



NGUYỄN THÚY VÂN

KỸ THUẬT SỐ



* GT 172053 *

NHÀ XUẤT BẢN
KHOA HỌC VÀ KỸ THUẬT



NGUYỄN THÚY VÂN

KỸ THUẬT SỐ

*Sách được dùng làm giáo trình
cho các trường đại học kỹ thuật*

Tái bản có sửa chữa



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI - 2004

Chịu trách nhiệm xuất bản : PGs. Ts. TÔ ĐĂNG HẢI
Biên tập : NGUYỄN NGỌC, NGUYỄN ĐĂNG
Vẽ bìa : HƯƠNG LAN

In 1.000 cuốn khổ 19x 27 cm tại Xưởng in NXB Văn hoá dân tộc.

Giấy phép xuất bản số 1189 - 35 - 10/9/ 2003.

In xong và nộp lưu chiểu tháng 1/ 2004.

MỤC LỤC

Trang

Phần 1. ĐẠI SỐ BOOLE VÀ VI MẠCH SỐ

<i>Chương 1. Đại số Boole</i>	6
1- 1. Mở đầu	6
1- 2. Biến logic và hàm logic	6
1- 3. Các hàm logic cơ bản	8
1- 4. Các phương pháp biểu diễn hàm logic	10
1- 5. Các hệ thức cơ bản và hệ quả trong đại số logic	15
1- 6. Hệ hàm dù	16
1- 7. Các hàm NAND, NOR	16
1- 8. Hàm XOR	20
1- 9. Hàm tương đương	21
1- 10. Một số nhận xét tổng quát	21
<i>Bài tập</i>	23
<i>Chương 2. Tối thiểu hóa hàm Boole</i>	25
2- 1. Khái niệm về tối thiểu hóa hàm Boole	25
2- 2. Các phương pháp tối thiểu hóa hàm Boole	27
2- 3. Tối thiểu hóa bằng phương pháp Quine – Mc. Cluskey	28
2- 4. Phương pháp tối thiểu hóa dùng bảng Karnaugh cho hàm dạng ETT	32
2- 5. Tối thiểu hóa hàm ở dạng chuẩn tắc hội	34
<i>Bài tập</i>	35
<i>Chương 3. Vi mạch số</i>	36
3- 1. Định nghĩa và phân loại	36
3- 2. Phân loại theo bản chất của tín hiệu điện vào ra của vi mạch	36
3- 3. Phân loại theo mật độ tinh hợp	37
3- 4. Phân loại theo công nghệ chế tạo	38
3- 5. Những thông số kỹ thuật của vi mạch số	40
3- 6. Đóng vỏ IC	42
3- 7. Giới hạn nhiệt độ	43
3- 8. Công nghệ đơn cực (MOS - METAL OXID SEMI- CONDUCTER)	44
3- 9. Công nghệ lưỡng cực	45
3- 10. Các mạch ra TTL	46
<i>Tổng quan về mạch số</i>	53

Phần 2. MẠCH TỔ HỢP

Chương 4. Thiết kế và phân tích mạch tổ hợp	58
4- 1. Mô hình toán học	58
4- 2. Phân tích mạch tổ hợp	59
4- 3. Thiết kế mạch tổ hợp	60
Bài tập	67
Chương 5. Các mạch tổ hợp thường gặp	68
5- 1. Bộ cộng nhị phân một cột số	68
5- 2. Bộ trừ nhị phân một cột số	71
5- 3. Bộ so sánh (Com parator)	72
5- 4. Mạch tạo và kiểm tra chẵn lẻ	75
5- 5. Mạch phân loại ngắn	77
5- 6. Bộ dồn kênh (MUX) hay bộ chọn dữ liệu (Data Selector)	78
5- 7. Bộ phân kênh (DEMUX)	79
5- 8. Mạch chuyển mã	80
5- 9. Một số ví mạch tổ hợp	91
Bài tập	94

Phần 3. MẠCH DÂY

Chương 6. Các phần tử nhỏ cơ bản	98
6- 1. Giới thiệu	98
6- 2. Định nghĩa và phân loại	98
6- 3. Các loại Flip Flop và điều kiện đồng bộ	100
6- 4. RS - FF	102
6- 5. T - FF	104
6- 6. JK - FF	105
6- 7. D - FF	107
6- 8. FF làm việc như một mạch chốt	109
6- 9. Xác định đầu vào kích cho FF	111
6- 10. Chuyển đổi giữa các loại FF	113
6- 11. Một số ví mạch FF	116
Bài tập	116
Chương 7. Những khái niệm cơ bản về mạch dây	119
7- 1. Một số khái niệm cơ bản về mạch dây	119
7- 2. Các phương pháp mô tả mạch dây	122
7- 3. Chuyển đổi giữa hai mô hình Mealy và Moore	125
7- 4. Các bước thiết kế mạch dây	128
7- 5. Thiết kế mạch dây từ đồ hình trạng thái	129
7- 6. Thiết kế mạch dây từ bảng của Otomat	136
7- 7. Thiết kế mạch dây từ lưu đồ thuật toán	137
Bài tập	143

Chương 8. Bộ đếm	145
8–1. Định nghĩa và phân loại bộ đếm	145
8–2. Mã của bộ đếm	148
8–3. Các bước thiết kế bộ đếm	150
8–4. Bộ đếm thuận, nhị phân, đồng bộ với hệ số đếm $K_d = 2^n$	151
8–5. Các bộ đếm nghịch, nhị phân, đồng bộ với hệ số đếm $K_d = 2^n$	155
8–6. Các bộ đếm Gray thuận, đồng bộ với hệ số đếm $K_d = 2^n$	157
8–7. Các bộ đếm đồng bộ thuận nghịch	161
8–8. Các bộ đếm thập phân, đồng bộ	161
8–9. Bộ đếm nhị phân, không đồng bộ với $K_d = 2^n$	167
8–10. Bộ đếm thập phân, không đồng bộ	169
8–11. Bộ đếm đặt lại trạng thái	170
8–12. Bộ đếm Johnson	173
8–13. Bộ đếm vòng	174
8–14. Một số vấn đề cần chú ý khi thiết kế bộ đếm	174
8–15. Một số ví mạch đếm thường gặp	179
<i>Bài tập</i>	183
Chương 9. Bộ ghi dịch	186
9–1. Giới thiệu	186
9–2. Thanh ghi dịch 4 bit nạp vào nối tiếp hoặc song song, ra nối tiếp, dịch phải	187
9–3. Bộ ghi dịch 4 bit, dịch trái, dịch phải	188
9–4. Đồ hình tổng quát của bộ ghi dịch	190
9–5. Thiết kế bộ đếm dùng bộ ghi dịch	192
9–6. Tạo dãy tín hiệu tuần hoàn dùng thanh ghi dịch	194
9–7. Bộ đếm vòng	198
9–8. Bộ đếm vòng xoắn (Bộ đếm Johnson)	201
9–9. Bộ ghi dịch với hàm hồi tiếp cộng modul	205
9–10. Một số ví mạch ghi dịch	210
<i>Bài tập</i>	212
Chương 10. Mạch dãy đồng bộ	213
10–1. Phân tích mạch dãy đồng bộ	213
10–2. Các bước thiết kế mạch dãy đồng bộ	218
10–3. Mạch dãy đồng bộ dùng mô hình Moore và mô hình Mealy	226
10–4. Tối thiểu hoá trạng thái	235
10–5. Mã hóa trạng thái	239
10–6. Mạch kiểm tra dãy tín hiệu vào	244
<i>Bài tập</i>	246
Chương 11. Mạch dãy không đồng bộ	250
11–1. Các bước thiết kế một mạch dãy không đồng bộ	250
11–2. Một ví dụ về thiết kế mạch dãy không đồng bộ	253
11–3. Hiện tượng chu kỳ và chạy đua trong mạch dãy không đồng bộ	258

Trang

11– 5. Tối thiểu hóa và mã hóa trạng thái trong mạch dãy không đồng bộ	261
11– 6. Một số thí dụ thiết kế mạch dãy không đồng bộ	264
<i>Bài tập</i>	172
Chương 12. Thiết kế mạch số dùng MSI, LSI	275
12– 1. Mở đầu	275
12– 2. Các bộ chọn dữ liệu hay dòn kênh	276
12– 3. Các ứng dụng của MUX	277
12– 4. Bộ phân kênh (DEMUX) và bộ giải mã (DECODER)	284
12– 5. Ứng dụng của bộ giải mã	284
12– 6. ROM	287
12– 7. Các ứng dụng của ROM	288
12– 8. Các mảng logic lập trình PLA	294
12– 9. Các ứng dụng của PLA	297
<i>Bài tập</i>	300
Chương 13. Mạch số học	301
13– 1. Bộ song song n bit	301
13– 2. Bộ cộng nhớ nhanh	303
13– 3. Mã bù	305
13– 4. Biểu diễn số nhị phân nguyên trong máy tính	307
13– 5. Cộng trừ dùng mã bù 2	308
13– 6. Cộng trừ dùng mã bù 1	310
13– 7. Hiện tượng tràn	311
13– 8. Bộ cộng trừ liên tiếp	312
13– 9. Cộng số thập phân dùng bộ cộng MSI số nhị phân	314
13– 10. Dùng số bù cho phép tính thập phân	315
13– 11. Cộng trừ số thập phân	317
13– 12. Đơn vị số học và logic (ALU)	319
13– 13. Thiết kế một đơn vị số học logic 4 bit	319
13– 14. Bộ nhân nhị phân dùng mạch tổ hợp	323
13– 15. Bộ nhân nhị phân dùng ROM	325
13– 16. Bộ nhân nhị phân dùng phương pháp dịch và cộng	326
13– 17. Một số vi mạch	330
<i>Bài tập</i>	332

Đáp số của bài tập

Tài liệu tham khảo

LỜI NÓI ĐẦU

Trong những năm gần đây công nghệ vi điện tử phát triển rất mạnh mẽ. Sự ra đời của các vi mạch cỡ lớn, cực lớn với giá thành giảm nhanh, khả năng lập trình ngày càng cao đã mang lại những thay đổi sâu sắc trong ngành kỹ thuật điện tử. Mạch số, ở những mức độ khác nhau đã và đang thâm nhập vào tất cả các thiết bị điện tử thông dụng và chuyên dụng. Tình hình đó đòi hỏi Kỹ thuật số – một giáo trình cơ sở cho các ngành Kỹ thuật điện tử, Kỹ thuật máy tính, Tin học, Điều khiển tự động, Thông tin, Đo lường điện tử ... phải có những cải tiến phù hợp.

Cuốn Kỹ thuật số này nhằm đáp ứng nhu cầu tiếp cận kỹ thuật hiện đại và chương trình đào tạo của Bộ Giáo dục và Đào tạo, trong khuôn khổ đề tài khoa học cấp Nhà nước KC-01 : "Đổi mới đào tạo ngành Điện tử – Tin học – Viễn thông".

Cuốn sách gồm 3 phần

- **Phần 1 :** *Dại số Boole và vi mạch số, trình bày cơ sở toán và kỹ thuật của mạch số*
- **Phần 2 :** *Mạch tổ hợp, giới thiệu những vấn đề lý thuyết cơ bản của mạch tổ hợp, những mạch tổ hợp thường gặp và ứng dụng*
- **Phần 3 :** *Mạch dãy, trình bày các phương pháp phân tích, thiết kế mạch dãy, các mạch dãy thường gặp và ứng dụng.*

Ngoài những kiến thức cơ bản nhất về mạch số, cách tra cứu và sử dụng các vi mạch số để có thể giải quyết được các bài toán phân tích và thiết kế mạch số với các loại vi mạch cỡ khác nhau, chúng tôi đưa ra nhiều ví dụ cụ thể và bài tập ở mỗi chương. Phần giải đáp các bài tập này được đặt ở cuối sách.

Các chương liên quan đến vi mạch đều có giới thiệu một số vi mạch (thông dụng nhất là họ TTL)

Cuốn sách là giáo trình đồng thời còn có thể dùng làm tài liệu tham khảo cho sinh viên năm cuối và nghiên cứu sinh của các ngành Kỹ thuật điện tử, Máy tính, Tin học ... và các ngành liên quan.

Để bổ sung và hoàn chỉnh những kiến thức đã trình bày ở đây bạn đọc có thể tham khảo cuốn "Thiết kế lôgic mạch số" Nhà xuất bản KH và KT – 1996 của cùng tác giả.

Tập 2 tiếp theo sẽ đề cập đến các vấn đề lý thuyết và kỹ thuật cụ thể như : Hazards, chẩn đoán sai lầm, mô phỏng logic, tự động phân tích và thiết kế mạch số, thiết kế dùng các modul có sẵn, thiết kế bộ logic và số học (ALU), bộ điều khiển, micro – processor, phối ghép các mạch logic công nghệ khác nhau, phối ghép mạch logic với mạch công suất.

Trong lần in lại này, tác giả đã cố sửa chữa, bổ sung song chắc
ăn không thể tránh khỏi còn sai sót. Tác giả mong nhận được ý kiến
đóng góp của bạn đọc để cuốn sách được hoàn chỉnh hơn trong lần
xuất bản sau.

Thư góp ý xin gửi về Khoa Công nghệ thông tin Trường Đại học
Bách khoa thành phố Hồ Chí Minh hoặc Nhà xuất bản Khoa học và Kỹ
thuật 70 Trần Hưng Đạo - Hà Nội.

TÁC GIẢ

Phần 1

ĐẠI SỐ BOOLE VÀ VI MẠCH SỐ

CHƯƠNG 1

ĐẠI SỐ BOOLE

1_1. MỞ ĐẦU

Trong mạch số, các tín hiệu thường cho ở hai mức điện áp, ví dụ : 0 V và 5 V. Những linh kiện điện tử dùng trong mạch số làm việc ở một trong hai trạng thái, ví dụ : transito lưỡng cực làm việc ở chế độ khoá hoặc là tắt hoặc là thông, trạng thái từ dư của một vật liệu từ có thể là $+B_r$ hoặc $-B_r$ (hình 1-1).

Do vậy, để mô tả các mạch số người ta dùng hệ nhị phân (binary), hai trạng thái của các linh kiện trong mạch được mã hóa tương ứng là 0 hoặc 1.

Một bộ môn đại số phát triển từ cuối thế kỷ 19 mang tên chính người sáng lập ra nó : đại số Boole và còn được gọi là đại số Logic, thích hợp cho việc mô tả mạch số.

Đại số Boole là công cụ toán học quan trọng để thiết kế và phân tích mạch số. Các kỹ sư, các nhà chuyên môn trong lĩnh vực điện tử, tin học, thông tin, điều khiển... đều cần phải nắm vững công cụ này dùng nó làm chìa khoá để đi sâu vào mọi lĩnh vực có liên quan đến kỹ thuật số.

1_2. BIẾN LOGIC VÀ HÀM LOGIC

1_2_1. BIẾN LOGIC

Xét một tập hợp B chỉ chứa hai phần tử 0 và 1, $B = \{ 0, 1 \}$. X_i được gọi là biến logic nếu như $X_i \in B$ (binary), tức là X_i chỉ có thể lấy hai giá trị là 1 hoặc 0.

Biến logic biểu thị hai trạng thái hay hai tính chất đối lập nhau như đúng và sai, sống và chết, dương và âm... Trong kỹ thuật, biến logic thường được mã hóa như sau :

Điện thế :

$$X_i = 0 \text{ tương ứng với } U = 0 \text{ V}$$

$$X_i = 1 \text{ tương ứng với } U = 5 \text{ V}.$$

Trong cách mã hóa này, mức logic "1" có điện thế cao hơn mức logic "0" nên được gọi là logic dương, nếu mã hóa ngược lại ta có logic âm, tức là :

$$X_i = 0 \text{ tương ứng với } U = 5 \text{ V}$$

$$X_i = 1 \text{ tương ứng với } U = 0 \text{ V}.$$

Trạng thái từ dư :

$$\begin{aligned} X_i = 0 &\text{ tương ứng với } +B_r \\ X_i = 1 &\text{ tương ứng với } -B_r \end{aligned}$$

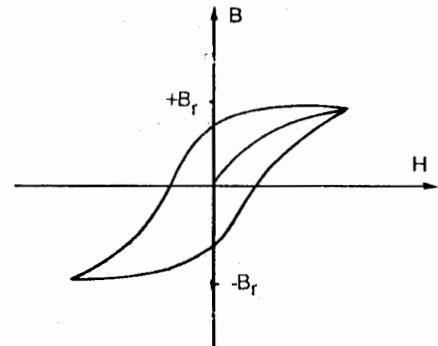
1-2.2. HÀM LOGIC

Hàm f được gọi là hàm logic nếu như f là hàm của một tập biến logic và bản thân f cũng chỉ lấy hai giá trị 0 hoặc 1 hay nói cách khác $f \in B$.

$$f = f(X_n, X_{n-1}, \dots, X_1) \in B$$

$$X_i \in B \text{ với } i = 1 \div n.$$

Nhận xét : Một tập hợp n biến logic có thể biểu diễn 2^n tổ hợp giá trị khác nhau (hình 1-2).



Hình 1-1. Đường cong từ trễ của vật liệu sắt từ.

Thứ tự	X_n	X_{n-1}	X_{n-2}	...	X_2	X_1
1	0	0	0	...	0	0
2	0	0	0	...	0	1
.				...		
2^n	1	1	1	...	1	1

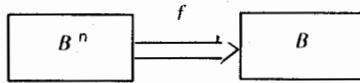
Hình 1-2. Các tổ hợp biến của n biến logic.

Ví dụ : Một tập 2 biến logic X_1 và X_2 sẽ có $2^2 = 4$ tổ hợp khác nhau như trong hình 1-3 :

Thứ tự	X_2	X_1
1	0	0
2	0	1
3	1	0
4	1	1

Hình 1-3. Các tổ hợp biến của X_1, X_2 .

Hàm logic f là một ánh xạ từ không gian B^n vào không gian B được biểu diễn như hình 1-4:



Hình 1-4. Mô tả hàm logic f .

$$f = f(X_n, X_{n-1}, \dots, X_1)$$

với :

$$f \in B = \{0, 1\}$$

$$X = X_n, X_{n-1}, \dots, X_1 \in B^n.$$

1_3. CÁC HÀM LOGIC CƠ BẢN

Như ta đã biết, với n biến logic sẽ có 2^n tổ hợp biến khác nhau, ứng với mỗi tổ hợp biến, hàm logic có thể lấy hai giá trị khác nhau. Như vậy, với n biến có thể có 2^{2^n} hàm khác nhau. Sau đây sẽ xét chi tiết các hàm logic cơ bản.

1_3.1. CÁC HÀM MỘT BIẾN

Giá trị của các hàm một biến $f = f(X_I)$ được cho trong hình 1-5.

X_1	f_o	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

Hình 1-5. Các hàm một biến.

Nhận xét :

$f_o \equiv 0$: hàm hằng 0 tức f_o luôn = 0.

$f_3 \equiv 1$: hàm hằng 1 tức f_3 luôn = 1.

$f_1 = X_I$: hàm lặp lại giá trị của X_I .

$f_2 = \bar{X}_I$: hàm đảo hoặc hàm phủ định của X_I (ký hiệu là NOT).

1_3.2. CÁC HÀM HAI BIẾN

Giá trị của 16 hàm hai biến khác nhau (f_o, f_1, \dots, f_{15}) được cho trong bảng trên hình 1-6 :

X_1	X_2	f_o	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	

X_1	X_2	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Hình 1-6. Các hàm hai biến.

Nhận xét : Các hàm đối xứng qua trục giữa f_7, f_8 là phủ định của nhau.

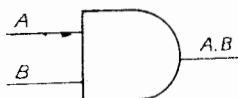
Ví dụ : $f_0 = \bar{f}_{15}$; $f_6 = \bar{f}_9$; ...

Một số hàm đặc biệt :

+ $f_0 \equiv 0$ với $\forall X$: được gọi là hàm hằng 0

+ $f_{15} \equiv 1$ với $\forall X$: được gọi là hàm hằng 1

+ $f_1 = X_1 \cdot X_2$ bằng 1 khi và chỉ khi $X_1 = X_2 = 1$. Hàm được gọi là hàm "Và" (AND). Mạch thực hiện hàm này có ký hiệu như hình 1-7.



Hình 1-7.

+ $f_7 = X_1 + X_2$ bằng 0 khi và chỉ khi $X_1 = X_2 = 0$.

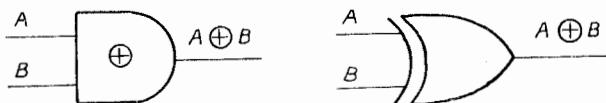
Hàm bằng 1 khi ít nhất 1 trong các biến của hàm bằng 1. Đây là hàm "Hoặc" (OR) và được ký hiệu trên hình 1-8.



Hình 1-8.

+ $f_6 = X_1 \oplus X_2$; $f_6 = 1$ khi và chỉ khi $X_1 \neq X_2$.

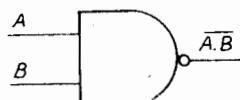
Đây là hàm không tương đương hay còn gọi là hàm cộng module 2 hay hàm cộng với sự loại trừ (XOR), hình 1-9.



Hình 1-9.

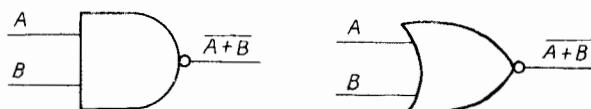
Tương ứng ta có các hàm đối xứng với các hàm trên :

+ $f_{14} = \bar{f}_1$: gọi là hàm "Và - phủ định" (NAND), hình 1-10.



Hình 1-10.

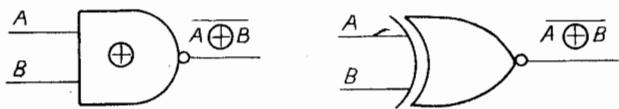
+ $f_8 = \bar{f}_7$: gọi là hàm "Hoặc - phủ định" (NOR), hình 1-11.



Hình 1-11.

+ $f_9 = \bar{f}_6$: gọi là hàm "tương đương".

Ký hiệu : $f_9 = X_1 \sim X_2$, hình 1-12.



Hình 1-12.

1

Mở rộng cho trường hợp n biến ta có :

Hàm AND : $f = X_n \cdot X_{n-1} \cdots \cdot X_1 = 1$ khi và chỉ khi $X_n = X_{n-1} = \dots = X_1 = 1$

Hàm OR : $f = X_n + X_{n-1} + \dots + X_1 = 0$ khi và chỉ khi $X_1 = X_2 = \dots = X_n = 0$

Các hàm logic nêu trên sẽ được xét kỹ trong các chương sau.

1_4. CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM LOGIC

Trước hết ta xét khái niệm hàm xác định đầy đủ và hàm không xác định đầy đủ.

Hàm xác định đầy đủ là hàm có trị số xác định với mọi tổ hợp biến. Hàm không thỏa mãn điều kiện trên là hàm không xác định đầy đủ. Tại những tổ hợp biến mà trị số của hàm không xác định (có thể là "0" hoặc "1") giá trị của hàm sẽ được ký hiệu bằng dấu "X".

1_4.1. BẢNG GIÁ TRỊ CỦA HÀM

Tương tự như trong đại số thông thường, một hàm logic có thể được biểu diễn bởi bảng giá trị của hàm đó. Bảng này sẽ có $n + 1$ cột (trong đó n cột là giá trị của biến và 1 cột là giá trị của hàm), 2^n hàng tương ứng với 2^n tổ hợp giá trị khác nhau của n biến vào. Ứng với mỗi tổ hợp giá trị biến ghi giá trị của hàm tương ứng. Bảng này còn được gọi là bảng chân lý hay bảng chức năng.

Ví dụ : Một hàm 3 biến có giá trị được cho trong hình 1-13.

Ở đây ta quy ước X_3 là bit có trọng số lớn nhất. X_1 là bit có trọng số nhỏ nhất. Trị thập phân của tổ hợp biến được xác định bằng công thức sau :

$$X_3 \cdot 2^2 + X_2 \cdot 2^1 + X_1 \cdot 2^0$$

Nhược điểm của phương pháp này là công kênh, đặc biệt nếu số biến lớn .

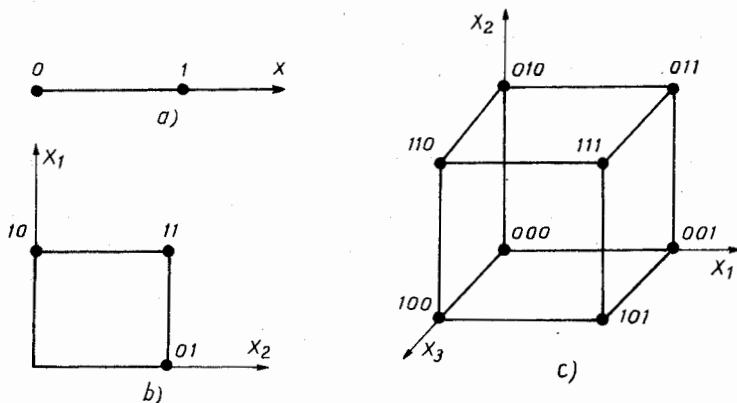
Ưu điểm của nó là trực quan, dễ nhìn, khó nhầm lẫn.

Giá trị thập phân của tổ hợp biến	X_3	X_2	X_1	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	X
3	0	1	1	X
4	1	0	0	0
5	1	0	1	1
6	1	1	0	X
7	1	1	1	1

Hình 1-13. Biểu diễn hàm 3 biến f trên bảng chân lý.

1_4.2. PHƯƠNG PHÁP HÌNH HỌC

Miền xác định của hàm được chuyển thành miền không gian n chiều. Mỗi tổ hợp giá trị biến được biểu diễn bởi một điểm trong không gian đó. Hàm n biến tương ứng với không gian n chiều nghĩa là sẽ có 2^n điểm. Ứng với mỗi điểm sẽ ghi giá trị của hàm tương ứng. Hai điểm nằm trên một cạnh sẽ chỉ khác nhau ở một biến duy nhất. Hình 1-14 biểu diễn hình học cho các hàm 1, 2 và 3 biến.



Hình 1-14. Biểu diễn hình học hàm logic. a . Hàm 1 biến ; b . Hàm 2 biến ; c . Hàm 3 biến.

Nhược điểm của phương pháp này là khi số biến lớn, hình vẽ phức tạp ($n = 4 : 2$ khối lập phương ; $n = 5 : 4$ khối lập phương).

1_4.3. BIỂU DIỄN BẰNG BIỂU THỨC ĐẠI SỐ

Ta công nhận không chứng minh định lý sau :

Định lý : Một hàm logic n biến bất kỳ luôn luôn có thể biểu diễn dưới dạng chuẩn tắc tuyển (CTT) đầy đủ hoặc chuẩn tắc hội (CTH) đầy đủ.

- Dạng CTT đầy đủ : là tuyển của nhiều thành phần, mỗi thành phần là hội (tích) gồm đầy đủ n biến.

- Dạng CTH đầy đủ : là hội của nhiều thành phần, mỗi thành phần là tuyển (tổng) gồm đầy đủ n biến.

Cách viết hàm số dưới dạng CTT đầy đủ :

- Chỉ quan tâm đến các tổ hợp biến mà hàm có giá trị bằng 1. Số lần hàm bằng 1 sẽ chính là số tích của biểu thức.

- Trong mỗi một tích (hội) các biến có giá trị bằng 1 được giữ nguyên, còn các biến có giá trị bằng 0 lấy phủ định, nghĩa là nếu giá trị của $X_i = 1$ thì trong tích sẽ được viết là X_i còn nếu $X_i = 0$ thì trong tích sẽ viết là : X_i phủ định (\bar{X}_i).

- Hàm f bằng tổng các tích đó.

Cách viết hàm số dưới dạng CTH đầy đủ :

- Chỉ quan tâm đến các tổ hợp biến mà hàm có giá trị bằng 0. Số lần hàm bằng 0 sẽ chính là số tổng của biểu thức.

- Trong mỗi một tổng (tuyển) các biến có giá trị bằng 0 được giữ nguyên, còn các biến có giá trị bằng 1 lấy phủ định, nghĩa là nếu giá trị của $X_i = 0$ thì trong tích sẽ được viết là X_i còn nếu $X_i = 1$ thì trong tích sẽ viết là \bar{X}_i .

- Hàm f bằng tích các tổng đó.

Ví dụ : Ta lấy lại ví dụ trong mục 1-4 ở hình 1-13.

Dạng CTT : hàm số $f = 1$ tại các tổ hợp giá trị biến ứng với giá trị thập phân là 0, 5, 7 và được viết trong bảng ở hình 1-15.

Tổ hợp giá trị biến thập phân	Tổ hợp giá trị biến $X_3 \ X_2 \ X_1$	Tích thành phần
0	0 0 0	$\bar{X}_3 \cdot \bar{X}_2 \cdot \bar{X}_1$
5	1 0 1	$X_3 \cdot \bar{X}_2 \cdot X_1$
7	1 1 1	$X_3 \cdot X_2 \cdot X_1$

Hình 1-15. Các tích (hội) đầy đủ của hàm f tại các giá trị thập phân là 0,5,7 .

Như vậy $f = \bar{X}_3 \cdot \bar{X}_2 \cdot \bar{X}_1 + X_3 \cdot \bar{X}_2 \cdot X_1 + X_3 \cdot X_2 \cdot X_1$

Dạng CTH : hàm số $f = 0$ tại các tổ hợp biến ứng với giá trị thập phân là : 1 và 4 và được biểu diễn trong hình 1-16.

Tổ hợp giá trị biến thập phân	Tổ hợp giá trị biến $X_3 \ X_2 \ X_1$	Tổng thành phần
1	0 0 1	$X_3 + X_2 + \bar{X}_1$
4	1 0 0	$\bar{X}_3 + X_2 + X_1$

Hình 1-16. Các tổng đầy đủ ứng với giá trị thập phân là 1 và 4.

Như vậy $f = (X_3 + X_2 + \bar{X}_1) (\bar{X}_3 + X_2 + X_1)$

Ưu điểm của phương pháp này là ngắn gọn .

Để cho giá trị một hàm logic, thường ký hiệu như sau :

- Đối với CTT :

$$f = \sum 0,5,7 \text{ với } N = 2, 3, 6 ;$$

0, 5, 7 là giá trị thập phân của các tổ hợp biến mà giá trị của hàm bằng 1; 2, 3, 6 là giá trị thập phân của các tổ hợp biến mà giá trị của hàm không xác định.

- Đối với CTH :

$$f = \prod 1, 4 \text{ với } N = 2, 3, 6 ;$$

1, 4 là giá trị thập phân của tổ hợp biến mà giá trị của hàm bằng 0; 2, 3, 6 là giá trị thập phân của tổ hợp biến mà giá trị của hàm không xác định.

1.4.4. BIỂU DIỄN BẢNG KARNAUGH

Nguyên tắc xây dựng bảng :

- Để biểu diễn hàm logic n biến cần xây dựng bảng gồm có 2^n ô, mỗi ô tương ứng với một tổ hợp biến.

- Các ô cạnh nhau hoặc đối xứng nhau chỉ khác nhau 1 biến. Các cột và hàng của bảng được ghi các tổ hợp giá trị biến sao cho những cột và hàng cạnh nhau hoặc đối xứng nhau chỉ khác nhau 1 biến.

- Trong các ô ghi giá trị của hàm ứng với giá trị của tổ hợp biến tại ô đó.

Đối với dạng CTT thì các ô tương ứng với $f = 0$ thường được để trống. Đối với dạng CTH thì các ô tương ứng với $f = 1$ thường được để trống. Tại các ô mà hàm số không xác định được đánh dấu X .

Bảng Karnaugh cho trường hợp hàm hai biến được biểu diễn trong hình 1-17.

	X_2		
X_1		0	1
0		0	1
	$\bar{X}_1 \quad \bar{X}_2$	$\bar{X}_1 \quad X_2$	

a)

	X_2		
X_1		0	1
0		0	1
	2	3	

b)

Hình 1-17. Bảng Karnaugh cho hàm 2 biến.

a. Các tổ hợp biến được biểu diễn trong bảng ; b. Ví dụ với $f = \sum 1,2$ và $N = 3$.

Trong hình 1-17a các số ghi ở góc trái trên là giá trị thập phân tương ứng của các tổ hợp biến với qui ước X_2 có trọng số là 2^0 , X_1 có trọng số là 2^1 .

	$X_2 \quad X_3$		
X_1		00	01
0	0 $\bar{X}_1 \bar{X}_2 \bar{X}_3$	1 $\bar{X}_1 \bar{X}_2 X_3$	3 $\bar{X}_1 X_2 \bar{X}_3$
1	4 $X_1 \bar{X}_2 \bar{X}_3$	5 $X_1 \bar{X}_2 X_3$	7 $X_1 X_2 \bar{X}_3$
	2 $\bar{X}_1 X_2 \bar{X}_3$	6 $X_1 X_2 \bar{X}_3$	

(a)

	$X_2 \quad X_3$		
X_1		00	01
0	0	1 1	3 1
1	4 X	5	7 1

(b)

Hình 1-18 . Bảng Karnaugh của hàm 3 biến.

a. Các tổ hợp biến được biểu diễn trong bảng; b. Biểu diễn của hàm $f = \sum 1, 3, 7$ với $N = 2, 4$.

Bảng Karnaugh của hàm 3 biến được biểu diễn trong hình 1-18a. Trong đó các số ghi ở góc trái trên mỗi ô là giá trị thập phân của tổ hợp biến ứng với ô đó, với qui ước X_1 là cột có trọng số lớn nhất 2^2 , còn X_0 là cột có trọng số bé nhất 2^0 .

Hình 1-18b là ví dụ cho hàm $f = \sum 1, 3, 7$ với $N = 2, 4$.

Từ hình 1-18b ta có :

$$f = \bar{X}_1 \bar{X}_2 X_3 + \bar{X}_1 X_2 X_3 + X_1 X_2 X_3$$

đây là dạng CTT đầy đủ của hàm số với giả thiết coi giá trị hàm số bằng 0 tại các tổ hợp biến 2 và 4.

Trên hình 1-19 là biểu diễn trên bảng Karnaugh của hàm 4 biến CTH với :

$$f = \prod 1, 7, 13 \text{ với } N = 2, 3, 11, 15.$$

Dạng CTH của hàm số (với giả thiết tại các tổ hợp biến 2,3,11,15 hàm có giá trị bằng 1) được viết như sau :

$$f = (X_1 + X_2 + X_3 + \bar{X}_4)(X_1 + \bar{X}_2 + \bar{X}_3 + \bar{X}_4)(\bar{X}_1 + \bar{X}_2 + X_3 + \bar{X}_4)$$

		$X_3 X_2$	$X_3 X_3$		
		00	01	11	10
$X_1 X_2$	00	$X_1 + X_2 + X_3 + X_4$	$X_1 + X_2 + X_3 + \bar{X}_4$	$X_1 + X_2 + \bar{X}_3 + \bar{X}_4$	$X_1 + X_2 + \bar{X}_3 + X_4$
01	00	$X_1 + \bar{X}_2 + X_3 + X_4$	$X_1 + \bar{X}_2 + X_3 + \bar{X}_4$	$X_1 + \bar{X}_2 + \bar{X}_3 + \bar{X}_4$	$X_1 + \bar{X}_2 + \bar{X}_3 + X_4$
11	00	$\bar{X}_1 + \bar{X}_2 + X_3 + X_4$	$\bar{X}_1 + \bar{X}_2 + X_3 + \bar{X}_4$	$\bar{X}_1 + \bar{X}_2 + \bar{X}_3 + \bar{X}_4$	$\bar{X}_1 + \bar{X}_2 + \bar{X}_3 + X_4$
10	00	$\bar{X}_1 + X_2 + X_3 + X_4$	$\bar{X}_1 + X_2 + X_3 + \bar{X}_4$	$\bar{X}_1 + X_2 + \bar{X}_3 + \bar{X}_4$	$\bar{X}_1 + X_2 + \bar{X}_3 + X_4$

(a)

		$X_3 X_4$		
		00	01	11
$X_1 X_2$	00	0	1	3
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

(b)

Hình 1-19. Bảng Karnaugh của hàm 4 biến.

a . Các tổ hợp biến ; b . Bảng Karnaugh của hàm $f = \prod 1, 7, 13$ với $N = 2, 3, 11, 15$.

Tương tự bảng Karnaugh cho các hàm 5 và 6 biến được biểu diễn trong hình 1-20 , các số ghi trên mỗi ô là giá trị thập phân của tổ hợp binh ứng với ô đó (qui ước X_1 là cột có trọng số lớn nhất).

$X_1 X_2$	$X_3 X_4$	00	01	11	10	$X_5 = 1$		
00	00	0	2	6	4	5	7	3
01	01	8	10	14	12	13	15	11
11	11	24	26	30	28	29	31	27
10	10	16	18	22	20	21	23	19
								17

a)

$X_1 X_2 X_3$	$X_4 X_5 X_6$	000	001	011	010	110	111	101	100
000	000	0	1	3	2	6	7	5	4
001	001	8	9	11	10	14	15	13	12
011	011	24	25	27	26	30	31	29	28
010	010	16	17	19	18	22	23	21	20
110	110	48	49	51	50	54	55	53	52
111	111	56	57	59	58	62	63	61	60
101	101	40	41	43	42	46	47	45	44
100	100	32	33	35	34	38	39	37	36

b)

Hình 1-20. Bảng Karnaugh của các hàm 5,6 biến.

a . Hàm 5 biến; b . Hàm 6 biến.

1 5. CÁC HỆ THỨC CƠ BẢN VÀ HỆ QUẢ TRONG ĐẠI SỐ LOGIC

1. $A + 0 = A$	2. $A \cdot 1 = A$
3. $A + 1 = 1$	4. $A \cdot 0 = 0$
5. $A + A = A$	6. $A \cdot A = A$
7. $A + \bar{A} = 1$	8. $A \cdot \bar{A} = 0$
9. $A + B = B + A$	10. $A \cdot B = B \cdot A$
11. $A + AB = A$	12. $A(A + B) = A$
13. $AB + A\bar{B} = A$	14. $(A + B)(A + \bar{B}) = A$
15. $A + B + C = (A + B) + C = A + (B + C)$	16. $A \cdot B \cdot C = A(B \cdot C) = (A \cdot B)C$
17. $A + B = \bar{A} \cdot \bar{B}$	18. $A \cdot B = \bar{A} + \bar{B}$

Nhận xét : Nếu thay phép cộng bằng phép nhân , giá trị 0 bằng giá trị 1 và ngược lại thì phương trình ở vế phải chuyển thành phương trình ở vế trái và ngược lại.

Các tính chất 9 và 10 được gọi là tính chất giao hoán.

Tính chất 11 và 12 được gọi là phép nuốt.

Tính chất 13 và 14 được gọi là phép dán.

Tính chất 15 và 16 được gọi là tính kết hợp.

Tính chất 17 và 18 là định lý Demoorgan.

1_6. HỆ HÀM ĐỦ

Định nghĩa : Xét tập hợp $F = \{ f_1(X_1, X_2, \dots, X_n), \dots, f_m(X_1, X_2, \dots, X_n) \}$ trong đó $f_i(X_1, \dots, X_n)$ là một hàm logic n biến , với $i = 1 \div m$.

Tập F là một hệ đủ nếu một hàm logic bất kỳ có thể biểu diễn được bằng một số hữu hạn các hàm $f_i(X_1, X_2, \dots, X_n)$ của F .

Ta biết rằng mọi hàm logic có thể biểu diễn bằng các phép tính : (+) (:) và (-) tức là các hàm AND, OR, NOT là một hệ hàm đầy đủ .

Như vậy muốn chứng minh một hệ hàm cho trước là một hệ đủ ta chỉ cần chứng minh nó có thể biểu diễn các hàm AND, OR và NOT .

Ví dụ : Chứng minh (+, -) là một hệ hàm đủ cần chứng minh hệ này thực hiện được các phép AND, OR, NOT.

- Dương nhiên hệ thực hiện được phép OR, NOT.
- Còn phải chứng minh hệ thực hiện được phép AND.

Thật vậy , phép AND :

$$A \cdot B = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A} + \overline{B}}$$

được thực hiện bởi phép OR và phép NOT. Vậy (+, -) là một hệ đầy đủ.

Ý nghĩa :

Chỉ dùng các hàm hệ đủ có thể xây dựng được một hàm logic bất kỳ.

Một số hệ hàm đủ thông dụng :

- AND, OR, NOT.
- AND, NOT.
- OR, NOT.
- NAND.
- NOR.

1_7. CÁC HÀM NAND VÀ NOR

Các hàm NAND và NOR được sử dụng nhiều hơn các hàm khác vì trong việc chế tạo chúng dùng công nghệ TTL và CMOS có những ưu điểm sau :

- + Giá thành thấp;
- + Có thời gian trễ nhỏ;
- + Công suất tổn hao nhỏ.

Vì vậy chỉ khi có lý do đặc biệt người ta mới không sử dụng chúng vào việc thiết kế mạch logic.

1_7.1. HÀM NAND (VÀ - ĐÁO)

Định nghĩa : Hàm NAND hai biến được biểu diễn bởi phương trình $f = \overline{A \cdot B}$. Ký hiệu của hàm NAND được cho trong hình 1-21a , còn bảng chân lý cho trong hình 1-21b.

Nhận xét: Hàm NAND hai đầu vào sẽ bằng 0 khi và chỉ khi hai đầu vào đều bằng 1.

Mở rộng cho n đầu vào , hàm NAND chỉ bằng 0 khi tất cả các đầu vào đều bằng 1.

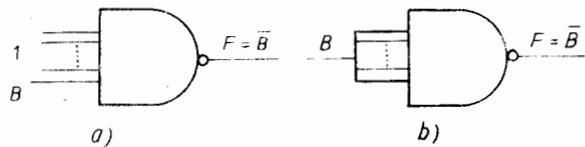
Dùng hàm NAND để xây dựng các hàm logic khác.

NAND là một hệ hàm đủ nên có thể dùng để thực hiện các hàm khác.

a. Tạo hàm NOT (đảo)

NAND có thể sử dụng như một cổng đảo nếu nối $n - 1$ đầu vào của cổng NAND với mức logic 1, đầu vào còn lại chọn làm đầu vào của mạch NOT (hình 1-22a).

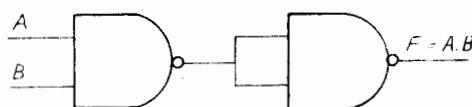
Ta có thể xây dựng mạch NOT bằng cách nối tất cả các đầu vào của mạch NAND với nhau thành đầu vào của mạch NOT (hình 1-22b).



Hình 1-22. Dùng mạch NAND để tạo hàm NOT

b. Tạo hàm AND (Và)

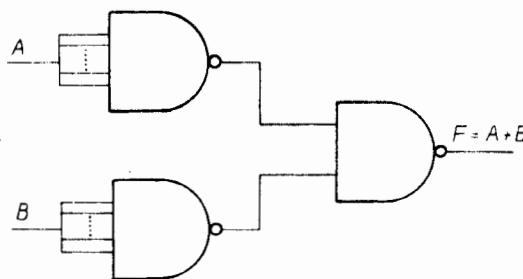
Hàm NAND là đảo của hàm AND do vậy hàm AND được xây dựng từ hàm NAND bằng cách mắc như hình 1-23.



Hình 1-23. Dùng mạch NAND để tạo hàm AND

c. **Tạo hàm OR (Hoặc)**

Như vậy hàm OR có thể được xây dựng từ các mạch NAND như hình 1-24.



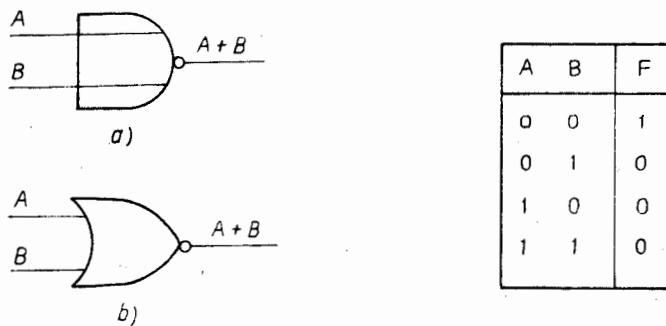
Hình 1-24. Dùng NAND để tạo hàm OR.

1_7.2. HÀM NOR (HOẶC - ĐÁO)

Định nghĩa : Hàm NOR được biểu diễn bằng phương trình $f = \overline{A + B}$. Ký hiệu hàm được chỉ ra trong hình 1-25a,b, bảng chân lý cho trong hình 1-25c.

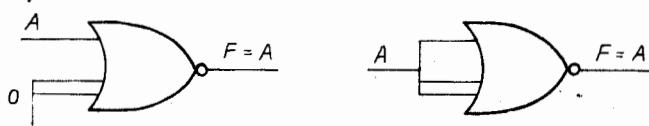
Nhận xét : Hàm NOR hai đầu vào sẽ bằng 0 khi chỉ cần một trong hai đầu vào bằng 1. Mở rộng, hàm NOR n đầu vào sẽ bằng 1 khi và chỉ khi tất cả các đầu vào đều bằng 0.

Hàm NOR là một hệ hàm dù giống như hàm NAND, vì vậy có thể dùng nó để xây dựng các hàm cơ bản khác.



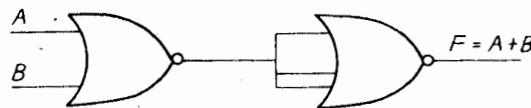
Hình 1-25. Hàm NOR.
a. Vật ; b. Ký hiệu ; c. Bảng chân lý.

a. **Tạo hàm NOT (hình 1.26)**



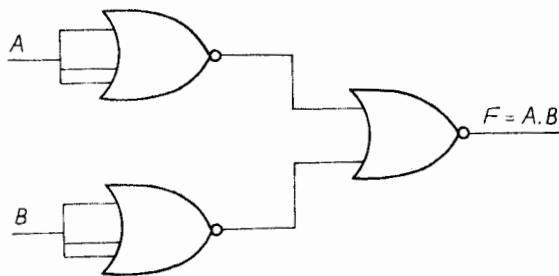
Hình 1-26. Dùng mạch NOR để tạo hàm NOT.

b. Tao hàm OR (hình 1-27)



Hình 1-27. Dùng mạch NOR để tạo hàm OR.

c. Tao hàm AND (hình 1-28)



Hình 1-28. Dùng mạch NOR để tạo hàm AND.

Kết luận : Mạch NAND (NOR) có thể dùng để xây dựng mọi hàm logic cơ bản AND, OR, NOT.

Mạch NAND, NOR là những hệ đú.

1_7.3. PHƯƠNG PHÁP THIẾT KẾ MẠCH DÙNG CÁC CÔNG NAND VÀ NOR

I. Dùng hàm NAND

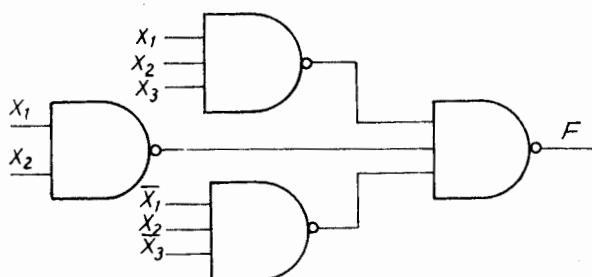
Hai bước cơ bản để thiết kế mạch dùng hàm NAND :

- Viết hàm logic xây dựng ở dạng chuẩn tắc tuyển;
- Thực hiện định lý DeMoorgan với toàn bộ hàm các thành phần không biến đổi.

Ví dụ : Dùng mạch NAND để thiết kế hàm f sau :

$$\begin{aligned} f &= X_1X_2X_3 + X_1X_2 + \overline{X_1X_2X_3} \\ &= \overline{\overline{X_1X_2X_3} + X_1X_2 + \overline{X_1X_2X_3}} \\ &= \overline{\overline{X_1X_2X_3} \cdot \overline{X_1X_2} \cdot \overline{X_1X_2X_3}} \end{aligned} \quad (\text{CTT})$$

Sơ đồ thực hiện hàm f được biểu diễn ở hình 1-29.



Hình 1-29. Sơ đồ hàm f chỉ dùng mạch NAND.

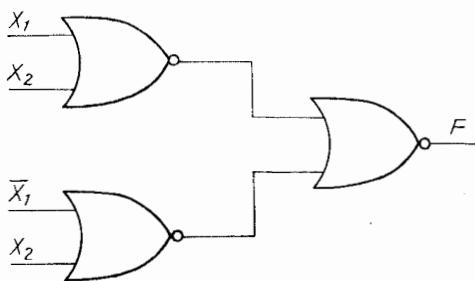
2. Dùng hàm NOR

Các bước thực hiện để xây dựng hàm bất kỳ chỉ dùng mạch NOR :

- Viết hàm logic cần xây dựng ở dạng chuẩn tắc hội.
 - Thực hiện định lý DeMorgan với toàn bộ hàm các thành phần không biến đổi.
- Ví dụ : Chỉ dùng mạch NOR thiết kế hàm sau :

$$\begin{aligned}
 &= (X_1 + X_2)(\bar{X}_1 + X_3) \quad (\text{CTH}) \\
 &= (X_1 + X_2)(\bar{X}_1 + \bar{X}_3) \\
 &= \overline{\overline{X}_1 + X_2} + \overline{\bar{X}_1 + \bar{X}_3}
 \end{aligned}$$

Sơ đồ mạch thực hiện được trong hình 1-30.



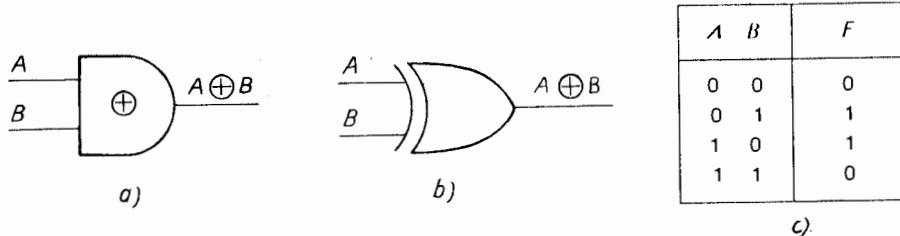
Hình 1-30. Sơ đồ thực hiện hàm / chỉ dùng mạch NOR.

1_8 HÀM XOR

1_8.1. ĐỊNH NGHĨA

Hàm XOR hai biến được biểu diễn bởi phương trình $F = A\bar{B} + \bar{A}B$.

Có 2 cách ký hiệu hàm vẫn được sử dụng trong các tài liệu hiện nay như trong hình 1-31a và 1-31b. Bảng chân lý của hàm được cho trong hình 1-31c.



Hình 1-31. Hàm XOR.
a,b. Ký hiệu ; c. Bảng chân lý.

Ký hiệu : $F = A \oplus B$ hoặc
 $F = A \neq B$.

Nhận xét : Từ bảng chân lý cho trong hình 1-31c của hàm XOR hai đầu vào ta thấy hàm sẽ bằng 1 khi và chỉ khi 2 đầu vào có giá trị khác nhau vì vậy hàm còn được gọi là hàm không tương đương hay hàm cộng modul 2 vì :

$0 \oplus 0 = 0$	$0 \oplus 1 = 1$	$1 \oplus 1 = 0$
------------------	------------------	------------------

1_8.2. TÍNH CHẤT

- Tính giao hoán : $A \oplus B = B \oplus A$
 Tính kết hợp : $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
 Tính phân bố : $A (B \oplus C) = AB \oplus AC$

Ngoài ra dựa vào định nghĩa ta cũng có các tính chất sau :

$$\begin{array}{ll} A \oplus 0 = A & A \oplus A = 0 \\ A \oplus 1 = \bar{A} & A \oplus \bar{A} = 1 \end{array}$$

$$\begin{array}{l} \overline{A \oplus B} = \bar{A} \oplus \bar{B} = A \oplus \bar{B} \\ A \oplus B = \bar{A} \oplus \bar{B} \\ A \oplus B = C \Leftrightarrow A \oplus C = B \Leftrightarrow B \oplus C = A \end{array}$$

1_9. HÀM TƯƠNG ĐƯƠNG

1_9.1. ĐỊNH NGHĨA

Hàm tương đương hai biến được biểu diễn bởi phương trình :

$$f = A B + \bar{A} \bar{B}$$

Ký hiệu hàm tương đương : $f = A \sim B$

Hàm tương đương có hai đầu vào sẽ bằng 1 khi hai đầu vào có giá trị bằng nhau.

1_9.2. TÍNH CHẤT

$A \sim 0 = \bar{A}$	$A \sim \bar{A} = 0$	$\overline{A \sim B} = \bar{A} \sim \bar{B} = A \sim \bar{B}$
$A \sim A = 1$	$A \sim B = B \sim A$	$\overline{A \sim B} = \bar{A} \sim \bar{B}$
$A \sim 1 = A$	$A \sim (B \sim C) = (A \sim B) \sim C$	$\overline{A \sim B} = C \Leftrightarrow A \sim C = B \Leftrightarrow B \sim C = A$

1_10. MỘT SỐ NHẬN XÉT TỔNG QUÁT

Đối với trường hợp nhiều biến , có thể biểu diễn các hàm NAND , NOR , XOR , tương đương như sau.

1_10.1. NAND

$$f = \overline{\prod_{i=1}^n a_i} = a_1 a_2 \dots a_n \quad (1-1)$$

$f = 1$ khi có ít nhất một biến $a_i = 0$, $\forall i = 1 \dots n$

Chứng minh dựa vào tính chất : $A \cdot 0 = 0$ khi có một biến a_i nào đó bằng 0

$\Rightarrow a_1 \cdot a_2 \dots a_{i-1} \cdot 0 \cdot a_{i+1} \dots a_n = 0$ vậy sau khi lấy phủ định sẽ có :

$$f = \overline{\prod_{i=1}^n a_i} = 1$$

1_10.2. HÀM NOR

$$f = \overline{\sum_{i=1}^n a_i} = a_1 + a_2 + \dots + a_n$$

$f = 0$ khi có ít nhất 1 biến $a_i = 1$ hay $f = 1$ khi và chỉ khi $a_i = 0$ với $\forall i$.

Chứng minh : Dựa vào tính chất $A + 1 = 1$. Khi $a_i = 1 ; a_j = 0 (\forall j \neq i)$

$$\Rightarrow a_1 + a_2 + \dots + a_{i-1} + 1 + \dots + a_n = 1$$

$$\Rightarrow f = \overline{\sum_{i=1}^n a_i} = 0$$

1_10.3. HÀM XOR

$$f = \sum_{i=1}^n a_i \quad (1-2)$$

$$Nhận xét : \quad f = \sum_{i=1}^n a_i = 1$$

khi một số lẻ các biến $a_i = 1 ; \forall i = 1 \dots n$.

Chứng minh : Dựa vào tính chất :

$$0 \oplus 0 = 0$$

$$A \oplus 0 = A$$

$$1 \oplus 1 = 0.$$

Do vậy, nếu số biến $a_i = 1$ là một số chẵn và bằng $2m$. Ta có thể chia thành m cặp $1 \oplus 1$, hàm f có dạng :

$$f = \underbrace{0 \oplus 0 \oplus \dots \oplus 0}_{(n - 2m) \text{ số } 0} \oplus \underbrace{(1 \oplus 1) \oplus (1 \oplus 1) \oplus \dots \oplus (1 \oplus 1)}_{m \text{ cặp } (1 \oplus 1)}$$

$$= 0 \oplus 0 \oplus 0 \oplus \dots \oplus 0 \\ = 0$$

Nếu số biến số $a_i = 1$ là một số lẻ và bằng $2m + 1$, tương tự như trên, ta cũng chia thành m cặp $1 \oplus 1$ và cuối cùng còn thừa một biến $a_i = 1$, do vậy :

$$f = 0 \oplus 1 = 1.$$

1_10.4. HÀM TƯƠNG ĐƯƠNG

$$f = \sum_{i=1}^n a_i \quad (1-3)$$

$$f = \sum_{i=1}^n a_i = 0 \quad (1-4)$$

khi một số lẻ các biến $a_i = 0$; ($\forall i = 1 \dots n$)

Chứng minh : Ta có $1 \sim A = A$ nên ta có thể bỏ qua các biến $a_i = 1$ trong biểu thức của f , chỉ xét các biến $a_i = 0$.

- Nếu số các biến $a_i = 0$ là chẵn và bằng $2m$ thì có n cặp $0 \sim 0 = 1$. Do đó : $f = 1$.
- Nếu số các biến $a_i = 0$ là lẻ và bằng $2m + 1$ thì f có dạng : $f = 0 \sim 1 = 0$.

BÀI TẬP

1.1. Một hội đồng giám khảo gồm 3 người . Lập bảng chân lý cho hàm báo hiệu nếu đa số uỷ viên trong hội đồng giám khảo bỏ phiếu thuận.

1.2. Lập bảng chân lý cho hàm sau :

$$Y^4 = A B \bar{D} + \bar{B} C D + \bar{A} \bar{C}$$

1.3. Dùng các phép tính NOT , AND , OR để viết lại các hàm cộng Modul (ký hiệu là \oplus) và tương đương (ký hiệu là \sim) :

$$\begin{aligned} F_3 &= A \oplus B \oplus C \\ F_4 &= A \oplus B \oplus C \oplus D \\ T_3 &= A \sim B \sim C \\ T_4 &= A \sim B \sim C \sim D \end{aligned}$$

Khái quát sự phụ thuộc của kết quả vào số lượng các giá trị 0 và 1 của các biến số trong trường hợp hàm n biến .

1.4. Dùng bảng chân lý để chứng minh đẳng thức sau

$$A \oplus (B \sim C) = (A \oplus B) \sim C$$

1.5. Cho trước các đẳng thức sau

$$\begin{aligned} A \oplus B &= \bar{A} \oplus \bar{B} \\ A \oplus \bar{B} &= \bar{A} \oplus B = \bar{A} \oplus \bar{B} = A \sim B \\ A \oplus B &= \bar{A} \sim B = A \sim \bar{B} = \bar{A} \sim \bar{B} \end{aligned}$$

Dùng các đẳng thức trên để chứng minh :

$$A \oplus B \oplus C = A \sim B \sim C$$

1.6. Xác định \bar{Y} , biết $Y = (A + B \bar{C}) \bar{D}$

1.7. Hãy chứng minh các đẳng thức sau :

$$A + \bar{A}B = A + B$$

$$AB + \bar{A}C + BC = AB + \bar{A}C$$

$$AB + \bar{A}C = (A + C)(\bar{A} + B)$$

1.8. Đơn giản các biểu thức sau :

$$Y = \bar{A} + AB\bar{C} + (\bar{A} + AB\bar{C})(A + \bar{A}\bar{B}C)$$

$$Y = (AB + CD)(A\bar{C} + B\bar{D})$$

$$Y = A\bar{C}D + \bar{A}C\bar{D} + \bar{A}\bar{B}\bar{C} + CD$$

$$Y = B(AC + \bar{A}\bar{C}) + A\bar{C} + A\bar{C}$$

$$Y = ACD + ABC + D(\bar{A} + \bar{B}) + \bar{A}CD$$

1.9. Chứng minh rằng nếu : $A = BC + \bar{B}C$ thì :

$$\bar{A} + AB = 0$$

$$AB = AC$$

$$AB + A\bar{C} + C = \bar{C}D$$

1.10. Với các giá trị nào của các biến số $\bar{A}B + C\bar{D} = 0$ thì

$$AB + \bar{C}(\bar{A} + \bar{D}) = AB + BD + \bar{B}\bar{D} + \bar{A}\bar{C}D$$

CHƯƠNG 2

TỐI THIỂU HOÁ HÀM BOOLE

2_1. KHÁI NIỆM TỐI THIỂU HOÁ HÀM BOOLE

Trước đây, khi kỹ thuật vi điện tử chưa phát triển, tối thiểu hoá hàm Boole là một trong những vấn đề cơ bản của lý thuyết tổng hợp mạch logic. Kỹ thuật vi điện tử ra đời, đặc biệt sự ra đời của các mạch tích hợp cỡ vừa (MSI), cỡ lớn (LSI) và cực lớn (VLSI) làm cho việc tối thiểu hoá không còn ý nghĩa như trước nữa. Tuy nhiên, trong quá trình phân tích và thiết kế các mạch logic đơn giản, nhiều lúc vẫn dùng đến một số khái niệm liên quan đến vấn đề tối thiểu hoá. Trong phần này sẽ trình bày những kiến thức cơ bản nhất về tối thiểu hoá hàm Boole.

Xét ví dụ sau :

Cho hàm $f(X_3, X_2, X_1)$ có bảng chân lý được biểu diễn ở hình 2-1a.

Biểu diễn hàm ở dạng chuẩn tắc tuyến (CTT) đầy đủ :

$$f = \bar{X}_3 X_2 \bar{X}_1 + X_3 X_2 \bar{X}_1 + X_3 X_2 X_1 \quad (2-1)$$

hoặc ở dạng chuẩn tắc hội (CTH) đầy đủ :

$$f = (X_3 + X_2 + X_1)(X_3 + X_2 + \bar{X}_1)(\bar{X}_3 + X_2 + X_1)(\bar{X}_3 + X_2 + \bar{X}_1). \quad (2-2)$$

Sơ đồ thực hàm f ở dạng CTT đầy đủ và CTH đầy đủ được biểu diễn trên hình 2-1,b và 2-1,c.

Mặt khác nhìn vào bảng chân lý và tận dụng tổ hợp biến tại đó hàm không xác định, ta có thể viết :

$$f = X_2 \quad (2-3)$$

Rõ ràng biểu thức (2-3) sẽ cho một sơ đồ đơn giản hơn rất nhiều (hình 2-1,d).

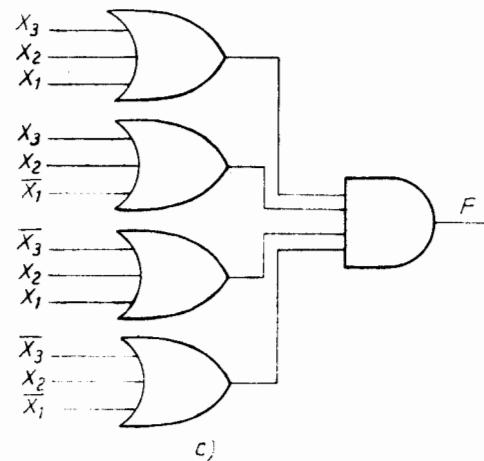
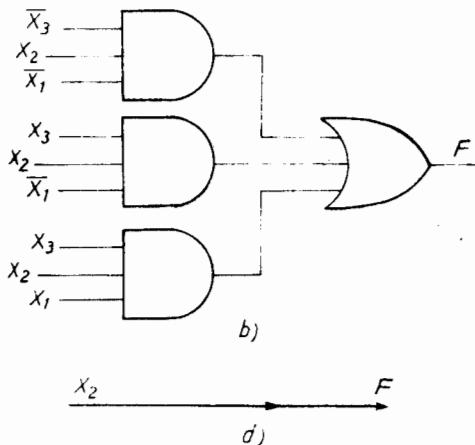
Tất nhiên không phải với hàm nào cũng chỉ cần nhìn vào bảng chân lý là tìm ngay được dạng biểu diễn đơn giản nhất. Do số trường hợp ta phải dùng đến công cụ "tối thiểu hoá hàm Boole". Thực chất của vấn đề tối thiểu hoá là tìm dạng biểu diễn đại số đơn giản nhất của hàm. Dạng đơn giản nhất ở đây còn tùy thuộc vào hàm được biểu diễn ở dạng CTT hay ở dạng CTH.

Sau đây sẽ xét vấn đề tối thiểu hoá hàm biểu diễn ở dạng CTT không xác định đầy đủ và đưa ra cách áp dụng cho hàm ở dạng CTH ở phần cuối.

Nếu ký hiệu số tích của hàm ở dạng CTT là n và giả thiết rằng sơ đồ thực hiện mạch chỉ gồm hai tầng: tầng 1 là các mạch AND (thực hiện các tích) và tầng 2 là một mạch OR (thực hiện phép tuyển các tích ở tầng 1). Các mạch AND và OR có số đầu vào không hạn chế.

X_3	X_2	X_1	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

a)



Hình 2-1. Hàm logic và các sơ đồ thực hiện.

- a. Bảng chân lý của hàm ; b. Sơ đồ thực hiện hàm ở dạng CTT đầy đủ ;
c. Sơ đồ thực hiện hàm ở dạng CTH đầy đủ ; d. Sơ đồ thực hiện hàm đơn giản.

Để đánh giá độ đơn giản trong cách biểu diễn ở dạng CTT người ta thường dùng một trong hai thông số sau.

a. Tổng số đầu vào ở tầng 1 : C^a

$$C^a = \sum_{i=1}^n S_i \quad (2-4)$$

với S_i : số biến trong tích thứ i của hàm;
 n : số tích của hàm.

b. Tổng số đầu vào của mạch (cả 2 tầng)

$$C^b = C^a + n . \quad (2-5)$$

Ví dụ với hàm ở dạng CTT có sơ đồ như ở hình 2-1.b, ta có :

$$C^a = \text{Số đầu vào tầng một} = 3 + 3 + 3 = 9;$$

$$C^b = \text{Số đầu vào của mạch} = 9 + 3 = 12.$$

Kết quả của tối thiểu hóa phải đưa ra được một cách biểu diễn hàm sao cho C^a hoặc C^b là cực tiểu. Đối với một hàm số cho trước C_{\min}^a , C_{\min}^b là duy nhất, tuy nhiên có thể có nhiều cách biểu diễn ứng với C_{\min}^a , C_{\min}^b ấy. Trong các phân trình bày sau này ta sẽ đánh giá độ tối thiểu của hàm số qua C^a .

Để cho tiện, ta ký hiệu C^a là C , C_{\min}^a là C_{\min} . Nhờ quan hệ của C^a và C^b có thể suy ra C^b và C_{\min}^b tương ứng.

2-2. CÁC PHƯƠNG PHÁP TỐI THIỂU HOÁ HÀM BOOLE

Việc tối thiểu hóa hàm Boolean nói chung có thể đưa về một trong hai nhóm :

- Biến đổi đại số;
- Thuật toán .

Cơ sở toán của các phương pháp này là các định lý, hệ quả và tính chất của đại số Boolean.

2.2.1. PHƯƠNG PHÁP TỐI THIỂU HOÁ BẰNG BIẾN ĐỔI ĐẠI SỐ

Hàm được đưa về biểu diễn ở dạng biểu thức và biến đổi một cách trực tiếp theo xu hướng giảm dần giá trị của C . Sự rút gọn thực hiện trên cơ sở các định lý :

$$A + \bar{A} = 1, A \cdot \bar{A} = 0, A + A = A, A \cdot A = A \quad (2-6)$$

Ví dụ cho hàm số :

$$\begin{aligned} f &= \bar{A}X + AX + A\bar{X} && (C = 6) \\ &= (\bar{A}X + AX) + (AX + A\bar{X}) \\ &= X(A + \bar{A}) + A(X + \bar{X}) \\ &= X + A. && (C_{\min} = 2) \end{aligned}$$

Do tính trực quan của phương pháp, kết quả đưa ra nhiều khi không biết được là đã tối thiểu hay chưa. Đây không phải là phương pháp chặt chẽ cho phép tự động hóa quá trình tối thiểu hóa.

2.2.2. NHÓM PHƯƠNG PHÁP TỐI THIỂU HOÁ THEO THUẬT TOÁN

Tiêu biểu là hai phương pháp sau :

- Bảng Karnaugh
- Quine - Mc. Cluskey.

Dưới đây ta sẽ nghiên cứu cả hai phương pháp này cho hàm biểu diễn ở dạng CTT xác định không đầy đủ, nghĩa là có những tổ hợp hợp biến mà tại đó hàm không xác định. Hàm xác định đầy đủ chỉ là một trường hợp riêng của hàm xác định không đầy đủ.

2-3. TỐI THIỂU HÓA BẰNG PHƯƠNG PHÁP QUINE-MC.CLUKEY

2-3.1. MỘT SỐ KHÁI NIỆM

a. Đỉnh

Đỉnh là một tích gồm đầy đủ các biến của hàm ban đầu (nếu hàm có n biến đỉnh là tích n biến)

Đỉnh 1 là đỉnh tại đó hàm số bằng 1.

Đỉnh 0 là đỉnh tại đó hàm số bằng 0.

Đỉnh không xác định là đỉnh tại đó hàm không xác định.

Thông thường khi cho một hàm số ở dạng CTT người ta cho tập các đỉnh 1 (L) và tập các đỉnh không xác định (N) của hàm ban đầu.

Ví dụ : Tối thiểu hóa hàm $f(X_3, X_2, X_1)$;

có $L = 2, 3, 7$ (các đỉnh 1);

$N = 1, 6$ (các đỉnh không xác định ký hiệu là X).

Các đỉnh này được gọi tên theo tích tương ứng theo mã nhị phân của chúng hoặc số thập phân tương ứng với mã nhị phân này (hình 2.2).

Số thập phân	1(X)	2	3	6(X)	7
Mã nhị phân	001	010	011	110	111
Tích	$\bar{X}_3 \bar{X}_2 X_1$	$\bar{X}_3 X_2 \bar{X}_1$	$\bar{X}_3 X_2 X_1$	$X_3 X_2 \bar{X}_1$	$X_3 X_2 X_1$

Hình 2-2. Ký hiệu các đỉnh.

b. Tích cực tiểu

Là một tích mà tại đó hàm bằng 1 hoặc không xác định với thành phần các biến không bô bót đi được nữa.

Tích cực tiểu là biểu diễn của 1 nhóm 2^k đỉnh (gồm những đỉnh 1 và đỉnh không xác định). Ta nói nó phủ những đỉnh này hay các đỉnh này được chứa trong nó. Nhóm 2^k đỉnh này là cực đại do vậy tích biểu diễn nó sẽ có số biến cực tiểu – tích cực tiểu. (Nếu hàm ban đầu có n biến, tích cực tiểu phủ 2^k đỉnh, thì số biến của tích là $n - k$).

Ý nghĩa : Tích cực tiểu là tích có số biến ít nhất phủ 2^k đỉnh 1 hoặc X của hàm số. Cơ sở toán của việc tìm các tích cực tiểu là áp dụng phép dán :

$$A X + A \bar{X} = A. \quad (2-7)$$

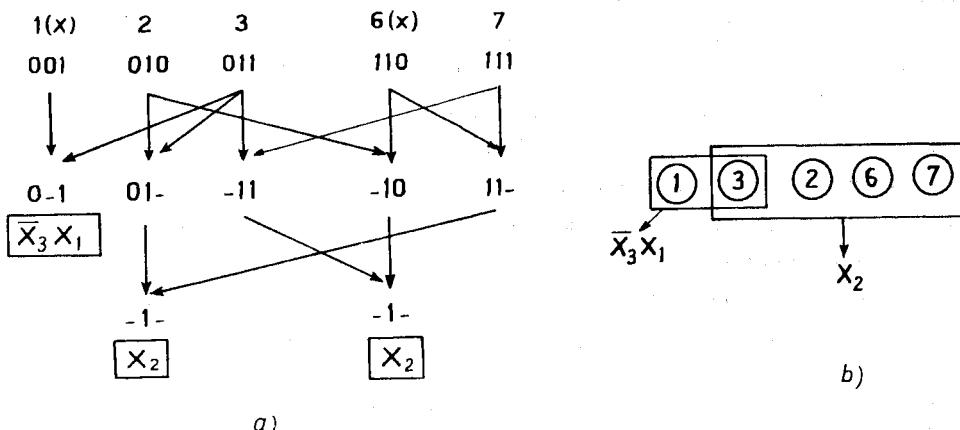
Việc tìm các tích cực tiểu của hàm cho ví dụ trên được biểu diễn trong hình 2-3a, Sơ đồ phủ đỉnh của các tích này cho ở hình 2-3b.

Đỉnh đánh dấu là đỉnh 1 của hàm số và đỉnh này được phủ duy nhất bởi một tích cực tiểu.

c. Tích quan trọng

Tích quan trọng là 1 tích cực tiểu và phủ ít nhất 1 đỉnh đánh dấu.

Ý nghĩa : Tối thiểu hoá hàm f nghĩa là tìm phủ tối thiểu của hàm f - phủ hết các đỉnh 1 của hàm số . Nếu một đỉnh 1 của hàm số chỉ được phủ duy nhất bởi một tích cực tiểu thì tích đó nhất định phải có mặt trong phủ tối thiểu. Tích đó chính là tích quan trọng.

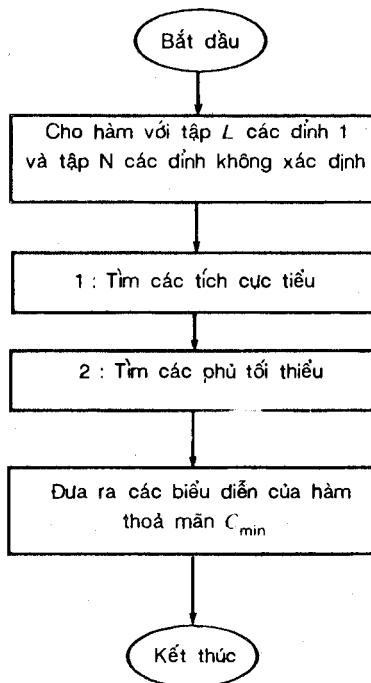


Hình 2-3. Tích cực tiểu.

a. Tìm các tích cực tiểu ; b. Sơ đồ phủ đỉnh của tích cực tiểu .

2-3.2 PHƯƠNG PHÁP QUINE-McCLUSKEY

Quá trình tối thiểu hoá gồm các bước sau(hình 2-4) :



Hình 2-4. Các bước tối thiểu hóa.

Ví dụ : Tối thiểu hóa hàm $f(X_1, X_2, X_3, X_4)$ có
 $L = 2, 3, 7, 12, 14, 15$
 $N = 6, 13.$

Giai đoạn 1 : Tìm các tích cực tiểu .

Giai đoạn này gồm các bước sau (hình 2-5) :

1. Biểu diễn các đỉnh 1 và đỉnh không xác định của hàm dưới dạng mā nhị phân bảng a.

2. Sắp xếp các tổ hợp mā trên theo số lượng chữ số 1 có trong chúng. ta thu được bảng mới gồm các nhóm có lượng chữ số 1 bằng 0, 1, 2, 3 ... (bảng b)

3. So sánh mỗi tổ hợp thuộc nhóm thứ i với một tổ hợp thuộc nhóm thứ $i + 1$. Nếu 2 tổ hợp đó chỉ khác nhau ở 1 cột số thì kết hợp 2 tổ hợp đó thành một tổ hợp mới, trong đó thay cột số khác nhau của hai tổ hợp cũ bằng một gạch ngang (-) đồng thời đánh dấu ký hiệu V vào 2 tổ hợp cũ (bảng c). Ở đây chúng ta đã sử dụng tính chất :

$$X Y + X \bar{Y} = X .$$

4. Loại bỏ các tổ hợp giống nhau trong bảng mới thành lập và lập lại bước 3, cho đến khi hết khả năng kết hợp các tổ hợp với nhau thì thôi (bảng d).

5 . Tập hợp các tổ hợp nằm trong bảng cuối cùng và các tổ hợp không có đánh dấu V chính là tập các tích cực tiểu của hàm Z. Giai đoạn một kết thúc ở đây.

Bảng a		Số chữ số 1	Bảng b		Bảng c		Bảng d	
2	0010	1	2	0010 V	2,3	001-V	2, 3, 6, 7	0- 1 -
3	0011		3	0011 V	2,6	0-10 V	6, 7, 14, 15	-1 1 -
6	0110		6	0110 V	3,7	0-11 V	12,13, 14,15	1 1 --
7	0111		12	1100 V	6,7	011 - V		
12	1100		7	0111 V	6,14	-110 V		
13	1101		13	1101 V	12,13	110 - V		
14	1110		14	1110 V	12,14	11-0 V		
15	1111		15	1111 V	7,15	-111 V		
					13,15	11-1 V		
					14,15	111 - V		

Hình 2-5. Tìm các tích cực tiểu.

Như vậy tập các tích cực tiểu (Z) của hàm f gồm các khối sau :

0-1- (phù các đỉnh 2, 3, 6, 7) : $\bar{X}_1 X_3$

-11- (phù các đỉnh 6, 7, 14, 15) : $X_2 X_3$

11-- (phù các đỉnh 12, 13, 14, 15) : $X_1 X_2$

Giai đoạn 2 : Tìm phủ tối thiểu .

Phủ tối thiểu phải phủ hết các đỉnh 1 của hàm và có C đạt C_{\min} . Do vậy giai đoạn này

chỉ quan tâm đến các đỉnh 1 mà không cần xét các đỉnh không xác định của hàm. Một khái niệm để đạt được C_{\min} phải tiến hành loại trừ các tích cực tiểu không cần thiết và giữ lại các tích quan trọng "nhất thiết phải có mặt trong kết quả tối thiểu".

Quá trình xác định các tích quan trọng tiến hành theo nhiều bước và dựa vào khái niệm "tích quan trọng ở bước thứ i ". Một số các ký hiệu được sử dụng :

L_i : tập các đỉnh 1 đang xét ở bước thứ i ;

Z_i : tập các tích cực tiểu đang ở bước thứ i ;

E_i : tập các tích quan trọng ở bước thứ i , với $i = 0, 1, 2, 3, \dots$

Cho trước L và Z (tập các tích cực tiểu – được xác định ở giai đoạn 1). xác định C_{\min} .

Giai đoạn 2 được tiến hành như sau :

1. Với $i = 0$

$$L_0 = L = \{2, 3, 7, 12, 14, 15\}$$

$$Z_0 = Z = \{X_1 X_3, X_2 X_3, X_1 X_2\}$$

Xác định E_0 (tập các tích quan trọng ở bước 0) từ L_0 và Z_0 theo cách sau :

Lập một bảng trong đó mỗi hàng tương ứng với 1 tích cực tiểu thuộc Z_0 , mỗi cột tương ứng 1 đỉnh thuộc L_0 . Dán dấu X vào ô (m, n) nếu tích cực tiểu ở hàng thứ m phủ đỉnh ở cột thứ n .

Xét từng cột, cột nào chỉ có một dấu X , thì đánh dấu \otimes . Đinh tương ứng với cột đó là đỉnh đánh dấu, tích cực tiểu tương ứng có chứa đỉnh đó là tích quan trọng (bảng hình 2-6).

L_0	2	3	4	12	14	15
Z_0						
$X_1 X_3$	\otimes	\otimes	X			
$X_2 X_3$			X		X	X
$X_1 X_2$				\otimes	X	X

Hình 2-6. Xác định tập các tích quan trọng E_0 .

Với ví dụ này, các đỉnh đánh dấu là : 2, 3, 12 và tập các tích quan trọng E_0 là :

$$E_0 = (X_1 X_3, X_1 X_2).$$

2. Với $i = 1$

$L_1 = L_0$ – Các đỉnh 1 được phủ ở E_0 : loại khỏi L_0 những đỉnh 1 mà E_0 đã phủ.

$Z_1 = Z_0 - E_0$ – Các tích không cần thiết : loại khỏi Z_0 các tích quan trọng nằm trong E_0 và các tích không cần thiết.

Lập 1 bảng tương tự như trên ứng với L_1 và Z_1 , từ bảng đó tìm tập các tích quan trọng E_1 .

3. Tiếp tục lặp lại :

$L_{i+1} = L_i$ – Các đỉnh thuộc các tích nằm trong E_i

$Z_{i+1} = Z_i - E_i$ – Các tích không cần thiết.

Lập bảng cho L_{i+1} và Z_{i+1} để tìm E_{i+1} .

Lặp lại bước này cho đến khi $L_k = \emptyset$ thì dừng lại. Nhận được kết quả :

$$C_{\min} = \bigvee_{j=0}^{k-1} E_j$$

Ở ví dụ trên, $L_1 = \emptyset$ nên

$$C_{\min} = E_0 = \{ \bar{X}_1 X_3, X_1 X_2 \}.$$

Hàm tìm được ở dạng tối thiểu :

$$f = \bar{X}_1 X_3 + X_1 X_2.$$

2_4. PHƯƠNG PHÁP TỐI THIỂU HOÁ BẢNG KARNAUGH CHO HÀM DẠNG CHUẨN TẮC TUYỀN

Phương pháp này được tiến hành theo các bước như sau :

1. Biểu diễn hàm đã cho trên bảng Karnaugh

2. Xác định các tích cực tiểu của hàm

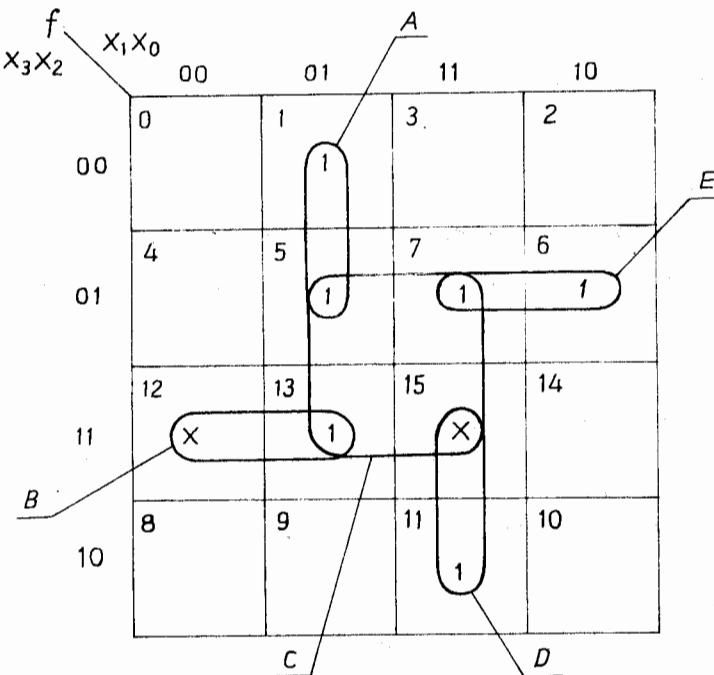
Tích cực tiểu được tìm bằng cách dán 2^k ô có đánh dấu 1 hoặc X với k tối đa. Các ô này kề nhau hoặc đối xứng với nhau trên bảng Karnaugh.

3. Tìm phủ tối thiểu

Chọn 1 số ít nhất các nhóm tích cực tiểu sao cho phủ được hết các đỉnh 1 của hàm. Dưới đây là một số ví dụ minh họa.

Ví dụ 1. Hãy tối thiểu hóa hàm

$$f(X_3, X_2, X_1, X_0) = \sum 1, 5, 6, 7, 11, 13 ; \quad N = 12,15$$



Hình 2-7. Tối thiểu hóa trên bảng Karnaugh.

Hàm được cho trên bảng Karnaugh 4 biến, các tích cực tiểu được xác định như hình 2-7, với :

$$A = \bar{X}_3 \bar{X}_1 X_0$$

$$B = X_3 X_2 \bar{X}_1$$

$$C = X_2 X_0$$

$$D = X_3 X_1 X_0$$

$$E = \bar{X}_3 X_2 X_1$$

Nhóm tối thiểu các tích cực tiểu phủ hết các đỉnh 1 của hàm là : A , C , D , E . Vậy :

$$f(X_3, X_2, X_1, X_0) = \bar{X}_3 \bar{X}_1 X_0 + X_2 X_0 + X_3 X_1 X_0 + \bar{X}_3 X_2 X_1$$

$$(C_{\min} = 11).$$

Ví dụ 2. Cho hàm $f(X_4, X_3, X_2, X_1, X_0)$ được biểu diễn trên bảng Karnaugh như ở hình 2-8.

$X_3 X_2$	$X_1 X_0$	00	01	11	10	10	11	01	00
$X_3 X_2$	00	1	1	X		1			
$X_3 X_2$	01	1	X	1	1	1	1	1	
$X_3 X_2$	11	1				1		1	
$X_3 X_2$	10					1			

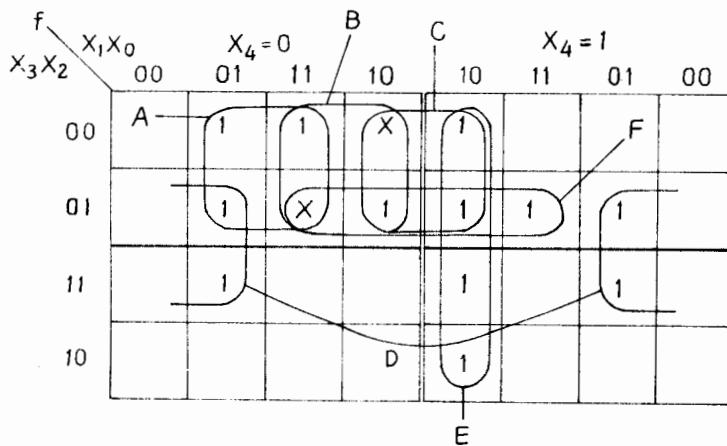
Hình 2-8. Bảng Karnaugh của hàm f.

Các tích cực tiểu của hàm được trình bày ở hình 2-9.

Nhận thấy A , D , F , E là các tích quan trọng đồng thời cũng phủ hết các đỉnh của hàm, như vậy $C_{\min} = (A, D, E, F)$.

$$f(X_4, X_3, X_2, X_1) = A + D + E + F = \bar{X}_4 \bar{X}_3 X_0 + X_2 \bar{X}_1 X_0 + X_4 X_1 \bar{X}_0 + \bar{X}_3 X_2 X_1$$

$$C_{\min} = 12.$$



Hình 2-9. Các tích cực tiểu của hàm f.

2_5. TỐI THIỂU HOÁ HÀM Ở DẠNG CHUẨN TẮC HỘI

Tương tự như hàm biểu diễn ở dạng CTT, để đánh giá độ đơn giản của hàm biểu diễn ở dạng CTH, người ta cũng dùng 2 thông số sau.

a) *Tổng số đầu vào tầng I :*

$$C^A = \sum_{i=1}^n S_i \quad (2-7)$$

S_i là số biến của tổng thứ i , n là số tổng của hàm.

b) *Tổng số đầu vào của mạch :*

$$C^B = C^A + n \quad (2-7)$$

Ví dụ :

$$\begin{aligned} f(X_3, X_2, X_1) &= (X_3 + X_2 + X_1)(\bar{X}_3 + X_2 + X_1)(X_3 + \bar{X}_2) \\ \Rightarrow C^A &= 3 + 3 + 2 = 8. \end{aligned}$$

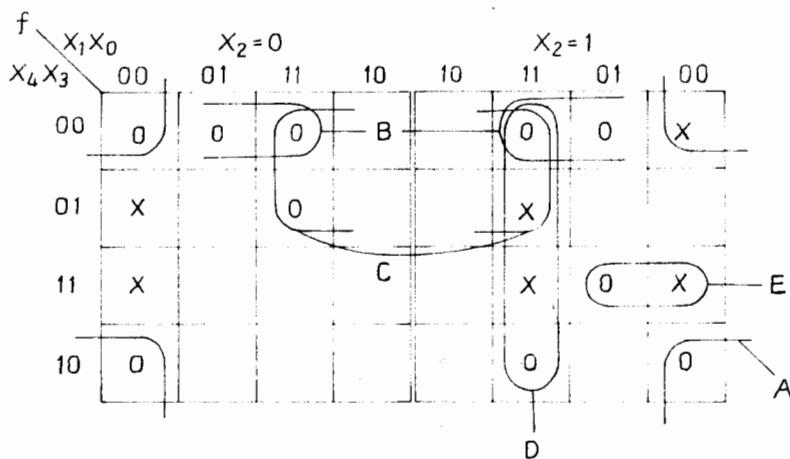
Việc tối thiểu hoá hàm viết dưới dạng chuẩn tắc hội (CTH) cũng tương tự như tối thiểu hoá hàm viết dưới dạng chuẩn tắc tuyến (CTT), khác ở chỗ :

- Thay các đinh 1 (tích tại đó hàm bằng 1 trong CTT) bằng các đinh 0 (tổng tại đó hàm bằng 0 trong dạng CTH).
- Thay tổng các tích (trong CTT) bằng tích các tổng (trong CTH) khi biểu diễn hàm.

Sau đây chúng ta sẽ xét một ví dụ tối thiểu hoá hàm ở dạng CTH, dùng phương pháp bảng Karnaugh.

Ví dụ : Tối thiểu hoá hàm

$$\begin{aligned} f &= \prod (0, 1, 3, 5, 7, 11, 16, 20, 23, 29) \text{ các đinh 0} \\ N &= (4, 8, 15, 24, 28, 31) \text{ các đinh không xác định.} \end{aligned}$$



Hình 2-10. Tối thiểu hoá hàm ở dạng CTH.

Biểu diễn hàm này trên bảng Karnaugh (hình 2-10). Sau khi tìm các tổng cực tiểu, ta thấy các tổng A, B, C, D, E là các tổng quan trọng, phủ tất cả các đỉnh 0 của hàm, ta có :

$$f_{\min}(X_4 X_3, X_2, X_1, X_0) = \\ (X_3 + X_1 + X_0)(X_4 + X_3 + \bar{X}_0)(X_4 + \bar{X}_1 + \bar{X}_0)(\bar{X}_2 + \bar{X}_1 + \bar{X}_0)(\bar{X}_4 + \bar{X}_3 + \bar{X}_2 + X_1) \\ C_{\min}^* = 13.$$

BÀI TẬP

2.1. Tối thiểu hoá hàm sau ở dạng CTT theo phương pháp Quine-Mc. Cluskey :

$$f(X_3, X_2, X_1, X_0) = \sum 0, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 15.$$

2.2. Tối thiểu hoá hàm sau ở dạng chuẩn tắc hội theo phương pháp Quine-Mc Cluskey :

$$f(X_3, X_2, X_1, X_0) = \prod(3, 4, 6, 7, 8, 9, 12, 13, 14)$$

2.3. Tối thiểu hoá hàm sau ở dạng chuẩn tắc tuyển theo phương pháp bảng Karnaugh :

$$f(X_3, X_2, X_1, X_0) = \sum 0, 1, 2, 5, 7, 10, 14, 15$$

2.4. Tối thiểu hoá hàm sau ở dạng chuẩn tắc hội theo phương pháp bảng Karnaugh :

$$f(X_4, X_3, X_2, X_1, X_0) = \prod (0, 1, 12, 13, 16, 17)$$

$$N = 8, 9, 24, 25$$

2.5. Viết dạng đại số đơn giản nhất cho các hàm sau :

$$Y_a(A, B, C) = \sum (0, 2, 3, 4, 6)$$

$$Y_b(A, B, C) = \prod (0, 1, 4, 5, 6).$$

CHƯƠNG 3

VI MẠCH

3_1. ĐỊNH NGHĨA VÀ PHÂN LOẠI

Vi mạch – (Integrated circuits) là những linh kiện điện tử rất nhỏ bé nhưng phức tạp, mỗi vi mạch có một chức năng xác định và được chế tạo bằng một công nghệ thích hợp. Vi mạch hiện đại thường đa năng và có thể sử dụng linh hoạt trong nhiều thiết bị điện tử khác nhau.

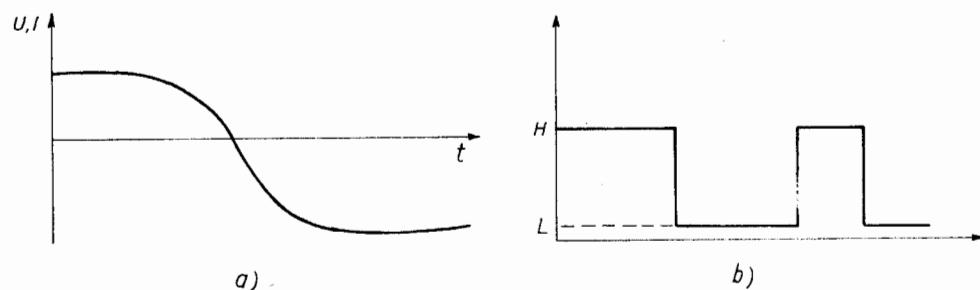
Có nhiều cách phân loại vi mạch :

- Phân loại theo bản chất của tín hiệu điện vào, ra của vi mạch.
- Phân loại theo mức độ tích hợp.
- Phân loại theo công nghệ chế tạo.

3_2. PHÂN LOẠI VI MẠCH THEO BẢN CHẤT CỦA TÍN HIỆU ĐIỆN VÀO, RA

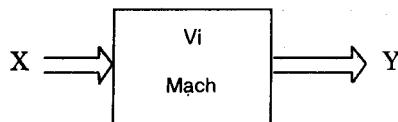
Tín hiệu điện được phân thành 2 loại :

- Tín hiệu tương tự (analog) : biên độ biến thiên liên tục theo thời gian trong một giới hạn cho phép (hình 3-1,a). Ví dụ : điện áp hoặc dòng điện xoay chiều.
- Tín hiệu số (digital) : biên độ tín hiệu chỉ có thể quy vào 2 giá trị hữu hạn mang ý nghĩa logic 0 hay 1, ứng với 2 mức thấp L (low) và cao H (high). Tín hiệu gián đoạn theo thời gian (hình 3-1,b).



Hình 3-1. Phân loại tín hiệu điện .
a. Tín hiệu liên tục ; b. Tín hiệu số.

Nếu ký hiệu : X là tín hiệu vào của vi mạch, Y là tín hiệu ra của vi mạch, theo bản chất của tín hiệu X , Y có những loại mạch sau :



X, Y : tín hiệu tương tự

: Vi mạch tương tự

X, Y : tín hiệu số

: Vi mạch số (Vi mạch logic)

X : Tương tự (Analog)

{ Mạch biến đổi tương tự - số

Y : Số (Digital)

{ Analog - Digital Converter (ADC)

X : Số (Digital)

{ Mạch biến đổi số - tương tự

Y : Tương tự (Analog)

{ Diginal - Analog Converter (DAC)

Ở đây chỉ xét đến các vi mạch số là vi mạch có tín hiệu vào và tín hiệu ra đều là tín hiệu số. Các vi mạch tương tự, các bộ ADC và DAC được trình bày trong các tài liệu khác.

Vi mạch số bao gồm từ các cổng logic đơn giản như các mạch AND, OR, ... các mạch lật, các mạch đếm, các mạch dồn kênh, phân kênh ... cho đến các loại mạch phức tạp hơn hẳn như bộ nhớ, các mạch vào/ra, các mạch điều khiển ngoại vi, các mạch vi tính, vi xử lý ... Vi mạch số được sử dụng rộng rãi trong mọi lĩnh vực.

3-3. PHÂN LOẠI THEO MẬT ĐỘ TÍCH HỢP

Mật độ tích hợp được định nghĩa là tổng số những phần tử tích cực (transistor) hoặc cổng logic (tùy theo cách đánh giá khác nhau chưa thống nhất) chứa trên một đơn vị diện tích của mảnh tinh thể bán dẫn trong vi mạch.

Mức tích hợp được định nghĩa là tổng số những phần tử tích cực hoặc cổng logic chứa trên mảnh tinh thể bán dẫn của vi mạch.

Loại mạch	Số transito	Số cổng logic	Năm sản xuất	Ví dụ
SSI	Hàng chục	Vài	Đầu những năm 60	7400, 7406, 7400, ...
MSI	Hàng trăm	Vài chục	Giữa những năm 60	Giải mã, bộ ghi, bộ nhớ nhỏ MUX DEMUX, ... 7447, 7483, ...
LSI	Hàng nghìn	Vài trăm	Cuối 60 giữa 70	Bộ nhớ lớn, MP 4.8 bit, bộ phối ghép vào/ra
VLSI	Hàng vạn	Vài nghìn	Cuối 70 đầu 80 đến nay	Bộ nhớ cực lớn, MP 16/32 bit, bộ điều khiển đa năng phức tạp 8086, Z8000, 68000 NS3200, NH7200.

Hình 3-2. Giới thiệu các loại vi mạch theo mức tích hợp.

Những thông số này tuy không phải luôn đúng nhưng cũng nói lên mức độ phức tạp của vi mạch ấy. Theo mức độ tích hợp, có những loại sau :

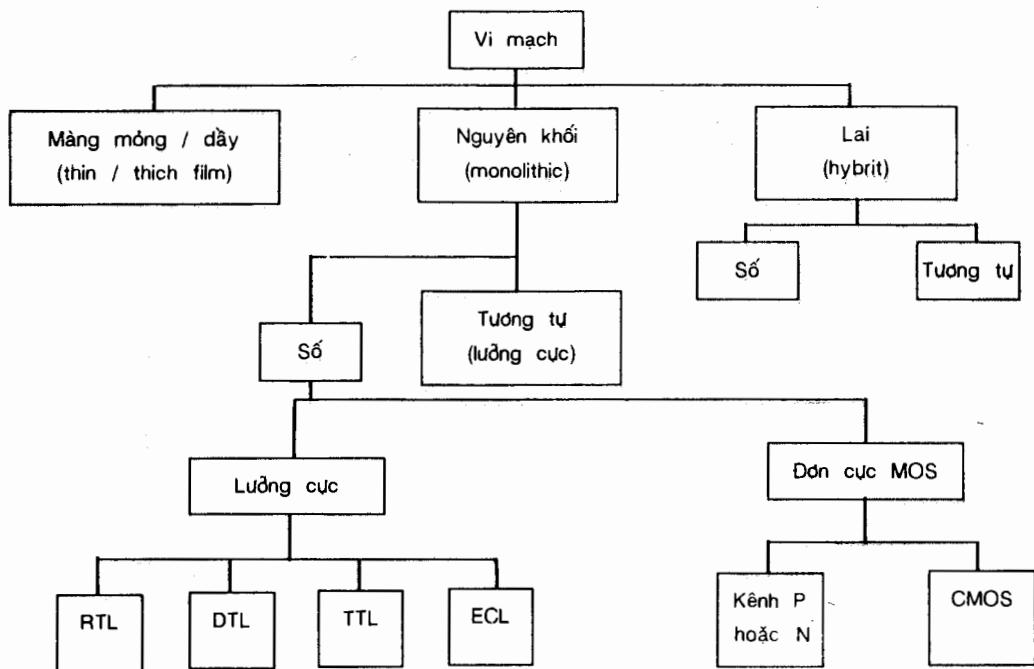
- a. Vi mạch cỡ nhỏ
- SSI (Small Scale Intergration)
- b. Vi mạch cỡ vừa
- MSI (Medium Scale Intergration)
- c. Vi mạch cỡ lớn
- LSI (Large Scale Intergration)
- d. Vi mạch cực lớn
- VLSI (Very Large Scale Intergration)

Bảng cho ở hình 3-2 giới thiệu một số loại vi mạch số theo các mức tích hợp khác nhau.

3_4. PHÂN LOẠI THEO CÔNG NGHỆ CHẾ TẠO

Hầu hết các loại vi mạch đều được chế tạo từ vật liệu bán dẫn. Sau phát minh vĩ đại về transistor (1974), công nghiệp bán dẫn bắt đầu phát triển nhanh chóng và đã làm đảo lộn nền công nghiệp điện tử. Đầu tiên vật liệu chủ yếu là germani, từ những năm 1960 trở lại đây thì vai trò chính lại chuyển sang silic. Các linh kiện quang - điện tử và siêu cao tần mới ra đời thường dùng vật liệu gallium arsenide có tính năng hứa hẹn. Chúng cho phép đạt tốc độ dịch chuyển nhanh hơn silic từ 2 đến 6 lần. Công nghệ mới nhất dựa trên hiệu ứng Josephson cho phép đạt tốc độ dịch chuyển và mật độ tích hợp rất cao nhưng còn đang ở giai đoạn thử nghiệm, khó khăn đặc biệt là phải giữ vật liệu siêu dẫn ở điều kiện nhiệt độ cực thấp.

Về mặt chế tạo, IC có thể chia ra làm 4 loại (hình 3-3): IC khối rắn (monolithic), IC màng mỏng/màng dày (thin/thick film) và IC lai (hybrid).



Hình 3-3. Phân loại vi mạch theo công nghệ chế tạo.

3-4.1. VI MẠCH BÁN DẪN KHỐI RÂN

Mạch tích hợp bán dẫn khối rân (monolithic) là mạch được xây dựng hoàn toàn trên một đơn vị tinh thể chất bán dẫn là Si. Sau đó chất bán dẫn sẽ được khuếch tán vào trong chất nền để tạo thành nhiều loại mặt ghép khác nhau. Những mặt ghép này có thể tạo thành diốt, transito hay điện trở.

Những vật liệu bán dẫn được khuếch tán vào trong chất nền dưới dạng hơi và đọng lại trên chất nền sau hàng loạt các quá trình "tạo mask" trong điều kiện nhiệt độ cao.

Quá trình tạo "mask" là quá trình trong đó người ta tiến hành oxy hoá bế mặt chất bán dẫn, tức là lấp kín bế mặt của nó bằng SiO_2 . Sau đó phủ một lớp cảm quang lên trên bế mặt SiO_2 . Dạng mạch được thu nhỏ, chụp lên phim tạo thành khuôn sáng. Đặt khuôn lên trên bế mặt chất bảo vệ, chiếu ánh sáng vào ta được các dạng yêu cầu. Dùng hoá chất ăn mòn các rãnh, bỏ chất cảm quang để thực hiện khuếch tán chất bán dẫn vào. "Mặt nạ" được tạo bằng bằng phương pháp quang học nên còn được gọi là phương pháp quang khắc.

Vi mạch monolithic có 2 loại : mạch lưỡng cực và mạch MOS (Metal – Oxide – Semi – conductor : kim loại – oxit – bán dẫn). Ngày nay , vi mạch MOS trở nên phổ biến hơn vì MOS dễ chế tạo, tốn ít diện tích hơn nên cho khả năng tích hợp lớn.

3-4.2. VI MẠCH TÍCH HỢP MÀNG MỎNG VÀ MÀNG DÂY

Các IC màng mỏng và màng dây được chế tạo bằng cách lồng đọng những vật liệu nhất định trên một để cách điện (chẳng hạn như gốm, sứ ...). Sau hàng loạt các quá trình tạo "mặt nạ", trên để được tạo thành điện trở, điện dung hay điện cảm. Các diốt , transito ... ít được chế tạo theo phương pháp này. Transito thường được chế tạo bằng phương pháp thông thường, với kích thước nhỏ (thường dùng FET). Do mạch này có thể chế tạo rất nhỏ nên nó cũng có những ưu điểm giống như mạch tích hợp nguyên khối (IC monolithic). Hơn nữa, khả năng chịu đựng của các phần tử của nó tốt hơn IC monolithic. Vì vậy, đối với những mạch đòi hỏi chính xác cao người ta thường sử dụng mạch tích hợp màng mỏng và màng dây.

3-4.3. VI MẠCH LAI (IC HYBRID)

Vi mạch lai là sự kết hợp của hai loại vi mạch trên. Mạch lai có thể bao gồm nhiều tinh thể monolithic được ghép nối với nhau thành một khối , nó cũng có thể là sự kết hợp giữa mạch monolithic với mạch màng mỏng thụ động. Dùi khi các IC monolithic , IC màng mỏng / màng dây được kết hợp với các phần tử bán dẫn để thực hiện các chức năng riêng.

Do tận dụng được ưu điểm của cả hai loại mạch nói trên, vi mạch lai có ưu điểm hơn hẳn các loại mạch khác. Chẳng hạn, do kích thước vô cùng nhỏ của mạch monolithic , công suất của mạch nhỏ. Để nhận được công suất lớn , trong mạch lai có thể kết hợp một mạch monolithic công suất nhỏ với transito công suất lớn. Hoặc dùng mạch lai có thể tạo một mạch chính xác cao bằng sự phối hợp giữa một mạch monolithic và mạch màng mỏng chính xác cao.

Do phải phối hợp nhiều công nghệ nên đồng thời với sự hoàn hảo, mạch tổ hợp lai đắt tiền hơn các loại mạch khác.

3_5. NHỮNG THÔNG SỐ CỦA VI MẠCH SỐ

Những thông số quan trọng của vi mạch số là :

- Mức logic (Logic Levels)
- Trễ truyền đạt (Propagation Delay)
- Công suất (Power)
- Độ ổn định nhiễu (Noise Immunity)
- Khả năng mắc tải vào, ra (Fan in , Fan out).

3_5.1. MỨC LOGIC

Mức logic là giá trị điện áp vào, ra được qui định cho các số nhị phân 0 và 1. Thường người ta chỉ ra giá trị danh định cho 2 mức logic.

Trong thực tế, giá trị điện áp có thể biến đổi chút ít do khả năng của các phần tử trong mạch, do sự biến đổi của nguồn cung cấp, do nhiệt độ và một số thông số khác. Thông thường, các nhà chế tạo sẽ đưa ra giá trị lớn nhất và nhỏ nhất của điện áp cho phép đối với mức logic 0 và logic 1.

Mức logic là thông số quan trọng của vi mạch số. Nhờ thông số này có thể dễ dàng nhận biết được những trạng thái logic ra và vào bằng cách đo mức logic bằng vôn kế hay oxylo. Hiểu biết về những mức logic sẽ cho phép phân tích sự hoạt động của mạch.

3_5.2. TRỄ TRUYỀN ĐẠT

Trễ truyền đạt là khoảng thời gian để đầu ra của mạch có đáp ứng đổi với sự thay đổi mức logic của đầu vào.

Trễ truyền đạt là tiêu chuẩn để đánh giá tốc độ làm việc của mạch. Tốc độ làm việc tương ứng với tần số lớn nhất mà mạch vẫn hoạt động đúng. Rõ ràng, trễ truyền đạt càng nhỏ càng tốt hay tốc độ làm việc càng lớn càng tốt.

Hình 3-4 minh họa thời gian trễ truyền đạt trong mạch số. Trên hình vẽ chỉ ra đầu vào, đầu ra tương ứng. Một chuyển biến từ $0 \Rightarrow 1$ ở đầu vào gây ra sự chuyển biến từ $1 \Rightarrow 0$ ở đầu ra và ngược lại. Nhận thấy, sự chuyển biến ở đầu ra xảy ra ở một thời điểm xác định sau khi có chuyển biến đầu vào. Đó là trễ truyền đạt.

Trễ truyền đạt thường được tính toán ở điểm 50% biên độ trên các sườn trước và sườn sau tương ứng giữa xung vào và xung ra.

Có 2 loại trễ truyền đạt :

- Trễ xảy ra khi đầu ra thay đổi từ mức cao (high) xuống mức thấp (low).
- Trễ xảy ra khi đầu ra thay đổi từ mức thấp lên mức cao.

Do cấu tạo của mạch logic, trễ đối với 2 loại chuyển biến thường là khác nhau. Chúng giống nhau về mức, gần nhau về giá trị nhưng không tương đương.

Độ rộng sườn trước (t_{s1}) và **độ rộng sườn sau** (t_{s2}) của xung vào và xung ra được định nghĩa :

Độ rộng sườn trước t_{s1} (độ rộng sườn sau t_{s2}) là khoảng thời gian để biên độ xung thay đổi trong khoảng từ 10% đến 90% giá trị biên độ cực đại của nó.

Đối với hầu hết các vi mạch số ngày nay, trễ truyền đạt là rất nhỏ. Trễ truyền đạt có thể nhỏ cỡ 1ns (nano giây). Một vài loại mạch logic có thời gian trễ lớn cỡ vài trăm nano giây ($n \times 100$ ns). Độ rộng sườn trước và độ rộng sườn sau thường nhỏ hơn thời gian trễ.

Do chế tạo, do đường dẫn mạch và vài thông số khác, trễ truyền đạt có thể biến thiên đáng kể so với giá trị rất nhỏ của nó đã được chỉ ra trong catalog. Khi đó trễ truyền đạt sẽ bị tăng lên.

Khi mắc liên tiếp nhiều mạch logic thì trễ truyền đạt của toàn mạch sẽ bằng tổng các trễ truyền đạt của từng tầng.

3.5.3. CÔNG SUẤT

Một thông số quan trọng khác của vi mạch là công suất. Đối với vi mạch phải quan tâm đến nhiều loại công suất.

Công suất tiêu tán

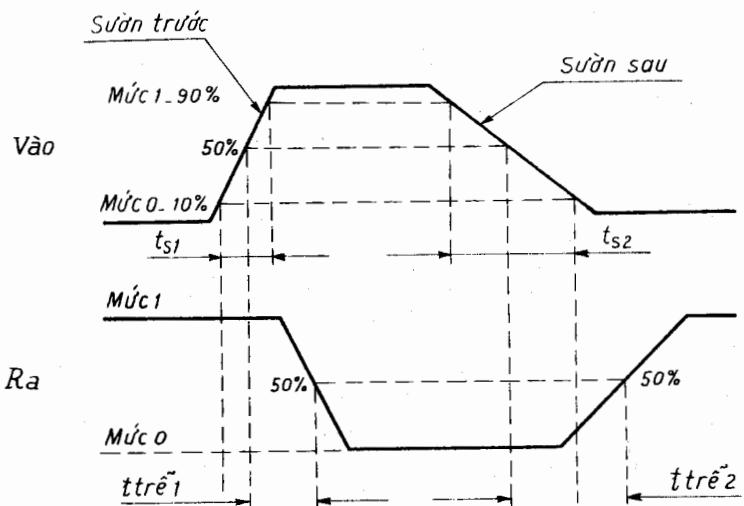
Đây là tiêu chuẩn để đánh giá lượng công suất tiêu thụ (tổn hao) trên các phần tử trong vi mạch. Công suất tiêu tán thường cỡ vài mW đối với một vi mạch số và là giá trị trung bình giữa công suất tiêu tán khi đầu ra ở mức 0, 1 (các công suất này thường là khác nhau). Công suất tiêu tán lớn tức là sự tiêu thụ năng lượng điện lớn. Tất nhiên mong muốn công suất tiêu tán càng nhỏ càng tốt.

Công suất tiêu tán có ý nghĩa đặc biệt quan trọng trong các thiết bị xách tay hay các thiết bị dùng pin. Để làm giảm sự tiêu thụ pin và đảm bảo sử dụng pin được lâu dài, người ta cố gắng làm giảm công suất tiêu tán.

Công suất điều khiển

Công suất điều khiển là công suất của tín hiệu điều khiển ở đầu vào bảo đảm sự hoạt động đúng của mạch.

Rõ ràng công suất điều khiển càng nhỏ càng tốt.



Hình 3-4. Trễ truyền đạt.

3_5.4. ĐỘ ỔN ĐỊNH NHIỀU

Độ ổn định nhiều là tiêu chuẩn đánh giá độ nhạy của mạch logic đối với tạp âm xung trên đầu vào của mạch.

Độ ổn định nhiều (tĩnh) là giá trị điện áp nhiều tối đa trên đầu vào không làm thay đổi trạng thái logic của mạch – còn được gọi là mức ổn định nhiều. Độ ổn định nhiều khi đầu ra ở mức 0, 1 là khác nhau. Do vậy, người ta phân biệt độ ổn định nhiều mức 0 và độ ổn định nhiều mức 1.

Tạp âm được xem như là bất cứ một tín hiệu nào không mong muốn phát sinh trong bản thân thiết bị hoặc được đưa từ ngoài vào, đè lên những mức logic chuẩn của hệ thống. Tạp âm này có thể biến đổi chậm theo mức một chiều, hoặc là ở tần số cao, thời gian tồn tại ngắn. Tạp âm có thể xảy ra ngẫu nhiên hay lặp đi lặp lại. Trong mọi trường hợp tín hiệu tạp âm có thể gây ra sự chuyển biến trạng thái sai nhầm.

Tất cả các mạch số logic đều có độ ổn định nhiều khá cao. Độ ổn định nhiều của hầu hết các mạch logic đều trong khoảng từ 10% – 50% của điện áp cung cấp.

Trong một số trường hợp, tạp âm bị loại bỏ bởi chính mạch logic với tư cách là "phản tử chuyển biến chậm" : một số tạp âm có tần số cao trong tự nhiên hay các tạp âm xung có thời gian tồn tại ngắn đến mức mạch logic không phản ứng kịp để gây chuyển biến trạng thái.

Độ ổn định tạp âm là một tham số quan trọng đối với các vi mạch số bởi lẽ hầu hết các mạch số trong quá trình chuyển biến trạng thái đều sinh ra một lượng tạp âm nhất định. Hơn nữa, nhiều thiết bị số được sử dụng trong môi trường công nghiệp có nhiều lớn.

3_5.5. KHẢ NĂNG MẮC TÀI VÀO, RA

Hệ số tải đầu ra

Hệ số tải đầu ra là số tải có thể nối được với đầu ra của mạch mà vẫn đảm bảo được sự hoạt động tin cậy, đảm bảo tốc độ, giới hạn nhiệt độ và các thông số khác.

Do hạn chế bởi công suất ra tối đa của mạch, do hình dạng của mạch, ... tất nhiên số lượng tải có thể nối với mạch logic là có hạn.

Có 2 cách cơ bản để nối tải vào đầu ra :

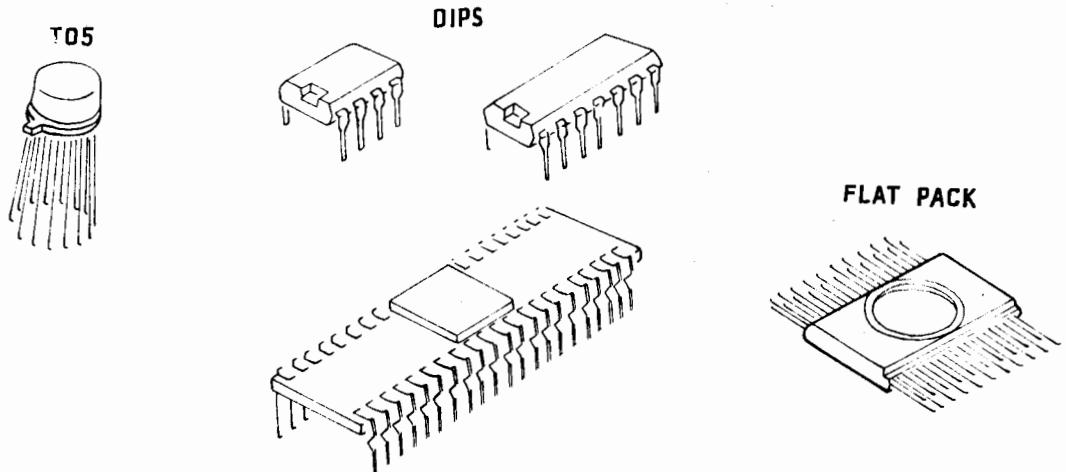
- **Dất chung** : Các tải được nối giữa đầu ra của mạch và đất.
- **Nguồn chung** : Các tải được nối giữa các đầu ra của mạch và nguồn cung cấp.

Hệ số tải đầu vào

Hệ số tải đầu vào là số đầu vào cực đại của mạch mà vẫn bảo đảm mạch làm việc tin cậy. Mong muốn hệ số tải đầu vào, ra lớn.

3_6. ĐÓNG VỎ IC

Có 3 phương pháp để đóng vỏ cho tinh thể silic là : phương pháp T05, đóng vỏ dạng hộp và đóng vỏ hai hàng chân song song. Chúng được minh họa ở hình 3-5.



Hình 3-5. Các loại đóng vỏ IC.

Dạng đóng vỏ IC đơn giản nhất là T05. Nó có hình dạng tiêu chuẩn giống như transito, tuy nhiên có một số cải tiến. Kiểu đóng rắn này hiện nay vẫn còn được sử dụng tuy không phổ biến.

Ưu điểm của nó là có thể tiêu tán được nhiệt tích luỹ. Vì vậy, kiểu này chủ yếu được sử dụng trong các IC tuyển tính.

Đóng vỏ dạng hộp

Đóng vỏ dạng hộp (Flat Pack) là kiểu đóng rắn IC với kích thước nhỏ nhất với mật độ cao. Do đóng rắn mật độ cao nên các IC thường được chế tạo bằng gỗm là vật liệu có thể chịu đựng được nhiệt độ cao. Mật khác vẫn phải giải quyết vấn đề tỏa nhiệt.

Flat - pack thường được sử dụng trong các hệ thống quân sự có độ tin cậy cao và thiết bị công nghiệp đặc biệt.

Đóng vỏ hai hàng chân song song

Kiểu đóng vỏ IC phổ biến nhất là kiểu hai hàng chân song song (Dual in Line Package). Nó hơi lớn hơn các kiểu đóng rắn khác về kích cỡ nhưng có rất nhiều ưu điểm : dễ lắp ráp và sử dụng. Các loại IC đóng vỏ kiểu DIP có kích thước khác nhau, thay đổi từ 8 chân đến 40 chân. Hầu hết các mạch SSI có 8,14 hay 16 chân. Các mạch MSI thường có 14, 16 hay 24 chân. Còn LSI, do kích thước lớn, đòi hỏi số lượng lớn đầu vào, đầu ra nên thường được thiết kế với 24, 28 và 40 chân.

Có nhiều kiểu vật liệu được sử dụng để đóng rắn. Thông dụng nhất và rẻ nhất là đóng gói chất dẻo. Đầu tiên các tinh thể IC được đặt vào khung kim loại và sau đó toàn bộ mạch được bao phủ bằng kỹ thuật đúc chất dẻo. Ngoài ra, người ta còn sử dụng kỹ thuật đóng rắn bằng gỗm. Nó có khả năng chịu đựng nhiệt độ cao hơn.

3_7. GIỚI HẠN NHIỆT ĐỘ

Hầu hết các IC đều được tính toán để có thể hoạt động trong một dải nhiệt độ kha rộng , thường là từ $-55^{\circ}\text{C} \div +125^{\circ}\text{C}$.

Những mạch đặc biệt có thể làm việc tốt ở ngoài giới hạn trên chỉ dùng trong các thiết bị có độ chính xác cao và thường là rất đắt.

Riêng loại IC đóng rắn bằng chất dẻo (plastic) có giới hạn nhiệt độ nhỏ hơn, thường từ $0^{\circ}\text{C} + 70^{\circ}\text{C}$.

Trong các mục 3-8 và 3-9 sẽ trình bày sơ lược đặc điểm của các vi mạch đơn cực và lưỡng cực. Việc đánh giá các vi mạch chủ yếu dựa vào các chỉ tiêu kỹ thuật đã nêu ở mục 3-3 như : công suất tiêu tán, thời gian trễ, độ ổn định nhiều, khả năng tải. Ngoài ra cũng phải xét đến một số thông số:

- Khả năng tích hợp (càng lớn càng tốt)
- Giá thành (càng rẻ càng tốt)
- Dễ chế tạo
- Dễ phối hợp (tương thích) với vi mạch công nghệ khác.

3-8. CÔNG NGHỆ ĐƠN CỰC (MOS - METAL OXID SEMI-CONDUCTER)

Đặc điểm

- Dễ chế tạo vì qui trình thực hiện đơn giản và ít công đoạn hơn công nghệ lưỡng cực do vậy giá thành rẻ.

- Mật độ tích hợp cao vì transito đơn cực nhỏ về kích thước và tiêu thụ rất ít điện năng.

- Công suất tiêu thụ nhỏ.

Tùy theo loại MOSFET được dùng mà công nghệ này còn được chia thành các loại sau :

3-8.1. HỌ PMOS

MOSFET dùng ở đây có kênh p nên gọi là PMOS (phản tử mang điện là các lỗ). Các lỗ vốn dịch chuyển khó hơn các điện tử n thành ra PMOS có tần số làm việc nhỏ (khoảng 1MHz) ; mật độ tích hợp lớn, công suất tiêu thụ nhỏ, dễ chế tạo. Tuy nhiên, họ này lại không tương hợp với TTL đòi hỏi nhiều điện áp nguồn nuôi khác nhau.

Các μP loại chậm , rẻ tiền thường được chế tạo bằng công nghệ này. Ví dụ : NEC μCOM 43/44/45 và TMS 1000.

3-8.2. HỌ MNOS

Kênh dẫn trong MOSFET là kênh điện tử n do đó đạt được tốc độ dịch chuyển cao hơn PMOS hàng chục lần. Mật độ tích hợp rất lớn. Ví dụ : RAM 256 Kbit thích hợp cho các VLSI, công suất tiêu thụ tương đương PMOS : trung bình 0,2 mW/cổng.

NMOS nói chung là tương hợp với TTL chỉ cần một nguồn nuôi duy nhất trong nhiều trường hợp. Các cải tiến của họ này như HMOS, XMOS, VMOS có mật độ tích hợp cao hơn, công suất tiêu thụ ít hơn, tần số làm việc lớn hơn.

Các LSI, VLSI và các μ P 16,32 bit thường được chế tạo bằng NMOS. Ví dụ : 8080 / 8085 / 8088 / 8086 , Z80 / Z8000 / Z80000 , MC 6800 / 68000 , NS 16000 / 32000. ..

3_8.3. HỌ CMOS

Ở đây sử dụng các cặp MOSFET kênh n và kênh p ở chế độ tải tích cực do đó công suất tiêu thụ rất nhỏ, trung bình $10 \mu\text{W}/\text{cổng}$. CMOS lại rất tin cậy vì ngưỡng đổi trạng thái logic bằng khoảng 1/2 điện áp nuôi. Tuy nhiên tần số làm việc và mức tích hợp có phần nào bị hạn chế so với NMOS. Các vi mạch CMOS hay được dùng trong những thiết bị điện tử y tế, quân sự, công nghiệp, Mặc dù còn đất song chúng có khả năng thay thế họ NMOS trong tương lai. Những loại SOSMOS (CMOS trên saphire) và HCMOS gần đây đã đạt được tốc độ xấp xỉ TTL trong khi mật độ tích hợp rất cao (như PMOS). Nhược điểm của chúng là khi làm việc ở 5 V thích hợp với TTL thì tốc độ bị giảm đi nhiều.

Ví dụ : μ P định trình 8 bit RCA-1802 , NSC 800 12 bit IMS 6100 và ROM 1Mbit 731000G.

Một số chỉ tiêu kỹ thuật của CMOS :

CMOS
- Thời gian trễ : 30-100ns
- Công suất tiêu tán : $0,01\text{mW}$ (1mW khi tần số 1MHz)
- Khả năng tải 50
- Độ ổn định nhiễu cao (45% U_{cc})
- Mức logic : Mức 0 = 0 V ; Mức 1 = Điện áp nguồn
- Các cổng logic cơ bản : NOR , NAND
- Nguồn cung cấp : $U_{cc} = 3-15$ V

3_9. CÔNG NGHỆ LƯƠNG CỰC

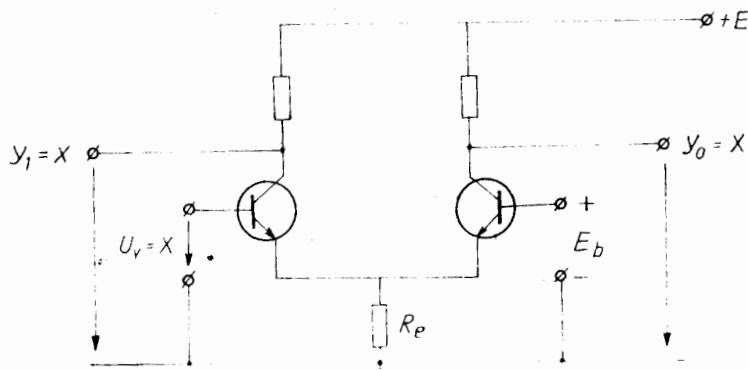
Thành phần cơ bản của các vi mạch công nghệ lưỡng cực là những transito lưỡng cực. Theo chế độ làm việc của các transit mà chia ra những họ sau đây :

3_9.1. HỌ ECL (MẠCH LOGIC GHÉP CỰC PHÁT - EMITOR COUPLER LOGIC)

Mạch cơ bản như hình 3-6.

Đặc điểm: Các transito đều làm việc ở chế độ không bão hòa. Do vậy tốc độ chuyển biến nhanh ($t_{c.mach} < 1$ ns), công suất tiêu thụ lớn (trung bình $25\text{mW}/1$ cổng), khó đạt được mật độ tích hợp cao. Chế tạo phức tạp. Điện áp nguồn $-5,2$ V , không tương hợp với các họ khác về mặt logic. Một số chỉ tiêu kỹ thuật của ECL :

ECL
- Thời gian trễ : 1-3 ns
- Công suất tiêu tán : 40 - 60mW
- Khả năng tải : 10 - 25
- Độ ổn định nhiễu : Cao
- Mức logic : Mức 0 = -1,75 V ; Mức 1 = -0,9 V
- Các cổng logic cơ bản : OR/NOR
- Nguồn cung cấp : $U_{cc} = -5,2$ V



Hình 3-8. Mạch logic ghép cực phát.

3_9.2. HỘ IIL (MẠCH LOGIC TÍCH HỢP PHUN – INTERGRATED INJECTION LOGIC)

Công nghệ này mới ra đời còn chưa được sử dụng nhiều nhưng có nhiều triển vọng vì :

- Mật độ tích hợp rất cao gần như PMOS
- Công suất tiêu tán nhỏ chỉ thua họ CMOS
- Dễ chế tạo, rẻ như công nghệ MOS
- Có thể ổn định và tin cậy như họ MOS
- Có thể chế tạo phần tương tự chung với phần số trong cùng một vi mạch. Ví dụ : μP định trình 16bit F9445, SBP9900.

3_9.3. HỘ TTL (TRANSISTOR – TRANSISTOR – LOGIC)

Hộ này được dùng rộng rãi trong mọi lĩnh vực và đã trở thành một tiêu chuẩn : gọi là sự tương hợp TTL để xét các hộ khác về mức logic

$$"0" : 0 \text{ V} \div 0,8 \text{ V}$$

$$"1" : 2,4 \text{ V} \div 5 \text{ V}$$

Đặc điểm : transito làm việc trong chế độ bão hòa nên thời gian trễ lớn khoảng vài nanô giây, công suất tiêu thụ nhỏ hơn công suất tiêu thụ của mạch ECL vài lần ; mật độ tích hợp cao. TTL chỉ dùng một nguồn điện áp nuôi duy nhất là 5 V.

Một số ký hiệu của TTL :

Theo nhiệt độ công tác có các loại sau :

74	:	0°C	÷ 70°C
84	:	- 25°C	÷ 85°C
54	:	- 55°C	÷ 125°C

Quá trình cài tiến TTL đi theo 2 hướng chính :

- Tăng tốc độ : Dùng diốt Schottky (tạo bởi ghép một chất bán dẫn và kim loại). Đặc điểm của diốt Schottky là thời gian trễ rất nhỏ. Nhờ diốt, transito không làm việc ở chế độ bão hòa do vậy tăng tốc độ (3 ns) nhưng lại tiêu tốn điện năng gấp đôi (20 mW/cổng). Mới nhất là TTL - AS (Avanced Schottky) nhanh gần bằng ECL (1,2 ns).

- Giảm công suất tiêu thụ ít hơn 10 lần (1 mW/cổng) song lại chậm hơn 3 lần (30 ns).

Kết hợp cả 2 hướng cài tiến này đã có :

TTL - LS (10 ns, 2 mW/cổng)

TTL - ALS (3 ns, 1,25 mW/cổng)

Ngoài ra, người ta cũng chế tạo các TTL công suất cao dùng cho các mạch đòi hỏi công suất lớn . Ví dụ : 74H00 .

Một số chỉ tiêu kỹ thuật của TTL :

TTL
- Thời gian trễ : 10 ns
- Công suất tiêu tán : 10 mW
- Khả năng tải : 10
- Độ ổn định nhiễu : Cao
- Mức logic : Mức 0 = +0,4 V ; Mức 1 = +3,6 V
- Các cổng logic cơ bản : NAND/NOR
- Nguồn cung cấp : $U_{cc} = 5 \text{ V} \pm 10\%$

TTL là loại vi mạch hiện nay đang được sử dụng rộng rãi, các mạch ra của TTL có nhiều loại khác nhau và được giới thiệu trong mục 3-10.

3-10. CÁC MẠCH RA TTL

Các mạch ra của vi mạch ngày càng được cải tiến để đáp ứng nhu cầu sử dụng ngày càng cao của thực tế . Phần này sẽ giới thiệu một số mạch ra của vi mạch loại TTL - là loại được sử dụng rộng rãi và nêu một số ứng dụng của các mạch ra này.

3-10.1. MẠCH RA BÌNH THƯỜNG (KÉO LÊN THỦ ĐỘNG - PASSIV-PULL-UP)

Mạch ra có sơ đồ như hình 3-7, trong đó C_L là tụ kỵ sinh gồm $C_{lắp ráp}$ và C tải.

Nguyên lý :

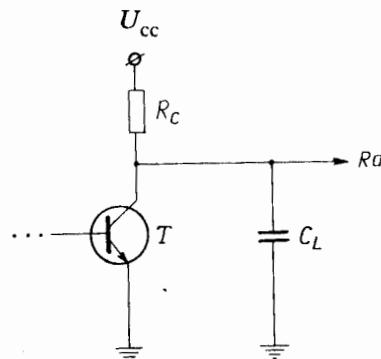
- Khi T tắt $U_{ra} = "H" \approx 5$ V.

Tụ C_L được nạp $T_n = R_c \cdot C_L$.

- Khi T thông bão hòa $U_{ra} = U_{ceb}$
 ≈ 0 V = "L".

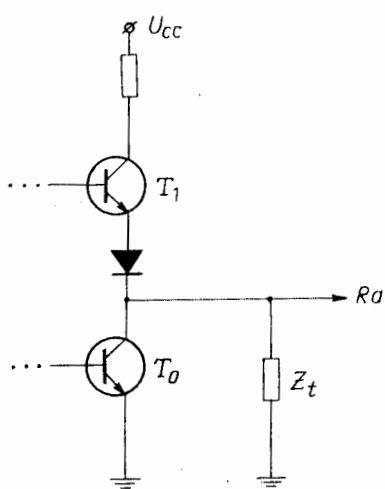
Tụ C_L phóng điện, $T_f = C_L / (R_c + R_{ceb})$.

Nhận thấy rằng : khi T tắt, T_n lớn
 \Rightarrow thời gian quá độ tăng \Rightarrow giảm tốc độ
 chuyển biến của mạch. Nhưng nếu giảm
 $R_c \Rightarrow T$ làm việc ở chế độ tích cực \Rightarrow công
 suất tiêu tán trên mạch tăng. Do vậy mạch
 ra này ít được dùng, phải cải tiến mạch ra.

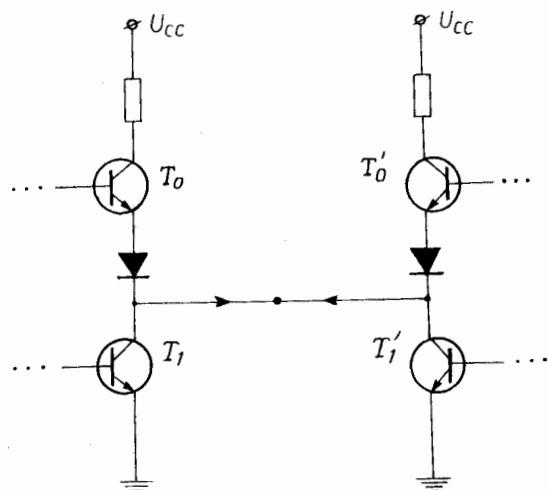


Hình 3-7. Mạch ra kéo lên thủ động.

3-10.2. MẠCH RA CỘT CHẠM HAY KÉO LÊN TÍCH CỤC (TOTEMPOLE HAY ACTIVE - PULL-UP)



Hình 3-8. Mạch ra kéo lên tích cực.



Hình 3-9. Nối chung hai mạch ra cột chạm.

Mạch ra có sơ đồ như hình 3-8.

Mạch có đặc điểm : $Z_{ra} = Z_T$ bão hòa, do vậy Z_{ra} luôn luôn nhỏ. cụ thể :

$$Z_{ra1} = Z_{T1\text{ b}\ddot{\text{h}}}; U_{ra} = "H".$$

$$Z_{ra1} = Z_{T0\text{ b}\ddot{\text{h}}}; U_{ra} = "H"$$

Ưu điểm : Tốc độ chuyển mạch cao.

Nhược điểm : Có khả năng hỏng mạch khi nối chung các đầu ra.

Ví dụ : khi nối chung đầu ra của hai mạch như hình 3-9, mạch sẽ hỏng khi T_1 và T_o cùng thông hoặc T_1' và T_o cùng thông (một đầu ra ở mức cao, một đầu ra ở mức thấp). Xác xuất xảy ra hỏng mạch tăng nhanh khi nối chung nhiều mạch ra.

3-10.3. MẠCH RA HỎ COLECTO (OPEN COLLECTOR)

Sơ đồ nguyên lý của tăng ra hở colecto được cho ở hình 3-10.

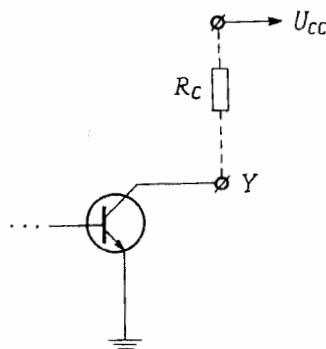
Mạch có cực thu để hở có thể được sử dụng như mạch logic thông thường bằng cách mắc thêm R_c lên nguồn. Giá trị của R_c phụ thuộc vào tải mắc ở đầu ra. Tải có thể là các mạch TTL khác hay đèn LED ... Thông thường giá trị $R_c =$ vài trăm $\Omega \div$ vài $k\Omega$.

Ưu điểm : - Có thể chọn R_c thích hợp với tình trạng của tải.

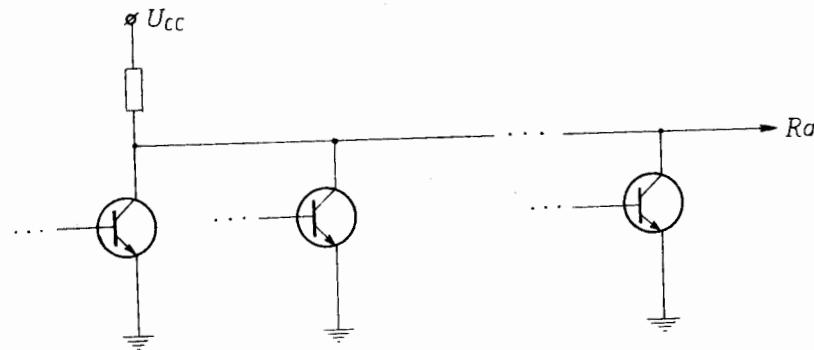
- Có thể nối chung các đầu ra và dùng chung một điện trở R_c . Lúc này, điểm nối chung thực hiện chức năng AND với mức "H" các đầu ra (gọi là wire AND hay dot AND).

Thật vậy, khi nối chung nhiều đầu ra (hình 3-11) mạch hoạt động như sau :

- RA = "H" khi tất cả các cổng đều có mức ra là "H".
- RA = "L" khi ít nhất 1 trong các cổng có mức ra là "L".



Hình 3-10. Mạch ra hở colecto.



Hình 3-11. Nối chung nhiều đầu ra mạch ra hở colecto.

3-10.4. MẠCH RA 3 TRẠNG THÁI

Các mạch ra trên đều có nhược điểm : luôn luôn chịu nhiều ảnh hưởng của tải. Các mạch ra kéo lên tích cực hay hở colecto có trở kháng ra khi T tắt lớn, do vậy thời gian quá độ của mạch lớn. Mạch ra cột chạm tuy có trở kháng ra tại các trạng thái 0, 1 luôn luôn nhỏ nhưng lại chịu nhiều ảnh hưởng của tải, mặt khác không cho phép nối chung các đầu ra.

Năm 1969 tại trường đại học Toyota, người ta đã phát minh ra mạch ra 3 trạng thái. Ba trạng thái đó là :

- Trạng thái 0

- Trạng thái 1

- Trạng thái treo : Trở kháng cao.

