Chương 5

HỆ TUẦN TỰ

5.1. KHÁI NIỆM CHUNG

Mạch số được chia thành hai loại chính : Hệ tổ hợp và hệ tuần tự.

Đối với hệ tổ hợp: tín hiệu ngõ ra ở trạng thái kế tiếp chỉ phụ thuộc vào trạng thái hiện tại của ngõ vào, mà bất chấp trạng thái hiện tại của ngõ ra. Như vậy, khi các ngõ vào thay đổi trạng thái (bỏ qua thời gian trễ của tín hiệu đi qua phần tử logic) thì lập tức ngõ ra thay đổi trạng thái.

Đối với hệ tuần tự: Các ngõ ra ở trạng thái kế tiếp vừa phụ thuộc vào trạng thái hiện tại của ngõ vào, đồng thời còn phụ thuộc trạng thái hiện tại của ngõ ra.

Do đó, vấn đề thiết kế hệ tuần tự sẽ khác so với hệ tổ hợp và cơ sở thiết kế hệ tuần tự là dựa trên các Flip - Flop (trong khi việc thiết kế hệ tổ hợp dựa trên các cổng logic).

Mặc khác, đối với hệ tuần tự, khi các ngõ vào thay đổi trạng thái thì các ngõ ra không thay đổi trạng thái ngay mà chờ đến cho đến khi có một xung điều khiển (gọi là xung đồng hồ Ck) thì lúc đó các ngõ ra mới thay đổi trạng thái theo các ngõ vào. Như vậy hệ tuần tự còn có tính đồng bộ và tính nhớ (có khả năng lưu trữ thông tin, lưu trữ dữ liệu), nên hệ tuần tự là cơ sở để thiết kế các bộ nhớ.

5.2. BỘ ĐẾM

5.2.1. Đại cương

Bộ đếm được xây dựng trên cơ sở các Flip - Flop (FF) ghép với nhau sao cho hoạt động theo một bảng trạng thái (qui luật) cho trước.

Số lượng FF sử dụng là số hàng của bộ đếm.

Bộ đếm còn được sử dụng để tạo ra một dãy địa chỉ của lệnh điều kiển, đếm số chu trình thực hiện phép tính, hoặc có thể dùng trong vấn đề thu và phát mã.

Có thể phân loại bộ đếm theo nhiều cách:

- Phân loại theo cơ sổ các hệ đếm: Bộ đếm thập phân, bộ đếm nhị phân.

Trong đó bộ đếm nhị phân được chia làm hai loại:

+ Bộ đếm với dung lượng đếm 2^n.

+ Bộ đếm với dung lượng đếm khác 2^n (đếm modulo M).

- Phân loại theo hướng đếm gồm: Mạch đếm lên (đếm tiến), mạch đếm xuống (đếm lùi), mạch đếm vòng.
- Phân loại mạch đếm theo tín hiệu chuyển: bộ đếm nối tiếp, bộ đếm song song, bộ đếm hỗn hợp.
- Phân loại dựa vào chức năng điều khiển:

+ Bộ đếm đồng bộ: Sự thay đổi ngõ ra phụ thuộc vào tín hiệu điều kiển Ck.

+ Bộ đếm không đồng bộ.

Mặc dù có rất nhiều cách phân loại nhưng chỉ có ba loại chính: Bộ đếm nối tiếp (không đồng bộ), Bộ đếm song song (đồng bộ), Bộ đếm hỗn hợp.

5.2.2. Bộ đếm nối tiếp

5.2.2.1. Khái niệm

Bộ đếm nối tiếp là 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. Đối với loại bộ đếm này, các ngõ ra thay đổi trạng thái không đồng thời với tín hiệu điều khiển Ck (tức không chịu sự điều khiển của tín hiệu điều khiển Ck) do đó mạch đếm nối tiếp còn gọi là mạch đếm không đồng bộ.

5.2.2.2. Phân loại

- Đếm lên.
- Đếm xuống.
- Đếm lên /xuống.
- Modulo M.

a. Đếm lên

Đây là bộ đếm có nội dung đếm tăng dần. Nguyên tắc ghép nối các TFF (hoặc JKFF thực hiện chức năng TFF) để tạo thành bộ đếm nối tiếp còn phụ thuộc vào tín hiệu điều khiển Ck. Có 2 trường hợp khác nhau:

- Tín hiệu Ck tác động sườn xuống: TFF hoặc JKFF được nghép nối với nhau theo qui luật sau:

$$\mathbf{Ck_{i+1} = Q_i}$$

- Tín hiệu Ck tác động sường xuống: TFF hoặc JKFF được nghép nối với nhau theo qui luật sau:

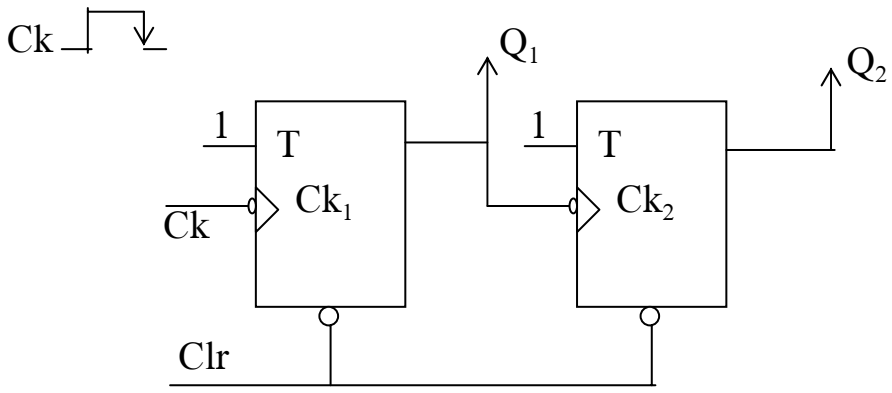
$$\mathbf{Ck_{i+1} = \overline{Q_i}}$$

Trong đó T luôn luôn giữ ở mức logic 1 (T = 1) và ngõ ra của TFF đứng trước nối với ngõ vào Ck của TFF đứng sau.

Để minh họa chúng ta xét ví dụ về một mạch đếm nối tiếp, đếm 4, đếm lên, dùng TFF.

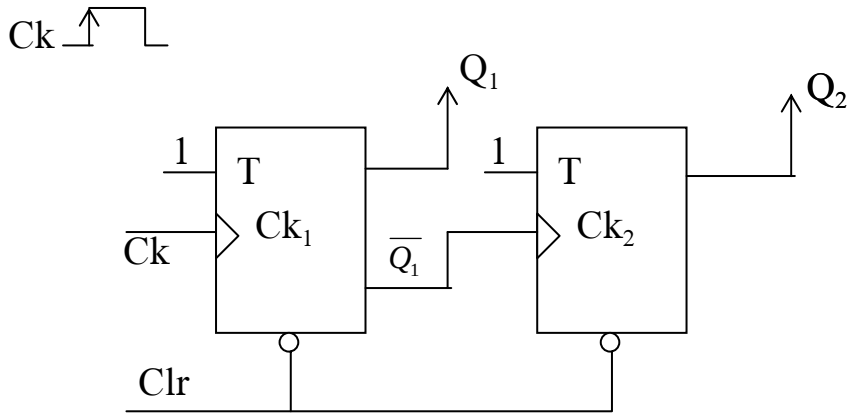
Số lượng TFF cần dùng: $4 = 2^2 \Rightarrow$ dùng 2 TFF.

Trường hợp Ck tác động theo sườn xuống (hình 5.1a):



Hình 5.1a

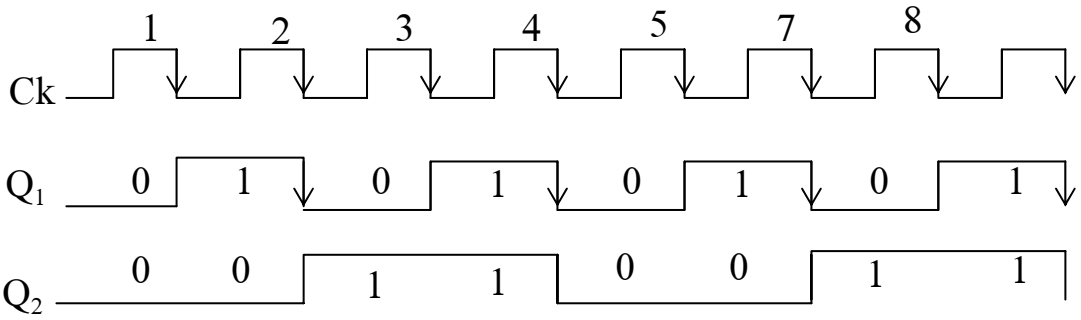
Trường hợp Ck tác động theo sườn lên (hình 5.1b):



H 5.1b

Trong các sơ đồ mạch này Clr (Clear) là ngõ vào xóa của TFF. Ngõ vào Clr tác động mức thấp, khi Clr = 0 thì ngõ ra Q của FF bị xóa về 0 (Q=0).

