

☎ CHƯƠNG 5 MẠCH TUẦN TỰ

★ CHỐT RS

- ♠ Chốt RS tác động mức cao
- ♠ Chốt RS tác động mức thấp

★ FLIPFLOP

- ♠ FF RS
- ♠ FF JK
- ♠ FF T
- ♠ FF D

★ MẠCH GHI DỊCH

★ MẠCH ĐẾM

- ♠ Đồng bộ
- ♠ Không đồng bộ
- ♠ Đếm vòng

Trong chương trước, chúng ta đã khảo sát các loại mạch tổ hợp, đó là các mạch mà ngõ ra của nó chỉ phụ thuộc vào các biến ở ngõ vào mà không phụ thuộc vào trạng thái trước đó của mạch. Nói cách khác, đây là loại mạch không có khả năng nhớ, một chức năng quan trọng trong các hệ thống logic.

Chương này sẽ bàn về loại mạch thứ hai: mạch tuần tự.

- Mạch tuần tự là mạch có trạng thái ngõ ra không những phụ thuộc vào tổ hợp các ngõ vào mà còn phụ thuộc trạng thái ngõ ra trước đó. Ta nói mạch tuần tự có **tính nhớ**. Ngõ ra Q_+ của mạch tuần tự là hàm logic của các biến ngõ vào $A, B, C \dots$ và ngõ ra Q trước đó.

$$Q_+ = f(Q, A, B, C \dots)$$

- Mạch tuần tự vận hành dưới tác động của xung đồng hồ và được chia làm 2 loại: **Đồng bộ** và **Không đồng bộ**. Ở mạch đồng bộ, các phần tử của mạch chịu **tác động đồng thời** của xung đồng hồ (C_K) và ở mạch không đồng bộ thì không có điều kiện này.

Phần tử cơ bản cấu thành mạch tuần tự là các flipflop

5.1 FLIP FLOP

Mạch flipflop (FF) là mạch dao động đa hài lưỡng ổn tức mạch tạo ra sóng vuông và có hai trạng thái ổn định. Trạng thái của FF chỉ thay đổi khi có xung đồng hồ tác động.

Một FF thường có:

- Một hoặc hai ngõ vào dữ liệu, một ngõ vào xung C_K và có thể có các ngõ vào với các chức năng khác.

- Hai ngõ ra, thường được ký hiệu là Q (ngõ ra chính) và \bar{Q} (ngõ ra phụ). Người ta thường dùng trạng thái của ngõ ra chính để chỉ trạng thái của FF. Nếu hai ngõ ra có trạng thái giống nhau ta nói FF ở **trạng thái cấm**.

Flipflop có thể được tạo nên từ mạch chốt (latch)

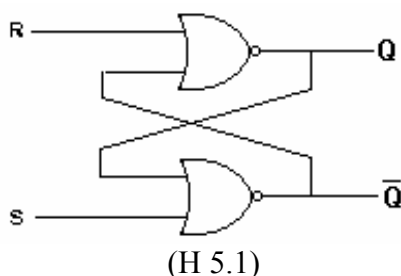
Điểm khác biệt giữa một mạch chốt và một FF là: FF chịu tác động của xung đồng hồ còn mạch chốt thì không.

Người ta gọi tên các FF khác nhau bằng cách dựa vào tên các ngõ vào dữ liệu của chúng.

5.1.1 Chốt RS

5.1.1.1. Chốt RS tác động mức cao:

(H 5.1) là chốt RS có các ngõ vào R và S tác động mức cao.



Các trạng thái logic của mạch cho ở bảng 5.1:

(Đối với mạch chốt vì không có tác động của xung đồng hồ nên ta có thể hiểu trạng thái trước là trạng thái giả sử, còn trạng thái sau là trạng thái khi mạch ổn định).

R	S	Q	Q ₊
0	0	0	0 Tác dụng nhớ
0	0	1	1 Q ₊ = Q
0	1	0	1 Đặt (Set)
0	1	1	1 Q ₊ = 1
1	0	0	0 Đặt lại (Reset)
1	0	1	0 Q ₊ = 0
1	1	0	Q ₊ = Q ₊ = 0 (Cấm)
1	1	1	

R	S	Q ₊
0	0	Q
0	1	1
1	0	0
1	1	Cấm

Bảng 5.1

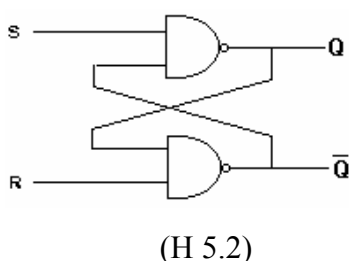
Bảng 5.2

Từ Bảng 5.1 thu gọn lại thành Bảng 5.2 và tính chất của chốt RS tác động mức cao được tóm tắt như sau:

- Khi R=S=0 (cả 2 ngõ vào đều không tác động), ngõ ra không đổi trạng thái.
- Khi R=0 và S=1 (ngõ vào S tác động), chốt được Set (tức đặt Q₊=1).
- Khi R=1 và S=0 (ngõ vào R tác động), chốt được Reset (tức đặt lại Q₊=0).
- Khi R=S=1 (cả 2 ngõ vào đều tác động), chốt rơi vào trạng thái cấm

5.1.1.2. Chốt RS tác động mức thấp:

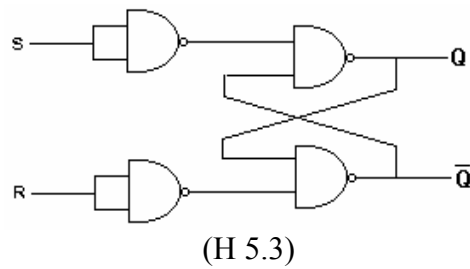
(H 5.2) là chốt RS có các ngõ vào R và S tác động mức thấp. Các trạng thái logic cho bởi Bảng 5.3



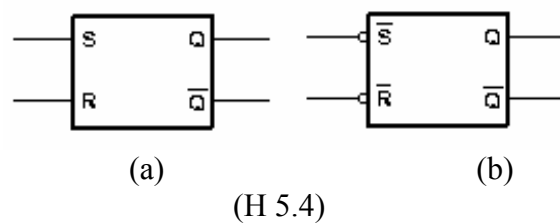
S	R	Q ₊
0	0	Cấm
0	1	1
1	0	0
1	1	Q

Bảng 5.3

Để có chốt RS tác động mức cao dùng cổng NAND, người ta thêm vào 2 cổng đảo ở các ngõ vào của mạch (H 5.2)

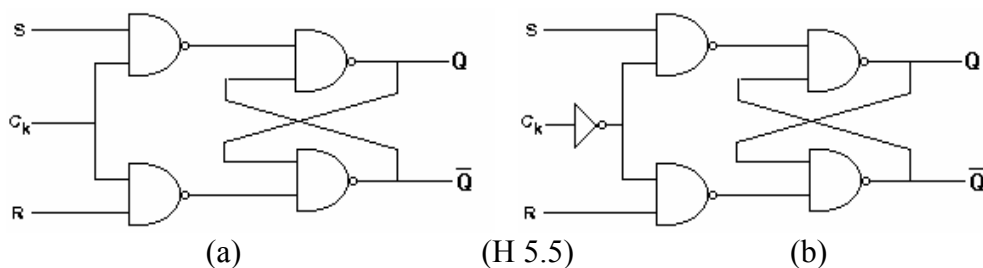


(H 5.4a) là ký hiệu chốt RS tác động cao và (H 5.4b) là chốt RS tác động thấp.



5.1.2 Flip Flop RS

Trong các phần dưới đây, ta luôn sử dụng chốt RS tác động mức cao dùng cổng NAND. Khi thêm ngõ vào xung C_K cho chốt RS ta được FF RS. (H 5.5a) là FF RS có các ngõ vào R, S và xung đồng hồ C_K đều tác động mức cao.



Hoạt động của FF (H 5.5a) cho bởi Bảng sự thật: (Bảng 5.4)

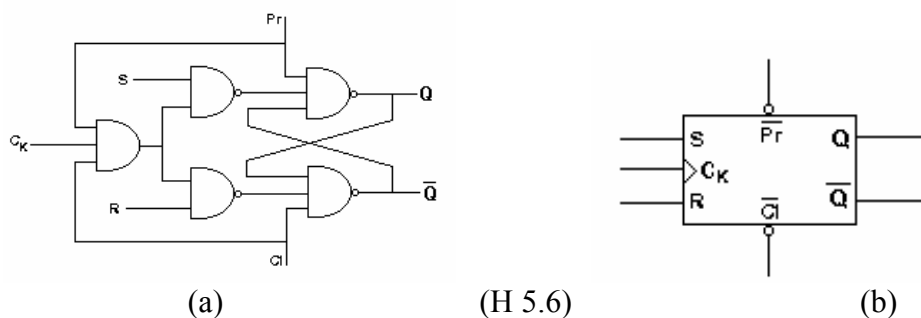
Vào			Ra
C_K	S	R	Q_+
0	x	x	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	Cắm

Bảng 5.4

Để có FF RS có xung đồng hồ tác động thấp chỉ cần thêm một cổng đảo cho ngõ vào C_K (H 5.5b). Ta có bảng sự thật giống Bảng 5.4, trừ ngõ vào C_K phải đảo lại

5.1.2.1. Flipflop RS có ngõ vào Preset và Clear:

Tính chất của FF là có trạng thái ngõ ra bất kỳ khi mở máy. Trong nhiều trường hợp, có thể cần đặt trước ngõ ra $Q=1$ hoặc $Q=0$, muốn thế, người ta thêm vào FF các ngõ vào Preset (đặt trước $Q=1$) và Clear (Xóa $Q=0$), mạch có dạng (H 5.6a) và (H 5.6b) là ký hiệu của FF RS có ngõ vào Preset và Clear tác động mức thấp.



Thay 2 cổng NAND cuối bằng hai cổng NAND 3 ngõ vào, ta được FF RS có ngõ vào Preset (Pr) và Clear (Cl).

- Khi ngõ Pr xuống thấp (tác động) và ngõ Cl lên cao ngõ ra Q lên cao bất chấp các ngõ vào còn lại.

