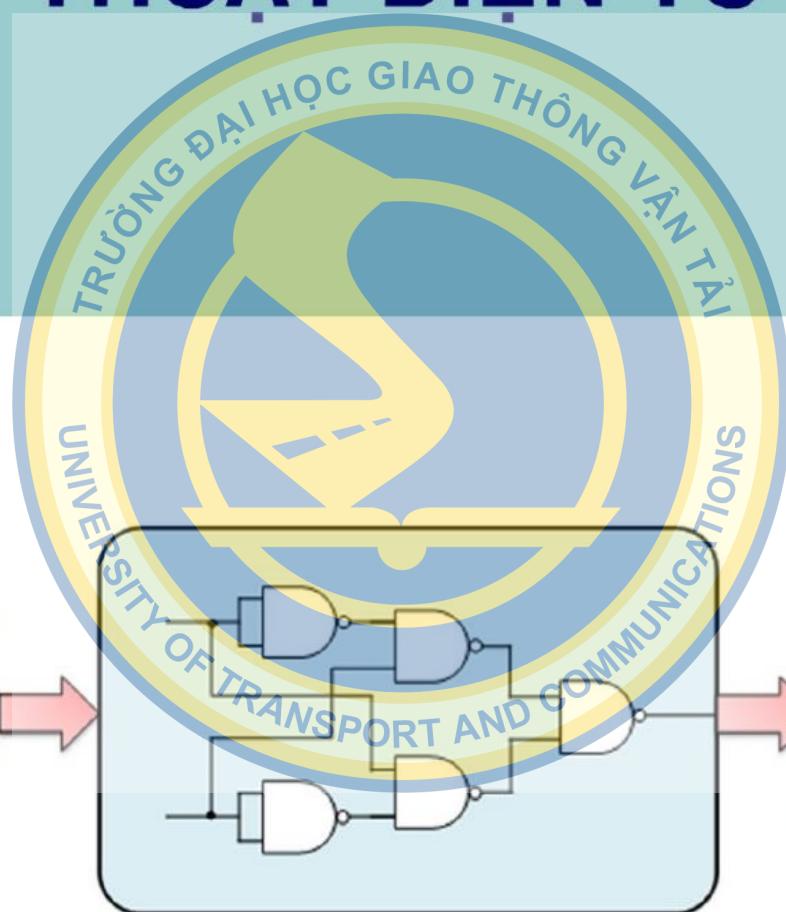


TS. ĐÀO THANH TOẢN (Chủ biên)

ThS. HỒ THÀNH TRUNG

KỸ THUẬT ĐIỆN TỬ SỐ



Phục vụ giảng dạy online 2022

Bộ môn KTĐT: dee.utc.edu.vn



NHÀ XUẤT BẢN GIAO THÔNG VẬN TẢI

Lời nói đầu

Điện tử số là nhánh quan trọng của kỹ thuật điện tử, xử lý các tín hiệu số trên cơ sở đại số logic, là cơ sở để tạo ra các sản phẩm điện tử nhỏ gọn, tiết kiệm năng lượng, và tích hợp nhiều tính năng. Giáo trình kỹ thuật số đề cập tới các nội dung: đại số logic, và mã hóa, chuyển đổi tín hiệu tương tự-số, các họ vi mạch số thực hiện các hàm logic, phương pháp thiết kế, tối ưu mạch điện tử số, xây dựng các mạch tổ hợp thường gặp trong tính toán, mã hóa, và truyền dữ liệu, cũng như các mạch điện tử số dãy có nhớ như thanh ghi dịch, bộ đếm,..

Cuốn sách phục vụ cho sinh viên chuyên ngành Kỹ thuật Điện-Điện tử và là tài liệu tham khảo hữu ích cho các chuyên ngành Kỹ thuật Điện tử, truyền thông và Kỹ thuật điều khiển và tự động hóa... cũng như đọc giả quan tâm đến lĩnh vực điện tử số.

Giáo trình Kỹ thuật Điện tử số do TS. Đào Thanh Toán là chủ biên, đồng thời là tác giả của chương 1,4,5,6,7; chương 2,3, phụ lục do ThS. Hồ Thành Trung biên soạn.

Trong quá trình hoàn thành cuốn giáo trình này, chúng tôi đã nhận được sự giúp đỡ, thảo luận, và góp ý của nhiều đồng nghiệp tại bộ môn Kỹ thuật Điện tử, Khoa Điện-Điện tử, Đại học Giao thông Vận tải, và các đồng nghiệp tại các Trường Đại học khác.

Mặc dù cố gắng sửa chữa, bổ sung cho tài liệu được hoàn chỉnh, nhưng chắc chắn không tránh khỏi những thiếu sót, hạn chế. Chúng tôi mong nhận được thêm góp ý của đọc giả để nâng cao hơn nữa chất lượng giáo trình này. Những ý kiến đóng góp cho chúng tôi xin gửi tới: Bộ môn Kỹ thuật Điện tử, Khoa Điện-Điện tử, Đại học Giao thông Vận tải.

Nhóm tác giả



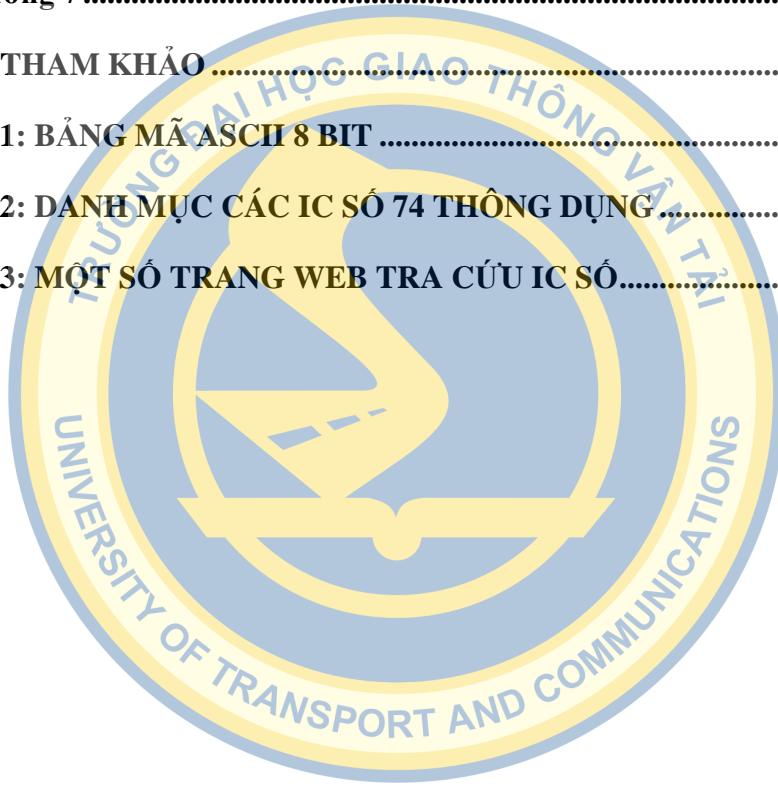
4 * KTĐTS

Mục lục

Lời nói đầu	3
Mục lục	5
Chương 1: GIỚI THIỆU VỀ KỸ THUẬT ĐIỆN TỬ SỐ	9
1.1. Hệ đếm	9
1.2. Chuyển đổi giữa các hệ đếm	10
1.3. Mã nhị phân	15
1.4. Tín hiệu số	23
1.5. Giới thiệu điện tử số.....	24
Ôn tập chương 1.....	27
Chương 2: CHUYỂN ĐỔI TÍN HIỆU TƯƠNG TỰ-SỐ.....	28
2.1. Giới thiệu	28
2.2. Chuyển đổi tương tự/số (ADC)	29
2.3. Chuyển đổi số/tương tự (DAC)	39
Ôn tập chương 2	48
Chương 3: ĐẠI SỐ LOGIC VÀ CÁC CỘNG LOGIC CƠ BẢN	50
3.1. Giới thiệu	50
3.2. Đại số LOGIC	50
3.3. Các công LOGIC	54
3.4. Chuyển đổi công đa dụng thành các công cơ bản.....	62
Ôn tập chương 3	65

Chương 4: VI MẠCH SỐ	66
4.1. Giới thiệu	66
4.2. Các đặc tính của vi mạch số.....	70
4.3. Các họ LOGIC sử dụng DIODE và TRANSISTOR lưỡng cực	74
4.4. Các họ LOGIC sử dụng TRANSISTOR hiệu ứng trường.....	83
4.5. Giao diện họ TTL và CMOS	87
Ôn tập chương 4	89
Chương 5: PHƯƠNG PHÁP THIẾT KẾ MẠCH ĐIỆN TỬ SỐ	90
5.1. Tổng quan	90
5.2. Các bước thiết kế mạch điện tử số	90
5.3. Phân tích mạch tổ hợp.....	103
5.4. Thiết kế sử dụng các cổng đa dụng.....	104
Ôn tập chương 5	112
Chương 6: MẠCH ĐIỆN TỬ SỐ TỔ HỢP THƯỜNG GẶP.....	114
6.1. Giới thiệu.....	114
6.2. Bộ cộng	114
6.3. Mạch trừ.....	117
6.4. Mạch so sánh	120
6.5. Mạch lập mã.....	121
6.6. Mạch giải mã	123
6.7. Mạch ghép kênh và phân kênh	130
6.8. Mạch tạo và kiểm tra chẵn và lẻ	135
Ôn tập chương 6	138

CHƯƠNG 7: MẠCH ĐIỆN TỬ SỐ TUẦN TỰ	139
7.1. Giới thiệu	139
7.2. FLIP-FLOP	140
7.3. Phân tích mạch tuần tự.....	153
7.4. Thiết kế mạch tuần tự	155
7.5. Bộ đếm	160
7.6. Thanh ghi dịch	173
Ôn tập chương 7	183
TÀI LIỆU THAM KHẢO	184
PHỤ LỤC 1: BẢNG MÃ ASCII 8 BIT	185
PHỤ LỤC 2: DANH MỤC CÁC IC SỐ 74 THÔNG DỤNG	195
PHỤ LỤC 3: MỘT SỐ TRANG WEB TRA CỨU IC SỐ	203





Chương 1

GIỚI THIỆU VỀ KỸ THUẬT ĐIỆN TỬ SỐ

1.1. HỆ ĐÉM

Hệ đếm là một tập các ký hiệu (bảng chữ số) để biểu diễn các số và xác định giá trị của các biểu diễn số.

Tổng quát: Nếu một hệ đếm có cơ số B thì một con số bất kỳ trong hệ đếm đó sẽ có giá trị trong hệ thập phân D thông thường với n chữ số bên trái và m chữ số bên phải dấu phẩy như sau:

$$D = a_n B^{n-1} + a_{n-1} B^{n-2} + \dots + a_1 B^1 + a_0 \cdot B^0 + b_1 B^{-1} + b_2 B^{-2} + \dots + b_m B^{-m} \quad (1.1)$$

Trong đó a_n là chữ số có nghĩa lớn nhất (MSD: Most Significant Digit), b_m là chữ số có nghĩa nhỏ nhất (LSD: Least Significant Digit).

Dựa vào công thức 1.1, có thể xây dựng được rất nhiều hệ đếm. Sau đây là một số hệ đếm thông dụng:

+ Hệ đếm mười (decimal numeration, thập phân): có cơ số là 10, các chữ số trong hệ đếm này là: 0, 1, 2, 3, 4, 5, 6, 7, 8 và 9. Đây là hệ đếm được sử dụng phổ biến trong đời sống hàng ngày.

Ví dụ 1.1: Con số $8899 = 8 \cdot 10^3 + 8 \cdot 10^2 + 9 \cdot 10^1 + 9 \cdot 10^0$ biểu diễn tám nghìn tám trăm chín mươi chín đơn vị theo nghĩa thông thường

+ Hệ đếm mười sáu (Hexadecimal numeration, hay còn gọi là thập lục phân): có cơ số 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: F10A trong hệ nhị phân sẽ biểu diễn giá trị

$$A = 15 \cdot 16^3 + 1 \cdot 16^2 + 0 \cdot 16^1 + 10 \cdot 16^0 = 4106 \text{ trong hệ đếm 10.}$$

+ Hệ đếm tám (Octal numeration): có cơ số là 8 với các chữ số 0, 1, 2, 3, 4, 5, 6 và 7

Ví dụ 1.2: con số 325 trong hệ octal biểu diễn giá trị

$$A = 3 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 = 213 \text{ trong hệ đếm 10.}$$

Bảng đổi chiều 16 con số đầu tiên trong các hệ đếm trên được tổng hợp ở bảng 1.1.

+ Hệ đếm hai (Binary numeration, hay còn gọi là nhị phân): có cơ số là 2, các chữ số trong hệ đếm này là 0 và 1.

Ví dụ 1.3: 1101 trong hệ nhị phân sẽ biểu diễn giá trị

$$A = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13 \text{ trong hệ đếm 10.}$$

Đây là hệ đếm được sử dụng trong mạch điện tử số, thiết bị kỹ thuật số, máy tính,..

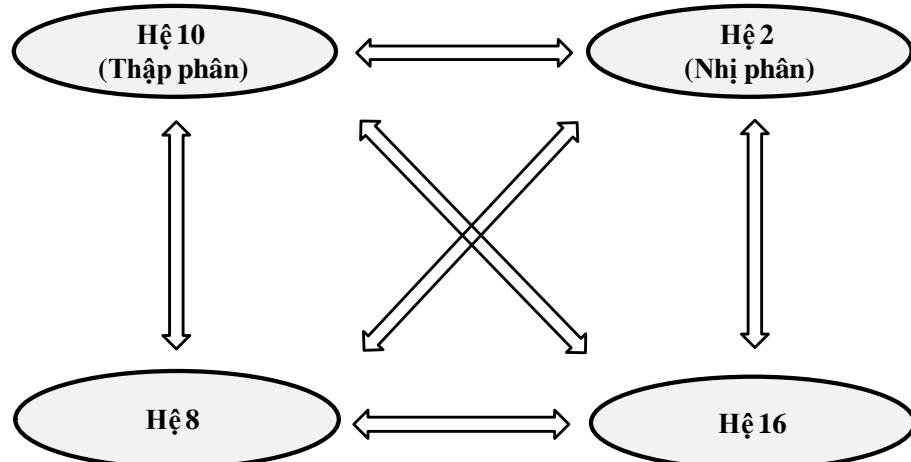
Bảng 1.1. Số và ký hiệu tương ứng trong các hệ đếm.

Số (Number)	Hệ 10 (Decimal)	Hệ 2 (Binary)	Hệ 8 (Octal)	Hệ 16 (Hexadecimal)
Không	0	0	0	0
Một	1	1	1	1
Hai	2	10	2	2
Ba	3	11	3	3
Bốn	4	100	4	4
Năm	5	101	5	5
Sáu	6	110	6	6
Bảy	7	111	7	7
Tám	8	1000	10	8
Chín	9	1001	11	9
Mười	10	1010	12	A
Mười một	11	1011	13	B
Mười hai	12	1100	14	C
Mười ba	13	1101	15	D
Mười bốn	14	1110	16	E
Mười lăm	15	1111	17	F
Mười sáu	16	10000	20	10
.....

Từ bảng 1.1 ta thấy, cùng là ký hiệu nhưng lại có nghĩa khác nhau khi được sử dụng với hệ đếm khác nhau, ví dụ ký hiệu “10” trong hệ thập phân nghĩa là “mười” (đọc là mười), còn trong hệ nhị phân là “hai” (đọc là một-không)...

1.2. CHUYỂN ĐỔI GIỮA CÁC HỆ ĐÉM

Chuyển đổi giữa các hệ đếm là cần thiết trong kỹ thuật, ví dụ: mạch điện tử số, hệ thống máy tính ngày nay làm việc với hệ đếm nhị phân, trong khi đời sống hàng ngày sử dụng hệ thập phân. Về nguyên tắc tất cả các hệ đếm có thể chuyển đổi lẫn nhau như mô tả ở hình 1.2.



Hình 1.2. Mô tả chuyển đổi giữa các hệ đếm thường dùng.

1.2.1. Hệ thập phân và hệ khác

a, **Hệ khác → hệ 10**

Chuyển từ hệ bất kỳ sang hệ 10, thực hiện theo công thức (1.1), đã trình bày ở mục 1.1.

b, **Hệ 10 → hệ khác**

Chuyển đổi một số thập phân cho một số trong B thì ta chia liên tục số thập phân đó cho B, cho đến khi thương số bằng 0, kết quả là tập hợp các số dư viết từ số dư cuối cùng (là MSD) đến số dư đầu tiên (là LSD).

Ví dụ 1.4: Chuyển đổi $(456)_{10}$ sang các hệ 2, 8, và 16

- Sang hệ 16

Phép chia thứ	Kết quả	Số dư
$456/16$	28	8
28/16	1	12
1/16	0	1

Như vậy $(456)_{10} = (1C8)_{16}$

- Sang hệ 8

Phép chia thứ	Kết quả	Số dư
$456/8$	57	0
57/8	7	1
7/8	0	7

Như vậy $(456)_{10} = (710)_8$

- Sang hệ 2

Phép chia thứ	Kết quả	Số dư
456/2	228	0
228/2	114	0
114/2	57	0
57/2	28	1
28/2	14	0
14/2	7	0
7/2	3	1
3/2	1	1
1/2	0	1

Ta được $(456)_{10} = (111001000)_2$

1.2.2. Hệ 2 và hệ khác

a, Hệ 2 → hệ khác

- **Hệ 2 sang hệ cơ số 8:** Tạo các nhóm 3 bit bắt đầu từ bit có ý nghĩa nhỏ nhất LSB, sau đó chuyển đổi sang hệ cơ số 8.

Ví dụ 1.5: Chuyển đổi $(101001101)_2$ sang các hệ 8

Bắt đầu từ số LSB, tạo các nhóm 3 bit: 101 001 101

Tương ứng trong hệ 8 : 5 1 5

Vậy $(101001101)_2 = (515)_8$

- **Hệ 2 sang hệ 10:** Chuyển từ hệ thập phân sang các hệ khác thực hiện theo công thức (1.1), đã trình bày ở mục 1.1.

