

KỸ THUẬT SỐ

ThS. Phạm Thị Đan Ngọc
Khoa Kỹ Thuật Điện Tử 2
Email: ngocptd@ptithcm.edu.vn

Ngày 17 tháng 11 năm 2014

Chương 4: Mạch tuần tự

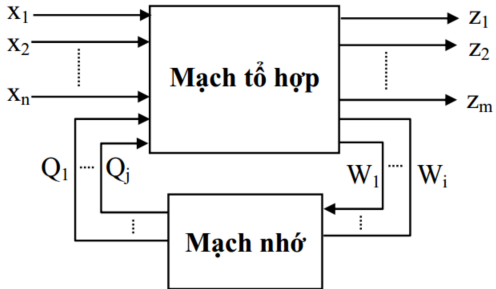
- 1 Giới thiệu
- 2 Phần tử nhớ của mạch tuần tự
- 3 Phương pháp mô tả mạch tuần tự
- 4 Phân tích và thiết kế mạch tuần tự
- 5 Mạch tuần tự thông dụng

4.1 Giới thiệu

4.1 Giới thiệu

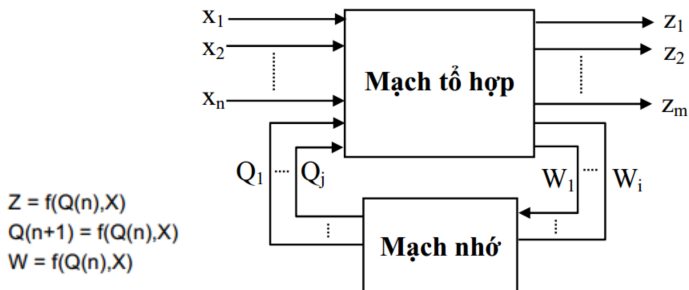
- Mô hình toán học:

$$\begin{aligned}
 Z &= f(Q(n), X) \\
 Q(n+1) &= f(Q(n), X) \\
 W &= f(Q(n), X)
 \end{aligned}$$



4.1 Giới thiệu

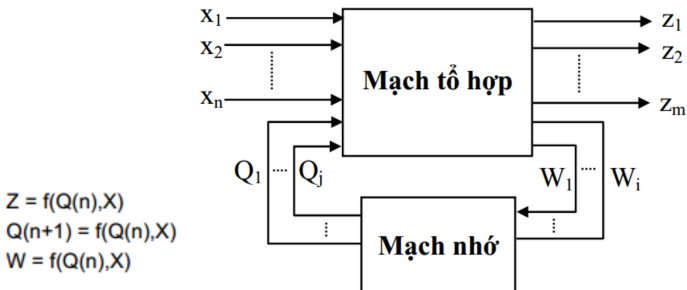
- Mô hình toán học:



- Đặc điểm: Trạng thái hoạt động của mạch điện không chỉ phụ thuộc vào các lối vào mà còn phụ thuộc vào trạng thái bên trong trước đó của mạch.

4.1 Giới thiệu

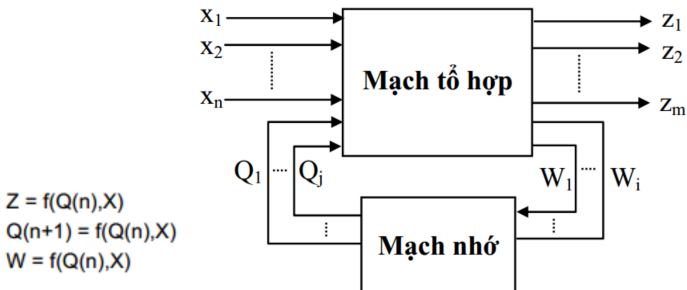
- Mô hình toán học:



- Đặc điểm: Trạng thái hoạt động của mạch điện không chỉ phụ thuộc vào các lối vào mà còn phụ thuộc vào trạng thái bên trong trước đó của mạch.
- Mạch tuần tự đồng bộ.

4.1 Giới thiệu

- Mô hình toán học:



- Đặc điểm: Trạng thái hoạt động của mạch điện không chỉ phụ thuộc vào các lối vào mà còn phụ thuộc vào trạng thái bên trong trước đó của mạch.
- Mạch tuần tự đồng bộ.
- Mạch tuần tự không đồng bộ.

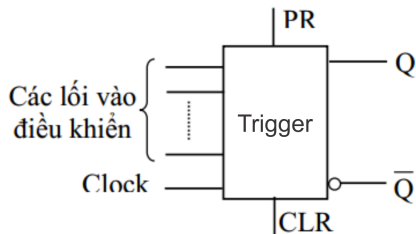
4.2 Trigger (Flip Flop)

4.2 Trigger (Flip Flop)

- Sơ đồ tổng quát

4.2 Trigger (Flip Flop)

- Sơ đồ tổng quát



4.2 Trigger

4.2 Trigger

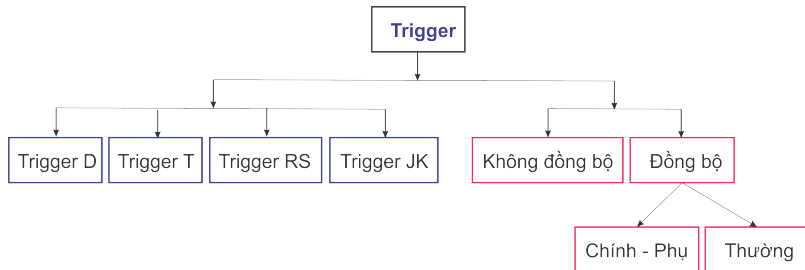
- Đây là phần tử cơ bản, phần tử này có thể nhớ một trong hai trạng thái.

4.2 Trigger

- Đây là phần tử cơ bản, phần tử này có thể nhớ một trong hai trạng thái.
- Trigger luôn có hai trạng thái ngược nhau.

4.2 Trigger

- Đây là phần tử cơ bản, phần tử này có thể nhớ một trong hai trạng thái.
- Trigger luôn có hai trạng thái ngược nhau.
- Có hai ngõ vào điều khiển, nó làm việc theo hai chế độ: đồng bộ và bất đồng bộ.



4.2 Trigger

4.2 Trigger

- Trigger không đồng bộ: ngõ ra của trigger thay đổi chỉ phụ thuộc tín hiệu ngõ vào.

4.2 Trigger

- Trigger không đồng bộ: ngõ ra của trigger thay đổi chỉ phụ thuộc tín hiệu ngõ vào.
- Trigger đồng bộ: ngõ ra của trigger thay đổi phụ thuộc tín hiệu ngõ vào và tín hiệu đồng bộ.