Giản đồ thời gian của mạch ở hình 5.1a :

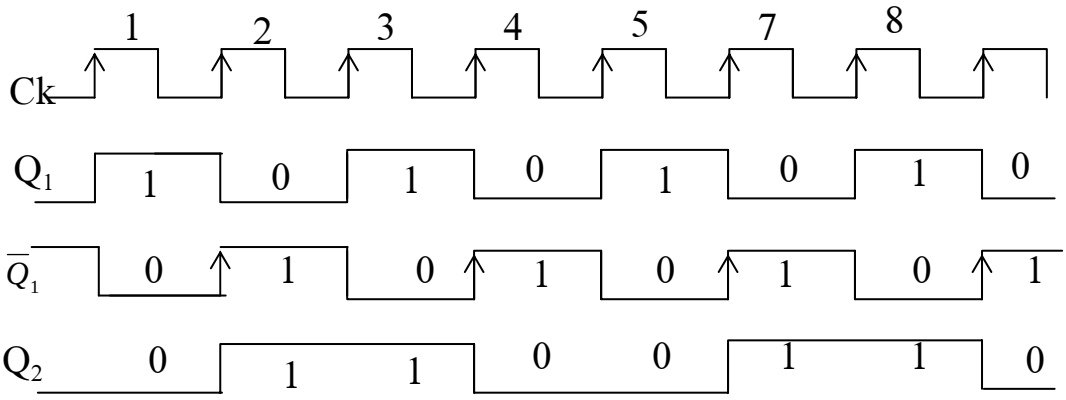


Hình 5.2a. Giản đồ thời gian mạch hình 5.1a

Bảng trạng thái hoạt động của mạch hình 5.1a:

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

Giản đồ thời gian mạch hình 5.1b :



Hình 5.2b. Giản đồ thời gian mạch hình 5.1b

Bảng trạng thái hoạt động của mạch hình 5.1b :

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

b. Đếm xuống

Đây là bộ đếm có nội dung đếm giảm dần. Nguyên tắc ghép các FF cũng phụ thuộc vào tín hiệu điều khiển Ck:

- Tín hiệu Ck tác động sườn xuống: TFF hoặc JKFF được nghép nối với nhau theo qui luật sau:

$$\mathbf{Ck_{i+1} = \overline{Q_i}}$$

- Tín hiệu Ck tác động sườn xuống: TFF hoặc JKFF được nghép nối với nhau theo qui luật sau:

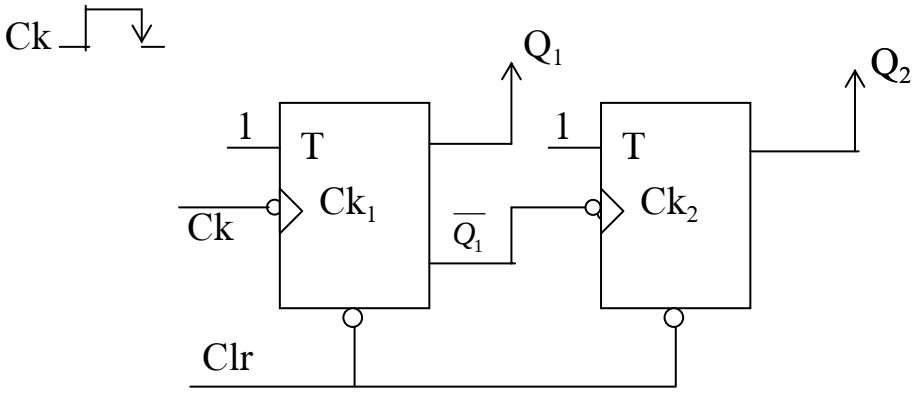
$$\mathbf{Ck_{i+1} = Q_i}$$

Trong đó T luôn luôn giữ ở mức logic 1 (T = 1) và ngõ ra của TFF đứng trước nối với ngõ vào Ck của TFF đứng sau.

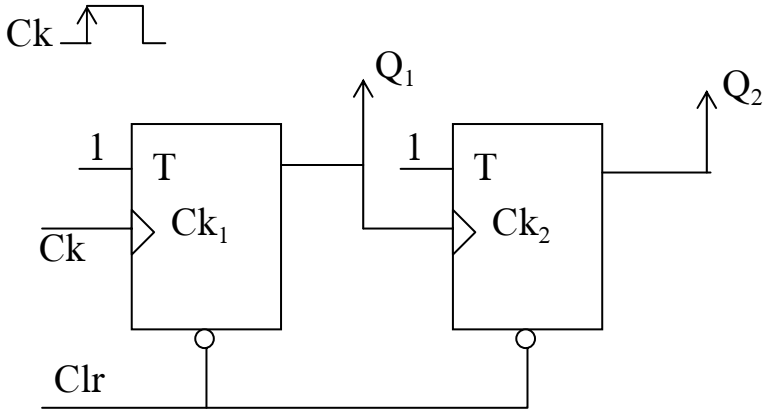
Ví dụ: Xét một mạch đếm 4, đếm xuống, đếm nối tiếp dùng TFF.

Số lượng TFF cần dùng: $4 = 2^2 \Rightarrow$ dùng 2 TFF.

Sơ đồ mạch thực hiện khi sử dụng Ck tác động sườn xuống và Ck tác động sườn lên lần lượt được cho trên hình 5.3a và 5.3b :

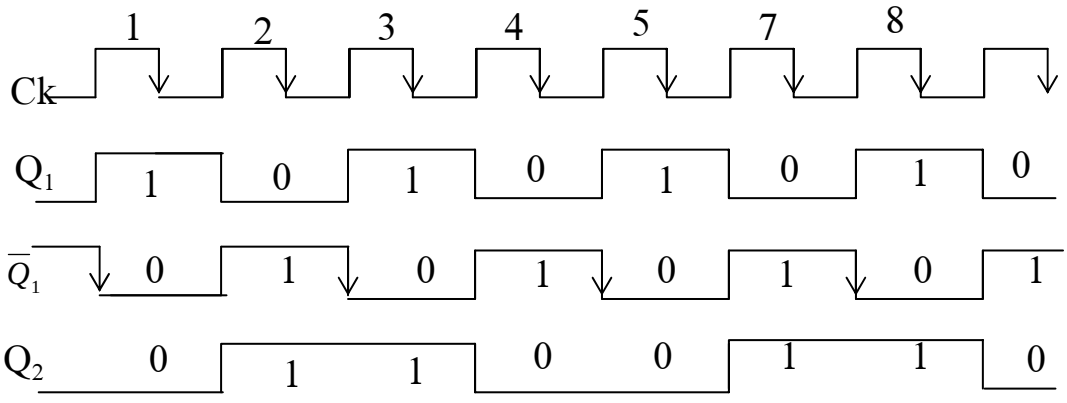


Hình 5.3a



H 5.3b

Giản đồ thời gian của mạch hình 5.3a :

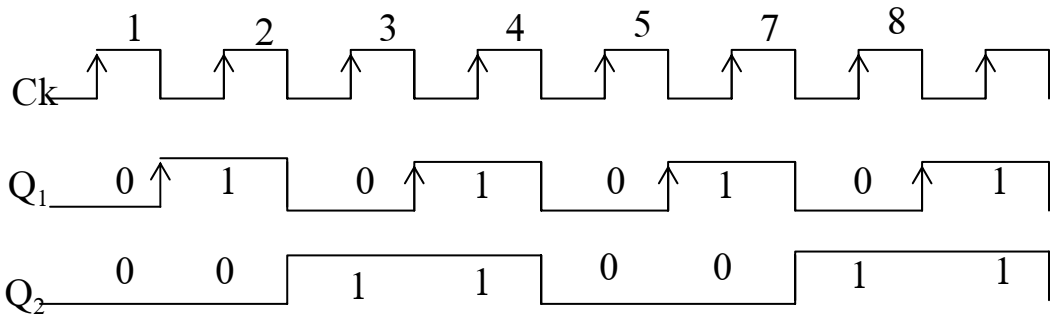


Hình 5.4a. Giản đồ thời gian mạch 5.3a

Bảng trạng thái hoạt động của mạch hình 5.3a:

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

Giản đồ thời gian của mạch hình 5.3b:



Hình 5.4b. Giản đồ thời gian mạch hình 5.3b

Bảng trạng thái hoạt động của mạch hình 5.3b :

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

c. Đếm lên/xuống:

Gọi X là tín hiệu điều khiển chiều đếm, ta quy ước:

+ Nếu X = 0 thì mạch đếm lên.
+ Nếu X = 1 thì đếm xuống.