- **Hệ 2 sang hệ 16:** Tạo các nhóm 4 bit bắt đầu từ bit có ý nghĩa nhỏ nhất LSB, sau đó chuyển đổi sang hệ 16

Ví dụ 1.6: Chuyển đổi $(101001101)_2$ sang các hệ 16

Bắt đầu từ bit LSB, tạo nhóm 4 bit : 1 0100 1101

Ở trường hợp này, cần thêm các số 0, để tạo đủ các nhóm 4 số: 0001 0100 1101

Tương ứng trong hệ 16 : 1 4 D

Vậy $(101001101)_2 = (14D)_{16}$

b, Hệ khác → hệ 2

Quá trình chuyển đổi từ hệ khác sang hệ 2 thực hiện ngược lại

- **Hệ 8 sang hệ 2:** Ánh xạ theo nhóm 3 bit

Ví dụ chuyển đổi (26)₈ sang các hệ 2

Tương ứng trong hệ 8: 2 6

Tương ứng trong hệ 2: 010 110

Vậy (26)₈ = (010110)₂

- *Hệ 10 sang hệ 2:* Theo nguyên tắc chia liên tiếp cho 2 như đã trình bày ở mục 1.2.1

- *Hệ 16 sang hệ 2:* Ánh xạ theo nhóm 4 bit

Ví dụ chuyển đổi (FE15)₁₆ sang các hệ 2

Tương ứng trong hệ 16: F E 1 5

Tương ứng trong hệ 2: 1111 1110 0001 0101

Vậy (FE15)₁₆ = (1111111000010101)₂

1.2.3. Chuyển đổi hệ 8 và 16

- *Hệ 8 sang 16:* có thể chuyển đổi qua hệ trung gian như hệ 2 hay hệ 10, chẳng hạn sử dụng hệ 2 làm hệ trung gian:

- + Chuyển đổi hệ 8 sang hệ 2.
- + Tạo các nhóm 4 bit bắt đầu từ bit có ý nghĩa nhỏ nhất LSB.
- + Chuyển đổi tương đương sang hệ 16.

Ví dụ 1.7: Chuyển đổi (26)₈ sang các hệ 16

Hệ 8 : 2 6

Tương ứng trong hệ 2: 010 110

Tạo các nhóm 4 bit: 0001 0110

Tương ứng trong hệ 16: 1 6

Vậy (26)₈ = (16)₁₆

- *Hệ 16 sang 8:*

+ Chuyển đổi hệ 16 sang hệ 2.

+ Tạo các nhóm 3 bit bắt đầu từ bit có ý nghĩa nhỏ nhất LSB.

+ Chuyển đổi tương đương sang hệ 8.

Ví dụ 1.8: Chuyển đổi (4A)₁₆ sang các hệ 8

Hệ 16 : 4 A

Tương ứng trong hệ 2: 0100 1010

Tạo các nhóm 3 bit: 001 001 010

Tương ứng trong hệ 8: 1 1 2

Vậy (4A)₁₆ = (112)₈

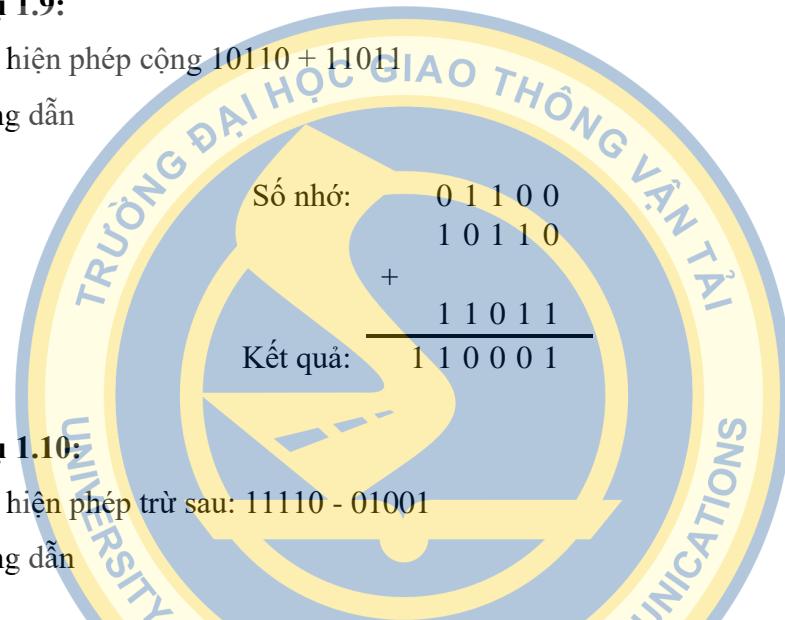
1.2.4. Phép tính với số nhị phân

Phép cộng	Phép trừ	Phép nhân	Phép chia
$0 + 0 = 0$	$0 - 0 = 0$	$0 \times 0 = 0$	$0 : 0 = X$ (cấm)
$0 + 1 = 1$	$1 - 1 = 0$	$0 \times 1 = 0$	$0 : 1 = 0$
$1 + 0 = 1$	$1 - 0 = 1$	$1 \times 0 = 0$	$1 : 0 = X$ (cấm)
$1 + 1 = 0$ (nhớ 1)	$0 - 1 = 1$ (mượn 1)	$1 \times 1 = 1$	$1 : 1 = 1$

Ví dụ 1.9:

Thực hiện phép cộng $10110 + 11011$

Hướng dẫn



Ví dụ 1.10:

Thực hiện phép trừ sau: $11110 - 01001$

Hướng dẫn



Ví dụ 1.11:

Thực hiện phép nhân sau: 110×010

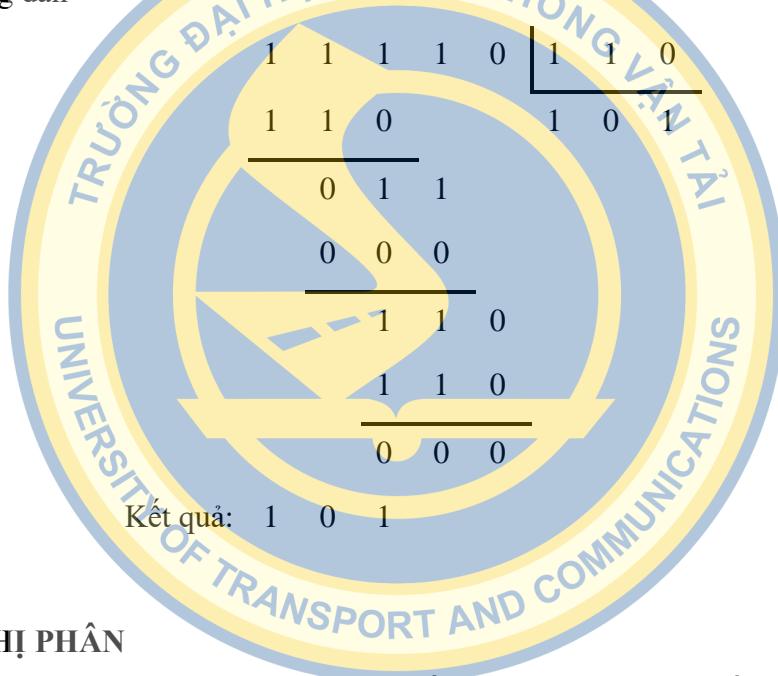
Hướng dẫn

$$\begin{array}{r}
 1 \quad 1 \quad 0 \\
 \times \\
 0 \quad 1 \quad 1 \\
 \hline
 1 \quad 1 \quad 0 \\
 1 \quad 1 \quad 0 \\
 0 \quad 0 \quad 0 \\
 \hline
 \text{Kết quả: } 1 \quad 0 \quad 0 \quad 1 \quad 0
 \end{array}$$

Ví dụ 1.12:

Thực hiện phép chia sau: $11110 : 110$

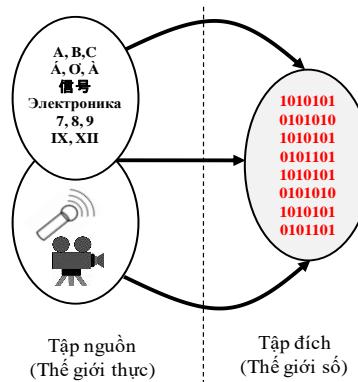
Hướng dẫn



1.3. MÃ NHỊ PHÂN

Mã hóa nhị phân là phép ánh xạ một đối tượng từ tập hợp nguồn, là tập số, tập ký tự, âm thanh, hình ảnh,... sang tập hợp đích, chứa các tổ hợp các số nhị phân. Tổ hợp các số nhị phân tương ứng với một số gọi là từ mã. Tập hợp các từ mã cho ta một bộ mã.

Mạch điện tử số làm việc với không chỉ với chữ số, chữ cái, ký tự. Ngoài việc sử dụng để mã hóa các đối tượng như con số, chữ cái, ký tự,... (quá trình số hóa) các từ mã còn được sử dụng trong việc biểu diễn các trạng thái bên trong của mạch điện tử số, mã lệnh (opcode) của các bộ vi xử lý.



Hình 1.2. Mô tả mã hóa nhị phân.

1.3.1 Mã có trọng số

- Mã nhị phân:

Với n bit, ta có thể mã hóa được 2^n trường hợp khác nhau, trong vị trí của một bit gắn với trọng số nhất định.

$$a_n 2^{n-1} + a_{n-1} 2^{n-2} + \dots + a_1 2^1 + a_0 2^0 \quad (1.2)$$

Bảng 1.2. Mã nhị phân có trọng số

Giá trị thập phân	n=1	n=2	n=3	n=4
	1	2 1	4 2 1	8 4 2 1
0	0	0 0	0 0 0	0 0 0 0
1	1	0 1	0 0 1	0 0 0 1
2		1 0	0 1 0	0 0 1 0
3		1 1	0 1 1	0 0 1 1
4			1 0 0	0 1 0 0
5			1 0 1	0 1 0 1
6			1 1 0	0 1 1 0
7			1 1 1	0 1 1 1
8				1 0 0 0
9				1 0 0 1
10				1 0 1 0
11				1 0 1 1
12				1 1 0 0
13				1 1 0 1
14				1 1 1 0
15				1 1 1 1

- Mã ASCII and Unicode

Mã ASCII (American Standard Code for Information Interchange: Mã tiêu chuẩn trao đổi thông tin của Hoa Kỳ), Mã ASCII là mã 7 bit, nên có $2^7 = 128$ nhóm mã, đủ để biểu thị tất cả ký tự của một bàn phím chuẩn cũng như các chức năng điều khiển trong máy tính và các thiết bị thông tin khác làm việc với văn bản. Bảng mã ASCII mở rộng sử dụng 8 bit để mã hóa. Tức là gồm 255 ký tự bao gồm cả 128 ký tự trong mã ASCII chuẩn, và các ký tự sau là các phép toán, các chữ có dấu và các ký tự để trang trí. Chi tiết về Bảng mã ASCII được trình bày ở phụ lục 1.

Unicode (mã thông nhất) là bộ mã chuẩn quốc tế được thiết kế để dùng làm bộ mã duy nhất không những bao gồm ngôn ngữ sử dụng bảng chữ cái latin, mà còn cho tất cả các ngôn ngữ khác nhau trên thế giới, kể cả các ngôn ngữ sử dụng ký tự tượng hình phức tạp như tiếng Trung, tiếng Nhật,... hiện nay mã Unicode sử dụng 32 bit để mã hóa (UTF 32). Sự ra đời của mã Unicode đáp ứng được yêu cầu về sự trao đổi thông tin toàn cầu và mạng xã hội hiện nay.

- Mã BCD (Binary-Coded Decimal: số thập phân được mã hóa bởi nhị phân) dùng số nhị phân 4 bit tương ứng thay thế cho từng số hạng trong số thập, các số thập phân lớn hơn 9 sẽ được mã hóa bằng ghép các tổ hợp từ 0→9. Ngoài mã BCD với trọng số 8 4 2 1, còn có mã BCD trọng số 4 2 2 1 và 5 4 2 1, như thống kê ở bảng 1.3.

Bảng 1.3. Các loại mã BCD.

Thập phân	Mã BCD		
	8 4 2 1	4 2 2 1	5 4 2 1
0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1	0 0 1 1
4	0 1 0 0	1 0 0 0	0 1 0 0
5	0 1 0 1	0 1 1 1	1 0 0 0
6	0 1 1 0	1 1 0 0	1 0 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0
Mã BCD 8421 còn gọi là NBCD (mã BCD bình thường)			

Ví dụ 1.13: 13 trong mã BCD 8421 là: 0001 0011

487 trong mã BCD 8421 là: 0100 1000 0111

Mã BCD sử dụng nhiều bit hơn số nhị phân nhưng quá trình biến đổi đơn giản hơn, có thể dễ dàng chuyển đổi từ thập phân sang nhị phân và ngược lại. Chỉ cần nhớ các nhóm mã 4 bit ứng với các ký số thập phân từ 0 đến 9.

1.3.2 Mã không trọng số

Thông dụng nhất là: Mã dư 3, Gray, Johnson,

- *Mã dư 3*: được tạo thành bằng cách cộng thêm 3 đơn vị (11) vào mã nhị phân.

Bảng 1.4. Tạo mã dựng mã dư 3 với $n = 1, 2, 3, 4$.

n=1		n=2		n=3		n=4		Giá trị thập phân
Nhị phân	Dư ba	Nhị phân	Dư ba	Nhị phân	Dư ba	Nhị phân	Dư ba	
0	Không lập mã	0 0	1 1	0 0 0	0 1 1	0 0 0 0	0 0 1 1	0
1		0 1	0 0	0 0 1	1 0 0	0 0 0 1	0 1 0 0	1
		1 0	0 1	0 1 0	1 0 1	0 0 1 0	0 1 0 1	2
		1 1	1 0	0 1 1	1 1 0	0 0 1 1	0 1 1 0	3
				1 0 0	1 1 1	0 1 0 0	0 1 1 1	4
				1 0 1	0 0 0	0 1 0 1	1 0 0 0	5
				1 1 0	0 0 1	0 1 1 0	1 0 0 1	6
				1 1 1	0 1 0	0 1 1 1	1 0 1 0	7
						1 0 0 0	1 0 1 1	8
						1 0 0 1	1 1 0 0	9
						1 0 1 0	1 1 0 1	10
						1 0 1 1	1 1 1 0	11
						1 1 0 0	1 1 1 1	12
						1 1 0 1	0 0 0 0	13
						1 1 1 0	0 0 0 1	14
						1 1 1 1	0 0 1 0	15

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*: là một loại mã không có trọng số, trong đó tổ hợp mã cạnh nhau chỉ khác nhau một bit, được giới thiệu vào năm 1947 bởi Frank Gray, tại Bell Labs, Hoa Kỳ. Có một số phương pháp để lập mã, nhưng cách đơn giản nhất là chúng tôi gọi là phương pháp soi gương, có nghĩa là khi xây dựng các tổ hợp ứng với số bit là n , các tổ hợp ứng với $n-1$ bit sẽ được liệt kê, sau đó các tổ hợp tiếp theo sẽ được thiết lập bằng cách soi gương, cuối cùng để hình thành đầy đủ các bít trong tổ hợp, nửa tổ hợp đầu sẽ thêm bit “0” còn nửa tổ hợp sau thì thêm bit “1” vào bên trái.

Bảng 1.5. Tạo mã dùng mã đư 3 với $n = 1, 2, 3, 4$.

$n = 1$	$n = 2$		$n = 3$			$n = 4$		Thập phân
0	0	0 0	0 0	0 0 0	0 0 0	0 0 0 0	0 0 0 0	0
1	1	0 1	0 1	0 0 1	0 0 1	0 0 0 1	0 0 0 1	1
“Gương”	1	1 1	1 1	0 1 1	0 1 1	0 0 1 1	0 0 1 1	2
	0	1 0	1 0	0 1 0	0 1 0	0 0 1 0	0 0 1 0	3
	Viết lại tổ hợp của $n=1$ ở nửa trên, các bit nửa dưới thêm bằng cách “soi gương”.	Nửa trên thêm bit “0”, Nửa trên thêm bit “1”.	1 0	1 1 0	1 1 0	0 1 1 0	0 1 1 0	4
			1 1	1 1 1	1 1 1	0 1 1 1	0 1 1 1	5
			0 1	1 0 1	1 0 1	0 1 0 1	0 1 0 1	6
			0 0	1 0 0	1 0 0	0 1 0 0	0 1 0 0	7
				1 0 0	1 1 0 0		1 1 0 0	8
				1 0 1	1 1 0 1		1 1 0 1	9
				1 1 1	1 1 1 1		1 1 1 1	10
				1 1 0	1 1 1 0		1 1 1 0	11
				0 1 0	1 0 1 0		1 0 1 0	12
				0 1 1	1 0 1 1		1 0 1 1	13
				0 0 1	1 0 0 1		1 0 0 1	14
				0 0 0	1 0 0 0		1 0 0 0	15

Mã Gray được sử dụng để tạo bảng Karnaugh trong quá trình tối thiểu hóa hàm logic, sử dụng trong truyền dẫn tín hiệu số vì có khả năng hạn chế nhiễu, và sử dụng để địa chỉ hóa bộ nhớ chương trình trong máy tính nhằm giảm công suất tiêu thụ.

+ Mã Johnson: là khi chuyển sang mã số kẽ tiếp sẽ thay 0 bằng 1 bắt đầu từ trái sang phải tới khi đạt toàn bit 1 sẽ bắt đầu thay 1 bằng 0 và cũng theo chiều từ trái sang phải. Mã Johnson được sử dụng trong biểu diễn các trạng thái của mạch đếm vòng xoắn (xem mục chương 5).