- Khi ngõ Cl xuống thấp (tác động) và ngõ Pr lên cao ngõ ra Q xuống thấp bất chấp các ngõ vào còn lại.

- Ngoài ra 2 ngõ vào Pr và Cl còn được đưa về 2 ngõ vào một cổng AND, nơi đưa tín hiệu C_K vào, mục đích của việc làm này là khi một trong 2 ngõ vào Pr hoặc Cl tác động thì mức thấp của tín hiệu này sẽ khóa cổng AND này, vô hiệu hóa tác dụng của xung C_K .

Bảng sự thật của FF RS có Preset và Clear (tác động thấp) cho ở bảng 5.5

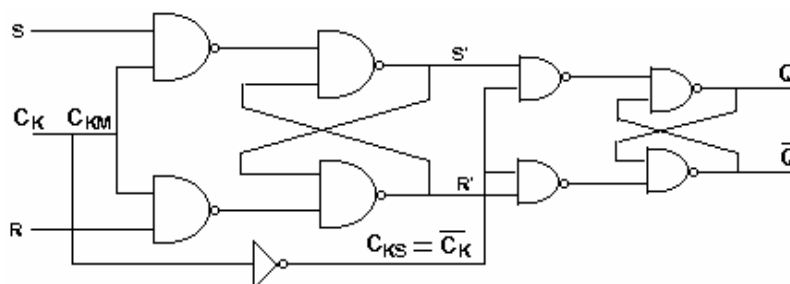
Pr	Cl	C_K	S	R	Q_+
0	0	x	x	x	Cấm
0	1	x	x	x	1
1	0	x	x	x	0
1	1	0	x	x	Q
1	1	1	0	0	Q
1	1	1	0	1	0
1	1	1	1	0	1
1	1	1	1	1	Cấm

Bảng 5.5

Lưu ý: Trên bảng 5.5, dòng thứ nhất tương ứng với trạng thái cấm vì hai ngõ vào Pr và Cl đồng thời ở mức tác động, 2 cổng NAND cuối cùng đều đóng, nên $Q_+=Q=1$.

5.1.2.2. Flipflop RS chủ tớ:

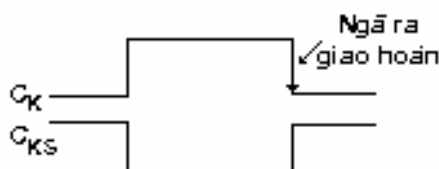
Kết nối thành chuỗi hai FF RS với hai ngõ vào xung C_K của hai FF có mức tác động trái ngược nhau, ta được FF chủ tớ (H 5.7).



(H 5.7)

Hoạt động của FF được giải thích như sau:

- Do C_{KS} của tầng tổ là đảo của $C_{KM} = C_K$ của tầng chủ nên khi $C_K=1$, tầng chủ giao hoán thì tầng tổ ngưng. Trong khoảng thời gian này, dữ liệu từ ngã vào R và S được đưa ra và ổn định ở ngã ra R' và S' của tầng chủ, tại thời điểm xung C_K xuống thấp, R' và S' được truyền đến ngã ra Q và \overline{Q} (H 5.8)



(H 5.8)

- Đối với trường hợp $R = S = 1$ khi $C_K = 1$ thì $R' = S' = 1$, nhưng khi C_K xuống thấp thì một trong hai ngã ra này xuống thấp, do đó mạch thoát khỏi trạng thái cấm, nhưng S' hay R' xuống thấp trước thì không đoán trước được nên mạch rơi vào trạng thái bất định, nghĩa là Q_+ có thể $= 1$ có thể $= 0$, nhưng khác với \overline{Q}_+ . Ta có bảng sự thật:

S	R	C_K	Q_+
0	0	\downarrow	Q
0	1	\downarrow	0
1	0	\downarrow	1
1	1	\downarrow	Bất định

Bảng 5.6

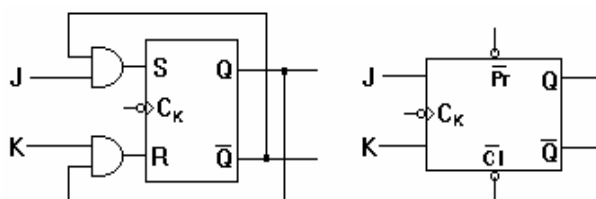
Tóm lại, FF RS chủ tớ đã thoát khỏi trạng thái cấm nhưng vẫn rơi vào trạng thái bất định, đồng thời ta được FF có ngã vào xung đồng hồ tác động bởi cạnh xuống của tín hiệu C_K .

Để có FF RS có ngã vào xung đồng hồ tác động bởi cạnh lên của tín hiệu C_K ta có thể dời cổng NOT đến ngã vào FF chủ và cho tín hiệu C_K vào thẳng FF tớ.

Mặc dù thoát khỏi trạng thái cấm nhưng FF RS chủ tớ vẫn còn trạng thái bất định nên người ta ít sử dụng FF RS trong trường hợp $R=S$.

5.1.3 Flipflop JK

FF JK được tạo ra từ FF RS theo sơ đồ như (H 5.9a).



(a)
(H 5.9)

(b)

(H 5.9b) là ký hiệu FF JK có ngõ vào Pr và Cl tác động thấp.
Bảng sự thật 5.7 (Để đơn giản, ta bỏ qua các ngõ vào Pr và Cl)

J	K	Q	\bar{Q}	$S=J\bar{Q}$	$R=KQ$	C_K	Q_+
0	0	0	1	0	0	↓	Q
0	0	1	0	0	0	↓	Q
0	1	0	1	0	0	↓	Q=0
0	1	1	0	0	1	↓	0
1	0	0	1	1	0	↓	1
1	0	1	0	0	0	↓	Q=1
1	1	0	1	1	0	↓	1
1	1	1	0	0	1	↓	0

J	K	C_K	Q_+
0	0	↓	Q
0	1	↓	0
1	0	↓	1
1	1	↓	\bar{Q}

Bảng 5.8

Bảng 5.7

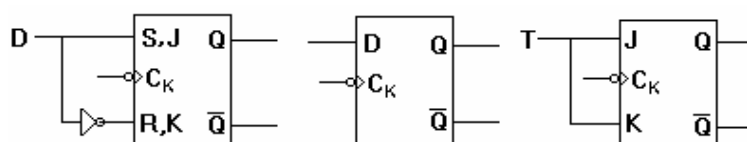
Bảng 5.8 là bảng rút gọn, suy ra từ bảng 5.7

Kết quả từ bảng 5.8 cho thấy:

FF JK đã thoát khỏi trạng thái cấm và thay vào đó là **trạng thái đảo** (khi $J=K=1$ thì $Q_+=\bar{Q}$). Người ta lợi dụng trạng thái đảo này để thiết kế mạch đếm

5.1.4 FlipFlop D

Thiết kế từ FF RS (hoặc JK) bằng cách nối một cổng đảo từ S qua R (hoặc từ J qua K). Dữ liệu được đưa vào ngõ S (J) mà bây giờ gọi là ngõ vào D (H 5.10a&b) và bảng 5.9 cho thấy các trạng thái của FF, cụ thể là mỗi khi có xung C_K tác động dữ liệu từ ngõ vào sẽ xuất hiện ở ngõ ra.



(a)

(b)

(c)

(H 5.10)

D	C_K	Q_+
0	↓	0
1	↓	1

Bảng 5.9

T	C_K	Q_+
0	↓	Q
1	↓	\bar{Q}

Bảng 5.10

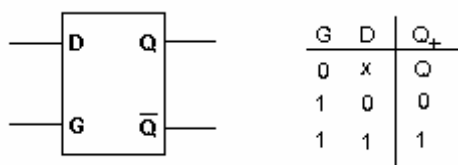
5.1.5 FlipFlop T

Nối chung hai ngõ vào J và K của FF JK ta được FF T (H 5.10c). Tính chất của FF T thể hiện trong bảng sự thật 5.10:

- Khi $T=0$, **FF không đổi trạng thái** dù có tác động của C_K .
- Khi $T=1$, **FF đổi trạng thái** mỗi lần có xung C_K tác động.

5.1.6 Mạch chốt D

Mạch chốt D hoạt động giống FF D, chỉ khác ở điểm ngõ vào xung đồng hồ C_K được thay bằng ngõ vào cho phép G, và tác động bằng mức chứ không bằng cạnh (H 5.11) và Bảng 5.11.

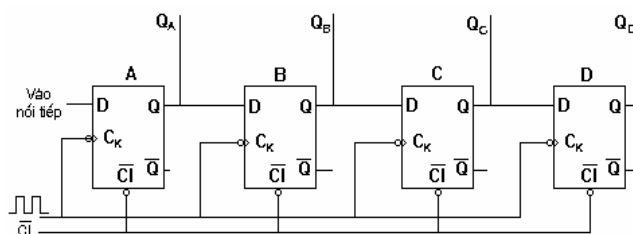


(H 5.11)

Bảng 5.11

5.2 MẠCH GHI DỊCH

5.2.1 Sơ đồ nguyên tắc và vận chuyển (H 5.12)



(H 5.12)

(H 5.12) là sơ đồ một mạch ghi dịch 4 bit đơn giản, mạch gồm 4 FF D nối thành chuỗi (ngõ ra Q của FF trước nối vào ngõ vào D của FF sau) và các ngõ vào C_K được nối chung lại (các FF chịu tác động đồng thời). Mạch ghi dịch này có khả năng dịch phải.

Ngõ vào D_A của FF đầu tiên được gọi là **ngõ vào dữ liệu nối tiếp**, các ngõ ra Q_A, Q_B, Q_C, Q_D là các **ngõ ra song song**, ngõ ra của FF cuối cùng (FF D) là **ngõ ra nối tiếp**.

Trước khi cho mạch hoạt động, tác dụng một xung xóa vào các ngõ vào \overline{Cl} (đưa các chân \overline{Cl} đã được nối chung xuống thấp rồi lên cao) để các ngõ ra $Q_A = Q_B = Q_C = Q_D = 0$.

