

## MỤC LỤC

CHƯƠNG 1: .....	4
HỆ THỐNG ĐẾM VÀ MÃ .....	4
I. BIỂU DIỄN SỐ TRONG CÁC HỆ THỐNG ĐẾM .....	4
1. Khái niệm cơ bản .....	4
2. Các hệ đếm thông dụng .....	5
3. Biểu diễn số trong các hệ đếm .....	6
II. HỆ ĐẾM HAI (NHỊ PHÂN) .....	7
III. MÃ HOÁ HỆ SỐ 10 .....	9
1. Khái niệm về mã hoá hệ số .....	9
2. Các mã thông dụng .....	9
CHƯƠNG 2: .....	11
ĐẠI SỐ BOOLEAN.....	11
I. KHÁI NIỆM CHUNG .....	11
1. Mở đầu .....	11
2. Một số khái niệm cơ bản .....	11
II. BIẾN VÀ HÀM LOGIC.....	12
1. Khái niệm về biến và hàm logic .....	12
2. Các hàm logic sơ cấp.....	13
3. Hệ hàm đầy đủ.....	17
III. Phương pháp biểu diễn hàm logic .....	18
1. Phương pháp dùng bảng giá trị của hàm .....	18
2. Phương pháp hình học.....	18
3. Phương pháp biểu thức đại số .....	18
4. Phương pháp dùng bảng Karnaugh .....	19
CHƯƠNG 3.....	21
TỐI THIỂU HOÁ HÀM BOOLEAN .....	21
I. PHƯƠNG PHÁP TỐI THIỂU HOÁ .....	21
1. Khái niệm tối thiểu hoá .....	21
2. Phương pháp tối thiểu hoá hàm logic bằng biến đổi đại số .....	22
3. Nhóm các phương pháp tối thiểu hoá theo thuật toán.....	22
CHƯƠNG 4: .....	27
GIỚI THIỆU VI MẠCH SỐ .....	27
I. ĐỊNH NGHĨA VÀ PHÂN LOẠI .....	27
1. Định nghĩa .....	27
2. Phân loại vi mạch theo bản chất của tín hiệu vào / ra .....	27
3. Phân loại theo mật độ tích hợp .....	28
4. Phân loại theo công nghệ chế tạo .....	28
II. CÁC THÔNG SỐ CHÍNH CỦA VI MẠCH SỐ .....	30
1. Mức logic .....	30
2. Đặc tính truyền đạt .....	30
3. Các thông số về dòng điện. ....	31
4. Công suất tiêu thụ.....	32
5. Hệ số tải FAN-IN; FAN-OUT .....	33
6. Khoảng lè chống nhiễu (Noise Margin).....	34
7. Thời gian truyền đạt và thời gian quá độ .....	35
8. Dạng vỏ IC .....	36
9. Giới hạn nhiệt độ.....	36
III. CÔNG NGHỆ IC SỐ .....	37
1. Công nghệ đơn cực (công nghệ MOS - Metal Oxide Semiconductor) .....	37

2. Công nghệ lưỡng cực .....	39
3. Giao tiếp TTL-CMOS và CMOS-TTL. ....	43
CHƯƠNG 5: .....	46
PHÂN TÍCH VÀ THIẾT KẾ MẠCH TỔ HỢP.....	46
I. MÔ HÌNH TOÁN HỌC CỦA MẠCH TỔ HỢP.....	46
II. PHÂN TÍCH MẠCH TỔ HỢP.....	47
III. THIẾT KẾ MẠCH TỔ HỢP.....	47
1. Bài toán thiết kế và các bước thực hiện .....	47
2. Thiết kế mạch tổ hợp 2 tầng và nhiều tầng .....	47
3. Thiết kế một hệ hàm tổ hợp.....	50
CHƯƠNG 6: .....	51
MỘT SỐ MẠCH TỔ HỢP THƯỜNG GẶP .....	51
I. BỘ CỘNG NHỊ PHÂN MỘT CỘT SỐ.....	51
1. Phân tích bài toán .....	51
2. Xây dựng sơ đồ .....	52
II. BỘ TRỪ NHỊ PHÂN MỘT CỘT SỐ .....	53
II. BỘ TRỪ NHỊ PHÂN MỘT CỘT SỐ .....	54
III. BỘ SO SÁNH – COMPARATOR.....	55
1. Bộ so sánh 2 số nhị phân 1 bit.....	55
2. Bộ so sánh n bit .....	56
IV. BỘ TẠO VÀ KIỂM TRA CHẶN LẼ - PARITY GENERATOR AND CHECKER ...	57
IV. BỘ TẠO VÀ KIỂM TRA CHẶN LẼ - PARITY GENERATOR AND CHECKER ...	58
1. Mạch tạo bit chẵn lẻ .....	58
2. Mạch kiểm tra chẵn lẻ .....	58
V. MẠCH PHÂN LOẠI NGẮT .....	60
VI. BỘ CHỌN KÊNH VÀ PHÂN KÊNH (MULTIPLEXER AND DEMULTIPLEXER) 60	
1. Bộ chọn kênh.....	60
2. Bộ phân kênh.....	61
VII. BỘ CHUYỂN MÃ.....	61
1. Các loại mã tiêu biểu.....	61
2. Mạch mã hoá - lập mã (ENCODER) .....	62
3. Mạch giải mã (DECODER) .....	64
4. Thiết kế mạch chuyển mã.....	67
5. Một số vi mạch chuyển mã thông dụng .....	67
CHƯƠNG 7: .....	70
CÁC PHẦN TỬ NHỚ CƠ BẢN.....	70
I.KHÁI NIỆM CHUNG .....	70
II. ĐỊNH NGHĨA VÀ PHÂN LOẠI .....	70
1. Định nghĩa .....	70
2. Phân loại FF .....	70
2. Phân loại FF .....	71
3. Biểu diễn FF.....	71
III. CÁC LOẠI FF VÀ ĐIỀU KIỆN ĐỒNG BỘ.....	71
1. Flip-Flop kiểu RS.....	71
2. JK Flip-Flop. ....	73
3. D Flip-Flop.....	75
4. Flip-Flop kiểu T. ....	76
IV. CHUYỂN ĐỔI GIỮA CÁC LOẠI FF .....	77
CHƯƠNG 8.....	79
BỘ ĐẾM .....	79

I. ĐỊNH NGHĨA VÀ PHÂN LOẠI .....	79
1. Định nghĩa: .....	79
2. Phân loại .....	79
3. Đồ hình trạng thái của bộ đếm: .....	80
II. MÃ CỦA BỘ ĐẾM .....	81
III. CÁC BƯỚC THIẾT KẾ BỘ ĐẾM .....	82
IV. MỘT SỐ DẠNG BỘ ĐẾM THƯỜNG GẶP .....	82
1. Bộ đếm thuận, đồng bộ, hệ số đếm $K_d = 2^n$ , dùng mã nhị phân .....	82
2. Các bộ đếm nghịch, nhị phân, đồng bộ với hệ số đếm chẵn .....	87
3. Các bộ đếm nhị phân, đồng bộ với hệ số đếm chẵn .....	88
4. Thiết kế bộ đếm có mạch khởi động .....	89
V. LƯU Ý KHI THIẾT KẾ BỘ ĐẾM: .....	90
VI. MỘT SỐ IC ĐẾM THƯỜNG GẶP: .....	90
CHƯƠNG 9 .....	91
MÔ TẢ VÀ THIẾT KẾ MẠCH DÂY .....	91
I. KHÁI NIỆM CƠ BẢN .....	91
1. Mô hình tổng quát .....	91
2. Phương pháp mô tả mạch dây .....	91
II. CÁC BƯỚC THIẾT KẾ MẠCH DÂY .....	94
1. Thiết kế mạch dây từ bảng trạng thái của Otomat .....	94
2. Thiết kế mạch dây từ đồ hình trạng thái .....	96
CHƯƠNG 10 .....	97
THANH GHI DỊCH .....	97
I, KHÁI QUÁT .....	97
1. Định nghĩa: .....	97
2. Cấu tạo .....	97
3. Phân loại .....	98
II. ĐỒ HÌNH TỔNG QUÁT .....	98
1. Sơ đồ khối: .....	101
2. Các bước thiết kế .....	102
IV. MẠCH TẠO DÂY TÍN HIỆU TUẦN HOÀN .....	104
1. Sơ đồ khối: .....	104
2. Các bước thiết kế .....	105
V. BỘ ĐẾM VÒNG, VÀ BỘ ĐẾM VÒNG XUẤN .....	107
VI. BỘ GHI DỊCH VỚI HÀM HỒI TIẾP LÀ HÀM CỘNG MODULE 2 .....	107
1. Bộ ghi dịch với hàm hồi tiếp là hàm cộng module có $L_{\max} = 2^n - 1$ .....	107
2. Ví dụ ứng dụng mạch loại này để thiết kế bộ đếm: .....	108
CHƯƠNG 11 .....	112
MẠCH DÂY ĐỒNG BỘ .....	112
Mạch dây đồng bộ là một mạch số bao gồm các mạch tổ hợp và các phần tử nhớ FF, mạch hoạt động theo sự đồng bộ của xung nhịp $C_k$ .....	112
I. PHÂN TÍCH .....	112
II. THIẾT KẾ .....	112
1. Các bước thiết kế: .....	112
III. TỐI THIỂU HOÁ: .....	114
1, Phương pháp Caldwell: .....	114
2. Phương pháp phân hoạch: .....	119
3. Phương pháp dùng bảng so sánh .....	120
IV. MÃ HOÁ TRẠNG THÁI .....	123
V. MẠCH DÂY ĐỒNG BỘ DÙNG MÔ HÌNH MOORE VÀ MÔ HÌNH MEALY .....	123

CHƯƠNG 12.....	128
MẠCH DÂY KHÔNG ĐỒNG BỘ.....	128
I. CÁC BƯỚC THIẾT KẾ: .....	128
II. MÃ HOÁ VÀ TỐI THIỂU HOÁ THÁI TRONG MẠCH DÂY KHÔNG ĐỒNG BỘ. .....	130
III. HIỆN TƯỢNG CHU KỲ VÀ CHẠY ĐUA .....	131
1. Hiện tượng chu kỳ:.....	131
2. Hiện tượng chạy đua: .....	132
IV. CÁC VÍ DỤ MINH HOẠ .....	133
1. Ví dụ 1:.....	133
2. Ví dụ 2.....	137
CHƯƠNG 13.....	141
THIẾT KẾ MẠCH SỐ DÙNG MSI VÀ LSI .....	141
I. KHÁI NIỆM: .....	141
II. BỘ DỒN KÊNH MUX.....	142
1. Sơ đồ tổng quát: .....	142
2. Các ứng dụng .....	142
II. ROM(READ ONLY MEMORY-BỘ NHỚ CHỈ ĐỌC).....	143
1. Sơ đồ cấu tạo: .....	143
2. ứng dụng.....	143
Chương 3 :  KHẢO SÁT BỘ NHỚ BÁN DẪN .....	152
Hình 4.1 Sơ đồ cấu trúc đơn giản của ROM16 x8 .....	153

## PHẦN 1

### ĐẠI SỐ BOOLEAN VÀ VI MẠCH SỐ

#### CHƯƠNG 1: HỆ THỐNG ĐẾM VÀ MÃ

##### ***I. BIỂU DIỄN SỐ TRONG CÁC HỆ THỐNG ĐẾM***

###### ***1. Khái niệm cơ bản***

- + ***Hệ thống đếm*** là tổ hợp các quy tắc gọi và biểu diễn các con số có giá trị xác định
- + ***Chữ số*** là những ký hiệu dùng để biểu diễn một con số
- + ***Phân loại hệ thống đếm*** gồm 2 loại là hệ thống đếm theo vị trí và hệ thống đếm không theo vị trí

. ***Hệ thống đếm theo vị trí*** là hệ thống mà trong đó giá trị về mặt số lượng của mỗi chữ số phụ thuộc vào vị trí của chữ số đó nằm trong con số

Ví dụ: trong hệ đếm thập phân:           Con số 1278 có số 8 chỉ 8 đơn vị

Con số 1827 có số 8 chỉ  $8 \cdot 10^3$  đơn vị

Như vậy tùy vào vị trí khác nhau trong con số mà chữ số biểu diễn giá trị khác nhau.

. ***Hệ thống đếm không theo vị trí*** là hệ thống mà giá trị về mặt số lượng của mỗi chữ số không phụ thuộc vào vị trí của chữ số đó nằm trong con số.

Ví dụ: trong hệ đếm La mã trong các con số IX, XX hay XXXIX đều có X để biểu diễn giá trị 10 trong hệ thập phân mà không phụ thuộc vào vị trí của nó trong con số.

Nhận xét: hệ thống đếm không theo vị trí công kênh khi biểu diễn giá trị lớn do đó ít sử dụng. Do vậy, khi nói tới hệ thống đếm người ta hiểu đó là hệ thống đếm theo vị trí và gọi tắt là **hệ đếm**.

## 2. Các hệ đếm thông dụng

Nếu một hệ đếm có cơ sở là N thì một con số bất kỳ trong hệ đếm đó sẽ có giá trị trong hệ thập phân thông thường như sau:

$$A = a_{n-1} \cdot N^{n-1} + a_{n-2} \cdot N^{n-2} + \dots + a_1 \cdot N^1 + a_0 \cdot N^0$$

Trong đó  $a_k$  là các chữ số lập thành con số ( $k = 0, 1 \dots n-1$ ) và  $0 < a_k < N-1$

Sau đây là một số hệ đếm thông dụng:

+ Hệ đếm mười (thập phân): có cơ sở là 10, các chữ số trong hệ đếm này là: 0, 1, 2, 3, 4, 5, 6, 7, 8 và 9.

ví dụ: con số  $1278 = 1 \cdot 10^3 + 2 \cdot 10^2 + 7 \cdot 10^1 + 8 \cdot 10^0$  biểu diễn một nghìn hai trăm bảy mươi tám đơn vị theo nghĩa thông thường

+ Hệ đếm hai (nhị phân): có cơ sở là 2, các chữ số trong hệ đếm này là 0 và 1

ví dụ: 1011 trong hệ nhị phân sẽ biểu diễn giá trị

$$A = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 11 \text{ trong hệ đếm 10 thông thường}$$

+ Hệ đếm mười sáu (thập lục phân – hexa): có cơ sở là 16 với các chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E và F

ví dụ: 8E trong hệ đếm hexa sẽ biểu diễn giá trị

$$A = 8 \cdot 16^1 + 14 \cdot 16^0 = 142 \text{ trong hệ đếm 10 thông thường}$$

+ Hệ đếm tám (bát phân – octa): có cơ sở là 8 với các chữ số 0, 1, 2, 3, 4, 5, 6 và 7

vd: con số 12 trong hệ octa biểu diễn giá trị

$$A = 1 \cdot 8^1 + 2 \cdot 8^0 = 10 \text{ trong hệ đếm thông thường}$$

Bảng đối chiếu 16 con số đầu tiên trong các hệ đếm trên

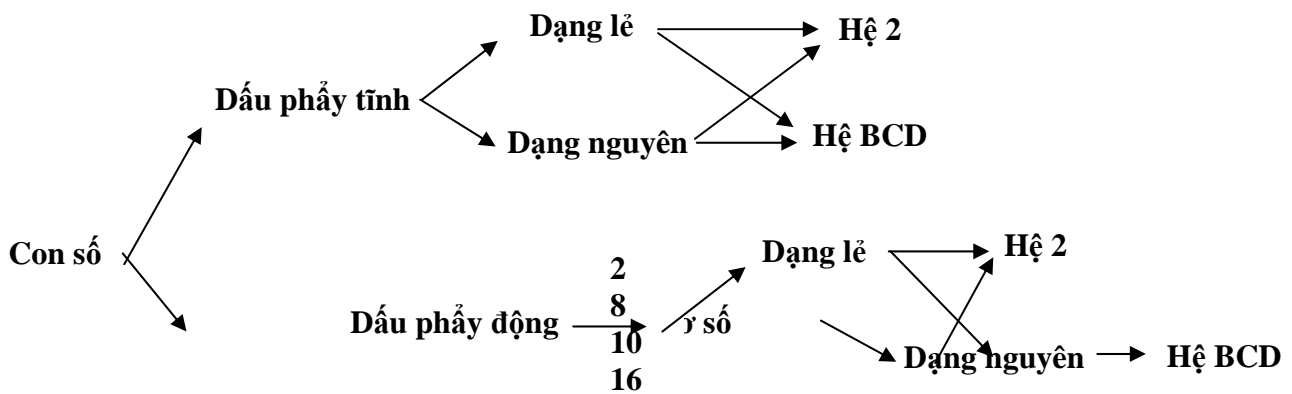
Hệ 10	Hệ 2	Hệ 16	Hệ 8
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11

10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

### 3. Biểu diễn số trong các hệ đếm

Một số trong hệ 10 được biểu diễn với các thành phần: dấu (+ hoặc -), phần nguyên, dấu phẩy (,) và phần lẻ

Khi các con số được xử lý bởi các mạch số thì các con số này phải được biểu diễn dưới dạng hệ 2 hoặc dạng mã nào đó tạo thành từ các số hệ 2 như mã BCD, mã Gray ...). Do vậy, các con số có thể biểu diễn theo sơ đồ sau:



#### **Dấu phẩy tĩnh:**

Dạng nguyên: dấu phẩy luôn ở sau chữ số cuối bên phải. ví dụ: “1001,”

Dạng lẻ: dấu phẩy luôn ở trước chữ số đầu bên trái. ví dụ: “,1001”

#### **Dấu phẩy động:**

Chuyển số thành dạng chuẩn hoá dùng lũy thừa

ví dụ: 12,78 chuyển thành  $(,1278).10^2$

**Dấu** : quy ước lấy giá trị 1 chỉ dấu âm và giá trị 0 chỉ dấu dương

ví dụ: 1 0101 trong hệ 2 chỉ số -5 trong hệ đếm 10

0 1001 trong hệ 2 chỉ số +9 trong hệ đếm 10

Tuy nhiên, người ta cũng còn thường sử dụng số bù để biểu diễn số âm như sau:

*Số bù 1*: dùng số 1 để biểu diễn dấu âm và phần giá trị thực hiện phép lấy phần bù cho mọi chữ số (chuyển 1 thành 0 và 0 thành 1 cho mọi chữ số)

ví dụ: số bù 1 của  $-0101$  là  $1\ 1010$

*Số bù 2*: dùng 1 để biểu diễn dấu âm còn phần giá trị đổi ra số bù 1 sau đó cộng thêm 1 vào hàng đơn vị

ví dụ: số bù 2 của  $-0101$  là  $1\ 1011$

*Số bù 9*: dùng 1 để biểu diễn dấu âm còn phần giá trị trở thành một số sao cho tổng của số mới và số cũ ở mỗi hàng bằng 9

ví dụ: số bù 9 của  $-0011\ 0100\ 0010$  (bằng  $-342$  theo hệ mười)

là  $1\ 0110\ 0101\ 0111$  (bằng  $-657$  theo hệ mười)

*Số bù 10*: lấy số bù 9 cộng thêm 1 đơn vị

ví dụ: số bù 9 của  $-0011\ 0100\ 0010$

là  $1\ 0110\ 0101\ 1000$  (bằng  $-658$  theo hệ mười)

## II. HỆ ĐẾM HAI (NHỊ PHÂN)

### 1. Các phép tính số học trong hệ đếm 2 (module 2)

+ Phép cộng: Dựa trên các nguyên tắc sau

$0 + 0$	0
$1 + 0$	1
$0 + 1$	1
$1 + 1$	10 (0 nhớ 1)

+ Phép trừ: Dựa trên các nguyên tắc sau

$0 - 0$	0
$1 - 0$	1
$1 + 1$	0
$10 - 1$	1

+ Phép nhân: Dựa trên các nguyên tắc sau

$0 \cdot 0$	0
$1 \cdot 0$	0
$0 \cdot 1$	0

1 . 1	1
-------	---

+ Phép chia: thực hiện như với hệ thập phân

## 2. Chuyển đổi giữa hệ 2 và hệ 10

Trong khi con người sử dụng hệ đếm 10 thì các mạch gia công và xử lý số liệu lại sử dụng hệ đếm 2 nên việc chuyển đổi giữa hai hệ đếm này là rất quan trọng.

### a. Chuyển đổi từ hệ 2 sang hệ 10

Một con số trong hệ 2 có giá trị trong hệ 10 là:

$$A = a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

trong đó  $a_k = 0$  hoặc  $1$  (với  $k = 0, 1, 2, \dots, n-1$ )

ví dụ: chuyển đổi con số 1001 trong hệ 2 sang hệ 10 như sau:

$$A = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9$$

### b. Chuyển đổi số từ hệ 10 sang hệ 2

Chuyển đổi từng phần nguyên và phần lẻ sau đó gộp lại

*Chuyển đổi phần nguyên* theo nguyên tắc chia và lấy phần dư

ví dụ: chuyển đổi số 17 hệ mười sang hệ hai như sau

Phần nguyên chia cho 2	0	1	2	4	8	17 ← số hệ 10
Số dư	1	0	0	0	1	← Số hệ 2

*Chuyển đổi phần lẻ* theo nguyên tắc nhân 2 trừ 1 như sau:

Đặt số 10 (phần lẻ) ở tận cùng bên trái. Nhân số hệ mười này với 2, nếu tích số lớn hơn 1 thì lấy tích số trừ đi 1, đồng thời ghi 1 xuống hàng dưới (hàng đặt hệ số cần tìm), nếu tích số nhỏ hơn 1 đặt 0 xuống hàng dưới, ghi sang cột 2 và tiếp tục tới khi hiệu số bằng 0 hoặc đạt số lẻ theo yêu cầu

ví dụ: chuyển đổi số 0,525 hệ mười sang hệ hai. áp dụng quy tắc trên ta có:

Hệ 10	$0,525 \times 2 = 1,05$	$0,05 \times 2 = 0,1$	$0,1 \times 2 = 0,2$	$0,2 \times 2 = 0,4$
<b>0,525</b>	$1,05 - 1 = 0,05$			
Hệ 2	1	0	0	0

Vậy số hệ 2 thu được là 0,1000

Từ 2 kết quả trên ta tìm được số hệ 2 tương ứng với số hệ 10 bằng cách gộp phần nguyên và phần lẻ với nhau

ví dụ:

Số hệ 10	Số hệ 2
17	10001



0,525	0,1000
17,525	10001,1000

### III. MÃ HOÁ HỆ SỐ 10

#### 1. Khái niệm về mã hoá hệ số

Để thực hiện việc chuyển đổi các con số giữa 2 hệ thống đếm 2 và 10 người ta sử dụng phương pháp biểu diễn 2 – 10. Phương pháp này gọi là *mã hoá các con số trong hệ đếm 10 bằng các nhóm mã hệ 2* (BCD – Binary Coded Decimal).

Các chữ số trong hệ 10 gồm các số từ 0 tới 9 do đó sẽ được biểu diễn bằng các hệ số hai có 4 chữ số. Nghĩa là thực hiện chuyển đổi một số hệ 2 sang hệ 10 ta phải thực hiện chuyển đổi với  $n = 4$

$$A = a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

$$A = 8a_3 + 4a_2 + 2a_1 + 1a_0$$

Trong đó, 8-4-2-1 gọi là trọng số và mã có quy luật trên gọi là mã BCD có trọng số tự nhiên hay mã BCD 8421

ví dụ:

Hệ 10	Mã BCD 8421
12	0001 0010
1278	0001 0010 0111 1000

Tuy nhiên, trên thực tế người ta còn sử dụng các mã BCD với trọng số khác nhau như: 7421, 5421, 2421 ...

*Chú ý:* Các con số biểu diễn bằng mã BCD 8421 và 7421 là duy nhất trong khi các mã BCD 5421 hay 2421 là không duy nhất.

#### 2. Các mã thông dụng

Khi sử dụng 4 chữ số hệ 2 ta sẽ có 16 tổ hợp khác nhau nhưng mã BCD chỉ sử dụng 10, do đó dư 6 tổ hợp. Bằng cách chọn 10 trong số 16 tổ hợp khác nhau người ta sẽ có nhiều loại mã khác nhau. Thông dụng nhất là: Mã BCD, Mã thừa 3, Mã Gray Ngoài ra có thể sử dụng 5 chữ số hệ 2 để mã hoá, ví dụ: Mã Johnson, Mã 2 trên 5 ...

+ Mã BCD: đã được trình bày ở trên

+ **Mã thừa 3:** được tạo thành bằng cách cộng thêm 3 đơn vị vào mã BCD 8421. Loại mã này được sử dụng rộng rãi trong thiết bị tính toán số học của hệ thống xử lý hoặc gia công các tín hiệu số.

+ **Mã Gray:** có đặc điểm là khi chuyển từ một mã số này sang mã số khác tiếp theo thì từ mã chỉ thay đổi tại cùng 1 vị trí của ký hiệu mã

+ **Mã 2 trên 5:** sử dụng 5 chữ số hệ 2 để biểu diễn các chữ số hệ 10. Mỗi tổ hợp luôn có 2 chữ số 1 và 3 chữ số 0.

+ **Mã Johnson:** sử dụng 5 chữ số hệ 2 với đặc điểm là khi chuyển sang mã số kế tiếp sẽ thay 0 bằng 1 bắt đầu từ **phải sang trái** tới khi đạt 11111 ( ứng với 5 trong hệ 10) sẽ bắt đầu thay 1 bằng 0 và cũng theo chiều từ **phải sang trái**.

*Bảng biểu diễn các chữ số hệ 10 theo các loại mã khác nhau*

Số hệ 10	Số hệ 2 (BCD- 8421)				Mã thừa 3				Mã Gray				Mã 2 trên 5					Mã Johnson				
	B 3	B 2	B 1	B 0	A 3	A 2	A 1	A 0	G 3	G 2	G 1	G 0	D 4	D 3	D 2	D 1	D 0	J 4	J 3	J 2	J 1	J 0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	1
2	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	0	0	1	1
3	0	0	1	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1
4	0	1	0	0	0	1	1	1	0	1	1	0	0	1	0	1	0	0	1	1	1	1
5	0	1	0	1	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1	1	1
6	0	1	1	0	1	0	0	1	0	1	0	1	1	0	0	0	1	1	1	1	1	0
7	0	1	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	1	1	1	0	0
8	1	0	0	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	1	0	0	0
9	1	0	0	1	1	1	0	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0

# CHƯƠNG 2: ĐẠI SỐ BOOLEAN

## I. KHÁI NIỆM CHUNG

### 1. Mở đầu

Kỹ thuật điện tử ngày nay được chia làm 2 nhánh lớn kỹ thuật điện tử tương tự và kỹ thuật điện tử số. Kỹ thuật điện tử số ngày càng thể hiện nhiều tính năng ưu việt về tốc độ xử lý, kích thước nhỏ gọn, khả năng chống nhiễu cao, tiêu thụ điện năng ít .... Do đó, điện tử số được ứng dụng rộng rãi trong nhiều lĩnh vực và ngày càng trở thành một phần thiết yếu hơn trong các hệ thống và thiết bị ở hầu hết các lĩnh vực có ứng dụng khoa học kỹ thuật và công nghệ mới (cơ khí, hoá học, y học...).

Hơn nữa, với sự phát triển của mạch tích hợp đã tạo nên sự thúc đẩy càng mạnh mẽ trong việc tạo ra những mạch số có độ phức tạp càng tăng. Nền công nghệ ban đầu chỉ tạo được các mạch tích hợp cỡ nhỏ (S.S.I) nhưng, ngày nay, việc sử dụng các mạch tích hợp cỡ vừa (M.S.I), cỡ lớn (L.S.I) và cực lớn (VLSI) ngày càng trở nên phổ biến.

Trong mạch số, tín hiệu đầu vào ở 1 trong 2 trạng thái logic 0 hoặc 1 và đầu ra cũng ở 1 trong 2 trạng thái 0 hoặc 1 tùy theo tín hiệu đầu vào và các phần tử trong mạch gọi là các cổng logic. Để mô tả mạch số người ta sử dụng công cụ toán học là **đại số Boolean (đại số logic)**. Đây là cơ sở toán học cho mọi lĩnh vực có liên quan đến kỹ thuật số.

### 2. Một số khái niệm cơ bản

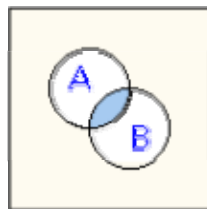
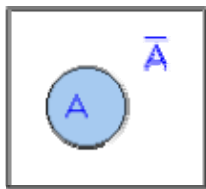
+ **Đại số logic**: là một tập hợp S của các đối tượng A, B, C ... trong đó xác định 2 phép toán cộng logic và nhân logic với các tính chất sau:

Tính chất	Tên gọi
S chứa $(A + B)$ và $(A.B)$	tính đóng kín
$A + B = B + A$ $A.B = B.A$	Luật giao hoán
$(A + B).C = A.C + B.C$ $A + B.C = (A + B).(A + C)$	Luật phân phối
$(A + B) + C = A + (B + C)$ $(A.B).C = A.(B.C)$	Luật kết hợp
$A + A = A$ $A.A = A$	
$A + B = B \Leftrightarrow A.B = A$	tính nhất quán
$A + 0 = A$ $A . 0 = 0$ $A + 1 = 1$ $A . 1 = A$	

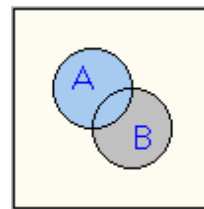
$A + \bar{A} = 1$ $A.\bar{A} = 0$	
$A.(A + B) \equiv A + A.B \equiv A$	Luật hấp thụ
$\overline{A + B} = \bar{A}.\bar{B}$ $\overline{A.B} = \bar{A} + \bar{B}$	Luật De Morgan
$A + \bar{A}B = A + B$ $\bar{A} + AB = \bar{A} + B$ $A.B + A.C + B.\bar{C} = A.C + B.\bar{C}$	
$\bar{\bar{A}} = A$ $\bar{1} = 0$ $\bar{0} = 1$	

+ Giản đồ Venn: đây là cách biểu diễn trực quan các phép toán trong đại số logic. Trên giản đồ Venn tập hợp S được biểu diễn bằng 1 ô vuông còn các phần tử A, B, C ... được biểu diễn bằng các miền nằm trong ô vuông đó. Miền không có trên giản đồ được coi bằng 0 và miền lớn nhất (toàn bộ ô vuông) được coi bằng đơn vị 1.

ví dụ: tập hợp S là một nhóm các sinh viên và được biểu diễn bởi toàn bộ miền trong hình



$A.B$  hay  $A \cap B$



$A+B$  hay  $A \cup B$

vuông; trong nhóm sinh viên đó có 2 nhóm phụ A và B, với sinh viên thuộc nhóm A có tóc nâu trong khi các sinh viên của nhóm B có mắt xanh.

Khi đó, phần giao của A và B bao gồm các sinh viên có cả mắt xanh và tóc nâu (A.B). Họ là thành viên của cả nhóm A và nhóm B.

Nhóm các sinh viên mà có tóc nâu hoặc mắt xanh có thể được biểu diễn:  $A+B$  (được xem như *hợp* của các nhóm)

## II. BIẾN VÀ HÀM LOGIC

### 1. Khái niệm về biến và hàm logic

+ Biến logic là một khái niệm dùng thay cho thuật ngữ mệnh đề tùy ý, mệnh đề này có thể đúng hoặc sai và không có khả năng một mệnh đề vừa đúng vừa sai, nghĩa là biến logic chỉ nhận một trong hai giá trị là đúng hoặc sai

Ví dụ, câu: “Hôm nay là thứ Năm và trời đang mưa” có thể được biểu diễn như sau:

$$C = A.B.$$

với A : hôm nay là thứ Năm.

B: trời đang mưa.

C: toàn bộ câu.

Khi nào thì toàn bộ câu là đúng?

Có thể thiết lập một bảng liệt kê các trường hợp *đúng(True)* hay *sai(False)* cho A và B:

A	B	C
sai	sai	sai
sai	đúng	sai
đúng	sai	sai
đúng	đúng	đúng

Nếu “1” được sử dụng để thay thế cho phát biểu đúng và “0” cho phát biểu sai thì bảng trên có thể được biểu diễn lại như sau:

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

Như vậy, toàn bộ câu là đúng khi A và B đều đúng còn các trường hợp khác C sai.

+ Một mệnh đề phức tạp được tạo thành từ các mệnh đề đơn giản ban đầu, nó nhận một trong 2 giá trị là đúng hoặc sai. Khi đó, ký hiệu là  $F(A, B, C \dots)$  hay  $F(x_1, x_2, x_3 \dots)$ , người ta gọi đó là hàm logic của các biến A, B, C ... hay của  $x_1, x_2, x_3 \dots$

+ Trong kỹ thuật số các giá trị đúng và sai của biến logic hay hàm logic được ký hiệu là 1 và 0 (đây đơn thuần là ký hiệu mà không phải là chữ số của hệ hai). Thêm nữa việc thực hiện các giá trị logic còn phụ thuộc vào việc chọn các trị số vật lý để biểu diễn.

Ví dụ: với vi mạch thuộc họ TTL người ta đưa ra 2 cách ký hiệu cho mức logic

. Mức logic dương:

$X_i = 1$             ứng với mức điện áp cao 5V

$X_i = 0$             ứng với mức điện áp thấp 0V

. Mức logic âm:

$X_i = 1$             ứng với mức điện áp thấp 0V

$X_i = 0$             ứng với mức điện áp cao 5V

## 2. Các hàm logic sơ cấp

### a. Hàm logic sơ cấp một biến

F <sub>i</sub>	A		F(A)	
	0	1	Biểu thức	Tên gọi
F1	0	0	0	Hằng số 0
F2	0	1	A	Lặp lại A YES
F3	1	0	$\bar{A}$	Đảo biến A NOT
F4	1	1	1	Hằng số 1

**b. Hàm logic hai biến**

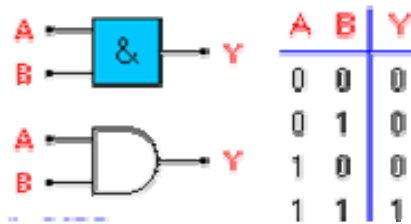
A	0	0	1	1	Ký hiệu và biểu thức đại số của hàm	Tên gọi của hàm
B	0	1	0	1		
F0	0	0	0	0	$F0 = 0$	Hằng số 0
F1	0	0	0	1	$F1 = A.B$	Nhân logic AND
F2	0	0	1	0	$F2 = A.\bar{B}$	Cấm B
F3	0	0	1	1	$F3 = A$	Lặp lại A YES / BUFFER
F4	0	1	0	0	$F4 = B.\bar{A}$	Cấm A INHIBITION
F5	0	1	0	1	$F5 = B$	Lặp lại B YES / BUFFER
F6	0	1	1	0	$F6 = A.\bar{B} + B.\bar{A} = A \oplus B$	Khác dấu / cộng module 2 XOR
F7	0	1	1	1	$F7 = A + B$	Cộng logic OR
F8	1	0	0	0	$F8 = A \downarrow B = \overline{A + B}$	Hàm Pierce NOR
F9	1	0	0	1	$F9 = A \sim B = A.B + \bar{A}.\bar{B}$	Đồng dấu
F10	1	0	1	0	$F10 = \bar{B}$	Bù của B NOT B
F11	1	0	1	1	$F11 = B \rightarrow A = A + \bar{B}$	Kéo theo A IMPLICATION
F12	1	1	0	0	$F12 = \bar{A}$	Bù của A NOT B
F13	1	1	0	1	$F13 = A \rightarrow B = \bar{A} + B$	Kéo theo B IMPLICATION
F14	1	1	1	0	$F14 = A/B = \overline{A.B}$	Hàm Sheffer

						NAND
F15	1	1	1	1	F15 = 1	Hằng số 1

### Các hàm logic sơ cấp

+ Hàm  $F(A,B) = A.B$

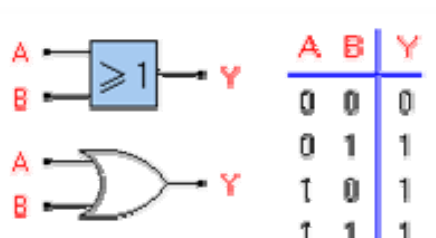
Hàm này thực hiện phép nhân logic của hai biến A và B. Phần tử thực hiện chức năng của hàm trên là phần tử AND (còn gọi là cổng AND). Một cổng AND có hai hay nhiều đầu vào và chỉ có một đầu ra. Đầu ra có mức logic 1 chỉ khi tất cả các đầu vào ở mức 1; và có mức 0 khi một trong các đầu vào ở mức 0. Hình dưới đây chỉ ra ký hiệu và bảng chân lý của cổng AND với 2 đầu vào.



**Tổng quát:** Hàm AND chỉ mang giá trị 1 khi các đầu vào đồng thời bằng 1

+ Hàm  $F(A,B) = A + B$

Hàm này thực hiện phép cộng logic. Phần tử thực hiện là phần tử OR (còn gọi là cổng OR). Cổng OR có mức logic cao khi có ít nhất một đầu vào ở mức 1; và chỉ khi cả 2 đầu vào ở mức logic 0 đầu ra cổng OR mới có mức logic 0. Hàm OR có ký hiệu và bảng chân lý như hình dưới đây:

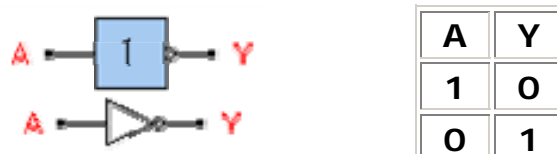


**Tổng quát:** Hàm OR chỉ mang giá trị 0 khi tất cả các đầu vào đồng thời bằng 0

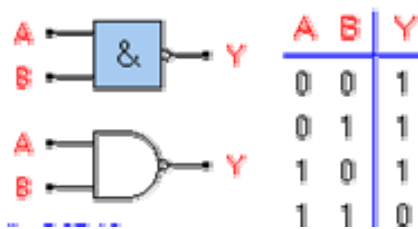
+ Hàm  $F(A) = \bar{A}$

Hàm này thực hiện phép lấy phần tử bù của A. Phần tử thực hiện hàm là phần tử NOT, thường được gọi là cổng đảo, có một đầu vào và một đầu ra. Trạng thái của đầu ra luôn ngược với đầu vào. Ký hiệu của mạch và bảng chân lý như sau:

+ Hàm  $F(A,B) = \overline{A.B}$



Hàm này còn gọi là hàm Sheffer. Phần tử mạch điện thực hiện hàm là phần tử NAND (cổng NAND). Về cơ bản, đây là một cổng AND theo sau là cổng NOT. Đầu ra có mức logic 0 chỉ khi tất cả đầu vào có mức logic 1. Dưới đây là ký hiệu và bảng trạng thái (bảng chân lý) của cổng NAND 2 đầu vào.

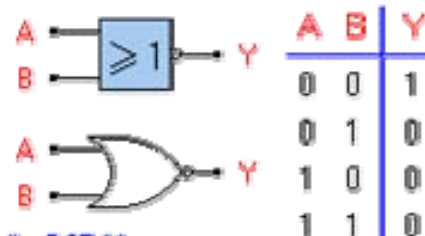


**Tổng quát:** Hàm NAND chỉ mang giá trị 0 khi tất cả các đầu vào đều có mức logic 1

+ Hàm  $F(A,B) = \overline{A \cdot B}$

Hàm này còn gọi là hàm Pierce. Phần tử mạch điện thực hiện hàm là phần tử **NOR (cổng NOR)**. Đây là cổng OR theo sau bởi cổng NOT. Đầu ra có mức logic thấp khi một hay nhiều đầu vào ở mức logic cao; và đầu ra có mức logic cao chỉ khi tất cả đầu vào ở mức thấp. Dưới đây là ký hiệu và bảng chân lý của hàm.

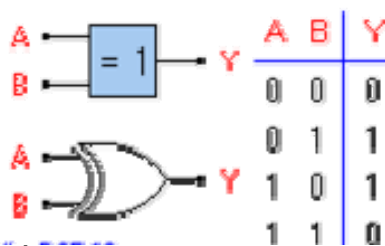
**Tổng quát:** hàm NOR chỉ mang giá trị 1 khi tất cả các đầu vào đều có mức logic 0



+ Hàm  $F(A,B) = A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$

Phần tử thực hiện hàm này là phần tử Exclusive OR (hay cổng XOR). Cổng này có 2 đầu vào. Cổng này là thành phần cơ bản của phép so sánh. Khi 2 đầu vào giống nhau, đầu ra ở mức logic 0; còn khi 2 đầu vào khác nhau, đầu ra có mức logic 1. Dưới đây là ký hiệu và bảng trạng thái.

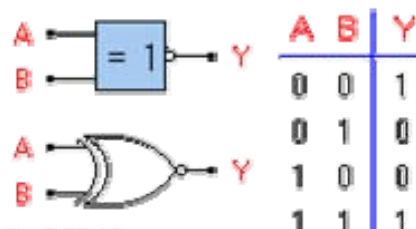
**Tổng quát:** hàm XOR cho giá trị 1 khi số các chữ số 1 trong tổ hợp là một số lẻ. Đây chính là



tính chất của hàm cộng module n biến

+ Hàm  $F(A,B) = \overline{A \oplus B} = A \sim B = A \otimes B = A \cdot B + \overline{A} \cdot \overline{B}$

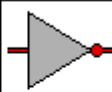
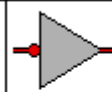
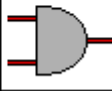

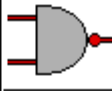

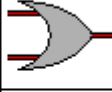
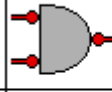

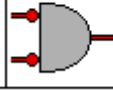
Hàm này gọi là hàm tương đương. Cổng logic thực hiện hàm này là cổng XNOR. Đây là sự kết hợp của hàm XOR và theo sau bởi hàm NOT. Khi 2 đầu vào giống nhau đầu ra ở mức logic 1; còn khi 2 đầu vào khác nhau, đầu ra có mức logic 0. Dưới đây là bảng chân lý và ký hiệu hàm





**Tổng quát:** hàm XNOR sẽ mang giá trị 1 khi số các chữ số 1 trong tổ hợp là một số chẵn (kể cả 0)

**Chú ý:** Với cùng một phần cứng như nhau nhưng nếu sử dụng với các mức logic khác nhau thì chức năng của các cổng sẽ thay đổi. Các cổng logic ở trên được thực hiện với kiểu logic dương. Nếu dùng logic âm thì ta có tương ứng như sau:

Positive Logic		Negative Logic	
INV			INV
AND			OR
NAND			NOR
OR			AND
NOR			NAND

### 3. Hệ hàm đầy đủ

Một hàm logic bất kỳ luôn được biểu diễn dưới dạng tổ hợp của các hàm sơ cấp ở trên. Tuy nhiên, trên thực tế không nhất thiết phải sử dụng hết các hàm sơ cấp đó mà chỉ cần một bộ phận của các hàm sơ cấp.

Một hệ hàm sơ cấp được gọi là đầy đủ nếu có thể biểu diễn một hàm logic bất kỳ bằng cách thực hiện các phép toán của đại số logic lên các phần tử của hệ hàm này.

Các hệ hàm sau được chứng minh là các hệ hàm đầy đủ:

- + Hệ hàm 1: gồm các hàm AND, OR, NOT
- + Hệ hàm 2: gồm các cổng AND, NOT
- + **Hệ hàm 3: NOR**
- + **Hệ hàm 4: NAND**
- + Hệ hàm 5: AND, NOT

...

Giải thích chi tiết hàm NOR và hàm NAND tạo thành các hàm khác như thế nào và trình bày phương pháp thiết kế mạch dùng cổng NOR và cổng NAND

### III. Phương pháp biểu diễn hàm logic

#### 1. Phương pháp dùng bảng giá trị của hàm

Phương pháp này sử dụng bảng ghi **mọi tổ hợp có thể** của biến và giá trị hàm tương ứng. Bảng này còn gọi là bảng hàm hay bảng chân lý (bảng sự thật)

ví dụ: Cho một hàm 3 biến có giá trị như trong bảng ứng với các tổ hợp của biến như sau:

X3	X2	X1	F
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	X

X là ký hiệu mà tại đó giá trị của hàm không xác định (có thể là 0 và có thể là 1)

*Nhận xét:* Phương pháp trên có ưu điểm là trực quan và rõ ràng nhưng nó tỏ ra cồng kềnh và quá rườm rà khi số biến tăng lên. Do đó phương pháp này chỉ dùng để biểu diễn cho các hàm sơ cấp hay các hàm có số biến nhỏ.

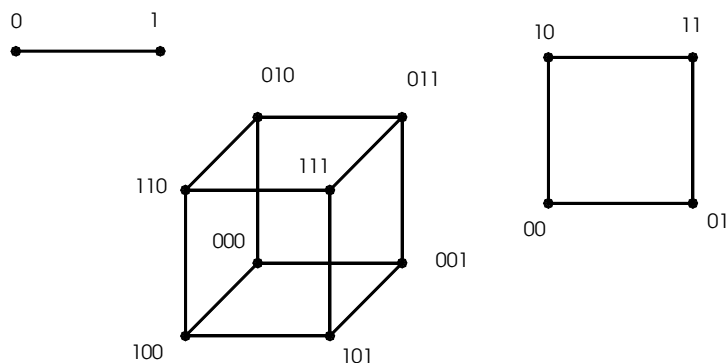
#### 2. Phương pháp hình học

Trong phương pháp này người ta biểu diễn n biến ứng với không gian n chiều. Mỗi tổ hợp của biến được biểu diễn bởi một điểm trong không gian đó

Như vậy, n biến sẽ biểu diễn bởi  $2^n$  điểm với quy ước 2 điểm trên cùng một cạnh chỉ khác nhau ở 1 biến duy nhất.

ví dụ: trường hợp 1, 2 và 3 biến biểu diễn như trong hình dưới đây

#### 3. Phương pháp biểu thức đại số



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

**Dạng chuẩn tắc tuyển đầy đủ** là tuyển của nhiều thành phần, mỗi thành phần là hội gồm đầy đủ n biến

**Dạng chuẩn tắc hội đầy đủ** là hội của nhiều thành phần, mỗi thành phần là tuyển gồm đầy đủ n biến

**a. Cách viết hàm số dưới dạng chuẩn tắc tuyển ( CTT ) đầy đủ:**

- + Số lần hàm bằng 1 sẽ là số tích của n biến
- + Trong mỗi tích các biến có giá trị 1 được giữ nguyên, các biến có giá trị 0 được lấy phủ định
- + Hàm F bằng tổng các tích trên

**b. Cách viết hàm số dưới dạng chuẩn tắc hội ( CTH ) đầy đủ:**

- + Số lần hàm bằng 0 sẽ là số tổng của biểu thức n biến
- + Trong mỗi tổng các biến có giá trị 0 được giữ nguyên, các biến có giá trị 1 được lấy phủ định
- + Hàm F bằng tích các tổng trên

ví dụ: Xây dựng hàm logic của các biến A, B ,C có các giá trị như sau:

$$F(0,0,0) = F(1, 0,0) = F(1,1,0) = 1$$

Các trường hợp khác bằng 0

Thực hiện các bước như trên ta có hàm F viết dưới dạng CTT và CTH như sau:

$$F(A, B, C) = \bar{A}.\bar{B}.\bar{C} + A.\bar{B}.\bar{C} + A.B.\bar{C} = \sum 0,4,6$$

$$F(A, B, C) = (A + B + \bar{C})(A + \bar{B} + C)(A + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + \bar{C}) = \prod 1,2,3,5,7$$

**4. Phương pháp dùng bảng Karnaugh**

Quy tắc xây dựng bảng:

- + Bảng có  $2^n$  ô để biểu diễn hàm n biến, mỗi ô cho một tổ hợp biến
- + Các ô cạnh nhau hay đối xứng nhau chỉ khác nhau 1 biến (ghi theo thứ tự của mã Gray). Các hàng và cột của bảng được ghi các tổ hợp giá trị biến sao cho hàng và cột cạnh nhau hay đối xứng nhau chỉ khác nhau 1 biến
- + Ghi giá trị của hàm ứng với tổ hợp tại ô đó

Chú ý: đối với CTT giá trị hàm bằng 0 được để trống

đối với CTH giá trị hàm bằng 1 được để trống

Hàm không xác định tại tổ hợp nào thì đánh dấu X vào ô đó

ví dụ: biểu diễn hàm sau bằng bảng Karnaugh

$$F(A, B, C) = \sum 0,2,5 \text{ với } N = 1, 4 \quad (\text{cách viết theo CTT})$$

$$F(A, B, C) = \prod 3,6,7 \text{ với } N = 1, 4 \quad (\text{cách viết theo CTH})$$

Với N là tập hợp của tổ hợp biến mà tại đó giá trị của hàm không xác định.

Thực hiện như các bước ở trên ta có bảng Karnaugh biểu diễn cho hàm F theo CTT như sau:

A \ BC	00	01	11	10
0	1	X		1
1	X	1		

Hoặc có thể biểu diễn hàm F theo CTH như sau:

A \ BC	00	01	11	10
0		X	0	
1	X		0	0

## CHƯƠNG 3

### TỐI THIỂU HOÁ HÀM BOOLEAN

#### I. PHƯƠNG PHÁP TỐI THIỂU HOÁ

##### 1. Khái niệm tối thiểu hoá

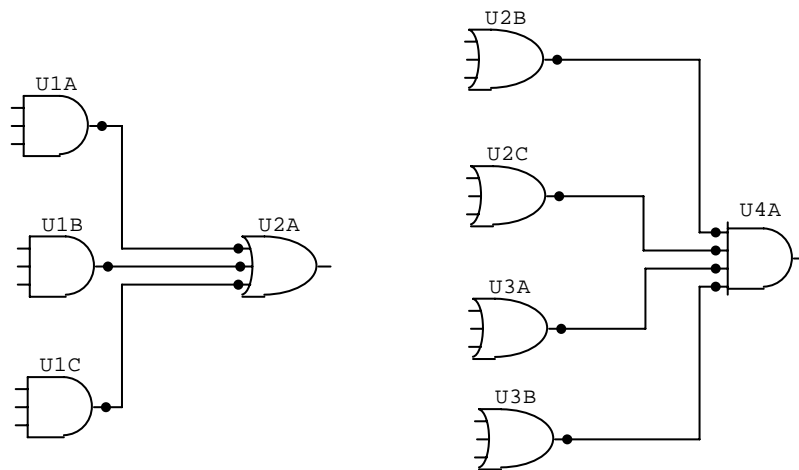
Tối thiểu hoá là tìm dạng biểu diễn đại số đơn giản nhất của hàm. Khi đó sẽ giảm được tối đa số cổng để thực hiện hàm. Đây là yêu cầu rất cần quan tâm vì nó giúp cho việc thực hiện mạch được đơn giản và hiệu quả.

Ví dụ: Cho hàm có dạng CTT và CTH đầy đủ như sau:

$$F = \overline{X_3} \cdot X_2 \cdot \overline{X_1} + X_3 \cdot X_2 \cdot \overline{X_1} + X_3 \cdot X_2 \cdot X_1$$

$$F = (X_3 + X_2 + X_1)(X_3 + X_2 + \overline{X_1})(\overline{X_3} + X_2 + X_1)(\overline{X_3} + X_2 + \overline{X_1})$$

Khi đó sơ đồ cổng thực hiện hàm sẽ có dạng:



Tuy nhiên nếu sử dụng bảng chân lý của hàm ta có:

X3	X2	X1	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

Từ bảng chân lý dễ dàng thấy  $F = X_2$ . Rõ ràng biểu thức này đơn giản hơn rất nhiều so với biểu thức ở trên, vì thế mạch lúc này cũng chỉ là một bộ đệm cho  $X_2$  mà thôi



Cũng có một số yếu tố khác ngoài giá thành ảnh hưởng đến độ phức tạp của mạch cần được quan tâm. Một trong các yếu tố là thời gian trễ truyền đạt, là khoảng thời gian tính từ lúc có sự thay đổi tại đầu vào tới khi có sự thay đổi kết quả tại đầu ra. Càng nhiều cổng được mắc nối tiếp với nhau thì thời gian trễ này càng lớn.

Ví dụ với hàm :  $f = A*B*C + A*B*\bar{C} + A*\bar{B}*C + A*\bar{B}*\bar{C}$  1)

là một dạng tối thiểu và đầu ra có mức trễ của cổng AND thêm với mức trễ của cổng OR.

Tuy nhiên, cũng với hàm này theo luật phân phối, ta được:

$$f = A*(B*C + B*\bar{C} + D). \quad 2).$$

Hàm này có thời gian trễ lớn hơn hàm trước vì nó gồm mức trễ của 3 cổng. Bởi thế, dù rẻ hơn, nó có thời gian trễ lớn hơn.

Một yếu tố đáng quan tâm khác là tải của đầu vào. Xét 1). tín hiệu A phải điều khiển 3 tải (3 cổng), trong khi với 2). tải chỉ có một cổng.