Sơ đồ tần số ra cuối cùng hoàn toàn giống như tần số ra cột chạm (hình 3-12,a), trong đó 2 transistor T_1 và T_0 tương đương 2 khóa điện tử (hình 3-12,b). Chỉ khác là nhờ tín hiệu điều khiển có thể tạo thêm trạng thái treo. Cụ thể :

Trạng thái 0 : T_0 thông bão hoà , T_1 tắt

$$U_{ra} = "L" \approx 0 \text{ V}$$

$$Z_{ra} = Z_{Tobh}$$

Trạng thái 1 : T_1 thông bão hoà , T_0 tắt.

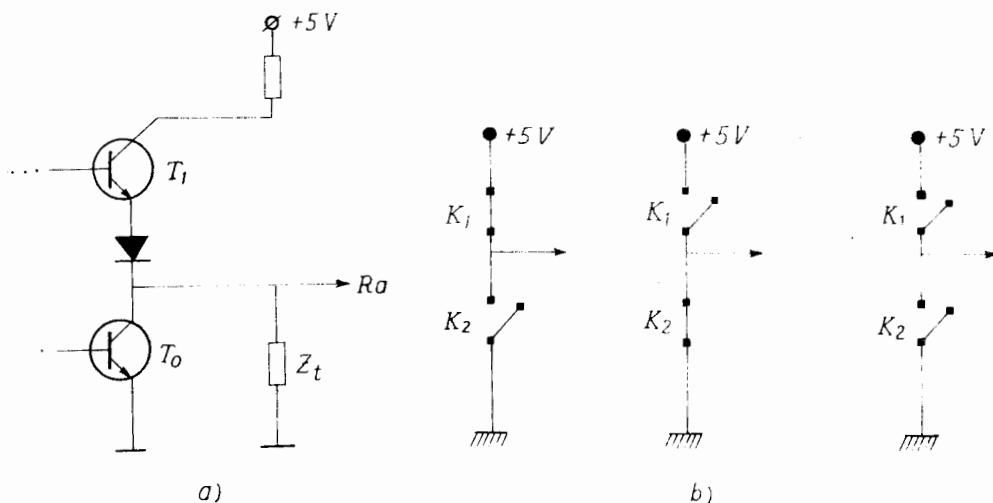
$$U_{ra} = "H" \approx 5 \text{ V}$$

$$Z_{ra} = Z_{Tibh}$$

Trạng thái treo : T_0 , T_1 đều tắt

$$Z_{ra} = \infty$$

Dầu vào, dầu ra không ảnh hưởng lẫn nhau.



Hình 3-12. Mạch ra 3 trạng thái.

a. Sơ đồ nguyên lý tần số ra cuối cùng; b. Sơ đồ tương đương của mạch ra 3 trạng thái.

Ví dụ : Mạch NAND đầu ra 3 trạng thái có sơ đồ nguyên lý như hình 3-13.

Nguyên lý : Xét tác dụng của X

- Khi $X = "L"$ T tắt mạch là mạch NAND vì :

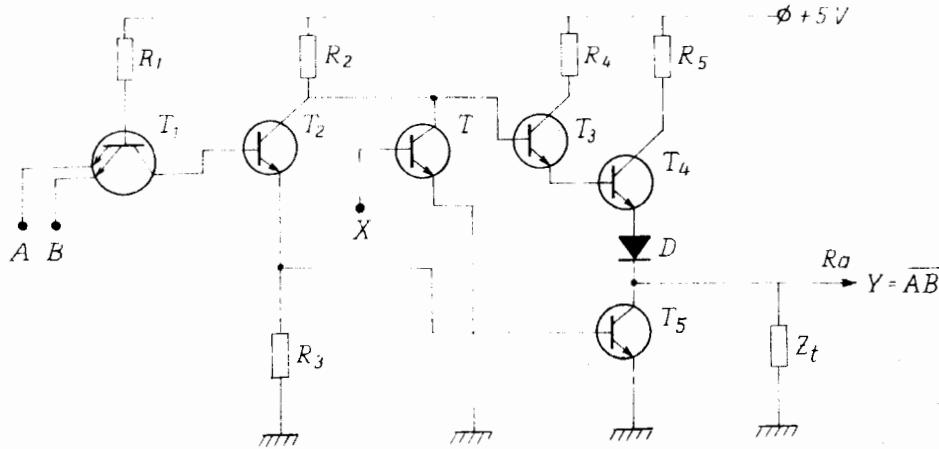
$$A, B = 1, 1 ; \text{ ra } Y = 0.$$

$$A, B = 0, 1 \text{ hoặc } 1, 0 \text{ hoặc } 0, 0 ; \text{ ra } Y = 1.$$

- Khi $X = "H"$ T thông bão hoà $\Rightarrow T_3$ tắt $\Rightarrow T_4$ tắt
 $\Rightarrow T_2$ tắt $\Rightarrow T_5$ tắt

$$A, B = \text{bất cứ giá trị nào} \Rightarrow \text{ra } Y = 1.$$

Vậy X là tín hiệu điều khiển trạng thái treo của mạch. Ở đây khi X ở mức "H" mạch có trở kháng cao, khi X ở mức "L" mạch hoạt động bình thường.



Hình 3-13. Sơ đồ mạch NAND ba trạng thái.

Ký hiệu của mạch 3 trạng thái (hình 3-14) :

Tác dụng của mạch 3 trạng thái :

1. Khi ở trạng thái treo đầu vào không ảnh hưởng đến đầu ra, do vậy có thể sử dụng mạch 3 trạng thái để tạo nên những đường thông tin 2 chiều.

Ví dụ : Sơ đồ sử dụng mạch 3 trạng thái tạo nên đường thông tin 2 chiều cho ở hình 3-15.

Nguyên lý làm việc :

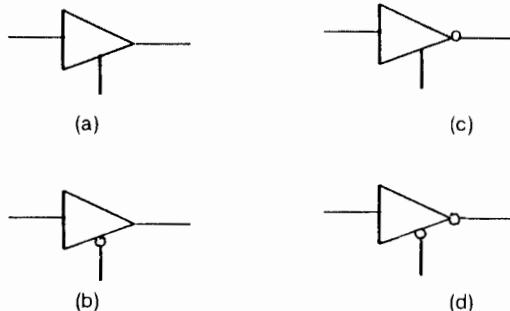
- Khi $X_{dk} = 1$: mạch 1 làm việc bình thường, thông tin đi từ A

$\Rightarrow B$ ($B = A$), mạch 2 ở trạng thái treo.

- Khi $X_{dk} = 0$: mạch 2 làm việc bình thường, thông tin đi từ B $\Rightarrow A$ ($A = B$), mạch 1 ở trạng thái treo.

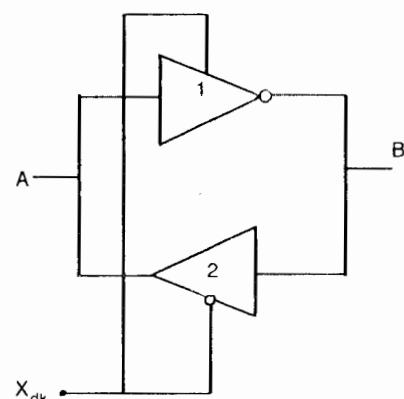
2. Nối chung các đầu ra 3 trạng thái. Mạch có đầu ra ba trạng thái có thể nối chung các đầu ra với nhau. Chỉ lưu ý phải tạo tín hiệu điều khiển sao cho tại một thời điểm bất kỳ chỉ có duy nhất một mạch hoạt động. Còn những mạch khác phải ở trạng thái trở kháng cao. Nếu có 2 mạch trở lên cùng làm việc thì có thể xảy ra hỏng mạch như khi nối chung các đầu ra của mạch ra cột chạm.

Vi mạch có đầu ra 3 trạng thái cho phép việc ghép chúng trên BUS chung trở nên dễ dàng.



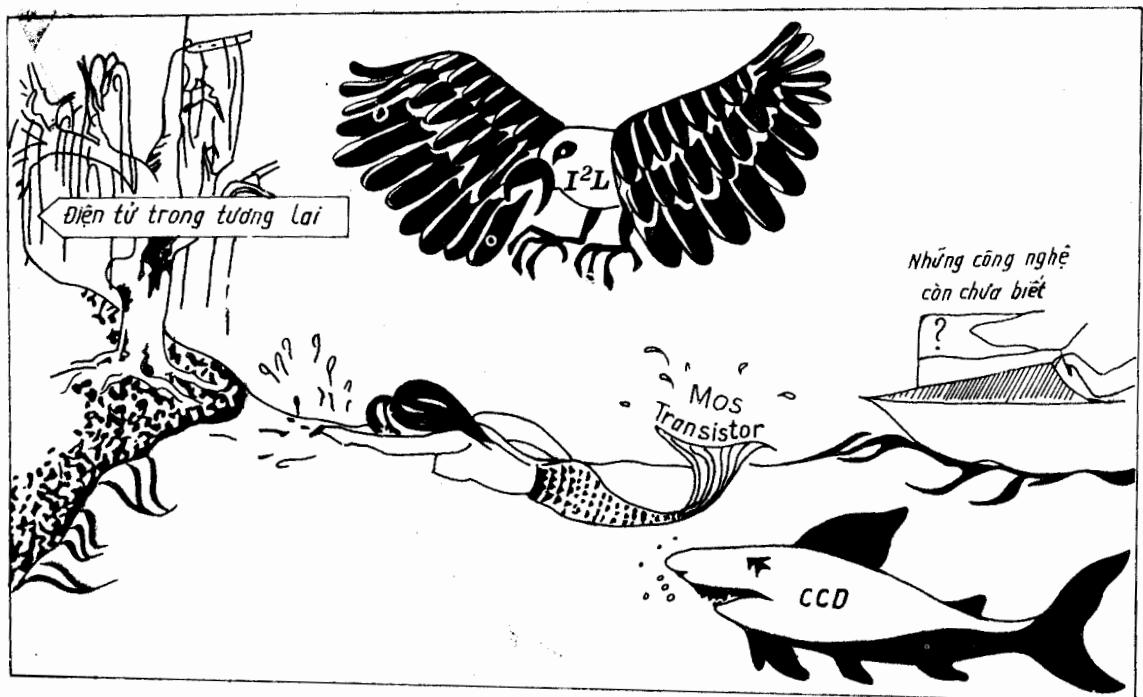
Hình 3-14. Ký hiệu của mạch ra 3 trạng thái.

- a. Bộ đệm không đảo với đầu vào điều khiển tích cực H
- b. Bộ đệm không đảo với đầu vào điều khiển tích cực L
- c. Bộ đệm đảo với đầu vào điều khiển tích cực H
- d. Bộ đệm đảo với đầu vào điều khiển tích cực L



Hình 3-15. Tao thông tin hai chiều từ mạch ra 3 trạng thái.

Kết luận : Kỹ thuật vi mạch phát triển nhanh chóng và đạt được những thành tựu không thể lường trước được. Con người vẫn đang tìm kiếm những công nghệ mới cho phép đạt được kết quả tốt hơn nữa. Cuộc cạnh tranh của công nghệ điện tử (hình 3.17) đã và sẽ không bao giờ kết thúc.



Hình 3-17. Cuộc thi đua của công nghệ điện tử.

TỔNG QUAN VỀ MẠCH SỐ

GIỚI THIỆU VỀ MẠCH SỐ

Các mạch số được chia làm 2 loại :

- Mạch tổ hợp (Combinational Circuits);
- Mạch dây (Sequential Circuits).

A. MẠCH TỔ HỢP

Là mạch mà tín hiệu ra chỉ phụ thuộc vào tín hiệu vào. Phương trình xác định tín hiệu ra của mạch là :

$$Y_i = f_i(X_1, X_2, \dots, X_n) \text{ với } \forall i = 1 \div m.$$

Y_i là tín hiệu ra ở đầu ra thứ i , các X_j ($j = 1 \div n$) là các tín hiệu vào, n là số các tín hiệu vào, m là số tín hiệu ra. Có thể gọi mạch tổ hợp là mạch không có trí nhớ.

B. MẠCH DÂY

Là mạch có tín hiệu ra phụ thuộc vào trạng thái trong của mạch và có thể phụ thuộc hoặc không phụ thuộc vào tín hiệu vào. Phương trình đặc trưng của mạch dây là :

$$Y_i = f_i(X_1, X_2, \dots, X_n, S_1, S_2, \dots, S_k) \text{ với } \forall i = 1 \div m.$$

Y_i là tín hiệu ra ở đầu ra thứ i , X_j ($j = 1 \div n$) là tín hiệu vào, S_t ($t = 1 \div k$) là trạng thái trong của mạch. Mạch dây có khả năng nhớ (lưu trữ) dữ kiện. Vì vậy mạch dây còn được gọi là mạch có trí nhớ.

Có thể coi mạch tổ hợp là một trường hợp riêng của mạch dây với số trạng thái trong của mạch là 1.

BÀI TOÁN THIẾT KẾ VÀ PHÂN TÍCH MẠCH SỐ

Bài toán phân tích là bài toán từ sơ đồ mạch cho trước xác định chức năng, dạng sóng, tính năng kỹ thuật ... của mạch đó.

Bài toán thiết kế là bài toán ngược của bài toán phân tích nghĩa là từ chức năng, dạng sóng, tính năng kỹ thuật ... đặt ra ban đầu, xây dựng được mạch thực hiện chức năng đó.

Quá trình thiết kế, phân tích, các hệ thống số gồm nhiều bước. Ví dụ bài toán thiết kế gồm các bước :

1. THIẾT KẾ LOGIC

Quá trình này chỉ liên quan đến khía cạnh logic của bài toán. Trên cơ sở này các vi mạch số đã có chỉ ra sơ đồ logic thực hiện mạch.

2. THIẾT KẾ MẠCH DIỆN TỬ

Đây là vấn đề liên quan đến việc thực hiện mạch điện tử cụ thể : chỉ ra sơ đồ điện tử thực hiện mạch.

3. THIẾT KẾ LAYOUT

Liên quan đến việc chế tạo mạch với công nghệ đã chọn, ví dụ : tiếp xúc, chiết dầy lớp oxít ...

Trong phạm vi của cuốn sách này chỉ để cập đến bài toán phân tích và thiết kế logic mạch số. Cần chú ý rằng sau khi đã có biểu thức logic của bài toán, trên cơ sở những vi mạch có sẵn phải xây dựng sơ đồ logic. Việc đó thực chất là thực hiện ánh xạ từ các giá trị logic 0 và 1 vào các giá trị điện thế L (Low – mức thấp) hay H (High – mức điện thế cao) của mạch vi điện tử. Phép ánh xạ này là tùy ý .

Nếu $\text{Logic } 0 \Rightarrow L$

$\text{Logic } 1 \Rightarrow H$

gọi là *logic dương*

Ngược lại nếu phép ánh xạ :

$\text{Logic } 0 \Rightarrow H$

$\text{Logic } 1 \Rightarrow L$

gọi là *logic âm*

Thông thường người ta dùng logic dương, trong các catalog cũng thường cho với các logic dương.

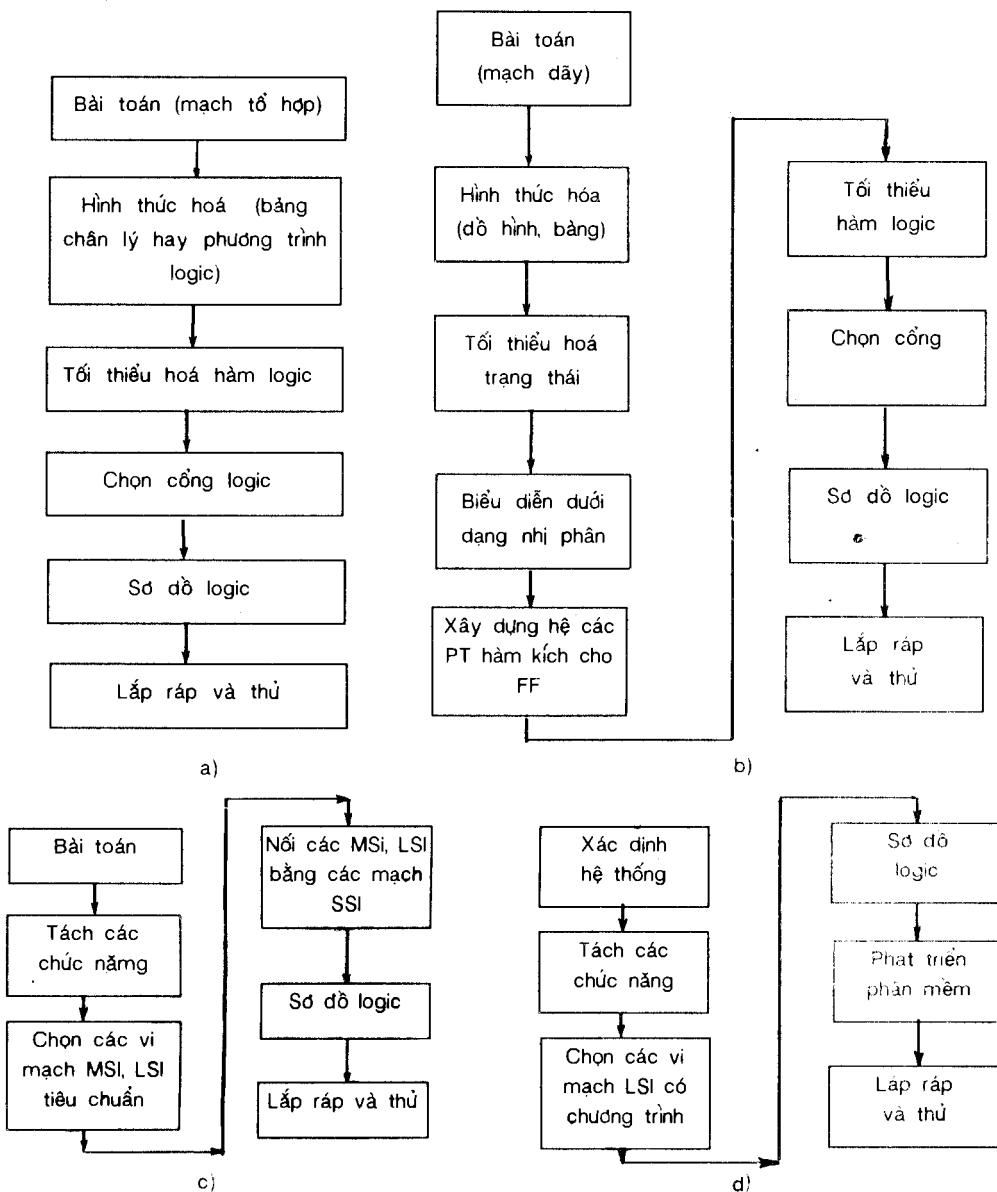
Cần lưu ý rằng : Giữa mạch dùng logic dương và mạch dùng logic âm có sự tương đương như bảng sau. Điều này rất có lợi khi thiết kế mạch cụ thể.

Logic dương	Logic âm
AND	OR
OR	AND
NAND	NOR
NOR	NAND
...	...

THIẾT KẾ LOGIC VÀ CÁC VI MẠCH CÓ LỚN

Mục tiêu chủ yếu đặt ra khi thiết kế là mạch phải thực hiện đúng những chức năng yêu cầu, đảm bảo độ tin cậy, dễ bảo dưỡng và đạt hiệu quả kinh tế cao. Muốn đạt được điều đó thì mạch được thiết kế cần phải đơn giản.

Các bước thiết kế mạch số được mô tả như hình sau :



Các bước thiết kế Logic mạch số :

- Các bước thiết kế mạch tổ hợp dùng SSI
- Các bước thiết kế mạch dây dùng các Flip-Flop và SSI
- Các bước thiết kế mạch số dùng MSI, LSI
- Các bước thiết kế mạch số dùng các vi mạch LSI có khả năng lập trình.

Trước đây, với những linh kiện rời rạc cổ điển như role, đèn điện tử, đèn bán dẫn và những vi mạch cỡ nhỏ SSI, thì một trong những nhiệm vụ quan trọng của thiết kế logic là tối thiểu hoá để đạt được số linh kiện là ít nhất.

Trong những năm gần đây, với sự phát triển mạnh mẽ của kỹ thuật vi điện tử, sự ra đời các mạch cỡ trung và cỡ lớn MSI, LSI, VLSI với giá thành ngày càng hạ đã làm cho quá trình thiết kế các hệ thống số có những thay đổi đáng kể. Với các vi mạch MSI, LSI, VLSI và những LSI lập trình được bài toán thiết kế là chọn các mạch đó, nối các mạch đó bằng các mạch phụ thêm như SSI và phát triển phần mềm... chứ không phải là tối thiểu hoá.

Các bước thiết kế đối với mỗi loại mạch số nói trên và các đặc điểm của mỗi loại mạch số sẽ được giới thiệu chi tiết ở các chương sau.

Phần 2

MẠCH TỐ HỢP

CHƯƠNG 4

THIẾT KẾ VÀ PHÂN TÍCH MẠCH TỔ HỢP

4-1. MÔ HÌNH TOÁN HỌC

Mô hình toán học tổng quát của mạch tổ hợp được mô tả như hình 4-1.

Trong mạch có n đầu vào, m đầu ra. Các X_i ($i = 1 \div n$) là các tín hiệu vào, các Y_j ($j = 1 \div m$) là các tín hiệu ra.

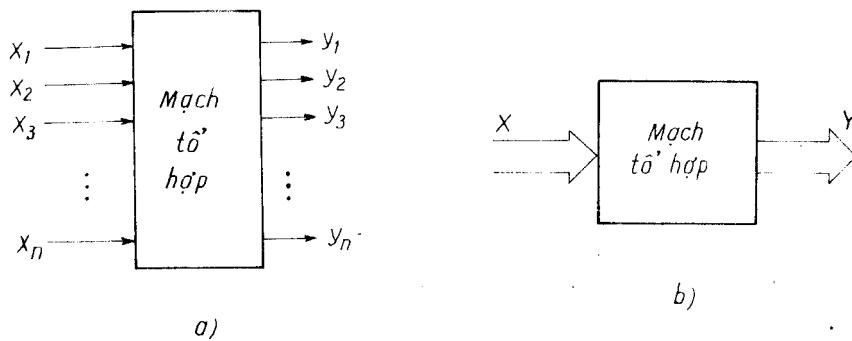
$X = \{X_1, X_2, \dots, X_n\}$: Tập các tín hiệu vào.

$Y = \{Y_1, Y_2, \dots, Y_m\}$: Tập hợp các tín hiệu ra.

Lúc đó mô hình toán học hình 4-1,a được mô tả một cách tổng quát như hình 4-1,b.

Mạch tổ hợp được mô tả bởi hệ m phương trình đại số Boole như sau :

$$Y_j = f_j(X_1, X_2, \dots, X_n) \text{ với } \forall j = 1 \div m$$



Hình 4-1. Mô hình toán học của mạch tổ hợp.

- Mô hình toán học mô tả cụ thể số tín hiệu vào và số tín hiệu ra;
- Mô hình toán học tổng quát của mạch tổ hợp.

Về mặt toán học có thể nói Otomat không có trí nhớ chính là mô hình toán học của mạch tổ hợp . Otomat không có trí nhớ được mô tả bởi phương trình sau :

$$O = (X, Y, f),$$

ở đây : X là bộ chữ vào; Y là bộ chữ ra; f là ánh xạ từ X vào Y .

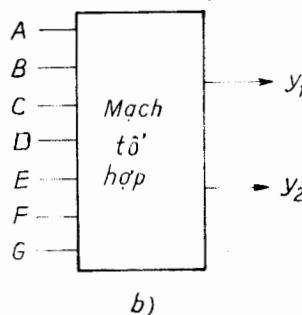
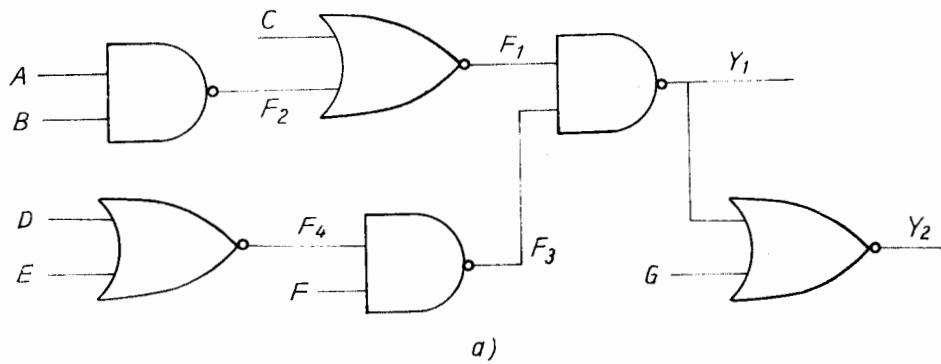
4-2. PHÂN TÍCH MẠCH TỔ HỢP

Bài toán phân tích là bài toán từ sơ đồ logic cho trước viết hàm logic của các đầu ra theo các đầu vào, và nếu cần thì còn phải chỉ ra dạng sóng của tín hiệu ra tương ứng với tín hiệu vào, xác định giá trị tín hiệu ở từng điểm trong sơ đồ. Thực hiện theo những bước sau :

- Đặt các biến phụ vào mỗi mạch đầu ra của mỗi mạch logic.
- Viết phương trình của các biến phụ đó. (viết lần lượt từ đầu vào cho đến đầu ra).
- Ở biểu thức cuối cùng, thay thế các giá trị tương ứng để rút ra được hàm logic cho các đầu ra cho sơ đồ đã cho.

Ví dụ : Phân tích mạch tổ hợp cho ở hình 4-2,a.

Từ sơ đồ logic cho ở hình 4-2,a lập được sơ đồ khối với các tín hiệu vào, tín hiệu ra như ở hình 4-2,b.



Hình 4-2. Phân tích mạch tổ hợp .

a. Sơ đồ logic cho trước ; b. Sơ đồ khối mạch tổ hợp cho ở hình 4-2,a.

Nhiệm vụ đặt ra là phải xác định được các tín hiệu ra phụ thuộc vào các tín hiệu vào như thế nào, tức là phải xác định được các hàm sau :

$$Y_1 = \varphi_1 (A, B, C, D, E, F, G)$$

$$Y_2 = \varphi_2 (A, B, C, D, E, F, G)$$

Bước 1: Đặt các biến phụ F_1, F_2, F_3, F_4 như trong hình 4-2,a. Ta có :

$$F_2 = \overline{A \cdot B}, F_1 = \overline{F_2 + C}, F_4 = \overline{D + E}, F_3 = \overline{F_4 \cdot F}.$$

Suy ra :

$$Y_1 = \overline{F_1 \cdot F_3}, Y_2 = \overline{Y_1 + G}.$$

Bước 2 : Viết phương trình các biến phụ đó :

$$\begin{aligned} Y_1 &= \overline{F_1 \cdot F_3} = \overline{F_1} + \overline{F_3} = F_2 + C + F_4 \cdot F \\ Y_2 &= \overline{Y_1 + G} = \overline{Y_1 \cdot G} = \overline{F_2 + C + F_4 \cdot F \cdot G} \end{aligned}$$

Bước 3 : Thay thế các giá trị F_2, F_4 vào các biểu thức Y_1, Y_2 :

$$\begin{aligned} Y_1 &= F_2 + C + F_4 \cdot F = \overline{A \cdot B} + C + \overline{D + E} \cdot F \\ &= \overline{A} + \overline{B} + C + \overline{D} \cdot \overline{E} \cdot \overline{F} \\ Y_2 &= \overline{F_2 + C + F_4 \cdot F \cdot G} = \overline{A \cdot B} + C + \overline{D + E} \cdot F \cdot \overline{G} \\ &= A + \overline{B} + C + \overline{D} \cdot \overline{E} \cdot F \cdot \overline{G} = A \cdot B \cdot \overline{C} (D + E + \overline{F}) \cdot \overline{G} \\ &= A \cdot B \cdot \overline{C} \cdot \overline{G} (D + E + \overline{F}). \end{aligned}$$

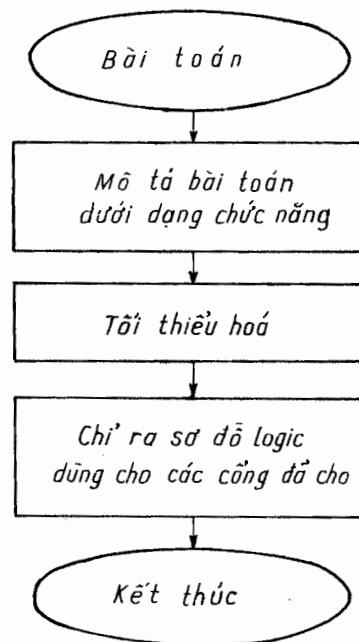
4-3. THIẾT KẾ MẠCH TỔ HỢP

4-3.1. BÀI TOÁN THIẾT KẾ VÀ CÁC BƯỚC THỰC HIỆN

Bài toán thiết kế là bài toán từ yêu cầu (chức năng, dạng sóng, tính năng kỹ thuật ...) xây dựng sơ đồ mạch thực hiện những phần tử cho trước.

Các bước thiết kế : Đối với trường hợp thiết kế dùng các phần tử rời rạc và các vi mạch cỡ nhỏ (SSI) thực hiện tuân tự theo các bước như biểu diễn ở lưu đồ trên hình 4-3.

Như đã trình bày, với sự phát triển nhanh chóng của kỹ thuật vi mạch việc thiết kế dùng các vi mạch cỡ trung bình và cỡ lớn MSI, LSI, VLSI, LSI lập trình được có nhiều thay đổi so với phương pháp thiết kế dùng các vi mạch cỡ nhỏ (SSI) và các



Hình 4-3. Các bước thiết kế mạch tổ hợp dùng các công tắc và các vi mạch cỡ nhỏ (SSI).

cổng. Những vấn đề này sẽ được đề cập đến trong một chương riêng (chương 12) ở cuối cuốn sách này.

Một vấn đề cần chú ý khi thiết kế một mạch tổ hợp có nhiều đầu ra là nên sử dụng những phần chung cho các hàm đó để sơ đồ thiết kế là đơn giản nhất.

4-3.2. THIẾT KẾ MẠCH TỔ HỢP 2 TẦNG VÀ NHIỀU TẦNG

a. Mạch 2 tầng

Ưu điểm

- Có thể thực hiện được mọi hàm logic;
- Có tốc độ cao;
- Việc phân tích và thiết kế mạch đơn giản.

Nhược điểm :

- Trong một số trường hợp thiết kế không nhận được sơ đồ đơn giản nhất;
- Thường yêu cầu các phần tử có số đầu vào lớn (trong trường hợp số đầu vào của phần tử cho trước không thỏa mãn chúng ta phải tăng số tầng của mạch lên).

Tầng 2 Tầng 1	AND	OR	NAND	NOR
AND	X	CTT	X	1. CTH 2. $\overline{\overline{f}}$, áp dụng D. (Demorgan)
OR	CTH	X	1. CTT 2. $\overline{\overline{f}}$, áp dụng D.	X
NAND	1. CTH 2. Các TP áp dụng D.	X	1. CTT 2. $\overline{\overline{f}}$, áp dụng D. (các thành phần không thay đổi)	X
NOR	X	1. CTT 2. Các TP áp dụng D.	X	1. CTH 2. $\overline{\overline{f}}$, áp dụng D. (các thành phần không thay đổi)

Hình 4-4. Cách thiết kế mạch hai tầng với các phần tử cho trước.

Giả thiết rằng :

- Các giá trị tín hiệu vào X_i, \bar{X}_i có sẵn.
- Trên cùng 1 tầng chỉ sử dụng một loại phần tử (AND, OR, NAND, NOR)

- Những phần tử này có số đầu vào không hạn chế.

Cách thiết kế mạch 2 tầng được cho ở hình 4-4.

Ví dụ : Cho hàm logic 3 biến $F(A, B, C)$ như hình 4-5.

Tối thiểu hóa hàm này, viết được phương trình hàm F như sau :

	$A\bar{B}$	$\bar{A}B$	$\bar{A}\bar{B}$	AB
C	00	01	11	10
0	1	1		
1		1	1	1

Hình 4-5. Biểu diễn hàm F trên bảng Karnaugh.

$$F = \bar{A} \bar{C} + B C + A C \quad : \quad \text{dạng chuẩn tắc tuyến (CTT)}$$

$$F = (\bar{A} + C)(A + B + \bar{C}) \quad : \quad \text{dạng chuẩn tắc hội (CTH).}$$

Nếu yêu cầu xây dựng sơ đồ mạch với tầng 1 và tầng 2 cho trước như sau :

1) Tầng 1 dùng AND, tầng 2 dùng OR :

Từ thuật toán đã nêu ở bảng hình 4-4 ta có :

Viết phương trình hàm F dưới dạng CTT :

$$F = \bar{A} \bar{C} + B C + A C$$

Từ phương trình của hàm F dễ dàng đưa ra được sơ đồ của mạch thực hiện hàm đó.

2) Tầng 1 dùng mạch OR, tầng 2 dùng AND :

Viết phương trình hàm F dưới dạng CTH :

$$F = (\bar{A} + C)(A + B + \bar{C})$$

3) Tầng 1 dùng NAND, tầng 2 dùng AND

Viết phương trình hàm F dưới dạng CTH :

$$F = (\bar{A} + C)(A + B + \bar{C})$$

Phủ định 2 lần từng thành phần rồi áp dụng Demoorgan :

$$F = \overline{\overline{(\bar{A} + C)}} \overline{\overline{(A + B + \bar{C})}} = \overline{(A \bar{C})} \cdot \overline{(A \bar{B} C)}$$

4) Tầng 1 dùng NOR, tầng 2 dùng OR

Viết hàm F dưới dạng CTT :

$$F = \bar{A} \bar{C} + B C + A C$$

Phủ định 2 lần từng thành phần rồi áp dụng Demoorgan :

$$F = \overline{\overline{\bar{A} \bar{C}}} + \overline{\overline{B C}} + \overline{\overline{A C}} = (\overline{\overline{A} + \overline{C}}) + (\overline{\overline{B} + \overline{C}}) + (\overline{\overline{A} + \overline{C}})$$

5) Tầng 1 dùng OR, tầng 2 dùng NAND

Viết hàm F dưới dạng CTT :

$$F = \bar{A} \bar{C} + B C + A C$$

Phủ định 2 lần hàm số rồi áp dụng Demoorgan :

$$F = \overline{\overline{\bar{A} \bar{C}}} + \overline{\overline{B C}} + \overline{\overline{A C}} = \overline{(A + C)} \cdot \overline{(B + C)} \cdot \overline{(A + C)}$$

6) Tầng 1 dùng NAND, tầng 2 dùng NAND

Viết hàm F dưới dạng CTT :

$$F = \overline{\overline{A} \overline{C}} + B C + A C .$$

Phủ định 2 lần hàm số rồi áp dụng Demoorgan (các thành phần không biến đổi)

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

7) Tầng 1 dùng AND, tầng 2 dùng NOR

Viết hàm F dưới dạng CTH :

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

Phủ định 2 lần hàm số rồi áp dụng Demoorgan

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

8) Tầng 1 dùng NOR, tầng 2 dùng NOR

Viết hàm F dưới dạng CTH :

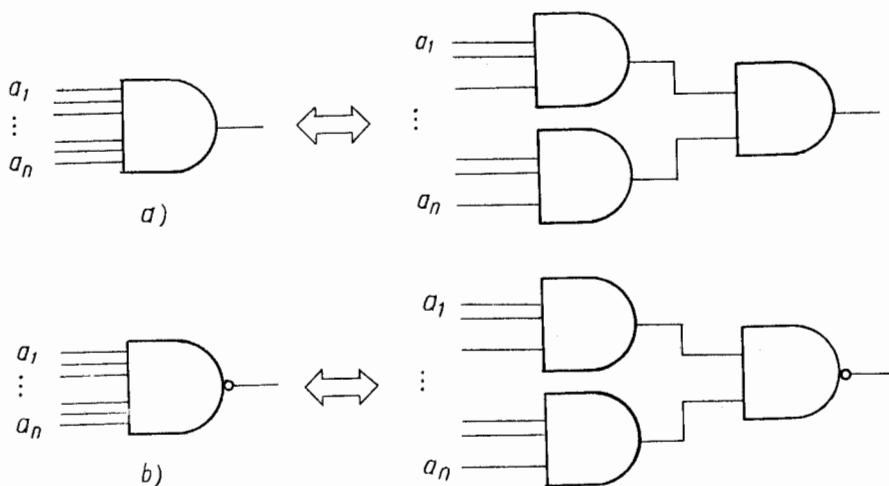
$$F = (\overline{A} + C) (A + B + \overline{C}) .$$

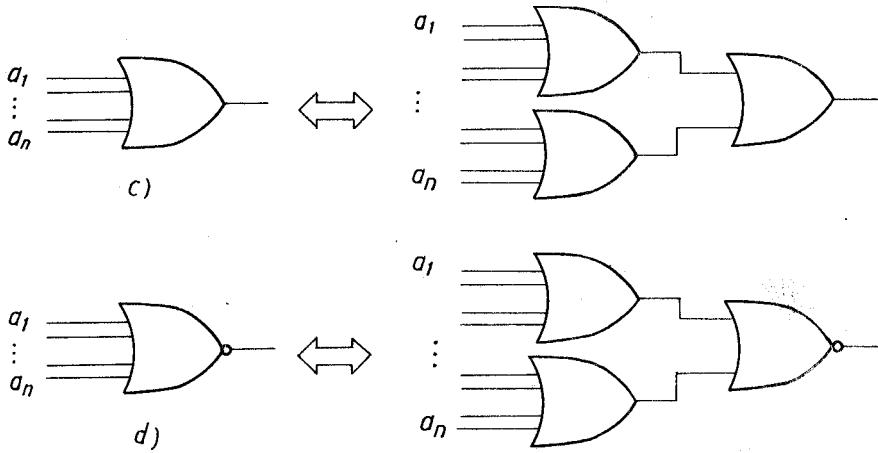
Phủ định 2 lần hàm số rồi áp dụng Demoorgan (các thành phần không biến đổi)

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

b. Mạch nhiều tầng

Khi số đầu vào lớn hơn số đầu vào cho phép của phần tử cho trước lúc đó phải tăng số tầng của mạch. Sự thay thế các mạch AND, OR, NAND, NOR bởi các phần tử cùng loại có số đầu vào nhỏ hơn được cho ở hình 4-6.





Hình 4-6. Thay thế các mạch nhiều đầu vào bởi các mạch cùng loại có số đầu vào ít hơn.

- Thay thế mạch AND;
- Thay thế mạch OR;
- Thay thế mạch NAND;
- Thay thế mạch NOR.

4-3.3. THIẾT KẾ MỘT HỆ HÀM TỔ HỢP

Bài toán:

Xây dựng sơ đồ thực hiện nhiều hàm logic cho trước sao cho sơ đồ là đơn giản nhất.

Giả thiết mạch được xây dựng có 2 tầng trong đó các đầu vào X_i, \bar{X}_i coi như có sẵn và số đầu vào, đầu ra của mỗi mạch không hạn chế.

Ví dụ : Thiết kế hệ 3 hàm tổ hợp

$$f(a, b, c, d)$$

$$g(a, b, c, d)$$

$$h(a, b, c, d)$$

Giá trị của các hàm được cho trong bảng Karnaugh hình 4-7.

		ab		cd			
		00	01	11	10		
		00			1		
		01	1				
		11	1	1			
		10			1		
f							

a)

		ab		cd			
		00	01	11	10		
		00			1	1	
		01		1			
		11				1	
		10			1	1	
g							

b)

		ab		cd			
		00	01	11	10		
		00				1	1
		01	1	1			
		11			1		
		10				1	
h							

c)

Hình 4-7. Bảng Karnaugh của các hàm f, g, h.

- Biểu diễn giá trị hàm f ;
- Biểu diễn giá trị hàm g ;
- Biểu diễn giá trị hàm h.

Hệ hàm tổ hợp này có thể thiết kế theo 2 cách sau :

Thiết kế riêng rẽ từng hàm số :

Từ các bảng Karnaugh biểu diễn các hàm đó, tiến hành tối thiểu hóa nhận được kết quả sau :

$$f = a b \bar{d} + \bar{a} \bar{b} d + \bar{a} c d \quad (C_{min} = 9)$$

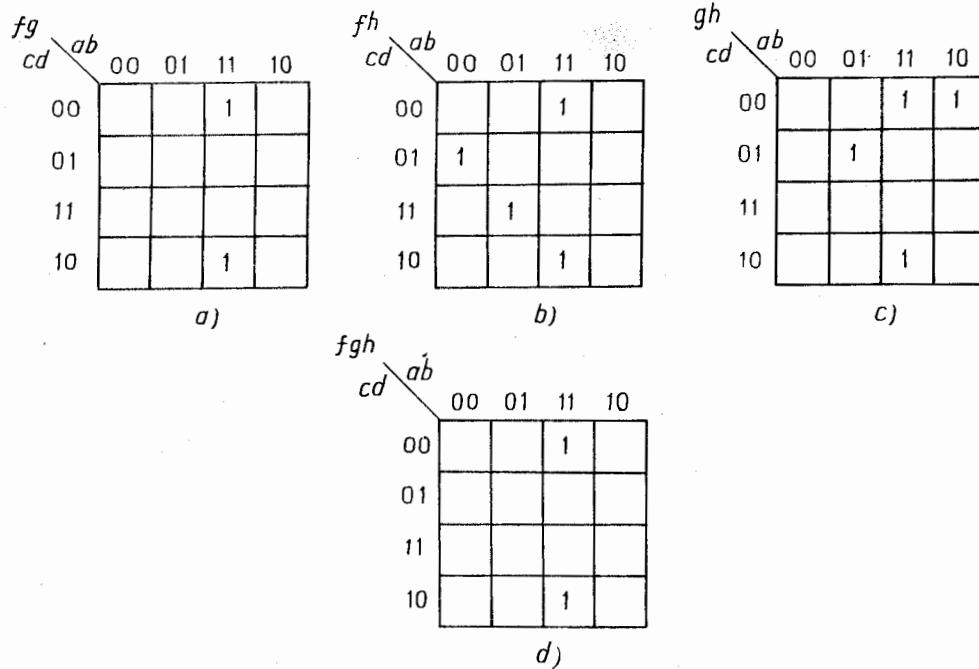
$$g = a \bar{d} + a \bar{b} c + \bar{a} b \bar{c} d \quad (C_{min} = 9)$$

$$h = a \bar{c} \bar{d} + \bar{a} \bar{c} d + \bar{a} b d + a b \bar{d} \quad (C_{min} = 12)$$

Vậy số đầu vào của mạch thực hiện cả 3 hàm là 30.

Thiết kế theo phương pháp sử dụng các phần chung của các hàm.

Lập bảng Karnaugh cho các cặp hàm fg , fh , gh và fgh như hình 4-8.



Hình 4-8. Bảng Karnaugh cho các cặp hàm fg , fh , gh , fgh .

Trên bảng Karnaugh cho từng hàm số f , g , h đánh dấu các phần chung nhau bằng β , những phần riêng của hàm này bằng 1 (hình 4-9).

Từ bảng Karnaugh hình 4-9, viết được biểu thức của các hàm f , g , h như sau :

$$f = a b \bar{d} + \bar{a} b c d + \bar{a} \bar{b} d$$

$$g = a b \bar{d} + a \bar{c} \bar{d} + \bar{a} b \bar{c} d + a \bar{b} c$$

$$h = a b \bar{d} + a \bar{c} \bar{d} + \bar{a} b \bar{c} d + \bar{a} \bar{b} \bar{c} d + \bar{a} b c d.$$

Với các công thức này, xây dựng được sơ đồ mạch thực hiện 3 hàm f , g , h như hình 4-10.

h	cd	ab	00	01	11	10
00					Ø	Ø
01	Ø	Ø				
11		Ø				
10			Ø			

a)

g	cd	ab	00	01	11	10
00					Ø	Ø
01				Ø		
11						1
10					Ø	1

b)

f	cd	ab	00	01	11	10
00					Ø	
01			Ø			
11			1	Ø		
10					Ø	

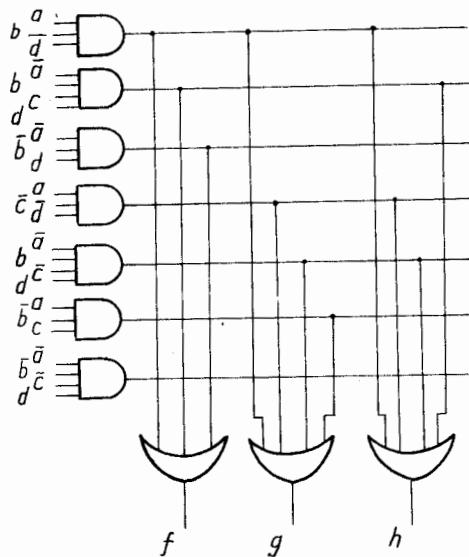
c)

Hình 4-9. Đánh dấu phần chung, riêng của các hàm f , g , h và các hàm chung của nó.

- Dánh dấu phần chung, riêng của hàm h với các hàm gh , gh , fh , fg ;
- Dánh dấu hàm chung, riêng của hàm g với các hàm fgh , gh , fg , h ;
- Dánh dấu hàm chung, riêng của hàm f với các hàm fgh , gh , fh , fg , h , g .

Với sơ đồ này, mạch chỉ có 24 đầu vào, dùng 7 mạch AND, 3 mạch OR trong khi đó sơ đồ thực hiện mạch theo cách thiết kế riêng rẽ từng hàm một có tới 30 đầu vào và phải dùng tới 10 mạch AND, 3 mạch OR.

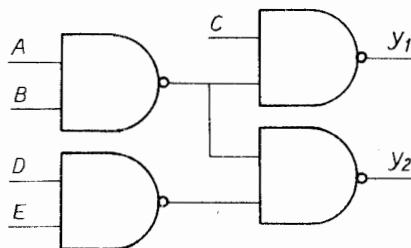
Chú ý rằng cách làm trên cũng hoàn toàn có thể áp dụng cho các hàm phủ định : \bar{f} , \bar{g} , \bar{h} , $\bar{f}\bar{g}$, $\bar{f}\bar{h}$, $\bar{g}\bar{h}$, $\bar{f}\bar{g}\bar{h}$.



Hình 4-10. Sơ đồ mạch thực hiện hệ hàm f , g , h theo phương pháp lấy phần chung.

BÀI TẬP

4.1. Phân tích mạch tổ hợp cho ở hình 4-11:



Hình 4-11.

4.2. Thiết kế mạch tổ hợp tầng 1 dùng NAND, tầng 2 dùng NAND có hàm logic như bảng trên hình 4-12.

f	cd	ab	00	01	11	10
00			1	1		
01					1	
11			1	1		
10						1

Hình 4-12.

f	a	bc	00	01	11	10
0			1			
1				1		1

Hình 4-13.

4.3. Cho hàm logic 3 biến (a, b, c) như bảng trên hình 4-13.

Xây dựng sơ đồ mạch thực hiện hàm này tầng 1 dùng AND, tầng 2 dùng NOR.

4.4. Thiết kế mạch tổ hợp thực hiện 3 hàm tổ hợp : $f(a, b, c, d)$; $g(a, b, c, d)$; $h(a, b, c, d)$ được cho trong bảng Karnaugh sau :

f	cd	ab	00	01	11	10
00					1	
01			1	1		1
11						
10			1			1

a)

g	cd	ab	00	01	11	10
00				1	1	1
01				1		
11						
10						1

b)

h	cd	ab	00	01	11	10
00			1		1	
01						
11			1			1
10						1

c)

Hình 4-14.

CHƯƠNG V

CÁC MẠCH TỔ HỢP THƯỜNG GẶP

Trên cơ sở lý thuyết về phân tích, thiết kế mạch tổ hợp đã trình bày ở chương 4, chương này sẽ giới thiệu một số mạch tổ hợp cụ thể như : bộ cộng (Adder), bộ so sánh (Comparator), bộ chuyển mã, bộ dồn kênh (*MUX*), phân kênh (*DEMUX*), bộ giải mã (DECODER) ... và phương pháp thiết kế chúng.

5-1. BỘ CỘNG NHỊ PHÂN MỘT CỘT SỐ

5-1.1. PHÂN TÍCH BÀI TOÁN

Tương tự như phép cộng 2 số thập phân mà chúng ta đã quen thuộc, phép cộng 2 số nhị phân cũng bắt đầu từ cột số có trọng số nhỏ nhất.

Ví dụ : Thực hiện phép cộng 2 số nhị phân A, B (4 bit) sau :

$$\begin{array}{r} C : \quad 1 \ 1 \ 1 \ 0 \\ A : \quad 1 \ 0 \ 1 \ 1 \\ B : \quad 0 \ 1 \ 1 \ 1 \\ \hline S : \quad 1 \ 0 \ 0 \ 1 \ 0 \end{array} \quad \Rightarrow \text{Dãy nhớ sau phép cộng từng bit}$$

$(11)_{10}$
 $(7)_{10}$

$(18)_{10}$

Từ phép cộng trên, nhận thấy ở mỗi cột số thứ i : thực hiện phép cộng chữ số A_i của số A với chữ số B_i của số B và với số nhớ C_{i-1} của cột số có trọng số nhỏ hơn bên cạnh đưa đến. Kết quả phải đưa ra chữ số của tổng tại cột đó (S_i) và số nhớ sang cột già hơn tiếp theo ngay cạnh đấy (C_i). Vì vậy bộ cộng nhị phân 1 cột số (hay 1 bit) phải có 3 đầu vào A_i, B_i, C_{i-1} và 2 đầu ra là S_i, C_i .

Sơ đồ khối của bộ cộng một cột số được mô tả ở hình 5-1,a với :

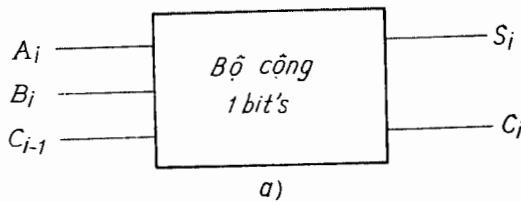
- A_i, B_i là chữ số ở cột thứ i của A, B ;
- C_{i-1} là số nhớ của cột có trọng số nhỏ hơn bên cạnh mang tới;
- S_i là chữ số của tổng ở cột thứ i ;
- C_i là số nhớ đưa đến cột có trọng số lớn hơn bên cạnh.

Bộ cộng nói trên được gọi là bộ cộng đầy đủ 1 bit (*FA* – Full Adder). Bảng chân lý của bộ cộng này được mô tả ở hình 5-1,b.

Biểu diễn tổng S_i và số nhớ C_i trên bảng Karnangh hình 5-1,c, tối thiểu hoá hai hàm này, nhận được kết quả :

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = A_i B_i + C_{i-1} (A_i + B_i)$$



A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

b)

S_i	B_i	C_{i-1}	00	01	11	10
A_i						
0				1		1
1	1		1			

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

C_i	B_i	C_{i-1}	00	01	11	10
A_i						
0					1	
1				1	1	1

$$C_i = A_i B_i + A_i C_{i-1} + B_i C_{i-1}$$

c)

Hình 5-1. Bộ cộng đầy đủ (FA) 1 bit .

a. Sơ đồ của bộ cộng; b. Bảng chân lý của bộ cộng đầy đủ 1 bit; c. Bảng Karnaugh của S_i , C_i .

5-1.2. XÂY DỰNG SƠ ĐỒ

Sơ đồ bộ cộng này có thể xây dựng theo 2 cách sau :

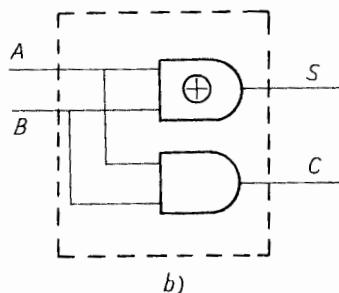
- Xây dựng sơ đồ bộ cộng 1 bit trực tiếp từ hệ phương trình S_i , C_i nói trên.
- Xây dựng sơ đồ bộ cộng 1 bit từ các bộ bán tổng (HA - Half Adder). Cách này thường hay được dùng trong thực tế.

Bảng chân lý của bộ bán tổng được cho ở hình 5-2,a.

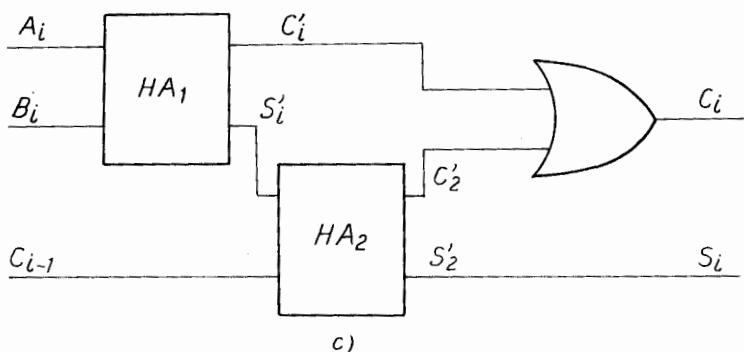
Từ bảng chân lý rút ra $S = a \oplus b$; $c = a \cdot b$. Sơ đồ của bộ cộng đầy đủ 1 bit xây dựng từ các bộ bán tổng được chỉ ra ở hình 5-2,c.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

a)



b)



Hình 5-2. Xây dựng bộ FA từ các bộ HA.

a. Bảng chức năng của bộ bán tổng (HA); b. Sơ đồ bộ bán tổng (HA); c. Sơ đồ bộ FA bằng các bộ HA.

Có thể kiểm tra được tính đúng đắn của sơ đồ này một cách dễ dàng. Theo sơ đồ 5-2c, viết được các phương trình đầu ra :

$$\begin{aligned}
 S_i &= S'_2 = S'_1 \oplus C_{i-1} = A_i \oplus B_i \oplus C_{i-1} \\
 C_i &= C'_1 + C'_2 = A_i B_i + C_{i-1} \cdot S'_1 = A_i B_i + C_{i-1} (A_i \oplus B_i) \\
 &= A_i B_i + (A_i B_i + A_i B_i) C_{i-1} = \\
 &= A_i (B_i + B_i C_{i-1}) + B_i (A_i + A_i + C_{i-1}) \\
 &= A_i (B_i + C_{i-1}) + B_i (A_i + C_{i-1}) \\
 C_i &= A_i B_i + C_{i-1} (A_i + B_i)
 \end{aligned}$$

S_i , C_i đúng là phương trình đầu ra của bộ cộng đầy đủ 1 bit.

5-2. BỘ TRỪ NHỊ PHÂN MỘT CỘT SỐ

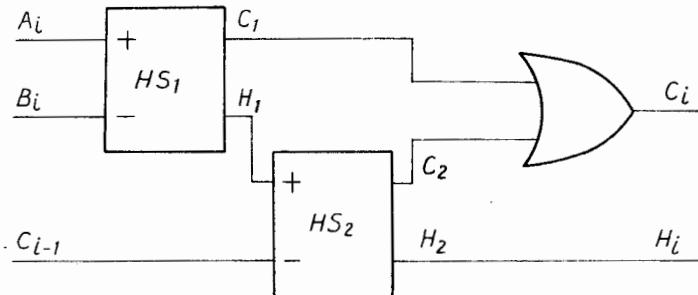
Cách làm hoàn toàn tương tự như khi xây dựng sơ đồ thực hiện bộ cộng nhị phân 1 cột số. Bảng chân lý của bộ trừ đầy đủ 1 bit (FS - Full Subtractor) cho ở hình 5-3,a.

A_i Chữ số thứ i của số bị trừ	B_i Chữ số thứ i của số trừ	C_{i-1} Nhớ từ cột có trọng số nhỏ hơn đưa đến	H_i Chữ số của hiệu	C_i Số nhớ đưa tới cột có trọng số lớn hơn
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

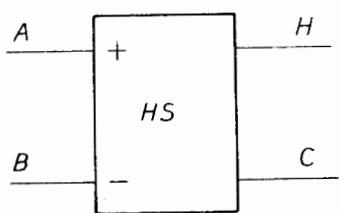
a)

A	B	H	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

b)



c)



Hình 5-3. Bộ trừ nhị phân một cột số.

a. Bảng chức năng của bộ bán trừ đầy đủ (FS) ; b. Bảng chức năng của bộ bán trừ (HS) ; c. Sơ đồ khối của bộ bán trừ (HS) ; d. Sơ đồ bộ FS dùng các bộ HS.

Tương tự như bộ cộng, ta có thể xây dựng bộ trừ đầy đủ 1 bit trực tiếp từ hệ phương trình viết từ bảng chân lý 5-3,a hoặc qua các bộ bán trừ (HS - Half Subtractor). Bảng chân lý của bộ bán trừ (HS) được mô tả ở hình 5-3,b, từ bảng chức năng này viết được hệ phương trình của bộ bán trừ :

$$\begin{aligned} H &= A \oplus B & (H \text{ là chữ số của hiệu}); \\ C &= \bar{A} B & (C \text{ là số nhớ sang cột tiếp theo}). \end{aligned}$$

Sơ đồ của khối bán trừ được mô tả ở hình 5-3,c. Trên cơ sở đó xây dựng bộ trừ đầy đủ 1 bit qua các bộ bán trừ như hình 5-3,d.

Trong thực tế, bộ trừ đầy đủ 1 bit ít được dùng mà người ta thường tìm cách biểu diễn các số theo dạng mã bù và biến phép trừ thành phép cộng với mã bù của nó. Điều này sẽ được xét chi tiết trong chương 13.

5_3. BỘ SO SÁNH (COMPARATOR)

Trong nhiều trường hợp phải so sánh 2 số nhị phân A và B để chỉ ra được mối quan hệ giữa chúng : $A > B$ hay $A < B$ hay là $A = B$.

Mạch tổ hợp thực hiện chức năng nói trên được gọi là bộ so sánh. Trước tiên chúng ta hãy xét bộ so sánh 2 số nhị phân 1 bit và sau đó thiết kế các bộ so sánh 2 số nhị phân nhiều bit .

5_3.1. BỘ SO SÁNH 2 SỐ NHỊ PHÂN 1 BIT

Bảng chức năng của bộ so sánh này được cho trong hình 5-4,a. Từ bảng chức năng hình 5-4,a viết được các hàm đầu ra của bộ so sánh như sau :

$$\text{Hàm } (A_i = B_i) = A_i \overline{\oplus} B_i$$

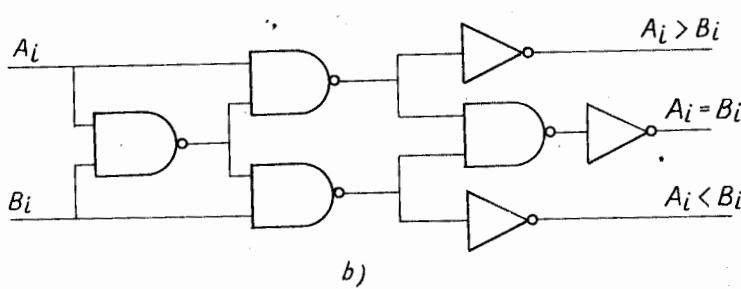
$$\text{Hàm } (A_i < B_i) = \bar{A}_i B_i$$

$$\text{Hàm } (A_i > B_i) = A_i \bar{B}_i$$

Trên cơ sở các hàm này, xây dựng được sơ đồ bộ so sánh 1 bit chỉ dùng các mạch NAND như hình 5-4,b.

A_i	B_i	$A_i = B_i$	$A_i < B_i$	$A_i > B_i$
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

a)



Hình 5-4. Bộ so sánh 1 bit.

a. Bảng chức năng ; b. Sơ đồ bộ so sánh chỉ dùng NAND.

5_3.2. BỘ SO SÁNH n BIT

Giả sử có 2 số nhị phân n bit A, B với :

$$A : \quad A_n \ A_{n-1} \ \dots \ A_1$$

$$B : \quad B_n \ B_{n-1} \ \dots \ B_1$$

trong đó qui ước A_1, B_1 là cột có trọng số nhỏ nhất, A_n, B_n là cột có trọng số lớn nhất.

Sơ đồ khối của bộ so sánh 2 số A, B được mô tả ở hình 5-5,a.

Để xây dựng sơ đồ mạch bộ so sánh này có thể dùng 2 cách như sau :

a. *Xây dựng sơ đồ trực tiếp từ các hàm sau*

$$A > B \quad (2n \text{ biến})$$

$$A = B \quad (2n \text{ biến})$$

$$A < B \quad (2n \text{ biến}).$$

Với phương pháp này thực chất là xây dựng một hệ 3 hàm logic, mỗi hàm có $2n$ biến.

b. *Xây dựng sơ đồ gián tiếp qua các bộ so sánh 1 bit*

Xét ví dụ cụ thể với $n = 3$

$$A = A_3 \ A_2 \ A_1$$

$$B = B_3 \ B_2 \ B_1$$

Để xây dựng được sơ đồ mạch so sánh này, phải cần 3 mạch so sánh 1 bit và các mạch phù trợ.

Dầu tiên so sánh 2 cột số có trọng lượng lớn nhất A_3, B_3 :

$$A_3 > B_3 : A > B$$

$$A_3 < B_3 : A < B$$

$$A_3 = B_3 : \text{so sánh tiếp}$$

Trong trường hợp $A_3 = B_3$ ta phải tiếp tục so sánh 2 chữ số A_2, B_2 có thể xảy ra trường hợp :

$$A_2 > B_2 : A > B$$

$$A_2 < B_2 : A < B$$

$$A_2 = B_2 : \text{so sánh tiếp}$$

Nếu $A_2 = B_2$ ta phải tiếp tục so sánh 2 chữ số A_1, B_1 :

$$A_1 > B_1 : A > B$$

$$A_1 < B_1 : A < B$$

$$A_1 = B_1 : A = B$$

Tóm lại

$$(A = B) \Leftrightarrow (A_3 = B_3)(A_2 = B_2)(A_1 = B_1)$$

$$(A > B) \Leftrightarrow (A_3 > B_3) + (A_3 = B_3)(A_2 > B_2) + (A_3 = B_3)(A_2 = B_2)(A_1 > B_1)$$

$$(A < B) \Leftrightarrow (A_3 < B_3) + (A_3 = B_3)(A_2 < B_2) + (A_3 = B_3)(A_2 = B_2)(A_1 < B_1)$$

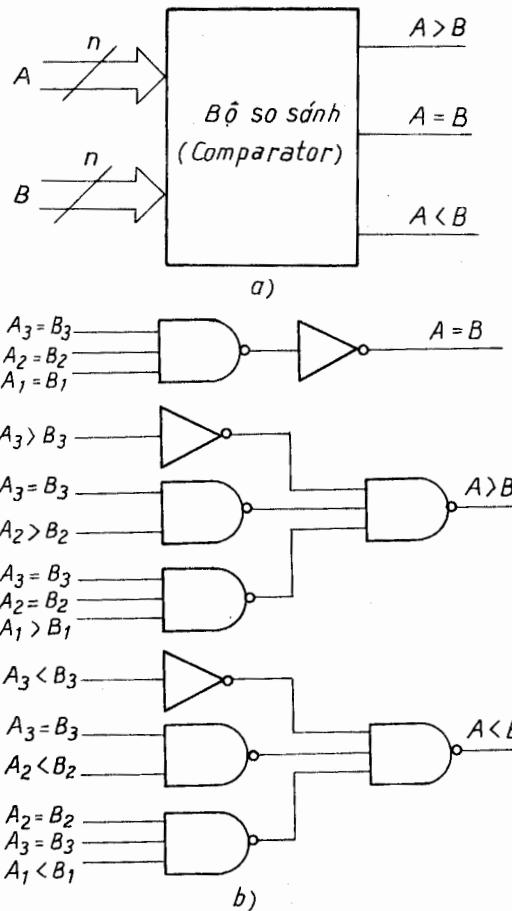
Trên cơ sở đó xây dựng được sơ đồ bộ so sánh 2 số 3 bit dùng các bộ so sánh 1 bit và các mạch *NAND* như hình 5-5.b. Trong đó

$$A_3 = B_3, A_3 > B_3, A_3 < B_3,$$

$$A_2 = B_2, A_2 > B_2, A_2 < B_2,$$

$$A_1 = B_1, A_1 > B_1, A_1 < B_1,$$

được đưa từ các đầu ra của 3 bộ so sánh 1 bit đến.



Hình 5-5. Bộ so sánh nhiều bit.

a. Sơ đồ khối của bộ so sánh n bit ; b. Sơ đồ của bộ so sánh 3 bit được xây dựng từ bộ so sánh 1 bit.

5_4. MẠCH TẠO VÀ KIỂM TRA CHẴN LẺ (PARITY GENERATOR / CHECKER)

Trong thông tin, khi truyền dữ liệu (Data) từ nơi này sang nơi khác cần phải xác định xem có lỗi của sự truyền hay không ? Và nếu có thể thì sửa lỗi đó.

Có nhiều phương pháp mã hoá dữ liệu để có thể làm được điều này. Phương pháp đơn giản nhất là thêm một bit vào dữ liệu được truyền đi để sao cho số chữ số 1 trong dữ liệu luôn luôn là 1 số lẻ hoặc chẵn. Bit đó được gọi là bit chẵn lẻ.

Bit chẵn : Nếu bit thêm vào có giá trị sao cho chữ số 1 trong dữ liệu là 1 số chẵn (*Even*).

Bit lẻ : Nếu bit thêm vào có giá trị sao cho số chữ số 1 trong dữ liệu là 1 số lẻ (*Odd*).

Ví dụ : Với mã ASCII, chữ số 9 trong hệ thập phân được mã hoá như sau :

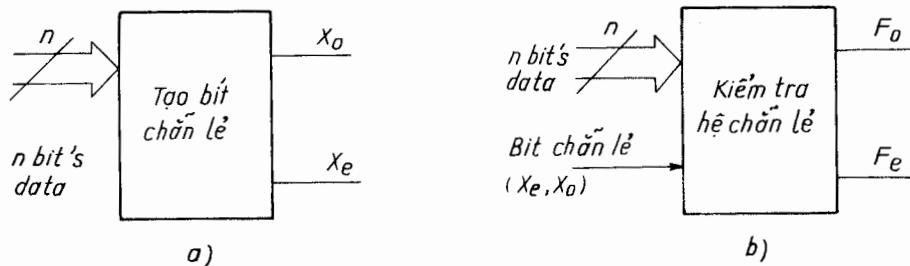
$(9)_{10} = X011.1001$; X là bit thêm vào nếu $X = 0$: hệ chẵn ; $X = 1$: hệ lẻ.

Để thực hiện được việc truyền dữ liệu theo kiểu đưa thêm bit chẵn, lẻ vào dữ liệu chúng ta cần phải :

1. Xây dựng sơ đồ tạo được bit chẵn, lẻ thêm vào n bit của dữ liệu.
2. Xây dựng sơ đồ kiểm tra được hệ là hệ chẵn hay lẻ với $(n + 1)$ bit ở đầu vào (n bit của dữ liệu, 1 bit chẵn, lẻ).

Sơ đồ khối của mạch tạo bit chẵn lẻ và mạch kiểm tra hệ chẵn lẻ được mô tả ở hình 5-6.

Để ví dụ, xét trường hợp đơn giản với dữ liệu có 3 bit . Trường hợp dữ liệu có nhiều bits hơn cách làm cũng hoàn toàn tương tự.



Hình 5-6. Sơ đồ khối mạch tạo, kiểm tra chẵn lẻ,
a. Mạch tạo bit chẵn lẻ ; b. Mạch kiểm tra hệ chẵn lẻ,

5_4.1. MẠCH TẠO BIT CHẴN LẺ

Gọi 3 bit của dữ liệu là d_1, d_2, d_3 và X_e, X_o là 2 bit chẵn, lẻ thêm vào dữ liệu. Bảng chân lý của mạch tạo bit chẵn lẻ cho trường hợp dữ liệu vào 3 bit được mô tả ở hình 5-7.

Ở đây X_o là giá trị bit phải thêm vào để hệ là hệ chẵn. X_e là giá trị bit phải thêm vào để hệ là hệ lẻ.

Từ bảng chân lý ta thấy $X_o = \bar{X}_e$ hay $X_e = \bar{X}_o$.

Do tính chất của hàm cộng Modul n biến : hàm bằng 1 nếu như trong n biến, số số biến bằng 1 là 1 số lẻ. Vậy viết ngay được :

$$X_e = d_1 \oplus d_2 \oplus d_3$$

$$X_o = \overline{X_e} = \overline{d_1 \oplus d_2 \oplus d_3}.$$

Vào			Ra	
d_1	d_2	d_3	X_e	X_o
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Hình 5-7 . Bảng chức năng của mạch tạo bit chẵn lẻ.

5.4.2. MẠCH KIỂM TRA CHẴN LẺ

Bảng chức năng của mạch kiểm tra tính chẵn lẻ của hệ được cho trong hình 5-8.

Vào				Ra	
d_1	d_2	d_3	X	F_e	F_o
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

Hình 5-8. Bảng chức năng của mạch kiểm tra chẵn, lẻ.

Từ bảng ta thấy :

- $F_e = 1$ nếu hệ là hệ chẵn (F_e chỉ ra tính chẵn của hệ);
 $F_o = 1$ nếu hệ là hệ lẻ (F_o chỉ ra tính lẻ của hệ).

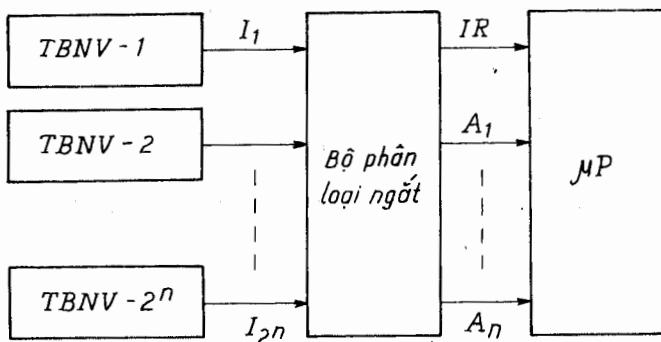
Hai hàm này luôn là phủ định của nhau. Mặt khác do tính chất của hàm cộng Modul, ta có :

$$F_o = d_1 \oplus d_2 \oplus d_3 \oplus X$$

$$F_o = \bar{F}_e .$$

5_5. MẠCH PHÂN LOẠI NGẮT

Trong kỹ thuật, thường có nhiều thiết bị ngoại vi được nối với máy tính, μP . Mạch phân loại ngắt (Interrupt Sorter) như mô tả ở hình 5-9.



Hình 5-9. Sơ đồ khối của bộ phân loại ngắt (IS).

Mạch có nhiệm vụ :

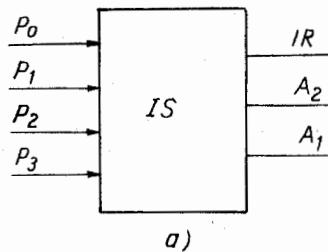
- Tại thời điểm t nếu có ít nhất 1 trong các thiết bị ngoại vi có yêu cầu ngắt thì mạch phải tạo ra tín hiệu yêu cầu ngắt IR (Interrupt Request) gửi tới μP .

- Nếu tại thời điểm t có nhiều thiết bị ngoại vi cùng gửi yêu cầu ngắt tới μP thì mạch phải chỉ ra được thiết bị ngoại vi nào (mã của thiết bị ngoại vi đó) cần ưu tiên giải quyết ngắt tại thời điểm đó. $IR = 1$ khi có ít nhất 1 trong các thiết bị ngoại vi có yêu cầu ngắt gửi tới μP . Các tổ hợp $(A_1, A_2, \dots, A_n)_2$ là các mã nhị phân tương ứng của các thiết bị ngoại vi.

Vậy bộ phân loại ngắt sẽ có 2^n đầu vào, $(n + 1)$ đầu ra như sơ đồ hình 5-9.

Ví dụ : Xây dựng sơ đồ bộ phân loại ngắt (IS) có 4 đầu vào, 3 đầu ra. Sơ đồ khối của bộ phân loại ngắt này được mô tả ở hình 5-10,a. Giả thiết các thiết bị ngoại vi sẽ được ưu tiên xử lý ngắt theo thứ tự 3, 2, 1, 0, các đầu ra A_1, A_2 được qui ước : A_1 biểu diễn bit có trọng số 2^0 , A_2 biểu diễn bit có trọng số 2^1 .

Bảng chức năng của bộ phân loại ngắt được biểu diễn trên hình 5-10,b.



a)

Vào				Ra		
P_0	P_1	P_2	P_3	IR	A_2	A_1
0	0	0	0	0	0	0
X	X	X	1	1	1	1
X	X	1	0	1	1	0
X	1	0	0	1	0	1
1	0	0	0	1	0	0

b)

Hình 5-10. Bộ phân loại ngắt 4 đầu vào, 3 đầu ra.

Từ bảng chức năng xác định được các hàm đầu ra như sau :

$$IR = P_0 + P_1 + P_3$$

$$A_2 = P_3 + P_2 \bar{P}_3$$

$$A_1 = P_3 + P_1 \bar{P}_2 \bar{P}_3$$

Trong đó $P_i = 1$ ($i = 1 \div 3$) chỉ ra rằng thiết bị ngoại vi thứ i có yêu cầu ngắt, $IR = 1$ khi có yêu cầu ngắt từ ít nhất một trong các thiết bị ngoại vi gửi tới μP .

5_6. BỘ DỒN KÊNH (MUX) HAY BỘ CHỌN DỮ LIỆU (DATA SELECTER)

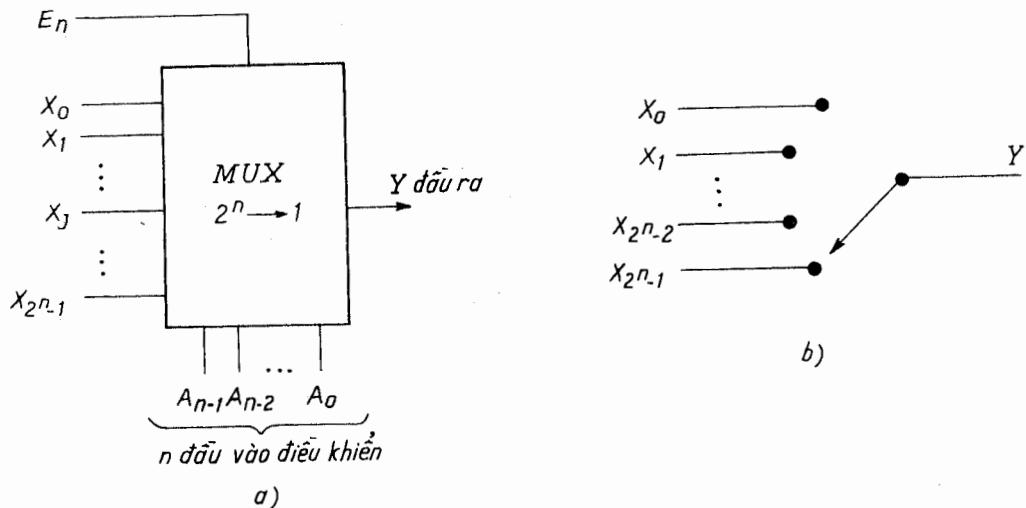
Định nghĩa

Bộ dồn kênh là mạch có 2^n đầu vào, n đầu vào điều khiển, 1 đầu vào chọn mạch và 1 đầu ra.

Tùy theo giá trị của n đầu vào điều khiển mà đầu ra sẽ bằng một trong những giá trị ở đầu vào (X_j). Cụ thể nếu giá trị thập phân của n đầu vào điều khiển bằng j thì $Y = X_j$. Nghĩa là nếu :

$$(A_{n-1} A_{n-2} \dots A_1 A_0)_2 = (J)_{10} \text{ thì } Y = X_j.$$

Sơ đồ khối của MUX $2^n \Rightarrow 1$ (2^n đầu vào, 1 đầu ra) được biểu diễn ở hình 5-11,a.



Hình 5-11. Bộ dồn kênh MUX $2^n \Rightarrow 1$

a. Sơ đồ khối ; b. MUX là một chuyển mạch điện tử.

Phương trình tín hiệu ra của MUX $2^n \Rightarrow 1$ như sau :

$$Y = X_0(\bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_1 \dots \bar{A}_0) + X_1(\bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_1 \dots \bar{A}_1 A_0) \\ + \dots + X_{2^n-1}(A_{n-1} A_{n-2} \dots A_1 \dots A_1 A_0)$$

Nếu ký hiệu \tilde{A}_i là tích bộ phận của $A_{n-1} A_{n-2} \dots A_i \dots A_1 A_0$.

$\tilde{A}_i = 1$ khi giá trị thập phân của số nhị thập phân $(A_{n-1} A_{n-2} \dots A_0)_{10} = i$.

Tổng quát có thể viết :

$$Y = X_0(\tilde{A}_0) + X_1(\tilde{A}_1) + \dots + X_i(\tilde{A}_i) + \dots + X_{2^n-1}X_{2^n-1}(\tilde{A}_{2^n-1})$$

Thực chất MUX là chuyển mạch điện tử dùng các tín hiệu điều khiển $(A_{n-1} A_{n-2} \dots A_0)$ để điều khiển sự nối mạch của đầu ra với 1 trong số 2^n đầu vào (hình 5-11,b).

Hiện nay bộ MUX được dùng như một phần tử vạn năng để xây dựng những mạch tổ hợp khác.

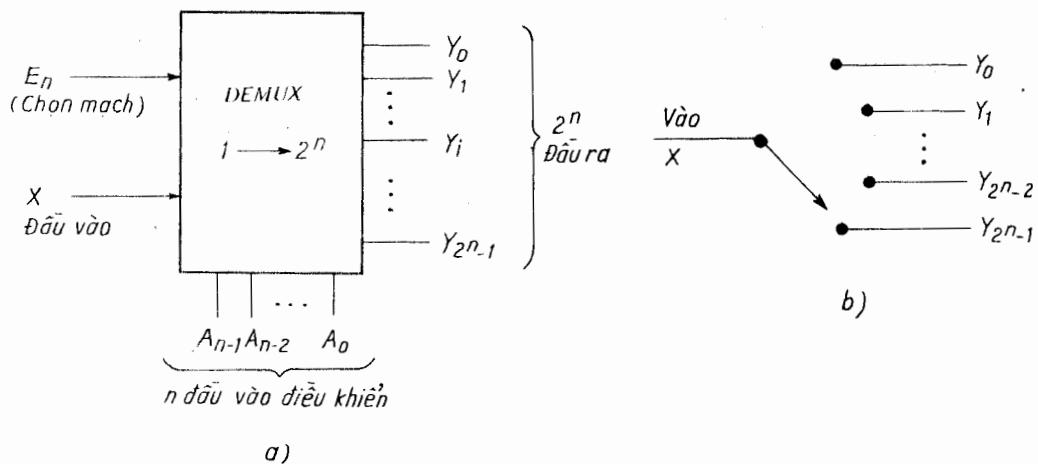
5-7. BỘ PHÂN KÊNH (DEMUX)

Định nghĩa

Bộ phân kênh là mạch có 1 đầu vào, n đầu vào điều khiển, 1 đầu vào chọn mạch và 2^n đầu ra.

Tùy theo giá trị của n đầu vào điều khiển mà đầu ra thứ i (Y_i) sẽ bằng giá trị của đầu vào. Cụ thể nếu gọi n đầu vào điều khiển là $A_{n-1} A_{n-2} \dots A_0$ thì $Y_i = X$ khi $(A_{n-1} A_{n-2} \dots A_1 A_0)_2 = (i)_{10}$.

Sơ đồ khối của bộ *DEMUX* 1 đầu vào 2^n đầu ra ($MUX 1 \rightarrow 2^n$) được biểu diễn ở hình 5-12,a.



Hình 5-12. Bộ phân kênh *DEMUX*.

a. Sơ đồ khối ; b. *DEMUX* là một chuyển mạch điện tử.

Phương trình tín hiệu ra của *DEMUX* $1 \Rightarrow 2^n$ như sau :

$$\begin{aligned} Y_0 &= X \cdot \bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_i \dots \bar{A}_0 \\ Y_1 &= X \cdot \bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_i \dots \bar{A}_1 A_0 \\ &\dots \dots \dots \dots \dots \dots \\ Y_{2^n-1} &= X \cdot A_{n-1} A_{n-2} \dots A_i \dots A_0 \end{aligned}$$

Ký hiệu \tilde{A}_i là tích bộ phận của $A_{n-1}, A_{n-2} \dots A_1 A_0$

$$\tilde{A}_i = 1 \text{ khi } (A_{n-1}, A_{n-2} \dots A_1 A_0)_2 = (i)_{10}$$

Tổng quát : $Y_j = X \cdot \tilde{A}_j$

Bộ phân kênh còn được gọi là bộ giải mã 1 trong 2^n . Tại một thời điểm chỉ có 1 trong số 2^n đầu ra ở mức tích cực.

5_8. MẠCH CHUYỂN MÃ

Trong các hệ thống điện tử dùng mạch số, dữ liệu (lệnh, số liệu) được truyền đi hay xử lý ở dạng từ (Word) nhị phân gồm các bit 0 và 1. Một từ n bit có thể biểu diễn cho 2^n phân tử tin khác nhau với giá trị thập phân từ : $0 \Rightarrow 2^n - 1$. Từ nhị phân n bits đó gọi là mã (Code) của phân tử tin tức. Có rất nhiều loại mã, phân này sẽ trình bày một số loại mã tiêu biểu thường dùng, sau đó sẽ xét đến các mạch logic thực hiện chuyển mã.

5.8.1. CÁC LOẠI MÃ TIÊU BIỂU

a. Các loại mã ký tự thường dùng

- Mã **ASCII** (American Standard Code for Information Interchange) : mã này dùng 8 bit để mã hoá cho 1 ký tự.
- Mã **EBCDI** (Extended Binary Coded Decimal Interchange Code) : dùng 8 bit để mã hoá cho 1 ký tự.
- Mã **BAUDOT** : mã này dùng 5 bit để biểu diễn cho 1 ký tự, nó thường được dùng trong bưu điện và một số loại teletype.

b. Các loại mã thường dùng để mã hoá các con số

Để mã hoá các con số thường dùng các loại mã : mã nhị phân, mã dư 3 (XS3), mã Gray, mã Gray dư 3, mã nhị phân của số thập phân (*BCD*), mã thập phân, mã 7 vạch ... Bảng mã của một số loại mã này được cho trong hình 5-13,a.

Mã có thể chia làm 2 loại : loại có trọng số (trọng số của các ký hiệu nhị phân được sắp xếp theo thứ tự $2^0, 2^1, 2^2, 2^n \dots$) và loại không có trọng số (trọng số của các ký hiệu nhị phân không phụ thuộc vào vị trí của chúng trong từ mã).

- Mã nhị phân :

Đây là loại mã có trọng số, trọng số của các ký hiệu nhị phân được xắp xếp từ thấp đến cao là 1, 2, 4, 8, ...

- Mã *BCD* (*binary coded decimal*)

Để mã hóa nhị phân cho 10 chữ số thập phân cần từ mã có độ dài 4 bit. Tùy theo cách sử dụng 10 trên 16 tổ hợp mã nhị phân 4 bit mà ta có các loại mã *BCD* khác nhau. Một số mã *BCD* thường gặp là : *BCD - Normal* (hay *NBCD*), 2421, 5121 ...

Mã *NBCD* là loại mã *BCD* đơn trị và có trọng số (8, 4, 2, 1). Ví dụ mã *NBCD* của 9 là 1001.

Mã 2421 và mã 5121 là loại mã *BCD* không đơn trị, nhưng chúng có đặc điểm là mã bù 9 của một số nào đó cũng chính là mã bù 1 của số đó. Ví dụ mã 5121 của số 4 là 0111, mã bù 9 của 4 là 5 (1000) chính là mã bù 1 của 0111. Đặc điểm này sẽ thể hiện rõ ưu điểm của nó khi xây dựng các mạch số học (sẽ xét trong chương 13).

- Mã *du 3* (XS3)

Mã này được tạo thành từ mã nhị phân bằng cách cộng thêm 0011 (3) vào từ mã nhị phân tương ứng.

- Mã *Gray* :

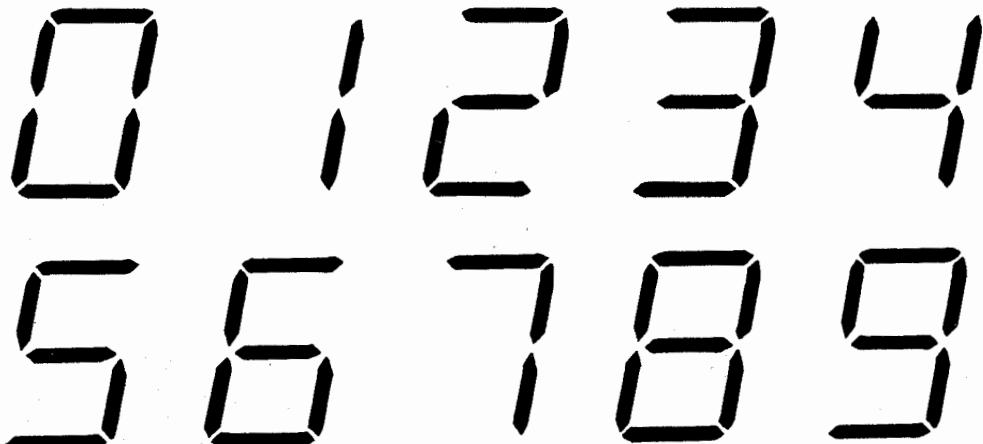
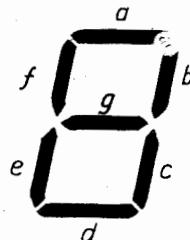
Là loại mã không có trọng số, các từ mã kế cận nhau chỉ khác nhau ở một biến số. Mã *Gray* được dùng để biểu diễn bảng Karnaugh.

- Mã *Gray du 3* :

Mã này cũng có đặc điểm như mã *Gray* và nó được tạo thành từ mã *Gray* bằng cách lêch đi 3 hàng.

Mã nhị phân	Dú 3	Gray	Dú 3 Gray	Mã nhị phân của số thập phân				Thập phân	7 vạch		
				BCD			Số thập phân				
				Normal 1	2421	5121					
0000	0011	0000	0010	0000	0000	0000	0	1000000000	1111110		
0001	0100	0001	0110	0001	0001	0001	1	0100000000	0110000		
0010	0101	0011	0111	0010	0010	0010	2	0010000000	1101101		
0011	0110	0010	0101	0011	0011	0011	3	0001000000	1111001		
0100	0111	0110	0100	0100	0100	0111	4	0000100000	0110011		
0101	1000	0111	1100	0101	1011	1000	5	0000010000	1011011		
0110	1001	0101	1101	0110	1100	1001	6	0000001000	1011111		
0111	1010	0100	1111	0111	1101	1010	7	0000000100	1110000		
1000	1011	1100	1110	1000	1110	1011	8	0000000010	1111111		
1001	1100	1101	1010	1001	1111	1111	9	0000000001	1111011		
1010	1101	1111	1011								
1011	1110	1110	1001								
1100	1111	1010	1000								
1101	0000	1011	0000								
1110	0001	1001	0001								
1111	0010	1000	0011								

a)



b)

Hình 5-13. Một số mã thường dùng để mã hóa con số.
a. Bảng mã; b. Đèn hiển thị 7 vạch và các con số tương ứng.

- **Mã sửa sai :**

Trong các mã sửa sai ngoài các bits mang thông tin còn có một số bit thêm vào để phát hiện và sửa sai. Mã sửa sai đơn giản nhất là mã chẵn lẻ, khi đó bit thêm vào gọi là bit chẵn lẻ (Parity bit).

- **Mã bảy vạch:**

Mã này dùng một từ mã có độ dài bằng 7 để biểu diễn một chữ số (hình 5-13,b).

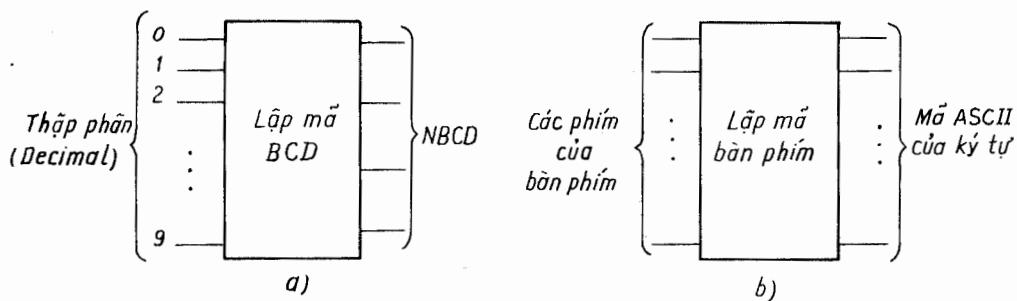
5.8.2. PHÂN LOẠI CÁC MẠCH CHUYỂN MÃ

Mạch chuyển mã là mạch thực hiện chức năng chuyển từ 1 loại mã này thành loại mã khác tương ứng. Ví dụ : Chuyển mã *NBCD* (*BCD Normal*) \Rightarrow mã dư 3 (*XS3*), mã dư 3 (*XS3*) \Rightarrow mã nhị phân, mã nhị phân \Rightarrow mã *Gray*, mã *Gray* \Rightarrow mã nhị phân, mã dư 3 \Rightarrow mã dư 3 *Gray* ...

Chi tiết hơn, còn có thể phân các mạch chuyển mã thành 2 loại sau :

a. **Mạch mã hóa – Lập mã (ENCODER)**

Mạch này có nhiệm vụ tạo ra các mã số. Ở mỗi thời điểm chỉ có một đầu vào ở mức tích cực. Ví dụ : mạch lập mã *NBCD* của số thập phân, mạch lập mã *ASCII* của bàn phím có sơ đồ khối được mô tả ở hình 5-14.



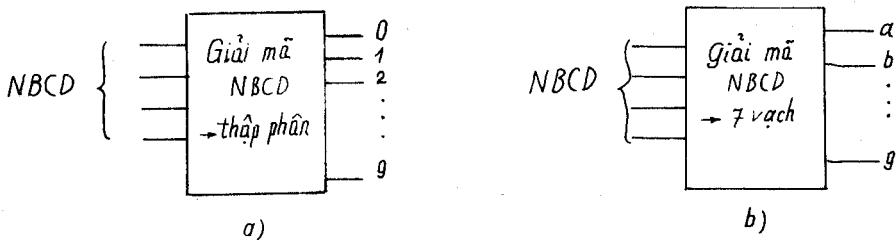
Hình 5-14. Sơ đồ khối của mạch lập mã.

a. Mạch lập mã *NBCD* ; b. Mạch lập mã *ASCII*.

Mạch lập mã ưu tiên (Priority Encoder) là mạch mà nếu tại một thời điểm t có thể có nhiều đầu vào ở mức tích cực thì mạch sẽ tạo ra mã của đầu vào được ưu tiên ở thời điểm đó.

b. **Mạch giải mã (DECODER)**

Mạch này có nhiệm vụ biến đổi mã nhận được thành mã ban đầu (ngược với mạch mã hóa). Sơ đồ khối của mạch giải mã *NBCD* sang thập phân và mạch giải mã *NBCD* sang mã 7 vạch (để hiển thị các con số) được mô tả ở hình 5-15.



Hình 5-15. Sơ đồ khối của mạch giải mã.

a. Mạch giải mã N BCD sang mã thập phân ; b. Mạch giải mã N BCD sang mã 7 vạch.

5-8.3. THIẾT KẾ MẠCH CHUYỂN MÃ

Mạch chuyển mã là một mạch tổ hợp có nhiều đầu vào và nhiều đầu ra. Các bước thiết kế mạch chuyển mã hoàn toàn tương tự như các bước thiết kế mạch tổ hợp nghĩa là :

- Lập bảng chân lý của mạch ;
- Tối thiểu các hàm ra ;
- Xây dựng sơ đồ mạch thực hiện.

Để cho vấn đề được rõ ràng, xét một số ví dụ cụ thể sau.

1. Mạch chuyển mã nhị phân sang mã Gray

Bảng chân lý của mạch được cho trong hình 5-16,a, sơ đồ khối của mạch như hình 5-16,b. Biểu diễn các hàm ra :

$$\begin{aligned} G_3 & (B_3, B_2, B_1, B_0); \\ G_2 & (B_3, B_2, B_1, B_0); \\ G_1 & (B_3, B_2, B_1, B_0); \\ G_0 & (B_3, B_2, B_1, B_0). \end{aligned}$$

của mạch trên bảng Karnaugh hình 5-16,c.

Tối thiểu hóa nhận được kết quả :

$$\begin{aligned} G_3 & = B_3 \\ G_2 & = \bar{B}_3 B_2 + B_3 \bar{B}_2 = B_2 \oplus B_3 \\ G_1 & = \bar{B}_1 B_2 + B_1 \bar{B}_2 = B_1 \oplus B_2 \\ G_0 & = \bar{B}_1 B_0 + B_1 \bar{B}_0 = B_1 \oplus B_0 \end{aligned}$$

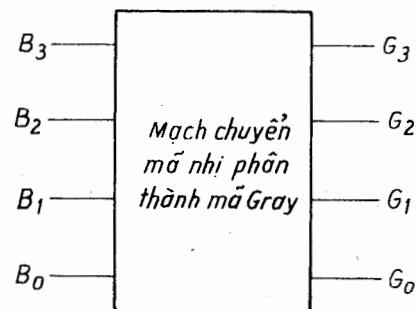
Căn cứ vào các hàm này lập được sơ đồ mạch thực hiện như hình 5-16,d (sơ đồ chỉ dùng mạch NAND).

2. Mạch chuyển mã Gray sang mã nhị phân

Bảng chân lý của mạch như đã cho trong hình 5-16,a. Từ bảng chia năng lập được sơ đồ khối như hình 5-17.

Mã nhị phân				Mã Gray			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

a)



b)

		B_3	B_2	B_1	B_0
		00	01	11	10
B_3	B_2	00			
		01			
B_3	B_2	11	1	1	1
		10	1	1	1

c)

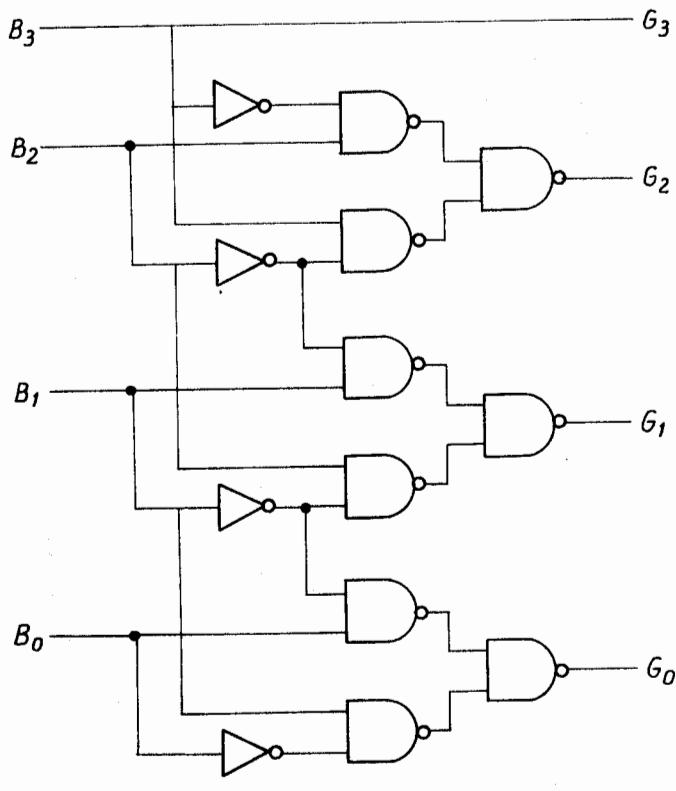
		G_2	B_3	B_2	B_1	B_0
		00	00	01	11	10
G_2	B_3	00				
		01	1	1	1	1
G_2	B_3	11				
		10	1	1	1	1

Hình 5-16.

B_3B_2	B_1B_0	00	01	11	10
00				1	1
01		1	1		
11		1	1		
10			1	1	

B_3B_2	B_1B_0	00	01	11	10
00		1			
01			1		1
11		1			1
10			1		1

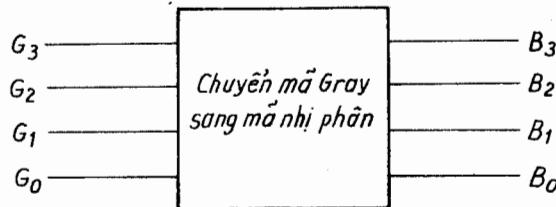
c)



d)

Hình 5-18. Mạch chuyển mã nhị phân \Rightarrow Gray.

- a. Bảng chúc năng ; b. Sơ đồ khối ; c. Bảng Karnaugh các hàm đầu ra ;
- d. Sơ đồ mạch thực hiện.



Hình 5-17. Sơ đồ khối của mạch chuyển mã Gray \Rightarrow nhị phân.

Lập bảng Karnaugh các hàm ra :

$$\begin{aligned}B_3(G_3, G_2, G_1, G_0); \\B_2(G_3, G_2, G_1, G_0); \\B_1(G_3, G_2, G_1, G_0); \\B_0(G_3, G_2, G_1, G_0)\end{aligned}$$

và tối thiểu hóa ta được kết quả :

$$\begin{aligned}B_3 &= G_3 \\B_2 &= G_2 \oplus G_3 \\B_1 &= G_1 \oplus G_2 \oplus G_3 \\B_0 &= G_0 \oplus G_1 \oplus G_2 \oplus G_3\end{aligned}$$

Từ hệ các hàm ra này dễ dàng lập được mạch thực hiện dùng các phần tử tổ hợp cho trước.

3. Mạch chuyển mã dư 3 (XS3) \Leftrightarrow dư 3 - Gray

Bảng chân lý cửa mạch cho trong hình 5-18,a. Từ bảng chân lý xác định được sơ đồ khối của mạch chuyển mã dư 3 (XS3) sang mã dư 3 - Gray (hình 5-18,b) và mạch chuyển mã dư 3 - Gray sang mã dư 3 (hình 5-18,c). Lập bảng Karnaugh cho các hàm ra và tối thiểu hóa các hàm này nhận được kết quả .

a) Mạch chuyển mã dư 3 \Rightarrow dư 3 - Gray

$$\begin{array}{ll}G^*_3 = B^*_3 & G^*_1 = B^*_1 \oplus B^*_2 \\G^*_2 = B^*_2 \oplus B^*_3 & G^*_0 = B^*_0 \oplus B^*_1\end{array}$$

Từ hệ các hàm ra này chúng ta dễ dàng lập được sơ đồ mạch thực hiện dùng các cổng cho trước.

b) Mạch chuyển mã dư 3 - Gray \Rightarrow dư 3

$$\begin{array}{ll}B^*_3 = G^*_3 & B^*_1 = G^*_1 \oplus G^*_2 \oplus G^*_3 \\B^*_2 = G^*_2 \oplus G^*_3 & B^*_0 = G^*_0 \oplus G^*_1 \oplus G^*_2 \oplus G^*_3\end{array}$$

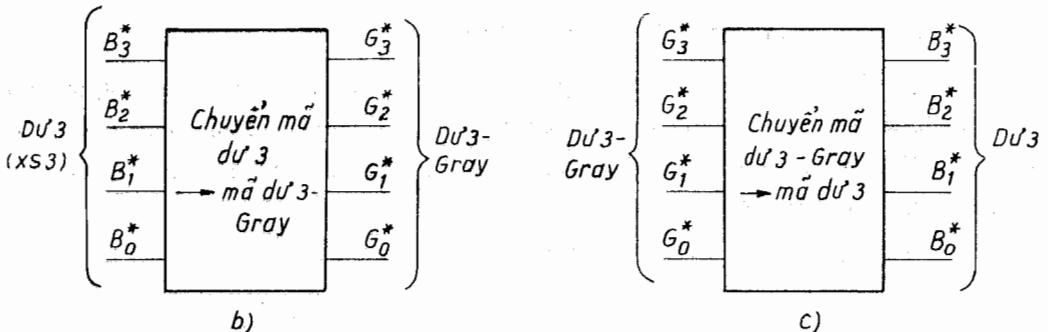
Từ các hàm ra này lập sơ đồ, mạch thực hiện dùng các cổng logic cho trước.

Tóm lại :

Việc chuyển mã nhị phân, dư 3 \Leftrightarrow Gray, dư 3 - Gray được mô tả như sơ đồ ở hình 5-19,a và 5-19,b.

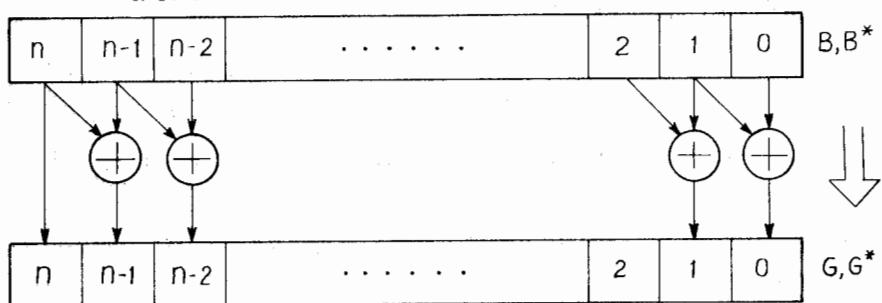
Mã dư 3				Mã dư 3 - Gray			
B_3^*	B_2^*	B_1^*	B_0^*	G_3^*	G_2^*	G_1^*	G_0^*
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1

a)

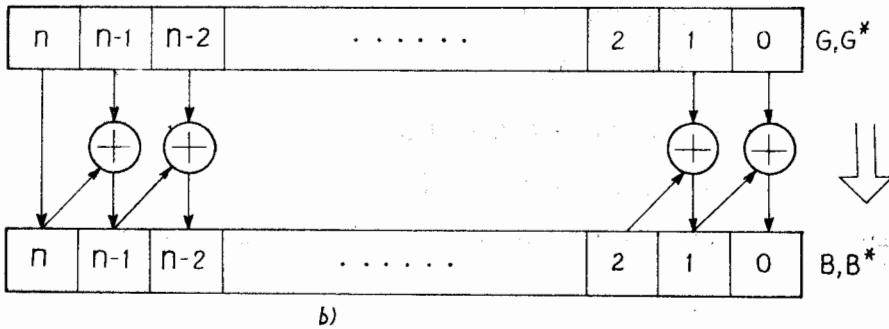


Hình 5-18. Mạch chuyển mã dư 3 \Leftrightarrow dư 3 - Gray.

- a. Bảng chúc năng ;
- b. Sơ đồ khối mạch chuyển mã dư 3 \Rightarrow dư 3 - Gray ;
- c. Sơ đồ khối mạch chuyển mã dư 3 - Gray \Rightarrow dư 3.



Hình 5-19.a



Hình 5-19. Chuyển đổi nhị phân, mã dư 3 \Leftrightarrow Gray, dư 3 - Gray.

4. Mạch chuyển mã NBCD \Rightarrow mã thập phân (NBCD to decimal Decoder)

Mã NBCD chỉ dùng 10 tổ hợp mã trong 16 tổ hợp mã nhị phân 4 bit, 6 tổ hợp mã còn lại ứng với các giá trị thập phân 10, 11, 12, 13, 14, 15 sẽ không xuất hiện ở đầu vào vì vậy ở đầu ra ta phải điền giá trị không xác định (dấu X).

Bảng chức năng của mạch cho trong hình 5-20,a, lập bảng Karnaugh cho các hàm ra $Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, Y_8, Y_9$ và tối thiểu hóa các hàm này nhận được kết quả như sau :

$Y_0 = \bar{B}_3 \bar{B}_2 \bar{B}_1 \bar{B}_0$	$Y_5 = B_2 \bar{B}_1 B_0$
$Y_1 = B_3 B_2 B_1 B_0$	$Y_6 = B_2 B_1 B_0$
$Y_2 = \bar{B}_2 B_1 \bar{B}_0$	$Y_7 = B_2 B_1 B_0$
$Y_3 = \bar{B}_2 B_1 B_0$	$Y_8 = B_3 \bar{B}_0$
$Y_4 = B_2 \bar{B}_1 \bar{B}_0$	$Y_9 = B_3 B_0$

Từ hệ các hàm đầu ra lập sơ đồ mạch thực hiện dùng các phần tử tổ hợp cho trước.

5. Mạch chuyển mã (giải mã) NBCD sang mã 7 vạch

Tên và sự bố trí các vạch sáng của đèn chỉ thị 7 vạch đã được cho ở hình 5-13,b. Bảng chức năng của mạch được biểu diễn trong hình 5-13,a.

Từ bảng chức năng lập được sơ đồ khối của mạch như hình 5-21. Lập bảng Karnaugh cho 7 hàm ra và tối thiểu hóa các các hàm này, nhận được kết quả :

$$a = C + A + \bar{A} \bar{B} \bar{C} + \bar{A} B D$$

$$b = \bar{B} + C D + \bar{C} D$$

$$c = B + C + D$$

$$d = A + C \bar{D} + \bar{B} \bar{C} \bar{D} + \bar{A} \bar{B} \bar{C} + B \bar{C} D$$

$$e = \bar{B} \bar{D} + C \bar{D}$$

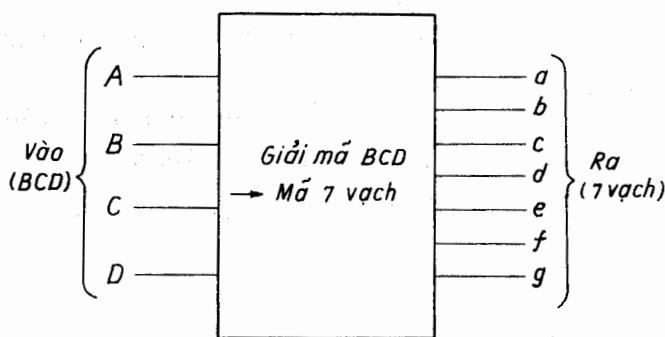
$$f = A + B \bar{C} + \bar{C} \bar{D} + B \bar{D}$$

$$g = A + B \bar{C} + C \bar{D} + \bar{B} C$$

Từ hệ các hàm ra này lập được sơ đồ thực hiện dùng các phần tử tổ hợp cho trước.

Giá trị thập phân	Nhi phân				Thập phân									
	B ₃	B ₂	B ₁	B ₀	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈	Y ₉
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	0	1	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	1	0	0	0	0
5	0	1	0	1	0	0	0	0	0	0	1	0	0	0
6	0	1	1	0	0	0	0	0	0	0	0	1	0	0
7	0	1	1	1	0	0	0	0	0	0	0	0	1	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0	1
9	1	0	0	1	0	0	0	0	0	0	0	0	0	1
10	1	0	1	0	X	X	X	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X	X	X	X

Hình 5-20. Bảng chức năng của mạch chuyển mã NBCD \Rightarrow Mã thập phân.



Hình 5-21 : Sơ đồ khối của mạch giải mã NBCD \Rightarrow 7 vạch.

5_9. MỘT SỐ VI MẠCH TỔ HỢP

Các vi mạch tổ hợp gồm từ những vi mạch cỡ nhỏ SSI đến các vi mạch cỡ vừa MSI, cỡ lớn LSI. Chúng gồm những mạch đơn giản như các cổng AND, OR, NOR, NAND, ... đến các mạch phức tạp như *MUX*, *DEMUX*, *ENCODER*, *DECODER* ... Cùng thực hiện một chức năng logic, lại có thể là những mạch khác nhau tuỳ thuộc vào mạch ra (kéo lên thụ động hay tích cực, 3 trạng thái hay hở Collector) hay các thông số kỹ thuật khác.

Phần này giới thiệu một số vi mạch tổ hợp loại TTL nhằm giúp độc giả bước đầu làm quen với cách tra cứu vi mạch.

IC mã hoá :

- | | |
|-------------|--|
| 74147/LS147 | : mã hoá ưu tiên <i>BCD</i> ; |
| 74148/LS148 | : mã hoá ưu tiên märk thái được ; |
| 74LS348 | : mã hoá ưu tiên märk thái được, đầu ra 3 trạng thái ; |
| 74184 | : chuyển đổi mã <i>BCD</i> \Rightarrow nhị phân, nhị phân \Rightarrow <i>BCD</i> ; |
| 74185 | : đổi <i>BCD</i> \Rightarrow nhị phân, nhị phân \Rightarrow <i>BCD</i> . |

IC giải mã :

- | | |
|-------------|---|
| 7441 | : giải mã <i>BCD</i> \Rightarrow thập phân, Đầu ra chịu điện áp cao (60 V); |
| 7442/LS42 | : <i>BCD</i> \Rightarrow thập phân ; |
| 7443 | : dư 3 \Rightarrow thập phân ; |
| 7444 | : giải mã Dư 3 – Gray sang thập phân ; |
| 7445 | : <i>BCD</i> \Rightarrow thập phân, dòng lớn (80 mA) ; |
| 7446 | : giải mã <i>BCD</i> \Rightarrow 7 vạch ; |
| 7447 | : đầu ra mức tích cực thấp ; |
| 7448 | : giải mã <i>BCD</i> \Rightarrow 7 vạch ; |
| 7449 | : đầu ra mức tích cực cao ; |
| 74141 | : <i>BCD</i> \Rightarrow thập phân, đầu ra chịu điện áp cao (60 V) ; |
| 74145/LS145 | : <i>BCD</i> \Rightarrow thập phân, dòng lớn 80 (mA) ; |
| 74LS138 | : giải mã giải 3 sang 8 ; |
| 74LS139 | : giải mã 4 sang 16 ; |
| 74159 | : giống 74154 nhưng có đầu ra hở (Open Collector). |
| 74155/LS155 | : giải mã 2 sang 4 . |

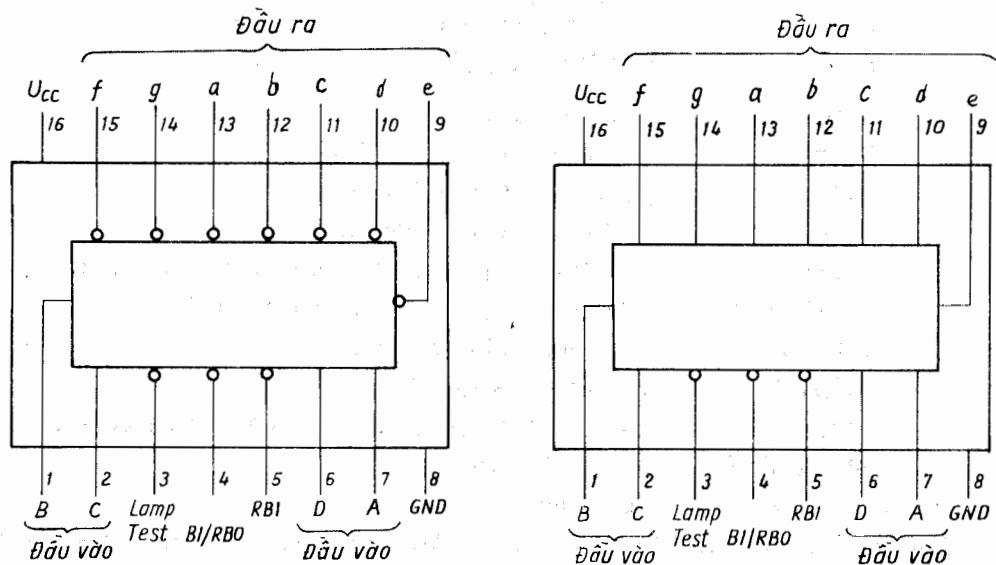
Ví dụ: Giải mã *BCD* sang 7 vạch :

Loại này gồm có các IC ký hiệu *MD54/DM7446a*, *47a*, *48*, *LS47*, *LS48*, *LS49* Các IC *46 A*, *47A* và *LS47* dùng mức tích cực thấp còn *LS48*, *48*, *LS49* sử dụng mức tích cực cao. Bảng thông số của các IC này cho ở hình 5-22.

Sơ đồ chân của IC được cho ở hình 5-23 và bảng chân lý hình 5-24.

Loại	Đầu ra				
	Mức tích cực	Trở kháng đầu ra	Dòng ra	Điện áp chịu đựng	Công suất tiêu tán
DM5446A	low	open-collector	40 mA	30 V	320 mW
DM5447A	low	open-collector	40 mA	15 V	320 mW
DM5448	high	2 kΩ pull-up	6.4 mA	5.5 V	265 mW
DM54LS47	low	open-collector	12 mA	15 V	35 mW
DM54LS48	high	2 kΩ pull-up	2 mA	5.5 V	125 mW
DM54LS49	high	open-collector	4 mA	5.5 V	40 mW
DM7446A	low	open-collector	40 mA	30 V	320 mW
DM7447A	low	open-collector	40 mA	15 V	320 mW
DM7448	high	2 kΩ pull-up	6.4 mA	5.5 V	265 mW
DM74LS47	low	open-collector	24 mA	15 V	35 mW
DM74LS48	high	2 kΩ pull-up	6 mA	5.5 V	125 mW
DM74LS49	high	open-collector	8 mA	5.5 V	40 mW

Hình 5-22. Bảng thông số của các IC giải mã $NBCD \Rightarrow 7$ vạch.



5446A/7446A (J),(N),(W);

5447A/7447A (J),(N),(W);

54LS47/74LS47 (J),(N),(W)

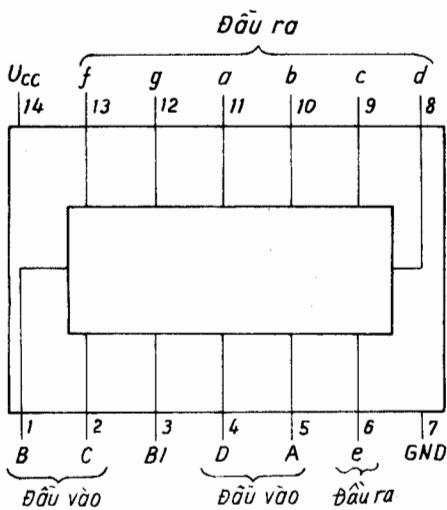
5448/7448 (J),(N),(W);

54LS48/74LS48 (J),(N),(W)

a)

b)

Hình 5-23 a, b.



54LS49/74LS49(J),(N),(W)

Hình 5-23.c. Sơ đồ chân của IC giải mã 7 vạch.

Thập phân hay chức năng	Đầu vào					Đầu ra						
	D	C	B	A	BI	a	b	c	d	e	f	g
0	L	L	L	L	H	H	H	H	H	H	H	L
1	L	L	L	L	H	L	H	H	L	L	L	L
2	L	L	H	L	H	H	H	L	H	H	L	H
3	L	L	H	H	H	H	H	H	H	L	L	H
4	L	H	L	L	H	L	H	H	L	L	H	L
5	L	H	L	H	H	H	L	H	H	L	H	H
6	L	H	H	L	H	L	L	H	H	H	H	H
7	L	H	H	H	H	H	H	H	L	L	L	L
8	H	L	L	L	H	H	H	H	H	H	H	H
9	H	L	L	H	H	H	H	H	L	L	H	H
10	H	L	H	L	H	L	L	L	H	H	L	H
11	H	L	H	H	H	L	L	H	H	L	L	H
12	H	H	L	L	H	L	H	L	L	L	H	H
13	H	H	L	H	H	H	L	L	H	L	H	H
14	H	H	H	L	H	L	L	L	H	H	H	H
15	H	H	H	H	H	L	L	L	L	L	L	L
Bl	X	X	X	X	L	L	L	L	L	L	L	L

Hình 5-24. Bảng chân lý của bộ giải mã 7 vạch (đầu ra tích cực ở mức cao).

BÀI TẬP

5.1. Hãy viết dạng đại số đơn giản nhất cho các hàm số sau và xâ dựng mạch thực hiện hàm số đó chỉ dùng các cổng *NAND* :

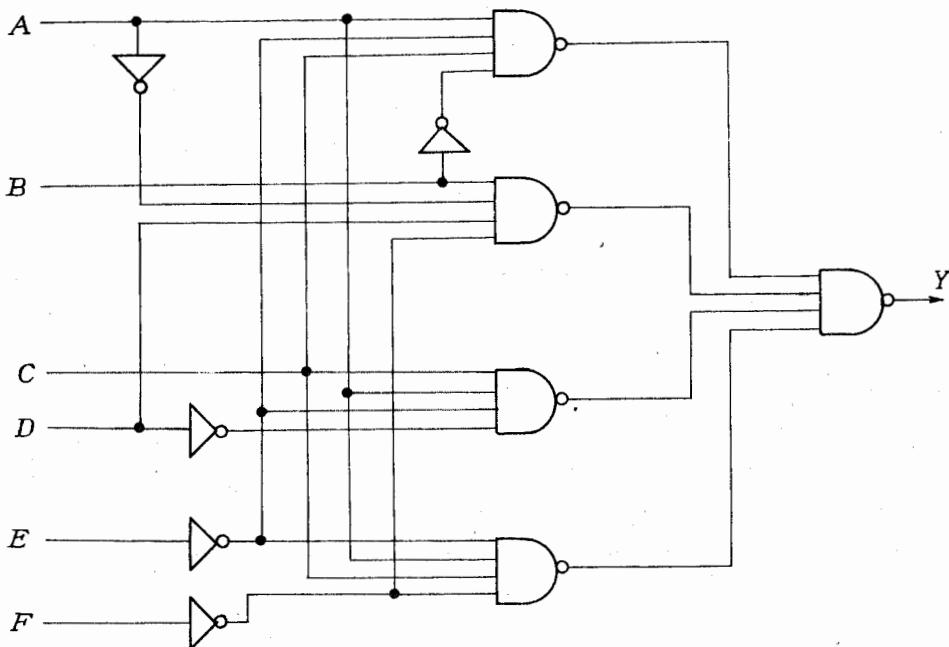
$$Y_a(A, B, C) = \sum (0, 2, 3, 4, 6)$$

$$Y_b(A, B, C) = \Pi(0, 1, 4, 5, 6)$$

(Với A, B, C là các biến – các đầu vào của mạch, C là bít trễ nhất).

5.2. Đơn giản mạch logic trên hình 5-25.

Vẽ lại mạch sau khi đã đơn giản hoá.



Hình 5-25.

5.3. Xác định mạch tổ hợp mô tả hàm sau sao cho khi xây dựng mạch, số linh kiện phải sử dụng là ít nhất (mạch đơn giản nhất) :

$$\begin{aligned} Y = & \overline{A} \overline{B} \overline{C} D + A B \overline{C} D + A \overline{B} \overline{C} \overline{D} + \overline{A} \overline{B} \overline{C} \overline{D} + \overline{A} B C D \\ & + A B C \overline{D} + \overline{A} B \overline{C} \overline{D} + A \overline{B} C D, \end{aligned}$$

Vẽ mạch tổ hợp đó.

5.4. Thực hiện hàm sau bằng những loại mạch khác nhau :

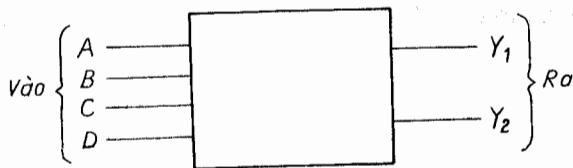
$$Y(A, B, C, D) = \sum (1, 3, 4, 5, 6, 7, 12, 13)$$

với qui ước D là bit trễ nhất.

5.5. Xây dựng mạch tổ hợp 4 đầu vào và 2 đầu ra chỉ dùng mạch *NAND* với 2 hàm ra Y_1, Y_2 :

$$Y_1 = \sum (3, 4, 5, 7, 13, 15)$$

$$Y_2 = \sum (3, 8, 9, 11, 13, 15)$$



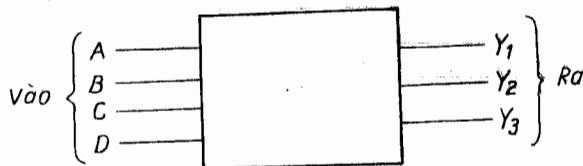
Hình 5-26. Cho bài tập 5.5.

5.6. Xây dựng mạch tổ hợp 4 đầu vào và 3 đầu ra Y_1, Y_2, Y_3 :

$$Y_1 (A, B, C, D) = \sum (10, 12, 14, 15) \quad \text{với } n = 7, 8.$$

$$Y_2 (A, B, C, D) = \sum (0, 7, 12) \quad \text{với } n = 2, 14, 15.$$

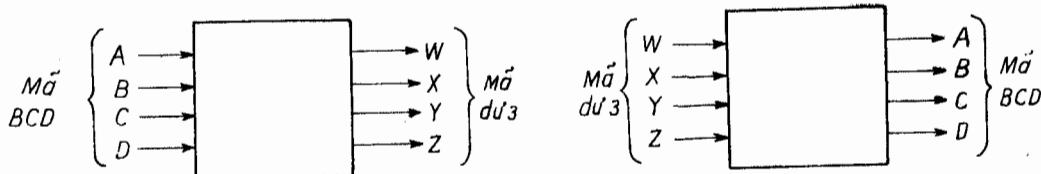
$$Y_3 (A, B, C, D) = \sum (2, 8, 14) \quad \text{với } n = 0, 10, 12.$$



Hình 5-27. Cho bài tập 5.6.

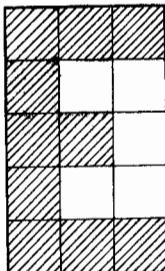
D là bit trê nhất và n là tập các đỉnh không xác định.

5.7. Thiết kế mạch tổ hợp biến đổi mã $NBCD$ sang mã dư 3 ($XS3$) và ngược lại (qui ước D và Z là các bit trê nhất).



Hình 5-28. Cho bài tập 5.7.

5.8. Thiết kế mạch điều khiển một hệ thống đèn đặt theo kiểu ma trận 3×5 . Với các lệnh điều khiển khác nhau, mạch sẽ hiển thị các chữ in hoa : C, E, F, P, T trên hình 5-29 là hiển thị chữ E.



Hình 5-29. Cho bài tập 5.8.

5.9. Thiết kế mạch tổ hợp làm nhiệm vụ tạo ra bit chẵn cho một từ mã 5 bits, một mạch khác kiểm tra tính chẵn của từ mã 5 bits đã được bổ xung thêm 1 bit chẵn.

5.10. Thiết kế mạch tổ hợp biến đổi mã *BCD* sang mã Hamming và mạch tổ hợp phát hiện sai và sửa sai mã Hamming 7/4.

Phần 3

MẠCH DÂY

CHƯƠNG 6

CÁC PHẦN TỬ NHỚ CƠ BẢN

6-1. GIỚI THIỆU

Như đã trình bày ở trên, mạch số được chia làm 2 loại :

1) *Mạch tổ hợp (Combinational Circuits)* là mạch tín hiệu chỉ phụ thuộc vào tín hiệu vào. Các phần tử cơ bản để xây dựng nên hệ tổ hợp là các mạch AND, OR, NAND, NOR, ...

2) *Mạch dây - Mạch tuần tự (Sequential circuits)* là mạch tín hiệu ra phụ thuộc không những vào tín hiệu vào mà còn phụ thuộc vào trạng thái trong của mạch, nghĩa là mạch có lưu trữ, nhớ các trạng thái. Để xây dựng mạch dây ngoài, các mạch tổ hợp cơ bản như AND, OR, NAND, NOR ... còn cần phải có các phần tử nhớ.

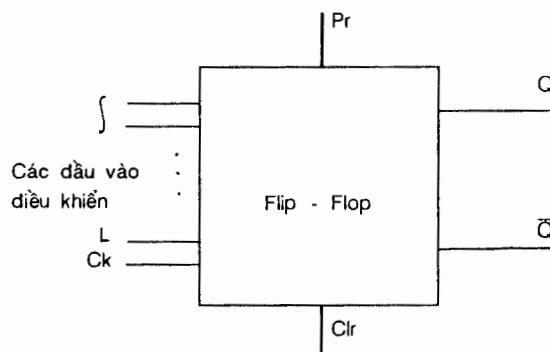
Chương này sẽ nghiên cứu các phần tử nhớ cơ bản để tạo thành mạch dây. Các mạch này được gọi là các Flip Flop (thường được viết tắt là FF), chúng là các phần tử nhớ đơn bit vì có khả năng nhớ được một chữ số nhị phân.

6-2. ĐỊNH NGHĨA VÀ PHÂN LOẠI

6-2.1. ĐỊNH NGHĨA

Flip - Flop (FF) là phần tử có khả năng lưu trữ (nhớ) 1 trong 2 trạng thái 0 hay 1.

FF có từ 1 đến một vài đầu vào điều khiển, có 2 đầu ra luôn luôn ngược nhau là Q và \bar{Q} . Tuỳ từng loại FF do chế tạo có thể còn có đầu vào xoá (thiết lập "0" - Clear), đầu vào thiết lập "1" (Preset). Ngoài ra FF còn thường hay có đầu vào đồng bộ (Clock). Sơ đồ khái tổng quát của một FF được cho ở hình 6-1.



Hình 6-1. Sơ đồ tổng quát cho một Flip - Flop.

Các ký hiệu về tính tích cực được chỉ ra trong hình 6-2.

Ký hiệu	Tính tích cực của tín hiệu
	Tích cực là mức thấp "L"
	Tích cực là mức cao "H"
	Tích cực là sườn dương của xung nhịp
	Tích cực là sườn âm của xung nhịp

Hình 6-2. Bảng ký hiệu về tính tích cực.

Ví dụ : Flip - Flop cho trong hình 6-3 là Flip - Flop loại JK, có đầu vào Pr (thiết lập "1"), tích cực ở "H", Clear (thiết lập ở "0"), tích cực ở "L", Ck tích cực ở sườn dương.

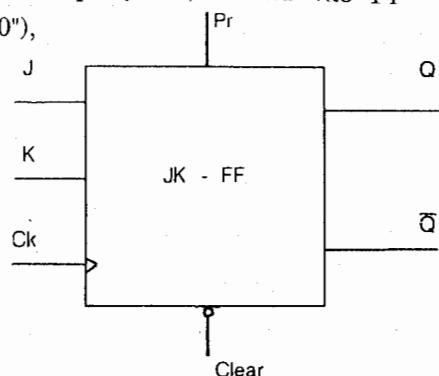
6.2.2. PHÂN LOẠI

Có nhiều cách phân loại FF.

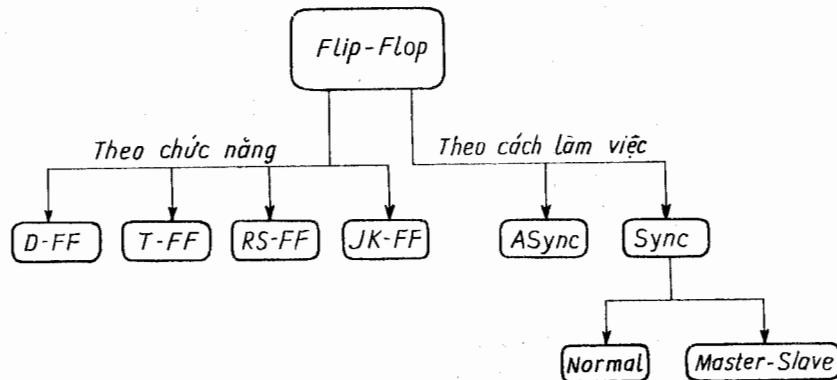
- Theo chức năng làm việc của các đầu vào điều khiển : hiện nay thường sử dụng loại FF một đầu vào D - FF, T - FF và loại FF hai đầu vào RS - FF và JK - FF, ngoài ra đôi khi còn có thể gặp loại FF nhiều đầu vào.

- Theo cách làm việc ta có loại FF không đồng bộ và đồng bộ. Đối với loại không đồng bộ, các tín hiệu điều khiển vẫn điều khiển được hoạt động của FF đúng không cần tín hiệu đồng bộ. Ngược lại, ở loại FF đồng bộ các tín hiệu điều khiển chỉ điều khiển được hoạt động của FF khi và chỉ khi có tín hiệu đồng bộ và tín hiệu này tích cực. Loại đồng bộ này lại được chia làm loại đồng bộ thường và loại đồng bộ chủ - tớ (Master - Slave).

Sơ đồ khái của sự phân loại FF được cho ở hình 6-4.



Hình 6-3. FF loại JK.



Hình 6-4. Cách phân loại Flip - Flop.

6.2.3. BIỂU DIỄN FLIP - FLOP

Để mô tả một *FF* người ta có thể dùng :

- Bảng chân lý (Bảng chức năng) ;
- Đồ hình chuyển đổi trạng thái ;
- Phương trình đặc trưng.

Các phương pháp biểu diễn này sẽ được trình bày chi tiết cho từng loại *FF* trong các phần tiếp theo.

6.3. CÁC LOẠI FLIPFLOP (FF) VÀ ĐIỀU KIỆN ĐỒNG BỘ

Theo chức năng có 4 loại *FF* cơ bản : *D*, *T*, *RS* và *JK*. Hình 6-5 là bảng chân lý của các loại *FF* với : hình 6-5,a cho *D* - *FF*; hình 6-5,b cho *T* - *FF*, hình 6-5,c cho *RS* - *FF* và hình 6-5,d cho *JK* - *FF*.

<i>D</i>	<i>Q</i>	<i>Q'</i>
0	0	0
0	1	0
1	0	1
1	1	1

a)

<i>T</i>	<i>Q</i>	<i>Q'</i>
0	0	0
0	1	1
1	0	1
1	1	0

b)

<i>R</i>	<i>S</i>	<i>Q</i>	<i>Q'</i>
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	cấm
1	1	1	cấm

c)

<i>J</i>	<i>K</i>	<i>Q</i>	<i>Q'</i>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

d)

Hình 6-5. Bảng chức năng của các *FF*.

- a. Bảng chân lý của *D* - *FF* ; b. Bảng chân lý của *T* - *FF* ;
- c. Bảng chân lý của *RS* - *FF* ; d. Bảng chân lý của *JK* - *FF*.

Các FF đều có thể xây dựng từ các mạch tổ hợp có hồi tiếp. Ta biết rằng mạch có hồi tiếp chỉ có thể làm việc tin cậy khi điều kiện sau đây được thỏa mãn :

Mạch không rơi vào trạng thái dao động dưới tác động của bất kỳ tập tín hiệu điều khiển vào nào. Điều này có nghĩa là, ứng với mỗi tổ hợp tín hiệu vào bất kỳ phải tồn tại ít nhất một trạng thái ổn định. Trạng thái ổn định là trạng thái thỏa mãn điều kiện $Q' = Q$.

Trên cơ sở các bảng chức năng của các FF ở hình 6-5, biểu diễn bảng trạng thái của các FF trên hình 6-6. Các bảng này biểu diễn trạng thái chuyển biến tới Q' với trạng thái ban đầu Q và các tín hiệu vào điều khiển. Khoanh tròn là những ô có $Q' = Q$, những ô đó tương ứng với trạng thái ổn định.

Q'	RS	00	01	11	10
Q		(0)	1	X	(0)
0		(1)	(1)	X	0
1					

(a)

Q'	JK	00	01	11	10
Q		(0)	(0)	1	1
0		(1)	0	0	(1)
1					

(b)

Q'	D	00	01
Q		0	1
0		(0)	(0)
1			

(c)

Q'	T	00	01
Q		(0)	1
0		(1)	0
1			

(d)

Hình 6-6. Bảng Karnaugh biểu diễn chức năng của các FF.

a. RS - FF; b. JK - FF ; c. D - FF; d. T - FF.

Từ bảng trạng thái trên hình 6-6 rút ra nhận xét rằng :

- Các $D - FF$ và $RS - FF$ có thể làm việc ở chế độ không đồng bộ vì với mỗi tập tín hiệu vào điều khiển D, RS luôn luôn tồn tại ít nhất 1 trong các trạng thái ổn định. Bởi vì tất cả các cột tín hiệu vào của bảng trạng thái ứng với hai loại FF này, mỗi cột đều có ít nhất 1 ô được khoanh tròn.

- Các $T - FF$ và $JK - FF$ không thể làm việc ở chế độ không đồng bộ vì mạch sẽ rơi vào trạng thái dao động nếu như tập tín hiệu vào là 1 1 với $JK - FF$ hoặc là 1 với $T - FF$. Điều này tương ứng là tại cột $T = 1$ đối với $T - FF$, $JK = 1 1$ với $JK - FF$ không có ô nào được khoanh tròn.

Như vậy các $D - FF$, $RS - FF$ có thể làm việc cả ở 2 chế độ : đồng bộ và không đồng bộ còn các $T - FF$ và $JK - FF$ chỉ có thể làm việc ở chế độ đồng bộ .

6_4. RS - FF

RS - FF là một *FF* có 2 đầu vào điều khiển *R,S*. *S* là đầu vào đặt (thiết lập "1" – Set) còn *R* là đầu vào xoá (Reset).

Sơ đồ khối của *RS - FF* không biểu diễn các đầu vào đồng bộ (*Ck*), xoá (Clear) và thiết lập (Preset) được cho ở hình 6-7,a. Bảng trạng thái của *RS - FF* được trình bày trong hình 6-7,b. Ba cột đầu *S, R* và *Q* là giá trị của các tín hiệu vào điều khiển (*S, R*) và trạng thái của *FF* (*Q*) tại thời điểm *t*. Các giá trị ở các hàng của 3 cột này biểu diễn tất cả các tổ hợp giá trị có thể có được của *S, R* và *Q*. Cột thứ tư là trạng thái sẽ chuyển biến tới (ký hiệu là *Q'*) của *FF* sau thời gian quá độ là $\delta't$: *Q'* là trạng thái ra của *FF* ở thời điểm tiếp theo $t + \delta't$.

Rút gọn bảng 6-7,b, nhận xét rằng :

- Khi $S = R = 0$ *FF* giữ nguyên trạng thái cũ $Q' = Q$.
- Khi $S = 0, R = 1$, đầu vào xoá (Reset) có tín hiệu, *FF* sẽ chuyển đến trạng thái 0 nghĩa là $Q' = 0$ dù rằng trước đó *FF* ở trạng thái 0 hay 1.
- Khi $S = 1, R = 0$, đầu vào thiết lập "1" (Set) có tín hiệu, *FF* sẽ chuyển đến trạng thái 1 nghĩa là $Q' = 1$.
- Tổ hợp tín hiệu vào $R = S = 1$ là tổ hợp tín hiệu vào cấm của *RS - FF*. Khi đó *FF* nhận được đồng thời 2 tín hiệu điều khiển thiết lập "1" ($S = 1$) và xoá ($R = 1$) trạng thái của *FF* sẽ không xác định.

Bảng trạng thái rút gọn của *RS - FF* được biểu diễn trên hình 6.6.c. Đồ hình chuyển đổi trạng thái của *RS - FF* được biểu diễn ở hình 6-7,d.