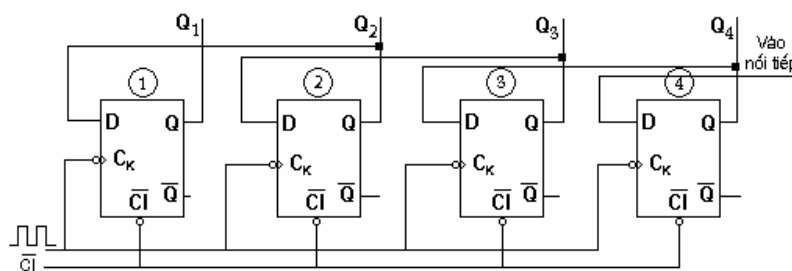
Cho dữ liệu vào D_A , sau mỗi xung đồng hồ, dữ liệu từ tầng trước lần lượt truyền qua tầng sau. (Giả sử D_A là chuỗi dữ liệu gồm 3 bit cao, 2 bit thấp rồi 1 cao và 1 thấp), trạng thái các ngõ ra của các FF cho ở Bảng 5.12

Vào			Ra			
Cl	C_K	D_A	Q_A	Q_B	Q_C	Q_D
0	x	x	0	0	0	0
1	↓	1	1	0	0	0
1	↓	1	1	1	0	0
1	↓	1	1	1	1	0
1	↓	0	0	1	1	1
1	↓	0	0	0	1	1
1	↓	1	1	0	0	1
1	↓	0	0	1	0	0

Bảng 5.12

Các mạch ghi dịch được phân loại tùy vào số bit (số FF), chiều dịch (phải/trái), các ngõ vào/ra (nối tiếp/song song).

Để có mạch dịch trái, dữ liệu nối tiếp đưa vào ngõ vào D của FF cuối cùng và các ngõ ra của FF sau nối ngược trở lại ngõ vào của FF trước (H 5.13)



(H 5.13)

Cho dữ liệu nối tiếp vào ngõ vào D của FF 4, sau mỗi xung đồng hồ, dữ liệu truyền từ tầng sau ra tầng trước. Giả sử chuỗi dữ liệu giống như trên, trạng thái các ngõ ra của các FF cho ở bảng 5.13

Vào			Ra			
Cl	C _K	D ₄	Q ₁	Q ₂	Q ₃	Q ₄
0	x	x	0	0	0	0
1	↓	1	0	0	0	1
1	↓	1	0	0	1	1
1	↓	1	0	1	1	1
1	↓	0	1	1	1	0
1	↓	0	1	1	0	0
1	↓	1	1	0	0	1
1	↓	0	0	0	1	0

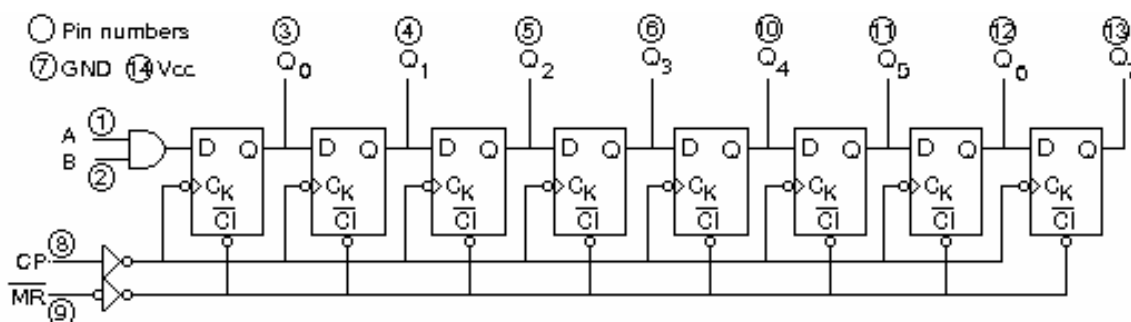
Bảng 5.13

5.2.2 Vài IC ghi dịch tiêu biểu

Trên thị trường hiện có khá nhiều loại IC ghi dịch, có đầy đủ các chức năng dịch phải trái, vào/ra nối tiếp, song song. Sau đây, chúng ta khảo sát 2 IC tiêu biểu:

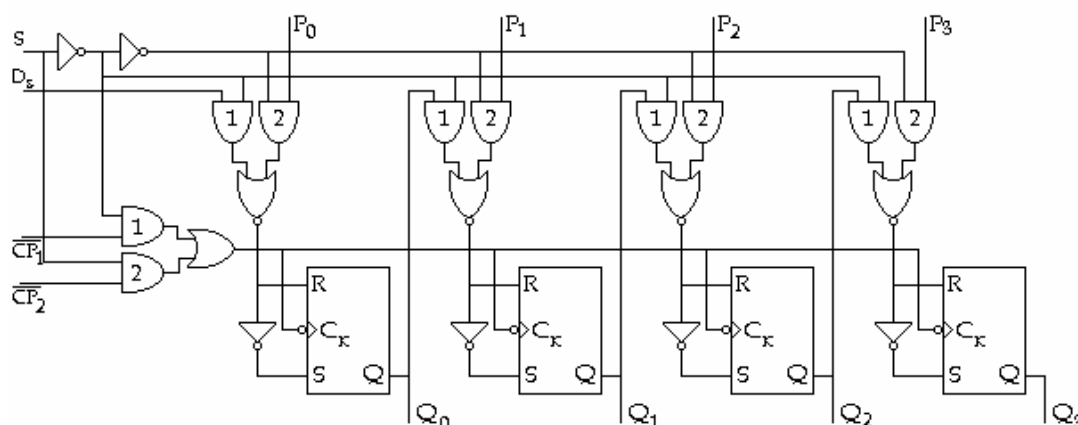
- IC **74164**: dịch phải 8 bit;
- IC **7495**: 4 bit , dịch phải, trái, vào/ra nối tiếp/song song .

5.2.2.1. IC 74164:



(H 5.14)

\overline{MR} : Master Reset, đây cũng là chân Clear của cả mạch, tác động thấp
CP: Clock pulse, ngõ vào xung đồng hồ tác động cạnh lên.

5.2.2.2. IC 7495:

(H 5.15)

Ý nghĩa các chân: **S**: Mode control input **Ds**: Serial Data input
P₀ - P₃ : Parrallel data inputs
CP₁ : Serial Clock **CP₂** : Parrallel clock
Q₀ - Q₃ : Parrallel outputs

Dưới đây là các bước thao tác để thực hiện các chức năng của IC

*** Nạp dữ liệu song song**

- Chuẩn bị dữ liệu ở các ngõ vào P₀ - P₃
- Cho S = 1, dữ liệu được đưa vào các ngõ vào của các FF, CP₁ bị khóa, CP₂ là ngõ vào C_K, dữ liệu xuất hiện ở ngõ ra Q₀ - Q₃ khi có cạnh xuống của C_K

*** Dịch phải**

- Sau khi đã nạp dữ liệu song song - Chuẩn bị dữ liệu nối tiếp.
- Cho S = 0
- Đưa dữ liệu nối tiếp vào ngõ vào Ds, CP₂ bị khóa, CP₁ là ngõ vào C_K, khi C_K tác động, dữ liệu sẽ dịch phải từng bit một trên các ngõ ra Q₀ - Q₃

*** Dịch trái**

- Nối ngõ ra FF sau vào ngõ vào song song của FF trước - P₃ là ngõ vào nối tiếp
- S = 1 để cách ly ngõ ra FF trước với ngõ vào FF sau
- CP₂ là ngõ vào xung C_K, dữ liệu sẽ dịch trái ứng với cạnh xuống của C_K.

Lưu ý: Mặc dù có 2 ngõ vào cho xung C_K nhưng khi sử dụng chúng thường được nối chung lại, lý do là vì ứng với một trạng thái của tín hiệu điều khiển S chỉ có một trong hai cổng AND mở để cho tín hiệu C_K đi qua.

5.2.3. Ứng dụng của ghi dịch

Ghi dịch có khá nhiều ứng dụng:

- Một số nhị phân khi dịch trái 1 bit, giá trị được nhân lên gấp đôi và được chia hai khi dịch phải một bit.

Thí dụ số 1010.00 = 10₁₀ khi dịch trái thành 10100.0 = 20₁₀ và khi dịch phải thành 101.000 = 5₁₀.

- Trong máy tính thanh ghi (tên thường gọi của mạch ghi dịch) là nơi lưu tạm dữ liệu để thực hiện các phép tính, các lệnh cơ bản như quay, dịch

- Ngoài ra, mạch ghi dịch còn những ứng dụng khác như: tạo mạch đếm vòng, biến đổi dữ liệu nối tiếp \leftrightarrow song song, dùng thiết kế các mạch đèn trang trí, quang báo. . . .

5.3 MẠCH ĐẾM

Lợi dụng tính đảo trạng thái của FF JK khi $J=K=1$, người ta thực hiện các mạch đếm.

Chức năng của mạch đếm là đếm số xung C_K đưa vào ngõ vào hoặc thể hiện số trạng thái có thể có của các ngõ ra.

Nếu xét khía cạnh tần số của tín hiệu thì mạch đếm có chức năng chia tần, nghĩa là tần số của tín hiệu ở ngõ ra là kết quả của phép chia tần số của tín hiệu C_K ở ngõ vào cho số đếm của mạch.

Ta có các loại: mạch đếm đồng bộ, không đồng bộ và đếm vòng.

5.3.1 Mạch đếm đồng bộ

Trong mạch đếm đồng bộ các FF chịu tác động đồng thời của xung đếm C_K .