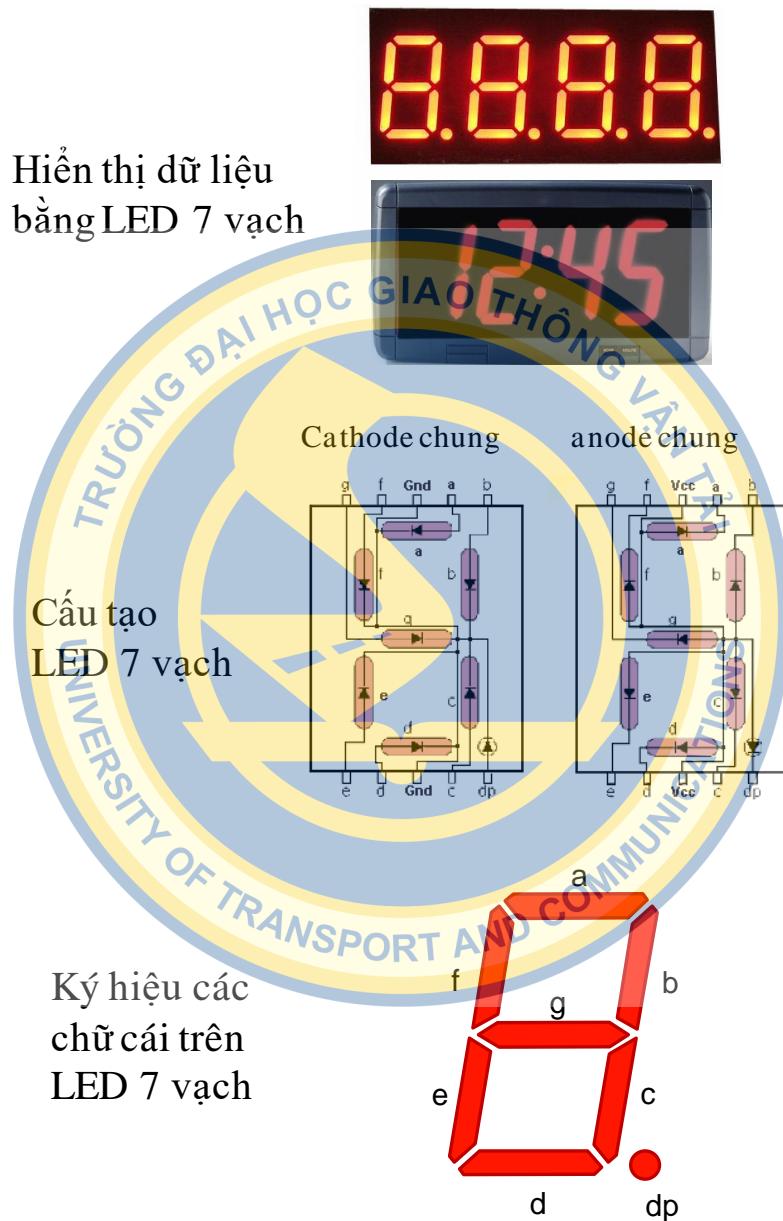
Bảng 1.6. Bảng mã Johnson với $n = 1-8$.

n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	Giá trị thập phân
0	00	000	0000	00000	000000	0000000	00000000	0
1	10	100	1000	10000	100000	1000000	10000000	1
	11	110	1100	11000	110000	1100000	11000000	2
	01	111	1110	11100	111000	1110000	11100000	3
	011	1111	11110	111100	1111000	11110000	111100000	4
	001	0111	11111	111110	1111100	11111000	111110000	5
		0011	01111	111111	1111110	11111110	111111100	6
		0001	00111	011111	1111111	11111111	111111110	7
			00011	001111	0111111	01111111	111111111	8
			00001	000111	0011111	00111111	011111111	9
				000011	0001111	00111111	001111111	10
				000001	0000111	00011111	000111111	11
					0000011	00001111	000011111	12
						1000000	00000111	13
							00000011	14
							00000001	15

1.3.3 Mã bảy vạch (seven-segment code)

Về cơ bản các 7 LED đơn được ghép thành cấu trúc số 8 và thêm 1 LED là dấu chấm thập phân (decimal point:dp). như hình 1.3. Các thiết bị hiển thị 7 vạch có hai loại tùy theo kiểu làm việc của LED:

- + Cathode chung, điều khiển bởi mức logic dương (LED sáng: “1”, LED tối: “0”)
- + Anode chung, điều khiển bởi mức logic âm (LED sáng: “0”, LED tối: “1”)



Hình 1.3. LED 7 vạch: một số dạng hiển thị thông tin, cấu tạo LED 7 vạch kiểu cathode và anode chung, và ký hiệu chữ cái tạo mã của một LED 7 vạch.

LED 7 vạch có thể biểu diễn tới 16 ký tự trong đó có 10 số và 6 chữ cái với bảng mã tông hợp ở bảng 1.7.

Bảng 1.7. Bảng mã 7 vạch với Cathode chung và Anode chung.

Hệ 16	LED 7 vạch	Cathode chung	Anode chung
		a b c d e f g	a b c d e f g
0		1 1 1 1 1 1 0	0 0 0 0 0 0 1
1		0 1 1 0 0 0 0	1 0 0 1 1 1 1
2		1 1 0 1 1 0 1	0 0 1 0 0 1 0
3		1 1 1 1 0 0 1	0 0 0 0 1 1 0
4		0 1 1 0 0 1 1	1 0 0 1 1 0 0
5		1 0 1 1 0 1 1	0 1 0 0 1 0 0
6		1 0 1 1 1 1 1	0 1 0 0 0 0 0
7		1 1 1 0 0 0 0	0 0 0 1 1 1 1
8		1 1 1 1 1 1 1	0 0 0 0 0 0 0
9		1 1 1 0 0 1 1	0 0 0 1 1 0 0
A		1 1 1 1 1 0 1	0 0 0 0 0 1 0
B		0 0 1 1 1 1 1	1 1 0 0 0 0 0
C		0 0 0 1 1 0 1	1 1 1 0 0 1 0
D		0 1 1 1 1 0 1	1 0 0 0 0 1 0
E		1 1 0 1 1 1 1	0 0 1 0 0 0 0
F		1 0 0 0 1 1 1	0 1 1 1 0 0 0

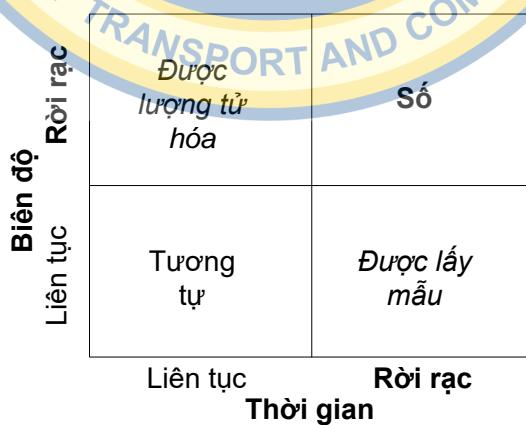
Bảng 1.8. Tổng hợp 16 tổ hợp đầu tiên của các mã thường gặp.

Giá trị thập phân	Mã nhị phân	Mã dư 3	Mã Gray	Mã Johnson	7 vạch (Cathode chung)
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0 0 0 0	1 1 1 1 1 1 0
1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 0 0 0 1	0 1 1 0 0 0 0
2	0 0 1 0	0 1 0 1	0 0 1 1	0 0 0 0 0 0 1 1	1 1 0 1 1 0 1
3	0 0 1 1	0 1 1 0	0 0 1 0	0 0 0 0 0 1 1 1	1 1 1 1 0 0 1
4	0 1 0 0	0 1 1 1	0 1 1 0	0 0 0 0 1 1 1 1	0 1 1 0 0 1 1
5	0 1 0 1	1 0 0 0	0 1 1 1	0 0 0 1 1 1 1 1	1 0 1 1 0 1 1
6	0 1 1 0	1 0 0 1	0 1 0 1	0 0 1 1 1 1 1 1	0 0 1 1 1 1 1
7	0 1 1 1	1 0 1 0	0 1 0 0	0 1 1 1 1 1 1 1	1 1 1 0 0 0 0
8	1 0 0 0	1 0 1 1	1 1 0 0	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1
9	1 0 0 1	1 1 0 0	1 1 0 1	1 1 1 1 1 1 1 0	1 1 1 0 0 1 1
10	1 0 1 0	1 1 0 1	1 1 1 1	1 1 1 1 1 1 0 0	1 1 1 1 1 0 1
11	1 0 1 1	1 1 1 0	1 1 1 0	1 1 1 1 1 1 0 0	0 0 1 1 1 1 1
12	1 1 0 0	1 1 1 1	1 0 1 0	1 1 1 1 0 0 0 0	0 0 0 1 1 0 1
13	1 1 0 1	0 0 0 0	1 0 1 1	1 1 1 0 0 0 0 0	0 1 1 1 1 0 1
14	1 1 1 0	0 0 0 1	1 0 0 1	1 1 0 0 0 0 0 0	1 1 0 1 1 1 1
15	1 1 1 1	0 0 1 0	1 0 0 0	1 0 0 0 0 0 0 0	1 0 0 0 1 1 1

1.4. TÍN HIỆU SỐ

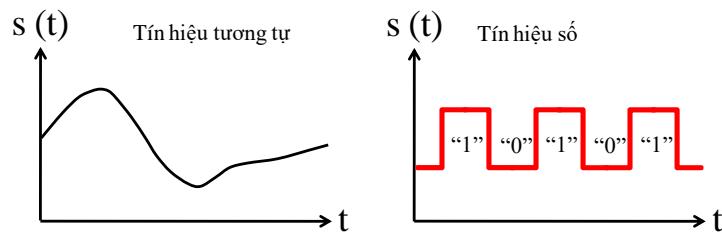
1.4.1. Tín hiệu tương tự và số

Tín hiệu được chia thành các dạng như sau tùy thuộc vào tính liên tục hay rời rạc của biên độ và thời gian.



Hình 1.4. Mối quan hệ giữa tham số của tín hiệu và loại tín hiệu.

- Tín hiệu tương tự (Analog signal) là tín hiệu biến đổi liên tục theo thời gian
- Tín hiệu số (Digital signal) là tín hiệu là tín hiệu lượng tử hoá, rời rạc theo thời gian và được mã hoá dưới dạng số.

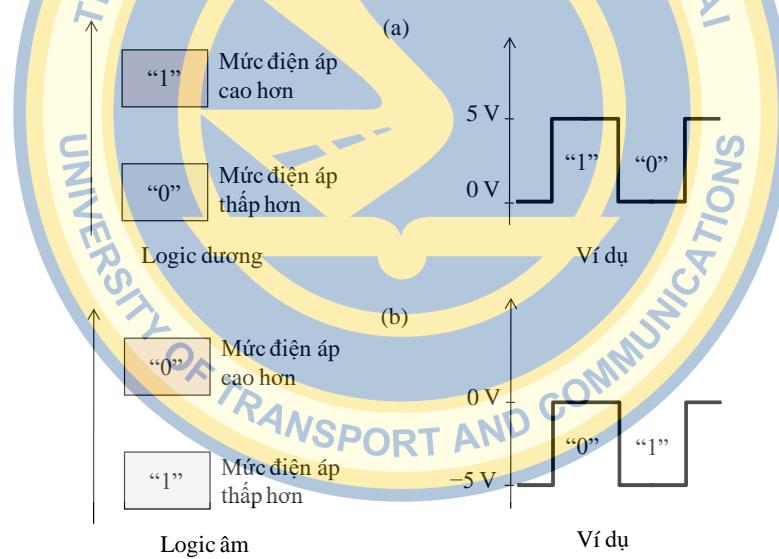


Hình 1.5. Dạng tín hiệu tương tự và số.

1.4.2. Các dạng biểu diễn tín hiệu số:

- Hệ logic dương: mức logic cao có điện áp hay dòng điện lớn hơn mức thấp và giá trị tín hiệu cao tương ứng logic 1 mức thấp có giá trị tín hiệu thấp tương ứng logic 0.
- Hệ logic âm: ngược lại, mức cao tương ứng logic 0. Mức thấp tương ứng logic 1.

Hệ logic dương được sử dụng phổ biến trong thực tế và hệ logic này cũng được chúng tôi sử dụng trong toàn bộ giáo trình này.



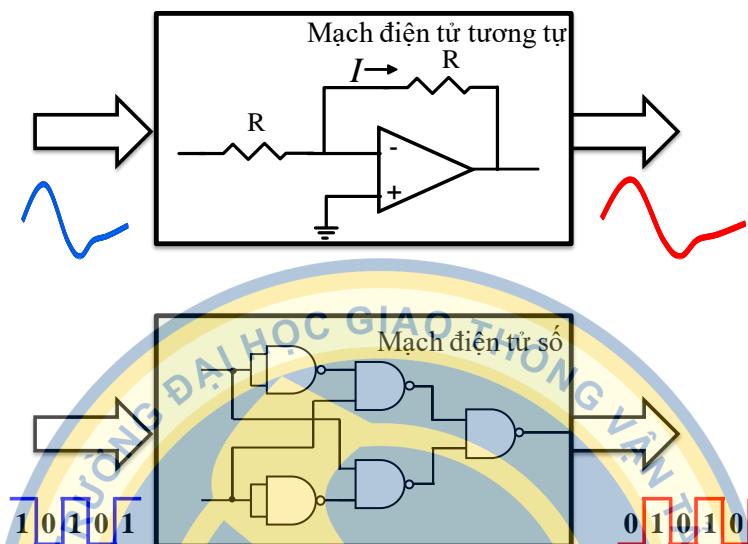
Hình 1.6. Logic dương và âm.

1.5. GIỚI THIỆU ĐIỆN TỬ SỐ

Điện tử số (Digital electronics) là nhánh của điện tử xử lý các tín hiệu số trên cơ sở đại số logic, sử dụng các công logic và flip-flop điện tử.

Mạch điện tử số (Digital electronic circuit): là mạch điện tử thực hiện chức năng của hàm logic sử dụng các công logic số và phần tử nhớ flip-flop.

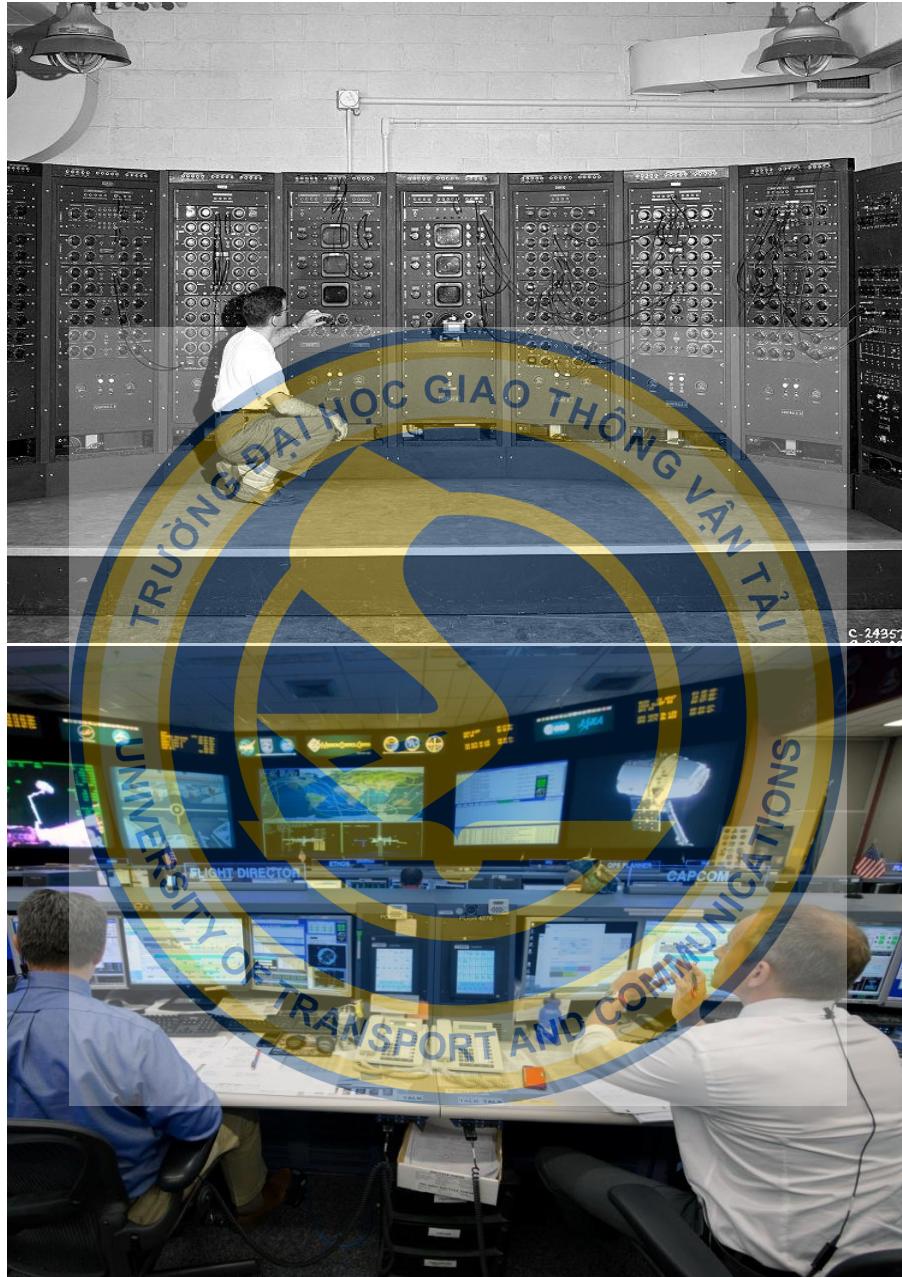
Ngày nay mạch điện tử số còn có thể được xây dựng bằng phương pháp lập trình các hàm logic trên máy tính hay thiết bị chuyên dụng, sau đó nạp vào các thiết bị có khả năng lập trình/xóa như: thiết bị logic khả trình (Programmable logic device: PLD), Vi điều khiển (Microcontroller: μ P), và bộ điều khiển logic lập trình được (Programmable logic controller: PLC). Sau khi được nạp, các thiết bị này sẽ hoạt động như một mạch điện tử số thiết kế. Phương pháp này có ưu điểm là linh động và dễ dàng chức năng hóa lại bằng phần mềm.



Hình 1.7. Mô hình mạch điện tử tương tự và mạch điện tử số.