Từ bảng chức năng trên hình 6-7,b nhận thấy rằng *RS - FF* chỉ thay đổi trạng thái từ $1 \Rightarrow 0$, nghĩa là *FF* được tắt (ở dòng 4) ký hiệu Toff và từ $0 \Rightarrow 1$, nghĩa là *FF* được bật ký hiệu là Ton với phương trình :

$$\text{Ton} (Q : 0 \Rightarrow 1) = S \bar{R} \bar{Q}$$

$$\text{Toff} (Q : 1 \Rightarrow 0) = \bar{S} R Q .$$

Nếu xét đến điều kiện cấm của tổ hợp tín hiệu vào khi $R = S = 1$ hay điều kiện cho phép của tín hiệu vào $RS = 0$ thì nhận được :

$$\text{Ton} = S \bar{Q}$$

$$\text{Toff} = R Q .$$

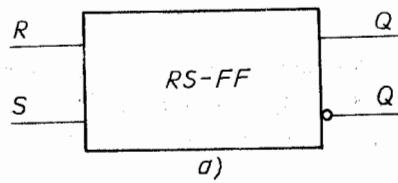
Phương trình đặc trưng của *RS - FF* được biểu diễn ở dạng :

$$Q' = f(R, S, Q)$$

Từ bảng chức năng trên hình 6-6,b, viết biểu thức ở dạng CTT và tối thiểu hóa ta được phương trình đặc trưng của *RS - FF* :

$$Q' = S + \bar{R} Q .$$

Từ phương trình đặc trưng có thể xây dựng sơ đồ thực hiện *RS - FF* chỉ dùng NAND (6-7,e) và chỉ dùng NOR (6-7,f).



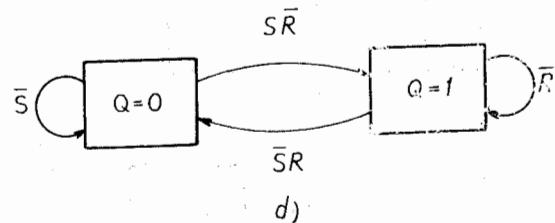
a)

Dòng	S	R	Q	Q'
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	X
8	1	1	1	X

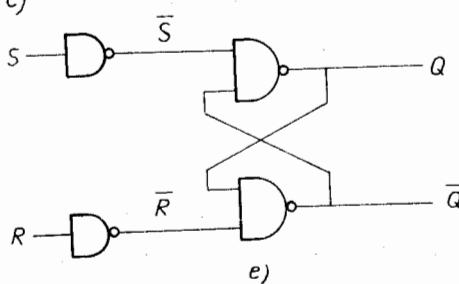
b)

R	S	Q'
0	0	Q
0	1	0
1	0	1
1	1	X

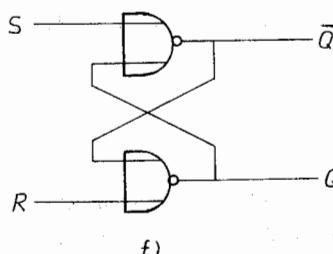
c)



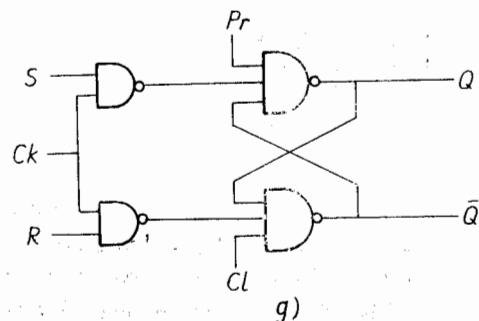
d)



e)



f)



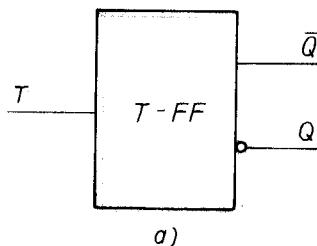
g)

Hình 6-7. RS - FF;

- a. Sơ đồ khối của RS - FF; b. Bảng chân lý của RS - FF; c. Bảng chức năng ngắn gọn của hàm RS - FF; d. Đồ hình trạng thái của RS - FF; e. Sơ đồ RS - FF không đồng bộ dùng NAND; f. Sơ đồ RS - FF không đồng bộ dùng NOR; g. Sơ đồ RS - FF đồng bộ dùng NAND (có xung nhịp, có Preset và Clear).

6_5. T - FF

T - FF là loại FF có một đầu vào điều khiển T . Số đồ khối của nó được biểu diễn trên hình 6-8,a. Bảng chân lý của T - FF được cho ở hình 6-8,b. Từ bảng này có thể đơn giản để nhận được bảng chức năng rút gọn 6-8,c và đồ hình trạng thái 6-8,d.



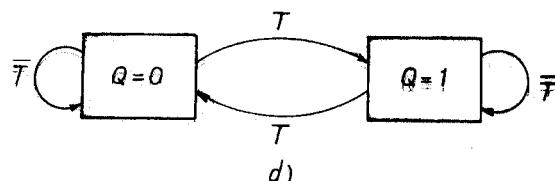
a)

T	Q	Q'
0	0	0
0	1	1
1	0	1
1	1	0

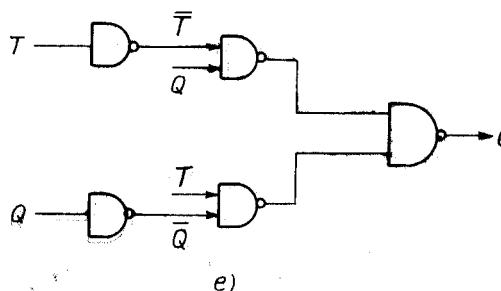
b)

T	Q'
0	Q
1	\bar{Q}

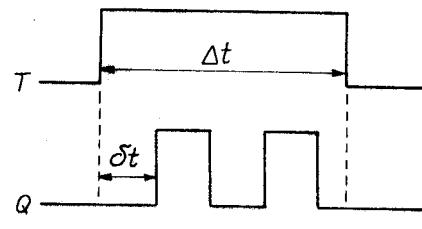
c)



d)



e)



f)

Hình 6-8. T - FF

- a. Sơ đồ khối của T - FF;
- b. Bảng chân lý của T - FF;
- c. Bảng chức năng rút gọn của hàm T - FF;
- d. Đồ hình trạng thái của T - FF;
- e. Sơ đồ T - FF không đồng bộ dùng NAND;
- f. Giản đồ dạng sóng của T - FF không đồng bộ.

Nhận thấy rằng khi $T = 0$, FF giữ nguyên trạng thái cũ, $Q' = Q$. Còn khi $T = 1$, FF lật trạng thái (Toggle). Cũng chính vì thế mà FF này có tên là T - FF.

Phương trình đặc trưng $T - FF$ được biểu diễn ở dạng :

$$Q' = \bar{T} Q + T \bar{Q} = T \oplus Q.$$

Sơ đồ thực hiện $T - FF$ không đồng bộ dùng NAND dựa trên phương trình đặc trưng của FF được cho ở hình 6-8,e.

Giả thiết rằng tín hiệu vào T tồn tại trong thời gian Δt và δt là thời gian trễ của mạch. Giả sử ban đầu của FF ở trạng thái 0, thì sau δt , FF sẽ lật lên trạng thái 1. Nếu tín hiệu vào T vẫn chưa kết thúc (tức là $\Delta t > \delta t$) thì mạch lại sẽ tiếp tục lật về trạng thái 0.

Quá trình cứ tiếp tục như vậy cho đến khi đầu vào T trở về 0. Mạch ở trạng thái dao động. Khi kết thúc xung điều khiển T , ta không biết đích xác mạch ở trạng thái nào. **Gần** đó dạng sóng của mạch được vẽ ở hình 6-8,f. Trong thực tế $\Delta t >> \delta t$ do vậy **mạch luôn luôn** ở trạng thái dao động khi $T = 1$. $T - FF$ là loại FF chỉ có thể làm việc ở chế độ đồng bộ.

Mạch điện vẽ ở hình 6-8,f trong thực tế $\Delta t >> \delta t$. Do vậy **mạch luôn luôn** ở trạng thái dao động khi $T = 1$. $T - FF$ là loại FF chỉ có thể làm việc ở chế độ đồng bộ.

6_6. JK - FF

$JK - FF$ là loại FF có 2 đầu vào điều khiển J, K . Sơ đồ khối của $JK - FF$ được biểu diễn trên hình 6-9,a bằng chân lý của $JK - FF$ cho ở hình 6-9,b và từ đó suy ra bảng chức năng rút gọn (hình 6-9,c) và đồ họa trạng thái (hình 6-9,d).

Từ bảng chân lý rút gọn 6-9,c nhận thấy rằng khi :

$$JK = 0\ 0, FF \text{ giữ nguyên trạng thái cũ} \quad (Q' = Q).$$

$$JK = 0\ 1, FF \text{ luôn luôn chuyển đến trạng thái } 0 \quad (Q' = 0).$$

$$JK = 1\ 0, FF \text{ luôn luôn chuyển đến trạng thái } 1 \quad (Q' = 1).$$

$$JK = 1\ 1, FF \text{ luôn luôn lật trạng thái} \quad (Q' = \bar{Q}).$$

Nếu so sánh với chức năng của $RS - FF$ ta nhận thấy có sự tương ứng giữa 2 FF này cụ thể K tương ứng với R , J tương ứng với S . Tuy nhiên $JK - FF$ khác so với $RS - FF$ ở chỗ không có tổ hợp tín hiệu vào cấm cụ thể khi $JK = 11$, FF luôn luôn lật trạng thái trong khi đối với $RS - FF$: $SR = 11$ và tổ hợp tín hiệu vào cấm.

Phương trình đặc trưng của $JK - FF$ có dạng :

$$Q' = J \bar{Q} + \bar{K} Q.$$

Viết lại phương trình này dưới dạng chỉ dùng các hàm NAND :

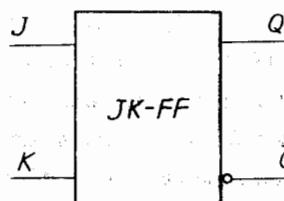
$$Q' = \overline{\overline{J} \overline{Q}} \cdot \overline{\overline{K} Q} = \overline{\overline{J} \overline{Q}} \cdot \overline{\overline{K} Q}$$

$$\bar{Q}' = \overline{J \bar{Q}} + K Q = \overline{\overline{J} \overline{Q}} \cdot \overline{K Q}.$$

Từ đó xây dựng được sơ đồ cho $JK - FF$ như hình 6-9,e. Như đã xét ở mục 6-3, $JK - FF$ cũng giống như $T - FF$ chỉ có thể làm việc ở chế độ đồng bộ. Thật vậy khi $J, K = 1\ 1$ mạch ở trạng thái dao động tương tự như $T - FF$ khi $T = 1$. Sơ đồ đầy đủ của $JK - FF$ đồng bộ với các đầu vào xoá, thiết lập (Clear, Preset) được biểu diễn ở hình 6-8,f.

Đối với mạch cho ở hình 6-9,f các tín hiệu điều khiển J, K chỉ tác động đến mạch khi Ck ở mức logic cao "H", 2 tín hiệu xoá (Clear) và thiết lập (Preset) là những tín hiệu không

đồng bộ, mức tích cực thấp "L". Sự hồi tiếp từ đầu ra về đầu vào sẽ làm cho mạch dao động khi xung đồng bộ Ck và các tín hiệu điều khiển J, K ở mức cao "H". Hiện tượng này được gọi là hiện tượng đua vòng quanh (race around) và có thể gây nên chuyển biến sai nhầm của mạch.

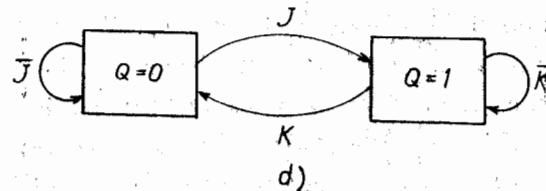


J	K	Q'
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

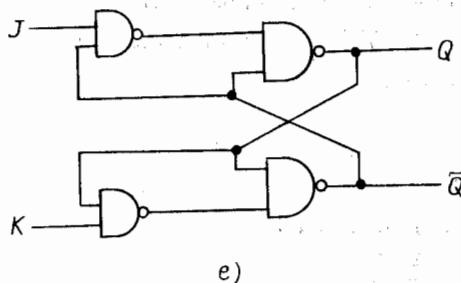
c)

Dòng	J	K	Q	Q'
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

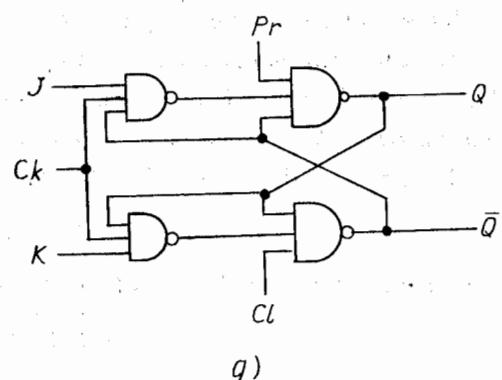
b)



d)



e)

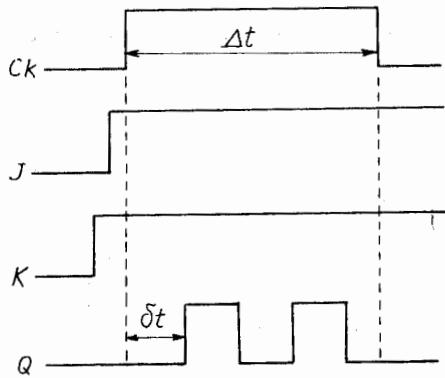


g)

Hình 6-9. JK - FF.

- a. Sơ đồ khối của JK - FF ; b. Bảng chức năng của JK - FF ; c. Bảng chức năng rút gọn của hàm JK - FF ; d. Đồ hình trạng thái của JK - FF ; e. Sơ đồ JK - FF không đồng bộ dùng NAND ; f. Sơ đồ JK-FF đồng bộ dùng NAND.

Nếu gọi Δt là khoảng thời gian tồn tại xung Ck , δt là thời gian quá độ của mạch, giản đồ dạng sóng miêu tả hiện tượng đua vòng quanh khi $Ck = "H"$, $J = K = "H"$ cho ở hình 6-10.



Hình 6-10. Hiện tượng đua vòng quanh (race around) của JK - FF.

Để khắc phục hiện tượng này dùng JK - FF loại chủ - tớ (Master – Slave). Sơ đồ khối và sơ đồ chi tiết của JK - FF loại chủ - tớ được biểu diễn ở hình 6-11.

JK - FF loại chủ tớ (Master-Slave) gồm 2 FF mắc liên tiếp FF – Master và FF – Slave, đầu vào của FF – Master là đầu vào của mạch, còn đầu ra của FF – Slave là đầu ra của mạch.

Trong hình 6-11, FF Master làm việc khi xung đồng bộ Ck ở mức cao "H", còn FF Slave làm việc khi Ck ở mức thấp "L". Khi xung nhịp ở "H" FF Master sẽ hoạt động theo sự điều khiển của tín hiệu điều khiển ở đầu vào, FF Slave không hoạt động. Khi xung nhịp ở mức "L" ngược lại FF Master không hoạt động còn FF Slave sẽ thay đổi trạng thái theo sự điều khiển của tín hiệu vào của nó cũng chính là đầu ra của FF Master. Như vậy khi xung nhịp ở "H" thông tin ở đầu vào được ghi nhận vào FF Master, còn khi xung nhịp "L" thông tin được đưa ra ở đầu ra của FF Slave cũng chính là đầu ra của mạch. Chính vì cấu tạo như vậy nên FF loại chủ tớ khắc phục được hiện tượng đua vòng quanh. Giản đồ dạng sóng của mạch điện được biểu diễn ở hình 6-11,c.

6_7. D - FF

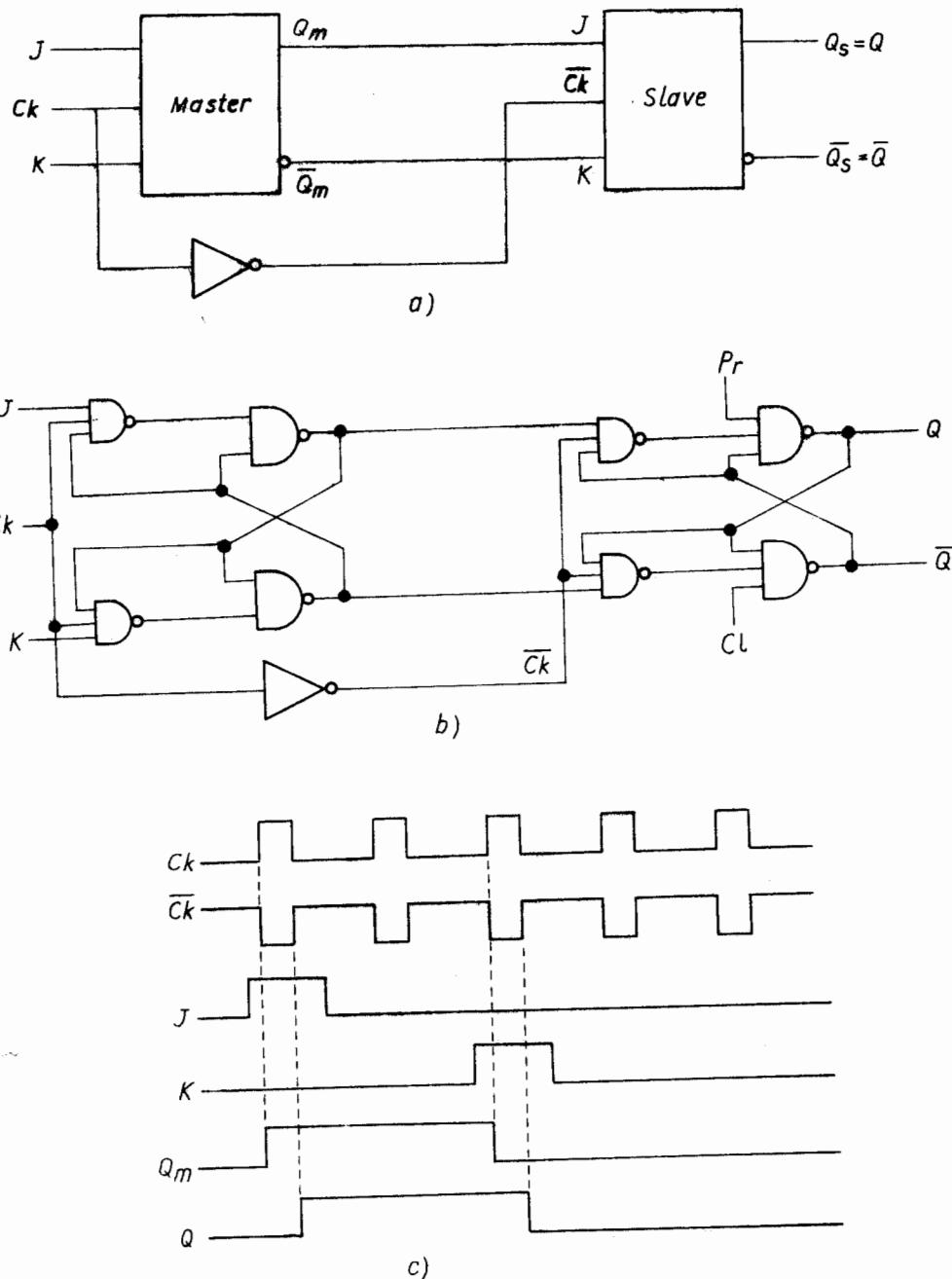
D - FF là loại FF có một đầu vào điều khiển D . Sơ đồ khối, bảng chân lý, bảng chân lý đơn giản, đồ hình trạng thái của loại FF này được biểu diễn trên hình 6-12a, 6-12b, 6-12c và 6-12d.

Phương trình đặc trưng của D - FF có dạng :

$$Q' = D$$

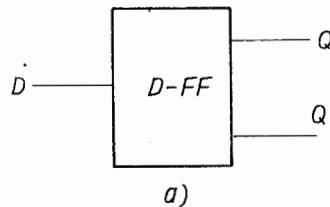
Nếu gọi δt là thời gian quá độ của mạch, thì D - FF là khâu trễ có thời gian trễ là δt . Đầu ra Q chính là sự trễ (Delay) của đầu vào một khoảng thời gian δt . Chính vì vậy mà FF này có tên là D - FF (D là viết tắt của Delay – trễ).

Như đã trình bày ở mục 6-3, D -FF có thể làm việc ở các chế độ đồng bộ và không đồng bộ.



Hình 6-11. JK -FF loại chủ tớ.

- a. Sơ đồ khối ; b. Sơ đồ JK -FF loại chủ tớ dùng NAND ; c. Giản đồ dạng sóng của JK -FF loại chủ tớ.



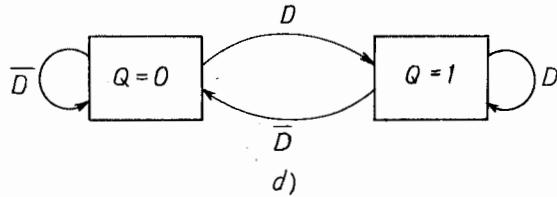
a)

D	Q	Q'
0	0	0
0	1	0
1	0	1
1	1	1

b)

D	Q'
0	0
1	1

c)



Hình 6-12. D - FF.

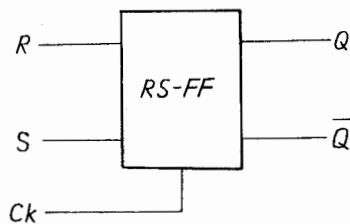
- a. Sơ đồ khối của D - FF ; b. Bảng chúc năng của D - FF ; c. Bảng chúc năng rút gọn của D - FF ;d. Đồ hình trạng thái của D - FF.

6_8. FF LÀM VIỆC NHƯ MỘT MẠCH CHỐT

Các FF có thể làm việc như một mạch chốt (Latch).

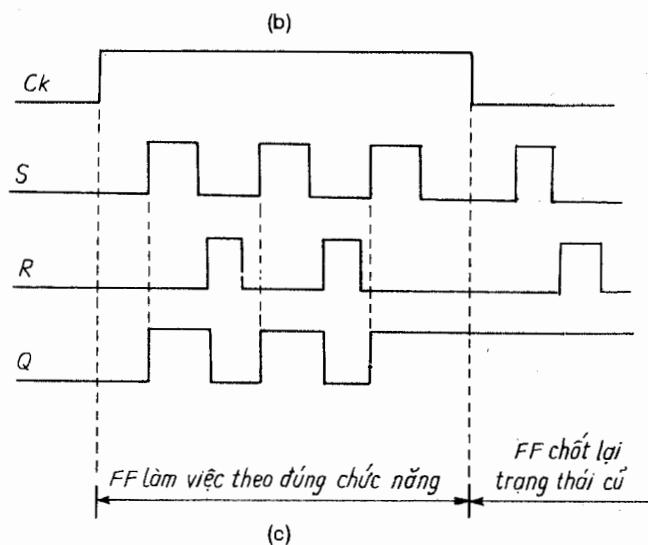
Trên hình 6-13,a là SR - FF đồng bộ. Khi Ck ở mức "H", đầu ra Q của FF có giá trị theo đúng bảng chân lý của FF. Khi Ck ở mức "L", đầu ra Q của FF giữ không đổi nghĩa là trạng thái của FF bị chốt lại (xem bảng chúc năng hình 6-13,b).

Tương tự giản đồ thời gian cho D - FF làm việc như một mạch chốt được biểu diễn ở hình 6-14.



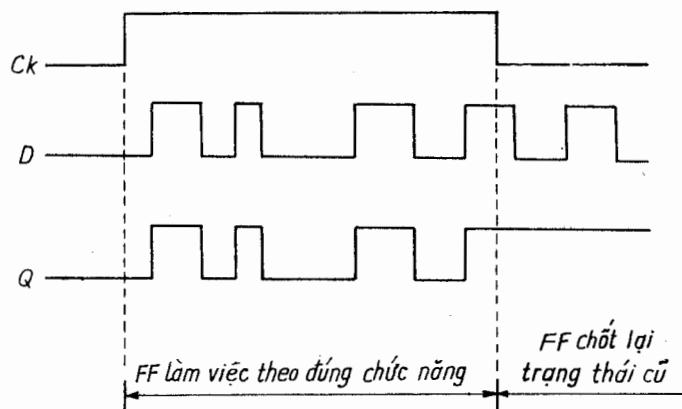
Hình 6-13.a.

Ck	S	R	Q'	Giải thích
0	0	0	Q	$Q = Q' \text{ FF bị chốt}$
0	0	1	Q	
0	1	0	Q	
1	0	0	Q	FF hoạt động theo đúng chức năng
1	0	1	0	
1	1	0	1	



Hình 6-13. SR - FF làm việc như một mạch chốt.

- a. Sơ đồ khối của SR - FF với Ck ; b. Bảng chức năng của SR - FF với Ck ;
- c. Dạng sóng của SR - FF.



Hình 6-14. Dạng sóng của D - FF làm việc như một mạch chốt.

6-9. XÁC ĐỊNH ĐẦU VÀO KÍCH (ĐẦU VÀO ĐIỀU KHIỂN) CHO FF

Trong nhiều trường hợp, đặc biệt khi thiết kế mạch dùng FF cần phải xác định đầu vào kích (đầu vào điều khiển) của FF ứng với sự chuyển đổi trạng thái cho trước $Q \Rightarrow Q'$.

Với mỗi FF, sự chuyển biến trạng thái $Q \Rightarrow Q'$ chỉ xảy ra trong 4 khả năng $0 \Rightarrow 0, 0 \Rightarrow 1, 1 \Rightarrow 0$ và $1 \Rightarrow 1$. Căn cứ vào chức năng của từng loại FF phải xác định giá trị đầu vào kích D, T, R, S, J, K , tương ứng với các chuyển đổi ấy.

6-9.1. ĐẦU VÀO KÍCH CHO D - FF

Do đặc điểm của $D - FF$ là tín hiệu ra ở thời điểm $t + \delta t$ chính là tín hiệu vào ở thời điểm t , nghĩa là $Q' = D$, ta nhận được bảng cho trong hình 6-15.

Hàng	Q	Q'	D
1	0	0	0
2	0	1	1
3	1	0	0
4	1	1	1

Hình 6-15. Bảng đầu vào kích cho $D - FF$.

6-9.2. ĐẦU VÀO KÍCH CHO T - FF

Như đã biết, T là viết tắt của chữ Toggle (lật). Khi $T = 0$, FF giữ nguyên trạng thái cũ, như vậy ứng với các trường hợp FF không thay đổi trạng thái : $0 \Rightarrow 0$ và $1 \Rightarrow 1$ thì $T = 0$ (hàng 1 và hàng 4). Ngược lại khi $T = 1$, FF sẽ lật trạng thái, như vậy với 2 trường hợp còn lại $0 \Rightarrow 1$ và $1 \Rightarrow 0$ thì $T = 1$. Hình 6-16 cho bảng đầu vào kích của $T - FF$.

Hàng	Q	Q'	T
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

Hình 6-16. Bảng đầu vào kích của $T - FF$.

6_9.3. ĐẦU VÀO KÍCH CHO RS - FF

Căn cứ theo chức năng hay đồ hình trạng thái của $RS - FF$ (hình 6-7,d) lập được bảng đầu vào kích cho $RS - FF$ hình 6-17.

- Ở hàng thứ 1 khi $Q \Rightarrow Q'$ là 0 $\Rightarrow 0$ đầu vào $K = 0$ còn R có thể lấy giá trị tùy ý 0 hay 1 ta ký hiệu là X .

- Hàng thứ 4 khi $Q \Rightarrow Q'$ là 1 $\Rightarrow 1$ đầu vào $R = 0$ còn S lấy giá trị tùy ý X .

- Khi FF thay đổi trạng thái 0 $\Rightarrow 1$ (hàng thứ 2) thì $R = 0, S = 1$.

- Cuối cùng khi FF chuyển từ 1 $\Rightarrow 0$ (hàng thứ 3) thì $R = 1$ và $S = 0$.

Hàng	Q	Q'	R	S
1	0	0	X	0
2	0	1	0	1
3	1	0	1	0
4	1	1	0	X

Hình 6-17. Bảng đầu vào kích cho $RS - FF$.

6_9.4. ĐẦU VÀO KÍCH CHO JK - FF

Tương tự như đối với $RS - FF$ với $J \approx S, K \approx R$:

- Ở hàng thứ 1 khi $Q \Rightarrow Q'$ là 0 $\Rightarrow 0, J = 0, K = X$

- Ở hàng thứ 4 khi $Q \Rightarrow Q'$ là 1 $\Rightarrow 0 J = X, K = 0$

Nhưng $JK - FF$ không có tổ hợp tín hiệu vào cấm, khi $J = K = 1$ FF lật trạng thái nên :

- Ở hàng thứ 2 khi $Q \Rightarrow Q'$ là 0 $\Rightarrow 1 J = 1 ; K = X$.

- Ở hàng thứ 3 khi $Q \Rightarrow Q'$ là 1 $\Rightarrow 0$ các đầu vào kích sẽ là $J = X$ và $K = 1$.

Bảng đầu vào kích cho $JK - FF$ được biểu diễn ở hình 6-18.

Hàng	Q	Q'	J	K
1	0	0	0	X
2	0	1	1	X
3	1	0	X	1
4	1	1	X	0

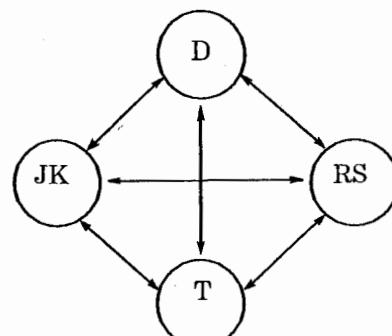
Hình 6-18. Bảng đầu vào kích cho $JK - FF$.

JK - FF khác phục được nhược điểm của *RS - FF* không có tổ hợp tín hiệu vào cấm, mặt khác *JK - FF* có số trường hợp các tín hiệu vào kích có giá trị không xác định lớn hơn so với các loại khác mà những giá trị không xác định này lại được lợi dụng trong quá trình thiết kế làm cho sơ đồ thực hiện đơn giản. Chính vì vậy *JK - FF* thường hay được sử dụng trong thực tế.

6_10. CHUYỂN ĐỔI GIỮA CÁC LOẠI FF

Có 4 loại *FF* đã giới thiệu là *D - FF*, *T - FF*, *RS - FF* và *JK - FF*. Trong thực tế có khi *FF* loại này lại được mắc mạch và được sử dụng như *FF* loại khác. Nội dung của phần này là xây dựng các *FF* yêu cầu từ các *FF* cho trước.

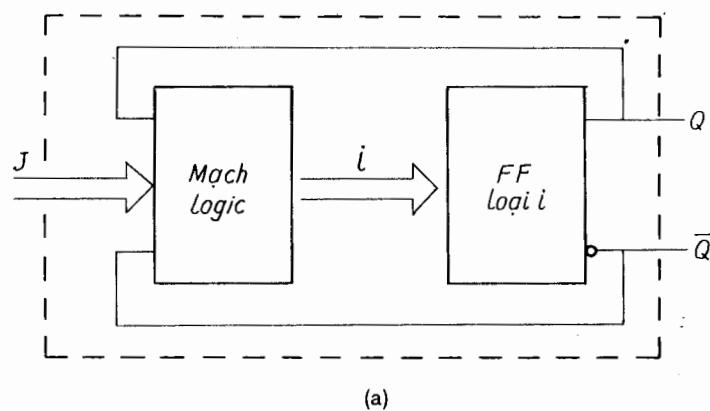
Với 4 loại *FF* có 12 khả năng chuyển đổi như trong hình 6-19.



Hình 6-19. Các khả năng chuyển đổi giữa các /

6_10.1. PHƯƠNG PHÁP CHUYỂN ĐỔI GIỮA CÁC LOẠI FF

Một trong những phương pháp để xây dựng *FF* loại *j* từ *FF* loại *i* cho trước được cho ở sơ đồ khối hình 6-20,a. (ở đây ký hiệu *i*, *j* là loại *FF*).



Q	Q'	JK	RS	D	T
0	0	0X	X0	0	0
0	1	1X	01	1	1
1	0	X1	10	0	1
1	1	X0	0X	1	0

(b)

Hình 6-20. Chuyển đổi giữa các loại *FF*.

a. Sơ đồ khối để chuyển *FF* loại *i* \Rightarrow *FF* loại *j*; b. Bảng đầu vào kích của các loại *FF*.

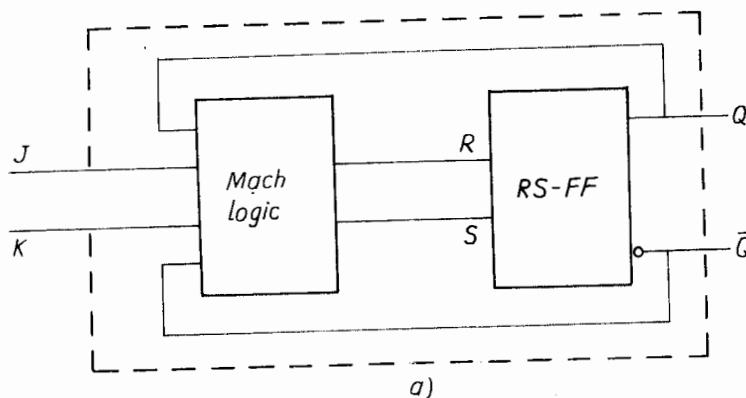
Trong sơ đồ 6-20,a các đầu vào j là các đầu vào của FF loại j cần thiết kế. Đầu ra của mạch logic là các đầu vào của FF loại i cho trước. Như vậy bài toán chuyển đổi từ FF loại i sang loại j là xây dựng mạch logic tổ hợp có các đầu vào là j và Q và các đầu ra là i biểu diễn bởi hệ hàm :

$$i = f(j, Q)$$

Để thực hiện chuyển đổi FF loại i sang FF loại j cần thực hiện các bước sau :

1. Xác định hệ hàm $i = f(j, Q)$ theo bảng các đầu vào kích của các FFs (hình 6-20,b).
2. Tối thiểu hóa các hàm này và xây dựng các sơ đồ.

Sau đây ta sẽ xét một số ví dụ xây dựng các FF từ các FF cho trước thường hay được sử dụng trong thực tế.



a)

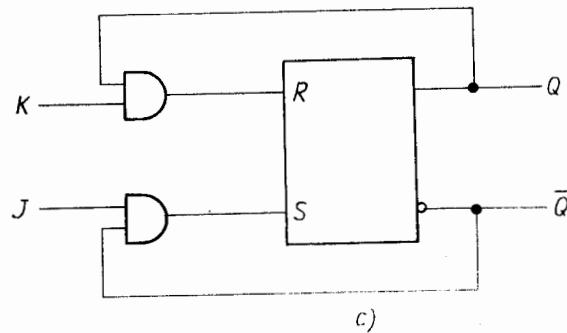
R	JK	00	01	11	10
Q	0	X	X	0	0
	1	0	(1)	(1)	0

$R = KQ$

S	JK	00	01	11	10
Q	0	0	0	(1)	(1)
	1	X	0	0	X

$S = J\bar{Q}$

b)



c)

Hình 6-21. Chuyển $RS - FF$ thành $JK - FF$.

- a. Sơ đồ khối ; b. Bảng Karnaugh của RS theo Q, J, K ; c. Sơ đồ thực hiện chuyển $RS - FF$ thành $JK - FF$.

Ví dụ :

Chuyển đổi từ $RS - FF$ thành $JK - FF$:

Sơ đồ khối của việc chuyển đổi này được cho ở hình 6-21,a. Nghĩa là cần phải thiết kế mạch logic tổ hợp của các hàm logic:

$$R = f(Q, J, K)$$

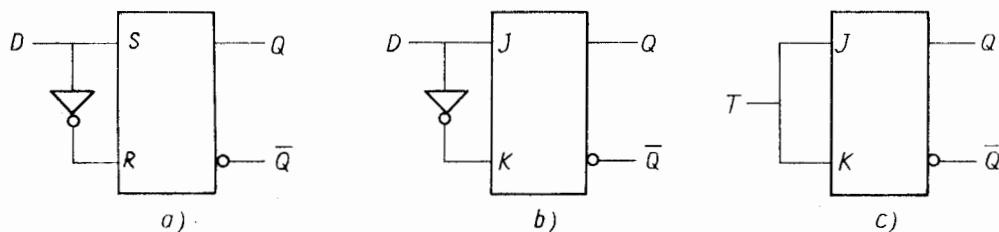
$$S = g(Q, J, K)$$

Từ bảng đầu vào kích của các FF hình 6-20,b nhận được bảng Karnaugh cho R, S với các biến vào là Q, J, K như hình 6-21,b. Tối thiểu hóa các hàm này ta có :

$$R = K \bar{Q}$$

$$S = J \bar{Q}$$

Mạch thực hiện sự chuyển đổi này được cho trong hình 6-21,c.



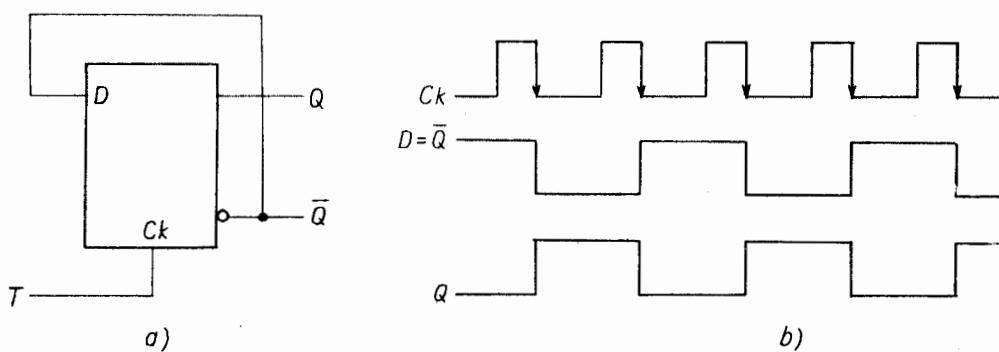
Hình 6-22. Một số sơ đồ chuyển đổi giữa các loại FF .

a. $SR - FF$ thành $D - FF$; b. $JK - FF$ thành $D - FF$; c. $JK - FF$ thành $T - FF$.

Chuyển $RS - FF$ thành $D - FF, JK - FF$ thành $T - FF$.

Với phương pháp hoàn toàn tương tự, nhận được các sơ đồ chuyển đổi giữa các loại FF khác như hình 6-22.

Chuyển đổi từ $D - FF$ thành $T - FF$. Một $D - FF$ có thể biến đổi thành $T - FF$ bằng cách nối đầu ra Q với đầu vào D . Lúc này T chính là đầu vào xung nhịp. Sơ đồ nối mạch được biểu diễn ở hình 6-23,a và dạng sóng của mạch được cho ở hình 6-23,b (với giả thiết $D - FF$ hoạt động với sườn âm của xung nhịp và trạng thái ban đầu của FF bằng 0).



Hình 6-23. $D - FF$ nối thành $T - FF$.

a. Sơ đồ mạch ; b. Dạng sóng của đầu ra.

6-11. MỘT SỐ VI MẠCH FF

1. 7470 : Flip -flop JK

- Xung nhịp Ck tích cực (tác động tại sườn dương);
- Đầu vào thiết lập, xoá (PR , CLS) tích cực ở mức thấp;
- Cố cồng AND ở đầu vào.

2. 7472 Flip -flop chủ tá

- Xung nhịp Ck tích cực ở mức cao;
- Cố cồng AND ở đầu vào;
- Đầu vào thiết lập, xoá tích cực ở mức cao.

3. 7473 và 74 LS 73 : Hai JK -FF

- Đầu vào xoá tích cực ở mức thấp;
- Xung nhịp Ck :
 - + 7473 : Tích cực ở mức cao;
 - + 74LS73 : Tích cực ở sườn âm.

4. 7476 và 74LS76 : Hai JK -FF

- Đầu vào thiết lập, xoá tích cực ở mức thấp;
- Xung nhịp Ck :
 - + 7476 : Tích cực ở mức cao;
 - + 74LS76 : Tích cực ở sườn âm.

5. 74107 và 74LS107 : Hai JK -FF

- Đầu vào xoá tích cực ở mức thấp;
- Xung nhịp Ck :
 - + 74107 : Tích cực ở mức cao;
 - + 74LS107 : Tích cực ở sườn âm.

6. 74LS112 : Hai JK -FF

- Xung nhịp Ck tích cực ở sườn âm;
- Đầu vào thiết lập, xoá tích cực ở mức thấp.

BÀI TẬP

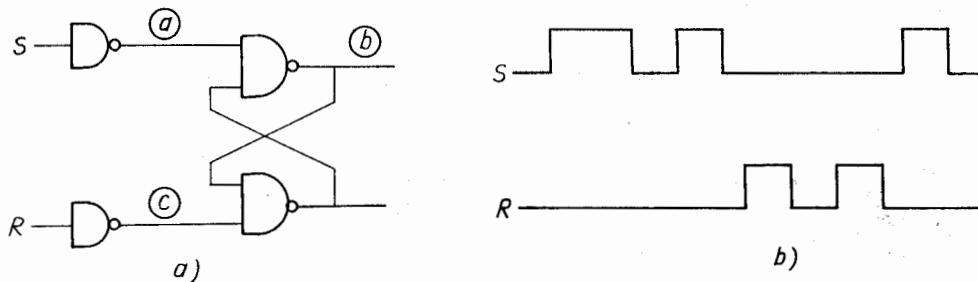
6.1. Cho các FF cơ bản loại : RS , JK , D , T và chốt D . Hãy xác định xem trong số các loại FF trên loại nào có thể thực hiện được mà không cần tín hiệu đồng bộ.

6.2. Viết các hàm logic của các FF loại RS , RS ngược, chốt D . Trên cơ sở đó xây dựng sơ đồ logic thực hiện các FF này dùng chỉ mạch $NAND$ hoặc NOR .

6.3. Một SR - FF cấu trúc từ các cồng $NAND$ cho ở hình (a). Hãy xác định các mức logic ở điểm a, b, c với các điều kiện sau :

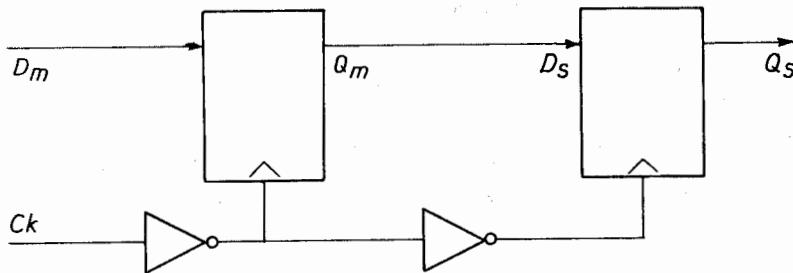
- a) $S = 0, R = 0$, và $Q = 0$;
- b) Giống (a) nhưng S đổi từ $0 \Rightarrow 1$;
- c) $S = 0, R = 0$ và R đổi từ $0 \Rightarrow 1$;

Dạng sóng của S và R cho ở hình 6-24, hãy vẽ dạng sóng tương ứng của Q với giả thiết giá trị ban đầu của $Q = 0$.



Hình 6-24. Cho bài tập 6.3.

6.4. Một $D - FF$ lật sườn gồm 2 chốt D như ở hình 6-25. Với đồ thị thời gian chứng tỏ rằng FF này nhận biết được dữ liệu vào tại sườn dương và cho ra tín hiệu tương ứng tại sườn âm của xung nhịp.



Hình 6-25. Cho bài tập 6.4.

6.5. Hãy vẽ đồ hình trạng thái ngoài cho FF có phương trình đặc tính là :

- a) $Q' = X \oplus Y \oplus Q$
- b) $Q' = X \approx Y \approx Q$

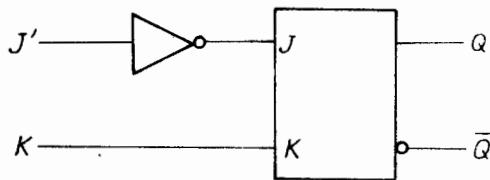
6.6. Với mạch logic ngoài chứng tỏ rằng một $D - FF$ lật sườn có thể chuyển thành một $JK - FF$ và chứng minh mạch cho ra tín hiệu chỉ phụ thuộc vào dữ liệu vào hiện có trên sườn dương của xung nhịp.

6.7. Một $JK - FF$ được biến đổi như ở hình 6-26 thành dạng $J'K - FF$. Hãy vẽ bảng trạng thái cho FF này và rút ra phương trình đặc tính của nó.

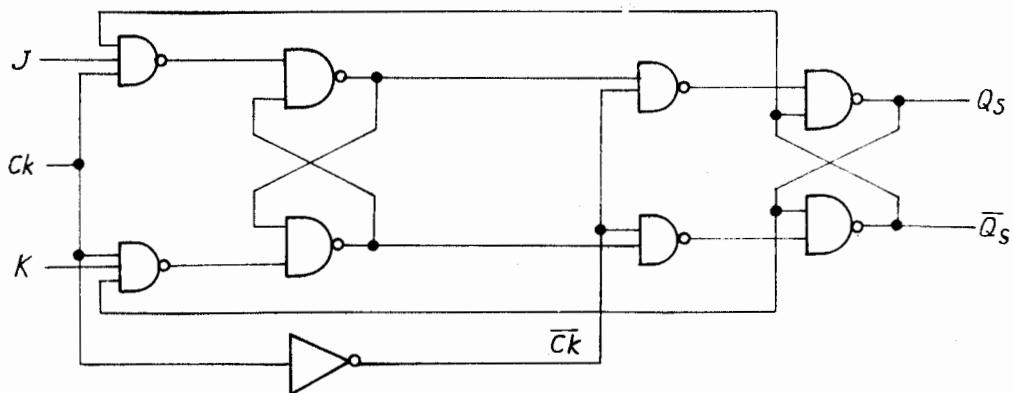
6.8. Một $JK - FF$ Master/Slave cho ở hình 6-27.

Giả thiết điều kiện đầu của FF là : $J = K = Q_m = Q_s = 0$.

Hãy vẽ các mức logic cho sự thay đổi sau (các thay đổi ở J và K xảy ra trong khoảng giữa các xung nhịp) :



Hình 6-26. Cho bài tập 6.7.



Hình 6-27. Cho bài tập 6.8.

- (I) $J : 0 \Rightarrow 1, \quad K : 0 \Rightarrow 0$, Xung nhịp 1
- (II) $J : 1 \Rightarrow 1, \quad K : 0 \Rightarrow 1$, Xung nhịp 2
- (II) $J : 1 \Rightarrow 0, \quad K : 1 \Rightarrow 0$, Xung nhịp 3
- (IV) $J : 0 \Rightarrow 1, \quad K : 0 \Rightarrow 0$, Xung nhịp 4

Hãy vẽ đồ thị thời gian biểu diễn dạng sóng của J, K, Q_m và Q_s cho một chu kỳ 4 xung nhịp.

Giả thiết với cùng các điều kiện đầu như trên, hãy xác định điều kiện cuối cùng của Q_s khi các đầu vào thay đổi theo thứ tự sau :

- (V) $Ck : 0 \Rightarrow 1, \quad J : 0 \Rightarrow 1, \quad Ck : 1 \Rightarrow 0$
- (IV) $J : 0 \Rightarrow 1, \quad Ck : 0 \Rightarrow 1, \quad K : 0 \Rightarrow 1, J : 1 \Rightarrow 0$

CHƯƠNG 7

MÔ TẢ VÀ THIẾT KẾ MẠCH DÃY

Trong chương này chúng ta sẽ xét những khái niệm cơ bản nhất về mạch dãy, các phương pháp mô tả mạch dãy, phân loại mạch dãy, các phương pháp thiết kế mạch dãy và lấy đó làm cơ sở lý thuyết cho các chương sau.

7_1. MỘT SỐ KHÁI NIỆM CƠ BẢN VỀ MẠCH DÃY

7_1_1. MÔ HÌNH TOÁN HỌC CỦA MẠCH DÃY

Otomat chính là mô hình toán học của mạch dãy. Otomat (A) là bộ sau đây :

$$A = (V, R, S, \delta, \lambda),$$

trong đó :

V là tập các tín hiệu vào : $\{V_1, V_2, \dots, V_n\} V_i \in V$

R là tập các tín hiệu ra : $\{R_1, R_2, \dots, R_m\} R_j \in R$

S là tập các trạng thái trong : $\{S_1, S_2, \dots, S_k\} S_h \in S$

δ là hàm chuyển biến trạng thái hay là ánh xạ $S \times V \Rightarrow S'$

λ là hàm ra hay là ánh xạ $S \times V \Rightarrow R$.

Những điều này được mô tả ở hình 7-1.

Tổng quát ta có :

Hàm ra : $R(t) = \lambda [V(t), S(t)]$ (7-1)

Hàm chuyển biến trạng thái : $S(t + \Delta t) = \delta [V(t), S(t)]$.

Từ nay về sau để cho gọn, ta ký hiệu :

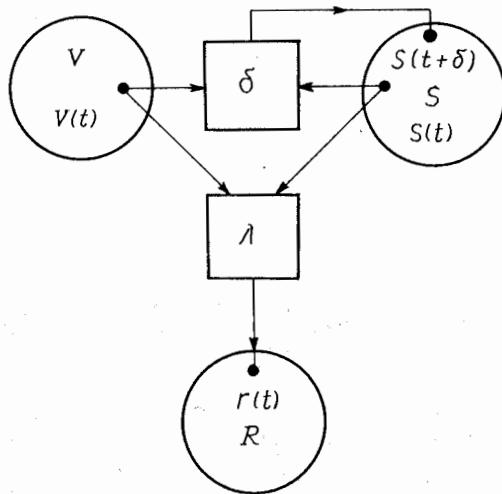
$$R(t) = R, S(t) = S, V(t) = V \text{ và } S(t + \Delta t) = S'.$$

Khi đó hệ (7-1) có thể viết như sau :

$$R = \lambda(V, S) \quad (7-2)$$

$$S' = \delta(V, S)$$

Mạch tổ hợp là một trường hợp riêng của mạch dãy khi số trạng thái trong (hay lực lượng của tập S) là 1.



Hình 7-1. Hàm chuyển biến trạng thái δ và hàm ra λ của Otomat.

7_1.2. MỘT SỐ KHÁI NIỆM

Otomat hữu hạn :

Otomat (A) được gọi là otomat hữu hạn nếu như các tập V , R , S hay ít nhất là tập S có lực lượng hữu hạn.

Otomat xác định :

Nếu các hàm δ , λ là các hàm xác định (đơn trị) thì A được gọi là otomat xác định.

Hình 7-2,a minh họa cho hàm δ của otomat xác định.

Nếu δ , λ là các hàm xác định trên toàn bộ miền $V \times S$ thì A là otomat xác định đầy đủ, ngược lại nếu δ , λ chỉ xác định trong 1 phần của miền $V \times S$ thì A được gọi là otomat xác định không đầy đủ.

Otomat không xác định :

Nếu các hàm δ , λ là các hàm không xác định (không đơn trị) thì A được gọi là otomat không xác định.

Hình 7-2,b minh họa cho hàm δ của Otomat không xác định.

Otomat xác suất :

Nếu δ , λ là các hàm chuyển biến trạng thái, hàm ra với xác suất cho trước thì A được gọi là otomat xác suất.

Hàm δ của otomat xác suất được mô tả ở hình 7-2,c.

Otomat mờ (Fuzzy Otomat) :

Nếu δ , λ không biết giá trị cụ thể mà chỉ biết một cách định tính thì otomat A được gọi là otomat mờ .

Hàm δ của otomat mờ được minh họa ở hình 7-2,d.

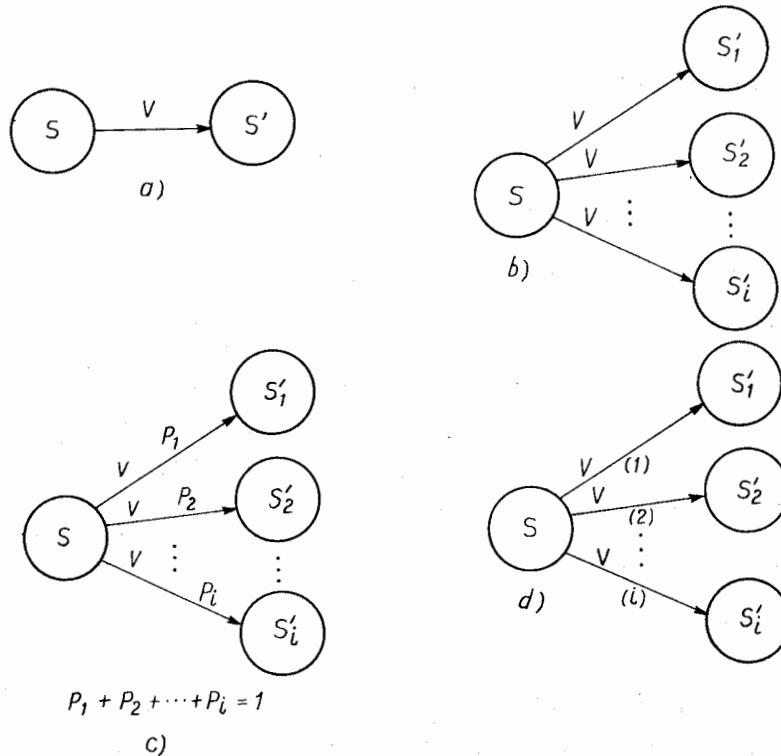
Otomat Mealy và Moore :

Otomat Mealy là otomat có hệ phương trình chuyển biến trạng thái và tín hiệu ra như ở hệ (7-2), tức là :

$$R = \lambda(V, S)$$

$$S' = \partial(V, S),$$

ở otomat loại này tín hiệu ra phụ thuộc cả vào tín hiệu vào lẫn trạng thái trong mạch



Hình 7-2. Các loại otomat.

- a. Otomat xác định ; b. Otomat không xác định ; c. Otomat xác suất ; d. Otomat mờ : (1)- Rất hay xảy ra, (2)- Hay xảy ra, (3)- ít xảy ra ; e. Otomat Moore.

Otomat Moore là otomat có hệ phương trình chuyển biến trạng thái và tín hiệu ra như sau :

$$R = \lambda (S) \quad (7-3)$$

$$S' = \delta (V, S)$$

Nghĩa là tín hiệu ra chỉ phụ thuộc vào trạng thái trong của mạch, hay nói cách khác ánh xạ λ được định nghĩa như sau :

$$D_\lambda \xrightarrow{\lambda} R \text{ ở đây } D_\lambda \subseteq S$$

Otomat Moore được minh họa ở hình 7-2,e.

Mạch dây đồng bộ :

Là mạch dây được điều khiển theo tín hiệu đồng bộ ở bên ngoài (xung nhịp Ck). Sự chuyển trạng thái của mạch từ trạng thái S_i đến trạng thái S_j chỉ xảy ra khi có xung nhịp tác động. Khoảng thời gian giữa 2 xung nhịp liên tiếp phải đủ lớn để hệ luôn luôn ở trạng thái ổn định trước khi có tác động bên ngoài.

Mạch dây không đồng bộ :

Là mạch dây mà sự hoạt động của mạch không có tín hiệu đồng bộ. Sự thay đổi của tín hiệu điều khiển là ngẫu nhiên. Đối với hệ này quá trình chuyển từ trạng thái vững S_i đến trạng thái vững S_j có thể lướt qua một số các trạng thái không vững trung gian.

Những mạch dây đồng bộ và không đồng bộ sẽ được xét chi tiết trong các chương 10 và 11.

7_2. CÁC PHƯƠNG PHÁP MÔ TẢ MẠCH DÂY

Thiết bị được thiết kế phải được mô tả bằng lời hay một số hình thức khác. Công việc đầu tiên của người thiết kế là phải phiên dịch các dữ kiện đó thành 1 hình thức mô tả hoạt động của thiết bị cần phải thiết kế một cách trung thực và duy nhất. Hay nói cách khác là phải hình thức hoá dữ liệu ban đầu.

Có 2 cách hình thức hoá thường được dùng, đó là đồ hình trạng thái và bảng.

7_2.1. BẢNG

1. Bảng chuyển đổi trạng thái

Các hàng của bảng này ghi các trạng thái trong, các cột ghi các tín hiệu vào. Giao điểm của hàng và cột là những ô, trong những ô này ghi trạng thái trong tiếp theo mà mạch sẽ chuyển đến ứng với tín hiệu vào và trạng thái hiện tại như đã ghi trên cột và hàng tương ứng. Bảng chuyển đổi trạng thái được mô tả như hình 7-3,a.

Tín hiệu vào

S'

	V	V_1	V_2	...	V_n
Trạng thái trong	S				
	S_1				
	S_2				
	.				
	S_{n-1}				
	S_n				

Trạng thái sẽ chuyển biến tới S'

a)

Tín hiệu vào

R

	V	V_1	V_2	...	V_n
Trạng thái trong	S				
	S_1				
	S_2				
	.				
	S_{n-1}				
	S_n				

Tín hiệu ra

b)

Tín hiệu vào

S/R

	V	V_1	V_2	...	V_n
Trạng thái trong	S				
	S_1				
	S_2				
	.				
	S_{n-1}				
	S_n				

Trạng thái sẽ chuyển biến tới và tín hiệu ra

c)

Hình 7-3. Mô tả hoạt động của thiết bị bảng bǎng.

a. Bảng chuyển đổi trạng thái; b. Bảng tín hiệu ra; c. Bảng chuyển đổi trạng thái / ra.

2. Bảng tín hiệu ra

Các hàng của bảng ghi các trạng thái trong, các cột ghi các tín hiệu vào. Giao điểm của hàng và cột là ô và trong ô này được ghi tín hiệu ra tương ứng. Bảng tín hiệu ra được mô tả như hình 7-3,b.

Có thể gộp 2 bảng chuyển đổi trạng thái và bảng tín hiệu ra thành một bảng chung gọi là bảng chuyển đổi trạng thái / ra (nhiều khi gọi tắt là bảng). Lúc đó trên các ô của bảng ta ghi trạng thái mà mạch sẽ chuyển biến đến và tín hiệu ra (S'/R) tương ứng với trạng thái hiện tại và tín hiệu vào.

Bảng chuyển đổi trạng thái và tín hiệu ra được mô tả ở hình 7-3,c.

Ví dụ : Một mạch dãy có :

Tập các tín hiệu vào $V = \{\alpha, \beta, \nu\}$;

Tập các tín hiệu trong $S = \{S_1, S_2, S_3, S_4, S_5\}$;

Tập các tín hiệu ra $R = \{0, 1\}$;

Chức năng của mạch được xác định bằng bảng chuyển đổi trạng thái và bảng ra như ở hình 7-4,a và 7-4,b .

S'

S \ V	α	β	γ
S_1	S_2	S_4	S_3
S_2	S_4	S_2	S_4
S_3	S_1	S_1	S_1
S_4	S_3	S_4	S_2
S_5	S_5	S_3	S_4

R

S \ V	α	β	γ
S_1	1	0	0
S_2	1	0	1
S_3	1	1	1
S_4	1	0	0
S_5	0	0	0

(a)

S/R

S \ V	α	β	γ
S_1	$S_2/1$	$S_4/0$	$S_3/0$
S_2	$S_4/1$	$S_2/0$	$S_4/1$
S_3	$S_1/1$	$S_1/1$	$S_1/1$
S_4	$S_3/1$	$S_4/0$	$S_2/0$
S_5	$S_5/0$	$S_3/0$	$S_4/0$

(b)

(c)

Hình 7-4. Mô tả mạch dãy bằng bảng .

a. Bảng chuyển đổi trạng thái ; b. Bảng tín hiệu ra ; c. Bảng chuyển đổi trạng thái và tín hiệu ra.

Cũng có thể gộp hai bảng trên thành 1 bảng chuyển đổi trạng thái / tín hiệu ra như ở hình 7-4,c.

7.2.2. ĐỒ HÌNH TRẠNG THÁI

Đồ hình trạng thái là 1 đồ hình có hướng gồm 2 tập :

M – Tập các đỉnh và K – Tập các cung có hướng .

a) Đối với mô hình Mealy thực hiện ánh xạ

Tập các trạng thái trong – Tập các đỉnh M ;

Tập các tín hiệu vào / ra – Tập các cung K .

Trên cung có hướng đi từ trạng thái trong S_i đến trạng thái trong S_j ghi tín hiệu vào / ra tương ứng.

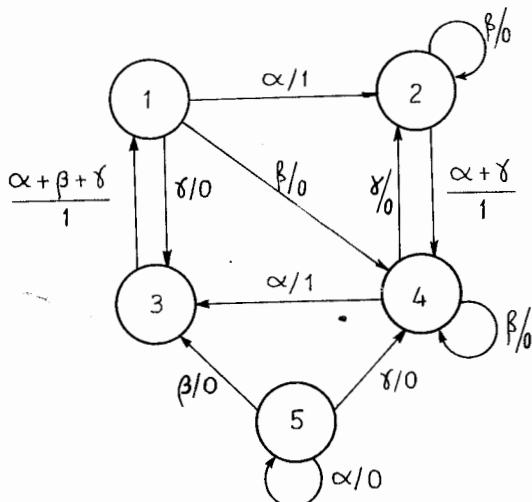
Như ví dụ của mạch dây có bảng chuyển đổi trạng thái, bảng ra ở hình 7-4 có đồ hình trạng thái cho mô hình Mealy như hình 7-5.

b) Đối với mô hình Moore

Vì tín hiệu ra chỉ phụ thuộc vào trạng thái trong của mạch, mà không phụ thuộc vào tín hiệu vào cho nên thực hiện ánh xạ :

Tập các trạng thái trong, tín hiệu ra – Tập các đỉnh M ;

Tập các tín hiệu vào – Tập các cung K .



Hình 7-5. Đồ hình trạng thái của mạch dây có bảng chuyển đổi trạng thái, bảng ra cho ở hình 7-4.

7.3. CHUYỂN ĐỔI GIỮA 2 MÔ HÌNH MEALY VÀ MOORE

Mealy và Moore là 2 mô hình toán học của mạch dây, hay nói cách khác là 2 cách biểu diễn khác nhau của mạch dây. Giữa 2 cách biểu diễn này là ánh xạ 1 – 1 và luôn luôn tồn tại thuật toán để chuyển từ mô hình này sang mô hình kia và ngược lại.

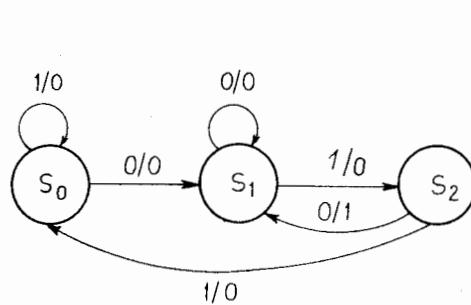
7.3.1. CHUYỂN TỪ MÔ HÌNH MEALY SANG MÔ HÌNH MOORE

Quá trình thực hiện chuyển đổi theo các bước sau :

Bước 1 : Ứng với mỗi cặp (S'/ R) của mô hình Mealy ta quy định một trạng thái tương ứng q của mô hình Moore.

Bước 2 : Thành lập bảng chuyển đổi trạng thái cho mô hình Moore : ghi mỗi trạng thái q (tương ứng với cặp s', r của mô hình Mealy) một tín hiệu ra R tương ứng.

Ví dụ : Mạch dãy được mô tả bởi mô hình Mealy có đồ hình trạng thái, bảng chuyển đổi trạng thái, tín hiệu ra như hình 7-6.



(a)

S/R

S	V	0	1
S_0	$S_1 / 0$	$S_0 / 0$	
S_1	$S_1 / 0$		$S_2 / 0$
S_2	$S_1 / 1$		$S_0 / 0$

(b)

Hình 7-6. Mô tả mạch dãy theo mô hình Mealy.

a. Đồ hình trạng thái ; b. Bảng chuyển đổi trạng thái / tín hiệu ra.

Bước 1 : Từ bảng chuyển đổi trạng thái / tín hiệu ra hình 7-6,b ta gán các trạng thái trong Q_j của mô hình Moore như sau :

$$S_1 / 0 : Q_0 ; S_0 / 0 : Q_1 ; S_2 / 0 : Q_2 \text{ và } S_1 / 1 : Q_3$$

Bước 2 : Thành lập bảng chuyển đổi trạng thái cho mô hình Moore :

Gán tín hiệu ra R cho mỗi trạng thái Q . Tín hiệu ra R chính là R của cặp (S, R) của mô hình Mealy. Các bước này được mô tả ở hình 7-7.

Mealy	Moore	
	Trạng thái	tín hiệu ra
$S_1 / 0$	Q_0	0
$S_0 / 0$	Q_1	0
$S_2 / 0$	Q_2	0
$S_1 / 1$	Q_3	1

Hình 7-7. Gán trạng thái, tín hiệu ra cho mô hình Moore.

Bảng chuyển đổi trạng thái / tín hiệu ra và đồ hình cho mô hình Moore được mô tả ở hình 7-8.

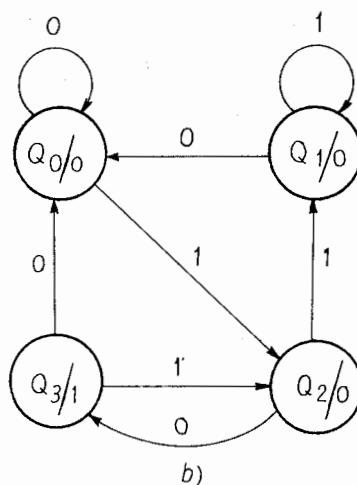
7.3.2. CHUYỂN TỪ MÔ HÌNH MOORE SANG MÔ HÌNH MEALY

Việc chuyển từ mô hình Moore sang mô hình Mealy đơn giản hơn so với việc chuyển từ mô hình Mealy sang mô hình Moore. Bởi vì, chỉ cần ghi thêm bên cạnh mỗi ô trong bảng chuyển đổi của mô hình Moore tín hiệu ra tương ứng. Sau đó tiến hành tối thiểu hóa các

trạng thái của mô hình Mealy nhận được. Vấn đề tối thiểu hóa trạng thái sẽ được xem xét một cách chi tiết trong chương 10.

Mealy	Moore			Tín hiệu ra
	Trạng thái hiện tại	Trạng thái sẽ chuyển biến đến $V_{\text{out}} = 0$	$V_{\text{out}} = 1$	
$S_1 / 0$	Q_0	Q_0	Q_2	0
$S_0 / 0$	Q_1	Q_0	Q_1	0
$S_2 / 0$	Q_2	Q_3	Q_1	0
$S_1 / 1$	Q_3	Q_0	Q_2	1

(a)



b)

Hình 7-8. Mô hình Moore tương ứng với mô hình Mealy hình 7-7.

a. Bảng chuyển đổi trạng thái / ra ; b. Đồ họa.

Nhận xét :

Mô hình Mealy và mô hình Moore là 2 mô hình dùng để biểu diễn hệ dãy, chúng luôn luôn tương đương với nhau.

Mô hình Mealy có hàm ra phụ thuộc cả tín hiệu vào lẫn trạng thái trong mạch còn mô hình Moore có hàm ra chỉ phụ thuộc vào trạng thái trong của mạch mà thôi.

Nói chung mô hình Mealy có số trạng thái trong ít hơn so với mô hình Moore, nhưng mô hình Moore lại có hệ hàm ra đơn giản hơn mô hình Mealy.

Trong thực tế cả 2 mô hình này đều được dùng để mô tả hệ dãy. Tuỳ từng trường hợp mà sử dụng mô hình loại nào để thiết kế mạch có sơ đồ đơn giản hơn, kinh tế hơn.

7_4. CÁC BƯỚC THIẾT KẾ MẠCH DÃY

Quá trình thiết kế mạch dãy được mô tả như ở lưu đồ hình 7-9.

1. Bài toán chưa được hình thức hóa

Ở đây nhiệm vụ thiết kế được mô tả bằng ngôn ngữ hoặc bằng lưu đồ thuật toán ... Nói chung là chưa được hình thức hóa.

2. Hình thức hóa

Phiên dịch các dữ kiện đó thành 1 hình thức mô tả hoạt động của mạch bằng cách hình thức hóa dữ kiện ban đầu ở dạng bảng trạng thái, bảng ra hay đồ hình trạng thái.

Rút gọn các trạng thái của mạch để nhận được số trạng thái trong ít nhất.

Chú ý rằng bước này tiến hành trên otomat chưa phải là nhị phân có V là tập các tín hiệu vào, R là tập các tín hiệu ra, S là tập các trạng thái trong.

3. Otomat nhị phân

Mã hoá tín hiệu vào, ra, trạng thái trong để nhận được otomat nhị phân có X là tập các tín hiệu vào, Y là tập tín hiệu ra, Q là tập các trạng thái trong.

4. Hệ hàm của mạch

Xác định hệ phương trình của mạch và tối thiểu hóa các phương trình này.

Nếu mạch dãy phải thiết kế dùng các FF và các mạch tổ hợp thì tùy theo yêu cầu mà chúng ta viết hệ phương trình các đầu vào kích cho loại FF đó. Điều này sẽ trình bày cụ thể hơn ở các phần sau.

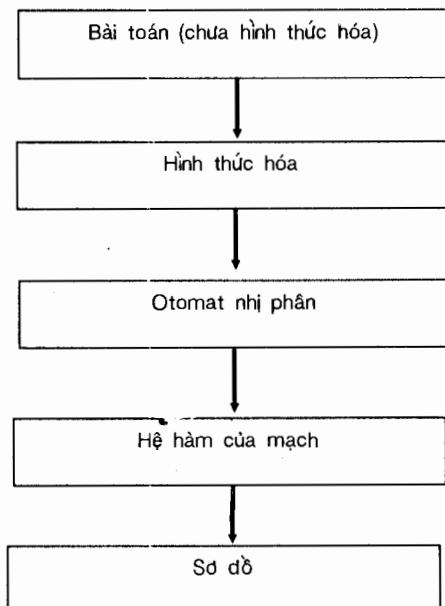
5. Xây dựng sơ đồ mạch thực hiện

Từ hệ phương trình của mạch đã viết được chúng ta xây dựng sơ đồ mạch thực hiện.

Trong nội dung của chương này sẽ giới thiệu phương pháp thiết kế mạch dãy , dùng các mạch logic tổ hợp và các FF . Dữ liệu ban đầu có thể là từ bảng trạng thái, bảng ra hoặc từ đồ hình trạng thái hoặc từ lưu đồ thuật toán (hình 7-10) ; trong hình này , con số ghi ở phần bên trái chỉ rõ mức của bước thiết kế.

Ví dụ :

Ở hàng đầu tiên : lưu đồ thuật toán, tương ứng với bước 1 của quá trình thiết kế : Bài toán chưa được hình thức hóa.



Hình 7-9. Các bước thiết kế mạch dãy.

Ở hàng thứ 3 là bước 3 của quá trình thiết kế : Otomat nhị phân với các tín hiệu vào X , tín hiệu ra Y và các trạng thái trong Q tương ứng với 2 cách biểu diễn ở dạng bảng otomat nhị phân hay ở dạng đồ hình nhị phân.

Ở hàng thứ 4 là bước 4 của quá trình thiết kế : Hệ hàm của mạch tương ứng với hệ hàm ra, hệ hàm đầu vào kích cho các FF loại D , T , RS và JK .

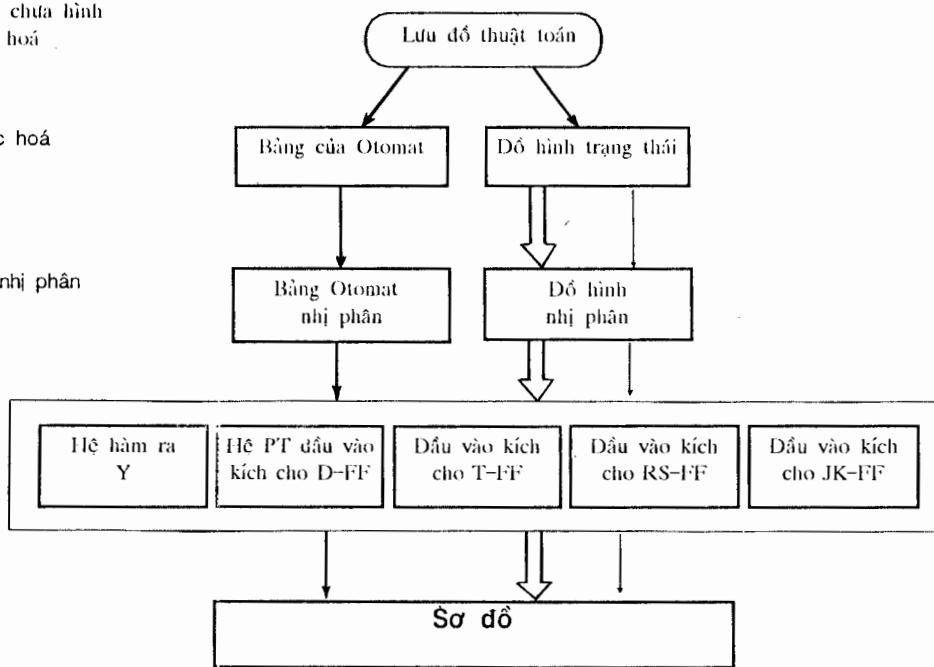
1. Bài toán chưa hình thức hoá

2. Hình thức hoá
 V, R, S

3. Otomat nhị phân
 X, Y, Q

4. Hệ hàm
của
mạch

5. Sơ đồ



Hình 7-10. Các phương pháp thiết kế mạch dây và các bước thiết kế.

7_5. THIẾT KẾ MẠCH DÂY TỪ ĐỒ HÌNH TRẠNG THÁI

Cho trước : Đồ hình trạng thái của mạch có tập tín hiệu vào V , tập tín hiệu ra R , tập trạng thái trong S (chưa mã hoá nhị phân).

Xác định : Hệ phương trình nhị phân của mạch (đã tối thiểu hoá). Trên cơ sở đó vẽ sơ đồ mạch thực hiện.

7_5.1. CÁC BƯỚC THIẾT KẾ

1. Mã hoá tín hiệu vào V , tín hiệu ra R , trạng thái trong S để chuyển otomat ban đầu thành otomat nhị phân có các tập tín hiệu vào X , tín hiệu ra Y , trạng thái trong Q .

2. Xác định hệ phương trình tín hiệu ra : $Y_i = f_i(X, Q)$. Phương trình này được xác định trên các cung với mô hình otomat kiểu Mealy, trên các đinh với mô hình otomat kiểu Moore. Tối thiểu các hàm này .

3. Xác định hệ phương trình hàm kích cho các FF và tối thiểu hóa các FF này.

Dưới đây sẽ trình bày thuật toán xác định phương trình đầu vào kích cho các FF_s từ đồ hình trạng thái.

Đối với FF - Q_i bất kỳ, sự thay đổi trạng thái từ Q_i đến Q'_i chỉ có thể có 4 khả năng như trên hình 7-11.

Trong đó các cung biểu diễn sự thay đổi trạng thái từ Q_i đến Q'_i được ký hiệu như sau : 0 \Rightarrow 0 là (0), 1 \Rightarrow 1 là (1), 0 \Rightarrow 1 là (2) và 1 \Rightarrow 0 là (3)

Từ quy ước đó, có thuật toán sau :

a. Thuật toán xác định phương trình đầu vào kích cho FF - Q_i loại D

Dựa trên cơ sở :

$$D_i = Q'_i$$

rút ra :

$$\begin{aligned} D_i &= Q'_i = \text{Tuyển tất cả các cung đi tới đỉnh có } Q_i = 1. \\ &= \sum \text{các cung loại (2), kể cả khuyên tại đỉnh đó tức là cung loại (1).} \\ &= \sum (1) + (2). \end{aligned}$$

Tối thiểu hóa hàm D_i vừa tìm được rút ra được phương trình đầu vào kích cho FF - Q_i loại D.

b. Thuật toán xác định phương trình đầu vào kích cho T - FF

Dựa trên cơ sở : T_i = Q_i \oplus Q'_i = \dot{Q}_i

trong đó \dot{Q}_i bằng 1 khi Q_i thay đổi trạng thái từ 0 \Rightarrow 1 hoặc từ 1 \Rightarrow 0, ta tiến hành như sau:

- Diện sự thay đổi giá trị Q_i vào các cung .

$$- T_i = \dot{Q}_i = \sum \text{các cung có } Q_i \text{ thay đổi (cung loại 2, loại 3)} = \sum (2) + (3)$$

Tối thiểu hóa hàm T_i vừa tìm được rút ra phương trình đầu vào kích cho T - FF thứ i.

c. Thuật toán xác định phương trình đầu vào kích cho JK - FF

Xác định :

T_{on} = \sum các cung mà Q_i được bật (Q_i thay đổi từ 0 \Rightarrow 1 - cung loại 2) = \sum (2). Dựa phương trình của T_{on} về dạng :

$$T_{on} = (T^*) \bar{Q}_i \Rightarrow \text{rút ra được } J = T^*.$$

T_{off} = \sum các cung mà Q_i được tắt (Q_i thay đổi từ 1 \Rightarrow 0 - cung loại 3) = \sum (3). Dựa phương trình của T_{off} về dạng :

$$T_{off} = (T^{**}) \bar{Q}_i \Rightarrow K = T^{**}.$$

d. Thuật toán xác định phương trình đầu vào kích cho RS - FF

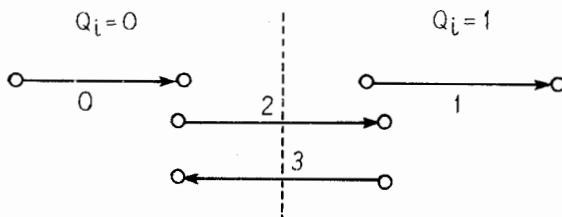
Fương trình đầu vào S của RS - FF được xác định như sau :

$$S = T_{on} + [\text{Các cung loại (1)}]$$

Phương trình đầu vào R của RS - FF được xác định như sau :

$$R = T_{off} + [\text{Các cung loại (0)}]$$

Các cung loại (1), các cung loại (0) để trong dấu [] ở biểu thức của S,R được lấy giá trị không xác định .Những giá trị này và những trạng thái không được sử dụng sẽ được lợi dụng để tối thiểu hóa sao cho biểu thức nhận được là đơn giản nhất.

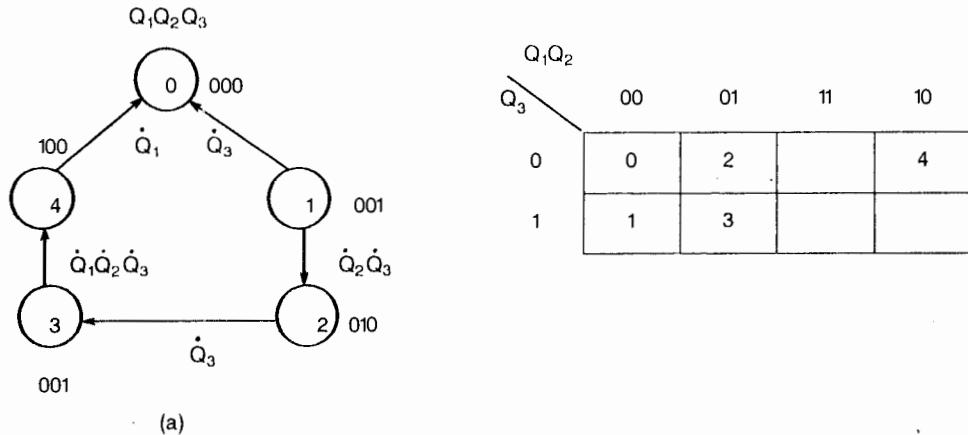


Hình 7-11. Các cung biểu diễn sự thay đổi trạng thái từ $Q_i \Rightarrow Q''_i$ của $FF - Q_i$.

7-5.2. VÍ DỤ

Để minh họa , xét ví dụ sau : Thiết kế mạch đếm đồng bộ có $K_d = 5$ có đồ hình trạng thái và mã hoá trạng thái cho ở hình 7-12, dùng :

- a) $D - FF$ và các mạch $NAND$.
- b) $T - FF$ và các mạch $NAND$.
- c) $JK - FF$ và các mạch $NAND$.
- d) $RS - FF$ và các mạch $NAND$.



Hình 7-12. Bộ đếm $K_d = 5$.

a. Đồ hình trạng thái ; b. Bảng mã hoá trạng thái.

Bộ đếm có $K_d = 5$ nên có 5 trạng thái 0, 1, 2, 3, 4. Để đơn giản , trên đồ hình ta không ghi các tín hiệu vào đếm và tín hiệu ra.Tín hiệu ra của bộ đếm chỉ xuất hiện khi bộ đếm đang ở trạng thái 4 và có tín hiệu vào đếm , lúc đó bộ đếm quay trở về trạng thái ban đầu 0 và cho ra tín hiệu ra.

Mạch có 5 trạng thái và do vậy được mã hoá ít nhất bằng 3 biến nhị phân tương ứng với 3 FF : Q_1 , Q_2 , Q_3 như trên bảng mã hoá trạng thái hình 7-13,b . Điện mã tương ứng vào các trạng thái trên đồ hình 7-13,a.

Dễ dàng viết được phương trình cho tín hiệu ra Y :

$$Q_1 \bar{Q}_2 \bar{Q}_3 \cdot X_d$$

Nếu lợi dụng những trạng thái không sử dụng đến khi mã hoá ,tối thiểu hàm này nhận được kết quả :

$$Y = Q_1 X_d$$

Vấn đề còn lại chỉ là xác định phương trình đầu vào kích cho các FF : Q_1 , Q_2 và Q_3 . Xét các trường hợp khi các FF này là :

a) $D - FF$

Dánh dấu các đỉnh có

$$Q_1 = 1 : \text{Định (4)}$$

$$Q_2 = 1 : \text{Định (2), (3)}$$

$$Q_3 = 1 : \text{Định (1), (3)}$$

$$D_1 = \sum \text{Các cung đi đến đỉnh (4)} = (3) = \bar{Q}_1 Q_2 Q_3.$$

$$D_2 = \sum \text{Các cung đi đến đỉnh (2), (3)} = (1) + (2) = \bar{Q}_1 \bar{Q}_2 Q_3 + \bar{Q}_1 Q_2 \bar{Q}_3.$$

$$D_3 = \sum \text{Các cung đi đến đỉnh (1), (3)} = (0) + (2) = \bar{Q}_1 \bar{Q}_2 \bar{Q}_3 + \bar{Q}_1 Q_2 \bar{Q}_3.$$

D_1	$Q_2 Q_3$	00	01	11	10
Q_1					
0				(1)	
1		X	X		X

$$D_1 = Q_2 Q_3$$

D_3	$Q_2 Q_3$	00	01	11	10
Q_1					
0	1				(1)
1		X	X		X

$$D_3 = \bar{Q}_1 \bar{Q}_3$$

D_2	$Q_2 Q_3$	00	01	11	10
Q_1					
0			(1)		
1		X		X	X

$$D_2 = \bar{Q}_2 Q_3 + Q_2 \bar{Q}_3 = Q_2 \oplus Q_3$$

Hình 7-13. Bảng Karnaugh cho đầu vào kích cho các $D - FF$.

Biểu diễn các hàm này trên bảng Karnaugh , trong đó các trạng thái không sử dụng được đánh dấu X (hình 7-13). Tối thiểu hoá , nhận được kết quả :

$$D_1 = Q_2 Q_3$$

$$D_2 = Q_2 \bar{Q}_3 + \bar{Q}_2 Q_3 = Q_2 \oplus Q_3$$

$$D_3 = \bar{Q}_1 \bar{Q}_3$$

b) Xác định phương trình đầu vào kích cho các T - FF

Điển sự thay đổi giá trị của Q_i (Q_i) vào các cung. Ví dụ : Khi mạch đếm từ trạng thái (0) \Rightarrow (1) (nghĩa là từ 000 \Rightarrow 001) Q_3 thay đổi giá trị từ 0 \Rightarrow 1 nên ta ghi Q_3 lên cung đó. Khi mạch chuyển từ trạng thái (1) \Rightarrow (2) (từ 001 \Rightarrow 010) : Q_1 không thay đổi vẫn giữ nguyên là 0 , Q_2 thay đổi từ 0 \Rightarrow 1 , Q_3 thay đổi từ 1 \Rightarrow 0 và ta ghi $Q_2 Q_3$ lên cung từ (1) \Rightarrow (3) (xem hình 7-12,a). Ta có :

$$T_1 = Q_1 = \sum \text{các cung có } Q_1 \text{ thay đổi} = (3) + (4) = \bar{Q}_1 Q_2 Q_3 + Q_1 \bar{Q}_2 \bar{Q}_3$$

$$T_2 = Q_2 = \sum \text{các cung có } Q_2 \text{ thay đổi} = (1) + (3) = \bar{Q}_1 \bar{Q}_2 Q_3 + \bar{Q}_1 Q_2 Q_3$$

$$T_3 = Q_3 = \sum \text{các cung có } Q_3 \text{ thay đổi} = (0) + (1) + (2) + (3) \\ = \bar{Q}_1 \bar{Q}_2 \bar{Q}_3 + \bar{Q}_1 \bar{Q}_2 Q_3 + \bar{Q}_1 Q_2 \bar{Q}_3 + \bar{Q}_1 Q_2 Q_3 .$$

Lập bảng Karnaugh cho các hàm trên (hình 7-14) rồi tối thiểu hoá các hàm trên với sự lợi dụng các trạng thái không sử dụng, nhận được kết quả :

$$T_1 = Q_1 + Q_2 Q_3$$

$$T_2 = Q_3$$

$$T_3 = \bar{Q}_1 .$$

		$Q_2 Q_3$	00	01	11	10
		0			1	
Q_1	0	1	X	X	X	
		1				

		$Q_2 Q_3$	00	01	11	10
		0		1	1	
Q_1	0	1				
		1				

		$Q_2 Q_3$	00	01	11	10
		0	1	1	1	1
Q_1	0	1	1	1	1	1
		1				

$$T_2 = Q_3$$

$$T_3 = \bar{Q}_1$$

Hình 7-14. Bảng Karnaugh cho các T - FF.

c) JK - FF

Chú ý khi viết các biểu thức T_{on} , T_{off} của FF thứ i ta cần phải đơn giản các biểu thức đó và đưa về dạng :

$$T_{on} = (T^*) \bar{Q}_i \text{ rồi suy ra } J_i = T^*$$

$$T_{off} = (T^{**}) Q_i \text{ rồi suy ra } K_i = T^{**}$$

Viết các biểu thức T_{on} , T_{off} cho các FF và từ đó xác định phương trình các đầu vào kích cho các FF đó như sau :

$$T_{on1} = \sum \text{Các cung mà } Q_1 \text{ được bật (chuyển từ } 0 \Rightarrow 1) = (3) = \bar{Q}_1 Q_2 Q_3$$

$$T_{off1} = \sum \text{Các cung mà } Q_1 \text{ tắt (chuyển từ } 1 \Rightarrow 0) = (4) = Q_1 \bar{Q}_2 \bar{Q}_3$$

$$T_{on2} = \sum \text{Các cung mà } Q_2 \text{ được bật (chuyển từ } 0 \Rightarrow 1) = (1) = \bar{Q}_1 \bar{Q}_2 Q_3$$

$$T_{off2} = \sum \text{Các cung mà } Q_2 \text{ tắt (chuyển từ } 1 \Rightarrow 0) = (3) = \bar{Q}_1 Q_2 Q_3$$

$Q_2 Q_3$

		00	01	11	10
		0			
		1	X	X	X
$J_1 = Q_2 Q_3$					

$Q_2 Q_3$

		00	01	11	10	
		0	(1)	X	X	1
		1		X	X	X
$J_3 = \bar{Q}_1$						

$Q_2 Q_3$

		00	01	11	10	
		0	(1)	1		X
		1	X	X	X	X
$J_2 = Q_3$						

$Q_2 Q_3$

		00	01	11	10	
		0	X	X	X	X
		1	1	X	X	X
$K_1 =$						

$Q_2 Q_3$

		00	01	11	10	
		0	X	(1)	1	
		1	X	X	X	
$K_2 = Q_3$						

$Q_2 Q_3$

		00	01	11	10	
		0	(X)	1	1	X
		1	X	X	X	X
$K_3 = 1$						

$$K_3 = 1$$

Hình 7-15. Bảng Karnaugh cho các JK - FF.

$$T_{on3} = \sum \text{Các cung mà } Q_3 \text{ được bật (chuyển từ } 0 \Rightarrow 1) = (0) + (2) = \bar{Q}_1 \bar{Q}_3$$

$$T_{off3} = \sum \text{Các cung mà } Q_3 \text{ tắt (chuyển từ } 1 \Rightarrow 0) = (1) + (3) = \bar{Q}_1 Q_3.$$

Biểu diễn các hàm này trên bảng Karnaugh (hình 7-15), lợi dụng các trạng thái không sử dụng (đánh dấu X) và tối thiểu hóa được kết quả :

$$J_1 = Q_2 Q_3 \quad K_1 = 1$$

$$J_2 = Q_3 \quad K_2 = Q_3$$

$$J_3 = Q_1 \quad K_3 = 1.$$

		R_1	$Q_2 Q_3$		
		00	01	11	10
Q_1	0	(X)	(X)		(X)
	1	1	X	X	(X)

$$R_1 = Q_1; \quad R_1 = \bar{Q}_2; \quad R_1 = \bar{Q}_3$$

		R_3	$Q_2 Q_3$		
		00	01	11	10
Q_1	0		(1)	(1)	
	1	X	X	X	X

$$R_3 = Q_3$$

		S_1	$Q_2 Q_3$		
		00	01	11	10
Q_1	0			(1)	
	1		X	X	X

$$S_1 = Q_2 Q_3$$

		S_3	$Q_2 Q_3$		
		00	01	11	10
Q_1	0	(1)			(1)
	1		X	X	X

$$S_3 = \bar{Q}_1 \bar{Q}_3$$

		R_2	$Q_2 Q_3$		
		00	01	11	10
Q_1	0	X		(1)	
	1	X	X	X	X

$$R_2 = Q_2 Q_3$$

		S_2	$Q_2 Q_3$		
		00	01	11	10
Q_1	0		(1)		X
	1		X	X	X

$$S_2 = \bar{Q}_2 Q_3$$

Hình 7-16. Bảng Karnaugh cho các RS - FF.

d) RS -FF

$$S_1 = T_{on1} + [\text{Các cung loại (1)}] = (3) + [\Phi].$$

$$R_1 = T_{off1} + [\text{Các cung loại (0)}] = (4) + [(0), (1), (2)]$$

$$S_2 = T_{on2} + [\text{Các cung loại (1)}] = (1) + [(2)]$$

$$R_2 = T_{off2} + [\text{Các cung loại (0)}] = (3) + [(0), (4)]$$

$$S_3 = T_{on3} + [\text{Các cung loại (1)}] = (0) + [(2) + [\Phi]]$$

$$R_3 = T_{off3} + [\text{Các cung loại (0)}] = (1) + [(3) + [(4)]].$$

Biểu diễn các hàm này trên bảng Karnaugh (hình 7-16) và tối thiểu hóa, nhận được kết quả :

$$S_1 = Q_2 Q_3$$

$$R_1 = Q_1 \text{ hoặc } R_1 = \bar{Q}_2 \text{ hoặc } R_1 = \bar{Q}_3$$

$$S_2 = \bar{Q}_2 Q_3$$

$$R_2 = Q_2 Q_3$$

$$S_3 = \bar{Q}_1 \bar{Q}_3$$

$$R_3 = Q_3 .$$

7_6. THIẾT KẾ MẠCH DÂY TỪ BẢNG CỦA OTOMAT

Cho trước : bảng chuyển đổi trạng thái, bảng ra của otomat ban đầu (chưa mã hoá nhị phân).

Cần phải xác định : Hệ phương trình nhị phân của mạch bao gồm hệ hàm ra, hệ hàm kích cho các FF. Trên cơ sở đó vẽ sơ đồ mạch.

Thực hiện theo các bước sau :

1. Mã hoá tín hiệu vào V , tín hiệu ra R , trạng thái trong của mạch S để chuyển ô mat ban đầu thành otomat nhị phân có : tập tín hiệu vào X , tập tín hiệu ra Y , tập trạng thái trong Q .

2. Lập bảng chuyển đổi trạng thái, bảng ra của otomat nhị phân ứng với sự mã hoá trên.

3. Dựa vào bảng các đầu vào kích cho các FF, xác định được đầu vào kích cho các FF ứng với sự chuyển đổi trong bảng trạng thái.

4. Viết phương trình đầu vào kích cho từng FF - Q_i , và các hàm ra rồi tối thiểu các hàm này. Trên cơ sở hệ hàm này xây dựng sơ đồ mạch.

Để hiểu rõ các bước trên, lại xét ví dụ trên : Thiết kế bộ đếm có $K_d = 5$, đồ hình trạng thái cho ở hình 7-12,a. Từ đó lập bảng chuyển đổi trạng thái như ở hình 7-17,a, mã hoá trạng thái như hình 7-17,b. Dựa vào hai bảng này và cần căn cứ vào bảng các đầu vào kích cho FF hình 1-17,c lập được bảng như hình 7-17,d. Dựa vào bảng này xác định được

ác phương trình các đầu vào kích cho các loại FF. Bảng Karnaugh và kết quả tối thiểu giống như ở mục 7-6 [các hình 7-13, 7-14, 7-15 và 7-16].

S	S'
0	1
1	2
2	3
3	4
4	0

a)

		Q ₁ Q ₂			
		00	01	11	10
Q ₃	0	0	2		4
	1	1	3		

b)

Q	Q'	D	T	R S	J K
0	0	0	0	X 0	0 X
0	1	1	1	0 1	1 X
1	0	0	0	1 0	X 1
1	1	1	1	0 X	X 0

c)

Q ₁	Q ₂	Q ₃	Q' ₁	Q' ₂	Q' ₃	D ₁	D ₂	D ₃	T ₁	T ₂	T ₃	R ₁ S ₁	R ₂ S ₂	R ₃ S ₃	J ₁ K ₁	J ₂ K ₂	J ₃ K ₃
0	0	0	0	0	1	0	0	1	0	0	1	X 0	X 0	0 1	0 X	0 X	1 X
0	0	1	0	1	0	0	1	0	0	1	1	X 0	0 1	1 0	0 X	1 X	X 1
0	1	0	0	1	1	0	1	1	0	0	1	X 0	0 X	0 1	0 X	X 0	1 X
0	1	1	1	0	0	1	0	0	1	1	1	0 1	1 0	1 0	1 X	X 1	X 1
1	0	0	0	0	0	0	0	0	1	0	0	1 0	X 0	X 0	X 1	0 X	0 X
1	0	1	X	X	X	X	X	X	X	X	X	X X	X X	X X	X X	X X	X X
1	1	0	X	X	X	X	X	X	X	X	X	X X	X X	X X	X X	X X	X X
1	1	1	X	X	X	X	X	X	X	X	X	X X	X X	X X	X X	X X	X X

d)

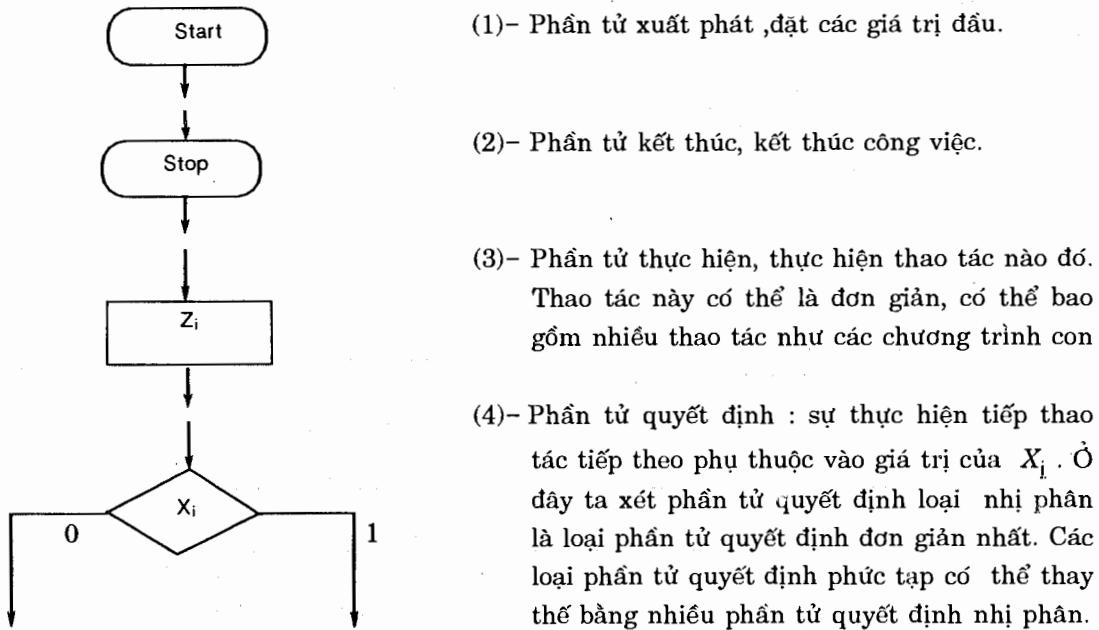
Hình 7-17. Thiết kế mạch dây từ bảng otomat.

- a. Bảng chuyển đổi trạng thái ; b. Mã hóa trạng thái ; c. Bảng đầu vào kích cho các FF ; d. Bảng đầu vào kích cho các FF ; e. Bảng trạng thái nhị phân và đầu vào kích cho các loại FF.

7-7. THIẾT KẾ MẠCH DÂY TỪ LUU ĐỒ THUẬT TOÁN

7-7.1. LUU ĐỒ THUẬT TOÁN VÀ CÁC PHẦN TỬ CƠ BẢN CỦA NÓ

Lưu đồ thuật toán là cách mô tả quan hệ một cách trực quan. Các phần tử cơ bản của lưu đồ thuật toán và các ký hiệu của các phần tử đó được biểu diễn ở hình 7-18.



Hình 7-18. Các phần tử cơ bản của lưu đồ thuật toán.

7_7.2. THUẬT TOÁN CHUYỂN TỪ LƯU ĐỒ THUẬT TOÁN THÀNH ĐỒ HÌNH OTOMAT MEALY

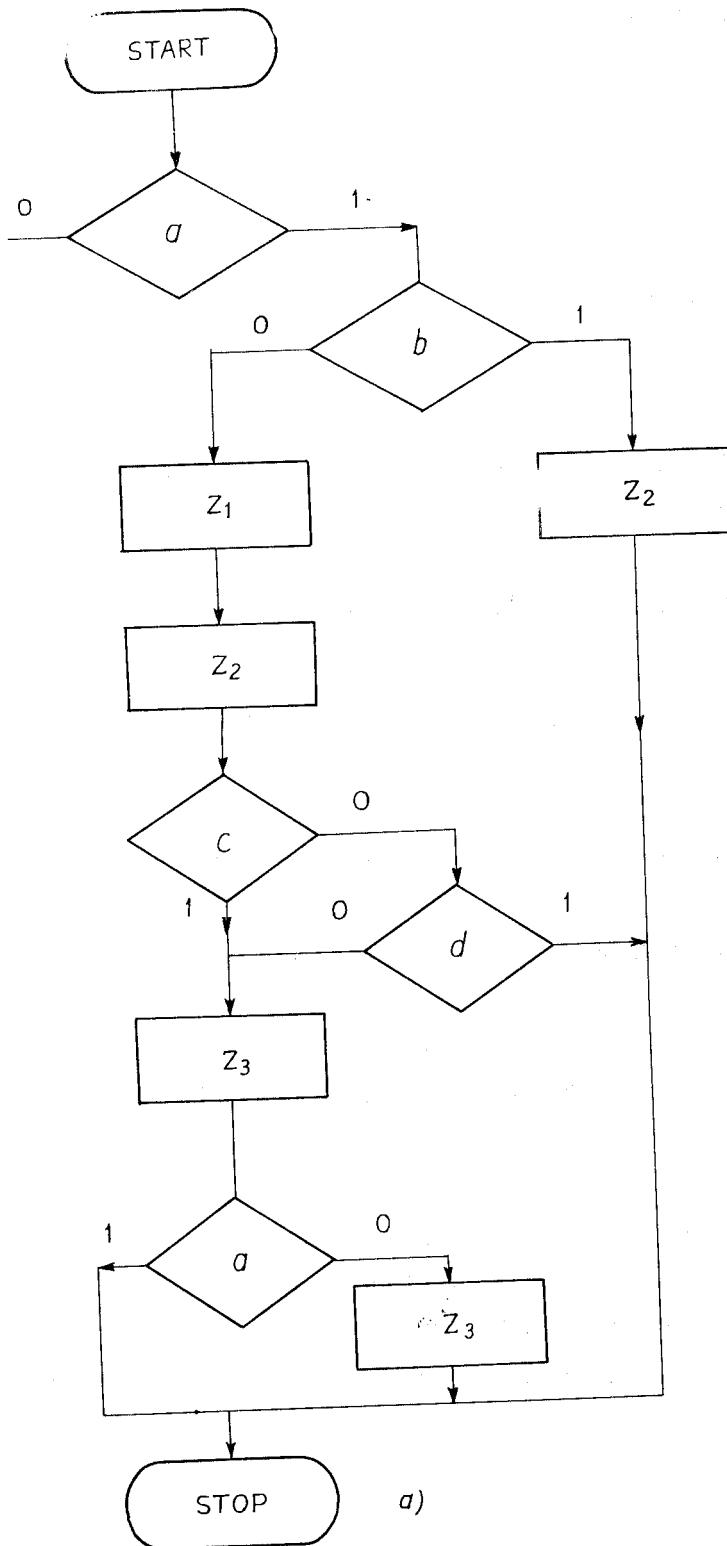
Thuật toán bao gồm các bước như sau :

- *Bước 1* : Ra của khối bắt đầu (Start) và vào của khối kết thúc (Stop) có thể được biểu diễn bằng cùng 1 trạng thái nếu như otomat khi kết thúc thuật toán lại quay về trạng thái ban đầu.
- *Bước 2* : Tất cả các đầu ra của khối thực hiện được gán trạng thái.
- *Bước 3* : Xây dựng đồ hình cho otomat với đỉnh là trạng thái , cung là chuyển biến tương ứng giữa các đỉnh.
- *Bước 4* : Gán lên các cung của đồ hình giá trị của tín hiệu vào, tín hiệu ra tương ứng.

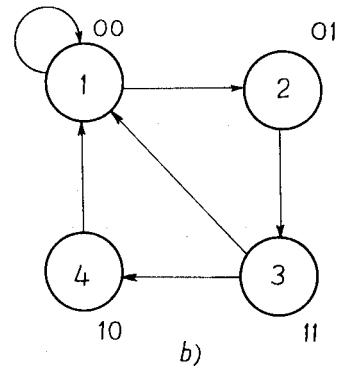
Để làm rõ thuật toán , xét ví dụ sau :

Cho lưu đồ thuật toán ở hình 7-19,a , áp dụng thuật toán đã giới thiệu chuyển lưu đồ thuật toán này thành đồ hình trạng thái cho mô hình Mealy như hình 7-19,b.

Với đồ hình trạng thái này, áp dụng phương pháp thiết kế mạch dây từ đồ hình trạng thái đã giới thiệu ở mục 7- 6, với sự mã hoá 4 trạng thái như ở hình 7-19,c, nhận được đồ hình nhị phân hình 7-16,d. và cuối cùng nhận được hệ phương trình của mạch.



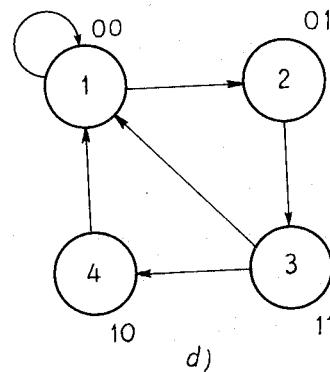
a)



b)

Q_1	0	1
0	1	2
1	4	3

c)



d)

Hình 7-19. Thiết kế từ LDTT (dùng mô hình Mealy).

- a. Lưu đồ thuật toán và sự gán trạng thái ; b. Đồ hình trạng thái ;
- c. Mã trạng thái; d. Đồ hình nhị phân.

Trong ví dụ này chúng ta viết hệ phương trình của mạch đối với $T - FF$ và ta có :

Hệ hàm ra :

$$Z_1 = a\bar{b} \quad (1)$$

$$Z_2 = ab \quad (1) + \quad (2)$$

$$Z_3 = (c + \bar{d}) \quad (3) + \bar{a} \quad (4)$$

Hệ hàm đầu vào kích cho $T - FF_1$ và $T - FF_2$:

$$T_1 = Y_1 = (2) + \bar{c}d \quad (3) + \quad (4)$$

$$T_2 = Y_2 = ab \quad (1) + \quad (3)$$

Thay (1) = $\bar{Y}_1 \bar{Y}_2$, (2) = $\bar{Y}_1 Y_2$, (3) = $Y_1 Y_2$ và (4) = $Y_1 \bar{Y}_2$ vào các phương trình trên và tối thiểu hóa, nhận được kết quả. Từ đó xây dựng sơ đồ mạch cần thiết kế.

7-7.3. THUẬT TOÁN CHUYỂN TỪ LƯU ĐỒ THUẬT TOÁN THÀNH OTOMAT MOORE

Thuật toán bao gồm các bước như sau :

- Bước 1 : Khối bắt đầu (Start) và khối kết thúc (Stop) sẽ được gán cùng 1 biến trạng thái nếu như Otomat khi kết thúc thuật toán lại quay về trạng thái ban đầu.
- Bước 2 : Mỗi khối thực hiện được gán 1 trạng thái.
- Bước 3 : Xây dựng đồ hình cho otomat với đỉnh là trạng thái, cung là chuyển biến có thể có giữa các đỉnh.
- Bước 4 : Gán lên các cung của đồ hình giá trị của tín hiệu vào, và trên các đỉnh tín hiệu ra tương ứng.

Sau khi đã nhận được đồ hình trạng thái của Otomat Moore, áp dụng phương pháp thiết kế mạch dây từ đồ hình đã được giới thiệu ở mục 7-7 nhận được kết quả bài toán thiết kế.

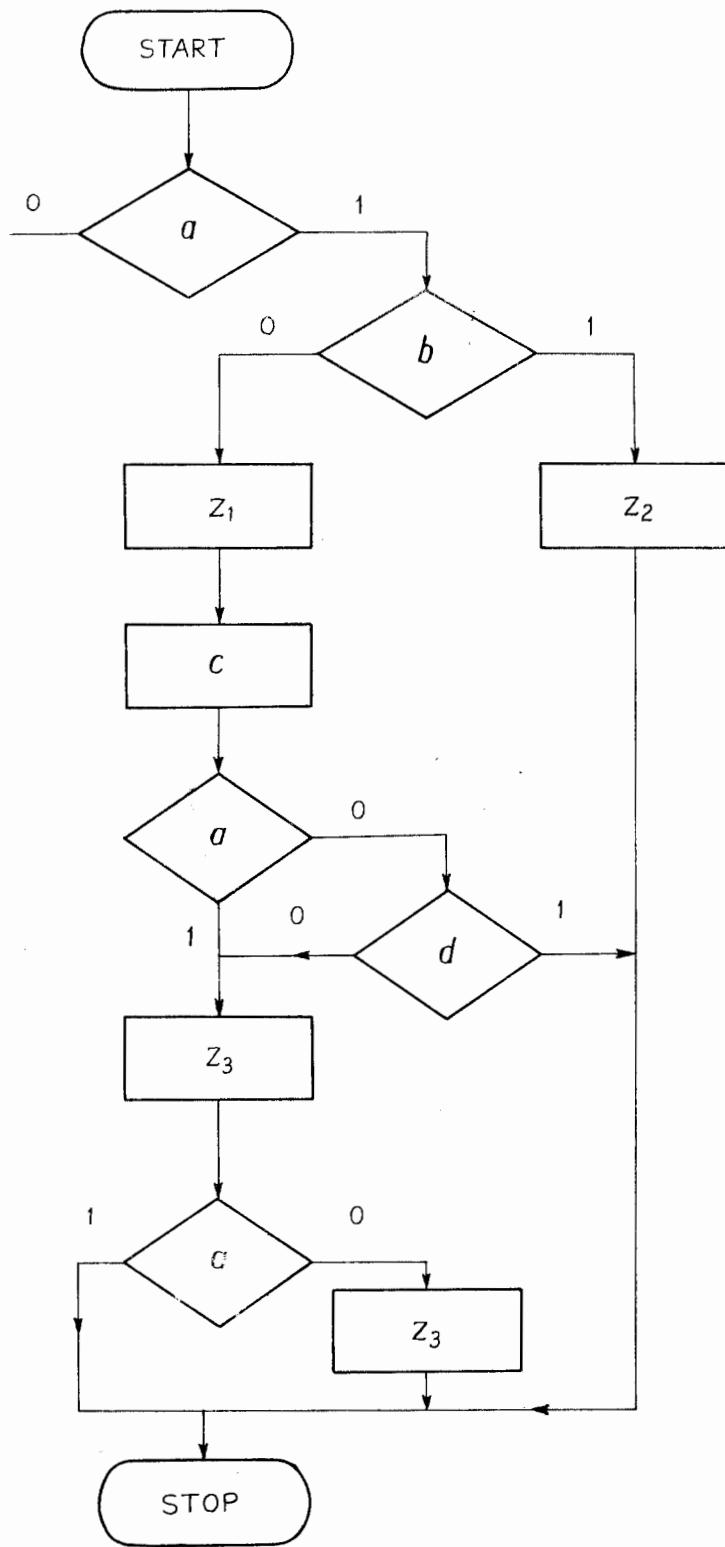
Xét ví dụ cụ thể sau :

Lưu đồ thuật toán cho ở hình 7-20,a. Áp dụng thuật toán đã giới thiệu ở trên ta nhận được đồ hình trạng thái otomat Moore ở hình 7-20,b. Mã hoá trạng thái như ở hình 7-20,c , nhận được đồ hình trạng thái nhị phân như hình 7-20,d.

Viết hệ phương trình đầu vào kích cho các FF (ở ví dụ này ta dùng $T - FF$) và phương trình tín hiệu ra như hình 7-20,e.

Nhận xét :

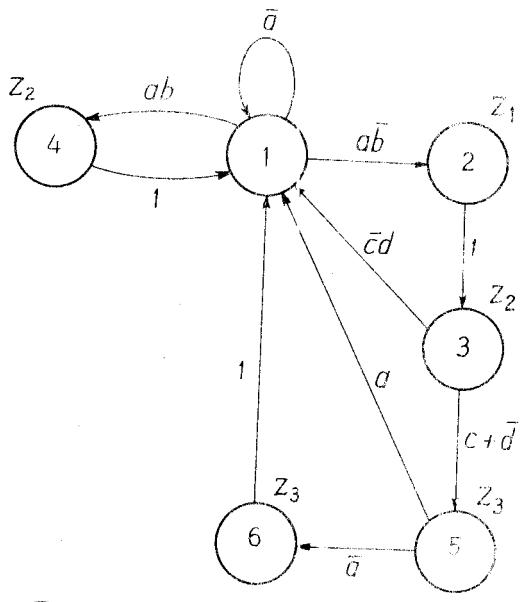
Thiết kế mạch dây từ lưu đồ thuật toán cho thấy rõ mối quan hệ giữa phần mềm (Software) và phần cứng (Hardware). Từ chương trình phần mềm có thể xây dựng mạch tương ứng thực hiện chức năng đó và ngược lại .



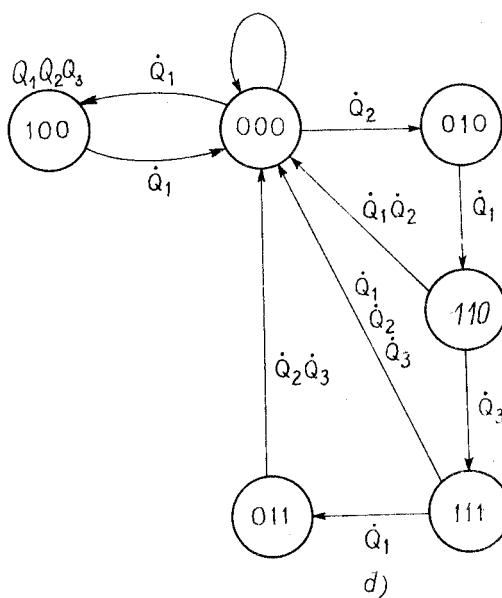
a)

$Q_3 Q_2$	00	01	11	10
0	1	2	3	4
1		6	5	

c)



b)



d)

$$\left. \begin{aligned} Q_1 &= ab(1) + \bar{ab}(2) + \bar{c}d(3) + \bar{a}(4) + a(5) \\ Q_2 &= a\bar{b}(1) + \bar{c}d(3) + a(5) + \bar{a}(6) \\ Q_3 &= (c+\bar{d})(3) + a(5) + \bar{a}(6) \\ Z_1 &= (2) \\ Z_2 &= (3) + (4) \\ Z_3 &= (5) + (6) \end{aligned} \right\}$$

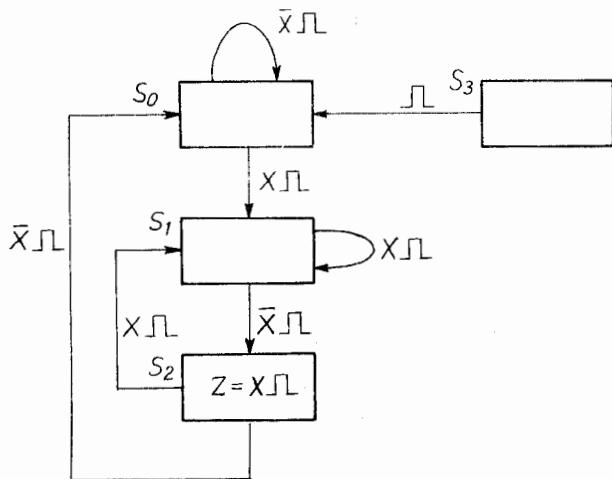
e)

Hình 7-20. Thiết kế từ LDTT (dùng mô hình Moore)

- a. LDTT và sự gán trạng thái ; b. Đồ hình trạng thái của otomat Moore ; c. Bảng mã hoá trạng thái ; d. Đồ hình nhị phân của otomat Moore ; e. Hệ phương trình của mạch.

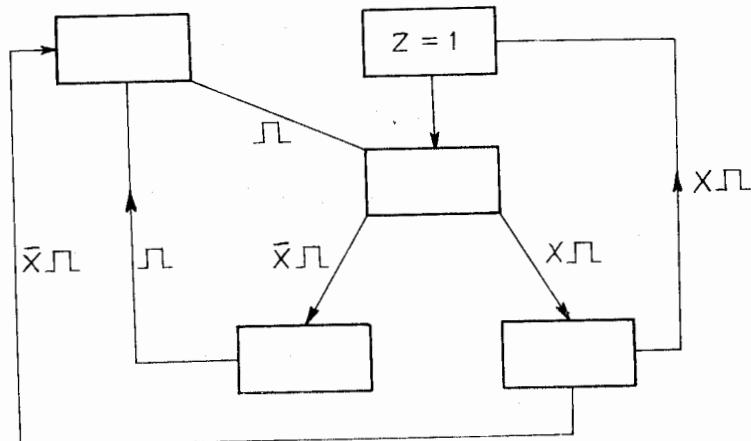
BÀI TẬP

7.1. Mạch dãy được mô tả bởi mô hình Mealy có đồ hình trạng thái như hình sau :
Hãy thực hiện việc chuyển đồ hình này sang đồ hình kiểu Moore.



Hình 7-21. Cho bài tập 7-1.

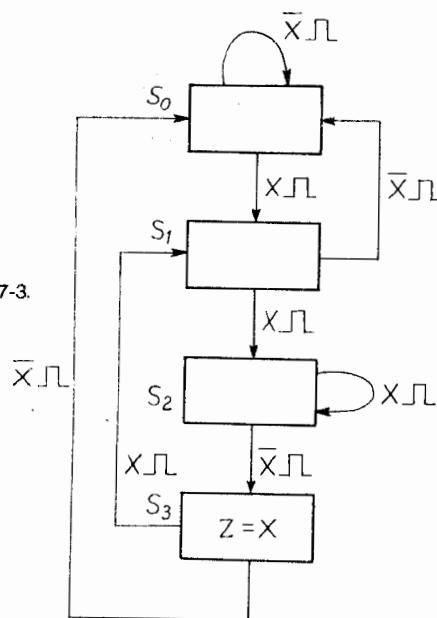
7.2. Mạch dãy được mô tả bởi mô hình Moore có đồ hình trạng thái cho trong hình sau :
Hãy mô tả mạch dãy này theo mô hình Mealy.



Hình 7-22. Cho bài tập 7-2.

7.3. Thiết kế mạch dãy đồng bộ có đồ hình trạng thái như hình sau :

7.4. Dùng $JK - FF$ và các mạch $NAND$ thiết kế mạch dãy có bảng trạng thái và tín hiệu ra như hình vẽ (với Z là tín hiệu ra, X là tín hiệu vào).



Hình 7-23. Cho bài tập 7-3.

S	S'/Z	
	X = 0	X = 1
S ₀	S ₁ /0	S ₁ /0
S ₁	S ₃ /0	S ₂ /0
S ₂	S ₀ /1	S ₀ /1
S ₃	S ₀ /0	S ₀ /0

Hình 7-24. Cho bài tập 7-4.

7.5. Thiết kế mạch dãy có bảng trạng thái, bảng ra cho trong hình sau (dùng JK - FF và các mạch NAND).

S	S'/Z	
	X = 0	X = 1
S ₀	S ₁ /0	S ₄ /0
S ₁	S ₂ /0	S ₂ /0
S ₂	S ₃ /0	S ₃ /1
S ₃	S ₀ /0	S ₀ /0
S ₄	S ₇ /0	S ₅ /0
S ₅	S ₆ /0	S ₆ /0
S ₆	S ₀ /1	S ₀ /1
S ₇	S ₃ /0	S ₆ /0

Hình 7-25. Cho bài tập 7-5.

CHƯƠNG 8

BỘ ĐẾM

Mạch đếm là mạch dãy đơn giản, cũng như các mạch dãy khác mạch đếm được xây dựng từ các phần tử nhớ là các Flip – Flop và các phần tử tổ hợp.

Các mạch đếm là thành phần cơ bản của các hệ thống số, chúng được sử dụng để đếm thời gian, chia tần số, điều khiển các mạch khác... Mạch đếm được dùng rất nhiều trong máy tính, trong thông tin. Để xây dựng bộ đếm, người ta có thể dùng mã nhị phân hoặc các loại mã khác như mã Gray, mã NBCD, mã vòng ...

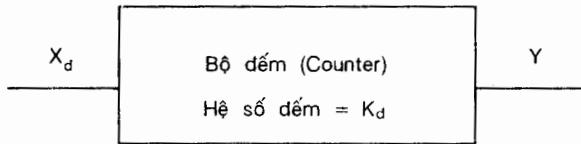
Chương này sẽ đưa ra những đặc điểm cơ bản nhất của bộ đếm và các phương pháp thiết kế bộ đếm.

8-1. ĐỊNH NGHĨA VÀ PHÂN LOẠI BỘ ĐẾM

8-1.1. ĐỊNH NGHĨA

Bộ đếm là một mạch dãy tuần hoàn có một đầu vào đếm và một đầu ra, mạch có số trạng thái trong bằng chính hệ số đếm (ký hiệu là K_d). Dưới tác dụng của tín hiệu vào đếm, mạch sẽ chuyển từ trạng thái trong này đến một trạng thái trong khác theo một thứ tự nhất định. Cứ sau K_d tín hiệu vào đếm, mạch lại trở về trạng thái xuất phát ban đầu.

Sơ đồ khối của bộ đếm được mô tả ở hình 8-1.



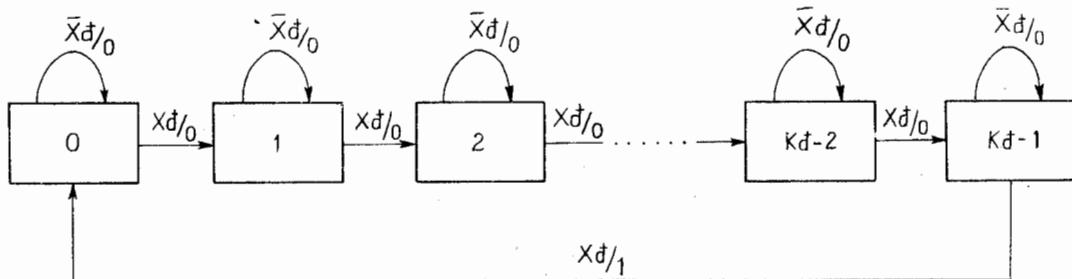
Hình 8-1. Sơ đồ khối của bộ đếm.

8-1.2. ĐỒ HÌNH TRẠNG THÁI TỔNG QUÁT CỦA BỘ ĐẾM

Đồ hình trạng thái của một bộ đếm có hệ số đếm bằng K_d được mô tả ở hình 8-2.

Khi không có tín hiệu vào đếm (X_d), mạch giữ nguyên ở trạng thái cũ. ($i \rightarrow i$), khi có tín hiệu vào đếm (X_d) mạch sẽ chuyển đến trạng thái kế tiếp ($i \rightarrow i + 1$).

Tính chất tuần hoàn của bộ đếm thể hiện ở chỗ : sau K_d tín hiệu vào X_d mạch lại quay trở về trạng thái xuất phát ban đầu .



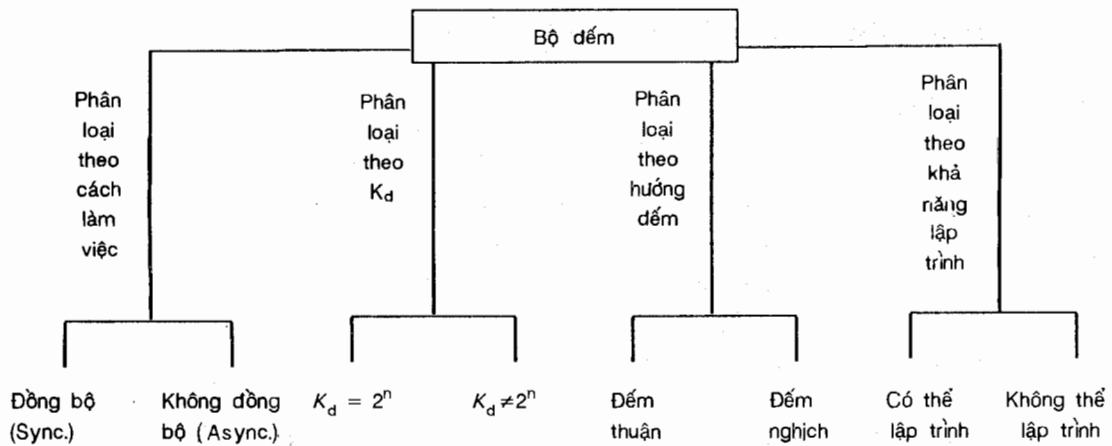
Hình 8-2. Đồ hình trạng thái của bộ đếm K_d .

Tín hiệu ra của bộ đếm chỉ xuất hiện ($Y = 1$) duy nhất trong trường hợp : bộ đếm đang ở trạng thái K_{d-1} và có tín hiệu vào X_d . Khi đó bộ đếm sẽ chuyển về trạng thái 0 .

Trong trường hợp cần hiển thị trạng thái của bộ đếm thì phải dùng thêm mạch giải mã .

8-1.3. PHÂN LOẠI BỘ ĐẾM

Có nhiều cách để phân loại bộ đếm, sau đây là một vài cách hay được dùng nhất (hình 8-3) .



Hình 8-3. Phân loại bộ đếm.

a. Phân loại theo cách làm việc :

- Bộ đếm đồng bộ (Synchronous)
- Bộ đếm không đồng bộ (Asynchronous)

Bộ đếm đồng bộ (Synchronous counter) là bộ đếm mà các FF dùng để mã hoá cho các trạng thái trong của bộ đếm thay đổi trạng thái cùng một lúc khi có tín hiệu vào đếm và mọi sự chuyển đổi trạng thái (từ S_i sang trạng thái mới S_j với i, j) đều không qua các trạng thái trung gian (hình 8-4,a).

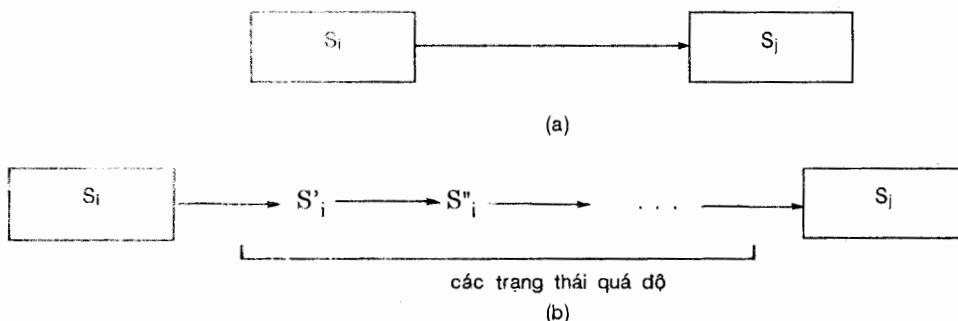
Đặc điểm của bộ đếm này là tín hiệu xung nhịp C_k được đưa đồng thời (song song) vào các **Flip-flop**.

b. Bộ đếm không đồng bộ (Asynchronous counter) :

Nếu trong bộ đếm tồn tại ít nhất một cặp chuyển biến trạng thái từ $S_i \Rightarrow S_j$ mà trong đó các FF khác nhau thay đổi trạng thái cùng một lúc thì bộ đếm đó được gọi là bộ đếm không đồng bộ.

Đặc điểm của bộ đếm này là tín hiệu xung nhịp C_k không được đưa đồng thời vào các FF .

Ví dụ sự chuyển từ trạng thái $S_i \Rightarrow S_j$ của bộ đếm không đồng bộ được mô tả như hình 8-4,b.



Hình 8-4. a) Sự chuyển trạng thái trong của bộ đếm đồng bộ;
b) Sự chuyển biến trạng thái trong của bộ đếm không đồng bộ.

Giả sử ban đầu bộ đếm không đồng bộ đang ở trạng thái cân bằng ổn định S_i , khi có tín hiệu vào đếm, chỉ có một số FF bị tác động, sau đó sự thay đổi trạng thái của các FF này lại dẫn đến sự thay đổi trạng thái của các FF khác. Quá trình cứ tiếp tục cho đến khi bộ đếm ở trạng thái cân bằng ổn định mới S_j . Nghĩa là trong quá trình chuyển đổi giữa hai trạng thái cân bằng ổn định, các FF không thay đổi trạng thái cùng một lúc. Các trạng thái trung gian giữa 2 trạng thái S_i và S_j còn gọi là các trạng thái quá độ. Quá trình chuyển qua các trạng thái trung gian trước khi đến trạng thái ổn định mới S_j làm cho tín hiệu trên đầu ra của bộ đếm bị "gợn sóng" (Ripple). Chính vì vậy bộ đếm không đồng bộ còn được gọi là Ripple Counter .

b. Phân biệt theo hệ số đếm (K_d) :

- Bộ đếm có hệ số đếm $K_d = 2^n$ (n là số tự nhiên);

- Bộ đếm có hệ số đếm $K_d \neq 2^n$.

Bộ đếm có hệ số đếm $K_d = 2^n$:

Ví dụ $K_d = 2, 4, 8, 16 \dots$ Bộ đếm này còn được gọi là bộ đếm có hệ số đếm cực đại hay chiều dài cực đại (Maximum Length), vì khi sử dụng n FF để mã hoá các trạng thái trong cho bộ đếm thì khả năng mã hoá tối đa là 2^n .

Bộ đếm có hệ số đếm $K_d \neq 2^n$:

Ví dụ $K_d = 3, 6, 10 \dots$ Vẫn sử dụng n FF để mã hoá các trạng thái trong cho bộ đếm cho nên sẽ có $2^n - K_d$ trạng thái trong không được sử dụng đến. Vì vậy khi thiết kế bộ đếm

này cần phải lưu ý đến các trạng thái không sử dụng đó, tức là cần phải có biện pháp làm cho bộ đếm thoát khỏi các trạng thái đó một cách hợp lý để trở về chu trình đúng mà vẫn phải đảm bảo bộ đếm được thiết kế là đơn giản.

c. Phân loại theo hướng đếm :

- Bộ đếm thuận (Up Counter);
- Bộ đếm nghịch (Down Counter).

Bộ đếm thuận là bộ đếm mà mỗi khi có tín hiệu vào đếm X_d thì trạng thái trong của bộ đếm tăng lên 1.

Ví dụ : Bộ đếm đang ở trạng thái $S_5 : 101$ khi có một tín hiệu vào đếm X_d nó sẽ chuyển sang trạng thái $S_6 : 110$.

Bộ đếm nghịch :

Là bộ đếm mà mỗi khi có tín hiệu vào đếm X_d thì trạng thái trong của bộ đếm giảm đi 1.

Ví dụ : Ban đầu bộ đếm đang ở trạng thái $S_9 : 1001$ khi có tín hiệu X_d nó sẽ chuyển sang trạng thái $S_8 : 1000$.

Chú ý rằng khái niệm thuận, nghịch chỉ là tương đối, chủ yếu là do vấn đề mã hoá các trạng thái trong của bộ đếm .

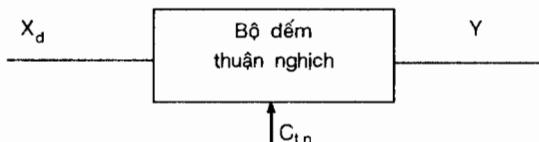
Trong thực tế người ta còn thiết kế bộ đếm thuận nghịch , bộ đếm này vừa có khả năng đếm thuận, vừa có khả năng đếm nghịch. Trong trường hợp này phải đưa thêm tín hiệu điều khiển đếm thuận hay đếm nghịch.

Sơ đồ khối của bộ đếm thuận nghịch được mô tả ở hình 8-5, trong đó C_{tn} là tín hiệu điều khiển việc đếm thuận hay đếm nghịch.

d. Phân loại theo khả năng lập trình :

- Bộ đếm có khả năng chương trình hoá (Programable Counter);
- Bộ đếm không có khả năng chương trình hoá.

Bộ đếm có khả năng chương trình hoá là bộ đếm có thể sử dụng với các hệ số đếm khác nhau, tùy thuộc vào tín hiệu điều khiển đưa vào nó. Do vậy bộ đếm này "mềm dẻo" , đa năng hơn các bộ đếm chỉ có 1 hệ số đếm cố định và ngày càng được sử dụng rộng rãi.



Hình 8-5. Sơ đồ khối của bộ đếm thuận nghịch.

8-2. MÃ CỦA BỘ ĐẾM

Quá trình đếm của bộ đếm là quá trình thay đổi từ trạng thái trong này đến trạng thái trong khác và mỗi trạng thái trong của bộ đếm được mã hoá bởi một mã cụ thể.

Cùng một bộ đếm, có thể có nhiều cách mã hoá các trạng thái trong khác nhau. Các cách mã hoá khác nhau sẽ tương ứng với những mạch thực hiện khác nhau mặc dù các mạch đó có cùng chức năng, hệ số đếm K_d ... Sau đây là một vài loại mã thường dùng để mã hoá trong bộ đếm.

8_2.1. MÃ NHỊ PHÂN

Mã nhị phân là loại mã mà các bit của nó có trọng số là : $1 - 2 - 4 - 8 - \dots - 2^{n-1}$, bit trẻ nhất ứng với $2^0 = 1$, bit tiếp theo ứng với $2^1 = 2, \dots$, và bit già nhất ứng với 2^{n-1} .

8_2.2. MÃ GRAY

Mã Gray là loại mã không có trọng số, khoảng cách Hamming giữa 2 từ mã kề nhau bằng một hay nói cách khác 2 từ mã kề nhau chỉ khác nhau một biến.

8_2.3. MÃ BCD (BINARY CODED DECIMAL)

Mã BCD là mã nhị phân mã hoá số thập phân.

Bảng cho ở hình 8-6,a là mã BCD - Normal.

Mã này dùng 4 chữ số nhị phân để mã hoá một chữ số thập phân. Nhóm 4 chữ số nhị phân này gọi là decard. Để mã hoá các số thập phân nhiều chữ số, người ta dùng các decard khác nhau, ví dụ decard hàng đơn vị, hàng chục, hàng trăm, hàng nghìn ...

Ví dụ : mã BCD của số 18 là : 0001.1000

8_2.4. MÃ JOHNSON

Mã Johnson là loại mã có đặc điểm :

- Nếu dùng n biến nhị phân thì sẽ mã hoá được tối đa là 2^n trạng thái.

Số thập phân	Mã nhị phân
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

(a)

- Hai từ mã kề nhau chỉ khác nhau ở một biến.

- Trong bảng mã các bit bằng 1 được "đẩy" dần lên từ bit trẻ nhất đến bit già nhất và khi đã "đẩy" hết thì nó lại với dần đi từ bit trẻ nhất.

A B	A B C	A B C D	A B C D E
0 0	0 0 0	0 0 0 0	0 0 0 0 0
1 0	1 0 0	1 0 0 0	1 0 0 0 0
1 1	1 1 0	1 1 0 0	1 1 0 0 0
0 1	1 1 1	1 1 1 1	1 1 1 1 0
	0 1 1	0 1 1 1	1 1 1 1 1
	0 0 1	0 0 1 1	0 1 1 1 1
		0 0 0 1	0 0 1 1 1
			0 0 0 1 1
			0 0 0 0 1

Hình 8-6. a) Mã BCD - Normal ;

b) Bảng mã Johnson.

Ví dụ bảng mã Johnson cho 2, 3, 4 và 5 biến nhị phân được cho ở hình 8-6, (với qui ước A là bit có trọng số nhỏ nhất).

8_2.5. MÃ VÒNG

Mã vòng có đặc điểm :

- Nếu dùng n biến nhị phân thì mã hoá được n trạng thái .
- Hai từ mã kế nhau luôn luôn khác nhau ở 2 biến .
- Trong từ mã chỉ có duy nhất 1 bit bằng 1, các bit khác bằng 0 . Chữ số 1 được dịch từ bit trẻ nhất đến bit già nhất tạo thành một vòng khép kín.

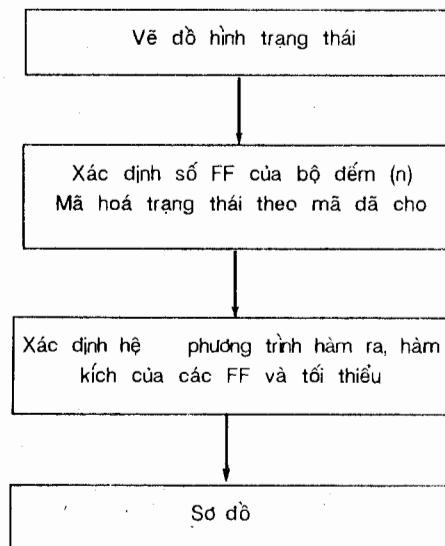
Ví dụ : mã vòng 6 bit có các từ mã như bảng sau (với qui ước A là bit có trọng số nhỏ nhất).

A	B	C	D	E	F
1	0	0	0	0	3
0	1	0	0	0	6
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

Hình 8-6. c) Bảng mã vòng 6 bit.

8_3. CÁC BƯỚC THIẾT KẾ BỘ ĐẾM

Các bước thiết kế bộ đếm nói chung cũng giống như các bước thiết kế mạch dây và được mô tả như hình 8-7.



Hình 8-7. Các bước thiết kế bộ đếm.

Bước 1. Vẽ đồ hình trạng thái của bộ đếm :

Căn cứ vào yêu cầu của bộ đếm cần thiết kế như : hệ số đếm (K_d) và một số các yêu cầu khác để xây dựng đồ hình mô tả hoạt động của bộ đếm.

Bước 2. Xác định số FF của bộ đếm , mã hoá các trạng thái trong của bộ đếm theo mã đã cho.