4.2 Trigger

- Trigger không đồng bộ: ngõ ra của trigger thay đổi chỉ phụ thuộc tín hiệu ngõ vào.
- Trigger đồng bộ: ngõ ra của trigger thay đổi phụ thuộc tín hiệu ngõ vào và tín hiệu đồng bộ.
- *Đồng bộ theo mức*: đồng bộ mức cao hoặc mức thấp.

4.2 Trigger

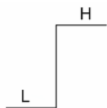
- Trigger không đồng bộ: ngõ ra của trigger thay đổi chỉ phụ thuộc tín hiệu ngõ vào.
- Trigger đồng bộ: ngõ ra của trigger thay đổi phụ thuộc tín hiệu ngõ vào và tín hiệu đồng bộ.
- *Đồng bộ theo mức*: đồng bộ mức cao hoặc mức thấp.
- *Đồng bộ theo sườn*: đồng bộ theo sườn dương hoặc sườn âm.

4.2 Trigger

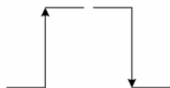
- Trigger không đồng bộ: ngõ ra của trigger thay đổi chỉ phụ thuộc tín hiệu ngõ vào.
- Trigger đồng bộ: ngõ ra của trigger thay đổi phụ thuộc tín hiệu ngõ vào và tín hiệu đồng bộ.
- *Đồng bộ theo mức*: đồng bộ mức cao hoặc mức thấp.
- *Đồng bộ theo sườn*: đồng bộ theo sườn dương hoặc sườn âm.
- *Đồng bộ theo kiểu xung*.

4.2 Trigger

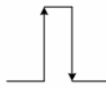
- Trigger không đồng bộ: ngõ ra của trigger thay đổi chỉ phụ thuộc tín hiệu ngõ vào.
- Trigger đồng bộ: ngõ ra của trigger thay đổi phụ thuộc tín hiệu ngõ vào và tín hiệu đồng bộ.
- *Đồng bộ theo mức*: đồng bộ mức cao hoặc mức thấp.
- *Đồng bộ theo sườn*: đồng bộ theo sườn dương hoặc sườn âm.
- *Đồng bộ theo kiểu xung*.



Đồng bộ theo mức



Đồng bộ theo sườn

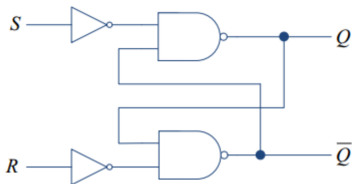


Đồng bộ theo xung

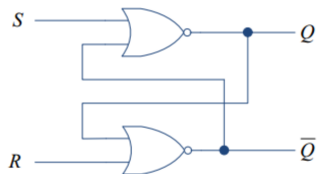
4.2.1 S-R flip-flop (Set-Reset flip-flop)

4.2.1 S-R flip-flop (Set-Reset flip-flop)

- FF cơ bản



a) SR flip flop sử dụng cổng NAND



b) SR flip flop sử dụng cổng NOR

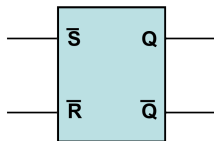
4.2.1 S-R flip-flop (Set-Reset flip-flop)

4.2.1 S-R flip-flop (Set-Reset flip-flop)

- SR FF sử dụng cổng NAND.

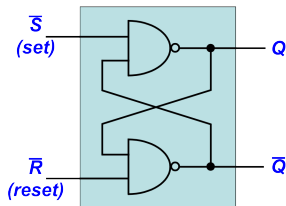
4.2.1 S-R flip-flop (Set-Reset flip-flop)

- SR FF sử dụng cổng NAND.



Bảng sự thật

S	R	Q	\overline{Q}	Mod hoạt động
0	0	1	1	Cắm
0	1	1	0	
1	0	0	1	
1	1	Q_0	$\overline{Q_0}$	Nhớ



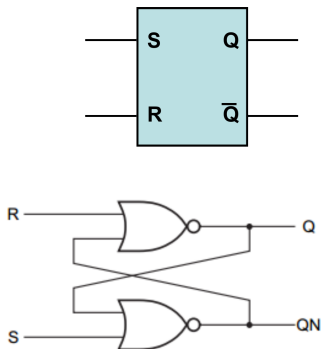
4.2.1 S-R flip-flop (Set-Reset flip-flop)

4.2.1 S-R flip-flop (Set-Reset flip-flop)

- SR FF sử dụng cổng NOR

4.2.1 S-R flip-flop (Set-Reset flip-flop)

- SR FF sử dụng cổng NOR



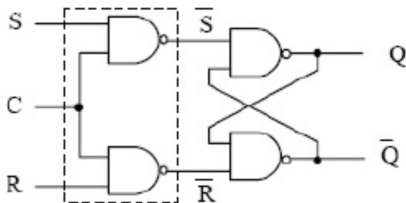
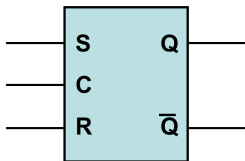
Bảng sự thật

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

Cấm

4.2.1 S-R flip-flop chịu tác động xung đồng hồ

4.2.1 S-R flip-flop chịu tác động xung đồng hồ



Bảng sự thật

C	S	R	Q	\overline{Q}	Mod hoạt động
0	0	0	1	1	Không đổi
0	0	1	1	0	Không đổi
0	1	0	0	1	Không đổi
0	1	1	Q_0	$\overline{Q_0}$	Không đổi
1	0	0	Q_0	$\overline{Q_0}$	Không đổi
1	0	1	0	1	Thiết lập
1	1	0	1	0	Xóa
1	1	1	1	1	Cắm

4.2.1 S-R flip-flop chịu tác động xung đồng hồ

4.2.1 S-R flip-flop chịu tác động xung đồng hồ

- Khảo sát giản đồ xung như sau:

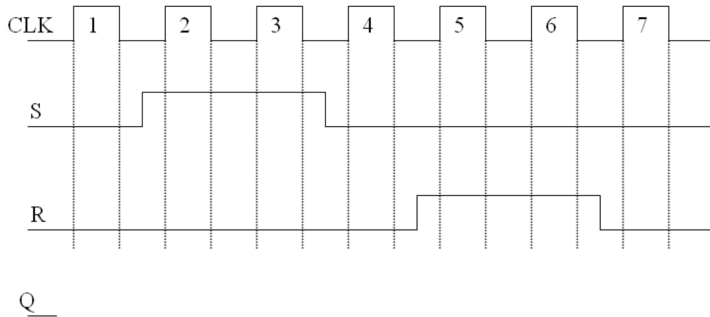
4.2.1 S-R flip-flop chịu tác động xung đồng hồ