- So với mạch điện tử tương tự, mạch điện tử số có các ưu điểm vượt trội:
 - + Dựa trên cơ sở của đại số logic có thể thiết kế và tối ưu mạch.
 - + Dễ tích hợp nhiều tính năng trên 1 chip.
 - + Các mạch số ít bị ảnh hưởng của nhiễu, giá trị chính xác của điện áp và dòng điện là không quan trọng như điện tử tương tự, mà chỉ là dải giá trị biểu thị cho logic 1 và 0.
 - + Nếu trong hệ thống lưu thông tin tương tự như băng từ, đĩa từ, hệ thống tương tự, các yếu tố như sự già hóa, sự tổn hao dù nhỏ nhất có thể làm thay đổi nội dung thông tin, thì với hệ thống số, dải giá trị biểu thị giá trị logic hay khoảng lè tránh nhiễu (noise margin, xem thêm chương 4), cho phép thông tin lưu trữ có độ bảo toàn cao hơn nhiều.
 - + Việc lưu trữ thông tin dễ dàng hơn, dung lượng thông tin lưu trữ lớn hơn nhiều, và truy cập dễ dàng và nhanh chóng.
 - + Dễ dàng điều khiển và cấu hình lại bằng phần mềm. Vì vậy, cho phép, cập nhật các tính năng mới mà không cần thay đổi phần cứng.
 - Bên cạnh đó, do các đối tượng của điện tử là các đại lượng vật lý trong thế giới tự nhiên và đa số chúng ở dạng tín hiệu tương tự: tín hiệu âm thanh từ micro; tín hiệu nhiệt độ, độ ẩm, nhịp tim, ... từ các cảm biến; tín hiệu áp lực ra loa;... , để tương

thích được với điện tử số, cần sử dụng các bộ biến đổi tương tự ↔ số. Vì vậy có thể sinh ra lỗi hay sai lệch thông tin do các quá trình này. Ở khía cạnh khác, với cùng khối lượng tính toán và xử lý tín hiệu, mạch số tiêu tốn năng lượng nhiều hơn và dễ sinh nhiệt hơn. Tuy nhiên, với sự phát triển của công nghệ chế tạo vi mạch và ngôn ngữ lập trình hiện nay, các vấn đề này không phải là lớn.



Hình 1.8. Máy tính của NASA, Hoa Kỳ: hình trên máy tính tương tự, chụp năm 1949, với kích thước chiếm cả căn phòng lớn; hình dưới là rất nhiều máy tính số trong một căn phòng, ảnh chụp năm 2014 (nguồn internet).

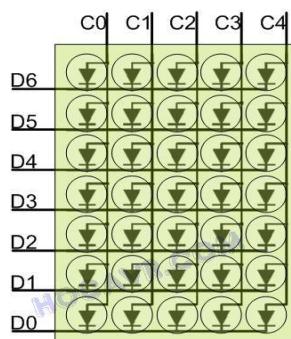
Ôn tập chương 1

Lý thuyết

- Khái niệm về hệ đếm? Nêu phương pháp chuyển đổi giữa các hệ đếm?
- Khái niệm tín hiệu số, tín hiệu tương tự?
- So sánh ưu điểm, nhược điểm mạch điện tử số và mạch điện tử tương tự?

Bài tập

- Chuyển hệ giá trị hệ 2 sau sang 8, 10, và 16
 - 11101
 - 101101
 - 1111011
 - 10100111
- Chuyển hệ giá trị trong hệ 8 sau sang 2, 10, và 16
 - 356
 - 7453
 - 4522
 - 53467
- Chuyển hệ giá trị hệ 10 sau sang 2, 8, và 16
 - 256
 - 3249
 - 88991
 - 435672
- Chuyển hệ giá trị hệ 16 sau sang 2, 8, và 10
 - 1EA
 - F10B
 - 1C01E
 - 36DF15
- Mã hóa số sinh viên trong lớp học sử dụng mã nhị phân.
- Mã hóa chữ cái tiếng Anh bằng mã Gray.
- Mã hóa chữ cái tiếng Việt bằng mã Dư 3.
- Thiết lập bảng mã cho LED ma trận như hình để hiển thị chữ A → Z, và các con số 0 → 9.



Chương 2

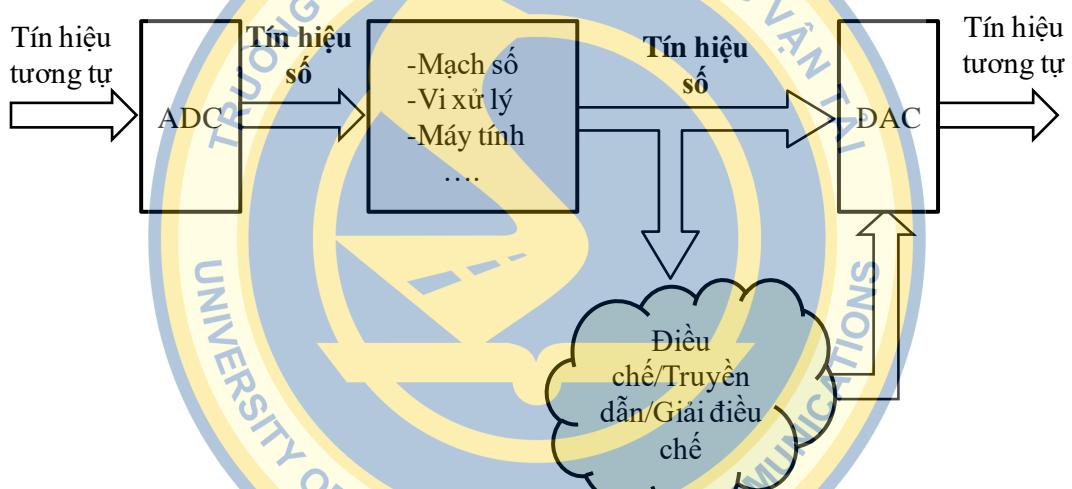
CHUYỂN ĐỔI TÍN HIỆU TƯƠNG TỰ - SỐ

2.1. GIỚI THIỆU

Như đã trình bày ở chương 1, các đại lượng vật lý ở đầu vào và đầu ra mạch điện tử số hầu hết có bản chất tương tự, như vậy cần phải thực hiện sự chuyển đổi từ dạng tương tự sang dạng số, trước khi đưa vào mạch và chuyển ngược lại từ dạng số đã xử lý sang dạng tương tự ở đầu ra. Chương 2, tập trung giới thiệu nguyên lý căn bản, các loại, và tham số kỹ thuật của bộ chuyển đổi tương tự/số và số/tương tự.

Tín hiệu trong tự nhiên là dạng tương tự như âm thanh, hình ảnh còn tín hiệu được xử lý trong các hệ thống máy tính để lưu giữ, gia công hay truyền dẫn lại là tín hiệu số. Vì vậy, có thể nói sự biến đổi qua lại giữa các tín hiệu từ dạng tương tự sang dạng số là cần thiết.

Vị trí của các bộ chuyển đổi tín hiệu trong hệ thống điện tử truyền thông nói chung được thể hiện trong hình dưới đây.



Hình 2.1. Sơ đồ khái niệm của một hệ thống điện tử truyền thông.

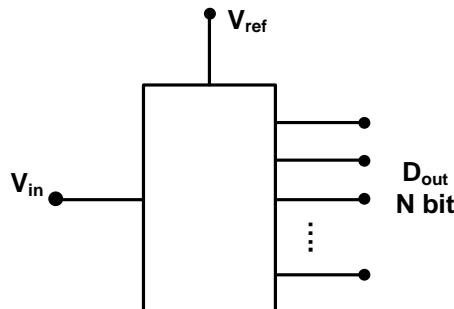
Trong đó, chuyển đổi tương tự-số (Analog-to-Digital Converter, ký hiệu: ADC hay A/D) là quá trình chuyển đổi tín hiệu liên tục (thường là điện áp) thành tín hiệu số để đưa vào hệ thống xử lý số.

Kết quả từ các hệ thống số là các đại lượng số cần được đổi thành tín hiệu tương tự để có thể tác động vào các hệ thống vật lý và thể hiện ra bên ngoài (ví dụ tái tạo âm thanh hay hình ảnh) hay dùng vào việc điều khiển sau đó (ví dụ dùng điện thế tương tự để điều khiển vận tốc động cơ). Mạch thực hiện chức năng này chính là mạch chuyển đổi số-tương tự (Digital-to-Analog Converter, ký hiệu là DAC hay D/A).

Phản tiếp theo đây sẽ giới thiệu chi tiết về hai loại chuyển đổi tín hiệu này.

2.2. CHUYỂN ĐỔI TƯƠNG TỰ/SỐ (ADC)

2.2.1. Nguyên tắc hoạt động



Hình 2.2. Sơ đồ khối của ADC.

Bộ chuyển đổi tương tự / số biến đổi tín hiệu tương tự liên tục về thời gian và biến độ thành tín hiệu số như mô tả trong hình trên. Giống chuyển đổi từ hệ đếm 10 sang hệ nhị phân, tín hiệu tương tự đầu vào V_{in} được chuyển thành giá trị số n bit thông qua biểu thức (1) sau:

$$\frac{V_{in}}{V_{ref}} = s(t) = B_0 2^0 + B_1 2^1 + \dots + B_{n-2} 2^{n-2} + B_{n-1} 2^{n-1} \quad (2.1)$$

Trong đó: V_{ref} là giá trị điện áp tham chiếu hay giá trị điện áp tương tự lớn nhất có thể chuyển đổi, giá trị này là cố định và được đưa ra sau khi sản xuất vi mạch ADC, điện áp V_{in} trước khi đưa vào chuyển đổi được chia cho giá trị điện áp tham chiếu và ở chuyển đổi ngược lại: số tương tự, sẽ được nhân với giá trị này để đảm bảo khôi phục lại dạng tín hiệu gốc. B₀ là bit có trọng số nhỏ nhất (LSB) và B_{n-1} là bit có trọng số lớn nhất (MSB). Tập hợp giá trị B_{n-1} B_{n-2} ... B₁ B₀ ta được tín hiệu số đầu ra.

Quá trình chuyển đổi gồm 3 bước:

- Lấy mẫu (Sampling): tín hiệu tương tự, sau khi qua khối lấy mẫu sẽ bị rời rạc hóa về thời gian, với thời gian đều nhau. Tín hiệu rời rạc về thời gian được giữ (Holding) cho các quá trình chuyển đổi tiếp theo.

Để đảm bảo khôi phục lại tín hiệu tương tự, tần số lấy mẫu F_s phải tuân theo định lý lấy mẫu Nyquist-Shannon:

$$F_s \geq 2f_{max} \quad (2.2)$$

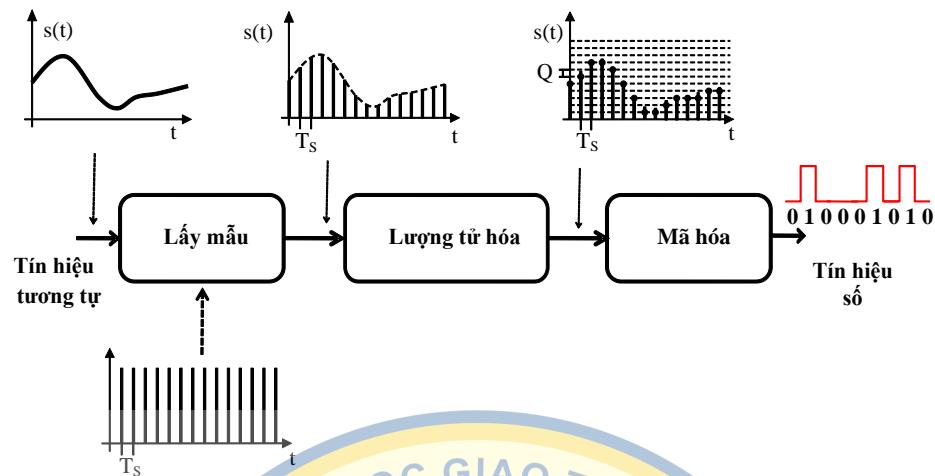
Trong đó f_{max} là tần số lớn nhất của tín hiệu tương tự.

- Lượng tử hóa (Quantization): tín hiệu rời rạc về thời gian tiếp tục được rời rạc về biến độ, tức là biến độ chỉ có giá trị ở một số mức nhất định. Mức lượng tử Q được tính theo công thức:

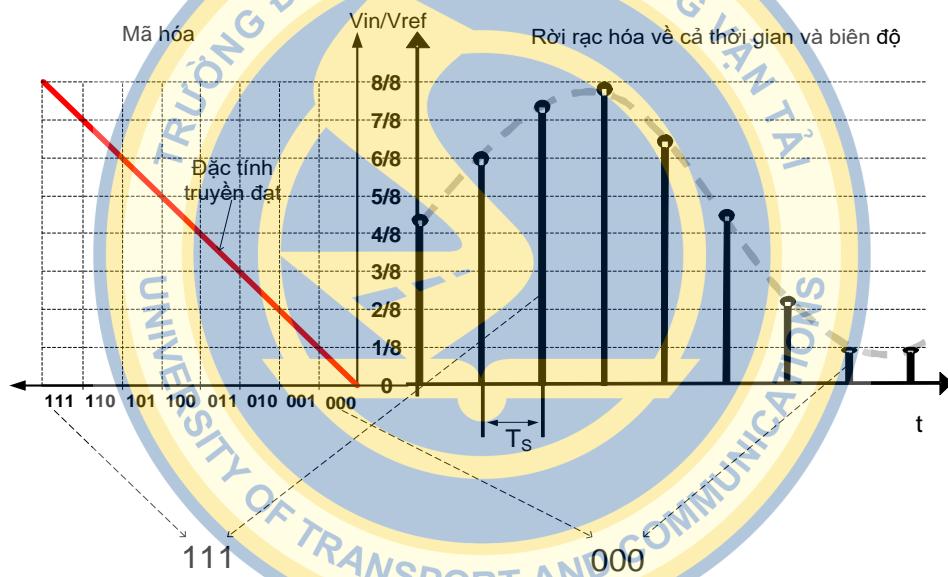
$$Q = \frac{V_{ref}}{2^n} \quad (2.3)$$

Mức lượng tử hay còn gọi là bậc số hóa, Q tương đương với giá trị của 1 LSB.

- Mã hóa (Encoding): tín hiệu được lượng tử hóa qua khôi mã hóa, mỗi mức biên độ được gán cho một tổ hợp nhị phân nhất định tùy thuộc vào loại mã sử dụng, nhưng phổ biến là mã nhị phân.



Hình 2.3. Mô tả chi tiết quá trình chuyển đổi tương tự / số.



Hình 2.4 Mô tả chi tiết quá trình chuyển đổi tương tự / số.

2.2.2. Thông số kỹ thuật cơ bản của ADC

– Dải điện áp đầu vào

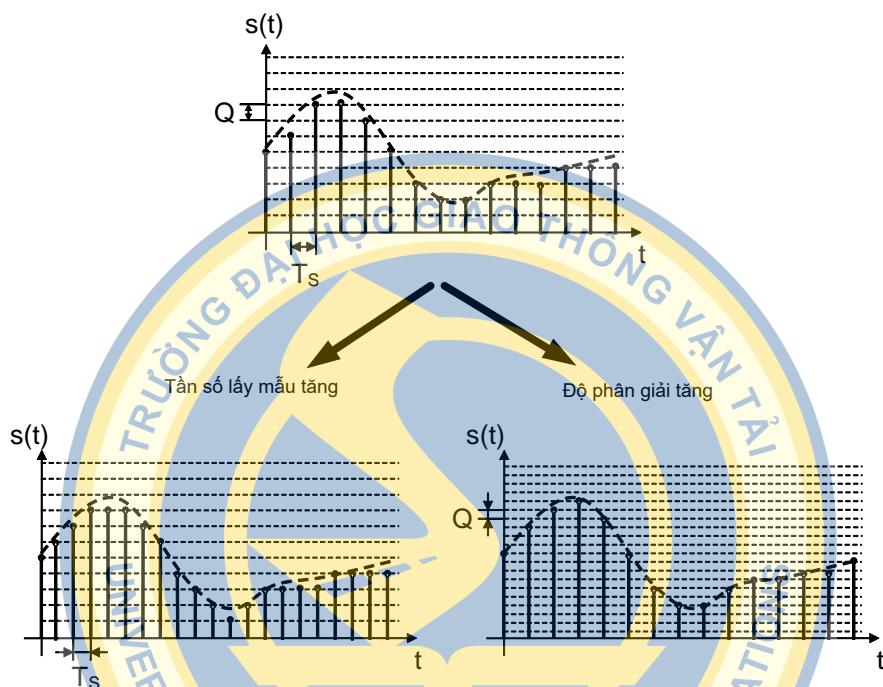
Dải điện áp đầu vào của một ADC được xác định bởi điện áp tham chiếu V_{ref} , nghĩa là từ $0 - V_{ref}$. ADC sẽ bị bão hòa với một điện áp tương tự cao hơn điện áp tham chiếu, do đó, khi thiết kế mạch sử dụng ADC, cần phải đảm bảo điện áp đầu vào không vượt quá điện áp tham chiếu.

Khi ADC hoạt động ở chế độ hai cực, nó cho phép điện áp âm đầu vào khi đó, phạm vi đầu vào tương tự là từ $-V_{ref}$ đến $+V_{ref}$.

- Tân số lấy mẫu và độ phân giải

Tân số lấy mẫu F_s là tần số của xung nhịp đưa vào khói lấy mẫu để giữ tín hiệu đầu vào tại những khoảng thời gian nhất định, đủ để thực hiện xong việc chuyển đổi. Như đã đề cập giá trị của F_s tối thiểu gấp đôi tần số lớn nhất của tín hiệu tương tự, nhưng, để đảm bảo chất lượng của quá trình chuyển đổi thì giá trị này thường lớn hơn nhiều lần.

Độ phân giải của một ADC biểu thị bằng số bit của tín hiệu số đầu ra. Khi này số mức lượng tử hóa chính là 2^n , nghĩa là tín hiệu đầu vào được chia thành 2^n mức từ 0 tới 2^n-1 . Số lượng bit càng nhiều sai số lượng tử càng nhỏ, độ chính xác càng cao.



Hình 2.5. Ảnh hưởng của tần số lấy mẫu và độ phân giải tới kết quả chuyển đổi.

