

Chương 5

Mạch tuần tự

Nội dung chương 5

- Khái niệm chung
- Các loại Flip – Flop
- Chuyển đổi giữa các loại FF
- Bộ đếm
- Thanh ghi dịch

Khái niệm chung

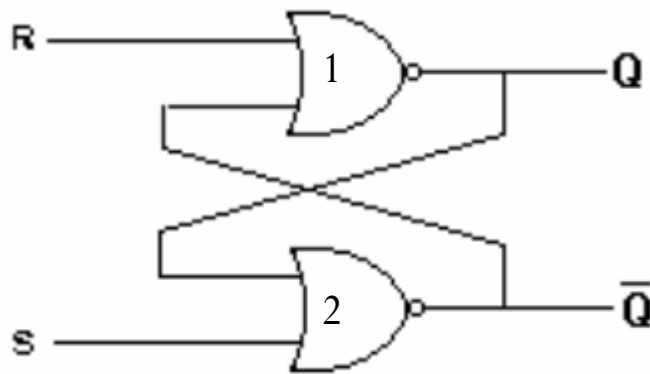
- Mạch tuần tự: các ngõ ra ở trạng thái kế tiếp vừa phụ thuộc vào trạng thái hiện tại của ngõ vào, vừa phụ thuộc trạng thái hiện tại của ngõ ra
- Khi các ngõ vào thay đổi trạng thái, các ngõ ra không thay đổi ngay mà chờ đến khi có xung đồng hồ
- Mạch tuần tự có tính đồng bộ và tính nhớ → cơ sở để thiết kế các bộ nhớ
- Cơ sở thiết kế mạch tuần tự là dựa trên Flip – Flop

Flip – Flop (FF)

- FF là mạch dao động đa hài hai trạng thái bền, được xây dựng trên cơ sở các cổng logic và hoạt động theo một bảng trạng thái cho trước
- Một FF thường có:
 - Một hoặc hai ngõ vào dữ liệu, một ngõ vào xung đồng hồ
 - Hai ngõ ra, thường ký hiệu Q (ngõ ra chính) và \bar{Q} (ngõ ra phụ)
- Phân loại FF:
 - FF không có tín hiệu điều khiển (không đồng bộ): Chốt (latch)
 - FF có tín hiệu điều khiển (đồng bộ): Chốt (latch) và FF

Flip – Flop không đồng bộ

- Chốt RS dùng cổng NOR: ngõ vào R và S tác động mức cao

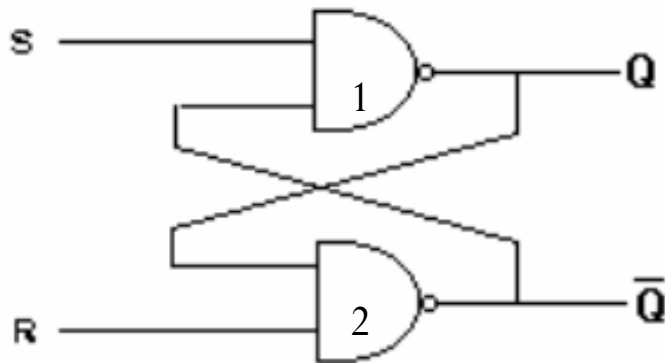


<i>S</i>	<i>R</i>	<i>Q</i>
0	0	Q^0
0	1	0
1	0	1
1	1	Cắm

- $S = 0, R = 1 \Rightarrow Q = 0$ hồi tiếp về cổng NOR 2 nên $\bar{Q} = 1$
- $S = 1, R = 0 \Rightarrow \bar{Q} = 0$ hồi tiếp về cổng NOR 1 nên $Q = 1$
- $S = 1, R = 1 \Rightarrow Q = \bar{Q} = 0$: Trạng thái cấm
- $S = R = 0$: giữ nguyên trạng thái trước đó
 - Nếu trạng thái trước đó có $Q = 0, \bar{Q} = 1$ hồi tiếp về cổng NOR 1 $\Rightarrow Q = 0$, chốt RS giữ nguyên trạng thái cũ
 - Nếu trạng thái trước đó có $Q = 1, \bar{Q} = 0$ hồi tiếp về cổng NOR 2 $\Rightarrow \bar{Q} = 0, Q = 1$ chốt RS giữ nguyên trạng thái cũ

Flip – Flop không đồng bộ (tt)

- Chốt RS dùng cổng NAND: ngõ vào R và S tác động mức thấp

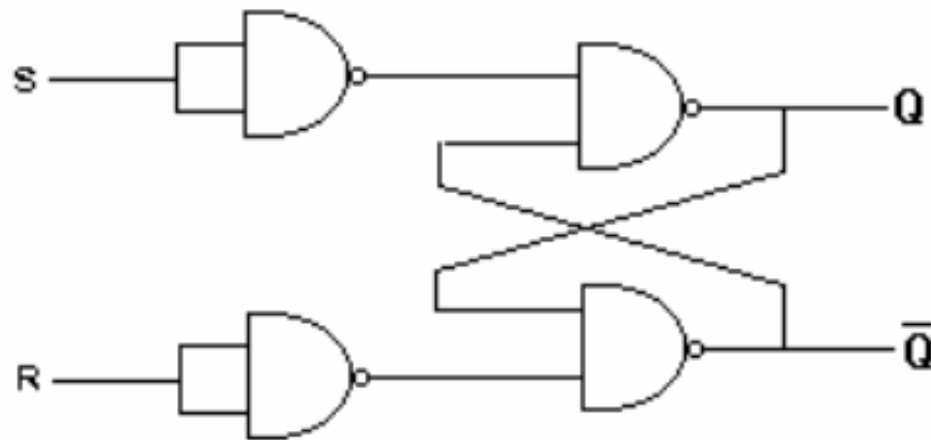


<i>S</i>	<i>R</i>	<i>Q</i>
0	0	Cấm
0	1	1
1	0	0
1	1	Q^0

- $S = 0, R = 1 \Rightarrow Q = 1$ hồi tiếp về cổng NAND 2 nên $\bar{Q} = 0$
- $S = 1, R = 0 \Rightarrow \bar{Q} = 1$ hồi tiếp về cổng NAND 1 nên $Q = 0$
- $S = R = 0 \Rightarrow Q = \bar{Q} = 1$: Trạng thái cấm
- $S = R = 1$: giữ nguyên trạng thái trước đó
 - Nếu trạng thái trước đó có $Q = 1, \bar{Q} = 0$ hồi tiếp về cổng NAND 1 $\Rightarrow Q = 1$, chốt RS giữ nguyên trạng thái cũ
 - Nếu trạng thái trước đó có $Q = 0, \bar{Q} = 1$ hồi tiếp về cổng NAND 2 $\Rightarrow \bar{Q} = 1, Q = 0$, chốt RS giữ nguyên trạng thái cũ

Flip – Flop không đồng bộ (tt)

- Chốt RS dùng cổng NAND: ngõ vào R và S tác động mức cao



<i>S</i>	<i>R</i>	<i>Q</i>
0	0	Q^0
0	1	0
1	0	1
1	1	<u>Cấm</u>

(Lưu ý: Sơ đồ này được sử dụng cho các phân tích chốt RS đồng bộ và RSFF về sau)

Flip – Flop đồng bộ

- Ck tác động mức 1
 - Ck tác động mức 0
 - Ck tác động sườn lên
 - Ck tác động sườn xuống
- Chốt (latch)
- FF



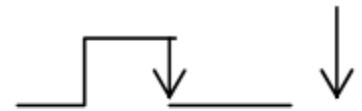
a.Mức 1



b.Mức 0



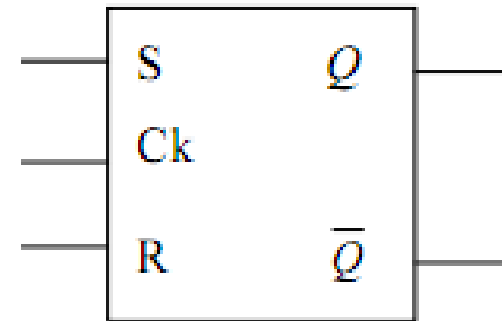
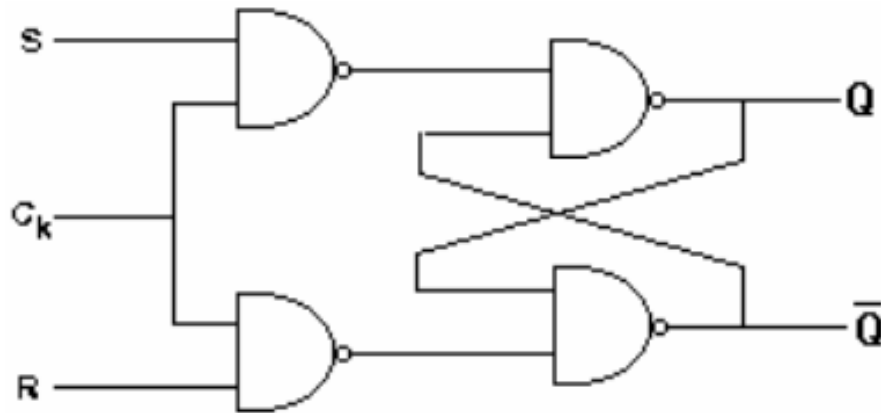
c.Sườn lên