Trước tiên phải xác định được n là số FF cần thiết để mã hoá cho K_d trạng thái trong của bộ đếm, n phải thoả mãn điều kiện sau :

- Đối với mã nhị phân và mã Gray : $n \geq \log_2 K_d$
- Đối với mã vòng : $n = K_d$
- Đối với mã Johnson : $n = (1/2) K_d$

Sau đó tiến hành mã hoá các trạng thái trong của bộ đếm theo mã đã cho (mã nhị phân, mã Gray, mã vòng, mã Johnson, ...).

Ví dụ : Xây dựng bộ đếm nhị phân có $K_d = 4$, gọi n là số FF thiết để mã hoá là n : $n \geq \log_2 4 \Rightarrow n = 2$. Như vậy phải dùng 2 FF để mã hoá các trạng thái trong cho bộ đếm. Sau đó tiến hành mã hoá trạng thái trong của bộ đếm theo mã nhị phân như sau : 00, 01, 10, 11

Bước 3. Xác định hàm kích của các FF và hàm ra :

Phương pháp xác định hàm kích cho các FF và hàm ra của bộ đếm có thể xác định theo hai cách sau :

- Dựa vào bảng chuyển đổi trạng thái, bảng ra để xác định các phương trình đầu vào kích cho các FF và phương trình của hàm ra.
- Dựa trực tiếp vào đồ hình chuyển đổi trạng thái viết các phương trình đầu vào kích cho các FFs và phương trình hàm ra.

Như đã biết, bộ đếm là một mạch dây có một tín hiệu ra và tín hiệu ra này chỉ xuất hiện ($Y = 1$) khi bộ đếm chuyển từ trạng thái trong $K_d - 1$ về trạng thái 0 (đối với bộ đếm thuận) và có tín hiệu vào đếm X_d . Do vậy việc xác định hàm ra của bộ đếm rất đơn giản. Trong các phần trình bày tiếp theo của chương này chỉ xét đến việc xác định các đầu vào kích cho FF_s .

Bước 4. Sơ đồ mạch thực hiện :

Từ các phương trình đầu vào kích các FF và phương trình hàm ra, đưa ra sơ đồ mạch thực hiện.

Để làm rõ các bước thiết kế bộ đếm, chúng ta hãy xét các ví dụ cụ thể trong chương này, trong đó luôn ký hiệu :

A là bit có trọng số nhỏ nhất	2^0
B là bit có trọng số	2^1
C là bit có trọng số	2^2
...	

8_4. BỘ ĐẾM THUẬN, NHỊ PHÂN, ĐỒNG BỘ VỚI HỆ SỐ ĐẾM $K_D = 2^n$

8_4.1. THIẾT KẾ BỘ ĐẾM NHỊ PHÂN ĐỒNG BỘ CÓ $K_D = 2$

$K_d = 2$, lập được đồ hình trạng thái như hình 8-8,a. Do $K_d = 2$, cho nên chỉ cần 1 FF để mã hoá cho hai trạng thái trong của bộ đếm, gọi FF đó là A.

Bảng chuyển đổi trạng thái và các giá trị đầu vào kích cho các FF như hình 8-8,b.

Nếu dùng $RS - FF$ để xây dựng bộ đếm :

Lập bảng Karnaugh cho các hàm R_A, S_A và xác định được :

$$S_A = \bar{A}$$

$$R_A = A.$$

Sơ đồ mạch thực hiện chỉ ra ở hình 8-8,c.

Nếu dùng $JK - FF$ để xây dựng bộ đếm, tương tự như trên cũng xác định được K_A, J_A :

$$K_A = J_A = 1$$

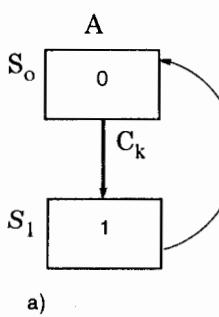
Sơ đồ mạch thực hiện được chỉ ra ở hình 8-8,d.

Tương tự nếu dùng $D - FF, T - FF$ để xây dựng bộ đếm thì chúng ta có :

$$D_A = \bar{A}$$

$$J_A = 1$$

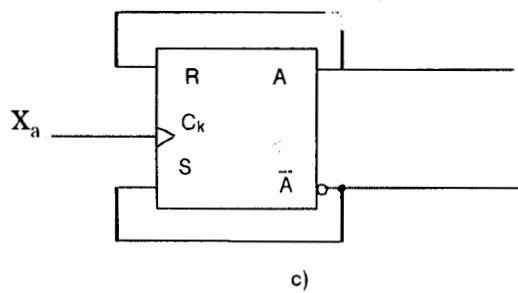
và sơ đồ mạch thực hiện như hình 8-8,e, f.



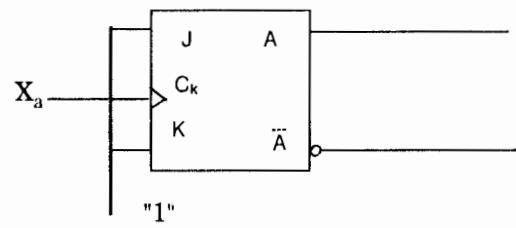
a)

S	A	A'	$R_A - S_A$	$K_A - J_A$	D_A	\bar{A}
S_o	0	1	0 1	X 1	1	1
S_1	1	0	1 0	1 X	0	1

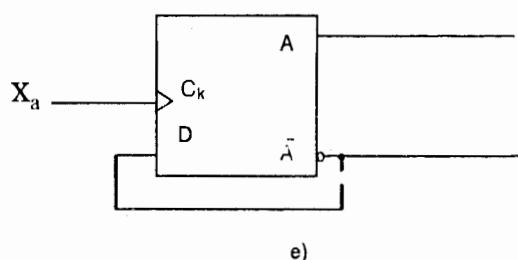
b)



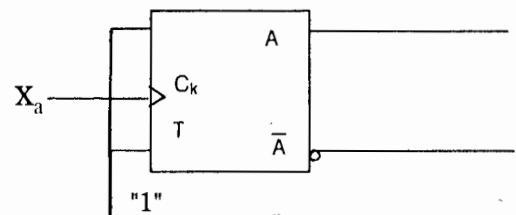
c)



d)



e)



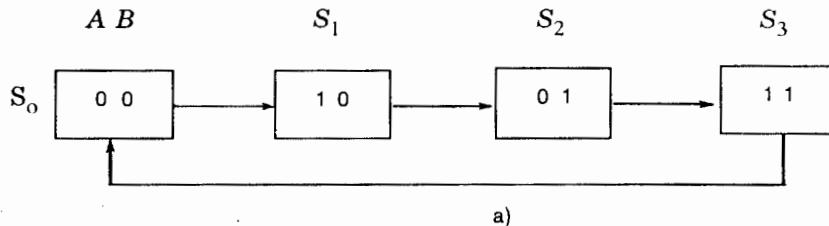
f)

Hình 8-8. Bộ đếm thuận , nhị phân , đồng bộ , $K_d=2$.

- a. Đồ hình trạng thái ; b. Bảng trạng thái và giá trị tương ứng của các đầu vào kích của các FF ;
- c. Sơ đồ dùng $RS - FF$; d. Sơ đồ dùng $JK - FF$; e. Sơ đồ dùng $D - FF$; f. Sơ đồ dùng $T - FF$.

4.2. THIẾT KẾ BỘ ĐẾM NHỊ PHÂN ĐỒNG BỘ CÓ $K_D = 4$

Do $K_d = 4$ lập được đồ hình chuyển đổi trạng thái, xác định được số FF cần dùng để thiết kế bộ đếm ($n = 2$) và mã hoá các trạng thái đó như trên hình 8-9,a ,trong đó 2 FF cần thiết để mã hoá các trạng thái là A và B .

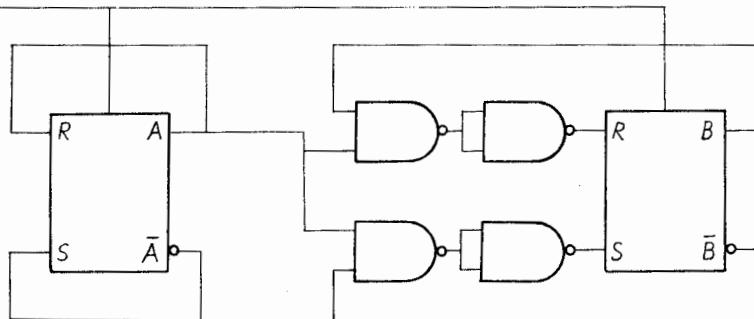


a)

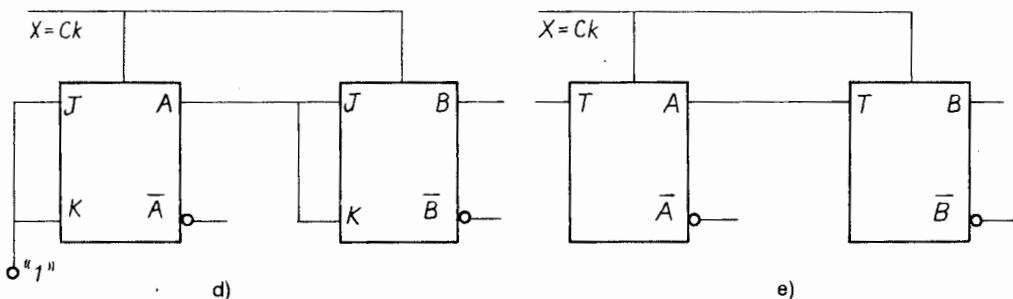
n	n + 1	Cho FF - A					Cho FF - B								
		A	B	A'	B'	R _A	S _A	K _A	J _A	T _A	R _B	S _B	K _B	J _B	T _B
0	0	0	0	1	0	0	1	X	1	1	X	0	X	0	0
1	0	1	0	0	1	1	0	1	X	1	0	1	X	1	1
0	1	0	1	1	1	0	1	X	1	1	0	X	0	X	0
1	1	1	1	0	0	1	0	1	X	1	1	0	0	X	1

b)

$X = Ck$



c)



Hình 8-9. Bộ đếm thuận nhị phân, đồng bộ, $K_d=4$.

- a. Đồ hình trạng thái ; b. Bảng trạng thái và giá trị tương ứng của các đầu vào kích của các FF ; c. Sơ đồ dùng RS - FF ; d. Sơ đồ dùng JK - FF ; e. Sơ đồ dùng T - FF.

Từ đó lập được bảng chuyển đổi trạng thái và các giá trị đầu vào kích cho từng loại *FF* như hình 8-9,b. Tối thiểu hoá hàm kích của các *FF*, nhận được kết quả :

Đối với *FF - A* :

$$\begin{aligned} R_A &= A ; & S_A &= \bar{A} \\ K_A &= J_A = 1 . & & \\ T_A &= 1 & & \end{aligned}$$

Đối với *FF - B* : K_d

$$\begin{aligned} R_B &= A B ; & S_B &= A \bar{B} \\ T_B &= A & & \\ J_B &= K_B = A . & & \end{aligned}$$

Sơ đồ mạch thực hiện ứng với *RS-FF* như ở hình 8-9,c , *JK-FF* (hình 8-9,d) và *T-FF* (hình 8-9,e).

8_4.3. THIẾT KẾ BỘ ĐÉM THUẬN , ĐÔNG BỘ NHỊ PHÂN , $K_D=8$

Tương tự như đối với các trường hợp $K_d = 2, 4$ lập đồ hình trạng thái ,bảng trạng thái và giá trị đầu vào kích cho các *FF*. Tối thiểu hoá các hàm này, nhận được kết quả :

Dùng *RS - FF* :

$$\begin{aligned} R_A &= A & S_A &= \bar{A} \\ R_B &= A B & S_B &= A \bar{B} \\ R_C &= A B C & S_C &= A B \bar{C} \end{aligned}$$

Dùng *T - FF* hoặc *JK - FF* :

$$\begin{aligned} T_A &= J_A = K_A = 1 \\ T_B &= J_B = K_B = A \\ T_C &= J_C = K_C = AB \end{aligned}$$

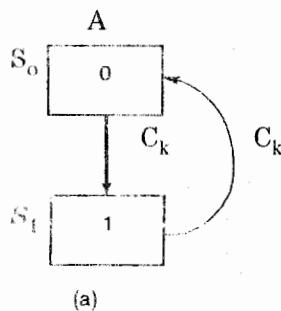
8_4.4. BỘ ĐÉM $K_D=2^n$

Quá trình thiết kế tiến hành tương tự như đã nêu ở phần trước. Trong trường hợp dùng *JK - FF* hoặc *T - FF* để xây dựng bộ đếm này, sẽ nhận được kết quả của các đầu vào kích cho các *FF* như sau :

$$\begin{aligned} T_A &= J_A = K_A = 1 \\ T_B &= J_B = K_B = A \\ T_C &= J_C = K_C = A B \\ T_D &= J_D = K_D = A B C = C J_C = C K_C \\ &\dots \\ T_N &= J_N = K_N = A B C \dots M = M J_M = M K_M . \end{aligned}$$

CÁC BỘ ĐẾM NGHỊCH, NHI PHÂN, ĐỒNG BỘ VỚI HỆ SỐ ĐẾM $K_d = 2^n$ (MAXIMUM LENGTH)

Phương pháp thiết kế hoàn toàn tương tự, chỉ khác với bộ đếm thuận về chiều vẽ trong đồ hình (ngược lại).



(a)

S	A	A'	$R_A - S_A$	$K_A - J_A$	D_A	T_A
S_0	0	1	0 1	X 1	1	1
S_1	1	0	1 0	1 X	0	1

(b)

Hình 8-10. Đồ hình trạng thái và bảng trạng thái của bộ đếm nghịch, $K_d = 2$.

$$K_d = 2$$

Nhận được kết quả :

- Dùng RS - FF : $R_A = A$ $S_A = \bar{A}$
- Dùng JK - FF : $J_A = K_A = 1$
- Dùng T - FF : $T_A = 1$
- Dùng D - FF : $D_A = \bar{A}$

$$K_d = 4$$

Trên hình 8-11 biểu diễn đồ hình, bảng trạng thái và giá trị các đầu vào kích của FF, của bộ đếm nghịch, nhị phân $K_d = 4$.

Thực hiện tối thiểu hóa ta được :

- Dùng JK - FF :

$$J_A = K_A = 1$$

$$J_B = K_B = A$$

- Dùng RS - FF :

$$R_A = A \quad S_A = \bar{A}$$

$$R_B = \bar{A} B \quad S_B = \bar{A} \bar{B}$$

- Dùng D - FF :

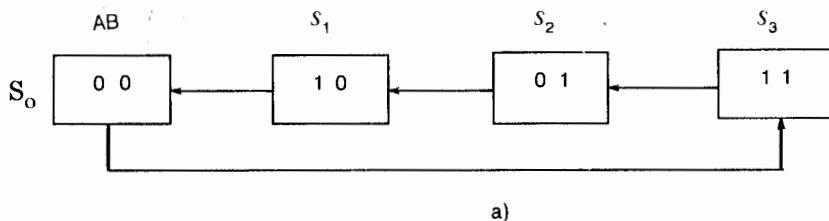
$$D_A = A$$

$$D_B = \bar{A} \bar{B} + A B$$

- Dùng T - FF hoặc JK - FF :

$$T_A = J_A = K_A = 1$$

$$T_B = J_B = K_B = \bar{A}$$



A	B	A'	B'	R _A	S _A	K _A	J _A	T _A	D _A	R _B	S _B	K _B	J _B	T _B	D _B	
0	0	1	1	0	1	X	1	1	1	0	1	X	1	1	1	1
1	0	0	0	1	0	1	X	1	0	X	0	X	0	0	0	0
0	1	1	0	0	1	X	1	1	1	1	0	1	X	1	1	0
1	1	0	1	1	0	1	X	1	0	0	X	0	X	0	1	1

b)

Hình 8-11. Đồ hình, bảng trạng thái và giá trị đầu vào kích thước của các FF
(bộ đếm nghịch, nhị phân, $K_d = 4$).

$$K_d = 2^n$$

Tổng quát ta có :

- Với $JK - FF$:

$$J_A = K_A = 1$$

$$J_B = K_B = \bar{A}$$

$$J_C = K_C = \bar{A} \bar{B} = J_B \bar{B} = K_B \bar{B}$$

$$\dots \\ J_N = K_N = \bar{A} \bar{B} \dots \bar{M} = J_M \bar{M} = K_M \bar{M}$$

- Với $RS - FF$:

$$R_A = A ;$$

$$S_A = \bar{A}$$

$$R_B = \bar{A} B = S_A B ;$$

$$S_B = \bar{A} \bar{B} = S_A \bar{B}$$

$$R_C = \bar{A} \bar{B} C = S_B C ;$$

$$S_C = \bar{A} \bar{B} \bar{C} = S_B \bar{C}$$

$$\dots$$

$$R_N = \bar{A} \bar{B} \dots \bar{M} N = S_M N ; \quad S_N = \bar{A} \bar{B} \dots \bar{M} \bar{N} = S_M \bar{N}$$

- Với $T - FF$:

$$T_A = 1$$

$$T_B = \bar{A}$$

$$T_C = \bar{A} \bar{B} = T_B \bar{B}$$

$$T_N = \bar{A} \bar{B} \dots \bar{M} = T_M \bar{M}$$

8_6. CÁC BỘ ĐẾM GRAY, THUẬN, ĐỒNG BỘ VỚI HỆ SỐ ĐẾM $K_D = 2^n$

Như phần trước đã nêu, mã Gray có các tính chất sau :

- Các từ mã kề cận nhau chỉ khác nhau ở một biến;
- Mã này có tính chất đối xứng.

Các bước thiết kế bộ đếm sử dụng mã Gray hoàn toàn tương tự như mã nhị phân.

Trong phần này sẽ thiết kế bộ đếm chỉ dùng $JK - FF$ và các mạch NAND. Việc xác định phương trình đầu vào kích của các FF dựa trực tiếp vào đồ hình chuyển đổi trạng thái như đã giới thiệu ở mục 7-5, nghĩa là đối với một $FF - i$ loại JK nào đó phải xác định các biểu thức T_{on_i} , T_{off_i} :

$$\begin{array}{ll} T_{on_i} = T^* & Q_i \Rightarrow J_i = T^* \\ T_{off_i} = T^{**} & Q_i \Rightarrow K_i = T^{**} \end{array}$$

Đối với các loại FF khác như $D - FF$, $T - FF$, $RS - FF$ cách thiết kế như đã giới thiệu ở chương 7.

Các ví dụ sau đây sẽ làm sáng tỏ điều này.

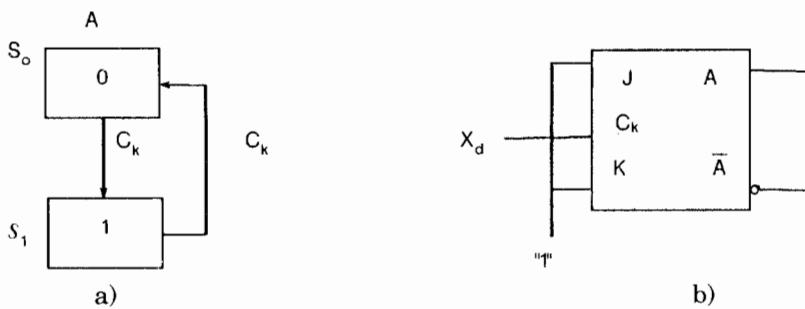
Bộ đếm Gray thuận có $K_d=2$

Đồ hình trạng thái và mã hóa trạng thái tương ứng cho ở hình 8-12,a.

Từ đồ hình viết được các phương trình đầu vào kích của FF như sau :

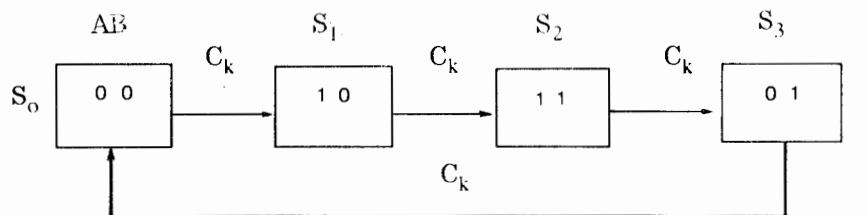
$$\begin{array}{l} T_{on_A} = S_o = A \Rightarrow J_A = 1 \\ T_{off_A} = S_1 = A \Rightarrow K_A = 1 \end{array}$$

Sơ đồ mạch thực hiện như ở hình 8-12,b.

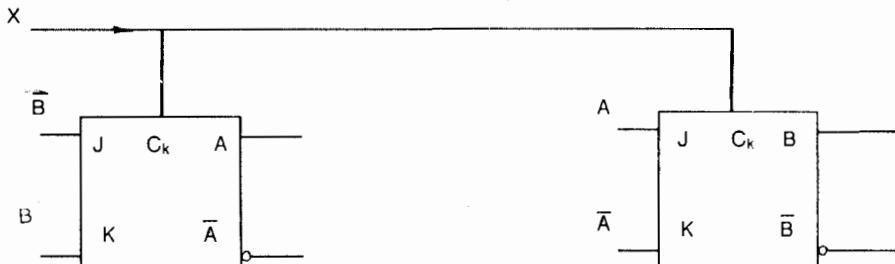


Hình 8-12. a). Đồ hình thực hiện của bộ đếm Gray, thuận, $K_d = 2^n$.

b) Sơ đồ thực hiện của bộ đếm Gray, thuận $K_d=2$.



a)



b)

Hình 8-13.a) Đồ hình trạng thái của bộ đếm Gray, $K_d = 4$;

b) Sơ đồ mạch thực hiện bộ đếm Gray $K_d = 4$.

$$K_d = 4$$

Đồ hình trạng thái và mã hóa như hình 8-13,a.

Từ đồ hình ta có :

$$T_{\text{onA}} = S_0 = \bar{A} \bar{B} \quad \Rightarrow J_A = \bar{B}$$

$$T_{\text{offA}} = S_2 = B A \quad \Rightarrow K_A = B$$

$$T_{\text{onB}} = S_1 = A \bar{B} \quad \Rightarrow J_B = A$$

$$T_{\text{offB}} = S_3 = \bar{A} B \quad \Rightarrow K_B = \bar{A}$$

Sơ đồ mạch thực hiện như hình 8-13,b.

$$K_d = 8$$

Đồ hình trạng thái và mã hóa các trạng thái của bộ đếm như hình 8-14,a. Từ đồ hình viết được các phương trình hàm kích cho các FF :

$$T_{\text{onA}} = S_0 + S_4 = \bar{A} (\bar{B} \bar{C} + BC) \quad \Rightarrow J_A = \bar{B} \bar{C} + BC$$

$$T_{\text{offA}} = S_2 + S_6 = A (B \bar{C} + \bar{B} C) \quad \Rightarrow K_A = B \bar{C} + \bar{B} C$$

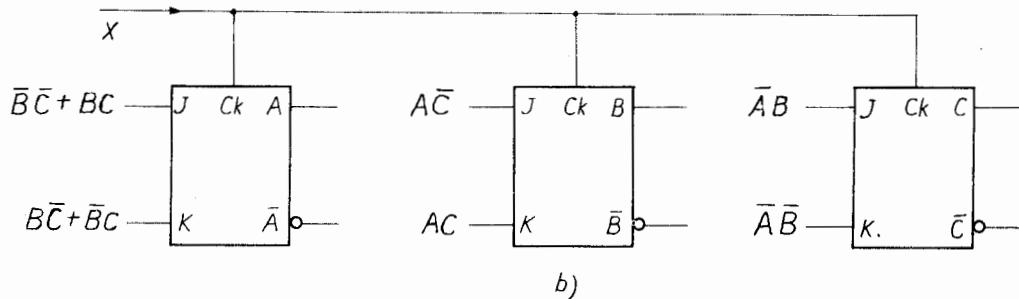
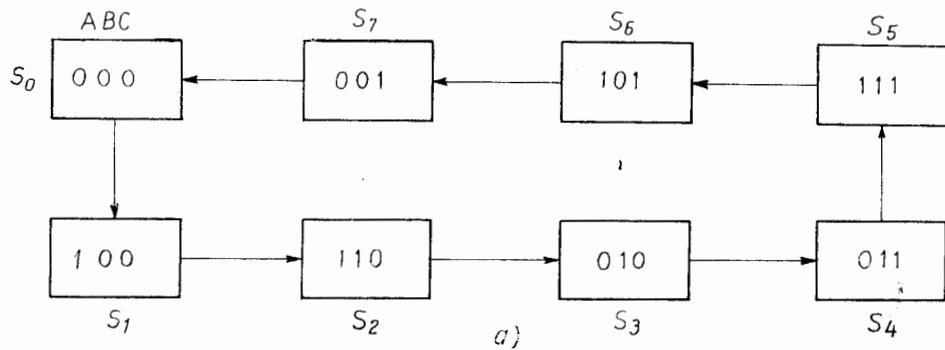
$$T_{\text{onB}} = S_1 = A \bar{B} \bar{C} \quad \Rightarrow J_B = A \bar{C}$$

$$T_{\text{offB}} = S_5 = A B C \quad \Rightarrow K_B = A C$$

$$T_{\text{onC}} = S_3 = \bar{A} B \bar{C} \quad \Rightarrow J_C = \bar{A} B$$

$$T_{\text{offC}} = S_7 = \bar{A} \bar{B} C \quad \Rightarrow K_C = \bar{A} \bar{B}.$$

Từ các phương trình hàm đầu vào kích cho các FF, xây dựng được sơ đồ mạch thực hiện như hình 8-14,b.



Hình 8-14. Bộ đếm . thuận , $K_d = 8$.

a . Đồ hình trạng thái ; b. Sơ đồ mạch thực hiện .

$$K_d = 16$$

Đồ hình trạng thái và mã hoá như hình 8-15,a.

Viết được các phương trình đầu vào cho các FF :

$$T_{onA} = S_0 + S_4 + S_8 + S_{12} \Rightarrow J_A = B C \bar{D} + \bar{B} \bar{C} \bar{D} + \bar{B} C D + B \bar{C} D$$

$$T_{offA} = S_2 + S_6 + S_{10} + S_{14} \Rightarrow K_A = B \bar{C} \bar{D} + \bar{B} C \bar{D} + B C D + \bar{B} \bar{C} D = \bar{J}_A$$

$$T_{onB} = S_1 + S_9 \Rightarrow J_B = A \bar{C} \bar{D} + A C D$$

$$T_{offB} = S_5 + S_{13} \Rightarrow K_B = A C \bar{D} + A \bar{C} D$$

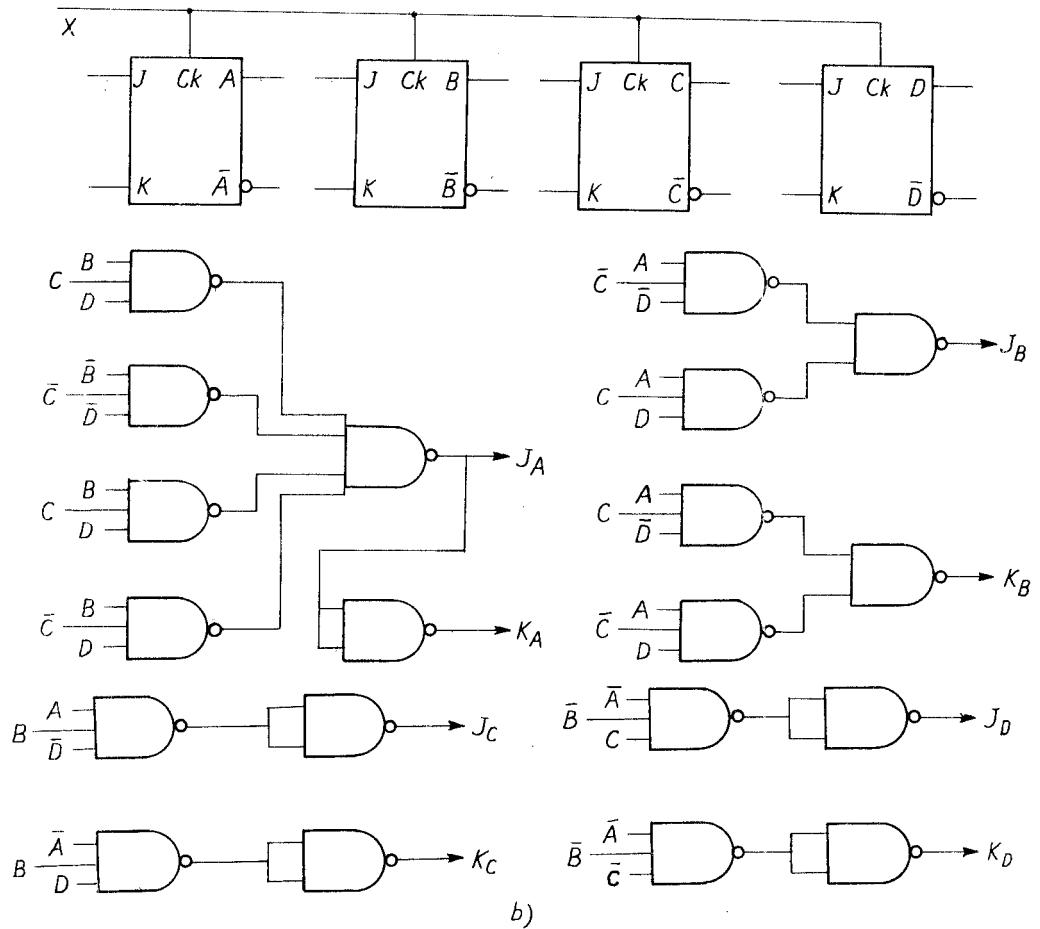
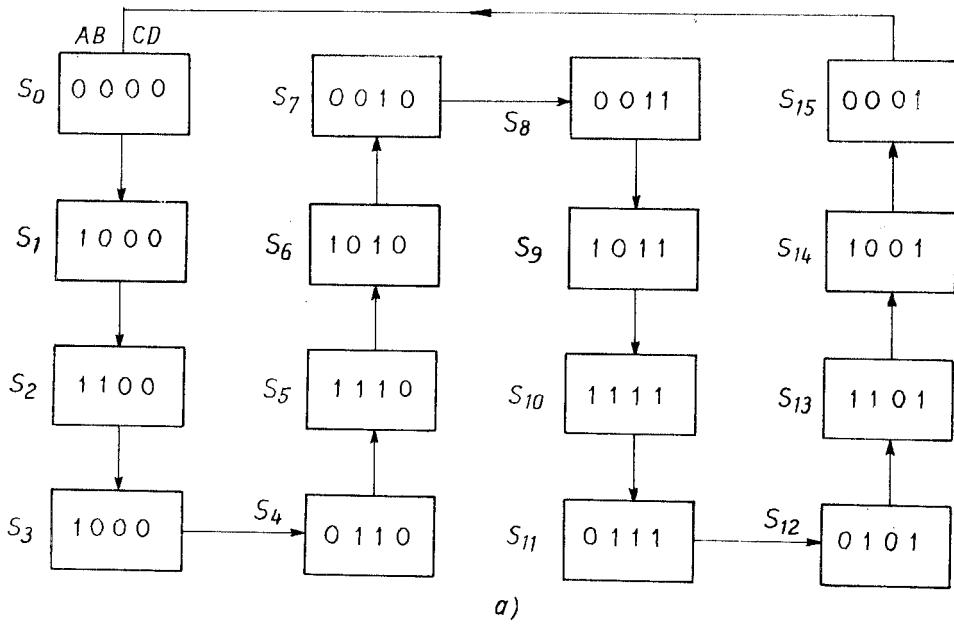
$$T_{onC} = S_3 \Rightarrow J_C = \bar{A} B \bar{D}$$

$$T_{offC} = S_{11} \Rightarrow K_C = \bar{A} B D$$

$$T_{onD} = S_7 \Rightarrow J_D = \bar{A} \bar{B} C$$

$$T_{offD} = S_{15} \Rightarrow K_D = \bar{A} \bar{B} \bar{C}$$

Lập được sơ đồ mạch thực hiện như hình 8-15,b.



Hình 8-15. a) Đồ hình trạng thái và mã hóa các trạng thái của bộ đếm Gray $K_d=2^4$.

b) Sơ đồ bộ đếm đồng bộ, mã Gray, $K_d = 16$.

Bộ đếm Gray nghịch, $K_d = 2^n$.

Áp dụng phương pháp thiết kế đã nêu ở trên với bộ đếm này, kết quả có nhận xét sau :

Muốn đổi chiều bộ đếm Gray $K_d = 2^n$, đồng bộ, chỉ cần đổi chỗ những đầu vào J, K của mỗi FF. Nghĩa là với mỗi $FF - i$ bất kỳ ta có :

Đếm thuận	Đếm nghịch
J_{ith}	$J_{ing} = K_{ith}$
K_{ith}	$K_{ing} = J_{ith}$

Trong đó J_i và K_i là đầu vào J, K của $FF - i$.

8_7. CÁC BỘ ĐẾM ĐỒNG BỘ THUẬN NGHỊCH

Để điều khiển một bộ đếm có thể làm việc cả hai chức năng : khi thì đếm thuận, khi thì đếm nghịch người ta dùng một tín hiệu điều khiển. Giả sử tín hiệu điều khiển đó là R , và qui ước như sau :

$R = 0$: Đếm thuận $R = 1$: Đếm nghịch

Khi đó các phương trình đầu vào kích cho bất kỳ một $FF - i$ nào đó (ví dụ dùng $JK - FF$) sẽ là :

$$\begin{aligned}J_i &= J_{ith} \cdot \bar{R} + J_{ing} \cdot R \\K_i &= K_{ith} \cdot \bar{R} + K_{ing} \cdot R\end{aligned}$$

8_8. CÁC BỘ ĐẾM THẬP PHÂN, ĐỒNG BỘ

8_8.1. KHÁI NIỆM

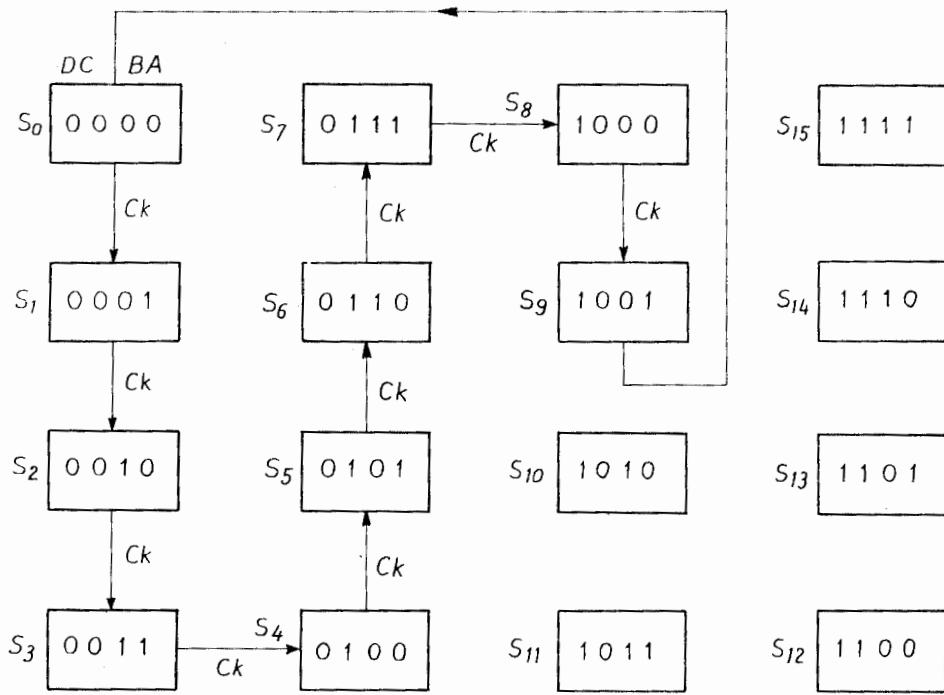
Để xây dựng bộ đếm thập phân có $K_d = 10$ phải dùng ít nhất 4 FF. Khi đó sẽ còn thừa 6 trạng thái không dùng đến (vì $4FF$ mã hoá được tối đa được là $2^4 = 16$ trạng thái). Trong quá trình thiết kế có thể lợi dụng các trạng thái này để sao cho phương trình hàm kích, hàm ra là tối giản.

Các bước thiết kế hoàn toàn tương tự như đối với bộ đếm nhị phân, Gray đã nêu ở trên.

8_8.2. BỘ ĐẾM THUẬN, ĐỒNG BỘ, MÃ NBCD (BCD-NORMAL)

Bộ đếm có đồ hình trạng thái (hình 8-16,a), bảng trạng thái và giá trị tương ứng của các tín hiệu vào kích của các FF được biểu diễn trên hình 8-16,b.

- Với Flip - Flop A : $J_A = K_A = 1$ (thấy ngay từ bảng).



a)

D C B A	D' C' B' A'	J _D K _D	J _C K _C	J _B K _B	J _A K _A
0 0 0 0	0 0 0 1	0 X	0 X	0 X	1 X
0 0 0 1	0 0 1 0	0 X	0 X	1 X	X 1
0 0 1 0	0 0 1 1	0 X	0 X	X 0	1 X
0 0 1 1	0 1 0 0	0 X	1 X	X 1	X 1
0 1 0 0	0 1 0 1	0 X	X 0	0 X	1 X
0 1 0 1	0 1 1 0	0 X	X 0	1 X	X 1
0 1 1 0	0 1 1 1	0 X	X 0	X 0	1 X
0 1 1 1	1 0 0 0	1 X	X 1	X 1	X 1
1 0 0 0	1 0 0 1	X 0	0 X	0 X	1 X
1 0 0 1	0 0 0 0	X 1	0 X	0 X	X 1

b)

Hình 8-18. Bộ đếm thuận, đồng bộ, mã NBCD (BCD- Normal).

- a. Đồ hình trạng thái; b. Bảng chuyển đổi trạng thái và đầu vào kích cho các FF loại JK của bộ đếm thập phân.

Với các FF - B, FF - C, FF - D lập bảng Karnaugh cho J_B, K_B, J_C, J_D, K_D và tối thiểu hóa (hình 8-17).

Vậy ta có : K_A = J_A = 1

$$K_B = A \text{ và } J_B = \bar{A} D$$

$$K_C = J_C = AB$$

$$K_D = A \text{ và } J_D = ABC .$$

		BA	DC	00	01	11	10
		00		1	X		X
		01		1	X		X
		11	X	X	X	X	X
		10			X	X	X

$$J_B = A \bar{D}$$

		BA	DC	00	01	11	10
		00		X	X	1	
		01		X	X	1	
		11	X	X	X	X	X
		10	X	X	X	X	X

$$K_B = A$$

		BA	DC	00	01	11	10
		00			1		
		01	X	X	X		X
		11	X	X	X		X
		10			X		X

$$J_C = A \cdot B$$

		BA	DC	00	01	11	10
		00		X	X	1	
		01				1	
		11	X	X	X	X	X
		10	X	X	X	X	X

$$K_C = A \cdot B$$

		BA	DC	00	01	11	10
		00				1	
		01				X	
		11	X	X	X		X
		10	X	X	X		X

$$J_D = A \cdot B \cdot C$$

		BA	DC	00	01	11	10
		00		X	X	X	
		01	X	X	X	X	X
		11	X	X	X	X	X
		10		1	X		X

$$K_D = A$$

Hình B-17. Bảng Karnaugh cho các đầu vào kích của các JK-FF.

Một trong những phương pháp để bộ đếm có thể loại bỏ sai nhầm (khi rơi vào một trong 6 trạng thái không sử dụng) là : xây dựng sơ đồ cho mạch tạo ra tín hiệu xung nhịp C_K^* .

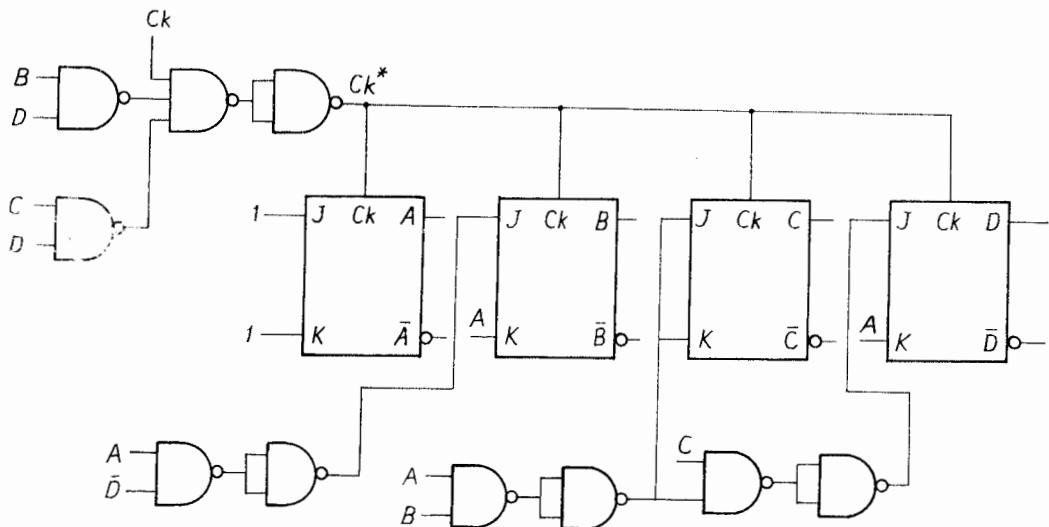
$C_K^* = 1$ khi có xung nhịp C_K và khi bộ đếm không ở một trong những trạng thái không sử dụng.

Sáu trạng thái không sử dụng có phương trình là :

$$BD + CD.$$

$$\text{Vậy } C_K^* = C_K (\overline{BD} + \overline{CD})$$

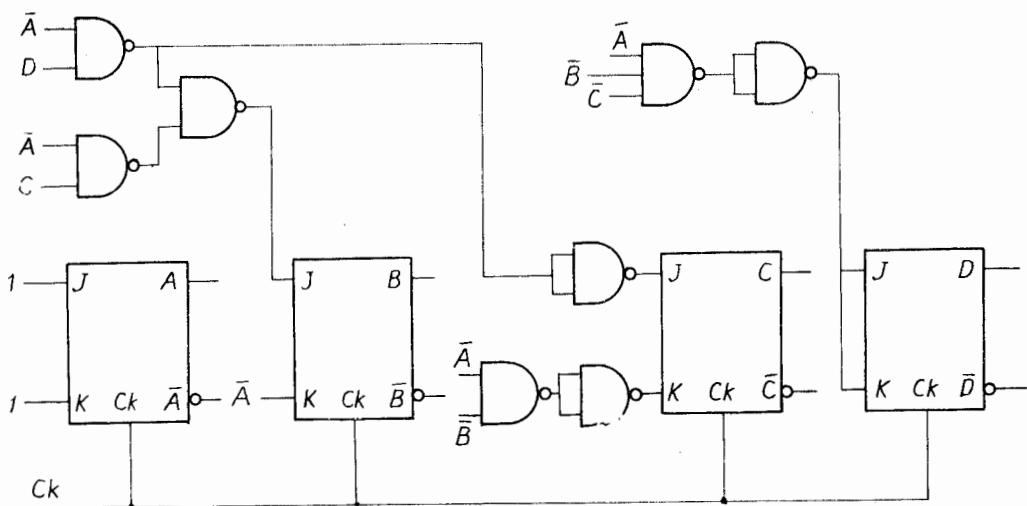
Sơ đồ tương ứng của bộ đếm được biểu diễn trên hình hình 8-18.



Hình 8-18. Sơ đồ bộ đếm thuần, đồng bộ, mã NBCD (BCD-Normal).

8.8.3. BỘ ĐẾM NGHỊCH MÃ NBCD

Với phương pháp tương tự như trên (chiều chuyển biến giữa các trạng thái là ngược lại so với bộ đếm thuần), ta nhận được kết quả :



Hình 8-19. Sơ đồ bộ đếm đồng bộ, thập phân, nghịch, mã NBCD.

$$J_A = K_A = 1$$

$$J_B = \bar{A} C + \bar{A} D$$

$$K_B = \bar{A}$$

$$J_C = \bar{A} D$$

$$K_C = A B$$

$$J_D = \bar{A} \bar{B} \bar{C}$$

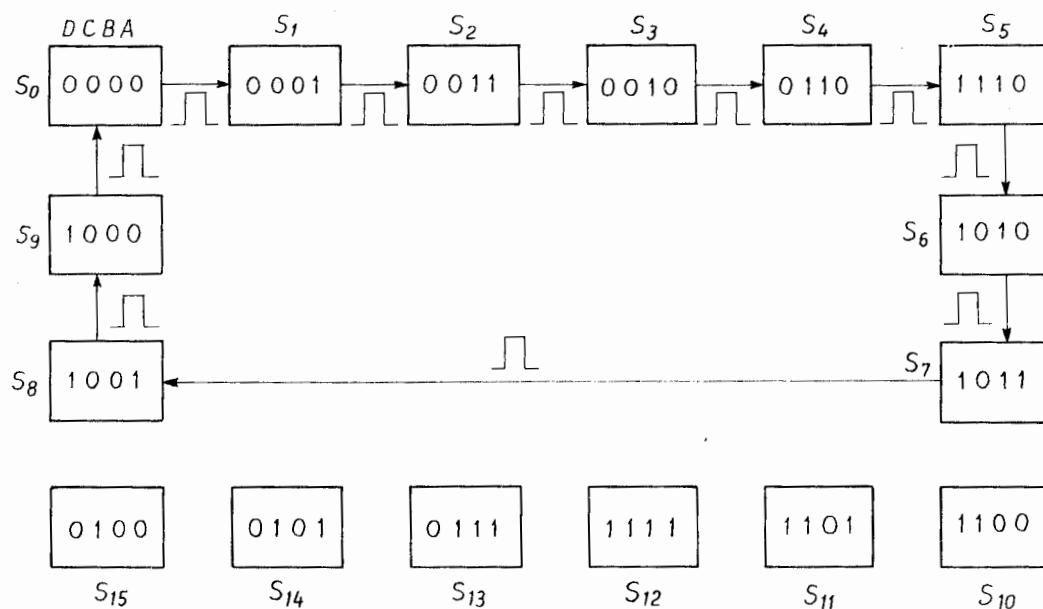
$$K_D = \bar{A} \bar{B} \bar{C} = J_D$$

Từ đó có sơ đồ mạch như hình 8-19.

8.8.4. BỘ ĐẾM ĐỒNG BỘ, THẬP PHÂN, THUẬN, MÃ GRAY

Dồ hình trạng thái của bộ đếm được biểu diễn trên hình 8-20,a. Bảng trạng thái và các đầu vào kích như hình 8-20,b.

Biểu diễn các hàm này trên bảng Karnaugh, và tối thiểu hoá (hình 8-20,c).



a)

Trạng thái hiện tại				Trạng thái tiếp theo				Đầu vào kích thích các FF							
D	C	B	A	D'	C'	B'	A'	J _D	K _D	J _C	K _C	J _B , K _B	J _A	K _A	
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	1	0	X	0	X	1	X	X	0
0	0	1	1	0	0	0	0	0	X	0	X	X	0	X	1
0	0	1	0	0	1	1	0	0	X	1	X	X	0	0	X
0	1	1	0	1	1	1	0	1	X	X	0	X	0	0	X
1	1	1	0	1	0	1	0	X	0	X	1	X	0	0	X
1	0	1	0	1	0	1	1	X	0	0	X	X	0	1	X
1	0	1	1	1	0	0	1	X	0	0	X	X	1	X	0
1	0	0	1	1	0	0	0	X	0	0	X	0	X	X	1
1	0	0	0	0	0	0	0	X	1	0	X	0	X	0	X

b)

Hình 8-20.

		BA	J _D	DC	00	01	11	10
00								
01		X	X	X	1			
11		X	X	X	X			
10		X	X	X	X			

$J_D = C$

		BA	K _D	DC	00	01	11	10
00		X						
01		X		X				
11		X		X				
10		1						

$K_D = \bar{B} \bar{A}$

		BA	J _C	DC	00	01	11	10
00								
01		X	X	X		X		
11		X	X	X		X		
10								

$J_C = B \bar{A} \bar{D}$

		BA	K _C	DC	00	01	11	10
00								
01		X		X				
11		X		X				
10		X		X				

$K_C = D$

		BA	J _B	DC	00	01	11	10
00								
01		1		X				
11		X		X				
10		X		X				

$J_B = A \bar{D}$

		BA	K _B	DC	00	01	11	10
00		X		X				
01		X		X				
11		X		X				
10		X		1				

$K_B = A D$

		BA	J _A	DC	00	01	11	10
00		1		X				
01		X		X				
11		X		X				
10		X		X				

$J_A = \bar{B} \bar{D} + B \bar{C} D$

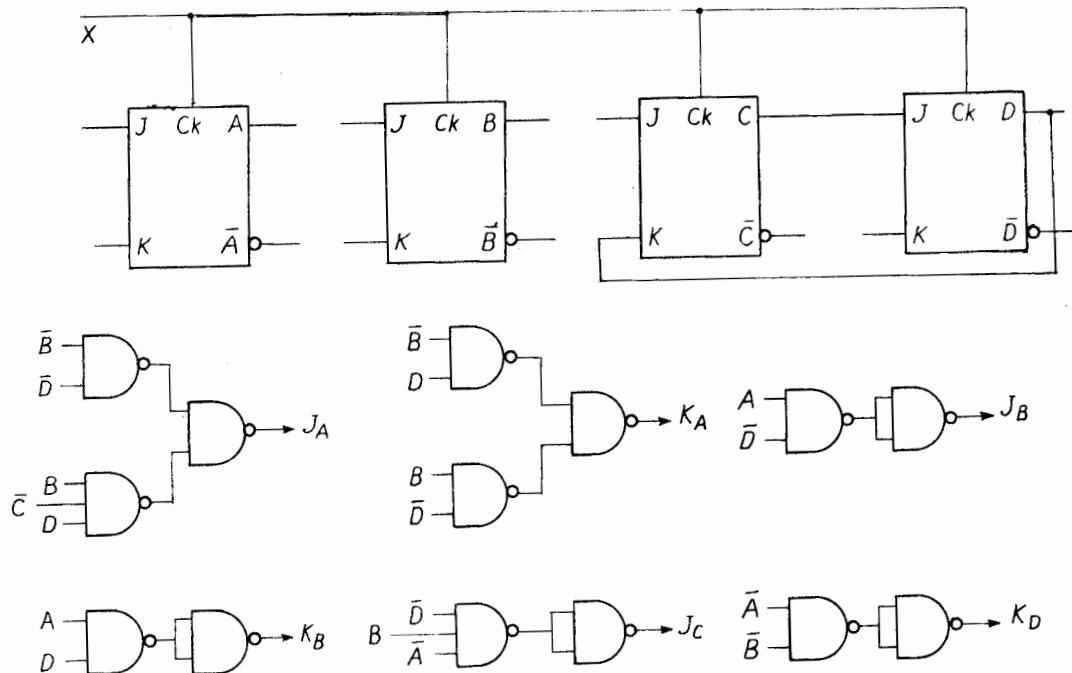
		BA	K _A	DC	00	01	11	10
00		X						
01		X		X				
11		X		X				
10		X		1				

$K_A = D \bar{B} + B \bar{D}$

Hình 8-20. Bộ đếm thập phân, dòng bộ, thuận, mã Gray.

a. Đồ hình trạng thái; b. Bảng trạng thái và hàm kích các FF; c. Bảng Karnaugh cho các đầu vào kích của FF.

Từ hệ phương trình đã tối thiểu hóa nhận được sơ đồ của bộ đếm như ở hình 8.20.



Hình 8-21. Sơ đồ bộ đếm thập phân, đồng bộ, mã Gray.

8-9. BỘ ĐẾM NHỊ PHÂN, KHÔNG ĐỒNG BỘ VỚI $K_d = 2^n$

Các bộ đếm này có sơ đồ rất đơn giản, với đặc điểm :

- Chỉ dùng một loại FF là loại $T - FF$ hoặc $JK - FF$. Nếu dùng $T - FF$ đầu vào T luôn được nối với 1 (luôn ở mức cao "H"), nếu dùng $JK - FF$ thì J luôn nối với K và nối lên 1 (luôn ở mức cao "H") .

- Đầu ra của FF ở tầng trước Q hoặc Q' (FF biểu diễn bit có trọng số nhỏ) luôn được đưa vào đầu xung nhịp cho tầng sau (FF biểu diễn bit có trọng số lớn hơn ngay cạnh đó). Khi đếm thuận lấy đầu ra thuận Q , ngược lại khi đếm nghịch lấy đầu ra nghịch Q (với giả thiết C_k tích cực tại sườn âm \downarrow).

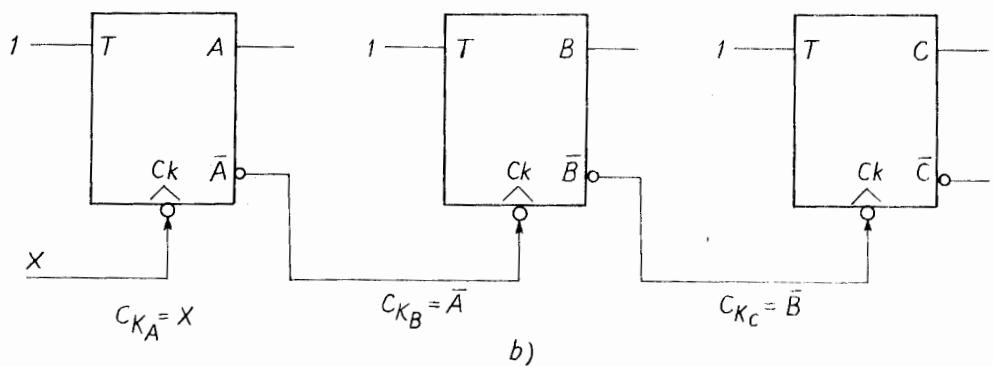
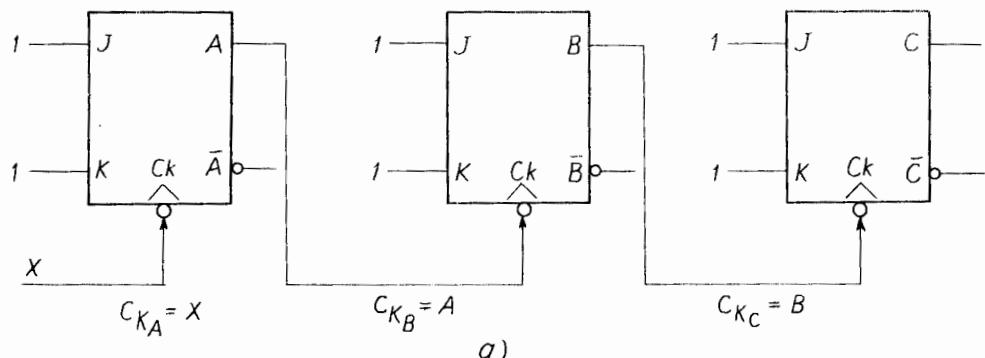
- Tín hiệu vào X_d luôn được đưa vào đầu vào nhịp của FF có trọng số bé nhất.

Ví dụ đối với bộ đếm nhị phân không đồng bộ $K_d = 2^n$ dùng các FF : A, B, C, \dots, N với A là cột trẻ nhất, N là cột có trọng số lớn nhất, ta có :

- Khi đếm thuận : $C_{KA} = X ; C_{KB} = A ; C_KC = B ; \dots C_{KN} = M ;$

- Khi đếm nghịch : $C_{KA} = X ; C_{KB} = A ; C_KC = B ; \dots C_{KN} = M ;$

Sơ đồ của bộ đếm nhị phân, dị bộ, $K_d = 8$, thuận, dùng $JK - FF$ được cho ở hình 8-22,a, đếm nghịch dùng $T - FF$ ở hình 8-22,b.

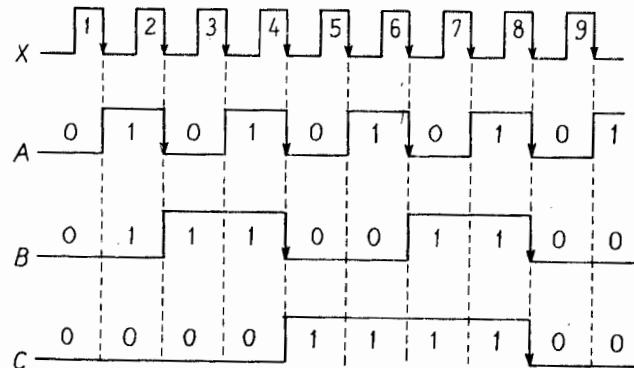


Hình 8-22. Sơ đồ bộ đếm nhị phân, không đồng bộ $K_d = 8$.

a. Đếm thuận dùng JK-FF ; b. Đếm nghịch dùng T-FF.

Dạng sóng ở các đầu ra của các FF của bộ đếm thuận $K_d = 8$ cho ở hình 8-23.

Đồ thị dạng sóng này là đồ thị lý tưởng vì không xét đến sự trễ của các FF. Trong thực tế sự trễ của các FF là không thể bỏ qua được, nó ảnh hưởng đến tần số làm việc của bộ đếm.



Hình 8-23. Dạng sóng trên các đầu ra của các FF A, B, C của bộ đếm thuận nhị phân, đิ bộ $K_d = 8$, C_k tích cực ở suồn sau.

Gọi t_{qd} là thời gian quá độ – trễ của một FF là thời gian từ khi có tác dụng tích cực xung nhịp tới khi FF thay đổi trạng thái.

T là khoảng thời gian giữa hai xung đếm X , hay chu kỳ của X .

n là số Flip – Flop dùng trong bộ đếm.

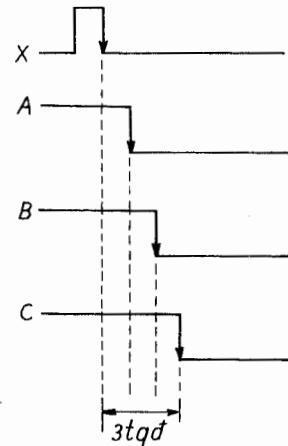
Đối với bộ đếm không đồng bộ, để mạch làm việc tin cậy phải bảo đảm sao cho khoảng thời gian giữa hai xung đếm X phải đủ lớn để ít nhất sau khi cả n Flip – Flop thay đổi trạng thái xung X tiếp theo mới được tác động lên bộ đếm.

Tóm lại ta có biểu thức :

$$f \leq f_{\max} = \frac{1}{T_{\min}} = \frac{1}{n \cdot t_{qd}}$$

Trong đó : f là tần số làm việc của bộ đếm, f_{\max} là tần số làm việc cực đại.

Quá trình chuyển từ trạng thái 111 đến trạng thái 000 trong bộ đếm thuận, không đồng bộ dùng 3 FF : A, B, C , xung nhịp C_k tác động ở sườn âm được biểu diễn trên hình 8-24.



Hình 8-24. Sự chuyển trạng thái của bộ đếm không đồng bộ.

8-10. BỘ ĐẾM THẬP PHÂN, KHÔNG ĐỒNG BỘ

Bảng trạng thái và đồ hình của bộ đếm này tương tự bộ đếm thập phân đồng bộ. Với bộ đếm đồng bộ ta đã có kết quả :

$$J_A = K_A = 1$$

$$J_B = A \bar{D}; \quad K_B = A$$

$$J_C = K_C = AB$$

$$J_D = ABC; \quad K_D = A$$

Các biểu thức J, K như trên đều có nhân thêm với X_d (trong quá trình thiết kế để đơn giản ta đã bỏ qua X_d này).

Đối với bộ đếm đồng bộ, việc nhân thêm với X_d được thực hiện bằng cách đưa tín hiệu X_d vào đầu vào C_k của tất cả các FF. Điều đó có thể lý giải một cách dễ dàng bởi vì ở các FF có C_k , các đầu vào điều khiển chỉ có tác động khi C_k ở tích cực – nghĩa là thực hiện chức năng AND với C_k .

Với bộ đếm dị bộ, sự thay đổi trạng thái của những FF này lại dẫn đến sự thay đổi trạng thái của những FF khác ..., thường chỉ lấy X_d đưa vào C_k của FF đầu tiên : FF – A còn các FF sau được điều khiển bởi các FF trước. Chú ý đến đặc điểm "dây chuyền" này và lưu ý rằng trong biểu thức J, K của cùng một FF phần chung của chúng có thể dùng làm C_k cho FF ấy.

Từ các hàm J , K ở trên cho bộ đếm đồng bộ, ta có phương trình C_k cho từng FF :

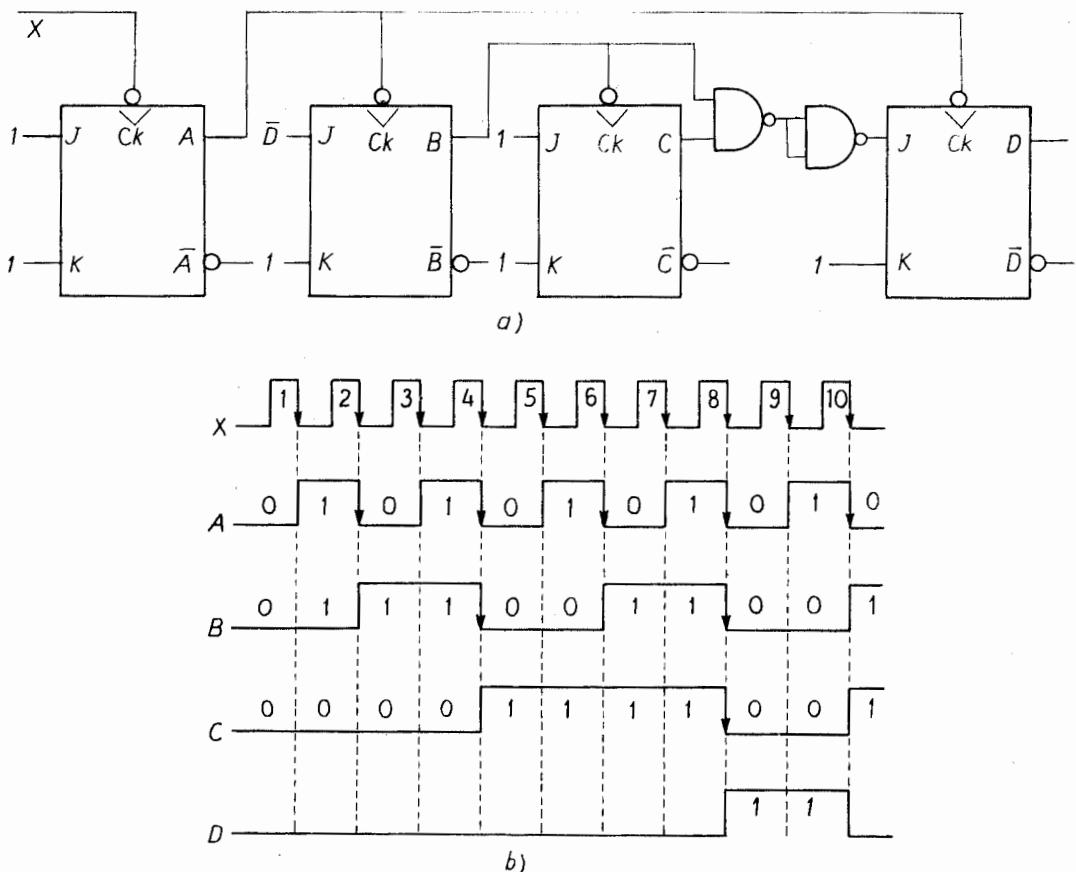
$$C_{kA} = X$$

$$C_{kB} = A$$

$$C_C = B$$

$$C_{kD} = A$$

Sơ đồ mạch thực hiện và dạng sóng trên các đầu ra của các FF được cho ở hình 8-25.



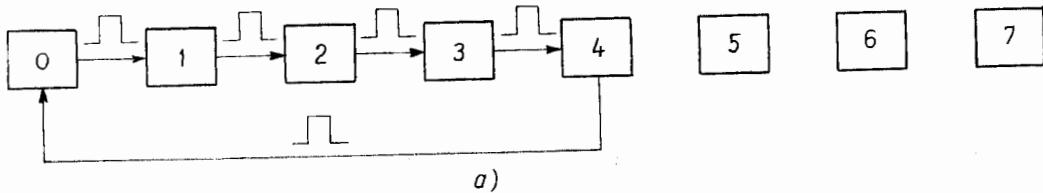
Hình 8-25. Bộ đếm thập phân, dị bộ, thuần.

a. Sơ đồ ; b. Giản đồ dạng sóng của các Flip - Flop.

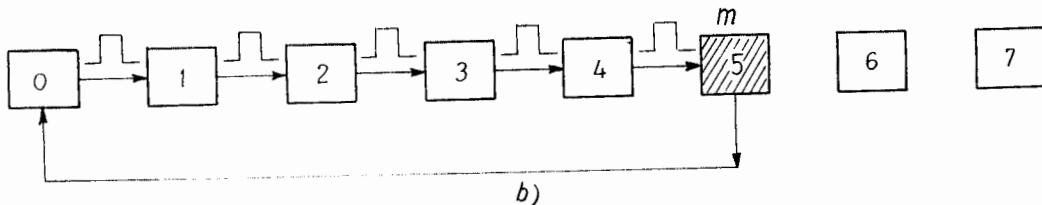
8_11. BỘ ĐẾM ĐẶT LẠI TRẠNG THÁI

Bộ đếm có $K_d = m \neq 2^n$ có thể thực hiện bằng cách :

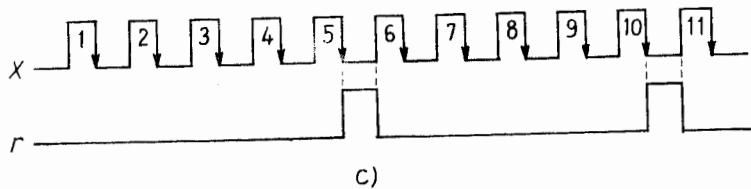
- Thiết kế trực tiếp (bỏ qua những trạng thái không sử dụng như hình 8-26,a).
- Sử dụng bộ đếm có sẵn, có K_d^* : để cho bộ đếm chuyển sang trạng thái m (xem hình 8-26,b), rồi dùng trạng thái này tạo tín hiệu điều khiển r để xoá tất cả các FF trở về trạng thái 0 (trạng thái ban đầu).



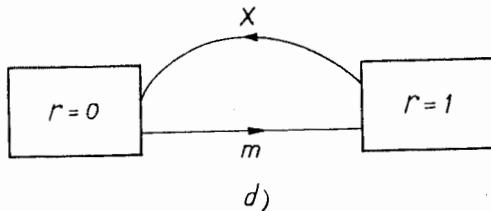
a)



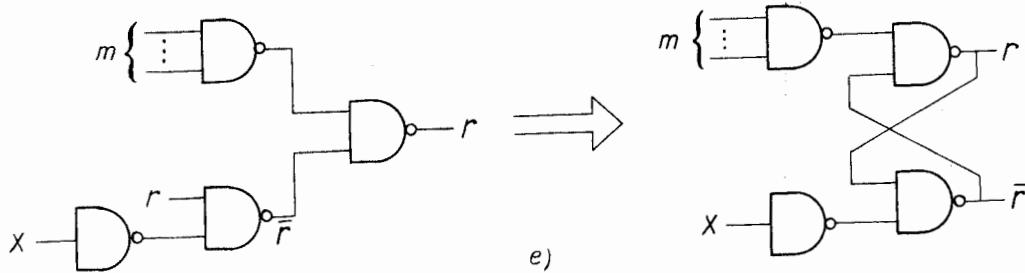
b)



c)



d)



Hình 8-26. Bộ đếm đặt lại trạng thái $K_d = 5$.

- a. Thiết kế trực tiếp ; b. Dùng bộ đếm có K_d để thiết bộ đếm $K_d = m$; c. Giản đồ dạng sóng của tín hiệu đặt lại r ; d. Đồ hình trạng thái của tín hiệu đặt lại r ; e. Mạch tạo tín hiệu đặt lại r .

Thiết kế trực tiếp đã được giới thiệu trong mục 8-3. Tất cả các bộ đếm trong các tiết trước của chương này đã được thiết kế trực tiếp. Trong phần này sẽ trình bày phương pháp thiết kế bộ đếm trên cơ sở sử dụng bộ đếm có sẵn. Với kỹ thuật vi mạch phát triển như hiện nay, cách này được sử dụng ngày càng nhiều.

Ví dụ : Thiết kế bộ đếm có $K_d = 5$ dùng phương pháp đặt lại trạng thái.

- Hình 8-26,c là giản đồ dạng sóng cho tín hiệu r của bộ đếm có $K_d = 5$.

- Hình 8-26,d là đồ hình trạng thái của tín hiệu r (tín hiệu thiết lập trạng thái ban đầu).

- Hình 8-26,e là sơ đồ tạo tín hiệu r dùng mạch NAND. Đầu vào mạch NAND gồm có 11 đầu để đưa X vào và các đầu mã nhị phân của m .

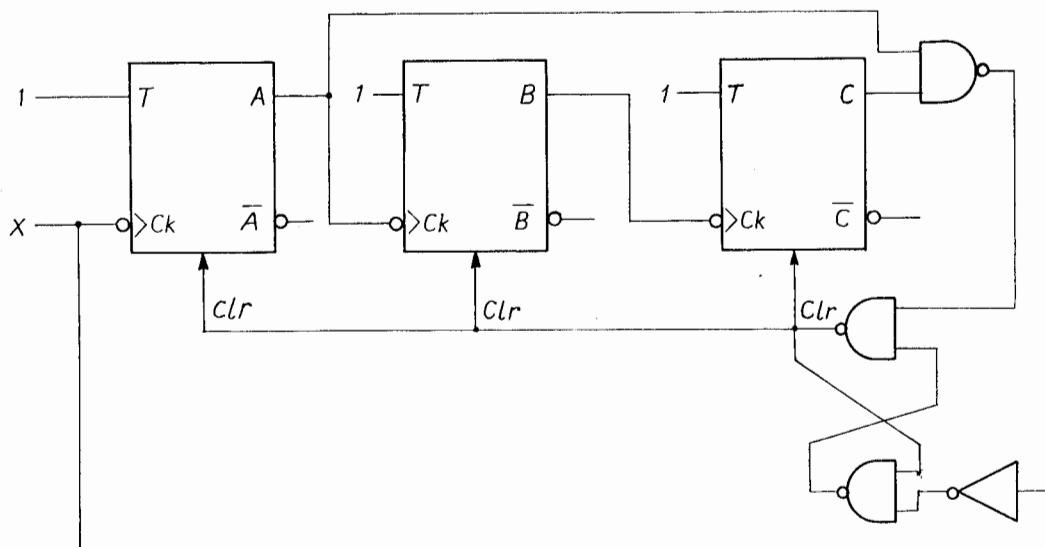
Nếu thiết kế cho bộ đếm thập phân thì $m = 10_{(10)} = 1010_{(2)} = D \bar{C} B \bar{A}$ và nếu chú ý tới các trạng thái không sử dụng thì $m = DB$.

Một số vấn đề cần lưu ý khi thiết kế bộ đếm đặt lại :

- Tín hiệu r phải tồn tại đủ lâu (tối sườn trước của xung nhịp tiếp theo) để tránh gây ra các chuyển biến sai nhầm.

- Đối với các bộ đếm không dùng mã Gray (nghĩa là các trạng thái kế cận nhau có thể khác nhau từ 2 biến trở lên, hay nói cách khác có nhiều FF chuyển biến cùng một lúc) : do thời gian quá độ của các FF nối chung là khác nhau nên dễ tạo ra sai nhầm, có thể sẽ xuất hiện m trong một thời gian ngắn và tạo ra tín hiệu r sai.

Nói chung các bộ đếm sử dụng đồ hình như kiểu 8-26,a là những bộ đếm cố định, trong khi đó các bộ đếm "đặt lại" thích hợp cho các bộ đếm có khả năng thay đổi hệ số đếm hay còn gọi là Programable counter. Việc thay đổi hệ số đếm được thực hiện bằng cách đưa tín hiệu vào điều khiển (m) thích hợp cho mạch tạo tín hiệu r .



Hình 8-27. Bộ đếm $K_d = 5$, không đồng bộ dùng nguyên lý "đặt lại".

Sau đây ta sẽ xét một số ví dụ minh họa cho phương pháp thiết kế bộ đếm dùng phương pháp đặt lại.

Ví dụ 1 :

Bộ đếm thập phân, thuận được tạo thành nhờ bộ đếm thuận, $K_d = 16$, nhị phân, nếu lấy tín hiệu r thiết lập lại 2 Flip-Flop B và D (giả thiết dùng 4 Flip-Flop A, B, C, D với A là bit trẻ nhất). Thực vậy, nếu bộ đếm đang ở trạng thái 1001(9) khi có tín hiệu vào đếm sẽ chuyển sang trạng thái :

1001 (9) \rightarrow 1010 (10) với bộ đếm $K_d = 16$

1001 (9) \rightarrow 0000 (0) với bộ đếm $K_d = 10$.

So sánh với 1010 thì 0000 chỉ khác ở hai bit B và D nghĩa là để xây dựng bộ đếm 10 ta cần dùng mạch của bộ đếm 16 chỉ cần thiết lập lại hai $FF : B$ và D .

Ví dụ 2 :

Bộ đếm thập phân, nghịch được tạo thành từ bộ đếm nghịch $K_d = 16$, nhị phân. Nếu lấy tín hiệu r thiết lập lại hai $FF : B$ và C .

Ví dụ 3 :

Bộ đếm $K_d = 5$ dùng bộ đếm không đồng bộ với đấu vào thiết lập lại có sơ đồ như hình 8-27.

8-12. BỘ ĐẾM JOHNSON (BỘ ĐẾM VÒNG XOĂN)

Dùng mã Johnson, với cách thiết kế như đã trình bày, nhận được kết quả:

- Nếu dùng $D - FF$ thì các $FF : A, B, C, \dots, N$ có phương trình hàm kích như sau:

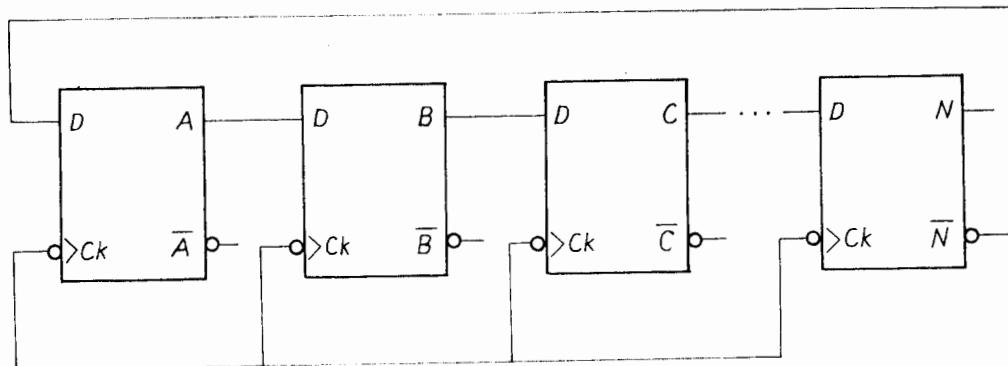
$$D_A = \bar{N}$$

$$D_B = A$$

$$D_B = B \quad \dots$$

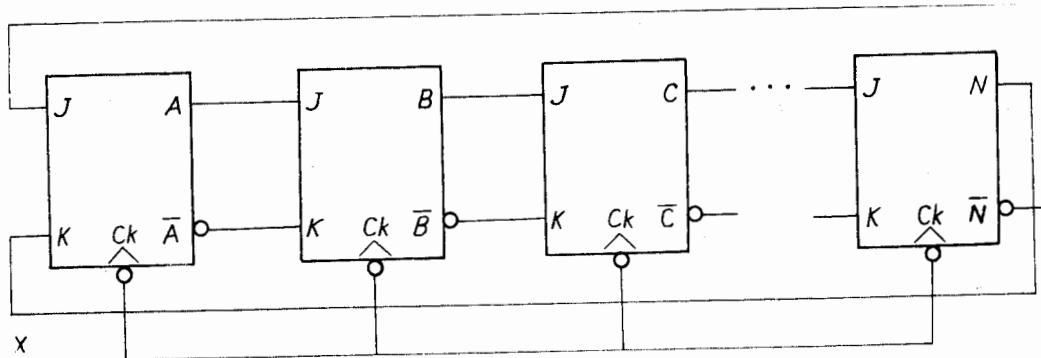
$$D_N = M$$

Sơ đồ mạch thực hiện được cho ở hình 8-28.



Hình 8-28. Bộ đếm Johnson dùng FF loại D .

- Nếu dùng FF loại JK có sơ đồ như hình 8-29.



Hình 8-29. Bộ đếm Johnson dùng FF loại JK .

8_13. BỘ ĐẾM VÒNG

Dùng mã vòng và phương pháp đã nêu để thiết kế.

- Nếu dùng các Flip - Flop loại D để thiết kế có sơ đồ như hình 8-30,a.

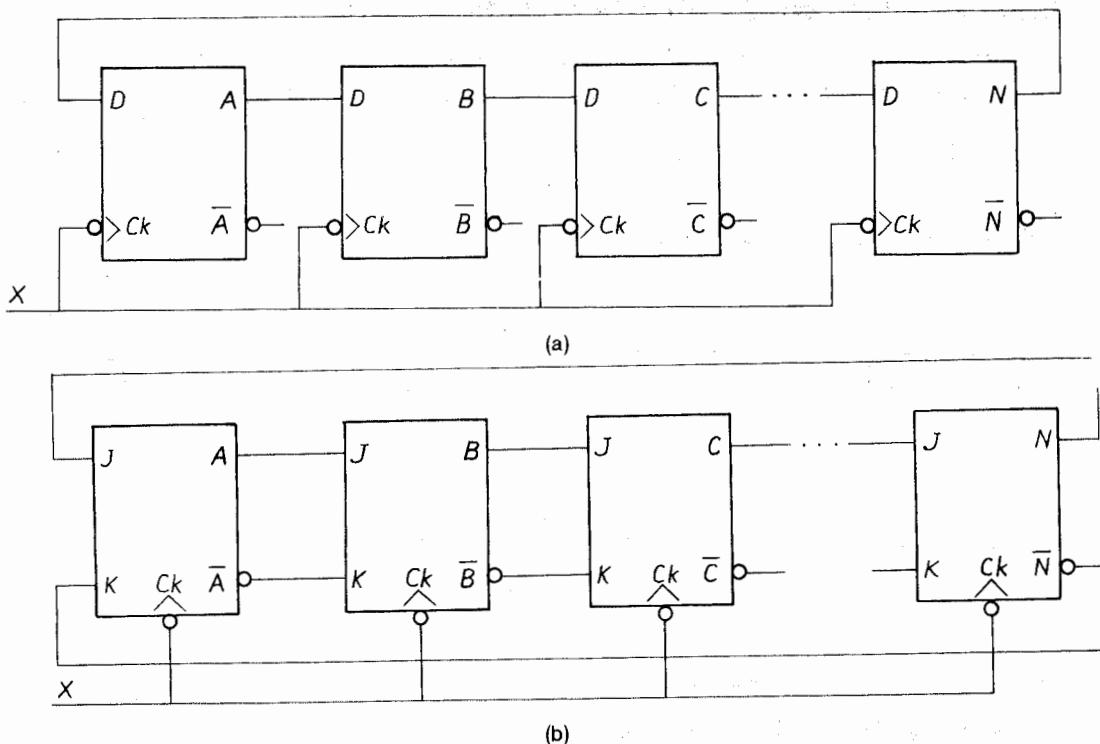
$$D_A = N$$

$$D_B = B \quad \dots$$

$$D_B = A$$

$$D_N = M$$

Nếu dùng JK - FF có sơ đồ như hình 8-30,b.



Hình 8-30. a) Bộ đếm vòng dùng D-FF ; b) Bộ đếm vòng dùng JK-FF.

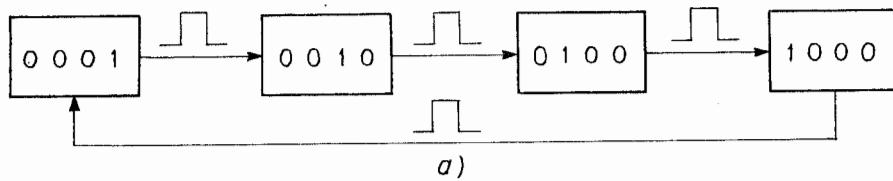
8_14. MỘT SỐ VẤN ĐỀ CẦN CHÚ Ý KHI THIẾT KẾ BỘ ĐẾM

8_14.1. ĐẶT VẤN ĐỀ

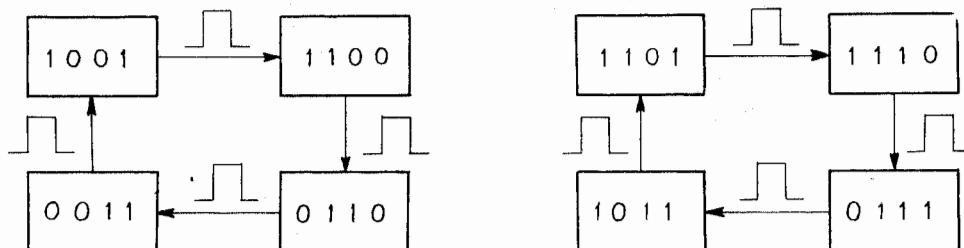
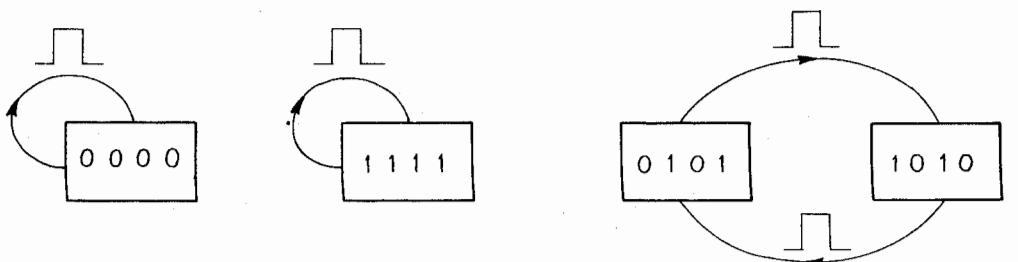
Khi thiết kế bộ đếm, trong các trường hợp không sử dụng hết các trạng thái (khi thiết kế bộ đếm có $K_d \neq 2^n$) sẽ có các trạng thái thừa. Nếu vì một nguyên nhân nào đó mà mạch bị rơi vào một trong các trạng thái thừa đó thì rất có thể mạch nằm mãi ở trạng thái đó không thoát ra được (trạng thái khoá - lock in). Phần này sẽ xét đến các hiện tượng ấy và nêu biện pháp khắc phục.

Ví dụ 1 :

Để thiết kế bộ đếm vòng $K_d = 4$ ta dùng 4 Flip - Flop A, B, C, D với A là bit có trọng số nhỏ nhất, D là bit có trọng số lớn nhất. Sự chuyển trạng thái theo đúng lý thuyết được mô tả ở hình 8-31,a . Còn lại 12 trạng thái không sử dụng, nếu mạch rơi vào một trong 12 trạng thái đó thì theo sơ đồ mạch của bộ đếm vòng (mục 8-13) sẽ có các đồ hình chuyển biến như hình 8-31,b.



a)



b)

Hình 8-31. Đồ hình trạng thái của bộ đếm vòng $K_d = 4$

a. Chu trình đúng; b. Chu trình sai.

Từ đồ hình trạng thái trên hình 8-31 dễ dàng thấy rằng nếu mạch rơi vào 1 trong 12 trạng thái sai đó thì mạch sẽ không trở về chu trình đúng được.

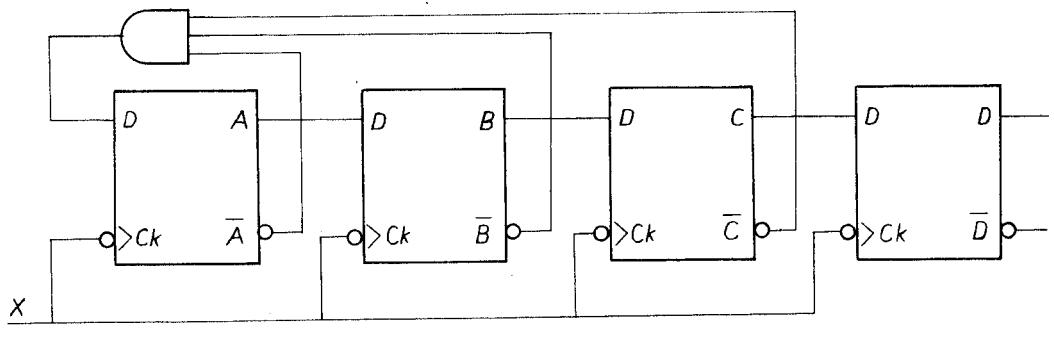
Để khắc phục hiện tượng này người ta đã xây dựng mạch hồi tiếp (mạch khởi động). Ví dụ như với mạch này người ta cắt đường phản hồi $D_A = D$ và thay vào đó là :

$$D_A = \overline{A} \cdot \overline{B} \cdot \overline{C}$$

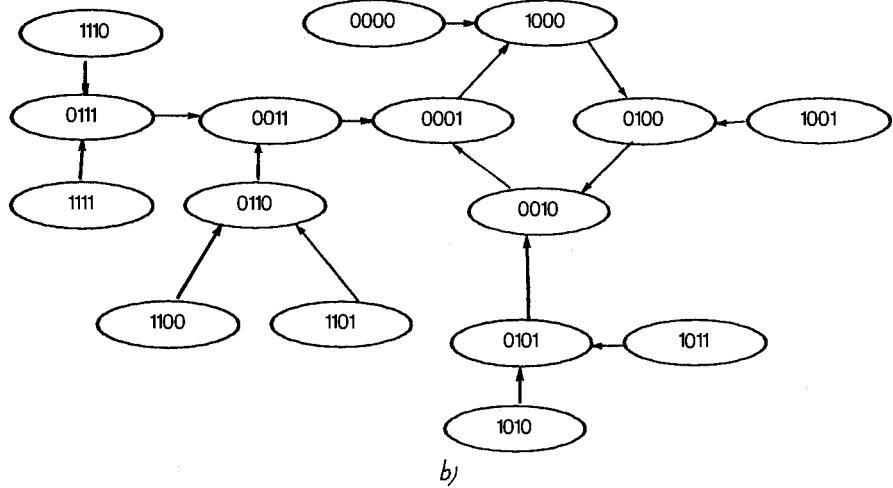
như hình 8-32,a. Đồ hình trạng thái của mạch được mô tả ở hình 8-32,b.

Ví dụ 2 :

Dùng 4 Flip-Flop A, B, C, D để xây dựng bộ đếm vòng xoắn (Johnson) $K_d = 8$. Sơ đồ mạch như hình 8-33,a , đồ hình chuyển biến trạng thái như hình 8-33,b.

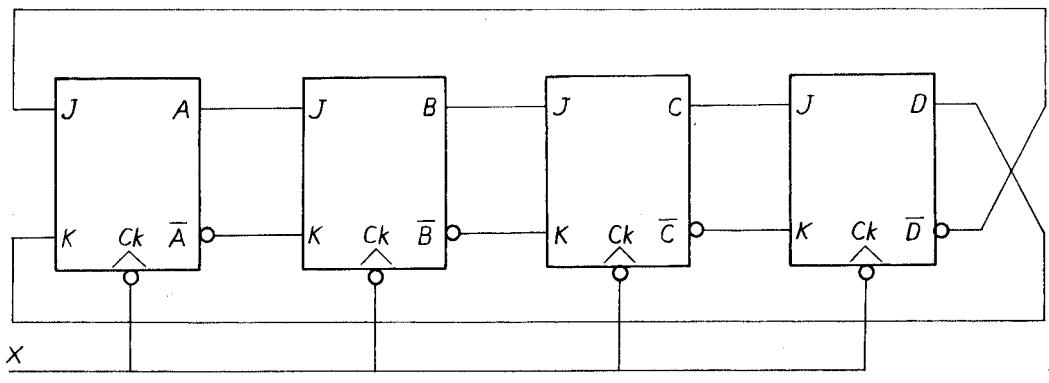


a)



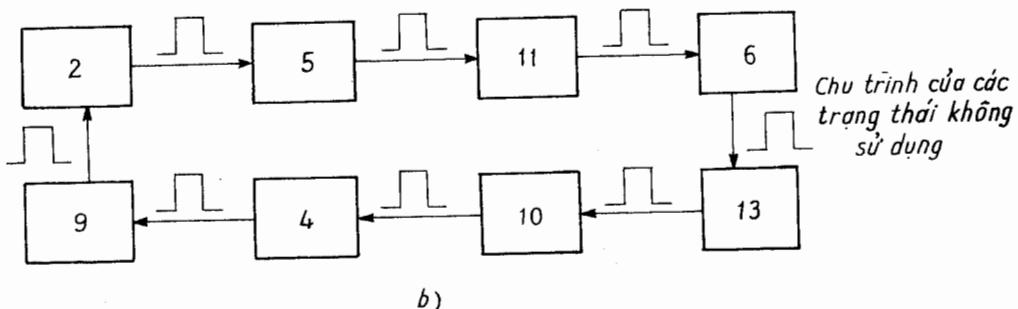
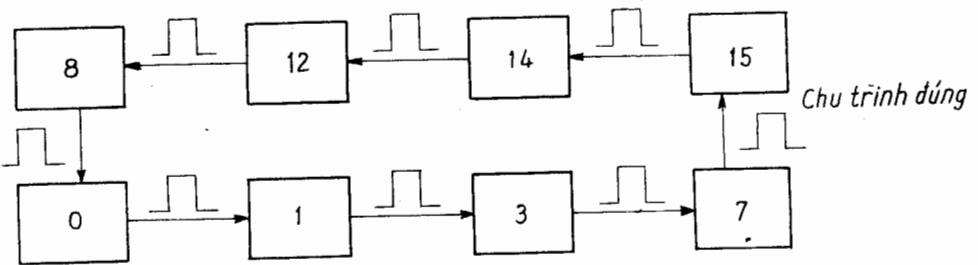
b)

Hình 8-32. Mạch khắc phục trạng thái “chết” của bộ đếm vòng, $K_d = 4$ và đồ hình của nó.



a)

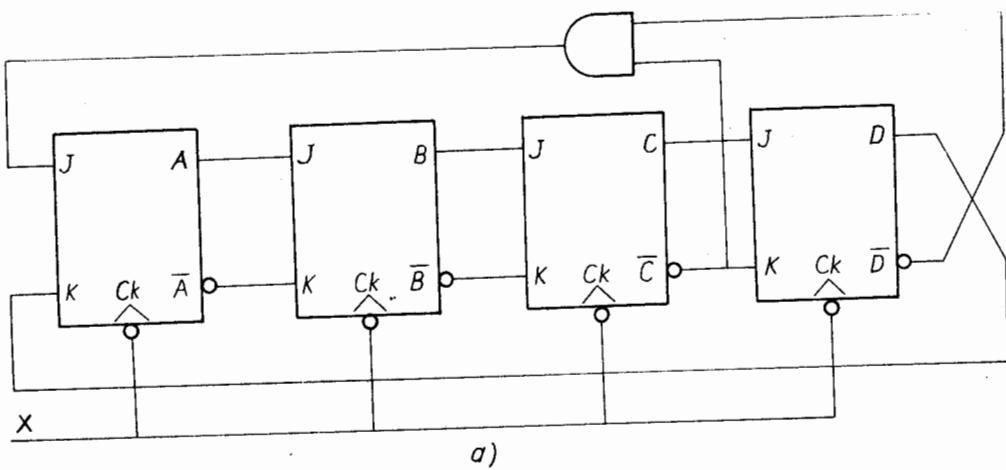
Hình 8-33.

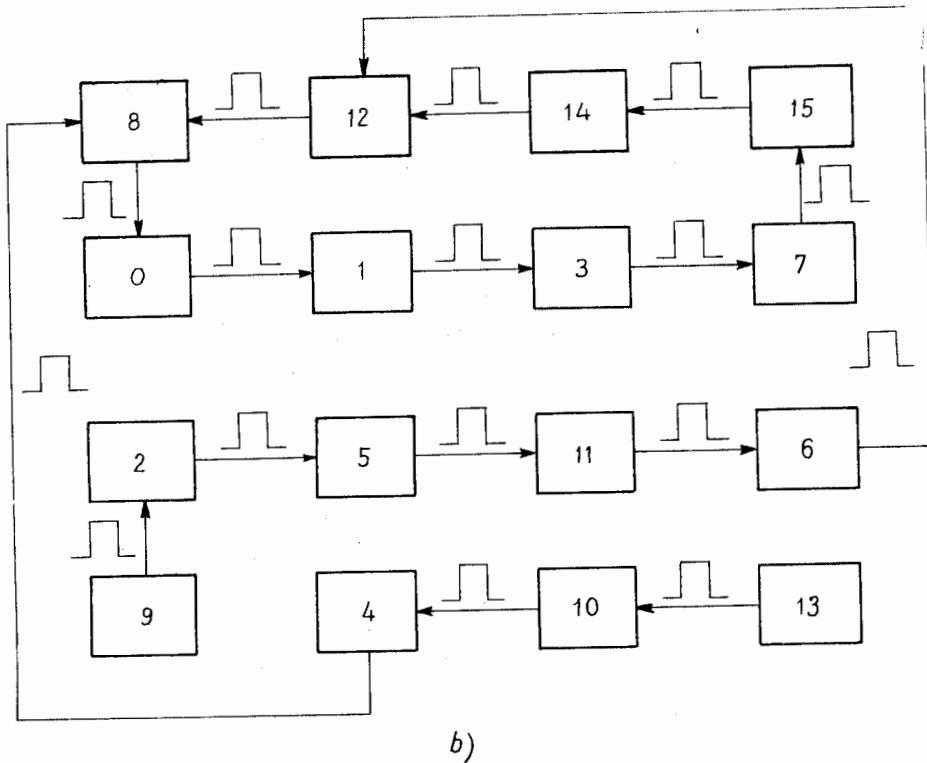


Hình 8-33. Bộ đếm vòng xoắn.

a. Sơ đồ ; b. Đồ hình.

Để khắc phục, người ta thay : $J_A = \bar{C} \cdot \bar{D}$ (hình 8-34,a). Đồ hình trạng thái của mạch sẽ tương ứng như hình 8-34,b.





b)

Hình 8-34. Mạch khắc phục trạng thái "chết" của bộ đếm Johnson $K_d = 8$ và đồ hình trạng thái của nó

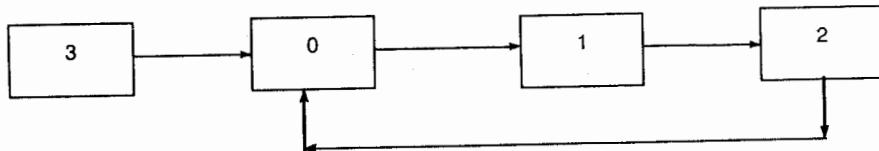
8-14.2. PHƯƠNG PHÁP THIẾT KẾ BỘ ĐẾM CÓ MẠCH KHỞI ĐỘNG

Cách thiết kế bộ đếm có mạch khởi động hoàn toàn giống như thiết kế các bộ đếm thông thường đã trình bày ở các tiết trước. Điều quan trọng là trên đồ hình trạng thái phải chỉ rõ sự chuyển biến từ các trạng thái không dùng đến về trạng thái trong vòng đếm đúng. Trên cơ sở đó thiết kế mạch. Sơ đồ của bộ đếm có mạch khởi động nói chung sẽ phức tạp hơn vì sẽ không còn các trạng thái không xác định, do vậy kết quả tối thiểu hóa nói chung sẽ lớn hơn.

Ví dụ :

Thiết kế bộ đếm đồng bộ, $K_d = 3$, dùng FF loại JK.

Trước hết ta chọn đồ hình trạng thái như hình 8-35.



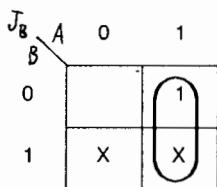
Hình 8-35. Đồ hình trạng thái của bộ đếm $K_d = 3$.

Bảng chuyển đổi trạng thái của bộ đếm và bảng Karnaugh của các đầu vào kích cho FF được biểu diễn trong hình 8-36.a và 8-36.b.

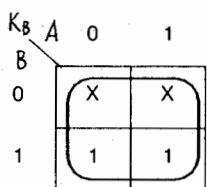
Sơ đồ của bộ đếm được cho ở hình 8.36.c.

B	A	B'	A'	J _A	K _A	J _B	K _B
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	0	0	X	1	0	X
1	1	0	0	X	1	X	1

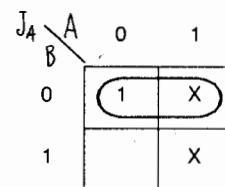
a)



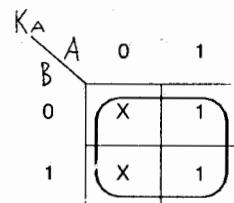
$$J_B = A$$



$$K_B = 1$$

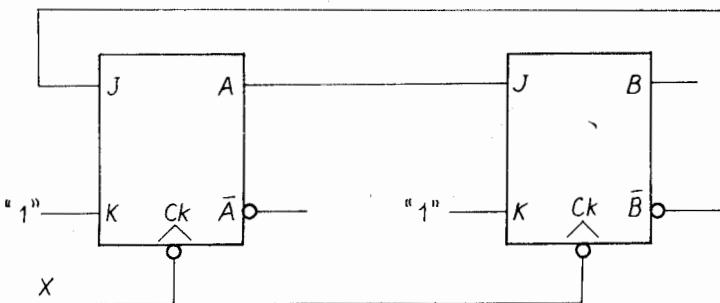


$$J_A = \bar{B}$$



$$K_A = 1$$

b)



c)

Hình 8-36.a) Bảng chuyển đổi trạng thái của bộ đếm;
b) Bảng Karnaugh của các đầu vào kích của FF; c) Mạch thực hiện bộ đếm $K_d = 3$ dùng JK-FF.

8-15. MỘT SỐ VI MẠCH ĐẾM THƯỜNG GẶP

1. VI MẠCH 7490, 74L90, 74LS90

Đây là bộ đếm thập phân gồm 1 bộ đếm 2 (dùng Flip-Flop A) và một bộ đếm 5 (dùng 3 Flip-Flop B, C, D theo thứ tự B là bit trẻ nhất và D là bit già nhất). Q_A, Q_B, Q_C, Q_D là các đầu ra của FF.

Vì mạch này gồm 14 chân, khi tra catalog ta gấp các ký hiệu NC là các chân không dùng.

Ký hiệu H(High) : mức cao

L(Low) : mức thấp

- 4 chân thiết lập : $R_o(1)$, $R_o(2)$, $R_g(1)$, $R_g(2)$
- Khi $R_o(1) = R_o(2) = H$ (ở mức cao) thì bộ đếm được xoá về 0 và các đầu ra ở mức thấp.

- $R_g(1)$ và $R_g(2)$ để thiết lập trạng thái 9 của bộ đếm, 2 đầu này nhất thiết phải có 1 đầu ở mức thấp và 2 đầu $R_o(1)$ và $R_o(2)$ không cùng ở mức cao thì mạch mới đếm.

Khi $R_g(1) = R_g(2) = H$ thì mạch được thiết lập ở trạng thái 9;

- A_{in} : dùng để đưa tín hiệu vào bộ đếm ở bộ đếm 2 ($FF - A$);

- B_{in} : dùng để đưa tín hiệu vào bộ đếm 5 (3 FF : B , C , D).

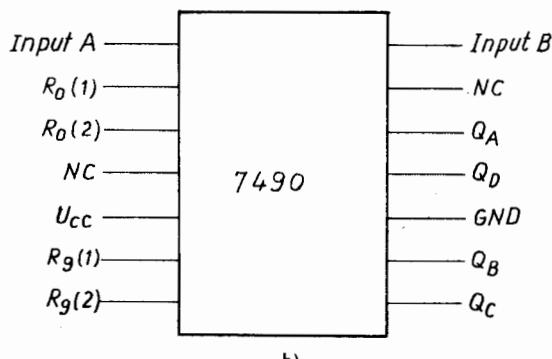
- Khi nối Q_A vào B_{in} và đưa tín hiệu đếm vào A_{in} thì bộ đếm sẽ đếm 10.

Trên hình 8-37 là bảng chân lý và minh họa sơ đồ chân của 7490.

Vì mạch 7490 được mắc mạch để tạo thành bộ đếm $K_d = 6$ như hình 8-38.

Vào thiết lập				R_a			
$R_o(1)$	$R_o(2)$	$R_g(1)$	$R_g(2)$	Q_D	Q_C	Q_B	Q_A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L				dễm
L	X	L	X				dễm
L	X	X	L				dễm
X	L	L	X				dễm

a)

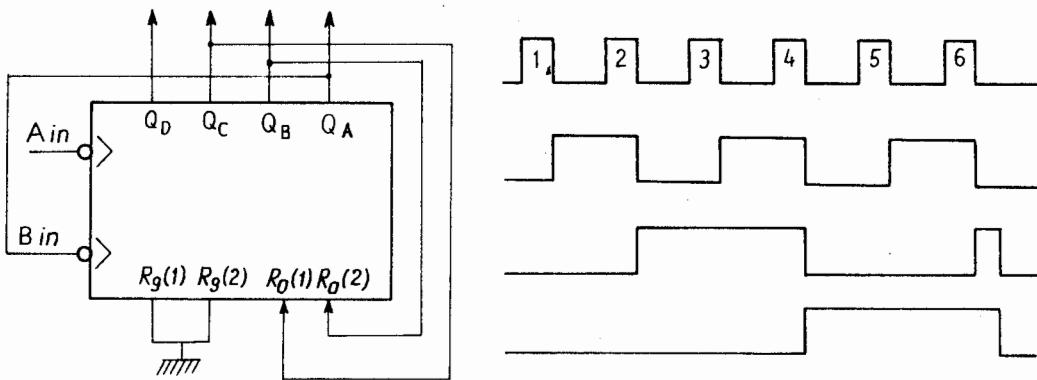


b)

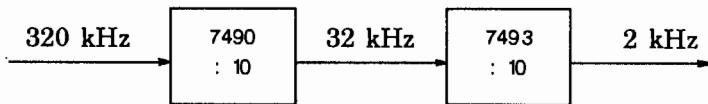
Hình 8-37.a) Bảng chân lý ; b) Sơ đồ chân của 7490.

- Dùng nguyên lý đặt lại ta có thể có được các bộ đếm 2, 4, 6, 8, 10 từ 1 vi mạch 7490, cách làm tương tự bộ đếm 6.

- Người ta cũng hay dùng các vi mạch đếm để chia tần số.



Hình 8-38. 7490 dùng làm bộ đếm 6 và đồ thị dạng sóng các đầu ra Q_A , Q_B , Q_C .

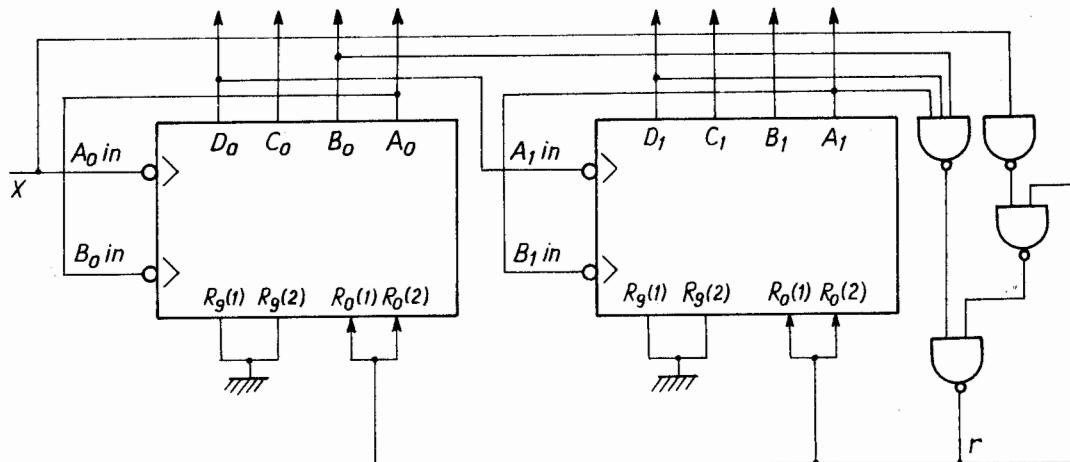


Hình 8-39. Mạch chia tần dùng 7490.

Ví dụ : Bộ đếm 7490 là chia 10 nếu dùng đếm 10 mắc 7490 với 1 vi mạch 7493 (đếm 16) ta có bộ chia tần số (chia 160) như hình 8-39.

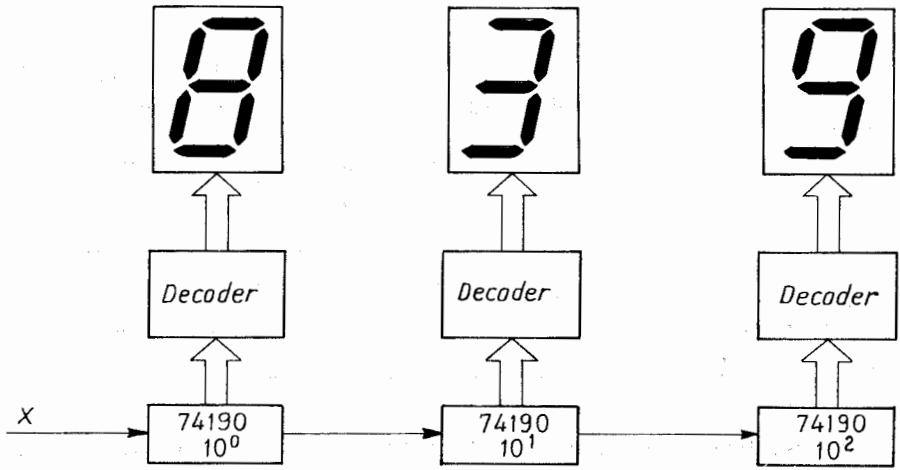
- Người ta dùng các vi mạch 7490 để đếm các số lớn hơn 10.

Ví dụ: Bộ đếm 92 như hình 8-40 dùng 2 vi mạch 7490 nối với nhau.

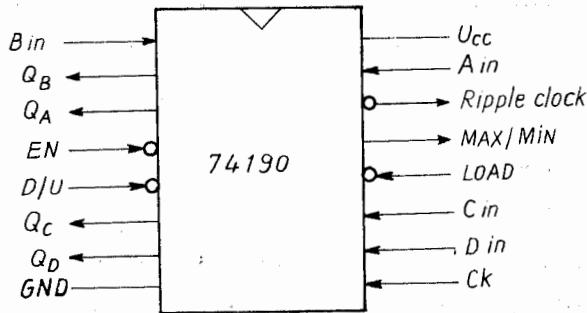


Hình 8-40. Hai vi mạch dùng làm bộ đếm 92.

Trên hình 8-42 là vi mạch đếm thập phân, thuận nghịch, đồng bộ, có đặt trước, có khả năng lập trình (Programmable).



Hình 8-41. Ba bộ đếm 10 dùng 7490 với các mạch hiển thị (giải mã 7 thanh).



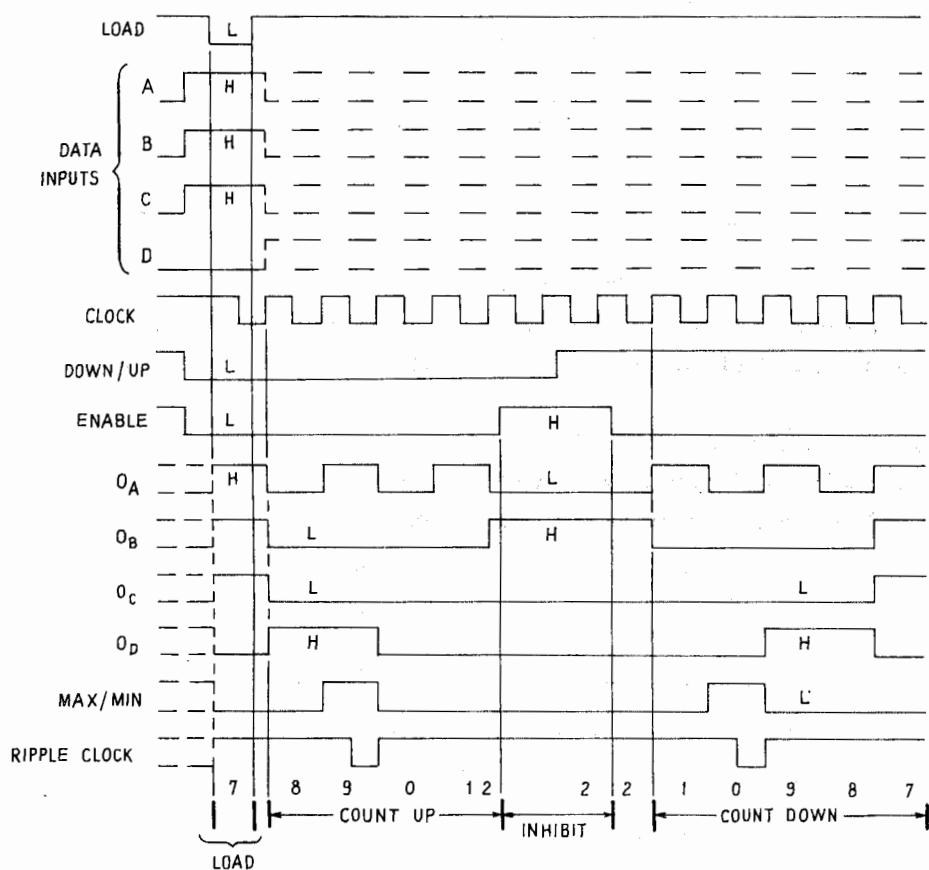
Hình 8-42. Sơ đồ chân của vi mạch 74190.

Chức năng của các chân IC:

- + EN : (Enable) cho phép mạch làm việc ở mức tích cực thấp. Khi EN ở mức thấp thì mạch sẽ đếm.
- + D/U (Down/Up) : điều khiển đếm thuận, nghịch, mức tích cực là mức thấp.
 - $D/U = H$: bộ đếm thuận;
 - $D/U = L$: bộ đếm nghịch.
- + Các đầu A, B, C, D là các đầu vào thiết lập trạng thái ban đầu cho Q_A, Q_B, Q_C, Q_D (in).
 - Q_A là bit đầu ra trễ nhất;
 - Q_D là bit đầu ra già nhất.
- + LOAD : Mức tích cực là mức thấp có tác dụng để nạp các trạng thái ban đầu vào cho Flip – Flop nhờ các đầu vào A, B, C, D.
- + Ripple Clock : là đầu ra, cho mức tích cực thấp khi bộ đếm chuyển từ 9 \Rightarrow 0 (đếm thuận) và từ 0 \Rightarrow 9 (đếm nghịch). Xung ra này có chiều dài bằng độ rộng xung nhịp C_K.

- + C_K : xung nhịp cho các Flip - Flop, tác động bằng sườn dương.
- + Max/Min : Mức tích cực sẽ là mức cao, có độ rộng bằng chu kỳ xung nhịp C_K khi bộ đếm chuyển từ 9 \Rightarrow 0 (đếm thuận) và 0 \Rightarrow 9 (đếm nghịch).

Giản đồ dạng sóng của 74190 được cho ở hình 8-43.



Hình 8-43. Giản đồ dạng sóng của vi mạch 74190.

BÀI TẬP

8.1. Thiết kế bộ đếm đồng bộ $K_d = 12$ dùng NAND và:

- $T - FF$;
- $RS - FF$;
- $JK - FF$;
- $D - FF$.

Xây dựng mạch giải mã cho bộ đếm này.

8.2. Thiết kế một dãy tín hiệu tuần hoàn dùng $JK - FF$ và mạch NAND như bảng trên hình 8-44.

Xung nhịp	C	B	A
1	0	0	1
2	1	0	0
3	0	1	0
4	1	0	1
5	1	1	0
6	0	1	1

Hình 8-44. Cho bài tập 8.2.

a. Kiểm tra chức năng của mạch (ở những trạng thái không sử dụng và chỉ ra rằng một trong những trạng thái đó là trạng thái khóa (lock – in). Chỉ ra cách để tránh trạng thái đó.

b. Vẽ dạng tín hiệu vào, ra ở A, B, C.

8.3. Biến đổi mã nhị phân cho ở bảng trong hình 8-45 thành mã Gray tương ứng với nó, thiết kế bộ đếm dùng JK – FF và NAND dùng mã đó.

P	Q	R	S
0	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
1	1	1	1

Hình 8-45. Cho bài tập 8.3.

8.4. Cho một FF loại JK có chức năng ở bảng hình 8-46.

a. Xác định đầu vào kích của FF này ứng với các chuyển biến trạng thái của FF.

b. Thiết kế bộ đếm có $K_d = 8$ (mã nhị phân) đồng bộ dùng FF loại này và các mạch NAND.

V	U	Q'
0	0	1
0	1	\bar{Q}
1	0	Q
1	1	0

Hình 8-46. Cho bài tập 8.4.

8.5. Cho mạch điện trên hình 8-47.

Mạch này dùng để xung ra Q có có khoảng lớn giữa bằng 14 chu kỳ xung nhịp.

a. Vẽ biểu đồ thời gian của tín hiệu.

b. Thiết kế bộ đếm và bộ điều khiển dùng JK - FF loại chủ từ và các cổng NAND.

8.6. Một bộ đếm loại dị bộ (ripple - counter) dùng 5 FF có thời gian trễ 30 ns, thời gian giải mã 50 ns.

a. Xác định f_{max} của bộ đếm;

b. Nếu bộ đếm làm việc ở f_{max} này, vẽ biểu đồ thời gian của mỗi FF trong bộ đếm khi bộ đếm chuyển từ trạng thái 01111 sang 10000.

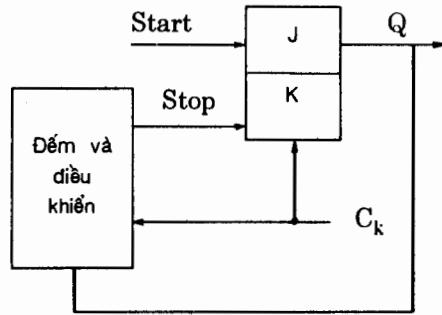
8.7. Vẽ biểu đồ thời gian của bộ đếm dị bộ:

a. Nhị phân 4 bits nghịch;

b. Nhị phân 4 bits thuận.

Giả thiết rằng các FF được lật ứng với sườn âm của xung nhịp.

8.8. Dùng 7490 và các cổng NAND thiết kế bộ đếm có hệ số đếm là 882.



Hình 8-47. Cho bài tập 8.5.

CHƯƠNG 9

BỘ GHI DỊCH

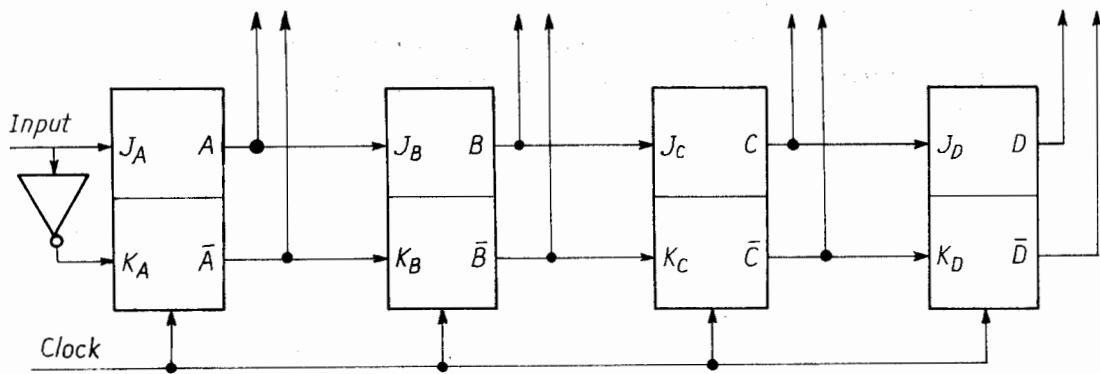
9_1. GIỚI THIỆU

9_1.1. ĐỊNH NGHĨA

Bộ ghi dịch (còn gọi là thanh ghi dịch) có khả năng ghi giữ và dịch thông tin (dịch phải hoặc dịch trái).

9_1.2. CẤU TẠO

Bộ ghi dịch gồm một dãy các phần tử nhớ đơn bit (FF) được mắc liên tiếp và đóng trong cùng một vỏ. Các FF sử dụng trong bộ ghi dịch thường là $D - FF$ hoặc là các loại khác nhưng mắc theo kiểu D . Để ghi n bit thông tin người ta sử dụng n FF , đầu ra của FF này được mắc vào đầu vào của FF tiếp theo. Bộ ghi dịch ghi được n bit thông tin được gọi là bộ ghi dịch n bit. Hình 9-1 là sơ đồ của một bộ ghi dịch 4 bit dùng $JK - FF$.



Hình 9.1. Bộ ghi dịch 4 bit dùng JK - FF.

Thông tin được nạp vào bộ ghi dịch từng bit một và được đồng bộ với xung nhịp C_k : bit đầu tiên được nạp vào FF đầu tiên, các bit thông tin khác đã được ghi giữ trong thanh ghi sẽ được dịch phải 1 bit, nghĩa là bit lưu trữ trong $FF - A$ sẽ chuyển sang $FF - B$, bit lưu trữ trong $FF - B$ sẽ chuyển sang $FF - C$... và cứ như vậy cho đến FF cuối cùng.

9_1.3. PHÂN LOẠI

Có nhiều cách phân loại bộ ghi dịch :

a) *Phân loại theo cách đưa thông tin vào, lấy thông tin ra*

- Vào nối tiếp, ra song song : thông tin được đưa vào thanh ghi dịch tuần tự từng bit một, số liệu được đưa ra đồng thời từ là tất cả n FF của thanh ghi được đọc cùng một lúc.
- Vào song song, ra song song : thông tin được đưa vào và lấy ra đồng thời ở n FF .
- Vào nối tiếp, ra nối tiếp : thông tin được đưa vào , lấy ra tuần tự từng bit một.
- Vào song song, ra nối tiếp : thông tin được đưa vào đồng thời cả n FF , lấy ra tuần tự từng bit một dưới sự điều khiển của xung nhịp.

b) *Phân loại theo đầu vào*

- Đầu vào đơn : mỗi FF trong bộ ghi dịch chỉ sử dụng một đầu vào điều khiển ví dụ đầu vào điều khiển D của $D - FF$ hoặc các loại FF khác nhưng mắc mạch theo kiểu D .
- Đầu vào đôi : Các FF trong bộ ghi dịch sử dụng cả hai đầu vào điều khiển ví dụ hai đầu vào điều khiển của $JK - FF$ hay $RS - FF$.

c) *Phân loại theo đầu ra*

- Đầu vào đơn :mỗi FF trong bộ ghi dịch chỉ có một đầu ra Q_i (hoặc \bar{Q}_i) được đưa ra chân của vi mạch.
- Đầu ra đôi :cả hai đầu ra của FF Q_i và \bar{Q}_i đều được đưa ra chân của vi mạch.

9_1.4. ỨNG DỤNG CỦA BỘ GHI DỊCH

Bộ ghi dịch được sử dụng rộng rãi để nhớ số liệu, chuyển số liệu song song thành nối tiếp, nối tiếp thành song song. Bộ ghi dịch là thành phần không thể thiếu được trong CPU của các hệ vi xử lý, trong các cổng vào ra có khả năng lập trình.

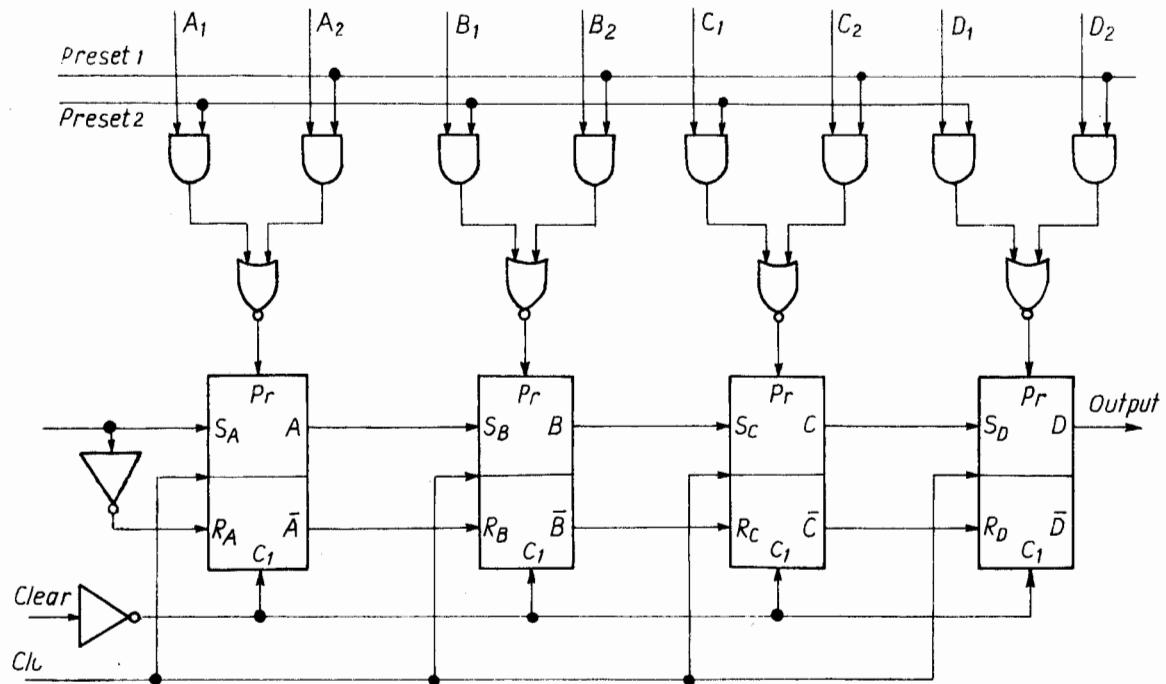
Bộ ghi dịch còn được dùng để thiết kế bộ đếm, tạo dây tín hiệu nhị phân tuần hoàn theo yêu cầu cho trước.

9.2. THANH GHI DỊCH 4 BIT NẠP VÀO NỐI TIẾP HOẶC SONG SONG RA NỐI TIẾP, DỊCH PHẢI

Sơ đồ bộ ghi dịch này được cho ở hình 9-2.

Bộ ghi dịch này có thể nạp thông tin vào nối tiếp hoặc song song. Đầu ra nối tiếp được lấy ra ở FF cuối cùng ($FF - D$). Việc nạp thông tin vào song song được thực hiện bởi

nội trong hai đầu vào Preset 1, Preset 2. Trước khi làm việc cần phải xoá tất cả các FF về 0" nhờ đầu vào xoá Clear. Thông tin trong bộ ghi dịch này được dịch phải.



Hình 9-2. Bộ ghi dịch 4 bit vào nối tiếp hoặc song song ra nối tiếp, dịch phải.

9_3. BỘ GHI DỊCH 4 BIT, DỊCH PHẢI, DỊCH TRÁI

Vì mạch 7495 là một vi mạch cho thanh ghi dịch loại này, sơ đồ nguyên lý của nó được cho ở hình 9-3.

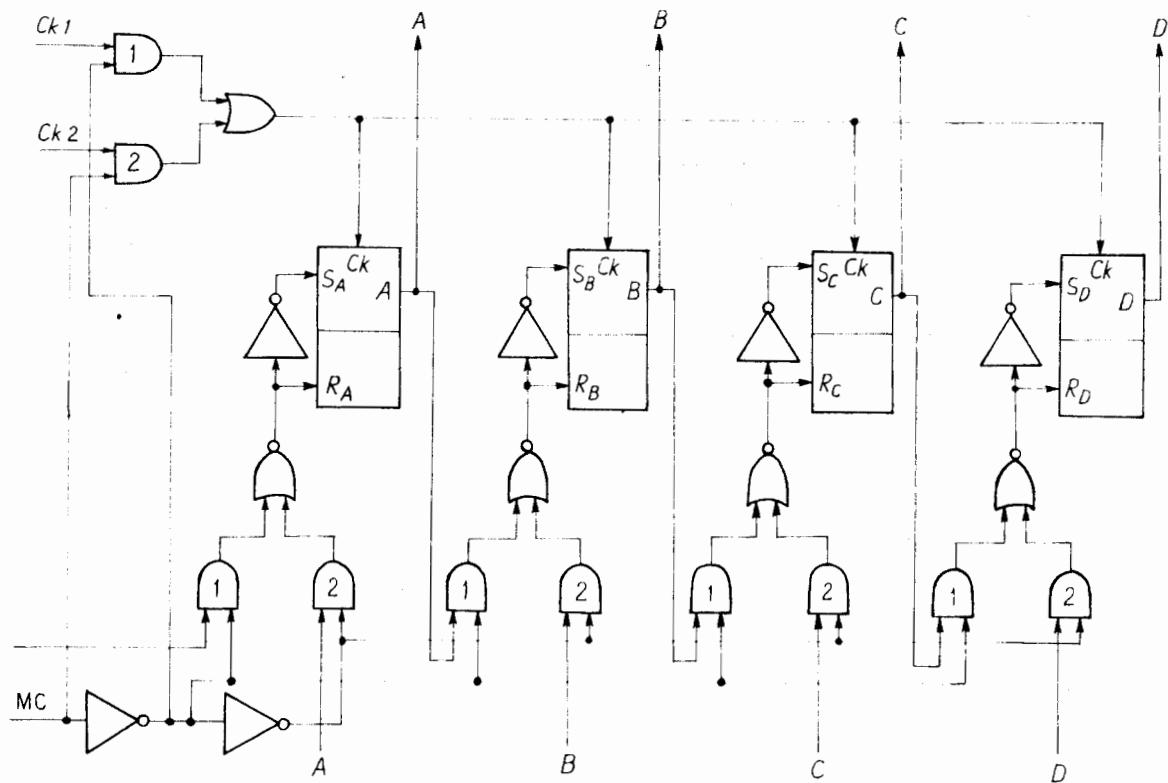
Mạch này có cả 4 khả năng vào, ra thông tin (vào song song ra nối tiếp, vào song song ra song song, vào nối tiếp ra nối tiếp, vào nối tiếp ra song song).

Đầu vào điều khiển *MC* (Mode Control) cho phép số liệu vào song song hay nối tiếp. Cụ thể :

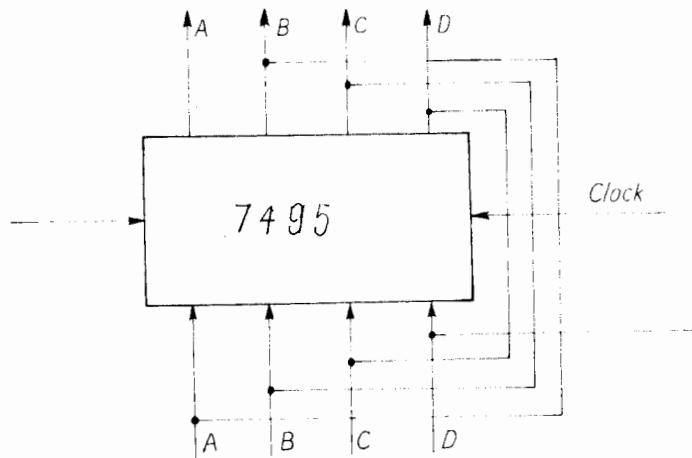
- $MC = 0$ dữ liệu được đưa vào nối tiếp nhờ xung nhịp Ck_1 (các cổng AND_1 mở).
- $MC = 1$ các cổng AND_2 mở nhờ xung nhịp Ck_2 , dữ liệu được đưa vào song song.

Bộ ghi dịch này có thể dịch phải, số liệu có thể lấy ra theo cả hai cách : song song (trên các đầu ra A, B, C, D) hoặc nối tiếp (trên đầu ra D).

Ngoài ra, với cách nối mạch như hình 9-4, bộ ghi dịch 7495 có thể dịch theo cả hai hướng phải và trái.



Hình 9-3. Sơ đồ vi mạch 7495.



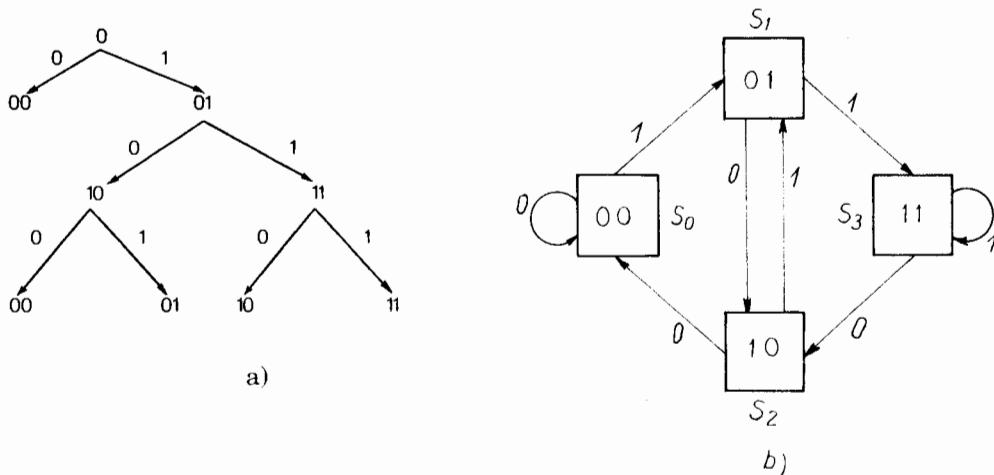
Hình 9-4. Sơ đồ nối chân của 7495 để dịch theo 2 hướng.

9_4. ĐỒ HÌNH TỔNG QUÁT (ĐỒ HÌNH DE BRUIJN) CỦA BỘ GHI DỊCH

Bảng chuyển đổi trạng thái của thanh ghi dịch 2 bit được chỉ ra trong hình 9-5,a. Nhờ bảng này ta có thể biết được tất cả các chuyển đổi trạng thái của mạch.

Ví dụ, nếu trạng thái ban đầu của bộ ghi dịch là 00 thì nó có thể chuyển đến một trong hai trạng thái tiếp theo : trạng thái 00 nếu đầu vào J của FF đầu tiên là 0 và là 01 nếu đầu vào J của FF đầu tiên là 1. Tương tự, nếu trạng thái ban đầu của bộ ghi dịch là 01 thì nó sẽ chuyển đến các trạng thái tiếp theo là 10 hoặc 11.

Từ bảng chuyển đổi trạng thái hình 9-5,a có thể biểu diễn ở dạng đồ hình tổng quát như trong hình 9-5,b. Các giá trị số ghi trên các cung chính là giá trị của hàm hồi tiếp (cung chính là đầu vào của FF đầu tiên) ứng với sự chuyển đổi đó.



Hình 9-5. Bộ ghi dịch 2 bit.

a. Bảng chuyển đổi trạng thái ; b. Đồ hình tổng quát - Đồ hình De Bruijn.

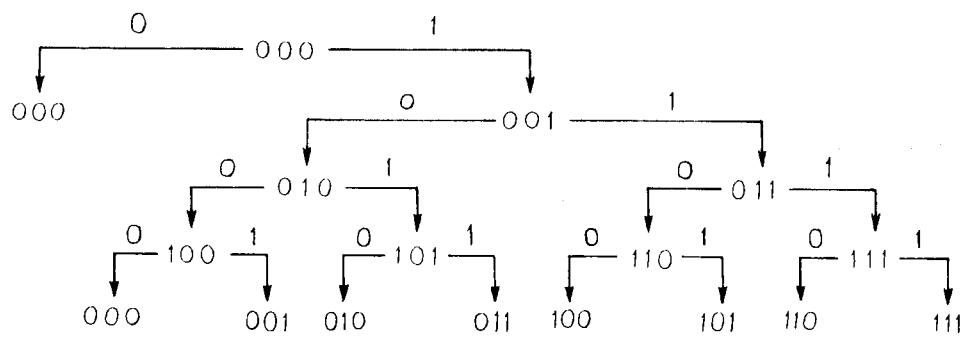
Từ đồ hình ta nhận thấy bộ ghi dịch sẽ bị "khoá" ở trạng thái 00 nếu hàm hồi tiếp là 0, hoặc ở trạng thái 11 nếu hàm hồi tiếp là 1. Đồ hình tổng quát còn được gọi là đồ hình De Bruijn.

Với cách làm tương tự ta có đồ hình De Bruijn của bộ ghi dịch 3 bit ở hình 9-6, của bộ ghi dịch bit ở hình 9-7.

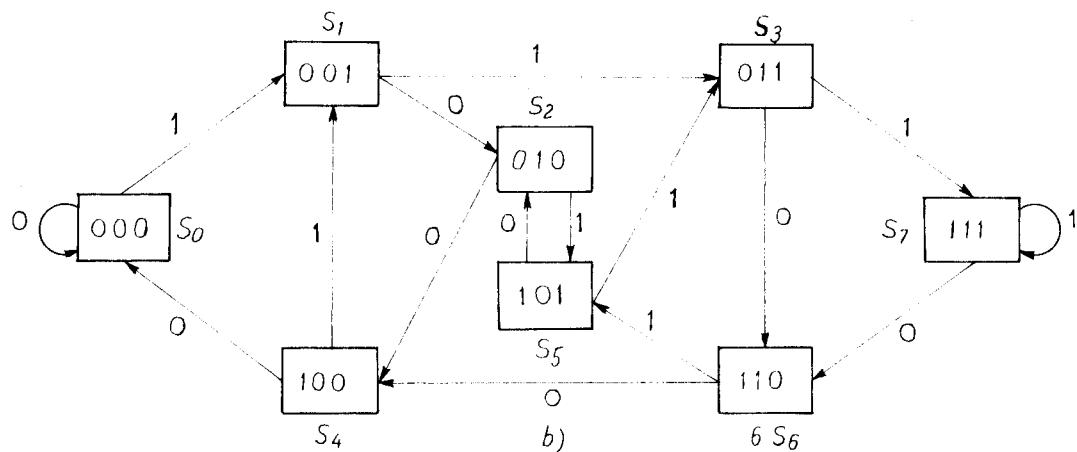
Nhận xét : Từ đồ hình De Bruijn của bộ ghi dịch luôn luôn xác định được trạng thái sẽ chuyển biến tới của bộ ghi dịch ứng với giá trị tương ứng của hàm hồi tiếp.

Ví dụ : nếu trạng thái ban đầu của bộ ghi dịch 4 bit là 0011(S_3) thì ứng với hàm hồi tiếp là 1 mạch sẽ chuyển biến tới trạng thái 0111 (S_7) theo đồ hình De Bruijn ở hình 9-7.

Đồ hình này được sử dụng để thiết kế bộ đếm, bộ tạo dãy tín hiệu nhị phân... như sẽ trình bày trong các phần sau ở chương này.