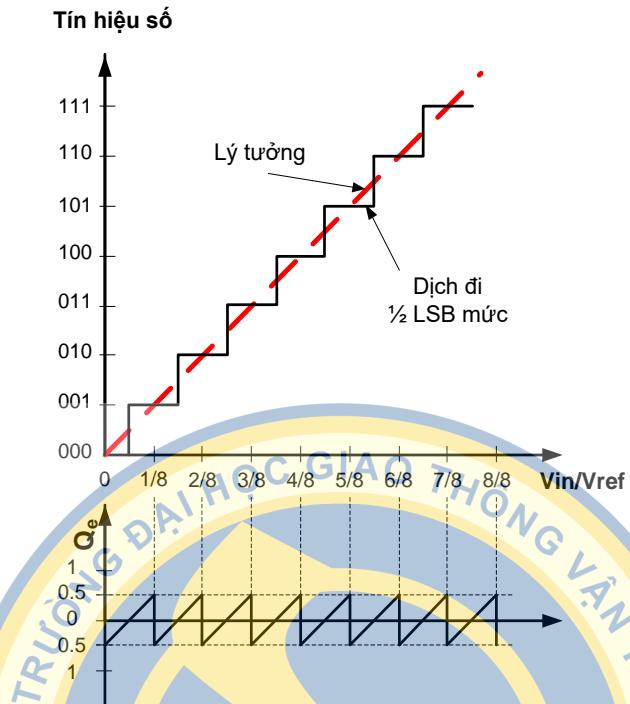
Hình trên cho thấy để tăng độ chính xác của việc chuyển đổi thì có thể tăng tần số lấy mẫu để tăng độ chi tiết về thời gian hoặc tăng độ phân giải để tăng độ chi tiết về mặt biên độ.

Chú ý là việc tăng tần số lấy mẫu và tăng độ phân giải sẽ làm mạch ADC trở nên phức tạp, tốn diện tích, tốn điện năng... nên lựa chọn hai thông số này phù hợp với tín hiệu đầu vào là điều rất quan trọng để cân đối giữa độ chính xác và các thông số khác của mạch. Hiện nay độ phân giải của các ADC trong khoảng 3-16 bit.

- Sai số lượng tử

Từ tín hiệu đầu vào liên tục có vô số các giá trị, đầu ra là giá trị rời rạc có hạn nên sẽ tạo ra sai số từ quá trình lượng tử hóa. Sai số lượng tử được định nghĩa là sự sai khác giữa giá trị đầu vào tương tự thực tế và giá trị chuyển đổi từ đầu ra số, được tính bằng công thức: $Q_e = 1/2 Q$.

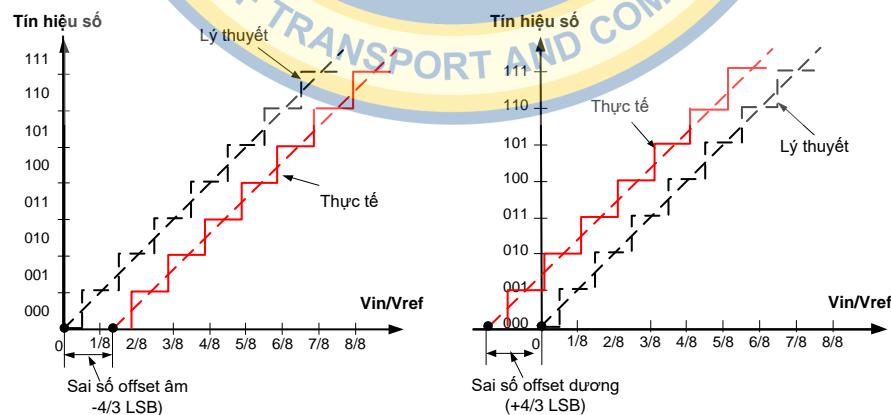
Theo lý thuyết, hàm truyền đạt lý tưởng của bộ ADC là một đường thẳng, tuy nhiên, với bộ ADC thực tế có dạng bậc thang với độ rộng bước Q (=LSB) như biểu diễn trong hình dưới đây. Từ hình này ta thấy lỗi lượng tử hóa nằm trong khoảng $\pm Q/2$.



Hình 2.6. Đặc tuyến truyền đạt lý tưởng và sai số lượng tử hóa của bộ ADC 3bit.

- Sai số lệch (offset error)

Sai số offset là sai số được xác định giữa sự chênh lệch của điểm offset danh định và điểm offset thực tế của bộ ADC. Điểm offset là giá trị giữa của bước chuyển khi đầu ra số bằng 0. Điểm offset danh định chính là gốc tọa độ. Trên thực tế hiện tượng này xuất hiện là khi tín hiệu tương tự đầu vào bằng 0 nhưng tín hiệu số đầu ra lại khác 0. Hình 2.7 biểu diễn đặc tuyến truyền đạt lý tưởng (nét đứt) và đường đặc tuyến thực tế (nét liền) khi có lỗi offset.

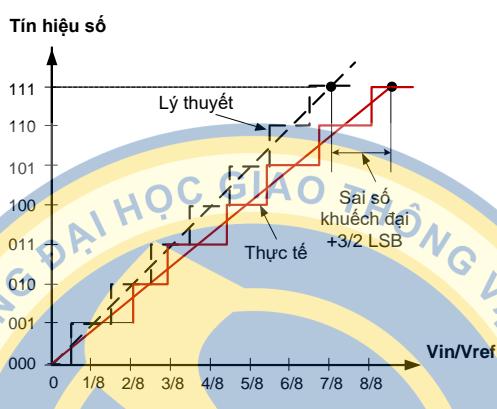


Hình 2.7. Sai số offset âm (hình trái) và dương (hình phải).

Sai số offset làm giới hạn phạm vi điện áp vào ADC. Sai số offset dương làm cho tín hiệu ra số bão hòa với điện áp tương tự vào nhỏ hơn giá trị cực đại Vref. Và sai số offset âm làm đầu ra nhận giá trị 0 khi điện áp đầu vào nhỏ nhất.

- Sai số khuếch đại (gain error)

Do hệ số khuếch đại của bộ ADC không đồng nhất trong suốt dải đo nên đường đặc tuyến thực tế có thể như hình 2.8 và vì thế xuất hiện sai số khuếch đại (gain error). Sai số khuếch đại được định nghĩa là độ lệch giữa điểm khuếch đại thực tế so với điểm trên đặc tuyến lý thuyết sau khi đã loại trừ sai số offset. Điểm khuếch đại định nghĩa là điểm giữa của bước chuyển đổi khi đầu ra tín hiệu số ở giá trị toàn thang (tất cả các bit đầu ra đều là 1).

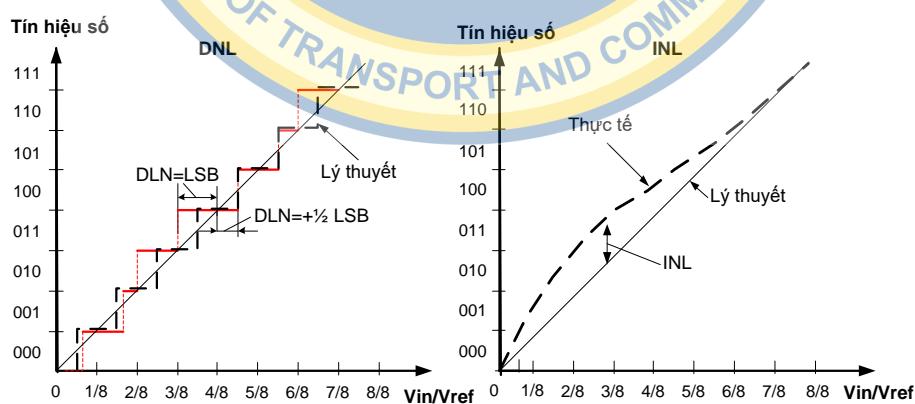


Hình 1.8. Sai số khuếch đại của ADC 3 bit

- Sai số phi tuyến (Nonlinearity)

+ *Sai số phi tuyến vi phân DNL (Differential Nonlinearity)*: là sự khác biệt giữa chiều rộng bước chuyển đổi thực tế và giá trị lý thuyết (LSB). Khi DNL nằm ngoài khoảng (-1 LSB; +1 LSB), sẽ làm mất mã khi một hay nhiều mã ở đầu ra

+ *Sai số phi tuyến tích phân INL (Integral Nonlinearity)*: là tổng các sai số phi tuyến vi phân, thông số này cho biết tính tuyến tính của ADC bằng cách xác định độ lệch của các điểm đặc tuyến thực tế so với đường lý thuyết.



Hình 2.9. Sai số DNL (hình trái) và INL (hình phải). DNL và INL xác định sau khi đã loại trừ sai số lệch và khuếch đại.

- Tỉ số tín hiệu trên nhiễu SNR

Tỉ số tín hiệu trên nhiễu SNR (Signal to Noise Ratio) là tỷ số giữa giá trị hiệu dụng của đầu vào lớn nhất và giá trị hiệu dụng của nhiễu, thông thường được tính bằng dB theo biểu thức sau:

$$\text{SNR} = 10 \lg \frac{P_{\text{signal}}}{P_{\text{noise}}} = 20 \lg \frac{V_{\text{in(max)}}}{P_{\text{noise}}} (\text{dB}) \quad (2.4)$$

Trong trường hợp tín hiệu đầu vào dạng hình sin với giá trị đỉnh-đỉnh bằng giá trị điện áp tham chiếu V_{ref} thì có thể tính SNR theo số bit n theo công thức:

$$\text{SNR}_{\text{max}} = 6,02 \times n + 1,76 \quad (2.5)$$

- Tỉ số tín hiệu trên nhiễu và méo SNDR (Signal to Noise and Distortion Ratio)

Đây là hệ số mang tính thực tế tốt hơn so với SNR, nó cho thấy ảnh hưởng của các loại nhiễu tác động tới ADC và được tính theo công thức:

$$\text{SNR} = 10 \lg \frac{V_{\text{FS}}^2 / 8}{n_Q^2 + n_D^2 + n_{\text{thermal}}^2 + n_{\text{jitter}}^2 + n_{\text{ps}}^2} \quad (2.6)$$

trong đó V_{FS} ($= V_{\text{ref}}$), n_Q là nhiễu lượng tử, n_D là nhiễu méo dạng, n_{thermal} là nhiễu nhiệt, n_{jitter} là nhiễu cho xung đồng hồ bị trượt, n_{ps} là nhiễu do nguồn cung cấp, tất cả đều là các giá trị hiệu dụng.

SNR là giá trị SNDR lý tưởng khi bỏ qua hết các loại nhiễu trừ nhiễu lượng tử.

- Số bit hiệu dụng ENOB (Effective number of bit)

Là số bit được sử dụng thay cho độ phân giải có công thức tính như sau:

$$\text{ENOB} = \frac{\text{SNDR(dB)} - 1,76}{6,02} \quad (2.7)$$

Thông số này cho biết chất lượng của thiết kế vì với thiết kế có số bit cho trước, giá trị của ENOB thể hiện thực tế đạt được là bao nhiêu so với giá trị tối đa (n) đã cho.

- Tốc độ chuyển đổi

Tốc độ chuyển đổi được xác định bởi thời gian cần thiết hoàn thành một lần chuyển đổi A/D. Thời gian này tính từ khi xuất hiện tín hiệu điều khiển chuyển đổi đến khi tín hiệu số đầu ra đã ổn định, bao gồm thời gian khởi động, lấy và giữ mẫu, và thời gian mã hóa.

- Công suất tiêu thụ

Thông số này phụ thuộc chính vào độ phức tạp trong cấu trúc của ADC, kích thước linh kiện sử dụng và giá trị điện áp nguồn cung cấp.

- Hệ số chất lượng FoM (Figure of Merit)

FoM là thông số được sử dụng nhiều nhất để đánh giá tổng quan chất lượng của các loại ADC khác nhau bằng cách xác định mối quan hệ giữa công suất tiêu thụ tổng (P) với tần số lấy mẫu và ENOB theo biểu thức sau:

$$\text{FoM} = \frac{P}{F_s 2^{\text{ENOB}}} \quad (2.8)$$

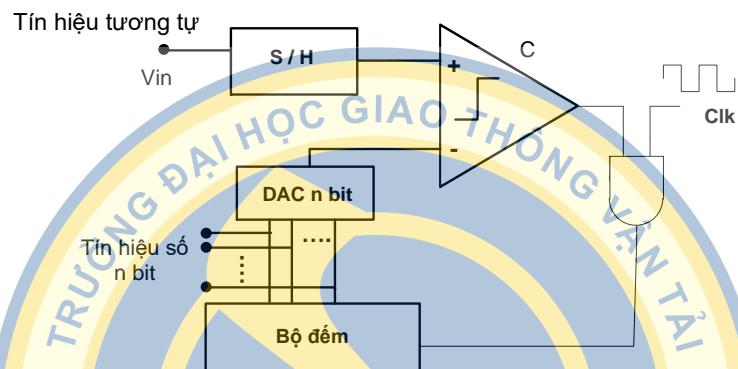
2.2.3. Các loại bộ chuyển đổi tương tự/số

- ADC đường dốc (Ramp ADC)

Sơ đồ khối cơ bản của ramp ADC được cho trong hình 2.10. Trong đó, S/H là khối Lấy mẫu và Giữ mẫu có nhiệm vụ giữ giá trị của điện áp đầu vào không đổi trong suốt thời gian chuyển đổi sang dạng số.

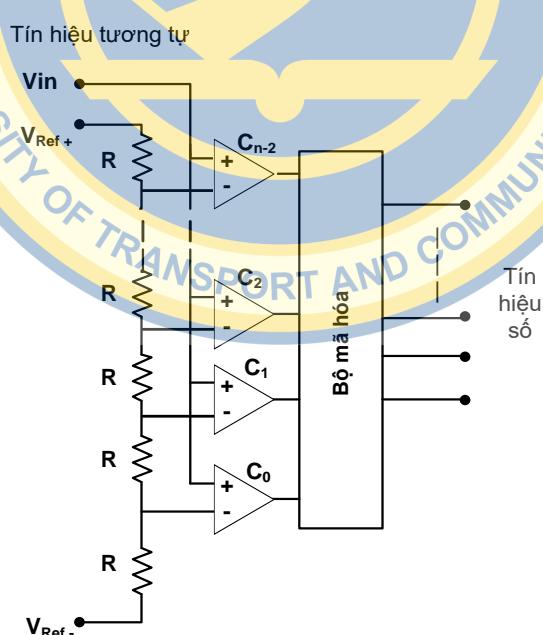
Ban đầu giá trị của bộ đếm được đặt là 0, giá trị của bộ đếm tăng dần và đưa tới DAC, để chuyển ngược lại thành tín hiệu tương tự và so sánh với tín hiệu đầu vào V_{in} . Khi tín hiệu đầu V_{in} vào bằng tín hiệu này thì quá trình đếm dừng và giá trị của bộ đếm thời điểm đó chính là kết quả tín hiệu số chuyển đổi.

Nhược điểm chính của loại ADC này là tốc độ chậm vì bộ đếm phải quét tất cả các giá trị từ nhỏ tới lớn để tìm ra giá trị gần nhất với giá trị đầu vào tương tự.



Hình 2.10. Sơ đồ khối cơ bản của Ramp ADC. Clk: xung nhịp. C: bộ so sánh.

- Flash ADC



Hình 2.11. Cấu trúc điển hình của một bộ flash ADC.

Hình 2.11 mô tả cấu trúc cơ bản của flash ADC. Trong một bộ flash ADC n bit có dãy ($2^n - 1$) bộ so sánh để so sánh tín hiệu đầu vào với tập tất cả các giá trị điện áp tham chiếu. Như vậy, các bộ so sánh hoạt động đồng thời để xác định giá trị đầu ra vì thế loại ADC này có tốc độ chuyển đổi rất nhanh. Tuy nhiên, do số lượng bộ so sánh sử dụng tăng mạnh khi số bit lớn nên flash ADC tiêu thụ công suất lớn và tốn diện tích khi độ phân giải tăng. Chẳng hạn với độ phân giải 3 bit thì mạch cần 7 bộ so sánh và khi độ phân giải lên 8 bit thì cần tới 255 bộ so sánh. Trên thực tế, flash ADC thường được sản xuất có độ phân giải từ 3 đến 6 bit với tần số từ cỡ MHz lên tới GHz.

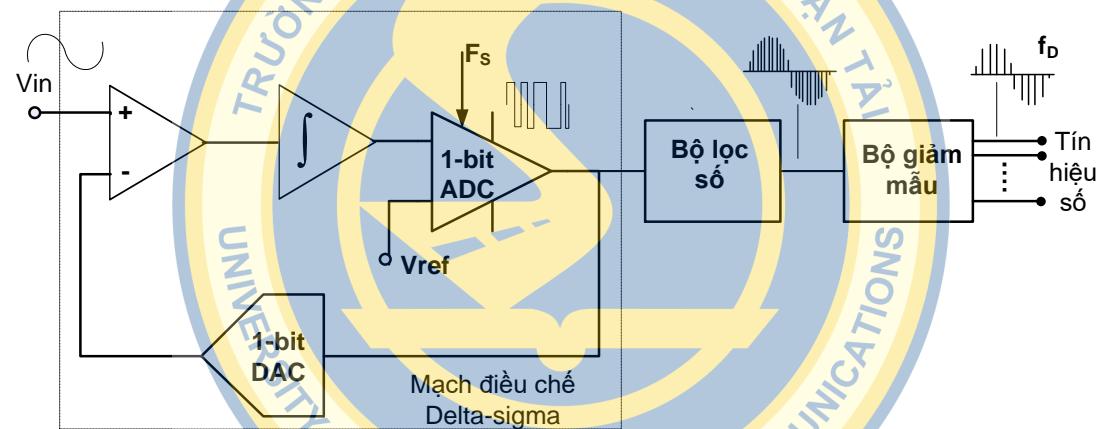
- Delta-Sigma ADC

Hình 2.12 là cấu trúc cơ bản của một Delta-Sigma ADC, loại ADC có độ chính xác cao nhất nhờ sử dụng bộ lọc số để loại bỏ sai số lượng tử.

Khối quan trọng nhất trong mạch là mạch điều chế Delta-Sigma với các khối khuếch đại vì sai (Delta), khối tích phân (Sigma), bộ so sánh (ADC 1 bit) và bộ DAC 1 bit.

Tần số lấy mẫu f_s được đưa vào bộ so sánh còn tần số f_D là tốc độ đọc dữ liệu ra của bộ giảm mẫu. Tỉ số giữa f_s/f_D được gọi là tỉ số loại trừ của bộ giảm mẫu.