d.Sườn xuống

Flip – Flop đồng bộ

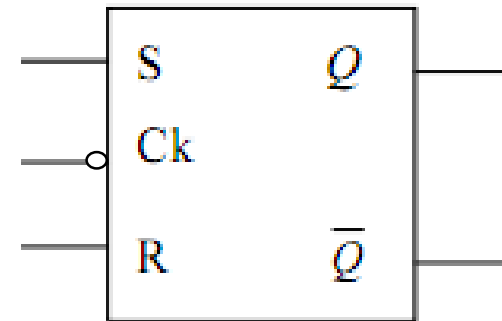
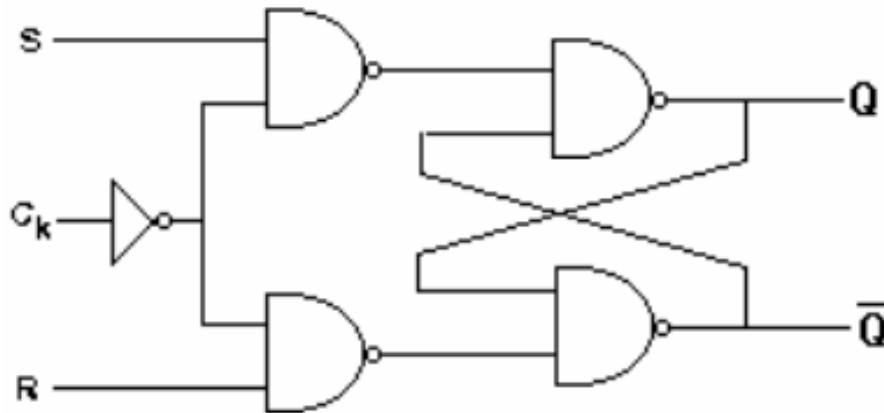
- Chốt RS: Ck tác động mức 1



S	R	Ck	Q
X	X	0	Q^0
0	0	1	Q^0
0	1	1	0
1	0	1	1
1	1	1	X

Flip – Flop đồng bộ

- Chốt RS: Ck tác động mức 0



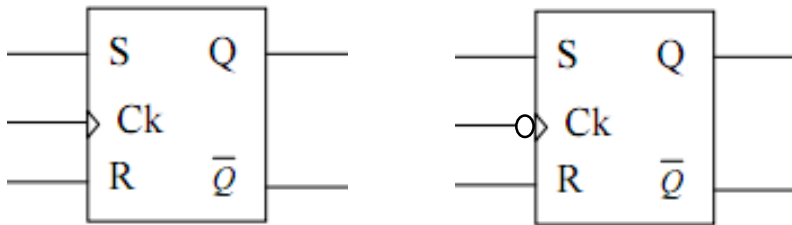
S	R	Ck	Q
X	X	1	Q^0
0	0	0	Q^0
0	1	0	0
1	0	0	1
1	1	0	X

Flip – Flop đồng bộ

- Ck tác động theo sườn (lên, xuống)
 - RSFF
 - JKFF
 - TFF
 - DFF

RSFF

- Kí hiệu:



- Bảng trạng thái:

S^n	R^n	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	X

S^n	R^n	Q^n	Q^{n+1}
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

RSFF (tt)

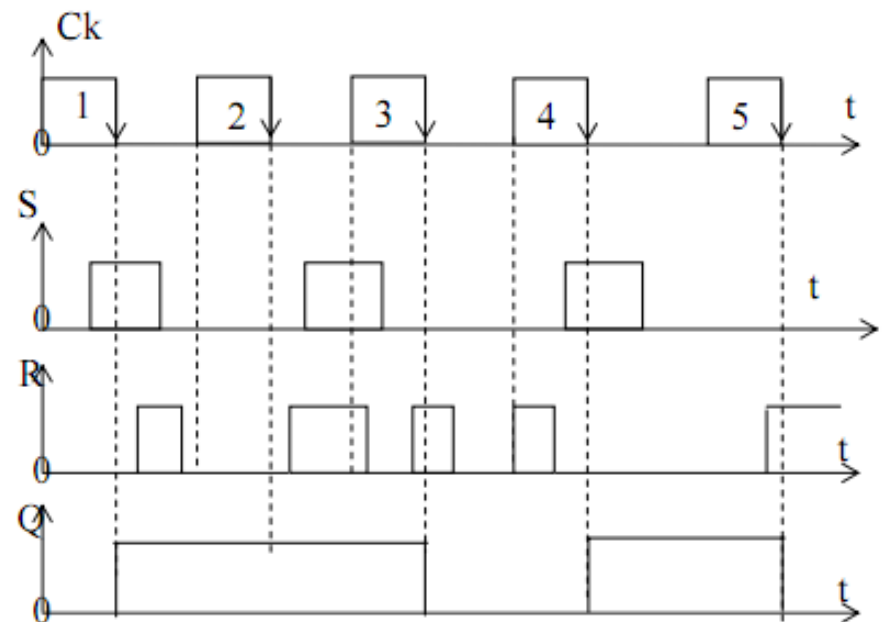
- Phương trình logic:

$Q^{n+1} \backslash Q^n \begin{matrix} S^n R^n \end{matrix}$	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$$Q^{n+1} = S^n + \overline{R^n} Q^n$$

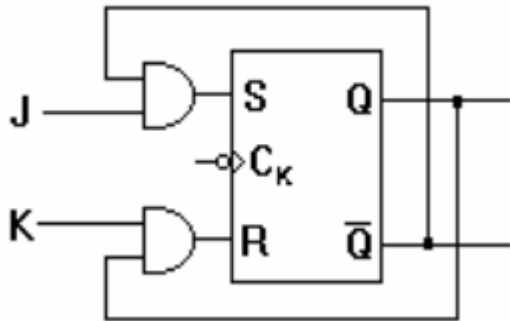
$(S^n \cdot R^n = 0)$

- Dạng sóng: (Ck tác động sườn xuống)



JKFF

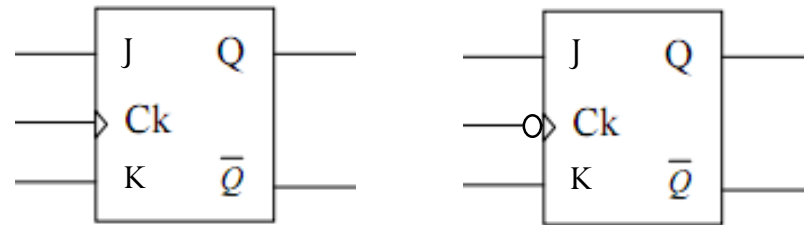
- Cải tiến từ RSFF



- Bảng trạng thái:

J^n	K^n	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	$\overline{Q^n}$

- Kí hiệu:



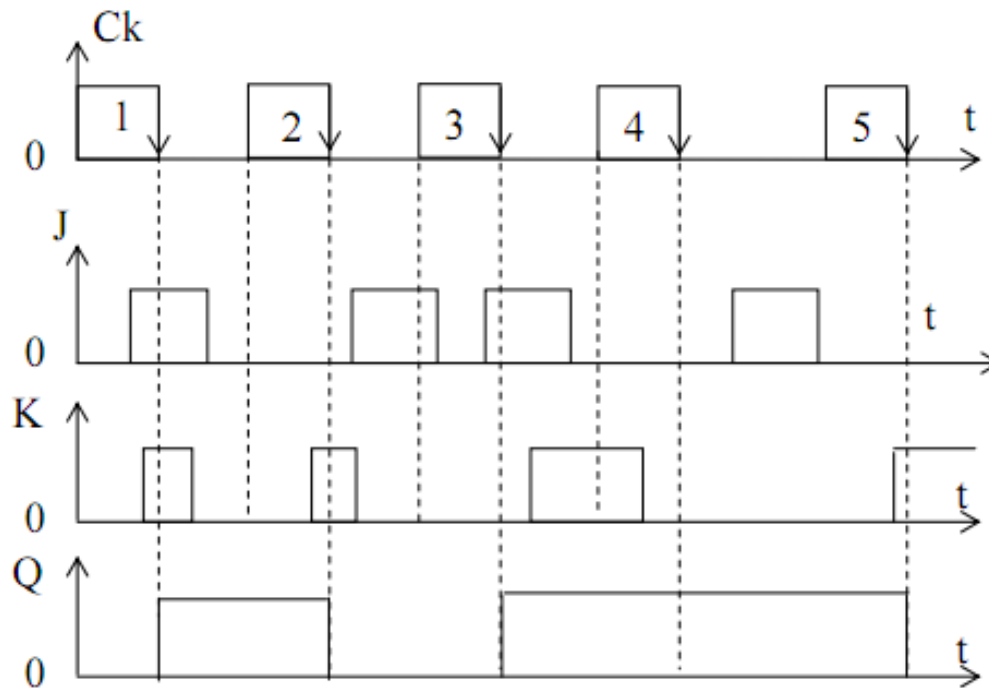
J^n	K^n	Q^n	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JKFF (tt)

- Phương trình logic:

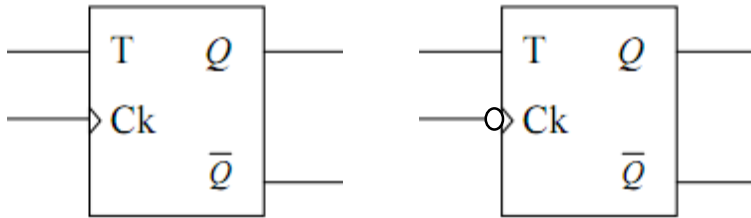
$$Q^{n+1} = J^n \overline{Q}^n + \overline{K}^n Q^n$$

- Dạng sóng: (Ck tác động sườn xuống)