4.2.1 S-R flip-flop chịu tác động xung đồng hồ

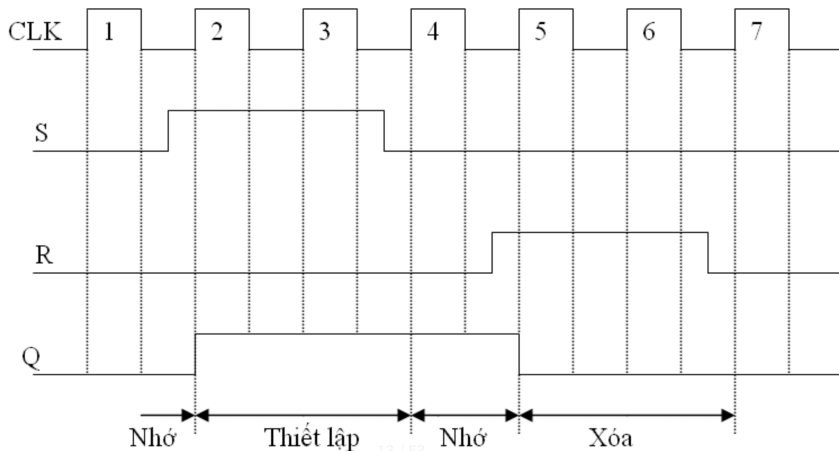
- Ví dụ: Vẽ đồ thị tín hiệu ngõ ra Q, biết rằng SR flip flop đồng bộ mức cao và các tín hiệu R, S được biểu diễn như hình bên dưới.

4.2.1 S-R flip-flop chịu tác động xung đồng hồ

- Ví dụ: Vẽ đồ thị tín hiệu ngõ ra Q, biết rằng SR flip flop đồng bộ mức cao và các tín hiệu R, S được biểu diễn như hình bên dưới.



4.2.1 S-R flip-flop chịu tác động xung đồng hồ



4.2.2 D Flip Flop - Delay

4.2.2 D Flip Flop - Delay

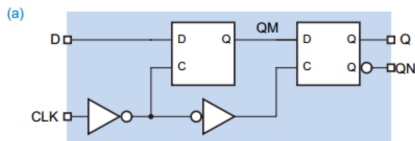
- D FF hoạt động ở hai chế độ: đồng bộ và bất đồng bộ.

4.2.2 D Flip Flop - Delay

- D FF hoạt động ở hai chế độ: đồng bộ và bất đồng bộ.
- D FF là một trường hợp đặc biệt của RSFF, chịu tác động của xung đồng hồ.

4.2.2 D Flip Flop - Delay

- D FF hoạt động ở hai chế độ: đồng bộ và bất đồng bộ.
- D FF là một trường hợp đặc biệt của RSFF, chịu tác động của xung đồng hồ.

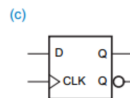


Thiết kế mạch dạng chốt D

(b)

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

Bảng chức năng



Hàm logic

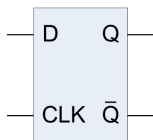
4.2.2 D Flip Flop - Delay

4.2.2 D Flip Flop - Delay

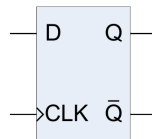
- D FF hoạt động theo chế độ đồng bộ.

4.2.2 D Flip Flop - Delay

- D FF hoạt động theo chế độ đồng bộ.



Đồng bộ theo mức



Đồng bộ theo sườn

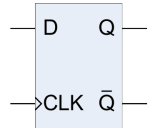
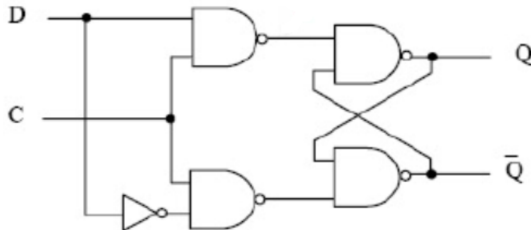
4.2.2 D Flip Flop - Delay

4.2.2 D Flip Flop - Delay

- D FF đồng bộ theo sườn.

4.2.2 D Flip Flop - Delay



- D FF đồng bộ theo sườn.



4.2.2 D Flip Flop - Delay

4.2.2 D Flip Flop - Delay

- Bảng sự thật

Ck	D	Q	\overline{Q}
	0	0	1
	1	1	0
0	x	Q_0	$\overline{Q_0}$

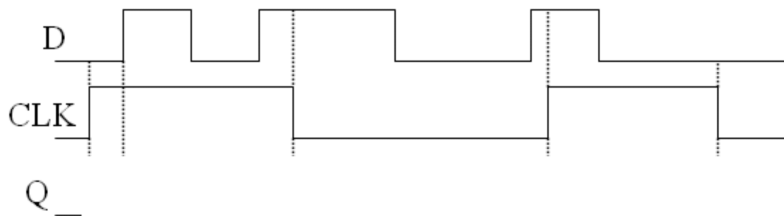
4.2.2 D Flip Flop - Delay

4.2.2 D Flip Flop - Delay

- Vẽ giản đồ của tín hiệu ngõ ra Q khi cho dạng tín hiệu ngõ vào D và xung clock tích cực mức cao.

4.2.2 D Flip Flop - Delay

- Vẽ giản đồ của tín hiệu ngõ ra Q khi cho dạng tín hiệu ngõ vào D và xung clock tích cực mức cao.



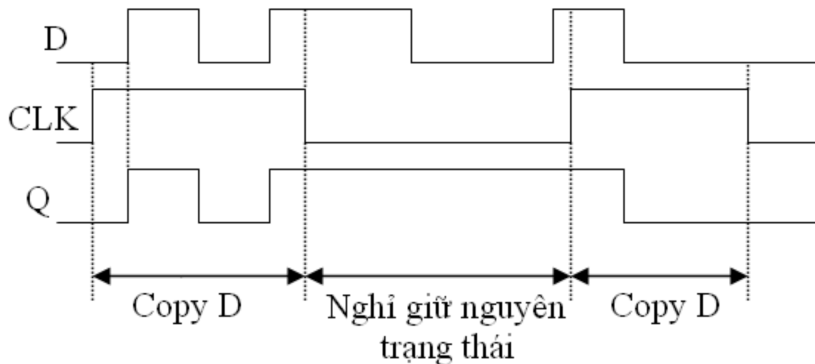
4.2.2 D Flip Flop - Delay

4.2.2 D Flip Flop - Delay

- Vẽ giản đồ của tín hiệu ngõ ra Q khi cho dạng tín hiệu ngõ vào D và xung clock tích cực mức cao.

