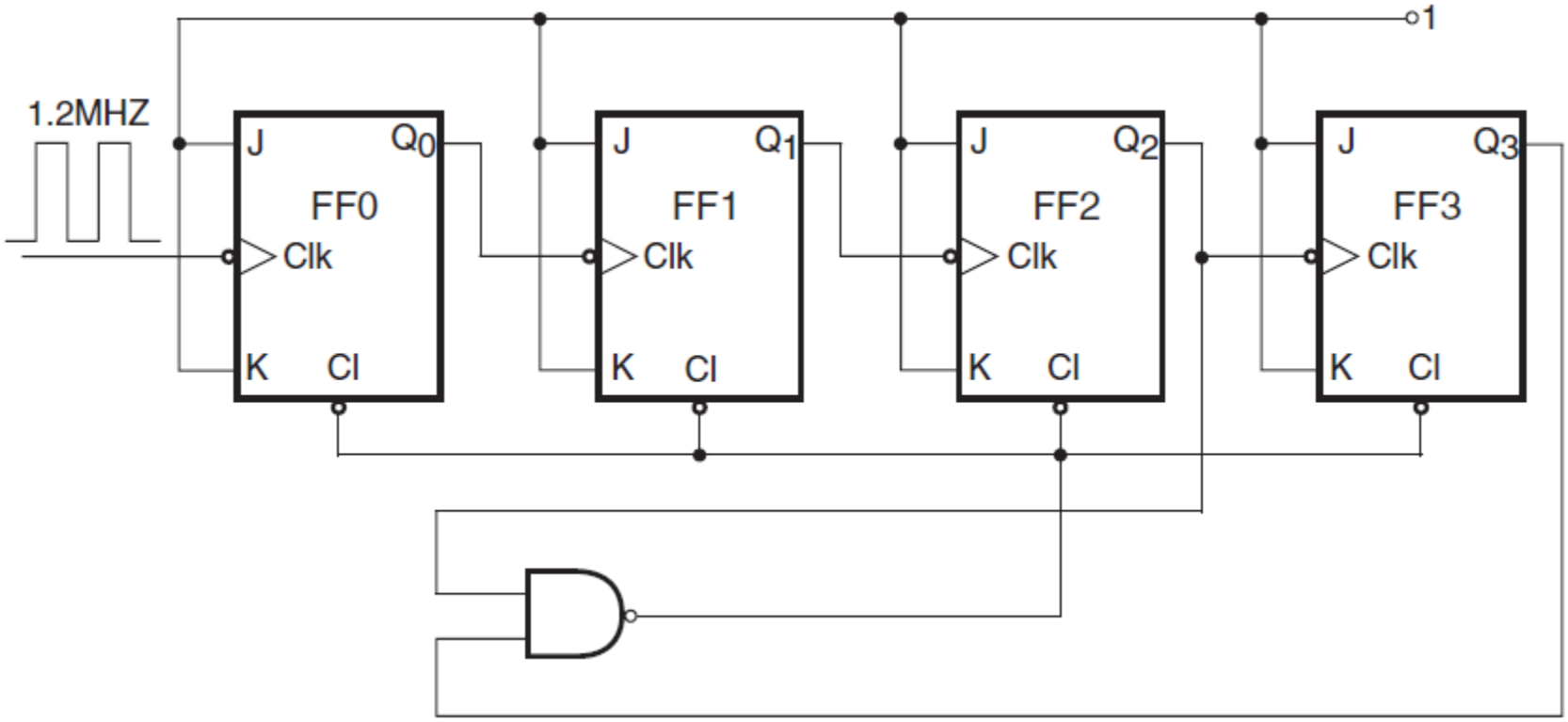


Bộ đếm không đồng bộ module < 2^N

- Các bước thiết kế bộ đếm không đồng bộ nhị phân bắt đầu từ 0000 và có cơ số X được tổng kết như sau:
  - Quyết định số lượng FF tối thiểu sao cho: $2^N \geq X$, kết nối các FF này như bộ đếm nhị phân không đồng bộ. Nếu $2^N = X$ thì kết thúc.
  - Xác định những FF ở mức logic CAO tại số đếm mà cơ số 10 tương đương =X. Chọn 1 cổng NAND với số lượng đầu vào bằng với số FF ở mức logic CAO. Ví dụ, với bộ đếm MOD-12, số đếm tương ứng là 1100 thì sẽ cần 2 FF ở mức logic CAO. Cổng NAND do đó sẽ là cổng 2 đầu vào.
  - Kết nối đầu ra Q của những FF đã được xác định vào đầu vào cổng NAND và đầu ra cổng NAND sẽ được đưa vào các đầu vào CLEAR không đồng bộ của tất cả các FF.

Ví dụ 8.12

- Tìm cơ số của bộ đếm sau và tần số của đầu ra FF Q_3.

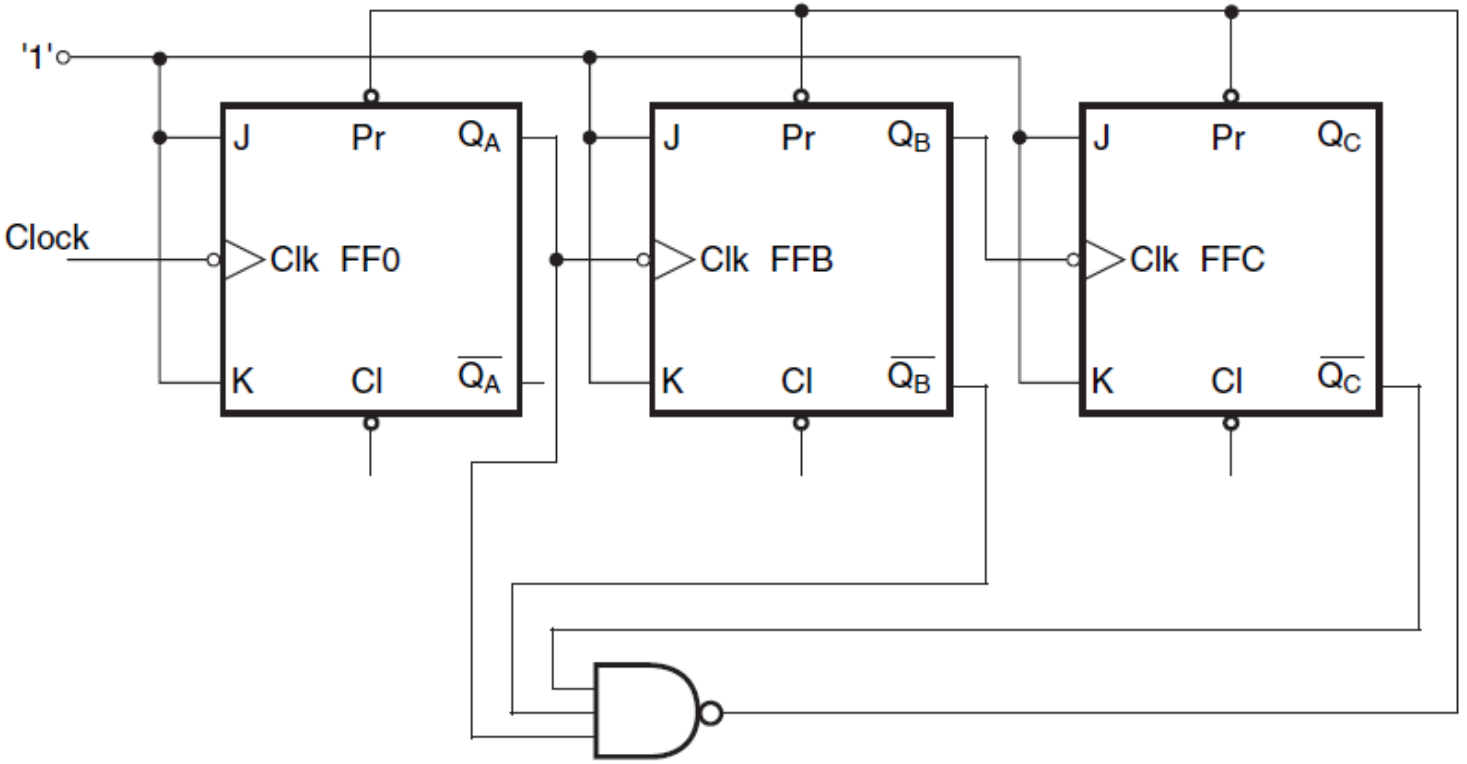


- Trả lời: Cơ số: 12

  Tần số Q_3: 100kHz

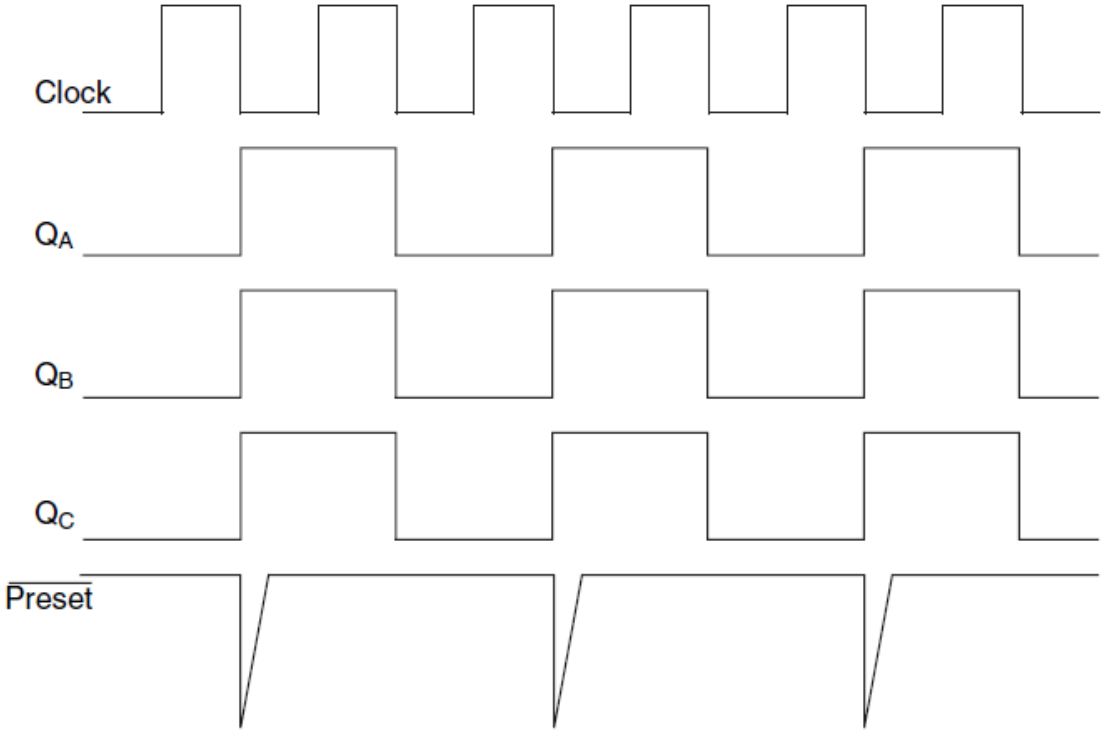
Ví dụ 8.13

- Thiết kế bộ đếm nhị phân không đồng bộ chỉ đếm hai trạng thái 000 và 111 và bỏ qua 6 trạng thái còn lại, sử dụng FF JK và vẽ dạng xung đầu ra.