Tới nay vẫn chưa có phương pháp tối ưu nào có thể thực hiện việc tối thiểu hoá một cách tối ưu. Việc tối thiểu hoá hàm logic có thể thực hiện bằng một trong hai cách cơ bản là:

+ Biến đổi đại số

+ Thuật toán

### 2. Phương pháp tối thiểu hoá hàm logic bằng biến đổi đại số

Trong trường hợp số biến ít và hàm được biểu diễn bằng phương pháp giải tích người ta có thể thực hiện biến đổi trực tiếp hàm theo các tính chất của đại số

Ví dụ: dùng phương pháp biến đổi đại số ta thực hiện rút gọn hàm f như sau:

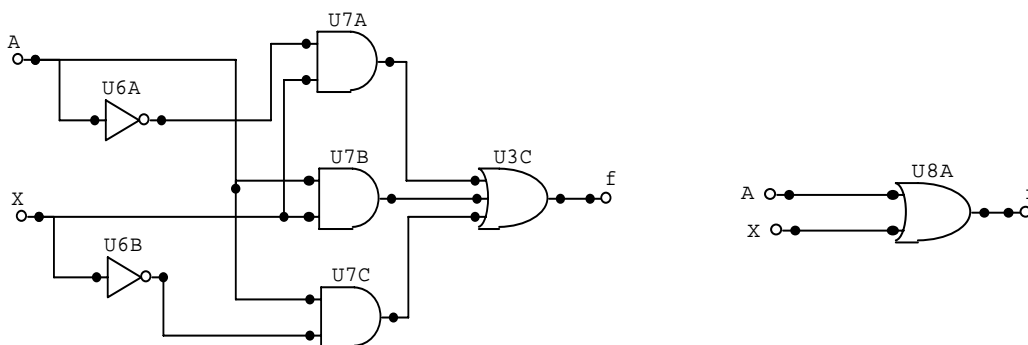
$$f = \bar{A}.X + A.X + A.\bar{X}$$

$$f = \bar{A}.X + A.X + A.X + A.\bar{X}$$

$$f = X(A + \bar{A}) + A(X + \bar{X})$$

$$f = X + A$$

rõ ràng là hàm f đã được đơn giản đi rất nhiều thay vì một hàm phức tạp



### 3. Nhóm các phương pháp tối thiểu hoá theo thuật toán

**Một số khái niệm:**

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

Đỉnh 1 là đỉnh mà tại đó hàm số bằng 1

Đỉnh 0 là đỉnh mà tại đó hàm số bằng 0

Đỉnh không xác định là đỉnh tại đó hàm không xác định (ký hiệu là X)

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

**Tích cực tiểu** là một tích mà tại đó hàm bằng 1 hoặc không xác định với thành phần các biến không bỏ bớt được nữa. Tích cực tiểu là biểu diễn của 1 nhóm  $2^k$  đỉnh. Tích cực tiểu này phủ các đỉnh hay các đỉnh chứa trong tích cực tiểu, nghĩa là dùng tích cực tiểu để biểu diễn tối đa số đỉnh với số biến ít nhất. Cơ sở toán học của việc tìm tích cực tiểu là áp dụng phép dán:

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

**Tích quan trọng** là một tích cực tiểu phủ ít nhất 1 đỉnh 1. Nó nhất thiết phải xuất hiện trong biểu thức cuối cùng của bài toán. Tập hợp các tích quan trọng chính là **phủ tối thiểu**, kết quả cuối cùng của bài toán.

**Chú ý:** Khi tiến hành với hàm viết dưới dạng CTH đầy đủ thì thay các đỉnh 1 bằng đỉnh 0. Các khái niệm tổng và tích cũng đổi chỗ cho nhau. Nghĩa là:

Đỉnh là tổng đầy đủ n biến

Biểu diễn hàm bằng tích các tổng

Tổng cực tiểu

Tổng quan trọng

Phủ tối thiểu là số tổng quan trọng ít nhất mà phủ hết được số đỉnh 0

Giá trị của biến sẽ giữ nguyên nếu có giá trị 0 và đảo nếu có giá trị 1

**Quá trình tối thiểu hoá gồm các bước như sau:**

+ Biểu diễn hàm số dưới dạng CTT đầy đủ với tập các đỉnh 1 và đỉnh không xác định hoặc CTH đầy đủ với tập các đỉnh 0 và đỉnh không xác định

+ Tìm các tích cực tiểu

+ Tìm các phủ tối thiểu

+ Đưa ra cách biểu diễn mới của hàm

### a. Phương pháp dùng bảng Karnaugh.

Bảng Karnaugh là một bảng có  $2^n$  ô, mỗi ô tương ứng với một tổ hợp trong bảng trạng thái và chứa các giá trị đầu ra tương ứng. Một đặc trưng của biểu đồ này là luôn sắp xếp sao cho chỉ có sự thay đổi của một biến khi chuyển từ ô này sang ô kề

Trong bảng ta sẽ viết được:

		L1					
		L2		R1			
		R2		00	01	11	10
00		0	0	0	0		
01		0	0	0	0		
11		*1	*1	0	1		
10		0	0	0	0		

chú ý đến 2 dấu hoa thị, ta

$$A = \overline{L1} . \overline{R1} . L2.R2 + \overline{L1} . R1.L2.R2$$

Sử dụng các định lý của Đại số Boolean, có thể viết lại:

$$\begin{aligned} A &= \overline{L1} . L2.R2.( \overline{R1} + R1) \\ &= \overline{L1} . L2.R2. 1 \\ &= \overline{L1} . L2.R2. \end{aligned}$$

Như vậy, hàm được tối thiểu hoá gồm một cổng AND 3 đầu vào.

Nguyên lý thiết lập biểu đồ Karnaugh tại các ô kề nhau, giá trị “1” được nhóm lại với Kích thước của nhóm là lũy thừa của 2 (ví dụ: 2 ô, 16 ô, 32 ô ...). Ví dụ 4 ô của cột thứ tư trong hình bên có thể được nhóm. Như vậy, toàn bộ được tối giản thành  $A . \overline{B}$ , chính là các phần tử của cả nhóm. Các phần tử có giá trị khác nhau sẽ không xuất hiện. Kết quả này cũng nhận nếu ta áp dụng các định lý của đại số Boolean này như sau:

		B			
		00	01	11	10
C	D	1	0	0	1
	0	1	0	0	1
	1	0	1	0	1
	0	1	0	0	1

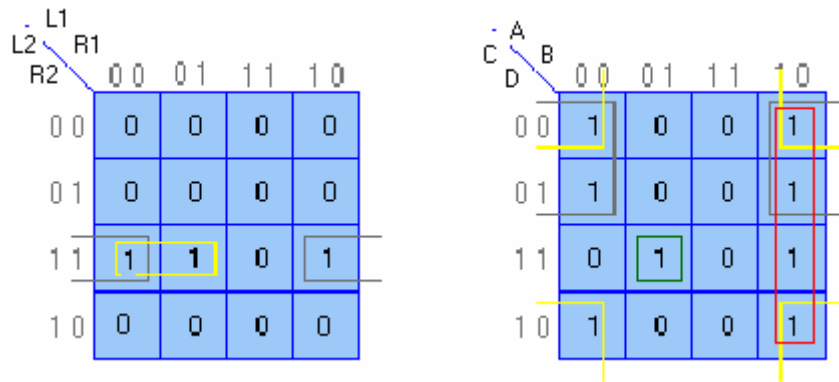
chính là nhau. ô, 4 ô, 8 bảng ở nhóm sẽ chung (C và D) được cho 4 ô

$$\begin{aligned} f &= A . \overline{B} . \overline{C} . \overline{D} + A . \overline{B} . C . D + A . \overline{B} . C . \overline{D} \\ &= A . \overline{B} . \overline{C} . (\overline{D} + D) + A . \overline{B} . C . (D + \overline{D}) \\ &= A . \overline{B} . \overline{C} + A . \overline{B} . C = A . \overline{B} . (C + \overline{C}) \\ &= A . \overline{B} \end{aligned}$$

*Chú ý:* Bảng Karnaugh, giống như bản đồ thế giới, phía bên phải sẽ tiếp liền phía bên trái, nên có thể nhóm các ô nằm đối diện nhau. Nguyên lý này cũng được áp dụng cho bên trên và bên dưới. (tức là chúng ta nhóm theo kiểu đối xứng hoặc liền kề)

Ví dụ, có thể nhóm 4 ô ở 4 góc của biểu đồ như hình dưới đây





Từ các nhận xét ở trên ta rút ra được *các bước tiến hành* tối thiểu hoá bằng bảng Karnaugh cho dạng CTT là:

1, Biểu diễn hàm đã cho trên bảng Karnaugh

2, Xác định các tích cực tiểu của hàm (tích cực tiểu tìm được bằng cách dán  $2^k$  ô có giá trị 1 hoặc X với k tối đa, các ô này gần kề hoặc đối xứng nhau)

3, Tìm phủ tối thiểu là chọn một số ít nhất các nhóm tích cực tiểu sao cho phủ hết được các đỉnh 1 của hàm

*Chú ý:* . Quá trình hoàn toàn tương tự khi hàm biểu diễn ở dạng CTH

. Khi lập bảng Karnaugh với CTT những ô bằng 0 nên để trống còn ở dạng CTH thì bỏ trống những ô có giá trị 1.

**b. Tối thiểu hoá bằng phương pháp Quine - Mc.Cluskey**

Phương pháp này được thực hiện cho hàm biểu diễn dưới dạng CTT

**Các bước tiến hành:**

**Bước 1:** Tìm tích cực tiểu

. Xác định đỉnh 1 và X

. Sắp xếp các tổ hợp biến theo số lượng chữ số 1 có trong chúng

. So sánh mỗi tổ hợp thuộc nhóm i với tổ hợp thuộc nhóm (i + 1). Nếu 2 tổ hợp đó chỉ khác nhau 1 cột số thì kết hợp 2 tổ hợp đó thành một tổ hợp mới, trong đó sử dụng dấu – thay cho cột số khác nhau. Đánh dấu vào 2 tổ hợp vừa kết hợp

. Loại bỏ các tổ hợp giống nhau và lặp lại bước trên cho đến khi hết các tổ hợp có khả năng kết hợp

. Tập hợp các tổ hợp trong bảng cuối và các tổ hợp không bị đánh dấu chính là tập các tích cực tiểu

**Bước 2:** Tìm phủ tối thiểu

. Lập bảng có cột là các giá trị có đỉnh là 1 (các giá trị này thường ghi theo hệ đếm 10 cho tiện theo dõi), hàng là các tích cực tiểu

. Đánh dấu X vào ô mà tích cực tiểu ở hàng phủ đỉnh ở cột. Cột có 1 dấu X chính là tích quan trọng

. Loại bỏ các cột đã được phủ trong tích quan trọng

. Loại các tích quan trọng khỏi hàng

. Lập bảng mới và tiếp tục quá trình đến khi tất cả các đỉnh đều được phủ

Ví dụ: Tối thiểu hoá hàm sau bằng phương pháp Quine – Mc. Cluskey như sau:

$$\text{Hàm } f = \sum(0,2,5,8,9,10,11)$$

		Sắp xếp lại		Thực hiện phép dán			
Hệ 10	Hệ 2	Hệ 10	Hệ 2				
0	0000	0	0000	(0,2)	00-0	(0,2,8,10)	-0-0
2	0010	2	0010	(0,8)	-000	(0,8,2,10)	-0-0
5	0101	8	1000	(2,10)	-010	(8,9,10,11)	10- -
8	1000	5	0101	(8,9)	100-	(8,10,9,11)	10- -
9	1001	9	1001	(8,10)	10-0		
10	1010	10	1010	(9,11)	10-1		
11	1011	11	1011	(10,11)	101-		

Vậy kết quả cuối cùng là: -0-0 và 10- -

$$\text{Hay } f = \overline{B.D} + \overline{AB}$$

## CHƯƠNG 4: GIỚI THIỆU VI MẠCH SỐ

### I. ĐỊNH NGHĨA VÀ PHÂN LOẠI

#### 1. Định nghĩa

Vi mạch là những linh kiện điện tử có một chức năng xác định và được chế tạo bằng một công nghệ riêng. Vi mạch hiện đại thường đa năng và có thể sử dụng linh hoạt trong nhiều thiết bị điện tử khác nhau

Người ta phân loại theo một số tiêu chí sau:

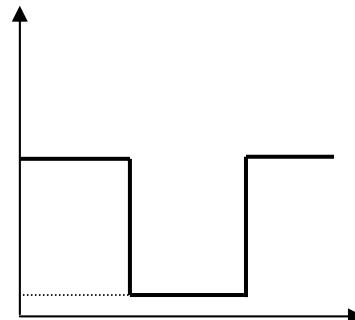
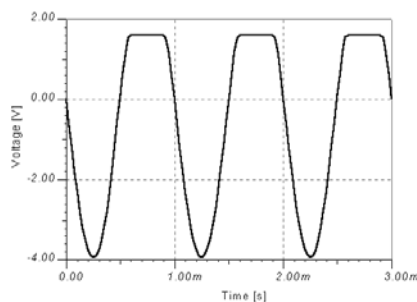
- + Phân loại theo bản chất của tín hiệu điện vào / ra của vi mạch
- + Phân loại theo mật độ tích hợp
- + Phân loại theo công nghệ chế tạo

#### 2. Phân loại vi mạch theo bản chất của tín hiệu vào / ra

Như đã biết, tín hiệu điện được phân thành 2 loại là tín hiệu tương tự và tín hiệu số.

- + Tín hiệu tương tự (analog) là tín hiệu có biên độ biến thiên liên tục theo thời gian
- + Tín hiệu số (digital) là tín hiệu có biên độ ở một trong hai giá trị hữu hạn mang ý nghĩa logic 0 hoặc 1, ứng với 2 mức thấp và cao. Tín hiệu số gián đoạn theo thời gian.

Nếu ký hiệu X, Y là tín hiệu vào và ra của vi mạch, theo bản chất của tín hiệu vào / ra này ta sẽ có các loại vi mạch sau:



Tín hiệu vào	Tín hiệu ra	Loại vi mạch
Tương tự	Tương tự	Tương tự
Số	Số	Số
Tương tự	Số	ADC / analog-digital converter
Số	Tương tự	DCA / digital-analog converter

Trong phạm vi của môn kỹ thuật số chúng ta chỉ xét tới vi mạch số, nghĩa là cả đầu vào lẫn đầu ra đều là tín hiệu số.

Các vi mạch số này bao gồm từ các cổng logic đơn giản như AND, OR, NOR, NAND ... flip-flop, MUX, DEMUX, Memory ... đến các loại mạch cực kỳ phức tạp như các bộ vi xử lý

### 3. Phân loại theo mật độ tích hợp

Mật độ tích hợp được định nghĩa là tổng các phần tử tích cực (transistor) hoặc cổng logic chứa trên một đơn vị diện tích của màng tinh thể bán dẫn trong vi mạch

ví dụ: Bộ vi xử lý Pentium III của Intel có mật độ tích hợp là 9triệu transistor trên 1 inch vuông

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

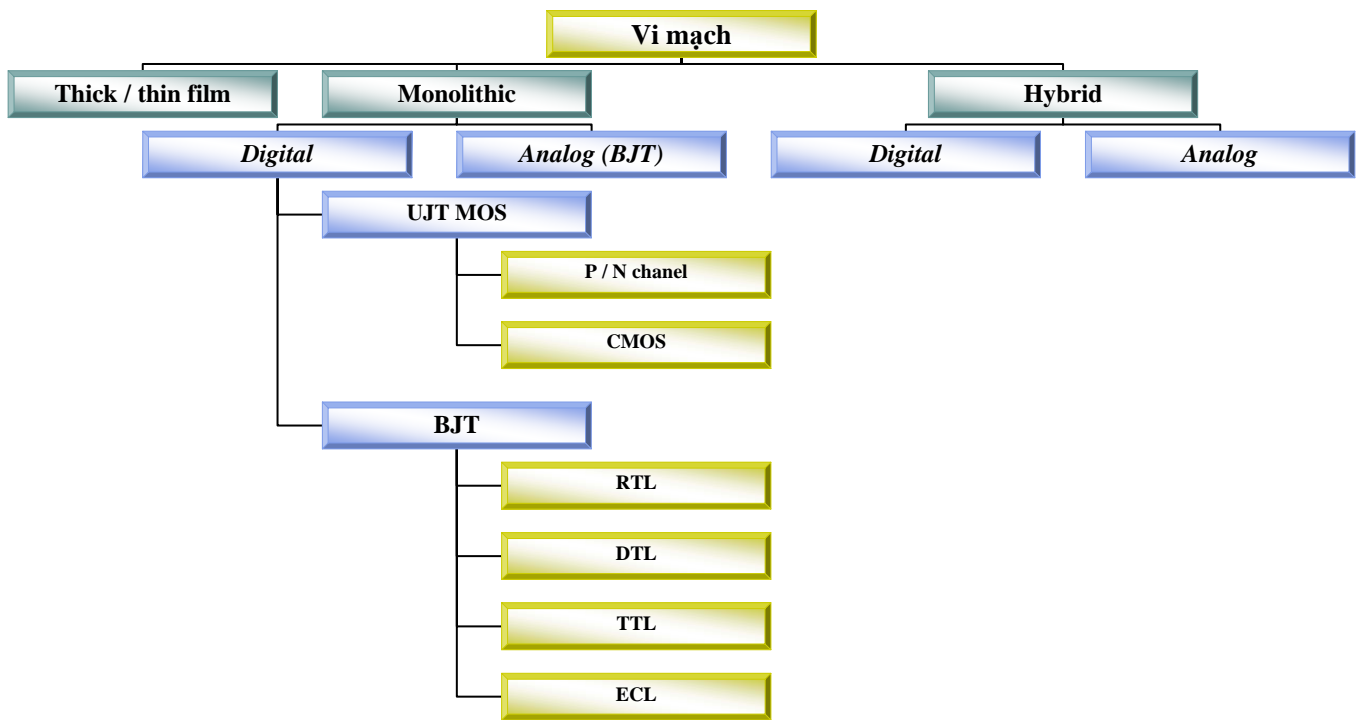
Những thông số trên phần nào cho thấy độ phức tạp của mạch. Phân loại theo mức độ tích hợp ta có

Loại mạch	Số transistor	Số cổng logic	Ví dụ
SSI – Vi mạch cỡ nhỏ	Hàng chục	1 - 10	Gate, flip-flop
MSI – Vi mạch cỡ trung bình	Hàng trăm	10 - 100	Gate, counter, shift-register, encoder, small memory
LSI – Vi mạch cỡ lớn	Hàng nghìn	100 - 1000	Larger Memory, microprocessor 4 / 8bit
VLSI - Vi mạch cỡ rất lớn	Hàng vạn	> 1.000	MP 16/32bit, console i/o 8086, Z8000
ULSI – Vi mạch cỡ cực lớn	Hàng triệu	> 10.000	MP 64bit

### 4. Phân loại theo công nghệ chế tạo

IC có thể chia ra làm 4 loại: IC màng mỏng/ màng dày; IC khối rắn; và IC lai

Dưới đây là các hướng phát triển vi mạch theo công nghệ chế tạo



### **a. Vi mạch màng mỏng / màng dày**

Các IC loại này được chế tạo bằng cách lắng đọng những vật liệu nhất định trên một đế cách điện (ví dụ như gốm, sứ...). Sau hàng loạt các quá trình tạo “mask” trên đế tạo thành điện trở, điện dung hay điện cảm. Các linh kiện tích cực như diode, transistor ... sẽ được chế tạo theo cách thông thường với kích thước nhỏ (thường là FET). Mạch này cho độ tích hợp khá cao nhưng không bằng loại đơn khối, tuy nhiên lại có khả năng chịu đựng điện áp và nhiệt tốt hơn. IC màng mỏng và màng dày được sử dụng cho các mạch đòi hỏi độ chính xác cao

### **b. Vi mạch bán dẫn khối rắn**

IC monolithic được tạo ra hoàn toàn trên một đơn vị tinh thể chất bán dẫn nền là Si, các chất bán dẫn khác sẽ được khuếch tán vào trong chất nền để tạo ra nhiều loại mặt ghép khác nhau. Những mặt ghép này có thể tạo thành điện trở, điện dung, diode hay transistor.

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 ở nhiệt độ cao.

Quá trình tạo mask là quá trình trong đó người ta tiến hành oxy hoá bề mặt chất bán dẫn, tức là lấp kín bề mặt của nó bằng SiO<sub>2</sub>. Sau đó phủ một lớp cảm quang lên trên bề mặt SiO<sub>2</sub>. Dạng mạch thu nhỏ, chụp lên phim tạo thành khuôn sáng. Đặt khuôn sáng lên bề mặt chất cảm quang, chiếu ánh sáng vào ta sẽ thu được dạng mạch theo yêu cầu. Dùng hoá chất ăn mòn các rãnh, loại bỏ chất cảm quang để thực hiện khuếch tán chất vào. Mask được tạo thành bằng phương pháp như trên gọi là phương pháp quang khắc.

Vi mạch monolithic có 2 loại là mạch lưỡng cực và mạch MOS, ngày nay vi mạch MOS trở nên phổ biến do dễ chế tạo, diện tích nhỏ nên khả năng tích hợp cao.

### **c. Vi mạch lai**

Đây là sự kết hợp của 2 loại vi mạch trên. IC lai có thể bao gồm nhiều tinh thể monolithic được ghép với nhau thành khối, đó cũng có thể là sự kết hợp giữa mạch monolithic với mạch màng mỏng thụ động.

IC lai mang đầy đủ ưu điểm của 2 loại vi mạch monolithic và màng mỏng / màng dày như kích thước nhỏ gọn mà công suất lại lớn, độ chính xác cao ...

## II. CÁC THÔNG SỐ CHÍNH CỦA VI MẠCH SỐ

### 1. Mức logic

Mức logic là giá trị điện áp vào / ra được quy định cho các số nhị phân 0 và 1. Mức logic là thông số quan trọng nhất của vi mạch số, nhờ thông số này mà ta có thể dễ dàng nhận biết được những trạng thái logic ra và vào bằng cách đo nhờ vôn kế hoặc oscilloscope.

Giữa các thông số khác nhau (điện áp, dòng, thời gian...) đặc trưng cho một họ logic thì các tham số điện tĩnh đặc biệt quan trọng bởi vì chúng xác định giới hạn dòng và áp tại đầu ra và đầu vào.

Mỗi trạng thái logic của linh kiện (High hay Low) được xác định bởi dải điện áp cho phép.

Tổng cộng sẽ có 4 dải điện áp, mỗi dải được xác định bởi 2 giới hạn điện áp; như vậy sẽ có 8 giá trị điện áp đặc trưng cho mỗi họ logic.

### Các mức và dải điện áp cho phép.

Ta có quan hệ điện áp đầu vào và ra sau:

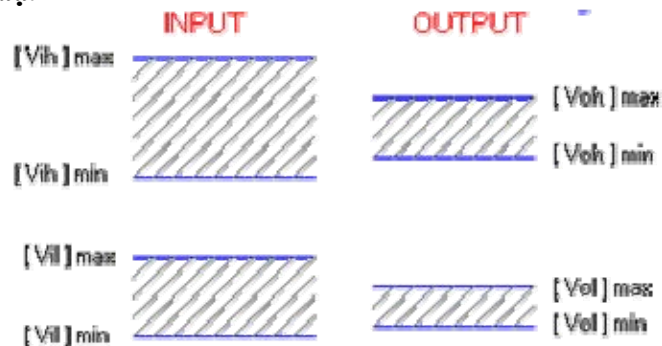
$$V_{ol_{max}} \leq V_{il_{max}}$$

$$V_{o1_{min}} \geq V_{i1_{min}}$$

$$V_{oh_{max}} \leq V_{ih_{min}}$$

$$V_{oh_{min}} \geq V_{ih_{min}}$$

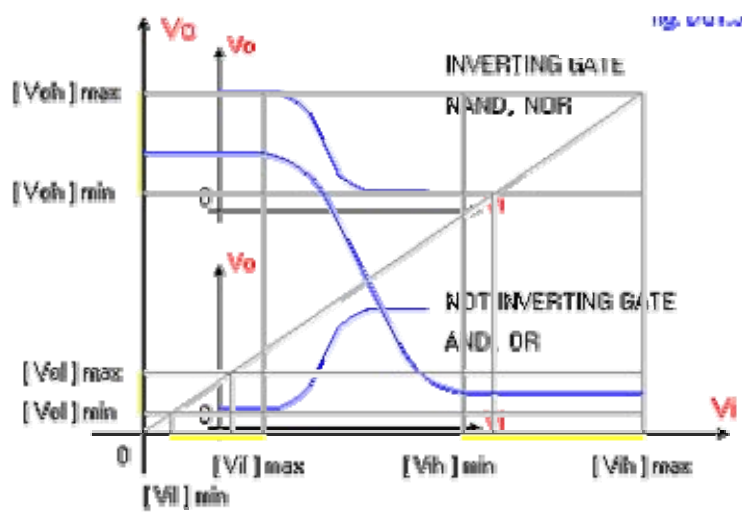
### 2. Đặc tính truyền đạt



Đường đặc tuyến truyền đạt (transfer characteristic) là đường cong chỉ ra mối quan hệ giữa điện áp vào và ra.

Dưới đây là đường đặc tuyến truyền đạt của một cổng đảo (trên) và một cổng không đảo (dưới).

Với cổng đảo, dải điện áp và các mức giới hạn sẽ được xác định như hình dưới đây:



Trong thực tế, điện áp vào và ra lớn nhất được cho bởi giá trị điện áp cung cấp  $V_{cc}$  và các giá trị nhỏ nhất là bằng zero tức bằng điện áp đất.

Như vậy, sẽ chỉ còn 4 giá trị điện áp giới hạn và có quan hệ:

$$V_{ol_{max}} \leq V_{il_{max}} .$$

$$V_{oh_{min}} \geq V_{ih_{min}} .$$

**Bảng so sánh giá trị điện áp vào và ra của các họ logic TTL, CMOS, HCT và họ ECL.**

Chú ý: Điện áp cung cấp khác nhau.

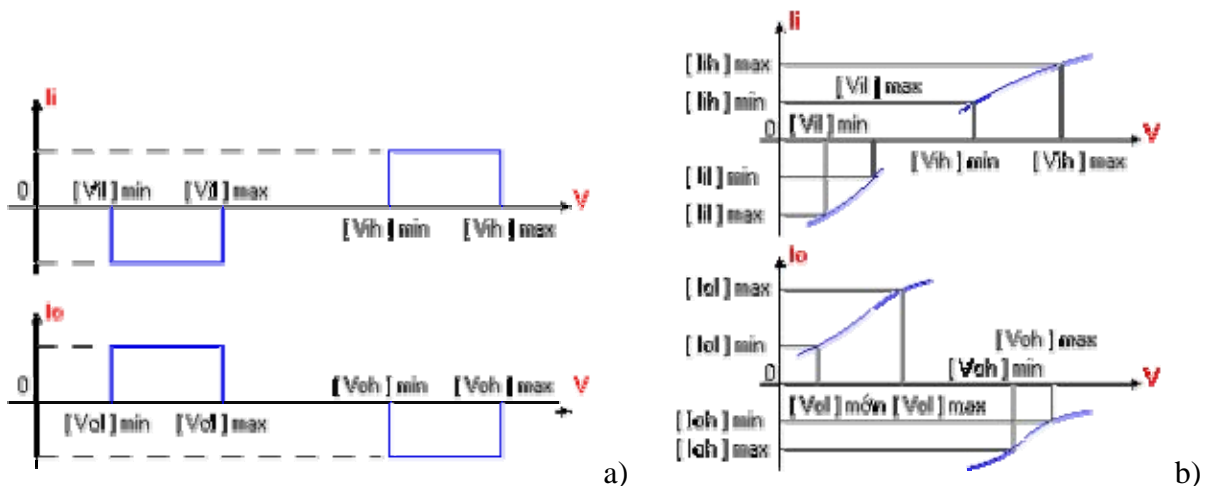
$I_L$	L-LS	CMOS	HCT	ECL
$V_{cc} [V]$	+5	+15	+5	-5.2
$[V_{ih}]_{min} [V]$	2	10	2	-1.2
$[V_{il}]_{max} [V]$	0.8	5	0.8	-1.4
$[V_{oh}]_{min} [V]$	2.4	14.85	4.4	-0.9
$[V_{ol}]_{max} [V]$	0.4	0.05	0.1	-1.7

### 3. Các thông số về dòng điện.

Các dải giới hạn về dòng điện cũng được định nghĩa tương tự như dải giới hạn điện áp. Các giá trị dòng ra là cao hơn các giá trị vào.

Chiều của dòng điện được quy ước như sau: chiều dương là chiều dòng đi vào cực của linh kiện còn chiều âm là chiều dòng đi ra khỏi linh kiện.

Ví dụ: các thông số đặc tính lý tưởng của cổng NAND được chỉ ra ở hình a. Tuy nhiên, trong thực tế các đường đặc tính này là các đường cong như trong hình b.



Nói chung, với các họ logic ta đều có:

$$V_{ol_{min}} = 0.$$

$$V_{oh_{max}} = V_{cc}.$$

bởi thế, chỉ còn 4 giá trị giới hạn về dòng điện:

$$I_{il_{max}} .$$

$$I_{ih_{max}} .$$

$I_{ol_{Min}}$ .

$I_{oh_{Min}}$ .

ta có quan hệ:

$I_{ol_{Min}} \geq I_{il_{Max}}$ .

$I_{oh_{Min}} \geq I_{ih_{Max}}$ .

Bảng sau sẽ chỉ ra các giá trị dòng ra và vào tương ứng với các họ logic TTL, CMOS và ECL.

	TTL-LS	CMOS	HCT	ECL
$V_{cc}$ [V]	+5	3 – 18	+5	-5.2
$I_{ih}$	20 $\mu$ A	0.1 $\mu$ A	0.1 $\mu$ A	300 $\mu$ A
$I_{il}$	-0.4 mA	-0.1 $\mu$ A	-0.1 $\mu$ A	-350 $\mu$ A
$I_{oh}$	-400 $\mu$ A	-0.4 mA	-5 mA	12 mA
$I_{ol}$	8 mA	0.4 mA	-5 mA	-12 mA

#### ***Dòng tiêu thụ trong trạng thái tĩnh.***

Tại trạng thái tĩnh, dòng cung cấp là tổng dòng tiêu thụ của từng linh kiện khi các cổng của nó là ổn định, không xảy ra sự chuyển trạng thái.

Các nhà sản xuất sẽ cung cấp các thông tin về dòng tĩnh “quiescent” cho từng linh kiện và dưới các điều kiện thử đặc biệt.

Bảng sau chỉ ra các giá trị dòng max cho một vài linh kiện của các họ logic.

	TTL-LS	CMOS	HCT	ECL
Device	74LS00	CD4011	74HCT00	HD10102
Gates	4 NANDs	4 NANDs	4 NANDs	4 OR / NOR
Maximum absorbed current [static state]	4.4 mA	0.5 $\mu$ A	2 $\mu$ A	80 mA

#### ***4. Công suất tiêu thụ.***

Công suất tiêu thụ bởi các linh kiện logic chia thành 2 loại: tĩnh và động.

Thành phần công suất tĩnh tạo nên do dòng tĩnh.

Thành phần động tạo nên do dòng điện yêu cầu để tích và phóng cho điện dung tải đầu ra; do dòng điện yêu cầu bởi các điện dung nội; và do dòng điện cần thiết để tạo trạng thái dẫn cho các Transistor đầu ra.

Với các linh kiện ECL, công suất tiêu thụ chủ yếu do hoạt động trong miền tích cực.

Công suất tiêu thụ được tính theo công thức:

$$P = I_{cc} \cdot V_{cc} + C_{pd} \cdot V_{cc}^2 \cdot f_i + \sum (C_L \cdot V_{cc}^2 \cdot f_o).$$

với:  $I_{cc}$ : dòng tĩnh.

$V_{cc}$ : điện áp cung cấp.

$f_i$ : tần số tín hiệu vào.



$f_o$  : tần số tín hiệu ra.

$C_{pd}$  : điện dung tương đương đầu vào.

$C_L$  : điện dung tải.

Thành phần công suất tĩnh tiêu thụ của các linh kiện LS-TTL cao hơn rất nhiều so với linh kiện CMOS nhưng lại nhỏ hơn so với linh kiện họ ECL.

Tổng công suất động của 1 linh kiện họ CMOS phụ thuộc chủ yếu vào tần số, không giống như linh kiện họ TTL.

Bảng công suất tiêu thụ của các linh kiện trong một số họ logic.

### 5. Hệ số tải FAN-IN; FAN-OUT

fig. D0	TL-LS	CMOS	HCT	ECL
Device	74LS00	CD4011	74HCT00	HD10102
Gates	4 NANDs	4 NANDs	4 NANDs	4 OR / NOR
Typical power dissipation (static state)	10 mW	0.15 $\mu$ W	11 $\mu$ W	100 mW

#### Hệ số tải đầu vào FAN-IN.

FAN-IN là tỷ số giữa dòng vào của 1 linh kiện cụ thể và dòng vào của 1 mạch chuẩn.

Thông thường, mạch được lấy làm chuẩn sẽ là 1 cổng logic cơ bản của cùng họ logic. Hệ số này được dùng nhiều trong quá khứ khi các họ logic mới được giới thiệu. Ngày nay, hệ số FAN-IN không được nhắc đến trong các giới thiệu sản phẩm data-sheet của các nhà sản xuất.

Hệ số tải FAN-OUT quan trọng hơn và được dùng nhiều hơn.

#### Hệ số tải đầu ra FAN-OUT.

FAN-OUT là tỷ số giữa dòng ra nhỏ nhất của 1 linh kiện logic và dòng ra của 1 linh kiện cụ thể được lấy làm chuẩn.

FAN-OUT cũng có thể được định nghĩa là số lớn nhất các cổng có thể được điều khiển từ 1 đầu ra, mà không làm vượt quá các giới hạn ra của linh kiện.

Hệ số FAN-OUT sẽ được tính với cả mức điện áp cao cũng như mức thấp và hệ số nhỏ hơn sẽ được chọn.

Trong trường hợp cổng LS-TTL, ta có:

$$FAN - OUT (H) = \frac{I_{oh}}{I_{ih}} = \frac{400 \mu A}{20 \mu A} = 20$$

$$FAN - OUT (L) = \frac{I_{ol}}{I_{il}} = \frac{8 mA}{0.4 mA} = 20$$

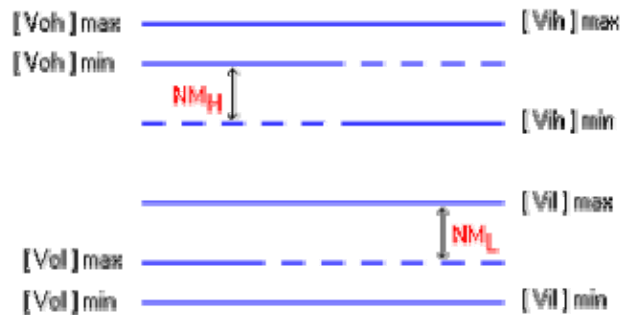
Bảng sau chỉ ra hệ số FAN-OUT của các họ logic:

TTL-LS	CMOS	HCT	ECL
--------	------	-----	-----

FAN-OUT	20	100	100	34
---------	----	-----	-----	----

### 6. Khoảng lề chống nhiễu (Noise Margin).

Nếu đầu ra của 1 cổng logic được nối với đầu vào của 1 cổng logic cùng họ, bất kể nhiều chông lần nào cũng không thể gây ra lỗi nếu biên độ của nó nhỏ hơn khoảng lề chống nhiễu.



Khoảng lề chống nhiễu (biểu diễn bởi NM) có đơn vị là Volts.

Tham số này được định nghĩa cho mức logic thấp ( $NM_L$ ) cũng như mức logic cao ( $NM_H$ ).

Ta có phương trình biểu diễn mối quan hệ của NM với các mức điện áp

$$NM_L = V_{il_{Max}} - V_{ol_{Max}}$$

$$NM_H = V_{oh_{Min}} - V_{ih_{Min}}$$

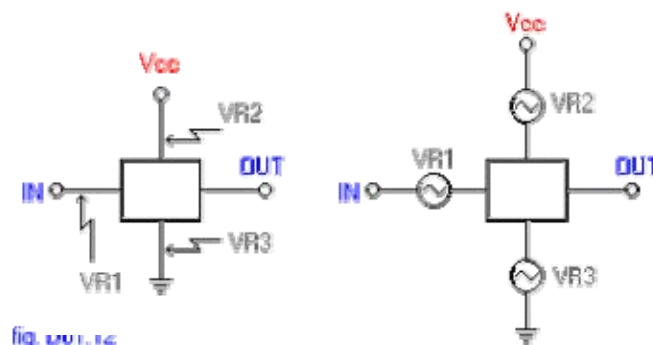
Thông thường, có một vài nguồn nhiễu ac, và ảnh hưởng của nó phụ thuộc vào các nhân tố sau:

trở kháng vào và ra, điện dung ảnh hưởng trên đường vào cũng như bản thân nhiễu của đường dây.

nhiễu từ nguồn cung cấp.

nhiễu đất.

Những yếu tố này tạo nên nhiễu như hình sau; nhiễu được biểu diễn như các nguồn điện áp.



Các nhiễu xung thông thường khó loại bỏ vì chúng được tạo nên bởi các sự cố mà rất khó phát hiện và chúng được truyền đi bởi các thành phần ký sinh.

Bảng dưới đây so sánh các loại nhiễu với các họ logic khác nhau

Số liệu trong bảng chỉ ra rằng khoảng lề chống nhiễu của họ logic CMOS cao hơn nhiều so với các họ logic khác. Như vậy, ta nên dùng họ CMOS trong môi trường nhiễu chẳng hạn trong môi trường công nghiệp.

	TTL-LS (+5V)	CMOS (+15V)	HCT [+5V]	ECL [- 5,2V]
NM <sub>H</sub>	0.7V	5V	2.4V	0.3V
NM <sub>L</sub>	0.3V	5V	0.7V	0.3V

### 7. Thời gian truyền đạt và thời gian quá độ

Có hai khoảng thời gian đặc trưng cho từng họ logic, trong đó *thời gian truyền đạt* là tham số quan trọng hơn. Nó là khoảng thời gian giữa thời điểm thay đổi mức logic vào và thời điểm xuất hiện thay đổi mức logic ra tương ứng. Nó sẽ xác định tốc độ lớn nhất của toàn mạch. *Thời gian quá độ* xác định tốc độ chuyển mức của tín hiệu ra.

Thông thường, mỗi linh kiện số sẽ phải chỉ rõ các thời gian truyền đạt sau:

$t_{PHL}$  : thời gian trễ với đầu ra chuyển từ mức cao xuống thấp.

$t_{PLH}$  : thời gian trễ với đầu ra chuyển từ mức thấp lên cao.

Các thời gian trễ này, phải được đo giữa các mức ngưỡng cụ thể, trong hầu hết các trường hợp, trùng với 50% khoảng thay đổi tín hiệu.

Tham số này chủ yếu dùng cho việc thiết kế các hệ thống logic vì khi kết quả thay đổi, thời gian sẽ phải xác định theo một cách đặc biệt cho mỗi thay đổi để chống lại các xung không mong muốn.

#### *Thời gian quá độ (transition time)*

Thông thường, với mỗi linh kiện số cũng phải nêu rõ các thời gian quá độ sau:

$t_{THL}$  : thời gian quá độ với đầu vào chuyển từ cao xuống thấp

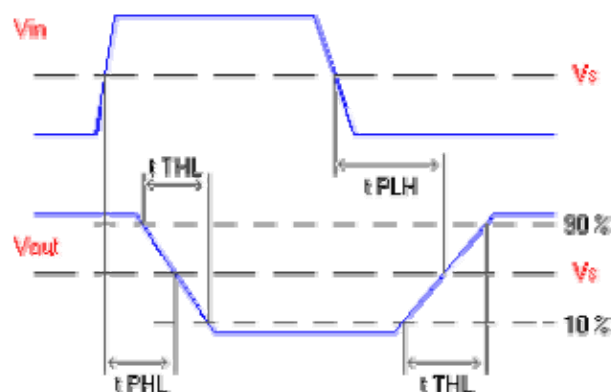
$t_{TLH}$  : thời gian quá độ với đầu vào chuyển từ thấp lên cao.

Thời gian được đo trong khoảng 10-90% thay đổi của tín hiệu.

Thời gian này cần khi thiết kế các mạch logic tuần tự, với các đầu vào kích, bởi vì tín hiệu kích không đủ nhanh linh kiện sẽ không lật trạng thái.

Hình sau biểu diễn thời gian trễ truyền đạt cũng như thời gian quá độ của một cổng đảo.

Bảng so sánh các giá trị thời gian của các họ logic.



Bảng này chỉ ra rằng họ ECL có tốc độ cao nhất, họ CMOS có tốc độ thấp nhất.

	TTL-LS 74LS00	CMOS CD4011B	HCT 74HCT00	ECL HD10102
$t_{PHL}$ [ns]	10	90	10	2.9
$t_{PLH}$ [ns]	10	90	10	2.9
$t_{THL}$ [ns]	---	80	5	3.3
$t_{TLH}$ [ns]	---	80	5	3.3

## 8. Dạng vỏ IC

Có 3 phương pháp để đóng gói cho tinh thể silic là: phương pháp T05, đóng vỏ dạng hộp và đóng vỏ hai hàng chân song song.

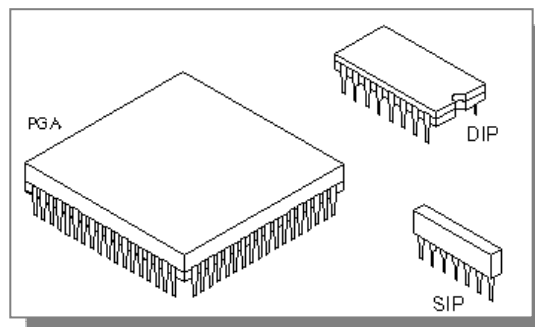
+ Đóng vỏ dạng T05, hình dạng này giống như của transistor, nghĩa là dạng mũ có nhiều chân. Kiểu đóng rắn này hiện nay ít được sử dụng nhưng do có khả năng tiêu tán nhiệt tốt nên chủ yếu được dùng cho IC tuyến tính.

+ Đóng vỏ 2 hàng chân song song / DIP, đây là cách phổ biến nhất để đóng vỏ IC. Nó lớn hơn kiểu đóng rắn nhưng có ưu điểm là dễ lắp ráp và sử dụng. Các loại IC đóng vỏ kiểu có số chân từ 8 tới hàng trăm chân.

Có nhiều kiểu vật liệu được sử dụng để đóng rắn, thông dụng và rẻ nhất là đóng gói chất dẻo. IC được đặt vào khung kim loại sau đó toàn bộ mạch được bao phủ bằng kỹ thuật đúc chất dẻo. Ngoài ra để tăng khả năng chịu nhiệt người ta còn dùng kỹ thuật đóng rắn bằng gốm.

+ Đóng vỏ dạng hộp / flat pack, đây là kiểu đóng vỏ cho các IC có mật độ tích hợp cao, thường gọi là IC dán.

IC flat pack thường được sử dụng cho các hệ thống yêu cầu độ tin cậy cao.



## 9. Giới hạn nhiệt độ

Hầu hết cá IC đều có thể hoạt động trong một dải nhiệt độ khá rộng từ  $-55$  tới  $+125^{\circ}\text{C}$ . Các mạch đặc biệt có thể làm việc ngoài dải trên tùy theo cấu tạo của chúng.

Với loại IC đóng rắn bằng chất dẻo thì giới hạn nhiệt độ nhỏ hơn (từ  $0$  tới  $+70^{\circ}\text{C}$ ) so với loại đóng rắn bằng gốm và thường được gắn thêm các cánh tản nhiệt hay thậm chí có cả quạt gió.

### III. CÔNG NGHỆ IC SỐ

#### 1. Công nghệ đơn cực (công nghệ MOS - Metal Oxide Semiconductor)

Công nghệ MOS có ưu điểm là dễ chế tạo vì công đoạn thực hiện ít quy trình hơn, mật độ tích hợp cao do transistor đơn cực có kích thước nhỏ và đặc biệt là tiêu thụ điện năng rất ít.

Dưới đây ta sẽ xem xét một số họ logic MOS thông dụng nhất

##### a. Họ logic PMOS

Các transistor MOSFET ở đây có dạng kênh P nên gọi là PMOS. Do các hạt mang điện là lỗ trống nên PMOS có tần số làm việc khá nhỏ (khoảng 1MHz) vì lỗ trống di chuyển khó hơn điện tử. PMOS có mật độ tích hợp cao, công suất tiêu thụ nhỏ và dễ chế tạo. Tuy nhiên họ này không tương hợp với TTL (họ logic rất phổ biến mà ta sẽ nói cụ thể ở phần sau) do đó mạch đòi hỏi nhiều điện áp nguồn nuôi khác nhau.

Công nghệ PMOS thường để chế tạo các bộ vi xử lý tốc độ chậm như NEC com 43/44/45 hay TMS 1000

##### b. Họ logic NMOS

MOSFET được sử dụng là MOSFET kênh N có hạt dẫn điện là điện tử nên đạt được tốc độ cao hơn PMOS hàng chục lần.

NMOS cho mật độ tích hợp rất lớn, công suất tiêu thụ cũng chỉ tương đương PMOS, khoảng 0,2mW/công

NMOS có khả năng tương thích với TTL nên chỉ cần một nguồn nuôi duy nhất.

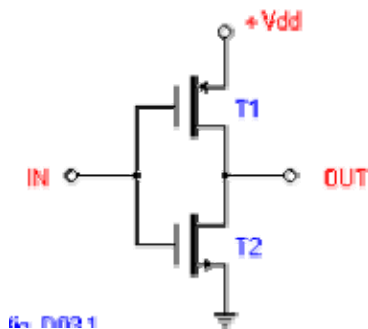
Họ NMOS có một số cải tiến thành các họ HMOS, XMOS hay VMOS có mật độ tích hợp cao hơn, công suất tiêu thụ nhỏ hơn nhưng tần số làm việc lại cao hơn.

Một số bộ vi xử lý được chế tạo theo công nghệ NMOS như 8080 / 8085 / 8086, Z80 / Z80000, MC 6800 / 68000 ...

##### c. Họ logic CMOS.

Họ CMOS sử dụng các cặp MOSFET kênh N và kênh P ở chế độ tải tích cực do đó công suất tiêu thụ nhỏ, 10  $\mu$ W/công. Ngưỡng đổi trạng thái bằng khoảng 1/2 điện áp nguồn nuôi.

ví dụ: hình dưới đây là sơ đồ của cổng NOT sử dụng công nghệ CMOS.



Mạch này gồm 2 Transistor trường khác loại, NMOS (T1) và PMOS (T2).

Đầu vào được nối tới cực cửa G và đầu ra nối tới cực máng D.

Điện áp cung cấp trong các mạch logic CMOS thường được ký hiệu  $V_{dd}$ .

##### Hoạt động

Khi đầu vào ở mức logic thấp, NMOS sẽ ngắt (vì  $V_{GS} \approx 0V$ ) và PMOS dẫn (vì  $V_{GS} \approx -V_{dd}$ ). Bởi thế, điện áp đầu ra có mức cao thực tế bằng  $V_{dd}$  (khi không tải).

Tương tự, khi đầu vào có mức logic cao, dẫn đến đầu ra có mức logic thấp bằng 0V (không tải).

Ưu điểm của việc sử dụng mạch 2 T khác loại (bù).

Việc sử dụng 2 T bù, khiến công nghệ CMOS có những ưu điểm so với các họ logic khác:

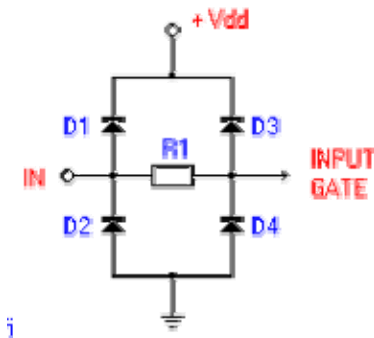
Giảm công suất tiêu thụ trong điều kiện tĩnh xuống khoảng vài  $\mu W$  (không có dòng tại mạch ra vì khi 1 T dẫn, T kia sẽ ngắt).

Khi chuyển trạng thái, sườn xung sẽ dốc hơn và có thời gian đổi xứng hơn, tức:  $t_{THL} = t_{TLH}$ .

Mức logic 0 và 1 tại đầu ra sẽ xấp xỉ 0V và  $V_{dd}$ .

Giảm dòng đầu vào trong điều kiện tĩnh, thậm chí về 0A do cực G được cách ly đối với MOS.

Tuy nhiên, ưu điểm của việc giảm công suất tiêu thụ do cực cửa G được cách ly đối với công nghệ MOS sẽ dẫn đến nhược điểm là: các đầu vào có thể lưu trữ các điện tích tĩnh điện tạo nên một lớp mỏng chất cách điện đọng lại trên kênh. Do đó, cần có mạch chống tĩnh điện tại đầu vào, nằm bên trong mạch tích hợp. Mạch này, về cơ bản là một nhóm các Diode được nối với nhau như hình dưới đây bởi thế điện áp  $V_{GS}$  không thể lớn hơn  $V_{dd}$  hay giảm xuống 0V.



Không giống các họ logic khác, công suất tiêu thụ của CMOS tăng nhanh khi tần số hoạt động tăng vì 2 lý do chính:

- + Số lần nạp và phóng trên một giây của các điện dung ký sinh (tạo bởi cực của G) tăng lên.
- + Trong khoảng thời gian chuyển mức logic, cả hai MOS đều dẫn.

Vì các lý do này, công suất tiêu thụ, mà được bỏ qua dưới điều kiện tĩnh, sẽ tăng khi tần số tăng, cho đến tần số khoảng vài MHz thì công suất tiêu thụ của họ CMOS sẽ xấp xỉ như các họ lưỡng cực.

### **Seri CMOS loại HC và HCT.**

Seri HC (CMOS tốc độ cao High Speed) được giới thiệu vào những năm bắt đầu thập kỷ 80. Loại này có tốc độ và dòng cao hơn CMOS chuẩn khoảng 10 lần, sơ đồ chân tương thích với họ TTL; khoảng lè chống nhiễu cao hơn TTL và  $V_{dd}$  từ 2 đến 6V. Khi làm việc với điện áp 5V như TTL tốc độ của các họ trên giảm đi rất nhiều

Seri mới này có công suất tiêu thụ thấp hơn họ TTL; khả năng chống nhiễu cao hơn; khả năng điều khiển đầu ra cao hơn và điện áp hoạt động từ 2 – 6V.

Vì điện áp ra của HC không tương thích với TTL nên seri HCT được phát triển, với cùng tính năng như HC nhưng có khả năng tương thích TTL với điện áp cung cấp  $V_{dd} = 5V$ .

Một số chỉ tiêu kỹ thuật của CMOS:

Thời gian trễ	30 – 100ns
---------------	------------

Công suất tiêu tán	0,01mW (1mW ở tần số 1MHz)
Khả năng tải	50
Độ ổn định nhiễu	~ 45% Vdd
Mức logic	Mức 0 bằng 0V; mức 1 bằng Vdd
Nguồn cung cấp	3 – 15V
Các cổng logic cơ bản	NOR; NAND

## 2. Công nghệ lưỡng cực

Thành phần cơ bản của các vi mạch công nghệ lưỡng cực là sử dụng các transistor lưỡng cực. Công nghệ này có một số họ cơ bản sau:

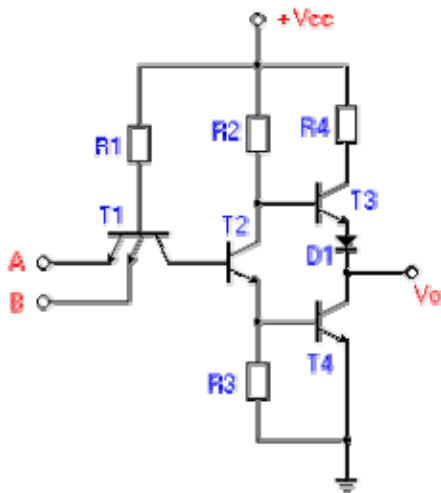
### a. Họ logic TTL (Transistor – Transistor – Logic)

Đây là họ vi mạch được sử dụng rộng rãi trong mọi lĩnh vực và trở thành tiêu chuẩn trong họ TTL cho các họ logic khác.

#### Đặc tính điện của cổng logic TTL.

	TL-LS	CMOS	HCT	ECL
Vcc [V]	+5	+15	+5	-5.2
[Vih] min [V]	2	10	2	-1.2
[Vil] max [V]	0.8	5	0.8	-1.4
[Voh] min [V]	2.4	14.95	4.4	-0.9
[Vol] max [V]	0.4	0.05	0.1	-1.7

Xét cổng logic cơ bản của họ TTL là cổng NAND được cho như hình dưới



Transistor T1 là loại nhiều emitter. Transistor T2 làm nhiệm vụ cung cấp 2 tín hiệu ngược pha; tín hiệu này điều khiển tầng ra gồm T3, D1 và T4.

Transistor T3 được gọi là “transistor nối nguồn (pull-up)” và hoạt động như một mạch lặp E khi đầu ra ở mức cao nó sẽ khiến cho trở kháng ra rất thấp.

Nếu cả hai đầu vào ở mức cao, mạch sẽ tiêu thụ dòng của mỗi đầu vào khoảng  $40\mu\text{A}$ .

Collector của T1 được nối với base của T2 và có mức điện áp  $2V_{BE}$ , tức là khoảng 1,4V. Diode tương đương của tiếp giáp base-collector của T1 lấy nguồn qua điện trở R1, do vậy được phân cực thuận; nhờ thế Transistor T2 rơi vào trạng thái bão hòa. Dòng Emitter

của T2 một phần chảy qua R3, một phần chảy vào base của T4 do đó, đưa T4 vào trạng thái bão hòa. Điện thế base của T3,  $V_{b3}$  có giá trị bằng với tổng  $V_{be}$  của T4 cộng với  $V_{ceSat}$  của T2.

Điện áp qua Emitter của T3 là:

$$V_{e3} = V_{ceSat} + V_{d1}$$

Do đó,  $V_{b3} = V_{e3}$  và transistor T3 ở trạng thái ngắt (OFF). Lúc này, Transistor T4 sẽ thông (ON), có dòng điện khoảng 16mA chảy qua và đầu ra có mức logic 0 tức điện áp đạt khoảng 400mV.

Giá trị logic 0 điển hình tại đầu ra là 220mV, với dòng điện đạt 16mA

Giá trị dòng này đủ để điều khiển 10 đầu vào logic TTL ở trạng thái 0.

Trở kháng ra Rout do T4 đạt khoảng 120hm.

Trạng thái OFF (một đầu vào tại mức thấp, đầu ra ở mức cao).

Xét trường hợp tối thiểu một đầu vào ở mức thấp (đầu vào không vượt quá 400mA).

Giá trị dòng lớn nhất khi đầu vào có mức logic 0 là khoảng 1,6mA, bởi thế một cổng với đầu ra mức thấp có thể điều khiển khoảng 10 cổng khác.