5.3.1.1 Mạch đếm đồng bộ n tầng, đếm lên

Để thiết kế mạch đếm đồng bộ n tầng (lấy thí dụ $n=4$), trước tiên lập bảng trạng thái, quan sát bảng trạng thái suy ra cách mắc các ngõ vào JK của các FF sao cho mạch giao hoán tạo các ngõ ra đúng như bảng đã lập. Giả sử ta dùng FF tác động bởi cạnh xuống của xung C_K (Thật ra, kết quả thiết kế không phụ thuộc vào chiều tác động của xung C_K , tuy nhiên điều này phải được thể hiện trên mạch nên ta cũng cần lưu ý). Với 4 FF mạch đếm được $2^4=16$ trạng thái và số đếm được từ 0 đến 15. Ta có bảng trạng thái:

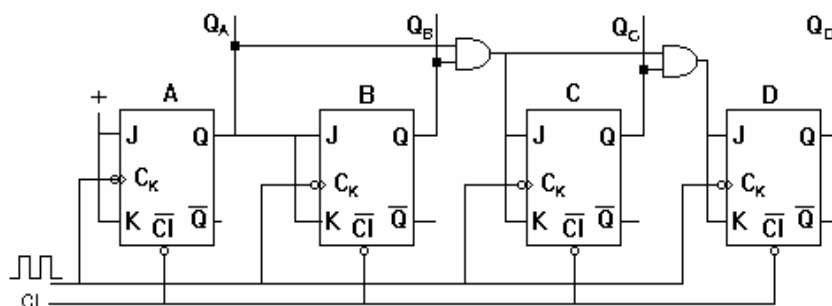
C_K	Q_D	Q_C	Q_B	Q_A	Số đếm
Xóa	0	0	0	0	0
1↓	0	0	0√	<u>1</u>	1
2↓	0	0	1	0	2
3↓	0	0√	<u>1</u> √	<u>1</u>	3
4↓	0	1	0	0	4
5↓	0	1	0√	<u>1</u>	5
6↓	0	1	1	0	6
7↓	0√	<u>1</u> √	<u>1</u> √	<u>1</u>	7
8↓	1	0	0	0	8
9↓	1	0	0√	<u>1</u>	9
10↓	1	0	1	0	10
11↓	1	0√	<u>1</u> √	<u>1</u>	11
12↓	1	1	0	0	12
13↓	1	1	0√	<u>1</u>	13
14↓	1	1	1	0	14
15↓	1√	<u>1</u> √	<u>1</u> √	<u>1</u>	15
16↓	0	0	0	0	0

Bảng 5.14

Nhận thấy:

- FF A đổi trạng thái sau từng xung C_K , vậy: $T_A = J_A = K_A = 1$
 - FF B đổi trạng thái nếu trước đó $Q_A = 1$, vậy $T_B = J_B = K_B = Q_A$
 - FF C đổi trạng thái nếu trước đó $Q_A = Q_B = 1$, vậy: $T_C = J_C = K_C = Q_A \cdot Q_B$
 - FF D đổi trạng thái nếu trước đó $Q_A = Q_B = Q_C = 1$, vậy:
 $T_D = J_D = K_D = Q_A \cdot Q_B \cdot Q_C = T_C \cdot Q_C$

Ta được kết quả ở (H 5.16)



(H 5.16)

5.3.1.2 Mạch đếm đồng bộ n tầng, đếm xuống

Bảng trạng thái:

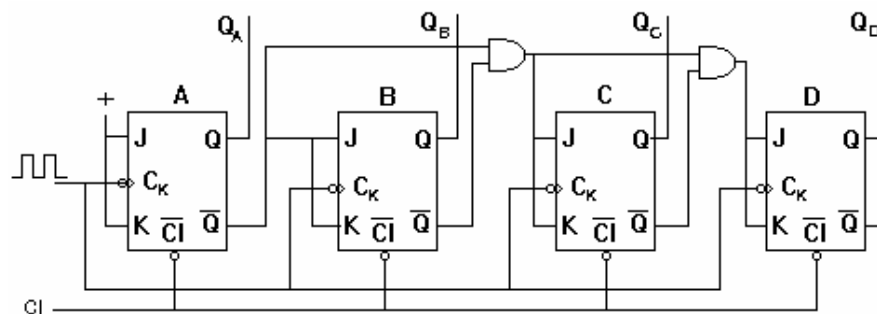
C_k	Q_D	Q_C	Q_B	Q_A	Số đếm
Xóa	0√	0√	0√	0	0
1↓	1	1	1	1	15
2↓	1	1	1√	0	14
3↓	1	1	0	1	13
4↓	1	1√	0√	0	12
5↓	1	0	1	1	11
6↓	1	0	1√	0	10
7↓	1	0	0	1	9
8↓	1√	0√	0√	0	8
9↓	0	1	1	1	7
10↓	0	1	1√	0	6
11↓	0	1	0	1	5
12↓	0	1√	0√	0	4
13↓	0	0	1	1	3
14↓	0	0	1√	0	2
15↓	0	0	0	1	1
16↓	0	0	0	0	0

Bảng 5.15

Nhân thấy:

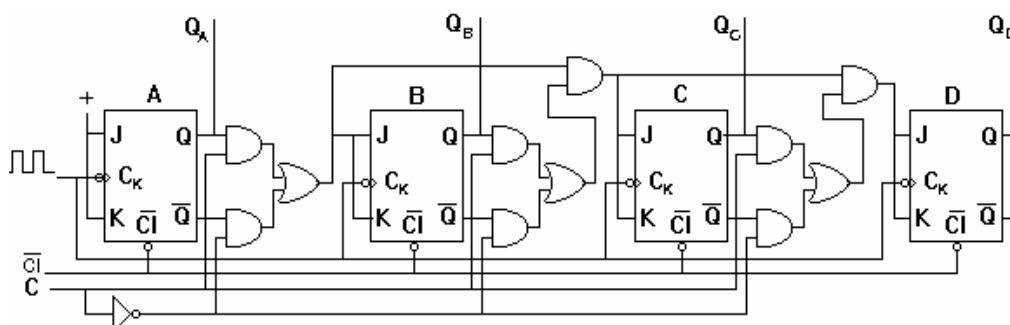
- FF A đổi trạng thái sau từng xung C_K , vậy: $T_A = J_A = K_A = 1$
 - FF B đổi trạng thái nếu trước đó $Q_A = 0$, vậy: $T_B = J_B = K_B = \overline{Q_A}$
 - FF C đổi trạng thái nếu trước đó $Q_A = Q_B = 0$, vậy: $T_C = J_C = K_C = \overline{Q_A} \overline{Q_B}$
 - FF D đổi trạng thái nếu trước đó $Q_A = Q_B = Q_C = 0$, vậy:

Ta được kết quả ở (H 5.17)



(H 5.17)

Để có mạch đếm n tầng, đếm lên hoặc xuống ta dùng một đa hợp 2→1 có ngõ vào điều khiển C để chọn Q hoặc \overline{Q} đưa vào tầng sau qua các cổng AND. Trong mạch (H.5.18) dưới đây khi $C=1$ mạch đếm lên và khi $C=0$ mạch đếm xuống.



(H 5.18)

Trong mạch (H 5.16) ta cần 2 cổng AND. Trong trường hợp tổng quát cho n tầng, số cổng AND là (n-2) như vậy thời gian tối thiểu để tín hiệu truyền qua mạch là:

$$T_{\min} = T_{\text{PFF}} + T_{\text{P_AND}}(n-2)$$

Tần số cực đại xác định bởi:

$$f_{\max} = \frac{1}{T_{\min}} = \frac{1}{t_{\text{PEF}} + (n-2)T_{\text{PAND}}}$$

Để gia tăng tần số làm việc của mạch, thay vì dùng các cổng AND 2 ngõ vào ta phải dùng cổng AND nhiều ngõ vào và mắc theo kiểu:

$$\begin{array}{ll} T_A = J_A = K_A = 1 & T_B = J_B = K_B = Q_A \\ T_C = J_C = K_C = Q_A \cdot Q_B & T_D = J_D = K_D = Q_A \cdot Q_B \cdot Q_C \end{array}$$

Như vậy tần số làm việc không phụ thuộc vào n và bằng:

$$f_{\max} = \frac{1}{T_{\text{PFE}} + T_{\text{PAND}}}$$

5.3.1.5 Mạch đếm đồng bộ Modulo - N ($N \neq 2^n$)

Để thiết kế mạch đếm modulo - N, trước nhất ta phải chọn số tầng.

Số tầng n phải thỏa điều kiện:

$$2^{n-1} < N < 2^n$$

Thí dụ thiết kế mạch đếm 10 ($N = 10$).

$$2^{4-1} < 10 < 2^4$$

Vậy số tầng là 4

Có nhiều phương pháp thiết kế mạch đếm đồng bộ modulo-N.

Sau đây ta khảo sát hai phương pháp : dùng **hàm Chuyển** và **MARCUS**

* Phương pháp dùng hàm Chuyển (Transfer function)

Hàm Chuyển là hàm cho thấy có sự thay đổi trạng thái của FF. Mỗi loại FF có một hàm Chuyển riêng của nó.

Hàm Chuyển được định nghĩa như sau: hàm có trị 1 khi có sự thay đổi trạng thái của FF ($Q_+ \neq Q$) và trị 0 khi trạng thái FF không đổi ($Q_+ = Q$).

Chúng ta chỉ thiết kế mạch đếm dùng FF JK do đó ta chỉ xác định hàm Chuyển của loại FF này.

Bảng trạng thái của FF JK (Bảng 5.16)

C_K	J	K	Q	Q_+	H
\downarrow	0	0	0	0	0
\downarrow	0	0	1	1	0
\downarrow	0	1	0	0	0
\downarrow	0	1	1	0	1
\downarrow	1	0	0	1	1
\downarrow	1	0	1	1	0
\downarrow	1	1	0	1	1
\downarrow	1	1	1	0	1

Bảng 5.16

Dùng Bảng Karnaugh ta suy ra được biểu thức của H: $H = \overline{JQ} + KQ$

Để thiết kế mạch đếm cụ thể ta sẽ xác định hàm H cho từng FF trong mạch, so sánh với biểu thức của hàm H suy ra J, K của các FF. Dưới đây là một thí dụ.