4.2.2 D Flip Flop - Delay

- Vẽ giản đồ của tín hiệu ngõ ra Q khi cho dạng tín hiệu ngõ vào D và xung clock tích cực mức cao.



4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

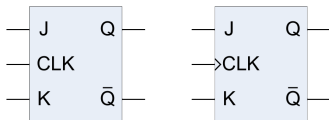
- Khắc phục nhược điểm của SRFF.

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- Khắc phục nhược điểm của SRFF.
- JK FF chỉ hoạt động ở chế độ đồng bộ.

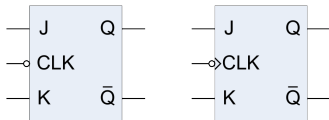
4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- Khắc phục nhược điểm của SRFF.
- JK FF chỉ hoạt động ở chế độ đồng bộ.



Tích cực mức cao

Tích cực sườn dương



Tích cực mức thấp

Tích cực sườn âm

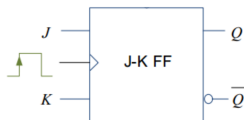
4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- Hàm logic và bảng sự thật.

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- Hàm logic và bảng sự thật.



Ck	J	K	Q	\overline{Q}
	0	0	Q_0	$\overline{Q_0}$
	0	1	0	1
	1	0	1	0
	1	1	$\overline{Q_0}$	Q_0





4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- Bảng kích thích của JK FF.

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- Bảng kích thích của JK FF.

Ck	Q	Q ₀	J	K
	0	0	0	x
	0	1	1	x
	1	0	x	1
	1	1	x	0

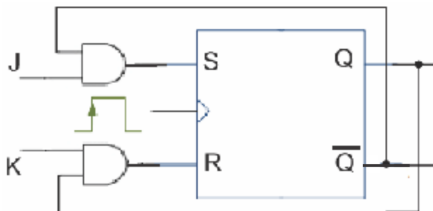
4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- JK FF hồi tiếp 2 ngõ ra về hai ngõ vào S và R qua cổng AND.

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- JK FF hồi tiếp 2 ngõ ra về hai ngõ vào S và R qua cổng AND.



$$S = J \cdot \overline{Q}$$

$$R = k \cdot Q$$

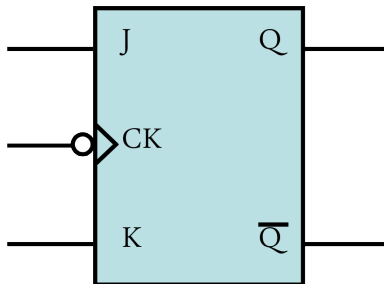
4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- Sự hồi tiếp từ ngõ ra đưa về ngõ vào của JK FF làm cho hai ngõ ra bị dao động, dẫn đến sự không ổn định. Do đó, để tránh hiện tượng này, người ta chế tạo ra JK FF chịu tác động cạnh xuống của xung đồng hồ.

4.2.3 JK Flip Flop - Jack Kilby Flip-Flop

- Sự hồi tiếp từ ngõ ra đưa về ngõ vào của JK FF làm cho hai ngõ ra bị dao động, dẫn đến sự không ổn định. Do đó, để tránh hiện tượng này, người ta chế tạo ra JK FF chịu tác động cạnh xuống của xung đồng hồ.



4.2.4 JKFF và DFF với hai ngõ vào trực tiếp

4.2.4 JKFF và DFF với hai ngõ vào trực tiếp

- PR và CL là hai ngõ vào trực tiếp có nhiệm vụ đưa ngõ ra Q lên mức cao hoặc về mức thấp.

4.2.4 JKFF và DFF với hai ngõ vào trực tiếp

- PR và CL là hai ngõ vào trực tiếp có nhiệm vụ đưa ngõ ra Q lên mức cao hoặc về mức thấp.
- *Khi $PR = 0$ và $CL = 1$:* lúc này bất chấp trạng thái ngõ vào D (JK), bất chấp xung lệnh ck mà ngõ ra Q sẽ ở mức cao.

4.2.4 JKFF và DFF với hai ngõ vào trực tiếp

- PR và CL là hai ngõ vào trực tiếp có nhiệm vụ đưa ngõ ra Q lên mức cao hoặc về mức thấp.
- *Khi $PR = 0$ và $CL = 1$* : lúc này bất chấp trạng thái ngõ vào D (JK), bất chấp xung lệnh ck mà ngõ ra Q sẽ ở mức cao.
- *Khi $PR = 1$ và $CL = 0$* : lúc này bất chấp trạng thái ngõ vào D (JK), bất chấp xung lệnh ck mà ngõ ra Q sẽ ở mức thấp.

4.2.4 JKFF và DFF với hai ngõ vào trực tiếp

- PR và CL là hai ngõ vào trực tiếp có nhiệm vụ đưa ngõ ra Q lên mức cao hoặc về mức thấp.
- *Khi $PR = 0$ và $CL = 1$* : lúc này bất chấp trạng thái ngõ vào D (JK), bất chấp xung lệnh ck mà ngõ ra Q sẽ ở mức cao.
- *Khi $PR = 1$ và $CL = 0$* : lúc này bất chấp trạng thái ngõ vào D (JK), bất chấp xung lệnh ck mà ngõ ra Q sẽ ở mức thấp.
- *Khi $PR = 1$ và $CL = 1$* : lúc này ngõ ra Q sẽ phụ thuộc vào trạng thái ngõ vào D (JK) và xung lệnh ck.



4.2.4 JKFF và DFF với hai ngõ vào trực tiếp

- PR và CL là hai ngõ vào trực tiếp có nhiệm vụ đưa ngõ ra Q lên mức cao hoặc về mức thấp.
- *Khi $PR = 0$ và $CL = 1$:* lúc này bất chấp trạng thái ngõ vào D (JK), bất chấp xung lệnh ck mà ngõ ra Q sẽ ở mức cao.
- *Khi $PR = 1$ và $CL = 0$:* lúc này bất chấp trạng thái ngõ vào D (JK), bất chấp xung lệnh ck mà ngõ ra Q sẽ ở mức thấp.
- *Khi $PR = 1$ và $CL = 1$:* lúc này ngõ ra Q sẽ phụ thuộc vào trạng thái ngõ vào D (JK) và xung lệnh ck.