TFF

- Kí hiệu:



- Bảng trạng thái:

T^n	Q^{n+1}
0	Q^n
1	$\overline{Q^n}$

- Phương trình logic:

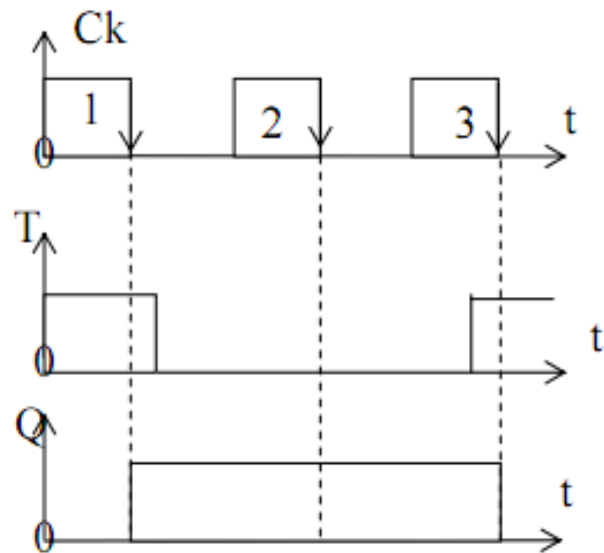
$$Q^{n+1} = \overline{T^n} Q^n + T^n \overline{Q^n}$$

$$Q^{n+1} = T^n \oplus Q^n$$

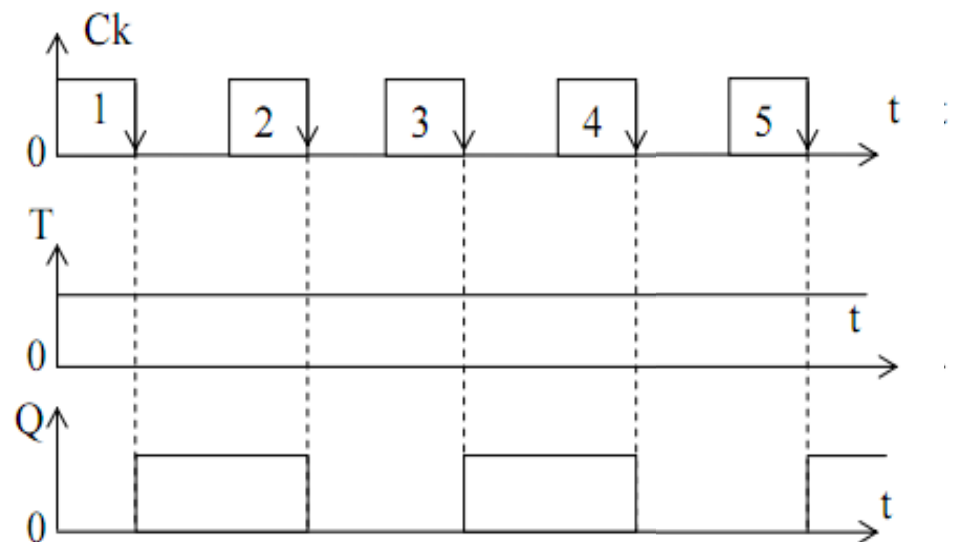
T^n	Q^n	Q^{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

TFF (tt)

- Dạng sóng: (Ck tác động sườn xuống)



Khi ngõ vào T luôn bằng 1:

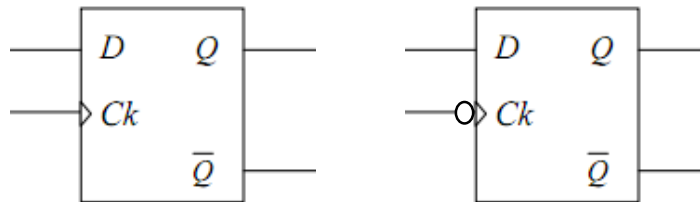


$$f_Q = \frac{f_{Ck}}{2}$$

- Vậy khi $T = 1$, TFF đóng vai trò mạch chia tần số xung đồng hồ
- Nếu ghép n TFF với nhau và các ngõ vào T luôn bằng 1 ta có $f_Q = \frac{f_{Ck}}{2^n}$

DFF

- Kí hiệu:



- Bảng trạng thái:

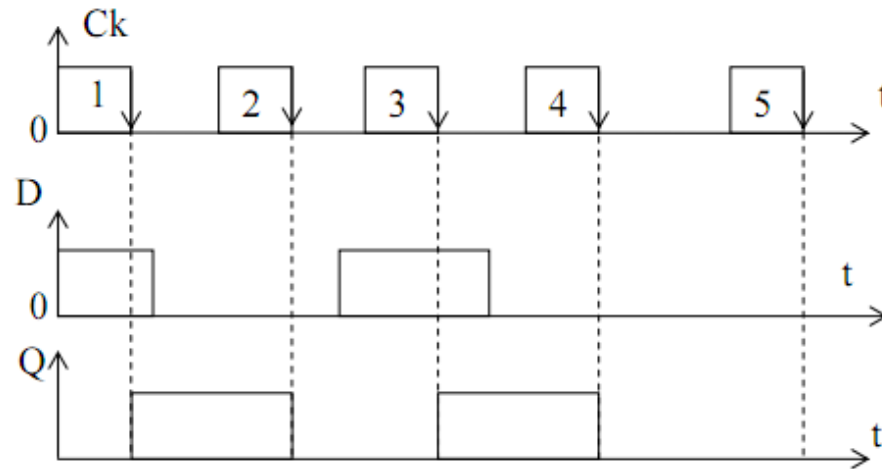
D^n	Q^{n+1}
0	0
1	1

D^n	Q^n	Q^{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

- Phương trình logic: $Q^{n+1} = D^n$

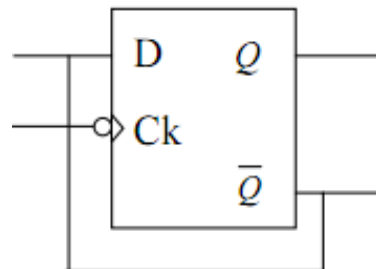
DFF (tt)

- Dạng sóng:



- Ứng dụng:

- Lưu trữ dữ liệu để chế tạo bộ nhớ và thanh ghi
- Chia tần số nếu mắc hồi tiếp ngõ ra \bar{Q} về ngõ vào D



Chuyển đổi giữa các loại Flip – Flop

- Biến đổi trực tiếp: dùng các tiên đề và định lý của đại số Boole
- Dùng bảng đầu vào kích và bảng Karnaugh

Chuyển đổi giữa các loại Flip – Flop

- Biến đổi trực tiếp

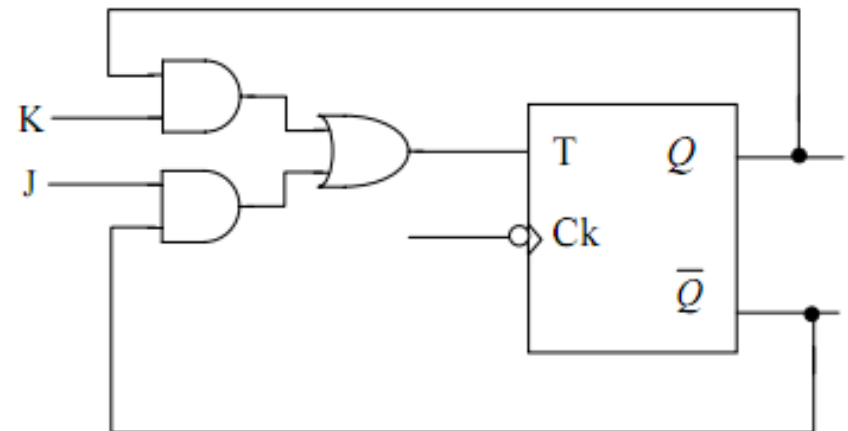
- TFF \rightarrow JKFF

TFF có phương trình: $Q^{n+1} = T^n \oplus Q^n$

JKFF có phương trình: $Q^{n+1} = J^n \overline{Q}^n + \overline{K}^n Q^n$

$$\rightarrow T^n \oplus Q^n = J^n \overline{Q}^n + \overline{K}^n Q^n$$

$$\begin{aligned} T^n &= Q^n \oplus (J^n \overline{Q}^n + \overline{K}^n Q^n) \\ &= Q^n \overline{(J^n \overline{Q}^n + \overline{K}^n Q^n)} + \overline{Q}^n (J^n \overline{Q}^n + \overline{K}^n Q^n) \\ &= Q^n (\overline{J}^n + Q^n) (\overline{K}^n + \overline{Q}^n) + \overline{Q}^n J^n \\ &= Q^n \overline{J}^n \overline{K}^n + Q^n \overline{K}^n + \overline{Q}^n J^n \\ &= Q^n \overline{K}^n + \overline{Q}^n J^n \end{aligned}$$