Thiết kế mạch đếm 10 đồng bộ dùng FF JK

Bảng trạng thái của mạch đếm 10 và giá trị của các hàm H tương ứng:

C_K	Q_D	Q_C	Q_B	Q_A	Q_D	Q_C	Q_B	Q_A	H_D	H_C	H_B	H_A
					+	+	+	+				
1 \downarrow	0	0	0	0	0	0	0	1	0	0	0	1
2 \downarrow	0	0	0	1	0	0	1	0	0	0	1	1
3 \downarrow	0	0	1	0	0	0	1	1	0	0	0	1
4 \downarrow	0	0	1	1	0	1	0	0	0	1	1	1
5 \downarrow	0	1	0	0	0	1	0	1	0	0	0	1
6 \downarrow	0	1	0	1	0	1	1	0	0	0	1	1
7 \downarrow	0	1	1	0	0	1	1	1	0	0	0	1
8 \downarrow	0	1	1	1	1	0	0	0	1	1	1	1

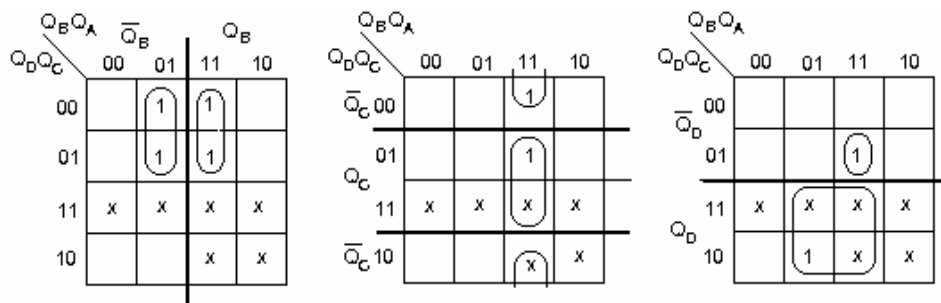
9↓	1	0	0	0	1	0	0	1	0	0	0	1
10↓	1	0	0	1	0	0	0	0	1	0	0	1

Bảng 5.17

Từ bảng 5.17, ta thấy:

$$H_A = 1 = Q_A + \overline{Q_A} \Rightarrow J_A = K_A = 1$$

Để xác định H_B , H_C và H_D ta phải vẽ bảng Karnaugh

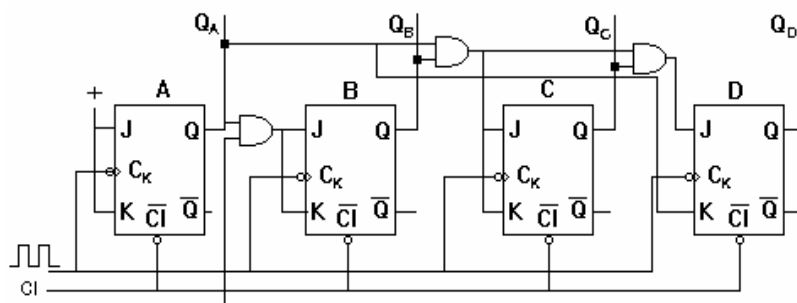


$$\begin{aligned} H_B &= \overline{Q_D} Q_A \overline{Q_B} + \overline{Q_D} Q_A Q_B & H_C &= Q_B Q_A \overline{Q_C} + Q_B Q_A Q_C & H_D &= Q_C Q_B Q_A \overline{Q_D} + Q_A Q_D \\ &\Rightarrow J_B = K_B = \overline{Q_D} Q_A & \Rightarrow J_C = K_C = Q_B Q_A & \Rightarrow J_D = Q_C Q_B Q_A, K_D = Q_A \end{aligned}$$

(H 5.19)

Ghi chú: Trong kết quả của hàm H ta muốn có chứa Q và \overline{Q} tương ứng để suy ra ngay các trị J và K nên ta đã chia bảng Karnaugh ra làm 2 phần chứa Q và \overline{Q} và nhóm riêng từng phần này.

Từ các kết quả này, ta vẽ được mạch (H 5.20)



(H 5.20)

Bây giờ ta có thể kiểm tra xem nếu như vì một lý do nào đó, số đếm rơi vào các trạng thái không sử dụng (tương ứng với số từ 10 đến 15) thì khi có xung đồng hồ trạng thái tiếp theo sẽ như thế nào? Mạch có quay về để đếm tiếp?

Áp dụng các hàm chuyển có được, ứng với mỗi trạng thái Q của từng FF trong các tổ hợp không sử dụng, ta tìm trị H tương ứng rồi suy ra Q_+ , ta được bảng kết quả sau:

C_K	Q_D	Q_C	Q_B	Q_A	H_D	H_C	H_B	H_A	Q_D +	Q_C +	Q_B +	Q_A +
↓	1	0	1	0	0	0	0	1	1	0	1	1
↓	1	0	1	1	1	1	0	1	0	1	1	0
↓	1	1	0	0	0	0	0	1	1	1	0	1
↓	1	1	0	1	1	0	0	1	0	1	0	0
↓	1	1	1	0	0	0	0	1	1	1	1	1
↓	1	1	1	1	0	1	0	1	0	0	1	0

Bảng 5.18

Từ bảng kết quả ta có kết luận:

- Khi ngã ra rơi vào trạng thái 10_{10} (1010), nó sẽ nhảy tiếp vào trạng thái 11_{10} (1011) rồi sau đó nhảy về 6_{10} (0110) (Dòng 1 và 2)
- Khi ngã ra rơi vào trạng thái 12_{10} (1100), nó sẽ nhảy tiếp vào trạng thái 13_{10} (11 01) rồi sau đó nhảy về 4_{10} (0100) (Dòng 3 và 4)
- Khi ngã ra rơi vào trạng thái 14_{10} (1110), nó sẽ nhảy tiếp vào trạng thái 15_{10} (1111) rồi sau đó nhảy về 2_{10} (0010) (Dòng 5 và 6).

Tóm lại, nếu có một sự cố xảy ra làm cho số đếm rơi vào các trạng thái không sử dụng thì sau 1 hoặc 2 số đếm nó tự động quay về một trong các số đếm từ 0 đến 9 rồi tiếp tục đếm bình thường.

★ Phương pháp MARCUS

Phương pháp MARCUS cho phép xác định các biểu thức của J và K dựa vào sự thay đổi của Q_+ so với Q

Từ bảng trạng thái của FF JK (Bảng 5.7) ta có thể viết lại Bảng 5.19:

Q	Q_+	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Bảng 5.19

Để thiết kế mạch, ta so sánh Q_+ và Q để có được bảng sự thật cho J, K của từng FF, sau đó xác định J và K.

Thí dụ thiết kế lại mạch đếm 10 bằng phương pháp MARCUS

Bảng sự thật cho J, K của từng FF

C_K	Q_D	Q_C	Q_B	Q_A	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A
1↓	0	0	0	0	0	x	0	x	0	x	1	x
2↓	0	0	0	1	0	x	0	x	1	x	x	1
3↓	0	0	1	0	0	x	0	x	x	0	1	x
4↓	0	0	1	1	0	x	1	x	x	1	x	1
5↓	0	1	0	0	0	x	x	0	0	x	1	x
6↓	0	1	0	1	0	x	x	0	1	x	x	1
7↓	0	1	1	0	0	x	x	0	x	0	1	x
8↓	0	1	1	1	1	x	x	1	x	1	x	1
9↓	1	0	0	0	x	0	0	x	0	x	1	x
10↓	1	0	0	1	x	1	0	x	0	x	x	1

Bảng 5.20

Ghi chú: Trong bảng 5.20, không có các cột cho Q_+ , tuy nhiên ta có thể thấy ngay là dòng bên dưới chính là Q_+ của dòng bên trên, như vậy kết quả có được từ sự so sánh dòng trên và dòng ngay dưới nó.

Ta thấy ngay $J_A = K_A = 1$

Dùng bảng Karnaugh để xác định các hàm còn lại

Nhận thấy các FF B và C có thể xác định chung cho J và K (cùng vị trí 1 và x), FF D được xác định J và K riêng

$Q_B Q_A$	$Q_D Q_C$	00	01	11	10
00		1	1		
01		1	1		
11	x	x	x	x	
10			x	x	

$$J_B = K_B = \overline{Q_D} Q_A$$

$Q_B Q_A$	$Q_D Q_C$	00	01	11	10
00			1		
01			1		
11	x	x	x	x	
10			x	x	

$$J_C = K_C = Q_B \cdot Q_A$$

$Q_B Q_A$	$Q_D Q_C$	00	01	11	10
00					
01			1		
11	x	x	x	x	
10	x	x	x	x	

$$J_D = Q_C \cdot Q_B \cdot Q_A$$

$Q_B Q_A$	$Q_D Q_C$	00	01	11	10
00	x	x	x	x	
01	x	x	x	x	
11	x	x	x	x	
10		1	x	x	

$$K_D = Q_A$$

(H 5.21)

Ta được lại kết quả trên.

Trên thị trường có khá nhiều IC đếm:

- 4 bit BCD: 74160, 74162, 74190, 74192, 4192, 4510, 4518. . .
- 4 bit nhị phân: 74161, 74163, 74191, 74193, 4193, 4516, 4520. . .
- 8 bit nhị phân: 74269, 74579, 74779. . .

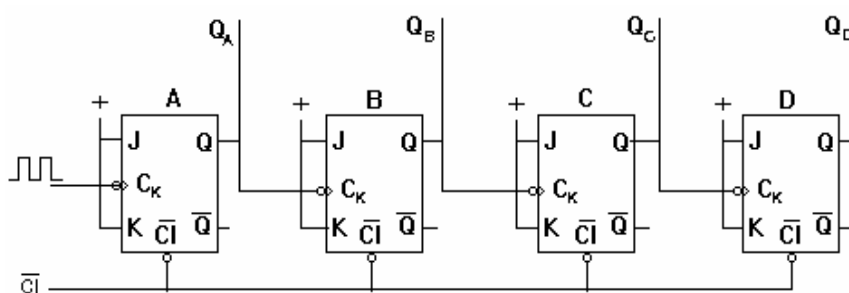
5.3.2 Mạch đếm không đồng bộ

Là các mạch đếm mà các FF không chịu tác động đồng thời của xung C_K .

Khi thiết kế mạch đếm không đồng bộ ta phải quan tâm tới **chiều tác động của xung đồng hồ C_K** .