Ta xét 2 trường hợp của tín hiệu Ck:

- Xét tín hiệu Ck tác động sườn xuống:

  Lúc đó ta có phương trình logic:

$$Ck_{i+1} = \overline{X}.Q_i + X\overline{Q_i} = X \oplus Q_i$$

- Xét tín hiệu Ck tác động sườn lên:

  Lúc đó ta có phương trình logic:

$$Ck_{i+1} = \overline{X}.\overline{Q_i} + X.Q_i = \overline{X \oplus Q_i}$$

d. Đếm modulo M:

Đây là bộ đếm nối tiếp, theo mã BCD 8421, có dung lượng đếm khác 2^n.

Ví dụ: Xét mạch đếm 5, đếm lên, đếm nối tiếp.

Số lượng TFF cần dùng: Vì $2^2 = 4 < 5 < 8 = 2^3 \Rightarrow$ dùng 3 TFF.

Vậy bộ đếm này sẽ có 3 đầu ra (chú ý: Số lượng FF tương ứng với số đầu ra).

Bảng trạng thái hoạt động của mạch:

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

Nếu dùng 3 FF thì mạch có thể đếm được 8 trạng thái phân biệt (000 $\rightarrow$ 111 tương ứng 0$\rightarrow$7). Do đó, để sử dụng mạch này thực hiện đếm 5, đếm lên, thì sau xung Ck thứ 5 ta tìm cách đưa tổ hợp 101 về 000 có nghĩa là mạch thực hiện việc đếm lại từ tổ hợp ban đầu. Như vậy, bộ đếm sẽ đếm từ 000 $\rightarrow$ 100 và quay về 000 trở lại, nói cách khác ta đã đếm được 5 trạng thái phân biệt.

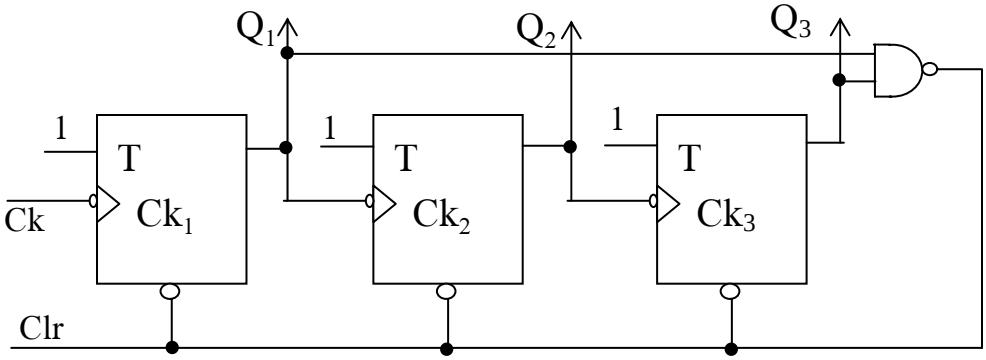
Để xóa bộ đếm về 000 ta phân tích: Do tổ hợp 101 có 2 ngõ ra Q_1, Q_3 đồng thời bằng 1 (khác với các tổ hợp trước đó) $\rightarrow$ đây chính là dấu hiệu nhận biết để điều khiển xóa bộ đếm. Vì vậy để xóa bộ đếm về 000:

- 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 có 2 ngõ vào.

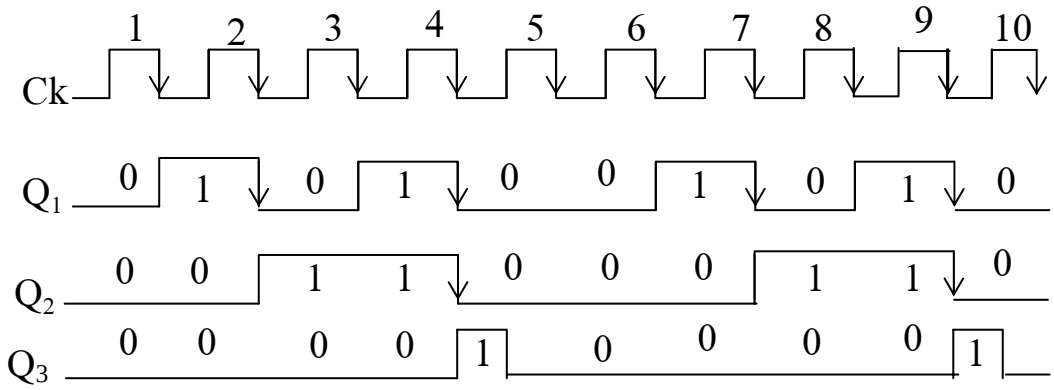
Như vậy sơ đồ mạch đếm 5 là sơ đồ cải tiến từ mạch đếm 8 bằng cách mắc thêm phần tử cổng NAND (hoặc cổng AND) có hai ngõ vào (tùy thuộc vào chân Clr tác động mức logic 0 hay mức logic 1) được nối đến ngõ ra Q_1 và Q_3, và ngõ ra của cổng NAND (hoặc AND) sẽ được nối đến ngõ vào Clr của bộ đếm (cũng chính là ngõ vào Clr của các FF).

Trong trường hợp Clr tác động mức thấp sơ đồ mạch thực hiện đếm 5 như trên hình 5.5 :



Hình 5.5. Mạch đếm 5, đếm lên

Giản đồ thời gian của mạch:

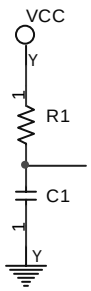


Hình 5.6. Giản đồ thời gian mạch đếm 5, đếm lên.

Chú ý:

Do trạng thái của ngõ ra là không biết trước nên để mạch có thể đếm từ trạng thái ban đầu là 000 ta phải dùng thêm mạch xóa tự động ban đầu để xóa bộ đếm về 0 (còn gọi là mạch RESET ban đầu). Phương pháp thực hiện là dùng hai phần tử thụ động R và C.

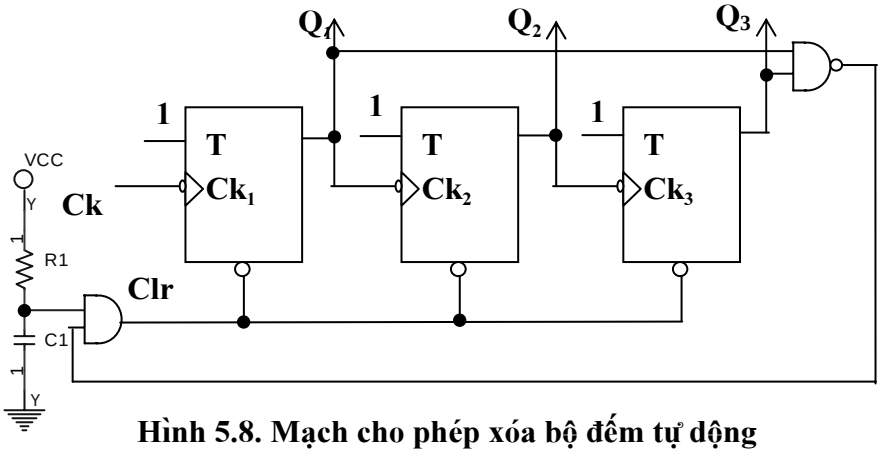
Trên hình 5.7 là mạch Reset mức 0 (tác động mức 0). Mạch hoạt động như sau: Do tính chất điện áp trên tụ C không đột biến được nên ban đầu mới cấp nguồn Vcc thì $V_C = 0 \Rightarrow$ ngõ ra Clr = 0 và mạch có tác động Reset xóa bộ đếm, sau đó tụ C được nạp điện từ nguồn qua điện trở R với thời hằng nạp là $\tau = RC$ nên điện áp trên tụ tăng dần, cho đến khi tụ C nạp đầy thì điện áp trên tụ xấp xỉ bằng Vcc $\Rightarrow$ ngõ ra Clr = 1, mạch không còn tác dụng reset.



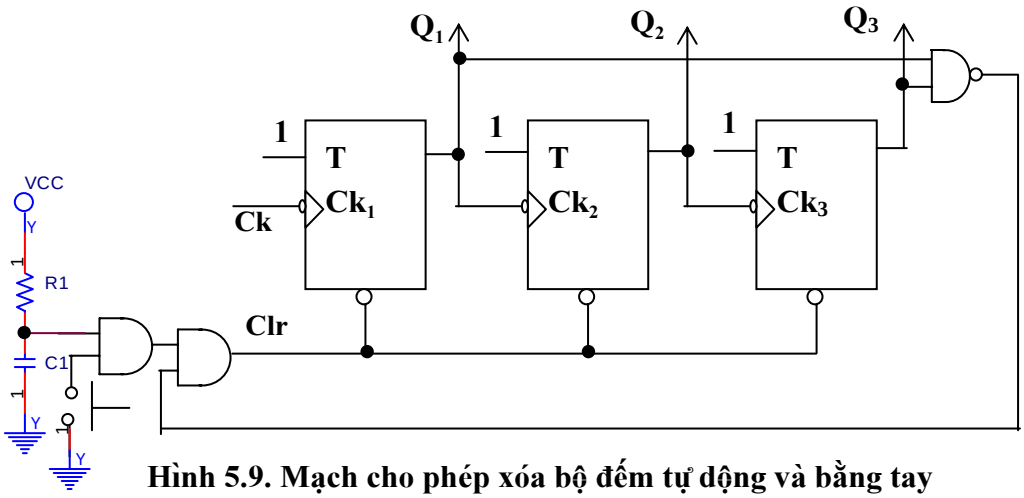
Hình 5.7. Mạch Reset mức 0

Chú ý khi thiết kế: Với một FF, ta biết được thời gian xóa (có trong Datasheet do nhà sản xuất cung cấp), do đó ta phải tính toán sao cho thời gian tụ C nạp điện từ giá trị ban đầu đến giá trị điện áp ngưỡng phải lớn hơn thời gian xóa cho phép thì mới đảm bảo xóa được các FF.