Mặc dù có độ chính xác cao nhưng do mạch có cấu trúc phức tạp nên loại ADC này thường tiêu thụ năng lượng lớn khi độ phân giải tăng lên.



Hình 2.12. Cấu trúc điển hình của một mạch Delta-Sigma ADC.

- SAR ADC

Cấu trúc cơ bản của một SAR ADC bao gồm mạch lấy mẫu, mạch so sánh, mạch DAC và thanh ghi SAR logic.

So với các loại ADC khác thì SAR ADC có cấu trúc đơn giản nhất nên nó cũng tiêu thụ năng lượng ít nhất. Xu hướng của điện tử hiện đại là tạo ra các thiết bị nhỏ gọn, dễ dàng mang theo và tiêu thụ năng lượng. Vì vậy, càng ngày SAR ADC càng được sử dụng rộng rãi.

Nguyên tắc hoạt động của SAR ADC được thể hiện qua các bước sau:

1. Giá trị logic 1 được đưa vào đầu vào của thanh ghi dịch. Sau mỗi bit chuyển đổi bit 1 này sẽ dịch sang phải một bit. $B_{n-1} = 1$ còn tất cả các bit khác bằng 0.

2. Bit có trọng số lớn nhất, MSB, của SAR là D_{n-1} được đặt giá trị ban đầu là 1 trong khi các bit khác đều bằng 0.

3. Do đầu ra của khối SAR điều khiển DAC với giá trị 100...0 nên đầu ra của DAC sẽ có giá trị bằng $V_{ref}/2$.

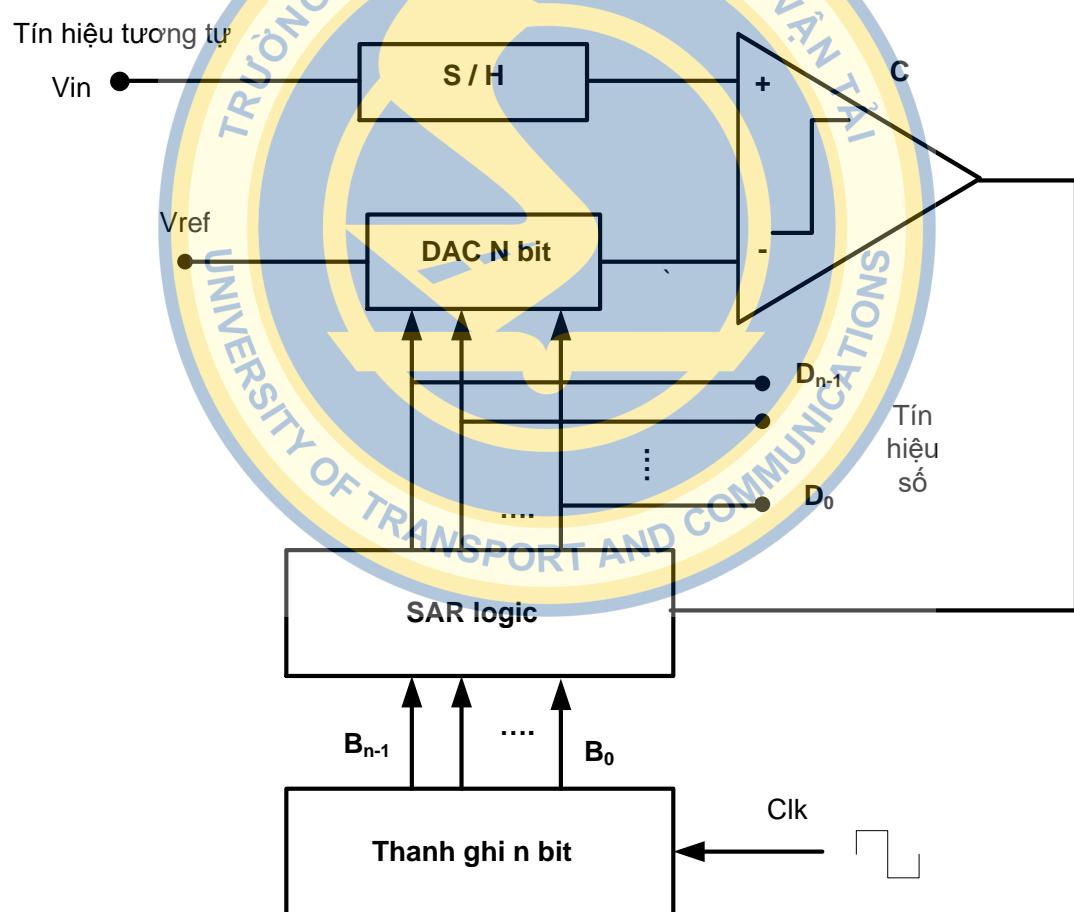
4. V_{in} được so sánh với $V_{ref}/2$. Nếu $V_{ref}/2$ lớn hơn V_{in} thì đầu ra của bộ so sánh mang giá trị 0 và thiết lập lại $D_{n-1} = 0$. Ngược lại, $V_{ref}/2$ nhỏ hơn V_{in} thì đầu ra bộ so sánh mang giá trị 1 và D_{n-1} giữ nguyên giá trị 1. D_{n-1} là giá trị MSB thực sự của mã đầu ra.

5. Bit 1 trong thanh ghi chuyển sang vị trí tiếp theo, $B_{n-2} = 1$ trong khi các bit khác đều bằng 0.

6. Trong khi D_{n-1} giữ nguyên giá trị của lần chuyển đổi trước thì D_{n-2} được thiết lập bằng 1 còn D_{n-3} đến D_0 bằng 0. Lúc này đầu ra của DAC sẽ được cộng thêm $V_{ref}/4$ với giá trị trước đó.

7. V_{in} lại được so sánh với đầu ra của DAC. Nếu đầu ra của DAC lớn hơn V_{in} thì D_{n-2} được thiết lập lại giá trị 0, ngược lại thì nó giữ nguyên giá trị 1.

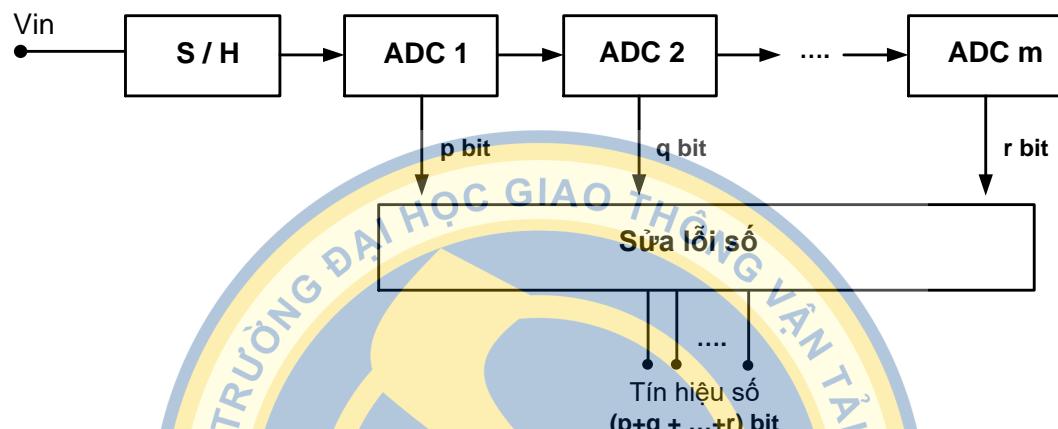
8. Quá trình được lặp lại tới khi đầu ra của DAC hội tụ gần nhất với giá trị V_{in} .



Hình 2.13. Cấu trúc cơ sở của SAR ADC.

- ADC đường ống (Pipelined ADC)

Mạch pipelined ADC về thực chất là sự nối tiếp của nhiều mạch ADC để có được độ phân giải cao. Như đã phân tích ở các mạch ADC trên, khi cần độ phân giải cao thì cấu trúc mạch sẽ trở nên công kênh vì vậy nếu chia thành nhiều tầng ADC với độ phân giải thấp hơn thì sẽ hiệu quả hơn. Trong cùng một cấu trúc pipelined có thể sử dụng nhiều loại ADC khác nhau. Khi này mạch phát huy tối đa ưu thế của mỗi loại ADC như tốc độ, độ chính xác và tiết kiệm năng lượng tương ứng với loại flash, Delta-Sigma và SAR ADC. Ví dụ, một pipelined ADC gồm một mạch flash ADC 3 bit và Delta-Sigma ADC 10 bit sẽ là một bộ chuyển đổi chính xác, tốc độ cao với độ phân giải 13 bit.



Hình 2.14. Sơ đồ khái niệm của pipelined ADC.

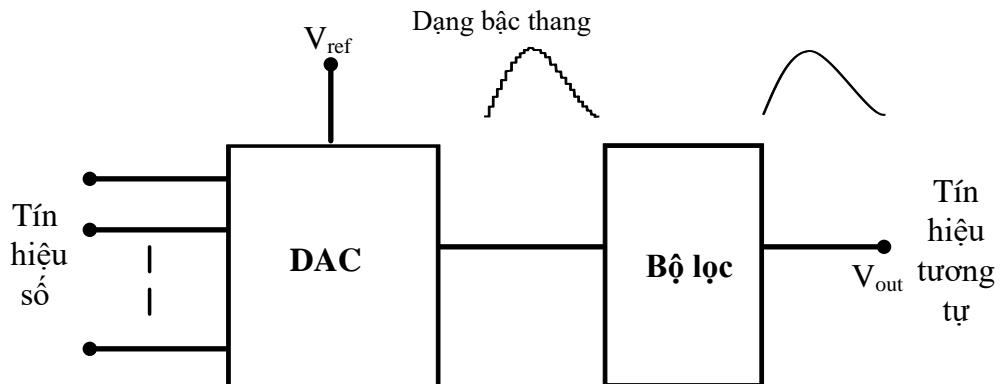
Bảng dưới đây tóm tắt các đặc điểm chính của các loại ADC vừa trình bày để thuận tiện cho việc nhận ra những ưu nhược điểm và phạm vi ứng dụng của mỗi loại ADC.

Bảng 2.1. So sánh các loại chuyển đổi ADC.

Loại ADC	Độ phân giải	Tốc độ	Công suất tiêu thụ	Chi phí
RAMP	Cao >12 bit	Thấp $<1\text{KHz}$	Trung bình	Thấp
Flash	Thấp 4-9 bit	Cao $100\text{MHz} - \text{GHz}$	Lớn	Cao
Delta - Sigma	Cao >12 bit	Thấp - Cao $\text{KHz} - \text{GHz}$	Trung bình	Trung bình
SAR	Trung bình - Cao 6-20 bit	Thấp - Cao $100\text{Hz} - \text{GHz}$	Thấp – Cực thấp	Thấp
Pipelined	Trung bình - Cao 8 - 18 bit	Trung bình – Cao $\text{MHz} - \text{GHz}$	Trung bình	Trung bình

2.3. CHUYỂN ĐỔI SỐ/TƯƠNG TỰ (DAC)

2.3.1. Nguyên tắc hoạt động

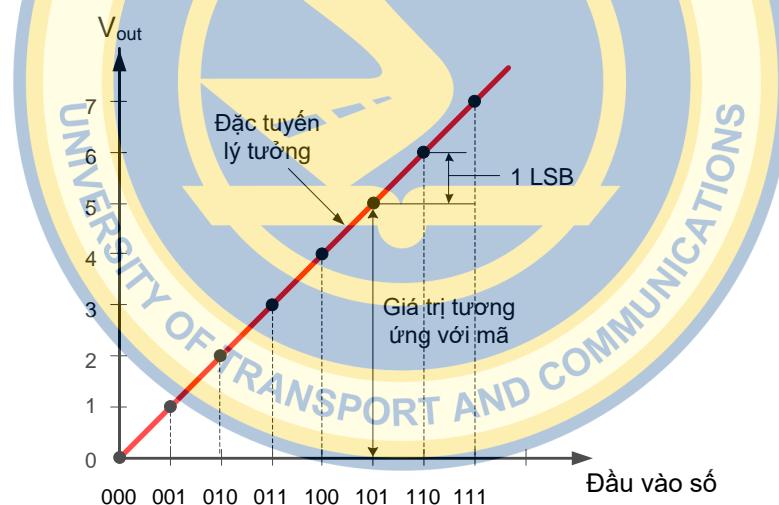


Hình 2.15. Sơ đồ khối của mạch chuyển đổi số - tương tự.

Các từ mã đầu vào N bit được đưa vào để chuyển đổi thành giá trị điện áp hoặc dòng điện tương ứng ở đầu ra theo biểu thức sau:

$$V_{out} = \frac{V_{ref}}{2^n} \sum_{i=0}^n D_i 2^i \quad (2.9)$$

Trong đó LSB được định nghĩa là giá trị gia tăng nhỏ nhất xuất hiện ở đầu ra. Công thức tính giống như phần trình bày trong ADC.



Hình 2.16. Đặc tuyến truyền đạt lý tưởng của bộ DAC 3 bit.

Giống như ADC, đặc tuyến truyền đạt lý tưởng của bộ DAC lý tưởng là đường thẳng thẳng hàng. Ví dụ, trong hình trên độ rộng mỗi bước chuyển đổi có giá trị bằng 1 LSB và bảng chuyển đổi giá trị số sang tương tự được cho trong bảng sau:

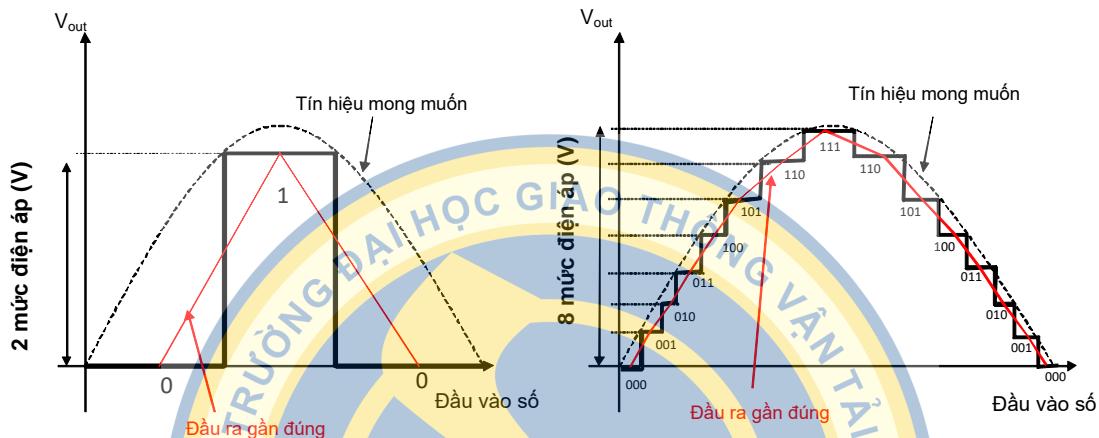
Tín hiệu số	000	001	010	011	100	101	110	111
Tương tự đầu ra	0	1	2	3	4	5	6	7

2.3.2. Thông số kỹ thuật cơ bản của DAC

- Độ phân giải

Dộ phân giải của bộ biến đổi DAC được định nghĩa là thay đổi nhỏ nhất có thể xảy ra ở đầu ra tương tự bởi kết quả của một thay đổi ở đầu vào số. Độ phân giải của DAC phụ thuộc vào số bit, DAC có càng nhiều bit thì độ phân giải càng nhỏ hơn. Độ phân giải luôn bằng trọng số của LSB, vì đó là khoảng thay đổi của V_{out} khi giá trị của đầu vào số thay đổi từ bước này sang bước khác.

Hình 2.17 mô tả sự chuyển đổi tín hiệu tương ứng với độ phân giải khác nhau để thấy được sự ảnh hưởng của độ phân giải tới dạng tín hiệu sau chuyển đổi DAC.



Hình 2.17. Dạng tín hiệu được tạo ra bởi DAC có độ phân giải 1 bit và độ phân giải 3 bit.

Ngoài ra, để so sánh với giá trị chuyển đổi toàn thang FS, độ phân cũng được tính theo phần trăm:

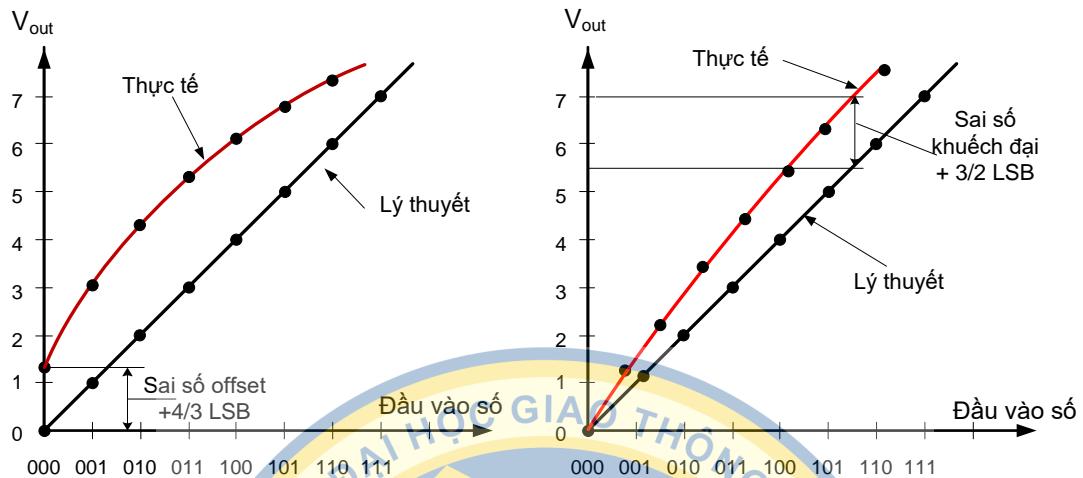
$$\text{Độ phân giải (\%)} = \frac{\text{LSB}}{\text{FS}} \times 100\% \quad (2.10)$$

Ví dụ, nếu điện áp tham chiếu là 2 V thì độ phân giải LSB tính theo giá trị điện áp và độ phân giải tính theo tỉ lệ phần trăm với giá trị toàn thang FS được xác định tương ứng theo số bit như bảng:

Độ phân giải n	2^n	$\text{LSB} = V_{ref} / 2^n$	FS	FS
2 bit	4	500 mV	1.75 V	28.57 %
4 bit	16	125 mV	1.875 V	6.67 %
8 bit	256	7.81 mV	1.992 V	0.392 %
10 bit	1024	1.95 mV	1.998 V	0.0976 %
16 bit	65536	30.518 μV	1.99997 V	0.0015 %
...