4.2.4 JKFF và DFF với hai ngõ vào trực tiếp

4.2.4 JKFF và DFF với hai ngõ vào trực tiếp

D FF

PR	CL	Ck	D	Q	\overline{Q}
0	1	x	x	1	0
1	0	x	x	0	1
1	1		0	0	1
1	1		1	1	0

JK FF

PR	CL	Ck	J	K	Q	\overline{Q}
0	1	x	x	x	1	0
1	0	x	x	x	0	1
	1		0	0	Q_0	$\overline{Q_0}$
	1		0	1	0	1
	1		1	0	1	0
	1		1	1	$\overline{Q_0}$	Q_0

4.2.4 Bảng kích thích của các Flip-Flop

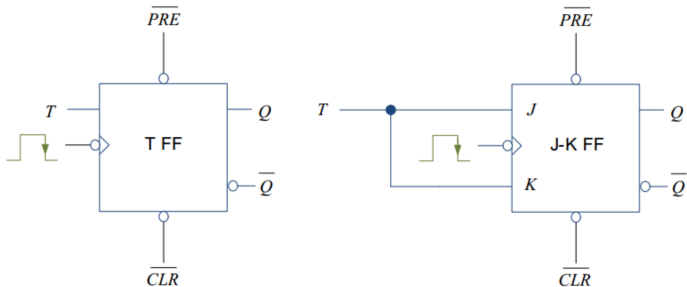
4.2.4 Bảng kích thích của các Flip-Flop

Bảng kích thích các loại FF

Q_n	Q_{n+1}	S	R	D	J	K	T
0	0	0	X	0	0	X	0
0	1	1	0	1	1	X	1
1	0	0	1	0	X	1	1
1	1	X	0	1	X	0	0


4.2.5 T FF

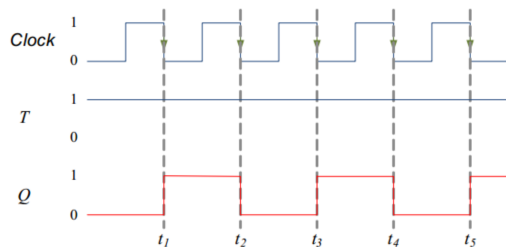
4.2.5 T FF



4.2.5 T FF

4.2.5 T FF

Ck	T	Q	\overline{Q}
	0	Q_0	$\overline{Q_0}$
	1	$\overline{Q_0}$	Q_0



4.2.5 T FF

4.2.5 T FF

- Nhận xét:

4.2.5 T FF

- Nhận xét:
 - Khi ngõ vào $T = 1$ thì ngõ ra Q sẽ đảo lại trạng thái trước đó của nó mỗi khi gặp cạnh xuống của xung đồng hồ. Do đó, dạng sóng của ngõ ra Q sẽ có tần số bằng $1/2$ dạng sóng ngõ vào của xung lệnh ck .

4.2.5 T FF

- Nhận xét:
 - Khi ngõ vào $T = 1$ thì ngõ ra Q sẽ đảo lại trạng thái trước đó của nó mỗi khi gặp cạnh xuống của xung đồng hồ. Do đó, dạng sóng của ngõ ra Q sẽ có tần số bằng $1/2$ dạng sóng ngõ vào của xung lệnh ck .
 - Dựa vào đặc điểm chia đôi tần số của T FF, người ta thiết lập mạch đếm nhị phân.

4.2.6 MS FF - Master Slave

4.2.6 MS FF - Master Slave

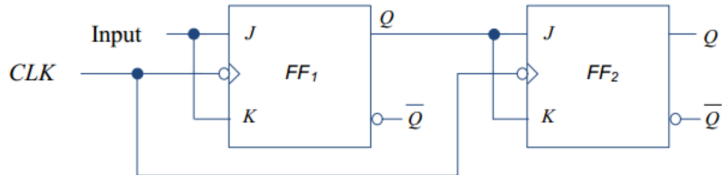
- Do hiện tượng trễ trong quá trình truyền đạt và các FF làm việc ở tần số cao thì các ngõ ra không thay đổi kịp theo xung nhịp. Điều này có thể dẫn đến sự chuyển đổi trạng thái sai.

4.2.6 MS FF - Master Slave

- Do hiện tượng trễ trong quá trình truyền đạt và các FF làm việc ở tần số cao thì các ngõ ra không thay đổi kịp theo xung nhịp. Điều này có thể dẫn đến sự chuyển đổi trạng thái sai.
- FF chủ tố được sử dụng nhằm cách ly dữ liệu vào với dữ liệu ngõ ra.

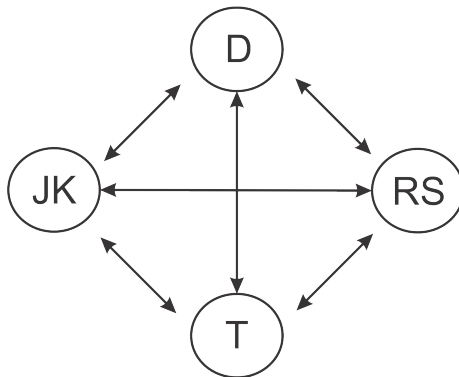
4.2.6 MS FF - Master Slave

4.2.6 MS FF - Master Slave



4.2.7 Sự chuyển đổi giữa các loại FF

4.2.7 Sự chuyển đổi giữa các loại FF



4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

- Cách 1: Dùng phương trình đặc trưng.

4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

- Cách 1: Dùng phương trình đặc trưng.
 - Sử dụng phép biến đổi logic đưa phương trình đặc trưng của FF đã cho về dạng phương trình đặc trưng của FF cần tìm.

4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

- Cách 1: Dùng phương trình đặc trưng.
 - Sử dụng phép biến đổi logic đưa phương trình đặc trưng của FF đã cho về dạng phương trình đặc trưng của FF cần tìm.
 - Từ phương trình đặc trưng, viết hàm kích thích của FF cần tìm.

4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

- Cách 1: Dùng phương trình đặc trưng.
 - Sử dụng phép biến đổi logic đưa phương trình đặc trưng của FF đã cho về dạng phương trình đặc trưng của FF cần tìm.
 - Từ phương trình đặc trưng, viết hàm kích thích của FF cần tìm.
 - Vẽ sơ đồ tương ứng.

4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

- Cách 1: Dùng phương trình đặc trưng.
 - Sử dụng phép biến đổi logic đưa phương trình đặc trưng của FF đã cho về dạng phương trình đặc trưng của FF cần tìm.
 - Từ phương trình đặc trưng, viết hàm kích thích của FF cần tìm.
 - Vẽ sơ đồ tương ứng.
- Cách 2: Dùng bảng hàm kích thích:

4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

- Cách 1: Dùng phương trình đặc trưng.
 - Sử dụng phép biến đổi logic đưa phương trình đặc trưng của FF đã cho về dạng phương trình đặc trưng của FF cần tìm.
 - Từ phương trình đặc trưng, viết hàm kích thích của FF cần tìm.
 - Vẽ sơ đồ tương ứng.
- Cách 2: Dùng bảng hàm kích thích:
 - Xác định hệ hàm theo bảng kích thích của FF.