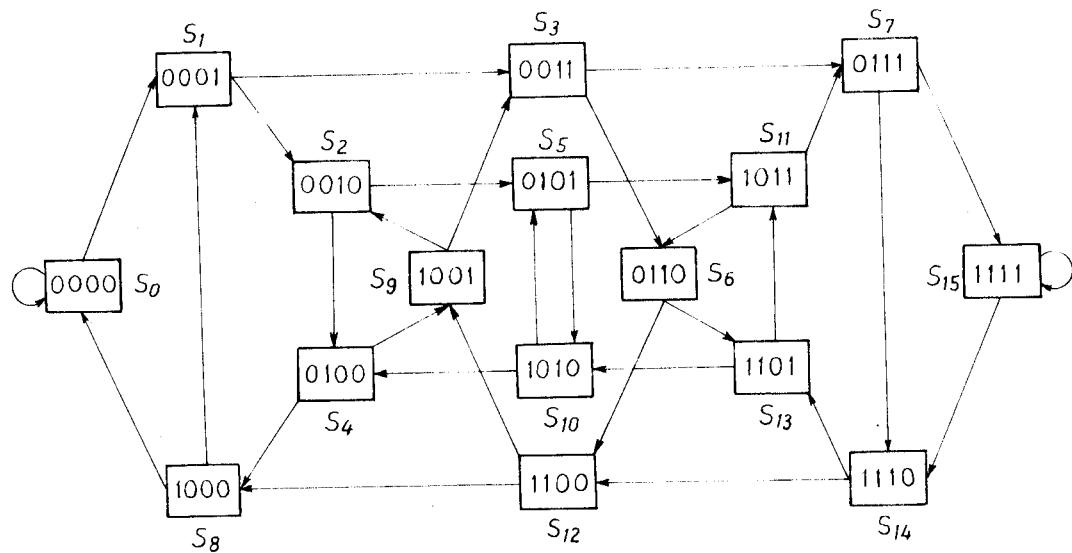


a)



b)

Hình 9-6. Đồ hình De Bruijn của bộ ghi dịch 3 bit.



Hình 9-7. Đồ hình De Bruijn của bộ ghi dịch 4 bit.

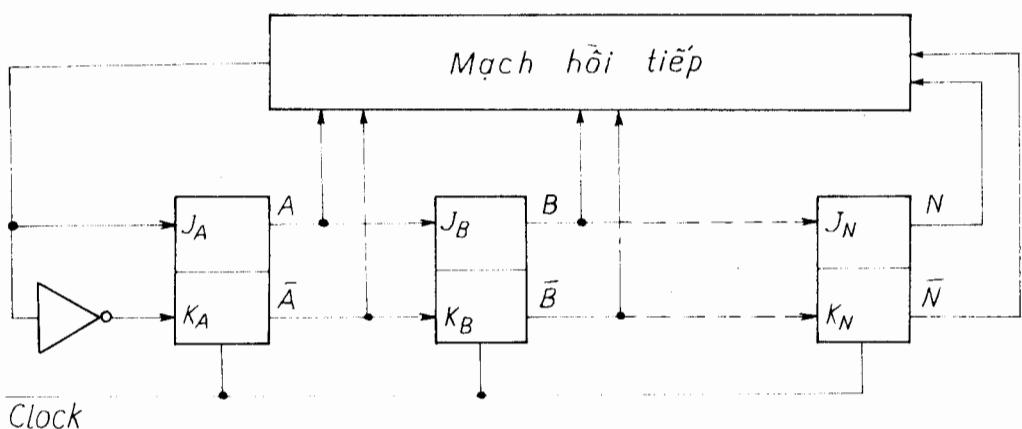
9_5. THIẾT KẾ BỘ ĐẾM DÙNG BỘ GHI DỊCH

9_5.1. SƠ ĐỒ KHỐI

Từ đồ hình tổng quát (đồ hình De Bruijn) của bộ ghi dịch nhận thấy rằng : xuất phát từ một trạng thái ban đầu nào đó , ứng với một dãy tín hiệu hồi tiếp $f_{ht} = D_A = J_A$ xác định, sẽ có một dãy xác định các trạng thái sẽ chuyển biến tới. Nếu dãy tín hiệu của f_{ht} được chọn thích hợp sao cho dãy chuyển biến trạng thái tạo thành một chu trình kín thì đồ hình trạng thái của mạch chính là đồ hình của một bộ đếm (đã xét trong chương 8).

Như vậy bài toán thiết kế bộ đếm dùng bộ ghi dịch trở thành bài toán thiết kế hàm hồi tiếp cung cấp cho đầu vào của bộ ghi dịch sao cho ứng với hàm này các trạng thái của mạch sẽ chuyển biến theo một chu trình kín, số trạng thái trong của chu trình bằng K_d .

Sơ đồ khối của bộ đếm dùng thanh ghi dịch có hàm hồi tiếp được chỉ ra ở hình 9-8.



Hình 9-8. Sơ đồ khối của bộ ghi dịch có hồi tiếp dùng để thiết kế bộ đếm.

Các FF được mác với nhau thành một thanh ghi dịch n bits. Đầu ra của các FF gồm cả Q và \bar{Q} được sử dụng để tạo hàm hồi tiếp đưa vào điều khiển đầu vào của thanh ghi (đầu vào kích của FF đầu tiên : $FF - A$). Giữa đầu vào và đầu ra của từng FF có mối quan hệ sau :

$$A' = f_{ht} (A, B, C, \dots, N)$$

$$B' = A$$

$$C' = B$$

$$\dots$$

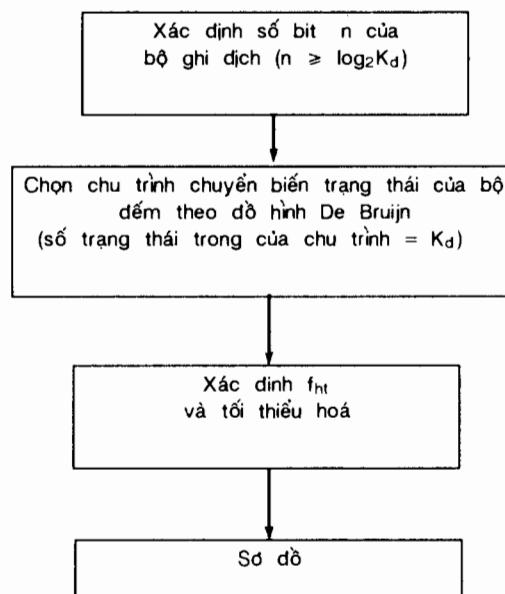
$$N' = M$$

Mạch hồi tiếp này có giá trị "0" hay "1" và được đưa vào đầu vào của $FF - A$, và khi có xung nhịp sẽ thiết lập trạng thái của $FF - A$ tương ứng.

Ví dụ nếu bộ ghi dịch đang ở trạng thái $N M \dots C B A = 0 0 \dots 0 0 1$, trạng thái tiếp theo của bộ ghi dịch sẽ là $0 0 \dots 0 1 0$ hoặc $0 0 \dots 0 1 1$ phụ thuộc vào giá trị của hàm hồi tiếp đưa vào đầu vào J của $FF - A$ là 0 hay 1.

9.5.2. CÁC BƯỚC THIẾT KẾ BỘ ĐẾM DÙNG THANH GHI DỊCH

Để thiết kế bộ đếm dùng bộ ghi dịch ta phải tiến hành theo các bước



Hình 9-9. Lưu đồ thiết kế bộ đếm dùng thanh ghi dịch.

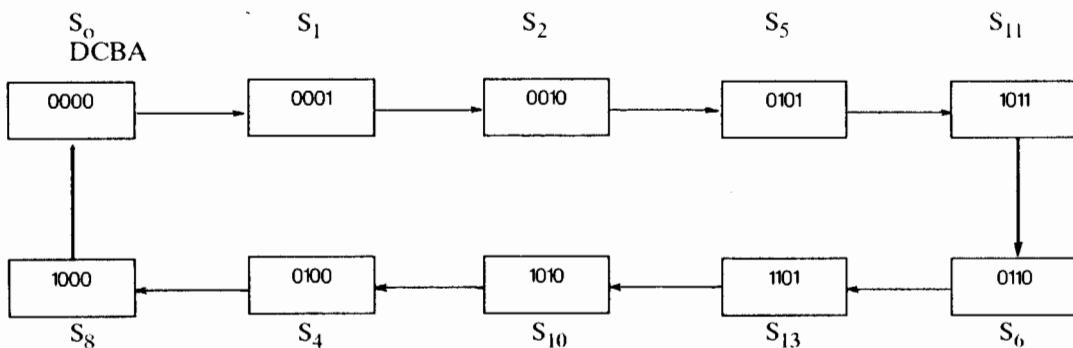
Ví dụ : thiết kế bộ đếm 10 dùng thanh ghi dịch.

Bước 1 : Xác định số bit của bộ ghi dịch $n = 4$

Bước 2 : Chọn chu trình chuyển biến trạng thái.

Theo đồ họa De Bruijn của bộ ghi dịch 4 bit (hình 9-7) giả sử ta chọn dãy trạng thái theo chu trình kín có 10 trạng thái khác nhau như trong hình 9-10,a.

Bước 3 : Xác định hàm hồi tiếp. Cũng theo đồ họa De Bruijn của bộ ghi dịch 4 bit, căn cứ vào dãy 10 trạng thái đã chọn ta có bảng trạng thái và giá trị của hàm hồi tiếp tương ứng (hình 9-10,b). Bảng Karnaugh của hàm được biểu diễn trên hình 9-10,c – trong đó các ô ghi 1 tương ứng với giá trị của hàm hồi tiếp là 1, những ô có dấu X tương ứng với các trạng thái không sử dụng, giá trị của f_{ht} là không xác định.



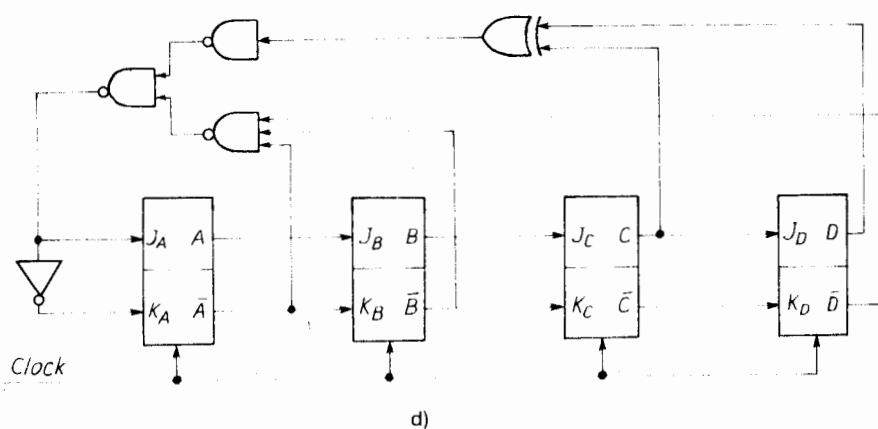
Hình 9.10 , a

S	D	C	B	A	f_{ht}
S_0	0	0	0	0	1
S_1	0	0	0	1	0
S_2	0	0	1	0	1
S_5	0	1	0	1	1
S_{11}	1	0	1	1	0
S_6	0	1	1	0	1
S_{13}	1	1	0	1	0
S_{10}	1	0	1	0	0
S_4	0	1	0	0	0
S_8	1	0	0	0	0

b)

		00	01	11	10
		DC	BA		
00		1		X	1
01			1	X	1
11		X		X	X
10			X		

c)



Hình 9-10. Thiết kế bộ đếm 10 dùng thanh ghi dịch.

- a. Đồ hình trạng thái ; b. Bảng trạng thái và giá trị của hàm hồi tiếp ; c. Bảng Karnaugh của hàm hồi tiếp ; d. Sơ đồ bộ đếm 10 dùng thanh ghi dịch.

9_6. TẠO DÃY TÍN HIỆU TUẦN HOÀN DÙNG THANH GHI DỊCH

Thanh ghi dịch còn có thể dùng để tạo ra một dãy tín hiệu tuần hoàn theo yêu cầu cho trước. Vấn đề là sơ đồ khối của mạch này như thế nào và làm sao để thiết kế được mạch thỏa mãn yêu cầu đã cho.

9_6.1. SƠ ĐỒ KHỐI CỦA MẠCH TẠO DÃY TÍN HIỆU TUẦN HOÀN DÙNG THANH GHI DỊCH

Gọi L là chu kỳ của dãy tín hiệu tuần hoàn cần phải tạo ra. Rõ ràng để tạo dãy này mạch phải tuần hoàn và có L trạng thái trong khác nhau. Mạch này theo định nghĩa chính là bộ đếm có $K_d = L$. Bộ đếm có thể xây dựng từ bộ ghi dịch và mạch hồi tiếp như đã trình bày ở hình 9-5.

Một vấn đề đặt ra là chỉ dùng bộ ghi dịch và mạch hồi tiếp đã đủ chưa? Nghĩa là tín

hiệu ra của mạch tạo dãy tín hiệu tuần hoàn có thể lấy trực tiếp từ các đầu ra của các FF s trong bộ ghi dịch không ?

Để trả lời câu hỏi này xét ví dụ sau : thiết kế mạch tạo dãy tín hiệu tuần hoàn $1 - 0 - 1 - 1 - 0 - 1 - 1 - 0$ ($L = 8$). Do $L = 8$ phải dùng 3 FF là A, B, C .

Giả sử lấy tín hiệu ra trực tiếp từ đầu ra của các $FF A$ hoặc B hoặc C (tín hiệu ở B, C xuất hiện chậm hơn ở A một hoặc hai nhịp). 8 tổ hợp của A, B, C tương ứng với dãy tín hiệu tuần hoàn $1 - 0 - 1 - 1 - 0 - 1 - 1 - 0$ được biểu diễn trên hình 9- 11.

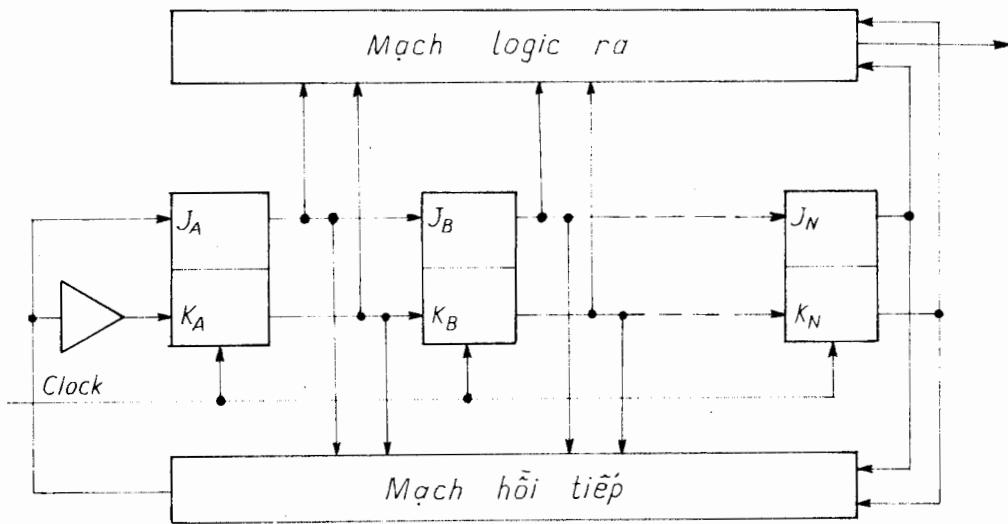
Clock	C	B	A
1	1	0	1
2	0	1	0
3	1	0	1
4	0	1	1
5	1	1	0
6	1	0	1
7	0	1	1
8	1	1	0

Hình 9-11. các tổ hợp của A, B, C ứng với dãy nhị phân
 $1 - 0 - 1 - 1 - 0 - 1 - 1 - 0$.

Từ hình 9-11 nhận thấy rằng : không thể lấy trực tiếp các đầu ra của các FF A, B, C làm đầu ra trực tiếp của mạch tạo dãy tín hiệu nhị phân vì :

Các trạng thái 101 xuất hiện 3 lần ; 110 xuất hiện 2 lần. Đối với trạng thái 101 trạng thái tiếp theo của nó không đơn trị, khi thì là 010, khi lại là 011.

Do vậy sơ đồ khối của mạch tạo dãy tín hiệu tuần hoàn ngoài bộ ghi dịch, mạch hồi tiếp, phải dùng thêm mạch logic ở đầu ra (hình 9-12).

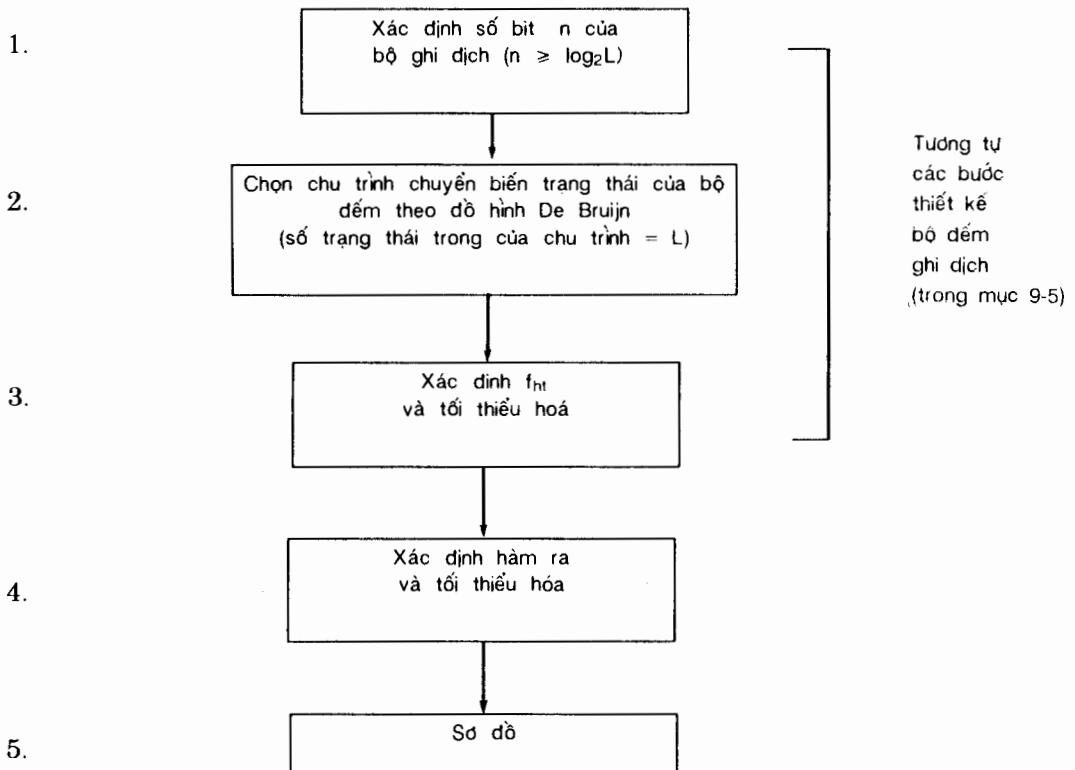


Hình 9-12. Sơ đồ khối của mạch tạo dãy tín hiệu nhị phân tuần hoàn dùng bộ ghi dịch.

9_6.2. CÁC BƯỚC THIẾT KẾ MẠCH TẠO DÃY TÍN HIỆU TUẦN HOÀN DÙNG BỘ GHI DỊCH

Bài toán thiết kế mạch tạo dây tín hiệu nhị phân tuần hoàn có chiều dài L dùng bộ ghi dịch có lưu đồ thuật toán như hình 9-13.

Các bước 1 – 3 tương tự các bước thiết kế bộ đếm dùng thanh ghi dịch.



Hình 9-13. Lưu đồ thuật toán thiết kế mạch tao dây tín hiệu tuần hoàn dùng bộ ghi dịch.

Để minh họa ta lấy lại ví dụ trên :xây dựng mạch tạo dây tín hiệu nhị phân tuần hoàn 1 - 0 - 1 - 1 - 0 - 1 - 1 - 0.

Bước 1 : Dãy tín hiệu tuần hoàn có chiều dài $L = 8$.

: Xác định số bit của bộ ghi dịch : $n = 3$ (vì $\log_2 8 = 3$).

Bước 2 : Chon chu trình chuyển biến trạng thái như sau :

$$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_5 \rightarrow S_3 \rightarrow S_7 \rightarrow S_6 \rightarrow S_4 \rightarrow S_0.$$

Đồ hình trang thái được chỉ ra trong hình 9-14.a.

Bước 3 : Xây dựng hàm hồi tiếp và tối thiểu hoá.

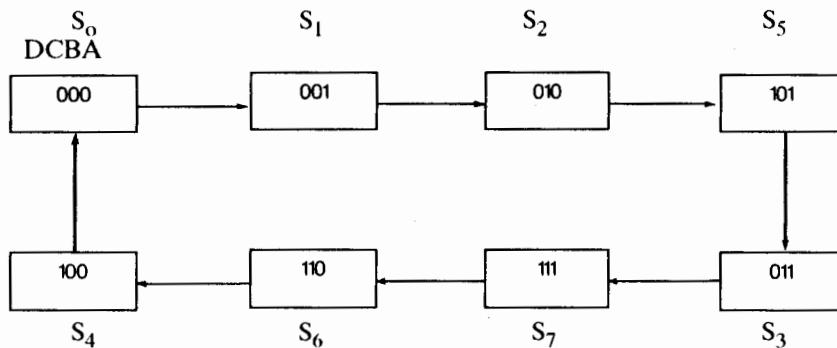
Cân cứ vào chương trình chuyển biến trạng thái đã chọn, dựa vào đồ hình De Bruijn của bộ ghi dịch 3 bit (hình 9-6) xác định được giá trị tương ứng của hàm hồi tiếp f_{ht} , ta nhận được bảng giá trị của hàm hồi tiếp như ở hình 9-14.b.

Tối thiểu hóa hàm hồi tiếp (f_{ht}) trên bảng Karnaugh (hình 9-14,c), ta được kết quả sau :

$$f_{ht} = B \bar{C} + \bar{A} \bar{C} + A B \bar{C}$$

Bước 4 : Xác định hàm logic ra và tối thiểu hóa. Nếu ký hiệu Q_A là đầu ra của mạch tạo dãy tín hiệu tuần hoàn hàm logic ra Q_A có giá trị như ở hình 9-14.b. Tối thiểu hóa hàm này trên bảng Karnaugh hình 9-14.d, ta có :

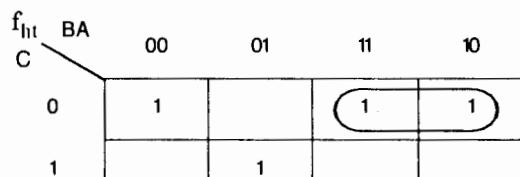
$$Q_A = \bar{A} \bar{C} + \bar{A} B + A C$$



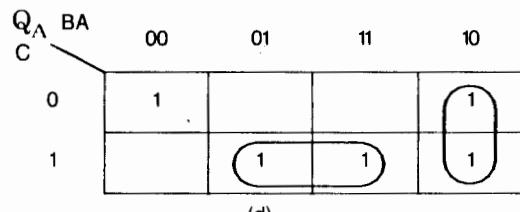
(a)

S	C	B	A	f_{ht}	Q
S_0	0	0	0	1	1
S_1	0	0	1	0	0
S_2	0	1	0	1	1
S_5	1	0	1	1	1
S_3	0	1	1	1	0
S_7	1	1	0	0	1
S_6	1	0	0	0	1
S_4	0	1	0	0	0

(b)



(c)



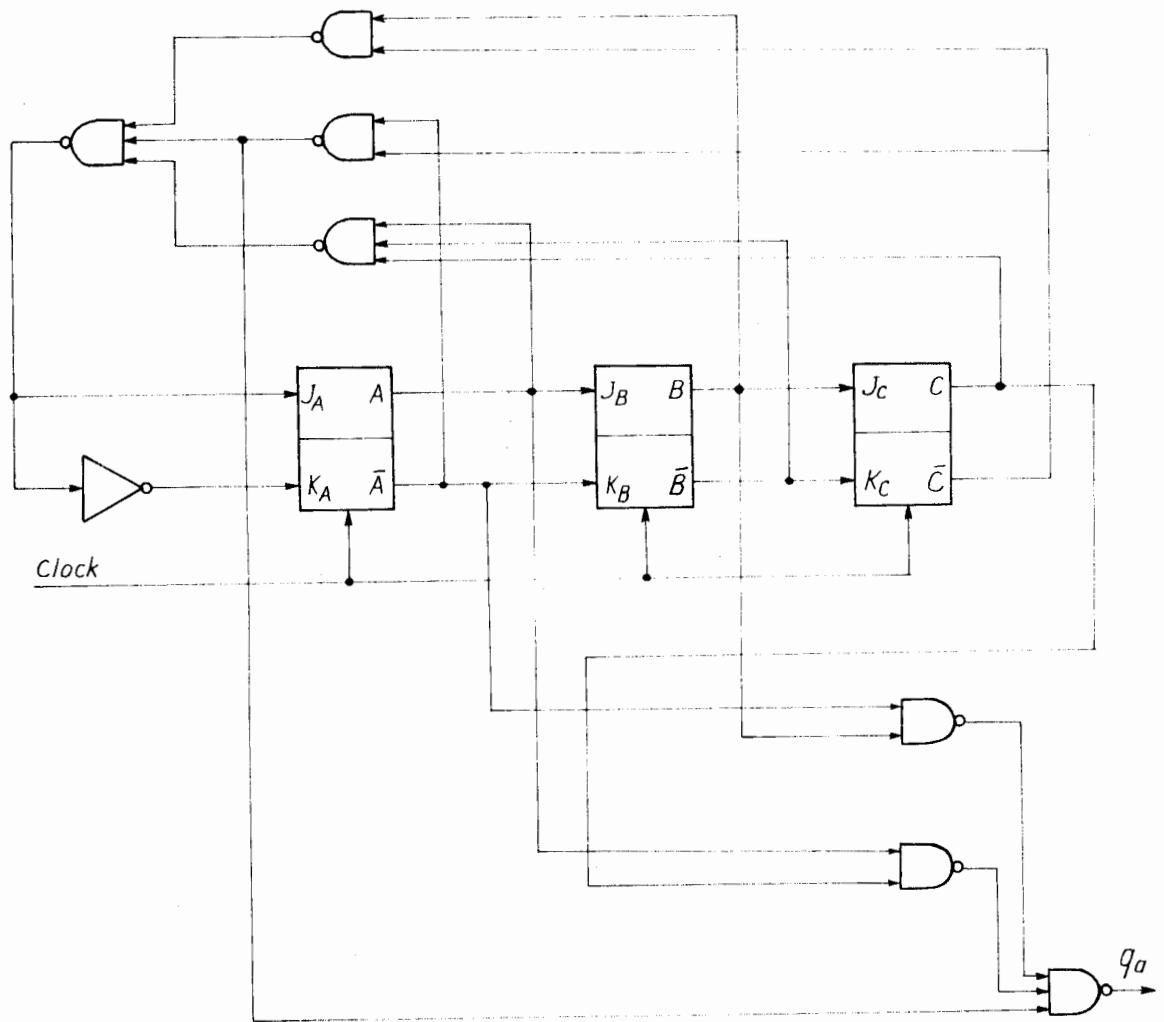
(d)

Hình 9-14. Thiết kế mạch tạo dãy tín hiệu nhị phân tuần hoàn.

- a. Đồ hình trạng thái của mạch;
- b. Bảng giá trị của hàm hồi tiếp (f_{ht}) và hàm ra (Q_A);
- c. Bảng Karnaugh của hàm hồi tiếp (f_{ht});
- d. Bảng Karnaugh của hàm ra (Q_A).

Bước 5 : Sơ đồ mạch.

Từ giá trị đã tối thiểu hóa của hàm hồi tiếp (f_{ht}) và hàm ra (Q_A), vẽ sơ đồ thực hiện mạch trên hình 9-15.



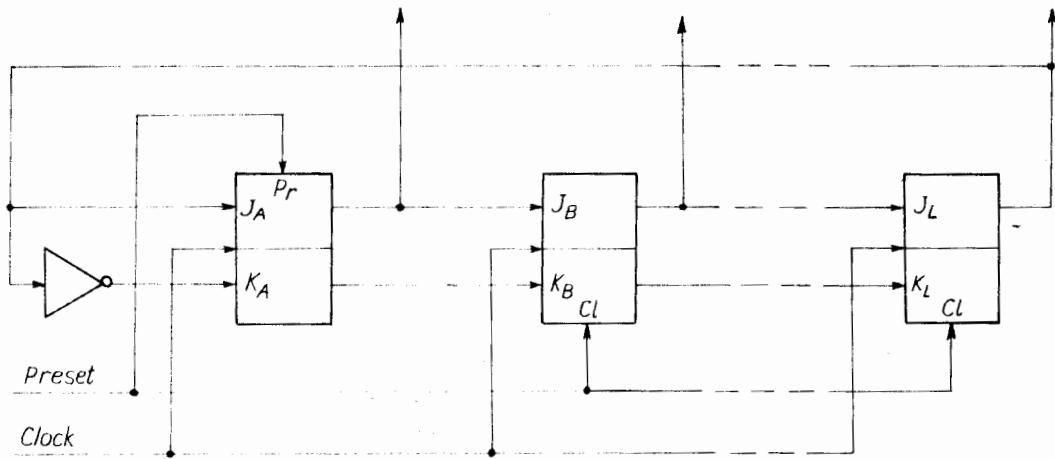
Hình 9-15. Sơ đồ mạch tạo dây tín hiệu nhị phân tuần hoàn dùng bộ ghi dịch.

9_7. BỘ ĐẾM VÒNG

Dạng đơn giản nhất của bộ đếm dùng thanh ghi dịch là bộ đếm vòng.

Trong bộ đếm vòng, hàm hồi tiếp được lấy từ đầu ra của FF cuối cùng và đưa về đầu vào của FF đầu tiên (hình 9-16).

Nếu mạch này dùng 10 FF là $A, B, C, D, E, F, G, H, I$ và J , thì đây là sơ đồ bộ đếm vòng thập phân. Dây trạng thái của bộ đếm vòng này được cho ở hình 9-17,a. Trong mạch này đã dùng 10 FF để xây dựng bộ đếm thập phân. Như vậy sẽ còn $2^{10} - 10$ trạng thái không sử dụng. Nếu vì một nguyên nhân khách quan nào đó bộ đếm rơi vào một trong những trạng thái không sử dụng (1014 trạng thái) thì từ sơ đồ cho ở hình 9-16 ta sẽ dễ dàng thấy rằng mạch sẽ không thể trở về chu trình hoạt động đúng được.



Hình 9-16. Sơ đồ của bộ đếm vòng.

Do vậy cần phải xây dựng hàm tự khởi động cho bộ đếm để sao cho nếu mạch ở một trong những trạng thái không sử dụng thì mạch sẽ tự động nhảy về chu trình trạng thái đúng.

Ví dụ khi bộ đếm vòng thập phân đang ở trạng thái sai : 0000000000 nghĩa là $A = B = C = D = E = F = G = H = I = J = 0$ thì mạch tự khởi động để bộ đếm trở về chu trình đúng 1000000000 ($A = 1, B = C = D = E = F = G = H = I = J = 0$) được cho như ở hình 9-17,b.

Trong bộ đếm vòng luôn luôn chỉ có 1 FF ở trạng thái "1", còn tất cả các FF khác ở trạng thái "0". Vì đặc điểm này, rất dễ dàng xây dựng hàm phát hiện những sai nhầm của bộ đếm vòng.

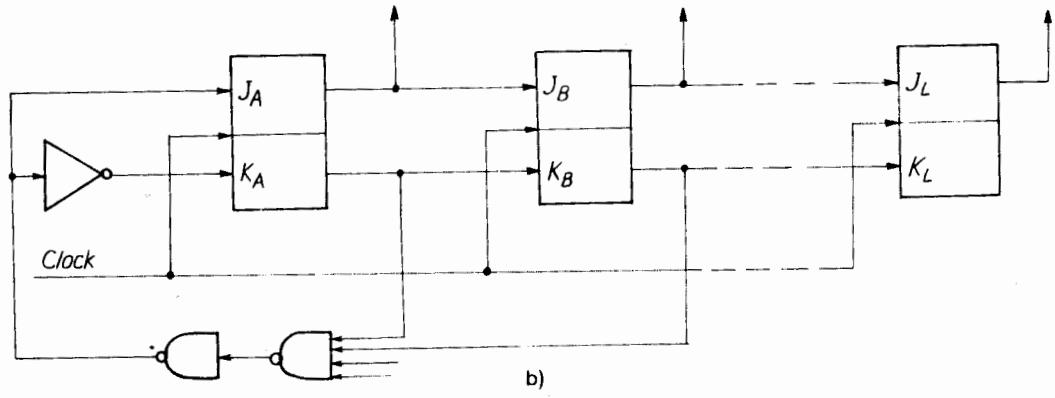
Ví dụ bộ đếm vòng dùng bộ ghi dịch 3 bit có hàm báo sai f_1 (khi có ít nhất 2 FF ở trạng thái "1") và f_0 (khi tất cả các FF đều ở "0") như sau :

$$f_1 = A \cdot B + A \cdot C + B \cdot C$$

$$f_0 = \overline{A} \cdot \overline{B} \cdot \overline{C}$$

Clock	L	I	H	G	F	E	D	C	B	A
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0

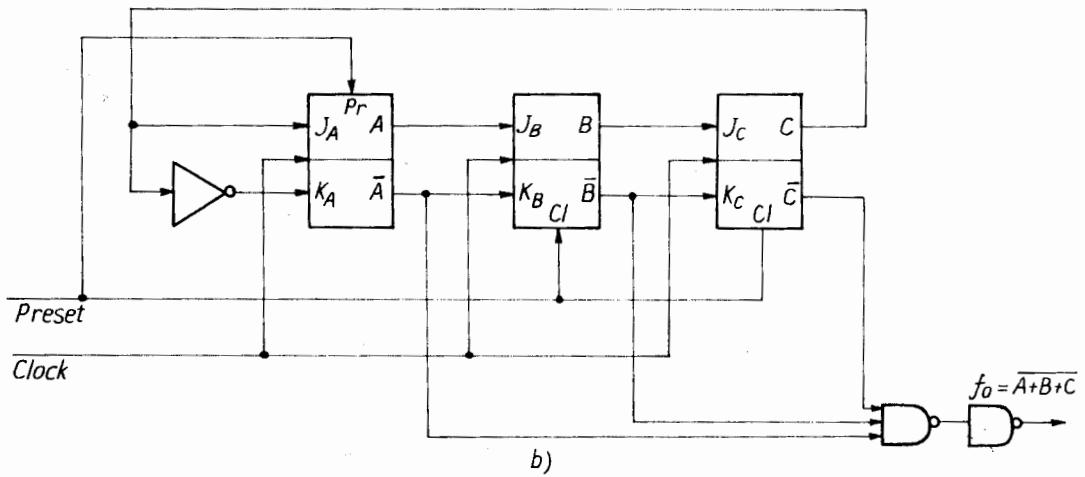
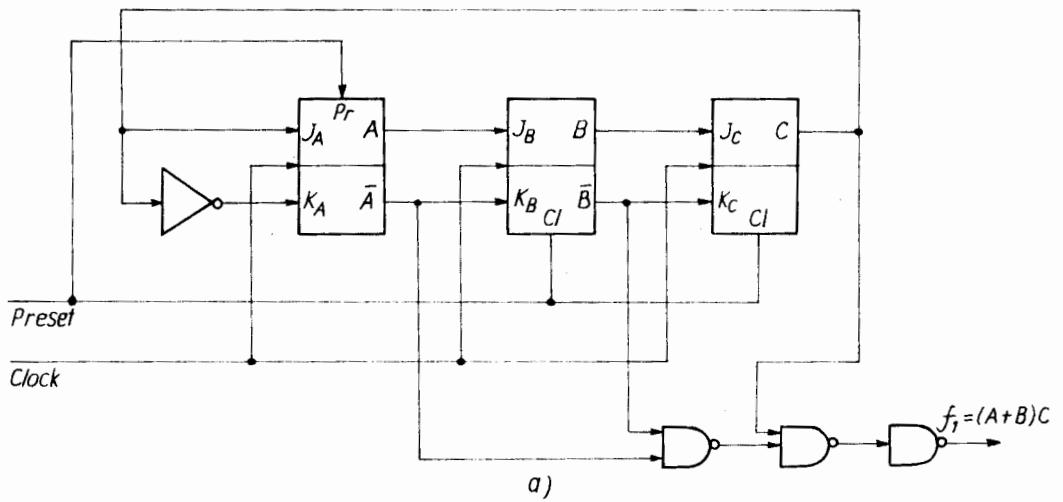
a)



Hình 9-17. Bộ đếm vòng thập phân.

a. Bảng trạng thái ; b. Mạch tự khởi động.

Sơ đồ của mạch báo sai được biểu diễn trên hình 9-18.



Hình 9-18. Mạch báo sai của bộ đếm vòng 3 bit.

a. Mạch báo sai f_1 ; b. Mạch báo sai f_0 .

9.8. BỘ ĐẾM VÒNG XOẮN (BỘ ĐẾM JOHNSON)

Bộ đếm vòng xoắn là một dạng đơn giản của bộ đếm dùng thanh ghi dịch. Bộ đếm vòng xoắn khác với bộ đếm vòng ở chỗ : hàm hồi tiếp được đưa từ đầu ra đảo của FF cuối cùng về đầu vào của FF đầu tiên. Nếu dùng bộ ghi dịch n bit, thì bộ đếm vòng xoắn có $K_d = 2^n$.

Hình 9-19,a biểu diễn một bộ đếm thập phân mác theo kiểu vòng xoắn, xây dựng từ một thanh ghi dịch 5 bit (5 FF_s là : A, B, C, D và E).

Giả thiết ban đầu tất cả các FF đều được xoá về "0", dãy 10 trạng thái được chọn từ đồ hình De Bruijn của thanh ghi dịch 5 bit và giá trị tương ứng của hàm hồi tiếp được cho ở hình 9-19,b.

Từ bảng Karnaugh (hình 9-19,c) ta nhận được hàm hồi tiếp :

$$f_{ht} = J_A = E$$

Để hiển thị giá trị của bộ đếm vòng xoắn ta phải dùng bộ giải mã. Bảng Karnaugh hình 9-19,d biểu diễn các hàm logic ở đầu ra của bộ giải mã, trong đó giá trị số ghi trong ô tương ứng với đầu ra của bộ giải mã. Tối thiểu hoá các hàm này, nhận được kết quả : $\overline{A} \overline{B} \overline{C} \overline{D} \overline{E}$.

$0 = \overline{A} \overline{E}$	$5 = A E$
$1 = A \overline{B}$	$6 = \overline{A} B$
$2 = B \overline{C}$	$7 = \overline{B} C$
$3 = C \overline{D}$	$8 = \overline{C} D$
$4 = D \overline{E}$	$9 = \overline{D} E$

Sơ đồ thực hiện giải mã cho đầu ra 0 được cho ở hình 9-19,e.

Từ đồ hình De Bruijn của bộ ghi dịch 5 bit ta còn nhận được 3 dãy chuyển biến trạng thái nữa độc lập với dãy chuyển biến trạng thái của bộ đếm vòng xoắn, ba dãy đó là :

- 1) $S_2 \rightarrow S_5 \rightarrow S_{11} \rightarrow S_{23} \rightarrow S_{14} \rightarrow S_{29} \rightarrow S_{26} \rightarrow S_{20} \rightarrow S_8 \rightarrow S_{17} \rightarrow S_2$
- 2) $S_4 \rightarrow S_9 \rightarrow S_{19} \rightarrow S_6 \rightarrow S_{13} \rightarrow S_{27} \rightarrow S_{22} \rightarrow S_{12} \rightarrow S_{25} \rightarrow S_{18} \rightarrow S_4$
- 3) $S_{10} \rightarrow S_{21} \rightarrow S_{10}$

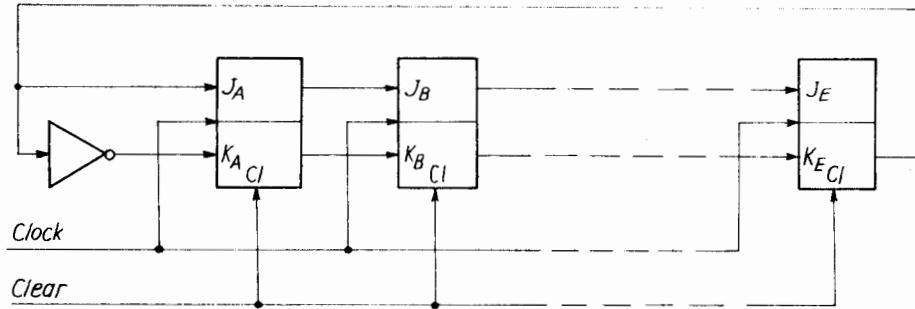
Về mặt lý thuyết, bộ đếm vòng xoắn không ở trong bất kỳ trạng thái của cả ba dãy này. Nhưng nếu vì một nguyên nhân khách quan bộ đếm rơi vào một trong bất kỳ trạng thái nào của ba dãy này thì do sơ đồ nguyên lý của mạch sẽ chỉ chuyển đến trạng thái tiếp theo thuộc cùng dãy đó khi có tín hiệu vào đếm. Như vậy bộ đếm không thể trở về chu trình hoạt động đúng được.

Bảng Karnaugh trên hình 9-19,f chỉ ra ba dãy không mong muốn đó, trong đó số ghi ở các ô đánh dấu trạng thái đó ở dãy 1, 2 hay 3. Cụ thể các ô đánh số 1 là những trạng thái

của dãy 1, các ô đánh số 2 ứng với các trạng thái của dãy 2, ô đánh số 3 ứng với trạng thái của dãy 3.

Nhận thấy rằng các trạng thái S_2, S_6, S_{10}, S_{14} là những trạng thái kế cận nhau trong bảng Karnaugh và chúng lại là đại diện cho đầy đủ cả 3 dãy trạng thái không sử dụng 1, 2 và 3. Hàm logic cho các trạng thái đó là :

$$f = \overline{A} B \overline{E}$$



a)

Clock	E	D	C	B	A	f_{ht}
0	0	0	0	0	0	1
1	0	0	0	0	1	1
2	0	0	0	1	1	1
3	0	0	1	1	1	1
4	0	1	1	1	1	1
5	1	1	1	1	1	0
6	1	1	1	1	0	0
7	1	1	1	0	0	0
8	1	1	0	0	0	0
9	1	0	0	0	0	0

b)

		f _{ht}								
		CB	00	01	11	10	10	11	01	00
ED	CB	00	1	X	X	X	X	1	1	1
		01	X	X	X	X	X	1	X	X
11		X					X		X	X
10			X	X	X	X	X	X	X	X

c)

CB
ED

	00	01	11	10	10	11	01	00
00	0	X	X	X	X	3	2	1
01	X	X	X	X	X	4	X	X
11	8	X	6	7	X	5	X	X
10	9	X	X	X	X	X	X	X

A = 0 A = 1

d)



e)

f_o
CB
ED

	00	01	11	10	10	11	01	00
00	S_2	S_6		2	1			
01	S_{10}	S_{14}		2	2		1	2
11		1			1		2	2
10		2	2	1	3		2	1

A = 0 A = 1

$f_o = \bar{A} B \bar{E}$

f)

Hình 9-19. Bộ đếm thập phân vòng xoắn.

- a. Sơ đồ ; b. Bảng trạng thái của bộ đếm vòng xoắn ; c. Bảng Karnaugh của hàm hồi tiếp ; d. Bảng Karnaugh của bộ giải mã thập phân ; e. Sơ đồ bộ giải mã cho đầu ra số 0 ; f. Bảng Karnaugh của hàm tự khởi động.

Hàm này được dùng để xóa tất cả các FF của bộ đếm. Như vậy nhờ cách này nếu bộ đếm rơi vào bất kỳ một trong những trạng thái không sử dụng nào của dây 1, dây 2 hay dây 3, thì sau một số xung nhịp bộ đếm sẽ chuyển đến một trong các trạng thái: S_2 , S_6 , S_{10} hay S_{14} và nhờ hàm xóa f mạch trở về chu trình hoạt động đúng.

Đối với bài toán của chúng ta, sau nhiều nhất là 10 xung nhịp, đầu vào xoá f_o sẽ xuất hiện, bộ đếm trở về chu trình hoạt động đúng.

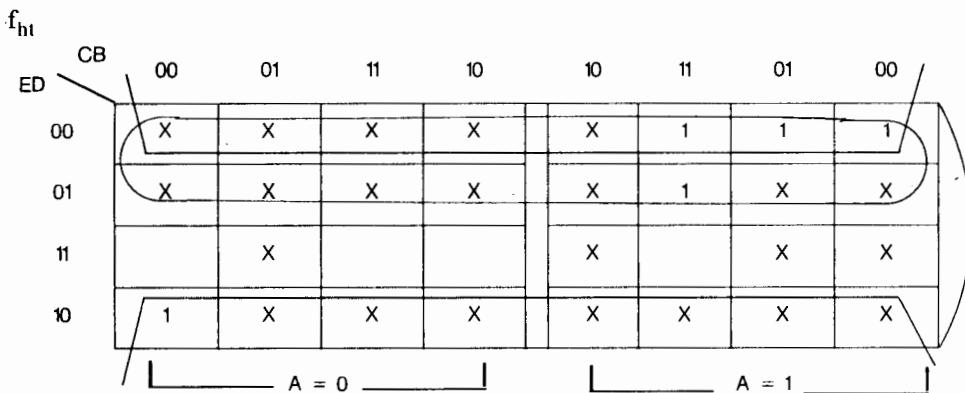
Như đã trình bày ở trên, nếu dùng thanh ghi dịch n bit thì bộ đếm vòng xoắn tương ứng có $K_d = 2^n$, nghĩa là hệ số đếm là số chẵn. Tuy nhiên với hàm hồi tiếp thích hợp, ta có thể xây dựng bộ đếm vòng xoắn có hệ số đếm lẻ ($K_d = 2^n - 1$) từ thanh ghi dịch n bits.

Ví dụ, có thể dùng thanh ghi dịch 5 bit để xây dựng bộ đếm vòng xoắn và loại đi trạng thái 00000 như ở bảng trạng thái hình 9-20,a. Trong bảng này, giá trị tương ứng của hàm hồi tiếp được viết ở cột cuối cùng của bảng. Biểu diễn hàm này trên bảng Karnaugh (hình 9-20,b) lưu ý tới các trạng thái không sử dụng, tối thiểu hoá, nhận được kết quả :

$$f = \overline{D} + \overline{E}.$$

Clock	E	D	C	B	A	f_{ht}
1	0	0	0	0	1	1
2	0	0	0	1	1	1
3	0	0	1	1	1	1
4	0	1	1	1	1	1
5	1	1	1	1	1	0
6	1	1	1	1	0	0
7	1	1	1	0	0	0
8	1	1	0	0	0	0
9	1	0	0	0	0	0

a)



$$f_{ht} = \overline{D} + \overline{E}$$

b)

Hình 9-20. Bộ đếm vòng xoắn dùng bộ ghi dịch 5 bit với $K_d = 9$ (loại đi trạng thái 00000);

a : Bảng trạng thái ; b : Bảng Karnaugh cho hàm hồi tiếp.

Tương tự như trên, nếu loại đi trạng thái 11111 thì hàm hồi tiếp sẽ là :

$$f = \overline{D} \overline{E}.$$

9_9. BỘ GHI DỊCH VỚI HÀM HỒI TIẾP LÀ HÀM CỘNG MODUL

Như đã trình bày ở trên, có thể dùng thanh ghi dịch và mạch hồi tiếp để xây dựng bộ iếm.

Bộ ghi dịch n bit có $f_{ht} = \overline{\text{đầu ra của } FF \text{ cuối}}$, là bộ đếm vòng, $K_d = n$.

Bộ ghi dịch n bit có $f_{ht} = \overline{\text{đầu ra của } FF \text{ cuối}}$, là bộ đếm vòng xoắn, $K_d = 2n$.

Ở đây xét bộ ghi dịch dùng hàm hồi tiếp cộng modul. Với hàm hồi tiếp này, hệ số đếm (K_d) hay độ dài của dãy tín hiệu tuần hoàn (L) của mạch có thể đạt được giá trị cực đại $K_{d\max} = L_{\max} = 2^n - 1$.

9_9.1. BỘ GHI DỊCH CÓ HÀM HỒI TIẾP CỘNG MODUL CÓ $L_{\max} = 2^n - 1$

Thanh ghi dịch 4 bit cho trong hình 9-21,a có hàm hồi tiếp :

$$f_{ht} = J_A = C \oplus D$$

Dựa vào đồ hình De Bruijn của bộ ghi dịch 4 bit (hình 9-7) và phương trình của hàm hồi tiếp đồng thời là hàm kích cho $FF - A$ dễ dàng xác định được trạng thái tiếp theo của bộ ghi dịch khi biết trạng thái hiện tại.

Ví dụ, nếu bộ ghi dịch đang ở trạng thái S_1 (0001 tức là $D = 0, C = 0, B = 0$ và $A = 1$) thì $f_{ht} = J_A = C \oplus D = 0 \oplus 0 = 0$, do vậy khi có xung nhịp Ck tiếp theo bộ ghi dịch sẽ chuyển đến trạng thái S_2 (0010 tức là $D = 0, C = 0, B = 1, A = 0$).

Bảng đầy đủ các chuyển biến trạng thái của thanh ghi dịch 4 bits này được biểu diễn trên hình 9-21,b.

Trạng thái S_o (0000) không xuất hiện trong bảng trạng thái vì nếu bộ ghi dịch ở trạng thái này, do hàm hồi tiếp $f_{ht} = J_A = C \oplus D$ nó sẽ nằm mãi ở trạng thái này không sao thoát ra được. S_o chính là *trạng thái "khoá"* hay còn gọi là *trạng thái "chết"*. Do vậy phải loại S_o .

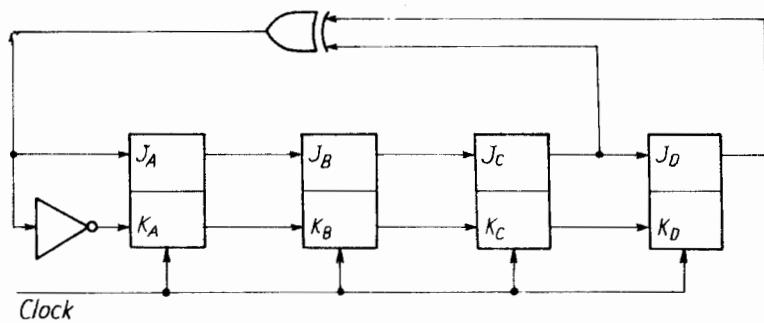
Thanh ghi dịch 4 bit này có số trạng thái khác nhau trong dãy chuyển biến trạng thái (L hay K_d) là 15 đạt L_{\max} .

Một số bộ ghi dịch n bits có hàm hồi tiếp cộng modul và chiều dài cực đại $L_{\max} = 2^n - 1$ được cho ở hình 9-22.

Mặt khác có thể có nhiều hàm hồi tiếp cộng modul khác nhau cùng cho ra L_{\max} . Hình 9-23 chỉ ra 3 hàm hồi tiếp khác nhau cùng cho L_{\max} của bộ ghi dịch 4 bit.

Các mạch dùng thanh ghi dịch có hồi tiếp cộng modul có thể được sử dụng như mạch tạo một dãy tín hiệu tuần hoàn. Nếu tín hiệu này được lấy trực tiếp trên đầu ra của FF thì dãy tín hiệu sẽ tuần hoàn có thời gian lặp lại T_{LL} được tính theo công thức :

$$T_{LL} = L_{\max} \cdot T_{Ck} = (2^n - 1) \cdot T_{Ck}.$$



a)

S	D	C	B	A	F
S ₁	0	0	0	1	0
S ₂	0	0	1	0	0
S ₄	0	1	0	0	1
S ₉	1	0	0	1	1
S ₃	0	0	1	1	0
S ₆	0	1	1	0	1
S ₁₃	1	1	0	1	0
S ₁₀	1	0	1	0	1
S ₅	0	1	0	1	1
S ₁₁	1	0	1	1	1
S ₇	0	1	1	1	1
S ₁₅	1	1	1	1	0
S ₁₄	1	1	1	0	0
S ₁₂	1	0	0	0	0
S ₈	1	1	0	0	1

b)

Hình 9-21. Bộ ghi dịch 4 bit với hàm hồi tiếp $f_{ht} = J_A = C \oplus D$.

a. Sơ đồ ; b. Bảng chuyển đổi trạng thái .

Số bit của bộ ghi dịch	Hàm hồi tiếp	Số bit của bộ ghi dịch	Hàm hồi tiếp
1	A	10	G \oplus J
2	A \oplus B	11	I \oplus K
3	B \oplus C	12	F \oplus H \oplus K \oplus L
4	C \oplus D	13	I \oplus J \oplus L \oplus M
5	D \oplus E	14	D \oplus H \oplus M \oplus N
6	E \oplus F	15	N \oplus O
7	F \oplus G	16	D \oplus M \oplus O \oplus P
8	D \oplus E \oplus F \oplus H	17	N \oplus Q
9	E \oplus I	18	K \oplus U

Hình 9-22. Một số bộ ghi dịch n bit có L_{max} với hàm hồi tiếp cộng modul.

S	D	C	B	A	f_{th}
S_1	0	0	0	1	1
S_3	0	0	1	1	1
S_7	0	1	1	1	0
S_{14}	1	1	1	0	1
S_{13}	1	1	0	1	1
S_{11}	1	0	1	1	0
S_6	0	1	1	0	0
S_{12}	1	1	0	0	1
S_9	1	0	0	1	0
S_2	0	0	1	0	1
S_5	0	1	0	1	0
S_{10}	1	0	1	0	0
S_4	0	1	0	0	0
S_8	1	0	0	0	0
S_0	0	0	0	0	1

a)

S	D	C	B	A	f_{th}
S_1	0	0	0	1	1
S_3	0	0	1	1	1
S_7	0	1	1	1	1
S_{15}	1	1	1	1	0
S_{13}	1	1	0	1	0
S_{11}	1	0	1	1	0
S_6	0	1	1	0	0
S_{12}	1	1	0	0	1
S_9	1	0	0	1	0
S_2	0	0	1	0	0
S_5	0	1	0	1	1
S_{10}	1	0	1	0	0
S_4	0	1	0	0	0
S_8	1	0	0	0	0
S_{14}	1	1	1	1	1

b)

S	D	C	B	A	f_{th}
S_1	0	0	0	1	0
S_3	0	0	1	1	0
S_7	0	1	1	1	0
S_{14}	1	1	1	0	0
S_{13}	1	1	0	1	1
S_{11}	1	0	1	1	1
S_6	0	1	1	0	1
S_{12}	1	1	0	0	0
S_9	1	0	0	1	1
S_2	0	0	1	0	1
S_5	0	1	0	1	0
S_{10}	1	0	1	0	0
S_4	0	1	0	0	0
S_8	1	0	0	0	0
S_0	0	0	0	0	1

c)

Hình 9-23. Một số hàm hồi tiếp hỗn hợp khác nhau của bộ ghi dịch 4 bit có L_{max} .

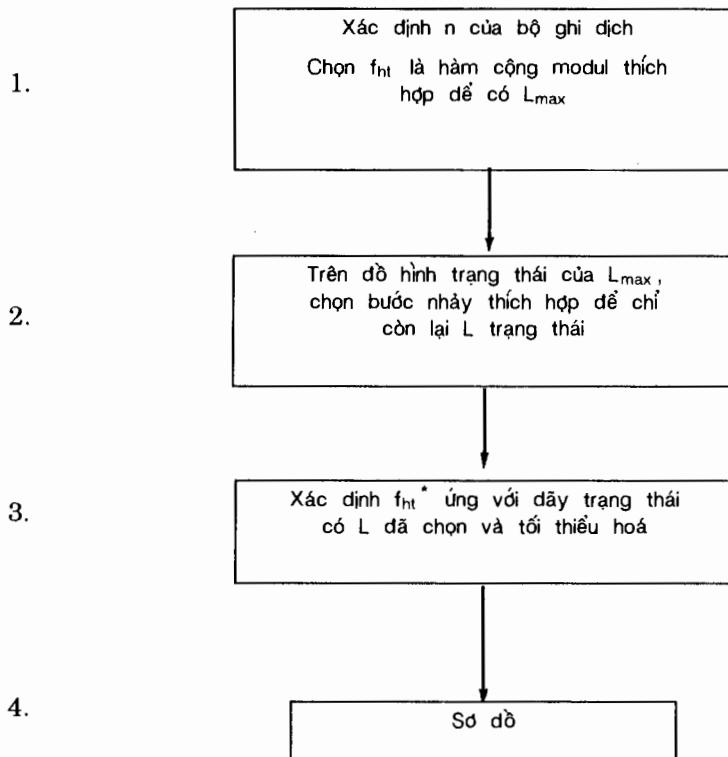
a. $f_{th} = C \sim D$; b. $f_{th} = A \oplus D$; c. $f_{th} = A \sim D$.

Ví dụ nếu tần số của xung nhịp là 1 MHz :

- với $n = 10$ $T_{LL} = 1.02$ ms;
- với $n = 20$ $T_{LL} = 1.05$ s;
- với $n = 30$ $T_{LL} = 1073.73$ s.

9_9.2. THIẾT KẾ BỘ ĐẾM DÙNG BỘ GHI DỊCH CÓ HÀM HỒI TIẾP CỘNG MODUL

Cũng có thể dùng bộ ghi dịch với hàm hồi tiếp cộng modul để xây dựng các bộ đếm có hệ số đếm bất kỳ ($K_d = L < L_{max}$) các bước tiến hành như hình 9-24.



Hình 9-24. Các bước thiết kế bộ đếm dùng bộ ghi dịch với hàm hồi tiếp cộng modul.

Ví dụ : Thiết kế bộ đếm thập phân dùng bộ ghi dịch với hàm hồi tiếp cộng modul.

- Bước 1 : $K_d = L = 10$ do vậy phải dùng thanh ghi dịch 4 bit ($n = 4$). Chọn hàm hồi tiếp là hàm cộng modul tương ứng để mạch đạt L_{max} . Như đã biết có thể chọn :

$$\begin{aligned}
 f_{ht} &= C \oplus D \text{ hoặc} \\
 &= C \sim D \text{ hoặc} \\
 &= A \oplus D \text{ hoặc} \\
 &= A \sim D .
 \end{aligned}$$

Giả sử chọn $f_{ht} = C \oplus D$.

- Bước 2 : Trên đồ hình trạng thái $L_{max} = 15$ của bộ ghi dịch 4 bit , với $f_{ht} = C \oplus D$ đã chọn (hình 9-25,a), chọn bước nhảy để loại đi 5 trạng thái chỉ còn lại 10 trạng thái ($L = 10$).

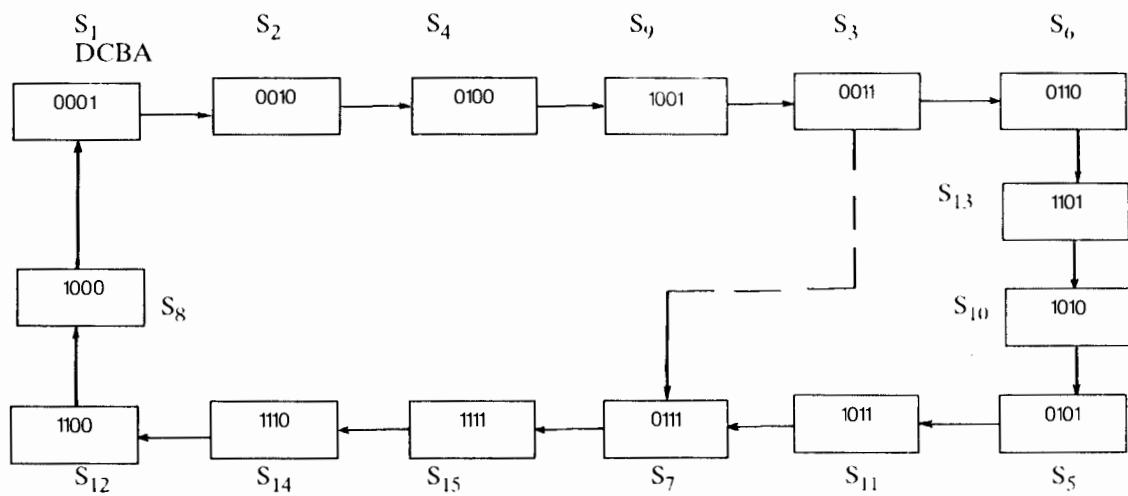
Ví dụ chọn bước nhảy từ S_3 về S_7 nghĩa là loại bỏ các trạng thái S_6 , S_{13} , S_{10} , S_5 , S_{11} khi mạch đang ở trạng thái S_3 không chuyển đến trạng thái S_6 mà chuyển đến S_7 (đường đứt nét trong hình 9-25,a).

- Bước 3 : Xác định f_{ht}^* . Dựa trên đồ họa De Bruijn của bộ ghi dịch 4 bit, xác định được giá trị của f_{ht}^* ứng với dây trạng thái đã chọn (hình 9-25,b).

Biểu diễn hàm này trên bảng Karnaugh hình 9-25,c, trong đó các trạng thái không sử dụng được đánh dấu X , trạng thái S_0 được đánh dấu 1 để tránh trạng thái "khoá" của S_0 . Tối thiểu hóa hàm này nhận được kết quả :

$$f_{ht}^* = C \oplus D + A B \bar{D} + \bar{A} \bar{B} \bar{D}$$

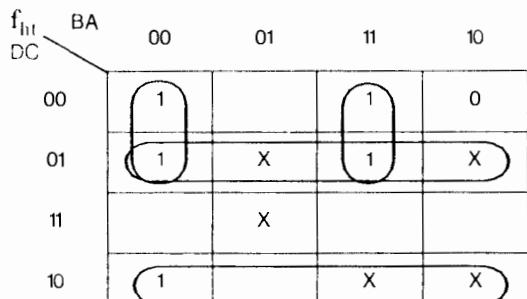
- Bước 4 : Sơ đồ thực hiện mạch được chỉ ra ở hình 9-25,d.



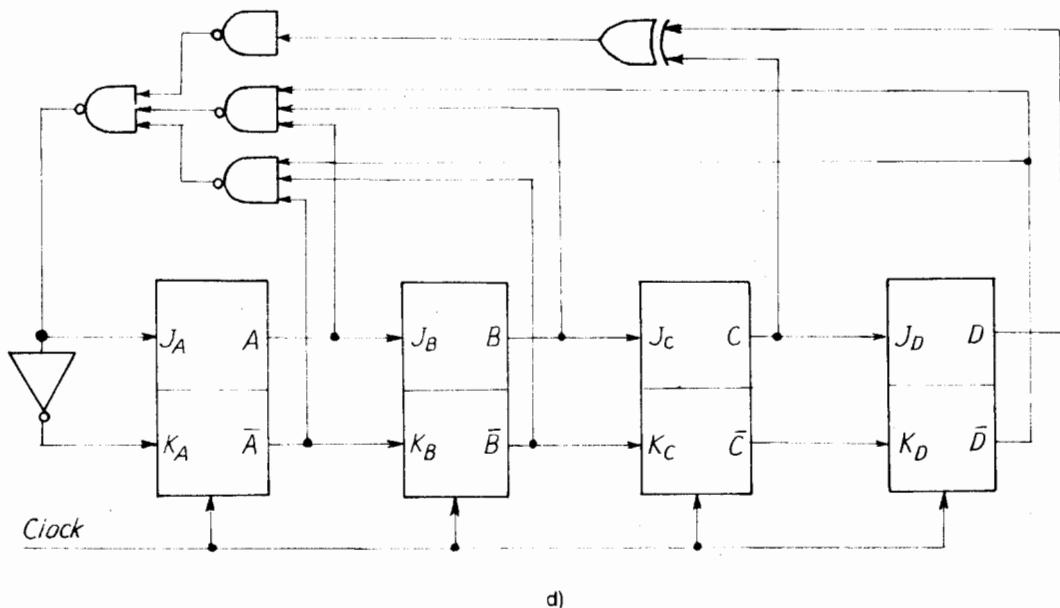
a)

S	D	C	B	A	f_{ht}
S_1	0	0	0	1	0
S_2	0	0	1	0	0
S_4	0	1	0	0	1
S_9	1	0	0	1	1
S_3	0	0	1	1	0
S_7	0	1	1	1	1
S_{15}	1	1	1	1	0
S_{14}	1	1	1	0	0
S_{12}	1	1	0	0	0
S_8	1	0	0	0	0

b)



c)



d)

Hình 9-25. Dùng bộ ghi dịch 4 bit có hàm hồi tiếp cộng modul để thiết kế bộ đếm có $K_d = L = 10$.

- a. Đồ hình chuyển đổi trạng thái ; b. Bảng trạng thái và giá trị tương ứng của f_{ht} ;
- c. Bảng Karnaugh của f_{ht} ; d. Sơ đồ .

9_9. MỘT SỐ VI MẠCH GHI DỊCH

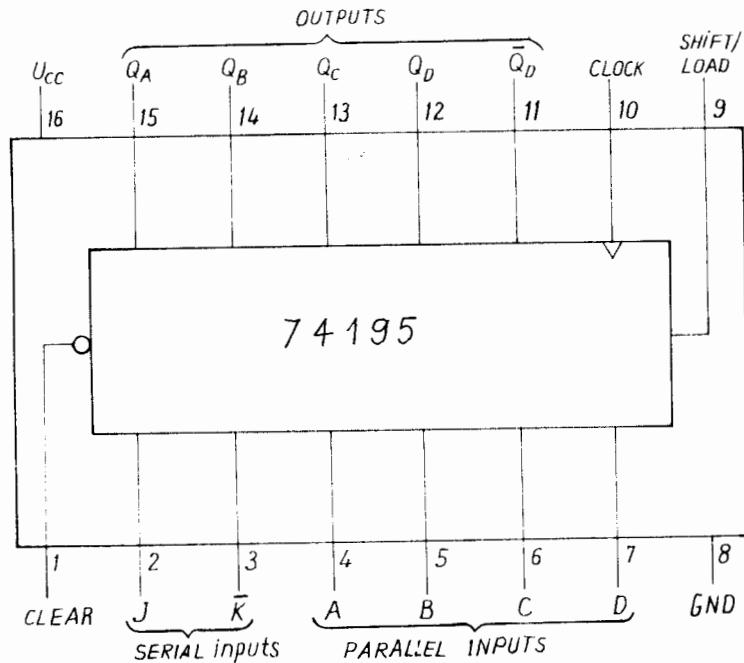
* 54/74 195	4 bit, vào song song hoặc nối tiếp theo kiểu JK ra song song, có đầu vào xoá
54/74 S 195	
54/74 LS 195 A	
* 54/74 L 99	Giống như trên không có đầu vào xoá
* 54/74 91 A	
54/74 L 91	8 bit, vào ra nối tiếp.
54/74 LS 91	
* 54/74 96	5 bit, vào song song hoặc nối tiếp, ra song song, có đầu vào xoá.
54/74 L 96	
54/74 LS 96	
* 54/74 164	8 bit, vào nối tiếp, ra song song, có đầu vào xoá, sử dụng RS - FF
54/74 L 164	
54/74 LS 164	
* 54/74 199	8 bit, vào song song hoặc nối tiếp theo kiểu JK ra song song, có đầu vào xoá

Phân tích vi mạch 54/74195

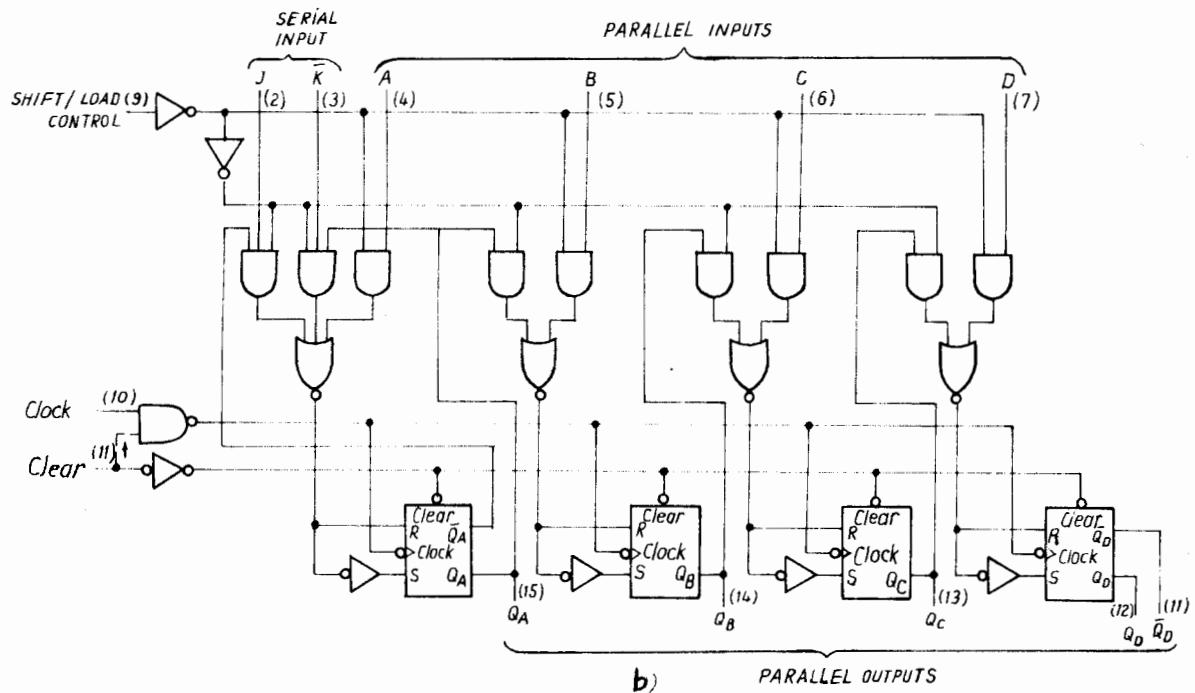
Vi mạch 74195 là bộ ghi dịch 4 bit, vào song song (trên các đầu vào A, B, C, D)

hoặc nối tiếp (J, K), ra song song (Q_A, Q_B, Q_C, Q_D) hoặc nối tiếp (Q_D và \bar{Q}_D). Bộ ghi dịch này có thể dịch phải (theo hướng từ bit trẻ nhất A đến bit già nhất D).

Sơ đồ chân của vi mạch 74195 được biểu diễn trên hình 9-26,a, sơ đồ nguyên lý hình 9-26,b và giản đồ thời gian hình 9-26,c. Ck tích cực ở sườn dương.

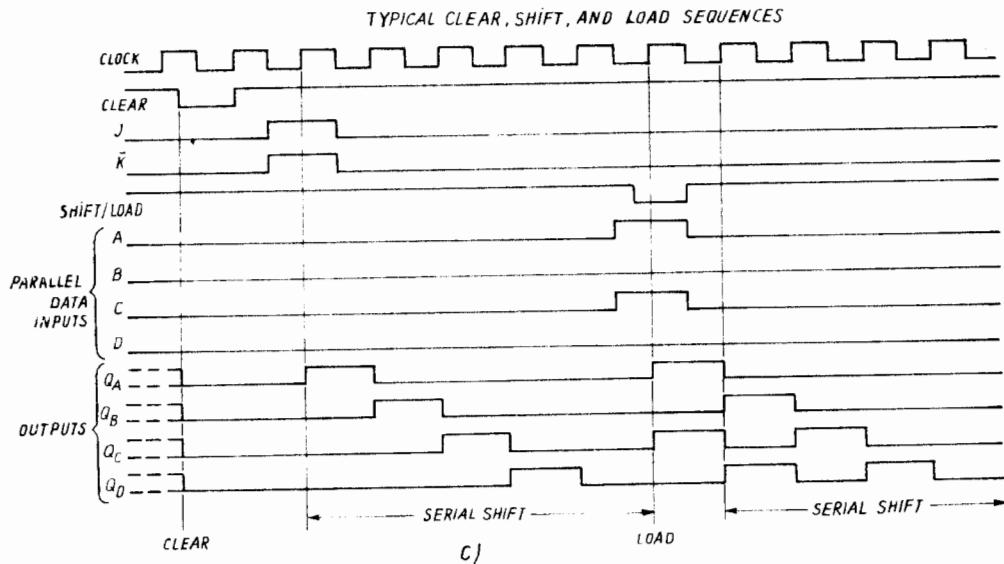


a)



b)

Hình 9-26. Vi mạch 74195.



Hình 9.26 . Vi mạch 74195.

a. Sơ đồ chân; b. Sơ đồ nguyên lý; c. Giản đồ dạng sóng.

- Shift/Load là tín hiệu để điều khiển dịch, nạp số liệu vào. Cụ thể :
 - + Shift/Load = " L" nạp số liệu vào song song, số liệu vào nối tiếp bị cấm.
 - + Shift/Load = " H " dịch phải, dữ liệu nối tiếp được đưa vào trên các đầu J, \bar{K} .

BÀI TẬP

9.1. Nội dung thanh ghi vào nối tiếp $DCBA = 0101$, với A là cột có trọng số nhỏ nhất. Dãy đầu vào là 10011, được nạp tuần tự vào thanh ghi từ trái qua phải, bit đầu tiên là bit có trọng số nhỏ nhất. Hãy vẽ đồ thị thời gian của 4 đầu ra của các FF A, B, C, D sau 5 xung nhịp.

9.2. Thiết kế bộ đếm $K_d = 12$ dùng thanh ghi dịch và mạch hồi tiếp.

9.3. Dùng thanh ghi dịch và mạch logic bổ xung để thiết kế bộ tạo dãy nhị phân $0 - 1 - 0 - 0 - 1 - 0 - 1 - 1 - 1 - 0 - 1$.

9.4. Vẽ đồ hình trạng thái thanh ghi sử dụng hàm hồi tiếp sau :

a. Bộ ghi dịch 4 bit, hàm phản hồi $f = B \oplus C$

b. Bộ ghi dịch 5 bit, hàm phản hồi $f = D \oplus E$

9.5. Dùng bộ ghi dịch 3 bit với hàm hồi tiếp là hàm cộng modul để thiết kế mạch tạo 2 dãy tín hiệu có độ dài là 5 và 7 dùng tín hiệu điều khiển là $m : m = 1$ tạo dãy tín hiệu có độ dài là 7, $m = 0$ tạo dãy tín hiệu có độ dài là 5).

9.6. Bộ ghi dịch 3 bit A, B, C có hàm hồi tiếp là cộng modul $B \oplus C$ (với ước A là bit có trọng số nhỏ nhất), được sử dụng để tạo ra dãy tuần hoàn các chữ số thập phân e (2.718282) dưới dạng mã NBCD trên 4 đầu ra P, Q, R, S .

Hãy thiết kế mạch thực hiện chức năng nói trên.

9.7. Một bộ đếm vòng xoắn 4 bit có 8 trạng thái . Vẽ đồ thị dạng sóng trên mỗi đầu ra của các FF . Xây dựng mạch phản hồi cần thiết để bỏ đi trạng thái 1111 (bộ đếm chỉ còn lại 7 trạng thái).

CHƯƠNG 10

MẠCH DÃY ĐỒNG BỘ

Chương này trình bày phương pháp cơ bản phân tích và thiết kế mạch dãy đồng bộ. Mạch dãy đồng bộ là một mạch số bao gồm các mạch tổ hợp và các phần tử nhớ (FF), hoạt động của mạch được đồng bộ bởi xung nhịp C_k . Trong thực tế để giảm nhỏ công suất tiêu thụ, thời gian trễ và để cho mạch thực hiện đơn giản, người ta thường thiết kế sơ đồ sử dụng các $JK - FF$ và các mạch NAND.

Để nắm vững các vấn đề thiết kế mạch dãy đồng bộ, trước tiên nghiên cứu vấn đề phân tích một mạch dãy đồng bộ. Bài toán phân tích và bài toán thiết kế là 2 bài toán ngược nhau và kinh nghiệm cho thấy bao giờ bài toán phân tích cũng đơn giản, dễ hơn bài toán thiết kế.

10_1. PHÂN TÍCH MẠCH DÃY ĐỒNG BỘ

10_1.1. CÁC BƯỚC PHÂN TÍCH MỘT MẠCH DÃY ĐỒNG BỘ

Như đã biết, bài toán phân tích là bài toán xác định chức năng của một mạch cho trước. Khi tiến hành phân tích một mạch dãy đồng bộ cần tiến hành theo các bước như trên hình 10-1.

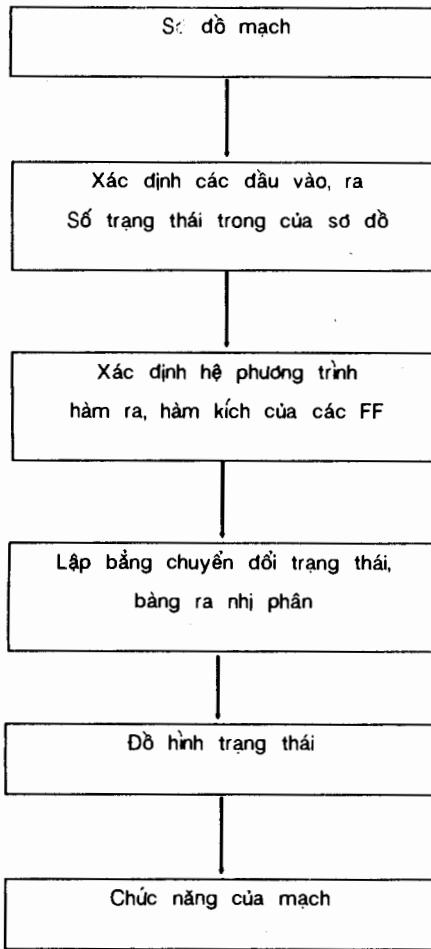
1) Sơ đồ mạch

Từ sơ đồ cho trước cần xác định chức năng từng phần tử cơ bản của sơ đồ, mối liên hệ giữa các phần tử đó.

2) Xác định các đầu vào, đầu ra, số trạng thái trong của mạch

Coi mạch như một " hộp đen " cần phải xác định số đầu vào, đầu ra của mạch, đặc điểm của các đầu vào, đầu ra đó. Để xác định được số trạng thái trong của mạch chúng ta cần phải xác định xem mạch được xây dựng từ bao nhiêu phần tử nhớ (các $JK - FF$) từ đó xác định được số trạng thái trong có thể có của mạch :

Gọi số FF là $n \rightarrow$ số trạng thái trong có thể có của mạch là 2^n .



Hình 10-1. Các bước phân tích một mạch dãy đồng bộ.

3) Xác định phương trình hàm ra, hàm kích của các FF

Dựa vào sơ đồ cho trước, chỉ ra được hệ phương trình hàm ra, hàm kích của các FF.

4) Lập bảng trạng thái, bảng ra nhị phân

Bảng trạng thái, bảng ra nhị phân là bảng biểu diễn quan hệ giữa trạng thái chuyển biến tới, tín hiệu ra nhị phân với trạng thái hiện tại và tín hiệu vào tương ứng.

Dựa vào các phương trình hàm kích, hàm ra đã xác định được ở trên, và dựa vào phương trình đặc tính của FF xác định được trạng thái chuyển biến tới và tín hiệu ra tương ứng với tín hiệu vào và trạng thái hiện tại của mạch.

5) Đồ hình trạng thái

Từ bảng trạng thái và bảng ra nhị phân đã lập được ở trên, xây dựng được đồ hình trạng thái và tín hiệu ra của mạch.

6) *Chức năng của mạch*

Sau khi lập được đồ hình trạng thái, dựa vào đồ hình đó xác định được chức năng của mạch. Cụ thể là từ sự phân tích đồ hình đó mà rút ra được chức năng của mạch.

Để làm sáng tỏ các bước trên, xét ví dụ sau :

10-1.2. VÍ DỤ

Phân tích một mạch dãy đồng bộ có sơ đồ được biểu diễn trên hình 10-2,a.

Bước 1 : Sơ đồ mạch trên hình 10-2,a.

Mạch có 2 đầu vào đó là :

- Tín hiệu vào X

- Xung nhịp tác động C_k .

Mạch có 1 đầu tín hiệu ra Z , sử dụng hai phần tử nhớ là hai $JK - FF$ (A và B).

Bước 2 : Xác định đầu ra, đầu vào, số trạng thái trong của mạch :

Mạch này có thể được biểu diễn bằng một "hộp đen" có 2 đầu vào là tín hiệu vào X và xung nhịp C_k . Có một đầu ra Z . Do mạch được xây dựng từ hai FF nên số trạng thái trong có thể có của mạch là 4. Cụ thể là :

$AB = 00, 01, 11, \text{ và } 10$ (hình 10-2,b).

Bước 3 : Xác định phương trình hàm ra và hàm kích cho các FF :

Từ sơ đồ cho trước (hình 10-2,a) rút ra được :

- Phương trình hàm ra :

$$Z = A \cdot B \cdot C_k \quad (10-1)$$

- Phương trình hàm kích cho các $FF A$ và B

$$\begin{aligned} J_A &= B \\ K_A &= 1 \\ J_B &= \overline{A} \\ K_B &= \overline{X} \cdot \overline{A} = \overline{X} + A \end{aligned} \quad (10-2)$$

Bước 4 . Bảng trạng thái, bảng ra nhị phân :

Dựa vào bảng trạng thái của $JK - FF$ (hình 10-2,c) có thể rút ra phương trình đặc tính của $JK - FF$ theo bảng Karnaugh hình 10-2,d :

$$Q' = J \overline{Q} + \overline{K} Q \quad (10-3)$$

trong đó Q là trạng thái hiện tại và Q' là trạng thái tiếp theo.

Thay J_A, K_A, J_B, K_B ở phương trình (10-2) vào phương trình (10-3) nhận được kết quả :

$$A' = \overline{A} \cdot B \quad (10-4)$$

$$B' = \overline{A} \cdot B \cdot X + \overline{A} \cdot \overline{B}$$

trong đó A', B' là trạng thái tiếp theo của A và B .

Dựa vào hệ phương trình chuyển đổi trạng thái (10-4) và phương trình hàm ra (10-1)

lập được bảng chuyển đổi trạng thái và bảng ra nhị phân (hình 10-2,e) theo trạng thái hiện tại (A, B) và tín hiệu vào X .

Tín hiệu ra $Z = A \cdot B \cdot C_k$.

$Z = 1$ khi và chỉ khi cả A và B đều bằng một và có xung nhịp tác động. Cho nên khoảng thời gian $Z = 1$ bao giờ cũng bằng khoảng thời gian tồn tại của xung nhịp (C_k).

Bước 5. Đồ hình trạng thái.

Từ bảng trạng thái, bảng ra nhị phân ở hình 10-2,e xây dựng được đồ hình trạng thái như hình 10-2,f (mô hình Mealy). Đồ hình gồm 4 trạng thái trong: S_0, S_1, S_2, S_3 . Các $JK - FF$ hoạt động với sườn âm của xung nhịp (sườn sau). Nhìn vào đồ hình trạng thái (hình 10-2,f) nhận thấy ở trạng thái trong S_2 ($AB = 11$) khi có C_k mạch sẽ đưa tín hiệu ra $Z = 1$. Điều này được mô tả trong hình 10-2,g như sau :

Mạch chuyển từ trạng thái $S_1 \rightarrow S_2$ tại thời điểm sườn sau (sườn âm) của xung nhịp thứ nhất. Trước khi có sườn âm của xung nhịp thứ 2 mạch vẫn ở trạng thái đó. Khi sườn âm của xung nhịp thứ 2 xuất hiện mạch sẽ chuyển từ trạng thái S_2 chuyển sang S_0 ($S_2 \rightarrow S_0$) và cho ra tín hiệu ra $Z = 1$.

Bước 6. Chức năng của mạch :

Nhìn vào đồ hình chuyển đổi trạng thái ta thấy có 2 con đường chuyển đổi trạng thái ($S_0 \rightarrow S_0$) là :

$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$ và $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$.

- Theo con đường $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$, tín hiệu ra $Z = 1$ sẽ được đưa ra cùng với thời điểm có xung nhịp thứ 3.

- Còn theo con đường $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$ sẽ không cho tín hiệu ra (hay $Z = 0$).

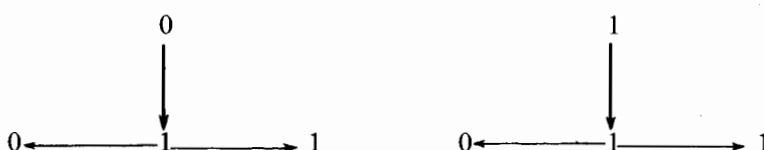
Theo con đường $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$, tín hiệu ra $Z = 1$ có sự phân tích sau :

Chuyển đổi trạng thái đầu tiên từ S_0 đến S_1 chỉ nhờ tác động của xung nhịp, không phụ thuộc vào tín hiệu vào $X = 1$, hay $X = 0$ (C_k).

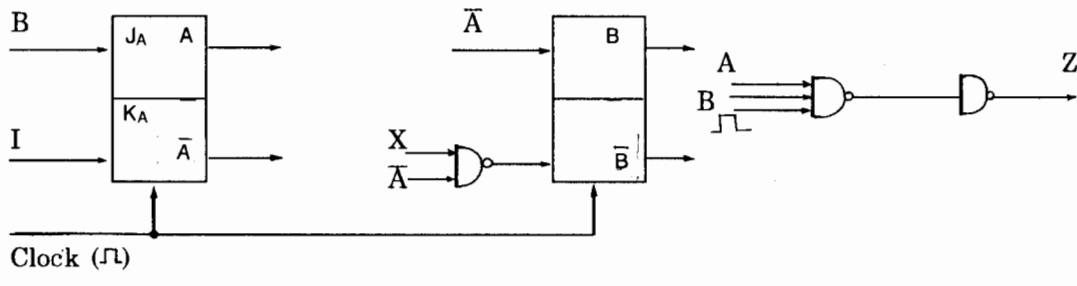
Chuyển đổi trạng thái thứ 2 từ $S_1 \rightarrow S_2$ nhờ tác động của tín hiệu vào $X = 1$ và xung nhịp C_k (X, C_k).

Còn chuyển đổi trạng thái thứ 3 từ $S_2 \rightarrow S_0$ chỉ nhờ tác động của xung nhịp và không phụ thuộc vào tín hiệu vào $X = 1$ hay $X = 0$ (C_k).

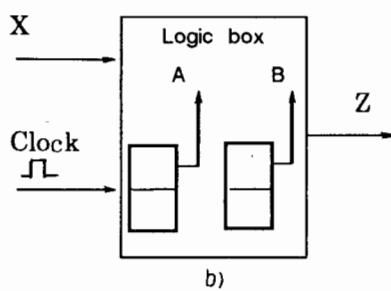
Như vậy mạch chỉ đưa ra tín hiệu ra $Z = 1$ khi đường chuyển đổi qua S_2 tức là mạch chỉ đưa ra tín hiệu ra $Z = 1$ khi dây tín hiệu vào X có dạng 010, 011, 110 hoặc 111. Có thể biểu diễn dây tín hiệu vào X để mạch có tín hiệu ra $Z = 1$ như sau :



Nói tóm lại mạch cho ở hình 10-2,a có chức năng kiểm tra dây tín hiệu X vào ở dạng chuỗi có độ dài bằng 3. Nếu chuỗi tín hiệu vào có dạng là một trong bốn dây : 010, 011, 110, 111, mạch sẽ cho tín hiệu ra $Z = 1$ tại thời điểm có xung nhịp thứ 3. Độ rộng của tín hiệu ra Z đúng bằng độ rộng của xung nhịp. Có thể thấy rõ điều đó trên đồ hình dạng sóng hình 10-2,h.



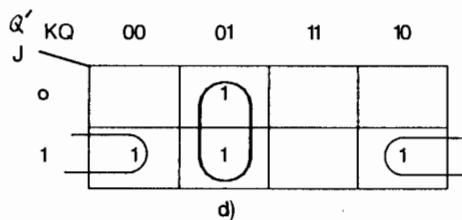
a)



b)

J	K	Q	$Q^{t+\delta t}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

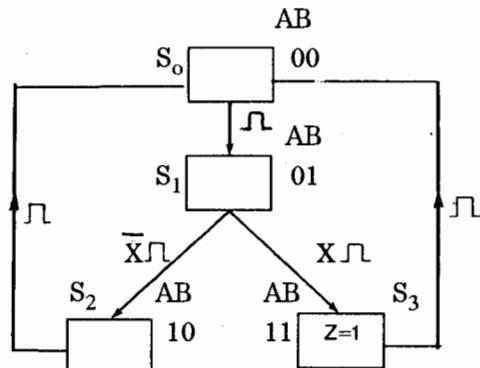
c)



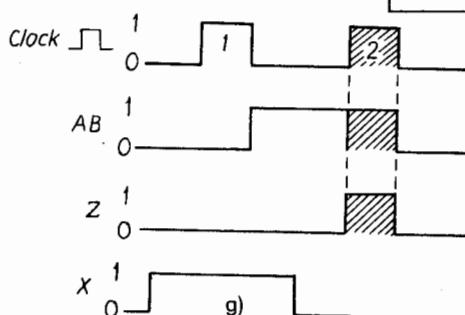
d)

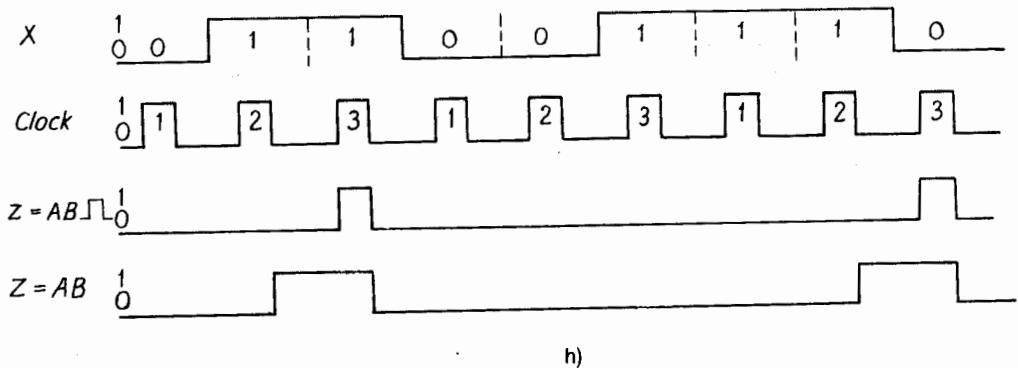
Trạng thái hiện tại	Trạng thái tiếp theo		Tín hiệu ra	
	X = 0 AB	X = 1 AB	X = 0 Z	X = 1 Z
S ₀	01	01	0	0
S ₁	10	11	0	0
S ₂	00	00	1	1
S ₃	00	00	0	0

e)



f)





Hình 10-2. Phân tích mạch dây đồng bộ.

- a. Sơ đồ mạch; b. Sơ đồ khối; c. Bảng chức năng của JK-FF ; d. Bảng Karnaugh cho phương trình đặc tính của JK-FF ; e. Bảng trạng thái và bảng ra nhị phân ; f. Đồ hình trạng thái ; g. Dạng sóng của tín hiệu ra ; h. Dạng sóng của mạch.

10.2. CÁC BƯỚC THIẾT KẾ MẠCH DÂY ĐỒNG BỘ

10.2.1. CÁC BƯỚC THIẾT KẾ MẠCH DÂY ĐỒNG BỘ

Quá trình phân tích một mạch dây đồng bộ trong phần trước cho thấy : để thiết kế một mạch dây đồng bộ tiến hành các bước theo thứ tự ngược với bài toán phân tích.

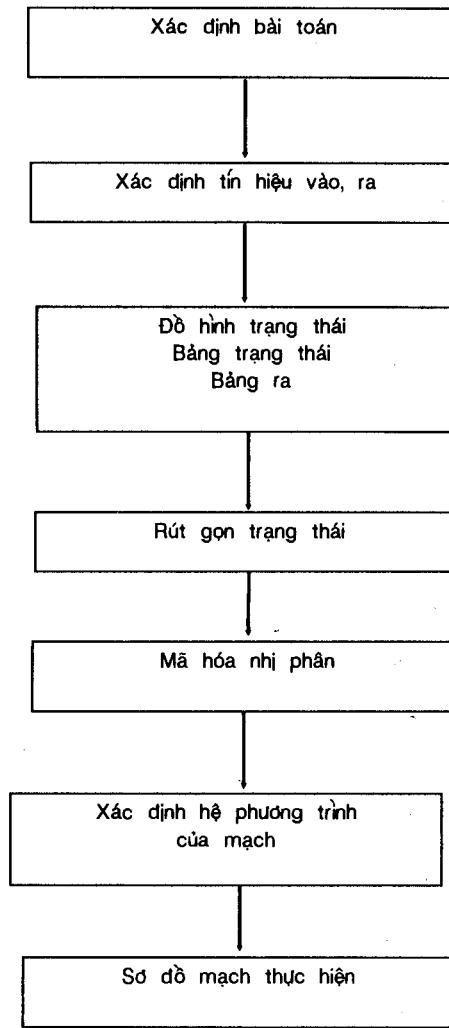
Tuy nhiên trong thiết kế còn có một đặc điểm hết sức quan trọng là mạch được thiết kế phải thực hiện chức năng cho trước với sơ đồ càng đơn giản càng tốt. Điều này không được đề cập đến trong quá trình phân tích.

Dễ dàng nhận thấy rằng : số trạng thái trong của mạch gần như luôn luôn tỷ lệ với số FF được dùng trong mạch. Vì vậy muốn thiết kế được mạch đơn giản thì phải giảm số trạng thái trong của mạch (nếu có thể) đến mức nhỏ nhất.

Tóm lại, để thiết kế một mạch dây đồng bộ ngoài các bước đã trình bày trong phần phân tích phải bổ sung thêm bước tối thiểu hóa trạng thái (hình 10-3).

Bước 1 : Xác định bài toán

Đây là bước đầu tiên và cũng là bước quan trọng để thực hiện tốt việc thiết kế mạch. Trước tiên phải xác định được yêu cầu đặt ra cho mạch thực hiện và phải xác định mạch được thiết kế từ những yếu tố nào? Trong bước này chúng ta không nên nhận vắn để một cách quá phức tạp mà phải xử lý thật linh hoạt để đưa ra phương pháp thực hiện tối ưu.



Hình 10-3. Các bước thiết kế mạch dây đồng bộ.

Bước 2 : Xác định tín hiệu vào ra :

Cũng như ở phần phân tích mạch dây đồng bộ đã nêu trên, coi mạch như một " hộp đen" có các đầu vào, ra như hình 10-2,b. Cần xác định được những đặc điểm cơ bản của các đầu vào ra đó.

Bước 3 : Đồ hình trạng thái, bảng trạng thái, bảng ra :

Nói chung việc xây dựng đồ hình trạng thái không dựa trên một quy tắc cụ thể nào mà phần lớn dựa vào kinh nghiệm. Phương pháp chung để giải loại bài toán này là dựa vào yêu cầu của bài toán đặt ra (tín hiệu vào, tín hiệu ra), từ đó lập được đồ hình trạng thái, xác định được điều kiện để cho mạch có thể chuyển đổi từ trạng thái này sang trạng thái tiếp theo và khi nào thì mạch cho tín hiệu ra.

Từ đồ hình trạng thái lập được ở trên, xây dựng được bảng trạng thái và bảng tín hiệu ra.

Bước 4 : Rút gọn trạng thái (tối thiểu hoá trạng thái)

Như đã biết số trạng thái trong của mạch gần như tỷ lệ với số FF được dùng trong mạch.

Do thiếu kinh nghiệm, nói chung ban đầu đồ hình lập ra chưa được đơn giản, cho nên nếu có thể cần phải giảm bớt (tối thiểu) các trạng thái để cho mạch có số trạng thái trong là nhỏ nhất.

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. Bài toán tối thiểu hoá trạng thái thực chất là tìm số lớp các trạng thái tương đương. Các phương pháp tối thiểu hoá trạng thái sẽ được trình bày chi tiết ở §10. 5.

Bước 5 : Mã hoá nhị phân (mã hoá trạng thái)

Sau khi tối thiểu hoá các trạng thái và đưa ra đồ hình trạng thái tương ứng, dùng các biến nhị phân để mã hoá cho các trạng thái trong đó. Số biến nhị phân dùng để mã hoá các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch.

Nếu số lượng trạng thái trong là N , số lượng biến nhị phân cần dùng là n , thì n phải thỏa mãn điều kiện :

$$n \geq \log_2 N .$$

Có rất nhiều cách mã hoá khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Vấn đề là phải mã hoá sao cho sơ đồ mạch thực hiện là đơn giản nhất.

Vấn đề mã hoá và phương pháp mã hoá trạng thái trong của mạch dây đồng bộ sẽ được xem xét một cách cụ thể ở mục 10_6.

Bước 6 : Xác định hệ phương trình của mạch

Sau khi đã mã hoá được các trạng thái trong của mạch, có thể xác định được hệ phương trình của mạch. Có hai cách xác định hệ phương trình của mạch, cụ thể là :

- Lập bảng chuyển đổi trạng thái và tín hiệu ra nhị phân, từ đó xác định được phương trình đầu vào kích cho các FF và phương trình của tín hiệu ra, sau đó chúng ta tiến hành tối thiểu hoá các phương trình đó.

- Dựa trực tiếp vào đồ hình trạng thái, viết hệ phương trình T_{on} , T_{off} của các FF và phương trình của tín hiệu ra, sau đó tiến hành tối thiểu hệ phương trình đó.

Bước 7 : Sơ đồ mạch thực hiện

Từ hệ phương trình đã xác định được, vẽ sơ đồ của mạch.

10 2.2. MỘT VÍ DỤ VỀ THIẾT KẾ MẠCH DÂY ĐỒNG BỘ

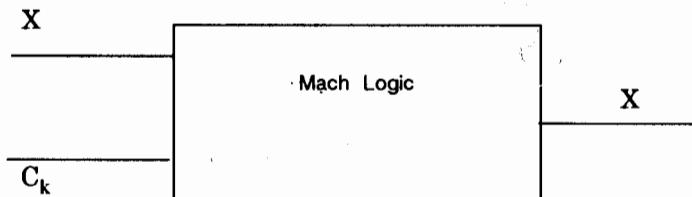
Thiết kế một mạch dây đồng bộ thực hiện nhiệm vụ kiểm tra dãy tín hiệu vào ở dạng nhị phân có độ dài bằng 3 được đưa vào liên tiếp trên đầu vào X . Nếu dãy tín hiệu vào có dạng là 010 hoặc 011 hoặc 110 hoặc 111, $Z = 1$. Trong các trường hợp khác $Z = 0$.

Bước 1 : Xác định bài toán

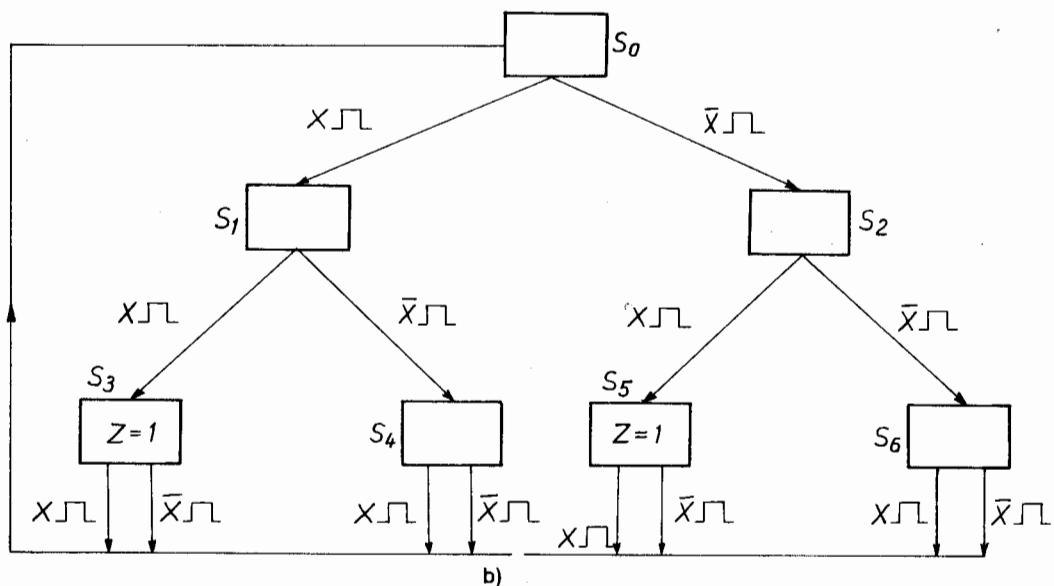
Mạch thiết kế có nhiệm vụ phát hiện dãy tín hiệu vào : nếu dãy tín hiệu vào có dạng là 010 hoặc 011 hoặc 110 hoặc 111 thì $Z = 1$ để báo hiệu là mạch đã nhận được một trong các dãy tín hiệu vào đó.

Bước 2 : Xác định tín hiệu vào, ra :

Mạch phải thiết kế là mạch đồng bộ, nên ngoài đầu vào X còn có đầu vào xung nhịp C_k , một đầu tín hiệu ra là Z (hình 10-4,a).



a)



b)

S	S'		Z	
	X = 0	X = 1	X = 0	X = 1
S ₀	S ₂	S ₁	0	0
S ₁	S ₄	S ₃	0	0
S ₂	S ₆	S ₅	0	0
S ₃	S ₀	S ₀	1	1
S ₄	S ₀	S ₀	0	0
S ₅	S ₀	S ₀	1	1
S ₆	S ₀	S ₀	0	0

c)

Hình 10-4. a) Mô tả các đầu tín hiệu vào ra của mạch; b) Đồ họa trạng thái của mạch,
c) Bảng trạng thái, bảng ra.

Bước 3 : Đồ hình trạng thái, bảng trạng thái, bảng ra :

Dựa vào các yêu cầu của tín hiệu vào, ra lập được đồ hình chuyển đổi trạng thái như hình 10-4,b.

Có thể giải thích đồ hình chuyển đổi trạng thái như sau :

- Trạng thái ban đầu là S_0 :

Khi tín hiệu vào là \bar{X} . C_k mạch sẽ chuyển đến trạng S_1 .

Khi tín hiệu vào là \bar{X} . C_k mạch sẽ chuyển đến trạng S_2 .

- Tương tự với trạng thái S_1 , mạch sẽ chuyển tới trạng thái S_3 khi tín hiệu vào là X . C_k và tới S_4 khi tín hiệu vào là \bar{X} . C_k .

- Ở trạng thái S_2 mạch sẽ chuyển tới trạng thái S_5 khi tín hiệu vào là \bar{X} . C_k và tới S_6 khi tín hiệu vào là \bar{X} . C_k .

- Nếu mạch ở 1 trong 4 trạng thái S_3 , S_4 , S_5 , S_6 : khi có tín hiệu vào là X . C_k hoặc X . C_k mạch sẽ chuyển về trạng thái ban đầu S_0 . Vậy khi dãy tín hiệu vào là 110 hoặc 111 (đường chuyển đổi trạng thái $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$) hay khi dãy tín hiệu vào là 010 hoặc 011 (đường $S_0 \rightarrow S_2 \rightarrow S_5 \rightarrow S_0$) thì mạch sẽ cho tín hiệu ra $Z = 1$ tại thời điểm của xung nhịp thứ 3. Với các đường chuyển đổi khác $Z = 0$. Từ đồ hình trạng thái lập được bảng trạng thái, bảng ra như hình 10-4,c.

Bước 4 : Tối thiểu hoá trạng thái

So sánh đồ hình 10-4,c với đồ hình của chính mạch đó trong [hình 10-2, f](#) nhận thấy rằng : đồ hình 10-4,c phức tạp hơn, có số trạng thái trong lớn hơn. Do vậy phải tiến hành tối thiểu hoá trạng thái.

Trong phần này sẽ giới thiệu phương pháp tối thiểu hoá Caldwell. Cơ sở lý thuyết của việc tối thiểu hoá là dựa vào khái niệm các trạng thái tương đương.

Định nghĩa các trạng thái tương đương :

Trạng thái S_i được gọi là tương đương với trạng thái S_j ($S_i \approx S_j$) khi và chỉ khi nếu lấy S_i và S_j là 2 trạng thái ban đầu thì với mọi dãy tín hiệu vào có thể có chúng luôn luôn cho dãy tín hiệu ra giống nhau.

Nếu có nhiều trạng thái tương đương với nhau từng đôi một thì chúng tương đương với nhau (tính chất bắc cầu). Để kiểm tra một nhóm các trạng thái xem chúng có tương đương với nhau không, có thể sử dụng bảng trạng thái và tín hiệu ra như sau :

- Nhóm các trạng thái tương đương phải có những hàng trong bảng tín hiệu ra giống nhau.

- Nhóm các trạng thái tương đương phải có những hàng trong bảng trạng thái ở cùng một cột (ứng với cùng một tổ hợp tín hiệu vào) là tương đương. Nghĩa là ứng với cùng một tổ hợp tín hiệu vào các trạng thái sẽ chuyển biến tới của chúng là tương đương.

Điều này cho thấy thủ tục kiểm tra tính tương đương của một nhóm các trạng thái phải tiến hành tuần tự từng bước cho đến nhóm trạng thái cuối cùng. Nếu nhóm trạng thái cuối cùng này là tương đương thì nhóm trạng thái được kiểm tra kiểm tra là tương đương.

Quy tắc Caldwell :

Những hàng (tương ứng với trạng thái trong) của bảng chuyển đổi trạng thái và tín hiệu ra, sẽ được kết hợp với nhau và được biểu diễn 1 hàng chung - đặc trưng (trạng thái đặc trưng) cho chúng nếu như chúng thỏa mãn cả 2 điều kiện sau :

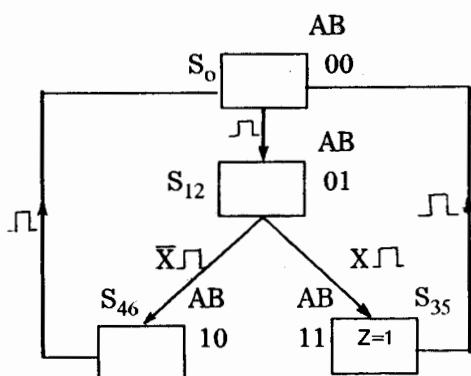
- 1) Các hàng tương ứng trong ma trận ra giống nhau.
- 2) Trong ma trận ra, các hàng tương ứng phải thỏa mãn một trong ba điều sau :
 - Các hàng trong ma trận trạng thái giống nhau.
 - Các trạng thái ở trong cùng một cột nằm trong nhóm trạng thái được xét.
 - Các trạng thái ở trong cùng một cột là các trạng thái tương đương.

X	0	1
S_0	S_2 $Z = 0$	S_1 $Z = 0$
S_1	S_{46} $Z = 0$	S_{35} $Z = 0$
S_2	S_{46} $Z = 0$	S_{35} $Z = 0$
S_{35}	S_o $Z = 1$	S_o $Z = 1$
S_{46}	S_o $Z = 0$	S_o $Z = 0$

d)

X	0	1
S_2	S_{12} $Z = 0$	S_{12} $Z = 0$
S_{12}	S_{46} $Z = 0$	S_{35} $Z = 0$
S_{35}	S_o $Z = 1$	S_o $Z = 1$
S_{46}	S_o $Z = 1$	S_o $Z = 1$

e)



f)

A	B	Mã hóa S
0	0	S_o
0	1	S_{12}
1	1	S_{35}
1	0	S_{46}

g)

Hình 10-4. d) Bảng trạng thái, bảng ra sau khi gộp S_3 với S_5 , S_4 với S_6 ; e) Bảng trạng thái, bảng ra sau khi gộp S_1 với S_2 ; f) Đồ hình chuyển đổi trạng thái tối giản ; g) Bảng mã hóa trạng thái.

Sau khi đã thay thế các trạng thái tương đương bằng một trạng thái chung đặc trưng cho chúng, lặp lại các công việc tìm các trạng thái tương đương (các hàng tương đương)

khác, tới khi nào không thể tìm được các hàng (các trạng thái) tương đương với nhau nữa thì dừng lại. Số trạng thái trong bảng trạng thái và tín hiệu ra lúc đó là tối thiểu.

Nhược điểm của phương pháp này là khi số trạng thái (số hàng) của bảng trạng thái và tín hiệu ra là quá lớn thì công việc tối thiểu hoá mất nhiều thời gian.

Áp dụng quy tắc Caldwell vào bài toán trên, trong bảng hình 10-4,c trạng thái S_4 tương đương với S_6 ($S_4 \approx S_6$), S_3 tương đương với S_5 ($S_3 \approx S_5$). Thay thế các trạng thái tương đương bằng một trạng thái chung đặc trưng cho chúng. Ví dụ thay thế S_4, S_6 bởi S_{46} ; thay thế S_3, S_5 bởi S_{35} . Từ đó lập được bảng trạng thái, bảng ra trên hình 10-4,d.

Nhìn vào bảng 10-4,d ta thấy ngay S_1 tương đương với S_2 , gộp S_1 và S_2 ta được bảng như hình 10-4,e

Ta không thể tìm thấy 2 trạng thái nào tương đương trong bảng ở hình 10-4,e. Buộc tối thiểu hoá kết thúc ở đây với số trạng thái tối thiểu là 4 gồm $S_0, S_{12}, S_{35}, S_{46}$.

Từ bảng trạng thái, bảng ra đã tối thiểu hình 10-4,e có thể xây dựng được đồ hình trạng thái như hình 10-4,f.

Bước 5: Mã hoá trạng thái :

Sau khi đã tối thiểu hoá mạch còn 4 trạng thái $S_0, S_{12}, S_{35}, S_{46}$. Mã hoá bốn trạng thái này bằng hai biến nhị phân A, B như hình 10-4,f (Nếu thiết kế mạch dùng FF ta sẽ dùng 2 FF ký hiệu A, B tương ứng với 2 biến nhị phân trên)

Bước 6 : Xác định hệ phương trình của mạch.

Có 2 cách xác định hệ phương trình của mạch :

a) Cách 1 :

Dựa vào bảng trạng thái, bảng ra để xác định đầu vào kích cho các FF và phương trình tín hiệu ra. Từ bảng hình 10-4,e thay thế các trạng thái trong bởi mã của chúng như bảng 10-4,f nhận được bảng trạng thái, bảng ra nhị phân như hình 10-4,g. Cũng trên bảng này biểu diễn các đầu vào kích của các FF - A và B là J_A, K_A, J_B, K_B tương ứng với các giá trị của tín hiệu vào X . Giá trị của các đầu vào kích đó được xác định theo bảng 10-4,h. J_A, K_A, J_B, K_B được biểu diễn và tối thiểu hoá theo bảng Karnaugh hình 10-4,i. Cụ thể :

$$J_A = B$$

$$K_A = 1$$

$$J_B = \bar{A}$$

$$K_B = \bar{X} + A$$

Phương trình tín hiệu ra Z được xác định dựa vào bảng chuyển đổi trạng thái, bảng ra (hình 10-4,e) và bảng mã hoá như sau : $Z = A \cdot B \cdot C_k$

b) Cách 2 :

Dựa trực tiếp vào đồ hình viết phương trình T_{on}, T_{off} của từng FF và phương trình tín hiệu ra.

Trạng thái hiện tại	Trạng thái tiếp theo		Các đầu vào của FF							
	X = 0	X = 1	X = 0		X = 1		X = 0		X = 1	
AB	AB	AB	J _A	K _A	J _A	K _A	J _A	K _A	J _A	K _A
00	01 Z = 0	01 Z = 0	0	x	0	x	1	x	1	x
01	10 Z = 0	11 Z = 0	1	x	1	x	x	1	x	0
11	00 Z = 1	00 Z = 1	x	1	x	1	x	1	x	1
10	00 Z = 0	00 Z = 0	x	1	x	1	0	x	0	x

h)

Q	Q'	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

i)

X	AB	00	01	11	10
0		1	x		x
1		1	x		x

X	AB	00	01	11	10
0		x	x	1	1
1		x	x	1	1

$$J_A = B$$

X	AB	00	01	11	10
0		1	x	x	
1		1	x	x	

$$K_A = \bar{A}$$

X	AB	00	01	11	10
0		x	1	1	x
1		x		1	x

$$J_B = \bar{A}$$

$$K_B = \bar{X} + A$$

k)

Hình 10-4, h) Bảng trạng thái nhị phân và giá trị tương ứng các đầu vào kích của JK-FF; i) Bảng giá trị đầu vào kích của JK - FF; k) Bảng Karnaugh của các hàm kích FF.

Đổi với JK - FF nếu :

$$T_{on Q} = T^* \cdot \bar{Q}$$

$$\Rightarrow J_Q = T^*$$

$$T_{off Q} = T^{**} \cdot Q$$

$$\Rightarrow K_Q = T^{**}$$

Khi đó, với ví dụ này :

$$T_{on A} = S_{12} X + S_{12} \bar{X} = S_{12} = \bar{A} B \Rightarrow J_K = B$$

$$T_{off A} = S_{35} + S_{46} = A \bar{B} + A B = A \Rightarrow K_A = 1$$

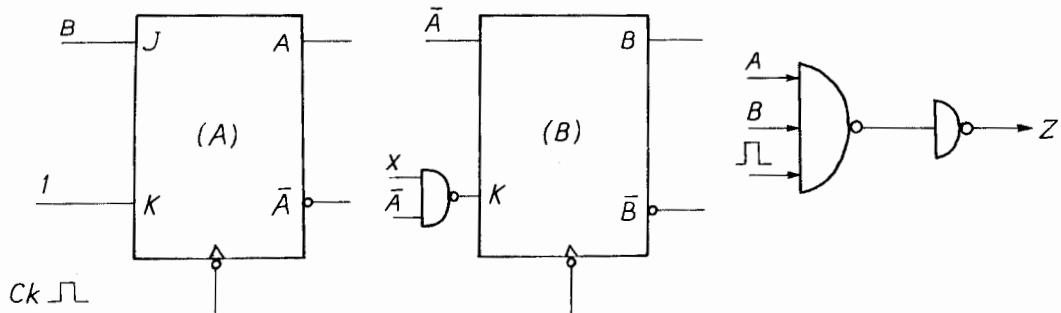
$$T_{on B} = S_o = \bar{A} \bar{B} \Rightarrow J_B = \bar{A}$$

$$T_{off B} = S_{12} \cdot \bar{X} + S_{35} = \bar{A} B \bar{X} + A B \Rightarrow K_B = \bar{A} \bar{X} + A = \bar{X} + A$$

Viết được phương trình hàm ra $Z = A \cdot B \cdot C_k$

Bước 7 : Sơ đồ mạch thực hiện

Từ phương trình hàm kích của các FF - A, FF - B và phương trình của hàm ra ta xây dựng được sơ đồ mạch thực hiện như hình 10-4,l.



Hình 10-4. l) Sơ đồ mạch.

10_3. MẠCH DÂY ĐỒNG BỘ DÙNG MÔ HÌNH MOORE VÀ MÔ HÌNH MEALY

Để mô tả mạch dây đồng bộ có thể dùng mô hình Mealy hay mô hình Moore. Mô hình Moore là mô hình mà tín hiệu ra chỉ phụ thuộc vào trạng thái trong của các mạch. Mô hình Mealy là mô hình mà tín hiệu ra phụ thuộc vào cả trạng thái trong của mạch và giá trị của tín hiệu vào X.

Ở phần này, qua các ví dụ cụ thể thiết kế mạch cho cả hai mô hình Mealy và Moore giúp ta hiểu rõ hơn về các mô hình này và ưu nhược điểm của từng loại.

10_3.1. VÍ DỤ 1

Thiết kế một mạch dây đồng bộ nhận biết dãy tín hiệu vào. Tín hiệu vào được đưa liên tiếp ở đầu vào của mạch ở dạng nhị phân. Mỗi lần dãy tín hiệu vào là 101, mạch sẽ cho

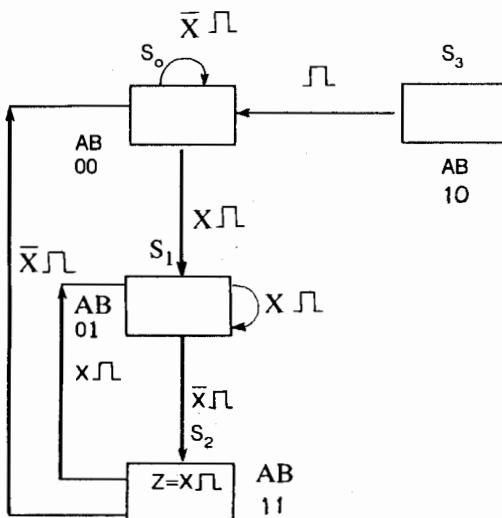
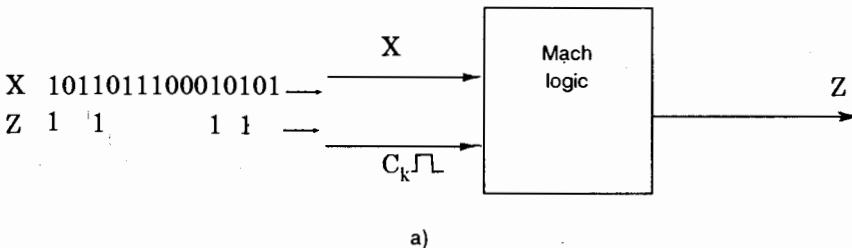
cho ra tín hiệu ra $Z = 1$. Các bit dữ liệu vào được đồng bộ với xung nhịp C_k .

Bước 1 : Xác định bài toán

Ta cần phải thiết kế 1 mạch dãy đồng bộ để phát hiện dãy tín hiệu vào. Nếu dãy tín hiệu vào có dạng 101 thì đầu ra $Z = 1$.

Bước 2 : Xác định tín hiệu vào, ra

Mạch có 2 đầu tín hiệu vào là X và xung nhịp C_k . Mạch có 1 đầu tín hiệu ra là Z (hình 10-5,a).



	X = 0	X = 1
S ₀	S ₀ Z = 0	S ₁ Z = 0
S ₁	S ₂ Z = 0	S ₁ Z = 0
S ₂	S ₀ Z = 0	S ₁ Z = 1
S ₃	S ₀ Z = 0	S ₀ Z = 0

c)

Hình 10-5. Mô phỏng mạch theo mô hình Mealy.

- a) Sơ đồ khối của mạch và tín hiệu ra ứng với dãy tín hiệu vào X (thứ tự của dãy vào X từ bên phải qua trái theo chiều mũi tên); b. Đồ hình chuyển đổi trạng thái;
- c. Bảng trạng thái và tín hiệu ra tương ứng.

Bước 3: Đồ hình trạng thái, bảng trạng thái và tín hiệu ra

Căn cứ vào yêu cầu của bài toán, dễ dàng lập được đồ hình chuyển đổi trạng thái ứng với hai mô hình Mealy và Moore.

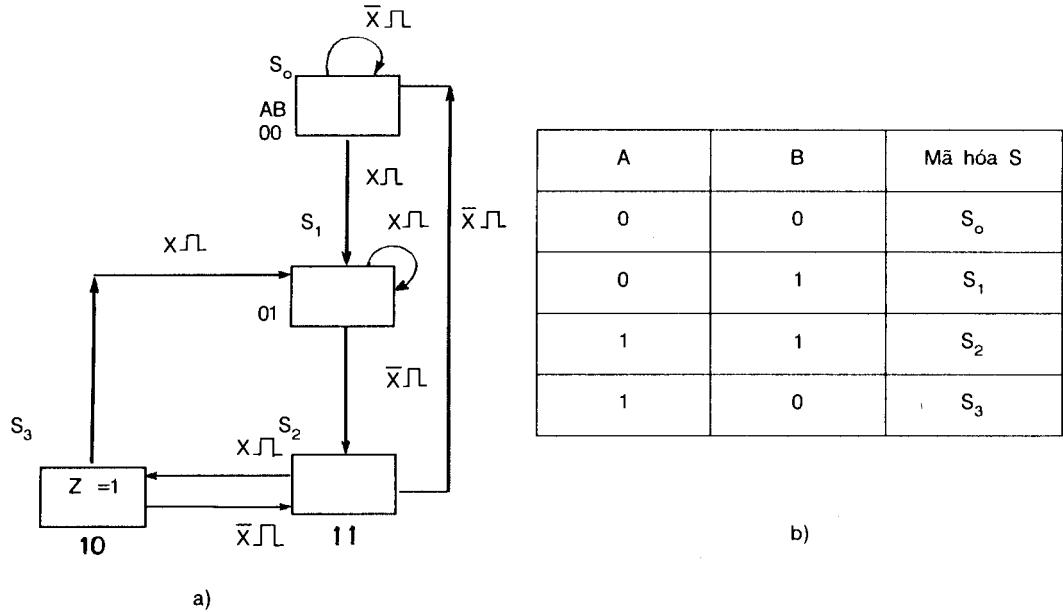
a) Với mạch Mealy hình 10-5,b và hình 10-5,c

Theo yêu cầu của bài toán, mạch chỉ cần sử dụng 3 trạng thái S_0, S_1, S_2 . Để mã hóa 3 trạng thái đó phải dùng ít nhất 2 biến nhị phân và sẽ có một trạng thái thừa không dùng

đến. Để tránh tình trạng mạch rơi vào trạng thái "khoá", thiết kế sao cho nếu mạch rơi vào trạng thái thừa đó thì ngay từ xung nhịp đầu tiên nó sẽ trở về S_0 (tức là trạng thái đúng) như hình 10-5,b. Bảng trạng thái và tín hiệu ra (hình 10-5,c) được xây dựng trực tiếp từ đồ hình 10-5,b.

b) Với mô hình Moore

Áp dụng thuật toán chuyển đổi từ mô hình Mealy sang mô hình Moore (mục 7_4) nhận được đồ hình trạng thái của mô hình Moore ở hình 10-6.



Hình 10-6. Mô hình Moore.

a. Đồ hình trạng thái ; b. Bảng trạng thái, bảng ra.

Với đồ hình của mô hình Moore, trạng thái S_3 chính là trạng thái cho tín hiệu ra $Z = 1$.

Bước 4 : Tối thiểu hoá trạng thái

Với ví dụ này ta có thể thấy ngay rằng số trạng thái trong bảng không thể tối thiểu hơn được nữa, nhưng tất nhiên trong trường hợp tổng quát, tối thiểu hoá trạng thái luôn là bước quan trọng và rất cần thiết khi thiết kế.

Bước 5 : Mã hoá trạng thái

Để mã hoá trạng thái, ta có thể dùng nhiều cách, ví dụ với bài toán này chọn cách mã hoá như bảng sau (hình 10-6,c) :

Cách mã hoá này hoàn toàn thống nhất với cách mã hoá trong hình 10-5,b và 10-6,a.

Bước 6 : Hệ phương trình của mạch (hàm kích các FF, hàm ra).

a) Với mô hình Mealy

$$T_{on A} = S_1 \bar{X} = \bar{A} B \bar{X} \Rightarrow J_A = B \bar{X}$$

A	B	Trạng thái
0	0	S_0
0	1	S_1
1	1	S_2
1	0	S_3

Hình 10-6. c

$$T_{\text{offA}} = S_3 + S_2 \bar{X} + S_2 X = S_3 + S_2$$

$$= A \bar{B} + A B = A$$

$$\Rightarrow K_A = 1$$

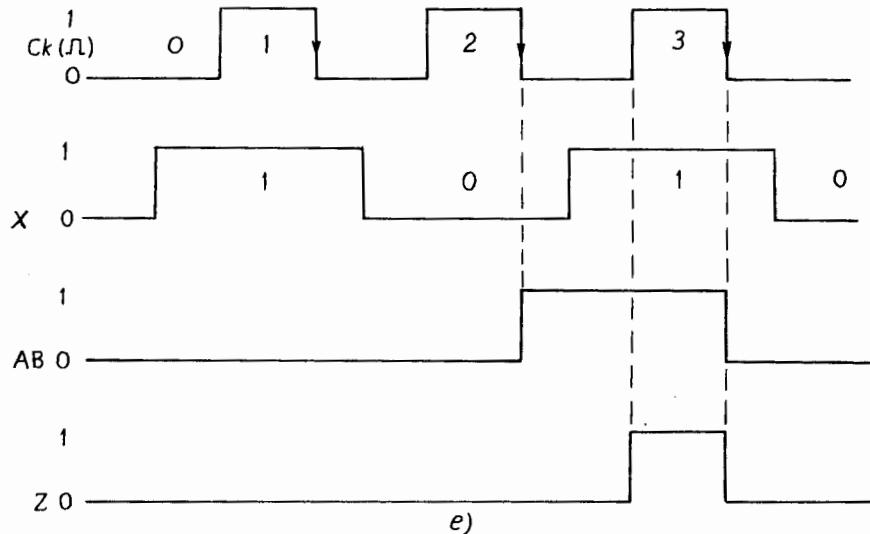
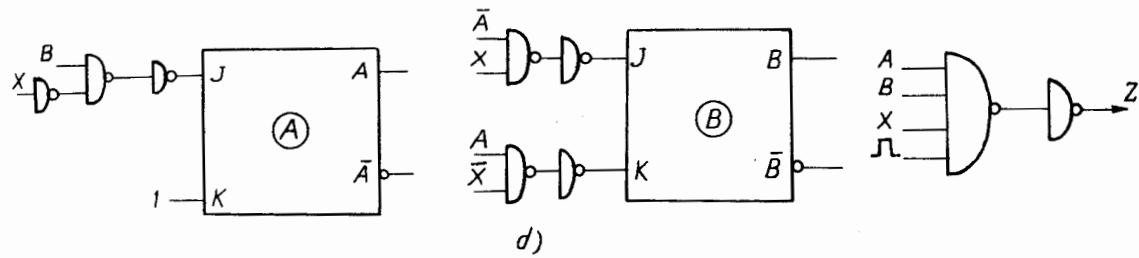
$$T_{\text{onB}} = S_0 X = \bar{A} \bar{B} X$$

$$\Rightarrow J_B = \bar{A} X$$

$$T_{\text{offB}} = S_2 \bar{X} = A B \bar{X}$$

$$\Rightarrow K_B = A \bar{X}$$

$$Z = A, B, X, C_k$$



Hình 10-6. d) Sơ đồ; e) Dạng sóng của mạch dùng mô hình Mealy.

b) Với Moore

$$T_{\text{onA}} = S_1 \bar{X} = \bar{A} B \bar{X} \Rightarrow J_A = B \bar{X}$$

$$T_{\text{offA}} = S_2 \bar{X} + S_3 X = A B \bar{X} + A \bar{B} X$$

$$= A (B \bar{X} + \bar{B} X) \Rightarrow K_A = B \bar{X} + \bar{B} X$$

$$T_{\text{onB}} = S_0 X + S_3 X + S_3 X$$

$$= \bar{A} \bar{B} X + A \bar{B}$$

$$= \bar{B} (\bar{A}X + A)$$

$$\Rightarrow J_B = A + X$$

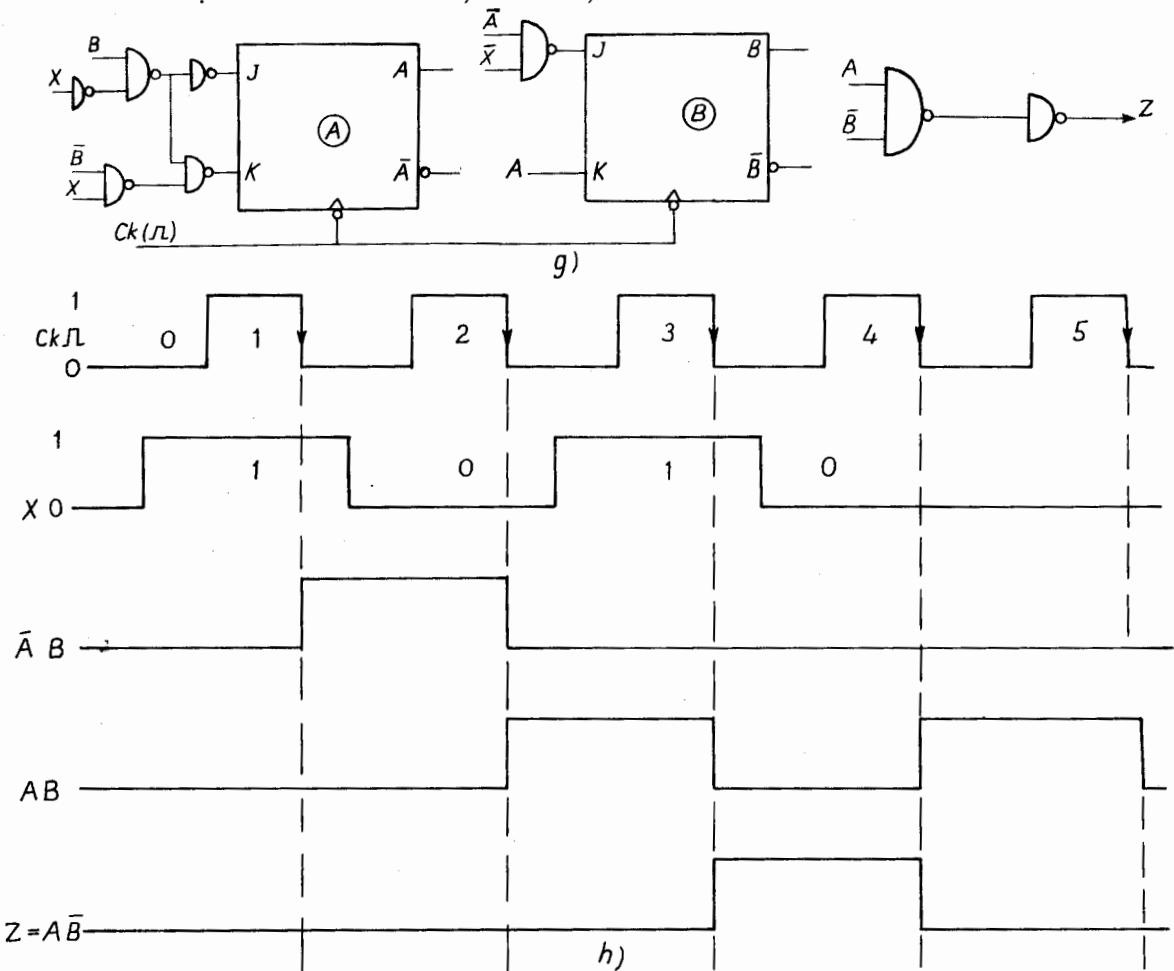
$$T_{off B} = S_2 X + S_2 \bar{X} = S_2$$

$$\Rightarrow K_B = A$$

$$Z = A \bar{B}$$

Bước 7 : Vẽ sơ đồ mạch.

Từ các phương trình hàm kích và hàm ra ở trên dễ dàng xây dựng được sơ đồ thực hiện mạch (hình 10-6,d và 10-6,g). Với mô hình Moore, mạch thực hiện có phức tạp hơn nhưng mạch ra của nó đơn giản hơn so với Mealy. Giản đồ dạng sóng cho mô hình Mealy và Moore được cho trên hình 10-6,e và 10-6,h.



Hình 10-6. g) Sơ đồ ; h) Dạng sóng của mạch dùng mô hình Moore.

10_3.2. VÍ DỤ 2

Thiết kế mạch kiểm tra dãy thông tin nhị phân được đồng bộ với xung nhịp đưa đến đầu vào X của một mạch dây đồng bộ. Nếu dãy tín hiệu vào gồm ba số liên tiếp là 011 hoặc 111 thì mạch sẽ cho tín hiệu ra $Z = 1$.

Bước 1 : Xác định bài toán

Mạch có nhiệm vụ kiểm tra dãy tín hiệu vào. Nếu dãy tín hiệu vào có dạng 011, 111 mạch sẽ cho ra một tín hiệu ra $Z = 1$ ở đầu ra.

Bước 2 : Xác định tín hiệu vào, ra

Ta có thể mô tả toàn bộ mạch bằng một " hộp đen" có hai đầu vào là : X , xung nhịp C_k và một đầu ra Z (hình 10-7,a).

Bước 3 : Đồ hình trạng thái, bảng trạng thái và tín hiệu ra

Căn cứ vào chức năng của mạch và các tín hiệu vào ra, lập được đồ hình trạng thái cho mô hình Mealy như hình 10-7. Từ đồ hình trạng thái hình 10-7,b, ta lập được bảng trạng thái và tín hiệu ra cho mô hình Mealy như hình 10-7,c.

Bước 4 : Tối thiểu hóa trạng thái

Từ bảng trạng thái và tín hiệu ra hình 10-7,c, tiến hành tối thiểu hóa trạng thái theo phương pháp Caldwell và được bảng trạng thái, tín hiệu ra hình 10-7,e (các bước tối thiểu được biểu diễn trên hình 10-7,d và 10-7,e).

Vậy sau khi đã tối thiểu hóa, số trạng thái trong của mạch là 4.

Đồ hình trạng thái đã tối thiểu được biểu diễn trên hình 10-7,f và là đồ hình cho mô hình Mealy. Đồ hình trạng thái cho mô hình Moore tương ứng nhận được bằng cách sau :

Trên đồ hình chuyển đổi trạng thái 10-7,f trạng thái S_o được chia thành 2 trạng thái S_{oA} và S_{oB} như hình 10-7,g. Lúc đó tín hiệu ra $Z = 1$ khi mạch ở trạng thái S_{oA} và chỉ phụ thuộc vào trạng thái S_{oA} , không phụ thuộc vào X , do vậy 10-7,g là đồ hình trạng thái cho mô hình Moore.

Bước 5 : Mã hóa trạng thái

Cho mô hình Mealy (hình 10-7,f) :

Mạch có 4 trạng thái trong là $S_o, S_{12}, S_{46}, S_{35}$.