Mạch cho phép xóa bộ đếm tự động (H 5.8) và bằng tay (H 5.9):



Hình 5.8. Mạch cho phép xóa bộ đếm tự động



Hình 5.9. Mạch cho phép xóa bộ đếm tự động và bằng tay

Ưu điểm của bộ đếm nối tiếp: Đơn giản, dễ thiết kế.

Nhược điểm: Với dung lượng đếm lớn, số lượng FF sử dụng càng nhiều thì thời gian trễ tích lũy khá lớn. Nếu thời gian trễ tích lũy lớn hơn một chu kỳ tín hiệu xung kích thì lúc bấy giờ kết quả đếm sẽ sai. Do đó, để khắc phục nhược điểm này, người ta sử dụng bộ đếm song song.

5.2.3. Bộ đếm song song

5.2.3.1. Khái niệm

Bộ đếm song song là bộ đếm trong đó các FF mắc song song với nhau và các ngõ ra sẽ thay đổi trạng thái dưới sự điều khiển của tín hiệu Ck. Chính vì vậy mà người ta còn gọi bộ đếm song song là bộ đếm đồng bộ.

Mạch đếm song song được sử dụng với bất kỳ FF loại nào và có thể đếm theo qui luật bất kỳ cho trước. Vì vậy, để thiết kế bộ đếm đồng bộ (song song) người ta dựa vào các bảng đầu vào kích của FF.

5.2.3.2. Mạch thực hiện

Đối với bộ đếm song song dù đếm lên hay đếm xuống, hoặc là đếm Modulo M (đếm lên/đếm xuống) đều có cách thiết kế chung và không phụ thuộc vào tín hiệu Ck tác động sườn lên, sườn xuống, mức 0 hay mức 1.

Các bước thực hiện :

- Từ yêu cầu thực tế xây dựng bảng trạng thái hoạt động của mạch.
- Dựa vào bảng đầu vào kích của FF tương ứng để xây dựng các bảng hàm giá trị của các ngõ vào dữ liệu (DATA) theo ngõ ra.
- Dùng các phương pháp tối thiểu để tối thiểu hóa các hàm logic trên.
- Thành lập sơ đồ logic.

Ví dụ: Thiết kế mạch đếm đồng bộ, đếm 5, đếm lên theo mã BCD 8421 dùng JKFF.

Trước hết xác định số JKFF cần dùng: Vì $2^2 = 4 < 5 < 8 = 2^3 \Rightarrow$ dùng 3 JKFF $\Rightarrow$ có 3 ngõ ra Q_1, Q_2, Q_3.

Ta có bảng trạng thái mô tả hoạt động của mạch như sau:

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

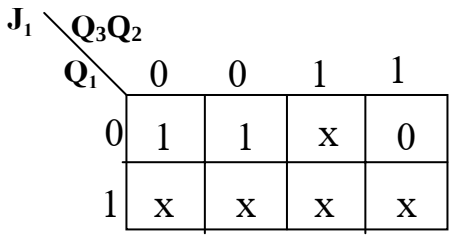
Ở chương 3 chúng ta đã xây dựng được bảng đầu vào kích cho các FF và đã có được bảng đầu vào kích tổng hợp như sau:

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

Từ đó ta suy ra bảng hàm giá trị của các ngõ vào data theo các ngõ ra như sau :

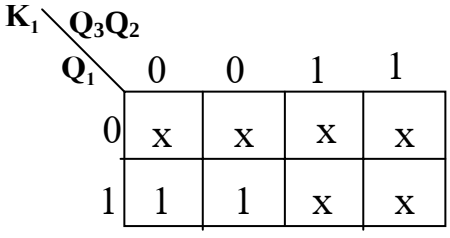
Xung vào	Trạng thái hiện tại			Trạng thái kế tiếp								
	Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
1	0	0	0	0	0	1	0	X	0	X	1	X
2	0	0	1	0	1	0	0	X	1	X	X	1
3	0	1	0	0	1	1	0	X	X	0	1	X
4	0	1	1	1	0	0	1	X	X	1	X	1
5	1	0	0	0	0	0	X	1	0	X	0	X

Lập bảng Karnaugh để tối thiểu hóa ta được:



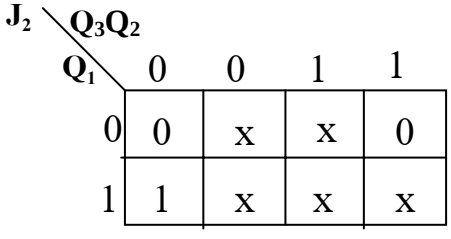
J_1 \ Q_3Q_2	00	01	11	10
Q_1 = 0	1	1	x	0
Q_1 = 1	x	x	x	x

$J_1 = Q_1$



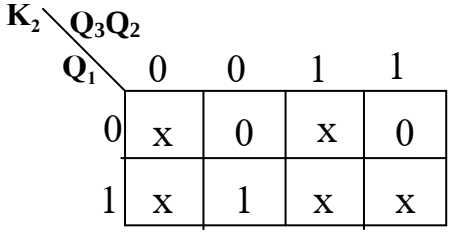
K_1 \ Q_3Q_2	00	01	11	10
Q_1 = 0	x	x	x	x
Q_1 = 1	1	1	x	x

$K_1 = 1 = Q_1$



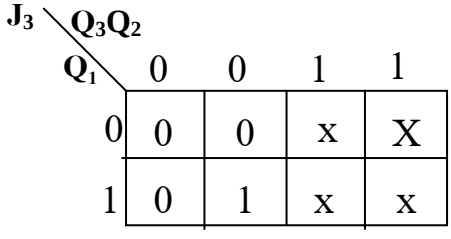
J_2 \ Q_3Q_2	00	01	11	10
Q_1 = 0	0	x	x	0
Q_1 = 1	1	x	x	x

$J_2 = Q_1$



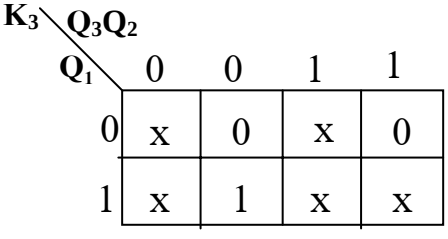
K_2 \ Q_3Q_2	00	01	11	10
Q_1 = 0	x	0	x	0
Q_1 = 1	x	1	x	x

$K_2 = Q_1$



J_3 \ Q_3Q_2	00	01	11	10
Q_1 = 0	0	0	x	X
Q_1 = 1	0	1	x	x

$J_2 = Q_1Q_2$

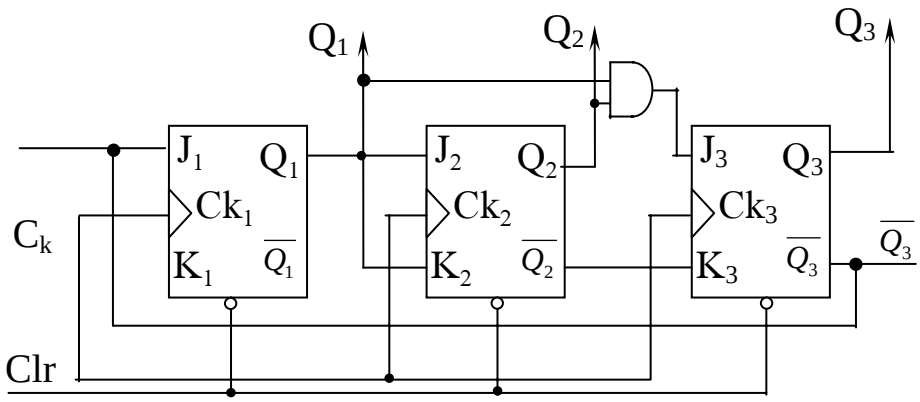


K_3 \ Q_3Q_2	00	01	11	10
Q_1 = 0	x	0	x	0
Q_1 = 1	x	1	x	x

$K_3 = 1 = Q_3$

Lưu ý: Khi thiết kế tính toán ta dùng các phương pháp tối thiểu để đưa về phương trình logic tối giản. Nhưng trong thực tế thì đôi lúc không phải như vậy. Ví dụ: $K_3 = 1$, $K_3 = Q_3$ hay $K_3 = \overline{Q_2}$ đều đúng, nhưng khi lắp ráp thực tế ta chọn $K_3 = \overline{Q_2}$ để tránh dây nối dài gây nhiễu cho mạch.