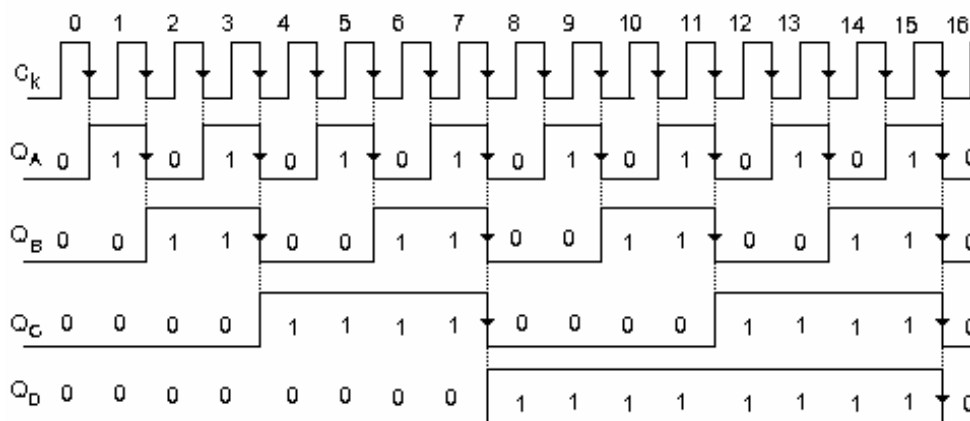
5.3.2.1. Mạch đếm không đồng bộ, n tầng, đếm lên (n=4):

Từ bảng trạng thái 5.14 của mạch đếm 4 bit, ta thấy nếu dùng FF JK tác động bởi **cạnh xuống của xung đồng hồ** thì có thể lấy ngõ ra của tầng trước làm xung đồng hồ C_K cho tầng sau, với điều kiện các ngõ vào JK của các FF đều được đưa lên mức cao. Ta được mạch đếm không đồng bộ, 4 bit, đếm lên (H 5.22).



(H 5.22)

(H 5.23) là dạng tín hiệu xung C_K và các ngõ ra của các FF



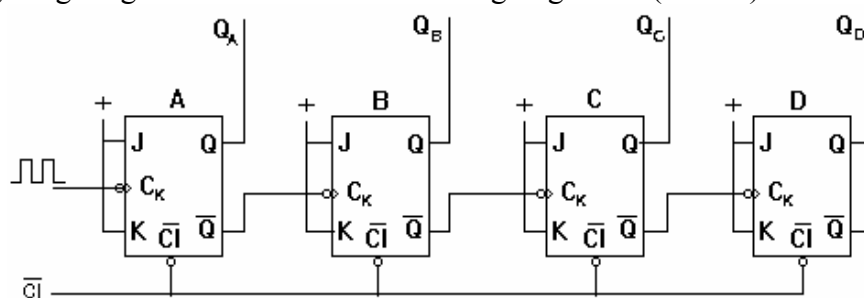
(H 5.23)

Tổ hợp các số tạo bởi các ngõ ra các FF D, C, B, A là số nhị phân từ 0 đến 15

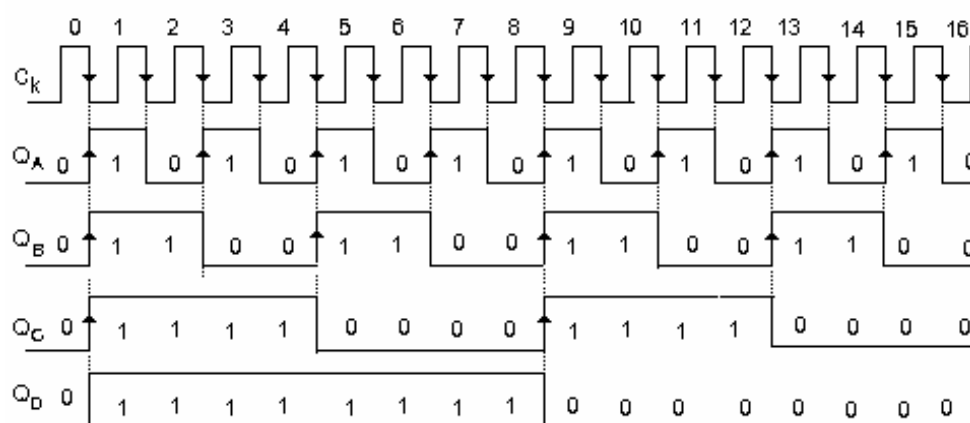
5.3.2.2. Mạch đếm không đồng bộ, n tầng, đếm xuống (n=4):

Để có mạch đếm xuống ta nối \bar{Q} (thay vì Q) của tầng trước vào ngõ vào C_K của tầng sau. (H 5.24) là mạch đếm xuống 4 tầng.

Dạng sóng ở ngõ ra các FF và số đếm tương ứng cho ở (H 5.25)



(H 5.24)



(H 5.25)

Quan sát tín hiệu ra ở các Flipflop ta thấy sau mỗi FF tần số của tín hiệu ra giảm đi một nửa, nghĩa là:

$$f_{Q_A} = \frac{f_{CK}}{2}$$

$$f_{Q_B} = \frac{f_{Q_A}}{2} = \frac{f_{CK}}{2^2} = \frac{f_{CK}}{4}$$

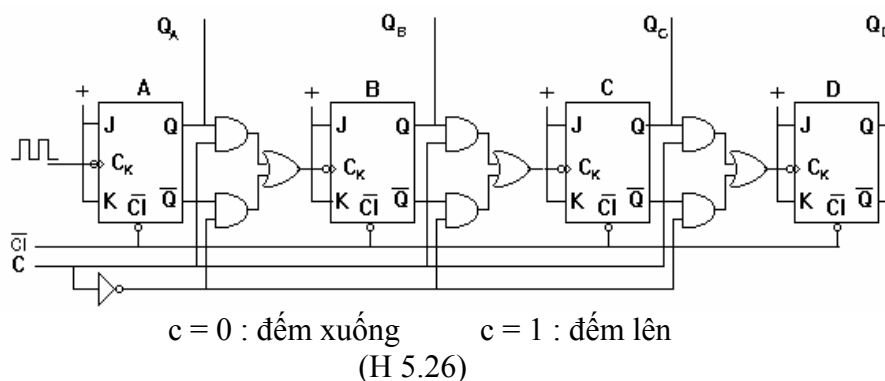
$$f_{Q_C} = \frac{f_{Q_A}}{4} = \frac{f_{CK}}{2^3} = \frac{f_{CK}}{8}$$

$$f_{Q_D} = \frac{f_{Q_A}}{8} = \frac{f_{CK}}{2^4} = \frac{f_{CK}}{16}$$

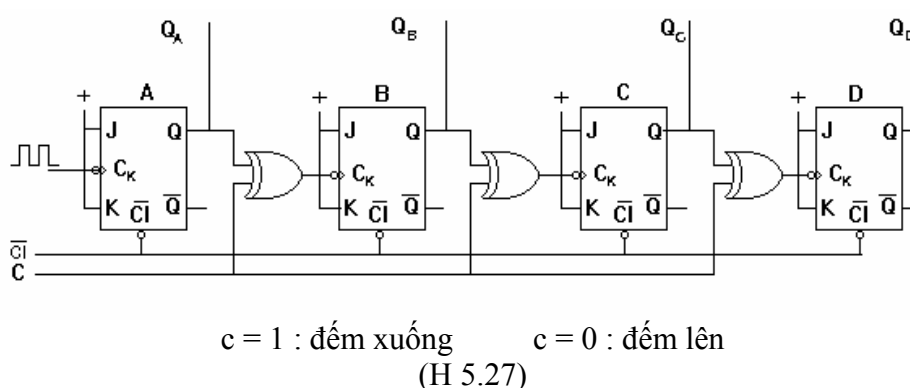
5.3.2.3. Mạch đếm không đồng bộ, n tầng, đếm lên, xuống (n=4):

Để có mạch đếm lên hoặc đếm xuống người ta dùng các mạch đa hợp 2→1 với ngõ vào điều khiển C chung để chọn Q hoặc \overline{Q} của tầng trước nối vào C_K tầng sau tùy theo yêu cầu về cách đếm.

Trong (H 5.26) , khi $C = 1$, Q nối vào C_K , mạch đếm lên và khi $C = 0$, \overline{Q} nối vào C_K , mạch đếm xuống



Trên thực tế, để đơn giản, ta có thể thay đa hợp 2→1 bởi một cổng EX-OR, ngõ điều khiển C nối vào một ngõ vào cổng EX-OR, ngõ vào còn lại nối với ngõ ra Q của FF và ngõ ra của cổng EX-OR nối vào ngõ vào C_K của FF sau, mạch cũng đếm lên/xuống tùy vào $C=0$ hay $C=1$.



5.3.2.4. Mạch đếm không đồng bộ modulo - N (N=10)

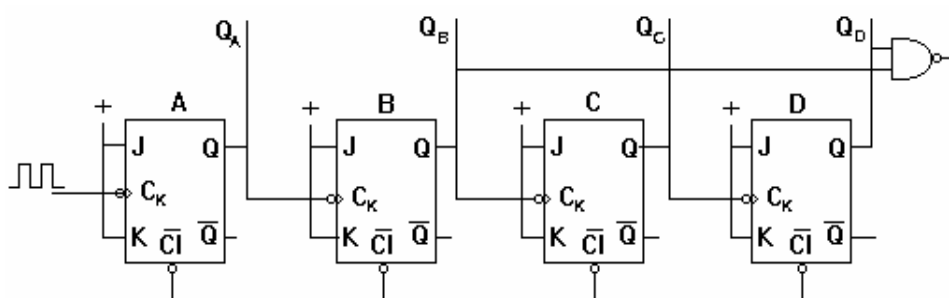
★ Kiểu Reset:

Để thiết kế mạch đếm kiểu Reset, trước nhất người ta lập bảng trạng thái cho số đếm (Bảng 5.21)

Quan sát bảng 5.21 ta thấy ở xung thứ 10, nếu theo cách đếm 4 tầng thì Q_D và Q_B phải lên 1. Lợi dụng hai trạng thái này ta dùng một cổng NAND 2 ngõ vào để đưa tín hiệu về xóa các FF, ta được mạch đếm ở (H 5.28).