- Để mã hóa cho các trạng thái ấy cần 2 biến nhị phân A, B , giả sử ta mã hóa như sau : $S_o(00); S_{12}(01); S_{46}(11); S_{35}(10)$

Cho mô hình Moore (hình 10-7,g) :

Mạch có 5 trạng thái trong, dùng 3 biến nhị phân A, B, C để mã hóa cho các trạng thái đó. Ví dụ chọn cách mã hóa :

$$S_{oA} : 000 \quad S_{12} : 001 \quad S_{oB} : 100 \quad S_{46} : 011 \quad S_{35} : 010$$

Bước 6 : Hệ phương trình của mạch

Xác định các phương trình này từ đồ hình trạng thái.

a) *Hệ phương trình của mạch theo mô hình Mealy :*

Nhìn vào đồ hình trạng thái hình 10-7,f ta có :

$$T_{onA} = S_{12} \cdot \bar{X} + S_{12} \cdot X = S_{12} = \bar{A} \cdot B \Rightarrow J_A = B$$

$$T_{offA} = S_{46} + S_{35} = A \cdot B + A \cdot \bar{B} \Rightarrow K_A = 1$$

$$T_{onB} = S_o = \bar{A} \cdot \bar{B} \Rightarrow J_B = \bar{A}$$

$$T_{offB} = S_{12} \cdot X + S_{46} = \bar{A} \cdot B \cdot X + A \cdot B \Rightarrow K_B = \bar{A} \cdot X + A = X + A$$

Và phương trình tín hiệu ra : $Z = A \cdot \bar{B} \cdot X \cdot C_k$.

b) Hệ phương trình của mạch theo mô hình Moore :

Dựa vào đồ hình trạng thái cho mô hình Moore (hình 10-7,f), có :

- Hệ phương trình hàm kích cho các FF :

$$\begin{aligned}
 T_{onA} = S_{35} \cdot \bar{X} + S_{46} &= \bar{A} B \bar{C} \bar{X} + A B C \Rightarrow J_A = B \bar{C} \bar{X} + B C \\
 &= B(\bar{C} \bar{X} + C) = B(C + \bar{X}) \\
 T_{offA} = S_{oB} &= A \bar{B} \bar{C} \Rightarrow K_A = \bar{B} \bar{C} \\
 T_{onB} = S_{12} \cdot X + S_{12} \bar{X} &= S_{12} = A \bar{B} \bar{C} \Rightarrow J_B = A C \\
 T_{offB} = S_{46} + S_{35} \bar{X} &= \bar{A} B C + \bar{A} B \bar{C} \bar{X} \Rightarrow K_B = \bar{A} C + \bar{A} \bar{C} \bar{X} \\
 &= \bar{A}(C + \bar{C} \bar{X}) = \bar{A}(C + \bar{X}) \\
 T_{onC} = S_{OA} + S_{OB} &= \bar{A} \bar{B} \bar{C} + A \bar{B} \bar{C} \Rightarrow J_C = \bar{A} \bar{B} + A \bar{B} = \bar{B} \\
 T_{offC} = S_{12} X + S_{46} &= \bar{A} \bar{B} C + \bar{A} B C \Rightarrow K_C = \bar{A} \bar{B} X + \bar{A} B \\
 &= \bar{A}(\bar{B} X + B) = \bar{A}(B + X).
 \end{aligned}$$

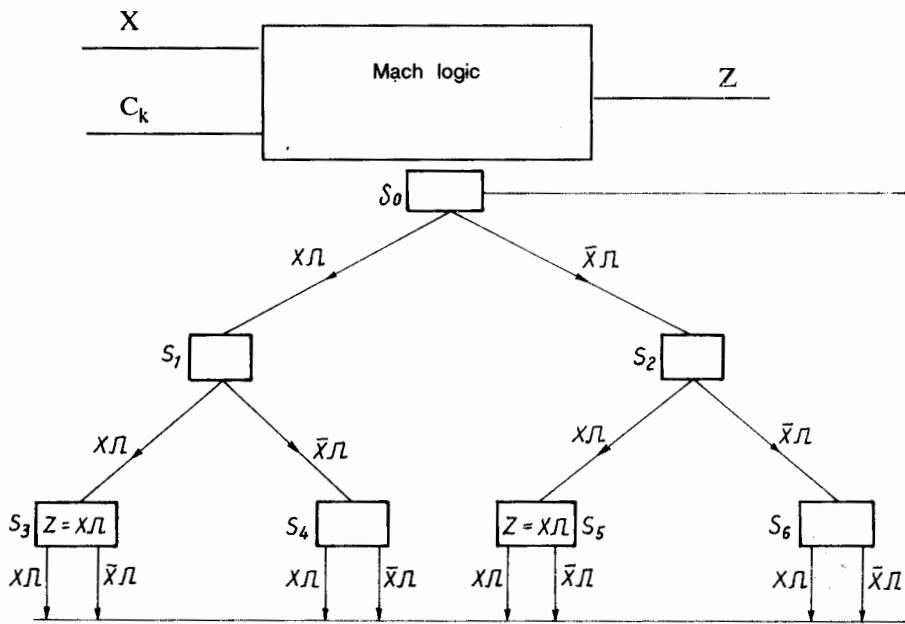
Và phương trình tín hiệu ra Z chỉ phụ thuộc vào trạng thái trong S_{oA} : $Z = \bar{A} \bar{B} \bar{C}$.

Bước 7 : Sơ đồ mạch thực hiện

a) *Sơ đồ mạch thực hiện theo mô hình Mealy :*

Từ hệ phương trình của mạch ở bước 6, ta lập được sơ đồ mạch thực hiện như hình 10-7,h.

b) *Sơ đồ mạch thực hiện theo mô hình Moore*



Hình 10-7. a) Sơ đồ khái; b) Đồ hình trạng thái cho mô hình Mealy.

X	0	1
S ₀	S ₂ Z = 0	S ₁ Z = 0
S ₁	S ₄ Z = 0	S ₃ Z = 0
S ₂	S ₆ Z = 0	S ₅ Z = 0
S ₃	S ₀ Z = 1	S ₀ Z = 1
S ₄	S ₀ Z = 0	S ₀ Z = 1
S ₅	S ₀ Z = 0	S ₀ Z = 1
S ₆	S ₀ Z = 0	S ₀ Z = 0

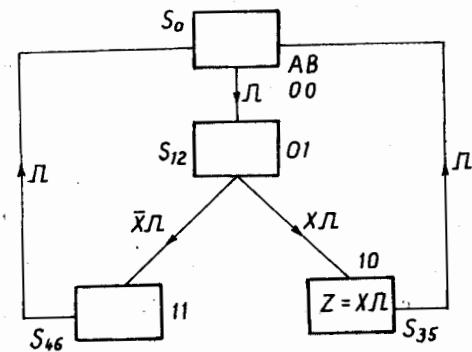
c)

X	0	1
S ₀	S ₂ Z = 0	S ₀ Z = 0
S ₁	S ₄₆ Z = 0	S ₃₅ Z = 0
S ₂	S ₄₆ Z = 0	S ₃₅ Z = 0
S ₃₅	S ₀ Z = 0	S ₀ Z = 1
S ₄₆	S ₀ Z = 0	S ₀ Z = 0

d)

X	0	1
S ₂	S ₁₂ Z = 0	S ₁ Z = 0
S ₁₂	S ₄₆ Z = 0	S ₃₅ Z = 0
S ₄₆	S ₀ Z = 0	S ₀ Z = 1
	S ₀ Z = 0	S ₀ Z = 0

e)



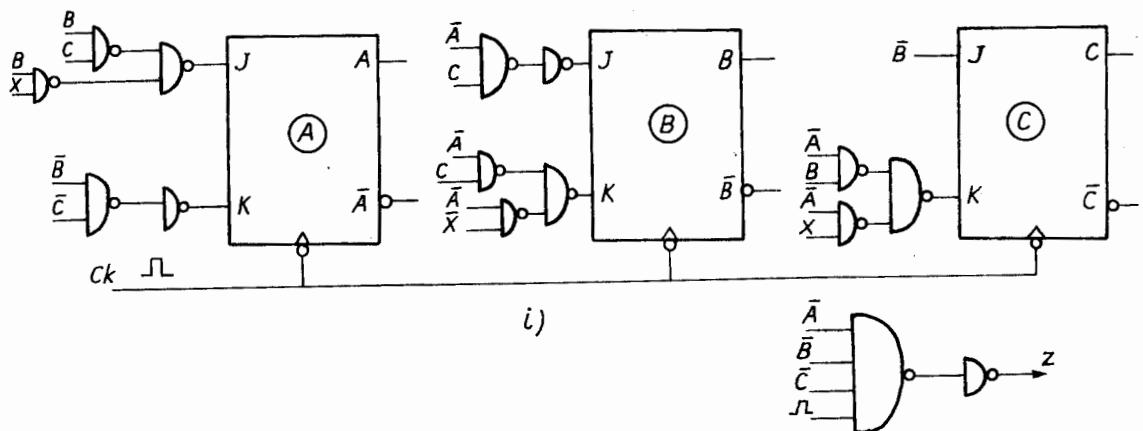
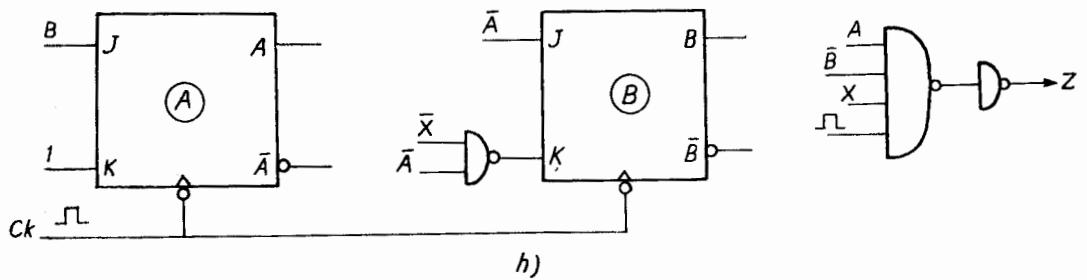
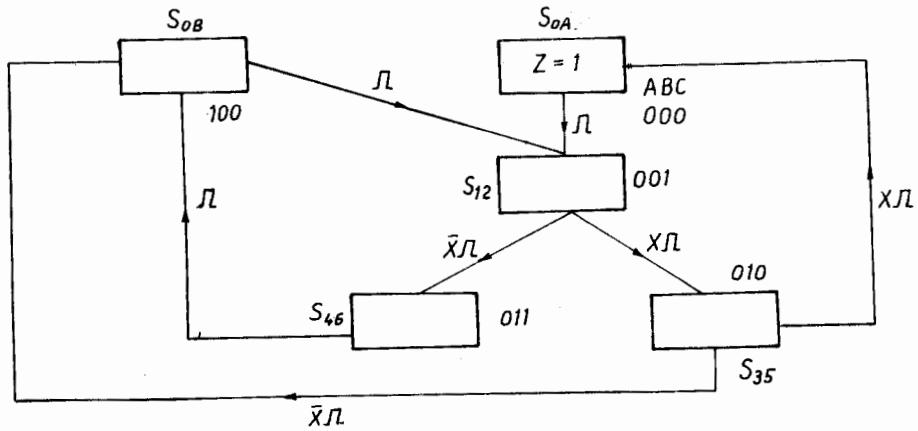
f)

Hình 10-7. c) Bảng trạng thái, bảng ra cho mô hình Mealy ; d) Các bước tối thiểu hoá trạng thái ; e) Các bước tối thiểu hoá trạng thái, f) Đồ hình trạng thái (đã tối thiểu) cho mô hình Mealy.

Kết luận :

Từ 2 ví dụ thiết kế mạch dãy dùng cả mô hình Moore và mô hình Mealy, có thể rút ra nhận xét sau :

- Mạch thực hiện dùng mô hình Moore nói chung có số trạng thái trong nhiều hơn so với mô hình Mealy. Cho nên trong nhiều trường hợp dẫn đến mạch thực hiện phải dùng nhiều phân tử nhớ (FF) hơn, làm cho mạch thực hiện phức tạp hơn.



Hình 10-7. g) Đồ hình trạng thái cho mô hình Moore; h) Sơ đồ mạch cho mô hình Mealy;
i) Sơ đồ mạch cho mô hình Moore.

- Để bù lại nhược điểm trên, phương trình tín hiệu ra theo mô hình Moore không phụ thuộc vào tín hiệu vào của mạch.

Vì vậy tùy từng trường hợp cụ thể, để đáp ứng các yêu cầu kỹ thuật khác nhau mà có thể sử dụng mô hình Moore hay mô hình Mealy.

10_5. TỐI THIỂU HÓA TRẠNG THÁI

Nhiệm vụ đặt ra cho người thiết kế là phải đưa ra mạch hoạt động theo đúng chức năng cho trước, có sơ đồ càng đơn giản càng tốt. Muốn vậy cần phải giảm nhỏ số trạng thái trong của mạch (nếu có thể) đến mức tối thiểu, bởi vì số trạng thái trong của mạch gần như tỷ lệ với số phần tử nhớ (FF) sử dụng trong mạch.

Ở tiết này, chúng ta sẽ nghiên cứu các phương pháp tối thiểu hóa trạng thái trong của một mạch dây đồng bộ.

Việc tối thiểu hóa trạng thái chủ yếu dựa vào khái niệm "trạng thái tương đương" (định nghĩa các trạng thái tương đương đã được giới thiệu ở mục 10_3 trong phần ví dụ). Bài toán tối thiểu hóa trạng thái chính là bài toán đi tìm lớp các trạng thái tương đương lớn nhất. Một lớp các trạng thái tương đương có thể được thay thế bằng một trạng thái đại diện cho chúng. Số lớp trạng thái tương đương chính là số trạng thái tối thiểu của mạch.

Ba phương pháp tối thiểu hóa trạng thái hay dùng là :

1) Phương pháp kiểm tra Caldwell :

Phương pháp này (đã được giới thiệu trong mục 10_3) có hạn chế là khi số trạng thái trong và số tín hiệu vào lớn thì công việc kiểm tra mất nhiều thời gian và bảng chuyển đổi công kẽm. Để khắc phục nhược điểm đó người ta đã đưa ra 2 phương pháp tối thiểu hóa khác thuận tiện hơn.

2) Phương pháp phân hoạch

Phương pháp này được dựa trên việc phân chia các trạng thái theo các lớp có các đặc điểm giống nhau. Sự phân chia thực hiện như sau :

1) Sự phân chia lần thứ nhất (P_1)

Từ bảng chuyển đổi trạng thái và tín hiệu ra ban đầu, phân chia các trạng thái ban đầu thành nhiều lớp, trong mỗi lớp là những trạng thái có tín hiệu ra như nhau khi tín hiệu vào như nhau.

2) Sự phân chia lần thứ 2 (P_2)

Xét các trạng thái trong cùng một lớp của P_1 : Nếu các trạng thái nằm trong cùng một lớp của P_1 có các trạng thái tiếp theo cũng nằm trong cùng một lớp của P_1 thì được xếp chung trong một nhóm. Nếu không thoả mãn thì phải tách chúng ra thành các nhóm khác nhau. Và sẽ được một tập hợp các nhóm mới P_2 .

Cứ tiếp tục lặp lại bước 2 sẽ thu được $P_3, P_4, P_5 \dots P_{i-1}, P_i$. Quá trình phân chia đó sẽ dừng lại khi $P_{i+1} = P_i$. Số nhóm của P_i chính là số trạng thái đã được tối thiểu hóa.

Ví dụ : Bảng chuyển đổi trạng thái và tín hiệu ra của một mạch dãy đồng bộ mô tả ở hình 10-8,a.

- Sự phân chia lần thứ nhất :

$$P_1 = (S_0, S_3, S_5) (S_1, S_4) (S_2, S_6)$$

- Sự phân chia lần thứ 2 :

$$P_2 = (S_3) (S_0, S_5) (S_1, S_4) (S_2, S_6)$$

Ta có $P_3 = P_2 = (S_3)(S_0, S_5) (S_1, S_4) (S_2, S_6)$

Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₀	S ₀ Z = 0	S ₄ Z = 1
S ₁	S ₄ Z = 1	S ₂ Z = 0
S ₂	S ₀ Z = 1	S ₃ Z = 1
S ₃	S ₅ Z = 0	S ₆ Z = 1
S ₄	S ₁ Z = 1	S ₂ Z = 0
S ₅	S ₅ Z = 0	S ₄ Z = 1
S ₆	S ₀ Z = 1	S ₃ Z = 1

a)

Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₀₅	S ₀₅ Z = 0	S ₁₄ Z = 1
S ₁₄	S ₁₄ Z = 1	S ₂₆ Z = 0
S ₂₆	S ₀₅ Z = 1	S ₃ Z = 1
S ₃	S ₀₅ Z = 0	S ₂₆ Z = 1

b)

Hình 10-8. Bảng chuyển đổi trạng thái, bảng ra của mạch.

a. Trước khi tối thiểu; b. Sau khi tối thiểu.

Như vậy các lớp trạng thái tương đương của mạch đã được xác định. Sau khi đã tối thiểu hoá, mạch chỉ có 4 trạng thái, bảng trạng thái biểu diễn ở hình 10-8,b.

3. Phương pháp dùng bảng so sánh (kéo theo)

Lý thuyết chung của phương pháp này như sau :

Giả sử ban đầu mạch có $n + 1$ trạng thái trong : $S_0, S_1, S_2, \dots, S_n$.

Lập bảng so sánh (bảng kéo theo) có các hàng và cột như hình 10-9.

Mỗi ô biểu diễn cặp so sánh giữa 2 trạng thái tương ứng (S_i, S_j) tại hàng và cột tương ứng với ô đó.

Tiến hành tối thiểu hóa như sau :

(1) Trong mỗi ô, nếu cặp trạng thái tương ứng có tín hiệu ra giống nhau với mọi tổ hợp tín hiệu vào (nghĩa là hàng trong bảng ra là như nhau) thì ghi vào đó cặp trạng thái mà nó chuyển biến tới ứng với từng tổ hợp tín hiệu vào.

Nếu tín hiệu ra khác nhau thì đánh dấu X vào đó.

(2) Kiểm tra những ô ghi các cặp trạng thái : ô này bị gạch đi (X) nếu như trong ô đó có một cặp trạng thái đã bị gạch ở ô tương ứng với chúng lúc trước.

Tiến hành như bước (2) cho đến khi nào không gạch được ô vào nữa thì dừng lại.

Những ô không bị gạch sẽ tương ứng với các cặp trạng thái tương đương S_i, S_j tại hàng và cột tương ứng với ô đó.

(3) Kiểm tra các trạng thái tương đương với S_i (theo cột) : trên bảng kéo theo sẽ kiểm tra từng cột từ phải (S_n) đến trái (S_0) để xác định các lớp trạng thái tương đương. Số lớp trạng thái tương đương của S_0 chính là số trạng thái của mạch (đã tối thiểu).

Để minh họa xét ví dụ sau :

Bảng chuyển đổi trạng thái và tín hiệu ra được mô tả ở hình 10-10,a. Nếu nhìn qua bảng thì ta không thể phát hiện được các trạng thái tương đương. Lập bảng kéo theo như hình 10-10,b.

Bước 1 : (hình 10-10,c).

Dánh dấu X vào những ô mà cặp trạng thái tương ứng với ô đó có hàng tương ứng trong bảng tín hiệu ra khác nhau.

Ở những ô không bị gạch, ghi những cặp trạng thái $S'_i - S'_j$ mà 2 trạng thái S_i, S_j sẽ chuyển biến tới ứng với tín hiệu vào $X = 0$ và $X = 1$.

Bước 2 :

Tiến hành kiểm tra các ô trong bảng 10-10,c gạch thêm một số ô mới, được bảng ở hình 10-10,d.

Bước 3 :

Viết các trạng thái tương đương theo từng cột, bắt đầu từ S_0 :

$$S_0 : (S_0)$$

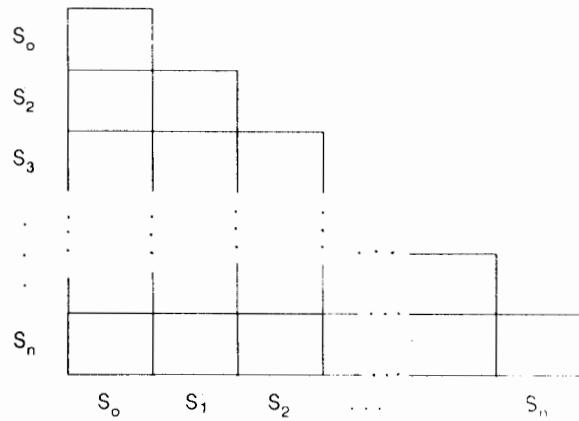
$$S_5 : (S_6) (S_5, S_7)$$

$$S_4 : (S_6) (S_5, S_7) (S_4)$$

$$S_3 : (S_6) (S_5, S_7) (S_4) (S_3)$$

$$S_2 : (S_6) (S_2, S_5, S_7) (S_4) (S_3) (S_1)$$

$$S_1 : (S_6) (S_2, S_5, S_7) (S_4) (S_3) (S_1)$$



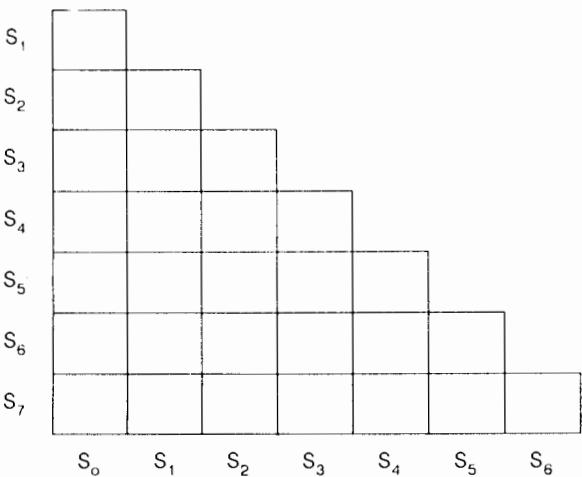
Hình 10-9. Bảng so sánh (bảng kéo theo).

Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₀	S ₀ Z = 0	S ₂ Z = 0
S ₁	S ₃ Z = 1	S ₀ Z = 0
S ₂	S ₅ Z = 0	S ₅ Z = 0
S ₃	S ₄ Z = 1	S ₁ Z = 0
S ₄	S ₆ Z = 1	S ₆ Z = 0
S ₅	S ₂ Z = 0	S ₂ Z = 0
S ₆	S ₁ Z = 1	S ₇ Z = 0
S ₇	S ₇ Z = 0	S ₂ Z = 0

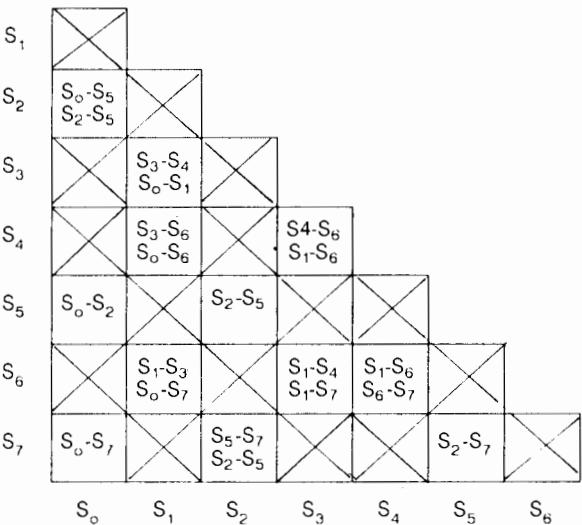
a) Bảng trạng thái ban đầu ví dụ.

Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₀₂₅₇	S ₀₂₅₇ Z = 0	S ₀₂₅₇ Z = 0
S ₁	S ₃ Z = 1	S ₀₂₅₇ Z = 0
S ₃	S ₄ Z = 1	S ₁ Z = 0
S ₄	S ₆ Z = 1	S ₆ Z = 0
S ₅	S ₁ Z = 1	S ₀₂₅₇ Z = 0

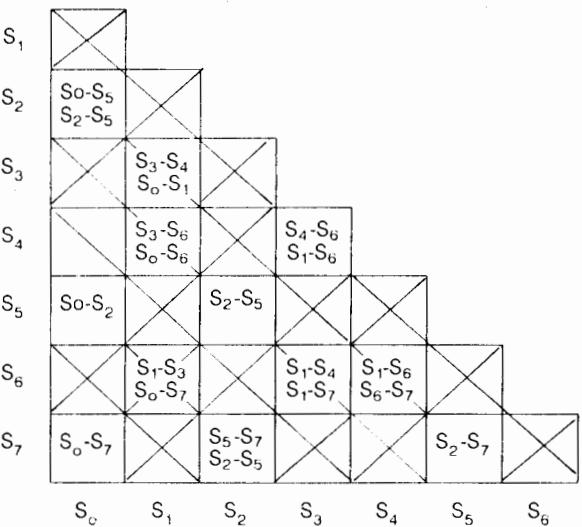
e) Bảng trạng thái
đang ra sau khi đã tối thiểu.



b) Bảng kéo theo.



c) Bảng kéo theo ở bước 1.



d) Bảng kéo theo bước hai.

Hình 10-10. Tối thiểu hóa trạng thái dùng bảng kéo theo.

Số trạng thái tối thiểu rằng số nhóm trạng thái tương đương của cột S_o . Mạch có số trạng thái tối thiểu là 5.

Lập được bảng chuyển đổi trạng thái và tín hiệu ra đã tối thiểu ở hình 10.10.e.

10-5. MÃ HÓA TRẠNG THÁI

Trong các ví dụ trước, các trạng thái trong của mạch được mã hoá một cách tùy ý. Có rất nhiều cách mã hoá khác nhau và mỗi cách cho một sơ đồ mạch thực hiện khác nhau. Cần phải lựa chọn cách mã hoá trạng thái sao cho đồ mạch thực hiện là đơn giản nhất. Sơ đồ mạch đơn giản có lợi về nhiều phương diện : trước hết là việc chế tạo và lắp ráp sẽ đơn giản hơn, tăng độ tin cậy của mạch, số vi mạch cần thiết sẽ ít hơn ...

Hiện nay chưa có phương pháp nào chỉ ra cách mã hoá tối ưu nhất để sơ đồ thực hiện đơn giản nhất.

Áp dụng 2 quy tắc mã hoá trình bày sau, sẽ nhận được sơ đồ mạch thực hiện tương đối đơn giản, tuy nhiên trong nhiều trường hợp mạch thực hiện chưa phải là đơn giản nhất.

Quy tắc 1

Nếu một trạng thái hiện tại S_i có thể chuyển biến đến nhiều trạng thái $S_{i1}', S_{i2}', \dots, S_{in}'$ thì các trạng thái S_{ij}' đó phải được mã hoá bằng các từ mã kế cận (nghĩa là chỉ khác nhau ở một biến).

Quy tắc 2

Nếu nhiều trạng thái $S_{j1}, S_{j2}, \dots, S_{jn}$ cùng chuyển biến đến một trạng thái tiếp theo S_j' thì các trạng thái $S_{j1}, S_{j2}, \dots, S_{jn}$ đó phải được mã hoá bằng các từ mã hoá kế cận nhau.

Để thấy được việc áp dụng 2 quy tắc mã hoá đó có lợi như thế nào, xét cho một ví dụ cụ thể sau :

Ví dụ

Thiết kế 1 mạch dãy đồng bộ phát hiện mã NBCD ở đầu vào là sai. Các mã NBCD được đưa vào liên tiếp từng bit một ở đầu vào X : bit đầu tiên là bit có trọng số lớn nhất. Các tín hiệu vào được đồng bộ với xung nhịp C_k . Và cứ đến bit thứ tư, nếu mã đó không đúng là mã NBCD thì mạch sẽ đưa ra tín hiệu ra $Z = 1$.

Bước 1 : Xác định bài toán

Ta phải thiết kế một mạch dãy đồng bộ kiểm tra dãy tín hiệu ở đầu vào X được đưa vào dưới dạng mã NBCD. Nếu mã đó sai thì mạch phải đưa ra tín hiệu ra $Z = 1$ ở đầu ra.

10 tổ hợp mã NBCD là : 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001

Các trường hợp sai là : 1010, 1011, 1100, 1101, 1110, 1111 khi đó $Z = 1$.

Bước 2 : Xác định tín hiệu vào ra

Biểu diễn mạch như một " hộp đen " có 2 đầu vào là :

- Đầu đưa tín hiệu vào X ở dạng mã NBCD (có độ dài = 4)

- Đầu vào xung nhịp C_k và có 1 đầu ra Z đưa tín hiệu ra $Z = 1$ để báo sai. Sơ đồ khối của mạch như hình 10-11.a.

Bước 3 : Đồ hình trạng thái, bảng trạng thái và tín hiệu ra

Căn cứ vào yêu cầu của bài toán và 2 bước trên, có thể lập được đồ hình chuyển đổi trạng thái như hình 10-11,b. Và từ đồ hình này lại lập được bảng chuyển đổi trạng thái, tín hiệu ra (hình 10-11,c).

Bước 4 : Tối thiểu hóa trạng thái

Ta nhận thấy số trạng thái trên đã là tối thiểu.

Bước 5 : Mã hóa trạng thái

Ở bước này sẽ đưa ra 3 cách mã hóa bất kỳ và một cách mã hóa theo 2 quy tắc đã nêu trên.

* Ba cách mã hóa bất kỳ được mô tả ở bảng (hình 10-11,d)

Mã hóa theo 2 quy tắc :

Quy tắc 1 :

$$(S_3, S_6)$$

$$(S_1, S_4)$$

$$(S_5, S_7)$$

Quy tắc 2 :

$$(S_2, S_7)$$

$$(S_5, S_7)$$

$$(S_3, S_6)$$

Kết hợp cả 2 qui tắc này phải mã hóa (S_3, S_6) (S_2, S_7) (S_5, S_7) (S_1, S_4) bởi các mã hóa kế cận. Giả sử chọn cách mã hóa được mô tả bởi bảng Kranangh trên hình 10-11,e.

Bước 6 : Xác định hệ phương trình của mạch

* Ba cách mã hóa bất kỳ :

Mã hóa cách 1 :

$$J_A = C X + \bar{B} \bar{C} \bar{X}$$

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

$$J_B = \bar{C} X + A \bar{C} + \bar{A} C \bar{X}$$

$$K_B = C \bar{X} + \bar{A} C + A B \bar{C}$$

$$J_C = \bar{A} \bar{B} X$$

$$K_C = \bar{B} \bar{X} + A \bar{B}$$

$$Z = A \cdot \bar{B} \cdot C \cdot C_k$$

Mã hóa cách 2 :

$$J_A = A + B \bar{C} + \bar{B} C + \bar{B} \bar{X}$$

$$K_A = \bar{B} + \bar{C} + X$$

$$J_B = \bar{C}X + A\bar{C} + \bar{A}CX$$

$$K_B = C\bar{X} + A\bar{C} + \bar{A}B\bar{C}$$

$$J_C = \bar{A}\bar{B}X$$

$$K_C = \bar{A}B + B\bar{X}$$

$$Z = \bar{A}, B, C, C_k$$

Mã hóa cách 3 :

$$J_A = CX + \bar{B}\bar{C}\bar{X}$$

$$K_A = B$$

$$J_B = A + C\bar{X}$$

$$K_B = AC + \bar{A}\bar{C}$$

$$J_C = \bar{A}\bar{B}X$$

$$K_C = AB + B\bar{X}$$

$$Z = A, B, C, C_k$$

* Theo phương pháp 2 quy tắc

Bảng trạng thái, và giá trị các đầu vào kích J, K được biểu diễn trên hình 10-11,f. Phương trình của các hàm kích cho các FF được tối thiểu hóa theo bảng Karnaugh hình 10-11,g kết quả :

$$J_C = \bar{B}$$

$$K_C = \bar{A} + \bar{B}\bar{X}$$

$$J_B = 1$$

$$K_B = 1$$

$$J_A = \bar{B} + C + X$$

$$K_A = \bar{B}C + B\bar{X} + B\bar{C}$$

Phương trình tín hiệu ra :

$$Z = A, \bar{B}, \bar{C}, C_k$$

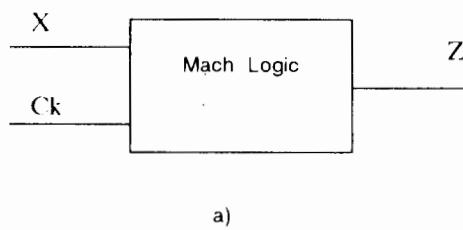
Bước 7 : Sơ đồ mạch thực hiện

* Ba phương pháp mã hóa tùy ý :

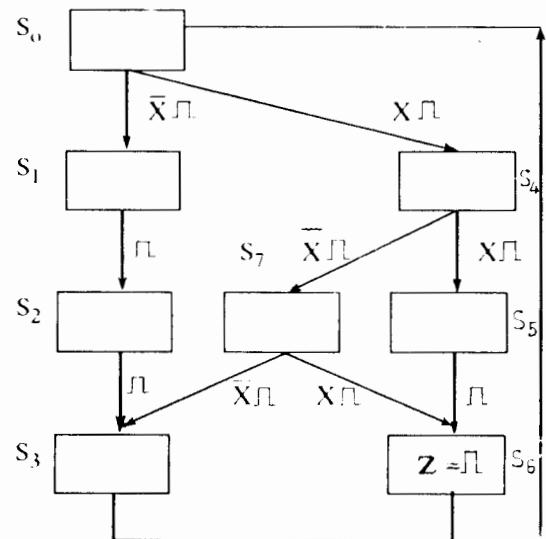
Cả 3 cách mã hóa này đều đưa đến mạch thực hiện khá phức tạp. Số lượng cổng cần thiết để thực hiện mạch được thống kê trong bảng ở hình 10-11,h, trong đó cách mã hóa 3 dùng ít mạch hơn cả.

* Phương pháp mã hóa theo 2 quy tắc :

Mạch thực hiện được mô tả ở hình 10-11,i.



a]



b)

Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₀	S ₁	S ₄
S ₁	S ₂	S ₂
S ₂	S ₃	S ₃
S ₃	S ₀	S ₀
S ₄	S ₁	S ₅
S ₅	S ₆	S ₆
S ₆	S ₀	S ₀
S ₇	S ₃	S ₆

c)

Trạng thái	Cách 1			Cách 2			Cách 3		
	C	B	A	C	B	A	C	B	A
S ₀	0	0	0	0	0	0	0	0	0
S ₁	0	0	1	0	0	1	0	0	1
S ₂	0	1	1	0	1	0	0	1	1
S ₃	0	1	0	0	1	1	0	1	0
S ₄	1	1	0	1	0	0	1	0	0
S ₅	1	1	1	1	0	1	1	0	1
S ₆	1	0	1	1	1	0	1	1	1
S ₇	1	0	0	1	1	1	1	1	0

d)

		BA	00	01	11	10	
		C	0	S_3	S_6	S_2	S_7
			1	S_1	S_4	S_0	S_5
0							
1							

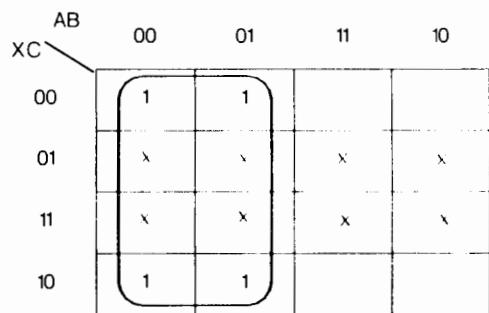
e

Hình 10-11. Thiết kế mạch phát hiện mã NBCD sai.

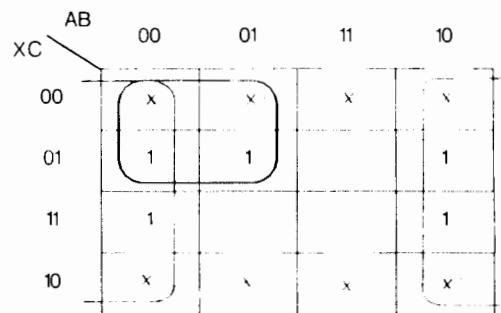
- a) Sơ đồ khối ; b) Dồ hình trang thái ; c) Bảng chuyển đổi trạng thái, bảng ra ;
d) 3 cách mã hóa trạng thái bất kỳ ; e) Bảng giá trị đầu vào kích của JK - FF.

Trạng thái hiện tại	Trạng thái tiếp theo		Các đầu vào của FF												
	X=0	X=1	X = 0		X = 1		X = 0		X = 1		X = 0		X = 1		
CBA	CBA	CBA	J _C	K _C	J _C	K _C	J _C	K _C	J _B	K _B	J _A	K _A	J _A	K _A	
S ₀	111	100	101	x	0	x	0	x	1	x	1	x	1	x	0
S ₁	100	011	011	x	1	x	1	1	x	1	x	1	x	1	x
S ₂	011	000	000	0	x	0	x	x	1	x	1	x	1	x	1
S ₃	000	111	111	1	x	1	x	1	x	1	x	1	x	1	x
S ₄	101	010	110	x	1	x	0	1	x	1	x	1	x	1	x
S ₅	110	001	001	x	1	x	1	x	1	x	1	x	1	x	1
S ₆	001	111	111	1	x	1	x	1	x	1	x	0	x	0	x
S ₇	010	000	001	0	x	0	x	x	1	x	1	x	1	x	1

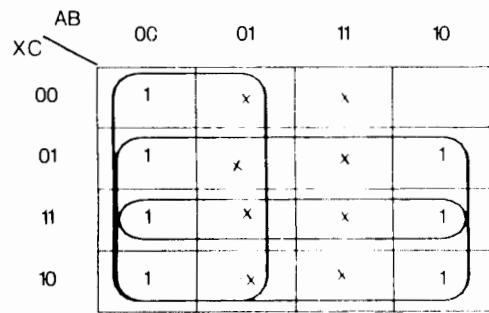
f)



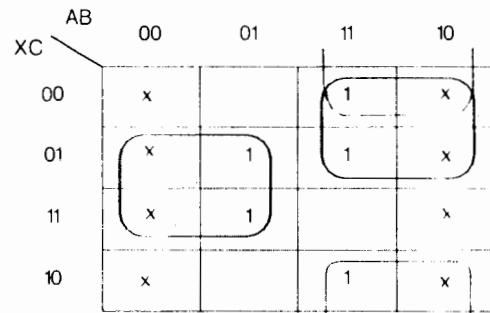
$$J_C = \overline{B}$$



$$K_C = \overline{A} + \overline{B} \cdot \overline{X}$$



$$J_A = \overline{B} + C + X$$



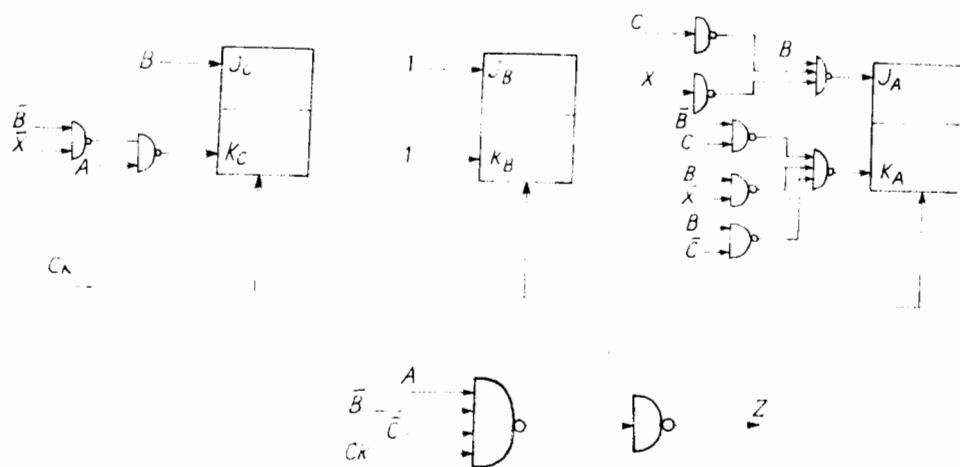
$$K_A = \overline{B} \cdot C + B \cdot \overline{X} + B \cdot \overline{C}$$

Hình 10-11. f) Bảng chuyển đổi trạng và các đầu vào kích cho FF;

g) Bảng Karnaugh của các hàm kích cho FF;

Gates NAND	Mã hóa cách 1	Mã hóa cách 2	Mã hóa cách 3
	22	18	16
Chips			
Dual 4 input	1	1	1
Triple 3 input	2	1	1
Quad 2 input	4	4	4
Total chips	7	6	6
Gate input	49	39	33

h)



Hình 10-11. h) So sánh sơ đồ thực hiện mạch với 3 cách mã hóa khác nhau ; i) Sơ đồ mạch dùng cách mã hóa theo 2 qui tắc.

10_6. MẠCH KIỂM TRA DÂY TÍN HIỆU VÀO

Trong các tiết trước chúng ta đã nghiên cứu các chi tiết cơ bản của công việc thiết kế một mạch dây đồng bộ. Nay giờ chúng ta ứng dụng những lý thuyết đó để thực hiện ví dụ thiết kế mạch dây đồng bộ sau :

Hãy thiết kế một mạch dây đồng bộ có dữ liệu đưa vào đầu vào X là một dây nhị phân được đồng bộ với xung nhịp. Nếu dây tín hiệu vào X là 111 thì mạch phải cho một tín hiệu ra $Z = 1$ trên đầu ra để báo hiệu là đã nhận dây tín hiệu như vậy. Dây tín hiệu vào có thể gối lên nhau.

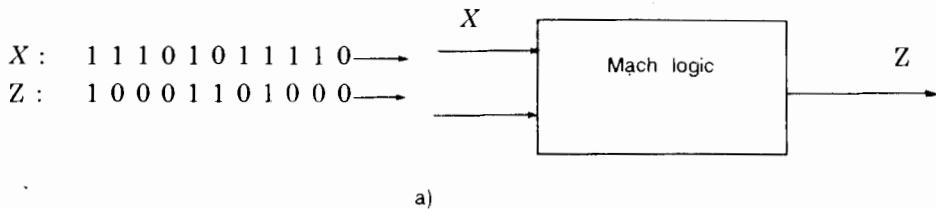
Ví dụ : $X : 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0$
 $Z : 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0$

Bước 1 : Xác định bài toán

Yêu cầu đặt ra là thiết kế một mạch dãy đồng bộ kiểm tra dãy tín hiệu vào có độ dài là 3. Nếu dãy tín hiệu đó là 111 cho ra 1 tín hiệu ra $Z = 1$ để báo hiệu.

Bước 2 : Tín hiệu vào, ra

Mạch gồm 2 đầu vào là : X đưa dữ liệu nhị phân và C_k đưa xung nhịp để đồng bộ ; và một đầu ra Z (hình 10-12,a).



a)

Bước 3 : Đồ hình chuyển đổi trạng thái

Căn cứ theo yêu cầu của đề thiết kế, có thể lập được đồ hình chuyển đổi trạng thái hình 10-12.b.

S_0 là trạng thái ban đầu của mạch. Các đường chuyển đổi trạng thái được ghi rõ trên đồ hình. Trạng thái S_3 là trạng thái không sử dụng, mạch sẽ thoát ra khỏi trạng thái S_3 khi nhận tín hiệu vào $X = 1$ và nó sẽ chuyển đến trạng thái S_1 .

Bước 4 : Tối thiểu hóa trạng thái

Có thể dễ dàng nhận thấy đồ hình chuyển đổi trạng thái đã là đơn giản nhất và không thể tối thiểu được nữa.

Bước 5 : Mã hoá trạng thái

Áp dụng 2 quy tắc mã hoá, mã hoá 4 trạng thái trong bởi 2 biến nhị phân A, B như sau :

$$S_0 (00); S_1 (01); S_2 (11); S_3 (10);$$

Bước 6 : Hệ phương trình của mạch

Với cách mã hoá trên, áp dụng cách thiết kế từ đồ hình trạng thái, nhận được hệ phương trình của mạch như sau :

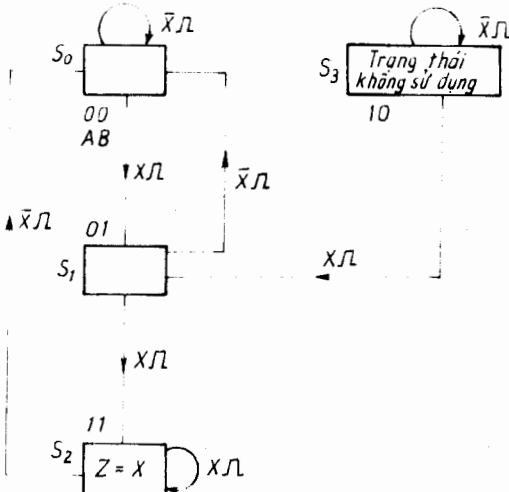
* Phương trình hàm kích cho các FF :

$$T_{onA} = S_1 \cdot X = \bar{A} B X \Rightarrow J_A = B X$$

$$T_{offA} = S_3 \cdot X + S_2 \cdot \bar{X} = A \bar{B} X + A B \bar{X} \Rightarrow K_A = \bar{B} X + B \bar{X}$$

$$T_{onB} = S_0 \cdot X + S_2 \cdot \bar{X} = \bar{A} \bar{B} X + A B X = B X \Rightarrow J_B = X$$

$$T_{offB} = S_2 \cdot \bar{X} + S_1 \cdot \bar{X} = A B \bar{X} + \bar{A} B \bar{X} = B \bar{X} \Rightarrow K_B = \bar{X}$$



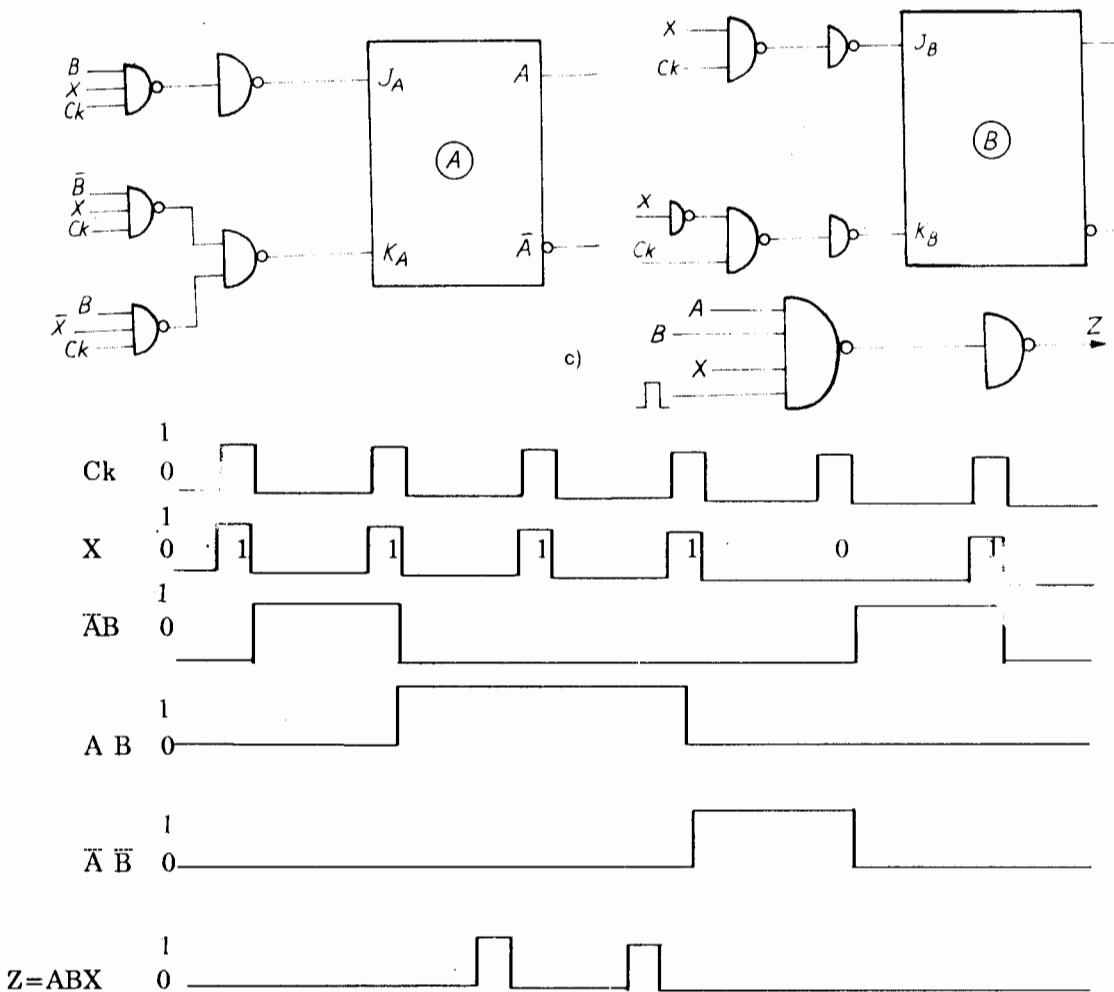
Hình 10-12. a) Sơ đồ khối ; b) Đồ hình chuyển đổi trạng thái.

* Phương trình tín hiệu ra $Z = A B X C_k$

Bước 7 : Sơ đồ mạch thực hiện

Từ hệ phương trình của mạch lập được sơ đồ mạch thực hiện như hình 10-12,c. Cần chú ý rằng khi viết hệ phương trình các đầu vào kích của mạch để đơn giản ta đã bỏ qua C_k trong các biểu thức. Trong sơ đồ đã thực hiện phép AND của C_k với tín hiệu vào.

Giản đồ thời gian của mạch được mô tả ở hình 10-12,d.

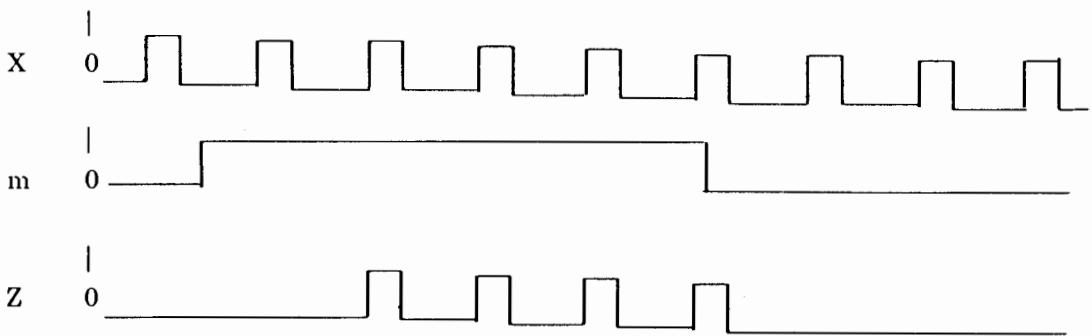


Hình 10-12. c) Sơ đồ của mạch ; d) Giản đồ dạng sóng của mạch.

BÀI TẬP

10.1. Tín hiệu xung nhịp X đưa tới một cổng được điều khiển đóng, mở bởi một tín hiệu m . Tín hiệu lựa chọn đóng hay mở cổng được xắp xếp sao cho mạch chỉ đưa ra những xung nhịp đầy đủ. Dạng sóng của mạch được biểu diễn như hình 10-13.

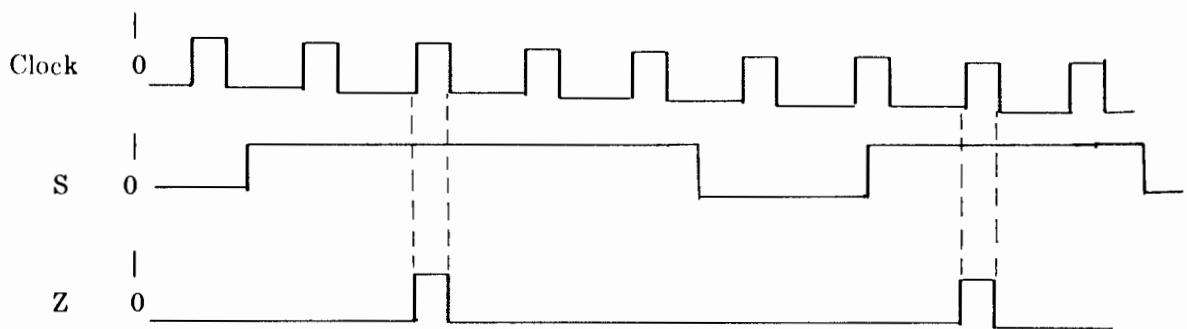
Hãy thiết kế một mạch dây đồng bộ thực hiện mô tả trên.



Hình 10-13. Cho bài tập 10.1

10.2. Một mạch được thiết kế cho qua xung nhịp đơn Z được điều khiển bởi công tắc S . Công tắc S được làm sao cho khoảng thời gian bật là tùy ý và dài hơn so với khoảng thời gian của chu kỳ xung nhịp.

Dạng sóng được biểu diễn như trên hình 10-14.



Hình 10-14. Cho bài tập 10.2

10.3. Một mạch dây có hai đầu vào tín hiệu X , xung nhịp và một đầu ra Z . Dữ liệu đầu vào X có dạng chuỗi và có độ dài bằng 4. Nếu dữ liệu vào có dạng một trong 3 chuỗi 1010, 0110, 0010 thì mạch sẽ cho tín hiệu ra trên đầu ra $Z = 1$. Hãy thiết kế mạch thực hiện chức năng trên dựa vào $JK-FF$ và cỗng NOR.

10.4. Một mạch dây Logic được sử dụng để kiểm tra tính chẵn lẻ của một dãy dữ liệu nhị phân liên tục được đưa đến đầu vào. Nếu số chữ số 1 nhận được là lẻ thì mạch sẽ đưa ra tín hiệu ra là $Z = 1$. Nếu hai chữ số 0 được đưa liên tiếp ở đầu vào thì mạch sẽ quay trở lại trạng thái ban đầu và lại bắt đầu kiểm tra dãy dữ liệu mới.

Thiết kế mạch thực hiện chức năng đó.

10.5. Thiết kế mạch dây đồng bộ thực hiện so sánh hai số nhị phân 4 bits (A và B) với bit đầu tiên là bit có trọng số lớn nhất. 3 đầu ra của mạch là :

$$Z_1 = 1 \text{ nếu } A > B$$

$$Z_2 = 1 \text{ nếu } A = B$$

$$Z_3 = 1 \text{ nếu } A < B$$

10.6. Một mạch dây đồng bộ có bảng chuyển đổi trạng thái, bảng ra như trên hình 10-15.

Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₀	S ₁ Z = 0	S ₄ Z = 0
S ₁	S ₀ Z = 0	S ₅ Z = 0
S ₂	S ₃ Z = 0	S ₅ Z = 1
S ₃	S ₂ Z = 0	S ₄ Z = 1
S ₄	S ₁ Z = 0	S ₂ Z = 0
S ₅	S ₀ Z = 0	S ₃ Z = 0
S ₆	S ₁ Z = 0	S ₀ Z = 0

Hình 10-15. Cho bài tập 10-6.

Hãy xác định chuỗi tín hiệu ra của mạch cho tất cả các dây tín hiệu vào có thể có độ dài 1, 2, 3, ... Sau đó tìm phân hoạch trạng thái .

10.7. Tối thiểu hóa trạng thái của hai mạch dây đồng bộ mà bảng trạng thái của chúng được cho trên hình 10-16, dùng :

- a. Qui tắc kết hợp Caldwell ;
- b. Phân hoạch ;
- c. Bảng kéo theo .

Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₀	S ₁ Z = 0	S ₂ Z = 0
S ₁	S ₃ Z = 0	S ₄ Z = 0
S ₂	S ₅ Z = 0	S ₆ Z = 0
S ₃	S ₀ Z = 1	S ₀ Z = 0
S ₄	S ₀ Z = 0	S ₀ Z = 0
S ₅	S ₀ Z = 0	S ₀ Z = 0
S ₆	S ₀ Z = 0	S ₀ Z = 1

Hình 10-16. Cho bài tập 10-7.

Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₀	S ₁ Z = 0	S ₈ Z = 0
S ₁	S ₂ Z = 0	S ₅ Z = 0
S ₂	S ₃ Z = 0	S ₄ Z = 0
S ₃	S ₀ Z = 0	S ₀ Z = 0
S ₄	S ₀ Z = 0	S ₀ Z = 1
S ₅	S ₆ Z = 0	S ₇ Z = 0
S ₆	S ₀ Z = 0	S ₀ Z = 0
S ₇	S ₀ Z = 0	S ₁ Z = 1

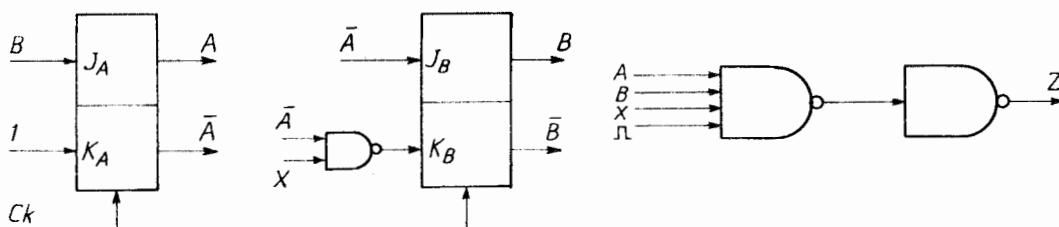
Trạng thái hiện tại	Trạng thái tiếp theo	
	X = 0	X = 1
S ₈	S ₉ Z = 0	S ₁₂ Z = 0
S ₉	S ₁₀ Z = 0	S ₁₁ Z = 0
S ₁₀	S ₀ Z = 0	S ₀ Z = 0
S ₁₁	S ₀ Z = 0	S ₀ Z = 0
S ₁₂	S ₁₃ Z = 0	S ₁₄ Z = 0
S ₁₃	S ₀ Z = 0	S ₇ Z = 0
S ₁₄	S ₀ Z = 0	S ₀ Z = 1

Hình 10-6. Cho bài tập 10.7.

10.8. Cho một mạch dây biểu diễn trên hình 10-17.

Hãy xác định :

- Bảng chuyển đổi trạng thái ;
- Đồ hình trạng thái ;
- Chức năng của mạch .



Hình 10-17. Cho bài tập 10.8.

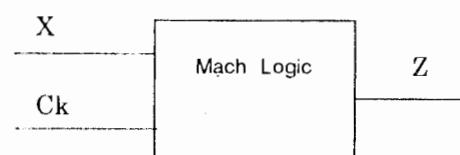
10.9. Dữ liệu nhị phân X được đưa liên tiếp ở đầu vào và được đồng bộ với xung nhịp. Thiết kế mạch mà tín hiệu ra Z sẽ bằng 1 khi dãy số 1101 được thu nhận ở đầu vào .

10.10. Thiết kế bộ đếm đồng bộ được điều khiển bởi hai tín hiệu A và B. Nếu :

A = 0, B = 0 : Bộ đếm không hoạt động ;

A = 0, B = 1 : Bộ đếm có $K_d = 4$;

A = 1, B = 0 : Bộ đếm có $K_d = 8$.



Hình 10-18. Cho bài tập 10.9.

CHƯƠNG 11

MẠCH DÃY KHÔNG ĐỒNG BỘ

Chương 10 đã nghiên cứu các mạch dãy đồng bộ, hoạt động của chúng được điều khiển bởi các xung nhịp. Tuy nhiên trong thực tế có những mạch lại được điều khiển bởi các sự kiện không tuân theo một qui luật nào cả. Chẳng hạn một hệ thống báo động cháy sẽ chỉ hoạt động khi có hiện tượng cháy. Những mạch dãy được điều khiển bởi những sự kiện ngẫu nhiên như vậy gọi là các mạch không đồng bộ.

Mạch dãy không đồng bộ có thể thiết kế :

- chỉ dùng những mạch *NAND*;
- dùng *RS - FF* không đồng bộ và các mạch *NAND*.

Việc thiết kế mạch dãy không đồng bộ dùng các *FF* loại không đồng bộ khác cũng hoàn toàn tương tự.

11_1. CÁC BƯỚC THIẾT KẾ MẠCH DÃY KHÔNG ĐỒNG BỘ

Các bước thiết kế một mạch dãy không đồng bộ được mô tả trong lưu đồ hình 11-1.

Nhận xét rằng : các bước thiết kế mạch dãy không đồng bộ cũng giống như thiết kế mạch đồng bộ (xem lại hình 10- 1). Chỉ khác ở bước 5, bước 6 :

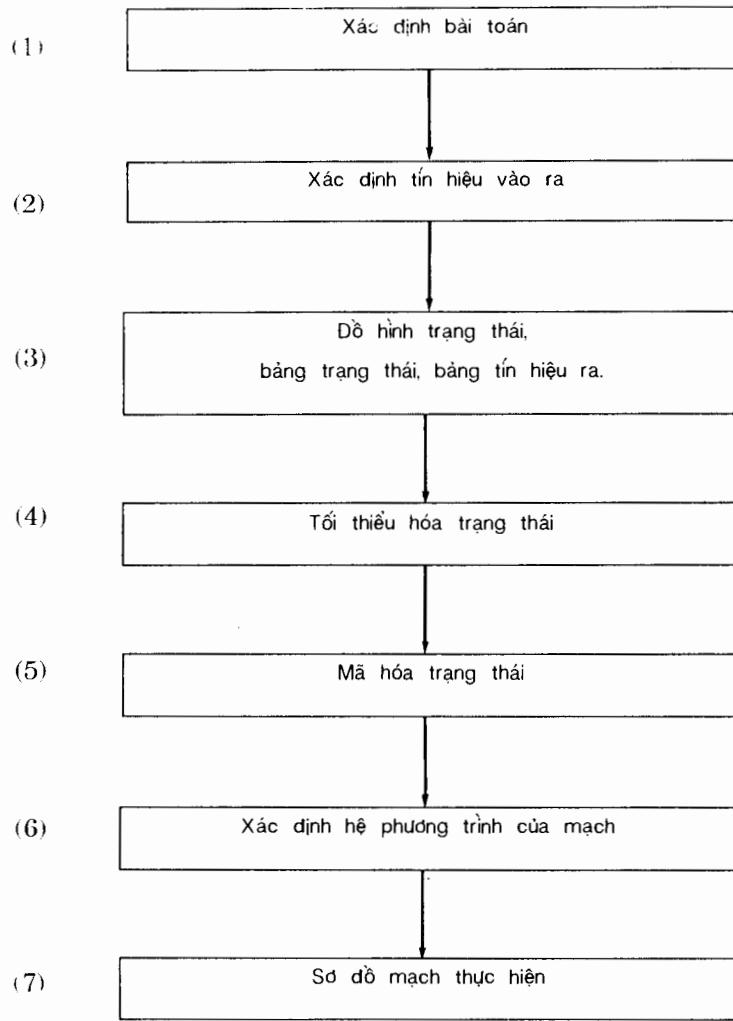
Bước 5 : Mã hoá trạng thái

Mạch không đồng bộ hoạt động không có sự tác động của xung nhịp cho nên trong mạch thường có các hiện tượng hoặc là chu kỳ hoặc là chạy đua làm cho hoạt động của mạch sai đi (sẽ xét cụ thể vấn đề này ở mục 11_3), vì vậy khi mã hoá trạng thái phải tránh hiện tượng này.

Bước 6 : Xác định hệ phương trình của mạch

Có hai cách để xác định hệ phương trình của mạch : Dựa vào bảng trạng thái và tín hiệu ra và dựa trực tiếp vào đồ hình. Cả hai cách đều có hai loại phương trình :

- + Phương trình của mạch chỉ dùng *NAND*.
- + Phương trình của mạch dùng *RS - FF* không đồng bộ và các mạch *NAND*



Hình 11-1. Các bước thiết kế một mạch dãy không đồng bộ.

Dưới đây là nội dung của từng phương pháp.

Cách 1: Dựa vào bảng trạng thái và tín hiệu ra

a. Chỉ dùng các mạch NAND :

Ký hiệu : A, B, \dots, N là các biến nhị phân dùng để mã hoá các trạng thái trong của mạch

X_1, X_2, \dots, X_m là các tín hiệu vào đã được mã hoá nhị phân.

Z_1, Z_2, \dots, Z_n là các tín hiệu ra đã được mã hoá nhị phân.

Dựa vào bảng trạng thái, bảng ra xác định được hệ phương trình :

$$A' = f_A(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

$$B' = f_B(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

$$N' = f_N(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_1 = g_1(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_2 = g_2(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

$$\dots \dots \dots \\ Z_n = g_n(A, B, \dots, N, X_1, X_2, \dots, X_m) .$$

Tối thiểu hoá hệ hàm này và viết phương trình ở dạng chỉ dùng *NAND*.

b. *Mạch dùng RS - FF và các mạch NAND* :

Trong bảng trạng thái, căn cứ vào sự thay đổi trạng thái của từng *FF* :

$A \Rightarrow A'$, $B \Rightarrow B'$, ..., $N \Rightarrow N'$, xác định được giá trị tương ứng của đầu vào kích R, S cho từng *FF*, từ đó viết được hệ phương trình :

$$R_A = \Phi_1(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$S_A = \Phi_2(A, \dots, N, X_1, X_2, \dots, X_m)$$

Tối thiểu hoá các hàm và viết phương trình ở dạng dùng *NAND*.

Tương tự với B, C và N ta cũng làm tương tự.

Ta xác định tín hiệu ra

$$Z = \Phi(A, \dots, N, X_1, X_2, \dots, X_m)$$

Tối thiểu hoá và viết phương trình ở dạng chỉ dùng *NAND*.

Cách 2: *Dựa trực tiếp vào đồ hình*

Ta có phương trình đầu vào kích (R, S) của *FF - A* như sau :

$$S_A = \text{tập hợp bật của } A + [(1)]$$

$$R_A = \text{tập hợp tắt của } A + [(0)] \text{ (xem mục 11-3)}$$

Điều này cũng hoàn toàn tương tự đối với *FF - B, FF - C ...*

a. *Nếu chỉ dùng mạch NAND*

Ta có phương trình của *RS - FF* như sau :

$$Q' = S + \bar{R} Q \quad \Rightarrow \quad A' = S_A + \bar{R}_A A$$

Sau đó ta phải tối thiểu hoá phương trình và viết dưới dạng chỉ dùng *NAND*. Đối với B, C, \dots, N ta cũng làm như vậy.

b. *Nếu dùng RS - FF không đồng bộ và các mạch NAND*

Ta cần xác định S_A, R_A theo các biến đầu vào và các biến nhị phân dùng để mã hoá :

$$R_A = \Phi_{1A}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$S_A = \Phi_{2A}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$R_B = \Phi_{2B}(A, \dots, N, X_1, X_2, \dots, X_m)$$

.

$$R_N = \Phi_{1N}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$S_N = \Phi_{2N}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_1 = \Psi_1(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_2 = \Psi_2(A, \dots, N, X_1, X_2, \dots, X_m)$$

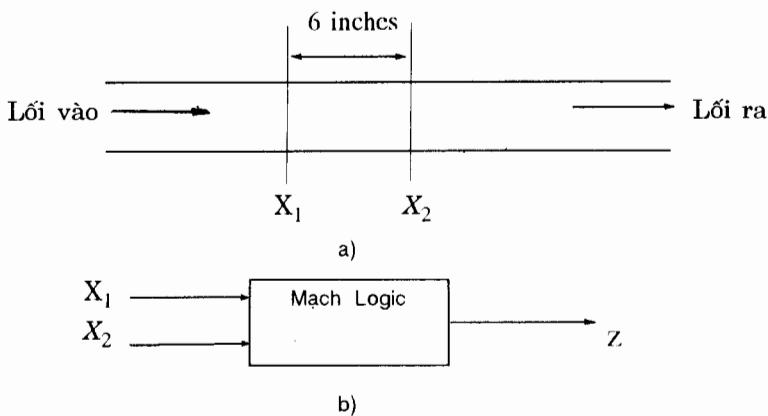
.

$$Z_n = \Psi_n(A, \dots, N, X_1, X_2, \dots, X_m)$$

- Tối thiểu hoá hệ phương trình : viết các phương trình ở dạng dùng mạch *NAND*. Để làm rõ các bước thiết kế mạch dây không đồng bộ ta hãy xét ví dụ cụ thể sau :

11-3. MỘT VÍ DỤ VỀ THIẾT KẾ MẠCH DÂY KHÔNG ĐỒNG BỘ

Một mạch dây không đồng bộ được thiết kế để đếm số người vào thăm một viện bảo tàng. Mạch gồm hai chùm sáng X_1, X_2 được bố trí cách nhau 6 inch như hình 11-2,a. Mạch được thiết kế sao cho mỗi lần đếm được một người. Cụ thể :

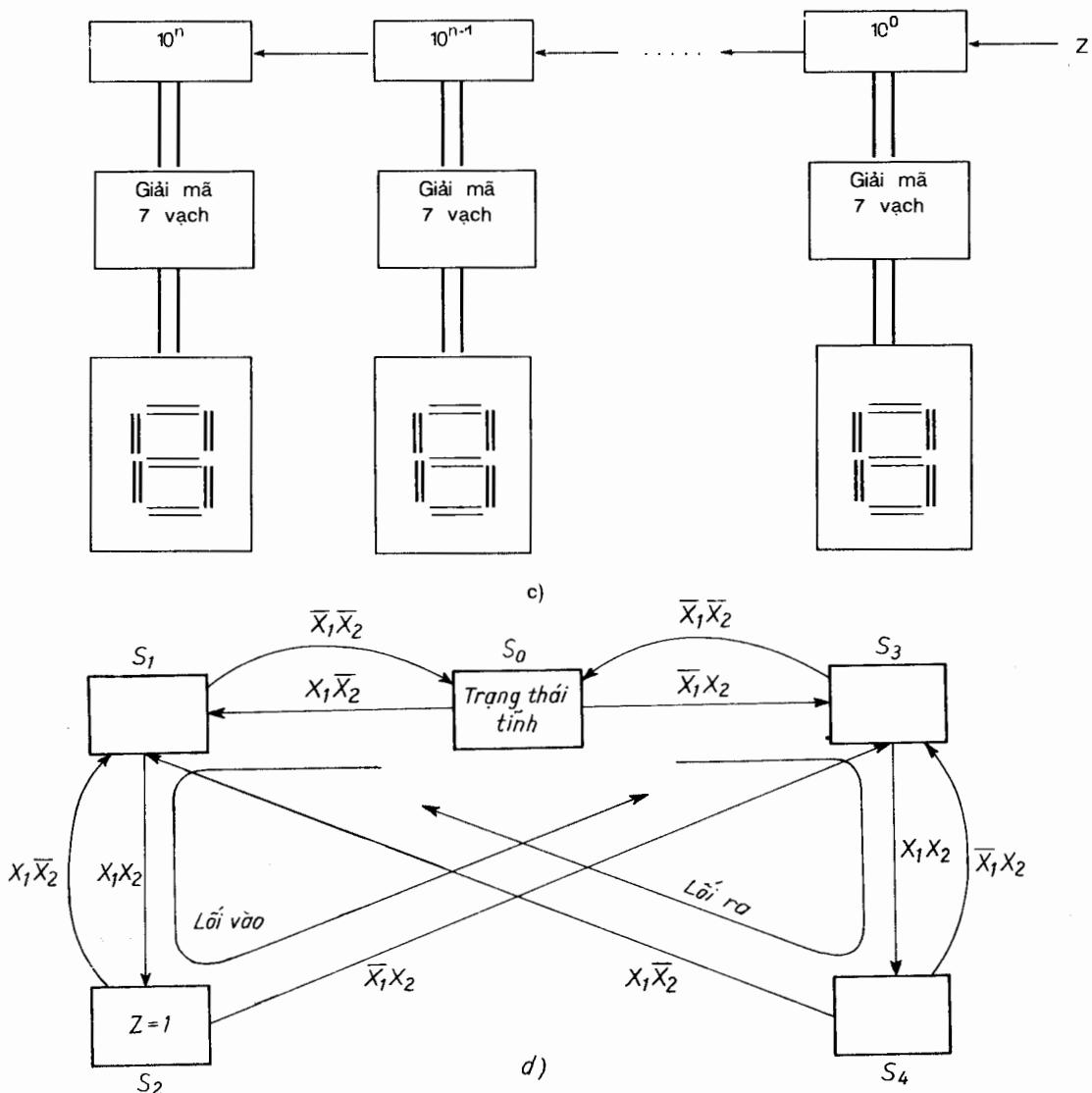


Hình 11-2. a) Bố trí các chùm sáng ở cửa vào viện bảo tàng;
b) Sơ đồ khối của mạch.

Khi có một người đi vào thì hai chùm sáng sẽ bị chắn liên tiếp. Đầu tiên X_1 bị chắn, tiếp đến cả X_1 và X_2 cùng bị chắn, sau đó đến X_2 bị chắn. Khi đó mạch cho ra tín hiệu $Z = 1$.

Sơ đồ khối của mạch tạo tín hiệu đếm Z được mô tả ở hình 11-2,b.

Hai đầu vào của mạch là X_1, X_2 . Khi một người đi vào sẽ ngăn chùm sáng thứ nhất (X_1) tiếp đó ngăn hai chùm sáng là X_1 và X_2 sau đó ngăn chùm sáng thứ hai là X_2 thì mạch cho tín hiệu ra là $Z = 1$. Đầu ra này được đưa vào đầu vào của một bộ giải mã như hình 11-2,c:



Hình 11-2. c) Bộ giải mã ; d) Đồ hình trạng thái của mạch đếm số người.

Giả sử qui ước : chùm sáng bị chấn = X ; ngược lại không bị chấn = \bar{X} .

Đồ hình trạng thái của mạch được mô tả trong hình 11-2,d.

S_0 là trạng thái ban đầu của mạch . Nếu một người đi vào sự chuyển đổi trạng thái của mạch sẽ là $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow S_3 \Rightarrow S_0$. Nếu một người đi ra thì quá trình chuyển đổi trạng thái của mạch sẽ là : $S_0 \Rightarrow S_3 \Rightarrow S_4 \Rightarrow S_1 \Rightarrow S_0$. Khi một người vào ngập ngừng sau đó lại quay ra ban đầu chấn chùm sáng X_1 sau đó quay ra mạch sẽ chuyển đổi trạng thái như sau : $S_0 \Rightarrow S_1 \Rightarrow S_0$ lúc đó mạch sẽ không đếm.

Tương ứng với đồ hình trạng thái hình 11-2,d, lập bảng trạng thái, bảng ra trên hình 11-2,e.

Trạng thái hiện tại	Trạng thái tiếp theo và tín hiệu ra							
	X_1 0	X_2 0	X_1 0	X_2 1	X_1 1	X_2 1	X_1 1	X_2 0
S_0	S_0 $Z = 0$		S_3 $Z = 0$				S_1 $Z = 0$	
S_1	S_0 $Z = 0$				S_2 $Z = 1$		S_1 $Z = 0$	
S_2			S_3 $Z = 0$		S_2 $Z = 1$		S_1 $Z = 0$	
S_3	S_0 $Z = 0$		S_3 $Z = 0$		S_4 $Z = 0$			
S_4			S_3 $Z = 0$		S_4 $Z = 0$		S_1 $Z = 0$	

Hình 11-2.e) Bảng trạng thái, bảng ra .

Bảng có 5 hàng ứng với 5 trạng thái hiện tại có thể xuất hiện và 4 cột ứng với một tổ hợp giá trị có thể của X_1, X_2 . Mỗi ô của bảng biểu diễn trạng thái tiếp theo và tín hiệu ra ứng với trạng thái hiện thời và giá trị của tín hiệu vào X_1, X_2 .

Ví dụ hàng đầu tiên của bảng ứng với trạng thái hiện tại là S_0 , nếu tín hiệu vào là $X_1X_2 = 01$ mạch sẽ chuyển tới trạng thái S_3 và tín hiệu ra $Z = 0$. Trong ô ghi rõ S_3 và $Z = 0$.

Trong bảng trạng thái 11-2.e những ô ở đó trạng thái được khoanh tròn là những ô có trạng thái tiếp theo bằng chính trạng thái hiện tại. Những trạng thái đó gọi là các trạng thái ổn định. Ví dụ trạng thái hiện tại là S_0 với tín hiệu vào là $X_1 X_2$ là 00, trạng thái tiếp theo của mạch vẫn được giữ nguyên là S_0 . Điều kiện cho trạng thái ổn định là $S' = S$, (trong đó S là trạng thái hiện tại còn S' là trạng thái tiếp theo).

Trên bảng có những ô trống. Những ô này tương ứng với các tổ hợp tín hiệu không xuất hiện ở đầu vào. Thật vậy do cách sắp xếp của hai chùm sáng, X_1 và X_2 không thể thay đổi giá trị cùng một lúc được. Ví dụ khi hai chùm sáng $X_1 X_2$ đều đang không bị chấn ($X_1 X_2 = 00$) tương ứng với trạng thái ban đầu S_0 , không thể ngay tức khắc cả hai chùm sáng đều bị chấn ($X_1 X_2 = 11$). Tổ hợp tín hiệu vào này không xuất hiện tương ứng với ô trống ở hàng thứ nhất.

Nếu bảng trạng thái, bảng ra có những ô trống thì có thể điền giá trị không xác định (X) vào các ô đó, rồi lợi dụng để tối thiểu hoá trạng thái, hệ phương trình của mạch. Để tối thiểu hoá trạng thái tiến hành như sau :

Có thể gán trạng thái tiếp theo và tín hiệu ra vào các ô trống sao cho hàng có ô trống có thể kết hợp với các hàng khác.

Ở bảng trạng thái, bảng ra trên hình 11-2,e : các hàng S_0, S_1, S_2 và S_3, S_4 có các trạng thái tiếp theo và tín hiệu ra tương ứng là giống nhau nếu như ta gán :

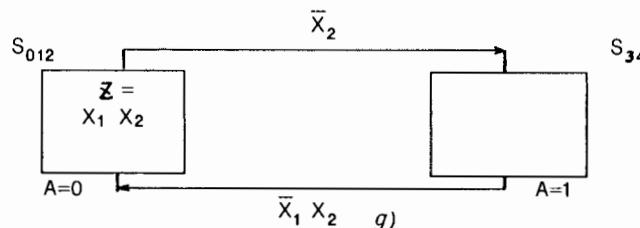
- ô trống của hàng đầu tiên (ứng với S_0) là $S_2/Z = 1$,
- ô trống của hàng thứ hai là $S_3/Z = 0$,
- ô trống của hàng thứ tư là $S_1/Z = 0$,
- ô trống của hàng thứ ba, hàng thứ năm là $S_0/Z = 0$.

Khi đó bảng 11-2,e được rút gọn thành bảng 11-2,f :

Từ bảng này ta có đồ hình trạng thái mới biểu diễn cho mạch như hình 11-2,g.

Trạng thái hiện tại	Trạng thái tiếp theo và tín hiệu ra							
	X_1 0	X_2 0	X_1 0	X_2 1	X_1 1	X_2 1	X_1 1	X_2 0
S_{012}	S_{012} $Z = 0$	S_{34} $Z = 0$	S_{012} $Z = 1$	S_{012} $Z = 0$				
S_{34}	S_{012} $Z = 0$	S_{34} $Z = 0$	S_{34} $Z = 0$	S_{012} $Z = 0$				

f)



g)

Hình 11-2.f) Bảng trạng thái, bảng ra của mạch sau khi đã tối thiểu hóa;

g) Đồ hình trạng thái của mạch.

Mạch chỉ có hai trạng thái , để mã hoá trạng thái cho mạch chỉ cần sử dụng 1 biến nhị phân A . Ví dụ mã hoá trạng thái $S_{012} : A = 0$ và trạng thái $S_{34} : A = 1$. Tín hiệu ra $Z = 1$ ở trạng thái S_{012} khi $X_1 X_2 = 11$ (hình 11-2,g).

Để thuận tiện cho việc viết phương trình của mạch cần nhắc lại qua về RS - FF.

Trong đó

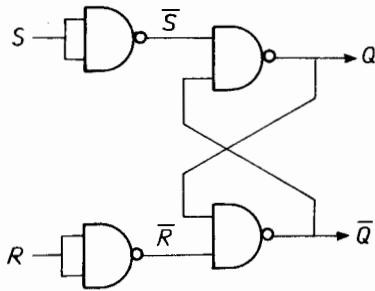
$$S = (\text{Tập hợp bật của } Q) + [(1)]$$

$$R = (\text{Tập hợp tắt của } Q) + [(0)]$$

Tập hợp bật của Q (T_{on}) là các cung mà Q chuyển từ $0 \Rightarrow 1$

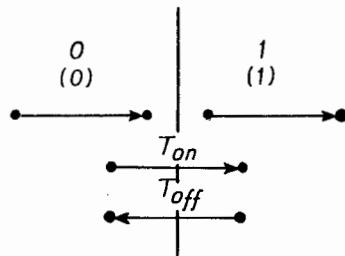
Tập hợp tắt của Q (T_{off}) là các cung mà Q chuyển từ $1 \Rightarrow 0$

Các cung $[(0)], [(1)]$ được lấy giá trị không xác định (X) và được lợi dụng để tối thiểu hóa .



R	S	Q'
0	0	Q
0	1	1
1	0	0
1	1	X

$$Q' = S + \bar{R} \cdot Q$$



L. cung	Q	Q'	R	S
0	0	0	X	0
T _{on}	0	1	0	1
T _{off}	1	0	1	0
1	1	1	0	X

Hình 11-2,h

Viết phương trình các hàm kích cho FF - A ta có :

$$S_A = (\text{Tập hợp bật của } Q) + [(1)] = \bar{A} \cdot \bar{X}_1 \cdot X_2 + A \cdot \bar{X}_1 \cdot X_2 = \bar{X}_1 \cdot X_2$$

$$\begin{aligned} R_A &= (\text{Tập hợp tắt của } Q) + [(0)] = A \cdot \bar{X}_2 + \bar{A} \cdot \bar{X}_1 \cdot \bar{X}_2 + \bar{A} \cdot X_1 \cdot \bar{X}_2 \\ &= A \cdot \bar{X}_2 + \bar{A} \cdot \bar{X}_2 = \bar{X}_2 \end{aligned}$$

Từ phương trình đặc trưng của RS - FF :

$$Q_A' = S_A + \bar{R}_A \cdot Q_A$$

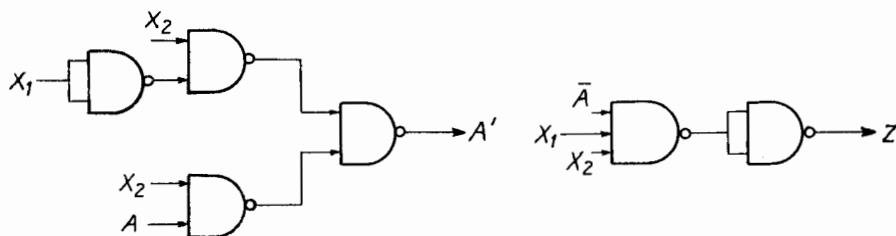
Thay giá trị của R_A , S_A vào biểu thức, nhận được kết quả :

$$A' = \bar{X}_1 \cdot X_2 + X_2 \cdot A = \overline{\bar{X}_1 \cdot X_2 + X_2 \cdot A} = (\overline{\bar{X}_1 \cdot X_2}) \cdot (\overline{X_2 \cdot A})$$

Phương trình tín hiệu ra :

$$Z = \bar{A} \cdot X_1 \cdot X_2$$

Sơ đồ mạch được biểu diễn như trên hình 11-2.i.

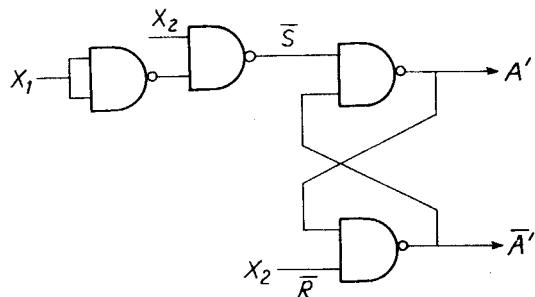


Hình 11-2.i) Sơ đồ mạch chỉ dùng *NAND*.

Nếu thiết kế mạch dùng *RS - FF* và các mạch *NAND* ta có :

$$S_A = \bar{X}_1 \cdot X_2 \quad R_A = \bar{X}_2$$

và mạch được biểu diễn trên hình 11-2,k.



Hình 11-2. k) Sơ đồ dùng *RS - FF* không đồng bộ .

11_3. HIỆN TƯỢNG CHU KỲ VÀ CHẠY ĐUA TRONG MẠCH KHÔNG ĐỒNG BỘ

Đặt vấn đề :

Đối với mạch dãy đồng bộ mã hoá trạng thái là làm sao cho sơ đồ thực hiện mạch là đơn giản nhất.

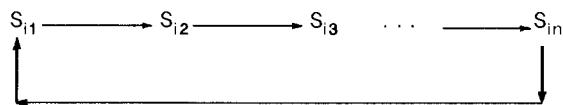
Đối với mạch dãy không đồng bộ trong mạch thường xảy ra các hiện tượng hoặc là chu kỳ hoặc là chạy đua. Những hiện tượng này làm cho mạch hoạt động sai lệch đi so với chức năng của nó. Vì vậy khi mã hoá trạng thái các mạch dãy không đồng bộ, điều đầu tiên là cần tránh các hiện tượng đó.

11_3.1. HIỆN TƯỢNG CHU KỲ TRONG MẠCH DÃY KHÔNG ĐỒNG BỘ

Định nghĩa

Hiện tượng chu kỳ là hiện tượng tại một tổ hợp tín hiệu vào nào đó, mạch liên tục chuyển từ trạng thái này sang trạng thái khác theo một chu kỳ kín. Nghĩa là trong quá trình đó không có trạng thái nào ổn định. Do vậy, khi thay đổi tín hiệu vào không xác định được mạch đang ở trạng thái nào trong dãy các trạng thái nói trên.

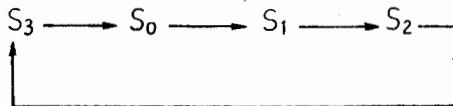
Ví dụ : ứng với một tổ hợp tín hiệu vào quá trình chuyển đổi trạng thái theo chu trình sau :



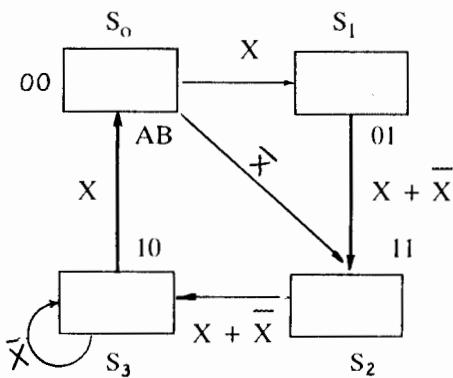
Trên bảng trạng thái hiện tượng chu kỳ được thể hiện ở chỗ : cột ứng với tổ hợp tín hiệu vào đó không có trạng thái nào được khoanh tròn (không có trạng thái nào ổn định).

Ví dụ : Đồ hình trạng thái của một mạch dãy không đồng bộ được biểu diễn trên hình 11-3,a. Việc mã hoá trạng thái sử dụng biến nhị phân A và B là hoàn toàn tùy ý . Từ đồ hình trạng thái, lập bảng trạng thái trên hình 11-3,b.

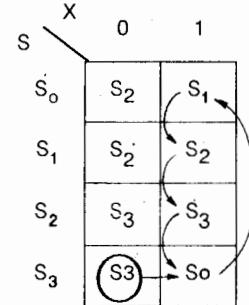
Giả thiết ban đầu mạch ở trạng thái S_3 ($AB = 10$) và $X = 0$. Sau đó tín hiệu vào X thay đổi từ 0 đến 1 mạch sẽ chuyển từ trạng thái S_3 sang S_0 . Nếu X vẫn bằng 1 mạch sẽ lặp lượt chuyển đến các trạng thái tiếp theo là $S_1, S_2 \dots S_0$ (hình 11-3,b) . Khi $X = 1$, chương trình chuyển đổi trạng thái như sau :



khi đó mạch không có trạng thái ổn định .



a)



b)

Hình 11-3. a) Đồ hình chuyển đổi trạng thái
b) Bảng chuyển đổi trạng thái .

11-3.2. HIỆN TƯỢNG CHẠY ĐUA TRONG MẠCH ĐỒNG BỘ

Định nghĩa

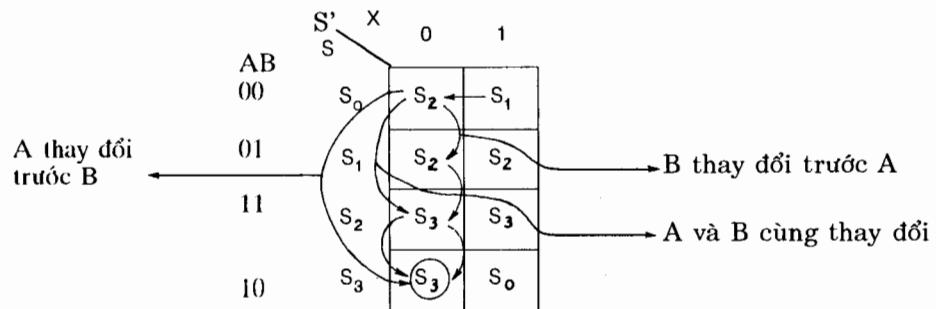
Hiện tượng chạy đua trong mạch không đồng bộ là hiện tượng : do tính không đồng nhất của các phần tử nhị phân dùng để mã hoá trạng thái, vì mạch hoạt động không đồng bộ, khi mạch chuyển trạng thái từ $S_i \Rightarrow S_j$ mạch có thể chuyển biến trạng thái theo những con đường khác nhau.

Nếu trạng thái cuối cùng của những con đường ấy là ổn định và duy nhất thì chạy đua này là chạy đua không nguy hiểm . Ngược lại, chạy đua nguy hiểm là những cách chuyển biến trạng thái khác nhau ấy cuối cùng dẫn tới các trạng thái ổn định khác nhau , có thể tới trạng thái khoá và không thoát ra được.

Ví dụ : Chạy đua không nguy hiểm : Một mạch dãy không đồng bộ có bảng trạng thái mô tả ở hình 11-3.

Nhìn vào bảng ta thấy nếu mạch đang ở trạng thái S_0 ($AB = 00$) tín hiệu vào X thay đổi từ 0 \Rightarrow 1 mạch sẽ chuyển trực tiếp tới trạng thái S_2 ($AB = 01$) và nếu X vẫn bằng

0 trạng thái tiếp theo của mạch sẽ là S_3 , nó sẽ là trạng thái ổn định cuối cùng của mạch nếu như X vẫn bằng 0.



Hình 11-4. Hiện tượng chạy đua không nguy hiểm trong mạch dãy không đồng bộ.

Mạch có thể thay đổi trạng thái theo những con đường khác nhau tùy thuộc vào thứ tự thay đổi (hay thời gian quá độ) của A và B :

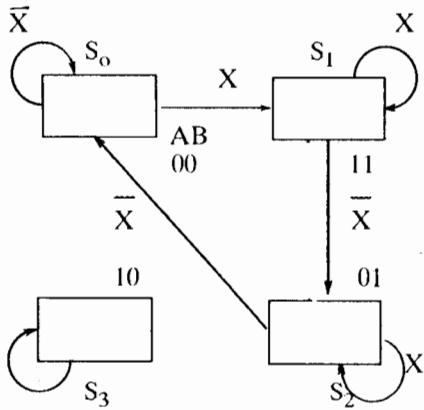
Nếu A và B thay đổi đồng thời mạch sẽ chuyển sang trạng thái S_2 rồi mới sang trạng thái S_3 .

- Nếu B thay đổi trước A thì mạch sẽ lần lượt chuyển qua S_1 , S_2 rồi mới sang S_3 .
- Tương tự nếu A thay đổi trước B mạch sẽ chuyển từ $S_0 \Rightarrow S_3$.

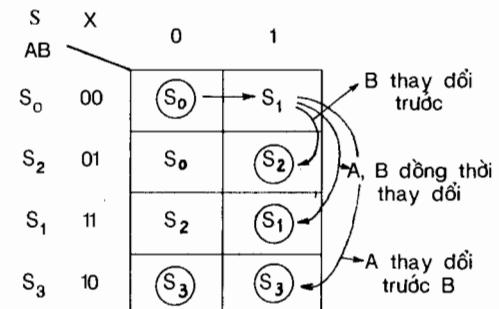
Nhận thấy rằng cả 3 con đường chuyển đổi đều dẫn đến một trạng thái ổn định S_3 . Hiện tượng chạy đua này là chạy đua không nguy hiểm.

Khi mạch đang ở trạng thái ổn định (được khoanh tròn), nó chỉ thay đổi trạng thái khi tín hiệu vào thay đổi.

* Chạy đua nguy hiểm: Đồ hình trạng thái của một mạch không đồng bộ mô tả ở hình 11-5.a.



a)



b)

Hình 11-5. Hiện tượng chạy đua nguy hiểm trong mạch không đồng bộ.

a. Đồ hình trạng thái; b. Bảng trạng thái.

Giả thiết trạng thái ban đầu của mạch là S_0 ($AB = 00$) và tín hiệu vào $X = 0$. Nếu X thay đổi từ 0 \Rightarrow 1 mạch sẽ chuyển đổi trạng thái như sau :

- Nếu A, B thay đổi đồng thời mạch sẽ chuyển tới trạng thái S_1 .
- Nếu B thay đổi trước A mạch sẽ chuyển tới trạng thái S_2 .
- Nếu A thay đổi trước B mạch sẽ chuyển tới trạng thái S_3 .

Ở đây trạng thái S_3 (xem hình 11-5,a) là trạng thái "khoá". Như vậy khi A thay đổi trước B thì mạch sẽ rơi vào trạng thái khoá và không thoát ra được.

Chạy đua này là chạy đua nguy hiểm.

11-4. TỐI THIỂU HOÁ VÀ MÃ HOÁ TRẠNG THÁI TRONG MẠCH DÂY KHÔNG ĐỒNG BỘ

11-4.1. TỐI THIỂU HOÁ TRẠNG THÁI

Tối thiểu hoá trạng thái là giảm bớt số trạng thái (nếu có thể) để mạch thiết kế là đơn giản và do vậy tin cậy hơn.

Cần lưu ý rằng tại những ô trống trong bảng chuyển đổi trạng thái và bảng ra (những ô này ứng với tổ hợp tín hiệu vào không xuất hiện) có thể lấy giá trị tùy ý để cho kết quả tối thiểu hoá trạng thái là tối giản.

11-4.2. MÃ HOÁ TRẠNG THÁI

Sử dụng các biến nhị phân để mã hoá các trạng thái trong của mạch. Gọi N là số trạng thái trong của mạch ; n là số lượng biến nhị phân dùng để mã hoá, n phải thỏa mãn : $n \geq \log_2 N$.

Đối với mạch dây đồng bộ chỉ cần tối thiểu hoá và mã hoá trạng thái sao cho mạch thực hiện là đơn giản nhất.

Đối với mạch dây không đồng bộ phải mã hoá trạng thái sao cho mạch thực hiện vừa đơn giản lại vừa tránh được hiện tượng chu kỳ và chạy đua.

Để tránh được hiện tượng chu kỳ, phải lưu ý sao cho với mọi tổ hợp tín hiệu vào thì mạch phải luôn luôn có một trạng thái ổn định.

Để tránh hiện tượng chạy đua, phải mã hoá trạng thái sao cho với tất cả các chuyển biến trạng thái có thể có của mạch chỉ có duy nhất một biến thay đổi mà thôi. Tức là các cặp từ mã hoá cho S_i, S_j ($S_i \Rightarrow S_j$) phải có khoảng cách Hamming bằng 1.

Để hiểu kỹ những vấn đề đó xét những ví dụ cụ thể sau.

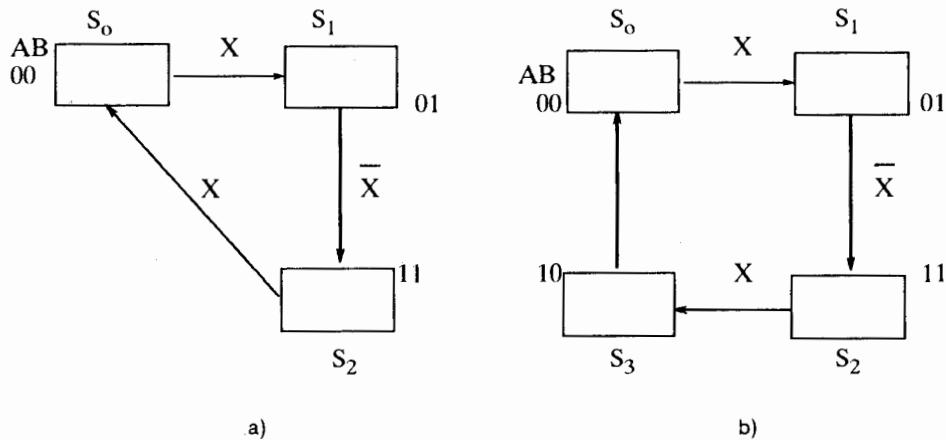
Ví dụ 1 :

Đồ hình trạng thái của một mạch dây không đồng bộ được mô tả trên hình 11-6a. Cần hai biến nhị phân A, B để mã hoá 3 trạng thái này. Giả sử chọn cách mã hoá như trong hình 11-6,a.

Với cách mã hoá này, khi thay đổi từ $S_2 \Rightarrow S_0$ cả hai biến nhị phân A và B đều thay đổi. Điều này dễ dẫn đến hiện tượng chạy đua trong mạch.

Do vậy, để tránh hiện tượng chạy đua, đưa thêm vào một trạng thái giả S_3 để cho sự thay đổi từ $S_2 \Rightarrow S_0$ thông qua trạng thái giả này bảo đảm trong quá trình thay đổi trạng

thái luôn chỉ có một biến thay đổi (xem hình 11.6,b). Đồ hình trạng thái mới này tránh được hiện tượng chạy đua .



Hình 11- 6. Tránh chạy đua trong mạch không đồng bộ.

- a. Đồ hình trạng thái ban đầu;
- b. Đồ hình trạng thái và mã hoá tránh chạy đua .

Khi sử dụng các trạng thái giả để mã hoá cho mạch cần lưu ý tìm cách cho mạch thoát khỏi các trạng thái giả đó . Phần lớn các trường hợp ta cho mạch thoát khỏi các trạng thái giả đó vô điều kiện.

Ví dụ 2 :

Đồ hình trạng thái của một mạch 4 trạng thái với sự mã hoá cho các trạng thái này dùng hai biến A và B được biểu diễn trên hình 11-7. Với đồ hình này tất cả các quá trình chuyển đổi trạng thái đều là chuyển đổi giữa các trạng thái kế cận (nghĩa là luôn luôn bảo đảm chỉ có một biến thay đổi) .

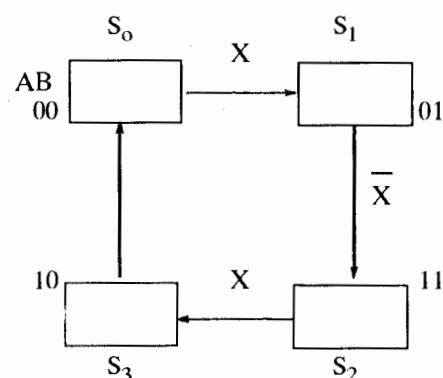
Ví dụ : $S_0 \Rightarrow S_1 ; S_1 \Rightarrow S_2 \dots$

Hình 11- 7: Đồ hình trạng thái với cách mã hoá không có chạy đua.

Giả sử trên đồ hình trạng thái của mạch có thêm sự chuyển đổi trạng thái giữa hai trạng thái giữa S_3 và S_1 (hình 11-8,a). Lúc này để mã hoá trạng thái sao cho tránh được hiện tượng

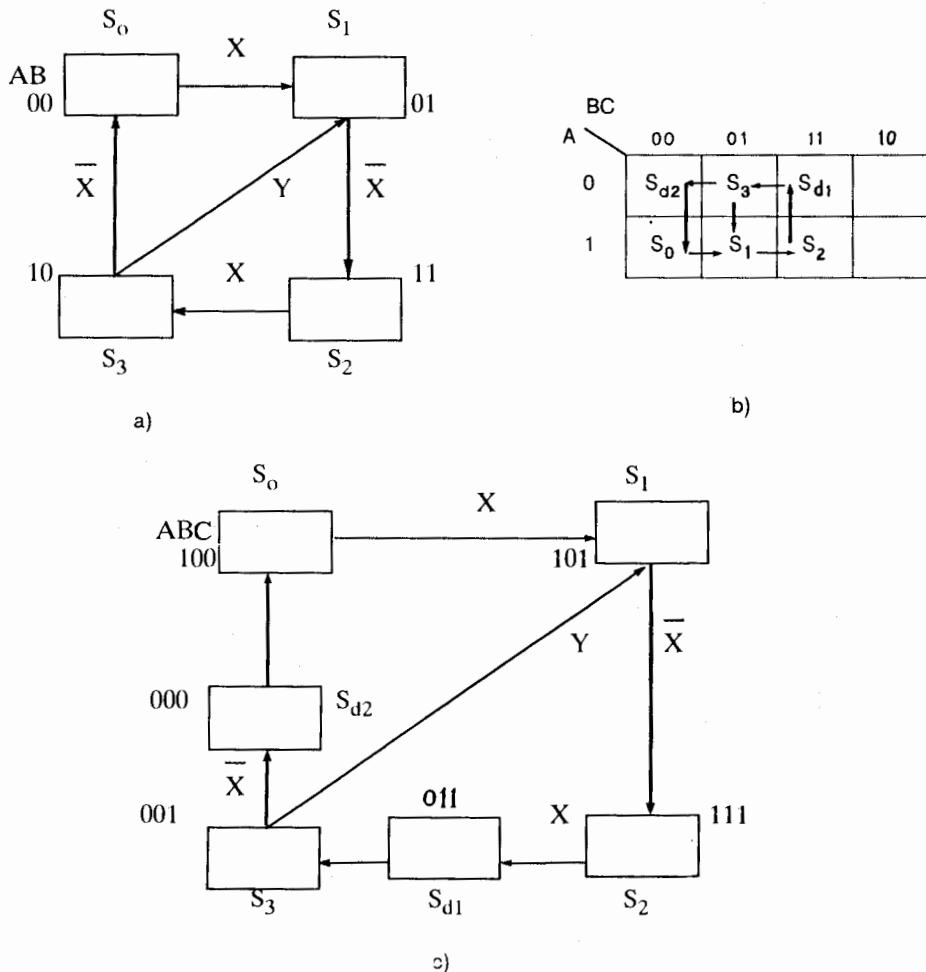
chạy đua không thể chỉ dùng hai biến nhị phân A, B được vì khi S_3 (10)

chuyển sang S_1 (01) cả A và B đều thay đổi. Như vậy để tránh hiện tượng chạy đua phải dùng thêm một biến nhị phân nữa (C).



Hình 11-7. Đồ hình trạng thái với cách mã hoá không có chạy đua.

Trên bảng Karnaugh của 3 biến nhị phân mã hoá để không có hiện tượng chạy đua cho các trạng thái của mạch (hình 11-8,b). Để tránh chạy đua ta thêm vào hai trạng thái giả là : S_{d1} , S_{d2} và các chuyển đổi $S_3 \Rightarrow S_0$, $S_2 \Rightarrow S_3$ được chỉ ra trên hình 11-8, b . Đồ hình trạng thái bao gồm hai trạng thái giả được mô tả trên hình 11-8,c.

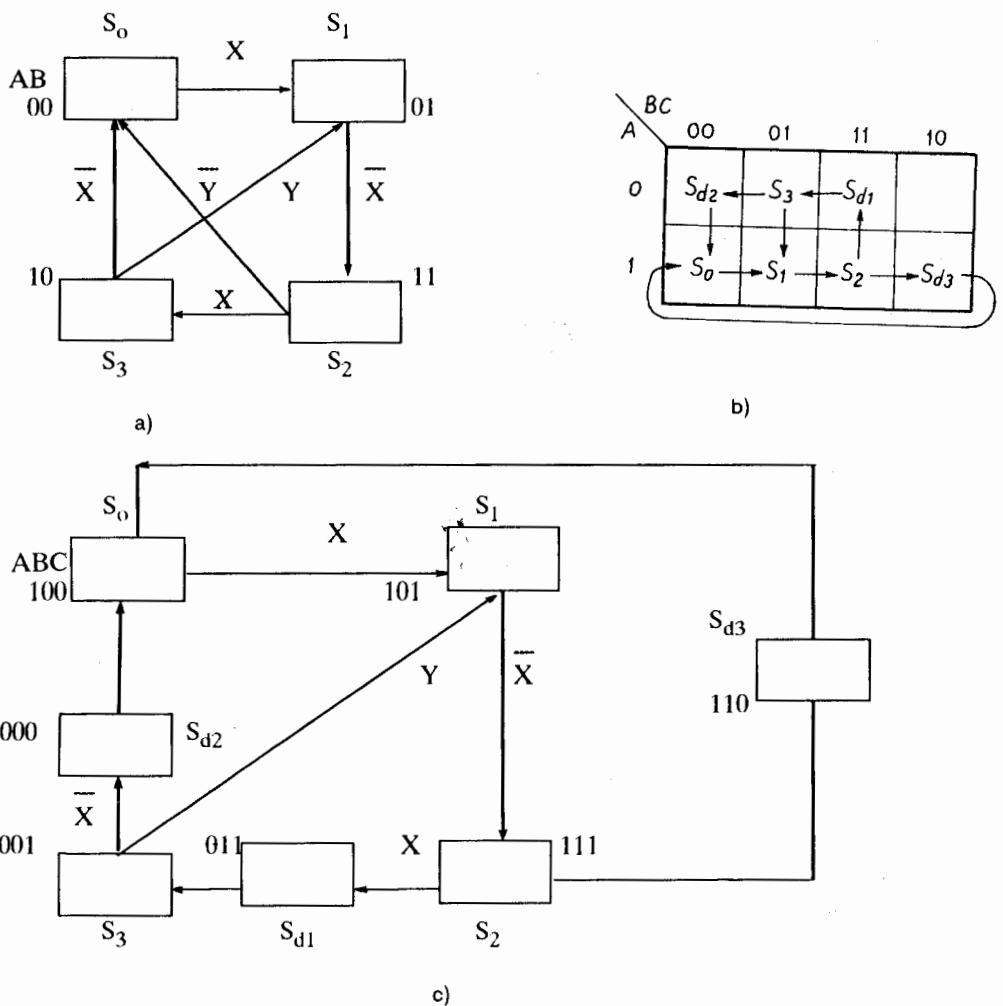


Hình 11-8. Mã hoá trạng thái tránh hiện tượng chạy đua .

- Có chạy đua ;
- Bảng mã hoá các trạng thái, có thêm trạng thái giả để tránh hiện tượng chạy đua ;
- Đồ hình trạng thái sau khi đã mã hoá tránh chạy đua .

Nếu như mạch lại có thêm chuyển đổi giữa hai trạng thái $S_2 \Rightarrow S_0$ (hình 11-9,a). Mã hoá trạng thái trong của mạch như bảng Karnaugh hình 11-8,b trong đó 4 trạng thái : S_0 , S_1 , S_2 , S_3 vẫn dùng mã như hình 11-8,b còn chuyển đổi giữa hai trạng thái S_2 , S_0 được thông qua trạng thái giả S_{d3} . Đồ hình trạng thái của mạch được biểu diễn trên hình 11-9,c.

Tóm lại khi mã hoá các trạng thái trong của mạch để tránh hiện tượng chu kỳ và chạy đua trong nhiều trường hợp phải dùng thêm biến nhị phân để mã hoá, đưa thêm các trạng thái giả , và sự chuyển đổi từ trạng thái giả về trạng thái sau là không điều kiện.



Hình 11-9. Mã hoá trạng thái để tránh hiện tượng chạy đua.

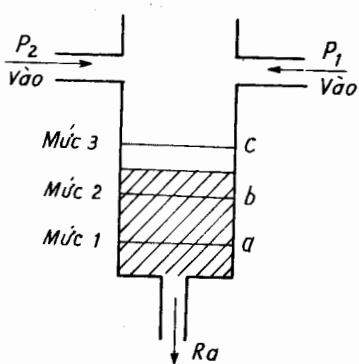
- Đồ hình chuyển đổi trạng thái ban đầu;
- Bảng mã hoá các trạng thái;
- Đồ hình chuyển đổi trạng thái sau khi đã mã hoá tránh đua.

11-5. MỘT SỐ VÍ DỤ VỀ THIẾT KẾ MẠCH DÃY KHÔNG ĐỒNG BỘ

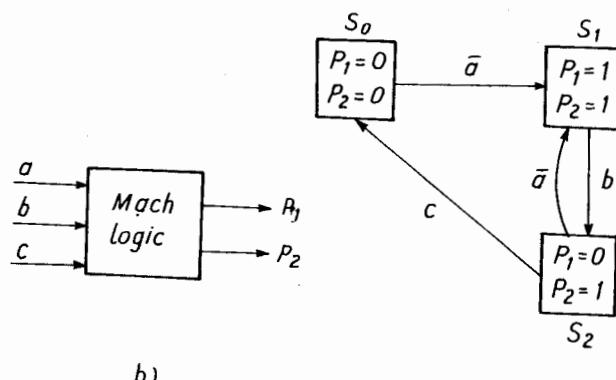
Để hiểu kỹ về mạch dây không đồng bộ và các vấn đề thiết kế mạch dây không đồng bộ, hãy xét một số ví dụ thiết kế.

11-5.1. VÍ DỤ 1

Thiết kế mạch điều khiển bơm nước vào một tháp nước nhờ hai bơm P_1 và P_2 . Cả hai bơm P_1 và P_2 được mở (bơm nước) khi nước ở dưới mức 1 và vẫn mở cho đến khi nước chưa đạt tới mức 2. Khi nước vừa đạt tới mức 2 thì bơm P_1 ngắt (không bơm nước), chỉ còn P_2 vẫn bơm. P_1 vẫn ngắt cho đến khi nước lại ở dưới mức 1. P_2 vẫn mở, chỉ khi nào nước đạt tới mức 3 thì P_2 mới ngắt. P_2 vẫn ngắt, chỉ mở khi nước lại xuống dưới mức 1.



a)

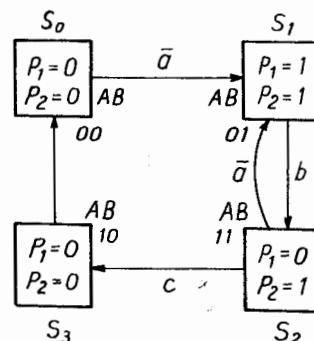


b)

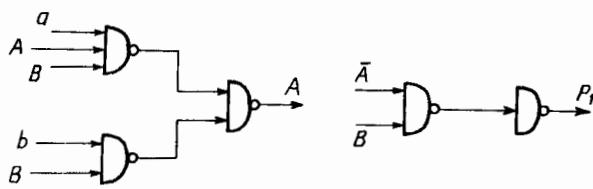
c)

		abc	000	100	110	111
		S ₀	S ₁ P ₁ =1 P ₂ =1	S ₀ P ₁ =0 P ₂ =0	S ₀ P ₁ =0 P ₂ =0	S ₀ P ₁ =0 P ₂ =0
S ₀		S ₁	(S ₁) P ₁ =1 P ₂ =1	(S ₁) P ₁ =1 P ₂ =1	S ₂ P ₁ =0 P ₂ =1	
S ₁	S ₂	S ₁ P ₁ =1 P ₂ =1	S ₂ P ₁ =0 P ₂ =1	S ₂ P ₁ =0 P ₂ =1	S ₀ P ₁ =0 P ₂ =0	
S ₂		S ₁ P ₁ =1 P ₂ =1	(S ₂) P ₁ =0 P ₂ =1	(S ₂) P ₁ =0 P ₂ =1	S ₀ P ₁ =0 P ₂ =0	

d)



e)



f)

Hình 11-10. Thiết kế mạch điều khiển bơm nước.

- a. Mô tả bài toán ; b. Sơ đồ khối cho bộ điều khiển bơm ; c. Đồ hình trạng thái của bộ bơm nước ; d. Bảng trạng thái ; e. Sơ đồ trạng thái đã sửa đổi ; f. Mạch điều khiển bơm được cài đặt.

Một bộ cảm biến được sử dụng để đo nước đưa ra các tín hiệu như sau :

$a = 1$ khi mức nước lớn hơn hoặc bằng mức 1, trường hợp khác $a = 0$

$b = 1$ khi mức nước lớn hơn hoặc bằng mức 2, trường hợp khác $b = 0$

$c = 1$ khi mức nước lớn hơn hoặc bằng mức 3, trường hợp khác $c = 0$.

Mã hoá cho trạng thái của bơm : $P = 1$ bơm mở ; $P = 0$ bơm ngắt .

Hình 11-10,a và 11-10,b biểu diễn nguyên tắc bơm nước vào tháp và sơ đồ khối của mạch.

Đồ hình trạng thái trong :

Từ các dữ liệu trên, lập được đồ hình trạng thái (hình 11-10,c) :

- Trạng thái S_0 tương ứng khi nước ở lớn hơn hoặc bằng mức "3" cả 2 bơm P_1, P_2 đều ngắt.

- Khi nước tụt xuống dưới mức "1" (\bar{a}) mạch chuyển sang trạng thái S_1 lúc đó cả 2 bơm đều mở.

- Đến khi nước đạt mức "2" (b) mạch chuyển sang trạng thái S_2 bơm P_1 vẫn bơm, bơm P_2 tắt . Có 2 trường hợp có thể xảy ra :

Nếu nước giảm xuống dưới mức "1" (\bar{a}) chuyển sang trạng thái S_1 : P_1 vẫn bơm, P_2 tắt .

Nếu nước tiếp tục dâng lên đến mức "3" (c) mạch sẽ chuyển sang trạng thái S_0 và cả 2 bơm đều tắt.

Bảng trạng thái :

Từ đồ hình trạng thái, lập được bảng trạng thái ở hình 11- 10,d. Do ý nghĩa vật lý của các biến vào , chỉ có thể xuất hiện ở đầu vào bốn bộ giá trị sau của abc : 000, 100, 110, 111.

Mã hoá trạng thái :

Từ bảng trạng thái hình 11- 10,d thấy rằng : ứng với mỗi cột của tổ hợp tín hiệu vào mạch luôn có ít nhất một trạng thái ổn định , nghĩa là trong mạch không có hiện tượng chu kỳ .

Như vậy, khi mã hoá trạng thái chỉ còn phải tránh hiện tượng chạy đua . Để mã hoá trạng thái của mạch, cần 2 biến nhị phân A, B . Và để tránh hiện tượng chạy đua thêm trạng thái giả S_3 như hình 11- 10,e.

Hệ phương trình của mạch :

Từ đồ hình trạng thái hình 11- 10,e viết được hệ phương trình của mạch như sau :

$$S_A = (\text{Tập hợp bật của } A) + [(1)] = b \cdot B$$

$$R_A = (\text{Tập hợp tắt của } A) + [(0)] = \bar{B} + B \bar{a} = \bar{B} + \bar{a}$$

$$S_B = (\text{Tập hợp bật của } B) + [(1)] = \bar{a} \bar{A}$$

$$R_B = (\text{Tập hợp tắt của } B) + [(0)] = c A$$

$$\text{Vậy : } A' = b \cdot B + (\bar{B} + \bar{a}) \cdot A = b \cdot B + a \cdot A \cdot B$$

$$B' = \bar{a} \cdot \bar{A} + \overline{c \cdot A} \cdot B = \bar{a} \cdot \bar{A} + (\bar{c} + \bar{A}) \cdot B$$

Và : $P_1 = \bar{A} \cdot B$

$$P_2 = \bar{A} \cdot B + A \cdot B = B$$

Mạch thực hiện : như vẽ ở hình 11-9, f.

11-6.2. VÍ DỤ 2

Thiết kế mạch kiểm tra dây tín hiệu vào $X_1 X_2$, mạch sẽ cho tín hiệu ra $Z = 1$ trên đầu ra Z khi 2 tín hiệu vào $X_1 X_2$ đưa liên tiếp ở đầu vào là $X_1 X_2 = 00, 10, 11$.

Tín hiệu vào, ra :

Sơ đồ khối của mạch được biểu diễn ở hình 11-11,a.

Đồ hình trạng thái, bảng trạng thái và bảng ra :

Từ các dữ kiện trên, vẽ được đồ hình trạng thái hình 11-11,b. Giữa 4 trạng thái này có 3 đường chuyển đổi trạng thái tương ứng với các tổ hợp tín hiệu vào :

00, 10, 11 ($\bar{X}_1 \bar{X}_2, X_1 \bar{X}_2$, và $X_1 X_2$). Theo đầu bài khi đó $Z = 1$.

Sau đó hoàn thiện đồ hình này bằng cách đưa thêm các đường chuyển đổi ứng với các tổ hợp tín hiệu vào khác.

Giả thiết rằng mạch được thiết kế hoạt động theo nguyên tắc : hai tín hiệu vào X_1, X_2 không thay đổi giá trị cùng một lúc. Ví dụ mạch chuyển trạng thái từ $S_0 \rightarrow S_1$ với tín hiệu vào là : $\bar{X}_1 \bar{X}_2$ thì dây tín hiệu vào tiếp theo chỉ có thể có thể là $\bar{X}_1 X_2$ hoặc $X_1 \bar{X}_2$.

Xét tương tự với các trạng thái khác được đồ hình như hình 11-11,c.

Tiếp tục hoàn thiện đồ hình với các tín hiệu vào để mạch ở trạng thái ổn định, nhận được đồ hình trạng thái hình 11-11,d. Từ đó lập bảng trạng thái, bảng ra hình 11-10,e.

Tối thiểu hóa trạng thái, mã hóa trạng thái

Dựa vào bảng 11-11,d, tối thiểu hóa trạng thái được bảng hình 11-11,f.

Ô trống ở bảng 11-11,f là ô ứng với tổ hợp tín hiệu vào không xuất hiện khi mạch ở trạng thái S_3 . Do vậy có thể chọn tùy ý, ví dụ chọn $S_3 \rightarrow S_0$, khi đó đồ hình tương ứng biểu diễn trên hình 11-11,g.

Mạch chỉ có chỉ có 3 trạng thái. Để mã hóa tránh chạy đua, đưa thêm vào một trạng thái giả và sự chuyển đổi từ trạng thái giả này sang trạng thái tiếp theo là không điều kiện. Cuối cùng được đồ hình trạng thái như hình 11-11,h (ở các đồ hình 11-11,g và 11-11,h để đơn giản không vẽ các cung tại đỉnh).

Hệ phương trình và sơ đồ mạch thực hiện.

Trong phần này để minh họa, trình bày hai phương pháp thiết kế khác nhau : thiết kế từ đồ hình chỉ dùng *NAND* và thiết kế từ bảng dùng *RS - FF* và các mạch *NAND*. Đối với các loại phẩn tử khác cách làm hoàn toàn tương tự.

a) *Thiết kế từ đồ hình, chỉ dùng các mạch NAND*

Từ đồ hình có :

$$\begin{aligned}
 S_A &= (\text{Tập hợp bật của } A) + [(1)] \\
 &= \bar{A} \bar{B} \bar{X}_1 X_2 + \bar{A} \cdot B (\bar{X}_1 + \bar{X}_2) + [A \bar{B} \bar{X}_1 X_2 + A B (\bar{X}_1 + \bar{X}_2)].
 \end{aligned}$$

Điền các giá trị của hàm trên vào bảng Karnaugh 4 biến (A, B, X_1, X_2) trong đó các giá trị của tập hợp bật điền "1"; các giá trị của cung loại (1) điền "X" (không xác định). Tối thiểu hóa hàm này được kết quả :

$$S_A = \bar{X}_1 X_2 + B \bar{X}_2.$$

Tương tự ta có :

$$R_A = (\text{Tập hợp tắt của } A) + [(0)] = \bar{B} \bar{X}_1 \bar{X}_2$$

$$A' = \bar{X}_1 X_2 + B \bar{X}_2 + (B + X_1 + X_2) A$$

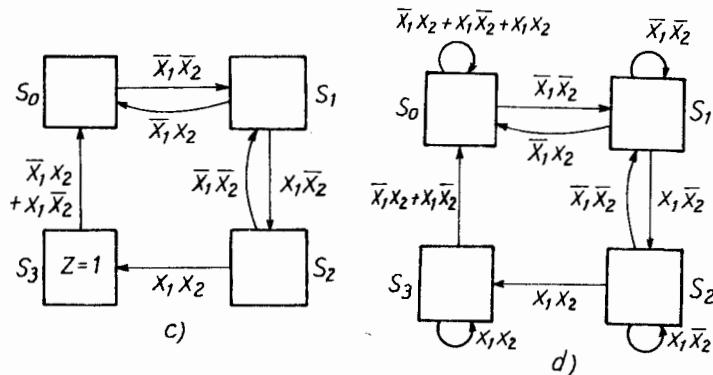
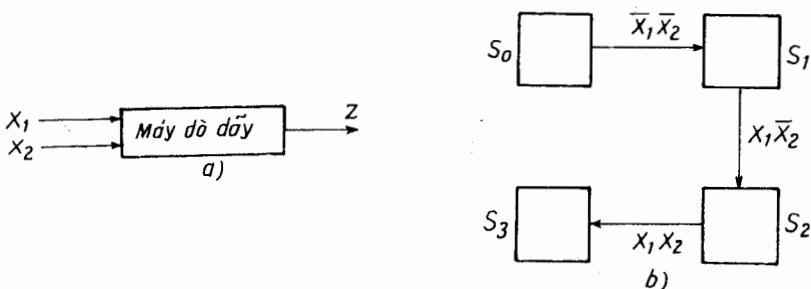
$$S_B = (\text{Tập hợp bật của } B) + [(1)] = \bar{A} X_1 X_2$$

$$R_B = (\text{Tập hợp tắt của } B) + [(0)] = A$$

$$B' = \bar{A} X_1 X_2 + \bar{A} B$$

$$Z = \bar{A} B.$$

Từ hệ phương trình của mạch vẽ được sơ đồ thực hiện mạch ở dạng chỉ dùng mạch *NAND* như hình 11-11,i.

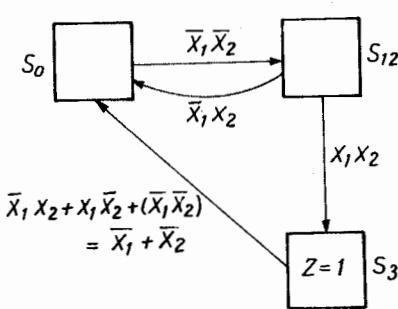


		00	01	11	10
		S_1	(S_0)	(S_0)	(S_0)
		S_1	$Z=0$	$Z=0$	$Z=0$
S_0		S_1			
S_1		S_0			S_2
S_2				S_3	(S_2)
S_3			S_0	(S_3)	S_0
			$Z=0$	$Z=1$	$Z=0$
			S_0	(S_3)	S_0
			$Z=0$	$Z=1$	$Z=0$

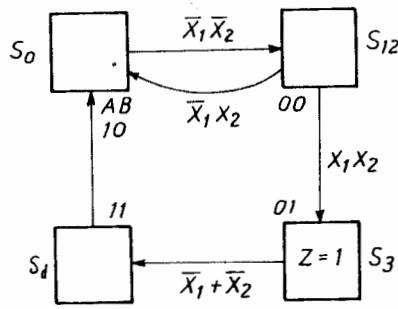
e)

		00	01	11	10
		S_{12}	(S_0)	(S_0)	(S_0)
		S_{12}	$Z=0$	$Z=0$	$Z=0$
S_0		S_{12}			
S_{12}		S_0	S_3	(S_2)	
S_3			S_0	(S_3)	S_0
			$Z=0$	$Z=1$	$Z=0$
			S_0	(S_3)	S_0
			$Z=0$	$Z=1$	$Z=0$

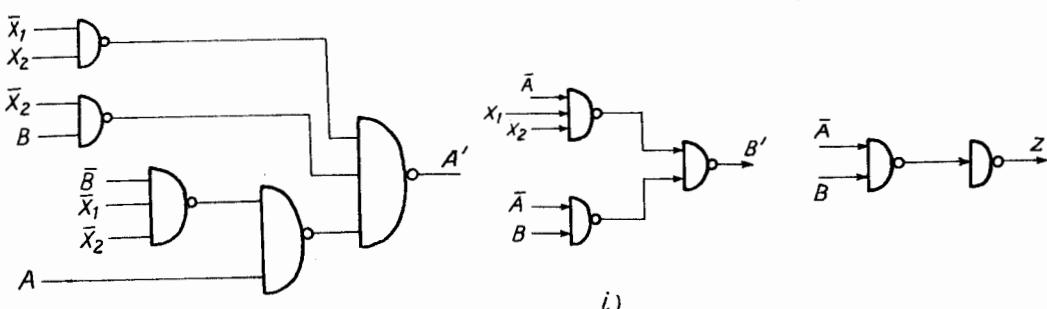
f)



g)



h)

**Hình 11- 11.** Thiết kế mạch kiểm tra dãy tín hiệu vào .

- a . Sơ đồ khối ; b . Đồ hình trạng thái (sơ lược) ; c, d . Hoàn thiện từng bước đồ hình trạng thái ; e . Bảng trạng thái, bảng ra ; f . Bảng trạng thái, bảng ra sau khi đã tối thiểu ; g . Đồ hình trạng thái sau khi đã tối thiểu ; h . Đồ hình trạng thái với trạng thái giả S_d và mã hoá tránh chạy đua ; i . Sơ đồ logic mạch thực hiện .

b) Thiết kế từ bảng trạng thái, dùng RS - FF không đồng bộ và các mạch NAND

Từ đồ hình trạng thái hình 11-11,h có bảng trạng thái hình 11-12,a. Bảng trạng thái nhị phân tương ứng được biểu diễn trên hình 11-12,b. Dựa vào cách xác định đầu vào kích cho RS - FF như bảng hình 11-12,c , xác định được giá trị tương ứng của R_A , S_A , R_B , S_B . Biểu diễn các hàm này trên bảng Karnaugh và tối thiểu hoá (hình 11-12,d), nhận được kết quả :

$$S_A = \bar{X}_1 X_2 + B \bar{X}_2$$

$$R_A = \bar{B} \bar{X}_1 \bar{X}_2$$

$$S_B = \bar{A} X_1 X_2$$

$$R_B = A$$

$$Z = S_3 = \bar{A} B .$$

Sơ đồ logic của mạch (dùng RS - FF không đồng bộ và các mạch NAND) được chỉ ra trên hình 11-12,e.

		$X_1 X_2$	00	01	11	10
		$A B$	S_{12} $Z=0$	S_0 $Z=0$	S_0 $Z=0$	S_0 $Z=0$
		S/Z	S_{12} $Z=0$	S_0 $Z=0$	S_3 $Z=1$	S_{12} $Z=0$
10						
00						
01						
11						

a)

		$X_1 X_2$	00	01	11	10
		$A B$	00	10	01	00
		$A' B'$	11	11	01	11
00						
01						
11						
10						

b)

Q^t	$Q^{t+\delta t}$	S	R
0	0	0	\emptyset
0	1	1	0
1	0	0	1
1	1	\emptyset	0

c)

S_A	$X_1 X_2$	00	01	11	10
$A B$					
00			1		
01		1	1		1
11	X	X	X	X	
10		X	X	X	X

$$S_A = \bar{X}_1 X_2 + B \bar{X}_2$$

R_A	$X_1 X_2$	00	01	11	10
$A B$					
00		X			
01				X	
11					
10		1			

$$R_A = \bar{B} \bar{X}_1 \bar{X}_2$$

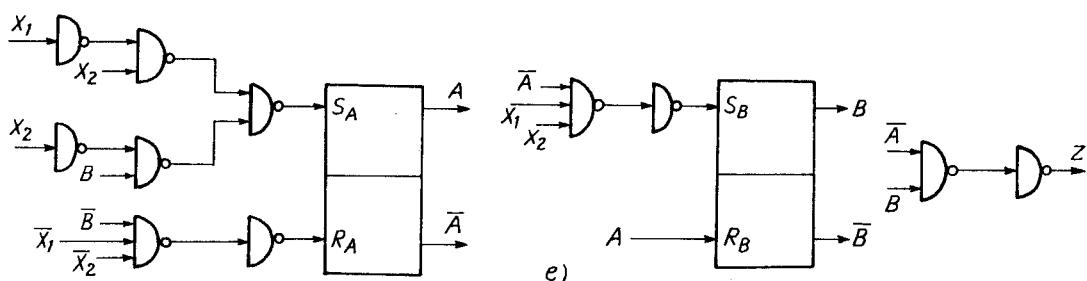
S_B	$X_1 X_2$	00	01	11	10
$A B$					
00				1	
01	X	X	X	X	
11					
10					

$$S_B = \bar{A} X_1 X_2$$

R_B	$X_1 X_2$	00	01	11	10
$A B$					
00		X	X		X
01					
11		1	1	1	1
10	X	X	X	X	

$$R_B = A$$

d)



Hình 11-12. Thiết kế mạch dùng RS - FF và mạch NAND.

- a. Bảng trạng thái và tín hiệu ra ; b. Bảng mã hoá các trạng thái ; c. Bảng chuyển đổi trạng thái của RS - FF ; d. Rút gọn phương trình các hàm đầu vào kích cho các FF ; e. Sơ đồ mạch thực hiện.

BÀI TẬP

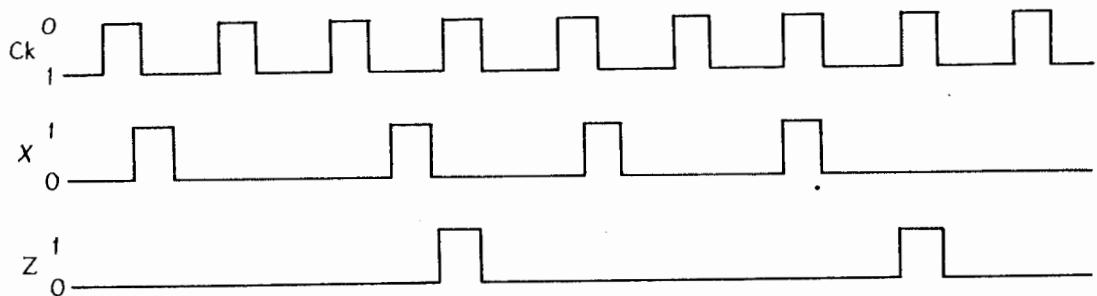
11.1. Thiết kế một mạch dãy không đồng bộ thực hiện một J^*K^* - FF có chức năng như sau :

- Nếu ban đầu $Q = 0$ khi $J^* = 1$, $K^* = 0$ và xung nhịp $Ck = 1$, tín hiệu ra Q sẽ chuyển từ 0 → 1 tại sườn âm (sườn sau) của xung nhịp.
- Nếu ban đầu $Q = 1$ khi $J^* = 0$, $K^* = 1$ và $Ck = 1$, tín hiệu ra Q sẽ chuyển từ 1 → 0 tại sườn âm của xung nhịp.
- Nếu $J^* = 1$, $K^* = 1$ và $Ck = 1$ thì FF sẽ lật trạng thái ($Q' = Q$). Vẽ giàn đồ dạng sóng cho các đầu vào, đầu ra của FF này .

11.2. Thiết kế một mạch dãy không đồng bộ thực hiện chức năng sau :

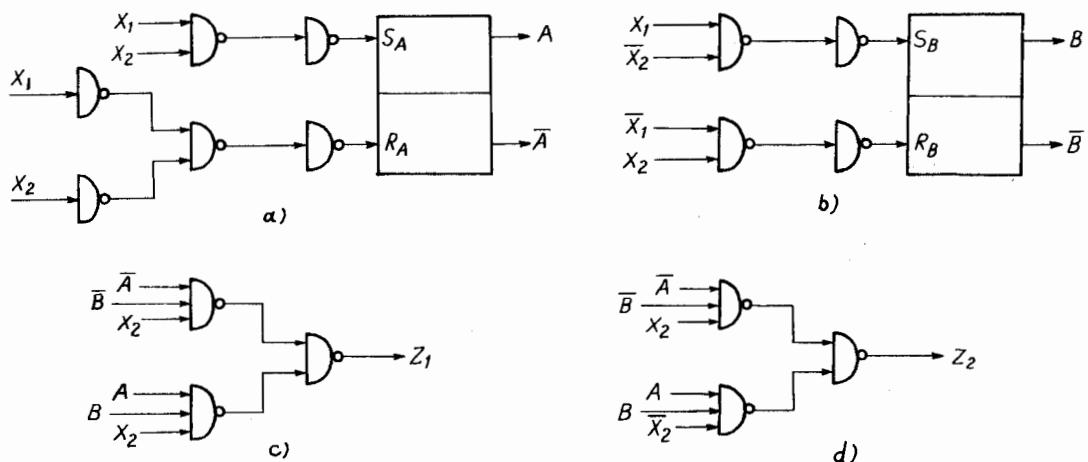
Mạch sẽ đưa ra một tín hiệu ra $Z = 1$ sau mỗi xung dữ liệu thứ 2 được đưa đến đầu vào X . Tín hiệu ra Z có độ rộng bằng độ rộng của Ck . Xung dữ liệu xuất hiện hoàn toàn ngẫu nhiên và giả thiết rằng thời gian nhỏ nhất giữa hai xung dữ liệu luôn lớn hơn chu kỳ của xung nhịp .

Gàin đồ dạng sóng của mạch được biểu diễn ở hình 11- 13 :



Hình 11- 13. Cho bài tập 11.2.

11.3. Cho mạch điện trên hình 11- 14.



Hình 11-14. Cho bài tập 11.3.

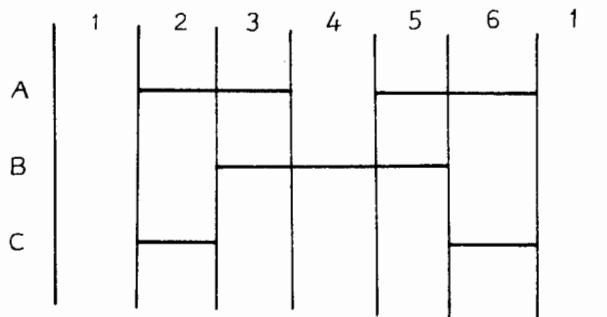
Phân tích chức năng của mạch, cụ thể :

a) Xác định bảng chuyển đổi trạng thái, bảng ra

b) Xác định đồ hình trạng thái.

c) Sử dụng bảng chuyển đổi trạng thái để xác định tín hiệu ra của mạch khi dãy tín hiệu vào $X_1 X_2 = 00, 01, 11, 10, 00, 01, 11, 01, 11, 10, 00$. Giả thiết điều kiện ban đầu là $X_1 = X_2 = 0$ và $A = B = 0$.

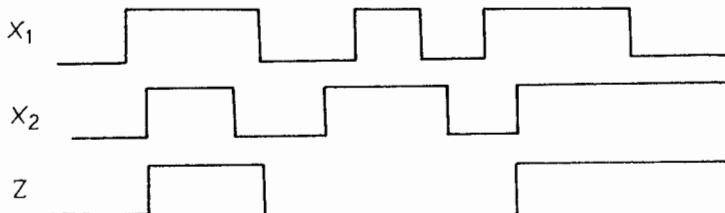
11.4. Thiết kế một mạch dãy không đồng bộ có giản đồ dạng sóng như trên hình 11-15:



Hình 11-15. Cho bài tập 11.4.

11.5. Một bộ phát hiện dãy kép (*A double - sequence detector*) có 2 đầu vào X_1, X_2 và 1 đầu ra Z .

Khi dãy tín hiệu vào $X_1 X_2$ liên tiếp là : 00, 10, 11 thì đầu ra $Z = 1$. Nếu dãy tín hiệu vào $X_1 X_2$ theo thứ tự ngược lại thì đầu ra trở về 0 ($Z = 0$) . Dạng sóng của mạch biểu diễn ở hình 11-16:



Hình 11-6. Dùng cho bài tập 11.5.

Vẽ đồ hình trạng thái, bảng trạng thái, bảng ra của mạch. Nếu có thể, rút gọn trạng thái và thiết kế mạch này dùng :

a) Những mạch *NAND* .

b) *SR - FF* và những mạch *NAND*.

11.6. X_1 và X_2 là 2 đầu vào của một mạch dãy không đồng bộ có 2 đầu ra Z_1 và Z_2 .

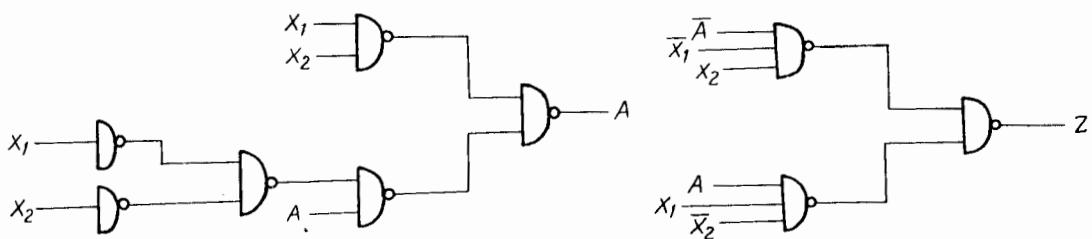
- Nếu $X_1 X_2 = 0 0$, tín hiệu ra $Z_1 Z_2 = 0 0$.

- Nếu X_1 thay đổi từ 0 đến 1 trước X_2 cũng thay đổi từ 0 đến 1, tín hiệu ra của mạch $Z_1 Z_2 = 0 1$.