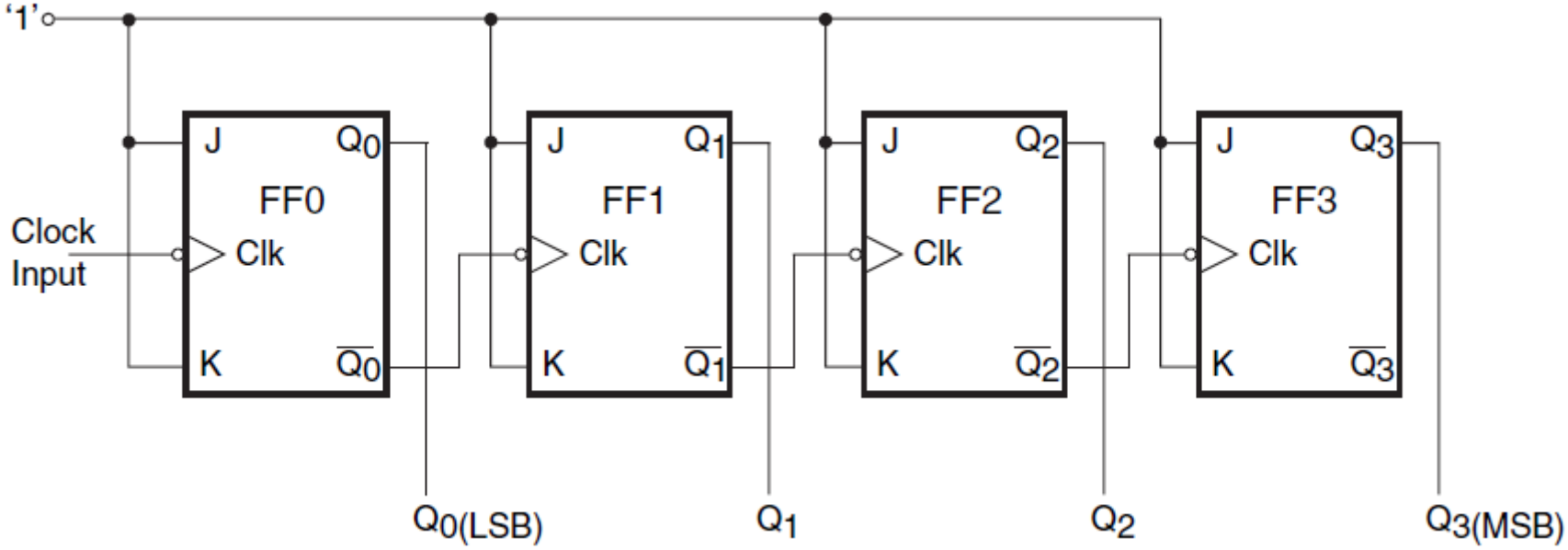
Ví dụ 8.13

- Thiết kế bộ đếm nhị phân không đồng bộ chỉ đếm hai trạng thái 000 và 111 và bỏ qua 6 trạng thái còn lại, sử dụng FF JK và vẽ dạng xung đầu ra.



Ví dụ 8.14

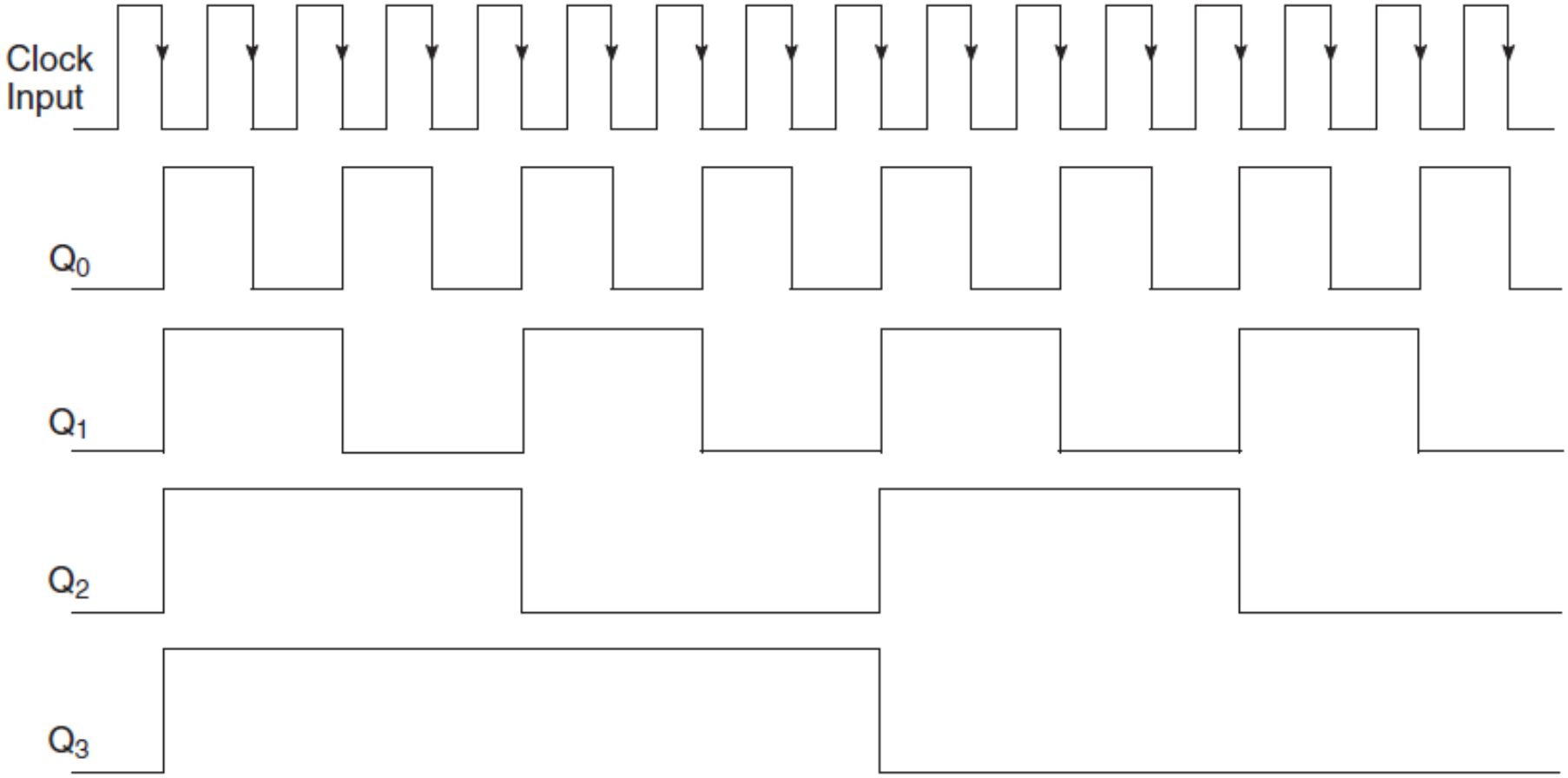
- Xét bộ đếm không đồng bộ sau, hãy viết thứ tự đếm nếu khởi tạo từ trạng thái 0000. Vẽ dạng sóng ra.



- Thứ tự đếm: 0000, 1111, 1110, 1101,1100, 1011, 1010, 1001, 1000, 0111, 0110, 0101, 0100, 0011, 0010, 0001 và 0000.

Ví dụ 8.14

- Dạng sóng ra:

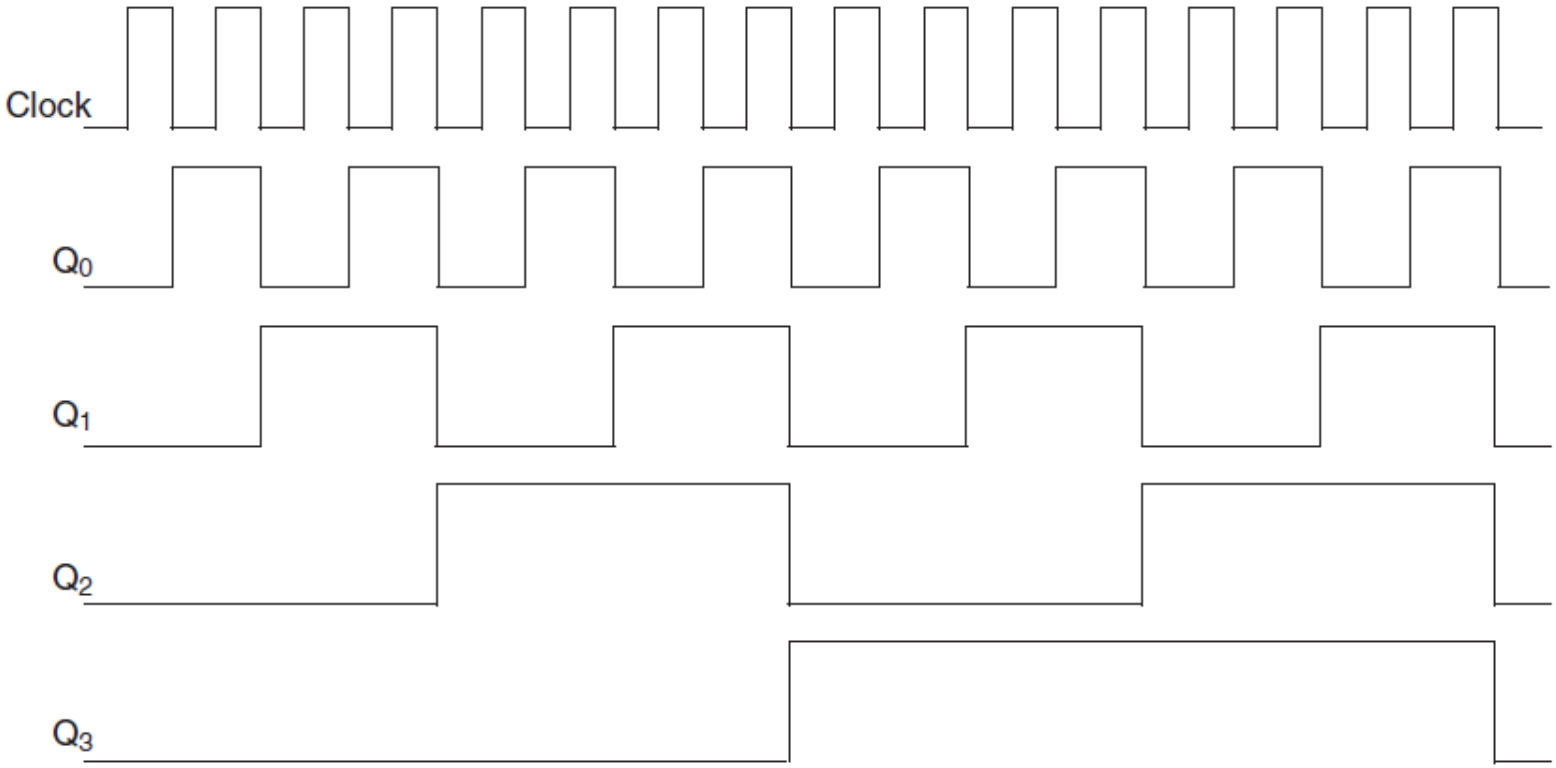


Bộ đếm không đồng bộ - Nhận xét

- Nếu bộ đếm từ FF tích cực sườn âm (H-L) và xung đồng hồ được đưa vào từ đầu ra Q thì bộ đếm theo thứ tự thuận (đếm lên).
- Nếu bộ đếm từ FF tích cực sườn âm (H-L) và xung đồng hồ được đưa vào từ đầu ra $\bar{Q}$ thì bộ đếm theo thứ tự nghịch (đếm xuống).
- Nếu bộ đếm từ FF tích cực sườn dương (L-H) và xung đồng hồ được đưa vào từ đầu ra Q thì bộ đếm theo thứ tự nghịch (đếm xuống).
- Nếu bộ đếm từ FF tích cực sườn dương (L-H) và xung đồng hồ được đưa vào từ đầu ra $\bar{Q}$ thì bộ đếm theo thứ tự thuận (đếm lên).

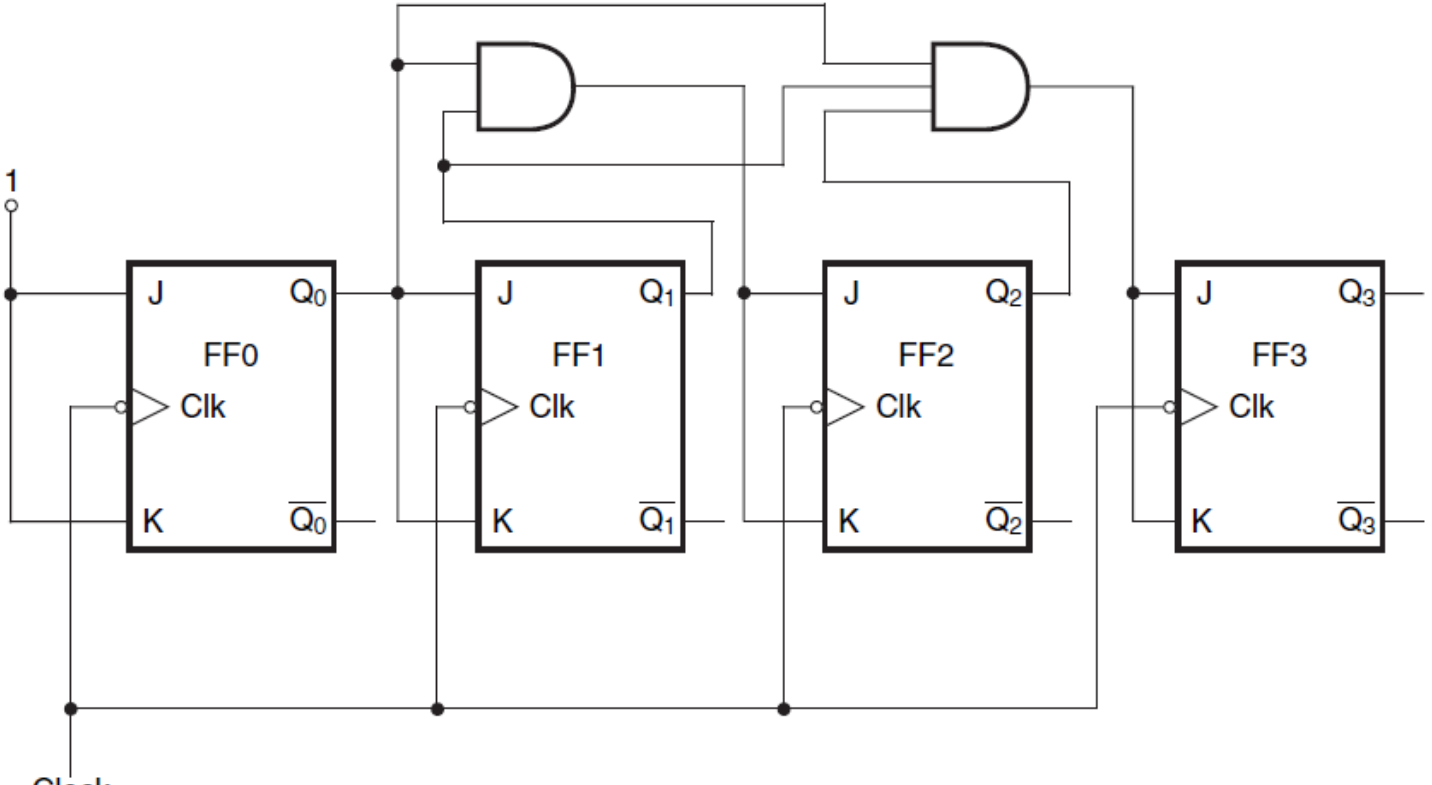
Bộ đếm đồng bộ (bộ đếm song song)

- Tất cả các FF được đồng bộ đồng thời bởi xung đồng hồ, tất cả FF thay đổi trạng thái tại cùng một thời điểm. Trễ truyền dẫn độc lập với số lượng FF.



Bộ đếm đồng bộ (bộ đếm song song)

- Tất cả các FF được đồng bộ đồng thời bởi xung đồng hồ, tất cả FF thay đổi trạng thái tại cùng một thời điểm. Trễ truyền dẫn độc lập với số lượng FF.



Bộ đếm đồng bộ (bộ đếm song song)

- Bộ đếm cơ số: Với tối thiểu N FF, có thể xây dựng bộ đếm cơ số nằm trong khoảng ($2^{N-1}+1 \rightarrow 2^{N}$)
  - Bộ đếm đồng bộ module 8
  - Bộ đếm đồng bộ module 10
  - Bộ đếm BCD
  - Bộ đếm với cơ số tùy ý

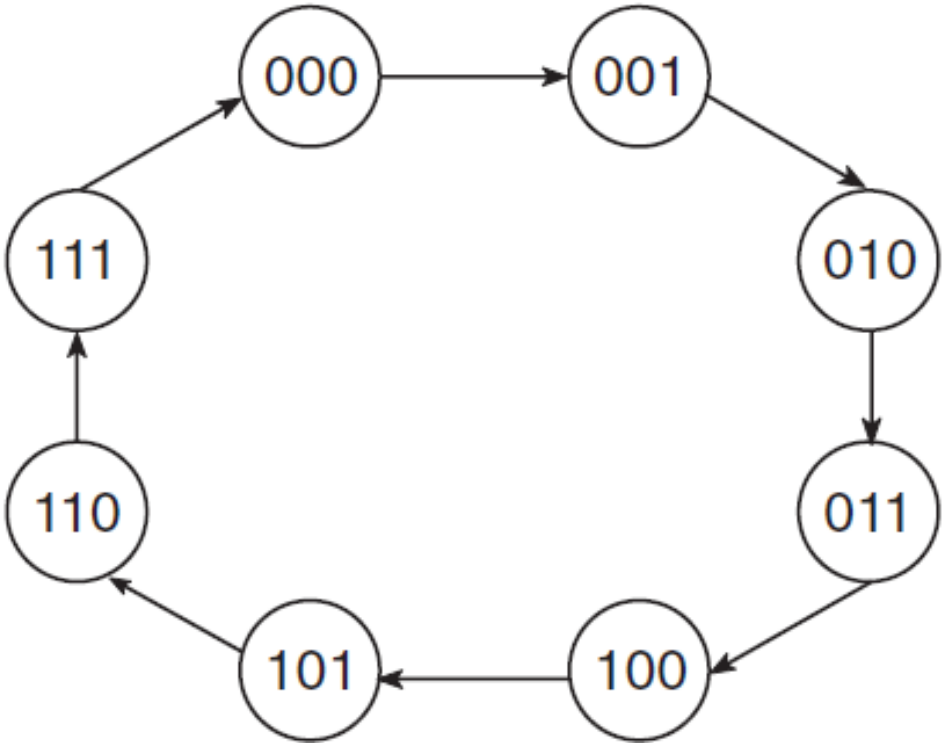
Thiết kế bộ đếm với số thứ tự tùy ý

- Bảng tác nhân kích thích của FF
- Liệt kê:
  - Trạng thái hiện tại
  - Trạng thái mong muốn tiếp theo
  - Các đầu vào FF cần thiết để đạt được trạng thái đó

Present state (Q_n)	Next state (Q_{n+1})	J	K	D
0	0	0	X	0
0	1	1	X	1
1	0	X	1	0
1	1	X	0	1