Số xung C_K vào	Số Q_D	Nhi Q_C	Phân Q_B	Ra Q_A	Số thập phân tương ứng
Xóa	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	0(1)	0	0(1)	0	10

Bảng 5.21



(H 5.28)

Mạch đếm kiểu Reset có khuyết điểm như:

- Có một trạng thái trung gian trước khi đạt số đếm cuối cùng.
- Ngõ vào C_I không được dùng cho chức năng xóa ban đầu.

* Kiểu Preset:

Trong kiểu Preset các ngõ vào của các FF sẽ được đặt trước thế nào để khi mạch đếm đến trạng thái thứ N thì tất cả các FF tự động quay về không.

Để thiết kế mạch đếm không đồng bộ kiểu Preset, thường người ta làm như sau:

- Phân tích số đếm $N = 2^n \cdot N'$ ($N' < N$) rồi kết hợp hai mạch đếm n bit và N' . Việc thiết kế rất đơn giản khi số $N' \ll N$
- Quan sát bảng trạng thái và kết hợp với phương pháp thiết kế mạch đếm đồng bộ (MARCUS hay hàm chuyển) để xác định JK của các FF.

Thí dụ, để thiết kế mạch đếm 10, ta phân tích $10 = 2 \times 5$ và ta chỉ cần thiết kế mạch đếm 5 rồi kết hợp với một FF (đếm 2)

Bảng trạng thái của mạch đếm 5.

Số xung C_K vào	Số Nhi Q_D	Phân Q_C	Ra Q_B	Số thập phân tương ứng
Xóa	0	0	0	0

1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	0	0	0	0

Bảng 5.22

Giả sử dùng FF JK có xung C_K tác động cạnh xuống.

Từ bảng 5.21, ta thấy có thể dùng tín hiệu ngõ ra FF B làm xung đồng hồ cho FF C và đưa J_C và K_C lên mức cao:

$$C_{KC} = Q_B; \quad J_C = K_C = 1$$

Các FF B và D sẽ dùng xung C_K của hệ thống và các ngõ vào JK được xác định nhờ hàm chuyển:

C_K	Q_D	Q_C	Q_B	H_D	H_B
1↓	0	0	0	0	1
2↓	0	0	1	0	1
3↓	0	1	0	0	1
4↓	0	1	1	1	1
5↓	1	0	0	1	0
	0	0	0		

Bảng 5.23

Dùng bảng Karnaugh xác định H_D và H_B rồi suy ra các trị J, K của các FF.

$$H_D = Q_C \cdot Q_B \cdot \overline{Q_D} + Q_D$$

$$\Rightarrow J_D = Q_C \cdot Q_B; \quad K_D = 1$$

$$H_B = \overline{Q_D} \cdot \overline{Q_B} + Q_B$$

$$\Rightarrow J_B = \overline{Q_D}; \quad K_B = 1$$

(H 5.29)

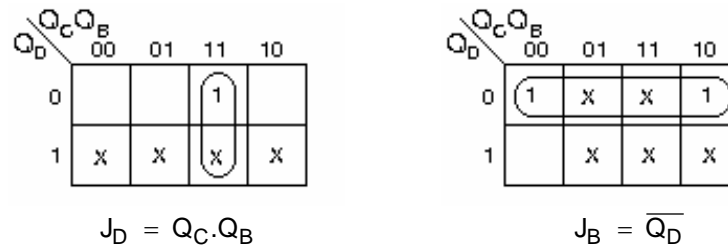
Có thể xác định J, K của các FF B và D bằng phương pháp MARCUS:

C_K	Q_D	Q_C	Q_B	J_D	K_D	J_B	K_B
1↓	0	0	0	0	x	1	x
2↓	0	0	1	0	x	x	1
3↓	0	1	0	0	x	1	x
4↓	0	1	1	1	x	x	1
5↓	1	0	0	x	1	0	x
	0	0	0				

Bảng 5.24

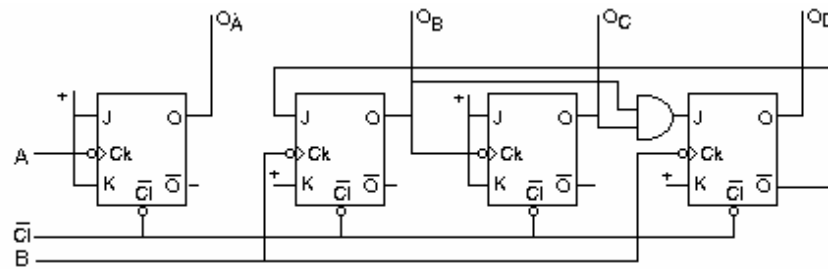
Ta có ngay $K_D = K_B = 1$

Dùng bảng Karnaugh xác định J_D và J_B



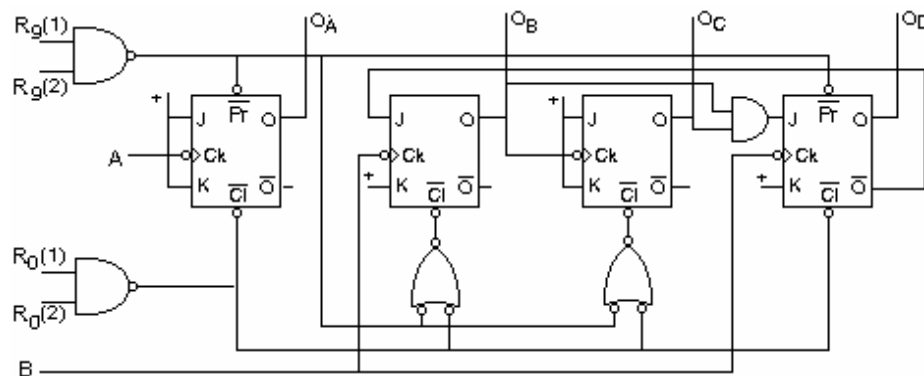
(H 5.30)

(H 5.31) là mạch đếm 10 thiết kế theo kiểu đếm 2x5 với mạch đếm 5 có được từ kết quả trên.



(H 5.31)

IC 7490 là IC đếm 10, có cấu tạo như mạch (H 5.31) thêm các ngõ vào Reset 0 và Reset 9 có sơ đồ mạch (H 5.32)



(H 5.32)

Bảng 5.25 là bảng sự thật cho các ngõ vào Reset

Reset Inputs				Outputs			
R ₀ (1)	R ₀ (2)	R ₉ (1)	R ₉ (2)	Q _D	Q _C	Q _B	Q _A
1	1	0	x	0	0	0	0
1	1	x	0	0	0	0	0
0	x	1	1	1	0	0	1
x	0	1	1	1	0	0	1
x	0	x	0	Đếm	Đếm	Đếm	Đếm
0	x	0	x	nt	nt	nt	nt
0	x	x	0	nt	nt	nt	nt
x	0	0	x	nt	nt	nt	nt

Bảng 5.25

Dùng IC 7490, có thể thực hiện một trong hai cách mắc:

- ★ Mạch đếm 2x5: Nối Q_A vào ngõ vào B, xung đếm (C_K) vào ngõ vào A

★ Mạch đếm 5x2: Nối Q_D vào ngõ vào A, xung đếm (C_K) vào ngõ vào B

Hai cách mắc cho kết quả số đếm khác nhau nhưng cùng một chu kỳ đếm 10. Tần số tín hiệu ở ngõ ra sau cùng bằng $1/10$ tần số xung C_K (nhưng dạng tín hiệu ra khác nhau).

Dưới đây là hai bảng trạng thái cho hai trường hợp nói trên.

Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

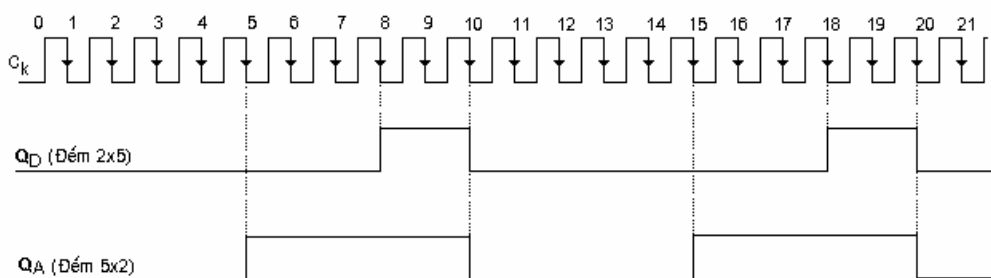
Bảng 5.26 : Đếm 2x5

Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1

Bảng 5.27 : Đếm 5x2

(H 5.33) cho thấy dạng sóng ở các ngõ ra của hai mạch cùng đếm 10 nhưng hai kiểu đếm khác nhau:

- Kiểu đếm 2x5 cho tín hiệu ra ở Q_D không đối xứng
- Kiểu đếm 5x2 cho tín hiệu ra ở Q_A đối xứng



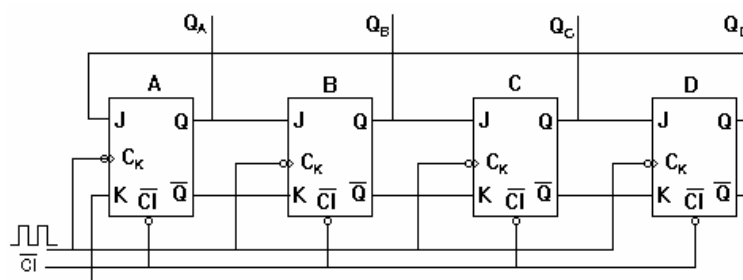
(H 5.33)

5.3.3 Mạch đếm vòng

Thực chất là mạch ghi dịch trong đó ta cho hồi tiếp từ một ngõ ra nào đó về ngõ vào để thực hiện một chu kỳ đếm. Tùy đường hồi tiếp mà ta có các chu kỳ đếm khác nhau

Sau đây ta khảo sát vài loại mạch đếm vòng phổ biến.