Sơ đồ logic: Hình 5.10



Hình 5.10. Sơ đồ mạch đếm 5, đếm lên, đếm song song

Giải thích hoạt động :

- Ban đầu dùng mạch RC xóa về 0 $\Rightarrow Q_1 = Q_2 = Q_3 = 0$.
  $J_1 = K_1 = 1$; $J_2 = K_2 = Q_2 = 0$; $J_3 = 0$, $K_3 = 1$.
- Khi Ck_1 ↑: Các trạng thái ngõ ra đều thay đổi theo trạng thái ngõ vào DATA trước đó.
  $J_1 = K_1 = 1 \quad \Rightarrow Q_1 = \overline{Q_1^0} = 1$.
  $J_2 = K_2 = 1 \quad \Rightarrow Q_2 = Q_2^0 = 0$.
  $J_3 = 0, K_3 = 1 \Rightarrow Q_3 = 1$ bất chấp trạng thái trước đó.
  (Hoặc $J_3 = 0, K_3 = 0 \Rightarrow Q_3 = Q_3^0 = 0$) $\Rightarrow Q_3Q_2Q_1 = 001$.
  Lúc đó: $J_1 = K_1 = \overline{Q_3} = 1$; $J_2 = K_2 = Q_1 = 1$; $J_3 = Q_2.Q_1 = 0$, $K_3 = 1$.
  (Hoặc $K_3 = Q_3 = 0$).
- Khi Ck_2 ↑ :
  $J_1 = K_1 = 1 \quad \Rightarrow Q_1 = \overline{Q_1^1} = 0$.
  $J_2 = K_2 = 1 \quad \Rightarrow Q_2 = \overline{Q_2^1} = 1$.
  $J_3 = 0, K_3 = 1 \Rightarrow Q_3 = 0$.
  (Hoặc $J_3 = 0, K_3 = 0 \Rightarrow Q_3 = Q_3^1 = 0$) $\Rightarrow Q_3\,Q_2\,Q_1 = 010$.
  Lúc đó: $J_1 = K_1 = \overline{Q_3} = 1$; $J_2 = K_2 = Q_1 = 0$; $J_3 = 0$, $K_3 = 1$.
  (Hoặc $K_3 = \overline{Q_2} = 0$).
- Khi Ck_3 ↑:
  $J_1 = K_1 = 1 \quad \Rightarrow Q_1 = \overline{Q_1^2} = 1$.
  $J_2 = K_2 = 0 \quad \Rightarrow Q_2 = Q_2^0 = 1$.
  $J_3 = 0, K_3 = 1 \Rightarrow Q_3 = 0$ bất chấp trạng thái trước đó.

(Hoặc $J_3 = 0, K_3 = 0 \Rightarrow Q_3 = Q_3^2 = 0$) $\Rightarrow$ $Q_3\,Q_2\,Q_1 = 011$.

Lúc đó: $J_1 = K_1 = \overline{Q_3} = 1$; $J_2 = K_2 = Q_1 = 1$; $J_3 = Q_2.Q_1 = 1$, $K_3 = 0$.

(Hoặc $K_3 = 1$).

- Khi Ck_4 ↑:

$J_1 = K_1 = 1 \quad \Rightarrow Q_1 = \overline{Q_1^3} = 0$.

$J_2 = K_2 = 1 \quad \Rightarrow Q_2 = \overline{Q_2^3} = 0$.

$J_3 = 0, K_3 = 1 \Rightarrow Q_3 = 1$ bất chấp trạng thái trước đó.

(Hoặc $J_3 = 0, K_3 = 0 \Rightarrow Q_3 = Q_3^0 = 0$) $\Rightarrow$ $Q_3\,Q_2\,Q_1 = 100$.

Lúc đó: $J_1 = K_1 = \overline{Q_3} = 1$; $J_2 = K_2 = Q_1 = 0$; $J_3 = Q_2.Q_1 = 0$, $K_3 = 1$.

(Hoặc $K_3 = Q_3 = 0$).

- Khi Ck_5 ↑:

$J_1 = K_1 = 1 \quad \Rightarrow Q_1 = Q_1^4 = 0$.

$J_2 = K_2 = 1 \quad \Rightarrow Q_2 = Q_2^4 = 0$.

$J_3 = 0, K_3 = 1 \Rightarrow Q_3 = 0$ bất chấp trạng thái trước đó.

$\Rightarrow$ $Q_3\,Q_2\,Q_1 = 000$.

Lúc đó: $J_1 = K_1 = \overline{Q_3} = 1$; $J_2 = K_2 = Q_1 = 0$; $J_3 = Q_2.Q_1 = 0$, $K_3 = 1$.

Mạch trở về trạng thái ban đầu.

5.2.4. Đếm thuận nghịch

Để thiết kế mạch cho phép vừa đếm lên vừa đếm xuống, ta thực hiện như sau:

- **Cách 1:** Lập hàm $J_{lên}$, $J_{xuống}$, $K_{lên}$, $K_{xuống}$ (giả sử ta dùng JKFF).

Gọi X là tín hiệu điều khiển. Xét 2 trường hợp:

+ Nếu quy ước X = 0: đếm lên; X = 1: đếm xuống.

Lúc đó ta có phương trình logic:

$J = \overline{X}.\,J_{lên} + X.\,J_{xuống}$

$K = \overline{X}.\,K_{lên} + X.\,K_{xuống}$

+ Nếu quy ước X = 1: đếm lên; X = 0: đếm xuống.

Lúc đó ta có phương trình logic:

$J = X.\,J_{lên} + \overline{X}.\,J_{xuống}$

$K = X.\,K_{lên} + \overline{X}.K_{xuống}$

- **Cách 2:** Lập bảng trạng thái.

Xung vào	X	Trạng thái h.tại	Trạng thái kế	J_3 K_3	J_2 K_2	J_1 K_1
1						
2						

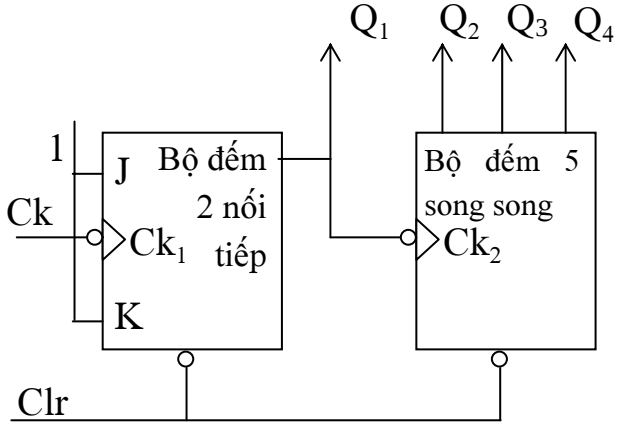
Sau đó thực hiện các bước giống như bộ đếm đồng bộ.

5.2.5. Đếm hỗn hợp

Bộ đếm hỗn hợp là bộ đếm mà trong đó bao gồm cả đếm nối tiếp và đếm song song. Đây là bộ đếm chế tạo khá nhiều trong thực tế và khả năng ứng dụng của bộ đếm hỗn hợp khá lớn so với bộ đếm song song.

Ví dụ: Bộ đếm 7490 bên trong bao gồm 2 bộ đếm đó là bộ đếm 2 nối tiếp và bộ đếm 5 song song. Hai bộ đếm này tách rời nhau. Do đó, tùy thuộc vào việc ghép hai bộ đếm này lại với nhau mà mạch có thể thực hiện được việc đếm thập phân hoặc chia tần số.