Lúc này, Transistor T1 sẽ dẫn, T2 và T4 rơi vào trạng thái ngắt. Điện áp trên collector của T2 là cao do vậy, T3 đạt bão hoà.

Dưới những điều kiện này, dòng đưa qua đầu ra đạt 400 $\mu$ A, đủ để điều khiển 10 cổng khác.

*Điện áp đầu ra Vo, khi đủ nguồn cung cấp, không nhỏ hơn 2,4V.*

Thực tế, với giá trị áp vào thấp hơn 800mV, điện áp ra điển hình đạt 3,3V.

Dòng điện ra trong điều kiện ngắn mạch có giá trị nhỏ nhất là 18mA và giá trị max là 58 mA, được giới hạn chủ yếu bởi R4.

Trở kháng ra ở mức cao là khoảng vài trăm Ohm.

### ***Quá trình chuyển trạng thái của cổng TTL.***

Để chuyển trạng thái từ 1 (OFF) về 0 (ON) một đầu vào sẽ có mức thế đất còn đầu vào kia nối với Vcc. Khi điện áp ở đầu vào ở mức thấp tăng, dòng điện đầu vào sẽ giảm và khi tăng đạt tới 0,8V T2 bắt đầu dẫn và điện áp trên collector của nó giảm. Kết quả, điện áp đầu ra giảm cho đến khi điện áp đầu vào đạt khoảng 1,4-1,5V, lúc này điện áp đầu ra có giá trị khoảng 2V. Điện áp trên base của T2 là khoảng 1,4V và do vậy, cả T2 và T4 đều dẫn.

Bắt đầu từ thời điểm này, điện áp đầu ra nhanh chóng giảm xuống giá trị  $V_{ccSat}$  của T4, tức là T2 đạt bão hoà còn T3 chuyển sang trạng thái ngắt (OFF).

Có một khoảng thời gian rất ngắn khi mà cả T3 và T4 đều dẫn; trong khoảng thời gian này có dòng chảy qua R4, T3, D1 và T4.

Dòng này được hạn chế chủ yếu bởi R4.

Khi chuyển từ trạng thái thấp (low) lên cao (High), ban đầu các đầu vào ở trạng thái cao.

Khi điện áp của một (hay nhiều) đầu vào giảm xuống 1,4V, T1 bắt đầu dẫn khiến cho T2 và T4 cũng rơi vào trạng thái dẫn. Dòng qua T2 giảm điện áp trên collector của T2 tăng khiến T3 rơi vào trạng thái dẫn vì thế đầu ra sẽ ở mức thấp.

Dù cho cổng TTL ở mức ON hay OFF trở kháng ra luôn thấp, cho phép cổng TTL có thể điều khiển tải dung kháng cao.

### ***Các nhánh phụ của họ TTL.***

Họ logic chuẩn TTL (STD) đã được thay đổi qua nhiều năm để có các tính năng tốt hơn, tạo nên các nhánh phụ (sub-families) của họ TTL.



Thực tế, các nhánh phụ của họ TTL chuẩn hoạt động nhanh hơn hay tiêu thụ công suất ít hơn so với họ TTL chuẩn.

Chúng gồm:

S TTL (Schottky TTL) : tốc độ tăng gấp 3 lần nhưng công suất tiêu thụ tăng lên tới 20mW/cổng.

AS TTL (Advanced Schottky): tốc độ gần bằng ECL (1 đến 2ns)

LS TTL (Low Power Schottky TTL) : cùng tốc độ nhưng công suất tiêu thụ giảm 5 lần. 10ns, 2mW/cổng

F TTL (Fast TTL) : tốc độ gấp 4 lần, công suất tiêu thụ giảm một nửa.

ALS TTL: 3ns, 1.25mW/cổng

Một số ký hiệu của TTL cho biết dải nhiệt độ công tác

74: 0°C - +70°C

84: -25°C - +85°C

54: -55°C - +125°C

Một số chỉ tiêu kỹ thuật của TTL chuẩn:

Thời gian trễ	10ns
Công suất tiêu tán	10mW
Khả năng tải	10
Độ ổn định nhiễu	Cao
Mức logic	mức 0 bằng +0,4V; mức 1 bằng +3,6V
Nguồn cung cấp	5V ± 10%
Các cổng logic cơ bản	NOR; NAND

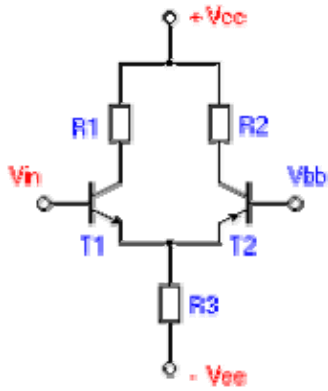
### **b. Họ logic ECL.**

Họ logic ECL (Emitter Coupled logic) được tạo ra sử dụng công nghệ lưỡng cực (giống như họ TTL).

Đây là họ logic có tốc độ hoạt động nhanh nhất trên thị trường. Nó đạt được tốc độ đó vì 2 lý do:

- + Tránh việc đưa các linh kiện tích cực vào trạng thái bão hòa.
- + Cho phép tiêu thụ công suất cao hơn trên mỗi cổng so với các họ logic khác.

Một thành phần chính trong họ ECL là bộ khuếch đại vi sai, trong đó 2 Transistor được ghép Emitter chung như trong hình dưới đây



Các đặc điểm của bộ khuếch đại vi sai:

Dòng emitter không đổi.

Dòng sẽ chảy từ Transistor này sang T kia, khi điện áp Vin đưa tới đầu vào của T thứ nhất nằm trong khoảng:

$$V_{BB} - 0,1V < V_{in} < V_{BB} + 0,1V.$$

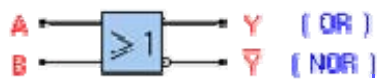
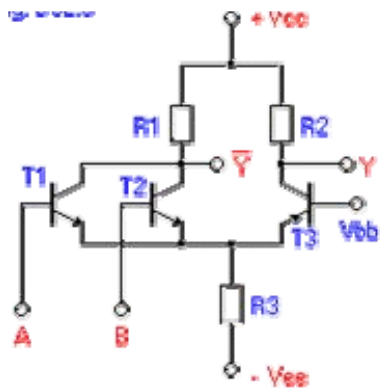
với  $V_{BB}$  : điện áp chuẩn đưa vào base của T thứ 2.

Điện áp đầu ra của mạch sẽ nhận một trong hai giá trị có thể và như vậy, nó hoạt động giống như mạch nhị phân. Bởi thế, mạch khuếch đại vi sai thông thường được xem như một mạch tương tự nhưng cũng là mạch số quan trọng.

Vì các mạch số so sánh được tạo nên từ mạch vi sai không có Transistor nào rơi vào trạng thái bão hoà, do vậy, họ logic ECL có tốc độ hoạt động rất nhanh và thời gian trễ thấp hơn 1ns.

Tuy vậy, tốc độ cao phải trả giá bằng việc tăng công suất tiêu thụ trên mỗi cổng so với họ TTL (khoảng 25mW/cổng)

Dưới đây là cổng OR và NOR với 2 đầu vào.

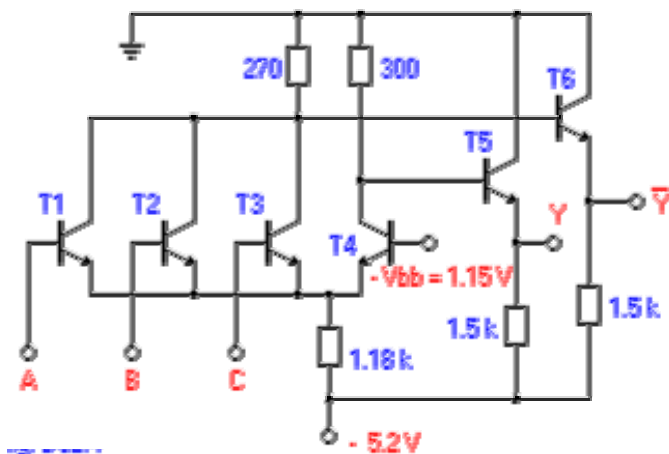


Mạch này tương tự như mạch hình trên chỉ khác là đầu vào sử dụng 2 Transistor mắc song song.

Nếu A và B ở mức thấp, T1 và T2 sẽ không dẫn trong khi T3 tích cực (dẫn). Lúc này, Y có mức thấp và Y có mức cao.

Nếu một trong hai đầu vào ở mức cao, dòng Emitter sẽ chảy qua R3 và dòng collector của T3 giảm gần về 0. Bởi thế điện áp tại điểm Y tăng và điện áp Y giảm. Như vậy, mạch logic thực hiện hàm OR tại đầu ra Y và hàm NOR tại đầu ra còn lại.

Một trong những nhược điểm của cấu trúc mạch ECL ở hình trên là các mức điện áp đầu ra có khác biệt so với đầu vào. Để khắc phục, mắc thêm 2 Transistor T4 và T5 theo kiểu CC, như trong hình dưới đây để đưa mức điện áp trở về đúng các giá trị yêu cầu.



Đây là cấu trúc cơ bản của cổng ECL với 3 đầu vào. Điện áp chuẩn  $-V_{BB}$  được tạo ra từ mạch bù nhiệt (không được chỉ ra ở hình vẽ). Mạch này tạo ra các mức điện áp :

$$V(0) = -1,7V.$$

$$V(1) = -0,9V.$$

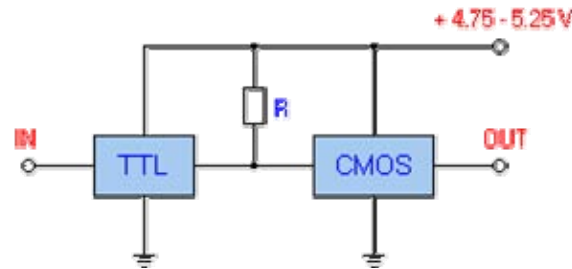
Khoảng lè chống nhiễu của họ logic này rất hẹp và điều này giải thích tại sao các cổng được cấp nguồn giữa đất và  $-V_{EE}$  (-5.2V) để làm giảm trở kháng trong.

### 3. Giao tiếp TTL-CMOS và CMOS-TTL.

Như giới thiệu ở phần trên ta thấy họ TTL và họ CMOS là 2 họ logic lớn nhất và được sử dụng nhiều nhất. TTL có ưu điểm về tốc độ còn CMOS lại có ưu điểm về công nghệ chế tạo đơn giản và tiêu thụ điện năng ít. Vì vậy, việc ghép nối giữa 2 họ logic là rất quan trọng dù rằng các nhà sản xuất khuyến nghị nên dùng cùng một họ logic trong một mạch điện tử.

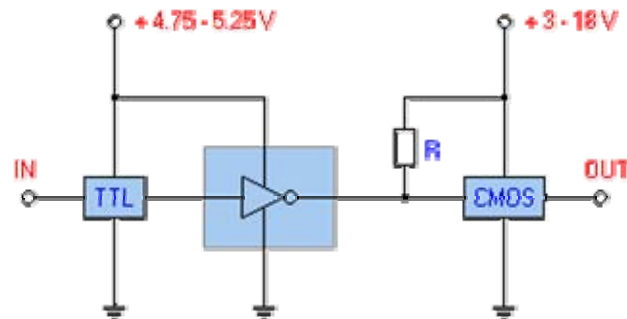
#### a. Giao tiếp TTL-CMOS.

Trường hợp đơn giản nhất của giao tiếp giữa linh kiện TTL và CMOS là khi ta chỉ có 1 nguồn cung cấp duy nhất là 5V.



Như ta thấy trong hình bên, một điện trở nối nguồn (pull-up-có giá trị khoảng vài KOhm) được sử dụng để kéo đầu ra có mức logic cao của cổng TTL (mà có giá trị nhỏ nhất là 2,4V, chưa được xem như là mức cao đối với CMOS) lên xấp xỉ 5V.

Khi các linh kiện có các nguồn cung cấp khác nhau giao tiếp với nhau ta phải sử dụng một linh kiện tương thích TTL với đầu ra hở collector hay hở cực máng D như hình sau:

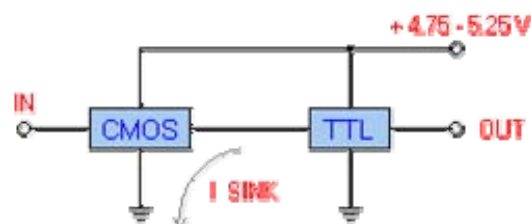


Bộ đệm điện hình là 7407 hay 7417 được cấp nguồn +5V, với điện trở nối nguồn khoảng vài KOhm giữa đầu ra và  $V_{DD}$  (có thể điều khiển nhằm thu được điện áp từ +3V đến +18V).

#### b. Giao tiếp CMOS-TTL.

Việc ghép nối trực tiếp linh kiện CMOS-TTL sử dụng cùng nguồn cung cấp +5V yêu cầu việc xem xét dòng rò của linh kiện CMOS để giữ mức điện áp đầu ra thấp của CMOS nằm trong phạm vi cho phép của linh kiện TTL.

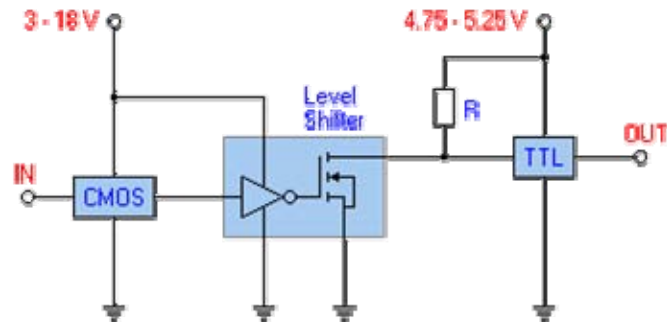
Trong Seri CMOS CD4000 B tất cả các linh kiện đều có thể điều khiển tối thiểu một



linh kiện TTL nhưng chỉ thuộc họ LS.

Để điều khiển một hay nhiều linh kiện của họ TTL STD, TTL S hay TTL F..., yêu cầu có một bộ đệm (chẳng hạn như CD4049 hay CD 4050).

Khi các linh kiện có nguồn cung cấp riêng khác nhau ghép nối với nhau, một bộ đệm CMOS với đầu ra hở cực máng D được sử dụng như hình dưới.



Tình huống này còn gặp phải khi cần truyền dữ liệu từ phần thu thập dữ liệu sử dụng linh kiện CMOS (chẳng hạn trong khu vực công nghiệp yêu cầu khoảng lè chống nhiễu cao-chỉ có với các linh kiện họ CMOS) tới hệ thống xử lý dùng các linh kiện họ TTL. Linh kiện được sử dụng nên là bộ đệm CMOS với đầu ra hở cực máng D- MM 74C906. Điện trở R nên nằm trong khoảng vài KOhm.

## **PHẦN II**

### **MẠCH TỔ HỢP**

## CHƯƠNG 5: PHÂN TÍCH VÀ THIẾT KẾ MẠCH TỔ HỢP

Mạch số được chia làm 2 loại là :

+ Mạch tổ hợp / Combinational Circuit

+ Mạch dãy / Sequential Circuit

**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  $i$ , có  $m$  đầu ra

$X_j$  là tín hiệu vào ở đầu vào thứ  $j$ , có  $n$  đầu vào

Người ta còn gọi mạch tổ hợp là mạch không có nhớ

**Mạch dãy** 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 dãy là:  $Y_i = f_i(X_1, X_2, \dots, X_n, S_1, S_2, \dots, S_k)$  với  $\forall i = 1 \div m$

$Y_i$  là tín hiệu ra ở đầu ra thứ  $i$ , có  $m$  đầu ra

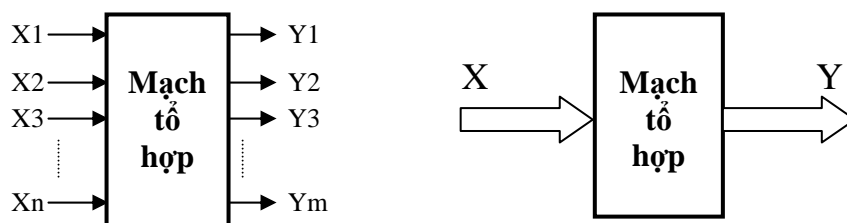
$X_j$  là tín hiệu vào ở đầu vào thứ  $j$ , có  $n$  đầu vào

$S_t$  là trạng thái trong của mạch

Mạch dãy có khả năng lưu trữ dữ liệu nên còn được gọi là mạch có nhớ.

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

### I. MÔ HÌNH TOÁN HỌC CỦA MẠCH TỔ HỢP



$X = \{ X_1, X_2, \dots, X_n \}$ : tập hợp các tín hiệu đầu vào

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

Khi đó mạch tổ hợp có thể được mô tả bởi hệ  $m$  phương trình đại số Boolean như sau:

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

Về mặt toán học có thể nói mô hình toán học của mạch tổ hợp chính là otomat không có nhớ, mô tả bằng phương trình:

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

Với  $X, Y$  là bộ chữ vào, ra và  $f$  là ánh xạ từ  $X$  vào  $Y$

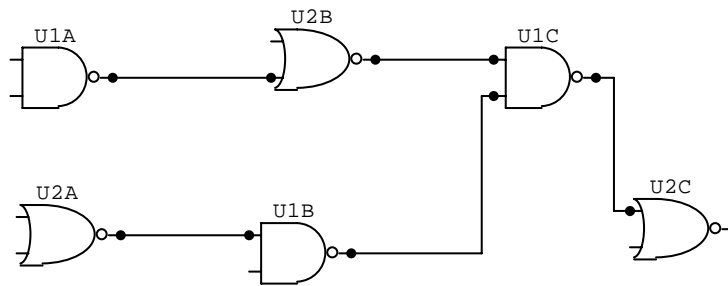
## II. PHÂN TÍCH MẠCH TỔ HỢP

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.

**Các bước phân tích mạch tổ hợp như 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)
- + Trong biểu thức cuối cùng, thay thế các biến phụ bằng các giá trị tương ứng để rút ra được hàm logic cho các đầu ra cho sơ đồ.

ví dụ: phân tích mạch tổ hợp cho ở hình dưới đây:



## III. THIẾT KẾ MẠCH TỔ HỢP

### 1. Bài toán thiết kế và các bước thực hiện

Đây là bài toán ngược với bài toán phân tích, đó là từ yêu cầu cho trước như chức năng, dạng sóng ... ta phải xây dựng sơ đồ mạch thực hiện những yêu cầu đó.

Trong phạm vi của chương này ta chỉ xét đến việc sử dụng các vi mạch cỡ nhỏ (SSI), thực hiện theo các bước sau:

- + Mô tả bài toán dưới dạng chức năng
- + Tối thiểu hoá
- + Chỉ ra sơ đồ logic dùng cho các cổng đã cho

### 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

Các cách thiết kế mạch hai tầng với các phần tử cho trước

Tầng1 / tầng 2	AND	OR	NAND	NOR
AND	X	CTT	X	1. CTH

				2. $\overline{\overline{f}}, D$
<b>OR</b>	CTH	X	1. CTT 2. $\overline{\overline{f}}, D$	X
<b>NAND</b>	1. CTH 2. $\overline{\overline{tp}}, D$	X	1. CTT 2. $\overline{\overline{f}}, D$	X
<b>NOR</b>	X	1. CTT 2. $\overline{\overline{tp}}, D$	X	1. CTH 2. $\overline{\overline{f}}, D$

Ghi chú:

$\overline{\overline{f}}$  : phủ định hai lần hàm f

$\overline{\overline{tp}}$  : phủ định hai lần từng thành phần

D: áp dụng luật Demoorgan

Các giá trị tín hiệu vào Xi và  $\overline{Xi}$  có sẵn

Trên cùng một tầng chỉ sử dụng một loại phân tử (AND, OR, NAND, và NOR)

Những phân tử này có số đầu vào không hạn chế

ví dụ: Cho hàm logic  $f = \sum 0,1,5,6,7$

Trước khi xây dựng sơ đồ ta cần thực hiện tối thiểu hoá hàm trên theo dạng CTT và CTH

Biểu diễn hàm f trên bảng Karnaugh

C / AB	00	01	11	10
0	1	1		
1		1	1	1

Từ bảng Karnaugh dễ dàng viết được:

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

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

Dựa vào bảng kết hợp đầu vào và đầu ra ta có thể xác định được sơ đồ mạch cho f như sau:

1. Tầng 1 dùng mạch AND, tầng 2 dùng mạch OR

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

2. Tầng 1 dùng mạch OR, tầng 2 dùng mạch AND

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

3. Tầng 1 dùng mạch OR, tầng dùng mạch NAND

+ Viết f dưới dạng CTT  $f = \overline{A}\overline{C} + B.C + A.C$



+ Phủ định hai lần hàm f, sau đó áp dụng 2 lần luật D

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

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

4. Tầng 1 dùng mạch NAND, tầng 2 dùng mạch AND

+ Viết f dưới dạng CTH  $f = (A + B + \overline{C})(\overline{A} + C)$

+ Phủ định 2 lần các thành phần và áp dụng De Morgan

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

$$f = \overline{\overline{A}.B.C.A.C}$$

5. Tầng 1 dùng mạch NAND, tầng 2 dùng mạch NAND

+ Viết hàm dưới dạng CTT  $f = \overline{A.C} + B.C + A.C$

+ Phủ định hai lần hàm f và áp dụng De Morgan

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

$$f = \overline{\overline{A.C}.B.C.A.C}$$

6. Tầng 1 dùng mạch NOR, tầng 2 dùng mạch OR

+ Viết hàm dưới dạng CTT  $f = \overline{A.C} + B.C + A.C$

+ Phủ định 2 lần các thành phần sau đó áp dụng D

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

$$f = \overline{A + C + \overline{B + C} + \overline{A + C}}$$

7. Tầng 1 dùng NOR, tầng 2 dùng mạch NOR

+ Viết hàm dưới dạng CTH  $f = (A + B + \overline{C})(\overline{A} + C)$

+ Phủ định 2 lần f và áp dụng D

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

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

8. Tầng 1 dùng mạch AND và tầng 2 dùng mạch NOR

+ Viết hàm f dưới dạng CTH  $f = (A + B + \overline{C})(\overline{A} + C)$

+ Phủ định 2 lần hàm số f và áp dụng D

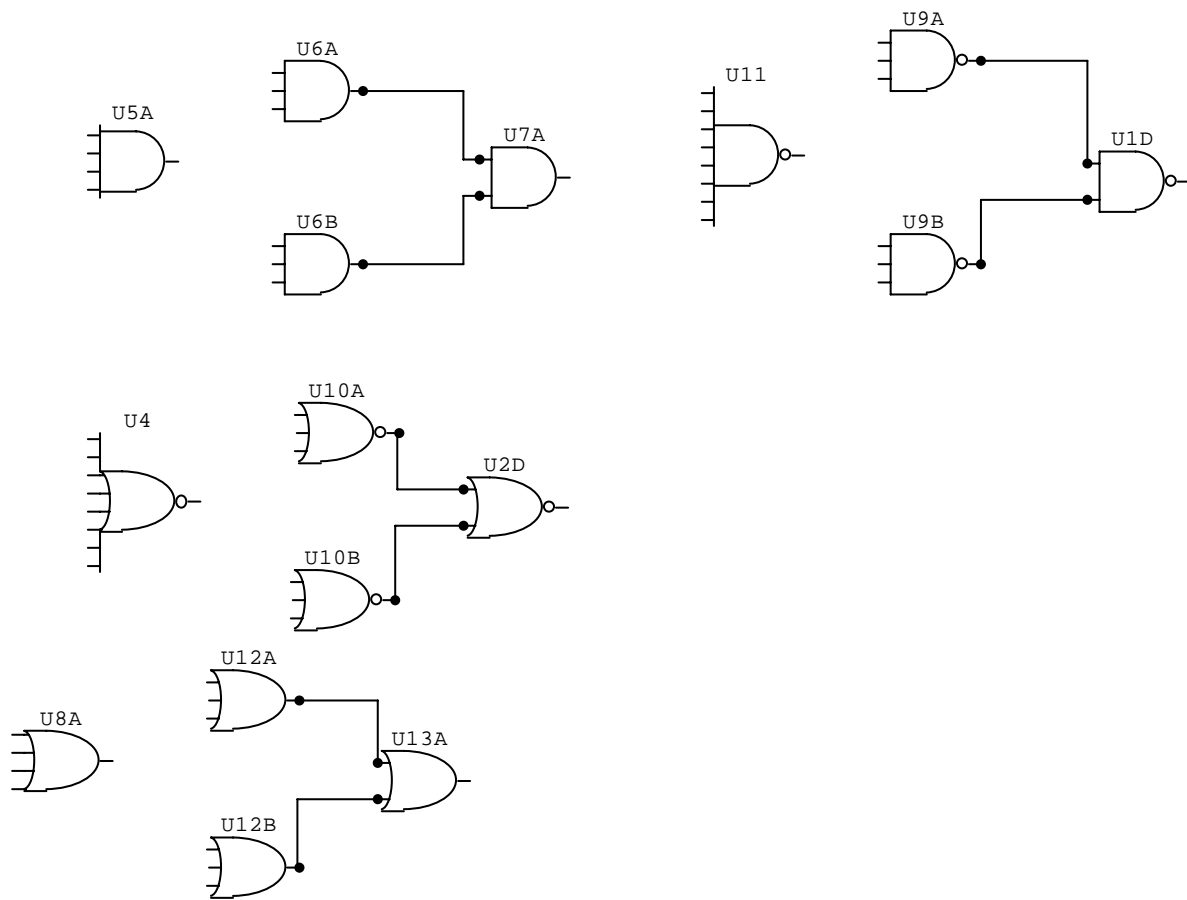
$$f = \overline{\overline{(A + B + \overline{C})} \overline{(\overline{A} + C)}}$$

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

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

### **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. Sử dụng các sơ đồ thay thế như sau:



### **3. Thiết kế một hệ hàm tổ hợp**

Có hai cách thiết kế một hàm tổ hợp là thiết kế riêng từng hàm hoặc thiết kế có phần chung để hạn chế số đầu vào.

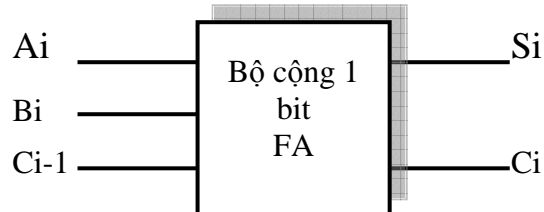
## CHƯƠNG 6:

### MỘT SỐ MẠCH TỔ HỢP THƯỜNG GẶP

#### I. BỘ CỘNG NHỊ PHÂN MỘT CỘT SỐ

##### 1. Phân tích bài toán

Mô hình toán học của bộ cộng đầy đủ 1 bit (FA – Full adder)



trong đó  $A_i$  và  $B_i$  là các số nhị phân thứ  $i$  của  $A$ ,  $B$  đưa vào cộng

$C_{i-1}$  là số nhớ của cột có trọng số nhỏ hơn bên cạnh ( của phép tính trước )

$S_i$  là là chữ số của tổng ở cột thứ  $i$   $S_i = A_i \oplus B_i \oplus C_{i-1}$

$C_i$  là số nhớ đưa đến cột có trọng số lớn hơn bên cạnh  $C_i = A_i.B_i + C_{i-1}(A_i + B_i)$

*Chú ý:* Phép cộng 2 số nhị phân luôn bắt đầu từ cột số có trọng số nhỏ nhất

Bảng chân lý của phép cộng đầy đủ một bit

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

## 2. Xây dựng sơ đồ

Có thể xây dựng bộ cộng theo 1 trong 2 cách như sau:

+ Xây dựng trực tiếp từ hệ phương trình của Si và Ci

+ Xây dựng từ các bộ bán tổng (HA – Half Adder). Đây là phương pháp được sử dụng nhiều trong thực tế và dưới đây ta sẽ xem xét tới phương pháp này.

Bộ bán tổng là bộ có bảng chân lý sau:

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

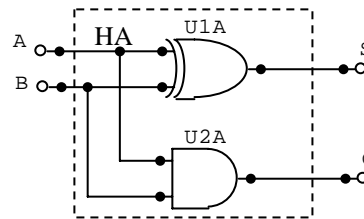
Từ bảng chân lý rút ra được:  $S = A \oplus B$   
 $C = A.B$

Như vậy sơ đồ của bộ bán tổng như sau:

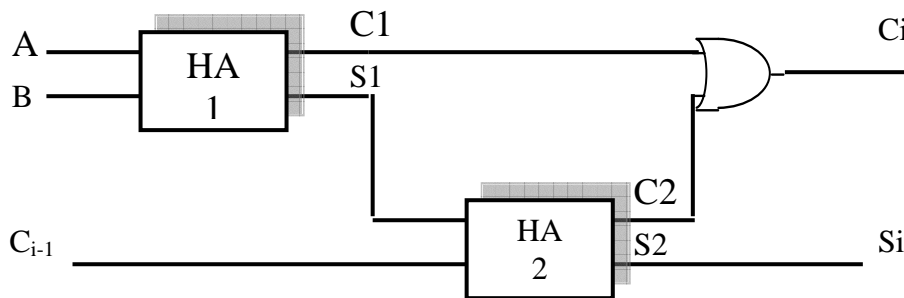
Từ phương trình của bộ tổng đầy đủ

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

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



ta xây dựng được sơ đồ của bộ tổng FA bằng 2 bộ HA và 1 cổng OR như sau:



chứng minh:

$$S_i = S_2 = C_{i-1} \oplus S_1 = C_{i-1} \oplus A \oplus B$$

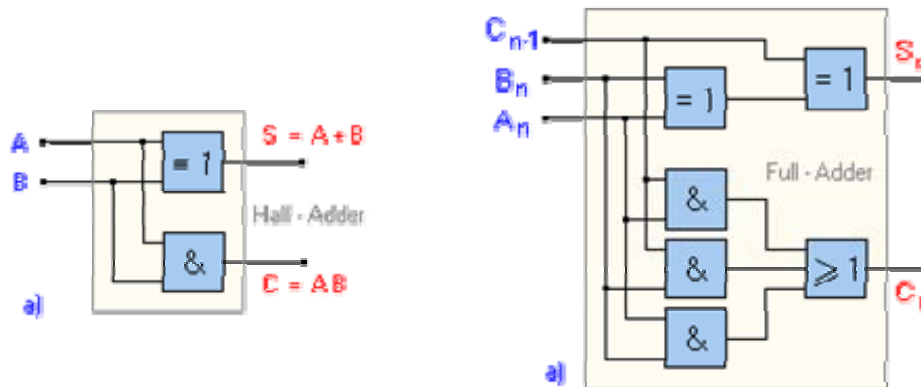
$$C_i = C_1 + C_2 = AB + C_{i-1}.S_1 = AB + C_{i-1}(A \oplus B)$$

$$C_i = AB + C_{i-1}(\overline{A}B + A\overline{B}) = A(B + \overline{B}.C_{i-1}) + B(A + \overline{A}.C_{i-1})$$

$$C_i = A(B + C_{i-1}) + B(A + C_{i-1}) = AB + C_{i-1}(A + B)$$

**Nguyên lý hoạt động của bộ cộng nhị phân.**

Rất nhiều mạch logic cần các thiết bị có khả năng cộng 2 số nhị phân. Một bộ cộng có thể tính toán một phép cộng nhị phân. Vì đầu ra phụ thuộc đầu vào tại một thời điểm xác định theo yêu cầu, nên sẽ sử dụng mạch logic tổ hợp.



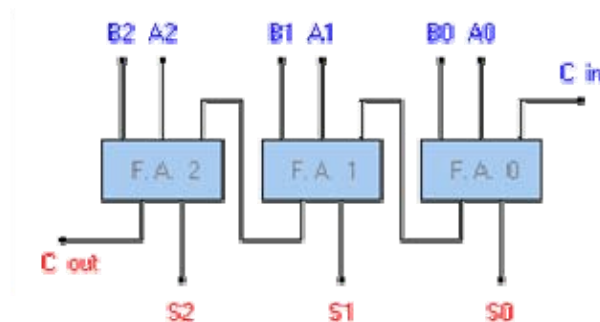
Hình trên là sơ đồ của 1 bộ bán tổng 1-bit và mạch toàn tổng. Sơ dĩ được gọi là bộ bán tổng vì nó không cộng “bit nhớ” tại đầu vào, một việc thường yêu cầu khi cộng những số có nhiều số hạng.

Để cộng các số với nhiều số hạng, mạch phải có khả năng xử lý thêm 1 đầu vào nữa. Đầu vào này là kết quả của phép cộng từ tầng trước. Mạch như vậy, được gọi là mạch toàn tổng (Full Adder).

### **Ghép nối tiếp các bộ cộng.**

Bộ toàn tổng sẽ là phần tử cơ sở cho việc xây dựng bộ cộng n-bit. Hình bên chỉ ra làm thế nào mà các đầu vào và ra của một phần tử cộng đơn lẻ có thể được nối với nhau để tạo thành bộ cộng 3-bit.

Hoàn toàn tương tự với các bộ cộng nhiều bit khác.



## II. BỘ TRỪ NHỊ PHÂN MỘT CỘT SỐ

Cách làm hoàn toàn tương tự như khi xây dựng bộ cộng nhị phân 1 cột số. Nghĩa là xây dựng bảng chân lý, tối thiểu hoá, sau đó xây dựng trực tiếp hoạt dùng bộ bán trừ để tạo ra bộ trừ đầy đủ

Bảng chân lý của bộ trừ đầy đủ (FS – Full Subtractor) như sau:

A <sub>i</sub> Chữ số thứ i của số bị trừ	B <sub>i</sub> Chữ số thứ i của số trừ	C <sub>i</sub> Số nhớ từ cột có trọng số nhỏ hơn	H <sub>i</sub> Chữ số thứ i của phép trừ	C <sub>i</sub> 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

Từ bảng chân lý ta tối thiểu hoá hàm H<sub>i</sub> và C<sub>i</sub> bằng bảng Karnaugh ta nhận được:

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

$$C_i = \bar{A}.B + C_{i-1}(\bar{A} + B)$$

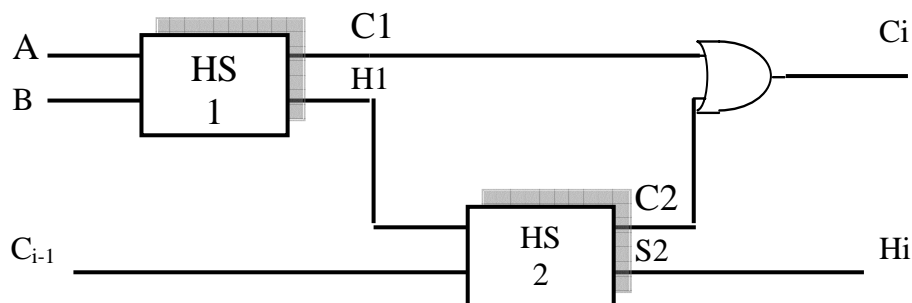
Bảng chân lý của bộ bán trừ (hệ số – Half Subtractor)

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

$$H = A \oplus B$$

$$C = \bar{A}.B$$

Khi đó ta có sơ đồ bộ bán trừ và bộ trừ đầy đủ như sau:



Chứng minh:

$$H_i = C_{i-1} \oplus H1 = C_{i-1} \oplus A \oplus B$$

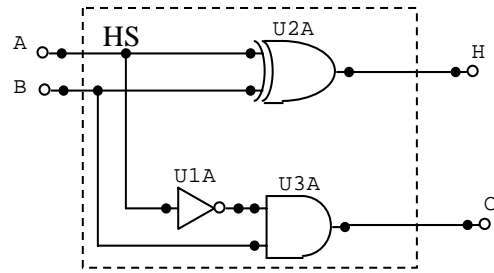
$$C_i = C1 + C2 = \overline{A}B + \overline{H1}.C_{i-1}$$

$$C_i = \overline{A}B + \overline{A \oplus B}.C_{i-1} = \overline{A}B + (\overline{A}B + AB).C_{i-1}$$

$$C_i = \overline{A}(B + \overline{B}.C_{i-1}) + B(\overline{A} + AC_{i-1})$$

$$C_i = \overline{A}(B + C_{i-1}) + B(\overline{A} + C_{i-1})$$

$$C_i = \overline{A}B + C_{i-1}(\overline{A} + B)$$



Chú ý: trên thực tế ít khi người ta sử dụng bộ trừ mà thường biến thành phép cộng hay dùng mã bù để sử dụng các bộ cộng

### III. BỘ SO SÁNH – COMPARATOR

Một bộ so sánh sẽ thực hiện phép so sánh 2 số nhị phân và kết quả sẽ được thể hiện tại đầu ra.

Bộ so sánh nhị phân thông thường có ba đầu ra:  $A=B$ ,  $A>B$ ,  $A<B$ . Vì các đầu ra chỉ phụ thuộc vào trạng thái tức thời của đầu vào nên mạch không có các phần tử nhớ.

#### 1. Bộ so sánh 2 số nhị phân 1 bit

Bảng chân lý cho các hàm ra của bộ so sánh như sau:

A	B	$A = B$	$A > B$	$A < B$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Từ bảng chân lý trên ta thấy:

Hàm  $(A = B) = \overline{A \oplus B} = \overline{A}B + A\overline{B}$

Hàm NXOR

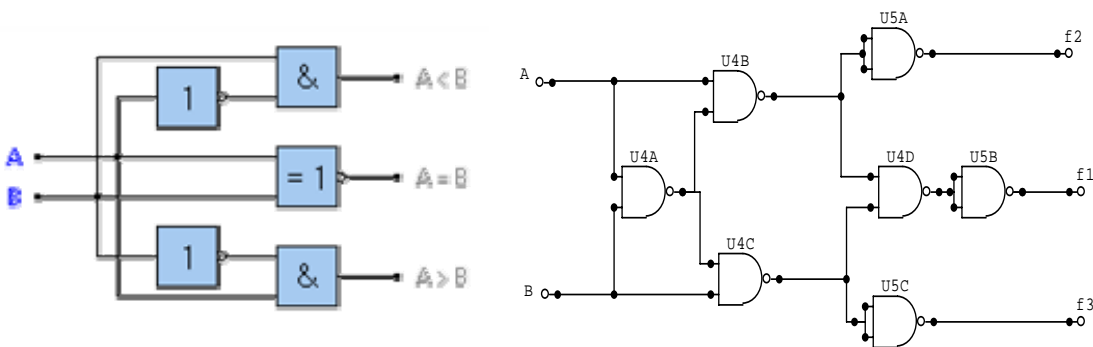
Hàm  $(A > B) = A.\overline{B}$

Hàm cấm B (inhibition)

Hàm  $(A < B) = \overline{A}B$

Hàm cấm A

Dưới đây là sơ đồ của bộ so sánh 1 bit theo nguyên lý và sơ đồ cụ thể dùng các cổng NAND.



với f1 là hàm  $(A = B)$

f2 là hàm  $(A > B)$

f3 là hàm  $(A < B)$

## 2. Bộ so sánh n bit

Giả sử có 2 số nhị phân n bit A và B được biểu diễn như sau:

$$\begin{matrix} A_n A_{n-1} \dots A_1 \\ B_n B_{n-1} \dots B_1 \end{matrix}$$

trong đó  $A_n, B_n$  là cột số có trọng số lớn nhất và  $A_1, B_1$  là cột số có trọng số nhỏ nhất

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

+ Xây dựng trực tiếp các hàm  $f_1, f_2$  và  $f_3$  (thực chất là xây dựng 1 hệ 3 hàm logic, mỗi hàm 2n biến)

+ Xây dựng sơ đồ gián tiếp từ các bộ so sánh 1 bit đã có sẵn

Xét ví dụ với n bằng 3

$$A = A_3 A_2 A_1$$

$$B = B_3 B_2 B_1$$

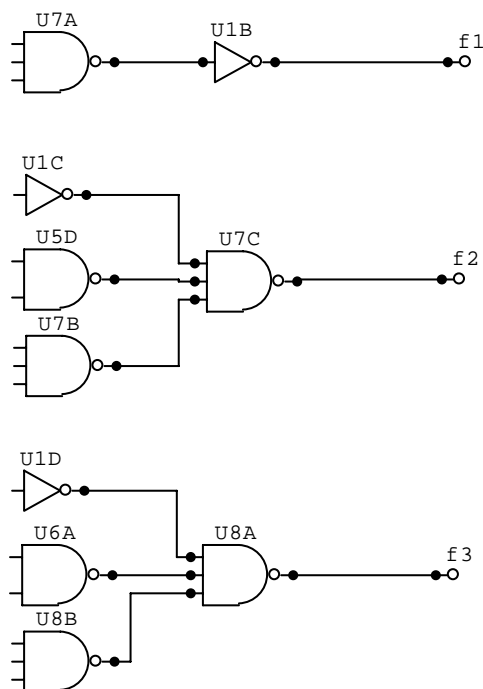
Khi đó ta thấy:

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

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

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

Sử dụng các bộ so sánh 1 bit ( $A_1, B_1$ ), ( $A_2, B_2$ ) và ( $A_3, B_3$ ), ta sẽ có sơ đồ mạch thực hiện so sánh 3 bit như sau:

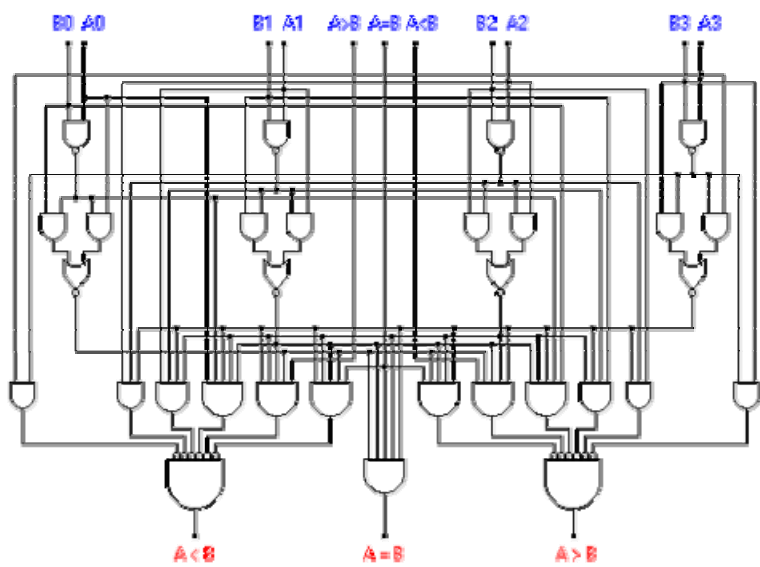
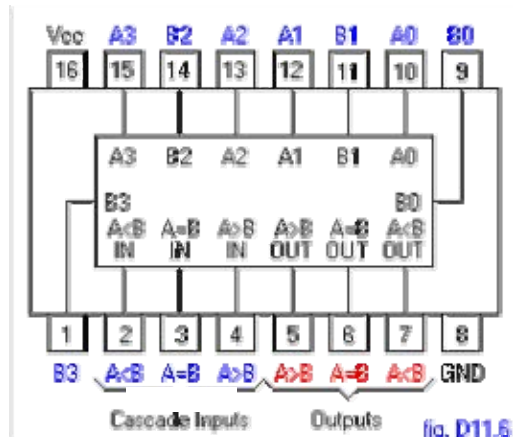


### Vì mạch so sánh nhị phân 4-bit.

Vì mạch SN74LS85 có một bộ so sánh các số nhị phân 4-bit.



Dưới đây là sơ đồ chân và sơ đồ cụ thể bên trong của vi mạch thực hiện



#### IV. BỘ TẠO VÀ KIỂM TRA CHẤM LẼ - PARITY GENERATOR AND CHECKER

Phương pháp kiểm tra chẵn lẻ là một phương pháp đơn giản nhất để xác định lỗi trong việc truyền dữ liệu. Phương pháp này được thực hiện bằng cách thêm 1 bit dữ liệu được truyền đi sao cho số chữ số 1 trong dữ liệu luôn là một số chẵn hoặc số lẻ. Bit thêm vào đó gọi là bit chẵn lẻ.

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

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

Để thực hiện được việc truyền dữ liệu theo kiểu đưa thêm bit chẵn, lẻ vào dữ liệu cần xây dựng:

+ Sơ đồ tạo được bit chẵn, lẻ thêm vào n bit dữ liệu

+ Sơ đồ kiểm tra được hệ là hệ chẵn hay lẻ với (n+1) bit ở đầu vào (n bit dữ liệu và 1 bit chẵn / lẻ)

##### 1. Mạch tạo bit chẵn lẻ

Xét ví dụ trong trường hợp dữ liệu 3 bit, trường hợp số bit lớn hơn có thể thực hiện hoàn toàn tương tự.

Gọi 3 bit dữ liệu là d1, d2, d3 và Xe, Xo là 2 bit chẵn, lẻ thêm vào dữ liệu. Xe là giá trị bit phải thêm vào để hệ là hệ chẵn, Xo là giá trị bit phải thêm vào để hệ là hệ lẻ. Bảng chân lý của mạch tạo bit chẵn lẻ cho trường hợp dữ liệu 3 bit như sau:

Vào			Ra	
d1	d2	d3	Xe	Xo
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

Từ bảng chân lý ta có:

$$Xe = d1 \oplus d2 \oplus d3$$

$$Xo = \overline{Xe} = \overline{d1 \oplus d2 \oplus d3}$$

##### 2. Mạch kiểm tra chẵn lẻ

Bảng chân lý của mạch kiểm tra tính chẵn lẻ của hệ đã cho

Vào				Ra	
d1	d2	d3	X	Fe	Fo
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

Từ bảng chân lý ta thấy

$$F_e = d1 \oplus d2 \oplus d3 \oplus X$$

$$F_o = \overline{F_e}$$

Fe chỉ ra tính chẵn của hệ, nghĩa là nếu hệ chẵn  $F_e = 1$

Fo chỉ ra tính lẻ của hệ, nghĩa là nếu hệ lẻ  $F_o = 1$

*Chú ý:* phương pháp kiểm tra chẵn lẻ chỉ phát hiện được lỗi đơn hoặc số lỗi là lẻ mà không phát hiện được lỗi chẵn và không có khả năng sửa lỗi

## V. MẠCH PHÂN LOẠI NGẮT

Mạch phân loại ngắt là mạch có nhiệm vụ:

+ Tại thời điểm  $t$  nếu có ít nhất 1 trong các thiết bị ngoại vi cùng gửi yêu cầu ngắt tới bộ VXL thì mạch phải tạo ra tín hiệu yêu cầu ngắt IR (interrupt request) tới bộ VXL

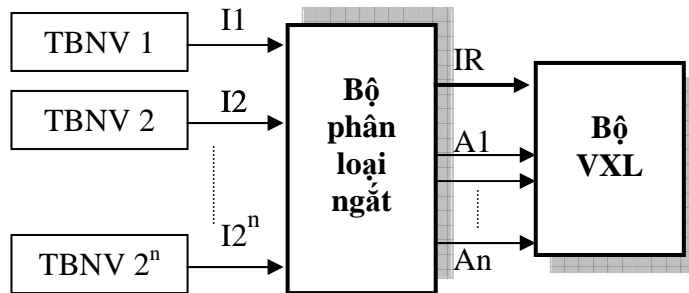
+ Tại thời điểm  $t$  nếu có nhiều thiết bị ngoại vi cùng gửi yêu cầu ngắt tới VXL thì mạch phải chỉ ra các thiết bị ngoại vi nào cần ưu tiên giải quyết ngắt tại thời điểm đó.

Sơ đồ khối của bộ phân loại ngắt được cho trong hình dưới đây

Trong đó ( $A_1, A_2 \dots A_n$ ) là các tổ hợp tương ứng với các thiết bị ngoại vi (TBNV)

IR = khi có ít nhất 1 trong  $2^n$  TBNV có yêu cầu ngắt tới bộ VXL

Như vậy bộ phân loại ngắt có  $2^n$  đầu vào và  $n+1$  đầu ra



Ví dụ: xây dựng sơ đồ bộ phân loại ngắt cho 4 thiết bị ngoại vi, với giả thiết các thiết bị ngoại vi được ưu tiên ngắt theo thứ tự P3, P2, P1, P0

Như vậy bộ phân loại ngắt sẽ có 4 đầu vào và 3 đầu ra ( IR và AB là địa chỉ của các TBNV). Khi đó ta có bảng chân lý của bộ phân loại ngắt đó như sau:

Vào				Ra		
P3	P2	P1	P0	IR	A	B
1	x	x	x	1	1	1
0	1	x	x	1	1	0
0	0	1	x	1	0	1
0	0	0	1	1	0	0
0	0	0	0	0	0	0

Từ bảng chân lý xác định được các hàm đầu ra như sau:

$$IR = P1 + P2 + P3$$

$$A = P3 + P2 \cdot \overline{P3}$$

$$B = P3 + P1 \cdot \overline{P2} \cdot \overline{P3}$$

## VI. BỘ CHỌN KÊNH VÀ PHÂN KÊNH (MULTIPLEXER AND DEMULTIPLEXER)

### 1. Bộ chọn kênh

Bộ dồn kênh là mạch có  $2^n$  đầu vào dữ liệu X, n đầu vào địa chỉ A, 1 đầu cho phép En và 1 đầu ra Y (có thể có mạch có thêm đầu ra  $\overline{Y}$ )

Nhiệm vụ của bộ chọn kênh là chuyển thông tin từ một đầu vào dữ liệu có địa chỉ được xác định nhờ các đầu vào địa chỉ đến đầu ra không đảo khi đầu vào cho phép ở trạng thái tích cực.

Tuỳ theo giá trị của n đầu vào địa chỉ mà đầu ra sẽ bằng một trong những giá trị ở đầu vào  $X_j$ . Cụ thể là nếu giá trị thập phân của tổ hợp  $(A_{n-1}A_{n-2}...A_0)$  bằng j thì  $Y = X_j$  nếu khi đó  $E_n = 1$ .

Hình bên là sơ đồ khối của một chọn 1 đầu vào từ  $2^n$  đầu vào

Hiện nay bộ MUX được dùng phân tử vạn năng để xây dựng những hợp khác, cụ thể là:

- + Tạo hàm logic
- + Tạo các dãy xung
- + Truyền dữ liệu

+ MUX được dùng như bộ thông tin ở dạng song song ở đầu vào tiếp ở đầu ra

- + Giải mã địa chỉ
- + ....

## 2. Bộ phân kênh

Bộ chọn kênh là mạch có 1 đầu vào dữ liệu X, n đầu vào địa chỉ A, 1 đầu cho và  $2^n$  đầu ra Y

Hình bên là sơ đồ khối của bộ

Nhiệm vụ của bộ phân kênh thực chức năng giải mã từ một địa chỉ của cho để tạo tín hiệu điều khiển của kênh chuyển thông tin từ một đầu vào dữ liệu có địa chỉ được xác định nhờ các đầu vào địa chỉ ra không đảo khi đầu vào cho phép ở trạng thái tích cực.

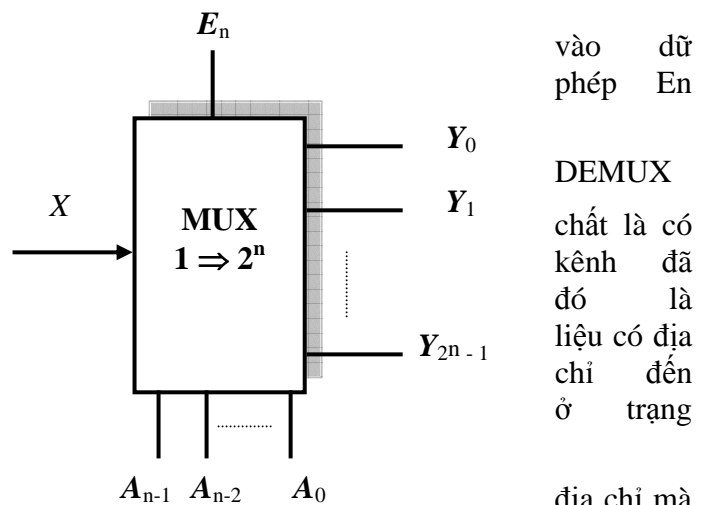
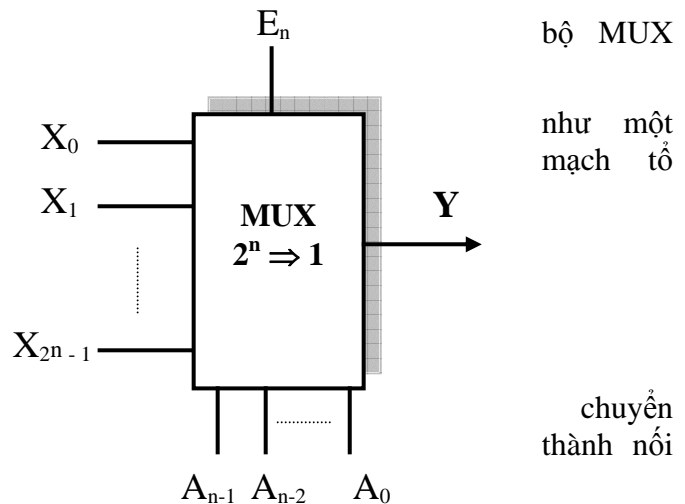
Tuỳ theo giá trị của n đầu vào địa chỉ mà đầu ra thứ i ( $Y_i$ ) sẽ bằng giá trị ở đầu vào X. Cụ thể là nếu giá trị thập phân của tổ hợp  $(A_{n-1}A_{n-2}...A_0)$  bằng i thì  $Y_i = X$  nếu khi đó  $E_n = 1$ .