5.3.3.1. Hồi tiếp từ Q_D về J_A và $\overline{Q_D}$ về K_A



(H 5.34)

Đối với mạch này, sự đếm vòng chỉ thấy được khi có đặt trước ngõ ra

- Đặt trước $Q_A = 1$, ta được kết quả như bảng 5.28.

C_K	Q_D	Q_C	Q_B	Q_A	Số TP
Preset	0	0	0	1	1
1↓	0	0	1	0	2
2↓	0	1	0	0	4
3↓	1	0	0	0	8
4↓	0	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮

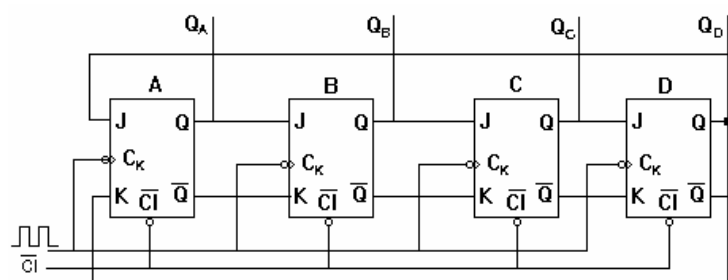
Bảng 5.28

Nếu đặt trước $Q_A = Q_B = 1$ ta có bảng 5.29

C_K	Q_D	Q_C	Q_B	Q_A	Số TP
Preset	0	0	1	1	3
1↓	0	1	1	0	6
2↓	1	1	0	0	12
3↓	1	0	0	1	9
4↓	0	0	1	1	3
⋮	⋮	⋮	⋮	⋮	⋮

Bảng 5.29

5.3.3.2. Hồi tiếp từ \overline{Q}_D về J_A và Q_D về K_A (H 5.35)



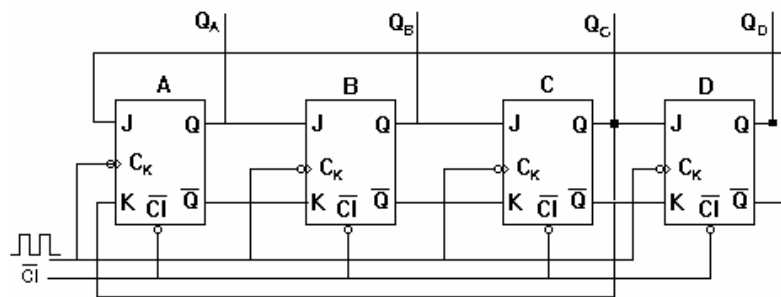
(H 5.35)

Mạch này còn có tên là mạch đếm Johnson. Mạch có một chu kỳ đếm mặc nhiên mà không cần đặt trước và nếu có đặt trước, mạch sẽ cho các chu kỳ khác nhau tùy vào tổ hợp đặt trước đó. Bảng 5.30 là chu kỳ đếm mặc nhiên.

C_K	Q_D	Q_C	Q_B	Q_A	Số TP
Preset	0	0	0	0	0
1↓	0	0	0	1	1
2↓	0	0	1	1	3
3↓	0	1	1	1	7
4↓	1	1	1	1	15
5↓	1	1	1	0	14
6↓	1	1	0	0	12
7↓	1	0	0	0	8
8↓	0	0	0	0	0

Bảng 5.30

5.3.3.3. Hồi tiếp từ \overline{Q}_D về J_A và Q_C về K_A (H 5.36)



(H 5.36)

C_K	Q_D	Q_C	Q_B	Q_A	Số TP
Preset	0	0	0	0	0
1↓	0	0	0	1	1
2↓	0	0	1	1	3
3↓	0	1	1	1	7
4↓	1	1	1	0	14
5↓	1	1	0	0	12
6↓	1	0	0	0	8
7↓	0	0	0	0	0

Bảng 5.31

Vài thí dụ thiết kế mạch đếm

1. Dùng FF JK thiết kế mạch đếm 6, đồng bộ

Bảng trạng thái và hàm chuyển mạch đếm 6:

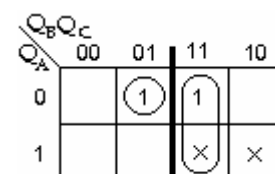
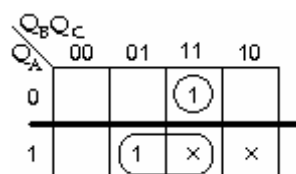
N	Q_A	Q_B	Q_C	Q_{A+}	Q_{B+}	Q_{C+}	H_A	H_B	H_C
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
2	0	1	0	0	1	1	0	0	1
3	0	1	1	1	0	0	1	1	1
4	1	0	0	1	0	1	0	0	1
5	1	0	1	0	0	0	1	0	1

Bảng 5.32

$$H_C = 1 \Rightarrow J_C = K_C = 1$$

Xác định J_A, K_A, J_B, K_B

Bảng Karnaugh cho hai hàm chuyển H_A & H_B



(H 5.37)

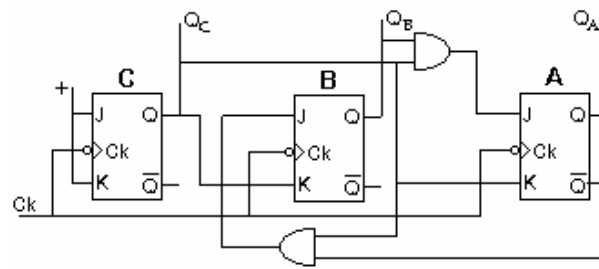
$$H_A = Q_B Q_C \overline{Q}_A + Q_C Q_A$$

$$\Rightarrow J_A = Q_B Q_C ; K_A = Q_C$$

$$H_B = \overline{Q}_A Q_C \overline{Q}_B + Q_C Q_B$$

$$\Rightarrow J_B = \overline{Q}_A Q_C ; K_B = Q_C$$

Mạch:



(H 5.38)

2. Thiết kế mạch đếm 7 không đồng bộ, dùng FF JK có ngõ vào xung đồng hồ tác động bởi cạnh lên của C_K .

Bảng trạng thái

N	Q_A	Q_B	Q_C	J_B	K_B	J_C	K_C
0↑	0	0	0	0	x	1	x
1↑	0	0	1	1	x	x	1
2↑	0	1	0	x	0	1	x
3↑	0	1	1	x	1	x	1
4↑	1	0	0	0	x	1	x
5↑	1	0	1	1	x	x	1
6↑	1	1	0	x	1	0	x
	0	0	0				

Bảng 5.33

Nhận xét bảng trạng thái ta thấy mỗi lần Q_B thay đổi từ 1 xuống 0 thì Q_A đổi trạng thái, mà FF có xung đồng hồ tác động bởi cạnh lên nên ta có thể lấy \overline{Q}_B làm xung đồng hồ cho FFA và $J_A = K_A = 1$.

FF B và FFC sẽ dùng xung đồng hồ hệ thống, dùng phương pháp MARCUS để xác định J & K của các FF này.

Ta thấy ngay $K_C = 1$

$Q_A \backslash Q_B Q_C$	00	01	11	10
0		1	x	x
1		1	x	x

$Q_A \backslash Q_B Q_C$	00	01	11	10
0	x	x	1	
1	x	x	x	1

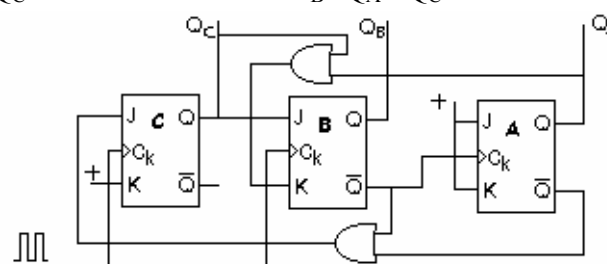
$Q_A \backslash Q_B Q_C$	00	01	11	10
0	1	x	x	1
1	1	x	x	

(H 5.39)

$$J_B = Q_C$$

$$K_B = Q_A + Q_C$$

$$J_C = \overline{Q}_A + \overline{Q}_B$$



(H 5.40)

BÀI TẬP

1. Thiết kế bộ đếm đồng bộ có dãy đếm sau: 000, 010, 101, 110 và lặp lại.
2. Làm lại bài 1. Thêm điều kiện các trạng thái không sử dụng 001, 011, 100 và 111 phải luôn luôn nhảy về 000 ở xung đồng hồ kế tiếp.
3. Thiết kế bộ đếm đồng bộ dùng FF-JK với dãy đếm sau: 000, 001, 011, 010, 110, 111, 101, 100, 000 . . .
4.
 - a. Thiết kế một mạch đếm đồng bộ dùng FF-JK tác động cạnh xuống, có dãy đếm như sau: 000, 001, 011, 111, 110, 100, 001. . . Những trạng thái không sử dụng được đưa về trạng thái 000 ở xung đồng hồ kế tiếp. Vẽ sơ đồ mạch.
 - b. Mắc nối tiếp một bộ đếm 2 (Dùng FF-JK, tác động cạnh xuống) với bộ đếm đã được thiết kế ở câu a. Vẽ dạng sóng ở các ngõ ra của bộ đếm giả sử trạng thái ban đầu của các ngõ ra đều bằng 0. Xác định dãy đếm của mạch.
5. Thiết kế mạch đếm đồng bộ modulo-12 dùng FF JK.
Dùng ngõ ra mạch đếm để điều khiển hệ thống đèn giao thông:
 - Đèn xanh cháy trong 40 s
 - Đèn vàng cháy trong 20s
 - Đèn đỏ cháy trong 40s
 - Đèn vàng và đỏ cùng cháy trong 20s. Chu kỳ lặp lạiCho chu kỳ xung đồng hồ là 10s.
6. Thiết kế mạch đếm đồng bộ dùng FF JK có ngõ vào điều khiển X:
 - Khi X=0 mạch đếm theo thứ tự 0, 2, 4, 6 rồi trở về 0
 - Khi X=1 mạch đếm 0, 6, 4, 2 rồi trở về 0.Các trạng thái không sử dụng trong hai lần đếm đều trở về 0 khi có xung đồng hồ