- Sai số lệch và sai số khuếch đại (offset error)

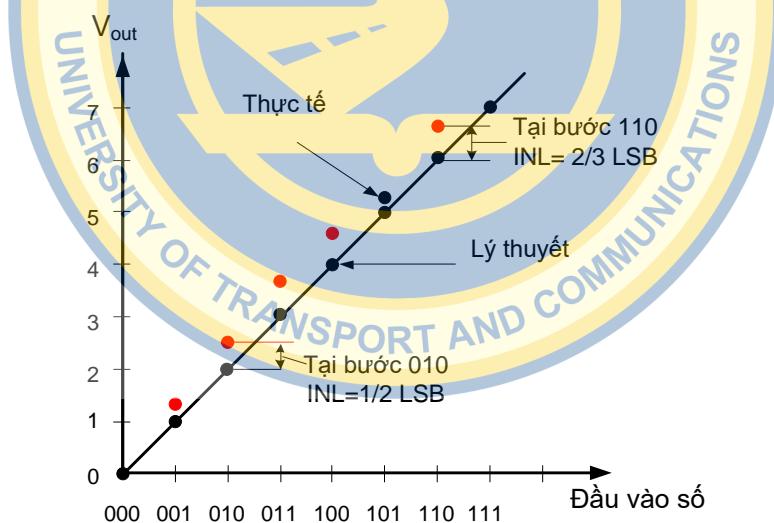
Sai số offset và sai số khuếch đại của DAC có cách xác định tương tự như trong ADC và được minh họa trong hình. Các sai số này gây sai số trong quá trình ánh xạ chuyên đổi từ tín hiệu số sang tương tự.



Hình 2.18. Cách xác định sai số offset (hình trái) và sai số khuếch đại (hình phải).

Đa số các chip DAC có tính năng điều chỉnh sai số lệch offset ngoài mạch, sẽ cho phép hiệu chỉnh giá trị độ lệch này bằng cách đặt mọi bit 0 ở đầu vào DAC và theo dõi đầu ra, cho đến khi nào đầu ra bằng 0V.

- Độ tuyến tính



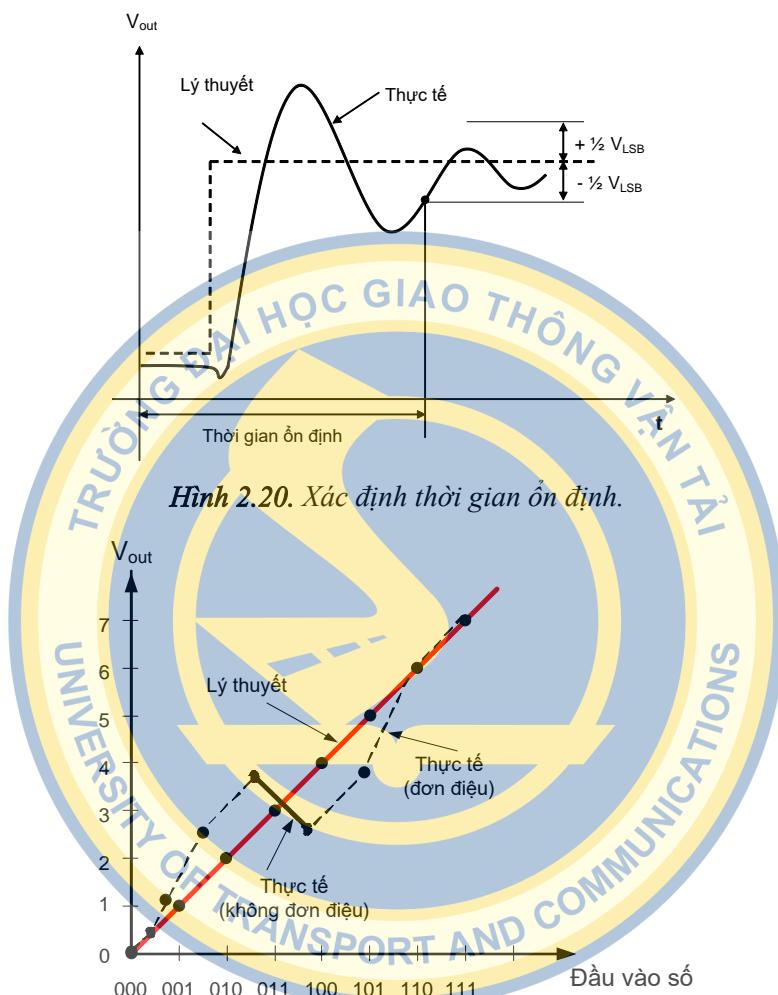
Hình 2.19. Đặc tuyến truyền đạt không tuyến tính gây sai số của ADC 3 bit.

Định nghĩa và cách xác định các tham số trên hoàn toàn tương tự như trong ADC và được minh họa trong hình 2.19. Trong hệ thống điện tử, tính tuyến tính rất quan trọng. Một ADC không tuyến tính sẽ tạo ra chuyển đổi không chính xác. Một DAC không tuyến tính sẽ làm cho tín hiệu tương tự khôi phục lại sẽ bị méo dạng.

- Thời gian ổn định

Thời gian ổn định (settling time) là thời gian cần thiết để đầu ra DAC tăng từ bậc thấp nhất đến bậc thang cao nhất khi đầu vào nhị phân biến thiên từ chuỗi bit toàn 0 đến chuỗi bit toàn 1. Thực tế thời gian ổn định là thời gian để đầu ra DAC ổn định trong phạm vi $\pm 1/2$ LSB của giá trị lý tưởng.

Hình 2.20 mô tả cách xác định thời gian ổn định của một DAC bằng cách tính tới thời điểm đầu ra lệch $\pm 1/2$ LSB so với giá trị lý tưởng (mong muốn).



- Tính không đơn điệu

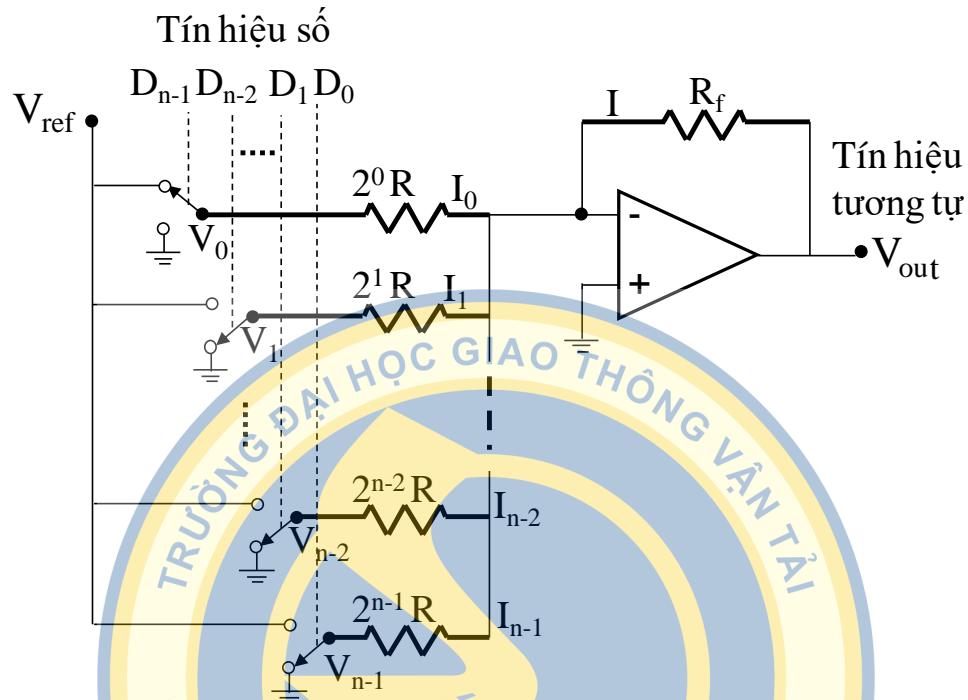
Tính đơn điệu (monotonicity) là đặc tính khi đầu vào số tăng thì đầu ra của DAC cũng tăng, ngược lại khi đầu ra không tăng hay giảm gọi là hiện tượng không đơn điệu (non-monotonicity).

Chú ý: Trong cả ADC và DAC, ngoài các tham số kể trên, còn có các tham số khác như mức logic cao, thấp, điện trở, điện dung ký sinh, của đầu vào; điện trở, điện dung của đầu ra; hệ số nhiệt, ...

2.3.3. Các cấu trúc DAC cơ bản

DAC có nhiều kiểu cấu trúc khác nhau từ đơn giản tới cực kỳ phức tạp. Phản tiếp theo đây sẽ giới thiệu những cấu trúc mang tính cơ bản và phổ biến nhất, bao gồm: DAC loại trọng số nhị phân, DAC loại R-2R.

- DAC trọng số nhị phân điện trở (Binary weighted resistor)



Hình 2.22. Sơ đồ cấu trúc của DAC loại trọng số nhị phân.

DAC loại trọng số nhị phân có sơ đồ nguyên lý như hình 2.22. Mạch kiểu mạch cộng đảo sử dụng IC khuếch đại thuật toán với các điện trở đầu vào có giá trị tăng theo trọng số: $R, 2^1 R, \dots, 2^{n-2} R, 2^{n-1} R$. Điện áp đầu ra V_{out} được xác định theo công thức sau:

$$V_{out} = -IR_f = -(I_0 + I_1 + \dots + I_{n-2} + I_{n-1})R_f \quad (2.11)$$

$$= -R_f \left(\frac{V_0}{2^0 R} + \frac{V_1}{2^1 R} + \dots + \frac{V_{n-2}}{2^{n-2} R} + \frac{V_{n-1}}{2^{n-1} R} \right) \quad (2.12)$$

Khi giá trị bit $D_i = 1$, chuyển mạch (thể hiện bằng đường chấm gạch nối với D_i) sẽ điều khiển để nối với nguồn tham chiếu V_{ref} , khi đó $V_i = V_{ref}$. Khi giá trị bit $D_i = 0$, chuyển mạch (thể hiện bằng đường chấm gạch nối với D_i) sẽ điều khiển để nối với điểm đất của mạch, khi đó $V_i = 0$. Như vậy, tổng quát có thể viết $V_i = V_{ref} D_i$

Như vậy điện áp tương tự đầu ra được xác định là:

$$= -R_f \left(\frac{V_0}{R 2^0} + \frac{V_1}{R 2^1} + \dots + \frac{V_{n-2}}{R 2^{n-2}} + \frac{V_{n-1}}{R 2^{n-1}} \right) \quad (2.13)$$

$$V_{out} = -R_f \left(\frac{V_{ref} D_{n-1}}{R 2^{n-1}} + \frac{V_{ref} D_{n-2}}{R 2^{n-2}} + \dots + \frac{V_{ref} D_1}{R 2^1} + \frac{V_{ref} D_0}{R 2^0} \right) \quad (2.14)$$

$$V_{out} = -R_f \left(\frac{V_{ref} D_0}{R 2^0} + \frac{V_{ref} D_1}{R 2^1} + \dots + \frac{V_{ref} D_{n-2}}{R 2^{n-2}} + \frac{V_{ref} D_{n-1}}{R 2^{n-1}} \right) \quad (2.15)$$

$$V_{out} = -V_{ref} R_f \left(\frac{D_0}{R 2^0} + \frac{D_1}{R 2^1} + \dots + \frac{D_{n-2}}{R 2^{n-2}} + \frac{D_{n-1}}{R 2^{n-1}} \right) \quad (2.16)$$

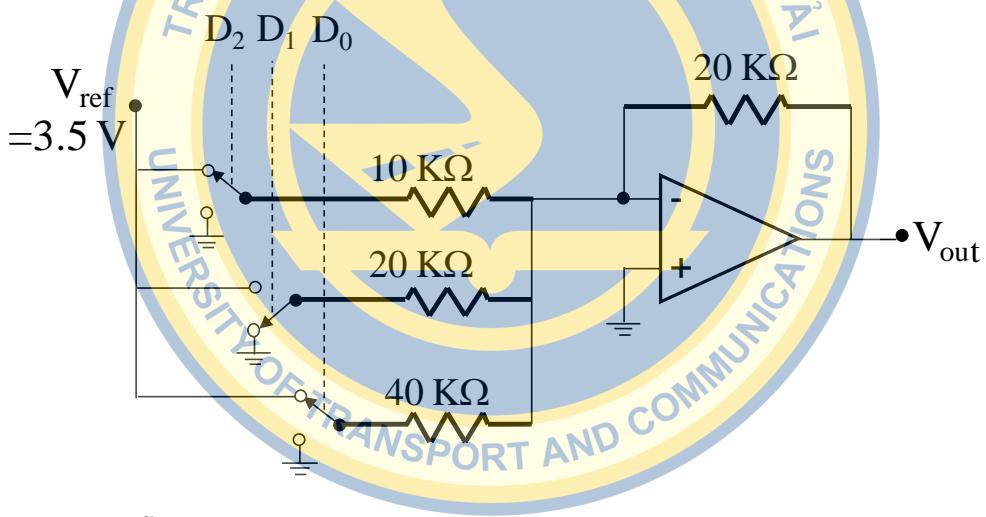
Chú ý:

Giá trị R trong công thức 2.16, không nhóm và ra ngoài để tránh hiểu nhầm, vì thực chất nó còn gắn với các giá trị trọng số $2^0, 2^1, \dots$ để tạo thành điện trở ở các nhánh đầu vào trong hình 2.22.

Ưu điểm của loại DAC này là tốc độ cao nhưng nhược điểm lớn là lại là khi số bit tăng lên làm cho các điện trở có trọng số lớn có giá trị rất lớn, rất có đam bảo độ chính xác và giá trị cho các điện trở lớn này, nhất là khi chế tạo DAC thành mạch tích hợp (IC). Ví dụ, nếu điện trở cơ bản $2^0 R = 1 K\Omega$, thì để chế tạo bộ DAC 12 bit, điện trở $2^{11} R = 2048/2 = 1024 K\Omega$. Một số thiết kế cũng sử dụng trọng số tụ điện.

Ví dụ 2.1:

Tính V_{out} của mạch ADC như hình dưới, khi $D_2 D_1 D_0$ nhận giá trị 011, 101



Hướng dẫn:

+ Khi $D_2 D_1 D_0$ nhận giá trị 011

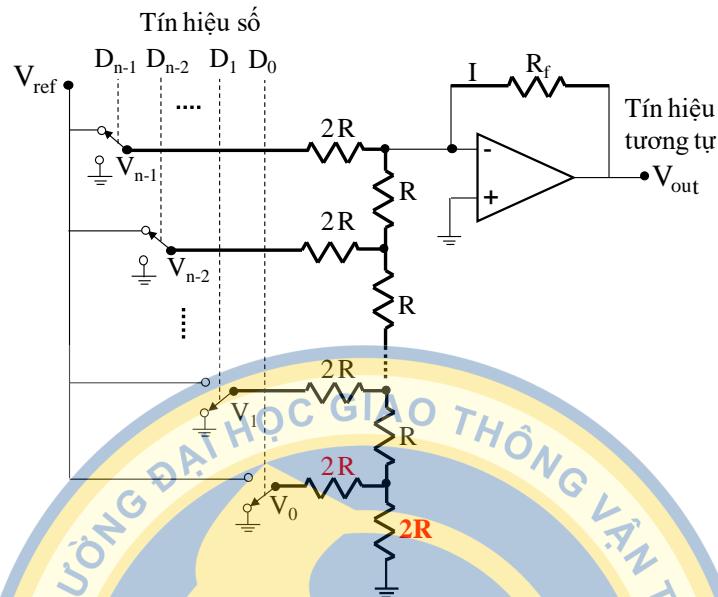
$$V_{out} = -3,5 \times 20 K\Omega \left(\frac{0}{10 K\Omega} + \frac{1}{20 K\Omega} + \frac{1}{40 K\Omega} \right) = -5,25 V$$

+ Khi $D_2 D_1 D_0$ nhận giá trị 101

$$V_{out} = -3,5 \times 20 K\Omega \left(\frac{1}{10 K\Omega} + \frac{0}{20 K\Omega} + \frac{1}{40 K\Omega} \right) = -0,875 V$$

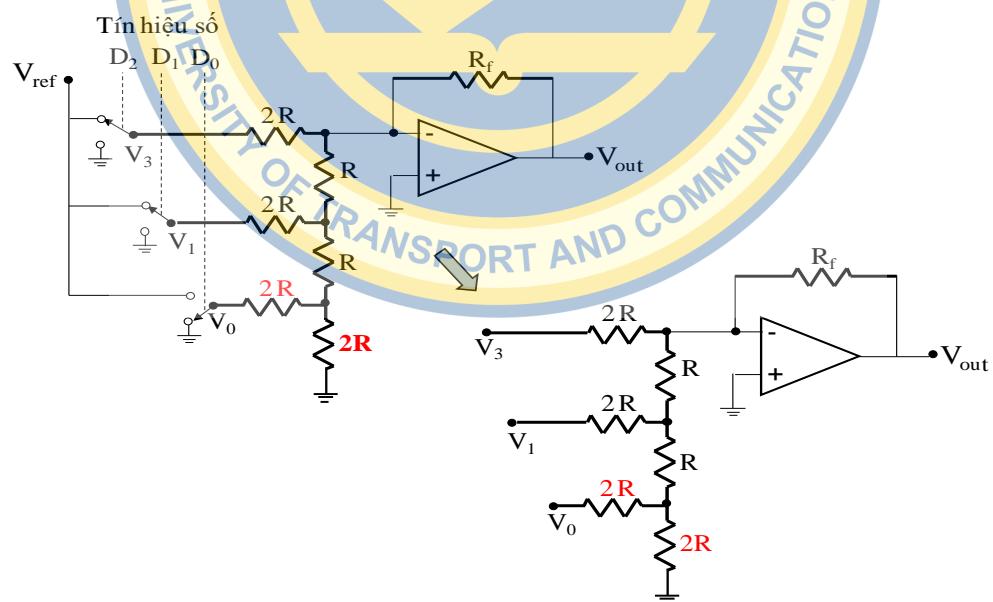
- DAC thang R/2R

DAC loại R/2R có dạng mạch như hình 2.23 dưới đây. Thay vì sử dụng các điện trở trọng số, DAC loại này chỉ sử dụng mạng điện trở R và 2R với mạch mắc theo kiểu hình thang.