## VII. BỘ CHUYỂN MÃ

Trong các hệ thống điện tử dùng mạch số, dữ liệu được xử lý và truyền đi dưới dạng từ nhị phân n bit, một từ n bit có thể biểu diễn cho  $2^n$  phần tử tin khác nhau. Từ nhị phân n bit gọi là mã (code) của phần tử tin tức. Có rất nhiều loại mã khác nhau được sử dụng cho từng mục đích khác nhau, dưới đây ta sẽ xét đến các mã tiêu biểu và các mạch tổ hợp thực hiện chuyển mã, gồm 2 loại là mã hoá và giải mã (ENCODER và DECODER)

### 1. Các loại mã tiêu biểu

#### a. Mã ký tự



+ Mã ASCII (American Standard Code for Information Interchange): dùng 8 bit để mã hoá cho bảng chữ cái và một số ký tự đặc biệt (mã này được sử dụng rộng rãi nhất để mã hoá ký tự cho các hệ thống xử lý văn bản)

+ Mã EBCDI (Extended Binary Coded Decimal Interchange): dùng 8 bit để mã hoá cho ký tự

+ Mã BAUDOT: dùng 5 bit để biểu diễn cho 1 ký tự, thường dùng cho teletype và bưu điện

## **b. Mã số**

Các loại mã thường sử dụng là nhị phân, dư 3, Gray, BCD ...

Phân này đã được mô tả cụ thể trong chương 1.

Ngoài ra còn một số loại mã đặc biệt như mã sửa sai là mã ngoài các bit mang thông tin còn có một số bit thêm vào để phát hiện và sửa lỗi, ví dụ: mã chẵn lẻ, mã CRC .... Mã 7 vạch có độ dài bằng 7 dùng để biểu diễn chữ số thập phân bằng đèn 7 thanh.

### **2. Mạch mã hoá - lập mã (ENCODER)**

Giả sử từ mã có  $n$  bit, khi đó sẽ có  $2n$  bộ giá trị khác nhau để biểu diễn cho các ký hiệu hoặc lệnh. Như vậy mỗi loại mã chỉ có số ký hiệu hoặc lệnh  $\leq 2n$

*Chú ý:* nếu  $N < 2n$  thì số tổ hợp không dùng đến có thể được dùng cho các mục đích khác, ví dụ như phát hiện hay sửa sai.

Dưới đây là ví dụ về việc mạch thực hiện mã hoá để tạo mã BCD 8421 (hay còn gọi là BCD tự nhiên, NBCD)

#### **Bộ mã hoá Thập phân sang BCD.**

Một bộ chuyển mã từ thập phân sang BCD, thường được gọi là bộ mã hoá, phải được sử dụng trong các hệ thống số vì các mạch logic về cơ bản là các thiết bị nhị phân.

Một bộ giải mã thập phân sang BCD là một mạch tổ hợp gồm 10 đầu vào (giá trị thập phân, từ 0 đến 9), và 4 đầu ra nhị phân. Bốn đầu ra thập phân để mã hoá các số không lớn hơn 9, cho nên được gọi là số BCD (Binary Coded Decimal). Trường hợp có nhiều hơn một đầu vào tích cực thì phải sử dụng bộ mã hoá ưu tiên, mà chỉ mã hoá đầu vào nào có trọng số lớn nhất. Như đã biết, mã BCD 8421 dùng 4 chữ số hệ 2 để mã hoá các con số từ 0 tới 9 của hệ 10 và có trọng số 8,4,2,1. Ta có bảng chân lý như sau:

Số hệ 10	A	B	C	D
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	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Từ bảng chân lý ta có:

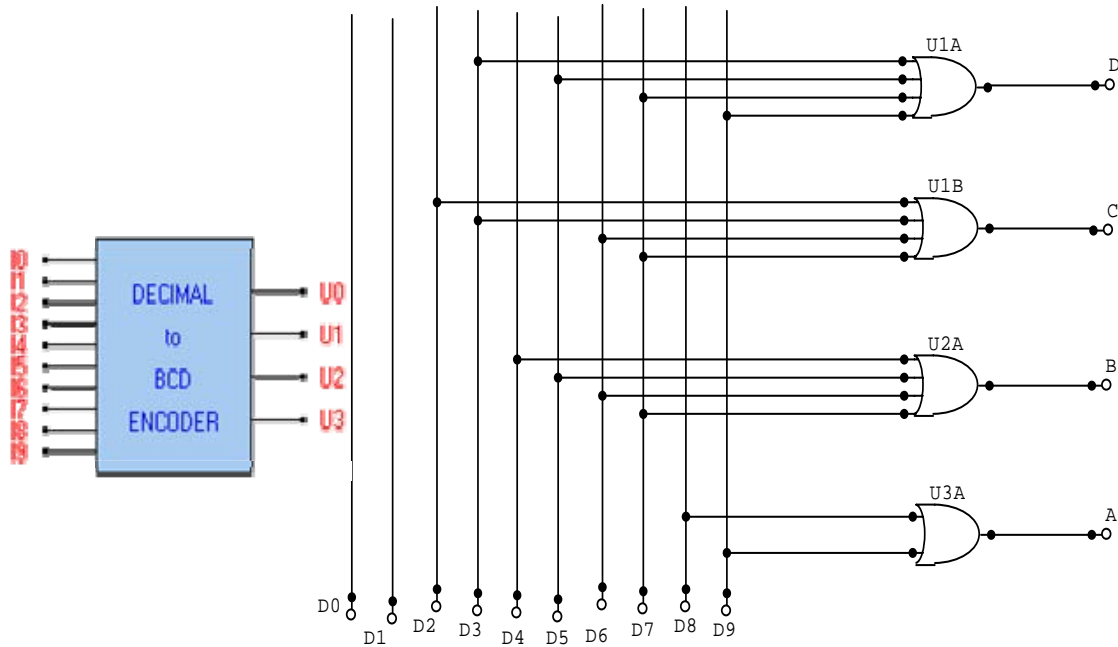
$$A = 8 + 9$$

$$B = 4 + 5 + 6 + 7$$

$$C = 2 + 3 + 6 + 7$$

$$D = 1 + 3 + 5 + 7 + 9$$

Như vậy mạch thực hiện mã hoá 10 – BCD 8421 có sơ đồ nguyên lý như sau:

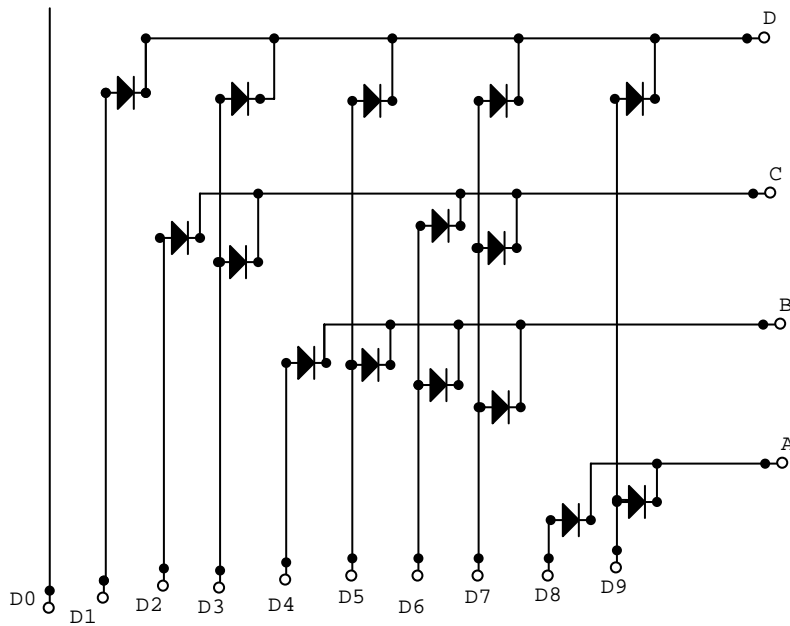


Khi một trong các đầu vào D0 – D9 có mức điện áp cao thì các đường ra ABCD sẽ có tín hiệu tương ứng. Ví dụ D5 có mức điện áp cao còn các đường khác có mức điện áp thấp, nghĩa là ta muốn mã hoá số 5, khi đó các đường ra B và D có mức điện áp cao còn A và C có mức điện áp thấp, tức ta có ABCD = 0101 như mong muốn.

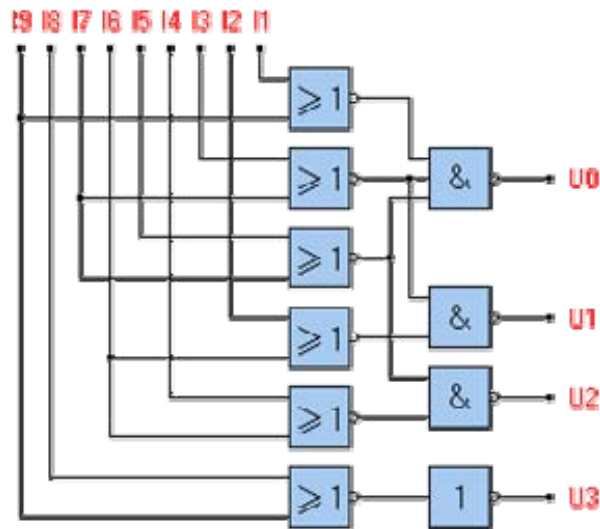
*Chú ý:* về cấu tạo, để đơn giản mạch OR thường chế tạo theo kiểu DL (diode logic).

Khi đó mạch trở thành như sau:

Chú ý: Mạch điện của bộ mã hoá không có mức ưu tiên (tức không có nhiều đường vào cùng



ở mức cao) được chỉ ra như ở hình D09.3. Bộ mã hoá không có đầu vào 0 vì, thông thường, nó không cần đến trong các mạch logic. Cũng có thể thực hiện bộ mã hoá trên theo sơ đồ dưới đây:



### 3. Mạch giải mã (DECODER)

Mạch này có chức năng ngược với bộ mã hoá, nghĩa là từ bộ bit  $n$  bit hệ 2 cần tìm lại được 1 trong  $N$  ký hiệu hoặc lệnh tương ứng.

#### Bộ giải mã BCD sang thập phân.

Bộ giải mã BCD sang hệ thập phân là một mạch tổ hợp có 4 đầu vào nhị phân và 10 đầu ra thập phân. Đầu vào là mã BCD và sẽ kích hoạt đầu ra tương ứng với đầu vào.

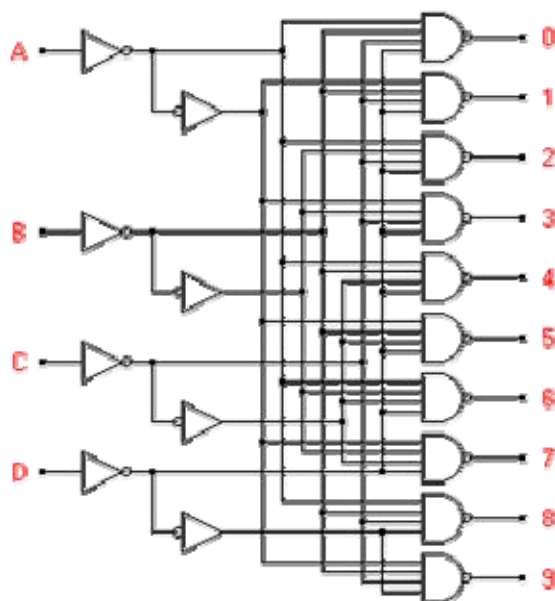


Dưới đây là bảng chân lý của bộ giải mã BCD-thập phân, với đầu ra tích cực ở mức logic âm, ứng với vi mạch SN 74LS42. Có thể thấy rằng các số lớn hơn 9 sẽ không kích hoạt bất kỳ đầu ra nào.

SN74LS42 là một vi mạch giải mã BCD – thập phân được sử dụng rộng rãi, dưới đây là

N	I3	I2	I1	I0	O0	O1	O2	O3	O4	O5	O6	O7	O8	O9
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0
10	1	0	1	0	1	1	1	1	1	1	1	1	1	1
11	1	0	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	0	0	1	1	1	1	1	1	1	1	1	1
13	1	1	0	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	0	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1

sơ đồ bên trong của vi mạch này để minh họa cho việc chuyển đổi mã.



### **Bộ giải mã BCD sang 7 vạch.**

Đèn 7 vạch được sử dụng để hiển thị dữ liệu được xử lý bởi thiết bị điện tử số. Chúng có thể hiển thị các số từ 0 đến 9 và các chữ cái từ A đến F và một vài ký tự khác.

Thiết bị hiển thị này có thể được điều khiển bởi bộ giải mã mà sẽ chiếu sáng các vạch (đoạn-segment) của đèn phụ thuộc vào số BCD tại đầu vào. Các bộ giải mã này cũng chứa các bộ đệm công suất để cấp dòng cho đèn, do vậy, nó còn được gọi là bộ điều khiển-giải mã (Decoder-Driver).

Bộ mã hoá này có 4 đầu vào tương ứng với 4 bit mã BCD và 7 đầu ra, mỗi đầu sẽ điều khiển một vạch của đèn 7 vạch. Hình dưới chỉ ra mô hình của các vạch trong thiết bị hiển thị (đèn) 7 vạch và các số có thể hiển thị.

*Hiển thị 7 vạch*

Đèn hiển thị 7 vạch bao gồm các vạch (đoạn sáng – segment) nhỏ. Chúng có thể biểu diễn tới 16 ký tự trong đó có 10 số và 6 chữ cái như hình dưới đây:



Các mã đầu vào từ 0 -9 hiển thị các chữ số của hệ thập phân. Các mã đầu vào từ 9-14 ứng với các ký hiệu đặc biệt như đã nêu, còn mã 15 sẽ tắt tất cả các vạch.

Đoạn sáng thứ 8 của đèn hiển thị là dấu chấm thập phân (dp). Các thiết bị hiển thị loại này có nhiều kiểu với màu sắc, kích thước khác nhau và có đặc tính phát sáng rất tốt.

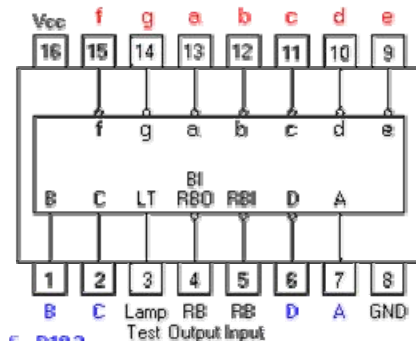
Về mặt điện, các LED hoạt động như diode chuẩn, chỉ khác là khi phân cực thuận đòi hỏi điện áp giữa anode và Cathode cao hơn. Để có cường độ sáng không đổi, thiết bị hiển thị phải được cấp đủ dòng.

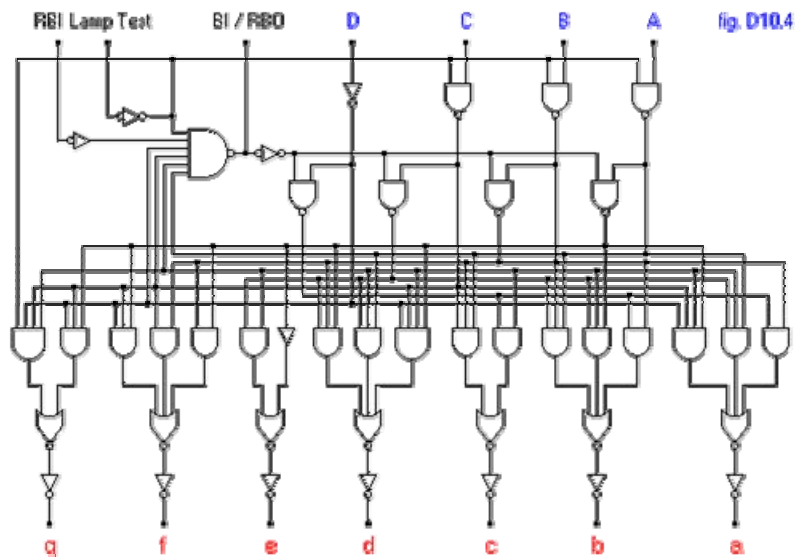
Các thiết bị hiển thị 7 vạch có thể có cực tính:

- + với kiểu cathode chung, điều khiển bởi mức logic dương.
- + với kiểu anode chung, điều khiển bởi mức logic âm.

**Vi mạch TTL 74LS47** là một bộ điều khiển- hiển thị được dùng phổ biến. Vi mạch này có các đầu ra đảo do đó sử dụng với LED anode chung. Hình dưới đây chỉ ra sơ đồ chân và sơ đồ mạch bên trong của vi mạch.

Chân LT (Lamp Test) được dùng để kiểm tra tình trạng hoạt động (sống hay chết) của các vạch; trong khi chân RB (Ripper Blanking) được dùng để tắt tất cả các vạch khi yêu cầu ở trạng thái không hiển thị số.





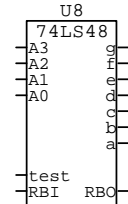
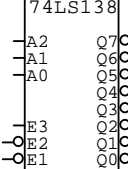
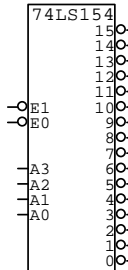
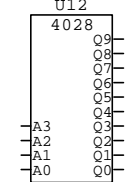
#### 4. Thiết kế mạch chuyển mã

Các bước thiết kế mạch chuyển mã hoàn toàn tương tự như thiết kế mạch tổ hợp, nghĩa là qua các bước sau:

- + Lập bảng chân lý của mạch
- + Tối thiểu hoá các hàm ra
- + Xây dựng sơ đồ mạch với loại cổng cho trước

#### 5. Một số vi mạch chuyển mã thông dụng

Tên vi mạch	Hình dạng thực tế
74147 Bộ mã hoá số hệ thập phân thành mã BCD 8421	
7447 / 74LS47 Chuyển đổi mã BCD thành mã 7 vạch để hiển thị theo hệ 10 bằng led 7 thanh có anode chung	
7442 Giải mã BCD 8421 thành số hệ thập phân	

<p>7448</p> <p>Chuyển đổi mã BCD thành mã 7 vạch để hiển thị theo hệ 10 bằng led 7 thanh có cathode chung</p>	
<p>74LS138</p> <p>Bộ giải mã / phân kênh 1 - 8</p>	
<p>74154</p> <p>Bộ giải mã / phân kênh 4 đường thành 16 đường</p>	
<p>4028</p> <p>Giải mã BCD 8421 – hệ 10</p>	

## **PHẦN III**

### **MẠCH DÂY**

## CHƯƠNG 7: CÁC PHẦN TỬ NHỚ CƠ BẢN

### I. KHÁI NIỆM CHUNG

Như đã nói, mạch dãy là mạch có tín hiệu ra không chỉ phụ thuộc vào tín hiệu vào mà còn phụ thuộc vào trạng thái trong của mạch, nghĩa là mạch có khả năng lưu trữ để nhớ trạng thái.

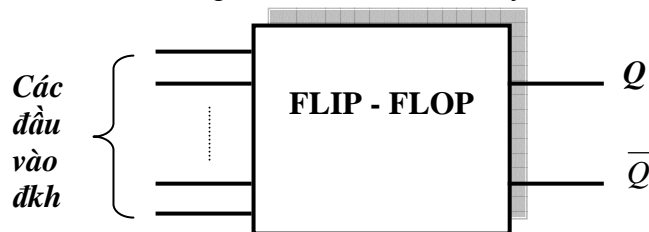
Các phần tử nhớ cơ bản để tạo thành mạch dãy được gọi là các flip-flop (mạch bập bênh), chúng là các phần tử nhớ đơn bit vì chỉ có khả năng nhớ được 1 chữ số nhị phân.

### II. ĐỊNH NGHĨA VÀ PHÂN LOẠI

#### 1. Định nghĩa

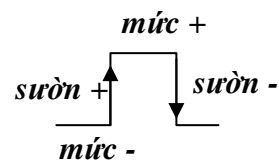
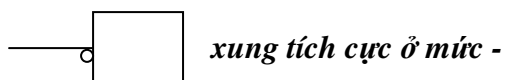
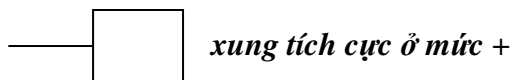
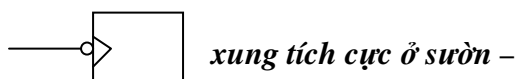
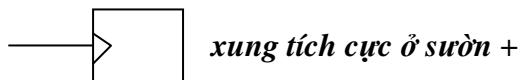
Flip – flop / FF là phần tử có khả năng lưu trữ 1 trong 2 trạng thái là 0 hoặc 1.

FF thường có nhiều đầu vào và 2 đầu ra có tính liên hợp (đầu ra này là đảo của đầu ra kia), ký hiệu là  $Q$  và  $\bar{Q}$ . Tên gọi của các đầu vào tùy thuộc vào từng loại FF, sẽ nói cụ thể



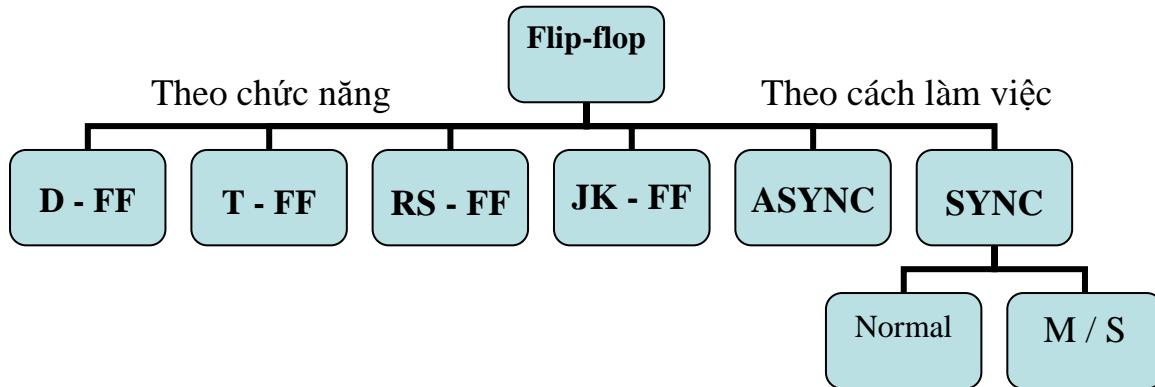
sau.

Ký hiệu về tính tích cực trong mạch FF:



## 2. Phân loại FF

Có thể phân loại FF theo 2 cách như sau:



## 3. Biểu diễn FF

Để mô tả một FF người ta có thể dùng 1 trong 3 cách sau:

- + Dùng bảng chân lý
- + Đồ hình chuyển đổi trạng thái
- + Phương trình đặc trưng

## III. CÁC LOẠI FF VÀ ĐIỀU KIỆN ĐỒNG BỘ

### 1. Flip-Flop kiểu RS

RS FF là mạch Flip-Flop đơn giản nhất chỉ có 2 đầu vào điều khiển R (reset – xóa) và S (set – thiết lập), RS-FF có thể được xây dựng từ 2 cổng NAND hay 2 cổng NOR. Hình dưới đây chỉ ra bảng trạng thái rút gọn và sơ đồ của mạch với các cổng NAND và ký hiệu của RS -

S	R	Q	$\bar{Q}$
0	0	$Q_n$	$\bar{Q}_n$
0	1	0	1
1	0	1	0
1	1	X	X

$Q_n$  = last state  
 X = indefinite state

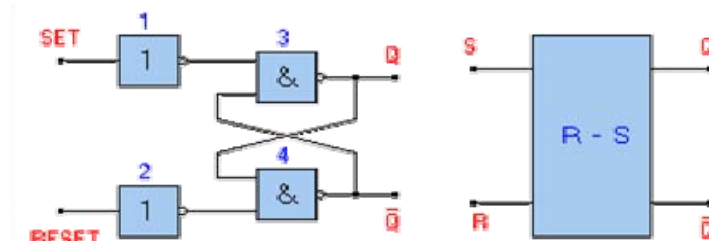
FF

R, S là các đầu vào điều khiển

$Q_n$  là trạng thái của FF tại thời điểm hiện tại t

Q là trạng thái sẽ chuyển tới của FF sau thời gian quá độ, tức trạng thái của FF ở thời điểm tiếp theo

Giả thiết, tại thời điểm bắt đầu, S=1 và R= 0. Mức đầu ra của cổng 1 là thấp (0) và điều này tạo nên trạng thái cao trên đầu ra của cổng 3 (Q=1). Tuy nhiên, đầu ra của cổng 2 ở mức



cao, bởi thế cổng 4 có cả hai đầu vào đều ở mức cao (từ cổng 2 và 3) nên đầu ra của nó sẽ ở mức thấp ( $\overline{Q} = 0$ ). Flip-Flop ở trạng thái SET và đầu ra  $Q = 1$  bất kể  $Q_n$  trước đó là 0 hay 1.

Khi  $S=0$  và  $R=1$ , Flip-Flop sẽ chuyển trạng thái và đầu ra:  $Q=0$ ;  $\overline{Q}=1$ . Trường hợp này, Flip-Flop được RESET hay xóa về 0, trạng thái logic 0 trên  $Q$  dù trước đó  $Q_n$  là 0 hay 1.

Trạng thái mà trong đó, cả hai đầu vào đều ở mức  $R = S = 0$  được gọi là trạng thái nhớ, vì đầu vào sẽ duy trì trạng thái trước đó,  $Q_n$ .

Nếu đầu vào SET và RESET đồng thời ở mức cao ( $S = R = 1$ ), ta sẽ có trạng thái sau:

$$Q = \overline{Q} = 1.$$

được coi là trạng thái *không xác định (không sử dụng hay cấm)* R-S Flip-Flop không được thiết kế để hoạt động trong trạng thái  $R=S=1$ .

*Nhận xét:*

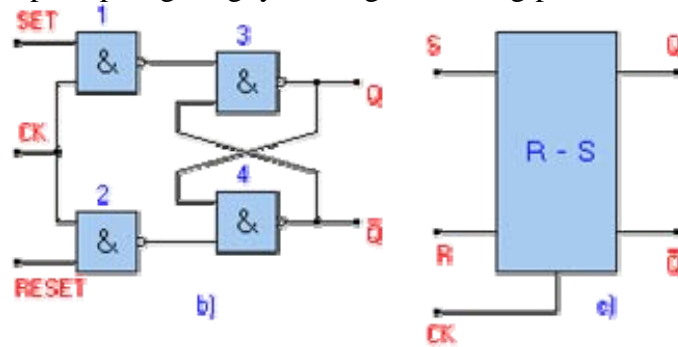
- + Phương trình đặc trưng của RS – FF là  $Q = S + Q_n \overline{R}$
- + S luôn đưa Q về giá trị 1
- + R luôn đưa Q về giá trị 0
- + FF tắt, tức chuyển trạng thái từ 1 sang 0 với phương trình  $T_{off} = \overline{S}RQ_n$
- + FF bật, tức chuyển trạng thái từ 0 sang 1 với phương trình  $T_{on} = S\overline{R}Q_n$

### RS Flip-Flop với đầu vào xung nhịp

Các hệ thống tuần tự thường yêu cầu các Flip-Flop thay đổi trạng thái đồng bộ với xung nhịp. Khi đó người ta coi FF như một mạch chốt hay RS FF đồng bộ hay RST FF hay RS FF nhịp. Điều này có thể thực hiện được bởi việc thay đổi mạch như sau:

Khi chưa có xung nhịp, Flip-Flop sẽ giữ nguyên trạng thái không phụ thuộc vào R và S

$t^n$	$t^{n+1}$		
CK	S	R	$Q_{n+1}$
↑	0	0	$Q_n$
↑	0	1	0
↑	1	0	1
↑	1	1	?
0	X	X	$Q_n$
1	X	X	$Q_n$



(trạng thái nhớ), nghĩa là trạng thái của FF bị chốt lại .

Khi có xung nhịp:

- nếu  $R = S = 0$ , đầu ra của Flip-Flop sẽ không đổi;
- nếu  $R = 0, S = 1$ , Flip-Flop sẽ có trạng thái đầu ra:  $Q = 1, \overline{Q} = 0$ ;
- nếu  $R = 1, S = 0$  ta sẽ có trạng thái đầu ra:  $Q = 0$  và  $\overline{Q} = 1$ .



Tóm lại: Khi không có xung nhịp FF không thay đổi trạng thái (không phụ thuộc vào tín hiệu đầu vào điều khiển) và chỉ khi có xung nhịp Ck mạch mới làm việc theo bảng chức năng (phụ thuộc vào tín hiệu đầu vào điều khiển)

### Các biến thể của RS – FF

Để sử dụng được cả tổ hợp cấm R = S = 1 người ta chế tạo các biến thể của RS – FF như FF R, FF S và FF E. Các FF này được sử dụng khá rộng rãi trong các khâu điều khiển của hệ thống số.

Flip – Flop R: ứng với tổ hợp cấm đầu ra Q = 0

Flip – flop S : ứng với tổ hợp cấm đầu ra Q = 1

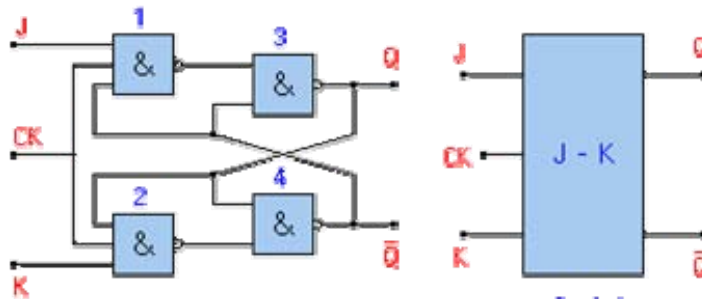
Flip – flop E: ứng với tổ hợp cấm FF không chuyển trạng thái

### 2. JK Flip-Flop.

JK – FF là một loại FF vạn năng và có nhiều ứng dụng

JK Flip-Flop cũng tương tự như một R-S khoá và có các đầu ra hồi tiếp về đầu vào như hình dưới đây

J	K	Q <sub>n+1</sub>
0	0	Q <sub>n</sub>
0	1	0
1	0	1
1	1	$\bar{Q}_n$



Một ưu điểm của J-K Flip-Flop là nó không có trạng thái không xác định như của R-S khi cả hai đầu vào ở mức 1.

Ví dụ, nếu J = K = 1; Q = 1 và  $\bar{Q} = 0$ ; khi có xung nhịp đến, chỉ có cổng 2 cho phép truyền dữ liệu vào, còn cổng 1 sẽ ngăn lại. Mức 0 tại đầu ra của cổng 2 sẽ khiến cho phần tử nhớ chuyển trạng thái. Như vậy, khi các đầu vào đều ở mức cao, đầu ra sẽ đảo hay lật (toggle) trạng thái tại mỗi xung nhịp vào.

Nhận xét:

+ Phương trình đặc trưng của JK – FF có dạng:  $Q = J \cdot \bar{Q} + \bar{K}Q$

+ Có sự tương ứng giữa JK và RS, J tương ứng với S, K tương ứng với R nhưng tổ hợp 11 trong JK vẫn được sử dụng mà không bị cấm như trong RS

+ JK = 00 FF luôn giữ nguyên trạng thái

JK = 01 FF luôn chuyển đến trạng thái 0

JK = 10 FF luôn chuyển đến trạng thái 1

JK = 11 FF luôn lật trạng thái

JK Flip-Flop chỉ có một khả năng cho trạng thái không xác định, đó là khi độ dài xung nhịp lớn hơn thời gian truyền đạt. Giả thiết, Flip-Flop đang ở trong trạng thái: Q = 0,  $\bar{Q} = 1$  và J = K = 1;

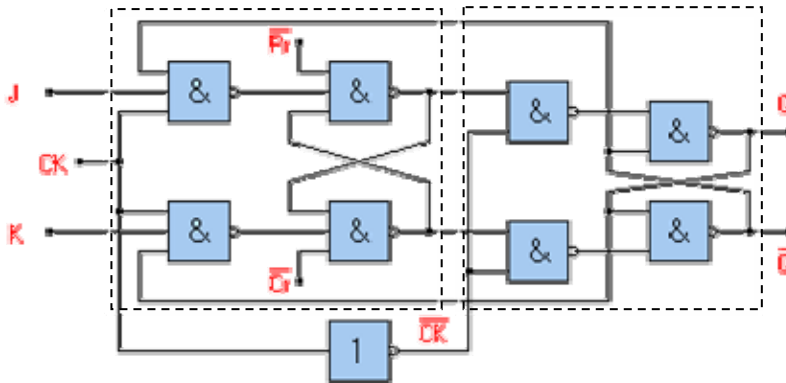
Khi có xung nhịp đến, đầu ra sẽ đảo trạng thái sau một khoảng thời gian truyền đạt “t” :

$$Q = 1 \text{ và } \overline{Q} = 0;$$

Tuy nhiên, do vẫn có xung nhịp kích thích, đầu ra sẽ hồi tiếp trở lại đầu vào khiến mạch có xu hướng dao động giữa 0 và 1. Bởi thế, tại thời điểm cuối của xung nhịp, trạng thái của Flip-Flop sẽ không được xác định. Hiện tượng này gọi là hiện tượng đua vòng quanh và có thể gây nên chuyển biến sai nhầm của mạch. Người ta khắc phục hiện tượng này bằng cách sử dụng mạch JK FF kiểu chủ tớ.

### JK Flip-Flop kiểu chủ tớ.

JK FF kiểu chủ tớ có sơ đồ cấu trúc như sau:



Mạch bao gồm 2 nửa giống nhau, mỗi nửa là một RS Flip-Flop, FF thứ nhất gọi là FF master (chủ) và FF thứ 2 gọi là FF slave (tớ). Đầu vào của FF chủ là đầu vào của mạch và đầu ra của FF tớ là đầu ra của mạch. Tín hiệu hồi tiếp từ đầu ra của FF tớ về đầu vào của FF chủ. Các xung đưa tới phần tớ là đảo với xung đưa tới phần chủ.

Các đầu vào Preset và Clear sẽ có chức năng giống như của đầu vào Set và Reset. Chúng tác động đến đầu ra một cách không đồng bộ, tức chúng sẽ thay đổi trạng thái đầu ra mà không phụ thuộc vào sự có mặt của xung nhịp; và chủ yếu để đưa đầu ra về một trạng thái đã biết nào đó. (người ta còn gọi đây là các đầu vào điều khiển trực tiếp)

Giả thiết các đầu vào này là không tích cực (khi  $Pr = Cl = 1$ ), khi có xung nhịp đến, Flip-Flop sẽ thay đổi trạng thái như trong bảng chân lý sau:

CK	J	K	$Q_{n+1}$
0	x	x	$Q_n$
	0	0	$Q_n$
	0	1	0
	1	0	0
	1	1	$\overline{Q_n}$

Với  $Q_{n+1}$ : trạng thái kế tiếp;

$Q_n$  : trạng thái trước đó.

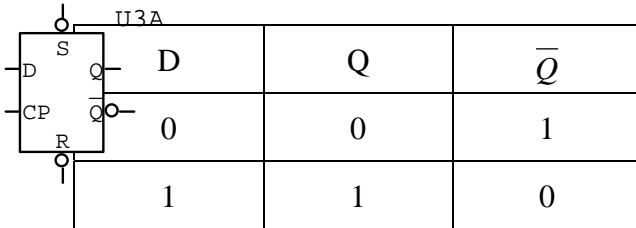
x: trạng thái không xác định.

Trong khoảng thời gian xung nhịp là cao, phần Tớ khoá, bởi thế các đầu ra Q và  $\overline{Q}$  sẽ không thay đổi. Khi xung nhịp chuyển từ 1 về 0, khối Tớ sẽ chuyển trạng thái trong khi khối

Chủ sẽ khoá. Nói cách khác, dữ liệu trên J và K trước tiên được truyền đến khối Chủ tại sườn tăng của của xung nhịp và truyền tới khối Tớ tại sườn xuống; như vậy, trạng thái không xác định của đầu ra như trường hợp J-K Flip-Flop sẽ được loại bỏ.

### 3. D Flip-Flop

D FF là loại FF chỉ có một đầu vào điều khiển D



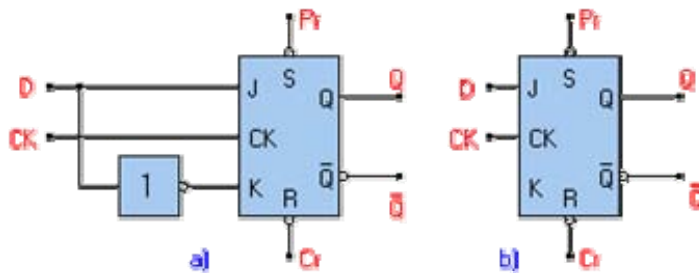
Phương trình đặc trưng của

D là  $Q = D$

Thực chất D FF chính là một khâu trễ có thời gian  $\Delta t$  là thời gian quá độ của mạch. Đầu ra Q chính là trễ của đầu vào sau khoảng thời gian  $\Delta t$ , vì vậy FF này có tên là D FF (delay FF)

#### Chế tạo D FF từ JK FF

Nếu từ một JK Flip-Flop thêm vào một bộ đảo như hình dưới thì đầu vào K luôn là bù của J và sẽ tạo nên mạch D Flip-Flop. Hoạt động của nó rất đơn giản, khi có xung đồng hồ đến, dữ liệu tại đầu vào sẽ được truyền và giữ nguyên tại đầu ra.

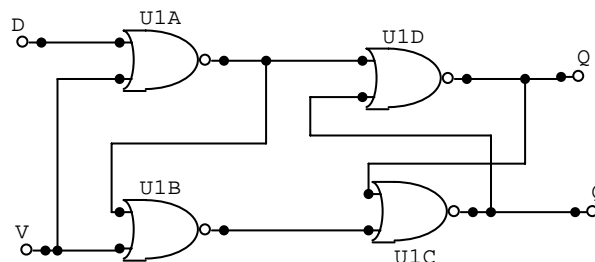


Ngoài ra cũng có thể chế tạo D FF từ RST FF bằng cách thêm cổng NOT giữa hai đầu vào S và R tương ứng với J và K như ở hình trên.

#### Biến thể của D FF

Trên thực tế người ta sử dụng biến thể của D là DV FF. Loại FF này có bảng trạng thái và sơ đồ xây dựng từ các cổng NOR như sau:

V	D	$Q_{n+1}$
1	0	0
1	1	1
0	0	$Q_n$
0	1	$Q_n$



Từ bảng trạng thái ta thấy:

+ Khi  $V = 1$  FF DV hoạt động như một FF D thông thường

+ Khi  $V = 0$  FF không đổi trạng thái với bất kỳ mức logic nào của D

#### 4. Flip-Flop kiểu T.

FF T là một FF có 2 đầu ra và 1 đầu vào T. T FF có bảng trạng thái như sau:

T	$Q_{n+1}$
0	$Q_n$
1	$\overline{Q_n}$

Khi  $T = 0$  FF giữ nguyên trạng thái

Khi  $T = 1$  FF lật trạng thái (toggle)

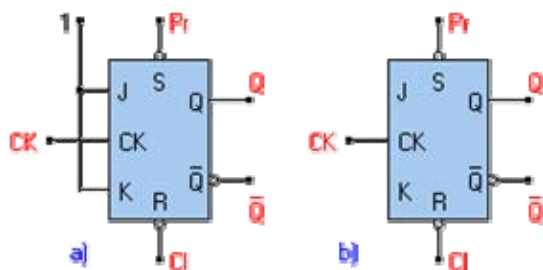
Phương trình đặc trưng của T FF:  $Q = T \oplus Q_n$

Như vậy mạch T FF thay đổi trạng thái tuần tự theo mỗi lần có xung kích thích

*Chú ý:* Khi đầu vào T có thời gian tồn tại ở mức logic cao trong một khoảng dài hơn so với thời gian chuyển trạng thái (thời gian trễ) của mạch thì mạch sẽ tiếp tục lật trạng thái tới khi hết thời gian tồn tại ở mức logic cao của T, quá trình đó làm cho việc xác định chính xác mạch đang ở trạng thái nào là không thể, do đó T chỉ có thể làm việc ở chế độ đồng bộ (vì thực tế thời gian tồn tại mức logic cao của T luôn lớn hơn rất nhiều thời gian trễ của mạch)

#### Chế tạo T FF từ JK FF

Rõ ràng T FF đơn giản là một JK Flip-Flop với cả J và K đều ở mức logic 1.



Vì  $J = K = 1$  nên Flip-Flop này sẽ lật (Toggle) trạng thái mỗi khi xung nhịp chuyển từ 1 về 0.

Hình bên là sơ đồ mạch và ký hiệu của T Flip-Flop.

#### Biến thể của T FF

Trên thực tế người ta sử dụng biến thể của T là TV FF. Loại FF này có bảng trạng thái như sau:

V	T	$Q_{n+1}$
1	0	$Q_n$
1	1	$\overline{Q_n}$
0	0	$Q_n$
0	1	$Q_n$

Từ bảng trạng thái ta thấy:

+ Khi  $V = 1$  FF TV hoạt động như một FF T thông thường

+ Khi  $V = 0$  FF không đổi trạng thái với bất kỳ mức logic nào của T

#### Nhận xét chung về chế độ làm việc của các loại FF:

+ Các D FF và RS FF có thể làm việc ở chế độ đồng bộ

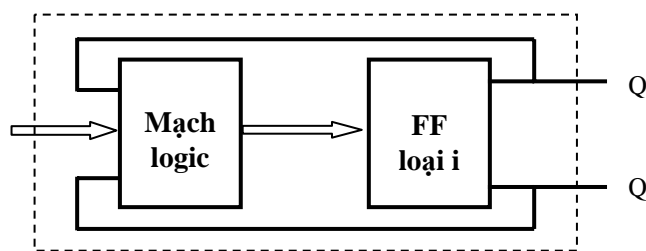
hoặc không đồng bộ vì với mỗi tập tín hiệu vào điều khiển luôn tồn tại ít nhất 1 trong các trạng thái ổn định ( $Q = Q_n$ )

+ 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 (chuyển trạng thái liên tục giữa 0 và 1). Khi JK = 11 hoặc T = 1 hai loại FF sẽ dao động, do đó chúng luôn phải làm việc ở chế độ đồng bộ.

#### IV. CHUYỂN ĐỔI GIỮA CÁC LOẠI FF

4 loại FF vừa xét ở trên có thể chuyển đổi lẫn cho nhau

Phương pháp chuyển đổi giữa loại FF i thành FF j được mô hình hoá theo sơ đồ sau:



Các bước thực hiện:

+ Xác định hệ hàm  $i = f(j, Q)$  theo bảng các đầu vào kích của các FF

+ Tối thiểu hoá các hàm này và xây dựng sơ đồ

Bảng đầu vào kích của các FF

$Q_n$	$Q_{n+1}$	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

ví dụ: thiết kế mạch tổ hợp chuyển đổi RS FF thành JK FF

Trước hết ra sẽ thiết kế mạch logic của hàm

$$R = f(Q, J, K)$$

$$S = g(Q, J, K)$$

Lập bảng Karnaugh của R theo Q, J, K ta có:

Q / JK	00	01	11	10

0	X	X		
1		1	1	

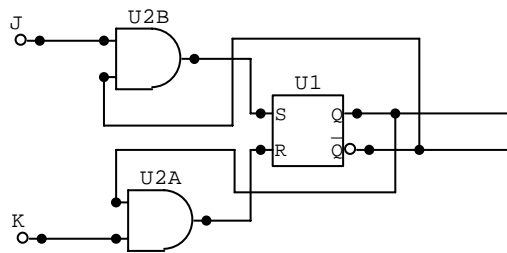
Vậy:  $R = QK$

Lập bảng Karnaugh của S theo Q, J, K ta có:

Q / JK	00	01	11	10
0			1	1
1	X			X

Vậy:  $S = J \cdot \bar{Q}$

Như vậy mạch thực hiện chuyển đổi từ RS FF sang JK FF sẽ có dạng như sau:



# CHƯƠNG 8 BỘ ĐẾM

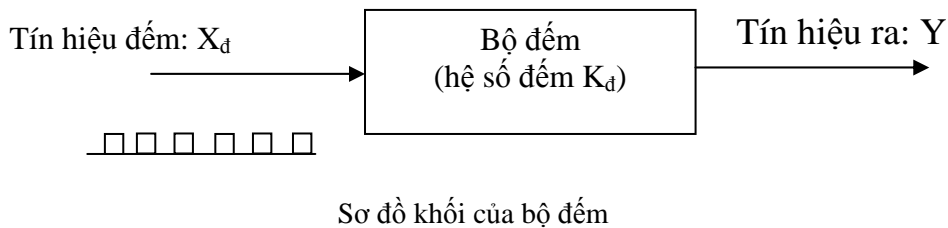
## I. ĐỊNH NGHĨA VÀ PHÂN LOẠI

### 1. Định nghĩa:

Bộ đếm là một mạch dãy tuần hoàn có các đặc điểm sau:

- + Một đầu vào( đếm) và một đầu ra(kết quả)
- + Số trạng thái trong bằng hệ số đếm

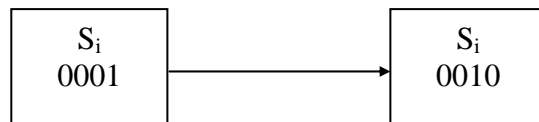
Dưới tác dụng của tín hiệu vào, bộ đếm sẽ chuyển trạng thái này đến trạng thái khác, theo một trình tự nhất định. Cứ sau  $K_d$  tín hiệu vào đếm, mạch lại trở về trạng thái ban đầu:



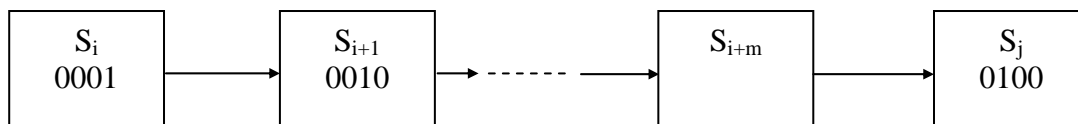
### 2. Phân loại

a, Phân loại theo cách làm việc:

+ Bộ đếm đồng bộ(Synchronous Counter) là bộ đếm mà các FF dùng(để mã hoá cho các trạng thái trong của bộ đếm) cùng một lúc khi có tín hiệu vào đếm mà không qua các trạng thái trung gian. Các tín hiệu xung nhịp( $C_k$ ) được đưa đồng thời vào các FF



+ Bộ đếm không đồng bộ(Asynchronous Counter), trong bộ đếm tồn tại ít nhất một cặp chuyển trạng thái( $S_i \rightarrow S_j$ ) mà trong đó các FF không thay đổi trạng thái cùng một lúc, tín hiệu xung nhịp  $C_k$  không được đưa đồng thời vào các FF.



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 FF này sẽ làm thay đổi trạng thái của FF khác, cho đến khi bộ đếm ở trạng thái cân bằng ổn định

b. Phân loại theo hệ số đếm:

- Bộ đếm có hệ số đếm  $K_d = 2^n$  (trong đó  $n$  là số tự nhiên), ví dụ  $K_d = 2, 4, 6, 8, 10, \dots$
- Bộ đếm có hệ số đếm  $K_d \neq 2^n$  (trong đó  $n$  là số tự nhiên), ví dụ  $K_d = 3, 5, 7, 9, 11, \dots$

c. Phân loại theo hướng đếm

- Bộ đếm thuận, còn gọi là bộ đếm tăng: mỗi khi có tín hiệu vào đếm giá trị của bộ đếm tăng lên 1

- Bộ đếm nghịch, còn gọi là bộ đếm giảm: mỗi khi có tín hiệu vào đếm giá trị của bộ đếm giảm đi 1.

Khái niệm bộ đếm thuận nghịch trong kỹ thuật số chỉ mang tính tương đối, tùy theo việc mã hoá (quan điểm) mà bộ đếm sẽ là thuận hay nghịch, có nghĩa là một bộ đếm sẽ là đếm thuận khi mã hoá kiểu này, nhưng sẽ là đếm nghịch khi mã hoá kiểu khác.

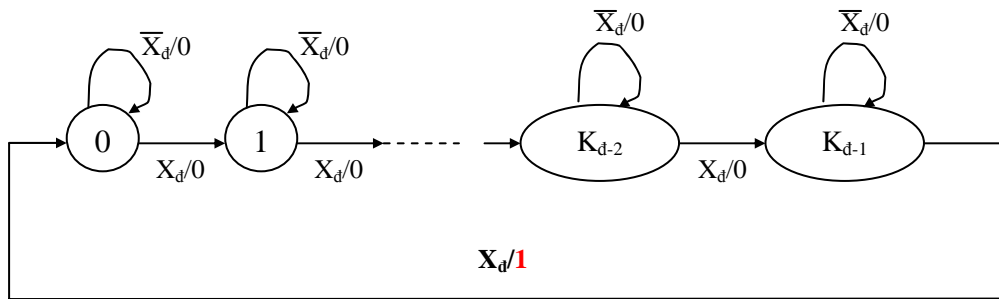
Có thể có loại thứ 3 là vừa thuận vừa nghịch tùy theo tín hiệu điều khiển.

d. Phân loại theo khả năng lập trình

- Bộ đếm có khả năng chương trình hoá
- Bộ đếm không có khả năng chương trình hoá.

Chú ý: một bộ đếm cụ thể thường bao gồm tất cả các đặc tính trên: ví dụ Bộ đếm có  $K_d=9$ , đếm tăng, đồng bộ, không thể chương trình hoá...

3. Đồ hình trạng thái của bộ đếm:



Đồ hình trạng thái của bộ đếm

- Đỉnh: là các trạng thái bên trong của bộ đếm, gồm có  $K_d$  trạng thái, tức là có  $K_d$  đỉnh.
- Cung: Là tín hiệu vào đếm/ kết quả ra đếm; khi không có tín hiệu vào đếm (tín hiệu vào đếm không tích cực) -  $X_d = 0$  (đảo), trạng thái của bộ đếm giữ nguyên; khi có tín hiệu vào đếm (tín hiệu vào đếm tích cực), bộ đếm thay đổi trạng thái. Tín hiệu ra của bộ đếm chỉ xuất hiện ( $Y=1$ ) khi bộ đếm ở trạng thái  $K_d-1$  ( $S_{K_d-1}$ ) và có tín hiệu vào  $X_d$

Tức là  $Y=(S_{K_d-1})X_d$

Và khi có tín hiệu ra mạch sẽ trở về trạng thái ban đầu, và quá trình đếm tiếp tục.

VD: Giả sử đếm số xe trên một đoạn đường, theo nguyên tắc có xe đi qua một vạch ngang thì tiến hành đếm

Khi đó, tín hiệu vào đếm tích cực  $X_d$  chính là xe đã đi qua vạch, giả sử khi tiến hành đếm chưa có xe nào đi qua, khi có 1 xe đi qua, thì trạng thái bộ đếm thay đổi: số xe tăng lên  $0 \rightarrow 1$ , cứ như thế cho đến 9, khi đã đến 9 thì nếu có một xe đi qua ta sẽ có 1\_0 xe, viết 0 nhớ 1 (tín hiệu ra là 1 - hàng chục), bộ đếm quay về trạng thái ban đầu 0 (ta hay nói là viết 0 nhớ 1).

**Chú ý:** muốn quan sát được trạng thái của bộ đếm, thì cần có mạch dải mã, như ví dụ trên thì tiếng Việt quy ước đọc là: một, hai, ba, bốn... thì người biết tiếng Việt sẽ hiệu là đã có 1, 2, 3, 4,... xe đã đi qua vạch.



## II. MÃ CỦA BỘ ĐẾM.

Giống như mạch dãy nói chung các trạng thái 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.

Các mã thường dùng để thiết kế bộ đếm:

### a. Mã nhị phân:

Mã nhị phân là loại mã mà các bit của nó có trọng số là 1-2-4-8-16-32-..., bit trẻ nhất(LSB) là  $1=2^0$ , bit tiếp theo tương ứng là  $2=2^1$ , ... , và bit già nhất MSB là  $2^{n-1}$ .

Dùng n bit nhị phân mã hoá được  $2^n$  trạng thái.

### b. Mã Gray:

Là loại mã không có trọng số, khoảng cách Hamming giữa 2 từ mã kề nhau là 1(2 từ mã kề nhau chỉ khác nhau một bit).

Dùng n bit nhị phân mã hoá được  $2^n$  trạng thái.

### c. Mã BCD

Là mã nhị phân mã hoá số thập phân, mã này dùng 4 chữ số nhị phân để mã hoá một chữ số thập phân, chúng được gọi là decard. Chỉ dùng 10 tổ hợp để mã hoá số từ 0(0000) đến 9(1001), các số lớn hơn thì sẽ dùng tổ hợp của chúng.

Ví dụ: 12=0001.0010

**Chú ý:** các loại mã trên xem ở bảng mã

### d. Mã Johnson:

Dùng n bit nhị phân, sẽ mã hoá được  $2n$  trạng thái, theo nguyên tắc:

+ Hai từ mã kề nhau chỉ khác nhau ở một bit.

+ Trong bảng mã các bit bằng 1 được đẩy dần lên từ bit trẻ nhất đến bit già nhất, và khi đẩy hết thì nó lại voi dần đi từ bit trẻ nhất(số bit 1 tăng dần từ trái sang phải, khi đầy thì lại giảm dần trái sang phải):

VD: n=2                      tổ hợp thứ nhất : 0 0

                                    tổ hợp thứ hai : 1 0

                                    tổ hợp thứ ba : 1 1

                                    tổ hợp thứ tư : 0 1

n=3                            tổ hợp thứ nhất : 0 0 0

                                    tổ hợp thứ hai : 1 0 0

                                    tổ hợp thứ ba : 1 1 0

                                    tổ hợp thứ tư : 1 1 1

                                    tổ hợp thứ năm : 0 1 1

                                    tổ hợp thứ sáu : 0 0 1

n=4                            0000

                                    1000

                                    1100

                                    1110

                                    1111

                                    0111

                                    0011

                                    0001

.....

### e. Mã vòng:

Mã vòng có nguyên tắc lập mã:

+ Dùng n bit nhị phân thì mã hoá được n trạng thái

- + 2 từ mã kề nhau luôn khác nhau 2 bit
- + Trong một từ mã chỉ có duy nhất một bit là 1 và được dịch từ bit trẻ nhất đến bit già nhất, các bit khác là 0.

VD: n=3            1 0 0  
                           0 1 0  
                           0 0 1

n =4                1 0 0 0  
                           0 1 0 0  
                           0 0 1 0  
                           0 0 0 1

.....