- Nếu X_2 thay đổi từ 0 đến 1 trước X_1 cũng thay đổi từ 0 đến 1, tín hiệu ra của mạch là $Z_1 Z_2 = 0 1$.

Trong cả 2 trường hợp tín hiệu ra giữ nguyên ở 0 1 và 1 0 cho tới lúc $X_1 X_2$ lại bằng 0 0.

11.7. Hãy phân tích mạch biểu diễn ở hình 11- 17 :

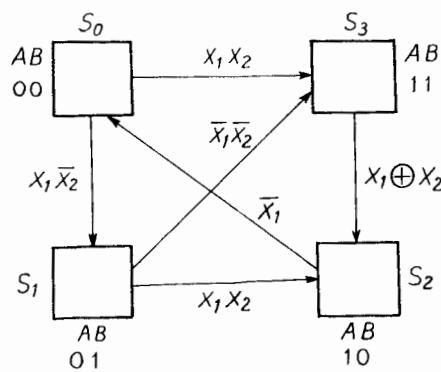


Hình 11- 17. Cho bài tập 11.7.

- Xác định bảng trạng thái.
- Xác định đồ hình trạng thái.
- Sử dụng bảng trạng thái để xác định tín hiệu ra khi dãy tín hiệu vào $X_1 X_2 = 0 0, 0 1, 1 1, 1 0, 1 1, 0 1, 0 0, 1 0, 0 0, 0 1$. Điều kiện ban đầu $X_1 = X_2 = A = 0$.

11.8. Đồ hình trạng thái cho một mạch có 4 trạng thái biểu diễn ở hình 11- 18:

- Xây dựng bảng trạng thái cho mạch và chỉ ra tất cả các chạy đua (races) cả nguy hiểm và không nguy hiểm . Với mỗi chạy đua (race), hãy chỉ ra tất cả các chuyển đổi trạng thái có thể có .
- Sửa đổi đồ hình trạng thái để tránh chạy đua trong mạch .
- Xác định hệ phương trình dùng mạch *NAND* cho mạch .



Hình 11-18. Cho bài tập 11.8.

11.9. Một mạch 3 đèn được hoạt động theo sự điều khiển của 3 chuyển mạch X, Y, Z . Nếu các chuyển mạch được bật theo thứ tự :

$X Y Z$ đèn L_1 sáng

$Y Z X$ đèn L_2 sáng

$Z X Y$ đèn L_3 sáng.

Nếu các chuyển mạch hoạt động sai thứ tự trên thì đèn đó sẽ sáng báo hiệu cho người điều khiển và người điều khiển phải đưa các chuyển mạch về trạng thái tĩnh ban đầu.

Thiết kế mạch này dùng các mạch *NAND*.

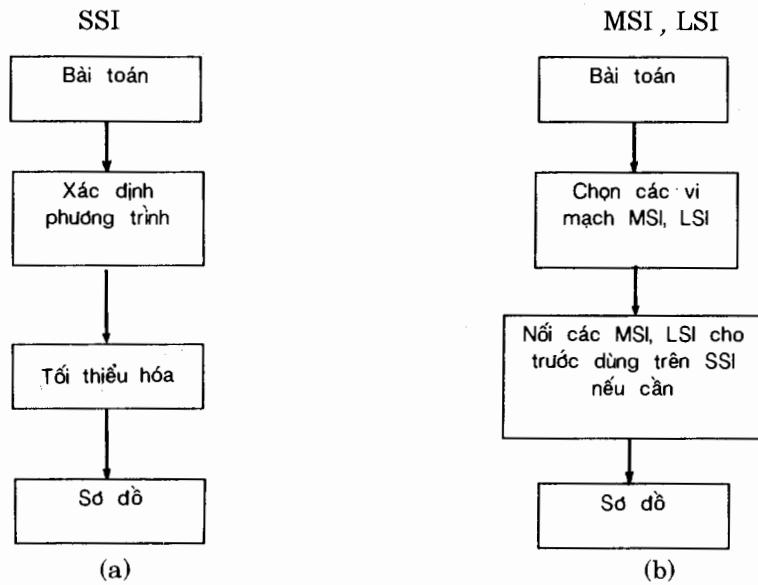
CHƯƠNG 12

THIẾT KẾ MẠCH SỐ DÙNG MSI VÀ LSI

12.1. MỞ ĐẦU

Các mạch *MSI* và *LSI* ngày càng được sử dụng rộng rãi, giá thành ngày càng hạ. Phương pháp thiết kế mạch số dùng *SSI* khác hoàn toàn với phương pháp thiết kế mạch số dùng *SSI*.

Hình 12-1 mô tả các bước thiết kế mạch số: dùng *SSI* (hình 12-1,a) dùng *MSI*, *LSI* (hình 12-1,b).



Hình 12-1. Các bước thiết kế mạch số dùng *SSI* và *MSI*, *LSI*.
a. Các bước thiết kế mạch số dùng *SSI*; b. Các bước thiết kế mạch số dùng *MSI*, *LSI*.

Giá thành của một mạch số gần như tỷ lệ thuận với số *IC* dùng trong mạch. Do vậy người thiết kế cần giải quyết bài toán : thay thế một số lớn các mạch *SSI* bằng một số ít các mạch *MSI* và *LSI*, dù rằng không sử dụng hết khả năng của các mạch *MSI* và *LSI* này nhưng như vậy nhiều khi vẫn kinh tế hơn. Trong chương này chúng ta sẽ làm quen với một số mạch *MSI* và *LSI* thường gặp như *MUX*, *DEMUX*, *DECODER*, *ROM*, *PLA* và ứng dụng của nó, đồng thời cũng tiến hành thiết kế mạch số dùng những vi mạch đó.

12.2. CÁC BỘ CHỌN DỮ LIỆU (DATA SELECTOR) HAY DÒN KÊNH (MULTIPLEXER-MUX)

Sơ đồ khối tổng quát của bộ chọn dữ liệu hay dòn kênh được mô tả như hình 12-2.

Trong thực tế

thường hay gặp các loại

MUX sau :

$MUX\ 2 \Rightarrow 1$

$MUX\ 4 \Rightarrow 1$

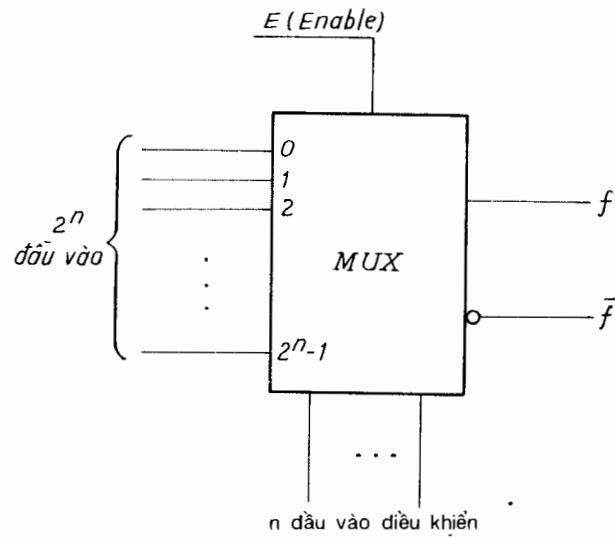
$MUX\ 8 \Rightarrow 1$

$MUX\ 16 \Rightarrow 1$

...

Ví dụ $MUX\ 4 \Rightarrow 1$

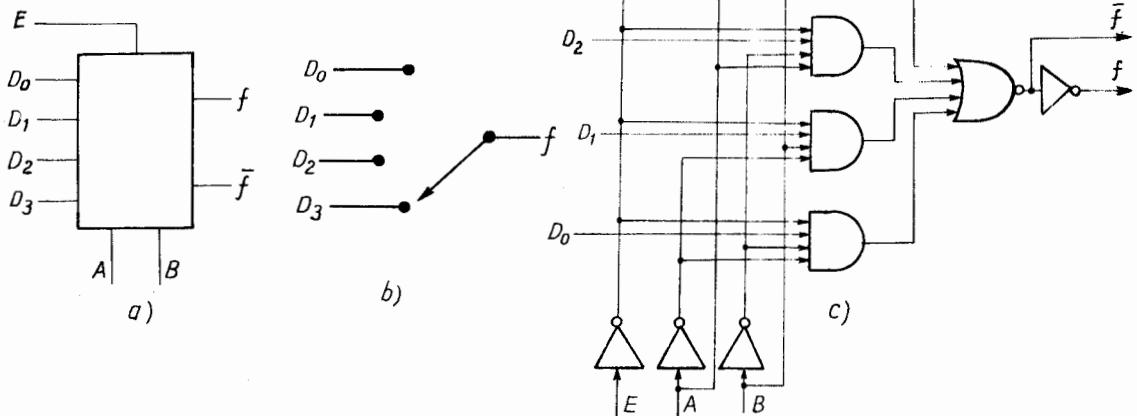
được mô tả trong hình 12-3. Hình 12-3,a là sơ đồ khối với 4 đầu đưa dữ liệu vào D_0, D_1, D_2, D_3 , hai đầu vào điều khiển việc lựa chọn một trong bốn đầu vào là A



Hình 12-2. Sơ đồ khối tổng quát của MUX.

và B. Một đầu đưa dữ liệu ra là F và một đầu đảo là F phủ định. Tín hiệu chọn mạch làm việc làm việc là E. Khi đầu vào điều khiển A và B có giá trị là 00 thì đầu ra F sẽ cho ra tín hiệu có ở đầu vào dữ liệu D_0 ($F = D_0$).

Tương tự như vậy nếu $AB = 01$ thì $F = D_1$, khi $AB = 10$ thì $F = D_2$, $AB = 11$ thì $F = D_3$.



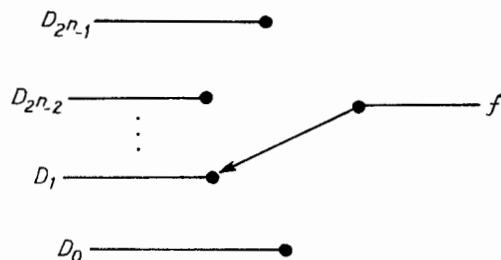
Hình 12-3. Mô tả MUX đầu vào.

a. Sơ đồ khối của MUX 4 đầu vào; b. Sự chọn 1 trong 4 đầu vào ; c. Xây dựng MUX 4 đầu vào từ các mạch tổ hợp.

12_3. CÁC ỨNG DỤNG CỦA MUX

12_3_1. MUX ĐƯỢC DÙNG LÀM BỘ CHỌN DỮ LIỆU HAY CHUYỂN MẠCH ĐIỆN TỬ

Dữ liệu vào được chọn để đưa tới đầu ra phụ thuộc vào tín hiệu hiệu điều khiển hay tín hiệu chọn. Điều này được miêu tả như trong hình 12-4.

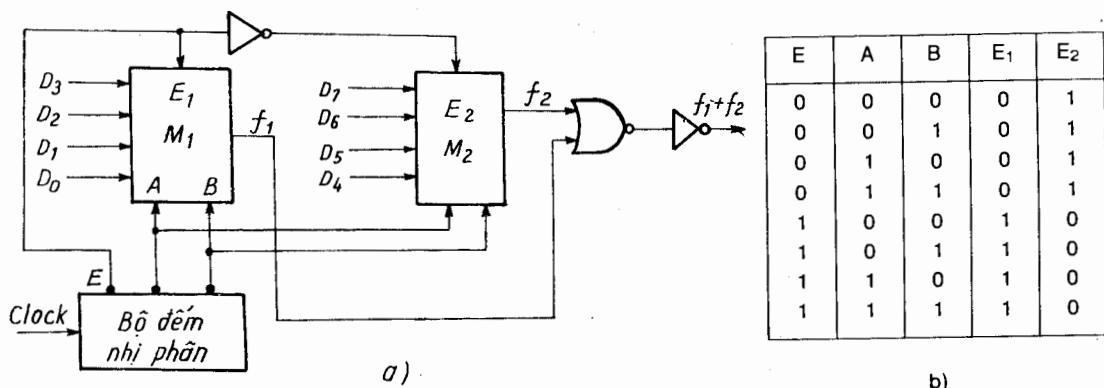


Hình 12-4. Bộ chọn dữ liệu dùng MUX.

Số các đường dữ liệu vào có thể tăng lên nhờ sử dụng MUX có nhiều đầu vào hoặc bằng cách mắc tổ hợp nhiều MUX có số đầu vào nhỏ.

Ví dụ : xây dựng MUX 8 đầu vào từ các MUX 4 đầu vào được minh họa ở hình 12-5,a. Việc điều khiển hoạt động của MUX này do một bộ đếm nhị phân 3 đầu ra thực hiện. Giá trị các đầu ra của bộ đếm này được cho trong hình 12-5,b.

Ví dụ xây dựng MUX 64 đầu vào từ các MUX 8 đầu vào được mô tả trong hình 12-6.

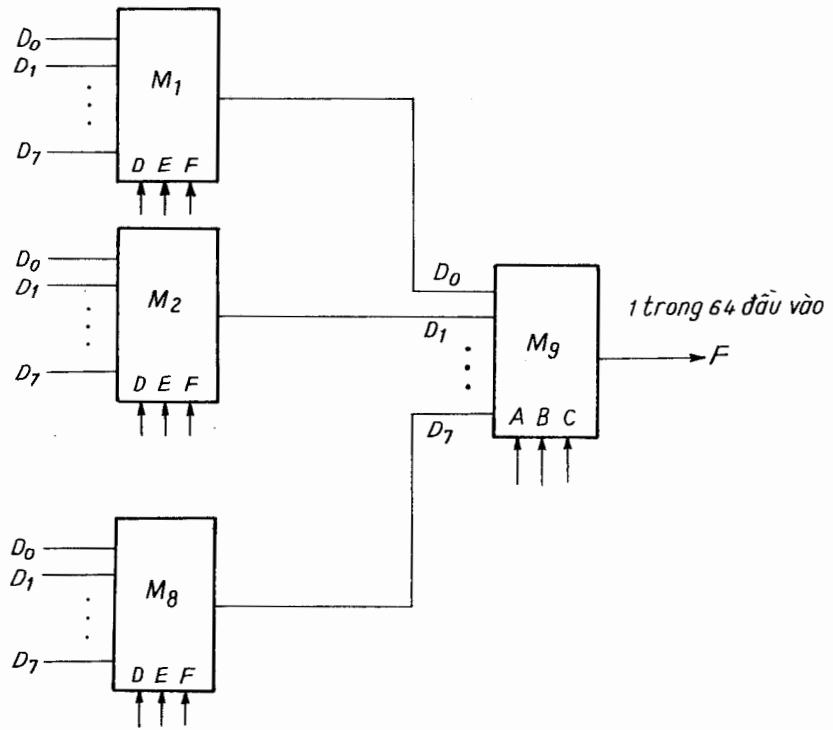


Hình 12-5. Tăng số đầu vào của MUX .

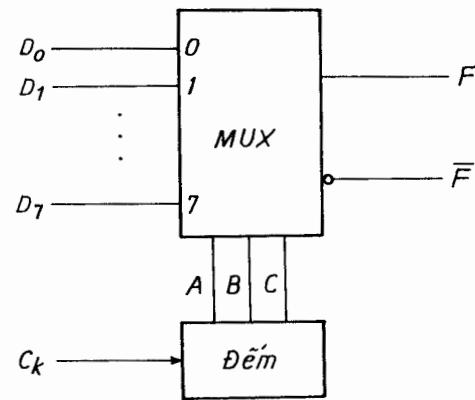
- Xây dựng MUX 8 đầu vào từ các MUX 4 đầu vào và các mạch phụ cận ;
- Bảng giá trị đầu ra của bộ đếm nhị phân.

12_3_2. BIẾN ĐỔI DẠNG THÔNG TIN VÀO SONG SONG THÀNH DẠNG NỐI TIẾP Ở ĐẦU RA

Hình 12-7 giới thiệu phương pháp biến đổi thông tin đưa đến đầu vào ở dạng song song 8 bit's thành dạng nối tiếp ở đầu ra. Việc điều khiển đưa tín hiệu ra được thực hiện nhờ một bộ đếm nhị phân 3 đầu ra như hình 12-7.



Hình 12-6. Tạo MUX 64 đầu vào từ 9 bộ MUX 8 đầu vào.



Hình 12-7. Biến đổi dạng thông tin vào song song thành dạng nối tiếp.

12.3.3. TẠO DÃY TÍN HIỆU NHỊ PHÂN TUẦN HOÀN

Muốn tạo dãy tín hiệu nhị phân tuần hoàn ta chỉ cần nối các đầu vào của *MUX* sử dụng với các mức logic nhất định. Việc thực hiện đưa tín hiệu ra nhờ một bộ đếm nhị phân mà các đầu ra của bộ đếm nhị phân này được đưa vào các đầu vào điều khiển của *MUX* đó. Ví dụ tạo dãy tín hiệu nhị phân tuần hoàn 10010011 dùng *MUX* 8 đầu vào và bộ đếm nhị phân 3 bit như hình 12-8.

12-3.4. TẠO HÀM LOGIC

Tổng quát : Một bộ MUX $2^n \Rightarrow 1$ có thể dùng để tạo hàm logic bất kỳ có $n + 1$ biến vào, trong đó n biến sẽ đưa vào n đầu điều khiển, còn 1 biến cùng với các hàng số 0 và 1 được đưa vào 2^n đầu vào còn lại tuỳ thuộc giá trị của hàm số.

Ví dụ 1: Sử dụng MUX $4 \Rightarrow 1$, tạo hàm 3 biến :

$$f = \bar{A} \bar{B} C + \bar{A} B C + \bar{A} B \bar{C} + A B C$$

Ta sẽ đưa hai biến vào đến các đầu vào điều khiển của MUX $4 \Rightarrow 1$. Có 3 khả

năng chọn 2 biến này trong 3 biến của hàm F và các khả năng đó là :

- a. 2 biến điều khiển là A và B ;
- b. 2 biến điều khiển là B và C ;
- c. 2 biến điều khiển là A và C .

Sau khi đã chọn biến điều khiển cần xác định các giá trị được đưa vào đầu vào MUX để thực hiện hàm f ban đầu. Có nhiều cách để xác định giá trị này. Ví dụ cách dùng bảng Karnaugh được tiến hành theo các bước sau :

- n biến điều khiển được phân thành 2^n vùng khác nhau trên bảng Karnaugh đánh dấu là $D_0, D_1 \dots, D_{2^n-1}$ (vùng D_i ứng với giá trị thập phân của n biến điều khiển là i);

- Diễn giá trị hàm số cho trước vào bảng Karnaugh;
- Tối thiểu hóa hàm đã cho trong từng vùng D_i , gọi hàm số này là D_i . D_i chính là giá trị đầu vào tại D_i của MUX ;

- Nếu D_i là các hàm một biến hoặc hằng 0, hằng 1 thì bài toán đã giải xong. Trong trường hợp ngược lại phải tiếp tục dùng MUX hoặc dùng các cổng logic để thực hiện các hàm D_i để đáp ứng các yêu cầu. Với trường hợp hai biến điều khiển là A và B ta có bảng Karnaugh hình 12-9,d biểu diễn giá trị của hàm F và từ đó xác định được giá trị của các hàm đưa vào các đầu vào của MUX $4 \Rightarrow 1$ là : $D_0 = C ; D_1 = 1 ; D_2 = 0 ; D_3 = C$. Sơ đồ thực hiện hàm với hai biến điều khiển A, B được mô tả ở hình 12-9,e. Tương tự với trường hợp dùng hai biến điều khiển là A, C và B, C , chúng ta có sơ đồ mạch thực hiện như hình 12-9,f và 12-9,g.

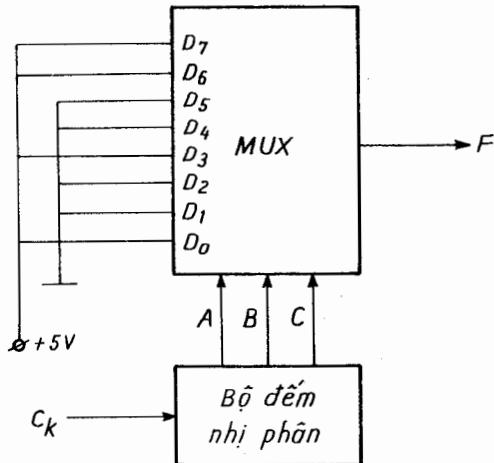
Ví dụ 2 : Thực hiện hàm 4 biến $f = \sum 0, 1, 5, 6, 7, 9, 10, 14, 15$.

- a. Dùng MUX $4 \Rightarrow 1$ và mạch $NAND$
- b. Dùng MUX $8 \Rightarrow 1$

a. *Dùng MUX $4 \Rightarrow 1$ và các mạch $NAND$*

Hai biến đưa vào đầu vào điều khiển là A và B .

Các ô trên bảng Karnaugh tương ứng với hàm của các đầu vào dữ liệu D_i được biểu diễn trong hình 12-10,a. Hàm f được biểu diễn trên bảng Karnaugh (hình 12-10,b). Từ đó ta xác định



Hình 12-8. Dùng MUX 8 đầu vào để tạo dây tín hiệu nhị phân tuần hoàn 10010011.

được các đầu vào của dữ liệu như sau :

$$D_0 = \bar{C} ; D_1 = C + D ; D_2 = \bar{C}D + C\bar{D} ; D_3 = C$$

Từ đó ta lập được sơ đồ thực hiện hàm f như hình 12-10,c.

	A	BC	00	01	11	10
0			D_0		D_1	
1			D_2		D_3	

a)

	A	BC	00	01	11	10
0			D_0	D_1		D_0
1			D_2	D_3		D_2

b)

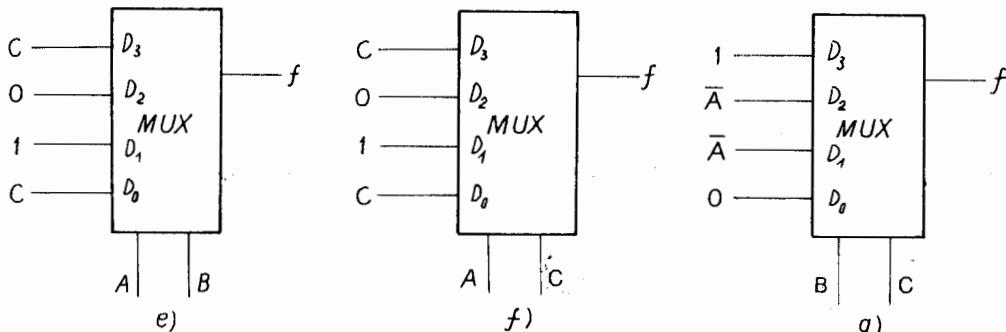
	A	BC	00	01	11	10
0			D_0	D_1	D_3	D_2
1						

c)

	A	BC	00	01	11	10
0				1	1	1
1					1	

d)

$$D_0 = C ; D_1 = 1 ; D_2 = 0 ; D_3 = C$$



Hình 12-9. Tạo hàm $f = ABC + ABC + ABC + ABC$ dùng MUX $4 \Rightarrow 1$.

a,b,c. Xác định hàm D_0, D_1, D_2, D_3 theo vị trí các ô của bảng Karnaugh, theo các biến điều khiển AB, AC, BC ; d. Mô tả hàm f bằng bảng Karnaugh và xác định các hàm D_i ; e. Sơ đồ tạo hàm f dùng MUX $4 \Rightarrow 1$ với hai biến điều khiển AB ; f. Sơ đồ tạo hàm f dùng MUX $4 \Rightarrow 1$ với hai biến điều khiển AC ; g. Sơ đồ tạo hàm f dùng MUX $4 \Rightarrow 1$ với hai biến điều khiển BC ;

Hai biến điều khiển là C, D :

Các hàm dữ liệu đầu vào của MUX $4 \Rightarrow 1$ được xác định theo các ô như hình 12-10,d. Tương tự ta cũng lập được sơ đồ thực hiện mạch như hình 12-10,e.

b. Dùng MUX 8 $\Rightarrow 1$

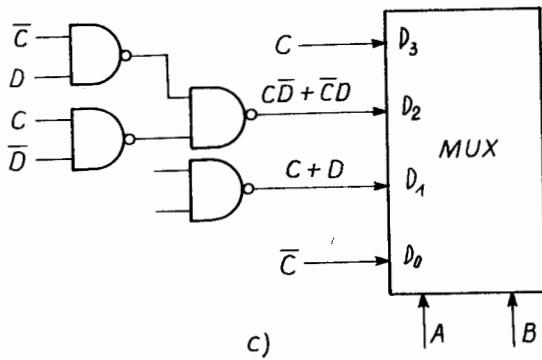
Chọn các biến điều khiển là A, B, C : Hàm đầu vào dữ liệu được xác định theo các ô như hình 12-11,a Bảng KARNAUGH của hàm f mô tả ở hình 12-11,b. Từ đó xác định được các hàm đầu vào dữ liệu như sau :

f	AB	CD	00	01	11	10
			D_0			
				D_1		
					D_3	
						D_2

a)

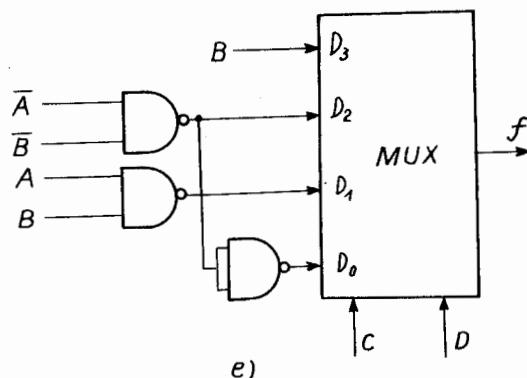
f	AB	CD	00	01	11	10
	00		1	1		
	01			1	1	1
	11				1	1
	10			1		1

b)



f	AB	CD	00	01	11	10	
f			D_3	D_2	D_1	D_0	
				D_0	D_1	D_3	D_2

d)

**Hình 12-10.** Tạo hàm f dùng MUX $4 \Rightarrow 1$.

- a. Các đường dữ liệu vào với biến điều khiển là AB ; b. Bảng Karnaugh của hàm $f = \sum 0, 1, 5, 6, 7, 9, 10, 14, 15$; c. Thực hiện hàm bằng MUX 4 đầu vào với 2 biến điều khiển là AB ; d. Các đường dữ liệu vào với biến điều khiển là CD ; e. Thực hiện hàm bằng MUX 4 đầu vào và các NANDs với 2 biến điều khiển là AB .

Lập được sơ đồ thực hiện mạch như hình 12-11,c.

Ví dụ 3 : Thực hiện hàm 5 biến sau dùng bộ dồn kênh 4 đầu vào (MUX $4 \Rightarrow 1$):

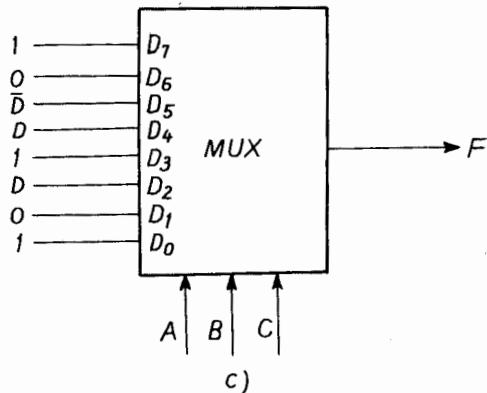
$$f = \sum 0, 1, 3, 6, 7, 8, 14, 15, 17, 18, 20, 21, 22, 24, 27, 28, 31.$$

	AB	CD	00	01	11	10
00	D_0			D_1		
01	D_2			D_3		
11	D_6			D_7		
10	D_4			D_5		

a)

	AB	CD	00	01	11	10
00			1	1		
01				1	1	1
11					1	1
10				1		1

b)

**Hình 12-11.** Tạo hàm f dùng $MUX\ 8 \Rightarrow 1$

- a. Các đường dữ liệu vào với biến điều khiển là $A\ B\ C$; b. Bảng Karnaugh của hàm $f = \sum 0, 1, 5, 6; 7, 9, 10, 14, 15$; c.Thực hiện hàm bằng $MUX\ 8$ đầu vào.

Cho bộ dòn kênh ở mức 1: chọn hai biến điều khiển là D và E . Biểu diễn f ban đầu trên bảng hình 12-12,a. Từ bảng này ta xác định được các hàm $D_{31}, D_{21}, D_{11}, D_{01}$ như sau:

$$D_{01} = \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + A \bar{B} C + A B \bar{C} + A B C$$

$$D_{11} = \bar{A} \bar{B} \bar{C} + A \bar{B} \bar{C} + A \bar{B} C$$

$$D_{21} = \bar{A} \bar{B} C + \bar{A} B C + A \bar{B} \bar{C} + A \bar{B} C$$

$$D_{31} = \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + \bar{A} B C + A B \bar{C} + A B C$$

Các hàm 3 biến này có thể tạo ra bằng bộ dòn kênh 4 đầu vào mà nó được coi như ở mức 2 của việc dòn kênh. Ở mức 2 này ta chọn B và C làm biến điều khiển như bảng Karnaugh hình 12-12,b. Theo bảng ta có các đầu vào dữ liệu đến các bộ dòn kênh ở mức 2 là :

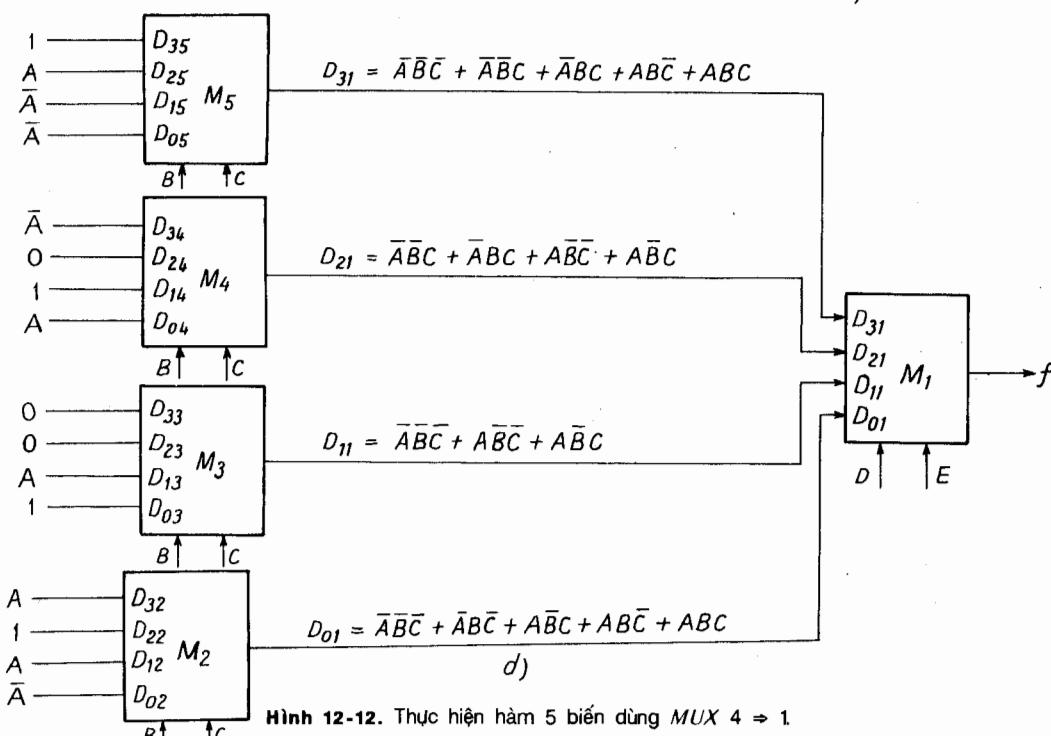
$$\begin{array}{llll} D_{01} = \bar{A} & D_{12} = A & D_{22} = 1 & D_{32} = A \\ D_{03} = 1 & D_{13} = A & D_{23} = 0 & D_{33} = 0 \\ D_{04} = A & D_{14} = 1 & D_{24} = 0 & D_{34} = \bar{A} \\ D_{05} = \bar{A} & D_{15} = \bar{A} & D_{25} = A & D_{35} = 1 \end{array}$$

Sơ đồ mạch tạo hàm f dùng 5 MUX $4 \Rightarrow 1$ cho trong hình 12-12,d.

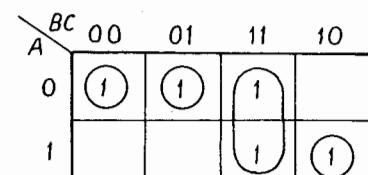
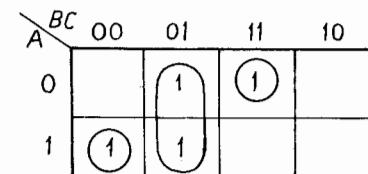
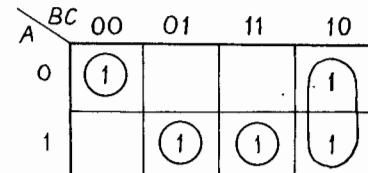
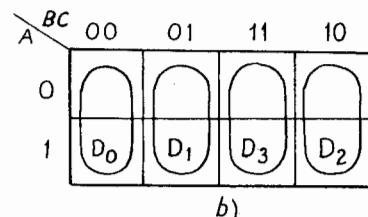
$f(A, B, C, D, E)$	$\bar{D} \cdot \bar{E}$	$\bar{D} \cdot E$	$D \cdot \bar{E}$	$D \cdot E$
$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \text{ (0)}$	$\bar{A} \cdot \bar{B} \cdot \bar{C}$			
$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D \cdot \bar{E} \text{ (1)}$		$\bar{A} \cdot \bar{B} \cdot \bar{C}$		
$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D \cdot E \text{ (3)}$				$\bar{A} \cdot \bar{B} \cdot \bar{C}$
$\bar{A} \cdot \bar{B} \cdot C \cdot \bar{D} \cdot \bar{E} \text{ (6)}$			$\bar{A} \cdot \bar{B} \cdot C$	
$\bar{A} \cdot \bar{B} \cdot C \cdot D \cdot \bar{E} \text{ (7)}$				$\bar{A} \cdot \bar{B} \cdot C$
$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D \cdot E \text{ (8)}$	$\bar{A} \cdot \bar{B} \cdot \bar{C}$			
$\bar{A} \cdot \bar{B} \cdot C \cdot D \cdot \bar{E} \text{ (14)}$		$A \cdot \bar{B} \cdot \bar{C}$	$\bar{A} \cdot B \cdot C$	
$\bar{A} \cdot \bar{B} \cdot C \cdot D \cdot E \text{ (15)}$				$\bar{A} \cdot B \cdot C$
$A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot E \text{ (17)}$		$A \cdot \bar{B} \cdot \bar{C}$	$A \cdot \bar{B} \cdot \bar{C}$	
$A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \text{ (18)}$	$A \cdot \bar{B} \cdot C$			
$A \cdot \bar{B} \cdot C \cdot \bar{D} \cdot \bar{E} \text{ (20)}$		$\bar{A} \cdot \bar{B} \cdot \bar{C}$	$A \cdot \bar{B} \cdot \bar{C}$	
$A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot E \text{ (21)}$			$A \cdot \bar{B} \cdot C$	
$A \cdot \bar{B} \cdot C \cdot \bar{D} \cdot \bar{E} \text{ (22)}$		$\bar{A} \cdot \bar{B} \cdot \bar{C}$		
$A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \text{ (24)}$	$A \cdot \bar{B} \cdot \bar{C}$			
$A \cdot B \cdot \bar{C} \cdot D \cdot E \text{ (27)}$				$A \cdot B \cdot \bar{C}$
$A \cdot B \cdot C \cdot \bar{D} \cdot \bar{E} \text{ (28)}$	$A \cdot B \cdot C$			
$A \cdot B \cdot C \cdot D \cdot E \text{ (31)}$			$A \cdot B \cdot C$	

a)

c)

Hình 12-12. Thực hiện hàm 5 biến dùng MUX 4 \Rightarrow 1.

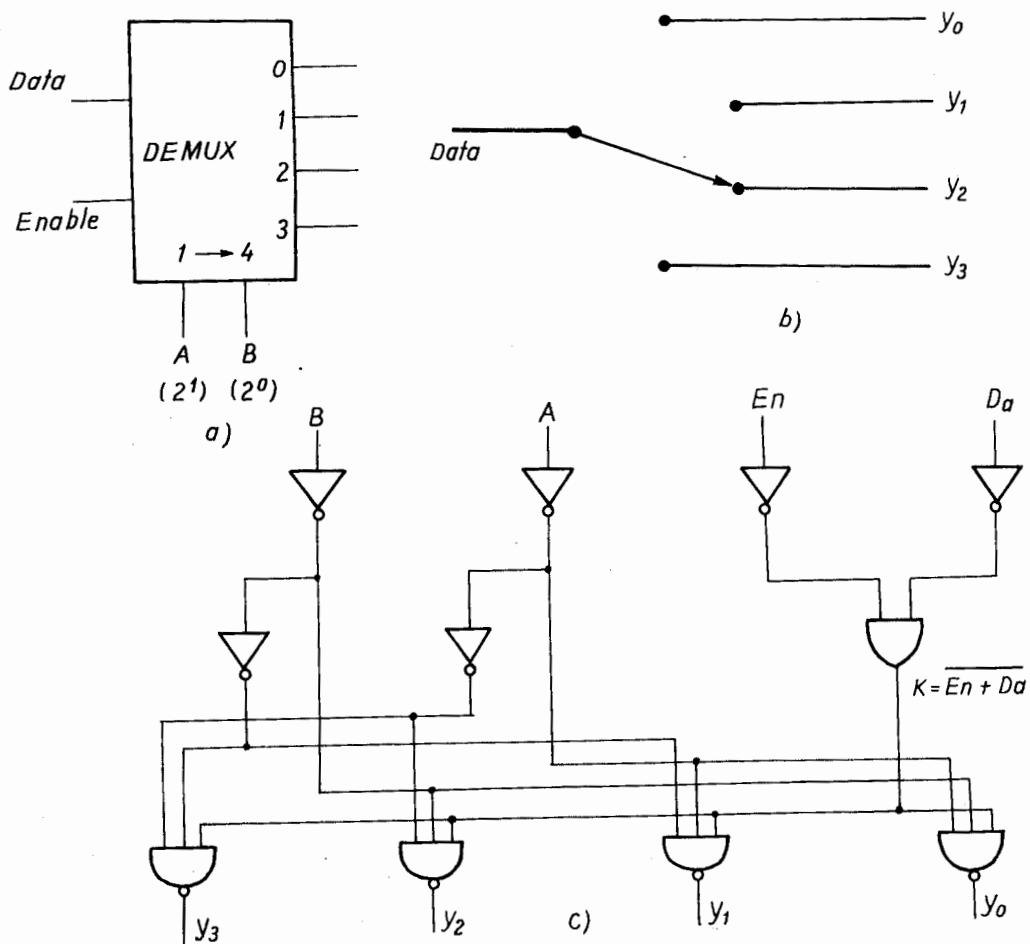
- a. Xác định đầu vào của MUX ở mức 1; b. Các đường dữ liệu vào với biến điều khiển B và C ;
c. Bảng Karnaugh xác định dữ liệu vào cho 4 MUX ở mức 2; d. Thực hiện mạch.



12_4. BỘ PHÂN KÊNH DEMUX VÀ BỘ GIẢI MÃ (DECODER)

Bộ phân kênh thực hiện chức năng ngược với bộ dồn kênh. Một bộ phân kênh tổng quát sẽ có : 1 đầu vào, n đầu vào điều khiển, 2^n đầu ra và một đầu vào chọn mạch En (Enable). Đầu ra thứ i khi giá trị thập phân của n đầu vào điều khiển bằng i .

Ví dụ : bộ phân kênh 1 đầu vào và 4 đầu ra($DEMUC 1 \Rightarrow 4$) có sơ đồ như hình 12-13,a. Hình 12-13,b mô tả nguyên lý làm việc của nó. Sơ đồ thực hiện $DEMUC 1 \Rightarrow 4$ từ các mạch tổ hợp cho trong hình 12-13,c.



Hình 12- 13. Mô tả bộ phân kênh $DEMUC 1 \Rightarrow 4$.

a. Sơ đồ khối; b. Sơ đồ mô tả nguyên lý làm việc; c. Sơ đồ thực hiện $DEMUC 1 \Rightarrow 4$ từ các mạch tổ hợp.

Khi $En = 0, Da = 0, K = \overline{En + Da} = 1$

$DEMUX$ hoạt động như một bộ giải mã $2 \Rightarrow 4$ với hai đầu vào *A*, *B* 4 đầu ra Y_0, Y_1, Y_2, Y_3 . Các đầu ra ở sơ đồ này có mức tích cực thấp. Trong các trường hợp khác tổ hợp biến *En*, *Da* thì $K = 0$. Các đầu ra luôn luôn ở mức cao.

Các bộ giải mã $DECODER$ và các bộ phân kênh $DEMUX$ khi làm việc như bộ giải mã có đặc điểm : mỗi đầu ra là 1 tích đầy đủ n biến ở đầu vào.

12_5. ỨNG DỤNG BỘ GIẢI MÃ

12_5.1. BỘ GIẢI MÃ DÙNG ĐỂ LÀM TÍN HIỆU CHỌN VỎ (CHIP SELECTOR - CS)

Bộ giải mã ngoài chức năng giải mã như tên gọi, nó còn được dùng để tạo tín hiệu chọn vỏ trong các mạch số nối chung và cho các vi mạch nhớ trong CPU (đơn vị xử lý trung tâm).

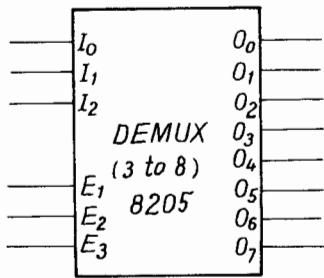
Ví dụ bộ giải mã Intel 8205 được dùng để chọn bộ nhớ cho bộ vi xử lý Intel 8085.

Bảng chức năng của 8205 cho trong hình 12-14,a

Sơ đồ khối của nó được mô tả trong hình 12-14,b

Ở đây 3 đầu vào cho phép làm việc của bộ giải mã Intel 8205 có mức tích cực sau :

Cách măc bộ giải mã Intel 8205 với bộ vi xử lý và các bộ nhớ như hình 12-14,c.

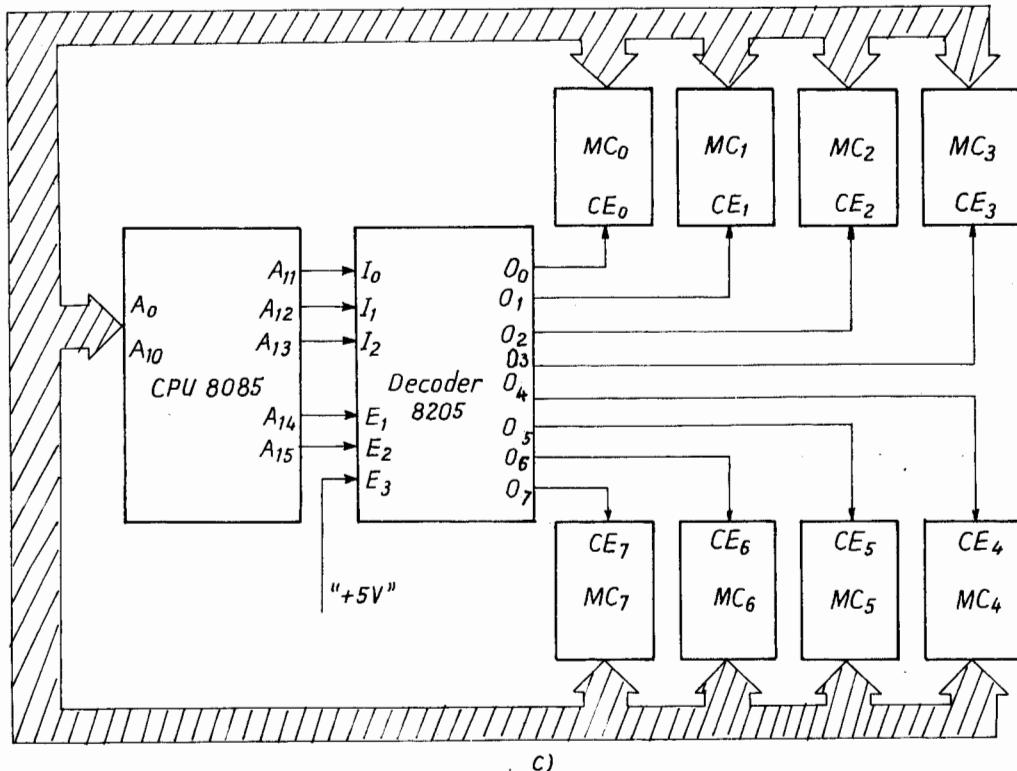


E₁, E₂ có mức tích cực thấp
E₃ có mức tích cực cao

a)

Địa chỉ			Cho phép			Đầu ra							
I ₂	I ₁	I ₀	E ₁	E ₂	E ₃	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇
0	0	0	0	0	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	1	0	0	0	1	1	1	0	1	1	1	1	1
0	1	1	0	0	1	1	1	1	0	1	1	1	1
1	0	0	0	0	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	1	1	1	0	1	1
1	1	0	0	0	1	1	1	1	1	1	1	0	1
1	1	1	0	0	1	1	1	1	1	1	1	1	0

b)



c)

Hình 12-14. Bộ giải mã Intel và ứng dụng của nó.

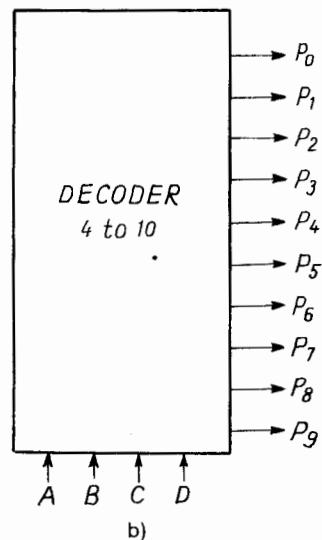
a. Sơ đồ khối; b. Bảng chức năng; c. Hệ thống chọn bộ nhớ cho vi xử lý 8085.

12_5.2. DÙNG BỘ GIẢI MÃ VÀ MỘT SỐ HÀM PHỤ (NẾU CẦN) ĐỂ TẠO HÀM SỐ CHO TRƯỚC

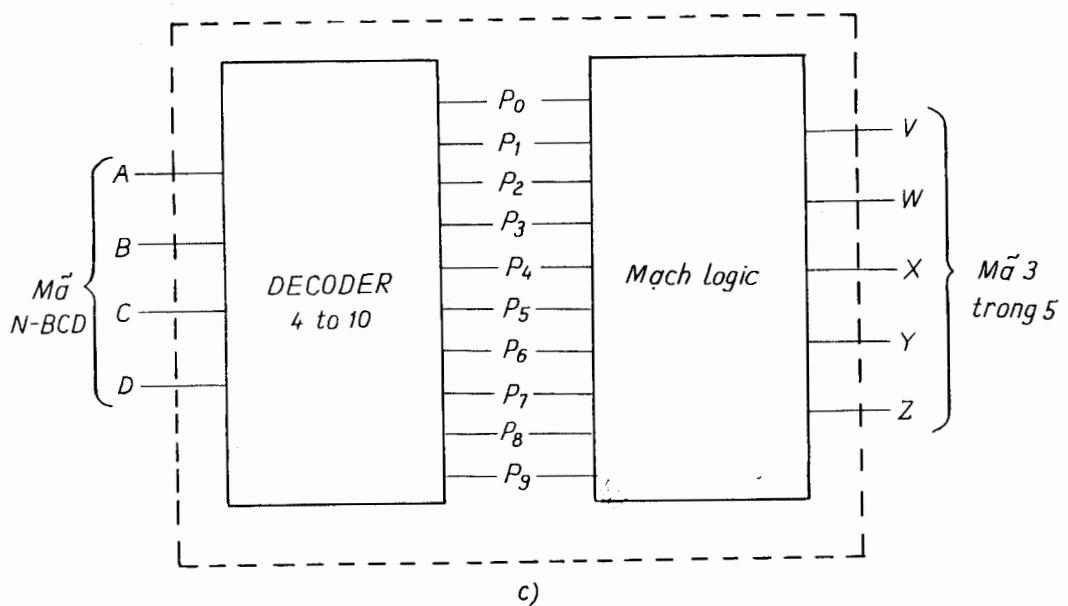
Như đã nhận xét ở trên, một bộ *DEMUC* khi làm việc như một bộ giải mã và một bộ giả mã luôn luôn có đặc điểm : mỗi đầu ra là một tích (hội) đầy đủ của tất cả các biến đầu vào. Mặt khác một hàm boolean bất kỳ luôn có thể biểu diễn dưới dạng CTT đầy đủ nghĩa là tuyển (hoặc – OR) của nhiều thành phần, mỗi thành phần là một tích (hội – AND) đầy đủ của các biến đầu vào. Do vậy muốn thực hiện một hàm Boolean cho trước ta chỉ cần xem hàm này là tuyển của những hội nào (tương đương đầu ra nào của bộ giải mã) rồi tuyển các thành phần ấy lại nhờ những mạch OR.

Phần tử P	NBCD				Mã 3 trong 5				
	A	B	C	D	V	W	X	Y	Z
P ₀	0	0	0	0	1	0	0	1	1
P ₁	0	0	0	1	1	0	1	0	1
P ₂	0	0	1	0	1	1	0	0	1
P ₃	0	0	1	1	0	0	1	1	1
P ₄	0	1	0	0	0	1	0	1	1
P ₅	0	1	0	1	0	1	1	0	1
P ₆	0	1	1	0	0	1	1	1	0
P ₇	0	1	1	1	1	0	1	1	0
P ₈	1	0	0	0	1	1	0	1	0
P ₉	1	0	0	1	1	1	1	0	0

a)



b)



Hình 12-15. Dùng bộ giải mã để tạo hàm số.

- a. Bảng chân lý của mạch chuyển mã NBCD;
- b. Sơ đồ khối của bộ giải mã 4 \Rightarrow 10;
- c. Sơ đồ khối mạch phải thiết kế.

Ví dụ : Dùng bộ giải mã $4 \Rightarrow 10$ và các mạch *NAND* để xây dựng mạch chuyển mã NBCD ở đầu vào thành mã 3 trong 5. Bảng cho ở hình 12-15,a, sơ đồ khối ở hình 12-15,b.

Vấn đề phải thiết kế mạch logic (hình 12-15,c).

Giả sử phải xây dựng mạch logic này chỉ dùng *NAND*. Theo bảng mã ta có :

$$V = P_0 + P_1 + P_2 + P_7 + P_8 + P_9$$

$$W = P_2 + P_4 + P_5 + P_6 + P_8 + P_9$$

$$X = P_1 + P_3 + P_5 + P_6 + P_7 + P_9$$

$$Y = P_0 + P_3 + P_4 + P_6 + P_7 + P_8$$

$$Z = P_0 + P_1 + P_2 + P_3 + P_4 + P_5$$

Như vậy

$$\bar{V} = P_3 + P_4 + P_5 + P_6$$

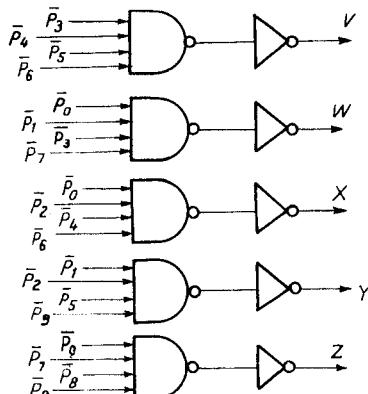
$$V = P_3 + P_4 + P_5 + P_6 \quad V = \bar{P}_3 \bar{P}_4 \bar{P}_5 \bar{P}_6$$

Tương tự :

$$\bar{W} = \bar{P}_0 \bar{P}_1 \bar{P}_3 \bar{P}_7 \quad \bar{Y} = \bar{P}_1 \bar{P}_2 \bar{P}_5 \bar{P}_9$$

$$\bar{X} = \bar{P}_0 \bar{P}_2 \bar{P}_4 \bar{P}_6 \quad \bar{Z} = \bar{P}_0 \bar{P}_7 \bar{P}_8 \bar{P}_9$$

Sơ đồ của phần mạch logic thêm vào cùng với bộ giải mã $4-10$ để chuyển mã NBCD như hình 12-15,d.



Hình 12-15. d) Sơ đồ phần mạch logic thêm vào dùng các mạch *NAND*.

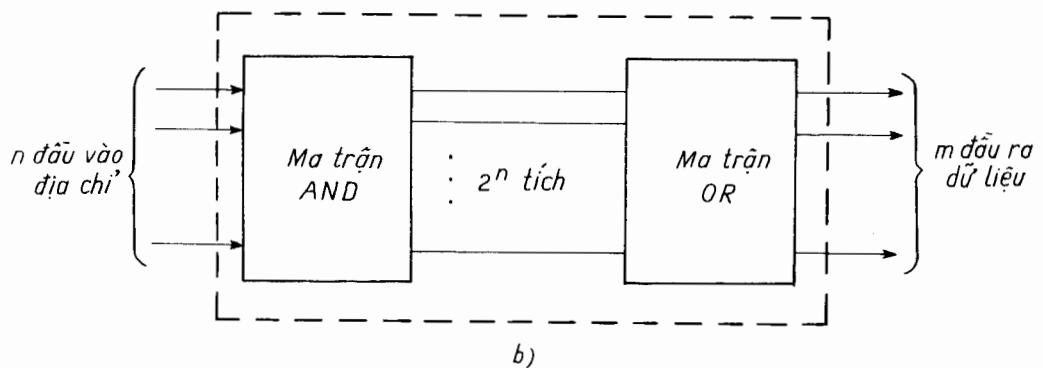
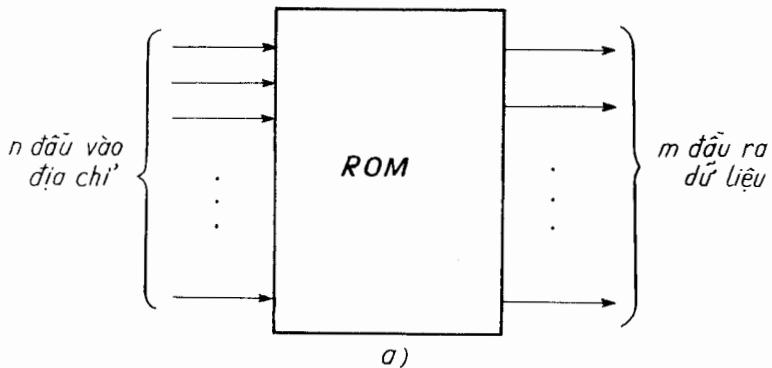
12-6. ROM (READ ONLY MEMORY - BỘ NHỚ CHỈ ĐỌC)

ROM như tên gọi của nó là bộ nhớ chỉ đọc. Sơ đồ khối của *ROM* cho trong hình 12-16,a. Cấu tạo bên trong của *ROM* gồm 2 mảng trật : Ma trận AND và ma trận OR như hình 12-16,b.

Tính chất chương trình của *ROM* thể hiện ở chỗ : Trong ma trận OR mỗi đầu ra là tuyển OR của các tích nào.

Ví dụ : Rom 3 đầu vào và 8 đầu ra (A, B, C và $Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8$) có sơ đồ nguyên lý của *ROM* cho trong hình 12-17,a. Ma trận nối và các hàm đầu vào, đầu ra của ma trận OR cho trong hình 12-17,b. Bảng chức năng của *ROM* cho trong hình 12-17,c.

Chú ý : E là đầu vào chọn mạch (cho phép hay không cho phép làm việc) có mức logic tích cực là mức thấp.



Hình 12-16.a) Sơ đồ khối của ROM ; b) Cấu tạo bên trong của ROM

12-7. CÁC ỨNG DỤNG CỦA ROM

1. Dùng làm bộ nhớ trong MT : ROM đúng như tên gọi của nó thường được dùng làm bộ nhớ trong máy tính. Ngoài ra ROM còn được dùng để thiết kế các mạch số tổ hợp và dây.

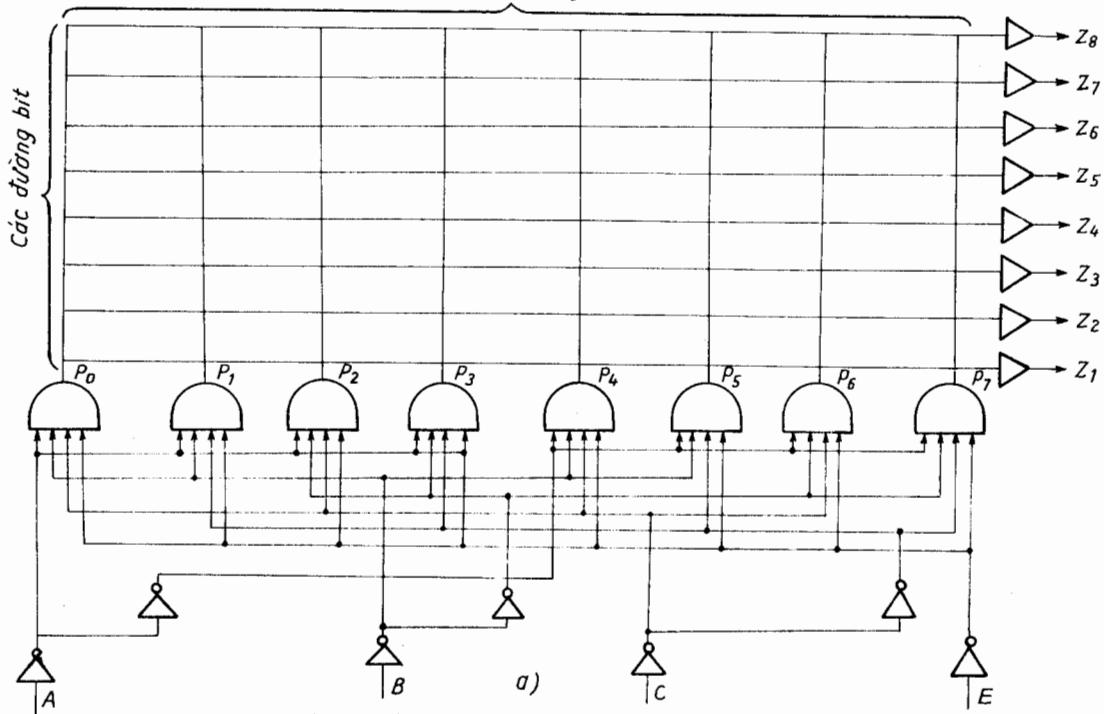
2. Dùng để thiết kế các mạch tổ hợp

Mạch tổ hợp có sơ đồ khối như hình 12-18.

Khi dùng ROM để thiết kế mạch tổ hợp ta coi X như địa chỉ vào của ROM và Y như các đầu ra số liệu của ROM. ROM được sử dụng như một bảng tra cứu (theo đúng bảng chân lý của mạch tổ hợp).

Ví dụ : Dùng ROM để xây dựng bộ giải mã nhị phân sang mã 7 vạch. Bảng chuyển đổi mã như hình 12-19,a. Các đầu vào được xem là đầu địa chỉ của ROM, các đầu ra 7 vạch

Các đường từ



$P_0 = \bar{A}\bar{B}\bar{C}$	$Z_8 = P_0 + P_5 + P_7$
$P_1 = \bar{A}\bar{B}C$	$Z_7 = P_2 + P_5$
$P_2 = \bar{A}BC$	$Z_6 = P_5$
$P_3 = \bar{A}BC$	$Z_5 = P_0 + P_2 + P_3 + P_4 + P_7$
$P_4 = A\bar{B}\bar{C}$	$Z_4 = P_0 + P_1 + P_5 + P_7$
$P_5 = A\bar{B}C$	$Z_3 = P_2 + P_3 + P_4 + P_6 + P_7$
$P_6 = AB\bar{C}$	$Z_2 = P_0 + P_7$
$P_7 = ABC$	$Z_1 = P_0 + P_2 + P_3 + P_4$

b)

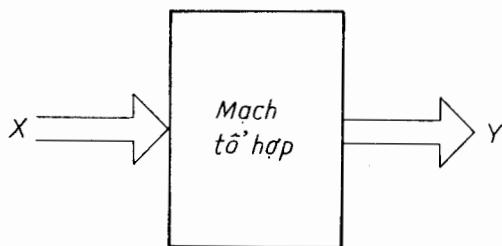
Địa chỉ			Đầu ra								
A	B	C	Z ₁	Z ₂	Z ₃	Z ₄	Z ₅	Z ₆	Z ₇	Z ₈	
0	0	0	1	1	0	1	1	0	0	1	
0	0	1	0	0	0	1	0	0	0	0	
0	1	0	1	0	1	0	1	0	1	0	
0	1	1	1	0	1	0	1	0	0	0	
1	0	0	1	0	1	0	1	0	0	0	
1	0	1	0	0	0	1	0	1	1	1	
1	1	0	0	0	1	0	0	0	0	0	
1	1	1	0	1	1	1	1	0	0	1	

c)

Hình 12-17. Ví dụ một ROM 3 đầu vào, 8 đầu ra.

- a. Sơ đồ nguyên lý và ma trận AND; b. Ma trận nối và các hàm đầu vào, đầu ra của ma trận OR; c. Bảng chân lý của ROM.

chính là đầu ra của *ROM*. Để biểu thị các đầu ra này được đưa vào mạch hiển thị (Led 7 thanh) như ở hình 12-19,b. Từ đó nhận được ma trận nối (ma trận OR) như ở hình 12-19,c.



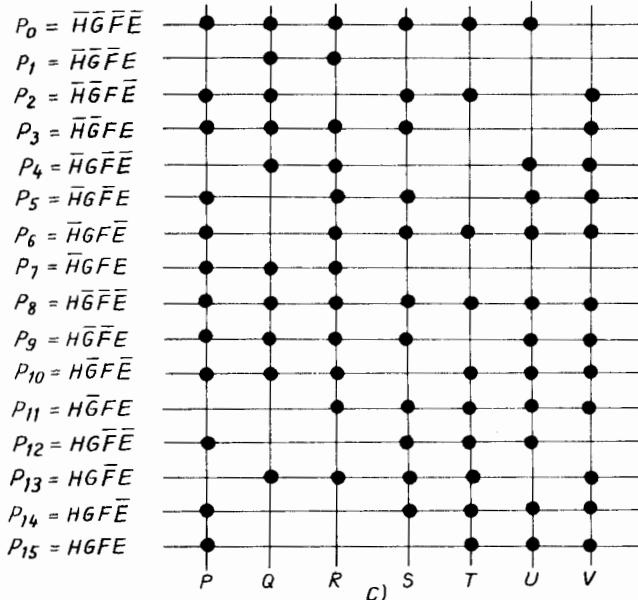
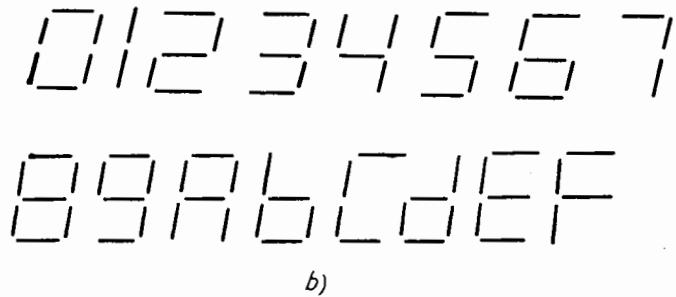
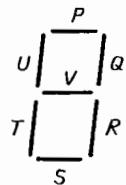
Hình 12-18. Sơ đồ khối của mạch tổ hợp.

3. Dùng *ROM* để thiết kế các mạch dãy đồng bộ và không đồng bộ : Sơ đồ khối của mạch dãy đồng bộ dùng *ROM* cho trong hình 12-20,a. Sơ đồ khối của mạch dãy đồng bộ dùng *ROM* cho trong hình 12-20,b.

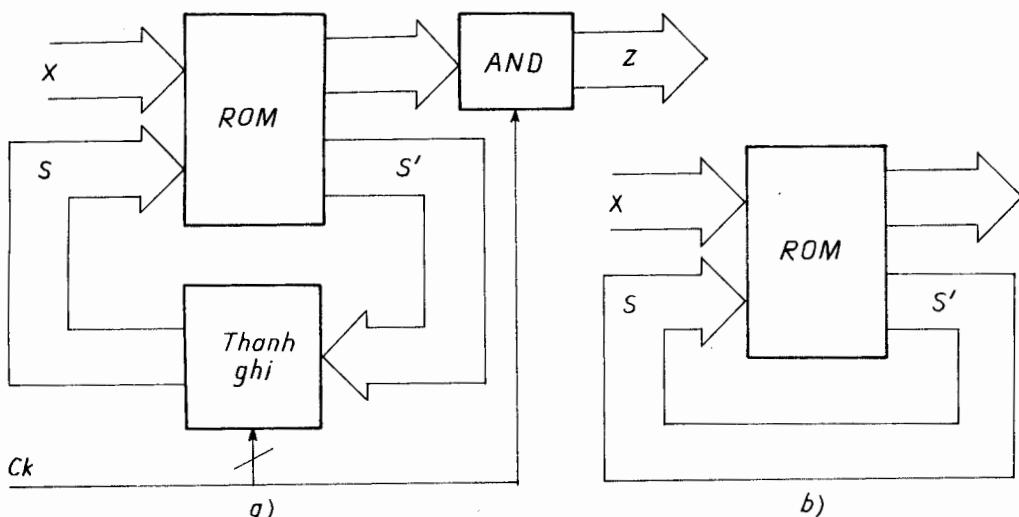
Mã 16	Vàos nhị phân				Ra 7 vạch						
	H	G	F	E	P	Q	R	S	T	U	V
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	1	1	1	0	1	1	1
B	1	0	1	1	0	0	1	1	1	1	1
C	1	1	0	0	1	0	0	1	1	1	0
D	1	1	0	1	0	1	1	1	1	0	1
E	1	1	1	0	1	0	0	1	1	1	1
F	1	1	1	1	1	0	0	0	1	1	1

Hình 12-19,a

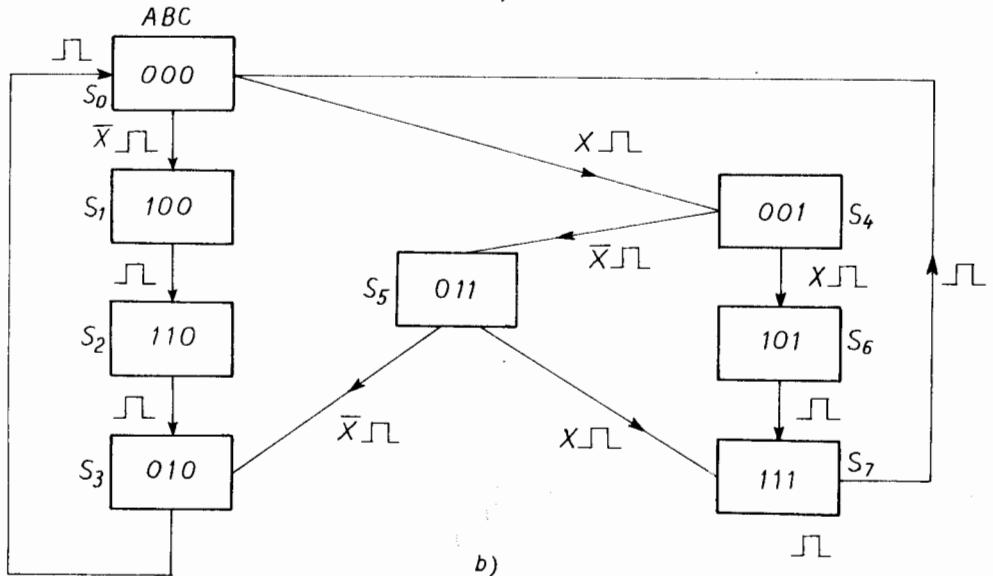
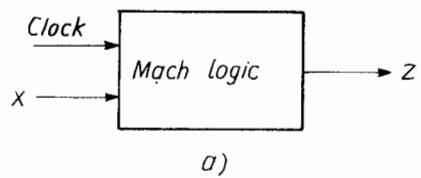
Ví dụ : Thiết kế mạch phát hiện mã *NBCD* ở đầu vào là sai. dùng *ROM*, các *FF* loại *JK-FF* và mạch *NAND*. Mã *NBCD* được đưa vào liên tiếp ở đầu vào , bit đầu tiên là bit có trọng số lớn nhất. Cứ đến bit thứ 4 nếu mã là sai, có tín hiệu ra *Z* = 1. Sơ đồ khối của mạch này được mô tả như hình 12-21,a. Căn cứ vào yêu cầu của đầu bài lập được đồ hình chuyển động trạng thái của mạch như hình 12-21,b. Mã hoá nhị phân các trạng thái của mạch bằng



Hình 12-19. Dùng ROM để xây dựng bộ giải mã 7 vạch.
a. Bảng mã; b. Led 7 thanh; c. Ma trận nối (ma trận OR).



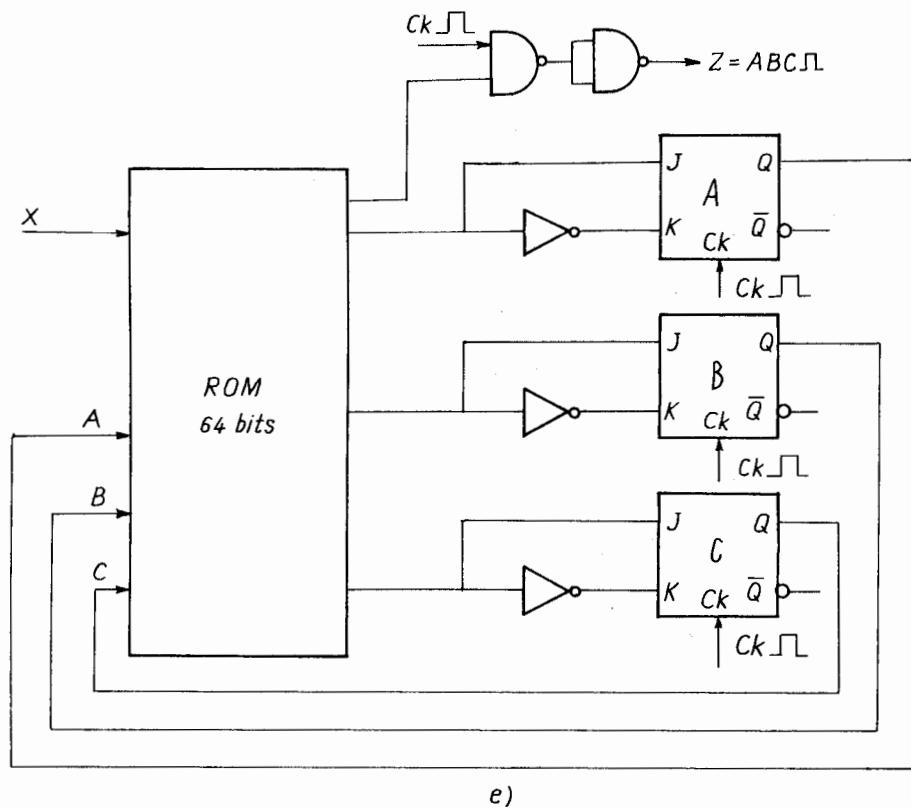
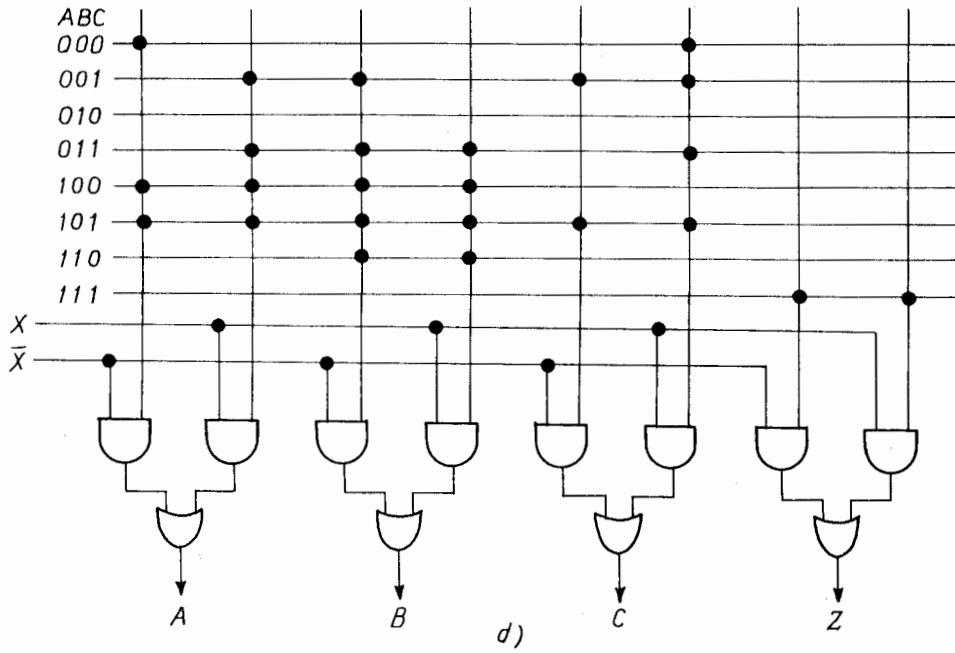
Hình 12-20. Sơ đồ khối mạch dãy đồng bộ dùng ROM;
a. Mạch dãy đồng bộ; b. Mạch dãy không đồng bộ.



A	B	C	X	A'	B'	C'	Z
0	0	0	0	1	0	0	0
0	0	0	1	0	0	1	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	0
1	1	0	0	0	1	0	0
1	1	0	1	0	1	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	1	1	0
1	0	1	0	1	1	1	0
1	0	1	1	1	1	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	1

c)

Hình 12-21a, b, c

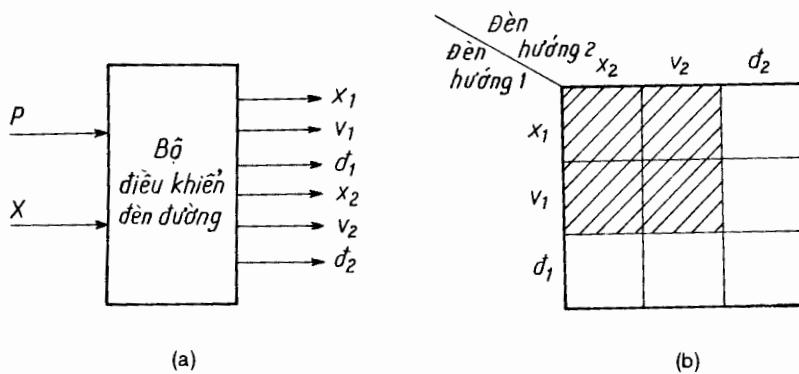


Hình 12-21. Thiết kế mạch phát hiện mã *NBCD* ở đầu vào là sai dùng *ROM*.

- a. Sơ đồ khối của mạch; b. Đồ hình chuyển đổi trạng thái của mạch; c. Bảng chuyển đổi trạng thái và tín hiệu ra; d. Ma trận nối; e. Sơ đồ mạch thực hiện dùng *ROM* và *JK-FF* và *NAND*.

3 biểu nhị phân A B C xem hình 12-21,b. Sau đó ta lập được bảng chuyển đổi trạng thái với các tín hiệu vào tương ứng và tín hiệu ra (Z) như hình 12-21,c. Từ bảng đó ta đưa ra được ma trận nối của ROM như hình vẽ 12-21,d. Mạch thực hiện dùng ROM , các $JK-FF$ và mạch $NAND$ như hình 12-21-e.

Ví dụ 2 : Thiết kế mạch dãy không đồng bộ dùng để điều khiển đèn giao thông ở một ngã tư đường phố dùng ROM (hình 12-22). Sơ đồ khối của bộ điều khiển như hình 12-22,a Trong đó các ký hiệu x_1, v_1, d_1 là các đèn xanh, vàng, đỏ của đường thứ nhất; x_2, v_2, d_2 là các đèn xanh, vàng, đỏ của đường thứ hai.



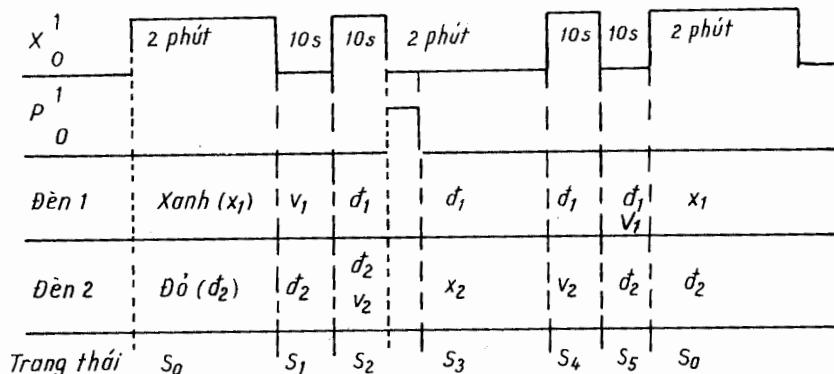
Hình 12-22.a) Sơ đồ khối của bộ điều khiển đèn đường;

b) Các tổ hợp trạng thái đèn của hai tuyến đường.

Các tổ hợp trạng thái của đèn được mô tả ở hình 12-22,b, trong đó những ô bị gạch tương ứng với các trường hợp cấm để bảo đảm điều kiện an toàn. Ví dụ: cả hai trực đường đều đèn xanh x_1, x_2 là một tổ hợp cấm, ...

Giả thiết rằng hoạt động của các đèn như hình 12-22,c lúc đó ta lập được đồ hình chuyển đổi trạng thái hình 12-22,d.

Mã hoá nhị phân của các trạng thái như hình 12-22,d ta lập được bảng chuyển đổi trạng thái và các tín hiệu ra (thể hiện bằng các đèn) ứng với các tín hiệu vào (X, P) như hình 12-22,e. Từ đó xây dựng được ma trận nối của ROM như hình 12-23,

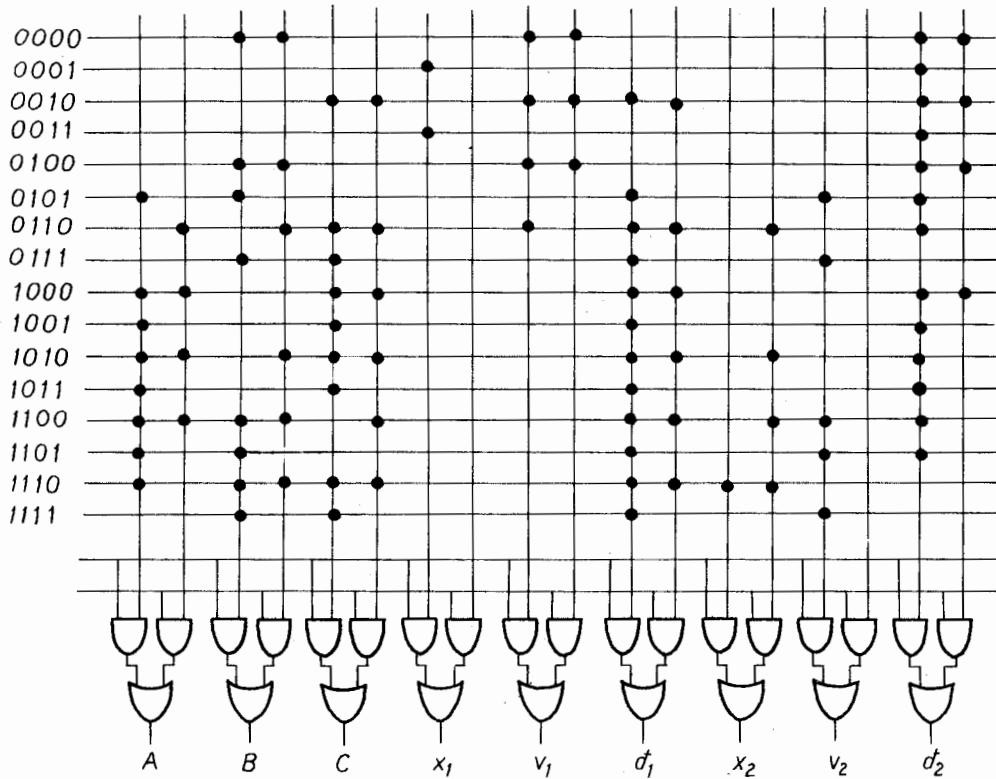


Hình 12-22.c

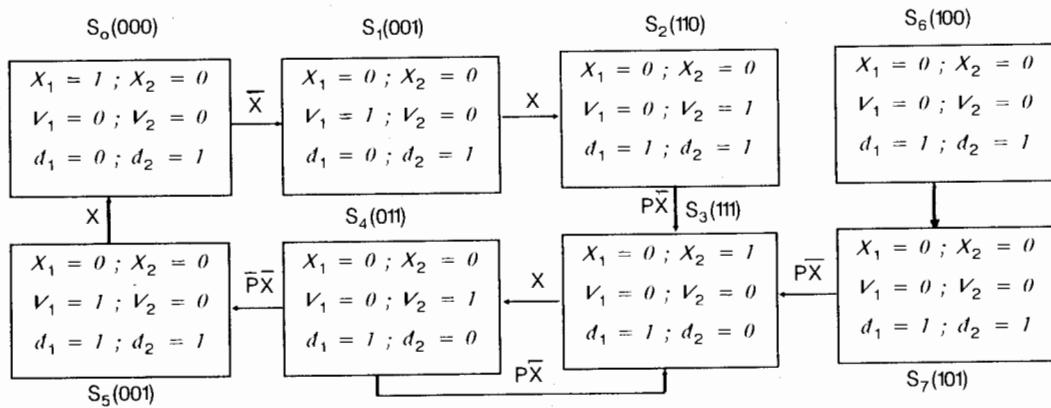
Trạng thái hiện tại					Đầu ra									
A	B	C	X	P	A	B	C	X_1	V_1	d_1	X_2	V_2	d_2	
0	0	0	0	0	0	1	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0	1	0	0	0	0	1
0	1	0	0	1	0	1	0	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	0	1	1	1
1	1	0	0	0	1	1	1	0	0	0	1	0	1	1
1	1	0	0	1	1	1	1	0	0	0	1	1	0	0
1	1	0	1	0	1	1	1	0	0	0	1	1	0	1
1	1	1	0	0	1	1	1	1	0	0	1	1	0	0
1	1	1	0	1	1	1	1	1	0	0	1	1	0	0
1	1	1	1	0	0	1	1	1	0	0	1	0	1	0
0	1	1	1	0	0	0	0	1	0	1	1	0	0	1
0	1	1	1	0	1	1	1	0	0	1	1	0	0	0
0	1	1	1	1	0	0	1	1	0	0	1	0	1	0
0	0	1	0	0	0	0	0	1	0	1	1	0	0	1
0	0	1	0	1	0	0	0	0	1	0	0	0	0	1
1	0	0	0	0	1	0	0	1	0	0	1	0	0	1
1	0	0	0	1	0	1	0	1	0	0	0	1	0	1
1	0	0	1	0	0	1	0	1	0	0	1	0	0	1
1	0	0	1	0	1	1	1	0	0	0	1	1	0	0
1	0	0	1	1	0	1	1	0	0	0	1	1	0	0
1	0	1	0	0	1	0	0	1	0	0	1	0	0	1
1	0	1	0	1	1	1	1	0	0	0	1	1	0	0
1	0	1	1	0	0	1	0	1	0	0	1	0	1	0
1	0	1	1	0	1	0	1	0	0	0	1	0	0	1
1	0	1	1	1	0	0	0	0	1	0	0	0	0	1
1	1	0	0	0	1	0	0	1	0	0	1	0	0	1
1	1	0	0	1	0	1	0	1	0	0	1	0	0	1
1	1	0	1	0	0	1	0	0	1	0	0	1	0	0
1	1	0	1	1	0	1	0	0	1	0	0	1	0	0
1	1	1	0	0	1	0	1	1	0	0	1	0	1	0
1	1	1	0	1	0	1	1	0	0	0	1	0	1	0
1	1	1	1	0	0	0	1	0	1	1	0	0	1	0
1	1	1	1	1	0	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	0	0	0	0	1	0	0	0	1
1	1	1	1	1	1	1	0	0	0	0	1	0	0	1
1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

e)

Hình 12-22.c) Giản đồ trạng thái của các đèn;
d) Đồ hình chuyển đổi trạng thái của các mạch; e) Bảng trạng thái;



Hình 12-23 Ma trận ROM.



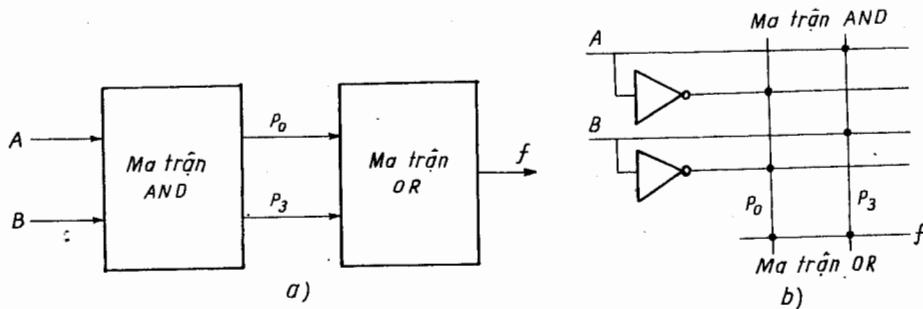
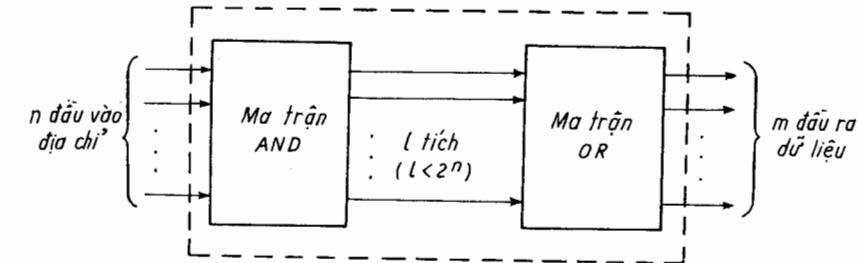
Hình 12-22.d

12-8. CÁC MÀNG LOGIC LẬP TRÌNH (PROGRAMMABLE LOGIC ARRAYS – PLA)

- PLA về mặt cấu tạo giống như ROM gồm 2 ma trận : ma trận AND, ma trận OR
- Sự khác nhau giữa PLA và ROM là ở chỗ : Đầu ra ma trận AND của PLA không đủ 2 tích như của ROM (cho n đầu vào) và mỗi tích có thể không đủ n biến. Sơ đồ tổng quát của PLA như hình 12-24, a.

PLA cho phép lập trình với cả 2 ma trận AND và OR (ROM chỉ cho phép lập trình với ma trận OR). Ví dụ : 1 PLA đơn giản cho ở hình 12-24. Sơ đồ khối của PLA này cho ở hình 12-24, b. Ma trận nối được mô tả ở hình 12-24, c.

Trong đó đầu ra của ma trận AND chỉ có 2 tích P_0 và P_3 (còn ở ROM có 4 tích là (P_0, P_1, P_2, P_3)).



Hình 12-24. Sơ đồ 1 PLA đơn giản.

a. Sơ đồ khối của PLA; b. Sơ đồ khối của PLA đơn giản; c. Ma trận nối

12-9. CÁC ỨNG DỤNG CỦA PLA

Tương tự như *ROM*, có thể dùng *PLA* để :

1. Tạo hàm Boolean (thiết kế mạch tổ hợp)

Ví dụ, dùng *PLA* để tạo các hàm 6 biến sau :

$$f_1 = A B C \bar{D} E \bar{F} + A E \bar{F} + A C D + \bar{B} \bar{D}$$

1 2 3 4

$$f_2 = A B \bar{E} \bar{F} + A \bar{C} \bar{D} + \bar{B} \bar{D} + E F$$

5 6 4 7

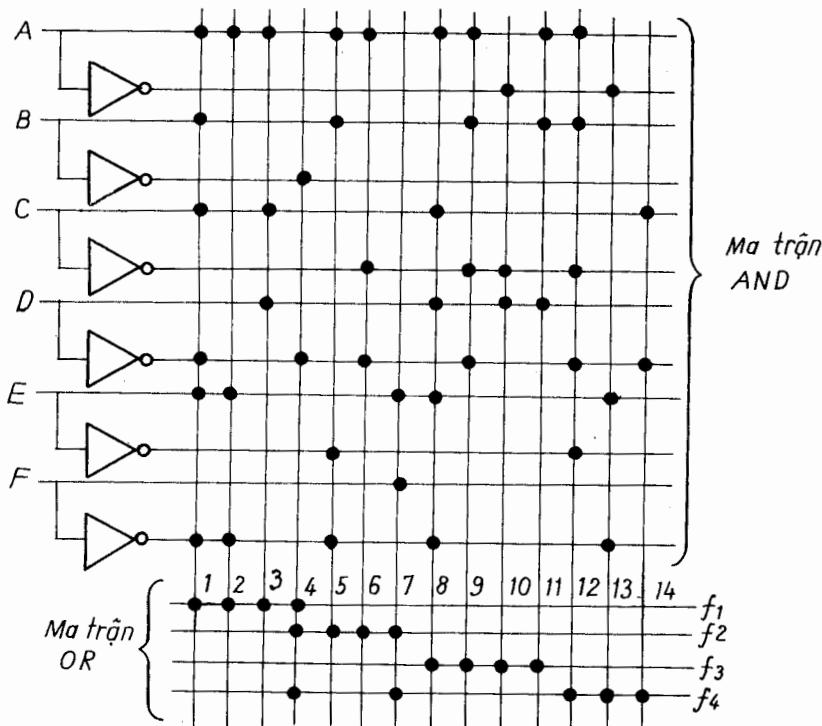
$$f_3 = A C D E \bar{F} + A B \bar{C} \bar{D} + \bar{A} \bar{C} D + A B D$$

8 9 10 11

$$f_4 = A B \bar{C} \bar{D} \bar{E} + \bar{A} E \bar{F} + \bar{B} \bar{D} + C \bar{D} + E F$$

12 13 4 14 7

Hệ phương trình trên gồm 14 tích riêng lẻ, các tích này hoặc là đầy đủ (gồm cả 6 biến) hoặc không đầy đủ. Ma trận AND phải có 12 đầu vào (tương ứng với 6 biến) và 14 đầu ra tương ứng với 14 tích. Ma trận OR phải có 14 đầu vào (14 tích) và 4 đầu ra tương ứng với 4 hàm số f_1, f_2, f_3 và f_4 . Sơ đồ *PLA* dùng để thực hiện 4 hàm trên như hình 12-25.



Hình 12-25. Sơ đồ *PLA* dùng để thực hiện các hàm 6 biến.

So sánh dung lượng của bộ nhớ khi dùng *PLA* và *ROM* nối chung dùng *PLA* luôn luôn tiết kiệm dung lượng nhớ hơn. Trong ví dụ của chúng ta ; Dung lượng của *PLA* bằng tổng dung lượng của ma trận AND và ma trận OR bằng $(12 \times 14) + (4 \times 14) = 224$ bit. Nếu tính dung lượng của *ROM* gồm ma trận AND và ma trận OR thì sẽ có $(12 \times 2^6) + (4 \times 2^6) = 1024$ bit gấp gần 5 lần dung lượng của *PLA*. Chỉ tính riêng dung lượng ma trận OR của *ROM* (Như cách tính dung lượng *ROM* trong catalog) thì đã có 256 bit, như vậy dùng *PLA* tiết kiệm 12,5%.

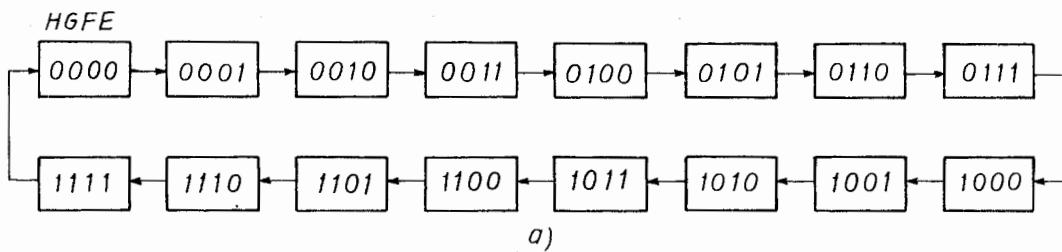
2) Thiết kế mạch dãy :

Cũng hoàn toàn tương tự như *ROM*, *PLA* có thể dùng để thiết kế mạch dãy. Trong thực tế, người ta đã chế tạo sẵn các *PLA* với kích thước của ma trận AND, Ma trận OR và một số *FF* loại nào đó cho trước. Người thiết kế tùy theo bài toán của mình mà chọn loại *PLA* thích hợp.

Ví dụ : thiết kế một bộ đếm 16 mã nhị phân có hiển thị 7 vạch dùng *PLA* và các *D-FF*.

Dồ hình trạng thái của bộ đếm này cho hình 12-26,a. Dùng 4 biến nhị phân để mã hoá : H G F E (E là cột có trọng số nhỏ nhất, H cột có trọng số lớn nhất)

Căn cứ vào bảng đầu mô kích cho *D-FF* hình 12-26,b chúng ta xác định được giá trị đầu vào *D* cho từng *FF* : H, G, F và E như hình 12-26,c. Xác định hàm đầu vào kích cho các *FF* đó và tối thiểu hóa các hàm đó theo bảng Karnaugh hình 12-26,d.



a)

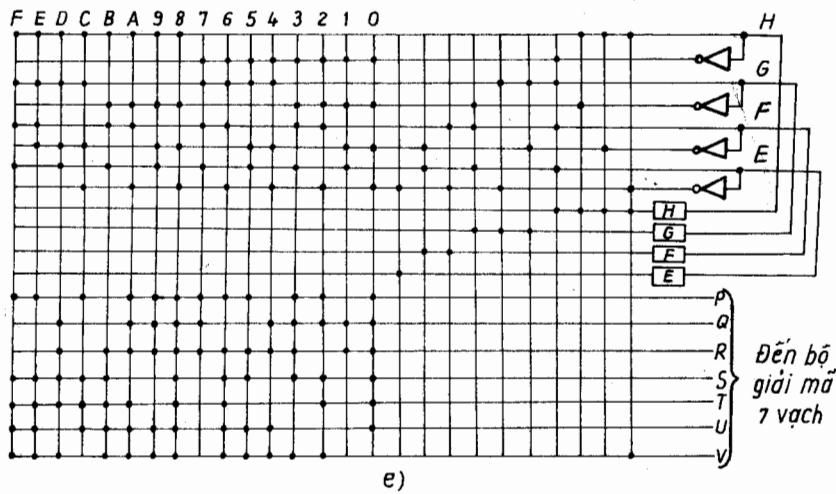
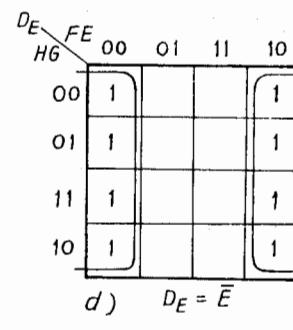
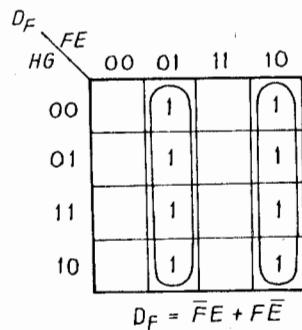
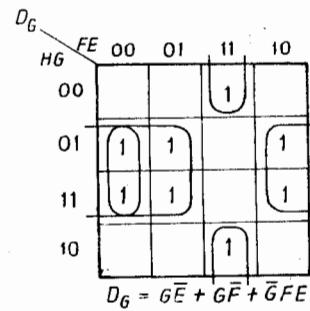
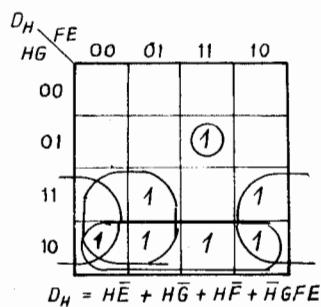
<i>Q</i>	<i>Q'</i>	<i>D</i>
0	0	0
0	1	1
1	0	0
1	1	1

b)

Trạng thái hiện tại								<i>Đầu vào Flip - Flop</i>			
<i>H</i>	<i>G</i>	<i>F</i>	<i>E</i>	<i>H'</i>	<i>G'</i>	<i>F'</i>	<i>E'</i>	<i>D_H</i>	<i>D_G</i>	<i>D_F</i>	<i>D_E</i>
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0

c)

Sơ đồ nối của PLA được chỉ ra trên hình 12-27,e.



Hình 12-26. a) Đồ hình chuyển động trạng thái của bộ đếm ; b) Bảng đầu vào kích cho D-FF ; c) Bảng chuyển đổi trạng thái và đầu vào kích cho FF ; d) Bảng Karnaugh của các đầu vào kích của FF ; e) Sơ đồ thực hiện bộ đếm 16 có hiển thị 7 vạch dùng PLA.

BÀI TẬP

12.1. Thực hiện các hàm Boole 5 biến sau dùng *MUX* 4 đầu vào

a) $f = \sum 0, 1, 2, 3, 4, 8, 9, 11, 13, 14, 18, 19, 20, 21, 25, 26, 29, 30, 31$

b) $f = \sum 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 18, 19, 22, 23, 24, 25, 26, 29, 30, 31$

12.2. Thực hiện các hàm Boole 6 biến :

$$f = \sum 0, 1, 3, 5, 7, 12, 14, 16, 18, 20, 22, 26, 28, 30, 32, 34, 37, 39, 41, 43, 45, 50, 51, 53, 60, 61, 62, 63$$

a) Bằng *MUX* 4 đầu vào và các mạch của *NAND*

b) Bằng *MUX* 8 đầu vào và *MUX* 4 đầu vào

12.3. Dùng mạch *NAND* và bộ giải mã $3 \rightarrow 8$ tạo các hàm sau:

$$f_1 = A B + \bar{A} \bar{B} \bar{C}$$

$$f_2 = A + B + \bar{C}$$

$$f_3 = \bar{A} B + A \bar{B}$$

12.4. Thiết kế mạch thực hiện hệ 4 phương trình sau :

$$f_1 = \bar{A} \bar{B} C + A B C$$

$$f_3 = A + B + \bar{C} D + \bar{A} D$$

$$f_2 = \bar{A} + \bar{B} + C + D$$

$$f_4 = A C D + A \bar{C} \bar{D} + B \bar{C} D + B C \bar{D}$$

dùng mạch *NAND* và bộ giải mã $4 \rightarrow 16$

12.5. Xây dựng *ROM* thực hiện :

a) NBCD \rightarrow mã dư 3 (XS3)

b) NBCD \rightarrow Gray dư 3

c) NBCD \rightarrow 2, 4, 2, 1

d) NBCD \rightarrow 6, 3, 1, -1

CHƯƠNG 13

MẠCH SỐ HỌC

Trong chương mạch tổ hợp đã giới thiệu các bộ cộng, trừ nhị phân một cột số. Mạch thực hiện là các vi mạch *SSI*.

Trong chương này sẽ giới thiệu mạch cộng trừ nhị phân nhiều cột số, cộng trừ mã *BCD* và các mạch nhân chia. Mạch thực hiện chủ yếu là các vi mạch *MSI*. Đơn vị số học và logic *ALU* (Arithmetic - Logic - Unit) là một thành phần cơ bản không thể thiếu được trong các μ P cũng được trình bày trong chương này.

13-1. BỘ CỘNG SONG SONG n BIT

Giả sử có hai số nhị phân n bit A và B , trong đó :

$$A = A_{n-1} A_{n-2} \dots A_0 \quad B = B_{n-1} B_{n-2} \dots B_0.$$

Để xây dựng mạch thực hiện phép cộng hai số A và B cần nắm vững nguyên tắc xây dựng một bộ cộng đầy đủ 1 bit, trên cơ sở đó xây dựng bộ cộng song song n bit. Cụ thể trong phần này sẽ giới thiệu phương pháp xây dựng bộ cộng song song 4 bit từ bộ cộng đầy đủ 1 bit.

13-1.1. BỘ CỘNG 1 BIT

Trong chương mạch tổ hợp đã giới thiệu cách thiết kế sơ đồ bộ cộng đầy đủ (Full Adder - viết tắt *FA*) 1 cột số.

Mạch có 3 đầu vào :

A_i là chữ số cột thứ i của số A

B_i là chữ số cột thứ i của số B

C_{i-1} là giá trị nhớ từ cột trái hơn bên cạnh đem đến.

Mạch có hai đầu ra :

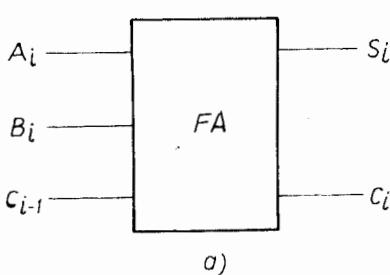
S_i là kết quả phép cộng ở cột thứ i .

C_i là giá trị nhớ sang cột già hơn tiếp theo.

Sơ đồ khối của *FA* 1 cột số được cho ở hình 13-1.a. Bảng chân lý của bộ cộng đầy đủ 1 bit được mô tả ở hình 13-1.b. S_i , C_i được biểu diễn trên bảng Karnaugh hình 13-1.c.

Từ bảng Karnaugh, tối thiểu hóa các hàm này nhận được kết quả :

$$\begin{aligned}
 S_i &= \bar{C}_{i-1} \bar{A}_i B_i + \bar{C}_{i-1} A_i \bar{B}_i + C_{i-1} \bar{A}_i \bar{B}_i + C_{i-1} A_i B_i \\
 &= C_{i-1} \oplus A_i \oplus B_i \\
 C_i &= A_i B_i + C_{i-1} (\bar{A}_i B_i + A_i \bar{B}_i) \\
 &= A_i B_i + C_{i-1} (A_i \oplus B_i).
 \end{aligned}$$



C_{i-1}	A_i	B_i	S_i	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

b)

S_i	$A_i B_i$	00	01	11	10
C_{i-1}					
0			1		1
1	1			1	

C_i	$A_i B_i$	00	01	11	10
C_{i-1}					
0				1	
1			1	1	1

c)

Hình 13-1. Bộ cộng 1 cột số.

a. Sơ đồ khối ; b. Bảng chân lý ; c. Bảng Karnaugh của S_i, C_i .

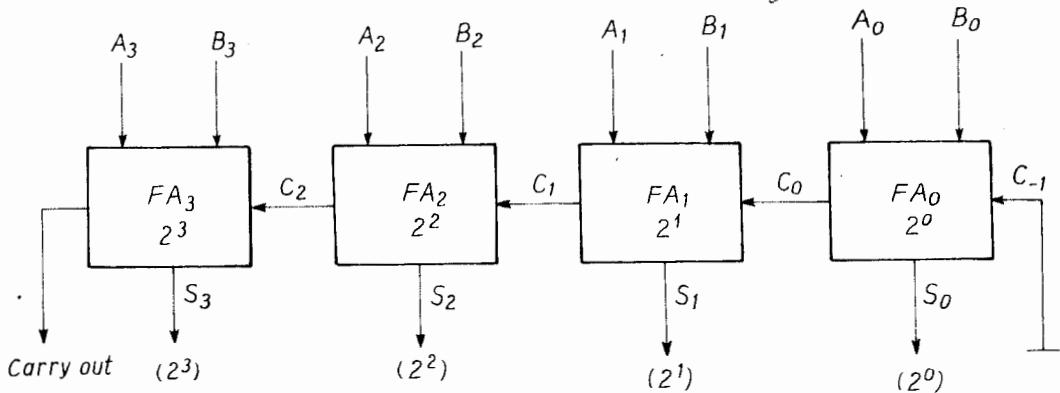
13-1.2. BỘ CỘNG SONG SONG 4 BIT

Giả sử ta có hai số nhị phân 4 bit A và B . Trong đó :

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0 ,$$

với A_0, B_0 là cột có trọng số bé nhất của hai số A và B (2^0) A_3, B_3 là cột có trọng số lớn nhất của hai số A và B (2^3). Từ bộ cộng nhị phân 1 bit ta có thể đưa ra mạch của bộ cộng song song 4 bit thực hiện phép cộng hai số A, B như hình 13-2.



Hình 13-2. Bộ cộng song song 4 bit.

Ví dụ : Có hai số A và B với $A = 1111$, $B = 1001$. Phép cộng hai số này được thực hiện như sau :

$$\begin{array}{r}
 A = \quad 1 \ 1 \ 1 \ 1 \\
 + \quad B = \quad 1 \ 0 \ 0 \ 1 \\
 C = \quad 1 \ 1 \ 1 \ 1 \\
 \hline
 \Sigma = \quad 1 \ 1 \ 0 \ 0 \ 0
 \end{array}$$

Nhược điểm của bộ cộng loại này :

Tín hiệu nhớ ở đầu ra (Carry Out) được tạo thành sau khi đi qua bốn bộ cộng $FA_0 \dots FA_3$, nếu mỗi bộ cộng FA_i là mạch hai tầng thì Carry Out sẽ xuất hiện ở đầu ra sau thời gian $T = 8$ lần thời gian quá độ với thời gian quá độ là thời gian quá độ của một tầng logic. Do vậy, thời gian trễ sẽ rất lớn đặc biệt khi số bit của mỗi số hạng tăng đủ lớn. Để khắc phục nhược điểm đó, người ta dùng bộ cộng nhớ nhanh hay là bộ cộng nhớ nhìn trước (Fast Carry hay Carry Look Ahead).

13-2. BỘ CỘNG NHỚ NHANH (FAST CARRY HAY CARRY LOOK AHEAD)

Như đã trình bày ở trên, trong bộ cộng song song n bit số nhớ truyền liên tiếp từ tầng này sang tầng khác làm cho thời gian thực hiện phép cộng lớn đặc biệt là khi n lớn. Do vậy cần có biện pháp khắc phục nhược điểm đó. Một trong những phương pháp đã được áp dụng là phương pháp nhớ nhanh (Fast Carry) hay là nhớ nhìn trước (Carry Look Ahead) sẽ được trình bày ở đây. Đối với một bộ cộng đầy đủ (FA) một cột số ta có :

$$C_i = (A_i \oplus B_i) C_{i-1} + A_i B_i \quad (13-1)$$

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

Đặt : $A_i \oplus B_i = P_i$
 $A_i B_i = G_i$ (13-2)

Thay (13-1) vào (13-2) ta có :

$$C_i = P_i C_{i-1} + G_i$$

$$S_i = P_i \oplus C_{i-1}$$

Đối với bộ cộng 4 bit's ta có :

$$C_0 = P_0 C_{-1} + G_0$$

$$C_1 = P_1 C_0 + G_1$$

$$C_2 = P_2 C_1 + G_2$$

$$C_3 = P_3 C_2 + G_3$$

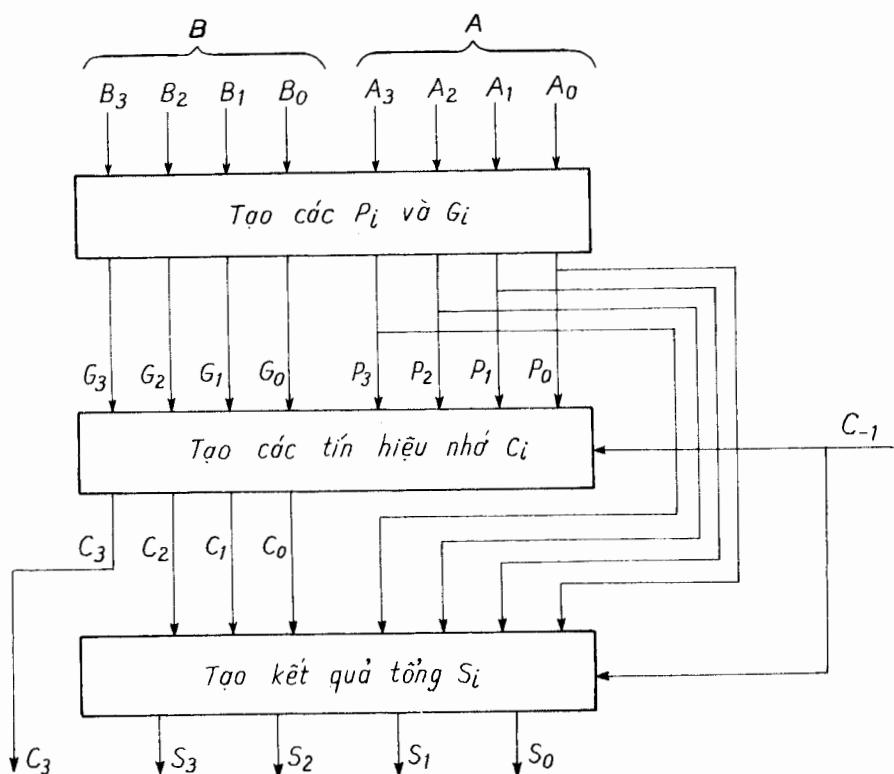
Thay C_0 vào C_1 , sau đó C_1 vào C_2 và C_2 vào C_3 ta được kết quả :

$$\begin{cases} C_1 = P_1 P_0 C_{-1} + P_1 G_0 + G_1 \\ C_2 = P_2 P_1 P_0 C_{-1} + P_2 P_1 G_0 + P_2 G_1 + G_2 \\ C_3 = P_3 P_2 P_1 P_0 C_{-1} + P_3 P_2 P_1 G_0 + P_3 P_2 G_1 + P_3 G_2 + G_3 \end{cases} \quad (13 - 3)$$

Các chữ số của tổng sẽ có giá trị như sau :

$$\begin{cases} S_0 = P_0 \oplus C_{-1} \\ S_1 = P_1 \oplus C_0 \\ S_2 = P_2 \oplus C_1 \\ S_3 = P_3 \oplus C_2 \end{cases} \quad (13 - 4)$$

Dùng hệ phương trình (13-3) và (13-4) để xây dựng bộ cộng song song nhớ nhanh. Dễ dàng nhận thấy rằng bộ cộng này gồm 3 khối chính, đó là :

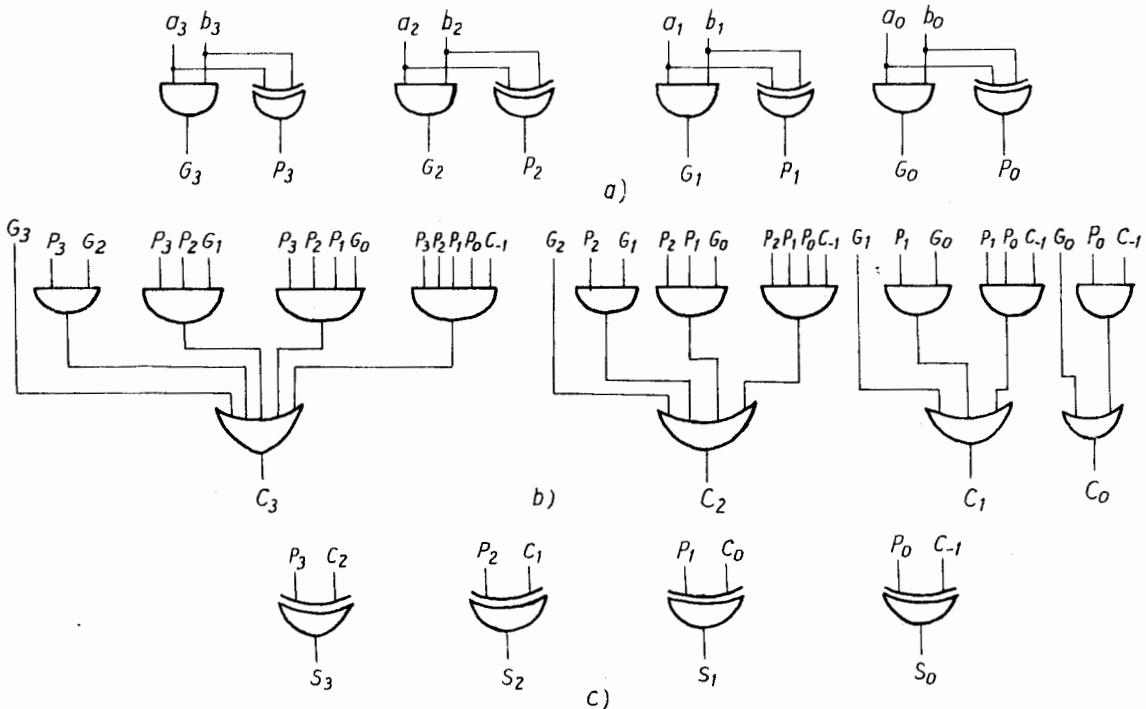


Hình 13-3. Sơ đồ khối bộ cộng song song 4 bit nhớ nhanh.

- Khởi tạo các giá trị P_i và G_i
- Khởi tạo các tín hiệu nhớ C_i
- Khởi tạo kết quả phép cộng S_i

Sơ đồ khối của bộ cộng song song 4 bit nhớ nhanh được mô tả trong hình 13-3.

Sơ đồ chi tiết của 3 khối này được chỉ ra ở hình 13-4,a, 13-4,b, 13-4,c.



Hình 13-4. Sơ đồ bộ cộng song song 4 bit nhớ nhanh.

- Sơ đồ logic khôi tạo P_i và G_i ;
- Sơ đồ logic khôi tạo nhớ C_i ;
- Sơ đồ logic khôi tạo kết quả của tổng S_i .

Nếu thiết kế bộ cộng song song 4 bit nói trên chỉ dùng mạch NAND, dễ dàng chứng minh được số mạch NAND cần cho các khôi và số mức logic của mạch như sau :

- Khởi tạo P_i và G_i cần 20 mạch NAND có 3 mức.
- Khởi tạo các tín hiệu nhớ C_i cần 18 mạch NAND có 2 mức
- Khởi tạo các chữ số của tổng S_i cần có 16 mạch NAND 3 mức.

Vậy mạch này cần 54 mạch NAND, để tạo được nhớ phải qua 5 mức logic và để tạo được kết quả của phép cộng phải qua 8 mức logic. Số mức logic còn đặc trưng cho thời gian thực hiện phép tính(thời gian quá độ của các mạch). Số mức logic để tạo các tín hiệu nhớ và kết quả của phương pháp này không đổi khi số bit n của các số hạng tăng. Vì vậy thời gian thực hiện phép cộng là không đổi khi n thay đổi.

13_3. MÃ BÙ

Như đã trình bày trong chương mạch tổ hợp, phân bộ cộng, trừ nhị phân một cột số : trong máy tính người ta không dùng mạch trừ mà tìm cách biểu diễn con số dưới dạng mã bù để có thể thay thế phép trừ bởi phép cộng với mã bù của nó.

Có hai loại mã bù dùng cho một bộ đếm cơ số r là : mã bù r và mã bù $r-1$. Máy tính thường hay tính toán với hệ đếm cơ số 2 và cơ số 10 cho nên có 4 loại mã bù cho nó, đó là : bù 2, bù 1 cho hệ đếm cơ số 2, bù 10 và bù 9 cho hệ đếm cơ số 10.

Mã bù của một số nguyên N trong hệ đếm cơ số r gồm n chữ số được định nghĩa như dưới đây.

13-3.1. BÙ R

Số bù r của số nguyên N ký hiệu là N^* và được xác định như sau :

$$N^* = r^n - N \quad (13 - 5)$$

Nếu $N = 000 \dots 0 = 0$ thì $N^* = r^n - 0 = r^n$

Ví dụ : a. Hệ đếm cơ số 2 :

N^* của 101100_2 là : $2^6 - 101100 = 10100$

N^* của 100101_2 là : $2^6 - 100101 = 11011$

b. Hệ đếm cơ số 10 :

N^* của 52460_{10} là : $10^5 - 52460 = 47540$

N^* của 1927_{10} là : $10^4 - 1927 = 8073$

13-3.2. BÙ R-1

Số bù $r - 1$ của số nguyên N ký hiệu là N^{**} , giá trị của N^{**} được xác định như sau :

$$N^{**} = r^n - 1 - N \quad (13 - 6)$$

Nếu số nguyên N có n cột số ký hiệu là $N_n N_{n-1} \dots N_1$ (với N_1 là cột có trọng số nhỏ nhất r^0 , N_n là cột có trọng số lớn nhất r^{n-1}).

Nếu N^{**} có n cột số ký hiệu là $N_n^{**} N_{n-1}^{**} \dots N_1^{**}$, Với N_1^{**} là cột có trọng số nhỏ nhất, N_n^{**} là cột có trọng số lớn nhất. Biết rằng :

$$r^n - 1 = \underbrace{(r-1)(r-1) \dots (r-1)}_{n \text{ chữ số mỗi chữ số đều là } r-1}(r-1) \dots .$$

Theo định nghĩa của N^{**} ta có :

$$N^{**} = (r - 1)(r - 1) \dots (r - 1)(r - 1) - N_n N_{n-1} \dots N_2 N_1$$

$$\text{Tổng quát : } N_i^{**} = (r - 1) - N_i \quad (13 - 7)$$

$$\text{với } \forall i = 1 \div n$$

Từ công thức (13-7), ta suy ra cách xác định mã bù $r - 1$ của số nguyên N như sau:

Mã bù $r - 1$ của một số nguyên N là một số có n cột số như của $N(n)$. Chữ số ở một cột bất kỳ được xác định bằng cách lấy $(r - 1)$ trừ đi chữ số tương ứng của số N . Điều này đúng với mọi cột thứ i ($i = 1 \div n$).

Ví dụ : Hệ đếm cơ số 2 :

Bù 1 của 101100 là : $N^{**} = 111111 - 101100 = 010011$

Bù 1 của 110100 là : $N^{**} = 111111 - 110100 = 001011$

Hệ đếm cơ số 10 :

Bù 9 của 52460 là $N^{**} = 99999 - 52460 = 47539$

Bù 9 của 19473 là $N^{**} = 99999 - 19473 = 80526$

Quan hệ giữa mã bù $r(N^*)$ và mã bù $r - 1(N^{**})$:

Từ các định nghĩa trên ta có : $N^* = N^{**} + 1$ (13 - 8)

Từ công thức (4) suy ra rằng muốn tìm mã bù r của một số nguyên N , ngoài cách dựa vào định nghĩa như công thức (1) ta có thể tìm mã bù $r - 1$ của nó rồi cộng thêm 1, nghĩa là làm theo thứ tự sau :

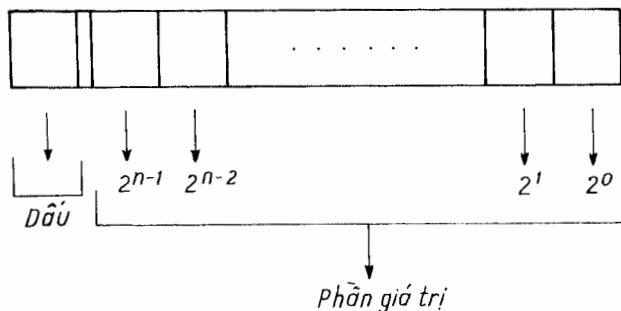
$$N \longrightarrow N^{**} \longrightarrow N^* = N^{**} + 1.$$

13-4. BIỂU DIỄN SỐ NHỊ PHÂN NGUYÊN TRONG MÁY TÍNH

Trong máy tính một số nhị phân nguyên có thể được biểu diễn theo các cách sau :

1. Dấu và phần giá trị.
2. Số bù 2.
3. Số bù 1.

Trong cả 3 cách biểu diễn đó, máy tính luôn luôn dùng một biến nhị phân để biểu diễn dấu cho số nhị phân nguyên đó. Cụ thể là : "0" cho số dương và "1" cho số âm. Cột dấu luôn luôn đứng cạnh cột có trọng số lớn nhất của số đó về phía trái và được ngăn cách với phần giá trị của số bởi dấu phẩy (hình 13-5).



Hình 13-5. Biểu diễn một số nhị phân nguyên.

13-4.1. BIỂU DIỄN SỐ NHỊ PHÂN NGUYÊN Ở DẠNG DẤU VÀ PHẦN GIÁ TRỊ

Nếu N là một số nhị phân nguyên thì N sẽ được biểu diễn như sau :

$$N = \begin{cases} 0, N \text{ nếu } N \geq 0 \\ 1, |N| \text{ nếu } N < 0 \end{cases}$$

Ví dụ: Số +13 được biểu diễn như sau : 0,1101; Số -13 được biểu diễn như sau : 1,1101.

13-4.2. BIỂU DIỄN SỐ NHỊ PHÂN NGUYÊN Ở DẠNG MÃ BÙ 2

Nếu N là một số nhị phân nguyên, được biểu diễn ở dạng mã bù 2 như sau :

$$N = \begin{cases} 0, N \text{ nếu } N \geq 0 \\ 1, N^* \text{ nếu } N < 0 \end{cases}$$

Trong đó N^* là mã bù 2 của N .

13-4.3. BIỂU DIỄN SỐ NHỊ PHÂN NGUYÊN Ở DẠNG MÃ BÙ 1 :

$$N = \begin{cases} 0, N \text{ nếu } N \geq 0 \\ 1, N^{**} \text{ nếu } N < 0 \end{cases}$$

Trong đó N^{**} là mã bù 1 của N .

Ví dụ : Số +13 có mã bù 2 = mã bù 1 = 0,1101

Số -13 có mã bù 1 = 1,0010

Số -13 có mã bù 2 = 1,0011

13-5. CÔNG VÀ TRỪ DÙNG MÃ BÙ 2

Như đã biết sử dụng mã bù có thể thay thế phép trừ các số bằng phép cộng mã bù của các số đó. Giả sử có 4 số +4, -4, +3, -3 và mã bù 2 của các số này là : 0.100 ; 1.100 ; 0.011 và 1.101. Hãy xét những tổng sau :

Tổng thứ nhất

$$\begin{array}{r} +4 : 0.100 \\ + \quad +3 : 0.011 \\ \hline 7 : 0.111 \end{array}$$

Tổng nhận được chính là mã bù 2 của 7.

Tổng thứ 2

$$\begin{array}{r} -4 : 1.100 \\ + \quad +3 : 0.011 \\ \hline -1 : 1.111 \end{array}$$

Tổng thứ 3

$$\begin{array}{r} +4 : 1.100 \\ + \quad -3 : 1.101 \\ \hline +1 : 10.001 \end{array}$$

Ở tổng này đã xuất hiện 1 giá trị nhớ từ cột dấu sang và nếu bỏ giá trị nhớ này thì tổng thu được chính là mã bù 2 của +1.

Tổng thứ tư :

$$\begin{array}{r}
 -4 : 1.100 \\
 +3 : 0.011 \\
 \hline
 -1 1.111
 \end{array}$$

Tổng thu được chính là mã bù 2 của -1.

Nhận xét :

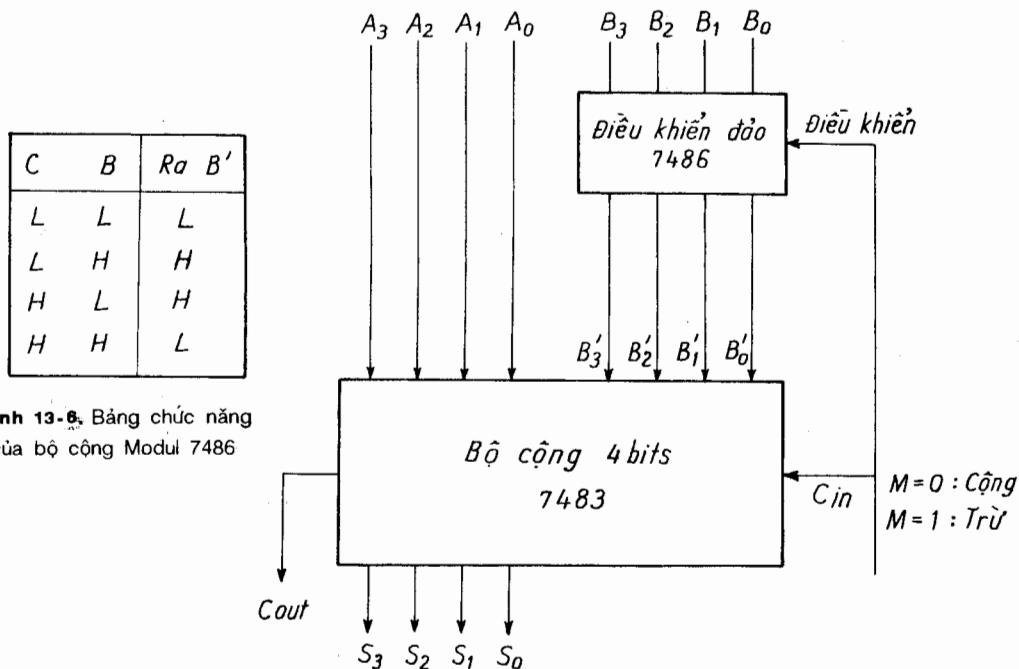
Phép cộng 2 số ở dạng mã bù 2 được tiến hành như bình thường kể cả cột dấu, nếu tổng nhận được có xuất hiện một giá trị nhỏ từ cột dấu sang thì bỏ giá trị đó đi.

Mạch thực hiện phép cộng 2 số ở dạng mã bù 2 dùng bộ cộng song song 4 bits nhị phân nhớ nhanh 7483 và bộ cộng modul 7486 như hình 13-7. Bảng chức năng của bộ cộng Modul 7486 được mô tả ở hình 13-6. Mạch ở hình 13-7 thực hiện phép cộng 2 số 4 bit A và B với:

$$A = A_3A_2A_1A_0$$

$$B = B_3B_2B_1B_0$$

trong đó A_3 và B_3 là bit dấu.



Hình 13-7. Mạch thực hiện cộng 2 số A, B ở dạng mã bù 2.

Nguyên tắc làm việc của mạch như sau :

A và B đã được biểu diễn ở dạng mã bù 2.

Nếu muốn thực hiện phép cộng ta đưa tín hiệu vào đầu M để điều khiển bộ cộng

modul và tín hiệu nhớ của bộ cộng 7483 là $M = "L" = 0$ lúc đó bộ cộng modul 7486 sẽ đưa ra số $B' = B'_3 B'_2 B'_1 B'_0$ trong đó $B'_i = B_i \oplus 0 = B_i$ với $i = 0 \div 3$.

Nếu muốn thực hiện phép trừ ta đưa tín hiệu vào đầu M là $M = "H" = "1"$ Lúc đó bộ cộng Modul 7486 sẽ đảo lại các bit B_i ($i = 1 \div 3$) và đưa kết quả ở đầu ra $B' = B'_3 B'_2 B'_1 B'_0$ trong đó $B'_i = B_i \oplus 0 = B_i$ ($i = 0 \div 3$) Và tín hiệu nhớ ban đầu của bộ cộng 7483 là $C_{in} = 1$, ta có $B' + C_{in}$ chính là mã bù 2 của đối số B ($B' + C_{in}$ là mã bù 2 của B). Ở bộ cộng 7483 ta thực hiện phép cộng như bình thường (cộng 2 số nhị phân 4 bit) trong đó bit S_3 chính là bít dấu.

13-6. CỘNG VÀ TRỪ DÙNG MÃ BÙ 1

Giả sử có 4 số là 4, 3, -4, -3, mã bù 1 của chúng tương ứng là 0. 100 0. 011 ; 1. 011 ; 1.100. Xét các tổng sau :

* Tổng 1 :

$$+4 : 0.100$$

$$+3 : 0.011$$

$$\begin{array}{r} +7 \\ \hline 0.111 \end{array}$$

Tổng thu được chính là mã bù 1 của 7.

* Tổng 2 :

$$-4 : 1.011$$

$$-3 : 1.100$$

$$\begin{array}{r} \hline -7 \\ \hline 10.111 \end{array}$$

Ở tổng thu được, xuất hiện giá trị nhớ từ cột dấu sang. Nếu ta lấy giá trị nhớ này đưa về cộng vào cột trẻ nhất của tổng ta được 1. 000 chính là mã bù 1 của -7.

* Tổng 3 :

$$+4 : 0.100$$

$$-3 : 1.100$$

$$\begin{array}{r} \hline +1 \\ \hline 10.001 \end{array}$$

Cũng tương tự nếu ta lấy giá trị nhớ đưa về cộng vào cột trẻ nhất của tổng thì ta được 0. 001 chính là mã bù 1 của 1.

* Tổng 4 :

$$-4 : 1.011$$

$$+3 : 0.011$$

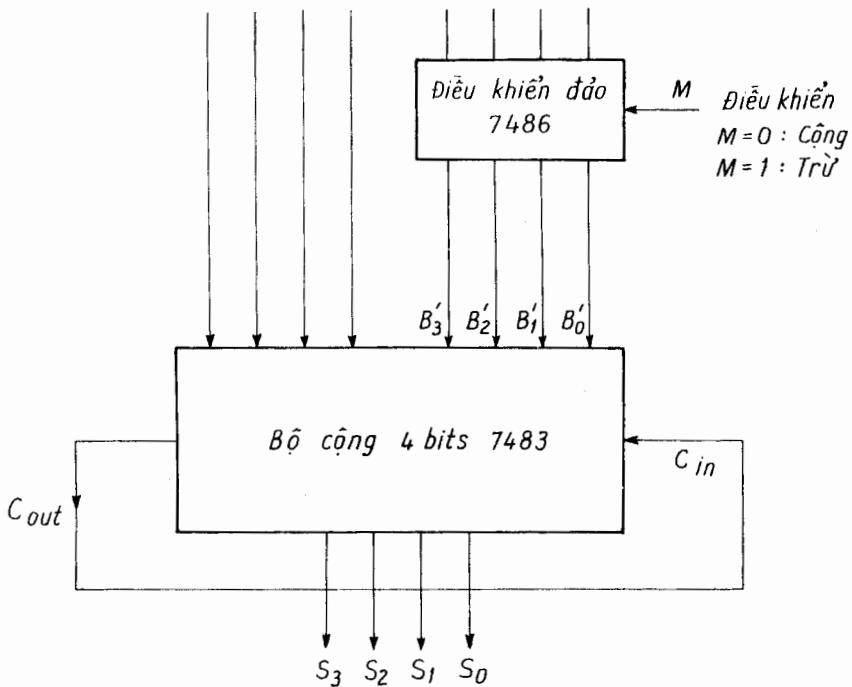
$$\begin{array}{r} \hline -1 \\ \hline 1.110 \end{array}$$

Tổng nhận được chính là mã bù 1 của -1.

Nhận xét :

Ta tiến hành cộng các số ở dạng mã bù như bình thường kể cả bit dấu. Trong trường hợp xuất hiện tín hiệu nhớ từ cột dấu đưa sang của tổng, thì tín hiệu đó được đưa về cộng vào cột số trẻ nhất của tổng.

Phép cộng trừ 2 số A, B ở dạng mã bù 1 trong đó A, B là các số 4 bit với bít có trọng số cao nhất là bit dấu, có thể thực hiện trên bộ cộng song song 4 bit nhớ nhanh 7483 như hình 13- 8



Hình 13-8. Cộng 2 số A , B ở dạng mã bù 1 nhờ mạch 7483 và 7486.

Như đã nêu ở mục 6 : Chức năng của bộ cộng Modul 7486 là đưa ra số B' được xác định bởi các bit $B'_3 B'_2 B'_1 B'_0$ trong đó $B'_i = B_i \oplus M$ ($i = 0 \div 3$). Nếu muốn thực hiện phép cộng thì đưa tín hiệu vào M là $L = "0"$; Muốn thực hiện phép trừ $A - B$ ta đưa ta đưa tín hiệu M là $H = "1"$ lúc đó

$$B' = B'_3 B'_2 B'_1 B'_0 = \bar{B}_3 \bar{B}_2 \bar{B}_1 \bar{B}_0$$

Ở bộ cộng 7483 tiến hành như bình thường. Nếu có tín hiệu nhớ từ cột dấu đưa sang ($Cout$) thì tín hiệu nhớ đó được đưa trở về cộng vào cột số trê nhất của tổng. Tổng S được xác định như sau : $S = S_3 S_2 S_1 S_0$ ở dạng mã bù 1 của tổng thực.

13-7. HIỆN TƯỢNG TRÀN

1. Định nghĩa :

Tràn là hiện tượng kết quả phép tính vượt quá khả năng biểu diễn của máy làm cho kết quả bị sai đi

2. Các ví dụ

Ví dụ 1 :

$$+1 : 0.001$$

$$+3 : 0.011$$

$$\dots\dots\dots$$

$$+4 : 0.100$$

Kết quả nhận được là đúng, không bị tràn

Ví dụ 2 :

$$\begin{array}{r} +5 \\ +6 \\ \hline +11 \end{array}$$

Kết quả nhận được là 1. 011 tức là -3 \Rightarrow sai

Hiện tượng tràn.

Ví dụ 3 :

$$\begin{array}{r} +5 \\ -3 \\ \hline +2 \end{array}$$

Kết quả nhận được đúng và không bị tràn

$$1.101$$

Ví dụ 4 :

$$\begin{array}{r} -5 \\ -6 \\ \hline -11 \end{array}$$

Kết quả nhận được là 0. 101 tức là +5 sai so với kết quả thực. Hiện tượng tràn.

$$10.101$$

3. Nguyên nhân tràn và phương pháp xác định tràn

Từ các ví dụ trên chúng ta có nhận xét về nguyên nhân gây nên tràn đó là phép cộng của 2 số cùng dấu mà kết quả vượt quá khả năng biểu diễn của máy.

Nó xảy ra khi 2 số đều dương mà kết quả lại âm $\bar{A}_n \bar{B}_n S$ hoặc 2 số đều âm mà kết quả lại là dương : $A_n B_n \bar{S}$

Trong đó :

A_n là dấu của số hạng thứ nhất A

B_n là dấu của số hạng thứ hai B

S là dấu của kết quả.

Ta có phương trình tràn là

$$\text{Tràn} = \bar{A}_n \bar{B}_n S + A_n B_n \bar{S}.$$

13-8. BỘ CỘNG TRỪ LIÊN TIẾP

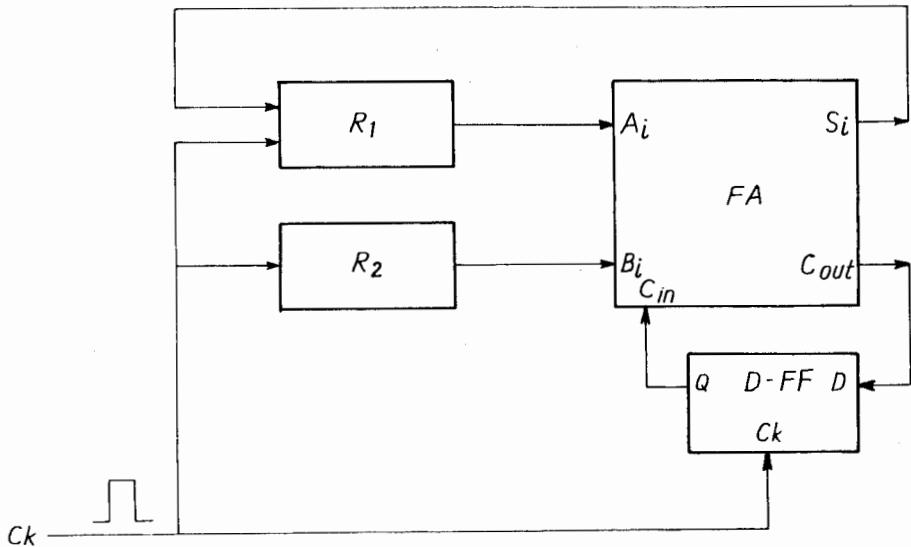
Cộng trừ liên tiếp là kết quả của quá trình thực hiện phép tính cộng (trừ) mà các bit của các số hạng được đưa liên tiếp vào đầu vào của 1 bộ cộng(trừ) 1 bit từ bit có trọng số nhỏ nhất đến bit có trọng số lớn nhất và kết quả cũng được đưa ra liên tiếp vào một bộ ghi dịch đồng bộ với xung nhịp tác động C_k từ bit có trọng số nhỏ nhất đến bit có trọng số lớn nhất.