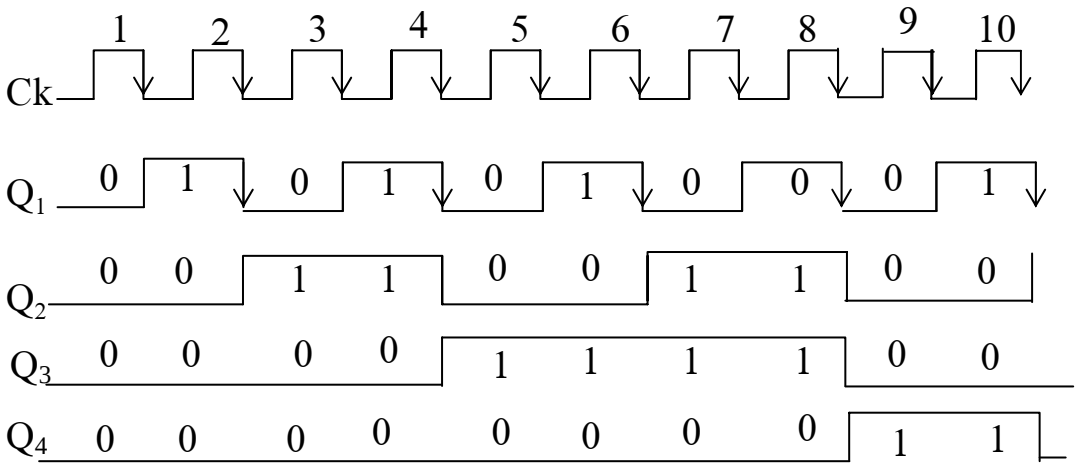
Trường hợp 1: 2 nối tiếp, 5 song song (hình 5.11).



Hình 5.11. Bộ đếm 2 nối tiếp ghép với bộ đếm 5 song song

Q_1 của bộ đếm 2 giữ vai trò xung Ck cho bộ đếm 5 song song.

Giản đồ thời gian của 2 nối tiếp 5 song song (hình 5.12) :



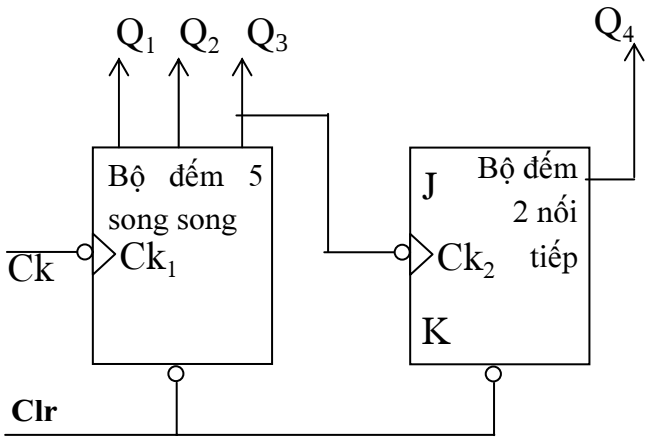
Hình 5.12. Giản đồ thời gian 2 nối tiếp ghép với 5 song song

Nhận xét: Cách ghép này dùng để đếm thập phân, nhưng không dùng để chia tần số.

Bảng trạng thái mô tả hoạt động của mạch:

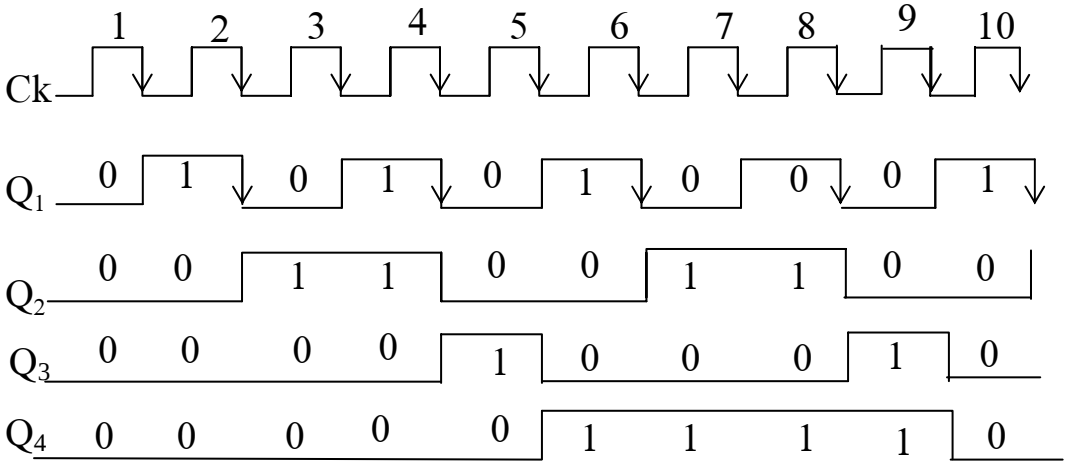
Xung vào	Trạng thái hiện tại				Trạng thái kế tiếp			
Ck	Q_4	Q_3	Q_2	Q_1	Q_4	Q_3	Q_2	Q_1
1	0	0	0	0	0	0	0	1
2	0	0	0	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	0	1	1	0	1	0	0
5	0	1	0	0	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	1	0	0	1	1	1
8	0	1	1	1	1	0	0	0
9	1	0	0	0	1	0	0	1
10	1	0	0	1	0	0	0	0

Trường hợp 2: 5 song song, 2 nối tiếp.



Hình 5.13. Bộ đếm 5 song song ghép với 2 nối tiếp

Q_3 của bộ đếm 5 song song giữ vai trò xung Ck cho bộ đếm 2.
Giản đồ thời gian của 5 song song nối tiếp 2.



Hình 5.14. Giản đồ thời gian 5 song song ghép 2 nối tiếp

Nhận xét: Cách ghép này không được dùng để đếm thập phân, nhưng lại thích hợp cho việc chia tần số.

Bảng trạng thái mô tả hoạt động của mạch :

Xung vào	Trạng thái hiện tại				Trạng thái kế tiếp			
Ck	Q_4	Q_3	Q_2	Q_1	Q_4	Q_3	Q_2	Q_1
1	0	0	0	0	0	0	0	1
2	0	0	0	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	0	1	1	0	1	0	0
5	0	1	0	0	0	1	0	1
6	1	0	0	0	1	0	0	1
7	1	0	0	1	1	0	1	0
8	1	0	1	0	1	0	1	1
9	1	0	1	1	1	1	0	0
10	1	1	0	1	0	0	0	0

5.3. THANH GHI DỊCH CHUYỂN VÀ BỘ NHỚ

5.3.1. Khái niệm

Thanh ghi dịch và bộ nhớ đều được ứng dụng trong lưu trữ dữ liệu, trong đó thanh ghi do khả năng lưu trữ cảu nó có hạng nên chỉ được sử dụng như bộ nhớ tạm thời (lưu kết quả các phép tính). Còn bộ nhớ có khả năng lưu trữ các bit gữ liệu khá lớn. Về mặc cấu tạo bênh trong nó được xây dựng trên cơ sở các thanh ghi (Nhiều thanh ghi hợp thành bộ nhớ)

5.3.2. Thanh ghi dịch chuyển

5.3.2.1. Khái niệm

Thanh ghi được xây dựng trên cơ sở các DFF (hoặc các FF khác thực hiện chức năng của DFF) và trong đó mỗi DFF sẽ lưu trữ 1 bit dữ liệu.

Để tạo thanh ghi nhiều bit, người ta ghép nhiều DFF lại với nhau theo qui luật như sau:

- Ngõ ra của DFF đứng trước được nối với ngõ vào DATA của DFF sau **($D_{i+1} = Q_i$) $\Rightarrow$ thanh ghi có khả năng dịch phải.**

- Hoặc ngõ ra của DFF đứng sau được nối với ngõ vào DATA của DFF đứng trước **($D_i = Q_{i+1}$) ⇒ thanh ghi có khả năng dịch trái.**

5.3.2.2. Phân loại

Phân loại theo số bit dữ liệu lưu trữ: 4 bit, 5 bit, 8 bit, 16 bit, 32 bit. Đối với thanh ghi lớn 8 bit, người ta không dùng họ TTL mà dùng họ CMOS.

Phân loại theo hướng dịch chuyển dữ liệu trong thanh ghi:

- Thanh ghi dịch trái.
- Thanh ghi dịch phải.
- Thanh ghi vừa dời phải vừa dời trái.

Phân loại theo ngõ vào dữ liệu:

- Ngõ vào dữ liệu nối tiếp.
- Ngõ vào dữ liệu song song: Song song không đồng bộ, song song đồng bộ.

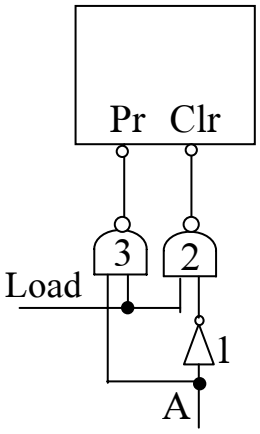
Phân loại theo ngõ ra:

- Ngõ ra nối tiếp.
- Ngõ ra song song.
- Ngõ ra vừa nối tiếp vừa song song.

5.3.2.3. Nhập dữ liệu vào FF

Nhập dữ liệu vào FF bằng chân Preset (Pr): (xem hình 5.15)

- Khi Load = 0 : Cổng NAND 3 và 2 khóa → ngõ vào Pr = Clr = 1 → FF tự do ⇒ dữ liệu A không nhập vào được FF.
- Khi Load = 1 : Cổng NAND 2 và 3 mở. Lúc đó ta có: Pr = $\overline{A}$, Clr = A.
  Nếu A = 0 → Pr = 1, Clr = 0 ⇒ Q = A = 0.
  Nếu A = 1 → Pr = 0, Clr = 1 ⇒ Q = A = 1.
  Vậy Q = A ⇒ dữ liệu A được nhập vào FF.