### III. CÁC BƯỚC THIẾT KẾ BỘ ĐẾM

Cũng giống như các bước thiết kế mạch dãy, gồm có 4 bước cơ bản như sau:

- Bước 1: Từ bài toán đã cho, xác định tín hiệu vào đếm, hệ số đếm  $K_d$ , từ đó vẽ đồ hình trạng thái của bộ đếm.
- Bước 2: Mã hoá các trạng thái trong và Xác định số FF(n) cần thiết của bộ đếm, theo điều kiện sau:

+ Mã nhị phân huặc mã Gray:  $n \geq \log_2 K_d$     n lấy cận trên, nguyên

Ví dụ ,  $K_d=8 \Rightarrow n \geq \log_2 8 = 3$ , chọn n=3

$K_d=10 \Rightarrow n \geq \log_2 10 \approx 3,4$  ; chọn n=4

+ Mã Johnson:  $n = K_d/2$

+ Mã Vòng:     $n=K_d$

Số FF cũng chính là số bit cần thiết để lập mã, mã hoá các trạng thái.

- Bước 3: Xác định hàm kích cho các FF, và xác định hàm ra (đã nói ở phần định nghĩa bộ đếm).

Tùy loại FF sử dụng mà có hàm vào kích khác nhau, có thể thực hiện dựa theo

2 yếu tố: + Dựa vào các bảng chuyển đổi trạng thái, bảng ra để xác định .

+ Dựa trực tiếp vào đồ hình trạng thái.

Các thứ nhất trực quan dễ thực hiện, nên được sử dụng nhiều hơn.

- Bước 4: Sơ đồ mạch thực hiện, Căn cứ vào 2 phương trình, để vẽ mạch nguyên lý:

+ Phương trình đầu vào kích

+ Phương trình tín hiệu ra

### IV. MỘT SỐ DẠNG BỘ ĐẾM THƯỜNG GẶP

#### 1. Bộ đếm thuận, đồng bộ, hệ số đếm $K_d = 2^n$ , dùng mã nhị phân.

Họ bộ đếm này chỉ khác nhau hệ số  $K_d$ , sau đây ta xét một số ví dụ

a,  $K_d = 4$

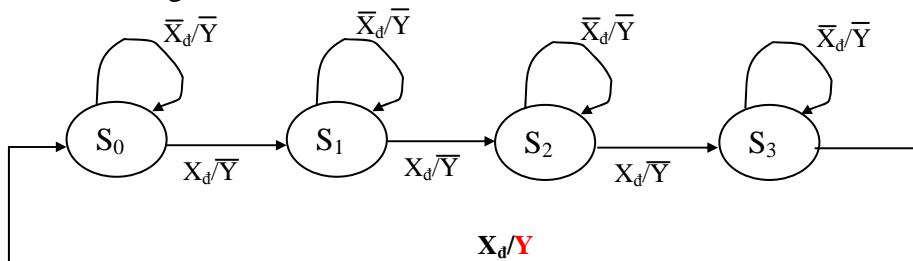
- Bước 1: Từ bài toán đã cho, xác định tín hiệu vào đếm, hệ số đếm  $K_d$ , từ đó vẽ đồ hình trạng thái của bộ đếm:

Giả sử: + Ký hiệu tín hiệu vào đếm là  $X_d$ : tích cực(1);  $\overline{X_d}$  không tích cực(0)

+ Ký hiệu tín hiệu ra đếm là  $Y$ : tích cực(1);  $\overline{Y}$  không tích cực(0)

$K_d=4$ , nên số trạng thái là 4, ký hiệu  $S_0, S_1, S_2, S_3$

Ta có đồ hình trạng thái như sau:

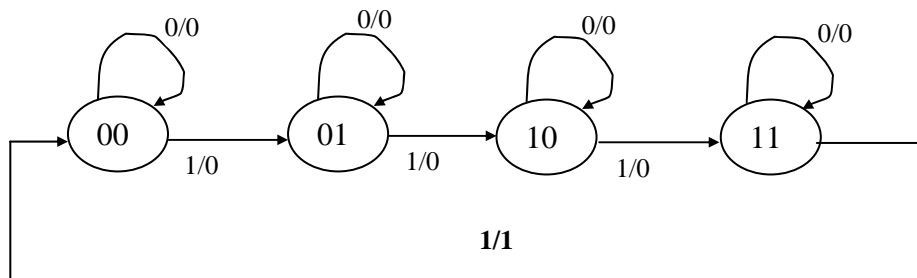


- Bước 2: Mã hoá các trạng thái trong và Xác định số FF(n) cần thiết của bộ đếm.  
 Số FF:  $n \geq \log_2 4 = 2$ . Đầu bài không yêu cầu loại FF, giả sử ta chọn JK-FF, ký hiệu lần lượt là A và B.

Ta có các tổ hợp mã hoá :

A B	Trạng thái
0 0	S <sub>0</sub>
0 1	S <sub>1</sub>
1 0	S <sub>2</sub>
1 1	S <sub>3</sub>

=> Dạng Automat nhị phân:



Nhìn đồ hình này ta thấy tất cả các thông số, ký hiệu đã được “số hoá”

- Bước 3: Xác định hàm kích cho các FF, và xác định hàm ra

Giả sử tại thời điểm t bộ đếm có tập các trạng thái S(i,j), tại thời điểm(t+1) thời điểm có xung đếm xuất hiện bộ đếm chuyển sang trạng thái S'(i,j), căn cứ vào 2 tập trạng thái này ta xác định được đầu vào kích cho hai FF A và B, theo bảng sau(xem phần đầu vào kích cho FF):

t		t+1		FF -A		FF -B	
A	B	A'	B'	J <sub>A</sub>	K <sub>A</sub>	J <sub>B</sub>	K <sub>B</sub>
0	0	0	1	0	x	1	x
0	1	1	0	1	x	x	1
1	0	1	1	x	0	1	x
1	1	0	0	x	1	x	1

Q->Q'	J K
0->0	0 x
0->1	1 x
1->0	x 1
1->1	x 0

Tiếp theo ta tiến hành tối thiểu hoá các hàm J và K

$$J_A$$

	A	0	1
B			
0			x
1		1	x

$$J_A=B$$

$$K_A$$

	A	0	1
B			
0		x	
1		x	1

$$K_A=B$$

$$J_B$$

	A	0	1
B			
0		1	1
1		x	x

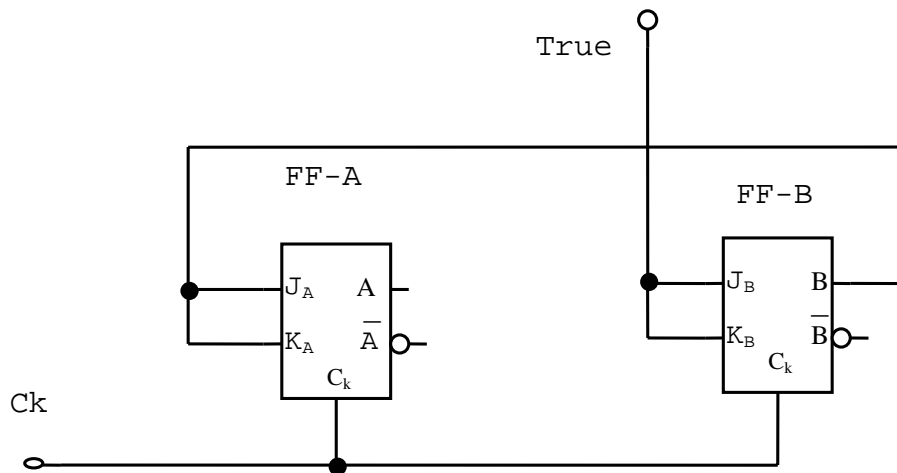
$$J_B=1$$

$$K_B$$

	A	0	1
B			
0		x	x
1		1	1

$$K_B=1$$

- Bước 4: Sơ đồ mạch thực hiện



a,  $K_d = 6$

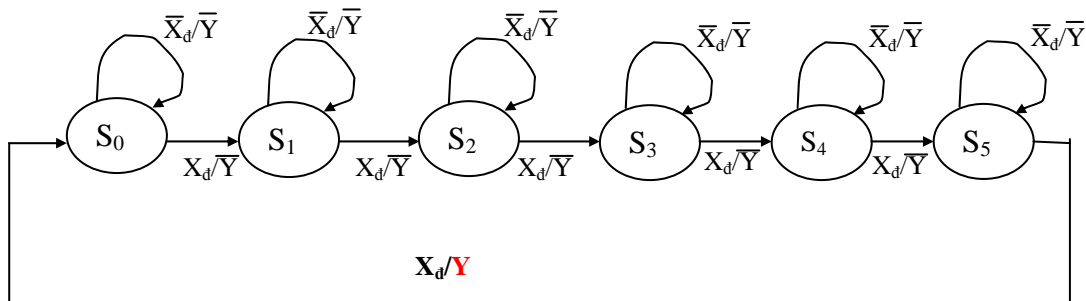
- Bước 1: Từ bài toán đã cho, xác định tín hiệu vào đếm, hệ số đếm  $K_d$ , từ đó vẽ đồ hình trạng thái của bộ đếm:

Giả sử: + Ký hiệu tín hiệu vào đếm là  $X_d$ : tích cực(1);  $\overline{X_d}$  không tích cực(0)

+ Ký hiệu tín hiệu ra đếm là  $Y$ : tích cực(1);  $\overline{Y}$  không tích cực(0)

$K_d=6$ , nên số trạng thái là 6, ký hiệu  $S_0, S_1, S_2, S_3, S_4, S_5$

Ta có đồ hình trạng thái như sau:



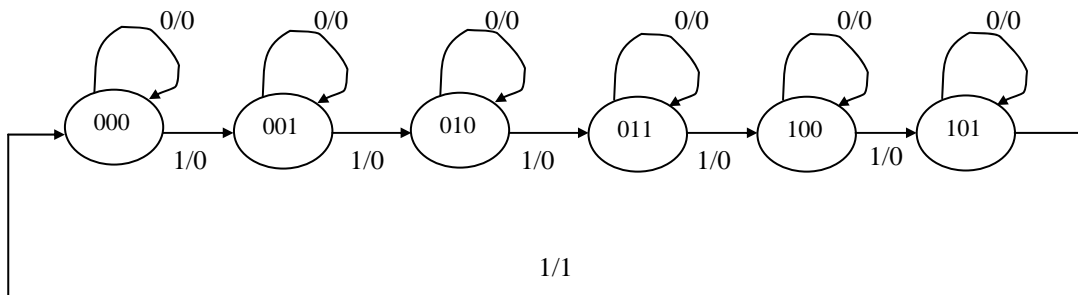
- Bước 2: Mã hoá các trạng thái trong và Xác định số FF(n) cần thiết của bộ đếm.

Số FF:  $n \geq \log_2 6$ , chọn  $n=3$ . Đầu bài không yêu cầu loại FF, giả sử ta chọn JK-FF, ký hiệu lần lượt là A, B, C. Như vậy sẽ có 8 tổ hợp trong khi chỉ cần 6 tổ hợp để mã hoá 6 trạng thái, giả sử không dùng 2 tổ hợp 110, 111

Ta có các tổ hợp mã hoá :

A BC	Trạng thái
0 0 0	$S_0$
0 0 1	$S_1$
0 1 0	$S_2$
0 1 1	$S_3$
1 0 0	$S_4$
1 0 1	$S_5$

=> Dạng Automat nhị phân:



- Bước 3: Xác định hàm kích cho các FF, và xác định hàm ra

Giả sử tại thời điểm  $t$  bộ đếm có tập các trạng thái  $S(i,j)$ , tại thời điểm  $(t+1)$  thời điểm có xung đếm xuất hiện bộ đếm chuyển sang trạng thái  $S'(i,j)$ , căn cứ vào 2 tập trạng thái này ta xác định được đầu vào kích cho ba FF A, B, C theo bảng sau (xem phần đầu vào kích cho FF):

t			t+1			FF -A		FF -B		FF -C	
A	B	C	A'	B'	C'	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	0	0	0	x	1	0	x	x	1

Tối thiểu hoá các hàm J và K

$J_A$ \ AB	00	01	11	10
C \ 0			x	x
C \ 1		1	x	x

$$J_A = \overline{BC}$$

$K_A$ \ AB	00	01	11	10
C \ 0	x	x	x	
C \ 1	x	x	x	1

$$K_A = C$$

	AB	00	01	11	10
C					
0			x	x	
1	1	x	x		

$$J_B = \bar{A}C$$

	AB	00	01	11	10
C					
0		x		x	x
1	x	1	x	x	

$$K_B = K_A = C$$

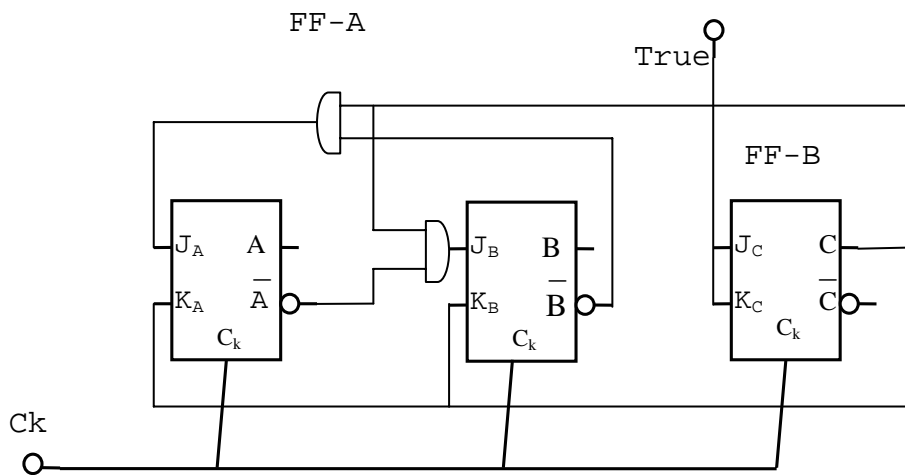
	AB	00	01	11	10
C					
0	1	1	x	1	
1	x	x	x	x	

$$J_C = 1$$

	AB	00	01	11	10
C					
0	x	x	x	x	
1	1	1	x	1	

$$K_C = 1$$

- Bước 4: Sơ đồ mạch thực hiện

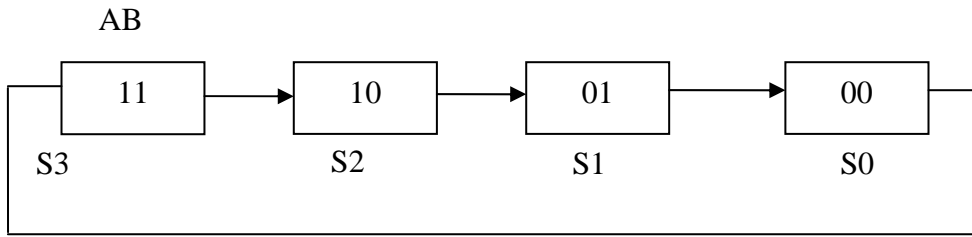


Bài tập: thiết kế với Kđ=8, 10, 12. Dùng JK-FF; D-FF; T-FF; RS-FF

## 2. Các bộ đếm nghịch, nhị phân, đồng bộ với hệ số đếm chẵn

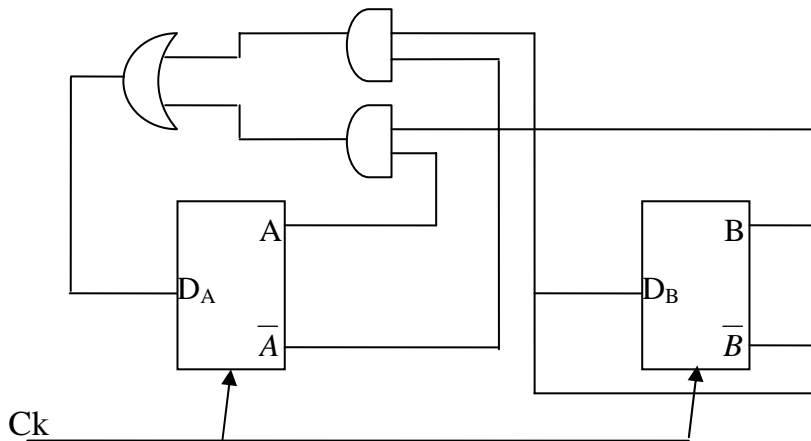
Phương pháp thiết kế giống như đếm thuận, chỉ khác ở đồ hình trạng thái: các trạng thái sẽ được chuyển từ 'cao' xuống 'thấp'

Ví dụ Kđ= 4



AB	A'B'	D <sub>A</sub>	D <sub>B</sub>
11	10	1	0
10	01	0	1
01	00	0	0
00	11	1	1

$$\Rightarrow D_A = \overline{A}B + AB \ ; \ D_B = \overline{B}$$



Bộ đếm Kđ=4

Sinh viên tự làm các phần sau:

- Bộ đếm mã Gray, đồng bộ Kđ=10
- Bộ đếm mã NBCD, đồng bộ Kđ=10, đếm thuận
- Bộ đếm mã NBCD, đồng bộ Kđ=10, đếm nghịch
- Bộ đếm mã JohnSon, đồng bộ Kđ=10, đếm nghịch
- Bộ đếm mã Vòng, đồng bộ Kđ=10, đếm nghịch

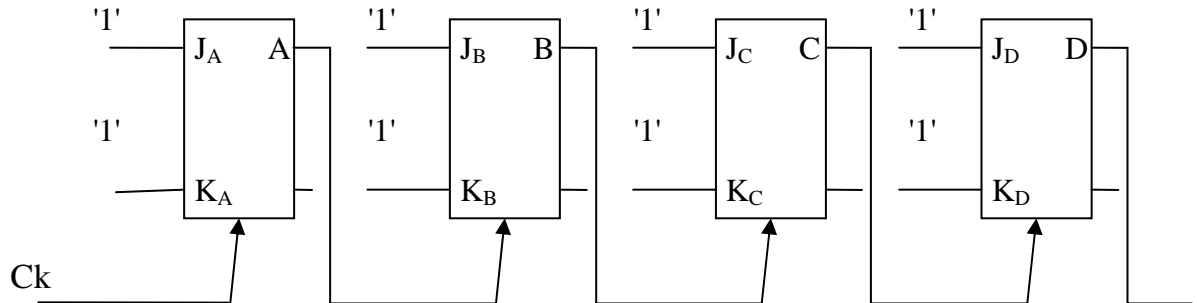
### 3. Các bộ đếm nhị phân, đồng bộ với hệ số đếm chẵn

Các bộ đếm kiểu này có cấu tạo rất đơn giản, có đặc điểm:

- Chỉ dùng một loại T-FF hoặc JK-FF, các đầu vào T và JK luôn nối với mức logic cao
- 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, khi đếm nghịch lấy Q'.



- Tín hiệu vào Xđ luôn được đưa vào đầu vào xung nhịp của FF có trọng số nhỏ nhất.



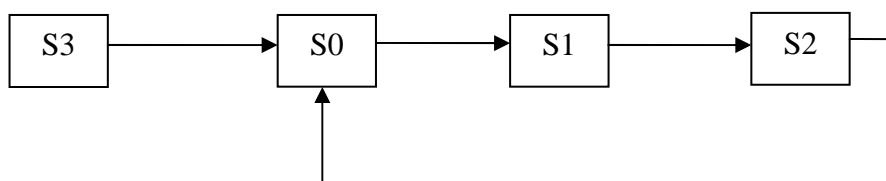
Bộ đếm thuận, không đồng bộ, dùng JK-FF

#### 4. Thiết kế bộ đếm có mạch khởi động

Đây là loại bộ đếm mà mỗi khi khởi động sẽ luôn có một trạng thái cho trước (ví dụ như các trò chơi để tăng thêm phần hấp dẫn, người ta luôn cho mỗi đội chơi một lượng điểm ban đầu nào đó).

Cách thiết kế cũng giống với các mạch đã xét, chỉ có quá trình chuyển trạng thái là khác một chút:

Minh họa: một bộ đếm khởi động luôn ở S3, trong quá trình hoạt động trạng thái sẽ biến đổi trong phạm vi từ S0 đến S2:



S	AB	AB
S0	00	01
S1	01	10
S2	10	00
S3	11	00

Từ bảng trạng thái, ta thấy trạng thái S3(11) sẽ chuyển về S0(00)

#### **V. LƯU Ý KHI THIẾT KẾ BỘ ĐẾM:**

Bộ đếm cũng là một mạch dãy, nên cũng có thể rơi vào trạng thái khoá hay trạng thái chu kỳ, cho nên trong quá trình thiết kế cần hạn chế khả năng này, chi tiết phương pháp xem ở chương Mạch dãy đồng bộ và chương Mạch dãy không đồng bộ.

#### **VI. MỘT SỐ IC ĐẾM THƯỜNG GẶP:**

- IC 7490
- IC 74L90
- IC 74LS90
- IC 74LS190
- IC 74LS192

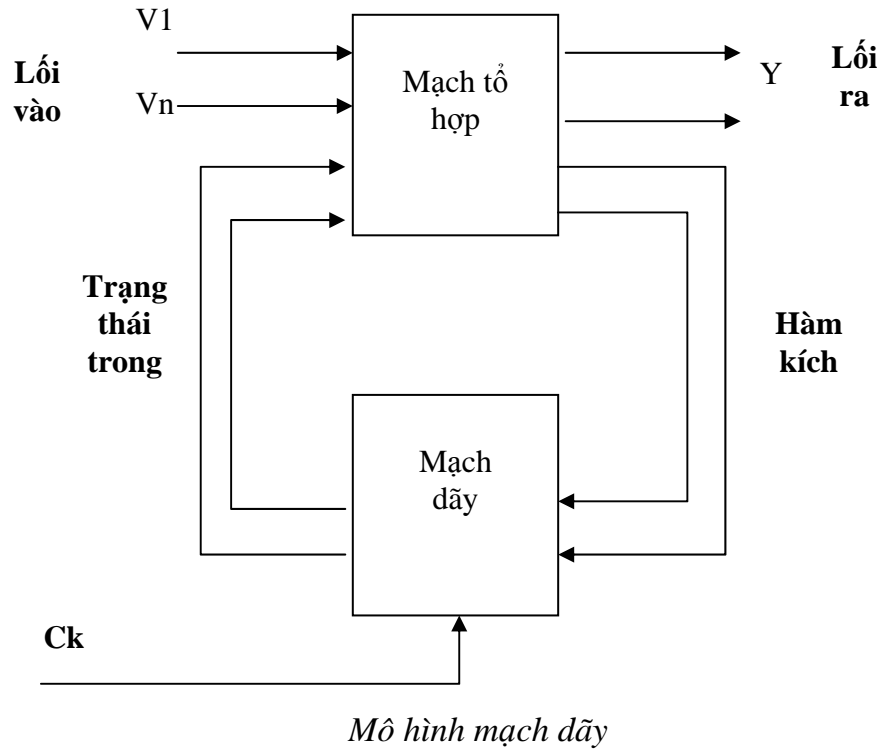
Ngoài chức năng đếm, các IC này còn có thể dùng để chia tần số theo nguyên tắc:  $n$  xung vào cho  $m$  xung ra, thì tần số sẽ giảm còn  $n/m$  lần.

Sơ đồ chân và ứng dụng cụ thể xem ở các tài liệu hướng dẫn sử dụng và tra chân IC

# CHƯƠNG 9 MÔ TẢ VÀ THIẾT KẾ MẠCH DẪY

## I. KHÁI NIỆM CƠ BẢN.

### 1. Mô hình tổng quát.



### 2 Phương pháp mô tả mạch dãy.

#### a. Mô hình toán học.

Ta có thể dùng một hệ phương trình toán học để biểu thị mối quan hệ vào ra của hệ tuần tự. Đối với mô hình tổng quát hình 6.1, nếu gọi:

V: là tập tín hiệu vào,

R: tập tín hiệu ra.

X :là tập hàm kích thích.

S: tập các trạng thái trong

thì hệ có thể được mô tả bởi các phương trình sau:

a. Otomat (hệ phương trình) Mealy:

$$R = f_1(V,S). \quad (1)$$

$$S' = f_2(X,S). \quad (2)$$

$$X = f_3(V,S). \quad (3)$$

Phương trình (1) được gọi là hàm ra, thể hiện mối quan hệ giữa đầu ra với tác động vào và biến trạng thái.

Phương trình (2) là hàm chuyển đổi trạng thái của các phần tử nhớ.

Phương trình (3) là hàm kích thích cho các phần tử nhớ.

Hệ phương trình trên được gọi là Otomat Mealy, trong khi nếu hàm ra chỉ phụ thuộc biến trạng thái mà không phụ thuộc biến vào thì hệ tuần tự có tên là Otomat Moore.

b. Otomat (hệ phương trình) Moore.

$$R = f_1(S),$$

$$S' = f_2(X,S),$$

$$X = f_3(V,S).$$

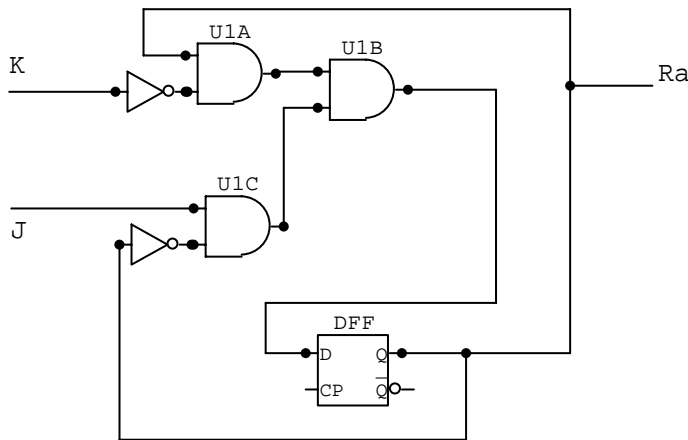
**b. Bảng trạng thái.**

Trong phương pháp này, ta thiết lập bảng để liệt kê mối quan hệ giữa R, trạng thái trong S' với tín hiệu vào V và trạng thái S. Bảng thu được có tên gọi là bảng chuyển đổi trạng thái. Nếu như đã biết loại FF, từ bảng chuyển đổi trạng thái, ta có thể suy ra hàm chuyển đổi trạng thái và hàm kích thích.

<b>V</b>	<b>S → S'</b>	<b>R</b>
V0	S0 → S'0	R0
V1	S1 → S'1	R1
V2	S2 → S'2	R2
V3	S3 → S'3	R3
...	...	...
Vn	Sn-1 → S'n-1	0

Bảng chuyển đổi trạng thái.

Minh họa: xét mạch chuyển đổi từ FF D sang FF JK như sau:



Mạch chuyển đổi FF D thành FF JK

Ta có thể biểu diễn bảng chuyển đổi trạng thái như sau:

<b>V JK</b>	<b>S → S'</b>	<b>R</b>
00	0 → 0	0
01	1 → 1	1
10	0 → 0	0

0 1	1 → 0	0
1 0	0 → 1	1
1 0	1 → 1	1
1 1	0 → 1	1
1 1	1 → 0	0

Nếu đặt :

00: V0 ;

01: V1;

10 : V2;

11: V3

và 0: S0; 1: S1,

ta sẽ có:

V	S → S'	R
V0	S0 → S0	S0
V0	S1 → S1	S1
V1	S0 → S0	S0
V1	S1 → S0	S0
V2	S0 → S1	S1
V2	S1 → S1	S1
V3	S0 → S1	S1
V3	S1 → S0	S0

c. Biểu diễn bằng đồ hình trạng thái.

Đồ hình trạng thái là một đồ hình có hướng gồm 2 tập:

M : tập đỉnh và K: tập các cung có hướng.

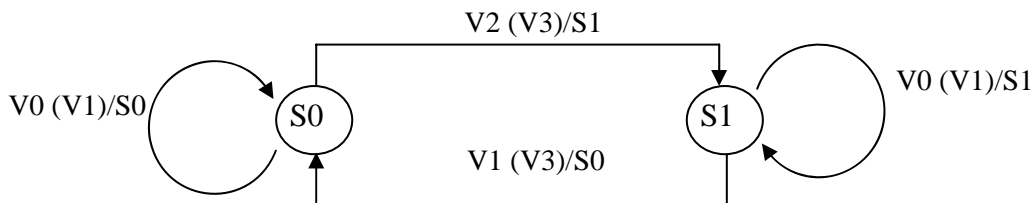
a. Mô hình Mealy

Tập đỉnh M là tập các trạng thái trong.

Tập các cung K là tập tín hiệu vào/ra.

Trên cung đi từ Si → Sj ghi tín hiệu vào/ra tương ứng.

Đối với minh hoạ chuyển đổi FF trên, ta sẽ xây dựng được đồ hình trạng thái Mealy như hình vẽ.



Mô hình Mealy.

b. Mô hình Moore

Vì tín hiệu ra chỉ phụ thuộc vào trạng thái trong của mạch mà không phụ thuộc tín hiệu vào cho nên :

Tập đỉnh M là tập trạng thái trong/tín hiệu ra.

Tập cung K là tập các tín hiệu vào.

## II. CÁC BƯỚC THIẾT KẾ MẠCH DÂY.

### - Bài toán chưa được hình thức hoá.

Nhiệm vụ thiết kế được mô tả bằng ngôn ngữ hoặc bằng lưu đồ thuật toán.. Nói chung là chưa được hình thức hoá.

### - Hình thức hoá

Phiên dịch các dữ kiện đó thành 1 hình thức mô tả hoạt động của mạch bằng cách hình thức hoá dữ liệu ban đầu ở dạng bảng trạng thái hay đồ hình trạng thái.

Rút gọn các trạng thái trong của mạch để nhận được số trạng thái trong là 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.

### - Otomat nhị phân.

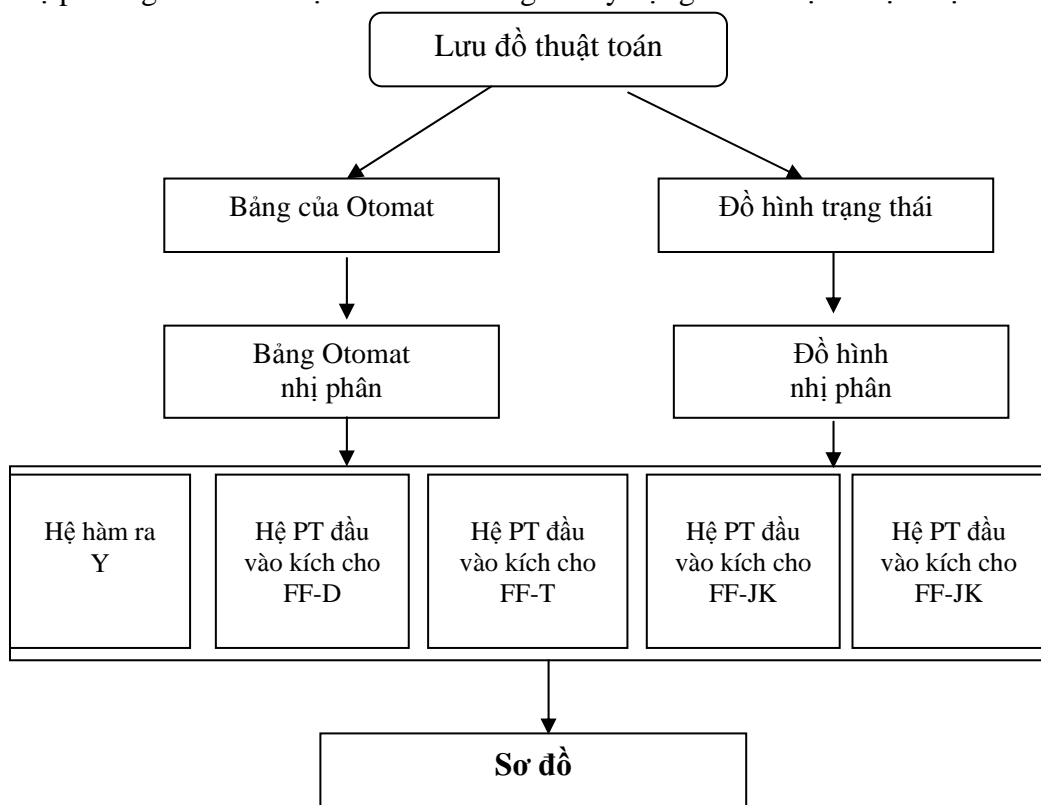
Mã hoá tín hiệu vào, ra, trạng thái trong để nhận được otomat nhị phân có X là tập tín hiệu vào, R là tập tín hiệu ra, Q là tập trạng thái trong.

### - Hệ hàm của mạch:

Xác định hệ phương trình logic của mạch và tối thiểu hoá các phương trình này.

### - Xây dựng sơ đồ mạch thực hiện.

Từ hệ phương trình của mạch đã viết chúng ta xây dựng sơ đồ mạch thực hiện.



Các bước thiết kế mạch dây.

### 1. Thiết kế mạch dây từ bảng trạng thái của Otomat.

#### Các bước:

1. Xác định sơ đồ khối chung.
2. Mã hoá tín hiệu vào V, ra R, trạng thái trong S và Q.
3. Lập bảng chuyển đổi trạng thái.
4. Xác định đầu vào kích thích cho các FF, viết hệ hàm kích thích và hàm ra.
5. Xây dựng sơ đồ mạch.

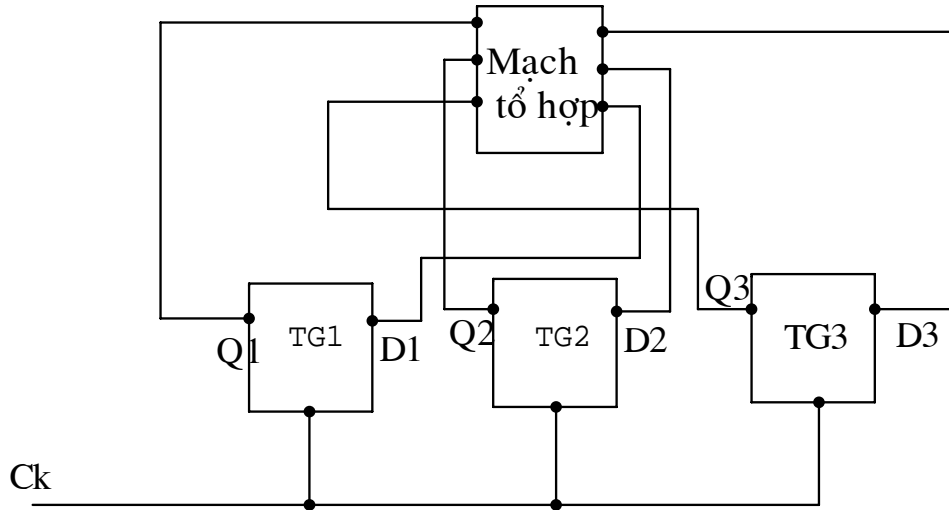
**Để hiểu rõ** thêm phương pháp thiết kế mạch dãy dùng bảng trạng thái ta sẽ xét bài toán minh họa sau:

Thiết kế mạch đếm đồng bộ thuận,  $K_d=5$  (đếm từ 0-4) dùng FF D theo phương pháp bảng trạng thái.

Quá trình giải như sau:

1. Xây dựng sơ đồ khối chung.

Để xây dựng mạch đếm,  $K_d=5$  ta dùng 3 FF. Hình vẽ sơ đồ khối chung được thể hiện như sau:



Sơ đồ khối chung của mạch đếm  $K_d=5$

Theo quy định, FF1 có trọng số nhỏ nhất. FF3 có trọng số lớn nhất.

2. Mã hoá.

S	S'
S0	S1
S1	S2
S2	S3
S3	S4
S4	S0

Bảng trạng thái trong của mạch

3. Xây dựng bảng chuyển đổi trạng thái.

	Q3	Q2	Q1	Q'3	Q'2	Q'1	D3	D2	D1
S0	0	0	0	0	0	1	0	0	1
S1	0	0	1	0	1	0	0	1	0
S2	0	1	0	0	1	1	0	1	1
S3	0	1	1	1	0	0	1	0	0
S4	1	0	0	0	0	0	0	0	0

4 Xác định đầu vào kích cho FF và hệ hàm kích.

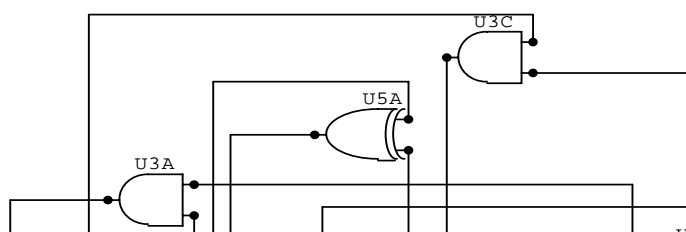
Sử dụng phương pháp tối thiểu hoá dùng bảng Karnaugh, ta sẽ tính được:

$$D1 = \overline{Q1} \cdot Q3$$

$$D2 = Q2 \oplus Q1 \quad D3 = Q1 \cdot Q2$$

6. Vẽ sơ đồ mạch.

Sau khi xây dựng được hệ hàm kích, ta sẽ vẽ mạch.



## 2. Thiết kế mạch dãy từ đồ hình trạng thái.

Gồm 5 bước thiết kế:

1. Xây dựng sơ đồ khối chung, đồ hình trạng thái.
2. Mã hoá V, R, S, Y, Q.
3. Xác định hệ phương trình tín hiệu ra  
 $Y = f(X, Q)$
4. Xác định hệ phương trình hàm kích cho các FF.  
 $X = f(V, Q)$ .

Dựa vào bảng hàm kích thích, xây dựng được thuật toán xác định phương trình đầu vào kích cho các FF.

Quy ước cung biểu diễn sự thay đổi  $Q_i \rightarrow Q_i'$  như sau:

$0 \rightarrow 0$  là cung loại 0.

$1 \rightarrow 1$  là cung loại 1.

$0 \rightarrow 1$  là cung loại 2.

$1 \rightarrow 0$  : cung loại 3.

a. Với FF D.

Do  $D_i = Q_i' =$  tuyến các cung đi tới đỉnh có  $Q_i=1$   
 $= \sum (1+2)$

b. Với FF T:

$T_i = \sum$  các cung có  $Q_i$  thay đổi  
 $= \sum (2+3)$

c. Loại FF JK.

+ Gọi  $T_{on}$  là  $\sum$  các cung mà  $Q_i$  bật =  $\sum(2)$

Sau đó thực hiện tối thiểu hoá

Nếu phương trình của  $T_{on}$  còn  $Q_i$ , tức là có dạng:

$$T_{on} = T * \overline{Q_i}$$

thì  $J = T^*$ .



Nếu phương trình của Ton không còn Qi thì :  $J=Ton$ .  
 + Gọi  $Toff = \sum \text{cung mà Qi tắt} = \sum(3)$ ; Sau đó thực hiện tối thiểu hoá.  
 Nếu phương trình của Toff còn Qi tức có dạng:  
 $Toff = (T^{**}) \cdot Qi$ .

thì  $K = T^{**}$ .

Nếu phương trình của Toff không còn Qi thì  $K = Toff$ .

d. Loại FF RS.

$S = Ton + [\text{cung loại (1)}]$ .

$R = Toff + [\text{cung loại (0)}]$ .

Chú ý : cung loại (1) và cung loại (0) trong dấu [ ] được lấy giá trị không xác định ;  
 Những giá trị này cùng với những trạng thái không sử dụng được dùng để tối thiểu hoá.

5. Vẽ sơ đồ mạch.

**Chú ý:** Ngoài ra còn có thể thiết kế mạch dây từ Lưu đồ Thuật toán, bằng cách chuyển về Đồ hình Mealy hoặc Đồ hình Moore

## CHƯƠNG 10 THANH GHI DỊCH

### I, KHÁI QUÁT

#### 1. Định nghĩa:

Thanh ghi dịch là một mạch dây, có khả năng ghi giữ và dịch bit thông tin (dịch phải hoặc trái)

Thanh ghi dịch được dùng: + Để nhớ số liệu

+ Để chuyển số liệu song song thành nối tiếp, và

ngược lại

+ Thiết kế bộ đếm, tạo dãy tín hiệu nhị phân tuần

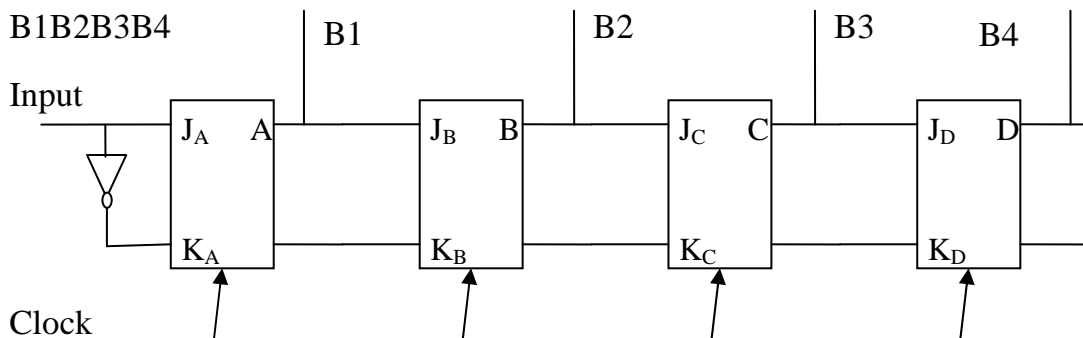
hoàn theo yêu cầu cho trước

...

#### 2. Cấu tạo

Thanh ghi dịch gồm dãy các phần tử nhớ đơn bit (FF) được mắc liên tiếp và đóng trong cùng một vỏ.

Các FF sử dụng trong thanh ghi dịch thường là D-FF hoặc FF mắc theo kiểu D-FF, số FF chính là số bit mà thanh ghi dịch lưu trữ được.



Thanh ghi dịch 4 bit dùng JK-FF

Thông tin được nạp vào thanh ghi dịch theo nguyên tắc từng bit đồng bộ với xung nhịp, bit đầu tiên được nạp vào FF đầu tiên, các bit thông tin đã được lưu trữ dịch phải 1 bit (bit lưu trong FF-A chuyển sang FF-B, FF-B chuyển sang FF-C...)

### 3. Phân loại

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
- Vào song, ra song song: thông tin được đưa vào và lấy ra đồng thời
- Vào nối tiếp, ra nối tiếp: thông tin được đưa vào và lấy ra tuần tự từng bit một
- Vào song song, ra nối tiếp: thông tin được đưa vào thanh ghi dịch đồng thời, số liệu được đưa ra tuần tự từng bit một.

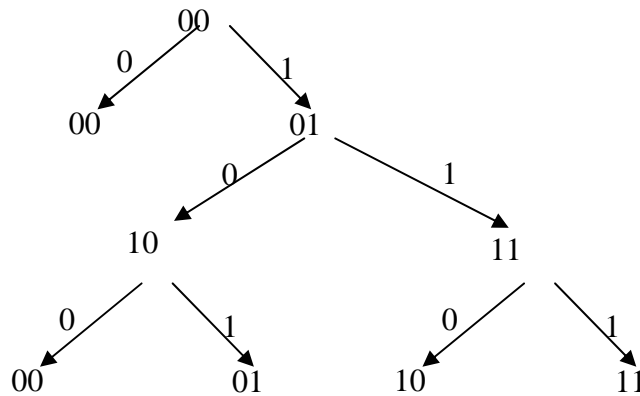
b, Phân loại theo đầu vào:

- Đầu ra đơn: mỗi FF trong thanh ghi dịch chỉ có một đầu ra Qi (hoặc đầu đảo), được đưa ra chân của vi mạch.
- Đầu ra đơn: Cả hai đầu ra của FF được đưa ra chân của vi mạch

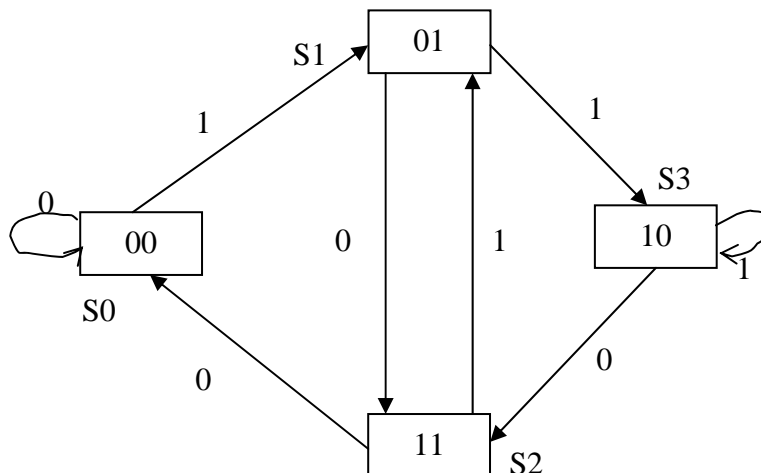
## II. ĐỒ HÌNH TỔNG QUÁT

Đồ hình tổng quát của thanh ghi dịch theo mô hình De Bruijn như sau:

- Thanh ghi dịch 2 bit:

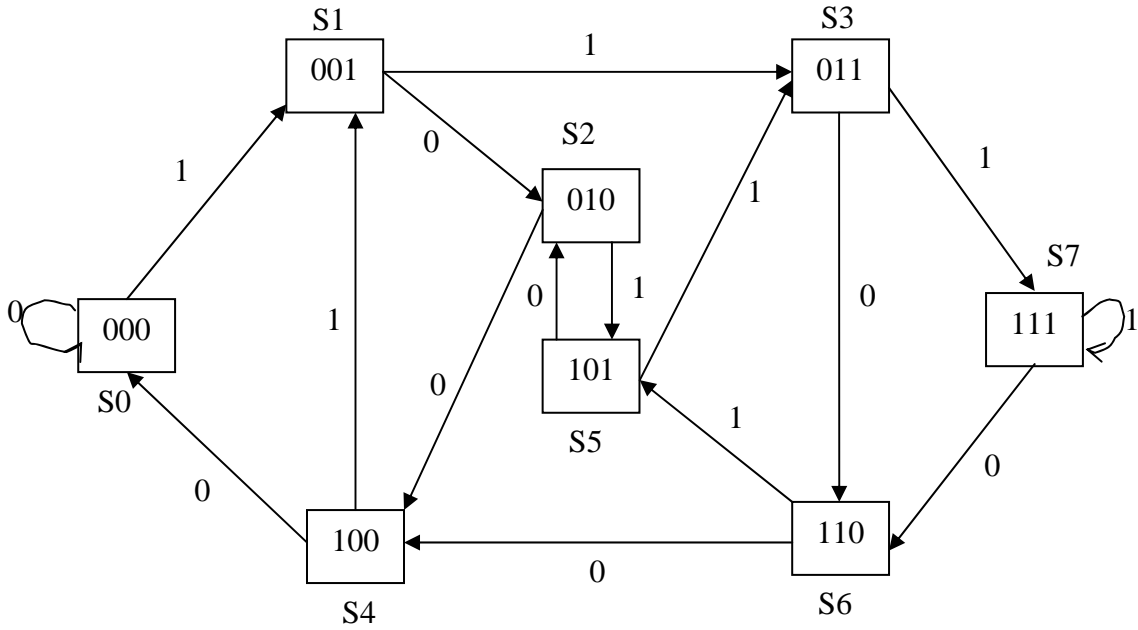


Bảng chuyển đổi trạng thái



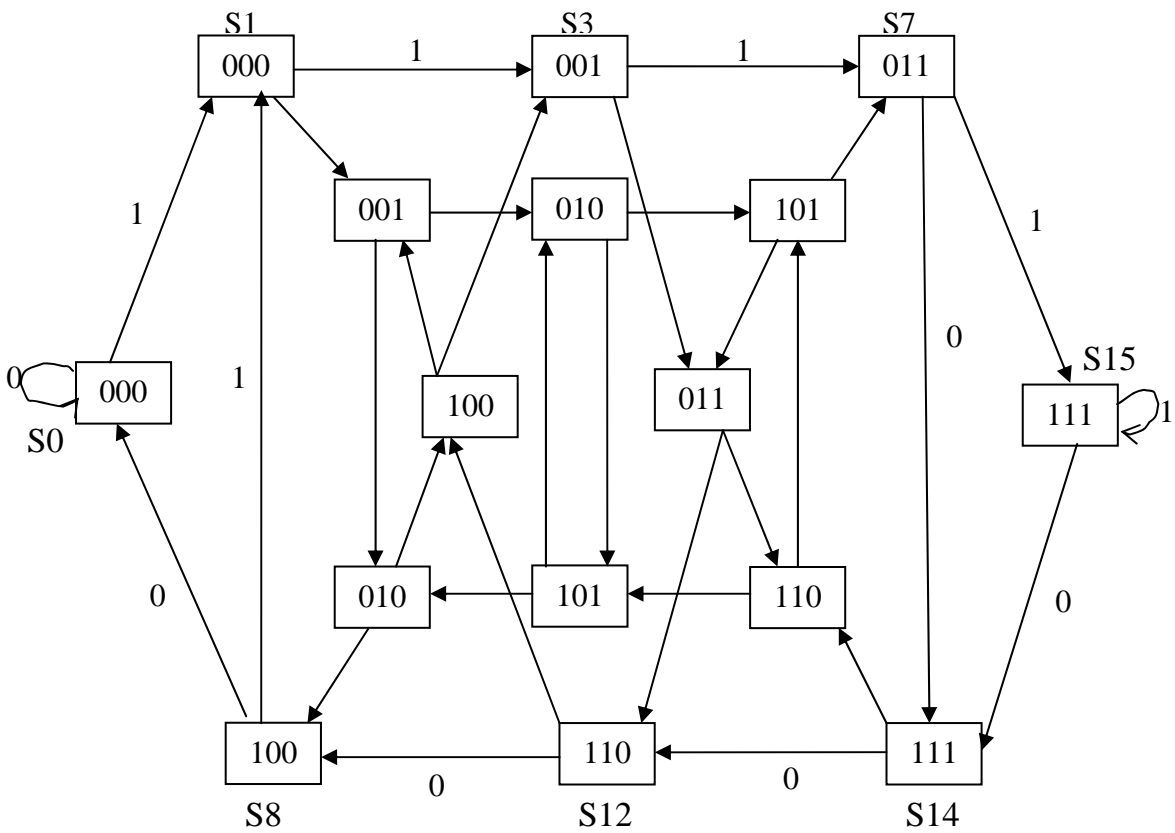
Từ đồ hình ta nhận thấy thanh ghi dịch sẽ bị khoá ở trạng thái 00 nếu hàm hồi tiếp là 0 và khoá ở trạng thái 11 nếu hàm hồi tiếp là 1.

- Thanh ghi dịch 3 bit:



Đồ hình tổng quát thanh ghi dịch 3 bit

- Thanh ghi dịch 4 bit:



Đồ hình tổng quát thanh ghi dịch 4 bit

### III. THIẾT KẾ BỘ ĐẾM DỪNG THANH GHI DỊCH

#### 1. Sơ đồ khối:

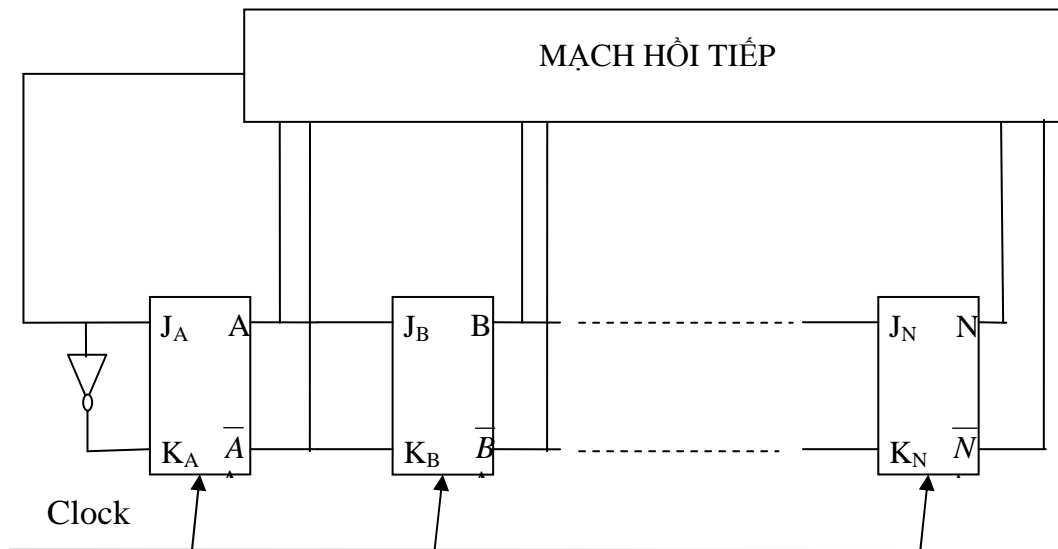
Từ đồ hình trạng thái của thanh ghi dịch ta thấy rằng: Xuất phát từ một trạng thái ban đầu bất kỳ, ứng với dãy tín hiệu hồi tiếp  $f_{ht}=D_A=J_A$  xác định, sẽ có dãy xác định các trạng thái sẽ chuyển tới, nếu dãy tín hiệu của  $f_{ht}$  được chọn thích hợp sao cho dãy chuyển biến trạng thái tạo thành một chu trình kín thì đồ hình trạng thái của mạch chính là đồ hình trạng thái của bộ đếm.