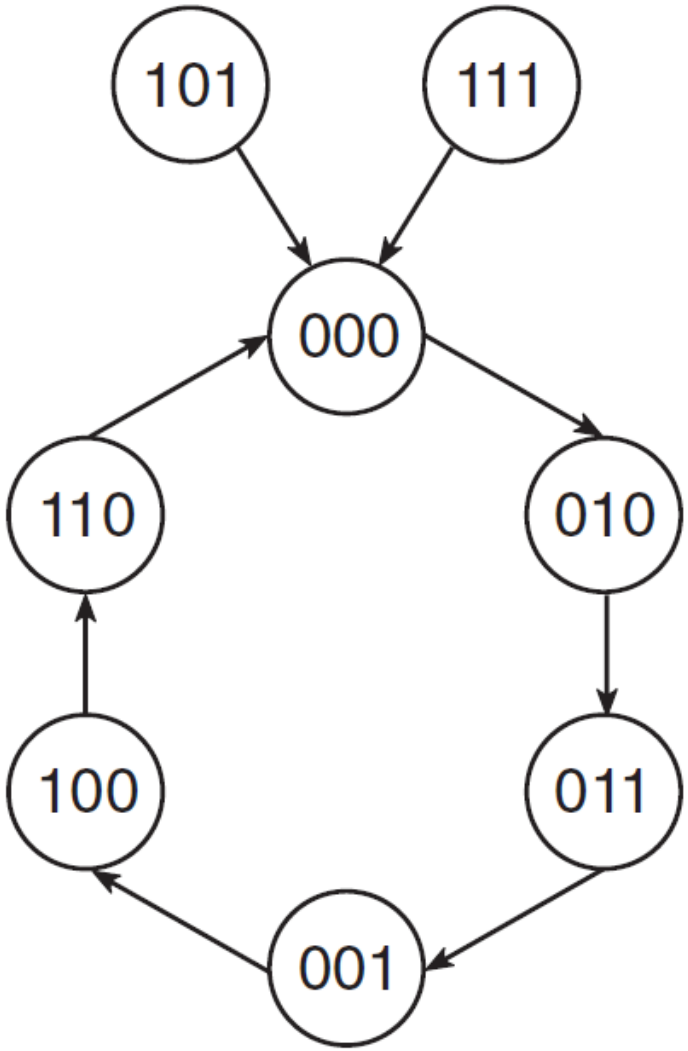
Thiết kế bộ đếm với số thứ tự tùy ý

- Vẽ đồ hình chuyển đổi trạng thái
  - Các trạng thái khác nhau được mô tả bởi các hình tròn
  - Mũi tên nối các hình tròn chỉ thứ tự chuyển đổi trạng thái sẽ diễn ra.



Thiết kế bộ đếm với số thứ tự tùy ý

- Các bước thiết kế:
  - Xác định số lượng FF cần thiết
  - Xác định các trạng thái không mong muốn
  - Vẽ đồ hình chuyển đổi trạng thái với cả các trạng thái không mong muốn
  - Các trạng thái không mong muốn nên được mô tả chuyển về bất kỳ trạng thái mong muốn nào



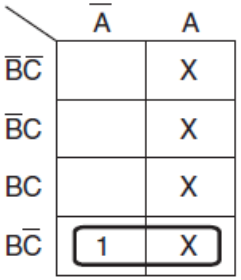
Thiết kế bộ đếm với số thứ tự tùy ý

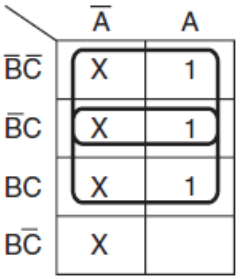
- Các bước thiết kế
  - Vẽ bảng tác nhân kích thích cho bộ đếm, trong đó liệt kê:
    - Các trạng thái hiện tại
    - Các trạng thái tiếp theo tương ứng với các trạng thái hiện tại
    - Đầu vào cho các FF theo yêu cầu

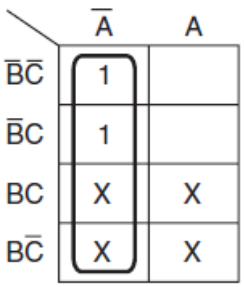
Present state			Next state			Inputs					
C	B	A	C	B	A	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	1	0	0	X	1	X	0	X
0	0	1	1	0	0	1	X	0	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	1	0	X	X	1	X	0
1	0	0	1	1	0	X	0	1	X	0	X
1	0	1	0	0	0	X	1	0	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X
1	1	1	0	0	0	X	1	X	1	X	1

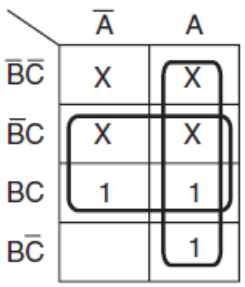
Thiết kế bộ đếm với số thứ tự tùy ý

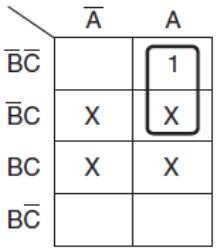
- Các bước thiết kế
  - Thiết kế các mạch logic cho các đầu vào J_A, K_A, J_B, K_B, J_C, K_C từ các đầu ra A, B, C…
  - Có thể sử dụng bảng Karnaugh cho mỗi đầu vào, tối thiểu hóa và biểu diễn bằng hàm Boolean.
  - Ví dụ:

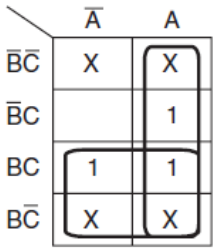












$$J_A = B.\overline{C}$$

$$K_A = \overline{B} + C$$

$$J_B = \overline{A}$$

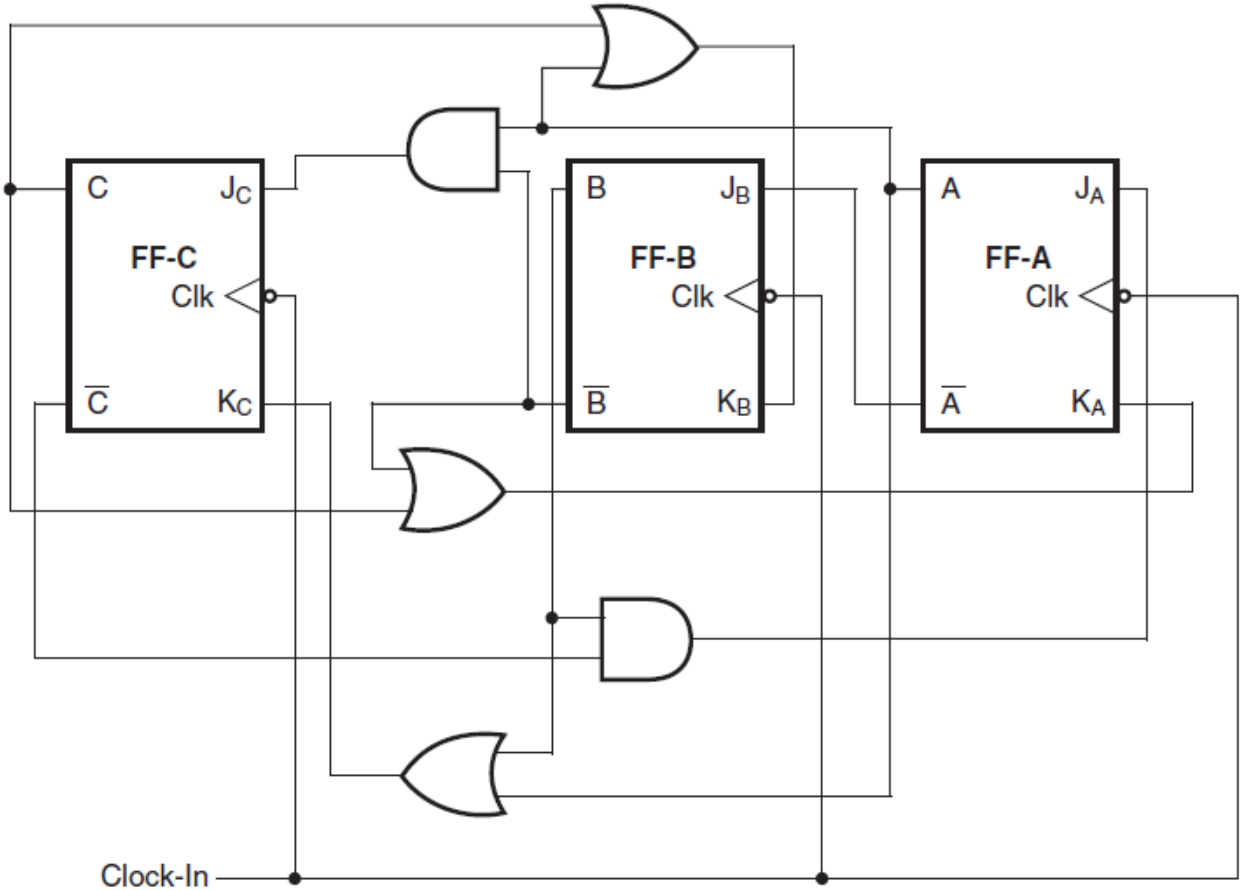
$$K_B = A + C$$

$$J_C = A.\overline{B}$$

$$K_C = A + B$$

Thiết kế bộ đếm với số thứ tự tùy ý

- Thực hiện mạch:



Ví dụ 8.15

- Cho bảng tác nhân kích thích như sau với X_1, X_2 là đầu vào của FF.

Present state (Q_n)	Next state (Q_{n+1})	Inputs	
		X_1	X_2
0	0	0	0
0	1	0	1
1	0	1	X
1	1	X	1

- Vẽ bảng tác nhân kích thích cho bộ đếm đồng bộ MOD-5, sử dụng FF để đếm chuỗi *000*, *001*, *011*, *101*, *110*, *000*, nếu trạng thái hiện tại là trạng thái không mong muốn, nó sẽ chuyển về 110 sau 1 xung đồng hồ.