4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

- Cách 1: Dùng phương trình đặc trưng.
 - Sử dụng phép biến đổi logic đưa phương trình đặc trưng của FF đã cho về dạng phương trình đặc trưng của FF cần tìm.
 - Từ phương trình đặc trưng, viết hàm kích thích của FF cần tìm.
 - Vẽ sơ đồ tương ứng.
- Cách 2: Dùng bảng hàm kích thích:
 - Xác định hệ hàm theo bảng kích thích của FF.
 - Tối thiểu hóa hàm Y.

4.2.7 Sự chuyển đổi giữa các loại FF

Phương pháp:

- Cách 1: Dùng phương trình đặc trưng.
 - Sử dụng phép biến đổi logic đưa phương trình đặc trưng của FF đã cho về dạng phương trình đặc trưng của FF cần tìm.
 - Từ phương trình đặc trưng, viết hàm kích thích của FF cần tìm.
 - Vẽ sơ đồ tương ứng.
- Cách 2: Dùng bảng hàm kích thích:
 - Xác định hệ hàm theo bảng kích thích của FF.
 - Tối thiểu hóa hàm Y.
 - Vẽ sơ đồ tương ứng.

4.2.7 Bảng kích thích của các Flip-Flop

4.2.7 Bảng kích thích của các Flip-Flop

Bảng kích thích các loại FF

Q_n	Q_{n+1}	S	R	D	J	K	T
0	0	0	X	0	0	X	0
0	1	1	0	1	1	X	1
1	0	0	1	0	X	1	1
1	1	X	0	1	X	0	0

4.2.7 Ví dụ chuyển đổi JK FF thành D FF

4.2.7 Ví dụ chuyển đổi JK FF thành D FF

		Q_n	
		0	1
J	D		
	0		X
	1	1	X

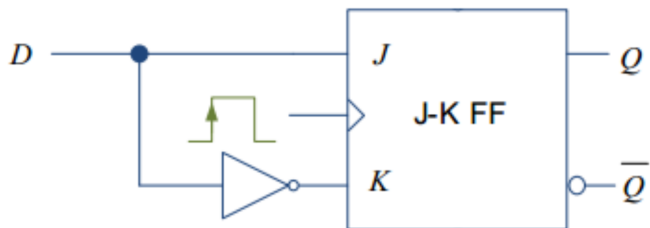
$$J = D$$

		Q_n	
		0	1
D	0	X	1
	1	X	

$$K = \overline{D}$$

4.2.7 Ví dụ chuyển đổi JK FF thành D FF

4.2.7 Ví dụ chuyển đổi JK FF thành D FF



4.3 Phương pháp mô tả mạch tuần tự

4.3 Phương pháp mô tả mạch tuần tự

4.3.1 Phương pháp mô tả mạch tuần tự bằng bảng

4.3 Phương pháp mô tả mạch tuần tự

4.3.1 Phương pháp mô tả mạch tuần tự bằng bảng

4.3.2 Phương pháp mô tả mạch tuần tự bằng đồ hình trạng thái

4.3.1 Phương pháp mô tả mạch tuần tự bằng bảng

4.3.1 Phương pháp mô tả mạch tuần tự bằng bảng

Trạng thái trong hiện tại Y_n	Trạng thái trong kế tiếp Y_{n+1}		Trạng thái ra Z_{n+1}	
	X_1	X_2	X_1	X_2
Y_1	Y_2	Y_1	Z_2	Z_1
Y_2	Y_2	Y_1	Z_2	Z_1

4.3.2 Phương pháp mô tả mạch tuần tự bằng đồ hình trạng thái

4.3.2 Phương pháp mô tả mạch tuần tự bằng đồ hình trạng thái

- Là sơ đồ biểu diễn trạng thái trong, trạng thái vào, trạng thái ra và quy luật chuyển đổi giữa chúng.

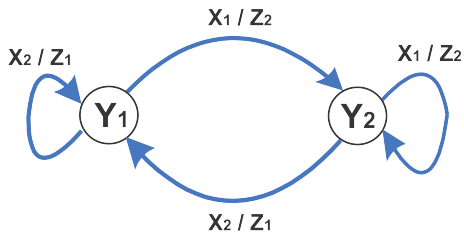
4.3.2 Phương pháp mô tả mạch tuần tự bằng đồ hình trạng thái

- Là sơ đồ biểu diễn trạng thái trong, trạng thái vào, trạng thái ra và quy luật chuyển đổi giữa chúng.
- Có hai loại sơ đồ trạng thái: Mealy và Moore.

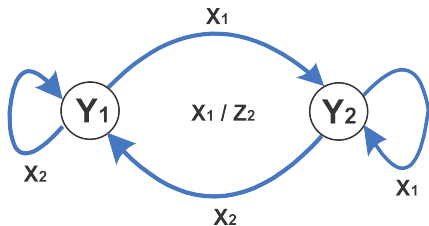
4.3.2 Phương pháp mô tả mạch tuần tự bằng đồ hình trạng thái

- Là sơ đồ biểu diễn trạng thái trong, trạng thái vào, trạng thái ra và quy luật chuyển đổi giữa chúng.
- Có hai loại sơ đồ trạng thái: Mealy và Moore.

Mealy



Moore



4.4 Phân tích và thiết kế mạch tuần tự

4.4 Phân tích và thiết kế mạch tuần tự

4.4.1 Phân tích mạch tuần tự.

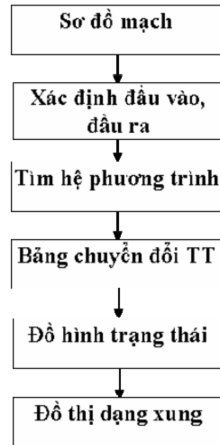
4.4 Phân tích và thiết kế mạch tuần tự

4.4.1 Phân tích mạch tuần tự.

4.4.2 Thiết kế mạch tuần tự.

4.4.1 Phân tích mạch tuần tự

- B1: Xác định chức năng từng phần tử
- B2: Xác định đầu vào, đầu ra, số trạng thái trong của mạch
- B3: Xác định pt hàm ra, pt hàm kích, pt chuyển đổi trạng thái
- B4: Lập bảng chuyển đổi trạng thái và tín hiệu ra
- B5: Vẽ đồ hình trạng thái
- B6: Vẽ đồ thị dạng xung
Nêu chức năng của mạch



4.4.2 Thiết kế mạch tuần tự

- *Bước 1:* Xác định bài toán, gán hàm và biến, tìm hiểu mối quan hệ giữa chúng.
- *Bước 2:* Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái và hàm ra.
- *Bước 3:* Rút gọn trạng thái (tối thiểu hoá trạng thái).
 - Việc tối thiểu hoá trạng thái chủ yếu dựa vào khái niệm trạng thái tương đương.
 - Các trạng thái tương đương với nhau có thể được thay bằng một trạng thái chung đại diện cho chúng.