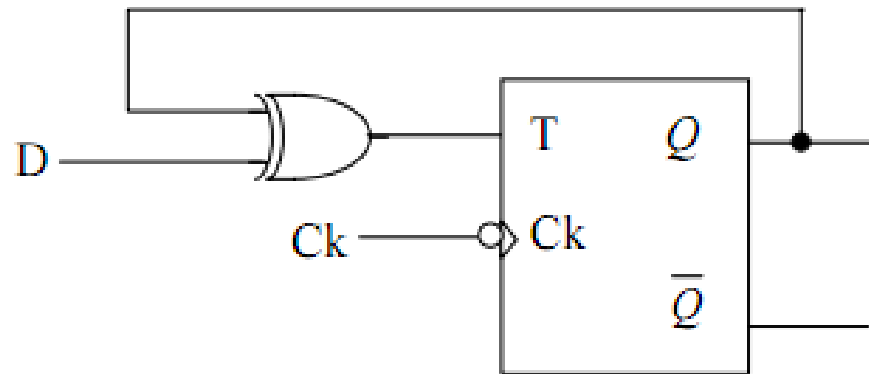


Chuyển đổi giữa các loại Flip – Flop

- Biến đổi trực tiếp

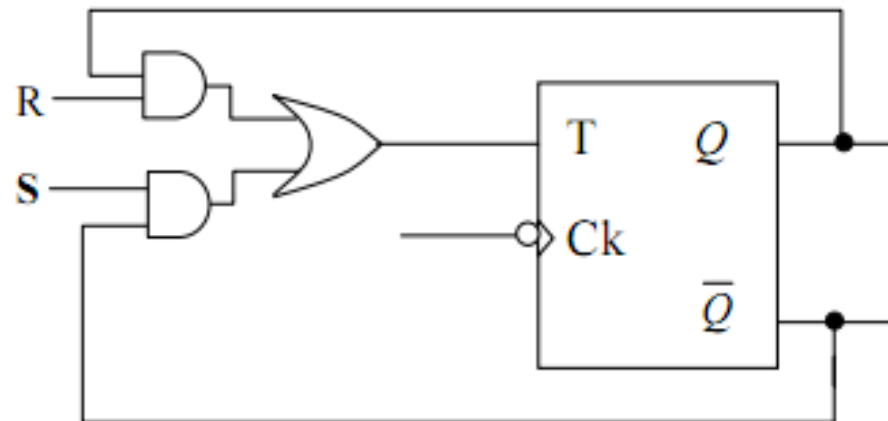
- TFF → DFF

$$T^n = D^n \oplus Q^n$$



- TFF → RSFF

$$T^n = Q^n R^n + \overline{Q^n} S^n$$

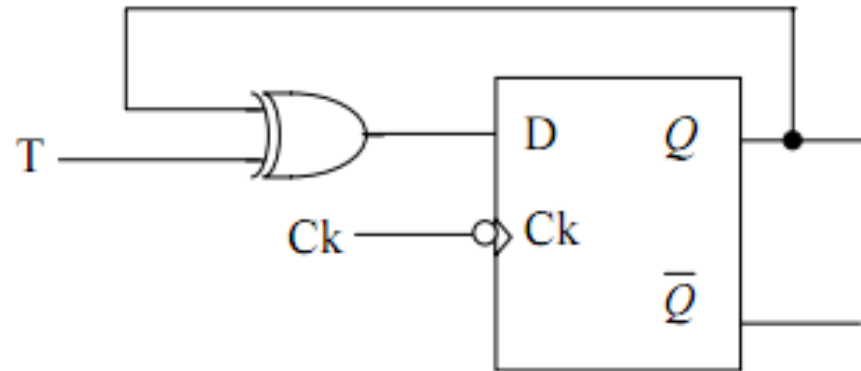


Chuyển đổi giữa các loại Flip – Flop

- Biến đổi trực tiếp

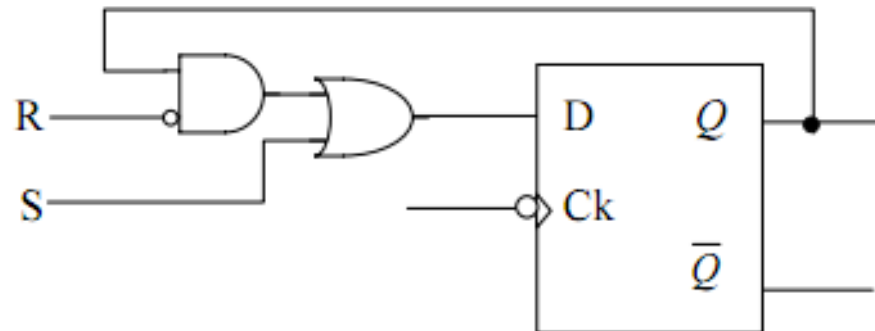
- DFF → TFF

$$D^n = T^n \oplus Q^n$$



- DFF → RSFF

$$D^n = S^n + R^n \bar{Q}^n$$

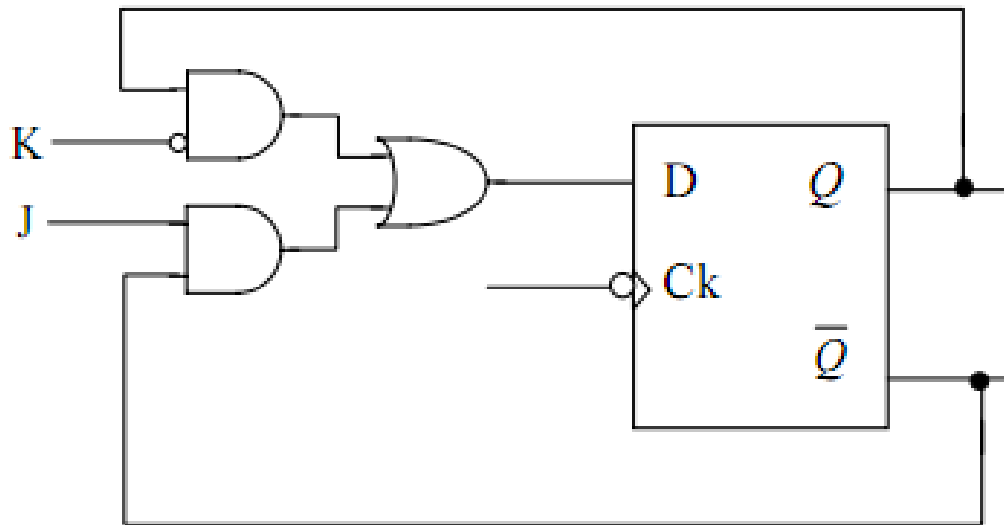


Chuyển đổi giữa các loại Flip – Flop

- Biến đổi trực tiếp

- DFF → JKFF

$$D^n = J^n \overline{Q^n} + \overline{K^n} Q^n$$



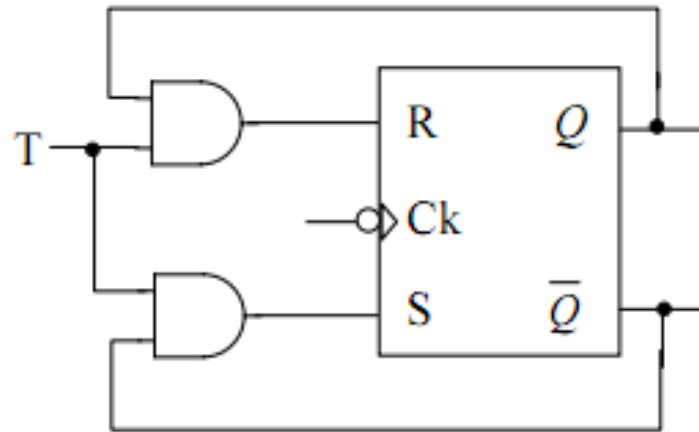
Chuyển đổi giữa các loại Flip – Flop

- Biến đổi trực tiếp

- RSFF \rightarrow TFF

$$S^n = T^n \overline{Q^n}$$

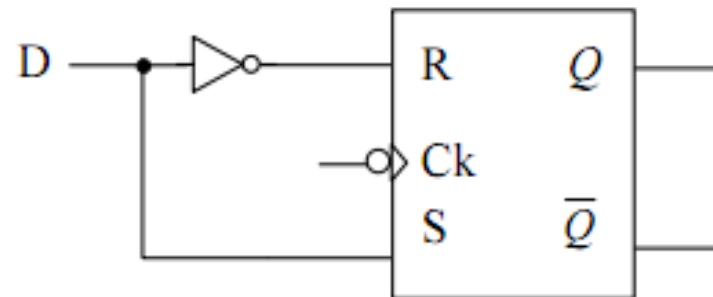
$$R^n = T^n Q^n$$



- RSFF \rightarrow DFF

$$S^n = D^n$$

$$R^n = \overline{D^n}$$



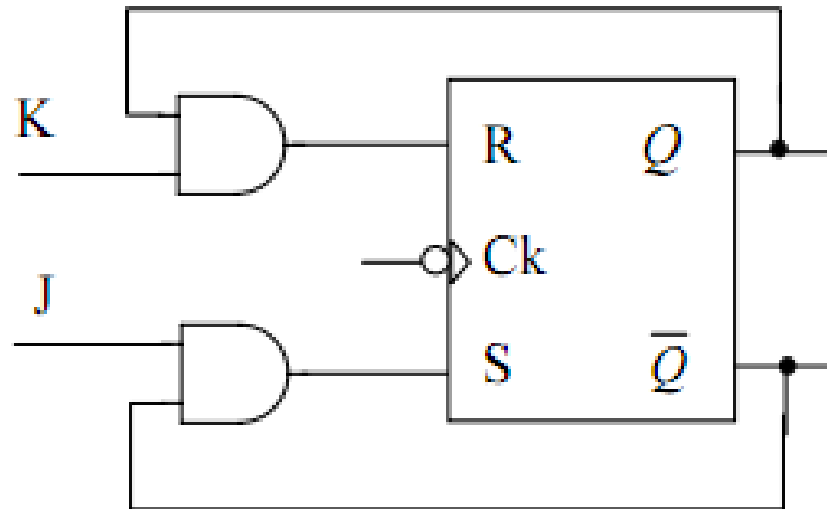
Chuyển đổi giữa các loại Flip – Flop

- Biến đổi trực tiếp

- RSFF \rightarrow JKFF

$$S^n = J^n \overline{Q}^n$$

$$R^n = K^n Q^n$$



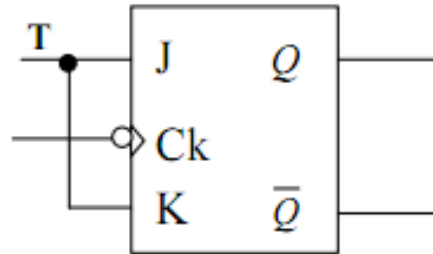
Chuyển đổi giữa các loại Flip – Flop

- Biến đổi trực tiếp

- JKFF \rightarrow TFF

$$J^n = T^n$$

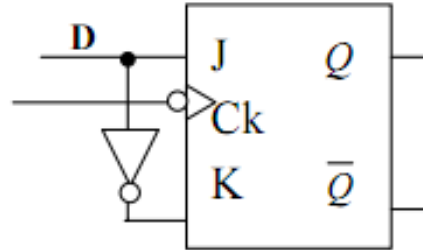
$$K^n = T^n$$



- JKFF \rightarrow DFF

$$J^n = D^n$$

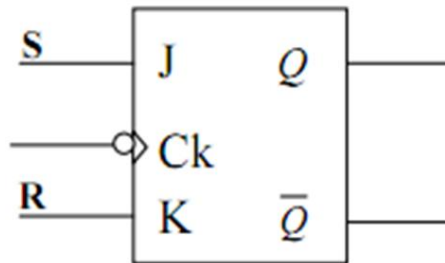
$$K^n = \overline{D^n}$$



- JKFF \rightarrow RSFF

$$J^n = S^n$$

$$K^n = R^n$$



Chuyển đổi giữa các loại Flip – Flop

- Dùng bảng đầu vào kích và bảng Karnaugh
 - Bảng đầu vào kích: Điều kiện ngõ vào để đạt yêu cầu thay đổi ở ngõ ra

Q^n	Q^{n+1}	S^n	R^n	J^n	K^n	T^n	D^n
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	1	0
1	1	X	0	X	0	0	1

Chuyển đổi giữa các loại Flip – Flop

- Dùng bảng đầu vào kích và bảng Karnaugh