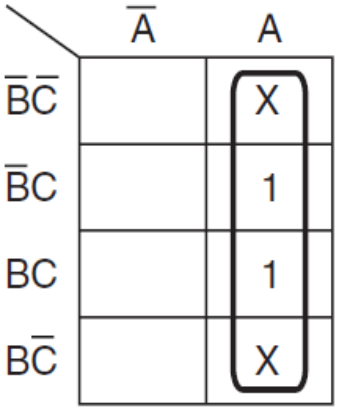
Ví dụ 8.15

- Bảng tác nhân kích thích:
  - Bộ đếm 5 trạng thái → 3 FFs
  - Các trạng thái không mong muốn được đưa về 110 (010, 100, 111 → 110)

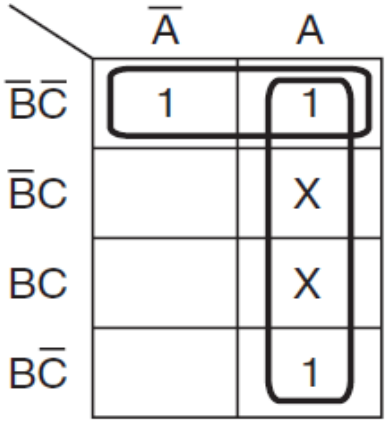
Present state			Next state			Inputs					
C	B	A	C	B	A	$X_1(A)$	$X_2(A)$	$X_1(B)$	$X_2(B)$	$X_1(C)$	$X_2(C)$
0	0	0	0	0	1	0	1	0	0	0	0
0	0	1	0	1	1	X	1	0	1	0	0
0	1	0	1	1	0	0	0	X	1	0	1
0	1	1	1	0	1	X	1	1	X	0	1
1	0	0	1	1	0	0	0	0	1	X	1
1	0	1	1	1	0	1	X	0	1	X	1
1	1	0	0	0	0	0	0	1	X	1	X
1	1	1	1	1	0	1	X	X	1	X	1

Ví dụ 8.15

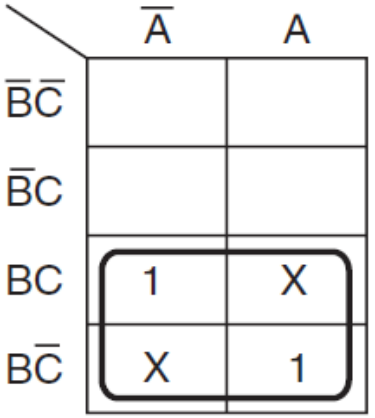
- Tối thiểu hóa từng đầu vào FF



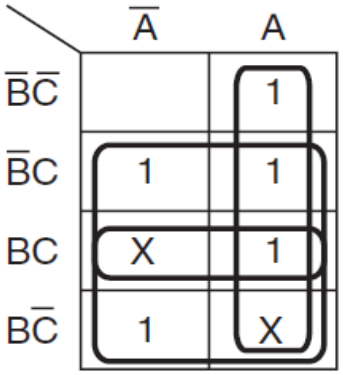
$X_1(A) = A$



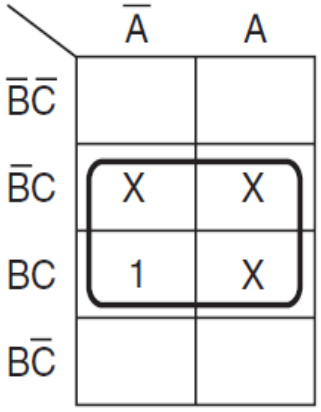
$X_2(A) = A + \overline{B}.\overline{C}$



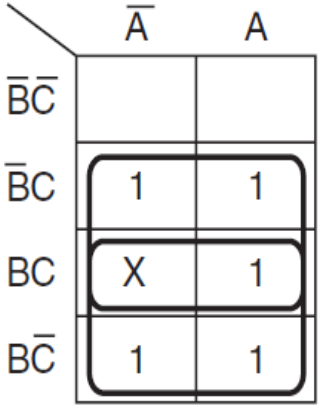
$X_1(B) = B$



$X_2(B) = A + B + C$



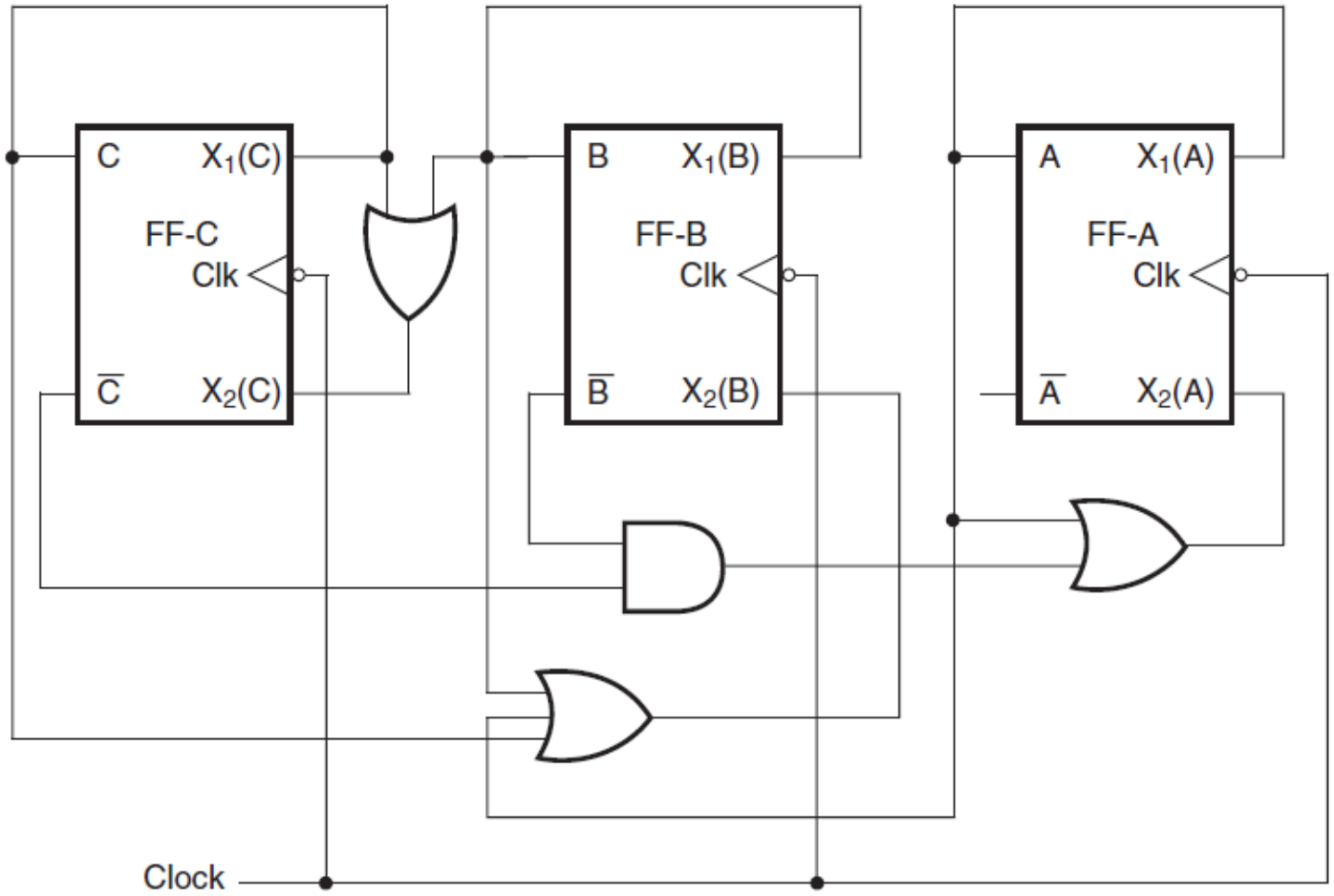
$X_1(C) = C$



$X_2(C) = B + C$

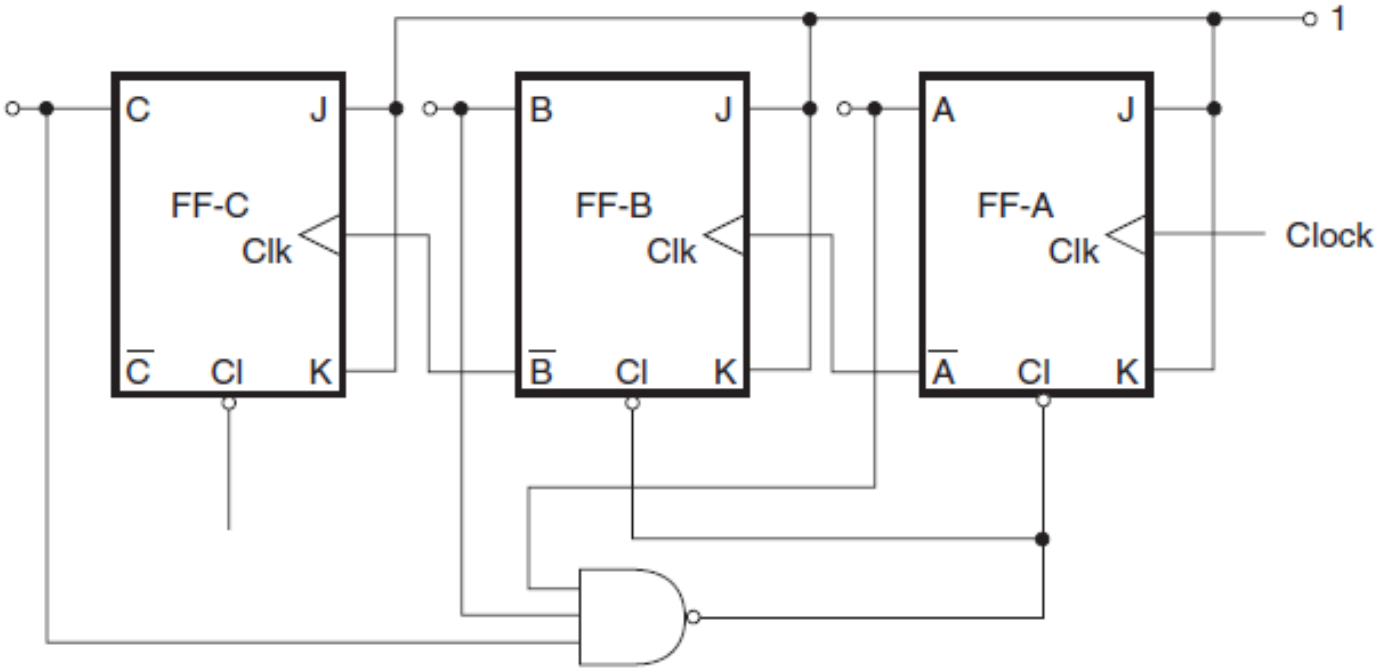
Ví dụ 8.15

- Thực hiện mạch:



Ví dụ 8.16

- Tìm thứ tự đếm của bộ đếm sau:



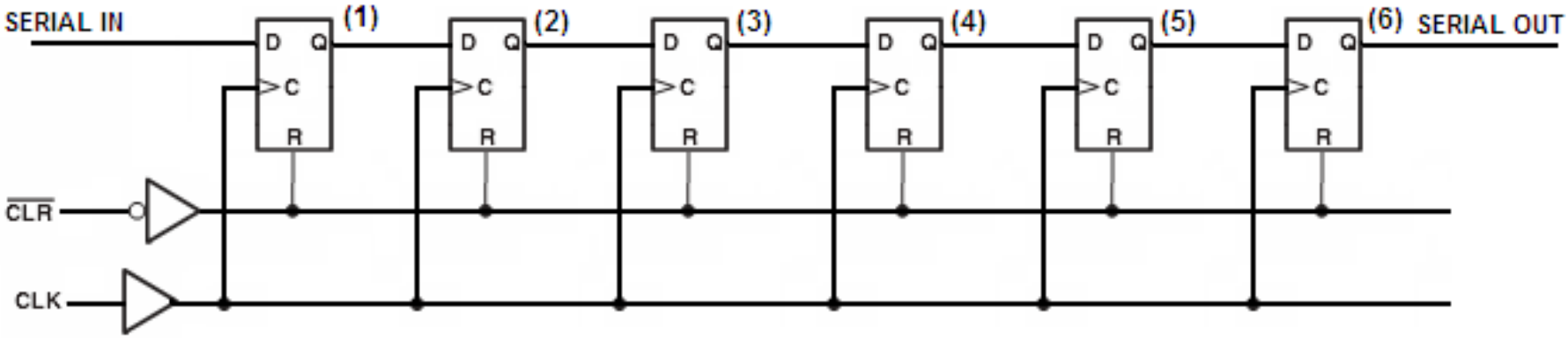
000, 001, 010, 011, 100, 101, 110, 000, . . .