Như vậy bài toán thiết kế bộ đếm dùng thanh ghi dịch chuyển 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 của bộ đếm dùng thanh ghi dịch như hình dưới đây, các FF được mắc với nhau thành thanh ghi dịch n bit, đầu ra của các FF gồm cả Q và  $\bar{Q}$  được dùng để tạo hàm hồi tiếp, đưa tới điều khiển đầu vào của thành ghi. Giữa đầu vào và đầu ra của FF có mối quan hệ:

$$A'=f_{ht}(A, B, C, D, \dots, N)$$

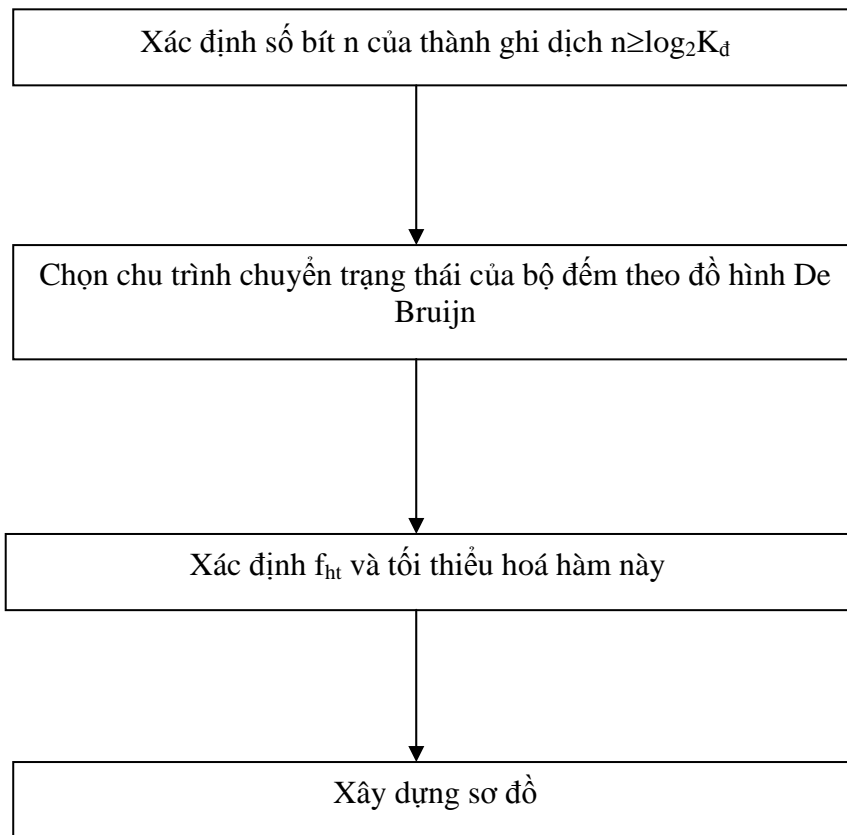
$$B'=A; C'=B; \dots; N'=M$$

Mạch hồi tiếp có giá trị 0 hay 1 và được đưa vào đầu vào của FF-A, khi có xung nhịp sẽ thiết lập trạng thái của FF- A tương ứng.



Bộ đếm dùng thanh ghi dịch

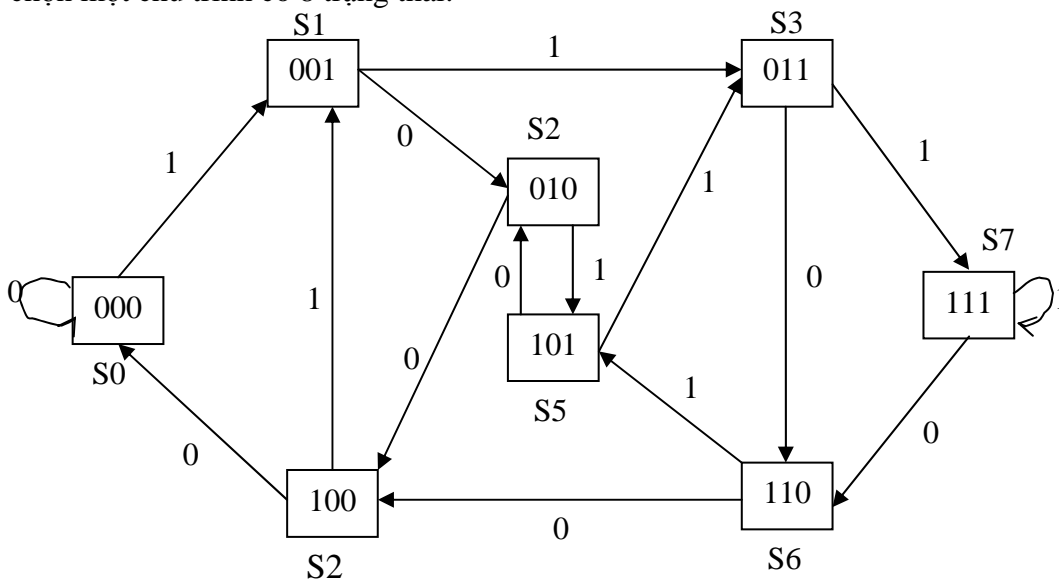
## 2. Các bước thiết kế



Ví dụ: Thiết kế bộ đếm  $K_d=8$

- Bước 1, Xác định số bit  $n=\log_2 8=3$

- Bước 2: Chọn chu trình chuyển trạng thái, căn cứ vào đồ hình trạng thái của thanh ghi dịch chọn một chu trình có 8 trạng thái:



Giả sử chọn: S0->S1->S3->S7->S6->S5-> S2->S4->S0

- Bước 3: Xác định hàm hồi tiếp, ký hiệu các FF lần lượt là CBA, ta có bảng mã hoá trạng thái và hàm hồi tiếp như sau:

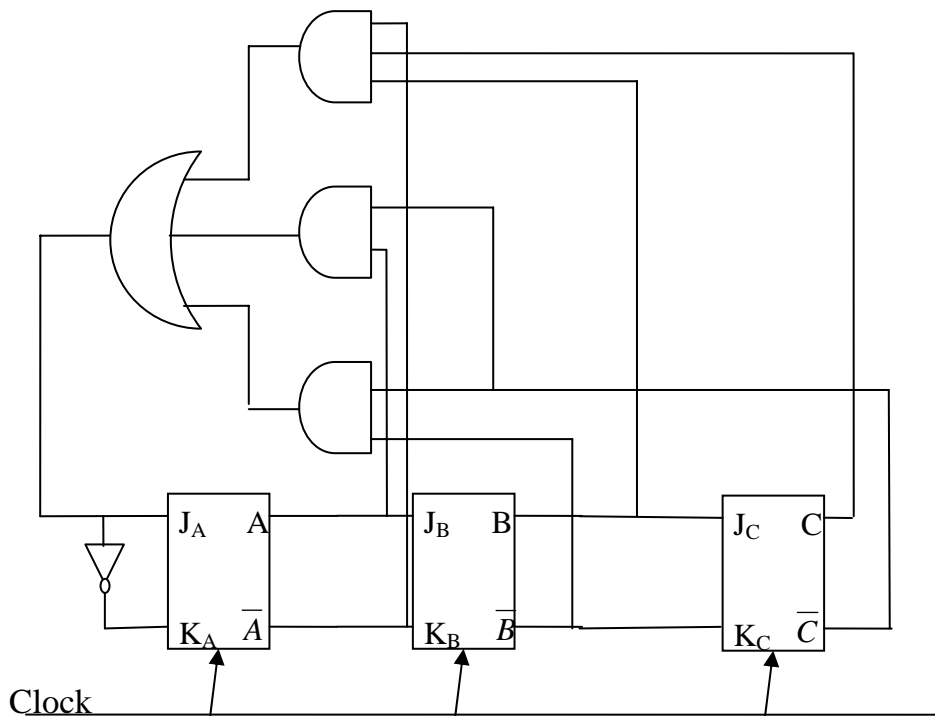
S	C	B	A	$f_{ht}$
S0	0	0	0	1
S1	0	0	1	1
S3	0	1	1	1
S7	1	1	1	0
S6	1	1	0	1
S5	1	0	1	0
S2	0	1	0	0
S4	1	0	0	0

Tối thiểu hoá  $f_{ht}$ :

C \ BA	00	01	11	10
0	1	1	1	
1				1

$$f_{ht} = \overline{C}\overline{B} + \overline{C}A + C\overline{B}\overline{A}$$

- Bước 4: Xây dựng sơ đồ:



**Bài tập:** thiết kế tương tự ứng với  $K_d=10, 12, 14$

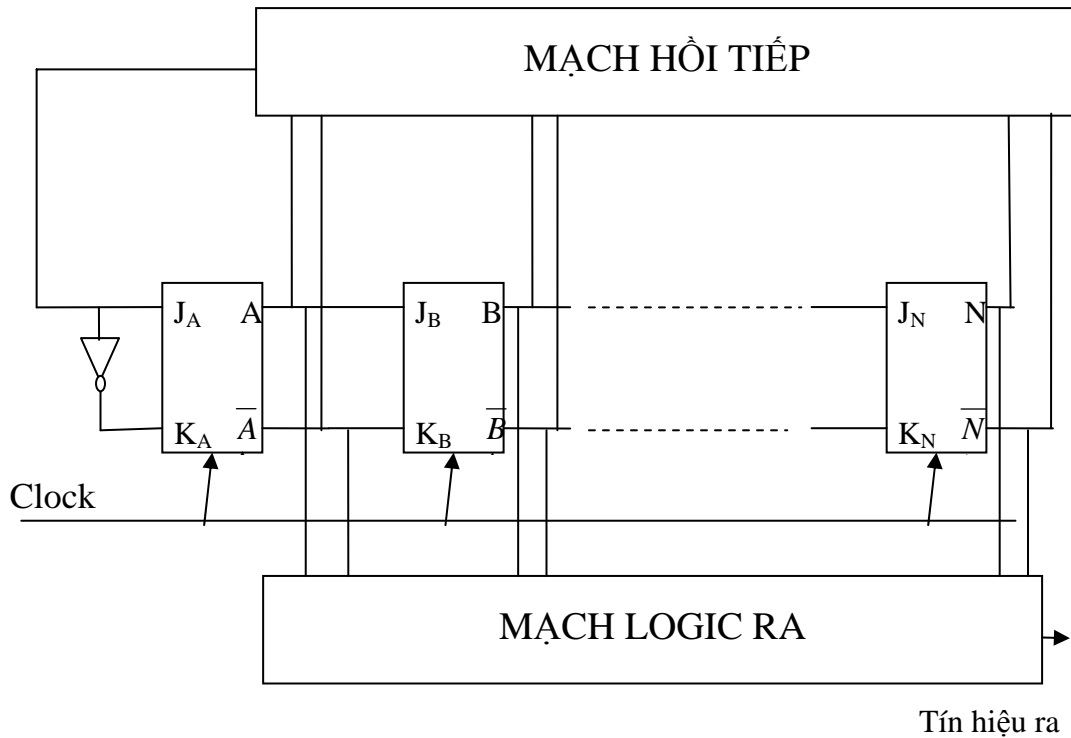
#### IV. MẠCH TẠO DÃY TÍN HIỆU TUẦN HOÀN

Thanh ghi dịch có thể dùng để tạo dãy tín hiệu tuần hoàn theo yêu cầu cho trước.

##### 1. Sơ đồ khối:

Gọi  $L$  là chu kỳ của dãy tín hiệu tuần hoàn cần phải tạo, tức là để tạo dãy tín hiệu này mạch phải tuần hoàn và có  $L$  trạng thái 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 giống như phần trên, thêm vào đó cần xây dựng một mạch tín hiệu ra để lấy dãy tín hiệu tuần hoàn.

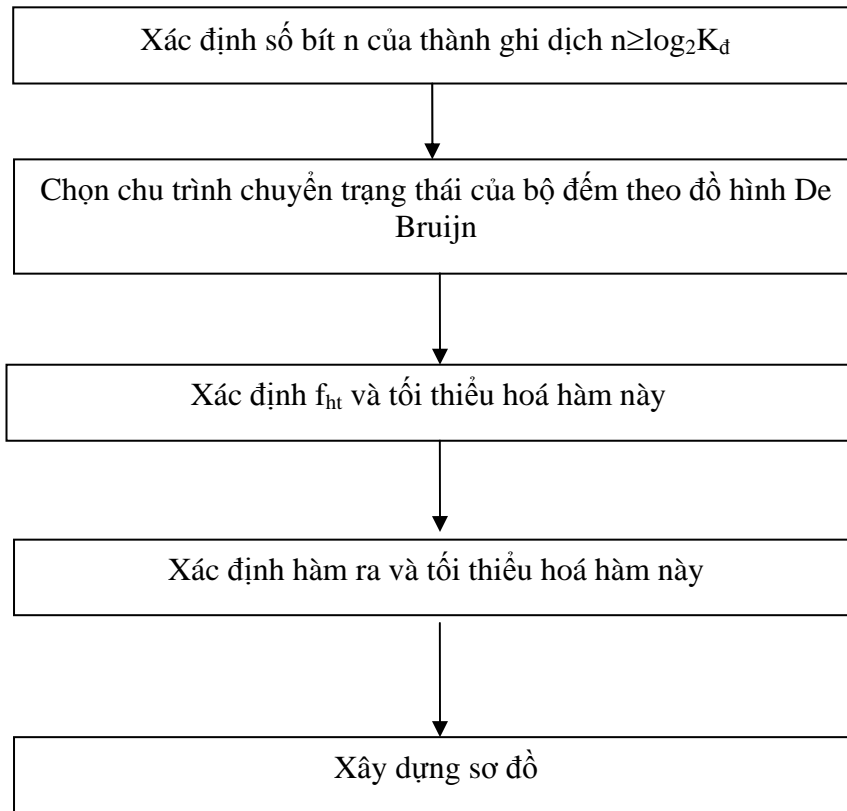
Ta có sơ đồ khối của mạch như sau:



Mạch tạo tín hiệu tuần hoàn dùng thanh ghi dịch



## 2. Các bước thiết kế



Minh họa: Thiết kế mạch tạo tín hiệu tuần hoàn:

$L=0-1-0-1-1-1-0-1$

- Bước 1: Tín hiệu có chiều dài  $L=8 \Rightarrow n=\log_2 8=3$

- Bước 2: chọn chu trình chuyển trạng thái:

Giả sử chọn:  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_7 \rightarrow S_6 \rightarrow S_5 \rightarrow S_2 \rightarrow S_4 \rightarrow S_0$

- Bước 3: Xây dựng hàm hồi tiếp và tối thiểu hoá

Ký hiệu các FF lần lượt là CBA, ta có bảng mã hoá trạng thái và hàm hồi tiếp như sau:

S	C	B	A	$f_{ht}$
S0	0	0	0	1
S1	0	0	1	1
S3	0	1	1	1
S7	1	1	1	0
S6	1	1	0	1
S5	1	0	1	0
S2	0	1	0	0
S4	1	0	0	0

Tối thiểu hoá  $f_{ht}$ :

<b>C</b> \ <b>BA</b>	00	01	11	10
0	1	1	1	
1				1

$$f_{ht} = \overline{C}B + \overline{C}A + C\overline{B}\overline{A}$$

- Bước 4: Xác định hàm ra:

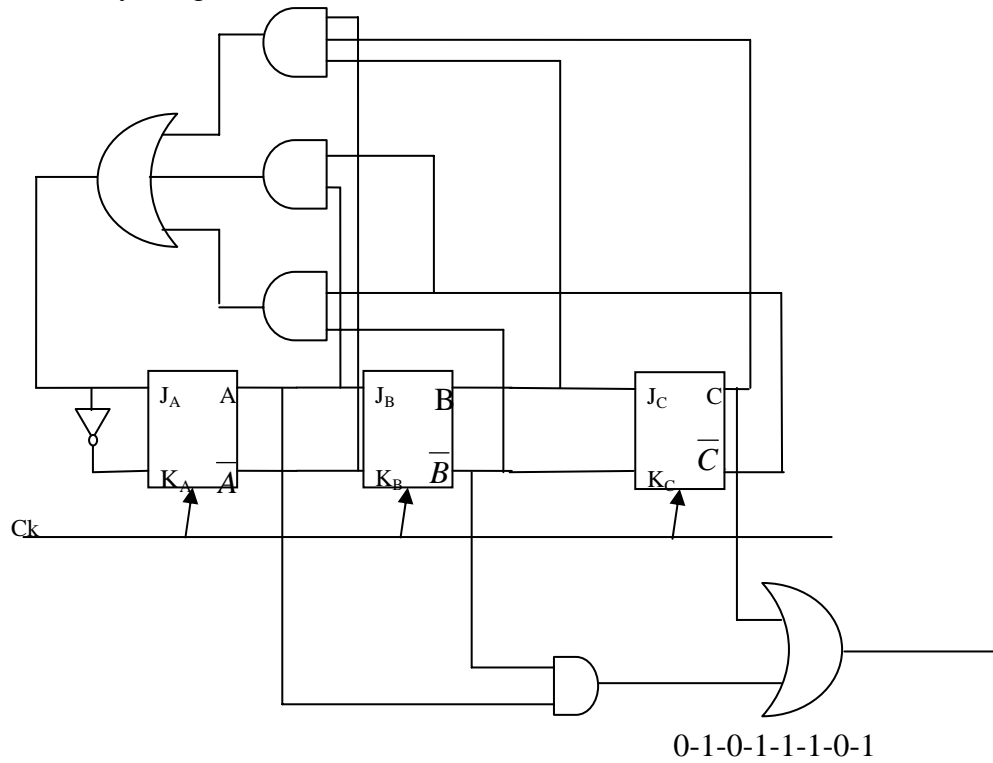
S	C	B	A	$f_{ht}$	L
S0	0	0	0	1	0
S1	0	0	1	1	1
S3	0	1	1	1	0
S7	1	1	1	0	1
S6	1	1	0	1	1
S5	1	0	1	0	1
S2	0	1	0	0	0
S4	1	0	0	0	1

Tối thiểu hoá  $f_{ht}$ :

<b>C</b> \ <b>BA</b>	00	01	11	10
0		1		
1	1	1	1	1

$$f_{ht} = C + \overline{B}A$$

-Bước 5: Xây dựng sơ đồ



- Bài tập: thiết kế mạch tạo chuỗi tín hiệu tuần hoàn:

L=0111000111

L=101110011110

## V. BỘ ĐẾM VÒNG, VÀ BỘ ĐẾM VÒNG XUẤN

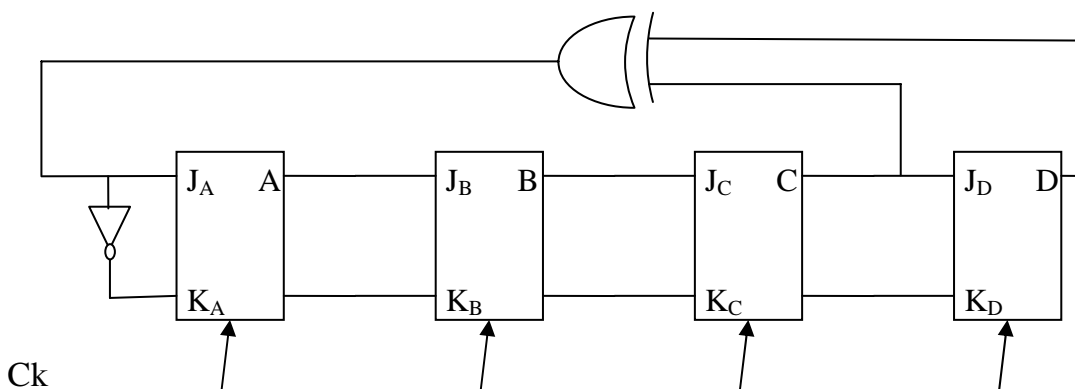
Trong bộ đếm vòng dùng thanh ghi dịch, hàm hồi tiếp được lấy từ đầu ra của FF cuối cùng và đưa vào FF đầu tiên, trong mạch này ở một thời điểm chỉ có một FF lưu trạng thái 1, ...các vấn đề khác tương tự như phần thiết kế bộ đếm.

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.

## VI. BỘ GHI DỊCH VỚI HÀM HỒI TIẾP LÀ HÀM CỘNG MODULE 2

### 1. Bộ ghi dịch với hàm hồi tiếp là hàm cộng module có $L_{max}=2^n-1$

Thanh ghi dịch 4 bit cho trong hình sau, có hàm hồi tiếp



$$f_{ht}=J_A=C \oplus D$$

Dựa vào đồ hình De Bruijn của bộ ghi dịch 4 bit và phương trình của hàm hồi tiếp đồng thời là hàm kích cho FF-A để 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 S1(0001, D=0,C=0,B=0,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 S2(0010, D=0,C=0,B=1,A=0), bảng đầy đủ như sau:

S	D	C	B	A	F
S1	0	0	0	1	0
S2	0	0	1	0	0
S4	0	1	0	0	1
S9	1	0	0	1	1
S3	0	0	1	1	0
S6	0	1	1	0	1
S13	1	1	0	1	0
S10	1	0	1	0	1
S5	0	1	0	1	1

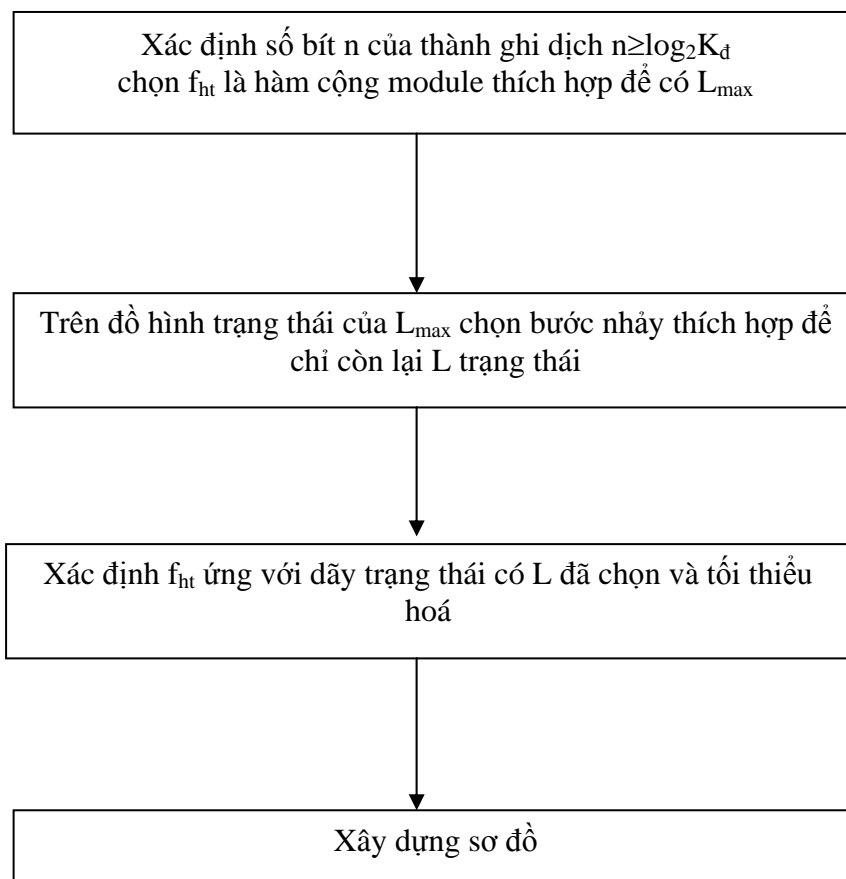
S11	1	0	1	1	1
S7	0	1	1	1	1
S15	1	1	1	1	0
S14	1	1	1	0	0
S12	1	1	0	0	0
S8	1	0	0	0	1

Trạng thái S0(0000) không xuất hiện trong bảng trạng thái, vì 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 mà không thoát ra được, đây chính là trạng thái khoá, nên phải loại S0.

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  $L_{max}$

## 2. Ví dụ ứng dụng mạch loại này để thiết kế bộ đếm:

+ Các bước thiết kế:

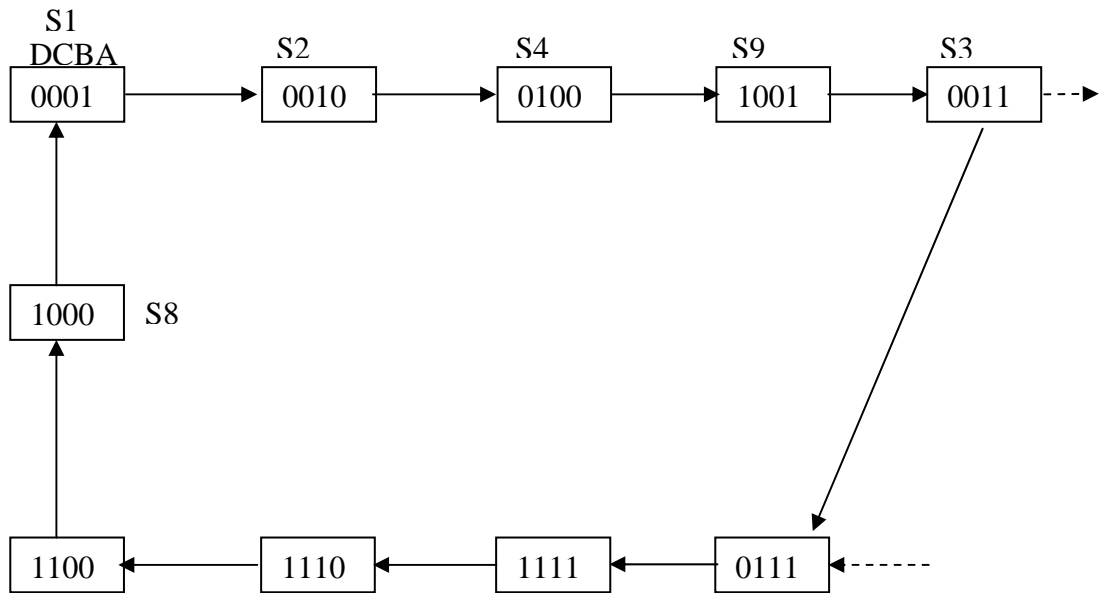


+ Minh hoạ: Thiết kế bộ đếm  $K_d=10$ , dùng bộ ghi dịch và hàm hồi tiếp cộng module 2  
- Bước 1:  $K_d=L=10$  do vậy  $n=4$ , chọn hàm hồi tiếp là hàm cộng module tương ứng để mạch đạt  $L_{max}$ , như đã biết để chọn:

$f_{ht}=C \oplus D$ , hoặc  $f_{ht}=A \oplus D...$

Giả sử chọn  $f_{ht}=C \oplus D$

- Bước 2: Trên đồ hình trạng thái  $L_{max}=15$  của bộ ghi dịch 4bit với  $f_{ht}=C \oplus D$  đã chọn, chọn bước nhảy để loại đi 5 trạng thái, giả sử chọn bước nhảy S3 về S7:

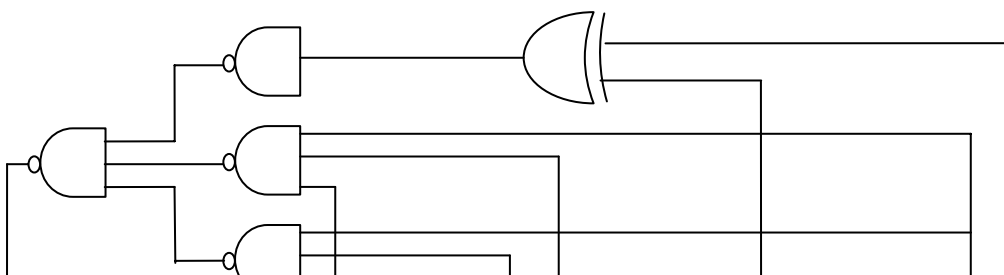


- Bước 3: Xác định  $f^*_{ht}$  dựa vào đồ hình De Bruijn của bộ ghi dịch 4 bit, xác định giá trị của  $f^*_{ht}$  ứng với dãy tín hiệu đã chọn:

S	D	C	B	A	$f^*_{ht}$
S1	0	0	0	1	0
S2	0	0	1	0	0
S4	0	1	0	0	1
S9	1	0	0	1	1
S3	0	0	1	1	0
S7	0	1	1	1	1
S15	1	1	1	1	0
S14	1	1	1	0	0
S12	1	1	0	0	0
S8	1	0	0	0	0

Tối thiểu hoá ta được  $f^*_{ht} = C \oplus D + ABD + \overline{ABD}$

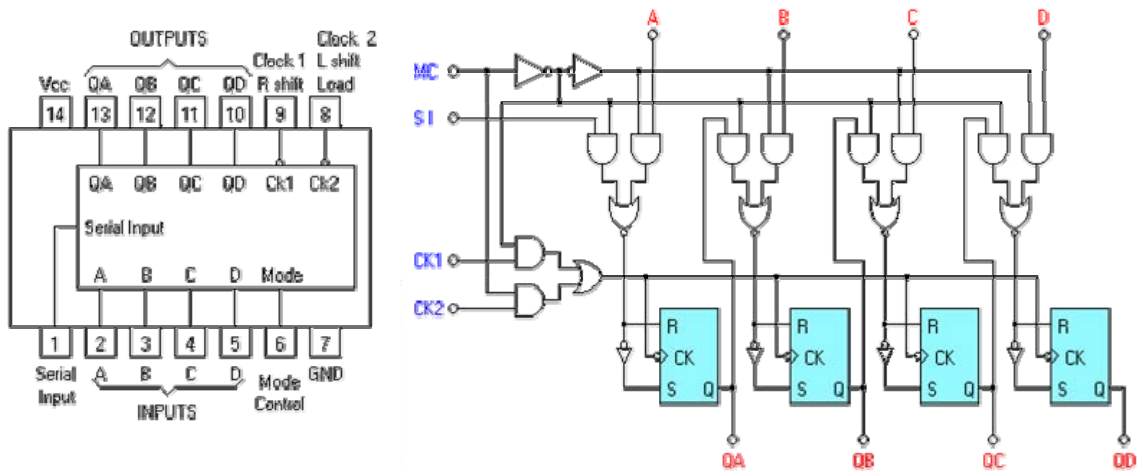
-Bước 4: Xây dựng sơ đồ :



**IC: Vi Mạch tích hợp SN 74LS95.**

Vi mạch SN 74LS95 chứa các thanh ghi dịch 4 bit với các đầu vào và ra có thể hoạt động theo kiểu song song hay nối tiếp; nó còn cho phép dịch phải hoặc dịch trái (chế độ dịch trái có thể thực hiện được với các kết nối thêm bên ngoài).

Sơ đồ chân và sơ đồ logic của vi mạch 74LS95





## CHƯƠNG 11 MẠCH DÂY ĐỒNG BỘ

**Mạch dây đồng bộ là một mạch số bao gồm các mạch tổ hợp và các phần tử nhớ FF, mạch hoạt động theo sự đồng bộ của xung nhịp  $Ck$ .**

### I. PHÂN TÍCH

Bài toán phân tích là bài toán xác định chức năng cho trước;

- Sơ đồ mạch:

Từ sơ đồ mạch cho trước cần xác định chức năng từng phần tử cơ bản của sơ đồ, và mối liên hệ giữa các sơ đồ đó

- Xác định 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, đặc điểm các đầu vào ra, xác định trạng thái của mạch

- Xác định phương trình hàm ra, hàm kích cho các FF

Dựa vào sơ đồ cho trước xác định hệ phương trình hàm ra, hàm kích cho các FF

- 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 đến, 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 xác định được ở trên, và dựa vào phương trình đặc tính của FF xác định được trạng thái chuyển tới và tín hiệu ra tương ứng với tín hiệu vào và trạng thái hiện tại của mạch.

- Đồ hình trạng thái:

Từ bảng trạng thái và bảng ra đã lập ở trên, xây dựng được đồ hình trạng thái và tín hiệu ra của mạch.

- 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 đó mà rút ra được chức năng của mạch.

### II. THIẾT KẾ

#### 1. Các bước thiết kế:

Các bước thiết kế ngược lại với các bước phân tích, tuy nhiên mạch thiết kế phải thực hiện chức năng cho trước với sơ đồ càng đơn giản càng tốt, điều này đồng nghĩa với việc tối thiểu hoá trạng thái.

- 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 và phải xác định mạch được thiết kế từ những yếu tố nào?

- Bước 2: Xác định tín hiệu vào/ra:

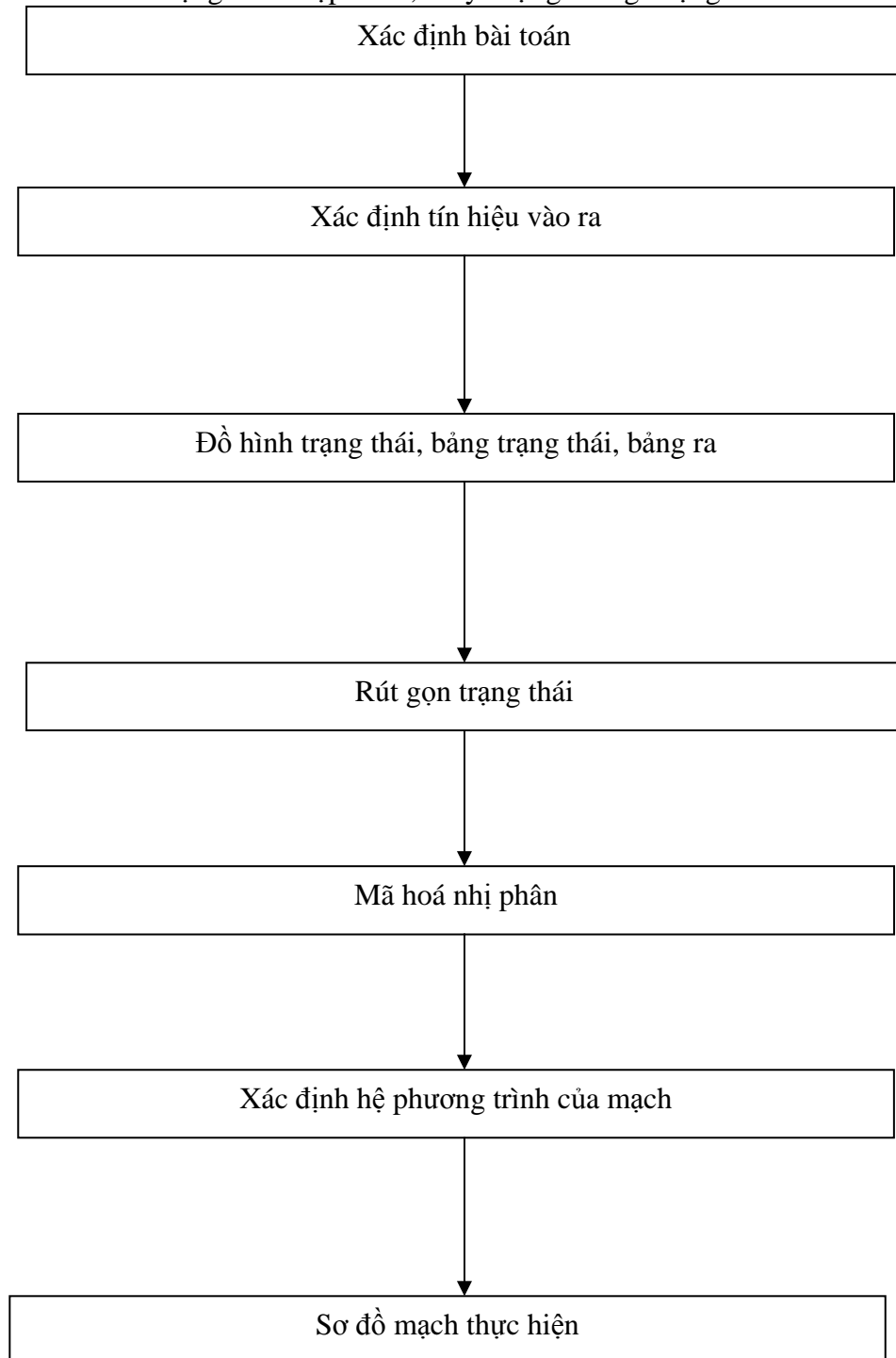
Xác định tín hiệu vào, ra cũng như đặc điểm của chúng

- Bước 3: Xây dựng đồ hình trạng thái, bảng trạng thái, bảng tín hiệu ra

Nói chung việc xây đồ hình trạng thái không dựa trên một quy tắc nào, mà phần lớn dựa vào **kinh nghiệm** của người thực hiện, nhưng quy tắc là dựa vào yêu cầu của bài toán đặt ra (tín hiệu vào ra), từ đó lập 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 trên, xây dựng bảng trạng thái và bảng tín hiệu



ra

- Bước 4: Tối thiểu hoá

Như đã biết số trạng thái trong của mạch gần như tỉ lệ với số FF được dùng trong mạch, việc tối thiểu hoá trạng thái **chủ yếu** dựa vào khái niệm tương đương, ta sẽ xét sau.

- Bước 5: Mã hoá nhị phân

Sau khi tối thiểu hoá các trạng thái, và đưa vào đồ hình trạng thái tương ứng, dùng các biến nhị phân để mã hoá cho các trạng thái, số biến nhị phân dùng để mã hoá các trạng thái phụ thuộc vào số trạng thái trong.

Nếu số lượng trạng thái trong là  $N$ , số lượng biến nhị phân dùng là  $n$ , thì  $n$  phải thoả mãn điều kiện:  $n \geq \log_2 N$

Có nhiều cách để mã hoá khác nhau, mỗi cách cho một sơ đồ khác nhau, yêu cầu phải mã hoá sao cho sơ đồ là đơn giản nhất.

- 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 hệ phương trình của mạch theo 2 cách:

+ 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 tín hiệu ra của mạch sau đó tiến hành tối thiểu hoá hệ phương trình đó.

- Bước 7: Sơ đồ mạch thực hiện

Từ các phương trình đã xác định được ta xây dựng sơ đồ mạch

### III. TỐI THIỂU HOÁ:

Nhiệm vụ đặt ra cho người thiết kế là phải đưa ra mạch hoạt động đúng chức năng cho trước, sơ đồ càng đơn giản càng tốt, muốn vậy cần giảm nhỏ số trạng thái trong.

Việc tối thiểu hoá chủ yếu dựa vào khái niệm "trạng thái tương đương", bài toán tối thiểu hoá trạng thái chính là bài toán đi tìm lớp trạng thái tương đương lớn nhất, và được thay bằng một trạng thái đại diện, số trạng thái tương đương chính là số trạng thái tối thiểu. Có 3 phương pháp tối thiểu hay dùng:

#### 1, Phương pháp Caldwell:

Phương pháp này được minh họa qua ví dụ sau:

Thiết kế một mạch dãy 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 đầu vào X, nếu dãy tín hiệu vào có dạng 010 hoặc 110 hoặc 111 thì tín hiệu ra  $Z=1$  để báo hiệu là mạch đã nhận được một trong các dãy tín hiệu vào đó.

- 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 010, hoặc 011, hoặc 110 hoặc 111 thì  $Z=1$  để báo hiệu mạch đã nhận được một trong các dãy tín hiệu đó.

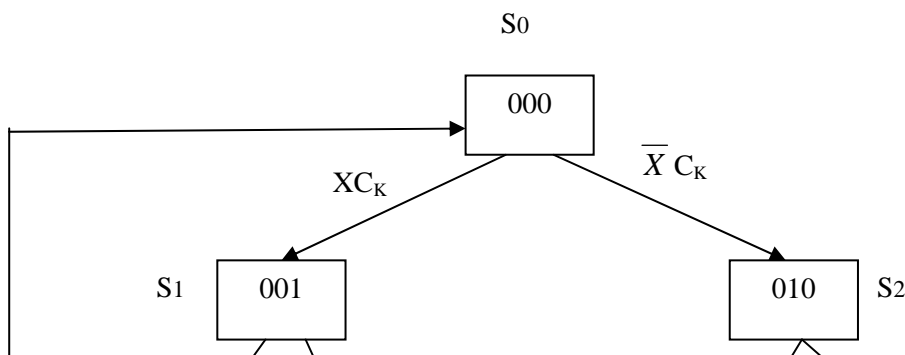
- Bước 2: Xác định tín hiệu vào ra

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, theo dạng sơ đồ khối như sau:



- 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 trạng thái như sau:

Sơ đồ chuyển đổi trạng thái:



Trạng thái ban đầu là  $S_0$  : khi tín hiệu vào là  $X C_K$  mạch chuyển đến  $S_1$ ,  $\overline{X} C_K$  mạch chuyển đến  $S_2$ .

Tương tự với trạng thái  $S_1$ , mạch sẽ chuyển đến  $S_3$  khi tín hiệu vào là  $X C_K$ , mạch chuyển đến  $S_2$  khi tín hiệu vào là  $\overline{X} C_K$

Tương tự với trạng thái  $S_2$ , mạch sẽ chuyển đến  $S_5$  khi tín hiệu vào là  $X C_K$ , mạch chuyển đến  $S_2$  khi tín hiệu vào là  $\overline{X} C_K$ .

Nếu mạch ở một trong 4 trạng thái  $S_3, S_4, S_5, S_6$ , khi có tín hiệu vào là  $X C_K$  hoặc  $\overline{X} C_K$  mạch sẽ chuyển đến 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 đường chuyển đổi khác  $Z=0$ , từ đồ hình trạng lập được bảng trạng thái như bảng sau:

S	S'		Z	
	X=0	X=1	X=0	X=1
S0	S2	S1	0	0
S1	S4	S2	0	0
S2	S6	S5	0	0
S3	S0	S0	1	1
S4	S0	S0	0	0
S5	S0	S0	1	1
S6	S0	S0	0	0

- Bước 4: Tối thiểu hoá trạng thái, ta sẽ dùng phương pháp Caldwell:

+ Trạng thái tương đương:

Trạng thái  $S_i$  được gọi là tương đương với  $S_j$  khi và chỉ khi nếu lấy  $S_i$  và  $S_j$  là 2 trạng thái ban đầu thì với mọi tín hiệu vào có thể có chúng luôn luôn cho tín hiệu ra giống nhau. Nếu có nhiều trạng thái tương đương với nhau từng đôi một thì chúng tương đương với nhau. Để kiểm tra nhóm các trạng thái 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 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 nhóm các trạng thái phải tiến hành tuần tự từng bước cho đến nhóm có trạng thái cuối cùng, nếu nhóm có trạng thái cuối cùng này là tương đương thì nhóm trạng thái được 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 kết hợp với nhau và được biểu diễn một hàng chung - đặc trưng (trạng thái đặc trưng) cho chúng nếu như chúng thoả mãn 2 điều kiện sau:

- 1- Các hàng tương ứng trong ma trận giống nhau
- 2- Trong ma trận ra, các hàng tương ứng phải thoả mãn một trong 3 điểm:
  - Các hàng trong ma trận phải giống nhau
  - Các trạng thái trong cùng một cột nằm trong nhóm trạng thái được xét
  - Các trạng thái ở trong cùng một cột là các trạng thái tương ứng

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, cho 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 lúc đó là tối thiểu.

Áp dụng quy tắc Caldwell vào bài toán trên, trong bảng trên:

S	S'		Z	
	X=0	X=1	X=0	X=1
S0	S2	S1	0	0
S1	S4	S2	0	0
S2	S6	S5	0	0
S3	S0	S0	1	1
S4	S0	S0	0	0
S5	S0	S0	1	1
S6	S0	S0	0	0

S4 tương đương S6, S3 tương đương S5, thay thế các trạng thái tương đương bằng trạng thái đặc trưng cho chúng, chẳng hạn thay thế S4, S6 bằng S46, S3 và S5 bằng S35, từ đó lập được bảng mới như sau:

S	X=0	X=1
S0	S2 Z=0	S1 Z=0
S1	S46 Z=0	S35 Z=0
S2	S46 Z=0	S35 Z=0
S35	S0 Z=1	S0 Z=1

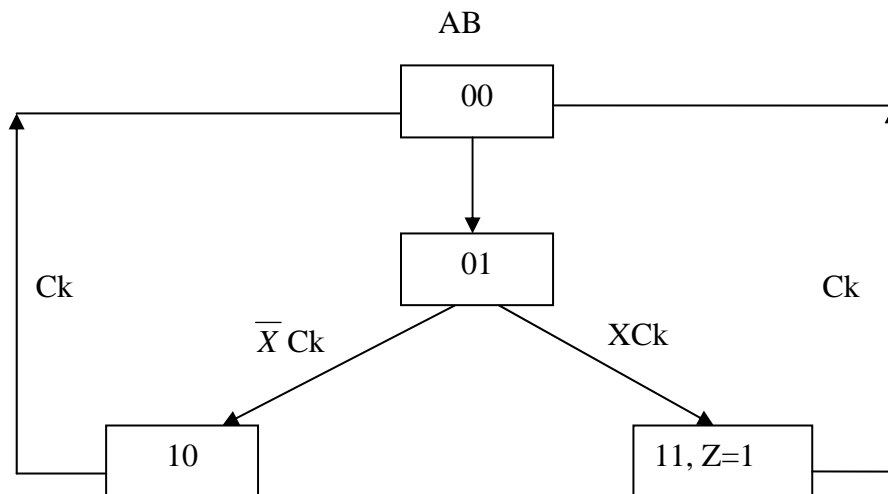
S46	S0 Z=0	S0 Z=0

Từ đây ta lại thấy S1~S2, bảng sẽ rút gọn như sau:

S	X=0	X=1
S0	S12 Z=0	S12 Z=0
S12	S46 Z=0	S35 Z=0
S35	S0 Z=1	S0 Z=1
S46	S0 Z=0	S0 Z=0

Từ bảng trên, không có trạng thái nào tương đương nữa, như vậy mạch còn lại 4 trạng thái, với đồ hình trạng thái như sau:

AB	S
00	S0
01	S12
11	S35
10	S46



- Bước 5: Mã hoá trạng thái, các trạng thái được mã hoá như bảng trên

- Bước 6: Xác định hệ phương trình của mạch.

Dựa vào bảng trạng thái, bảng ra để xác định đầu vào kích cho các FF và phương trình tín hiệu ra.

Đầu vào kích cho các FF-A và FF-B là  $J_A, K_A, J_B, K_B$  tương ứng với giá trị của tín hiệu vào X, theo bảng sau:

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	$J_A$	$K_A$	$J_A$	$K_A$	$J_B$	$K_B$	$J_B$	$K_B$
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

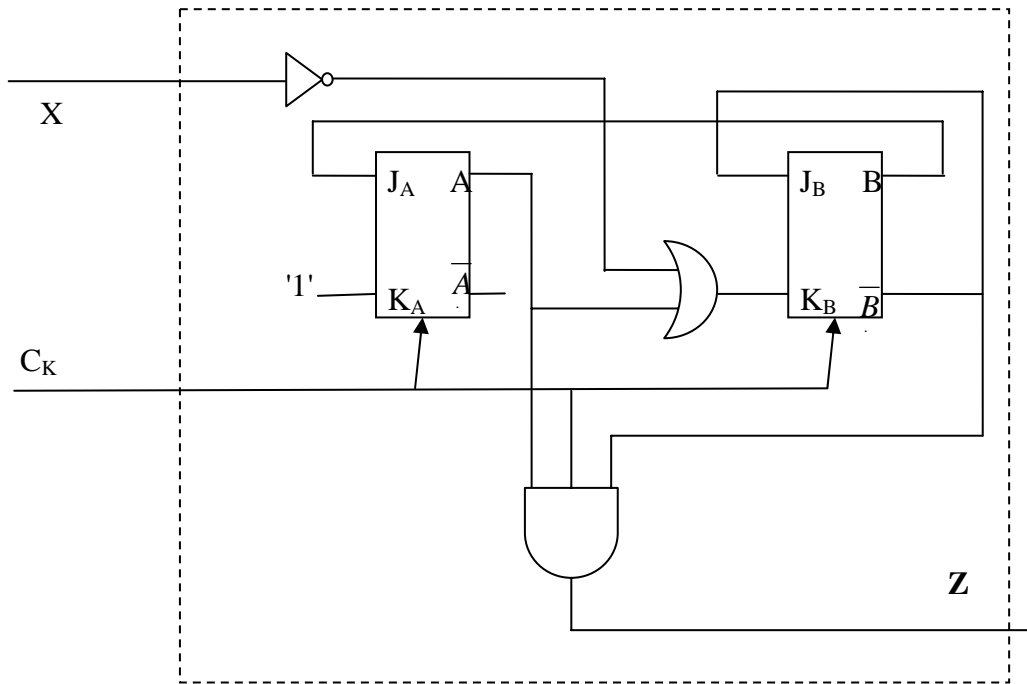
Sau khi tối thiểu hoá, ta thu được:

$$J_A=B; K_A=1; J_B=\bar{B}; K_B=\bar{X} + A;$$

Phương trình tín hiệu ra Z được xác định dựa vào bảng chuyển đổi trạng thái, bảng ra và bảng mã hoá :  $Z=A.B.C_K$

Ngoài cách này có thể dựa trực tiếp vào phương trình  $T_{on}$  và  $T_{off}$  để xác định.

- Bước 7: Sơ đồ mạch:



## 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:

- Sự phân chia lần thứ nhất(P1)

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.

- Sự phân chia lần 2(P2)

Xét các trạng thái trong cùng một lớp của P1: Nếu các trạng thái nằm trong cùng một lớp của P1 có các trạng thái tiếp theo cùng nằm trong cùng một lớp của P1 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 P2

Tiếp tục thực hiện như trên, sẽ được P3,P4,..., Pi. Quá trình phân chia đó sẽ dừng lại khi  $P_{i+1}=P_i$ , số nhóm  $P_i$  chính là số trạng thái đã được tối thiểu hoá.

Sau đây là một ví dụ minh hoạ:

Bảng chuyển đổi trạng thái và tín hiệu ra của một mạch dãy đồng bộ mô tả ở bảng sau:

Trạng thái hiện tại	Trạng thái tiếp theo	
	X=0	X=1
S0	S0 Z=0	S4 Z=1
S1	S4 Z=1	S2 Z=0
S2	S0 Z=1	S3 Z=1
S3	S5 Z=0	S6 Z=1
S4	S1 Z=1	S2 Z=0
S5	S5 Z=0	S4 Z=1
S6	S0 Z=1	S3 Z=1

+ Sự phân chia lần 1:

$$P1=(S0,S3,S5)(S1,S4)(S2,S6)$$

+ Sự phân chia lần 2:

$$P2=(S3)(S0, S5)(S1,S4)(S2,S6)$$

Ta có  $P3=P2=(S3)(S0, S5)(S1,S4)(S2,S6)$

Trạng thái hiện tại	Trạng thái tiếp theo	
	X=0	X=1
S05	S05 Z=0	S14 Z=1
S14	S14 Z=1	S26 Z=0
S26	S05 Z=1	S3 Z=1
S3	S05 Z=0	S26 Z=1

Như vậy các lớp trạng thái tương đương của mạch đã được xác định, sau khi tối thiểu hoá mạch chỉ có 4 trạng thái như trên.

### 3. Phương pháp dùng bảng so sánh

Giả sử ban đầu mạch có n-1 trạng thái trong S0, S1, S2, ..., Sn Lập bảng so sánh có các hàng và cột như hình vẽ.

Mỗi ô biểu diễn cặp so sánh giữa 2 trạng thái tương ứng(Si, Sj) tại hàng và cột tương ứng với ô đó. Tiến hành tối thiểu hoá như sau:

- Trong mỗi ô, nếu cặp trạng thái tương ứng có tín hiệu ra giống nhau với mọi tổ hợp tín hiệu vào(hàng trong bảng ra như nhau) thì ghi vào đó cặp trạng thái mà nó chuyển đến ứng với từng tổ hợp tín hiệu vào. Nếu tín hiệu ra khác nhau thì đánh dấu X vào đó.

- Kiểm tra những ô ghi 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 trước.



Tiến hành như bước 2 cho tới khi nào không gạch được ô nào 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.

- Kiểm tra các trạng thái tương đương với  $S_i$ : trên bảng kéo theo sẽ kiểm tra từng cột từ phải ( $S_n$ ) sang 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.

Minh họa:

Cho bảng chuyển đổi trạng thái và tín hiệu ra mô tả như bảng sau:

Trạng thái hiện tại	Trạng thái tiếp theo	
	X=0	X=1
S0	S0 Z=0	S2 Z=0
S1	S3 Z=1	S0 Z=0
S2	S5 Z=0	S5 Z=0
S3	S4 Z=1	S1 Z=0
S4	S6 Z=1	S6 Z=0
S5	S2 Z=0	S0 Z=0
S6	S1 Z=1	S7 Z=0
S7	S7 Z=0	S2 Z=0

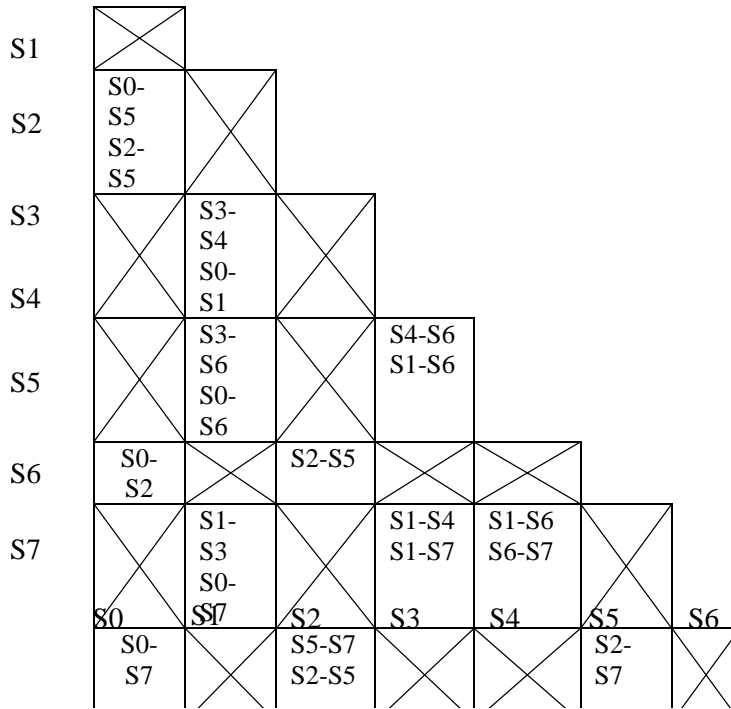
Sau đây là các bước tối thiểu hoá:

- Bước 1: Đánh 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. 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

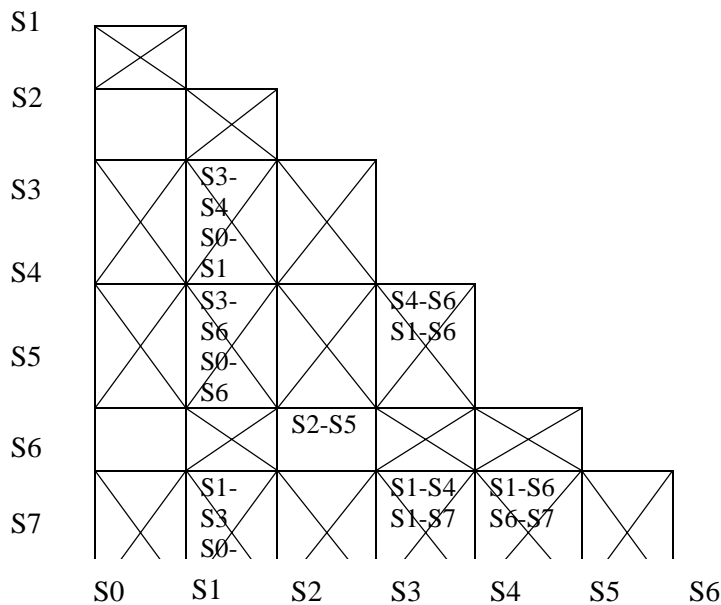
- Bước 2: Tiến hành kiểm tra các ô trong bảng

Kiểm tra các ô trong bảng sau:

S1							
S2							
S3							
S4							
S5							
S6							
S7							
	S0	S1	S2	S3	S4	S5	S6



Gạch thêm một số ô mới, được bảng ở hình sau



S5: (S6)(S5, S7)

S4: (S6)(S5, S7)(S4)

S3: (S6)(S5, S7)(S4)(S3)

S2: (S6)(S2, S5, S7)(S4)(S3)(S1)

S1: (S6)(S2, S5, S7)(S4)(S3)(S1)

Số trạng thái tối thiểu bằng số nhóm trạng thái tương đương cột S0, mạch có số trạng thái tối thiểu là 5.