4.4.2 Thiết kế mạch tuần tự

4.4.2 Thiết kế mạch tuần tự

- Bước 4: Mã hóa trạng thái

4.4.2 Thiết kế mạch tuần tự

- Bước 4: Mã hóa trạng thái
 - Số biến nhị phân dùng để mã hóa các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch.

4.4.2 Thiết kế mạch tuần tự

- Bước 4: Mã hóa trạng thái
 - Số biến nhị phân dùng để mã hóa các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch.
 - Có rất nhiều cách mã hóa khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Cần mã hóa sao cho sơ đồ mạch thực hiện là đơn giản nhất.

4.4.2 Thiết kế mạch tuần tự

- Bước 4: Mã hóa trạng thái
 - Số biến nhị phân dùng để mã hóa các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch.
 - Có rất nhiều cách mã hóa khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Cần mã hóa sao cho sơ đồ mạch thực hiện là đơn giản nhất.
- Bước 5: Xác định hệ phương trình của mạch. Có hai cách xác định:

4.4.2 Thiết kế mạch tuần tự

- Bước 4: Mã hóa trạng thái
 - Số biến nhị phân dùng để mã hóa các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch.
 - Có rất nhiều cách mã hóa khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Cần mã hóa sao cho sơ đồ mạch thực hiện là đơn giản nhất.
- Bước 5: Xác định hệ phương trình của mạch. Có hai cách xác định:
 - Lập bảng chuyển đổi trạng thái và tín hiệu ra, từ đó xác định các phương trình kích cho các FF.

4.4.2 Thiết kế mạch tuần tự

- Bước 4: Mã hóa trạng thái
 - Số biến nhị phân dùng để mã hóa các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch.
 - Có rất nhiều cách mã hóa khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Cần mã hóa sao cho sơ đồ mạch thực hiện là đơn giản nhất.
- Bước 5: Xác định hệ phương trình của mạch. Có hai cách xác định:
 - Lập bảng chuyển đổi trạng thái và tín hiệu ra, từ đó xác định các phương trình kích cho các FF.
 - Dựa trực tiếp vào đồ hình trạng thái, viết hệ phương trình và phương trình hàm ra.

4.4.2 Thiết kế mạch tuần tự

4.4.2 Thiết kế mạch tuần tự

- Cả hai cách đều có dạng phương trình:

4.4.2 Thiết kế mạch tuần tự

- Cả hai cách đều có dạng phương trình:
 - Phương trình của mạch chỉ dùng cổng NAND

4.4.2 Thiết kế mạch tuần tự

- Cả hai cách đều có dạng phương trình:
 - Phương trình của mạch chỉ dùng cổng NAND
 - Phương trình của mạch dùng SR FF không đồng bộ và các mạch AND.

4.4.2 Thiết kế mạch tuần tự

- Cả hai cách đều có dạng phương trình:
 - Phương trình của mạch chỉ dùng cổng NAND
 - Phương trình của mạch dùng SR FF không đồng bộ và các mạch AND.
 - Phương trình của mạch dùng các FF khác.

4.4.2 Thiết kế mạch tuần tự

- Cả hai cách đều có dạng phương trình:
 - Phương trình của mạch chỉ dùng cổng NAND
 - Phương trình của mạch dùng SR FF không đồng bộ và các mạch AND.
 - Phương trình của mạch dùng các FF khác.
- Bước 6: Vẽ sơ đồ thực hiện.

4.5 Mạch tuần tự thông dụng

4.5 Mạch tuần tự thông dụng

4.5.1 Bộ đếm

4.5 Mạch tuần tự thông dụng

4.5.1 Bộ đếm

4.5.2 Thanh ghi dịch

4.5.1 Bộ đếm

4.5.1 Bộ đếm

- Mạch đếm tuần tự không đồng bộ.

4.5.1 Bộ đếm

- Mạch đếm tuần tự không đồng bộ.
- Mạch đếm nhị phân đồng bộ.

4.5.1 Bộ đếm

- Mạch đếm tuần tự không đồng bộ.
- Mạch đếm nhị phân đồng bộ.
- Mạch chốt bằng D FF.

4.5.1 Bộ đếm

- Mạch đếm tuần tự không đồng bộ.
- Mạch đếm nhị phân đồng bộ.
- Mạch chốt bằng D FF.
- Mạch đếm không đồng bộ sử dụng D FF.

4.5.1 Bộ đếm

- Mạch đếm tuần tự không đồng bộ.
- Mạch đếm nhị phân đồng bộ.
- Mạch chốt bằng D FF.
- Mạch đếm không đồng bộ sử dụng D FF.
- Thiết kế mạch đếm đồng bộ bằng D FF.

4.5.1 Bộ đếm

- Mạch đếm tuần tự không đồng bộ.
- Mạch đếm nhị phân đồng bộ.
- Mạch chốt bằng D FF.
- Mạch đếm không đồng bộ sử dụng D FF.
- Thiết kế mạch đếm đồng bộ bằng D FF.
- Mạch đếm vòng xoắn.

4.5.1 Bộ đếm

- Mạch đếm tuần tự không đồng bộ.
- Mạch đếm nhị phân đồng bộ.
- Mạch chốt bằng D FF.
- Mạch đếm không đồng bộ sử dụng D FF.
- Thiết kế mạch đếm đồng bộ bằng D FF.
- Mạch đếm vòng xoắn.
- Mạch đếm vòng thuận.

4.5.2 Thanh ghi dịch

4.5.2 Thanh ghi dịch

- Thanh ghi dịch là phần tử không thể thiếu trong CPU, trong các hệ vi xử lý.

4.5.2 Thanh ghi dịch

- Thanh ghi dịch là phần tử không thể thiếu trong CPU, trong các hệ vi xử lý.
- Nó có khả năng nhớ số liệu và dịch thông tin (sang phải hoặc sang trái).

4.5.2 Thanh ghi dịch

- Thanh ghi dịch là phần tử không thể thiếu trong CPU, trong các hệ vi xử lý.
- Nó có khả năng nhớ số liệu và dịch thông tin (sang phải hoặc sang trái).
- Thanh ghi dịch được cấu tạo từ một dãy phần tử nhớ được mắc liên tiếp với nhau và một số cổng logic cơ bản hỗ trợ.