- Ví dụ: JKFF chuyển đổi thành RSFF

Ta cần tìm các hàm: $J^n = f(S^n, R^n, Q^n)$, $K^n = f(S^n, R^n, Q^n)$

Dựa vào bảng đầu vào kích, lập bảng Karnaugh:

J Q ⁿ	SR			
	00	01	11	10
0	0	0	X	1
1	X	X	X	X

K Q ⁿ	SR			
	00	01	11	10
0	X	X	X	X
1	0	1	X	0

$$J^n = S^n, K^n = R^n$$

Bộ đếm

- Xây dựng dựa trên cơ sở các Flip – Flop ghép với nhau
- Phân loại:
 - Theo hệ đếm: bộ đếm thập phân, bộ đếm nhị phân
 - Theo hướng đếm: mạch đếm lên, mạch đếm xuống, mạch đếm vòng
 - Theo tín hiệu chuyển: bộ đếm nối tiếp, bộ đếm song song, bộ đếm hỗn hợp
 - Chức năng điều khiển: bộ đếm đồng bộ, bộ đếm không đồng bộ

Bộ đếm nối tiếp

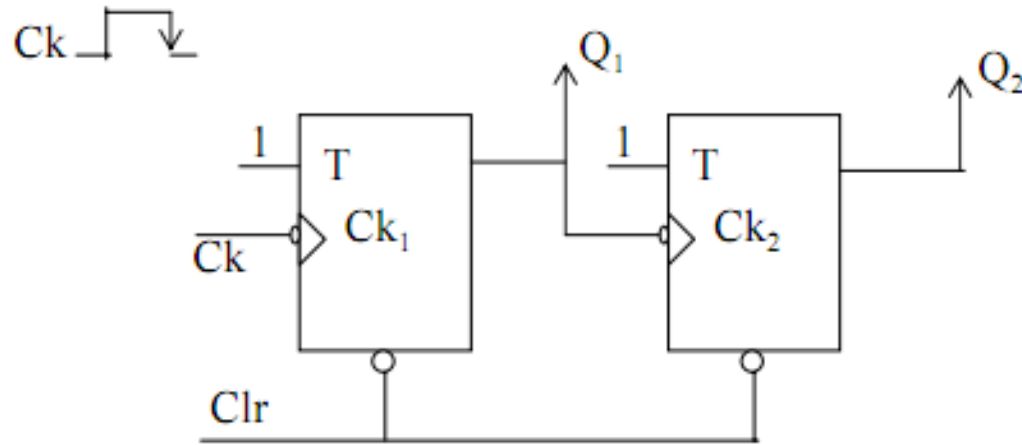
- Bộ đếm nối tiếp: bộ đếm trong đó các TFF hoặc JKFF giữ chức năng của TFF được ghép nối tiếp với nhau và hoạt động theo một loại mã duy nhất là BCD 8421
- Phân loại:
 - Đếm lên
 - Đếm xuống
 - Đếm lên/xuống
 - Modulo M
- Ưu điểm: đơn giản, dễ thiết kế
- Nhược điểm: với dung lượng lớn, sử dụng nhiều FF thì thời gian trễ tích lũy lớn → kết quả sai

Bộ đếm nối tiếp - Đếm lên

- Bộ đếm lên có nội dung đếm tăng dần
- T/(J, K) luôn luôn ở mức logic 1 và ngõ ra của TFF/JKFF đứng trước nối với ngõ vào Ck của TFF/JKFF đứng sau
 - Ck tác động sườn xuống: TFF hoặc JKFF ghép với nhau theo quy luật: $Ck_{i+1} = Q_i$
 - Ck tác động sườn lên: TFF hoặc JKFF ghép với nhau theo quy luật: $Ck_{i+1} = \overline{Q_i}$

Bộ đếm nối tiếp - Đếm lên (tt)

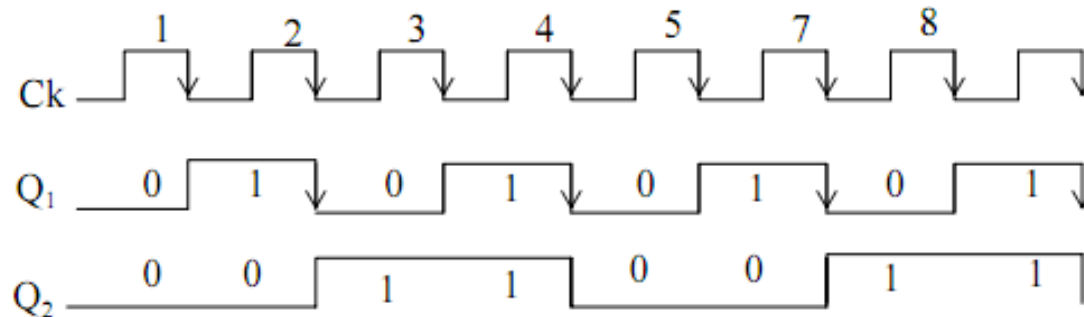
- Xét mạch đếm nối tiếp, đếm 4, đếm lên, dùng TFF:
 - Trường hợp Ck tác động theo sườn xuống:



Clr (Clear) là ngõ vào xóa của TFF; với Clr tác động mức thấp thì khi $Clr = 0$ thì ngõ ra Q của FF bị xóa về 0

Bộ đếm nối tiếp - Đếm lên (tt)

– Giải đồ thời gian:

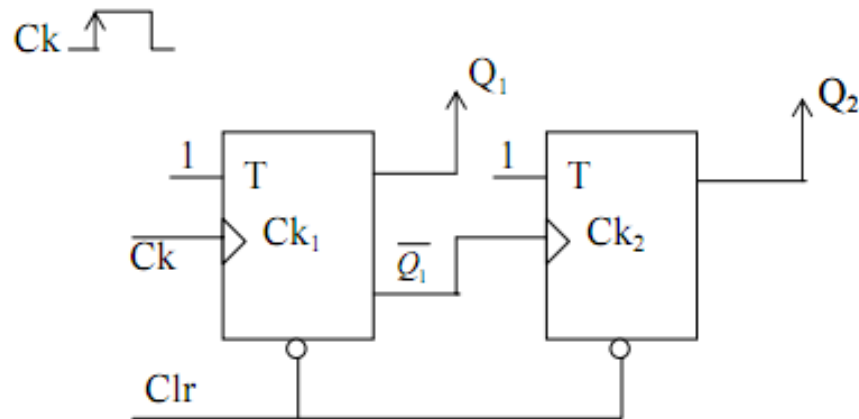


– Bảng trạng thái:

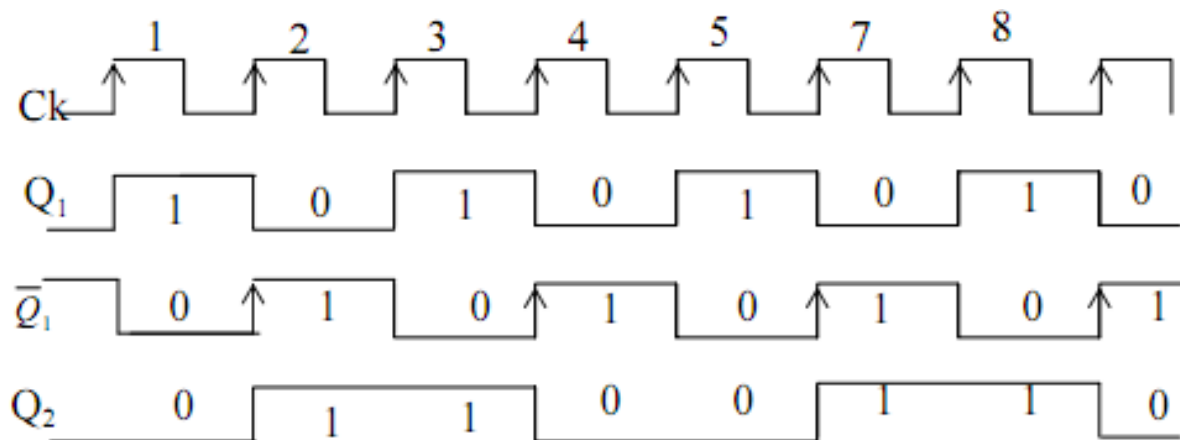
Xung vào	Trạng thái hiện tại		Trạng thái kế tiếp	
Ck	Q_2	Q_1	Q_2	Q_1
1	0	0	0	1
2	0	1	1	0
3	1	0	1	1
4	1	1	0	0

Đếm lên (tt)

- Trường hợp Ck tác động theo sườn lên:



- Giải đồ thời gian:



Bộ đếm nối tiếp - Đếm lên (tt)

– Bảng trạng thái:

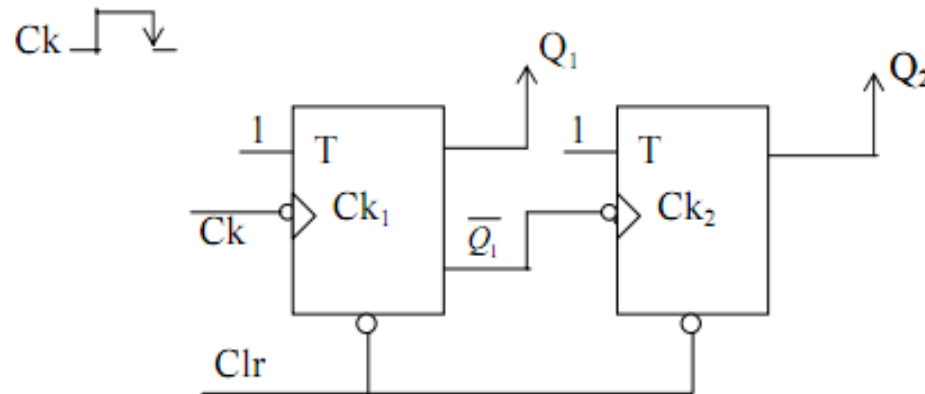
Xung vào	Trạng thái hiện tại		Trạng thái kế tiếp	
Ck	Q_2	Q_1	Q_2	Q_1
1	0	1	1	0
2	1	0	1	1
3	1	1	0	0
4	0	0	0	1

Bộ đếm nối tiếp - Đếm xuống

- Bộ đếm xuống có nội dung đếm giảm dần
- T/(J, K) luôn luôn ở mức logic 1 và ngõ ra của TFF/JKFF đứng trước nối với ngõ vào Ck của TFF/JKFF đứng sau
 - Ck tác động sườn xuống: TFF hoặc JKFF ghép với nhau theo quy luật: $Ck_{i+1} = \overline{Q_i}$
 - Ck tác động sườn lên: TFF hoặc JKFF ghép với nhau theo quy luật: $Ck_{i+1} = Q_i$

Bộ đếm nối tiếp - Đếm xuống (tt)

- Xét mạch đếm nối tiếp, đếm 4, đếm xuống, dùng TFF:
 - Trường hợp Ck tác động theo sườn xuống:

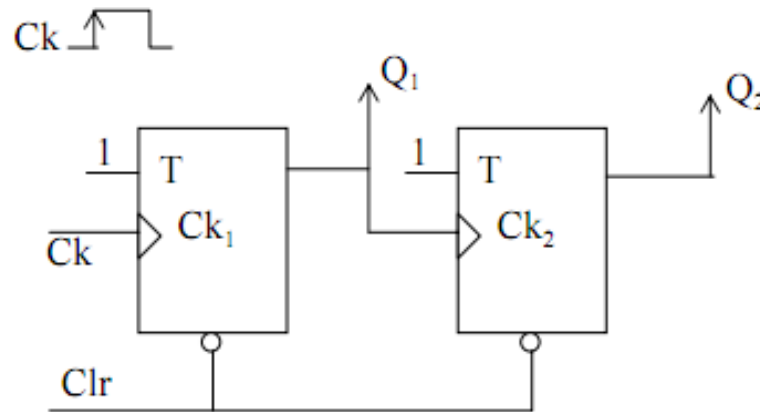


Bảng trạng thái:

Xung vào	Trạng thái hiện tại		Trạng thái kế tiếp	
Ck	Q ₂	Q ₁	Q ₂	Q ₁
1	0	0	1	1
2	1	1	1	0
3	1	0	0	1
4	0	1	0	0

Bộ đếm nối tiếp - Đếm xuống (tt)

- Trường hợp Ck tác động theo sườn lên:



Bảng trạng thái:

Xung vào	Trạng thái hiện tại		Trạng thái kế tiếp	
Ck	Q_2	Q_1	Q_2	Q_1
1	1	1	1	0
2	1	0	0	1
3	0	1	0	0
4	0	0	1	1

Bộ đếm nối tiếp - Đếm lên/xuống

- Bộ đếm lên/xuống vừa có thể đếm lên vừa có thể đếm xuống tùy thuộc vào tín hiệu điều khiển
- Gọi X là tín hiệu điều khiển, ta quy ước:
 - Nếu $X = 0$ thì đếm lên
 - Nếu $X = 1$ thì đếm xuống
- Trường hợp Ck tác động sườn xuống:
$$Ck_{i+1} = \bar{X}Q_i + X\bar{Q}_i = X \oplus Q_i$$
- Trường hợp Ck tác động sườn lên:
$$Ck_{i+1} = \bar{X}\bar{Q}_i + XQ_i = \overline{X \oplus Q_i}$$

Bộ đếm nối tiếp - Đếm modulo M

- Bộ đếm modulo M là bộ đếm nối tiếp, theo mã BCD 8421, có dung lượng đếm khác 2^n
- Xét mạch đếm 5, đếm lên, đếm nối tiếp: cần dùng 3 FF
 - Bảng trạng thái:

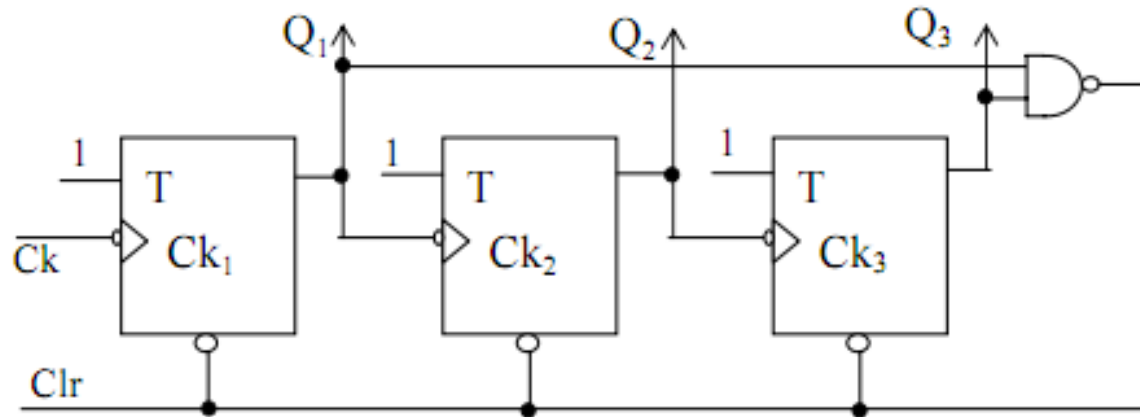
Xung vào	Trạng thái hiện tại			Trạng thái kế tiếp		
Ck	Q_3	Q_2	Q_1	Q_3	Q_2	Q_1
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	1
4	0	1	1	1	0	0
5	1	0	0	1/0	0	1/0

Bộ đếm nối tiếp - Đếm modulo M (tt)