Lập được bảng chuyển đổi trạng thái và tín hiệu ra đã tối thiểu như bảng sau:

Trạng thái hiện tại	Trạng thái tiếp theo	
	X=0	X=1
S0257	S0257 Z=0	S0257 Z=0
S1	S3 Z=1	S0257 Z=0
S3	S4 Z=1	S1 Z=0
S4	S6 Z=1	S6 Z=0
S6	S1 Z=1	S0257 Z=0

#### IV. MÃ HOÁ TRẠNG THÁI

Trong các minh hoạ trước đây các trạng thái của mạch được mã hoá tuỳ ý, có nhiều cách mã hoá khác nhau và mỗi cách cho một sơ đồ khác nhau, cần phải chọn cách mã hoá sao cho sơ đồ mạch thực hiện là đơn giản nhất. Hiện nay vẫn chưa có cách mã hoá nào là tối ưu nhất.

Thường thực hiện theo 2 cách sau để có được sơ đồ tương đối đơn giản

- Quy tắc 1:

Nếu một trạng thái hiện tại  $S_i$  có thể chuyển biến đến nhiều trạng thái  $S_i'1, S_i'2, \dots, S_i'n$  thì các trạng thái  $S_i'j$  phải được mã hoá bằng các từ mã kế cận (chỉ khác nhau 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 kế cận nhau.

#### V. MẠCH DÂY ĐỒNG BỘ DÙNG MÔ HÌNH MOORE VÀ MÔ HÌNH MEALY

Nhắc lại:

Mô hình Moore là mô hình mà tín hiệu ra chỉ phụ thuộc vào trạng thái trong của mạch.

Mô hình Mealy là mô hình mà tín hiệu ra phụ thuộc cả vào trạng thái trong và tín hiệu vào X.

Ta nghiên cứu 2 mô hình này qua một ví dụ:

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 theo dạng nhị phân, mỗi lần dãy tín hiệu vào là 101, mạch sẽ cho ra tín hiệu  $Z=1$ , các bit dữ liệu vào được đồng bộ với xung nhịp  $Ck$ .

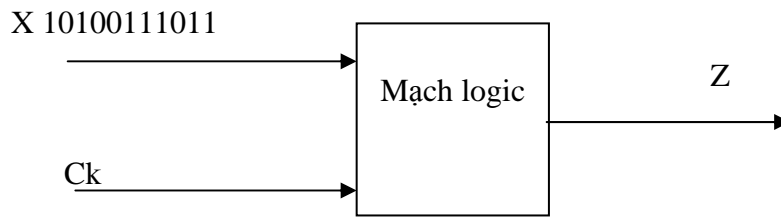
Giải:

- Bước 1: Xác định bài toán

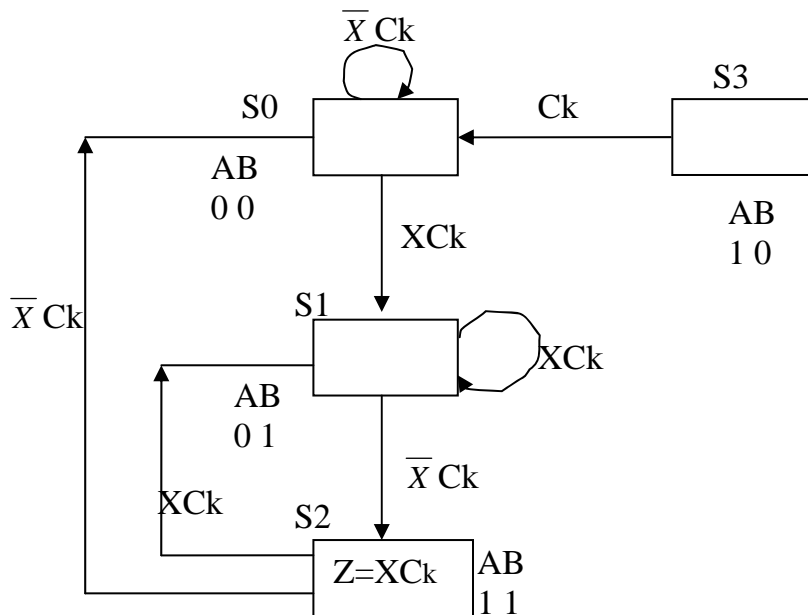
Theo đầu bài ra ta cần phải thiết kế 1 mạch dây đồng bộ để phát hiện dãy tín hiệu vào, nếu dãy tín hiệu vào có dạng 101 thì đầu ra  $Z=1$

- Bước 2: Xác định tín hiệu vào, ra:

Mạch có 2 đầu tín hiệu vào là X và xung nhịp  $Ck$ , và có một đầu tín hiệu ra Z, theo như sơ đồ khối sau:



- Bước 3: đồ hình trạng thái, bảng trạng thái, và tín hiệu ra  
 a, Nếu dùng mô hình Mealy:

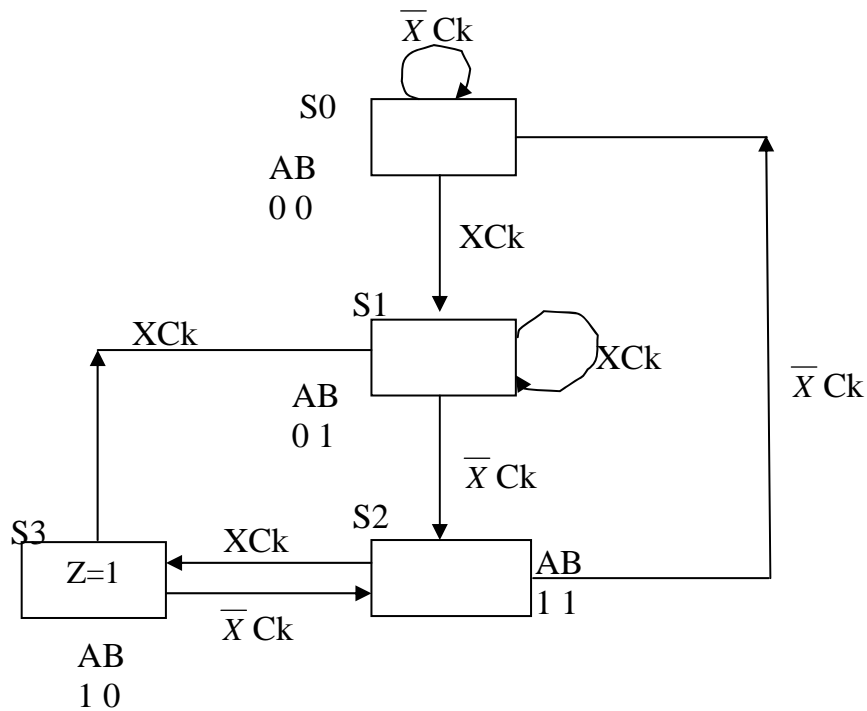


Theo yêu cầu của bài toán chỉ cần sử dụng 3 trạng thái để mã hoá S0, S1, S2. Như vậy dùng 2 bit nhị phân sẽ thừa 1 trạng thái, để tránh tình trạng rơi vào trạng thái khóa, ta thiết kế để mạch khi rơi vào trạng thái đó, thì ngay từ xung nhịp đầu tiên nó sẽ về S0. Ta có bảng trạng thái và tín hiệu ra:

b, Với mô hình Moore:

Chuyển từ mô hình Mealy sang Moore ta được sơ đồ và cách mã hoá như sau:

A	B	S
0	0	S0
0	1	S1
1	1	S2
1	0	S3



Trạng thái S3 chính là trạng thái cho tín hiệu ra Z=1

- Bước 4: Tối thiểu hoá

Với ví dụ này ta nhận thấy các trạng thái không thể tối thiểu thêm được nữa.

- Bước 5: Mã hoá trạng thái, với bài toán này áp dụng phương pháp mã hoá khác nhau 1 bit

A	B	S
0	0	S0
0	1	S1
1	1	S2
1	0	S3

- Bước 6: Hệ phương trình của mạch

a, Mô hình Mealy:

$$\begin{aligned} T_{ON A} &= S1 \bar{X} = \bar{A} \bar{B} \bar{X} & \Rightarrow J_A &= B \bar{X} \\ T_{Off A} &= S3 + S2 \bar{X} + S2 X = S2 + S3 = A & \Rightarrow K_A &= 1 \\ T_{ON B} &= S0 X = \bar{A} B X & \Rightarrow J_B &= \bar{A} X \\ T_{Off B} &= S2 \bar{X} = A B \bar{X} & \Rightarrow K_B &= A \bar{X} \end{aligned}$$

Và  $Z = A B X C_k$

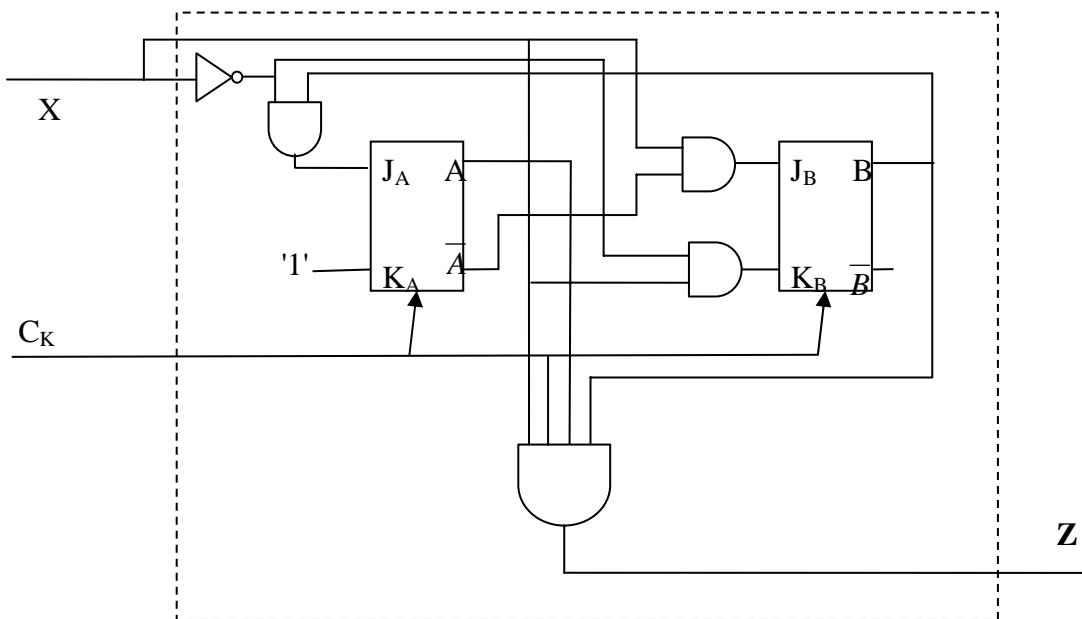
a, Mô hình Moore:

$$\begin{aligned} T_{ON A} &= S1 \bar{X} = \bar{A} \bar{B} \bar{X} & \Rightarrow J_A &= B \bar{X} \\ T_{Off A} &= S2 \bar{X} + S3 X = A B \bar{X} + A \bar{B} X & \Rightarrow K_A &= B \oplus X \\ T_{ON B} &= \bar{B} (A X + A) & \Rightarrow J_B &= A + X \\ T_{Off B} &= S2 \bar{X} + S2 X & \Rightarrow K_B &= A \end{aligned}$$

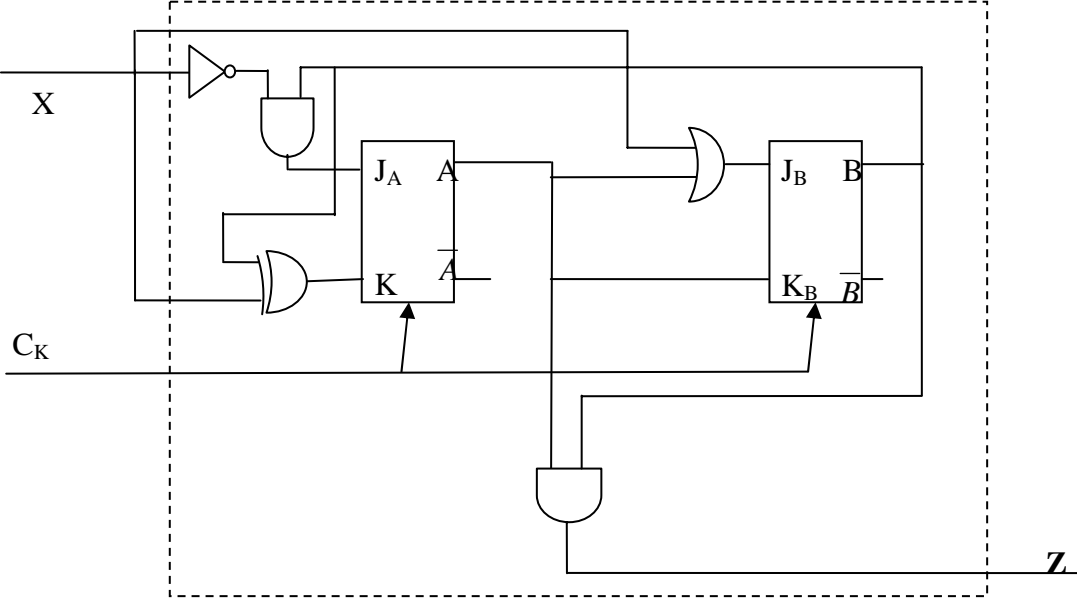
Và  $Z = A \bar{B}$

- Bước 7: Sơ đồ

a, Mô hình Mealy:



b, Mô hình Moore:



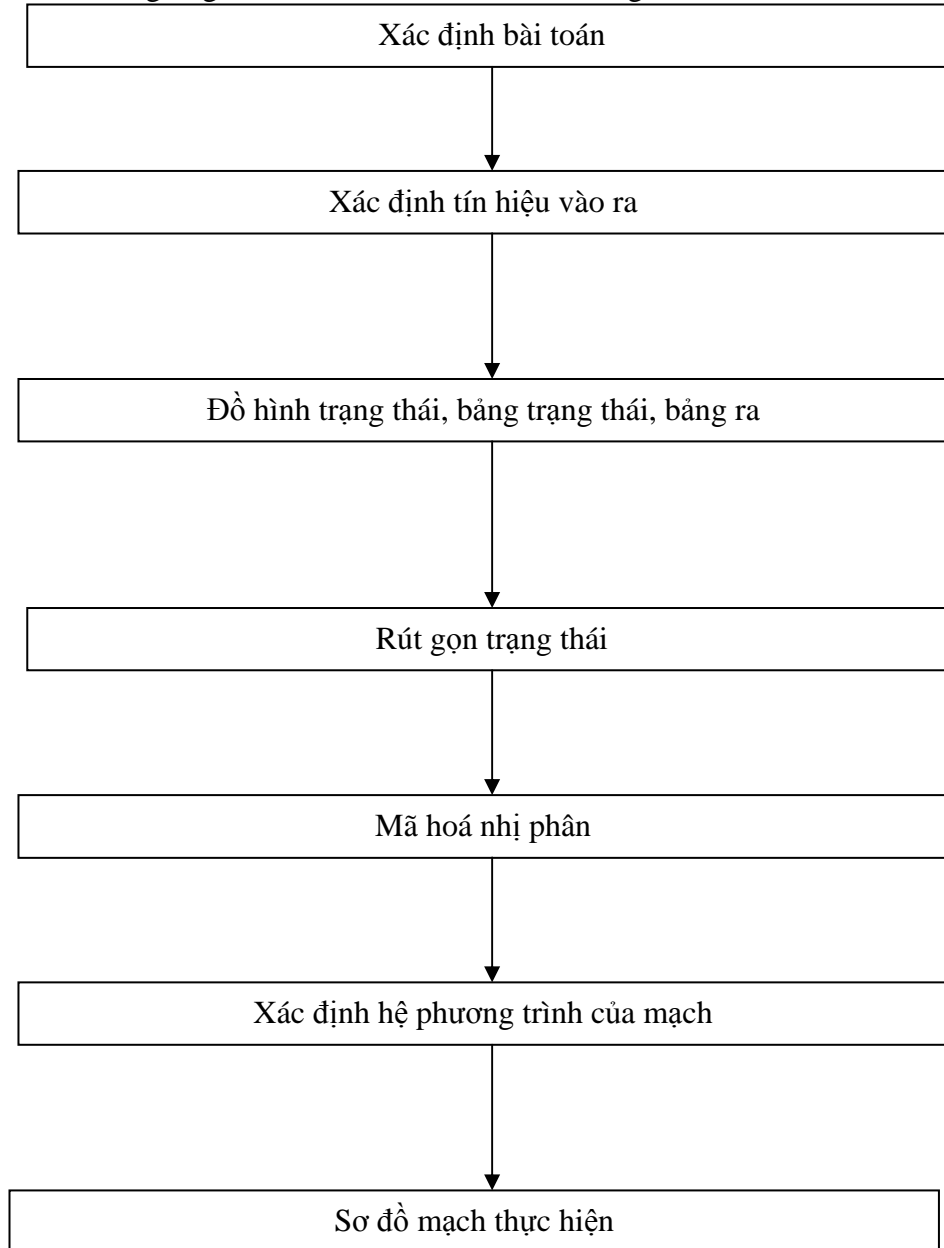
## CHƯƠNG 12 MẠCH DÂY KHÔNG ĐỒNG BỘ

Nếu mạch dây đồng bộ hoạt động theo sự điều khiển của xung nhịp Ck thì mạch dây không đồng bộ hoạt động theo sự điều khiển bởi các sự kiện mà không tuân theo quy luật.

Tóm lại tất cả các mạch dây mà được điều khiển bởi các sự kiện ngẫu nhiên thì được xếp vào nhóm mạch dây không đồng bộ.

### I. CÁC BƯỚC THIẾT KẾ:

Về cơ bản giống với các bước thiết kế mạch đồng bộ, chỉ khác ở bước 5 và bước 6



- Bước 5: Mã hoá nhị phân

Mạch không đồng bộ hoạt động không có sự tác động của xung nhịp, cho nên trong mạch thường xuất hiện hiện tượng chạy đua hoặc hiện tượng chu kỳ, dẫn tới sự hoạt động không chính xác của mạch. Cho nên khi mã hoá cần có biện pháp để loại trừ.



- Bước 6: Xác định hệ phương trình của mạch:

Dựa vào bảng trạng thái bảng tín hiệu ra và có thể dựa trực tiếp vào đồ hình. Cả 2 cách đều có 2 loại phương trình:

+ Phương trình của mạch chỉ dùng NAND

+ Phương trình của mạch chỉ dùng RS-FF không đồng bộ và các mạch NAND

Cách 1: Dựa vào bảng trạng thái và tín hiệu ra

**\*. Chỉ dùng mạch NAND:**

Ký hiệu A, B, ..., N là các biến nhị phân dùng để mã hoá 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:

$$A' = f_A(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

$$B' = f_B(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

....

$$N' = f_N(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

$$Z_1 = g_1(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

$$Z_2 = g_2(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

....

$$Z_n = g_n(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

Tối thiểu hoá hệ hàm này và viết phương trình ở dạng chỉ dùng NAND

**\*. Mạch dùng RS-FF và các mạch NAND**

Trong bảng mã 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:

$$R_A = F_1(A, B, N, X_1, X_1, \dots, X_m)$$

$$R_B = F_2(A, B, N, X_1, X_1, \dots, X_m)$$

Tối thiểu hoá hàm và viết phương trình ở dạng dùng NAND. Cách làm tương tự với B, C, ..., N

Cách 2: Dựa trực tiếp vào đồ hình trạng thái

Ta có phương trình đầu vào kích R, S của FF-A là:

$$S_A = \text{Tập hợp bật(on) của A} + [(1)]$$

$$R_A = \text{Tập hợp tắt(off) của A} + [(0)]$$

Tương tự cho B, C, ..., N

**\*. Nếu chỉ dùng NAND**

Ta có phương trình của RS-FF như sau:

$$Q' = S + \overline{RQ} \Rightarrow A' = S_A + \overline{R_A} A$$

Sau đó tiến hành tối thiểu hoá từng phương trình và viết dưới dạng chỉ dùng NAND.

Tiến hành tương tự với B, C, ..., N

**\* Nếu chỉ 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:

$$R_A = F_{1A}(A, B, N, X_1, X_1, \dots, X_m)$$

$$S_A = F_{2A}(A, B, N, X_1, X_1, \dots, X_m)$$

$$R_B = F_{2B}(A, B, N, X_1, X_1, \dots, X_m)$$

...

$$R_N = F_{1N}(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

$$S_N = F_{2N}(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

$$Z_1 = Y_1(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

$$Z_2 = Y_2(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

...

$$Z_n = Y_n(A, B, \dots, N, X_1, X_1, \dots, X_m)$$

Tối thiểu hoá hệ phương trình: viết các phương trình ở dạng dùng mạch NAND.

## II. MÃ HOÁ VÀ TỐI THIỂU HOÁ THÁI TRONG MẠCH DÂY KHÔNG ĐỒNG BỘ.

### 1. Mã hoá trạng thái:

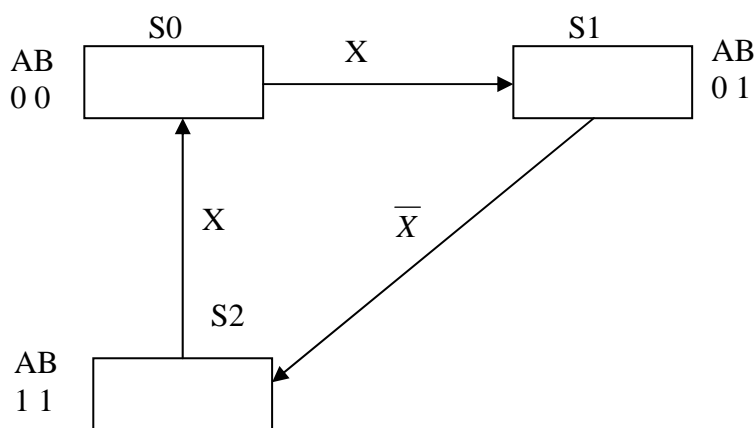
Nói chung mã hoá trạng thái ở loại mạch này về nguyên tắc cũng giống như hệ thống mạch dãy đã xem xét, nhưng lưu ý việc mã hoá phải tránh được hiện tượng chu kỳ và chạy đua.

Tránh hiện tượng chu kỳ: Phải lưu ý sao cho với mọi tổ hợp tín hiệu vào thì mạch luôn luôn có một trạng thái ổn định.

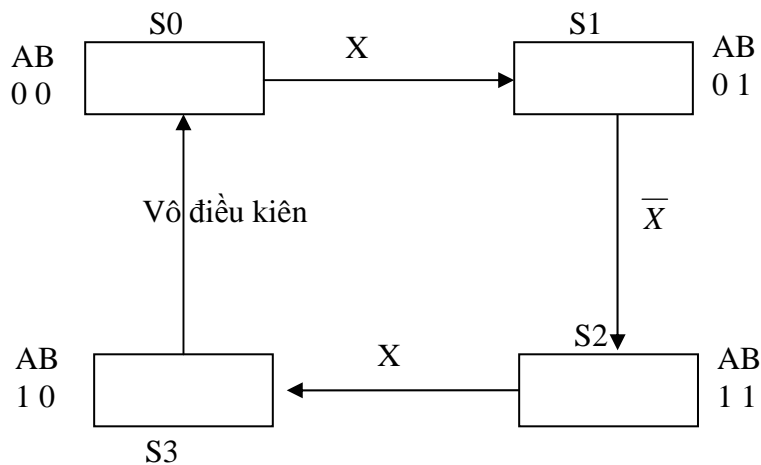
Tránh hiện tượng chạy đua: Phải lưu ý mã hoá sao cho với tất cả các chuyển biến trạng thái có thể có của mạch chỉ có duy nhất một biến thay đổi mà thôi, tức là các cặp từ mã mã hoá cho  $S_i$  và  $S_j$  ( $S_i \rightarrow S_j$ ) có khoảng cách Hamming là 1

Minh hoạ:

Cho đồ hình trạng thái của mạch dây không đồng bộ như hình sau:



Khi thay đổi từ  $S_2 \rightarrow S_0$  cả hai biến nhị phân đều thay đổi ( $11 \rightarrow 00$ ), điều này dẫn đến hiện tượng chạy đua trong mạch, để tránh ta đưa thêm biến  $S_3$  để sự thay đổi từ  $S_2 \rightarrow S_0$  thông qua  $S_3$ , tức là  $S_2 \rightarrow S_3 \rightarrow S_0$ , đảm bảo chỉ có một biến thay đổi, thường chọn điều kiện thoát khỏi trạng thái giả là vô điều kiện:



**Tóm lại:** Khi mã hoá các trạng thái trong của mạch, để tránh hiện tượng chu kỳ, và chạy đua, ta cần có trạng thái trung gian, thông thường tăng số bit mã hoá để tăng trạng thái, và lựa chọn trạng thái trung gian cho hợp lý, và phải thiết kế để việc thoát khỏi trạng thái trung gian là vô điều kiện

## 2. Tối thiểu hoá:

Giống như mạch dây đồng bộ, nhưng thường chú ý đến yếu tố mạch chạy tốt là được (không có hiện tượng chạy đua và chu kỳ)

### III. HIỆN TƯỢNG CHU KỲ VÀ CHẠY ĐUA

#### 1. Hiện tượng chu kỳ:

Hiện tượng chu kỳ là hiện tượng tại một tổ hợp tín hiệu 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.

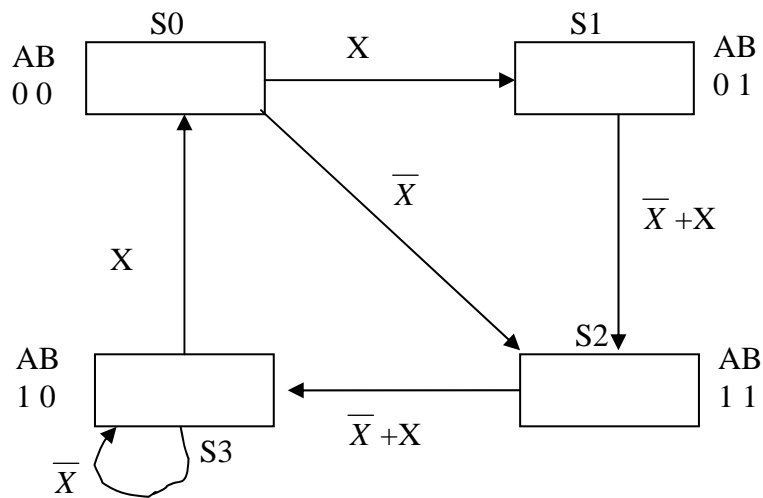
Ví dụ: ứng với một tổ hợp tín hiệu vào quá trình chuyển đổi trạng thái theo Hình vẽ dưới đây

Việc mã hoá trạng thái sử dụng biến nhị phân A và B là hoàn toàn tùy ý. Đồ hình trạng thái, và bảng trạng thái như sau:

Giả thiết ban đầu mạch ở trạng thái S3 (AB=10) và X=0, sau đó tín hiệu vào X thay đổi từ 0 → 1, mạch sẽ chuyển từ trạng thái S3 → S0.

Nếu X vẫn = 1 mạch sẽ chuyển lần lượt chuyển đến các trạng thái tiếp theo là S1, S2, ..., S0. Khi X= 1 ta có:

$S3 \rightarrow S0 \rightarrow S1 \rightarrow S2 \rightarrow S3$ , tức mạch không có trạng thái ổn định



$$X=1$$

X	0	1
S0	S2	S1
S1	S2	S2
S2	S3	S3
S3	S3	S0

## 2. Hiện tượng chạy đua:

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 nên khi chuyển trạng thái từ  $S_i \rightarrow S_j$  mạch có thể chuyển theo các đường khác nhau.

Nếu trạng thái cuối cùng của những con đường ấy là ổn định và duy nhất thì chạy đua là chạy đua không nguy hiểm, nhưng nếu cách chuyển trạng thái ấy dẫn đến các trạng thái ổn định khác nhau, có thể tới trạng thái khoá và không thoát ra được.

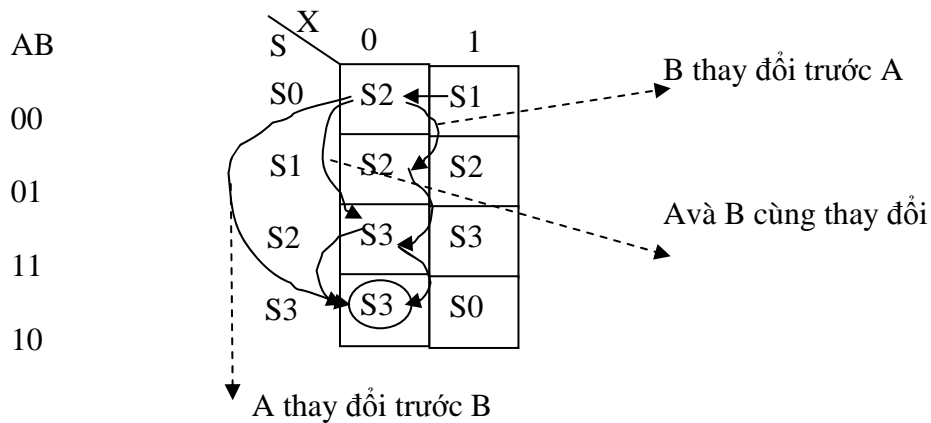
**Ví dụ chạy đua không nguy hiểm:** Một mạch dây không đồng bộ có bảng trạng thái mô tả ở hình sau:

Mạch có thể thay đổi trạng thái theo những con đường khác nhau tùy thuộc vào thứ tự thay đổi của A và B:

- Nếu A và B thay đổi đồng thời mạch sẽ chuyển sang trạng thái S2 rồi mới sang trạng thái S3.

- Nếu B thay đổi trước A thì mạch sẽ lần lượt chuyển qua S1, S2 rồi mới sang S3

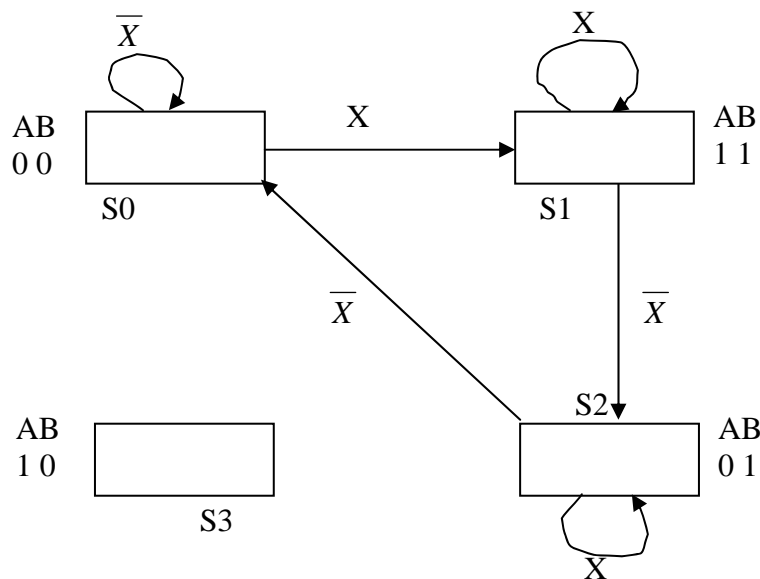
- Nếu A thay đổi trước B mạch sẽ chuyển từ S0 sang S3



Cả 3 con đường chuyển đổi đều dẫn đến cùng một trạng thái ổn định S3. Hiện tượng chạy đua này là chạy đua không nguy hiểm.

Khi mạch đang ở trạng thái ổn định(khoanh tròn), nó chỉ thay đổi trạng thái khi tín hiệu vào thay đổi.

**Ví dụ chạy đua nguy hiểm:** Đồ hình trạng thái của một mạch không đồng bộ mô tả ở hình sau:



Giả thiết trạng thái ban đầu của mạch là S0(AB=00) và tín hiệu vào X=0. Nếu X thay đổi từ 0→1 mạch sẽ chuyển đổi trạng thái như sau:

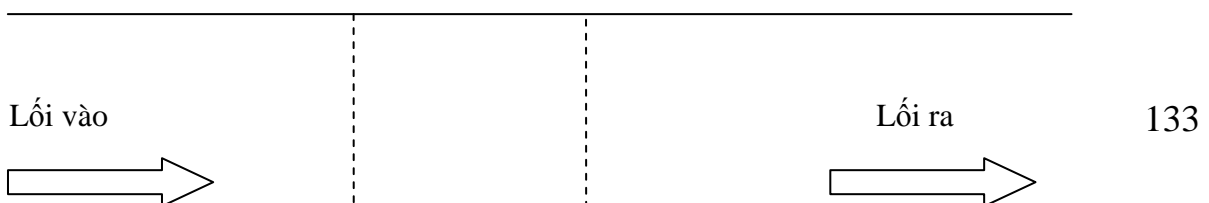
- Nếu A, B thay đổi đồng thời mạch sẽ chuyển đến S1
- Nếu B thay đổi trước A thì mạch sẽ chuyển đến S2
- **Nếu A thay đổi trước B mạch sẽ chuyển đến S3**

ở đây trạng thái S3 là trạng thái khoá, như vậy khi A thay đổi trước B mạch sẽ rơi vào trạng thái khoá và không thoát ra được, chạy đua khi này là chạy đua nguy hiểm

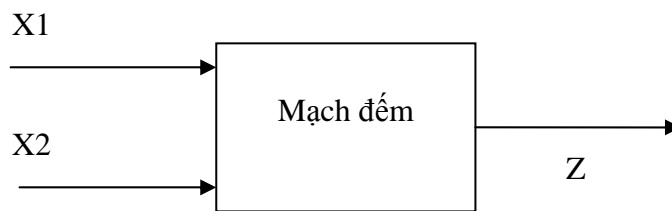
#### IV. CÁC VÍ DỤ MINH HOẠ

##### 1. Ví dụ 1:

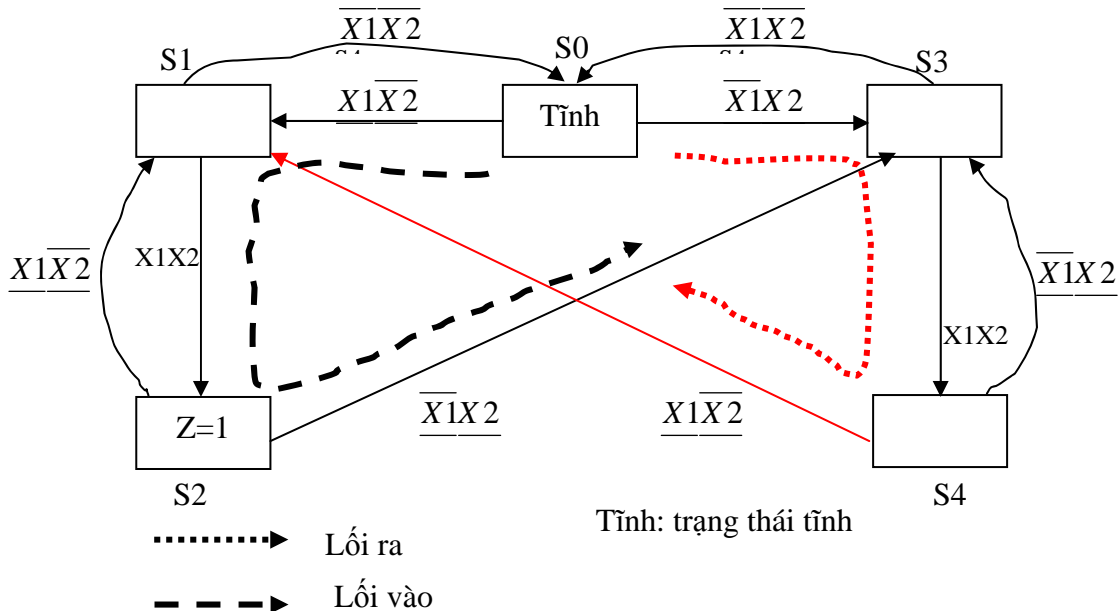
Thiết kế mạch đếm để đếm số người vào thăm một viện bảo tàng, mạch gồm 2 LED sáng X1 và X2 được bố trí như hình vẽ. Mạch thiết kế sao cho mỗi lần đếm được một người



Phân tích: Khi có một người đi vào thì 2 chùm sáng sẽ bị chắn liên tiếp:  
 - Đầu tiên X1 bị chắn, tiếp  
 - Tiếp theo X1 và X2 cùng bị chắn  
 - Sau đó X2 bị chắn, mạch có tín hiệu ra Z=1 (một người đã đi vào)  
 Mô hình mạch đếm như sau:



Quy ước : Chùm sáng bị chắn X; Chùm sáng không bị chắn  $\bar{X}$   
 Ta có đồ hình trạng thái của mạch như sau:



S0 là trạng thái ban đầu của mạch (trạng thái tĩnh)  
 - Nếu một người đi vào sự chuyển đổi trạng thái của mạch sẽ là  $S0 \rightarrow S1 \rightarrow S2 \rightarrow S3 \rightarrow S0$   
 - Nếu một người đi ra sự chuyển đổi trạng thái của mạch sẽ là  $S0 \rightarrow S3 \rightarrow S4 \rightarrow S1 \rightarrow S0$   
 - Nếu một người ngấp ngừng vào sau đó lại đi ra: ban đầu chắn X1, sau đó không chắn X1 nữa, trạng thái mạch:  $S0 \rightarrow S1 \rightarrow S0$  khi này mạch sẽ không đếm.

Lập bảng trạng thái, bảng ra

Trạng thái hiện tại	Trạng thái tiếp theo và tín hiệu ra							
	X1	X2	X1	X2	X1	X2	X1	X2
	0	0	0	1	1	1	1	0
S0	S0 / Z=0		S3 / Z=0					S0 / Z=0
S1	S0 / Z=0				S2 / Z=0			S1 / Z=0
S2			S3 / Z=0		S2 / Z=1			S / Z=0
S3								
S4	S0 / Z=0		S3 / Z=0		S4 / Z=0			
			S3 / Z=0		S4 / Z=0			S1 / Z=0

Bảng có 5 hàng ứng với 5 trạng thái hiện tại có thể xuất hiện và 4 cột, mỗi cột ứng với một tổ hợp giá trị có thể X1, X2. Mỗi ô của bảng biểu diễn trạng thái tiếp theo và tín hiệu ra tương ứng với trạng thái hiện thời và giá trị của tín hiệu vào X1, X2.

Ví dụ hàng đầu tiên của bảng ứng với trạng thái hiện tại là S0, nếu tín hiệu vào là X1X2=01 mạch sẽ chuyển tới trạng thái S3 và tín hiệu ra Z=0. Trong ô ghi rõ S3 và Z=0.

Trong bảng trạng thái, 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 là trạng thái ổn định, ví dụ trạng thái hiện tại là S0 với tín hiệu vào là X1, X2 là 00, trạng thái tiếp theo của mạch vẫn là S0. Điều kiện trạng thái ổn định là  $S'=S$  (S: trạng thái hiện tại, S' trạng thái tiếp theo).

Trên bảng có những ô trống, những ô này tương ứng các tổ hợp tín hiệu không xuất hiện ở đầu vào. Thật vậy do cách sắp xếp của 2 chum sáng X1, X2 đều đang không bị chặn(00) tương ứng với trạng thái ban đầu S0, không thể ngay tức khắc cả 2 chum sáng đều bị chặn(X1X2=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, cụ thể 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.

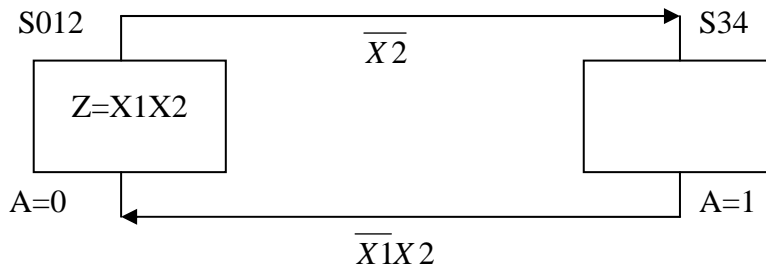
ở bảng trạng thái, bảng ra trên hình trên, các hàng S0, S1, S2 và S3, S4 có các trạng thái tiếp theo và tín hiệu ra tương ứng là giống nhau nếu ta gán:

- Ô trống của hàng đầu tiên(S0) là S2/Z=1
- Ô trống của hàng thứ 2 S3/Z=0
- Ô trống của hàng thứ 4 S1/Z=0
- Ô trống của hàng thứ 3, thứ 5 S0/Z=0

Khi đó ta có bảng tối thiểu như sau:

Trạng thái hiện tại	Trạng thái tiếp theo và tín hiệu ra							
	X1	X2	X1	X2	X1	X2	X1	X2
	0	0	0	1	1	1	1	0
S012	<del>S012</del>	<del>Z=0</del>	<del>S34</del>	<del>Z=0</del>	<del>S012</del>	<del>Z=1</del>	<del>S012</del>	<del>Z=0</del>
S34	<del>S012</del>	<del>Z=0</del>	<del>S34</del>	<del>Z=0</del>	<del>S34</del>	<del>Z=0</del>	<del>S012</del>	<del>Z=0</del>

Đồ hình trạng thái:



Sử dụng 1 bit nhị phân để mã hoá, S012:0; và S34:1

Mạch có tín hiệu ra Z=1 khi X1X2=11 ở trạng thái S012.

Viết phương trình hàm kích cho FF-A ta có:

$$S_A = (\text{Tập hợp bật của } Q) + [(1)] = \overline{AX1X2} + \overline{AX1X2} = \overline{X1X2}$$

$$R_A = (\text{Tập hợp tắt của } Q) + [(0)] = AX2 + \overline{AX1X2} + \overline{AX1X2} = X2$$

Mà phương trình đặc trưng của RS-FF:

$$Q'_A = S_A + \overline{R_A} Q_A \text{ thay giá trị của } R_A, S_A \text{ vào ta có:}$$

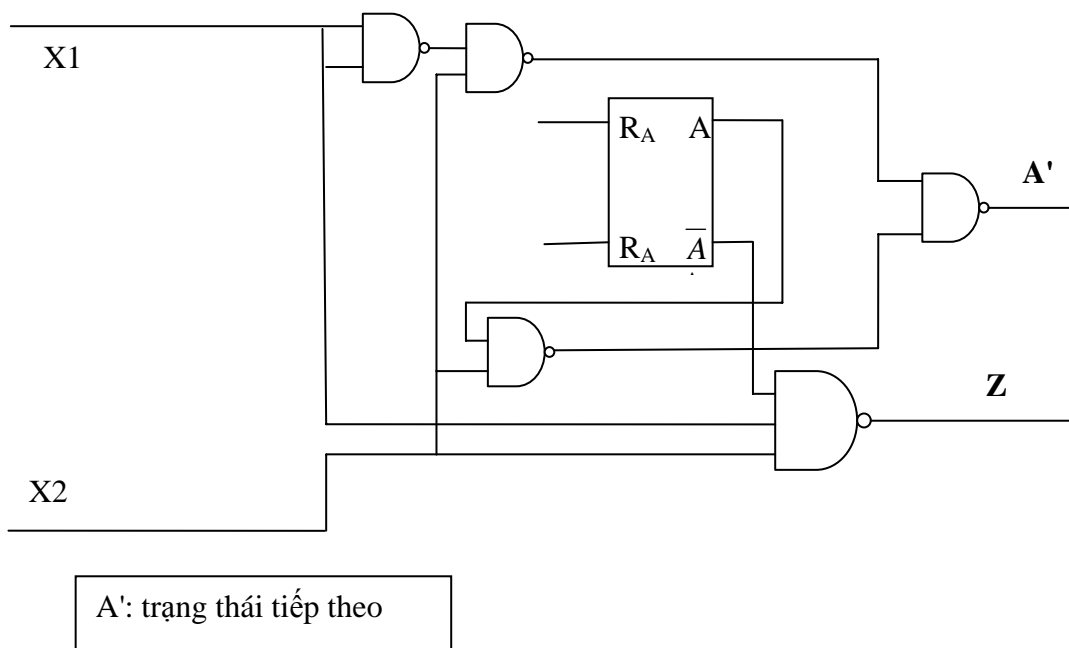
$$A' = \overline{X1X2} + X2A = \overline{X1X2} + X2A = (\overline{X1X2}).(\overline{X2A})$$

Phương trình tín hiệu ra:

$$Z = \overline{AX1X2}$$

Từ đó ta xây dựng được sơ đồ như sau:





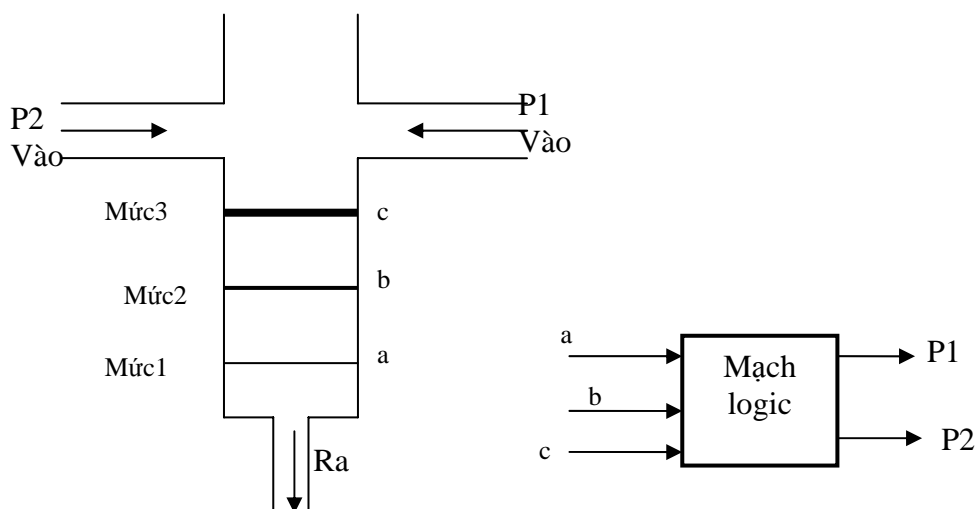
## 2. Ví dụ 2

Thiết kế mạch điều khiển bơm nước vào một ống nước nhờ 2 bơm p1 và P2, cả 2 bơm được mở để bơm nước khi mực nước ở dưới mức 1 và vẫn mở cho đến khi chưa đạt mức 2. Khi vừa đạt mức 2 thì bơm P1 ngắt, còn P2 vẫn bơm. Và P1 vẫn ngắt cho đến khi nước lại ở dưới mức 1, P2 vẫn mở, chỉ khi nước đạt mức3 thì P2 mới ngắt. Và P2 vẫn ngắt, chỉ mở khi nước lại xuống dưới mức 1

### Giải:

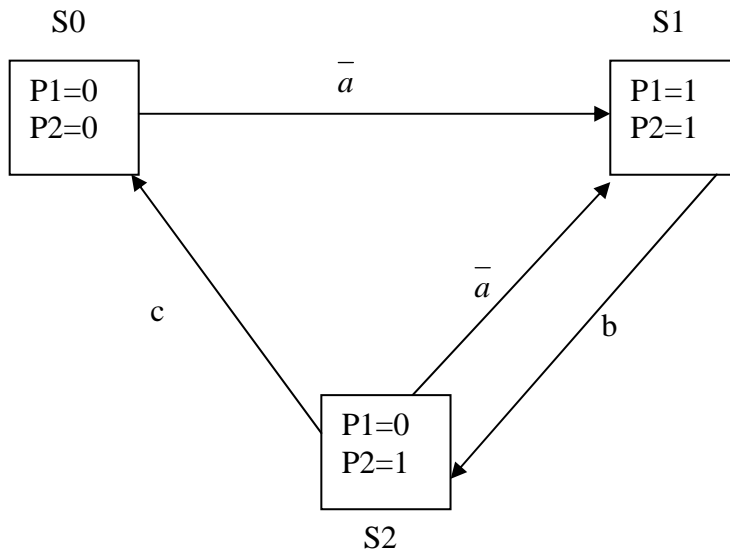
Mã hoá trạng thái:

- + a=1 khi mực nước lớn hơn hoặc bằng mức 1, trường hợp khác a=0
- + b=1 khi mực nước lớn hơn hoặc bằng mức 2, trường hợp khác b=0
- + c=1 khi mực nước lớn hơn hoặc bằng mức 3, trường hợp khác c=0
- + P=1 : Bơm mở; P=0 : bơm đóng



**Đồ hình trạng thái:**

Từ các dữ liệu đầu bài, lập đồ hình trạng thái như sau:



- Trạng thái S0 tương ứng khi nước lớn hơn hoặc bằng mức 3, cả 2 bơm P1 và P2 đều ngắt.
- Khi mức nước tụt xuống mức 1( $\bar{a}$ ), mạch chuyển sang trạng thái S1 lúc đó cả 2 bơm đều mở.
- Khi nước đạt mức 2(b), mạch chuyển sang trạng thái S2 bơm P1 vẫn bơm, P2 tắt. Có 2 khả năng xảy ra
  - + Nếu nước tiếp tục dâng lên ở mức 3(c) mạch sẽ chuyển sang trạng thái S0 và cả 2 bơm đều tắt.
  - + Nếu nước giảm xuống mức 1( $\bar{a}$ ) mạch chuyển sang trạng thái S1; P1 vẫn bơm, P2 tắt.

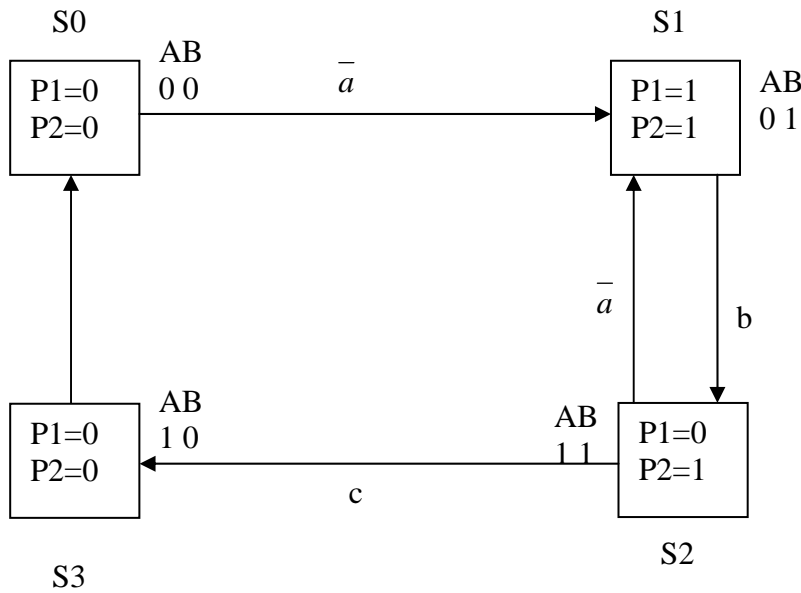
**Bảng trạng thái:**

Từ đồ hình trạng thái ta có bảng trạng thái, do ý nghĩa vật lý của các biến vào, nên chỉ có thể có 4 khả năng sau của abc: 000;100;110;111.