Hình 5.15

Tuy nhiên, cách này phải dùng nhiều cổng logic không kinh tế và phải dùng chân Clr là chân xóa nên phải thiết kế đồng bộ.

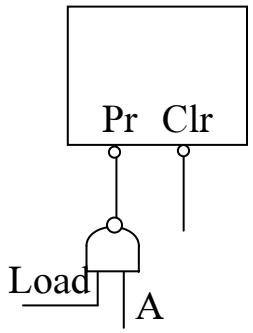
Để khắc phục những nhược điểm đó ta dùng mạch như trên hình 5.16 :

- Chân Clr để trống tương đương với mức logic 1.
- Khi Load = 0 : cổng NAND khóa → Pr = Clr =1 → FF tự do. Dữ liệu không được nhập vào FF.
- Khi Load = 1 : cổng NAND mở → Pr = $\overline{A}$.

  Giả sử ban đầu : Q = 0.

  Nếu A = 0 → Pr = 1, Clr = 1 ⇒ Q = Q^0 = 0.

  Nếu A = 1 → Pr = 0, Clr = 1 ⇒ Q = 1.

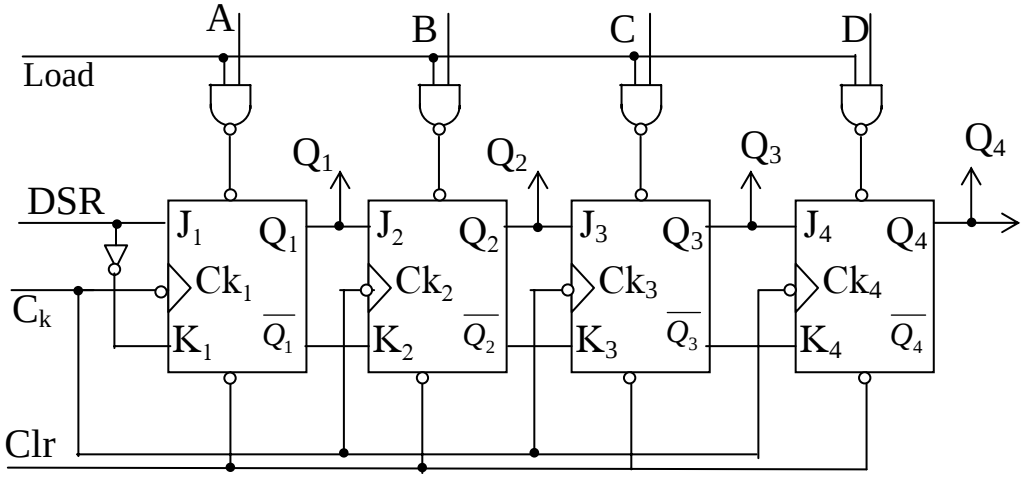


Hình 5.16

Vậy Q = A ⇒ Dữ liệu A được nhập vào FF.

Chú ý: Phương pháp này đòi hỏi trước khi nhập phải xóa FF về 0.

Ví dụ: Xét một thanh 4 bit có khả năng dời phải (hình 5.17).



Hình 5.17. Thanh ghi dịch phải

Trong đó:

- DSR (Data Shift Right): Ngõ vào Data nối tiếp (ngõ vào dịch phải).
- Q_1, Q_2,Q_3, Q_4 : các ngõ ra song song.

Để giải thích hoạt động của mạch, ta dựa vào bảng trạng thái của DFF.

Giả sử ban đầu : Ngõ vào nhập Load = 1 → A, B, C, D được nhập vào thanh ghi dịch → Q_1 = A, Q_2 = B, Q_3 = C, Q_4 = D.

Hoạt động dịch phải của thanh ghi:

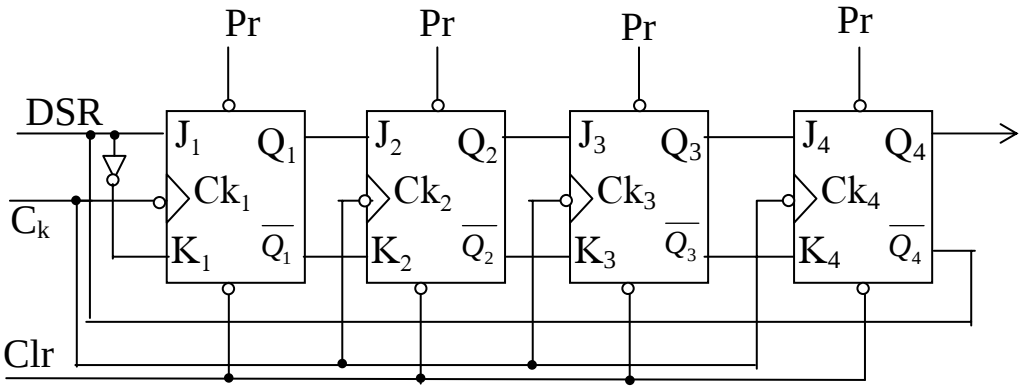
- Xét FF_1: $D = DSR_1$, $Q_1 = A$.
  Nếu $DSR_1 = 0 \rightarrow Q = 0$; nếu $DSR_1 = 1 \rightarrow Q = 1$.
  Kết luận: Sau một xung Ck tác động sườn xuống thì $Q_1 = DSR_1$.
- Lúc đó FF_2, FF_3,FF_4 : $Q_2 = A$, $Q_3 = B$, $Q_4 = C$.

Tức là sau khi Ck tác động sườn xuống thì 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 thế bằng các dữ liệu từ ngõ vào DATA nối tiếp DSR_1, DSR_2, DSR_3, DSR_4.

Ta có bảng trạng thái hoạt động của mạch:

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

Trường hợp ngõ ra $\overline{Q}$ bằng ngõ vào dữ liệu nối tiếp DSR (hình 5.18).



Hình 5.18.

Ta có bảng trạng thái hoạt động của mạch hình 5.18:

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

Đây là mạch được ứng dụng nhiều trong thực tế.

5.3.3. Bộ nhớ

5.3.3.1. Các khái niệm

- **Tế bào nhớ (Memory cell)**

  Đó là thiết bị hay mạch điện tử dùng để lưu trữ 1 bit.

  Ví dụ: FF để lưu trữ 1 bit, tụ điện khi nạp điện thì lưu trữ 1 bit, hoặc một điểm trên băng từ.

- **Từ nhớ (Memory word)**

  Là nhóm các bit ở trong một bộ nhớ.

  Ví dụ: Một thanh ghi gồm 8 DFF có thể lưu trữ từ nhớ là 8 bit.

  Trong thực tế, kích thước của từ nhớ có thể thay đổi trong các loại máy tính từ $4 \rightarrow 64$ bit.

- **Byte:**

  Một nhóm từ nhớ 8 bit.

- **Dung lượng bộ nhớ**

  Chỉ khả năng lưu trữ của bộ nhớ.

  Ví dụ: $1K = 2^{10}$; $2K = 2^{11}$; $4K = 2^{12}$; $1M = 2^{20}$.

- **Địa chỉ**

  Dùng để xác định các vùng của các từ trong bộ nhớ.

Xét bộ nhớ gồm 16 ngăn nhớ tương đương 16 từ, ta cần dùng 4 đường địa chỉ ($2^4 = 16 \rightarrow$ có 4 đường địa chỉ). Như vậy có mối quan hệ giữa địa chỉ và dung lượng bộ nhớ.

Ví dụ : Để quản lý được bộ nhớ có dung lượng là 8 Kbytes thì cần 13 đường địa chỉ.

- **Hoạt động đọc (READ)**

Đọc là xuất dữ liệu từ bộ nhớ ra ngoài.

Để đọc nội dung một ô nhớ cần thực hiện:

+ Đưa địa chỉ tương ứng vào các đường địa chỉ A.
+ Khi tín hiệu điều khiển đọc tác động thì lúc bấy giờ dữ liệu chứa trong các ngăn nhớ tương ứng với vùng địa chỉ xác định ở trên sẽ được xuất ra ngoài.

- **Hoạt động viết (WRITE)**

Viết là ghi dữ liệu từ bên ngoài vào bên trong bộ nhớ.

Muốn viết phải thực hiện:

+ Đặt các địa chỉ tương ứng lên các đường địa chỉ.
+ Đặt dữ liệu cần viết vào bộ nhớ lên các đường dữ liệu.
+ Tích cực tín hiệu điều khiển ghi.

Khi ghi dữ liệu từ bên ngoài vào bên trong bộ nhớ thì dữ liệu cũ sẽ mất đi và được thay thế bằng dữ liệu mới.