13-8.1. BỘ CỘNG LIÊN TIẾP

Sơ đồ thực hiện của bộ cộng liên tiếp được mô tả như hình 13-9.

Nguyên lý làm việc :

Trạng thái ban đầu $Q = 0$ do đó $Cin = 0$. Khi có xung nhịp thứ nhất đưa vào 2 bit A_0, B_0 của 2 số hạng Avà B được đưa vào bộ cộng FA lúc đó đầu ra của bộ cộng 1 bit FA đưa ra 2 tín hiệu S_0 và C_0 trong đó $S_0 = A_0 \oplus B_0 \oplus 0 ; C_0 = A_0 B_0 + [A_0 \oplus B_0] 0 = A_0 B_0 = D$.



Hình 13-9. Sơ đồ bộ cộng liên tiếp.

Lúc đó toàn bộ các bit của 2 số hạng \$A, B\$ (nằm trên 2 bộ ghi dịch \$R_1, R_2\$) được dịch sang phải 1 bit. Giá trị \$S_0\$ được đưa vào bit đầu tiên bên trái của \$R_1\$. Khi có xung nhịp \$C_k\$ thứ 2 lúc đó \$Q = C_0 = C_{in}\$, 2 bit \$A_1, B_1\$ của \$A\$ và \$B\$ được đưa vào đầu vào bộ cộng \$FA\$. Bộ cộng \$FA\$ đưa 2 tín hiệu ra :

$$S_1 = A_1 \oplus B_1 \oplus C_0 \text{ và } C_1 = A_1 B_1 + (A_1 \oplus B_1) C_0 = C_{out} = D$$

\$S_1\$ đưa vào bit đầu tiên bên phải của \$R_1\$ lúc này \$S_0\$ và các bit của \$A, B\$ đã được dịch sang phải 1 bit.

Quá trình cộng được thực hiện như thế cho đến khi kết quả bit cuối cùng của tổng \$S(S_{n-1})\$ được đưa vào bộ ghi dịch \$R_1\$ từng bit một. Bộ cộng này có nhược điểm là thời gian thực hiện phép cộng lớn.

13-8.2. BỘ CỘNG TRỪ LIÊN TIẾP

Sơ đồ thực hiện của bộ cộng trừ liên tiếp được minh họa ở hình 13-10

I là đầu đưa tín hiệu bắt đầu thực hiện phép tính. M là đầu đưa tín hiệu vào điều khiển $M = 0$: cộng ; $M = 1$: trừ ($A - B$ cụ thể $A_i - B_i$)

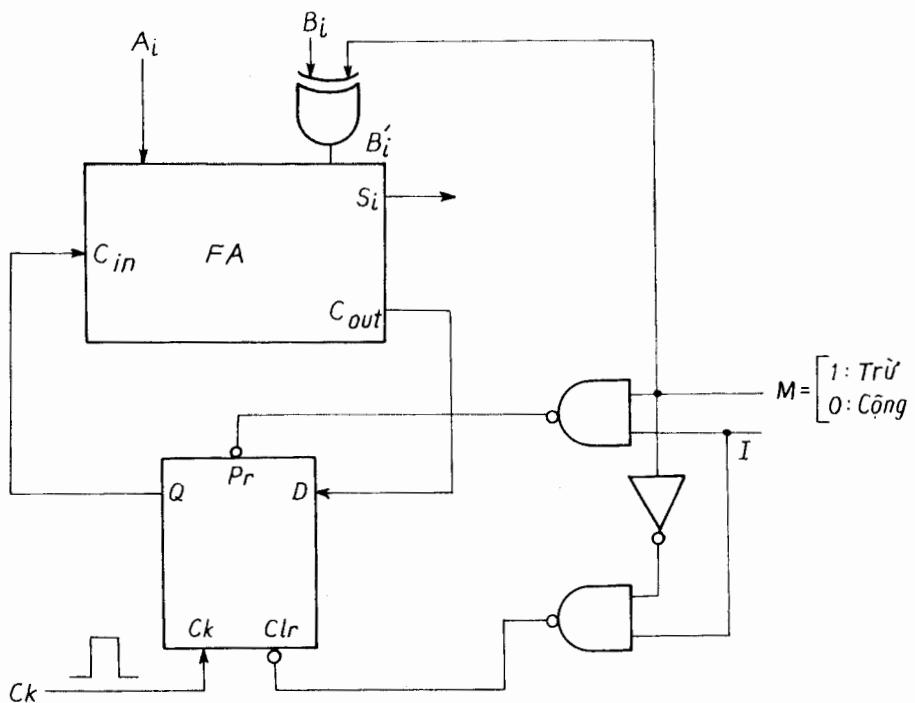
$$M = 0 : \text{cộng } B'_i = B_i \oplus M = B_i$$

Khi $I = 1$ thì \$Clr\$ của \$D\$-FF ở mức thấp. Lúc đó \$A_i, B_i, Q = C_{in}\$ được đưa vào bộ cộng từng bit 1 theo xung nhịp \$C_k\$ tác động. \$S_i\$ được gửi ra một bộ ghi dịch theo từng bit 1 đồng bộ với xung nhịp.

$$M = 1 : \text{trừ}$$

Khi $I = 1$ thì \$Pr\$ ở mức thấp. \$A_i, B_i\$ và \$C_{in} = C_{i-1}\$ được đưa vào bộ cộng từng bit 1 đồng bộ theo xung nhịp. \$B_i\$ qua bộ cộng Modul là \$B'_i

$B'_i = B_i \oplus M = B_i \oplus 1 = \bar{B}_i \Rightarrow$ Phép trừ được thực hiện ở bit thứ i . Kết quả \$S\$ được gửi ra một bộ ghi dịch theo từng bit 1 đồng bộ với xung nhịp.



Hình 13-10. Bộ cộng trừ liên tiếp.

13-9. CỘNG SỐ THẬP PHÂN DÙNG BỘ CỘNG MSI SỐ NHỊ PHÂN

Các chữ số thập phân ở đây được biểu diễn bởi mã NBCD (8421)

1. Khi kết quả ≤ 9

Kết quả đúng không phải hiệu chỉnh

$$\begin{array}{r} \text{Ví dụ } 4 + 5 : \quad 4 \quad : \quad 0100 \\ \qquad \qquad \qquad 5 \quad : \quad 0101 \\ \hline \qquad \qquad \qquad 9 \qquad \qquad 1001 \end{array}$$

2. Khi kết quả $9 < S \leq 15$ với Slù tổng của các số NBCD đó

$$\begin{array}{r} \text{Ví dụ :} \quad 7 \quad : \quad 0111 \\ \qquad \qquad \qquad 7 \quad : \quad 0111 \\ \hline \qquad \qquad \qquad 14 \qquad \qquad 1110 \end{array}$$

Kết quả vượt quá giá trị của các số NBCD và không có nhớ sang cột tiếp theo. Vì vậy phải hiệu chỉnh lại kết quả đó:

Ta lấy kết quả của phép cộng trừ đi 10 (thập phân) là 1010 tức là lấy kết quả cộng với 0110 (Mã bù 2 của 1010) được kết quả đúng và có nhớ.

Ví dụ : Từ kết quả của phép cộng trên là 1110 ta có :

1110
0110

$$\begin{array}{r} \text{Kết quả đúng và có nhớ sang cột tiếp theo} \\ \hline 1.0100 \end{array}$$

3. Khi kết quả $S : 15 < S \leq 19$:

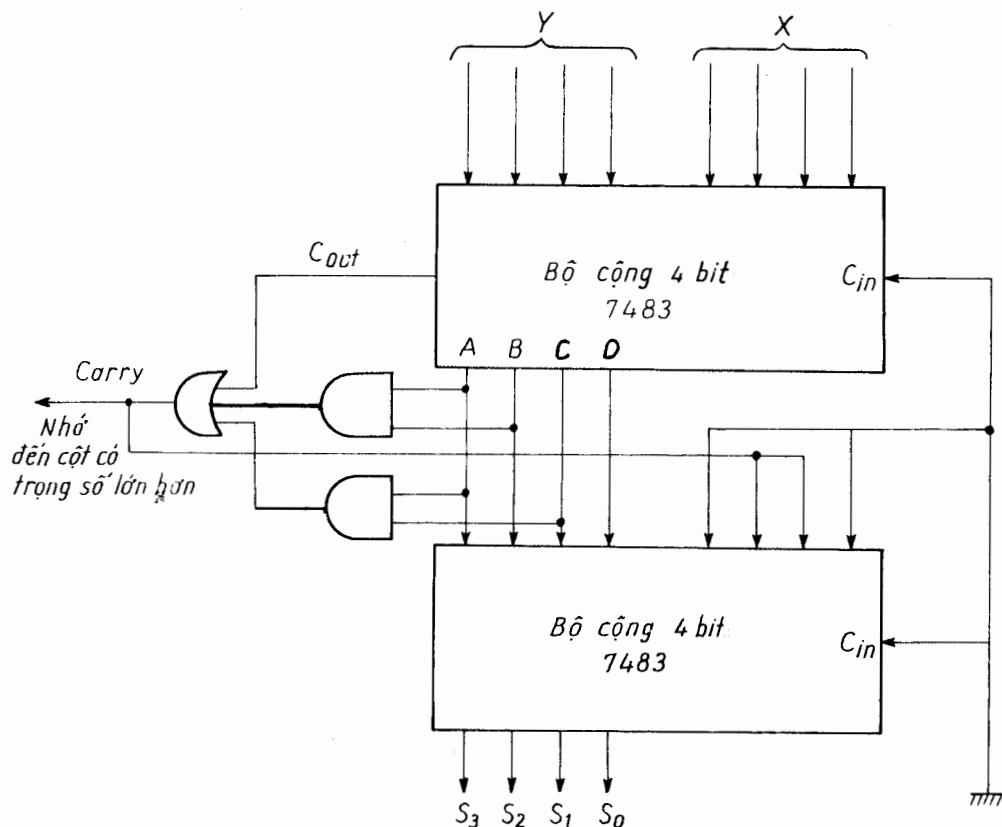
Kết quả nhận được sai và có nhớ sang cột tiếp theo do vậy phải hiệu chỉnh kết quả.

Khi đó tín hiệu nhớ $\text{Carry} = \text{Cout} + AB + AC$.

Mạch tạo nhớ chỉ dùng *NAND* có dạng:

$$\text{Carry} = \overline{\text{Cout}} \cdot \overline{AB} \cdot \overline{AC}$$

* Mạch thực hiện phép cộng 2 số *NBCD* : X, Y và có tạo tín hiệu nhớ đồng thời hiệu chỉnh kết quả được mô tả ở hình 13-11.



Hình 13- 11. Mạch cộng 2 số *NBCD* có tạo nhớ và hiệu chỉnh.

13-10. DÙNG SỐ BÙ CHO PHÉP TÍNH THẬP PHÂN

Phép trừ số thập phân cũng được thay thế bởi phép cộng với số bù của nó. Với số thập phân có 2 loại mà bù thường được dùng là :

1. Bù 10 :

N là số thập phân có n cột số có nghĩa : Mà bù 10 của nó là N^* với

$$N^* = 10^n - N$$

Ví dụ : $N = 972 \Rightarrow N^* = 10^3 - 972 = 28$

2. Bù 9 :

Với n là số thập phân có n cột số có nghĩa. Mà bù 9 của N là N^{**} với

$$N = 972 \Rightarrow N^{**} = 10^n - 1 - N = 10^3 - 1 - 972 = 27$$

+ Cách tìm mã bù 9 của số N :

$N^{**} = N_n^{**} N_{n-1}^{**} \dots N_i^{**} \dots N_2^{**} N_1^{**}$, trong đó $N_i^{**} = 9 - N_i$ với N_i là chữ số thứ i của N .

3. Ví dụ phép cộng, trừ dùng mã bù 9 và bù 10 :

a. Bù 10 : Một số nguyên N được biểu diễn ở mã bù 10 như sau :

$$N = \begin{cases} N & \text{nếu } N \geq 0 \\ N^* & \text{nếu } n < 0 \end{cases}$$

Ví dụ 1 : mã bù 10 của 48 là 48

mã bù 10 của -27 là 73

$+48$: 48 Ta nhận thấy tổng nhận được có xuất hiện

-27 : 73 giá trị nhớ sang cột tiếp theo. Nếu bỏ giá

$\underline{\quad}$ $\underline{\quad}$ trị này đi ta thu được kết quả đúng.

Ví dụ 2 : -48 : 52

$+27$: 27 79 là mã bù 10 của 21. Kết quả đúng

$\underline{-21}$ $\underline{\quad}$ 79

b. Cộng 2 số được biểu diễn ở mã bù 9 :

Một số nguyên N được biểu diễn ở mã bù 9 như sau :

$$N = \begin{cases} N & \text{nếu } N \geq 0 \\ N^* & \text{nếu } n < 0 \end{cases}$$

Ví dụ 1 : $+48$: 48 Tổng nhận được có xuất hiện nhớ từ cột

-27 : 72 có trọng lượng cao nhất sang. Nếu lấy giá

$\underline{\quad}$ $\underline{\quad}$ nhớ đưa về cộng với cột trẻ nhất ta sẽ được kết

21 120 quả đúng. Cụ thể ở đây $20+1=21$, 21 là kết quả đúng.

Người ta mã hóa dấu cho các số ở dạng bù 9 như sau

Dấu + : 0000

Ví dụ : +92 được mã hoá là 0.92 : 0000. 10010010

Dấu - : 1001

Ví dụ 2 : -92 được mã hoá là 9.92 : 1001. 10010010

13-11. CỘNG TRỪ SỐ THẬP PHÂN

1) Dùng mã bù 9 :

Mạch thực hiện phép cộng hoặc trừ số thập phân sử dụng mã bù 9 dùng mạch cộng FA (7483) được minh họa như hình 13-12a.

Trong đó M là đầu điều khiển đưa vào để chọn phép tính :

$M = 1$: trừ ; $M = 0$: Cộng

$M = 1$: Tín hiệu điều khiển sẽ chọn và đưa tín hiệu ra ở khối 4 Mux 2 $\Rightarrow 1$ là X_8, X_4, X_2, X_1 là dạng mã bù của X_8, X_4, X_2, X_1 .

$M = 0$: Tín hiệu ra của khối 4 MUX 2 $\Rightarrow 1$ là X_8, X_4, X_2, X_1

Bộ cộng FA 7438 là bộ cộng song song 4 bits nhớ nhanh.

Các giá trị của X'_8, X'_4, X'_2, X'_1 được xác định theo bảng hình 13-12b và 13-12c. Lúc đó ta có :

$$X'_8 = \bar{X}_8 \bar{X}_4 \bar{X}_2$$

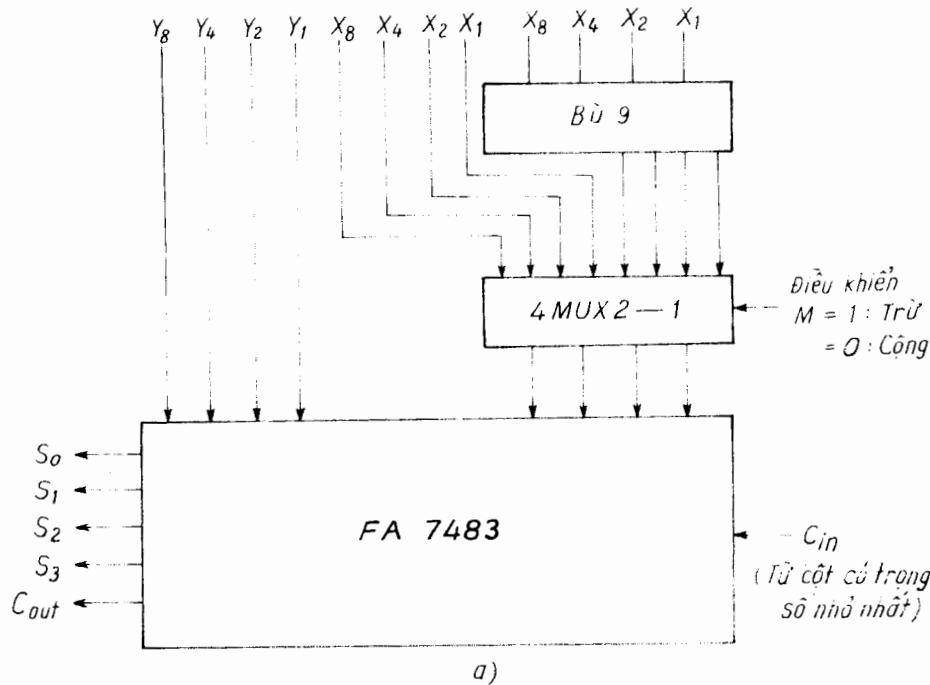
$$X'_4 = X_4 \cdot \bar{X}_2 + \bar{X}_4 \cdot X_2 = X_4 \oplus X_2$$

$$X'_2 = X_2$$

$$X'_1 = \bar{X}_1.$$

2. Dùng mã bù 10 :

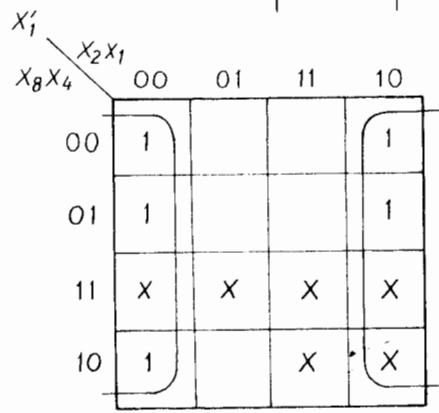
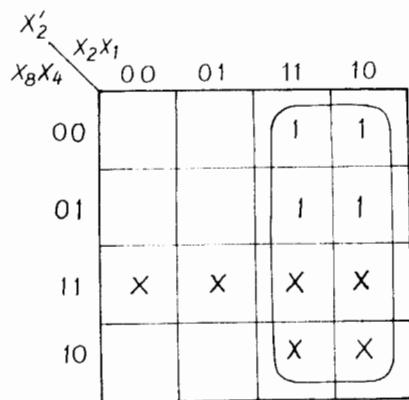
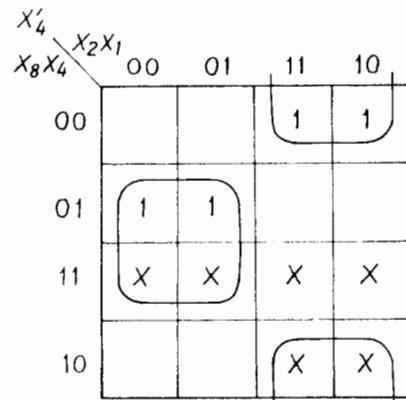
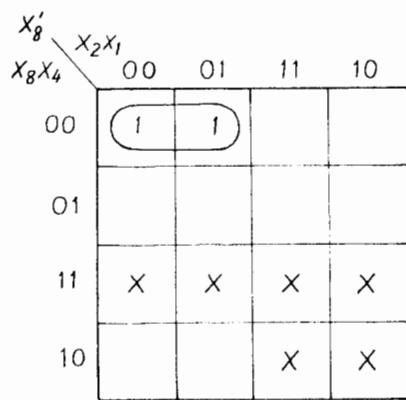
Đối với mã bù 10 chúng ta cũng có thể thực hiện phép tính cộng trừ một cách tương tự. Nhưng khi phân tích hàm đạt số phức tạp hơn nhiều dẫn đến sơ đồ thực hiện sẽ phức tạp hơn. Chúng ta cũng có thể sử dụng mã 3 để xây dựng bộ cộng trừ các số thập phân. Lúc đó sơ đồ thực hiện đơn giản và dễ thiết kế hơn nhiều so với việc sử dụng mã bù 10.



a)

X_8	X_4	X_2	X_1	X'_8	X'_4	X'_2	X'_1
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	-
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0

b)



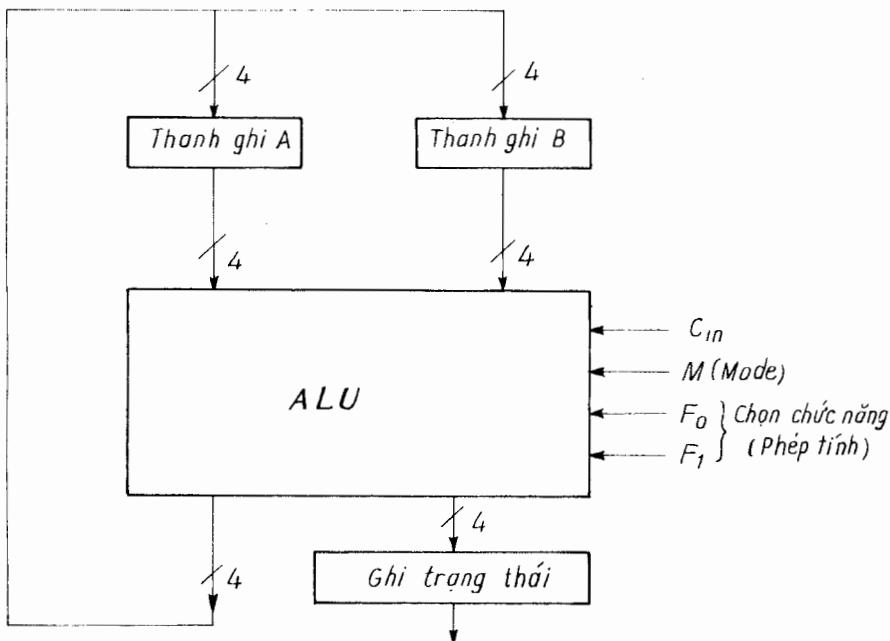
c)

Hình 13-12. a) Mạch cộng trừ số thập phân sử dụng mã bù 9 ; b, c) Mối liên hệ giữa các bit của mã bù với các bit của số ban đầu .

13-12. ĐƠN VỊ SỐ HỌC VÀ LOGIC (ALU)

Dơn vị số học và logic (Arithmetic - Logic Unit) là một thành phần cơ bản không thể thiếu được trong các MP. Nó bao gồm 2 khối chính đó là : khối logic và khối số học.

- Khối logic : Thực hiện các phép tính logic như là AND, OR, NOT, XOR.
 - Khối số học : Thực hiện các phép tính số học như là : cộng, trừ, tăng 1, giảm 1
- Sơ đồ khối của 1 đơn vị số học - logic ALU4 bits được mô tả như hình 13 - 13 :



Hình 13-13. Sơ đồ khối *ALU* 4 bit.

Trong đó :

M là đầu vào chọn phép tính số học hay logic. $F_0 F_1$ là 2 đầu vào chọn chức năng. Sau khi một phép tính số học hay logic được thực hiện kết quả sẽ được ghi lên 1 thanh ghi, ví dụ ở trên là thanh ghi A. Kết quả này có thể được sử dụng để thực hiện phép tính sau. Bộ *ALU* còn tạo ra các bit trạng thái chuyển đổi thanh ghi. Ví dụ : Carry out; nếu có nhớ, Zero nếu kết quả phép tính bằng 0. Sau đây là quá trình thiết kế 1 đơn vị số học - logic (*ALU*) 4 bits. Trong trường hợp *ALU* có số lượng bit lớn hơn cùng tiến hành tương tự.

13-13. THIẾT KẾ MỘT ĐƠN VỊ SỐ HỌC - LOGIC (ALU) 4 BIT

Như đã nêu ở trên, một đơn vị số học - logic phải thực hiện được các phép tính số học và logic như là : cộng, trừ, tăng 1 ... và AND, OR, NOT ... Căn cứ vào yêu cầu trên mà khối số học và logic có 2 khối chính đó là :

- Khối số học : Có chức năng thực hiện các phép tính số học : cộng, trừ ...

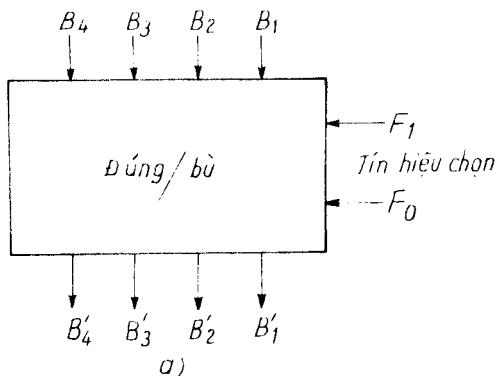
- Khối logic : Có chức năng thực hiện các phép tính logic : AND, OR ...

Chúng ta sẽ trình bày phương pháp thiết kế từng khối cơ bản một. Sau đó sẽ đưa ra cách ghép nối 2 khối đó như thế nào cho hợp lý.

13-13.1. KHỐI SỐ HỌC

Khối này bao gồm 2 bộ phận chính là :

1. Bộ đúng / bù có 2 đầu vào điều khiển là F_0F_1 để chọn phép tính như cộng trừ ... 4 đầu đưa số liệu vào , 4 đầu đưa số liệu ra đã được biến đổi để thuận tiện cho việc thực hiện phép tính. Sơ đồ khối của bộ đúng bù được mô tả như hình 13-14,a. Bảng chức năng của khối này được cho trong hình 13-14,b.

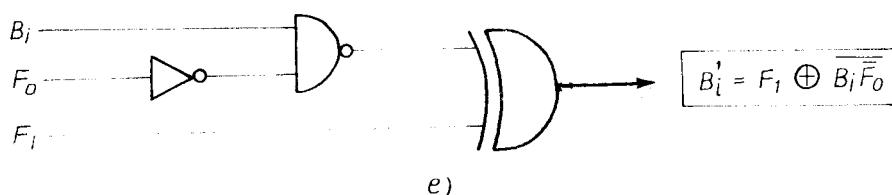
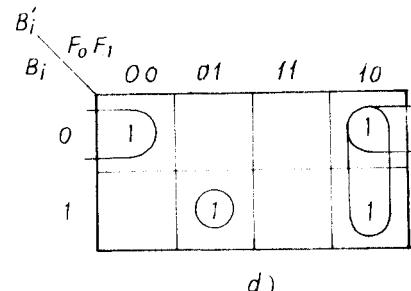


F_0F_1	B'_4	B'_3	B'_2	B'_1	Chú thích
0 0	\bar{B}_4	\bar{B}_3	\bar{B}_2	\bar{B}_1	Bù
0 1	B_4	B_3	B_2	B_1	Đúng
1 0	1	1	1	1	1
1 1	0	0	0	0	0

b)

F_0	F_1	B_i	B'_i	Ghi chú
0	0	0	1	
0	0	1	0	\bar{B}_i
0	1	0	0	
0	1	1	1	B_i
1	0	0	1	
1	0	1	1	1
1	1	0	0	
1	1	1	0	0

c)



Hình 13-14. a) Sơ đồ khối của bộ đúng/bù ; b) Bảng chức năng ; c) Bảng chân lý của B'_i ; d) Bảng Karnaugh của B'_i ; e) Sơ đồ tạo tín hiệu B'_i ;

Từ bảng chức năng này, lập được bảng chuyển đổi với 1 cột B_i bất kỳ của số hạng đưa vào, như hình 13-14,c.

Từ bảng hình 13-14,c ta tìm được quan hệ giữa B_i và F_0, F_1, B_i :

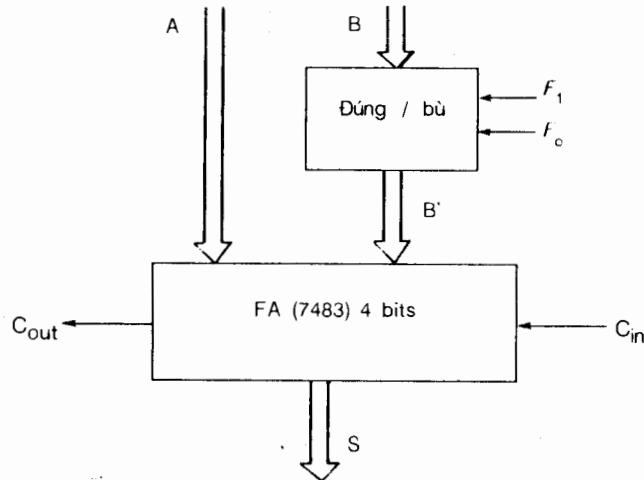
$$\begin{aligned} B_i &= F_0 \bar{F}_1 + \bar{B}_i \bar{F}_1 + B_i \bar{F}_0 F_1 = F_1 (B_i \bar{F}_0) + \bar{F}_1 (F_0 + \bar{B}_i) \\ &= F_1 (B_i \bar{F}_0) + \bar{F}_1 (\bar{B}_i F_0) \\ &= \overline{F_1 \oplus B_i \bar{F}_0} = F_1 \oplus (\bar{B}_i \bar{F}_0) = \bar{F}_1 \oplus (B_i \bar{F}_0). \end{aligned}$$

Từ phương trình biểu diễn mối quan hệ giữa B_i và F_0, F_1, B_i ta lập sơ đồ đưa ra tín hiệu B_i từ B_i, F_0, F_1 dùng các mạch tổ hợp, như hình 13-14,e.

2. Một bộ cộng song song 4 bit nhớ nhanh thường là 7483 :

Bộ cộng này có nhiệm vụ cộng các tín hiệu đưa vào và đưa ra kết quả là tổng S và giá trị nhớ Cout. Sơ đồ khối của bộ cộng này được mô tả ở hình 13-14,e.

Ta có sơ đồ khối tổng quát của khối số học như hình 13-14,f. Trong đó C_{in} là giá trị khối ban đầu.



Hình 13-14. f) Sơ đồ khối của bộ cộng trong khối số học.

Üng với các giá trị có thể có của F_0, F_1, C_{in} mà chúng ta có các trường hợp sau : (xem hình 13-14f)

- Khi $F_0 F_1 C_{in} = 000$:

Lúc đó : $S = A + \bar{B}$ bởi vì $B = \bar{B}$

- Khi $F_0 F_1 C_{in} = 001$:

$B = \bar{B}$ và $S = A + \bar{B} + C_{in} = A + (\bar{B} + 1) = A - B$

Tức là S bằng A cộng với mã bù 2 của B

- Khi $F_0 F_1 C_{in} = 010$:

$B = B$ và $S = A + B + C_{in} = A + B$

- Khi $F_0 F_1 C_{in} = 011$:

$B = B$ và $S = A + B + C_{in} = A + B + 1$

- Khi $F_0 F_1 C_{in} = 100$

$B = 1111$ (B, B', A, S là các số nhị phân 4 bit).

Lúc đó	$S = A + B' + C_{in} = A + 1111 = A - 1$ vì -1 có mã bù 2 là 1111
- Khi	$F_0 F_1 C_{in} = 101 :$ $B' = 1111$ Và $S = A + B' + C_{in} = A + (1111 + 1) = A$
Vì :	$(A + 1111) + 1 = (A - 1) + 1 = A$
- Khi	$F_0 F_1 C_{in} = 110 :$ $B' = 0000 \Rightarrow S = A + B' + C_{in} = A$
- Khi	$F_0 F_1 C_{in} = 111 :$ $B' = 0000 \Rightarrow S = A + B' + C_{in} = A + 1$

Ta lập được bảng chức năng của khối số học như hình 13-14,g. Trong đó F_0, F_1 là tín hiệu vào và C_{in} là tín hiệu nhớ ban đầu.

F_0	F_1	C_{in}	S	Chức năng
0	0	0	$A + \bar{B}$	A cộng với bù 1 của B
0	0	1	$A - B$	Trừ
0	1	0	$A + B$	Cộng
0	1	1	$A + B + C_{in}$	Cộng có nhớ ban đầu
1	0	0	$A - 1$	Giảm 1
1	0	1	A	Truyền A
1	1	0	A	Truyền A
1	1	1	$A + 1$	Tăng 1

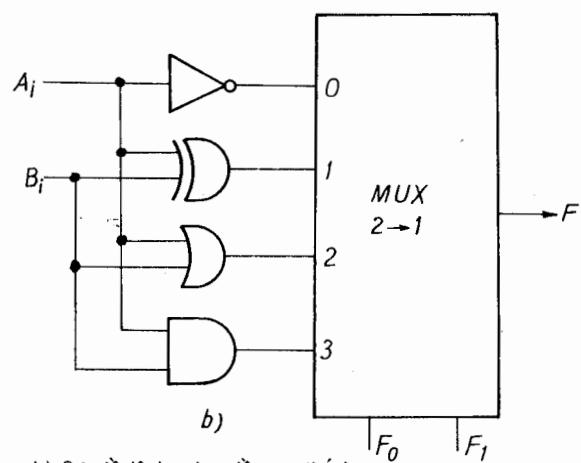
Hình 13-14.g) Bảng chức năng của khối số học.

13-13.2. KHỐI LOGIC

Khối logic có chức năng như sau : AND, OR, NOT, XOR. Bảng chức năng của khối logic được mô tả 13-15,a. Trong đó $F_0 F_1$ là hai tín hiệu điều khiển. Sơ đồ khối cho một khối logic cho cột thứ i được mô tả ở hình 13-15,b. Trong trường hợp khối logic 4 bit thì nó sẽ có 4 khối như vậy.

F_0	F_1	f	Chức năng
0	0	\bar{A}	NOT
0	1	$A \oplus B$	XOR
1	0	$A + B$	OR
1	1	$A.B$	AND

a)

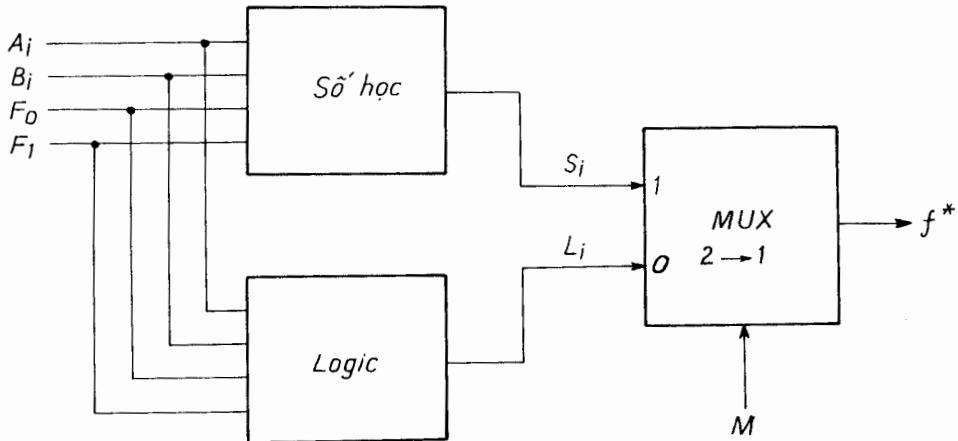


Hình 13-15. a) Bảng chức năng ; b) Sơ đồ logic cho đầu ra thứ i.

13-13.3. GHÉP NỐI HAI KHỐI SỐ HỌC VÀ LOGIC TRONG ĐƠN VỊ SỐ HỌC LOGIC ALU

Thiết kế khối số học logic (ALU) như thế nào để cho tại một thời điểm bất kỳ tùy thuộc vào yêu cầu của các tín hiệu vào nó chỉ thực hiện một chức năng nhất định hoặc là thực hiện các phép tính số học hoặc là thực hiện các chức năng logic. Có như vậy thì nó mới làm việc tin cậy được.

Để giải quyết yêu cầu này thì đối với mỗi cột của dữ liệu vào (A_i, B_i) cần phải có một thiết bị chọn lọc chức năng. Thông thường người ta dùng 1 MUX $2 \Rightarrow 1$. Sơ đồ khối ghép nối giữa 2 khối logic và số học cho mỗi cột các dữ liệu vào (A_i, B_i) được minh họa ở hình 13-16.



Hình 13-16. Sơ đồ khối cho một cột của ALU.

ALU được điều khiển thực hiện phép tính số học hay logic như tín hiệu M

$M = 0 \Rightarrow ALU$ thực hiện phép toán logic và đưa kết quả ra đầu ra f^* .

$M = 1 \Rightarrow$ thì ALU thực hiện phép toán số học và đưa kết quả ra đầu ra f^* .

Thực chất tín hiệu M ở đây là tín hiệu điều khiển chọn của MUX $2 \Rightarrow 1$.

13-14. BỘ NHÂN NHỊ PHÂN DÙNG MẠCH TỔ HỢP

Xét phép nhân 2 số nhị phân 4 bit với nhau. Trường hợp số bits lớn hơn cũng thực hiện hoàn toàn tương tự.

Giả sử có 2 số nhị phân 4 bit là A và B với :

$$A = A_3 \ A_2 \ A_1 \ A_0$$

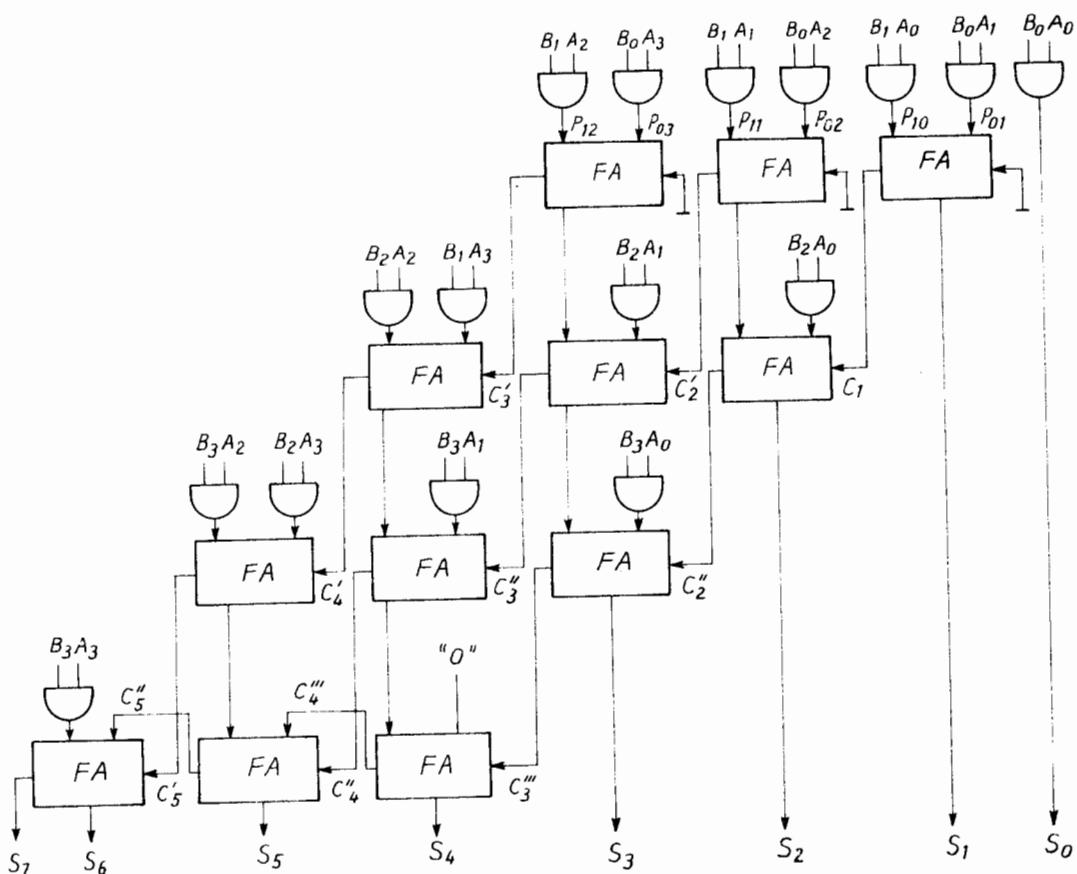
$$B = B_3 \ B_2 \ B_1 \ B_0$$

Trong đó (A_0, B_0) là các cột có trọng số nhỏ nhất, (A_3, B_3) là cột có trọng số lớn nhất.

Chúng ta cần tìm tích của chúng : $S = A \cdot B$. Có thể mô tả các bước của phép nhân như sau :

$$\begin{array}{r}
 A = \begin{matrix} A_3 & A_2 & A_1 & A_0 \end{matrix} \\
 \times \quad B = \begin{matrix} B_3 & B_2 & B_1 & B_0 \end{matrix} \\
 \hline
 \begin{matrix} P_{03} & P_{02} & P_{01} & P_{00} \\ P_{13} & P_{12} & P_{11} & P_{10} \\ P_{23} & P_{22} & P_{21} & P_{20} \\ P_{33} & P_{32} & P_{31} & P_{30} \end{matrix} \\
 \hline
 \begin{matrix} C'_5 & C'_4 & C'_3 & C'_2 & C_1 & S_1 & S_0 \\ C''_5 & C''_4 & C''_3 & C''_2 & & & \\ C'''_4 & C'''_3 & & & & & \end{matrix} \\
 \hline
 \begin{matrix} S_7 & S_6 & S_5 & S_4 & S_3 & S_2 & S_1 & S_0 \end{matrix}
 \end{array}$$

Với $P_{ij} = A_j B_i$. Sơ đồ thực hiện phép nhân 2 số A, B bởi trên được mô tả ở hình 13-17.

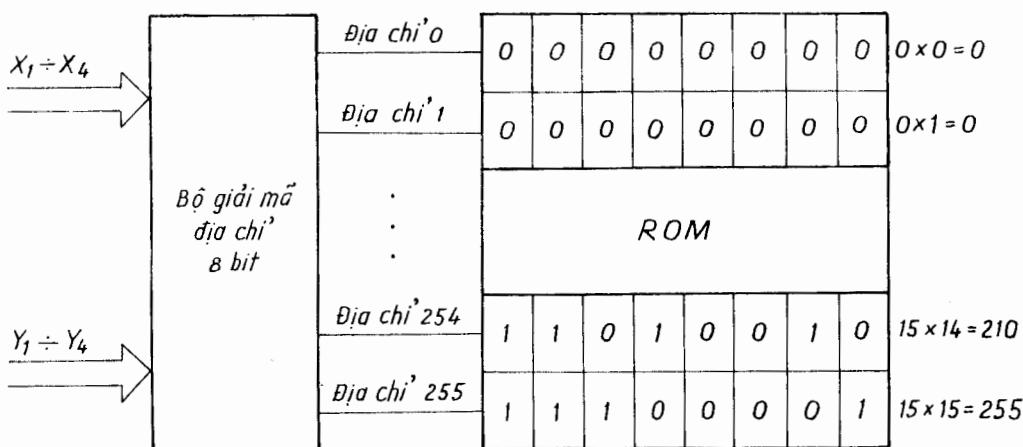


Hình 13-17. Sơ đồ mạch thực hiện phép nhân 2 số nhị phân 4 bit A và B.

Đối với bộ nhân thực hiện phép nhân song song như ở trên, khi số bit của số nhân và số bị nhân tăng lên thì số thiết bị cần dùng để xây dựng bộ nhân cũng tăng lên đáng kể. Nhưng đối với tốc độ phát triển của vi mạch, điều đó không đáng quan tâm lắm.

13-15. BỘ NHÂN NHỊ PHÂN DÙNG ROM

Phép nhân các số nhị phân có thể thực hiện bằng cách dùng các mạch ROM. Trong phương pháp này ROM được sử dụng như một bảng tra cứu, các đầu vào của địa chỉ của ROM sẽ tương ứng với số liệu của số nhân và số bị nhân, các đầu ra số liệu của ROM chính là kết quả của phép nhân. Ví dụ : Phép nhân 2 số nhị phân 4 bit dùng ROM có 8 đường địa chỉ, trong đó 4 đường cho số bị nhân và 4 đường cho số nhân. 8 đường số liệu ra của ROM là kết quả của phép nhân này. Dung lượng của ROM cần thiết là : $2^8 \times 8 \text{ bit} = 256 \text{ bit}$. Nội dung tương ứng của 1 ô nhớ trong ROM là tích số của 1 phép nhân. Sơ đồ minh họa ROM 8 đường địa chỉ và 8 đường số liệu để thực hiện phép nhân 2 số nhị phân 4 bit như hình 13-18.



Hình 13-18. Dùng ROM làm bộ nhân.

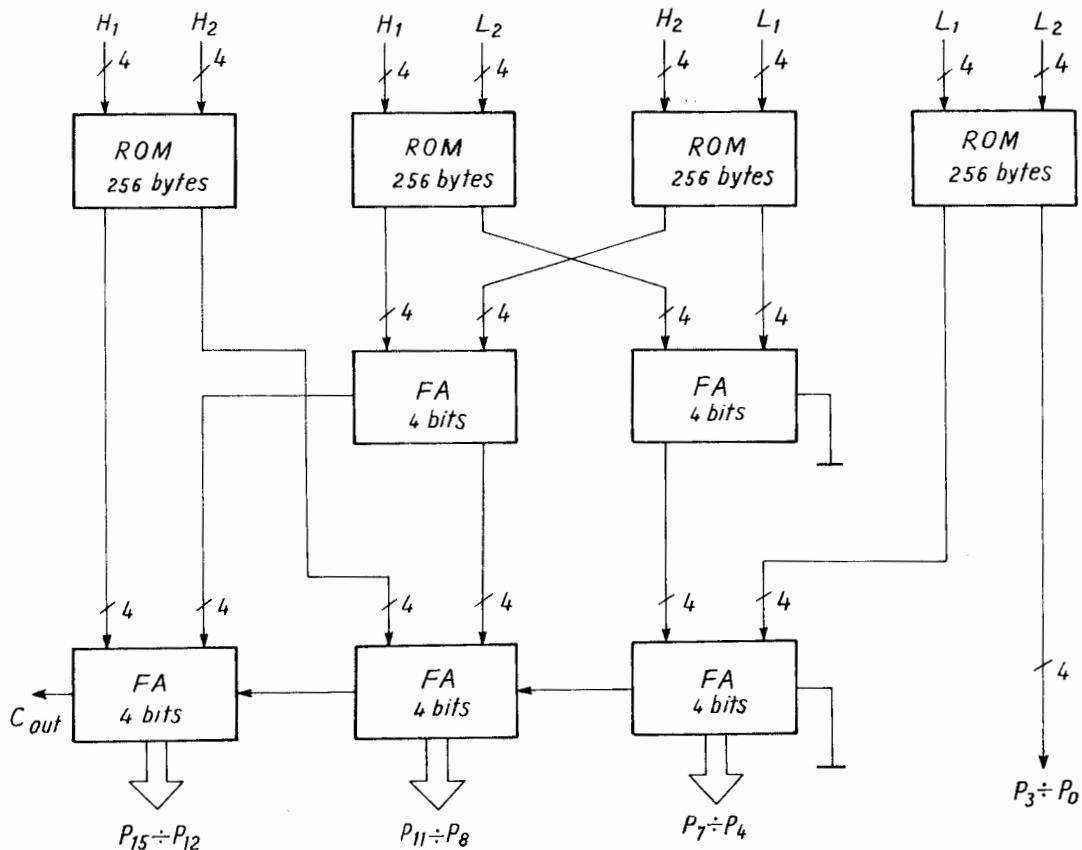
Nếu dùng 1 ROM để thực hiện phép nhân giữa 2 số thì dung lượng cần thiết sẽ tăng khi số bit của số nhân và số bị nhân tăng. Tuy nhiên có thể thực hiện phép nhân 2 số nhị phân bằng cách dùng nhiều ROM có dung lượng nhỏ hơn. Ví dụ : Thực hiện phép nhân 2 số nhị phân 8 bit là N_1 và N_2 . Nếu sử dụng một ROM để thực hiện phép nhân số N_1 , N_2 đó thì phải dùng ROM có 16 đường địa chỉ và 16 đường số liệu, dung lượng của ROM này là : $2^{16} \times 16 \text{ bit} = 2^{28} \text{ bit}$. Một khía cạnh khác có thể biểu diễn N_1 , N_2 dưới dạng:

$$\text{Với } N_1 = \underbrace{N_{18} N_{17} N_{16} N_{15}}_{H_1} \quad \underbrace{N_{14} N_{13} N_{12} N_{11}}_{L_1} = L_1 + 2^4 H_1$$

$$N_2 = \underbrace{N_{28} N_{27} N_{26} N_{25}}_{H_2} \quad \underbrace{N_{24} N_{23} N_{22} N_{21}}_{L_2} = L_2 + 2^4 H_2$$

$$N_1 \cdot N_2 = (L_1 + 2^4 H_1)(L_2 + 2^4 H_2) = 2^8 H_1 H_2 + 2^4 L_1 H_2 + 2^4 L_2 H_1 + L_1 L_2$$

Sơ đồ mạch thực hiện phép nhân này dùng các ROM 8 đường địa chỉ 8 đường số liệu được minh họa ở hình 13-19.



Hình 13-19. Mạch thực hiện phép nhân 2 số 8 bit dùng ROM : 256 byte.

13-16. BỘ NHÂN NHỊ PHÂN DÙNG PHƯƠNG PHÁP DỊCH VÀ CỘNG

Để đơn giản và dễ hiểu, trước hết chúng ta xét trường hợp nhân 2 số nhị phân 4 bit. Trong trường hợp số nhân và số bị nhân có số bit (số chữ số) lớn hơn cũng tiến hành hoàn toàn tương tự. Giả sử có 2 số nhị phân 4 bit \$A\$ và \$B\$. Trong đó \$A = 1110\$; \$B = 1010\$. Ta tiến hành tìm tích của chúng :

Số bị nhân	\$A\$:	1110
Số nhân	\$B\$:	1010
<hr/>		
Tính bộ phận	\$P_0\$	0000
	\$P_1\$	1110
	\$P_2\$	0000
	\$P_3\$	1110
<hr/>		
		10001100

\$P_i\$ (\$i = 0 \div 3\$) là tính bộ phận của bước thứ \$i\$.

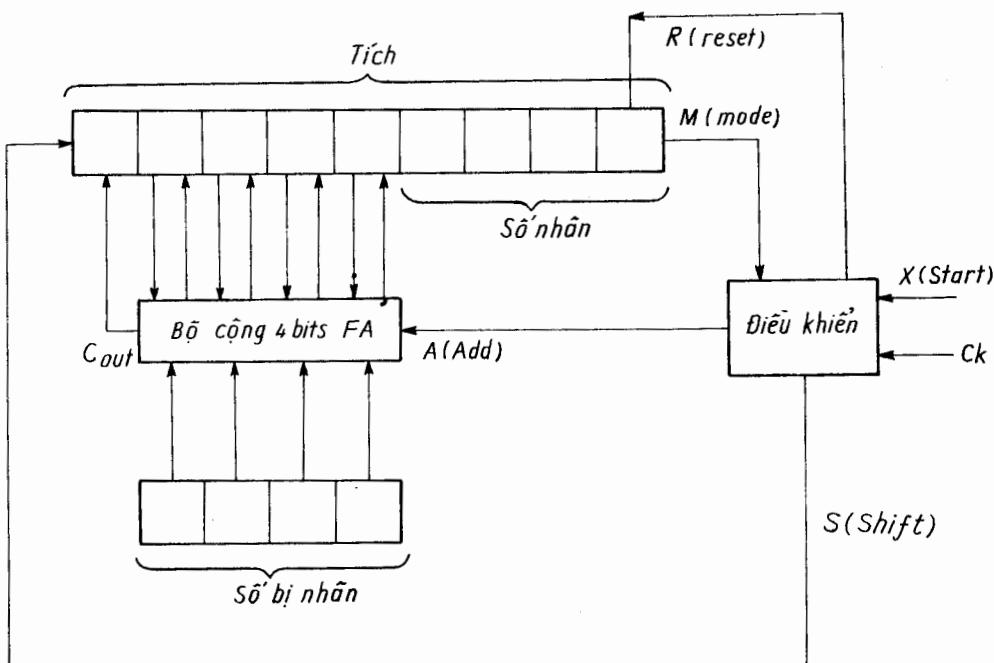
Thuật toán được trình bày như sau :

1. Phải xác định được chữ số thứ i của số nhân là 0 hay 1. Nếu là 1 thì P_i bằng số bị nhân ($P_i = A$). Nếu là 0 thì $P_i = 0$
2. Lần lượt tạo ra n tinh bộ phận P_i ($i = 0 \div n-1$) trong đó n là số bit (số chữ số) của số nhân.
3. Trong phép nhân tiến hành từ cột có trọng số nhỏ nhất thì sau mỗi bước tìm tích bộ phận, tích bộ phận đó được dịch trái 1 bit (1 vị trí) quá trình cứ tiếp tục như vậy cho đến bước thứ ($n - 1$) thì dừng lại và lấy tổng các tích bộ phận.

* Tuy nhiên chúng ta cũng có thể thực hiện như sau :

Cứ sau mỗi bước tìm bộ phận thì tích bộ phận đó được dịch trái 1 bit và ta cộng tích các bộ phận vừa tìm được với tổng các tích bộ phận đã tìm được trước đó.

Sơ đồ tiêu biểu thực hiện theo phương pháp này được mô tả ở hình 13-20



Hình 13-20. Sơ đồ bộ nhân dùng phương pháp dịch cộng 4 bit.

Sơ đồ bao gồm:

Bộ ghi dịch 9 bit dùng để ghi kết quả phép nhân (ban đầu số nhân được ghi ở 4 bit bên phải của bộ ghi này).

- Bộ ghi số nhân 4 bit;
- Bộ cộng nhị phân song song nhớ nhanh 4 bit;
- Bộ điều khiển phép nhân.

Bộ điều khiển phép nhân này có 4 đầu vào là :

- Xung nhịp **CL** để đồng bộ hoạt động của nó.
- Tín hiệu khởi động **x (Start)**.
- Tín hiệu đưa từ bit **M** của số nhân (bit thứ i với $i = 0 \div 3$)

Các đầu ra là :

- Xung điều khiển dịch s để điều khiển bộ ghi dịch 9 bits.
- Xung cộng A^* để điều khiển bộ cộng FA.
- Xung thiết lập lại R để xoá bit M về 0 khi cần thiết.

Nguyên lý hoạt động của mạch như sau :

Số bị nhân và số nhân được nạp vào các bộ ghi như đã nói trên. Khi có xung X cho phép bắt đầu, và xung nhịp C_k đưa vào, nếu bit M tức là bit trẻ nhất của số nhân bằng 1 ($M = 1$) thì khối điều khiển đưa ra xung điều khiển cộng A^K . Đồng thời giá trị số bị nhân được đưa vào bộ cộng để cộng với 4 bit (5, 6, 7, 8) của bộ ghi dịch 9 bit. Sau đó kết quả được gửi trở lại 5 bit (5, 6, 7, 8, 9) của bộ ghi dịch. Khi có xung nhịp C_k tiếp theo thì bộ điều khiển đưa ra 1 xung dịch s để điều khiển bộ ghi dịch 9 bit về phải 1 bit.

Nếu $M = 0$ thì bộ điều khiển chỉ đưa ra xung dịch s ở thời điểm xung nhịp thứ 2 chứ không đưa tín hiệu điều khiển cộng. Và cũng tại thời điểm xung nhịp thứ 1 bộ điều khiển đưa ra tín hiệu R để xoá bit M về 0 sau khi giá trị bit M đã được đưa vào bộ điều khiển.

Khi có xung nhịp thứ 3 quá trình lại thực hiện giống như quá trình đầu. Quá trình thực hiện phép nhân dừng lại khi tất cả các bit của số nhân đã được xoá hết và kết quả phép nhân được gửi lên bộ ghi dịch 9 bit.

Ở đây bộ điều khiển nhân thực chất là 1 mạch dãy đồng bộ với xung nhịp C_k . Chúng ta có thể thiết kế bộ điều khiển nhân như sau :

Thiết kế bộ điều khiển nhân dùng JK-FF và các mạch NAND :

Như đã nêu trên, bộ điều khiển nhân này có 3 đầu vào và 3 đầu ra và Chức năng của nó cũng đã được nói ở phần trên. Xem hình 13-20,a.

- Cần cứ vào chức năng của mạch điều khiển nhân chúng ta có thể lập được sơ đồ chuyển đổi trạng thái như hình 13-21,a.

- Nhìn vào bảng đồ hình chuyển đổi trạng thái, dễ dàng nhận thấy các trạng thái đã tối thiểu.

- Để mã hoá các trạng thái mạch cần phải dùng 3 biến nhị phân ABC .

Áp dụng hai qui tắc mã hoá trạng thái cho mạch dãy đồng bộ, mã hoá trạng thái:

Các trạng thái được mã hoá theo 2 qui tắc như sau :

- Hệ phương trình của mạch :

$$T_{onA} = S_2 \bar{M} + S_3 = \bar{A} B C \bar{M} + \bar{A} B \bar{C} = \bar{A} B (C \bar{M} + \bar{C}) = \bar{A} B (\bar{M} + C)$$

$$\Rightarrow J_A = B \bar{C} + B \bar{M}$$

$$T_{offA} = S_6 \bar{M} + S_7 = A \bar{B} C \bar{M} + A \bar{B} \bar{C} = A \bar{B} (C \bar{M} + \bar{C}) = A \bar{B} (\bar{M} + \bar{C})$$

$$\Rightarrow K_A = \bar{B} \bar{C} + \bar{B} \bar{M}$$

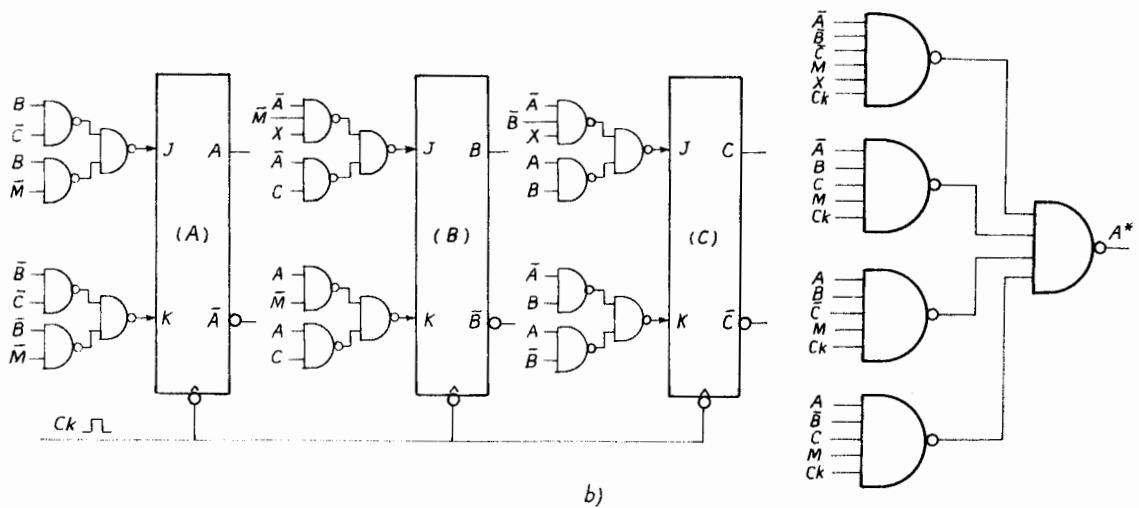
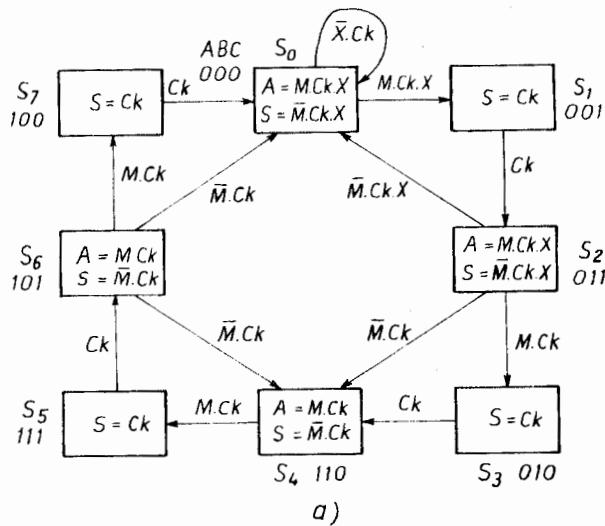
$$T_{onB} = S_0 \bar{M}X + S_1 = \bar{A} \bar{B} \bar{C} \bar{M}X + \bar{A} \bar{B} C = B[\bar{A}(\bar{C} \bar{M}X + C)] = \bar{B} \bar{A}(\bar{M}X + C)$$

$$\Rightarrow J_B = \bar{A} \bar{M}X + \bar{A} C$$

$$T_{offB} = S_4 \bar{M} + S_5 = A B \bar{C} \bar{M} + A B C$$

$$\Rightarrow K_B = A \bar{M} + A C$$

$$\begin{aligned}
 T_{\text{onC}} &= S_0 MX + S_0 \bar{M} X + S_1 M + S_4 \bar{M} = S_0 X + S_4 = \bar{A}\bar{B}\bar{C} X + \bar{ABC} \\
 \Rightarrow J_C &= ABX + AB \\
 T_{\text{offC}} &= S_2 M + S_2 \bar{M} + S_6 \bar{M} + S_6 M = S_2 + S_6 = (\bar{A}B + A\bar{B}) C \\
 \Rightarrow K_2 &= AB + A\bar{B} \\
 A^* &= [\bar{A}\bar{B}\bar{C} MX + \bar{ABC} M + ABC\bar{C} M + ABC\bar{C} M] \\
 &= M[\bar{A}\bar{B}\bar{C} X + \bar{ABC} + ABC\bar{C} + ABC\bar{C}] \\
 S &= (S_1 + S_3 + S_5 + S_7) + S_0 \bar{M} X + S_2 \bar{M} + S_4 \bar{M} + S_6 \bar{M}
 \end{aligned}$$



Hình 13-21.a) Đồ hình trạng thái của bộ điều khiển nhân; b) Sơ đồ logic của bộ điều khiển phép nhân.

13-17. MỘT SỐ VI MẠCH

13-17.1. GIỚI THIỆU MỘT SỐ VI MẠCH ĐIỂN HÌNH VÀ CHỨC NĂNG CỦA NÓ :

- Bộ cộng đầy đủ 1 bit 80
- Bộ cộng đầy đủ 2 bit 82
- Bộ cộng đầy đủ 4 bit 83A, LS83A
- Hai bộ cộng đầy đủ 1 bit 283, S283, LS283, H183, LS183
- Bốn mạch cộng/trừ nối tiếp LS385
- Bộ số học logic 4 bit 181, S181, LS181, AS181, LS381, AS381
- Tạo số nhớ nhanh S182, 182, AS182

13-17.2. PHÂN TÍCH CỤ THỂ MỘT VÀI VI MẠCH

a) 74182 Bộ tạo nhớ nhanh

Các đầu vào : G_0, G_1, G_2, G_3 :

Các đầu vào tạo nhớ : P_0, P_1, P_2, P_3 :

Đầu vào nhớ. C_n

Các đầu ra : $C_{n+x}, C_{n+y}, C_{n+z}$:

Các đầu ra nhớ. : G đầu ra tạo nhớ. P : đầu ra truyền nhớ.

Phương trình logic của đầu ra có dạng:

* Dữ kiện tác động ở mức thấp :

$$\begin{aligned}C_{n+x} &= G_0 + P_0 C_n \\C_{n+y} &= G_1 + P_1 G_n + P_1 P_0 C_n \\C_{n+z} &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_n\end{aligned}$$

$$\begin{aligned}\overline{G} &= \overline{G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0} \\P &= \overline{P_3 P_2 P_1 P_0}\end{aligned}$$

* Dữ kiện tác động ở mức cao :

$$\overline{C}_{n+x} = \overline{Y_0 (X_0 + C_n)}$$

$$\overline{C}_{n+y} = \overline{Y_2 \{X_2 + Y_1 [X_1 + Y_0 (X_0 + C_n)]\}}$$

$$\overline{C}_{n+z} = \overline{Y_2 \{X_2 + Y_1 [X_1 + Y_0 (X_0 + C_n)]\}}$$

$$Y = Y_3 (X_3 + Y_2) (X_3 + X_2 + Y_1) (X_3 + X_2 + X_1 + Y_0)$$

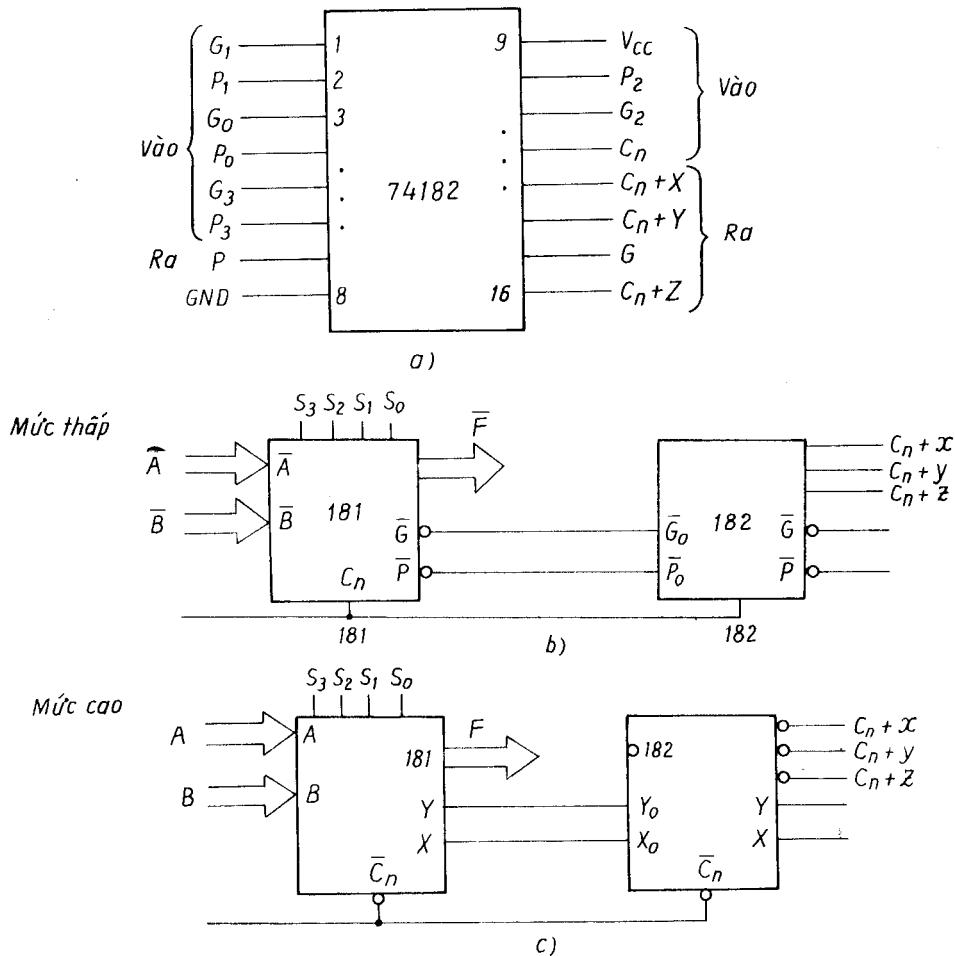
$$X = X_3 + X_2 + X_1 + X_0$$

Sơ đồ chân của vi mạch được cho ở hình 13-22, a.

b) 74181 :

74181 là vi mạch thực hiện chức năng của 1 đơn vị số học – logic (ALU). Nó có thể thực hiện 16 phép tính số học và 16 phép tính logic. Có 4 đầu chọn chức năng là : $S_0 S_1 S_2 S_3$. 74181 có thể tính toán trên 2 từ mã 4 bit tạo số nhớ nhanh bằng cách kết hợp với vi mạch 74182 (hình 13-22,b và c).

- + Dữ kiện tác động ở mức thấp : Hình 13 - 22,b.
- + Dữ kiện tác động ở mức cao : Hình 13 - 22,c.



Hình 13-22. Sơ đồ 74182.

BÀI TẬP

13.1. Thiết kế mạch logic tổ hợp thực hiện biến đổi số nhị phân 4 bits thành mã bù 2 tương ứng.

13.2. Thiết kế mạch logic tổ hợp thực hiện biến đổi số ở dạng mã BCD-N thành mã bù 9 tương ứng.

13.3. Thiết kế mạch logic chỉ dùng các mạch *NAND* 2 đầu vào thực hiện sự biến đổi mã BCD-N thành mã dư 3 (XS3) tương ứng.

13.4. Cho bảng mã dư 3 (XS3) tương ứng với 10 chữ số thập phân, hãy

a) Xây dựng thuật toán để cộng 2 số thập phân dùng mã XS3.

b) Sử dụng mã bù 9 để thực hiện phép toán : $(34)_{10} - (19)_{10}$

13.5. Thiết kế mạch cộng / trừ số thập phân dùng mã XS3 cho 3 chữ số thập phân dùng bộ cộng 7483 4 bit nhị phân (dùng mã bù 9).

13.6. Xây dựng thuật toán cho phép cộng 2 số dương cơ số 12. Với thuật toán này thiết kế bộ cộng / trừ 2 số cơ số 12, mã nhị phân ở cột thứ i nào đó dùng bộ cộng 7483 4 bits nhị phân và các mạch phụ.

13.7. Số thập phân được biểu diễn bằng mì 2-4-2-1 cho ở bảng trên hình 13-24.

a) Thiết kế thuật toán cho phép cộng 2 số thập phân biểu diễn bởi mã này.

b) Với thuật toán này xây dựng sơ đồ cho việc cộng 2 số thập phân biểu diễn bởi mã này, ở 1 cột thứ i nào đó dùng bộ cộng 7483 4 bit nhị phân và các mạch phụ.

13.8. Xây dựng mạch nhân 2 số nhị phân, 1 số 4 bit và 1 số 3 bit dùng các mạch AND và bộ cộng đầy đủ.

13.9. Một bộ nhân 2 số nhị phân dương 4 bit dùng phương pháp dịch và cộng có sơ đồ khối được biểu diễn như hình 13-25. Phép nhân được điều khiển bởi khối điều khiển.

Khối này có các đầu vào là :

- Xung nhịp clock
- Tín hiệu khởi động Start.
- Bit *M* của số nhân.

Số thập phân	Mã dư 3 (XS3)			
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Hình 13-23. Cho bài tập 13-4.

Số thập phân	Mã 2-4-2-1			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

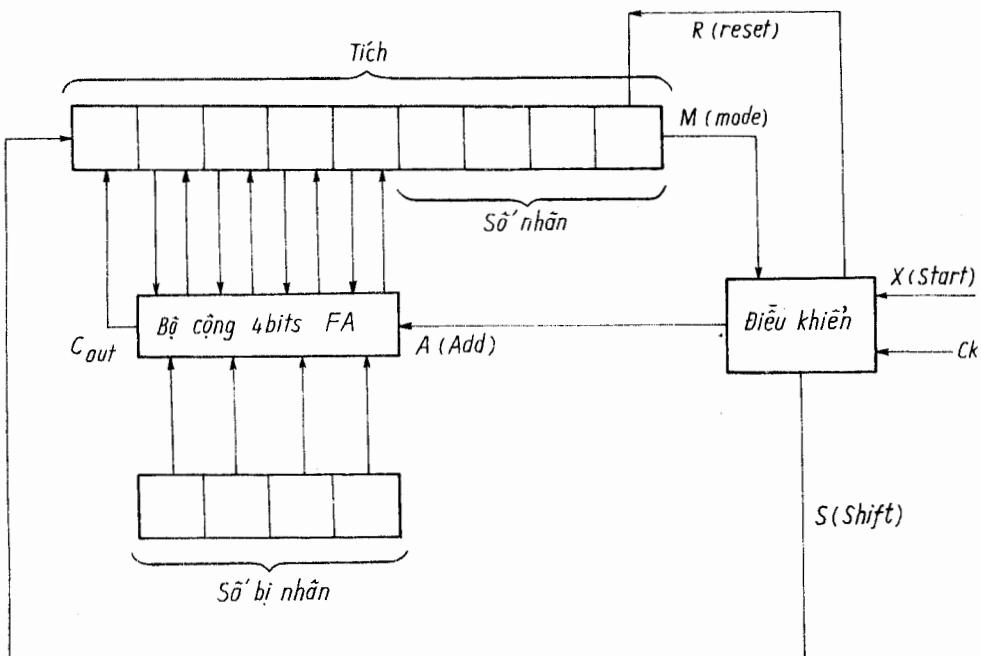
Hình 13-24. Cho bài tập 13-7.

Các đầu ra là :

- Xung dịch S
- Xung cộng A
- Xung thiết lập lại Reset.

Khối điều khiển phải thiết kế sao cho nếu $M = 1$ và $C_k = 1$ đưa ra xung cộng A , xung thiết lập lại R xoá M về 0. Khi có xung nhịp tiếp theo tạo xung dịch S .

Dùng phương pháp thiết kế mạch đồng bộ thiết kế bộ nhân này.



Hình 13-25. Cho bài tập 13.9.

ĐÁP SỐ CỦA BÀI TẬP

Chương 1

1.1. Giá trị đèn báo là Y với $Y = 1$ đèn sáng

$$Y = AB + BC + AC$$

Với các công tắc báo bù phiếu là A, B, C . Chúng có giá trị 1 nếu công tắc đóng, ngược lại thì là 0.

1.3. $F_{a_3} = A \bar{B} \bar{C} + \bar{A} B \bar{C} + \bar{A} \bar{B} C + ABC$

$$\begin{aligned} F_{a_4} = & A \bar{B} \bar{C} D + \bar{A} B \bar{C} \bar{D} + \bar{A} \bar{B} C \bar{D} + A B C \bar{D} + \\ & \bar{A} \bar{B} \bar{C} D + \bar{A} B C D + A \bar{B} C D + A B \bar{C} D \end{aligned}$$

$$F_{e_3} = A \bar{B} \bar{C} + \bar{A} B \bar{C} + \bar{A} \bar{B} C + ABC$$

$$\begin{aligned} F_{e_4} = & A \bar{B} \bar{C} D + \bar{A} B \bar{C} D + \bar{A} \bar{B} C D + ABCD + \bar{A} \bar{B} \bar{C} \bar{D} + \\ & A \bar{B} C \bar{D} + \bar{A} B C \bar{D} + A B \bar{C} \bar{D} \end{aligned}$$

1.5. $A \oplus B \oplus C = \bar{A} \oplus (\bar{B} \oplus \bar{C}) = \bar{A} \oplus (B \sim C) = \overline{A \oplus (B \sim C)} = A \sim B \sim C$

1.6. $\bar{Y} = \bar{A} \bar{B} + \bar{A} C + D$

1.8. a) $Y = \bar{A} + B \bar{C}$

b) $Y = \bar{A} + \bar{B} + CD$

c) $Y = \bar{A} \bar{B} + \bar{A} C + AD$

d) $Y = B + A\bar{C} + \bar{A} C$

e) $Y = ABC + (\bar{A} + \bar{B}) D$

1.10. $A = 1, B = C = 0, D = 1$

Chương 2

2.1. $f_{min1}(X_3, X_2, X_1, X_o) = \bar{X}_l \bar{X}_o + X_2 X_o + \bar{X}_2 X_1$

$$f_{min2}(X_3, X_2, X_1, X_o) = X_2 \bar{X}_l + X_l X_o + \bar{X}_2 \bar{X}_o$$

2.2. $f_{min1}(X_3, X_2, X_1, X_o) = (X_3 + X_1 + X_o)(X_3 + \bar{X}_l)(X_2 + X_o)$

2.3. $f_{min1}(X_3, X_2, X_1, X_o) = \bar{X}_3 \bar{X}_2 \bar{X}_l + \bar{X}_3 X_2 X_o + X_3 X_2 X_1 + X_2 X_l \bar{X}_o$

$$f_{min2}(X_3, X_2, X_1, X_o) = \bar{X}_3 \bar{X}_2 \bar{X}_o + \bar{X}_3 \bar{X}_l X_o + X_2 X_l X_o + X_3 X_l \bar{X}_o$$

2.4. $f_{min}(X_4, X_3, X_2, X_1, X_o) = \bar{X}_2 \bar{X}_l + X_3 \bar{X}_l$

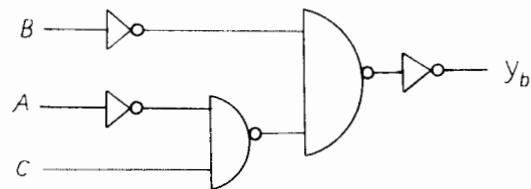
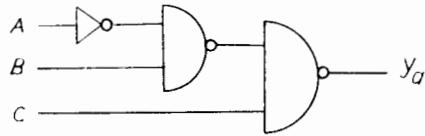
$$2.5. \quad Y_a = \overline{C} + \overline{A} B$$

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

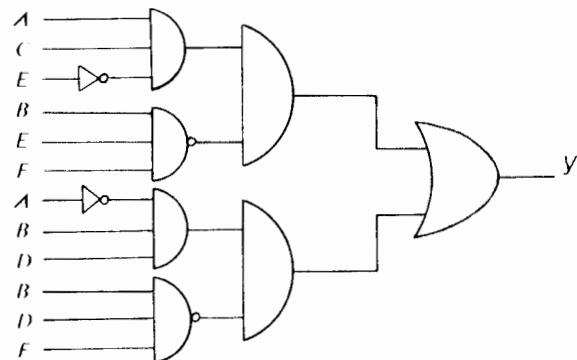
Chương 5

$$5.1. \quad Y_a = \overline{C} + \overline{A} B$$

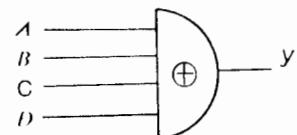
$$Y_b = \overline{B} (A + \overline{C})$$



$$5.2. \quad Y = A C \overline{E} \cdot \overline{B E F} + \overline{A} B D \cdot \overline{B D F}$$



$$5.3. \quad Y = A \oplus B \oplus C \oplus D$$



$$5.4. \quad Y = \overline{A} D + \overline{A} B + B \overline{C}$$

$$5.5. \quad Y_1 = \overline{B D} \quad \overline{A} B \overline{C} \quad \overline{A} \overline{B} C D$$

$$Y_2 = \overline{A D} \quad A \overline{B} \overline{C} \quad \overline{A} \overline{B} C D$$

$$5.6. \quad Y_1 = B \cdot C \cdot D + A \cdot \overline{D}$$

$$Y_2 = A \cdot B \cdot \overline{D} + BCD + \overline{A} \cdot \overline{B} \cdot \overline{D}$$

$$Y_3 = A \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot \overline{D}$$

$$5.7. \quad W = A + B \cdot (C + D) \quad A = W \cdot (X + YZ)$$

$$X = \overline{B} \cdot (C + D) + B \cdot \overline{(C + D)} \quad B = \overline{X} \cdot (\overline{Y} \cdot \overline{Z}) + XYZ$$

$$Y = \overline{(C + D)} + CD \quad C = \overline{Y} \cdot \overline{Z} \cdot (Y + Z)$$

$$Z = \overline{D} \quad D = \overline{Z}$$

$$5.8. \quad \begin{array}{ll} C \cdot E \cdot F \cdot P \cdot T & : Y_1 \Rightarrow Y_1 = A + C \\ C \cdot E \cdot F \cdot P & : Y_2 \Rightarrow Y_2 = C \\ E \cdot F \cdot P \cdot T & : Y_3 \Rightarrow Y_3 = A + B \\ C \cdot E \cdot T & : Y_4 \Rightarrow Y_4 = A \cdot \overline{C} + \overline{A} \cdot C \\ C \cdot E & : Y_5 \Rightarrow Y_5 = \overline{A} \cdot C \\ P & : Y_6 \Rightarrow Y_6 = A \cdot B \\ T & : Y_7 \Rightarrow Y_7 = A \cdot \overline{C} \end{array}$$

Với ABC là 3 bit nhị phân mã hóa cho 5 biến C, E, F, P .

5.9. Mạch tạo bit chẵn dùng hàm cộng Modul 5 biến

- Đầu đưa ra bit chỉ thị chẵn :

$$P = [(B \oplus C) \oplus A] \oplus (E \oplus D) = A \oplus B \oplus C \oplus D \oplus E$$

- Đầu báo hiệu chẵn (hàm cộng Modul 6 biến)

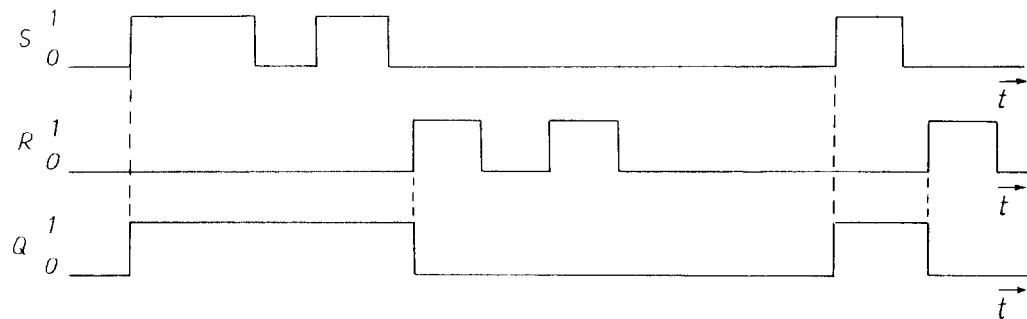
$$H = A \oplus B \oplus C \oplus D \oplus E \oplus P$$

Chương 6

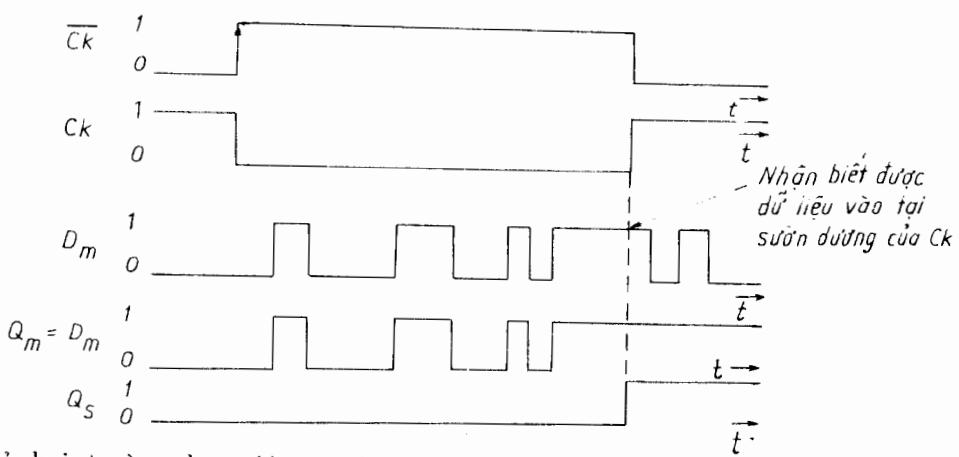
$$6.3. \quad (i) \quad a = 1, b = 0, c = 1$$

$$(ii) \quad a = 0, b = 1, c = 1$$

$$(iii) \quad a = 1, b = 0, c = 0$$



6.4.

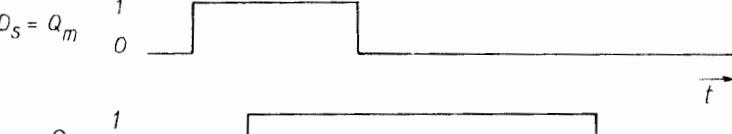
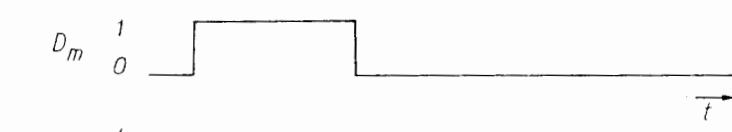
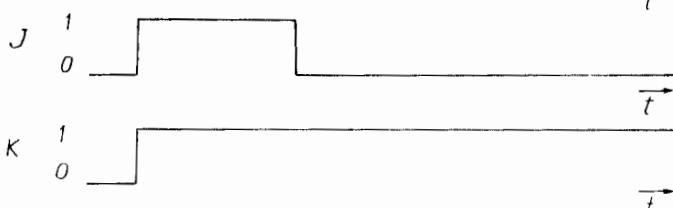
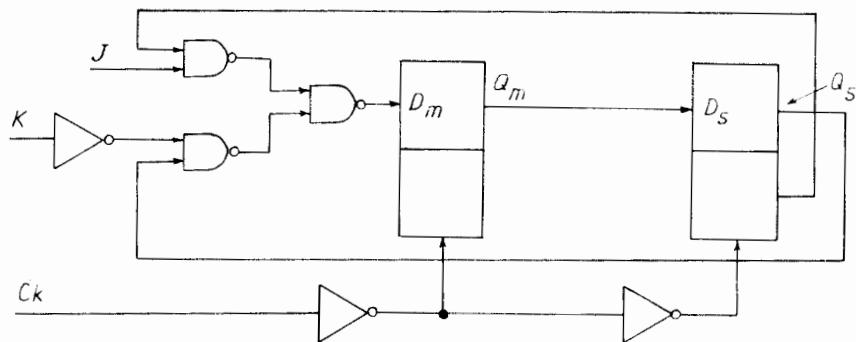


6.5. Cả hai trường hợp đều có dạng

$$(\bar{x}Y + x\bar{Y}) \text{ AND}$$



6.6.



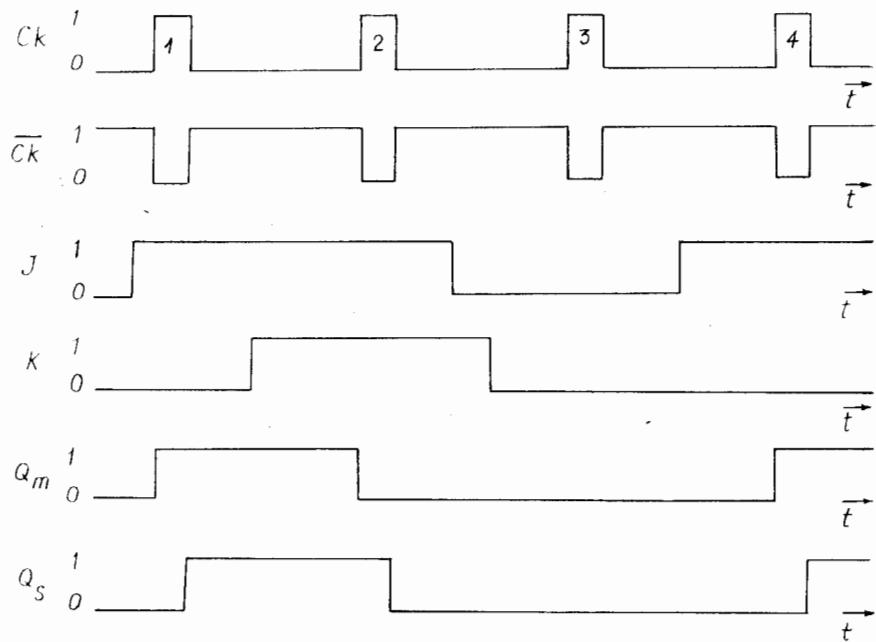
$$6.7. \quad Q^t + \delta t = (\bar{J} \cdot \bar{Q} + \bar{K}Q)^t$$

6.8. (i) tại sườn dương của xung nhịp thứ nhất

- (a) 1 → 0, (b) 0 → 1, (c) 1 → 1 (d) 0 → 0
 (e) 1 → 1, (f) 1 → 0 (g) 0 → 1 (h) 1 → 1

Tại sườn âm của xung nhịp thứ nhất

- (a) 0 → 1, (b) 1 → 1, (c) 1 → 0, (d) 0 → 1,
 (e) 1 → 1, (f) 0 → 0 (g) 1 → 1, (h) 1 → 0.

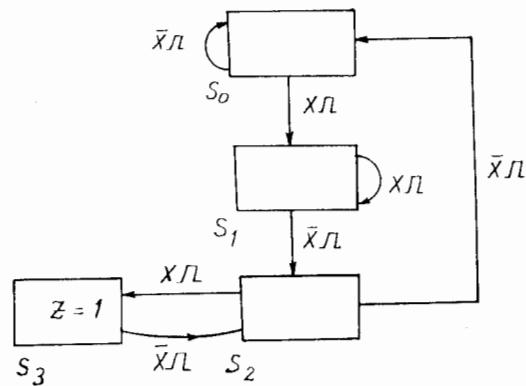


$$(v) \quad Q_s = 1$$

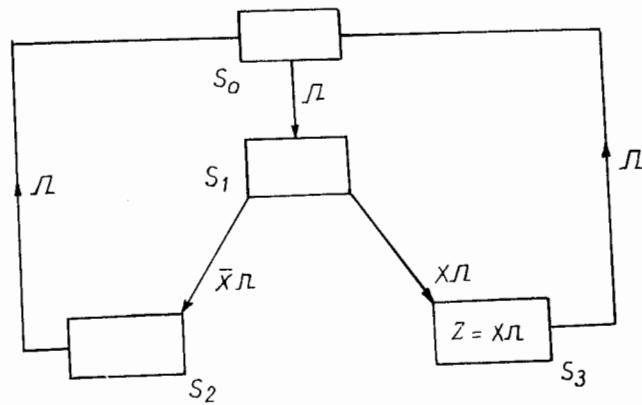
$$(vi) \quad Q_s = 1$$

Chương 7

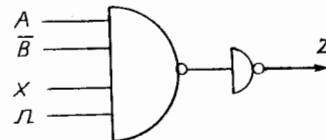
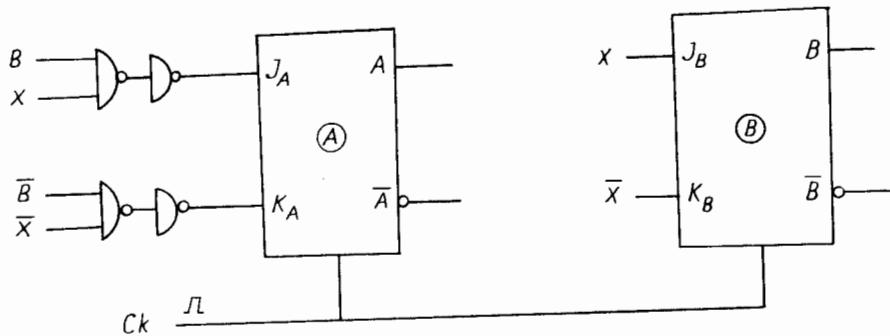
7.1.



7.2.

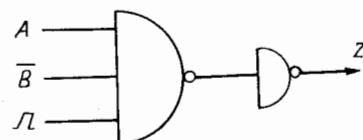
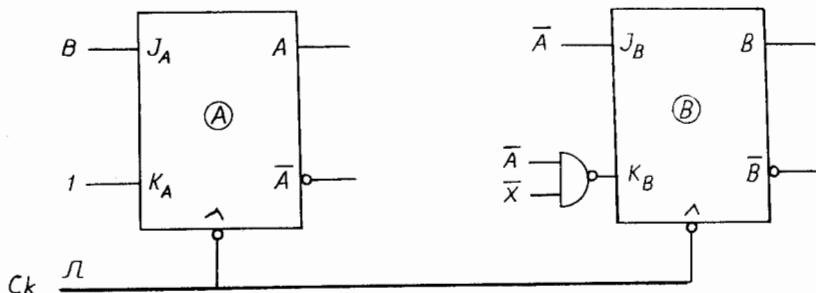


7.3.

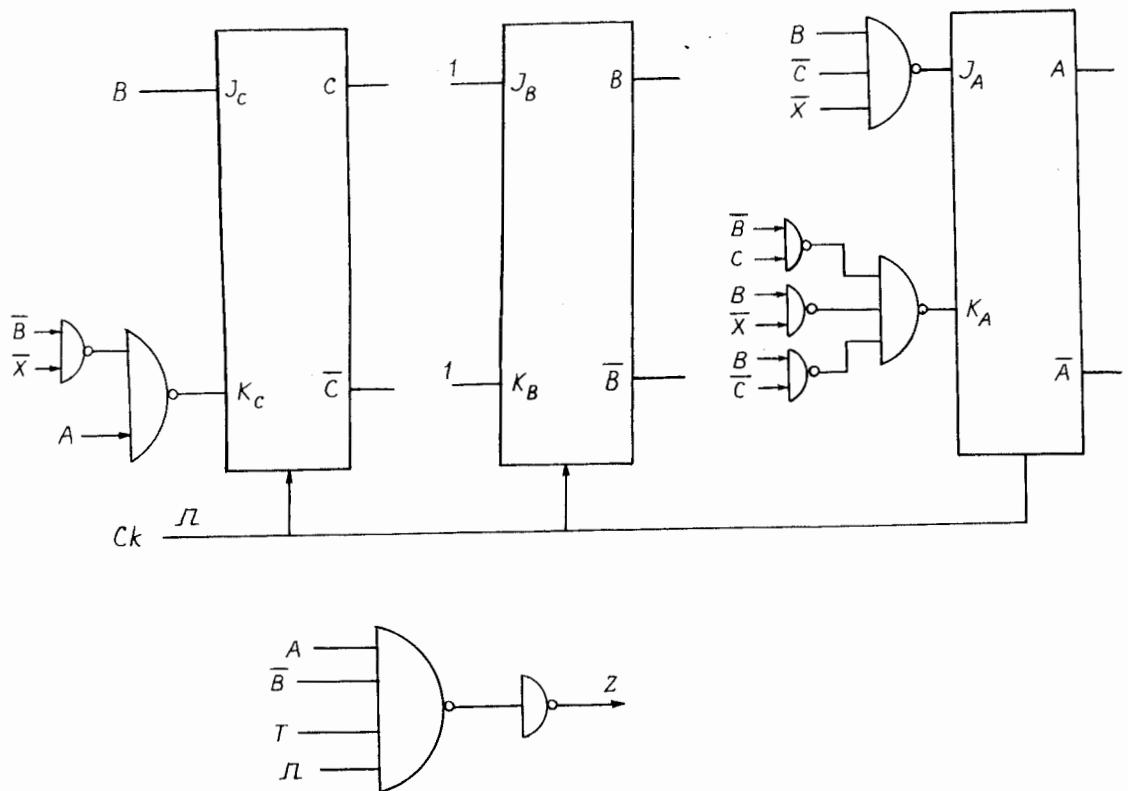


7.4. Với mã hóa các trạng thái theo hai biến AB như sau :

$S_0(00)$, $S_1(01)$, $S_2(10)$, $S_3(11)$



7.5. Ứng với trường hợp mã hóa cho các trạng thái dùng 3 biến nhị phân LBA như sau : $S_0(111)$, $S_1(100)$, $S_2(011)$, $S_3(000)$, $S_4(101)$, $S_5(110)$, $S_6(001)$, $S_7(010)$ ta có mạch thực hiện :



Chương 8

8.1. D là bit có trọng số lớn nhất của bộ đếm

$$\begin{aligned}
 T_D &= ABC + ABD, \quad T_C = A\bar{B}\bar{D}, \quad T_B = A, \quad T_A = 1, \quad S_D = ABC, \\
 R_D &= ABD \text{ or } BAC\bar{C}, \quad S_C = A\bar{B}\bar{C} \bar{D}, \quad R_C = ABC, \quad S_B = A\bar{B}, \\
 R_B &= AB, \quad S_A = \bar{A}, \quad R_A = A, \quad J_D = ABC, \quad K_D = AB, \quad J_C = A\bar{B}\bar{D}, \\
 K_C &= AB, \quad J_B = A, \quad K_B = A, \quad J_A = 1, \quad K_A = 1, \quad D_D = \bar{B}D + \bar{A}D + ABC, \\
 D_C &= \bar{B}C + \bar{A}C, \quad AB \bar{C} \bar{D}, \quad D_B = A\bar{B} + \bar{A}B, \quad D_A = \bar{A}.
 \end{aligned}$$

8.2. C là bit có trọng số lớn nhất của mạch dây

$$J_C = \bar{A} + \bar{B}, \quad K_C = \bar{A}, \quad J_B = C, \quad K_B = \bar{C}, \quad J_A = B, \quad K_A = \bar{B}$$

Trạng thái "khóa" là $(CBA) = 111$

8.3. D là bit có trọng số lớn nhất của bộ đếm

$$\begin{aligned}
 J_D &= RS, \quad K_D = \bar{Q}, \quad J_C = \bar{P}, \quad K_Q = P\bar{R}S, \quad J_B = \bar{P}S, \quad K_B = PS; \\
 J_A &= \bar{P}Q\bar{R} + PR; \quad K_A = P\bar{R} + \bar{P}R
 \end{aligned}$$

8.4. $Z^t Z^{t+\delta t}$ VU. C là bit có trọng số lớn nhất của bộ đếm

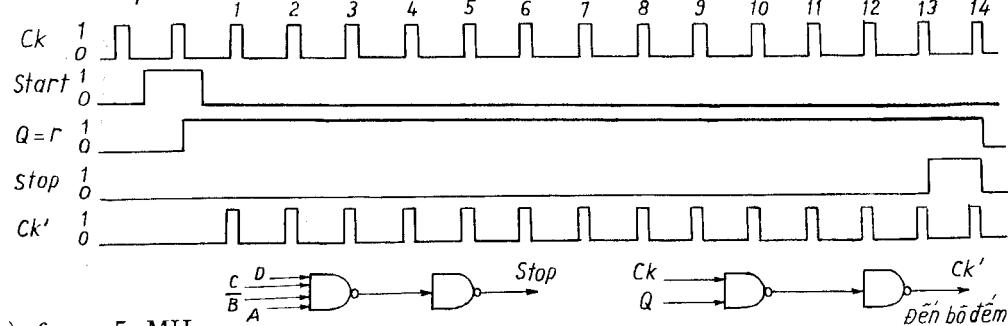
$$0 \quad 0 \quad 1 \quad \phi \quad P_C = \bar{A} + \bar{B}, \quad Q_C = AB, \quad P_B = \bar{A}, \quad Q_B = A,$$

$$0 \quad 1 \quad 0 \quad \phi \quad P_A = 0 \quad Q_A = 1$$

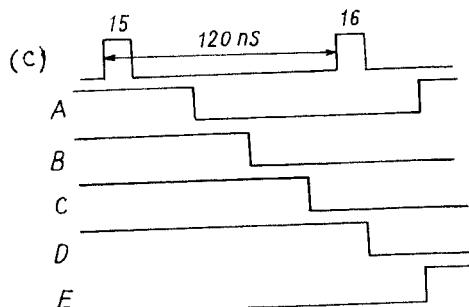
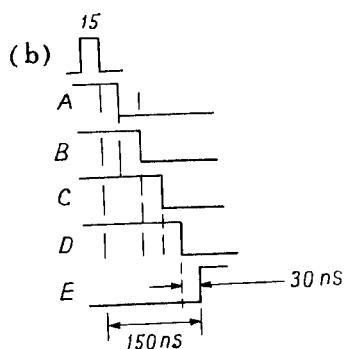
$$1 \quad 0 \quad \phi \quad 1$$

$$1 \quad 1 \quad \phi \quad 0$$

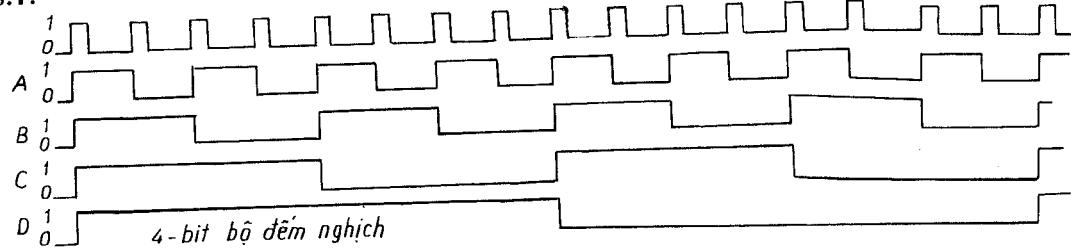
8.5



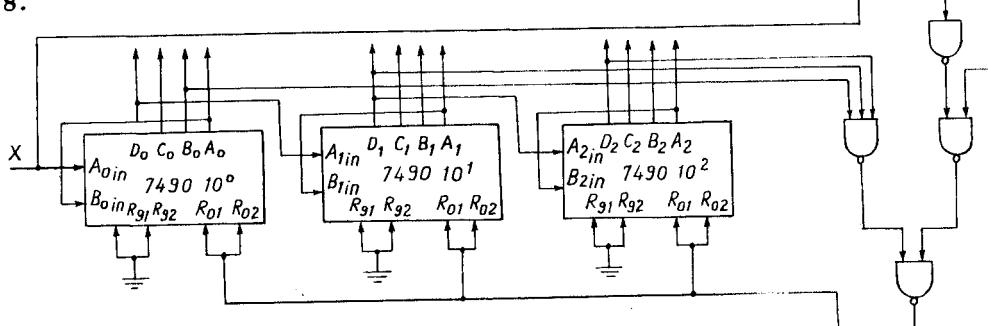
8.6. (a) $f_u = 5$ MHz



8.7.

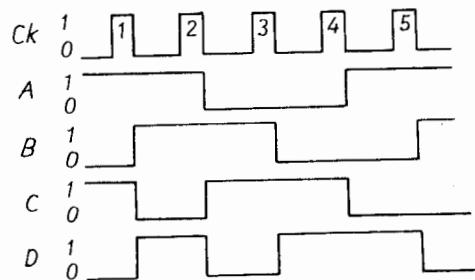


8.8.



Chương 9

9.1.



9.2. Chọn dãy chuyển đổi như sau :

$$S_0 - S_1 - S_2 - S_5 - S_{11} - S_7 - S_{15} - S_{14} - S_{13} - S_{10} - S_4 - S_8 - S_6$$

Hàm hồi tiếp (A là FF biểu diễn có trọng số nhỏ nhất trong bộ ghi dịch) :

$$f = B\bar{D} + A\bar{C}\bar{D} + \bar{A}\bar{B}C + A\bar{C}\bar{D} + \bar{A}\bar{C}\bar{D}, \quad o = \bar{A}\bar{B}\bar{C}\bar{D}\bar{D},$$

$$1 = \bar{A}\bar{C}\bar{D}, \quad 2 = \bar{B}\bar{C}\bar{D}, \quad 3 = \bar{A}\bar{B}\bar{C}\bar{D}, \quad 4 = \bar{A}\bar{C}\bar{D}, \quad 5 = A\bar{B}\bar{D}, \quad 6 = ABCD,$$

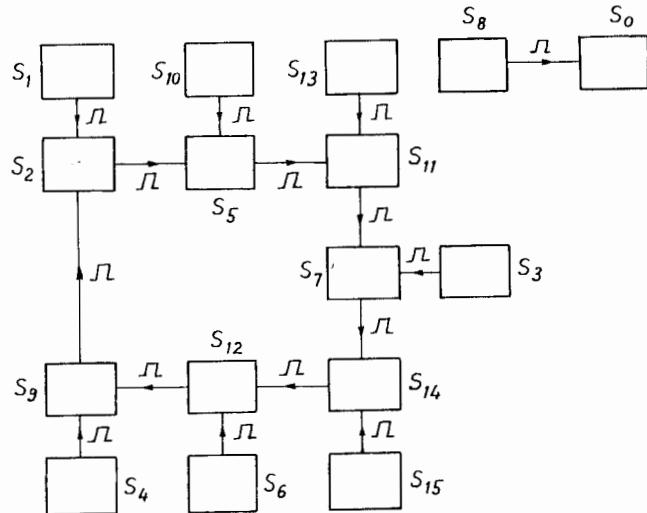
$$7 = \bar{A}\bar{B}C, \quad 8 = \bar{B}CD, \quad 9 = \bar{A}\bar{B}\bar{C}\bar{D}, \quad 10 = \bar{A}\bar{B}C, \quad 11 = \bar{A}\bar{B}D.$$

9.3. Bởi vì bộ ghi dịch được thiết kế tạo dãy nhị phân có modul 11. Chọn dãy 11 trạng thái cho bởi đồ hình chuyển đổi trạng thái Bruijn như hình 9-5a :

$$S_0 - S_1 - S_2 - S_5 - S_{11} - S_7 - S_{14} - S_{13} - S_{10} - S_4 - S_8 - S_6$$

Hàm hồi tiếp là $f = \bar{A}\bar{C}\bar{D} + BCD + A\bar{C}D + A\bar{B}\bar{C}\bar{D}$ với A là bit có trọng số nhỏ nhất của bộ ghi dịch. Đầu ra logic $g = D + AC$.

9.4. (a)



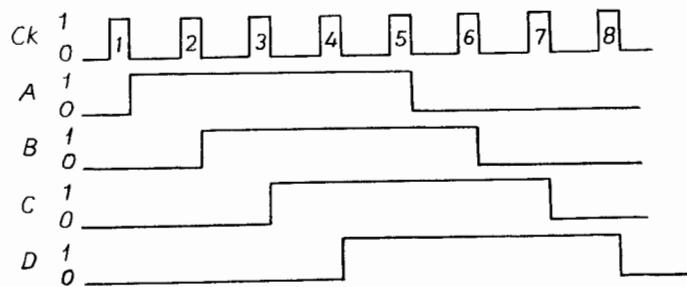
9.5. Hàm logic hồi tiếp

$$f = B \oplus C + \bar{A}\bar{B} + m\bar{B}$$

(A là bit có trọng số nhỏ nhất của thanh ghi dịch)

9.6. $P = \overline{ABC} + ABC\bar{C}$, $Q = \overline{A}\ \overline{C}$, $R = \overline{A}\ \overline{B} + \overline{B}\ \overline{C} + \overline{A}\ \overline{C} + ABC$,
 $S = A\overline{B}\overline{C} + \overline{A}\ \overline{C}$

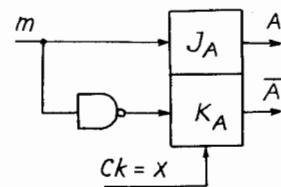
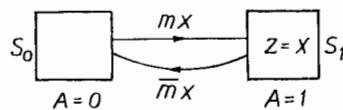
9.7. (a)



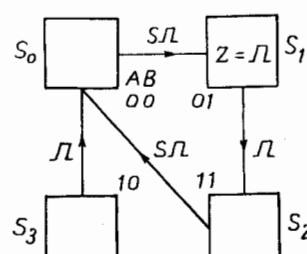
(b) $f = \overline{C}\ \overline{D}$

Chương 10

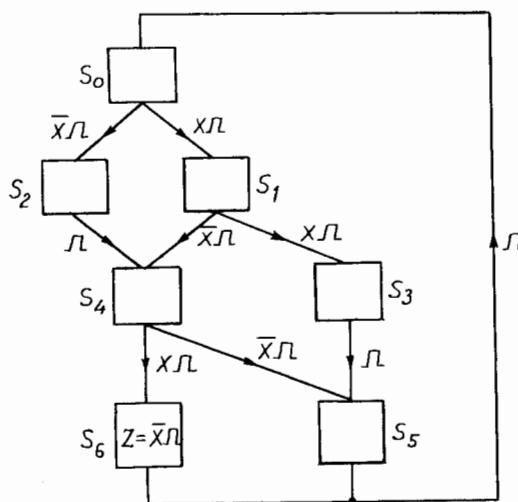
10.1. $S_A = \overline{Am}X$ $J_A = mX$
 $R_A = Am\bar{X}$ $K_A = \overline{m}X$
 $Z = AX$



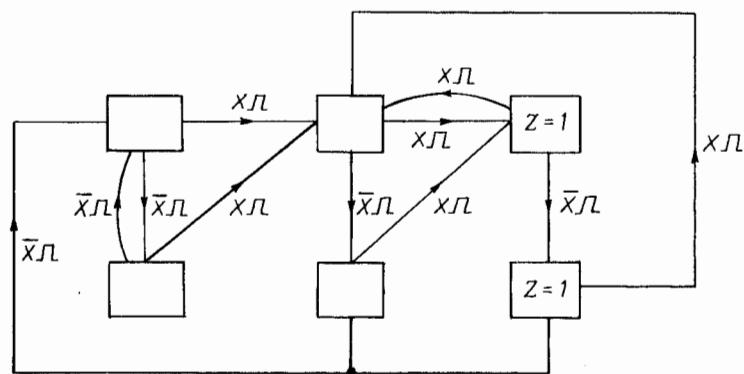
10.2. $J_A = B$ $J_B = \overline{AS}$
 $K_A = \overline{S} + \overline{B}$ $K_B = AS$
 $Z = \overline{AB}$



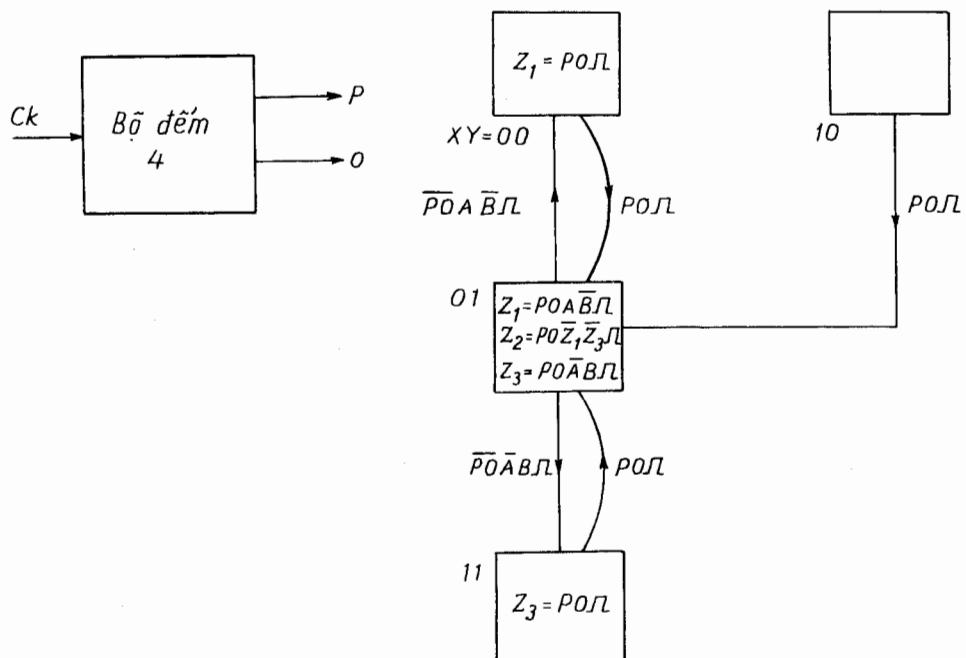
10.3



10.4



10.5



10.6.

A	B	0	1
0	S_0	S_3	
1	S_1	S_2	

a)

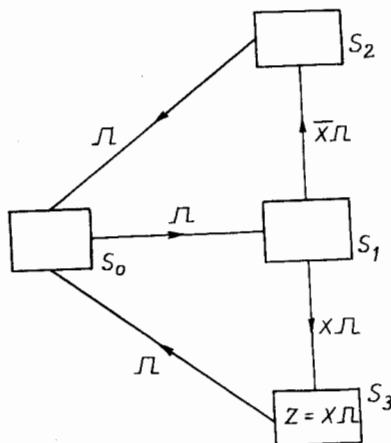
DC	BA	00	01	11	10
00	S_6	S_7	S_8	S_9	
01	S_5	S_3	S_4	S_2	
11			S_0	S_1	
10					

b)

$$10.7 \quad P_f = (S_0 S_1)(S_4 S_5)(S_2 S_3)(S_6)$$

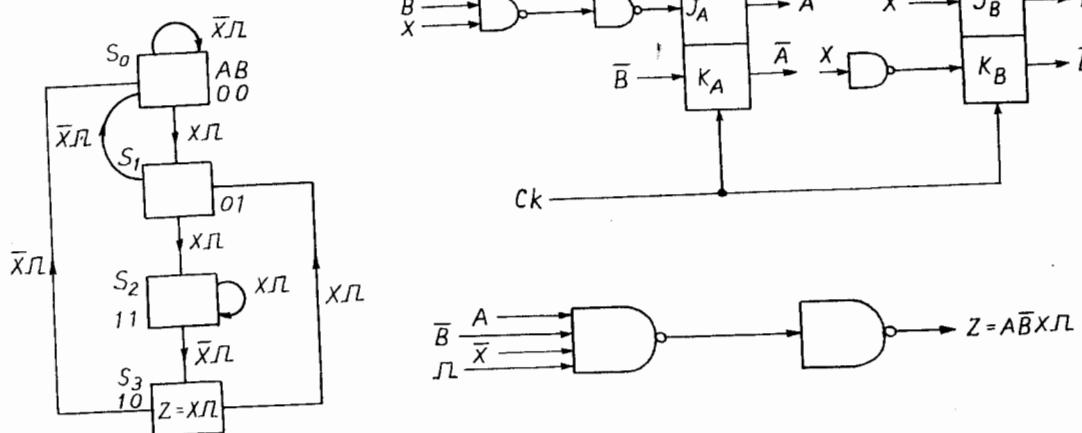
10.8

X	0	1
S ₀	S ₁ Z = 0	S ₁ Z = 0
S ₁	S ₂ Z = 0	S ₃ Z = 0
S ₂	S ₀ Z = 0	S ₀ Z = 0
S ₃	S ₀ Z = 0	S ₀ Z = 0



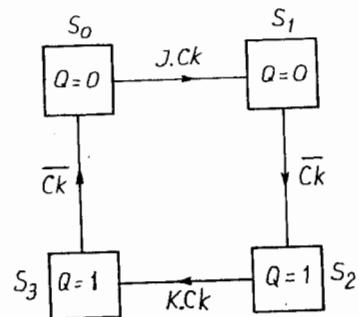
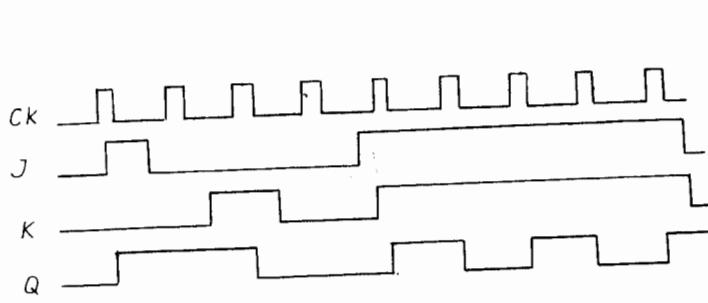
Phát hiện một dãy tín hiệu nhị phân có độ dài bằng 3 được đưa đến đầu vào. Nếu dãy đó có dạng 011, 111 thì tín hiệu ra Z = 1

10.9

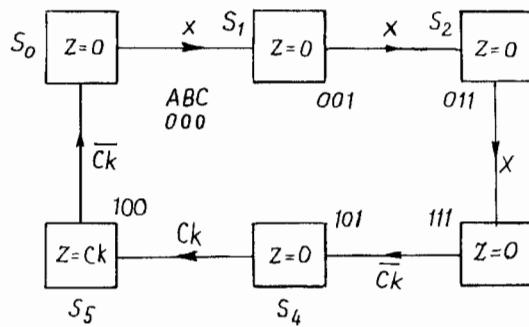


Chương 11

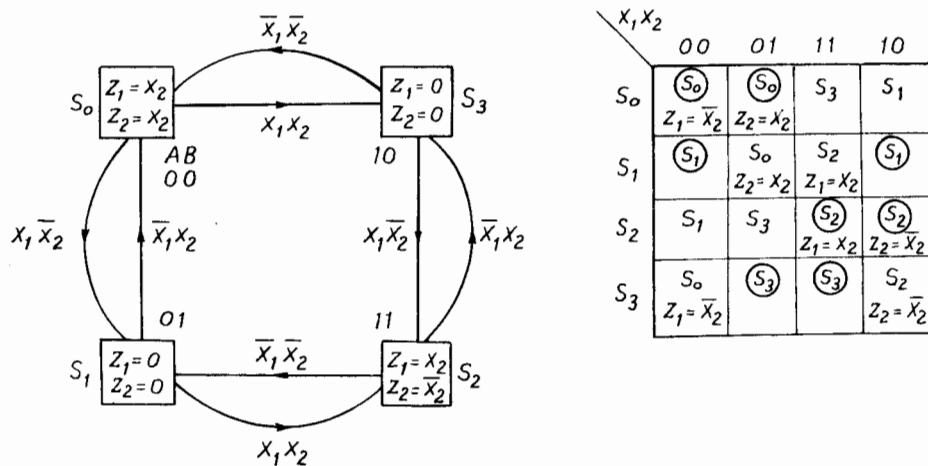
11.1



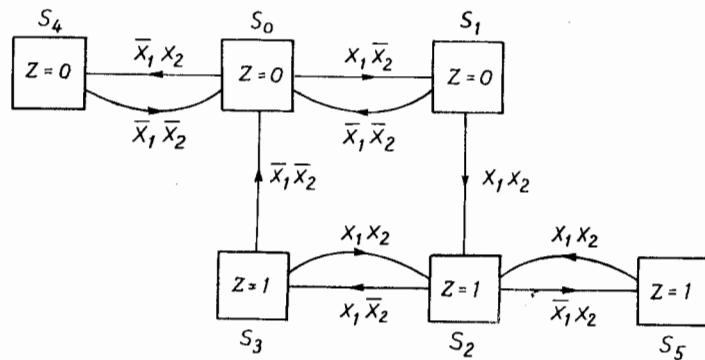
11.2. Dồ hình trạng thái cơ sở :



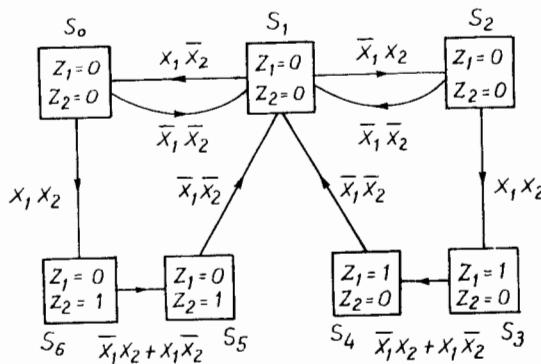
11.3



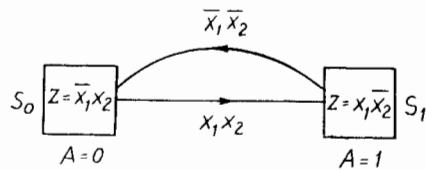
11.5. Dồ hình trạng thái của mạch



11.6



11.7

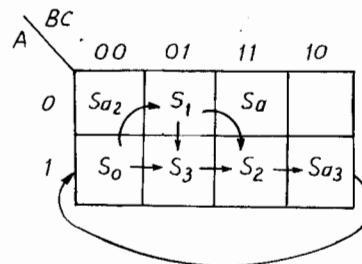


X_1X_2	00	01	11	10
S_0	(S_0) $Z=0$	(S_0) $Z=1$	S_1 $Z=0$	(S_0) $Z=0$
S_1	S_0 $Z=0$	(S_1) $Z=0$	(S_1) $Z=0$	(S_1) $Z=1$

11.8 Chạy đua 1 : $S_0 \rightarrow S_3$ với tín hiệu vào X_1X_2 - Nguy hiểm

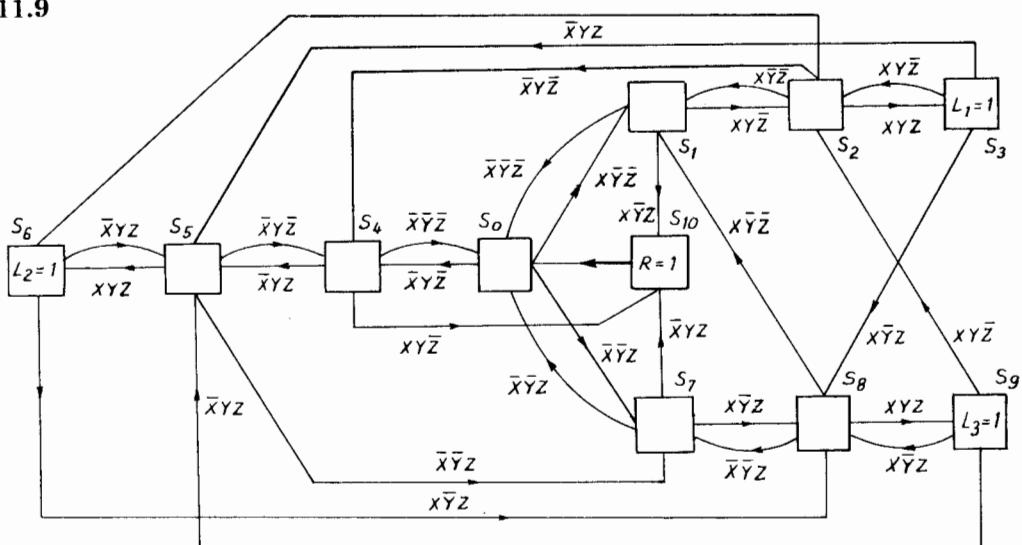
Chạy đua 2 : $S_1 \rightarrow S_2$ với tín hiệu vào X_1X_2 - Nguy hiểm

X_1X_2	00	01	11	10
S_0	(S_0)	(S_0)	S_3	S_1
S_1	S_3		S_2	(S_1)
S_3	(S_3)	S_2	(S_3)	S_2
S_2	S_0	S_0	(S_2)	(S_2)



Thêm 3 trạng thái
giả để mã hóa
tránh chạy đua

11.9



Chương 12

12.1. Các biến của hàm là $A \ B \ C \ D \ E$

- (a) Mức 1 : Có 1 MUX 4 đầu vào và tín hiệu điều khiển là D
Mức 2 : có 4 MUX $4 \rightarrow 1$, biến đưa vào đầu vào điều khiển là BC
 $MUX \ 1 \ (DE = 00)$ có $D_0 = A, D_1 = \dots, D_2 = \bar{A}, D_3 = \bar{\bar{A}}$
 $MUX \ 2 \ (DE = 01)$ có $D_0 = \bar{A}, D_1 = A, D_2 = 1, D_3 = 1$
 $MUX \ 3 \ (DE = 10)$ có $D_0 = 1, D_1 = 0, D_2 = A, D_3 = 1$
 $MUX \ 4 \ (DE = 11)$ có $D_0 = 1, D_1 = 0, D_2 = \bar{A}, D_3 = A$
- (b) Mức 1 : gồm 1 MUX $4 \rightarrow 1$, tín hiệu điều khiển là D và E
Mức 2 : gồm 4 MUX $4 \rightarrow 1$, tín hiệu điều khiển là B, C
 $MUX \ 1 \ (DE = 00) : D_0 = A, D_1 \cdot D_2 = 1, D_3 = 0$
 $MUX \ 2 \ (DE = 01) : D_1 = A, D_1 = \bar{A}, D_2 = 1, D_3 = A$
 $MUX \ 3 \ (DE = 10) : D_0 = A, D_1 = 1, D_2 = 1, D_3 = 1$
 $MUX \ 4 \ (DE = 11) : D_0 = A, D_1 = 1, D_2 = 1, D_3 = 1$

12.2. (a) Các đầu vào của $MUX \ 1$

$$D_0 = \bar{A} + \bar{B}, \ D_1 = \bar{A}\bar{B}, \ D_2 = 0, \ D_3 = A + B$$

Các đầu vào của $MUX \ 2$:

$$D_0 = \bar{A} \bar{B}, \ D_1 = A + \bar{B}, \ D_2 = A \bar{B}, \ D_3 = A$$

Các đầu vào của $MUX \ 3$:

$$D_0 = A + B, \ D_1 = \bar{A} B, \ D_2 = \bar{A} \bar{B}, \ D_3 = \bar{A} + B$$

Các đầu vào của $MUX \ 4$:

$$D_0 = \bar{A} \bar{B} + AB, \ D_1 = \bar{B}, \ D_2 = A\bar{B}, \ D_3 = AB$$

(b) $MUX \ 5$ có E và F là tín hiệu điều khiển

Các $MUX \ 1, 2, 3$ và 4 có B, C và D là tín hiệu điều khiển các đầu vào của $MUX \ 1$:

$$D_0 = 1, \ D_1 = 0, \ D_2 = 0, \ D_3 = \bar{A}, \ D_4 = \bar{A}, \ D_5 = \bar{A},$$

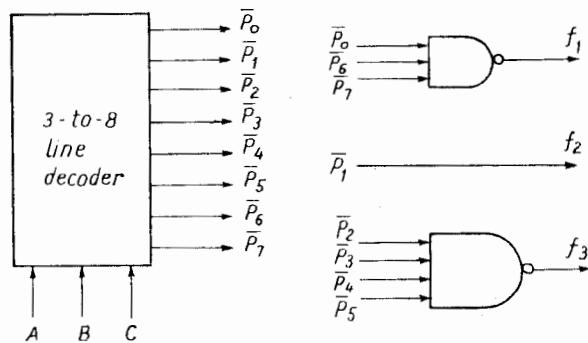
$$D_6 = 0, \ D_7 = A$$

Các đầu vào của $MUX \ 2$:

$$D_0 = A, \ D_1 = L, \ D_0 = A, \ D_0 = A, \ D_0 = 0, \ D_0 = 1,$$

$$D_0 = 0, \ D_7 = A$$

12.3



Chương 13

13.1. Số nhị phân $ABCD$ chuyển thành số bù 2 tương ứng là $PQRS$ trong đó :

$$P = A \oplus (B + C + D) \quad RR = C \oplus D$$

$$Q = B \oplus (C + D) \quad S = D$$

13.2. Số $NBCD$ là $ABCD$ chuyển thành dạng bù 9 tương ứng là $PQRS$ trong đó

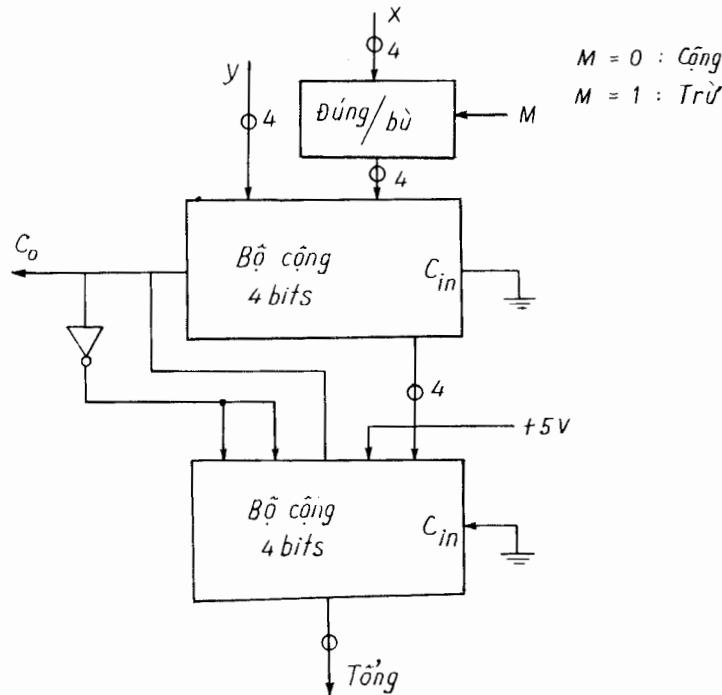
$$P = \overline{A} \ \overline{B} \ \overline{C}; \quad Q = B \oplus C, \quad R = C, \quad S = \overline{D}$$

13.4. Số $NBCD$ là $ABCD$ chuyển thành số dạng mà dư 3 (XS 3) là $PQRS$ trong đó :

$$D = A + BD + BC, \quad Q = \overline{B} D + \overline{B} C + B \ \overline{C} \ \overline{D},$$

$$R = C \sim D, \quad S = D$$

13.5.



13.6. Tổng của 2 số ở dạng nhị phân S

Nếu $(0)_{10} < S \leq (11)_{10}$ thì kết quả này cũng đúng với kết quả ở dạng có số 12.

Nếu $(11)_{10} < S \leq (23)_{10}$. Kết quả sai so với kết quả ở dạng cơ số 12, cần hiệu chỉnh bằng cách lấy $S + 0100$ để được kết quả đúng ở dạng cơ số 12

13.7. Tổng 2 số ở dạng nhị phân là S

(a) $(0)_{10} \leq S \leq (4)_{10}$. Kết quả này cũng đúng với kết quả ở dạng của 2421.

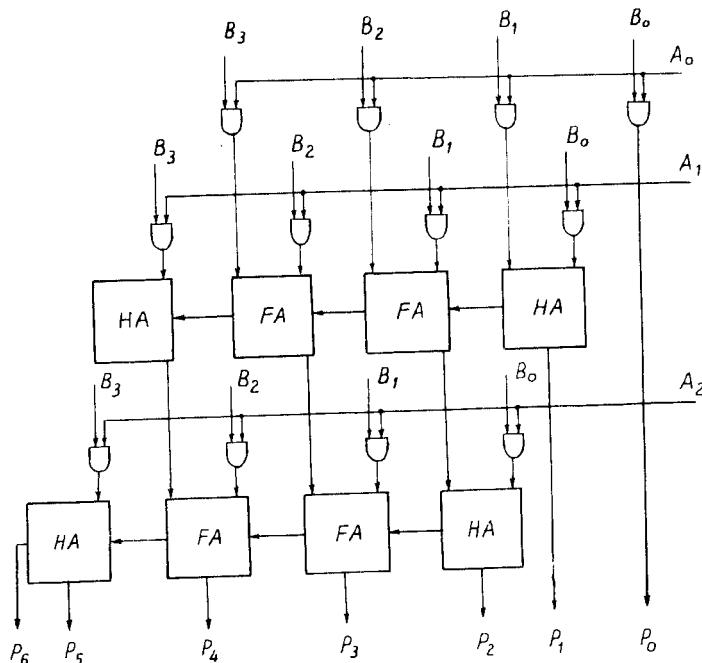
(b) $(4)_{10} < S \leq (8)_{10}$. Kết quả này sai so với kết quả ở dạng mã 2421 cần hiệu chỉnh bằng cách lấy $S + 0110$ để được kết quả đúng ở dạng mã 2421.

(c) $(11)_{10} \leq S \leq (19)_{10}$. Kết quả này đúng so với kết ở dạng mã 2421

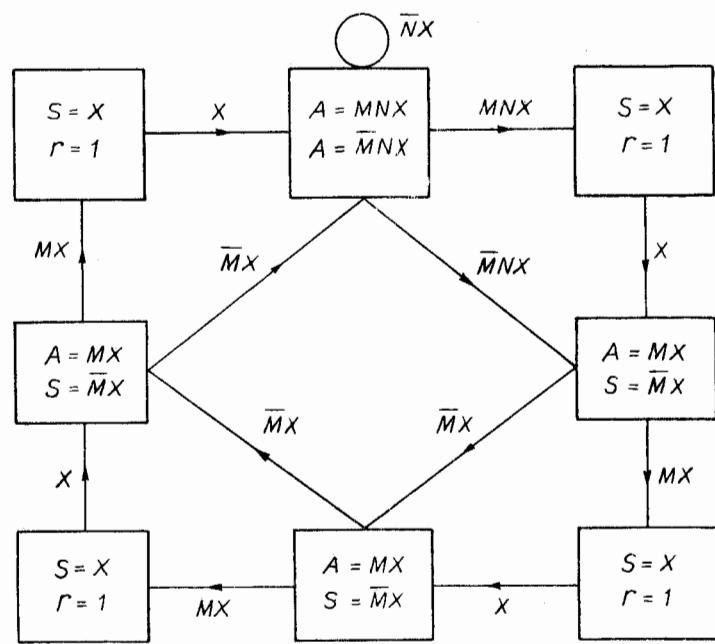
(d) $(22)_{10} \leq S \leq (26)_{10}$. Kết quả này sai so với kết quả ở dạng mã 2421 phải hiệu chỉnh bằng cách lấy $S + 1010$ để được kết quả đúng ở dạng mã 2421

(e) $(26)_{10} < S \leq (30)_{10}$. Kết quả này cũng đúng so với kết quả ở dạng mã 2421.

13.8. Sơ đồ mạch thực hiện



13.9. Đồ hình chuyển đổi trạng thái



Tài liệu tham khảo

- 1) Bùi Minh Tiêu. **Kỹ thuật số tập I, II** – Nhà xuất bản Đại học
- 2) Blakeslee, T.R. **Digital Design with Standard MSI and LSI** – Wiley 1985
- 3) Bennets, R.G. **Introduction to Digital Board Testing** – Arnold 1982
- 4) Bennets, R.G. **Design of Testable Logic Circuits** – Addison Wesley, Wokingham (1984)
- 5) Beuth und Schmusch. **Grundschaltungen** – Vogel Buchverlag 1989
- 6) Chappewill, S.G., R.P.R. Menon, J.F. Pelegrin and A.M. Schowe. "**Functional Simulation in the Lamp System**" – Proc. 13th Design Automation Conference. (1976)
- 7) Clare, C.R. **Designing Logic Systems using state machines** – McGraw-Hill, 1983
- 8) Comer, D.J. **Digital Logic and State Machine Design** – Holt, Rinehart & Winston, Eastbourne (1984).
- 9) David Green. "**Modern Logic Design**" – University of Manchester 1986
- 10) Don Lancaster. **Das CMOS – Kochbuch** – IWT Verlag 1991
- 11) Eberhard Kuhn. **Handbuch TTL – und CMOS – Schaltungen** – Huthig Buch Verlag Heidelberg 1992
- 12) Fletcher, W.I. **An Engineering Approach to Digital Design** – Prentice – Hall, 1990
- 13) Friedman, A.D. **Logical Design of Digital Systems** – Woodland Hills, Computer Science Press, 1987
- 14) **GAL Data Book.** Lattice 1992
- 15) Givne, D.D. **Introduction to Switching Circuit Theory** – Mc Graw – Hill, 1986
- 16) Greenfield, J.D. **Practical Digital Design Using ICs** – Wiley, 1990 Hill, F.J. and Petersen, G.R. **Introduction to Switching Theory and Logical Design** – Wiley, 1984
- 17) Johnson, D.E. **Digital Circuits and Microcomputers** – Prentice – Hall, 1989
- 18) Kline, R.M. **Structured Digital Design** – Prentice – Hall 1990
- 19) Lee, S.C. **Digital Circuits and Logic Design** – Prentice – Hall, 1986
- 20) Lee, S.C. **Modern Switching Theory and Digital Design** – Prentice – Hall, 1988
- 21) Lewin, D. **Logical Design of Switching Circuits** Macmillan, 1986
- 22) Lind, L.F. and Nelson, J.C.C. **Analysis and Design of Sequential Digital Systems**, Macmillan, 1987
- 23) Mano, M. **Digital Logic and Computer Design** – Prentice – Hall, 1989
- 24) Milos D. Eregegovac Tomis Lang. **Digital Systems and Hardware/Firmware Algorithms**
- 25) Nguyễn Xuân Quỳnh. **Lý thuyết mạch logic và kỹ thuật số** – Nhà xuất bản Đại học
- 26) PAL Device. Data Book Bipolar and CMOS, 1990
- 27) Peatman, J.B. **Digital Hardware Design** – McGraw – Hill, 1980
- 28) Programmable Logic Texas Instruments 1990
- 29) Seifart/**Digitalschaltungen** – Verlag Technik GmbH Berlin 1989
- 30) Triebel, W.A. **Integrated Digital Electronics** – Prentice – Hall, 1979
- 31) Wermer Kobitzsch/**Mikroprozessoren** – R.Oldenbourg Verlag München Wien 1989
- 32) W.Hiberg und R.Piloty/**MP und Ihre Anwendungen** – R.Oldenbourg Verlag München Wien 1988
- 33) Wickes, W.E. **Logic Design with Integrated Circuits** – Wiley, 1988
- 34) Zissos, D. **Problems and Solutions in Logic Design** – OUP, 1976

203235



Giá: 35.000đ