S \ abc	000	100	110	111
S0	S1 P1=1 P2=1	S0 P1=0 P2=0	S0 P1=0 P2=0	S0 P1=0 P2=0
S1	S1 P1=1 P2=1	S1 P1=1 P2=1	S2 P1=0 P2=1	
S2	S1 P1=1 P2=1	S2 P1=0 P2=1	S2 P1=0 P2=1	S0 P1=1 P2=1

**Mã hoá trạng thái:**

Từ bảng trên ta thấy, 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 chỉ cần tránh hiện tượng chạy đua, giải pháp thực hiện bằng việc thêm một trạng thái giả S3:



Hệ phương trình của mạch:

Từ đồ hình trạng thái, ta có hệ phương trình của mạch như sau:

$$S_A = (\text{Tập hợp bật của A}) + [(1)] = b.B$$

$$R_A = (\text{Tập hợp tắt của A}) + [(0)] = \bar{B} + B\bar{a} = \bar{B} + \bar{a}$$

$$S_B = (\text{Tập hợp bật của B}) + [(1)] = \bar{a}A$$

$$R_B = (\text{Tập hợp tắt của B}) + [(0)] = cA$$

Mà phương trình đặc trưng của RS-FF:

$$Q'_A = S_A + \bar{R}_A Q_A \text{ thay giá trị của } R_A, S_A \text{ vào ta có:}$$

$$Q'_A = A' = b.B + (\bar{B} + \bar{a}).A = b.B + a.A.B$$

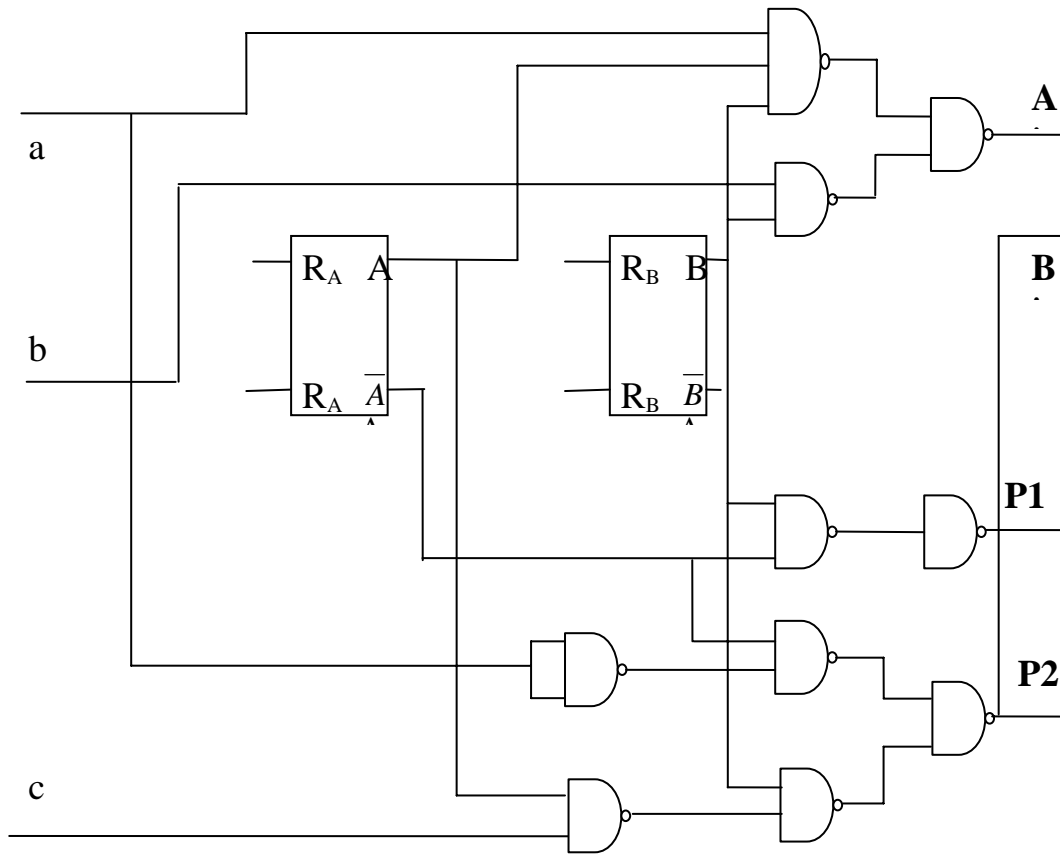
và  $B' = \bar{a}A + \bar{c}.A.B = \bar{a}A + (\bar{c} + \bar{A})B$

Phương trình tín hiệu ra:

$$P1 = \bar{A}.B$$

$$P2 = \bar{A}.B + AB = B$$

Từ đó ta xây dựng được sơ đồ như sau:

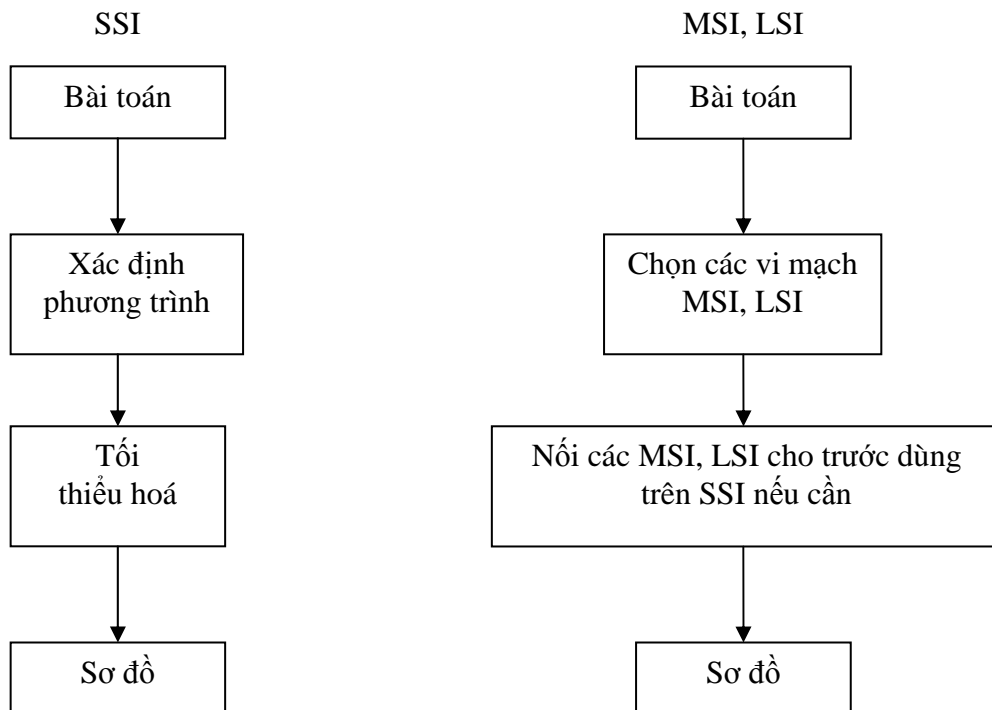


A', B': trạng thái tiếp theo

## CHƯƠNG 13 THIẾT KẾ MẠCH SỐ DÙNG MSI VÀ LSI

### I. KHÁI NIỆM:

Thiết kế mạch số dùng mạch tích hợp cỡ vừa MSI và tích hợp cỡ lớn LSI có một số đặc điểm khác với mạch SSI đã nghiên cứu:

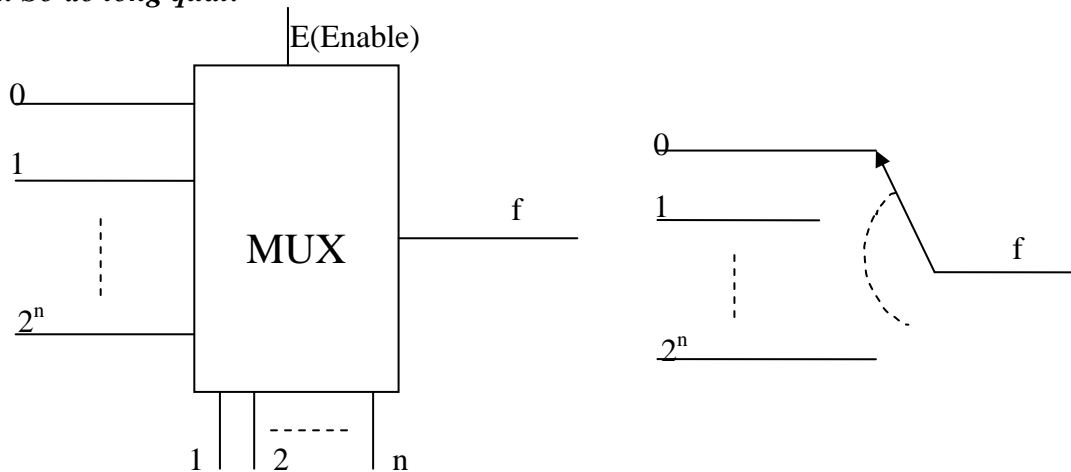


Thiết kế mạch số dùng MSI và LSI có thể không sử dụng hết khả năng của mạch, nhưng lại kinh tế và làm cho mạch tổng thể có kích thước nhỏ hơn nhiều.

Các vi mạch thường dùng dưới dạng MUX, DEMUX, CODER, DECODER, ROM, PLA...Sau đây là các ví dụ

## II. BỘ DÒNG KÊNH MUX

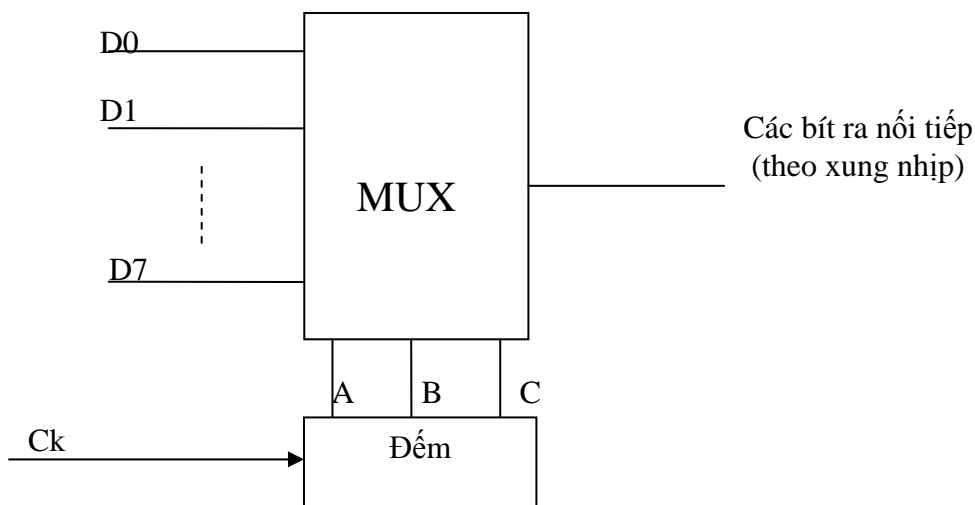
### 1. Sơ đồ tổng quát:



### 2. Các ứng dụng

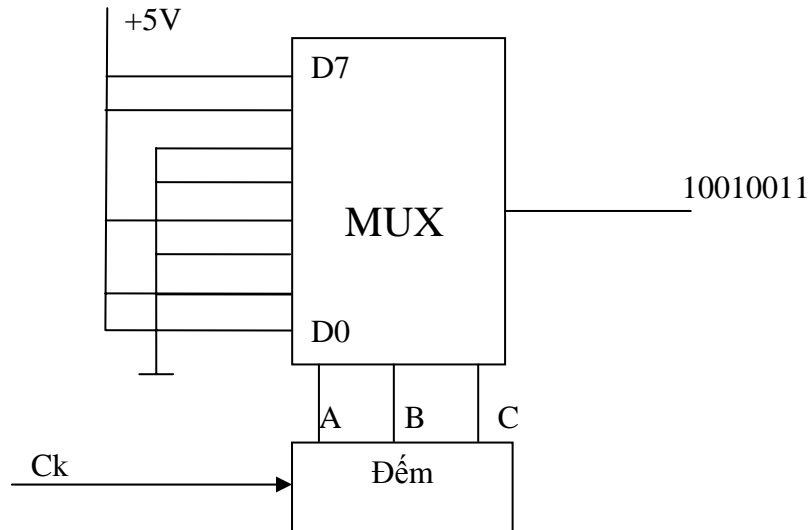
#### a. Chuyển đổi dữ liệu song song thành nối tiếp

Minh họa biến đổi 8 bit song song thành nối tiếp, việc điều khiển đưa tín hiệu ra thực hiện nhờ bộ đếm nhị phân 3 đầu ra



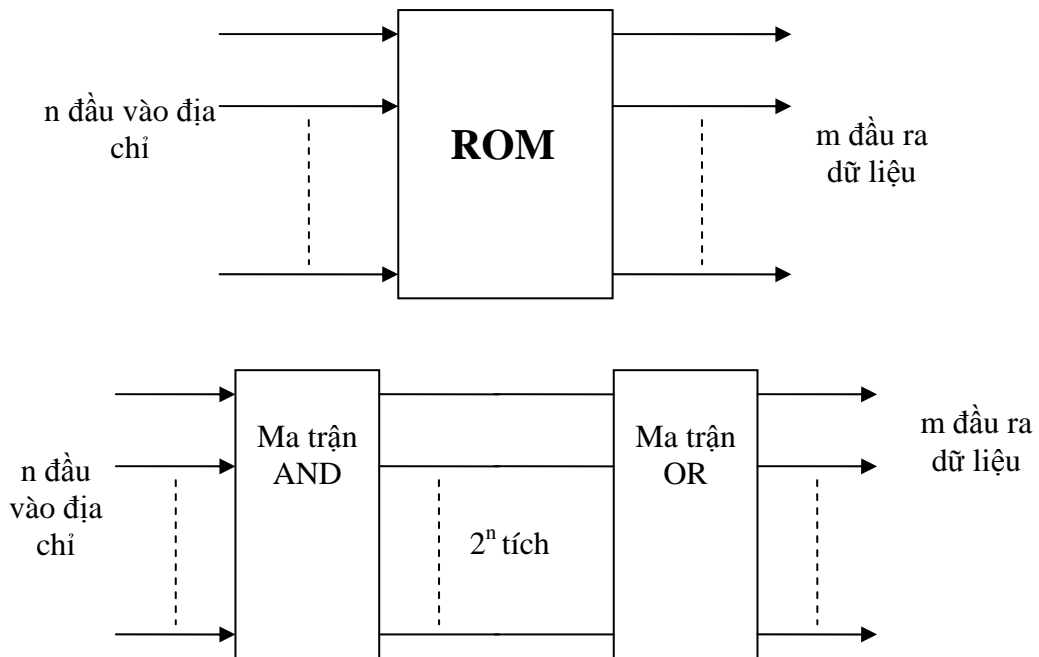
#### b. Mạch tạo dãy tín hiệu tuần hoàn:

Muốn tạo dãy tín hiệu tuần hoàn dùng MUX ta thực hiện nối các đầu vào MUX, và nối với 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 tín hiệu tuần hoàn 10010011:



## II. ROM (READ ONLY MEMORY - BỘ NHỚ CHỈ ĐỌC)

### 1. Sơ đồ cấu tạo:



### 2. ứng dụng

- Dùng làm bộ nhớ trong, lưu các chương trình cố định của máy tính.
- Dùng để thiết kế các mạch tổ hợp

Khi dùng ROM để thiết kế mạch tổ hợp, coi X như địa chỉ vào của ROM và Y như là các số liệu ra của ROM, ROM được sử dụng như một bảng tra cứu (giống bảng chân lý trong thiết kế mạch tổ hợp)

Địa chỉ			Đầu ra							
A	B	C	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8
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

Ví dụ: Dùng ROM để xây dựng bộ giải mã nhị phân sang 7 vạch, có bảng chân lý như trên, các đầu vào được xem là đầu địa chỉ của ROM, các đầu ra 7 vạch chính là đầu ra của ROM.

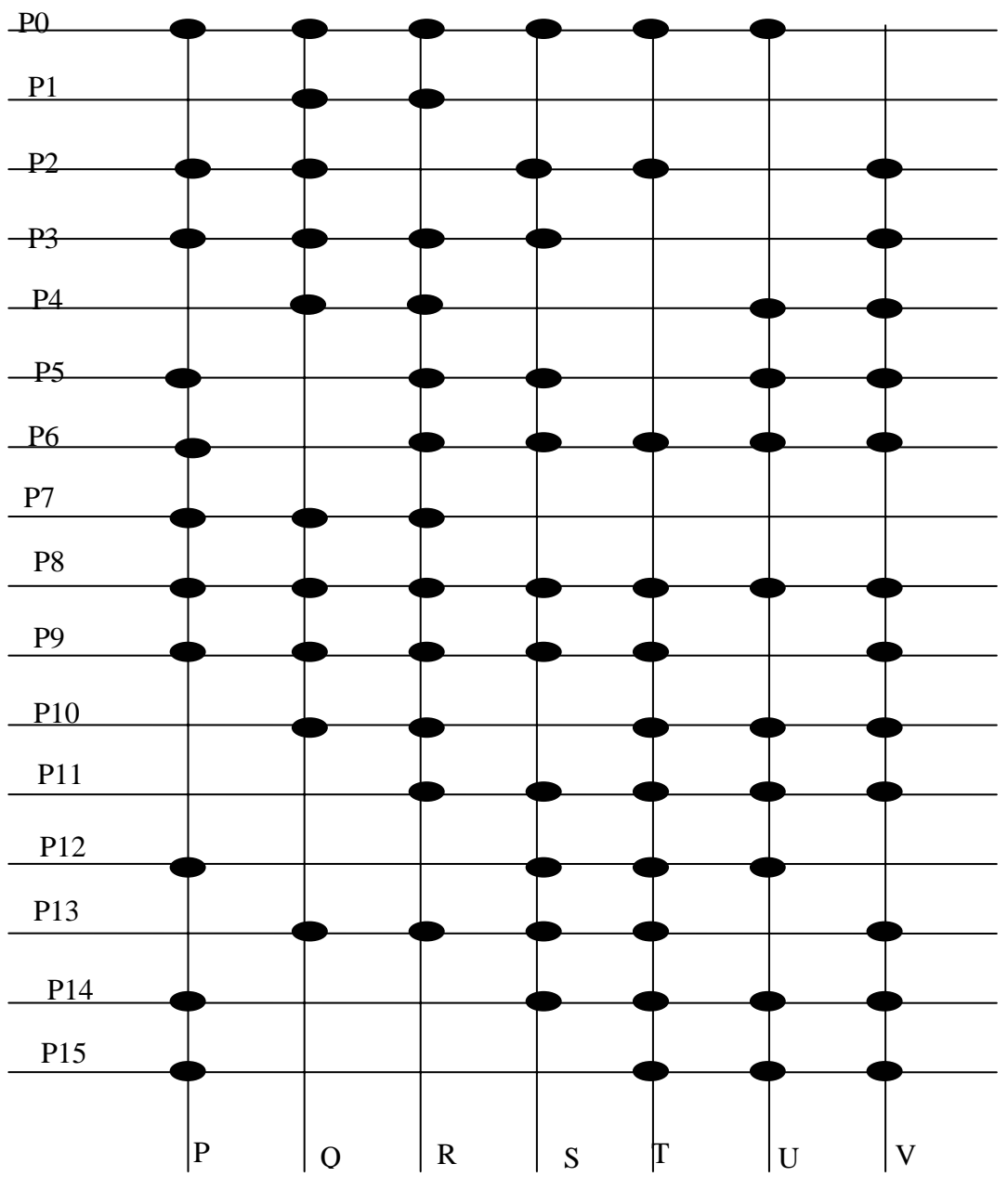
Mã 7 vạch:

Mã	Mã nhị phân				Mã 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	

Ma trận nối như sau:







## Chương 14 : CHUYỂN ĐỔI TƯƠNG TỰ – SỐ

### I. KHÁI NIỆM CHUNG

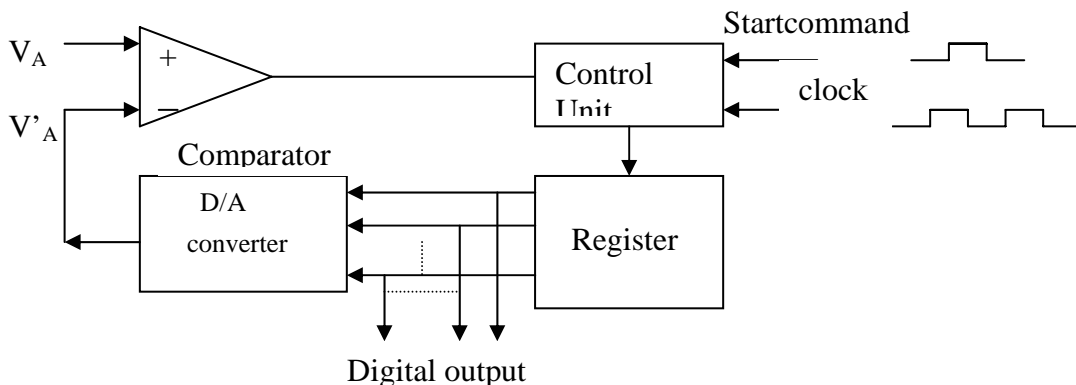
Ngày nay việc truyền đạt tín hiệu cũng như quá trình điều khiển và chỉ thị phần lớn được thực hiện theo phương pháp số. Trong khi đó tín hiệu tự nhiên có dạng tương tự như: nhiệt độ, áp suất, cường độ ánh sáng, tốc độ quay, tín hiệu âm thanh... Để kết nối giữa nguồn tín hiệu tương tự với các hệ thống xử lý số người ta dùng các mạch chuyển đổi tương tự sang số (ADC) nhằm biến đổi tín hiệu tương tự sang số hoặc trong trường hợp ngược lại cần biến đổi tín hiệu số sang tương tự thì dùng các mạch DAC (Digital Analog Converter).

### II. NGUYÊN TẮT THỰC HIỆN CHUYỂN ĐỔI ADC

Mạch chuyển đổi tín hiệu tương tự sang số, chuyển một tín hiệu ngõ vào tương tự (dòng điện hay điện áp) thành dạng mã số nhị phân có giá trị tương ứng.

Chuyển đổi ADC có rất nhiều phương pháp. Tuy nhiên, mỗi phương pháp đều có những thông số cơ bản khác nhau:

- + Độ chính xác của chuyển đổi AD.
- + Tốc độ chuyển đổi .
- + Dải biến đổi của tín hiệu tương tự ngõ vào



Hình 6.1 Sơ đồ khối tổng quát của mạch ADC

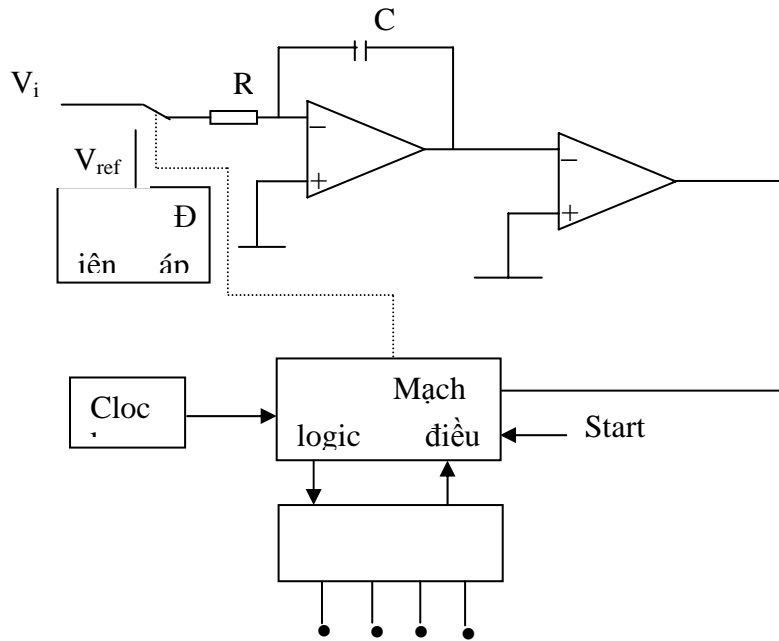
Hoạt động

- Đầu tiên kích xung start để bộ ADC hoạt động
- Tại một tần số được xác định bằng xung clock bộ điều khiển làm thay đổi thành số nhị phân được lưu trữ trong thanh ghi(Register).-Số nhị phân trong thanh ghi được chuyển thành dạng điện áp  $V'_a$  bằng bộ chuyển đổi DA.
- Bộ so sánh, so sánh  $V'_a$  với điện áp ngõ vào  $V_a$ . Nếu  $V'_a < V_a$  thì ngõ ra của bộ so sánh vẫn giữ mức cao. Khi  $V'_a > V_a$  ngõ ra của bộ so sánh xuống mức thấp và quá trình thay đổi số của thanh ghi ngưng. Lúc này  $V'_a$  gần bằng  $V_a$ , những số trong thanh ghi là những số cần chuyển đổi .

### III. CÁC PHƯƠNG PHÁP CHUYỂN ĐỔI AD

#### 1. Phương pháp tích phân (Intergration method)

Phương pháp tích phân cũng giống như phương pháp chuyển đổi ADC dùng tín hiệu dốc đôi (Dual-Slope-ADC). Cấu trúc mạch điện đơn giản hơn nhưng tốc độ chuyển đổi chậm.



Hình 6.2 : Sơ đồ nguyên lý cơ bản của mạch chuyển đổi AD dùng phương pháp tích phân

\* Hoạt động

-Khi có xung start mạch đếm đưa về trạng thái reset. Mạch logic điều khiển khóa K ở vị trí 1, điện áp tương tự  $V_{in}$  được nạp vào tụ điện C với thời hằng  $t_1$  tín hiệu ngõ ra của mạch tích phân giảm dần, và cho đến khi nhỏ hơn 0V thì ngõ ra của bộ so sánh lên mức 1, do đó mạch logic điều khiển mở công cho xung clock vào mạch đếm. Sau khoảng thời gian  $t_1$  mạch đếm tràn mạch logic điều khiển khóa K ở vị trí 0, khi đó điện áp âm  $V_{ref}$  được đưa vào ngõ vào của mạch tích phân, tụ điện C xả điện với tốc độ không đổi, sau khoảng thời gian  $t_2$  tín hiệu ngõ ra của mạch tích phân tăng dần, do đó ngõ ra của mạch so sánh xuống, mức thấp làm cho mạch logic điều khiển đóng công và báo kết thúc chuyển đổi. Trong suốt khoảng thời gian xả điện  $t_2$  mạch đếm vẫn tiếp tục đếm kết quả của mạch đếm cũng chính là tín hiệu số cần chuyển đổi tương ứng với điện áp tương tự ngõ vào  $V_{in}$ .

Mối quan hệ giữa điện áp ngõ vào  $V_{in}$  và điện áp chuẩn  $V_{ref}$  với  $t_1, t_2$

$$t_2 = t_1 \cdot V_{in} / V_{ref}$$

$t_1 = 2^n / f_{ck}$  : thời gian mạch đếm từ 0 đến khi tràn

$t_2 = N / f_{ck}$  : thời gian mạch đếm từ khi tràn đến kết quả sau cùng

-Biểu thức này không phụ thuộc vào thời hằng RC, cũng như số xung clock (nếu mạch làm việc ổn định).

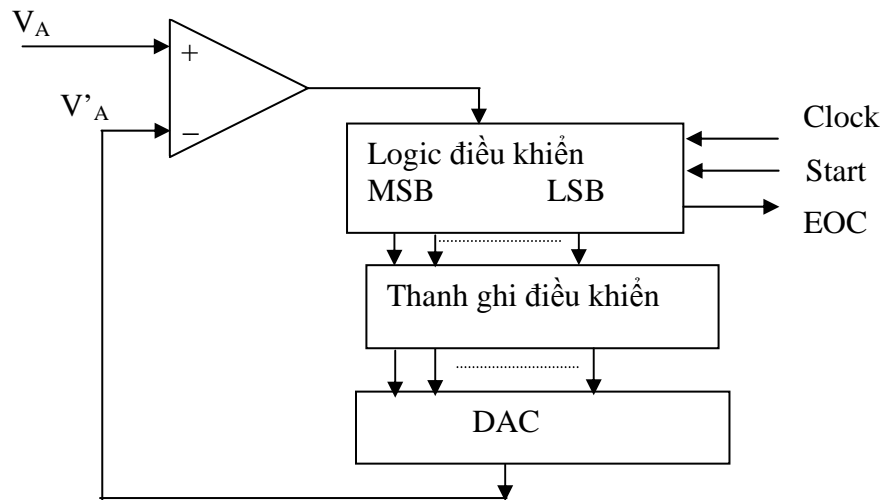
-Các tín hiệu tương tự  $V_{in}$  qua mạch tích phân nên các tín hiệu nhiễu đều bị loại bỏ.

-Nhược điểm của mạch này là thời gian chuyển đổi chậm, giữa  $2^n$  chu kỳ xung clock trong lần lấy tích phân trong thời gian  $t_1$  và  $N$  chu kỳ trong lần lấy tích phân trong thời gian  $t_2$ . Thời gian chuyển đổi lớn nhất khi  $t_1=t_2$ .

Thời gian chuyển đổi:  $T = t_1+t_2$

## 2. Phương pháp ADC xấp xỉ liên tiếp(Successive- Approximation ADC)

Đây là một trong những phương pháp được sử dụng rộng rãi. Tuy nhiên, mạch điện có phức tạp nhưng thời gian chuyển đổi ngắn hơn. Phương pháp chuyển đổi ADC xấp xỉ liên tiếp có thời gian chuyển đổi cố định không phụ thuộc vào điện áp ngõ vào.



Hình 6.3 : Sơ đồ khối chuyển đổi ADC dùng phương pháp xấp xỉ liên tiếp.

\* Hoạt động

Khi tác động cạnh xuống của xung start thì ADC bắt đầu chuyển đổi .

-Mạch logic điều khiển đặt bit có nghĩa lớn nhất(Most Significant Bit) của thanh ghi điều khiển lên mức cao và tất cả các bit còn lại ở mức thấp.Số nhị phân ra ở mạch thanh ghi điều khiển đượ cqua mạch DAC để tạo ra điện áp tham chiếu  $V'_a$ .

Nếu  $V'_a > V_a$  thì ngõ ra bộ so sánh xuống mức thấp ,làm cho mạch logic điều khiển xóa bit MSB xuống mức thấp.

Nếu  $V'_a < V_a$  thì ngõ ra của bộ so sánh vẫn ở mức cao và làm cho mạch logic điều khiển giữ bit MSB ở mức cao.

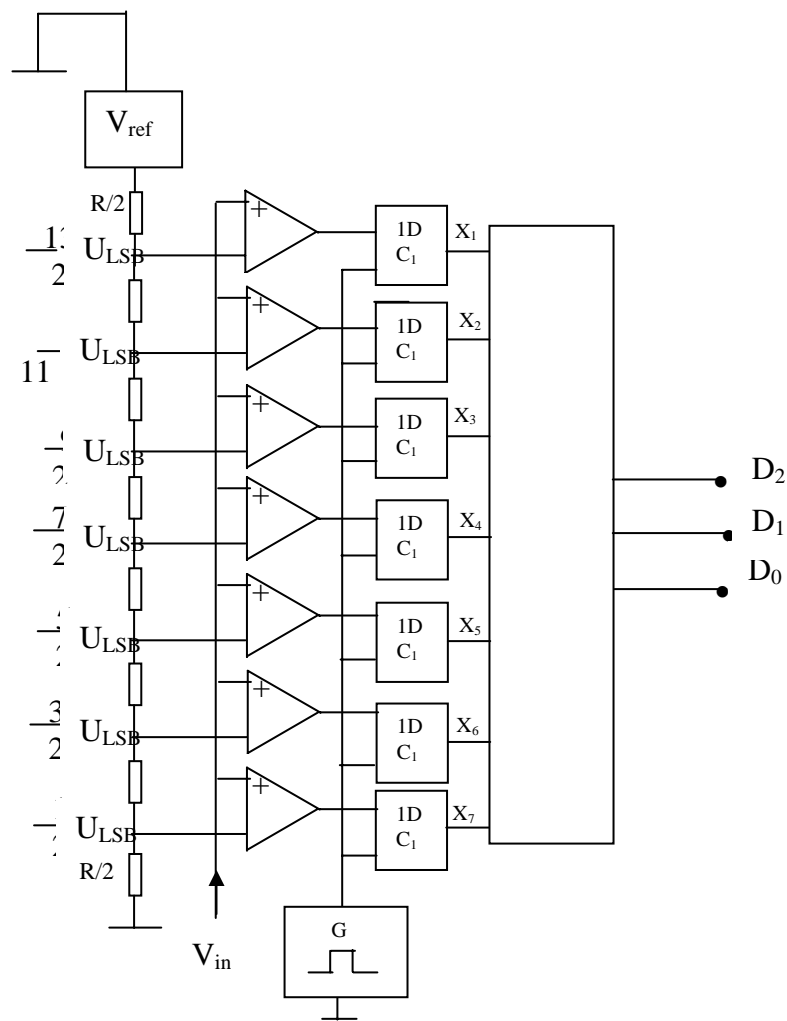
Tiếp theo mạch logic điều khiển đưa bit có nghĩa kế bit MSB lên mức cao và tạo ở ngõ ra khối DAC một điện áp tham chiếu  $v'_a$  rồi đem so sánh tương tự như bit MSB ở trên .Quá trình này cứ tiếp tục cho đến bit cuối cùng trong thanh ghi điều khiển. Lúc đó  $v'_a$  gần bằng  $V_a$  ngõ ra của mạch logic điều khiển báo kết thúc chuyển đổi.

Như vậy mạch đổi ra  $n$  bit chỉ mất  $n$  chu kỳ xung clock nên có thể đạt tốc độ rất cao. Tuy nhiên mạch ADC xấp xỉ liên tiếp lại không thể đáp ứng với tín hiệu tương tự vào biến đổi cực nhanh .

## 3. Phương pháp song song (parallel method)

Mạch ADC dùng nguyên tắc chuyển đổi song song hay còn gọi là phương pháp ADC nhanh, có cấu trúc mạch điện phức tạp nhưng tốc độ chuyển đổi rất cao .

Trong vài trường hợp người ta cần mạch chuyển đổi ADC có tốc độ rất cao vì những tín hiệu biến đổi nhanh nên khi chuyển sang dạng số người ta cần mạch ADC có tốc độ cao .



Hình 6.4 Sơ đồ khối mạch chuyển đổi AD dùng phương pháp song song

\* Hoạt động

Mạch bao gồm: khối so sánh song song và mạch mã hoá. Tín hiệu tương tự được vào các mạch so sánh cùng một lúc, các trạng thái ra của mạch so sánh được đưa vào các flip flop D để đưa đến bộ mã hóa, đầu ra của mạch mã hóa chính là đầu ra của mạch ADC.

Mạch so sánh và mạch mã hóa là loại mạch có tốc độ xử lý rất cao nên tổng thời gian trễ chỉ vài chục ns, nhờ vậy sự chuyển đổi xảy ra rất nhanh. Tuy nhiên với mạch ADC nhanh ở 3 bit thì nó đòi hỏi bảy bộ so sánh khi ở 6 bit thì cần đến 63 bộ so sánh đó là nhược điểm của mạch ADC dùng phương pháp so sánh.

Bảng sự thật của mạch chuyển đổi

Điện áp vào	Ngõ ra bộ so sánh							Tín hiệu số ngõ ra		
	$K_7$	$K_6$	$K_5$	$K_4$	$K_3$	$K_2$	$K_1$	$D_1$	$D_2$	$D_3$
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	1
2	0	0	0	0	0	1	1	0	1	0
3	0	0	0	0	1	1	1	0	1	1
4	0	0	0	1	1	1	1	1	0	0
5	0	0	1	1	1	1	1	1	0	1
6	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	1

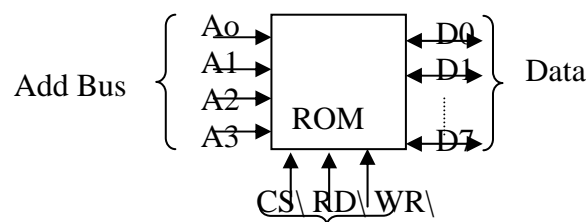
## Chương 3 : KHẢO SÁT BỘ NHỚ BÁN DẪN

Vi điều khiển (Microcontroller) là IC chuyên về xử lý dữ liệu điều khiển theo một chương trình, muốn vi điều khiển thực hiện một công việc gì thì người sử dụng phải lập trình. Chương trình phải được lưu trữ ở một bộ phận nào đó, để vi điều khiển nhận lệnh và thi hành, đôi khi trong lúc xử lý, chương trình của vi điều khiển cần nơi để lưu trữ tạm thời dữ liệu chính của bộ nhớ. Các bộ nhớ của vi điều khiển là các IC, các IC nhớ này có thể đọc dữ liệu ra, ghi dữ liệu vào hoặc chỉ đọc dữ liệu ra. Đôi khi bộ nhớ của vi điều khiển không đủ để lưu trữ những thông tin cần thiết khi chạy chương trình, khi đó phải dùng kỹ thuật mở rộng bộ nhớ.

### I BỘ NHỚ CHỈ ĐỌC (ROM: Read Only Memory)

Loại bộ nhớ này được thiết kế để lưu trữ các dữ liệu cố định. Trong lúc hoạt động bình thường dữ liệu mới không thể nào ghi được vào ROM, mã dữ liệu chỉ đọc ra từ ROM. ROM dùng để lưu trữ các chương trình của máy tính do không bị mất dữ liệu khi mất điện

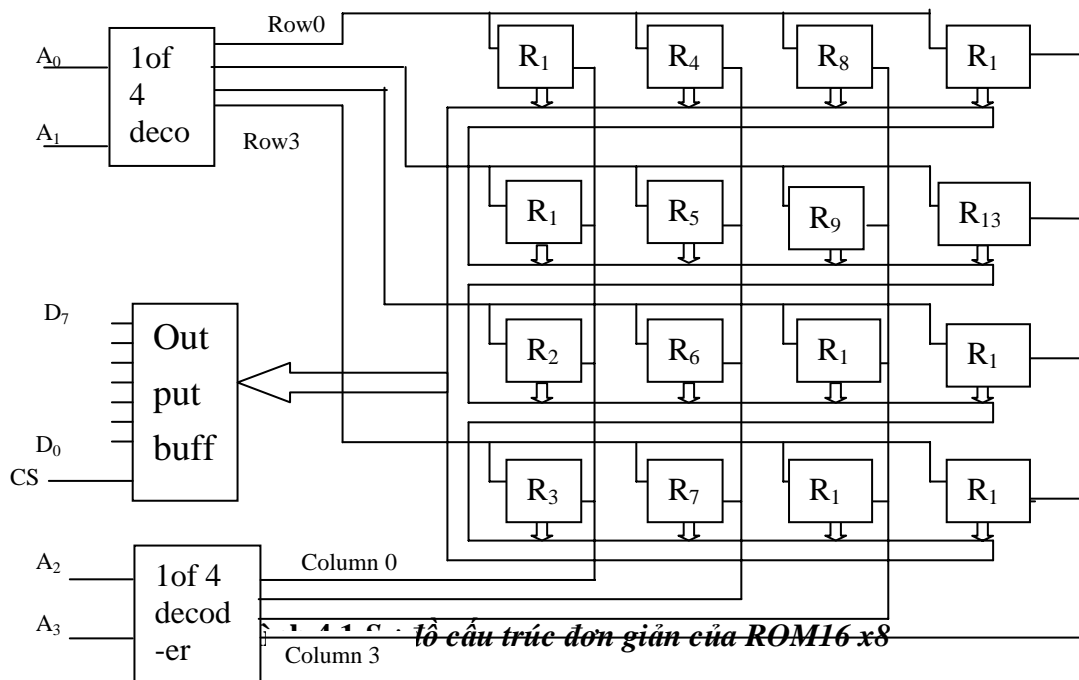
Sơ đồ ROM có dung lượng 32 x4bit



ROM có 3 bus: bus dữ liệu, bus địa chỉ, bus điều khiển. Với bộ nhớ ROM ở trên bus địa chỉ chỉ có 4 đường nên có dung lượng bộ nhớ là  $2^4=16$ . Bus dữ liệu có 8 đường, từ dữ liệu là 8bit hay 1byte, vậy bộ nhớ ROM này có dung lượng là 16byte. Bus điều khiển cho phép ROM hoạt động đọc hay viết, để đọc dữ liệu của ô nhớ nào phải cung cấp địa chỉ của ô nhớ đó tới các ngõ vào địa chỉ tác động đến ngõ vào cho phép CS.



## 1. Cấu trúc bên trong của ROM



Cấu trúc của ROM rất phức tạp, từ sơ đồ trên thì cấu trúc của ROM gồm có 4 phần chính

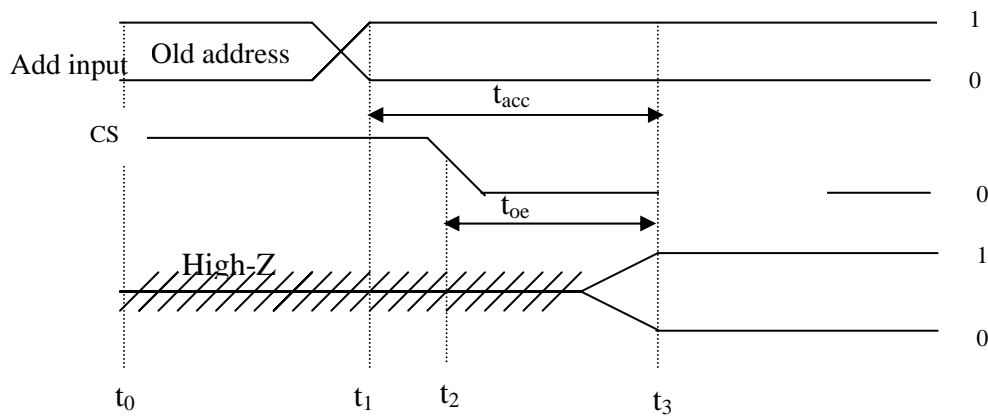
- + Giải mã hàng
- + Giải mã cột
- + Ma trận thanh ghi
- + Đệm ngõ ra
- Ma trận thanh ghi: Lưu trữ dữ liệu đã được lập trình từ ROM, mỗi thanh ghi chứa một từ dữ liệu, như trong trường hợp trên mỗi thanh ghi lưu trữ bốn từ dữ liệu bit. Ngõ ra của từ dữ liệu 8 bit được kết nối với bit dữ liệu bên trong. Mỗi thanh ghi có hai ngõ vào cho phép. Thanh ghi nào có hai ngõ vào cho phép ở mức cao thì dữ liệu sẽ gửi là bus dữ liệu.
- Giải mã địa chỉ: mã địa chỉ  $A_3A_2A_1A_0$  dùng để xác định thanh ghi nào trong ma trận được phép đặt từ dữ liệu 8bit lên bus dữ liệu. Hai bit địa chỉ  $A_0A_1$  được đưa đến bộ giải mã hai đường sang bốn đường để lựa chọn một trong bốn dòng, hai bit địa chỉ  $A_2A_3$  được đưa đến bộ giải mã thứ hai để chọn một trong bốn cột. Chỉ duy nhất một thanh ghi ở trong một hàng và một cột được chọn bởi một địa chỉ ở ngõ vào, và thanh ghi này được phép gửi dữ liệu lên bus.
- Đệm ngõ ra: dữ liệu do thanh ghi gửi ra sẽ được đưa vào bộ đệm, bộ đệm sẽ gửi dữ liệu ra các đường dữ liệu bên ngoài, khi tín hiệu điều khiển CS ở mức cao. Nếu CS ở mức thấp thì bộ đệm ngõ ra ở trạng thái tổng trở cao và các đường dữ liệu  $D_0 - D_7$  sẽ được thả nổi

## 2. Thời hằng truy xuất bộ nhớ ROM

Có một khoảng thời gian từ lúc áp đặt địa chỉ tới các ngõ vào địa chỉ của ROM đến lúc dữ liệu xuất hiện ở ngõ ra (trong lúc ROM hoạt động) thời gian này gọi là thời

gian trễ hay thời gian truy xuất. Khoảng thời gian từ lúc ngõ vào cho phép CS\ đến lúc dữ liệu xuất hiện gọi là thời gian cho phép xuất dữ liệu.

Giản đồ thời hằng truy xuất của Rom



### 3. Các loại bộ nhớ ROM

Maskable Programmed ROM (ROM mặt nạ): đây là loại ROM do nhà sản xuất nạp sẵn chương trình, khi đã nạp chương trình thì các bit trong ROM này không được thay đổi nữa.

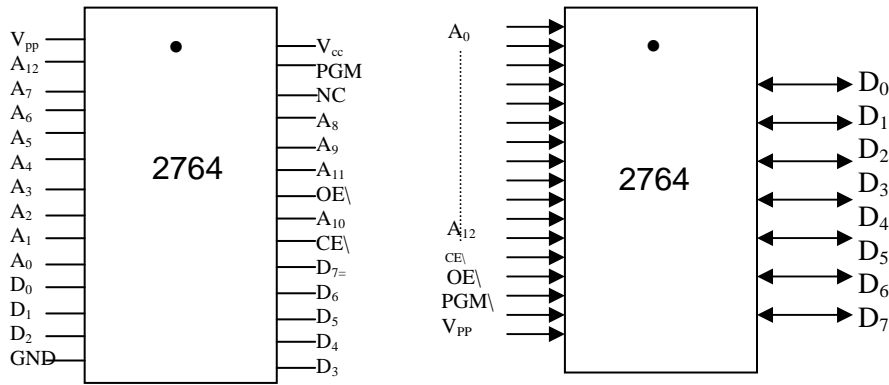
Programmable ROM (PROM): loại ROM này người sử dụng có thể nạp chương trình và chỉ nạp một lần không thể xóa được.

Erasable Programmable ROM (EPROM): loại ROM này có thể lập trình bởi người sử dụng và có thể xóa nạp nhiều lần. Để xóa dữ liệu trong EPROM phải dùng ánh sáng cực tím để xóa, để lập trình cho ROM phải dùng mạch nạp EPROM.

EPROM có hai điểm bất lợi: phải lấy EPROM ra khỏi socket để xóa và lập trình lại khi muốn thay đổi chương trình. Khi muốn thay đổi dữ liệu ô nhớ thì phải xóa dữ liệu của ô nhớ đó, nhưng khi dùng ánh sáng cực tím thì tất cả dữ liệu trong EPROM bị xóa sạch và phải nạp lại toàn bộ dữ liệu.

### 4. Khảo sát bộ nhớ EPROM 2764

Trong các mạch điều khiển dùng vi xử lý PROM được sử dụng rất phổ biến vì nó cho phép người sử dụng có thể nạp và xóa các chương trình dễ dàng theo yêu cầu của mỗi người. EPROM 2764 có dung lượng 8kbyte có sơ đồ chân và sơ đồ logic như sau:



Hình 4.2 Sơ đồ chân và sơ đồ logic EPROM 2764

– EPROM 2764 có 13 đường địa chỉ và 8 đường dữ liệu nên dung lượng của 2764 là  $2^{13}=8192$ byte dữ liệu hay 8kbyte ,có 2 nguồn cung cấp  $V_{cc}$  và  $V_{pp}$  ngõ vào  $V_{cc}$  luôn nối tới nguồn 5v ngõ vào  $V_{pp}$  được nối tới nguồn +5v khi EPROM đang làm việc ở chế độ đọc dữ liệu và nối tới nguồn 26v khi lập trình cho EPROM Hai ngõ vào điều khiển:

$OE\backslash$  được dùng để điều khiển bộ đệm cho phép dữ liệu của EPROM xuất ra ngoài hay không

$CE\backslash$  là ngõ vào cho phép có hai chức năng :khi hoạt động bình thường  $CE\backslash$  là tín hiệu cho phép để đọc dữ liệu từ EPROM,  $CE\backslash$  phải ở mức thấp để mạch điện bên trong lựa chọn dữ liệu và chuyển nó đến output buffer kết hợp với tín hiệu cho  $OE\backslash$  ở mức thấp, thì dữ liệu mới xuất ở các ngõ ra  $D_0-D_7$ . Khi  $CE\backslash$  ở mức cao thì EPROM ở trạng thái chờ (Standby). công suất tiêu tán lúc này 132mw.

Bảng trạng thái làm việc của EPROM

MO		$OE\backslash$	P	V	O
DE	$E\backslash$		$GM\backslash$	pp	utput
REA	$V_{il}$	$V_{il}$	V	V	D
D			ih	cc	out
STA		X	X	V	H
NDBY	ih			cc	ighZ
PRO		X	V	V	D
GAM	il		il	pp	in
PRO		$V_{il}$	V	V	D
GRAM	il		ih	pp	out

VERYFY						
PRO		X	X		V	H
GRAM INHIBIT	ih			pp	cc	ighZ

## II. BỘ NHỚ RAM

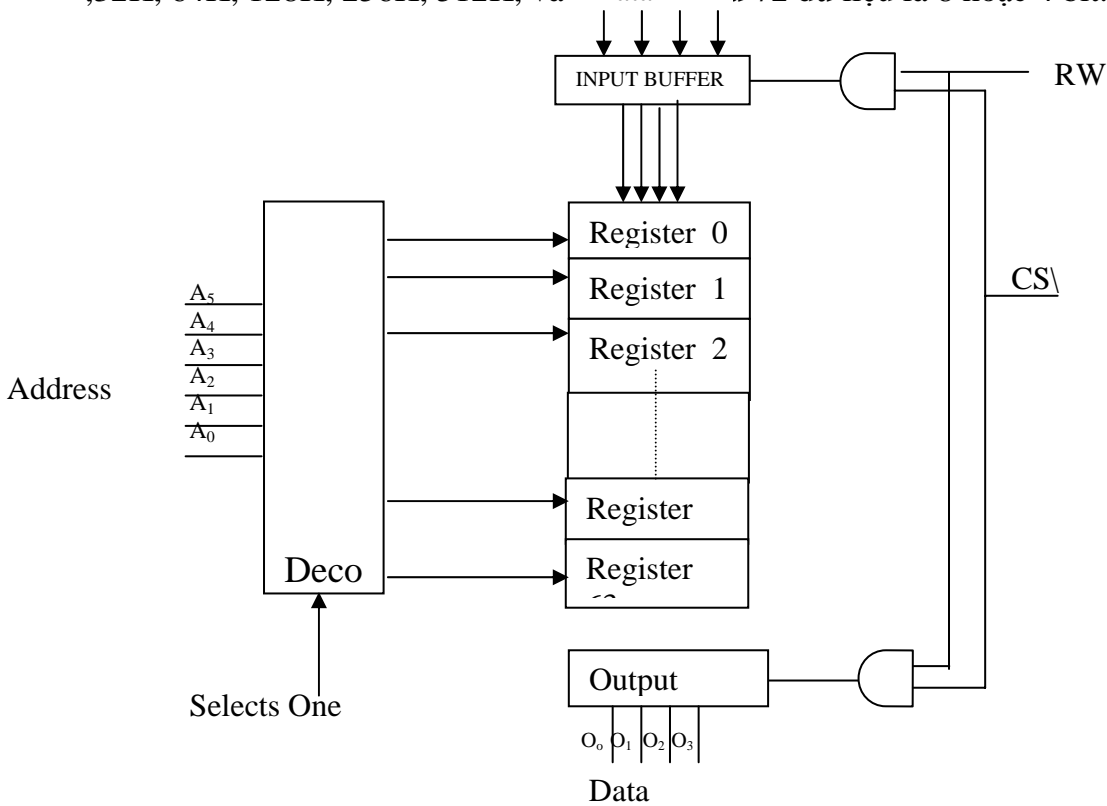
-Ram là bộ nhớ truy xuất ngẫu nhiên, có nghĩa là bất kỳ ô nhớ nào cũng dễ dàng truy xuất như những ô nhớ khác.

-Khuyết điểm của Ram là dữ liệu lưu trữ trong Ram sẽ mất khi mất điện.

-Ưu điểm chính của Ram là có thể đọc và ghi nhanh chóng

### 1. Cấu Trúc Của Ram

Tương tự như bộ nhớ Rom, bộ nhớ Ram cũng gồm có một số thanh ghi. mỗi thanh ghi lưu trữ 1 từ dữ liệu duy nhất và một dữ liệu duy nhất. Dung lượng của bộ nhớ Ram là 1K, 2K, 8K, 16K, 32K, 64K, 128K, 256K, 512K, và Data là 72 dữ liệu là 8 hoặc 4 bit.



Hình 4.5 Sơ đồ cấu trúc bên trong Ram 64x4

#### a. Hoạt động đọc dữ liệu từ Ram

Mã địa chỉ của ô nhớ cần đọc dữ liệu được đưa đến ngõ vào địa chỉ của Ram đồng thời ngõ tín hiệu điều khiển R/W phải ở mức logic 1 và ngõ vào cho phép (CS) phải ở mức logic 1. khi đó dữ liệu mới xuất hiện ở ngõ ra dữ liệu.

Khi R/W=1 sẽ không cho phép bộ đệm ngõ vào, do đó dữ liệu ngõ vào không ảnh hưởng gì đến ô nhớ đang truy xuất.

#### b. Hoạt động ghi dữ liệu lên Ram

Để ghi dữ liệu vào thanh ghi đã được lựa chọn bởi các ngõ vào địa chỉ của bộ nhớ Ram, đòi hỏi ngõ vào R/W=0 và CS=1. Tổ hợp hai mức logic này sẽ cho phép bộ đệm ngõ vào để đưa từ dữ liệu (4bit) ở các ngõ vào sẽ được nạp thanh ghi được chọn

Khi R/W ở mức thấp sẽ không cho phép bộ đệm ngõ ra và ngõ ra ở trạng thái tổng trở cao (trong lúc ghi dữ liệu). Khi ghi dữ liệu vào ô nhớ thì dữ liệu trước đó sẽ mất đi.

**c. Chip select (cs)**

Hầu hết các bộ nhớ đều có hoạt nhiều ngõ vào CS, được dùng để cho phép hoặc không cho phép bộ nhớ hoạt động trong nhiều trường hợp kết nối nhiều bộ nhớ. Khi không cho tất cả các ngõ vào dữ liệu và ngõ ra dữ liệu ở trạng thái tổng trở cao.

**d. Những chân data input-output**

Để giảm số chân cho một IC nhà chế tạo kết hợp 2 chức năng data input và data output thành một chân Input/output, chúng có chức năng của các chân I/O. Khi hoạt động đọc, các chân I/O hoạt động như là các chân xuất dữ liệu. Khi ghi dữ liệu, các chân I/O hoạt động như là các chân dữ liệu.

**2. Các loại Ram**

Ram được chia làm 2 loại:

-SRAM (Static RAM); là một loại linh kiện mà việc lưu trữ dữ liệu dựa vào nguyên tắc hoạt động của flip flop D. Dữ liệu vào tồn tại ở một trong hai trạng thái logic của mạch số.

DRAM (Dynamic Ram); là loại linh kiện nhớ mà dữ liệu lưu trữ như điện tích trữ trong tụ điện.