- **Bộ nhớ không bay hơi**

Chỉ loại bộ nhớ mà dữ liệu không mất đi khi mất nguồn điện.

- **Bộ nhớ bay hơi**

Chỉ loại bộ nhớ lưu trữ dữ liệu khi còn nguồn điện và khi mất nguồn điện thì dữ liệu sẽ bị mất.

- **RAM** (**R**andom **A**ccess **M**emory)

Bộ nhớ truy xuất ngẫu nhiên, đọc viết tùy ý, còn được gọi là RWM (Read/Write Memory). Đây là loại bộ nhớ cho phép đọc dữ liệu chứa bên trong ra ngoài và cho phép nhập dữ liệu từ bên ngoài vào trong.

- **ROM** (**R**ead **O**nly **M**emory)

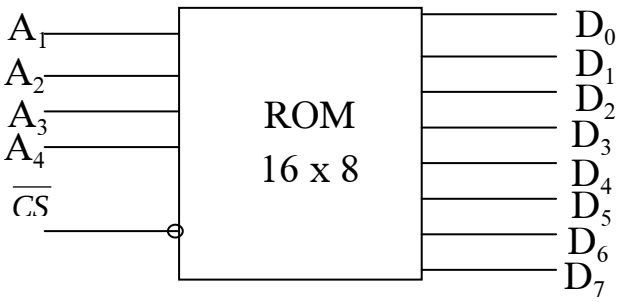
Bộ nhớ chỉ đọc. Chỉ cho phép đọc dữ liệu trong ROM ra ngoài mà không cho phép dữ liệu ghi dữ liệu từ bên ngoài vào trong bộ nhớ.

- **SM (Static Memory)**
  Bộ nhớ tĩnh là loại bộ nhớ lưu trữ dữ liệu cho đến khi mất điện áp cung cấp mà không cần làm tươi dữ liệu bên trong. Ví dụ: SRAM.
- **DM (Dynamic Memory)**
  Bộ nhớ động là loại bộ nhớ có thể mất dữ liệu khi điện áp cung cấp chưa bị mất, vì vậy cần có cơ chế làm tươi dữ liệu. Ưu điểm của loại bộ nhớ này là tốc độ truy xuất nhanh, giá thành hạ. Ví dụ: DRAM.
- **Bộ nhớ tuần tự**
  Ví dụ: Đĩa mềm, đĩa cứng, băng từ.

5.3.3.2.ROM (Read Only Memory)

- **MROM** (Mask ROM): Được lập trình bởi nhà sản xuất.
  Ưu và nhược điểm: Chỉ có tính kinh tế khi sản xuất hàng loạt nhưng lại không phục hồi được khi chương trình bị sai hỏng.
- **PROM** (Programmable ROM): Đây là loại ROM cho phép lập trình bởi nhà sản xuất. Nhược điểm: Nếu hỏng không phục hồi được.
- **EPROM** (Erasable PROM): Đó là loại PROM có thể xóa và lập trình lại. Có hai loại EPROM: EPROM được xóa bằng tia cực tím (Ultralviolet EPROM) và EPROM xóa bằng xung điện (Electrical EPROM). Tuổi thọ của EPROM phụ thuộc vào thời gian xóa.

Ứng dụng của ROM: Chứa chương trình điều khiển vào ra của máy tính, PC, µP, µC, ROM BIOS (ROM Basic Input/Output System). Dùng để chứa ký tự: ROM ký tự. Dùng để chứa các biến đổi hàm.

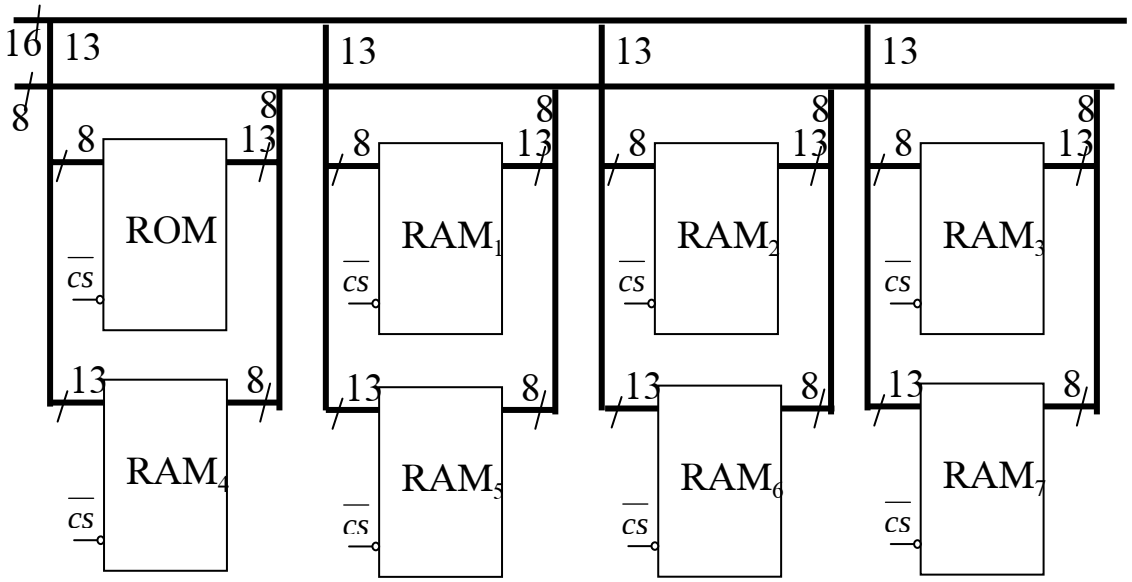


Hình 5.19. Sơ đồ khối của ROM 16x8 = 128 bit

5.3.3.3.RAM (Random Access Memory)

DRAM: Làm việc theo hai pha. Một pha chọn địa chỉ hàng, một pha chọn địa chỉ cột. Do đó, số chân địa chỉ thực hiện trên IC nhỏ hơn một nữa so với RAM hoặc ROM.

5.3.3.4.Tổ chức bộ nhớ



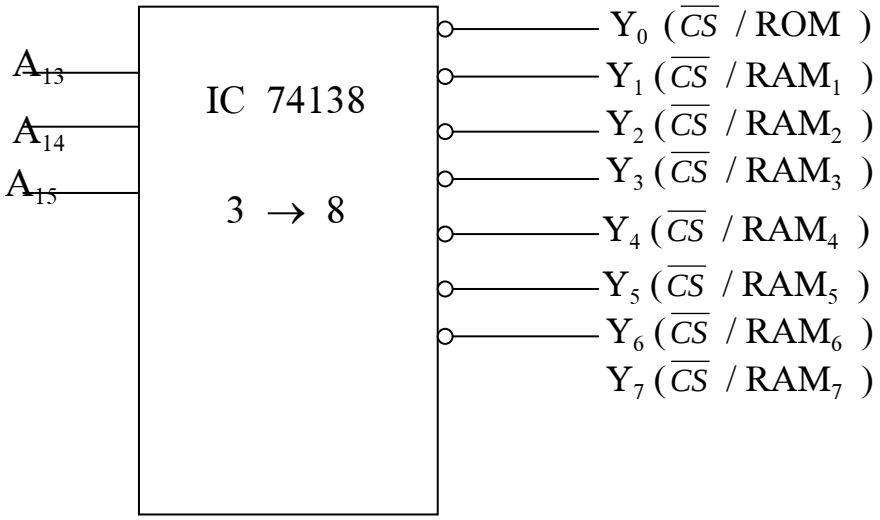
Hình 5.20. Tổ chức bộ nhớ

Giả sử CPU hay μP có 16 đường địa chỉ và 8 đường dữ liệu. Nếu dùng để quản lý bộ nhớ thì quản lý được dung lượng bộ nhớ tối đa là 64 Kbytes.

Giả sử 64 Kbytes phân thành các loại sau: 1 ROM 8K, và 7 RAM 8K.

Để chọn lần lượt từng bộ nhớ để xuất dữ liệu và vì còn thừa 3 dường địa chỉ là A_{13}, A_{14}, A_{15} nên ta dùng mạch giải mã từ $3 \rightarrow 8$.

Trên hình 5.21 là sơ đồ mạch giải mã địa chỉ dùng IC 74138.



Hình 5.21. Mạch giải mã địa chỉ

Bản đồ bộ nhớ của hệ thống:

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Địa chỉ Hex				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 H	ROM
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	F	F	F H	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0 H	RAM1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	F	F	F H	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0 H	RAM2
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5	F	F	F H	
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0 H	RAM3
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7	F	F	F H	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0 H	RAM4
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	9	F	F	F H	
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0 H	RAM5
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B	F	F	F H	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C	0	0	0 H	RAM6
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	D	F	F	F H	
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	E	0	0	0 H	RAM7
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	F	F	F	F H	