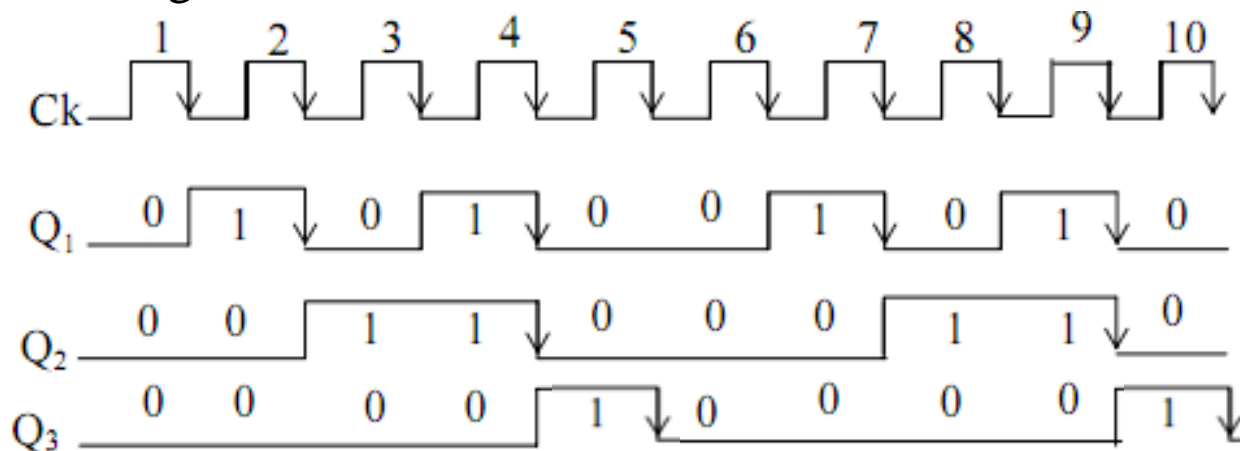
- Vấn đề đặt ra: Sau xung C_k thứ 5, ta tìm cách đưa tổ hợp 101 về 000 để mạch thực hiện đếm lại từ tổ hợp ban đầu
- Tổ hợp 101 có 2 ngõ ra Q_1 và Q_3 đồng thời bằng 1 (khác với các tổ hợp trước đó) \rightarrow dấu hiệu nhận biết để xóa bộ đếm về 000
- Để xóa bộ đếm về 000:
 - Đối với FF có ngõ vào Clr tác động mức 0 thì ta dùng cổng NAND 2 ngõ vào
 - Đối với FF có ngõ vào Clr tác động mức 1 thì ta dùng cổng AND 2 ngõ vào

Bộ đếm nối tiếp - Đếm modulo M (tt)

- Sơ đồ mạch:



- Giải đồ thời gian:



Bộ đếm song song

- Bộ đếm trong đó các FF mắc song song với nhau
- Các ngõ ra thay đổi trạng thái đồng thời dưới sự điều khiển của tín hiệu $Ck \rightarrow$ còn gọi là bộ đếm đồng bộ
- Sử dụng bất kỳ loại FF và theo bất kỳ quy luật đếm
- Không phụ thuộc tín hiệu Ck tác động sườn lên, sườn xuống, mức 0 hay mức 1
- Thiết kế dựa trên bảng đầu vào kích của FF

Bộ đếm song song (tt)

- Xét mạch đếm đồng bộ, đếm 5, đếm lên theo mã BCD 8421 dùng JKFF: cần dùng 3 JKFF
 - Bảng trạng thái:

Xung vào	Trạng thái hiện tại			Trạng thái kế tiếp		
Ck	Q_3	Q_2	Q_1	Q_3	Q_2	Q_1
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	1
4	0	1	1	1	0	0
5	1	0	0	0	0	0

Bộ đếm song song (tt)

- Bảng đầu vào kích:

Xung vào	Trạng thái hiện tại			Trạng thái kế tiếp								
	Q ₃	Q ₂	Q ₁	Q ₃	Q ₂	Q ₁	J ₃	K ₃	J ₂	K ₂	J ₁	K ₁
1	0	0	0	0	0	1	0	X	0	X	1	X
2	0	0	1	0	1	0	0	X	1	X	X	1
3	0	1	0	0	1	1	0	X	X	0	1	X
4	0	1	1	1	0	0	1	X	X	1	X	1
5	1	0	0	0	0	0	X	1	0	X	0	X

Bộ đếm song song (tt)

- Lập bảng Karnaugh để tối thiểu hóa:

J_1 \ Q_3Q_2		Q_1			
		00	01	11	10
0	1	1	x	0	
1	x	x	x	x	

$$J_1 = \overline{Q_3}$$

$K_1 \backslash Q_3 Q_2$		Q_1			
		00	01	11	10
0	x	x	x	x	
1	1	1	x	x	

$$K_1 = 1$$

$J_2 \backslash Q_3 Q_2$		Q_1			
		00	01	11	10
0	0	x	x	0	
1	1	x	x	x	

$$J_2 = Q_1$$

$K_2 \backslash Q_3 Q_2$		Q_1			
		00	01	11	10
0	x	0	x	0	
1	x	1	x	x	

$$K_2 = Q_1$$

Bộ đếm song song (tt)

- Lập bảng Karnaugh để tối thiểu hóa:

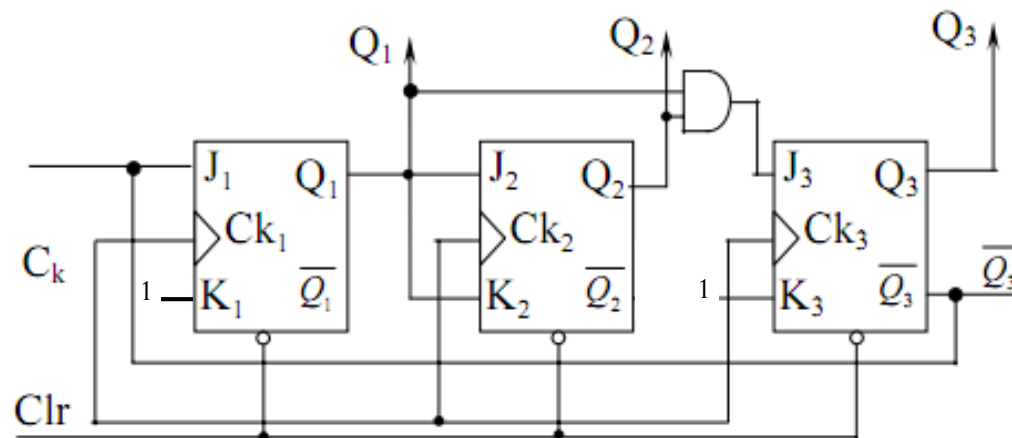
$J_3 \backslash Q_3 Q_2$		Q_1			
		00	01	11	10
Q_1	0	0	0	x	x
	1	0	1	x	x

$$J_3 = Q_1 Q_2$$

$K_3 \backslash Q_3 Q_2$		Q_1			
		00	01	11	10
Q_1	0	x	1	x	1
	1	x	1	x	x

$$K_3 = 1$$

- Sơ đồ logic:



Thanh ghi dịch chuyển

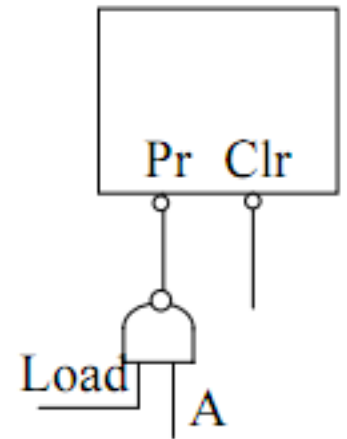
- Xây dựng dựa trên cơ sở các DFF hoặc các FF khác thực hiện chức năng của DFF
- Mỗi DFF lưu trữ 1 bit dữ liệu
- Để tạo thanh ghi nhiều bit, ghép các DFF lại với nhau theo quy luật:
 - Ngõ ra của DFF đứng trước được nối với ngõ vào dữ liệu của DFF sau ($D_{i+1} = Q_i$): thanh ghi có khả năng dịch phải
 - Ngõ ra của DFF đứng sau được nối với ngõ vào dữ liệu của DFF trước ($D_i = Q_{i+1}$): thanh ghi có khả năng dịch trái

Thanh ghi dịch chuyển (tt)

- Phân loại:
 - Theo hướng dịch chuyển dữ liệu:
 - Dịch trái
 - Dịch phải
 - Vừa dịch phải vừa dịch trái
 - Theo ngõ vào dữ liệu:
 - Ngõ vào dữ liệu nối tiếp
 - Ngõ vào dữ liệu song song
 - Theo ngõ ra:
 - Ngõ ra nối tiếp
 - Ngõ ra song song
 - Ngõ ra vừa nối tiếp vừa song song

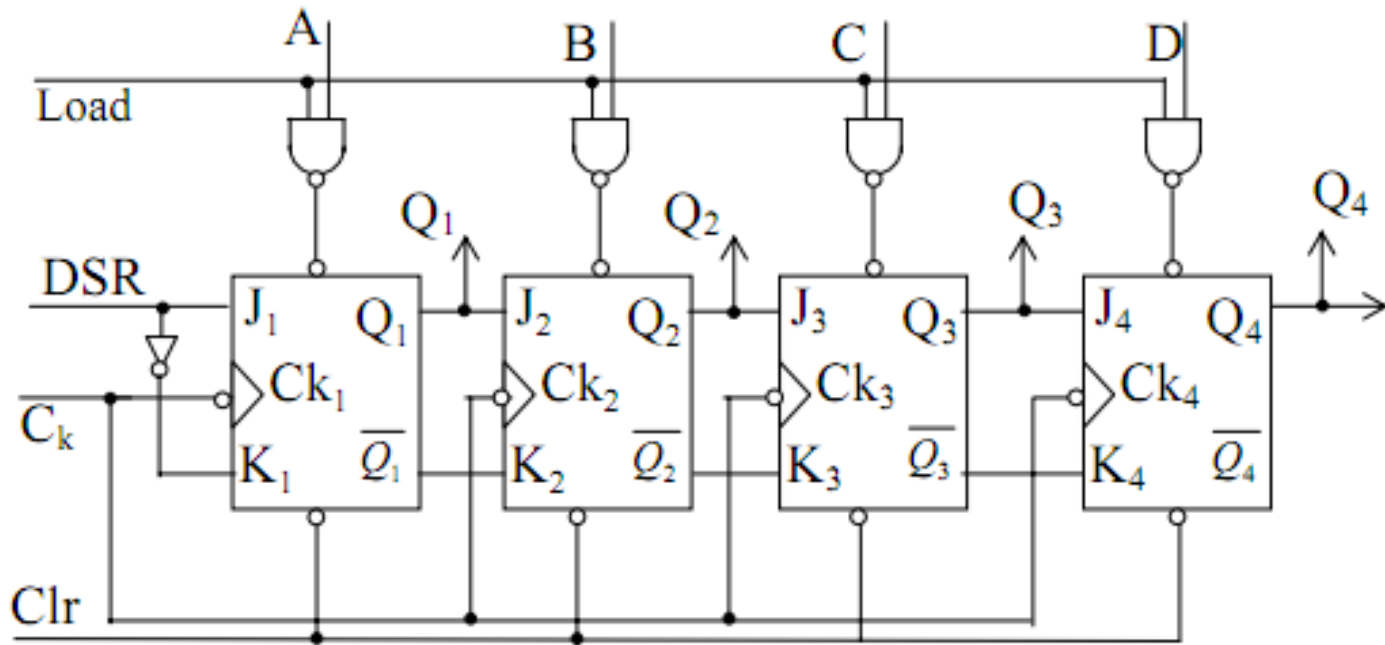
Thanh ghi dịch chuyển (tt)