8.5.2 Thanh ghi

- Thanh ghi có cấu tạo gồm các trigger nối với nhau
- Chức năng:
  - Để lưu trữ tạm thời thông tin
  - Dịch chuyển thông tin
- Lưu ý: cả thanh ghi và bộ nhớ đều dùng để lưu trữ thông tin, nhưng thanh ghi có chức năng dịch chuyển thông tin. Do đó, thanh ghi có thể sử dụng làm bộ nhớ, nhưng bộ nhớ không thể làm được thanh ghi.

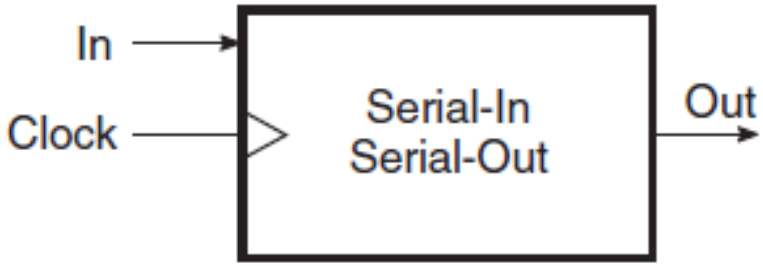
Thanh ghi dịch (Shift Register)

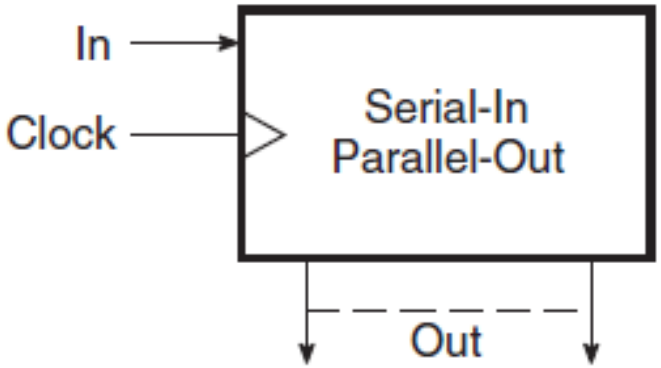
- Thanh ghi dịch được dùng để:
  - Biến đổi mã ***song song ⟷ nối tiếp***
  - Tạo trễ cho các dãy tín hiệu số
- Phần tử cơ bản của thanh ghi dịch là các D flip flop nối chuỗi 'nối tiếp' với nhau

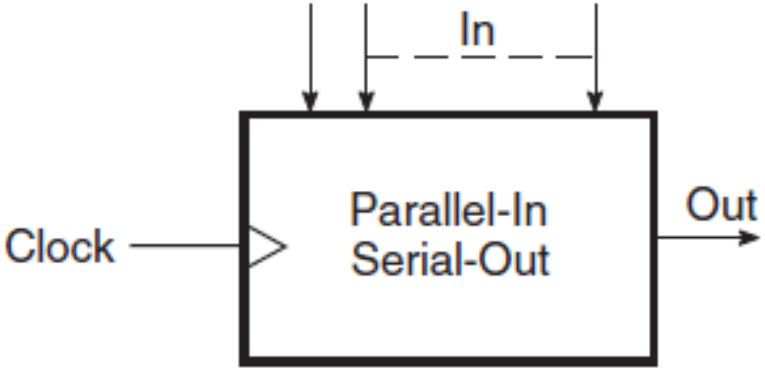


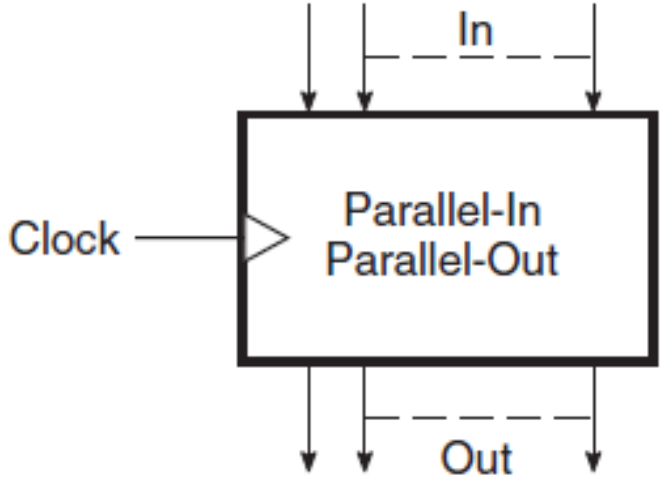
Thanh ghi dịch (Shift Register)

- Phân loại:



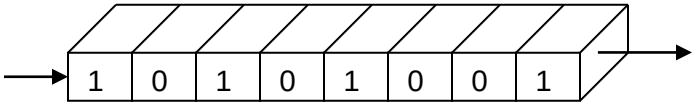




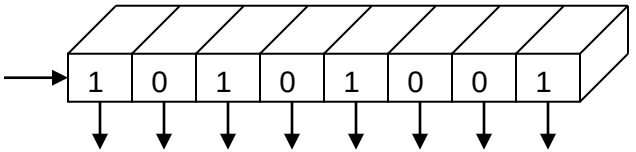


Vào ra thanh ghi

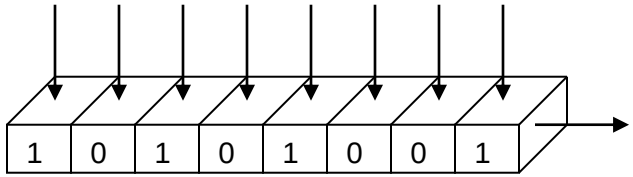
- Vào nối tiếp ra nối tiếp



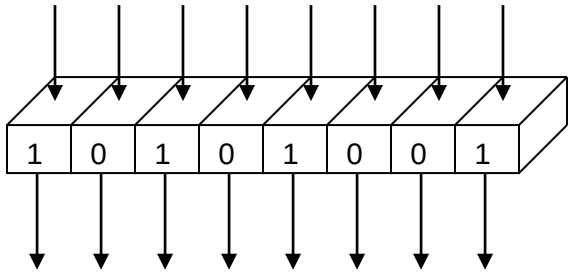
- Vào nối tiếp ra song song



- Vào song song ra nối tiếp

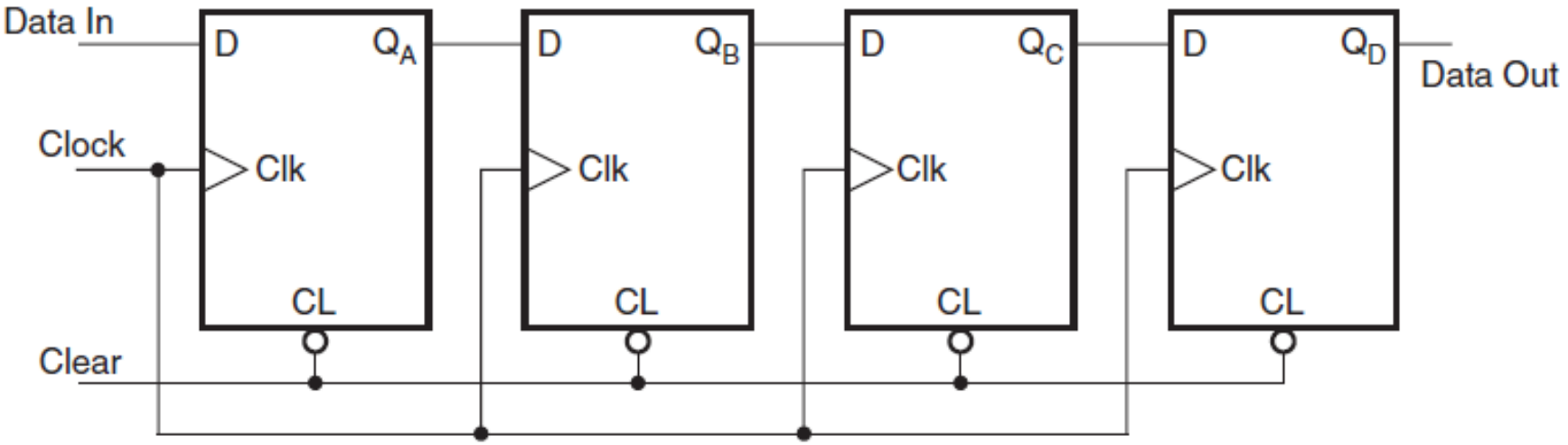


- Vào song song ra song song



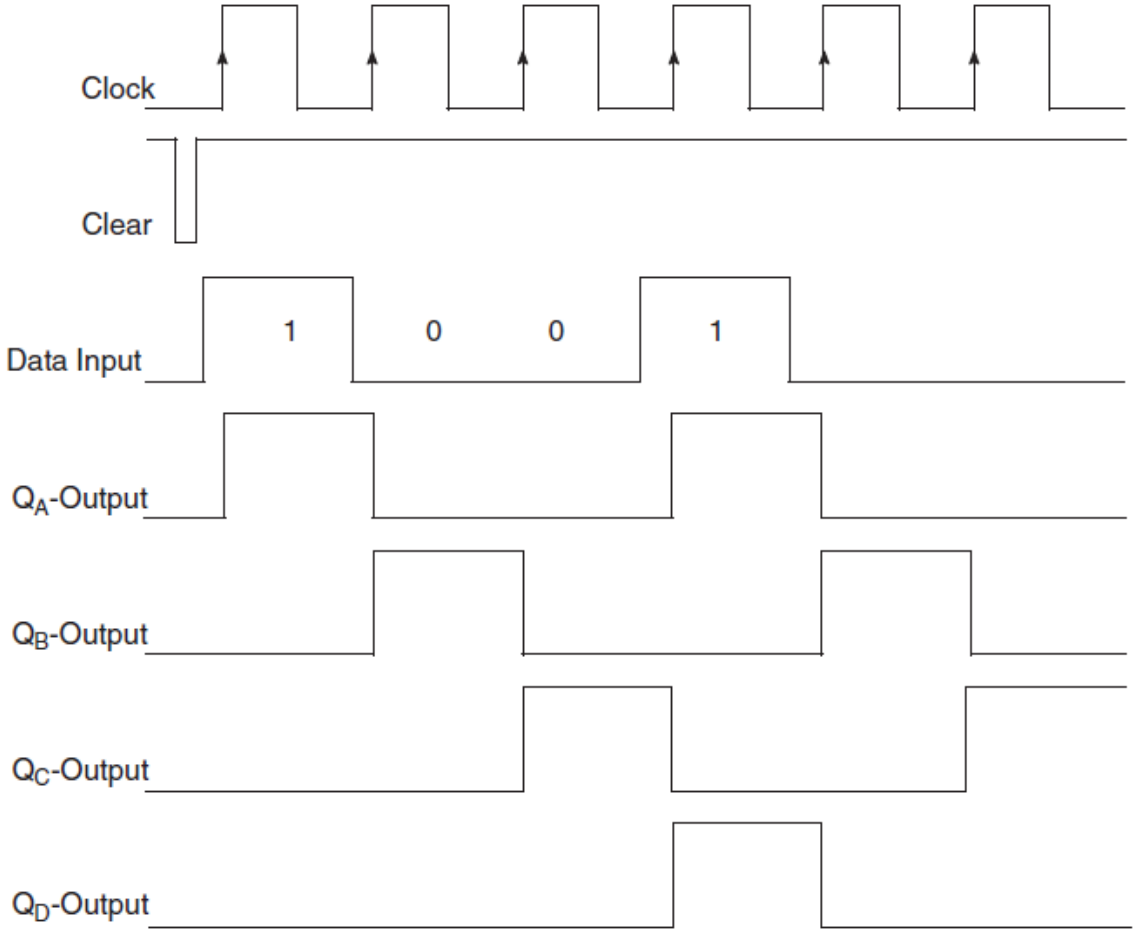
Vào nối tiếp – ra nối tiếp

- Sơ đồ mạch:



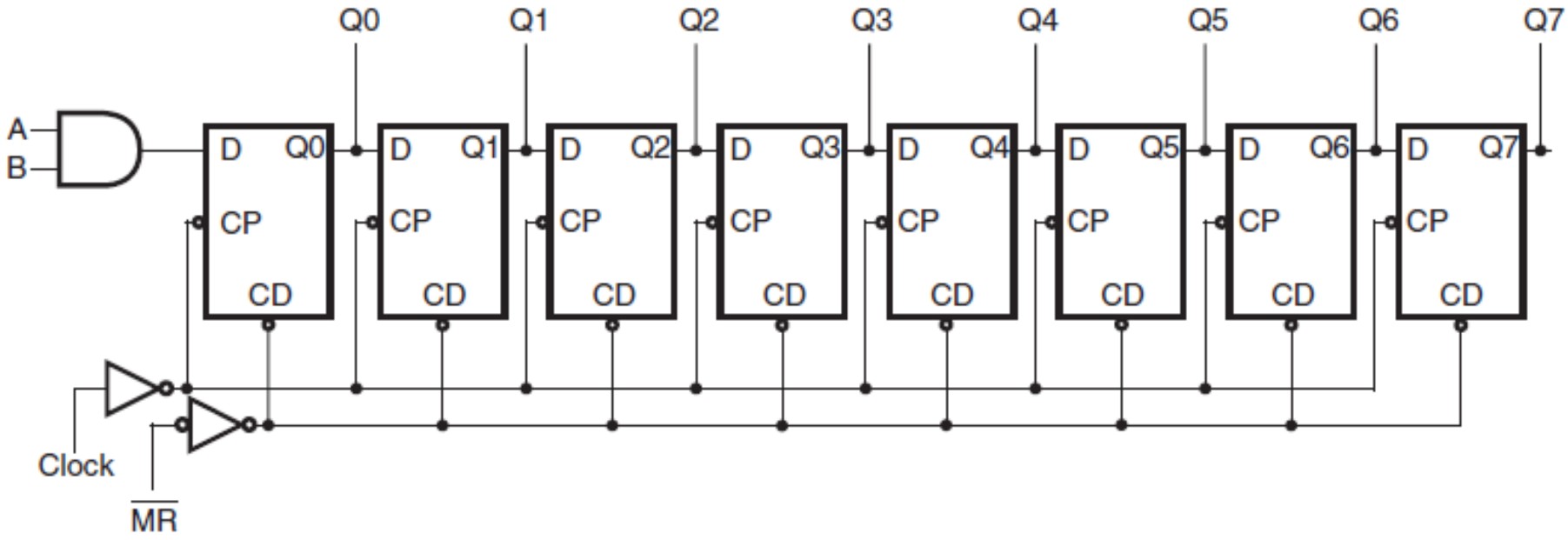
Vào nối tiếp – ra nối tiếp

- Biểu đồ thời gian:



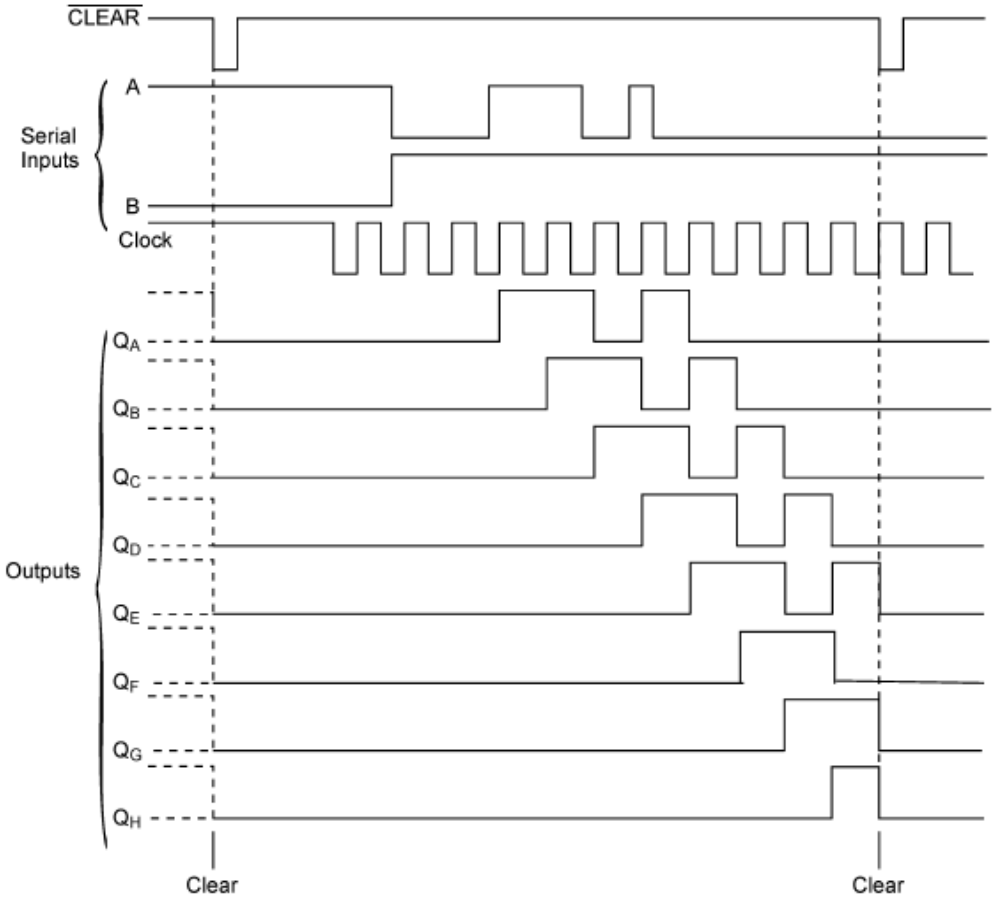
Vào nối tiếp – ra song song

- Sơ đồ mạch:



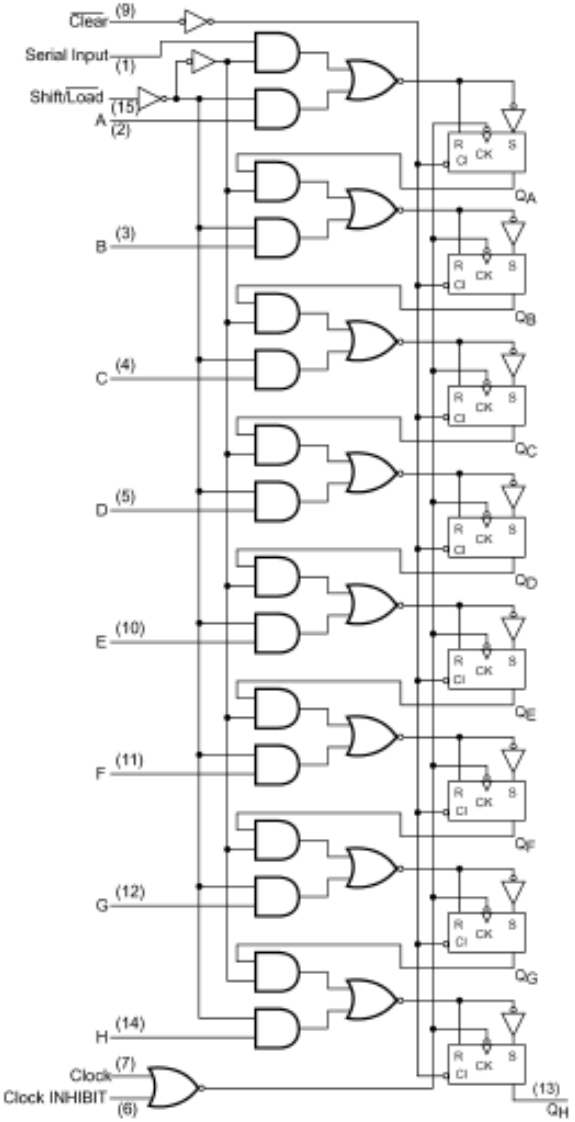
Vào nối tiếp – ra song song

- Biểu đồ thời gian:



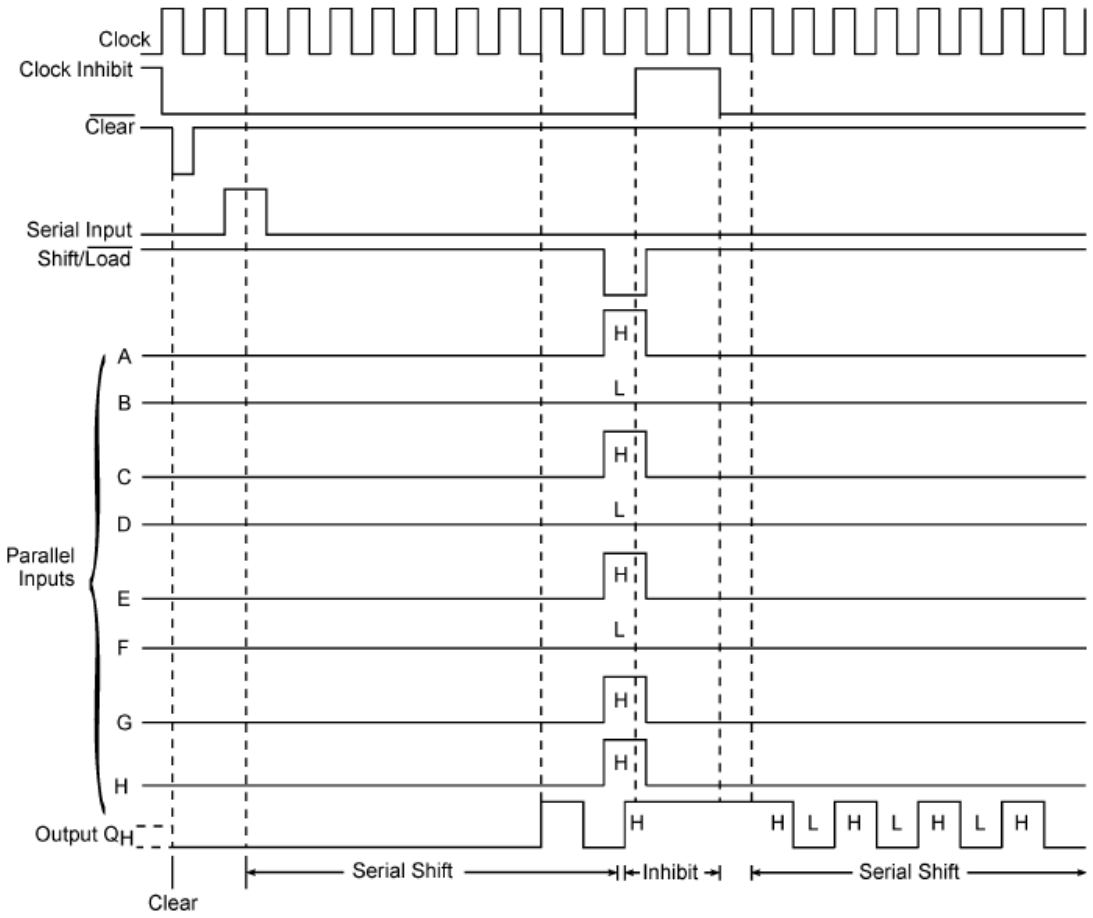
Vào song song – ra nối tiếp

- Sơ đồ mạch:



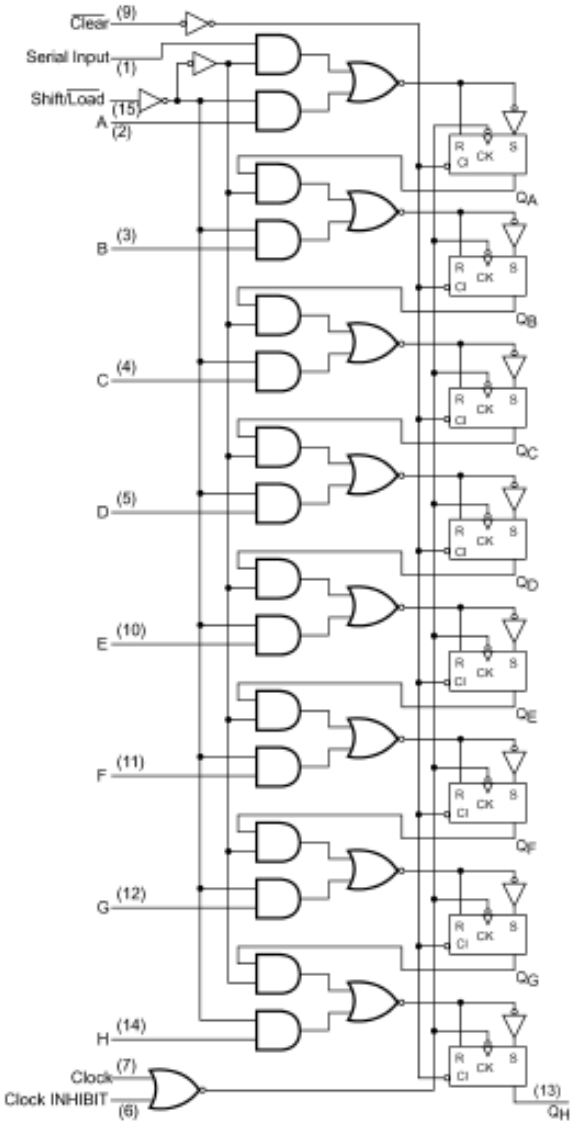
Vào song song – ra nối tiếp

- Biểu đồ thời gian:



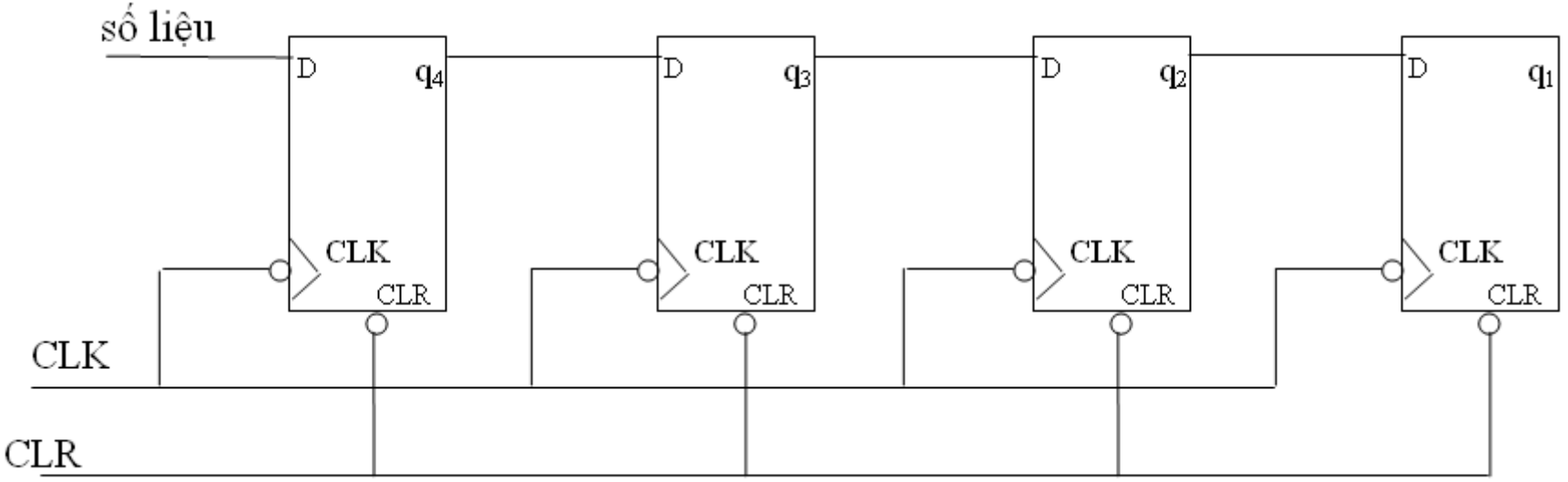
Vào song song – ra song song

- Sơ đồ mạch:



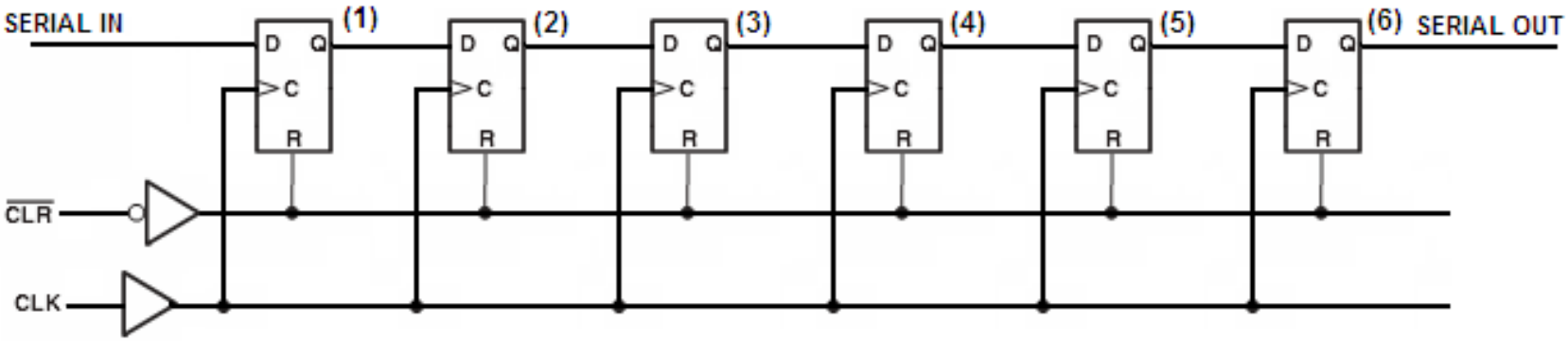
Thanh ghi dùng Trigger D

- Thanh ghi 4 bit vào nối tiếp ra song song dùng Trigger D



Thanh ghi dịch (Shift Register)

- Thanh ghi dịch được dùng để:
  - Biến đổi mã ***song song ⟷ nối tiếp***
  - Tạo trễ cho các dãy tín hiệu số
- Phần tử cơ bản của thanh ghi dịch là các D flip flop nối chuỗi 'nối tiếp' với nhau



Các loại thanh ghi dịch

- Các thanh ghi dịch được phân chia thành các loại sau:
  - Vào nối tiếp ra nối tiếp (SISO), ví dụ: 4006 (18 nhịp), 4517 (64 nhịp), 4557 (64 nhịp), 4562 (128 nhịp) …
  - Vào nối tiếp ra song song (SIPO), ví dụ: 4015 (4 bit), 4094 (8 bit), 74164 (8 bit) …
  - Vào song song ra nối tiếp (PISO), ví dụ: 4014, 4021, 74165, 74166... đều là các thanh ghi 8 bit
  - Vào song song ra song song (PIPO), ví dụ: 7495, 74195, 74395, 4035 (4 bit), 74323 (8 bit)...
  - Thanh ghi dịch vạn năng có thể dịch theo hai chiều, ví dụ 74194, 4194 (4 bit)

Kết thúc chương 8