4.5.2 Thanh ghi dịch

- Thanh ghi dịch là phần tử không thể thiếu trong CPU, trong các hệ vi xử lý.
- Nó có khả năng nhớ số liệu và dịch thông tin (sang phải hoặc sang trái).
- Thanh ghi dịch được cấu tạo từ một dãy phần tử nhớ được mắc liên tiếp với nhau và một số cổng logic cơ bản hỗ trợ.
- Muốn ghi và truyền một từ nhị phân n bit thì cần n phần tử nhớ (n FF). Trong các thanh ghi dịch thường dùng các FF đồng bộ như RS, T, JK, D.

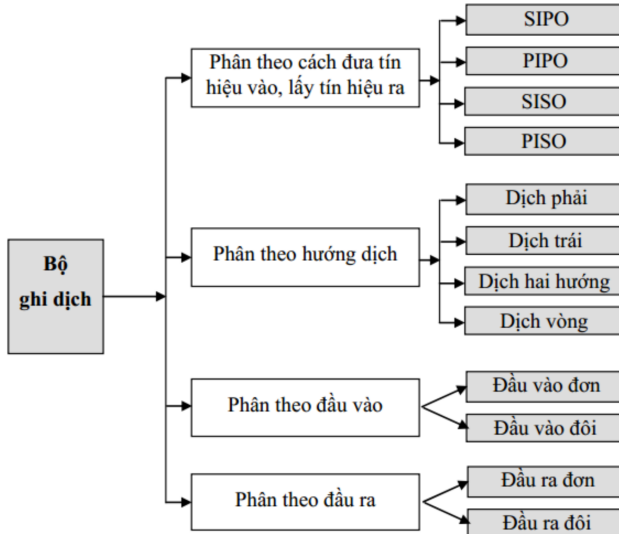
4.5.2 Thanh ghi dịch

- Thanh ghi dịch là phần tử không thể thiếu trong CPU, trong các hệ vi xử lý.
- Nó có khả năng nhớ số liệu và dịch thông tin (sang phải hoặc sang trái).
- Thanh ghi dịch được cấu tạo từ một dãy phần tử nhớ được mắc liên tiếp với nhau và một số cổng logic cơ bản hỗ trợ.
- Muốn ghi và truyền một từ nhị phân n bit thì cần n phần tử nhớ (n FF). Trong các thanh ghi dịch thường dùng các FF đồng bộ như RS, T, JK, D.
- Thông thường người ta hay dùng D FF hoặc các FF khác nhưng mắc theo kiểu D FF để tạo thành các thanh ghi.

4.5.2 Thanh ghi dịch

- Thanh ghi dịch là phần tử không thể thiếu trong CPU, trong các hệ vi xử lý.
- Nó có khả năng nhớ số liệu và dịch thông tin (sang phải hoặc sang trái).
- Thanh ghi dịch được cấu tạo từ một dãy phần tử nhớ được mắc liên tiếp với nhau và một số cổng logic cơ bản hỗ trợ.
- Muốn ghi và truyền một từ nhị phân n bit thì cần n phần tử nhớ (n FF). Trong các thanh ghi dịch thường dùng các FF đồng bộ như RS, T, JK, D.
- Thông thường người ta hay dùng D FF hoặc các FF khác nhưng mắc theo kiểu D FF để tạo thành các thanh ghi.
- Hoạt động dịch dữ liệu được thực hiện từng bit theo điều khiển của xung clock.

Phân loại



4.5.2 Thanh ghi dịch

4.5.2 Thanh ghi dịch

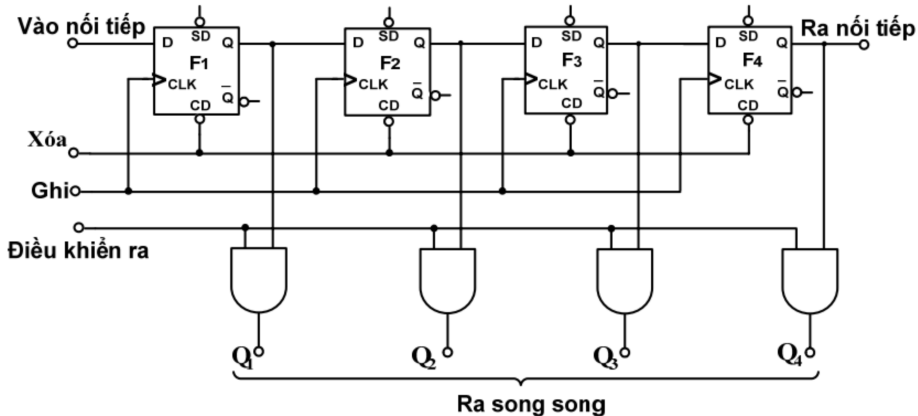
- Mạch ghi dịch vào nối tiếp ra song song: dữ liệu được nhập nối tiếp vào FF đầu tiên và lấy ra cùng lúc ở ngõ ra của tất cả các FF.

4.5.2 Thanh ghi dịch

- Mạch ghi dịch vào nối tiếp ra song song: dữ liệu được nhập nối tiếp vào FF đầu tiên và lấy ra cùng lúc ở ngõ ra của tất cả các FF.
- Mạch ghi dịch vào song song ra song song: người ta sử dụng 2 ngõ preset và clear của các FF để nhập dữ liệu và dữ liệu được lấy ra cùng lúc ở tất cả các ngõ ra của các FF.

Mạch ghi dịch vào nối tiếp ra song song

Mạch ghi dịch vào nối tiếp ra song song



Mạch ghi dịch vào nối tiếp ra song song

Mạch ghi dịch vào nối tiếp ra song song

Cách ghi dữ liệu vào:

- Số liệu: D_1, D_2, D_3, D_4
- Dịch phải, nhập D_4 trước.
- Sau 4 xung nhịp: ghi xong.

Cách lấy số liệu ra:

- Ra song song:

Sau 4 xung nhịp:

$$Q_1 Q_2 Q_3 Q_4 = D_1 D_2 D_3 D_4$$

Đặt “Điều khiển ra” = 1.

- Ra nối tiếp:

Lấy ra ở Q_4 (trigơ cuối cùng).

Sau 4 xung nhịp:

D_4 xuất hiện ở Q_4

Cần 3 xung nữa để $D_1 D_2 D_3$ được đưa ra Q_4 .

CLK	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	D_4	0	0	0
2	D_3	D_4	0	0
3	D_2	D_3	D_4	0
4	D_1	D_2	D_3	D_4
5	0	D_1	D_2	D_3
6	0	0	D_1	D_2
7	0	0	0	D_1

Mạch ghi dịch vào song song ra song song