- Nhập dữ liệu vào FF:
 - Dữ liệu được nhập vào FF bằng chân Preset (Pr)
 - Khi Load = 0: Pr = Clr = 1 (Chân Clr để trống ứng với mức logic 1)
 - FF tự do, dữ liệu không được nhập vào FF
 - Khi Load = 1: Pr = \bar{A}
 - Giả sử ban đầu: $Q = 0$
 - Nếu $A = 0 \rightarrow \text{Pr} = 1, \text{Clr} = 1 \rightarrow Q = Q^0 = 0$
 - Nếu $A = 1 \rightarrow \text{Pr} = 0, \text{Clr} = 1 \rightarrow Q = 1$
 - Vậy $Q = A$, dữ liệu được nhập vào FF
 - **Chú ý:** phương pháp này đòi hỏi trước khi nhập phải xóa FF về 0



Thanh ghi dịch chuyển (tt)

- Xét thanh ghi 4 bit dịch phải dùng JKFF:



Trong đó:

- DSR (Data Shift Right): ngõ vào dữ liệu nối tiếp
- Q_1, Q_2, Q_3, Q_4 : các ngõ ra song song

Thanh ghi dịch chuyển (tt)

- Hoạt động của mạch:
 - Giả sử ban đầu Load = 1 \rightarrow A, B, C, D được nhập vào thanh ghi dịch
 $Q_1 = A, Q_2 = B, Q_3 = C, Q_4 = D$
 - Xét FF_1 : $D = DSR_1, Q_1 = A$
 - Nếu $DSR_1 = 0 \rightarrow Q = 0$
 - Nếu $DSR_1 = 1 \rightarrow Q = 1$
 - Như vậy sau một xung Ck tác động sườn xuống thì $Q_1 = DSR_1$
 - Lúc đó tại FF_2, FF_3, FF_3 : $Q_2 = A, Q_3 = B, Q_4 = C$
 - Như vậy, sau khi Ck tác động sườn xuống, nội dung trong thanh ghi được dời sang phải 1 bit
 - Sau 4 xung, dữ liệu trong thanh ghi được xuất ra ngoài và nội dung DFF được thay bằng dữ liệu từ ngõ vào $DSR_1, DSR_2, DSR_3, DSR_4$

Thanh ghi dịch chuyển (tt)

- Bảng trạng thái :

Xung vào	Trạng thái hiện tại				Trạng thái kế			
	Q_1	Q_2	Q_3	Q_4	Q_1	Q_2	Q_3	Q_4
1	A	B	C	D	DSR_1	A	B	C
2	DSR_1	A	B	C	DSR_2	DSR_1	A	B
3	DSR_2	DSR_1	A	B	DSR_3	DSR_2	DSR_1	A
4	DSR_3	DSR_2	DSR_1	A	DSR_4	DSR_3	DSR_2	DSR_1

Thanh ghi dịch chuyển (tt)

- Trường hợp ngõ ra $\overline{Q_4}$ bằng ngõ vào dữ liệu nối tiếp DSR:

Bảng trạng thái:

Xung vào	Trạng thái hiện tại				Trạng thái kế			
	Q_1	Q_2	Q_3	Q_4	Q_1	Q_2	Q_3	Q_4
1	0	0	0	0	1	0	0	0
2	1	0	0	0	1	1	0	0
3	1	1	0	0	1	1	1	0
4	1	1	1	0	1	1	1	1
5	1	1	1	1	0	1	1	1
6	0	1	1	1	0	0	1	1
7	0	0	1	1	0	0	0	1
8	0	0	0	1	0	0	0	0

Bộ nhớ

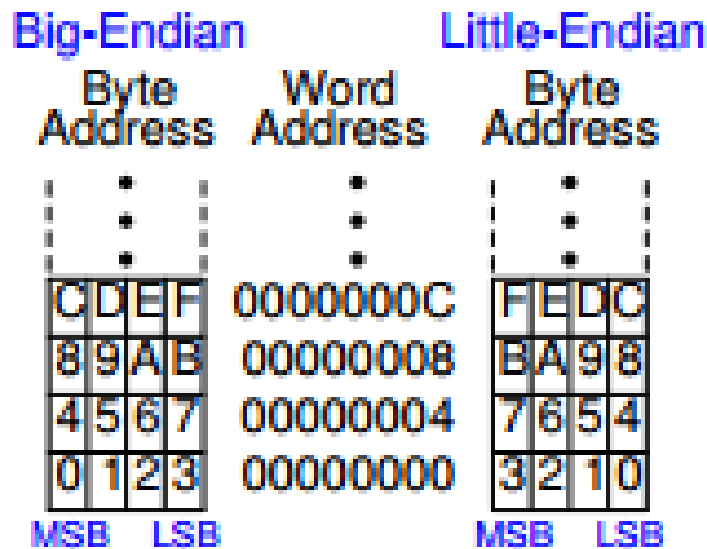
- Tế bào nhớ: thiết bị hay mạch điện tử dùng để lưu trữ 1 bit
- Từ nhớ (word): nhóm các bit ở trong một bộ nhớ, kích thước từ 8 – 64 bit
- Dung lượng bộ nhớ: khả năng lưu trữ của bộ nhớ (Kilo, Mega, Giga, Tera)
- Địa chỉ: dùng để xác định các vùng của các từ trong bộ nhớ, một bộ nhớ gồm 2^n từ thì cần n đường địa chỉ
 - Địa chỉ theo từ (word):
 - Địa chỉ theo byte:

Word Address	Data	
⋮	⋮	⋮
00000003	14FD02FF	Word 3
00000002	01010101	Word 2
00000001	1FE0A02C	Word 1
00000000	FFFFFFFF	Word 0

Word Address	Data	
⋮	⋮	⋮
0000000C	14FD02FF	Word 3
00000008	01010101	Word 2
00000004	1FE0A02C	Word 1
00000000	FFFFFFFF	Word 0

Bộ nhớ (tt)

- Địa chỉ byte kiểu Big endian và Little endian:
 - Big endian: Byte có trọng số lớn nhất (MSB) trong word lưu ở địa chỉ thấp nhất
 - Little endian: Byte có trọng số nhỏ nhất (LSB) trong word lưu ở địa chỉ thấp nhất



Bộ nhớ (tt)

- Hoạt động đọc:
 - Đọc là xuất dữ liệu từ bộ nhớ ra ngoài
 - Để đọc nội dung một ô nhớ cần thực hiện:
 - Đưa địa chỉ cần đọc vào các đường địa chỉ
 - Tích cực tín hiệu điều khiển đọc
- Hoạt động ghi:
 - Ghi là lưu dữ liệu từ bên ngoài vào bên trong bộ nhớ
 - Để ghi dữ liệu vào một ô nhớ cần thực hiện:
 - Đưa địa chỉ cần ghi vào các đường địa chỉ
 - Đưa dữ liệu cần ghi vào các đường dữ liệu
 - Tích cực tín hiệu điều khiển ghi

Bộ nhớ (tt)

- Phân loại:
 - ROM (Read Only Memory):
 - Bộ nhớ chỉ đọc, không cho phép ghi
 - Dữ liệu không mất khi mất điện áp cung cấp
 - MROM (Mask ROM): được lập trình bởi nhà sản xuất trong quá trình sản xuất, không phục hồi được nếu chương trình bị sai hỏng
 - PROM (Programmable ROM): cho phép lập trình bằng thiết bị đặc biệt và không thể xóa được
 - EPROM (Erasable Programmable ROM): có thể xóa (bằng tia cực tím) và lập trình lại
 - EEPROM (Electrically Erasable Programmable ROM): có thể xóa bằng điện và lập trình lại

Bộ nhớ (tt)

- RAM (Random Access Memory)
 - Bộ nhớ truy xuất ngẫu nhiên, đọc viết tùy ý
 - Dữ liệu khi mất điện áp cung cấp
- SRAM (Static RAM)
 - Lưu trữ dữ liệu cho đến khi mất điện áp cung cấp mà không cần làm tươi dữ liệu
- DRAM (Dynamic RAM)
 - Có thể mất dữ liệu khi điện áp cung cấp chưa bị mất, nên cần có cơ chế làm tươi dữ liệu
 - Tốc độ truy xuất chậm hơn SRAM nhưng giá thành thấp hơn