Hình 2.23. Mạch DAC n bit loại R-2R.

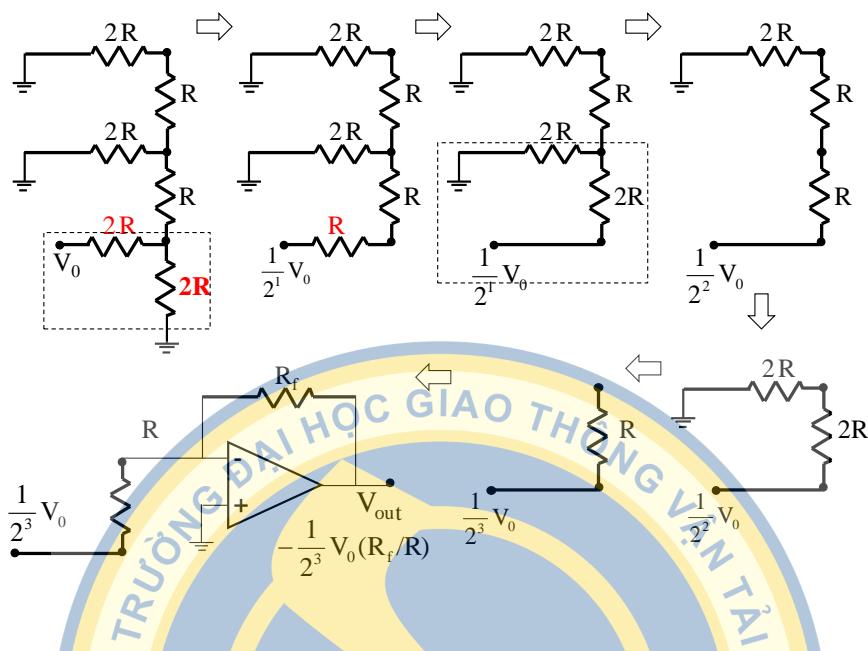
Để xây dựng công thức xác định điện tương tự cho bộ biến đổi DAC n bit, để đơn giản, trước hết ta xem xét bộ biến đổi cho 3 bit: D₂ D₁ D₀. Sơ đồ mạch điện tính từ V₀, V₁, và V₂ có dạng như hình 2.24



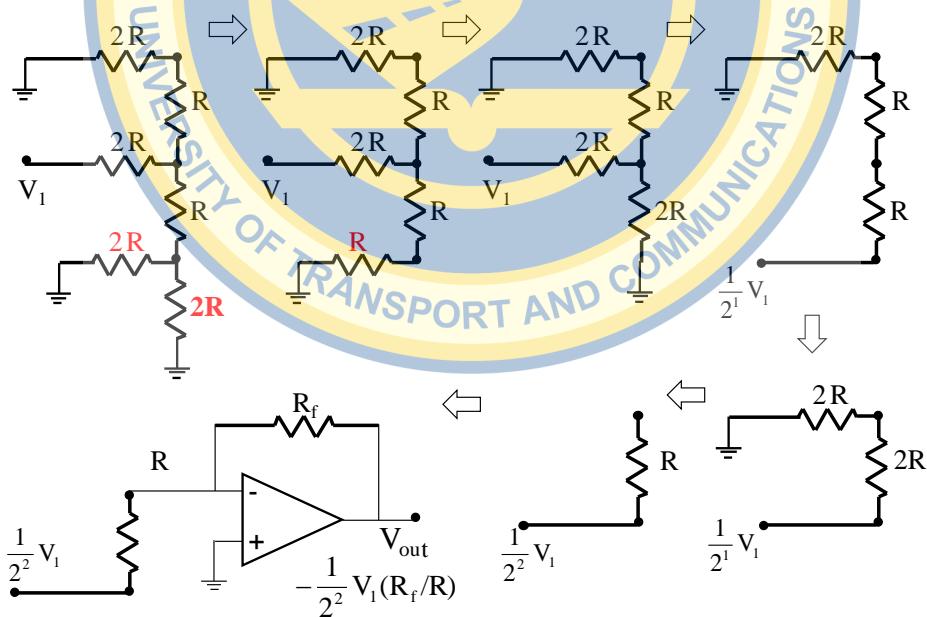
Hình 2.24. Mạch DAC 3 bit loại R-2R.

Áp dụng Định lý Thevenin các điện áp tác động đến đầu ra một cách riêng rẽ như sau:

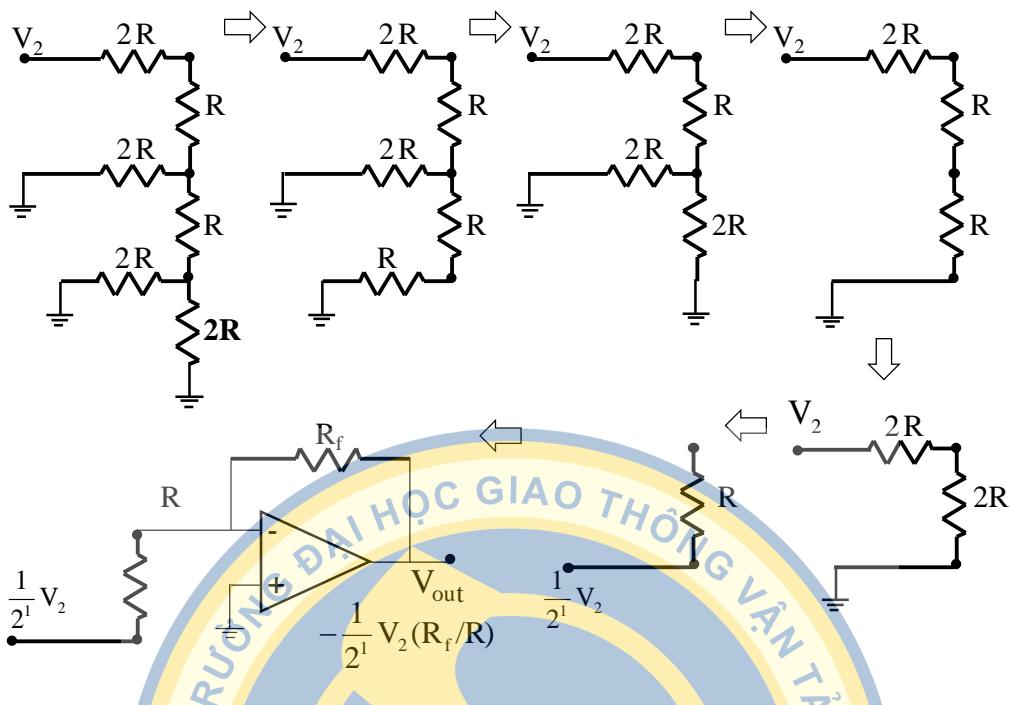
- Xét ảnh hưởng của V_0 :



- Xét ảnh hưởng của V_1 :



- Xét ảnh hưởng của V_2 :



Như vậy nếu xét của cả V_0 , V_1 , và V_2 ta có:

$$V_{out} = -R_f/R \left(\frac{V_0}{2^3} + \frac{V_1}{2^2} + \frac{V_2}{2^1} \right) \quad (2.17)$$

Xét đến tác động điều khiển của các bit D_0 , D_1 , và D_2

$$V_{out} = -R_f/R \left(\frac{R_{ref} D_0}{2^3} + \frac{R_{ref} D_1}{2^2} + \frac{R_{ref} D_2}{2^1} \right) \quad (2.18)$$

$$V_{out} = -V_{ref} R_f / R \left(\frac{D_0}{2^3} + \frac{D_1}{2^2} + \frac{D_2}{2^1} \right) \quad (2.19)$$

Tổng quát cho biến đổi n bit:

$$V_{out} = -V_{ref} R_f / R \left(\frac{D_0}{2^n} + \frac{D_1}{2^{n-1}} + \dots + \frac{D_{n-2}}{2^2} + \frac{D_{n-1}}{2^1} \right) \quad (2.20)$$

Công thức trên cho thấy thực chất DAC R-2R cũng là loại có trọng số nhị phân nhưng chỉ số lượng điện trở trong mạch phụ thuộc vào số bit, còn độ lớn của điện trở lại không phụ thuộc vào số bit. Phương pháp này thích hợp DAC nhiều bit.

Nếu chọn $R_f = R$:

$$V_{out} = -V_{ref} \left(\frac{D_0}{2^n} + \frac{D_1}{2^{n-1}} + \dots + \frac{D_{n-2}}{2^2} + \frac{D_{n-1}}{2^1} \right) \quad (2.21)$$

Ôn tập chương 2

Lý thuyết

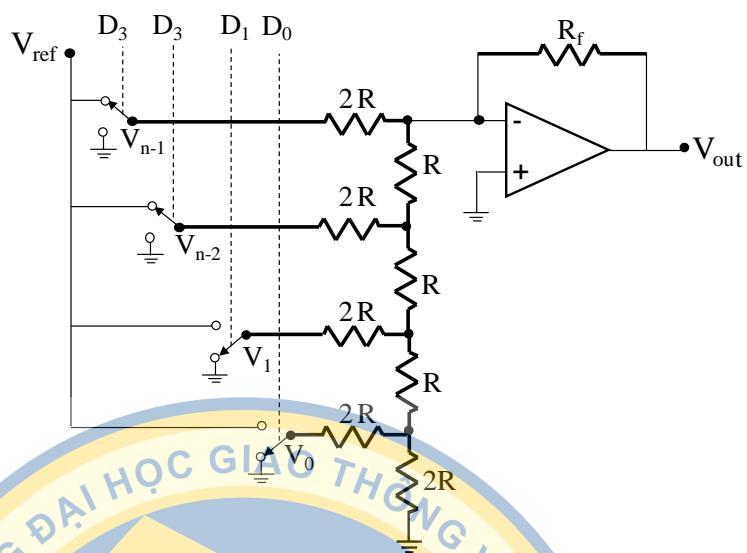
1. Nêu nguyên tắc hoạt động bộ chuyển đổi tương tự/số (ADC)?
2. Nêu các thông số kỹ thuật cơ bản của ADC?
3. Nêu nguyên tắc hoạt động bộ chuyển đổi số/tương tự (DAC)?
4. Nêu các thông số kỹ thuật cơ bản của DAC?

Bài tập

1. Xác định độ phân giải của bộ chuyển đổi ADC với điện áp toàn thang (FS) tương tự vào 3 V, với các trường hợp sau:
 - a. ADC 6 bit
 - b. ADC 8 bit
 - c. ADC 12 bit
 - d. ADC 16 bit
 - e. ADC 20 bit
 - f. ADC 24 bit
2. Một ADC 10 bit với điện áp toàn thang (FS) tương tự vào 4,5 V, xác định tín hiệu số ra trong các trường hợp:
 - a. Tín hiệu tương tự là 0,58 V
 - b. Tín hiệu tương tự là 1,52 V
 - c. Tín hiệu tương tự là 2,45 V
 - d. Tín hiệu tương tự là 3,06 V
 - e. Tín hiệu tương tự là 3,48 V
 - f. Tín hiệu tương tự là 4,24 V
3. Một bộ chuyển đổi DAC có tín hiệu tương tự đầu ra là 10.5 mV ứng với 00001010. Xác định tín hiệu tương tự khi tín hiệu số là
 - a. 00000100
 - b. 00011101
 - c. 10000101
 - d. 00110101
 - e. 10010100
 - f. 11111111

4. Cho mạch DAC như hình vẽ, $R = 5 \text{ k}\Omega$, $R_f = 10 \text{ k}\Omega$, $V_{ref} = 2,5 \text{ V}$, xác định điện áp V_{out} trong các trường hợp:

- a. 0000
- b. 0001
- c. 0010
- d. 0011
- e. 0100
- f. 0101
- g. 0110
- h. 0111
- i. 1000
- j. 1101
- k. 1111



Chương 3

ĐẠI SỐ LOGIC VÀ CÁC CÔNG LOGIC CƠ BẢN

3.1. GIỚI THIỆU

Như ta đã biết, các mạch số logic hoạt động ở chế độ nhị phân, nơi điện thế vào/ra sẽ có giá trị 0 hoặc 1. Việc chỉ định giá trị 0 và 1 biểu thị khoảng điện thế định sẵn. Đặc điểm này của mạch logic cho phép sử dụng đại số Boolean làm công cụ phân tích và thiết kế các hệ thống kỹ thuật số. Ta sẽ nghiên cứu mạch logic cơ bản nhất- công logic- là nền tảng để từ đó xây dựng nên tất cả các mạch logic và hệ thống kỹ thuật số khác. Chúng ta sẽ khảo sát cách mô tả và phân tích hoạt động của các công logic khác nhau và các mạch phức tạp, hình thành từ tổ hợp công logic, bằng đại số Boolean- cách sử dụng đại số Boolean đơn giản hóa biểu thức Boolean của mạch sao cho có thể xây dựng lại mạch bằng ít công logic nhất.

3.2. ĐẠI SỐ LOGIC

3.2.1. Biến và hàm logic

Đại số logic hay còn gọi là đại số Boole, do nhà bác học Georges Boole đề xuất năm 1854. Như đã đề cập, đại số logic là cơ sở toán học cho mạch điện tử số, máy tính số và rộng hơn là nền tảng kỹ thuật số.

- Trạng thái logic: là trạng thái của một thực thể nào đó, dưới góc độ logic thì một thực thể ở một trong hai trạng thái tồn tại hoặc không tồn tại.

- Biến logic: mô tả trạng thái logic của các thực thể, biến logic có giá trị là “0” hoặc “1”.

Thường được ký hiệu là các chữ cái: A, B, C, D, X₁, X₂, X₃...khi biến logic nhận giá trị “1” và \bar{A} , \bar{B} , \bar{C} , \bar{D} , \bar{X}_1 , \bar{X}_2 , \bar{X}_3 ...khi biến logic nhận giá trị “0”

- Hàm logic: diễn tả mối quan hệ của các biến, hàm logic cũng có giá trị là “0” hoặc “1”.

Hàm logic cũng thường được ký hiệu là các chữ cái: F₁, F₂, F₃, Y₁, Y₂, Y₃.....

Như vậy, F₁, F₂, F₃, Y₁, Y₂, Y₃, ..., = {0,1}

3.2.2. Biểu diễn hàm logic

3.2.2.1. Dưới dạng đại số

Một hàm logic với các tổ hợp biến khác nhau có thể biểu diễn bằng phương pháp đại số dưới hai dạng:

+ Tổng các tích (Sum of the product: SOP), còn được gọi là Chuẩn tắc tuyễn. Trong dạng SOP, hàm được biểu diễn dưới dạng dạng tổng (hoặc) của các tích (và), trong đó mỗi tích chứa một hay nhiều biến độc lập xuất hiện dưới dạng đảo hoặc không đảo.

Khi các tích chứa đầy đủ các biến gọi là tích chuẩn (Minterm), và hàm gọi là Tổng các tích đầy đủ (hay Chuẩn tắc tuyển đầy đủ)

Ví dụ 3.1: $F = AB + AC + BC$

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

$$\text{Dạng đầy đủ: } Z = \bar{A}\bar{B}C + ABC + A\bar{B}C$$

+ Tích các tổng (Product of the sum: POS), hay còn được gọi là Chuẩn tắc hội. Ở dạng POS, hàm được biểu diễn dưới dạng tích (và) của các tổng (hoặc), trong đó mỗi tổng chứa một hay nhiều biến độc lập xuất hiện dưới dạng đảo hoặc không đảo

Khi các tổng chứa đầy đủ các biến gọi là tổng chuẩn (Maxterm), và hàm gọi là Tích các tổng đầy đủ (hay Chuẩn tắc hội đầy đủ)

Ví dụ 3.2: $F_1 = (A+B)(A+C)(B+C)$

$$F_2 = (\bar{A} + C)(A + B)(\bar{B} + C)$$

$$\text{Dạng đầy đủ: } Z = (\bar{A} + \bar{B} + C)(A + B + C)(A + \bar{B} + C)$$

3.2.2.2. Dưới dạng bảng sự thật

Bảng sự thật để biểu diễn hàm logic có dạng như bảng 3.1, gồm các biến đặt bên trái và hàm đặt bên phải, trên các ô ghi tên biến, hàm, giá trị các biến, và giá trị hàm tương ứng.

Bảng 3.1. Cấu trúc bảng sự thật biểu diễn hàm logic.

Biến				Hàm			
A	B	C	...	F1	F2	F3	...
0/1	0/1	0/1	...	0/1	0/1	0/1	...
0/1	0/1	0/1	...	0/1	0/1	0/1	...
0/1	0/1	0/1	...	0/1	0/1	0/1	...
0/1	0/1	0/1	...	0/1	0/1	0/1	...
...

Ví dụ 3.3: Ví dụ một hàm 2 biến A, B, và hàm F với quan hệ thể hiện ở bảng sự thật sau:

A	B	F
0	0	1
0	1	0
1	0	1
1	1	0

Chú ý: dạng đại số và dạng bảng là tương đương, thực tế một hàm có thể biểu diễn theo dạng đại số, hay dạng bảng, hoặc cả hai.

Khi chuyển từ bảng sang đại số:

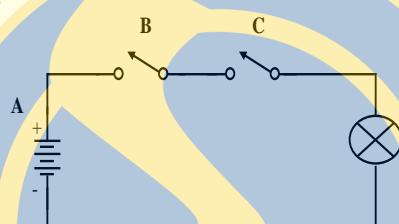
- Sang dạng tổng các tích SoP: Chỉ quan tâm đến các tổ hợp biến mà hàm có giá trị bằng 1, và số lần hàm bằng 1 chính là số tích của biểu thức, trong mỗi một tổng, các biến có giá trị bằng 1 được giữ nguyên, còn các biến có giá trị bằng 0 lấy phủ định.

Hàm trong bảng trên viết dưới dạng tổng các tích: $Z = \overline{A}\overline{B} + A\overline{B}$

- Sang dạng tích các tổng PoS: Chỉ quan tâm đến các tổ hợp biến mà hàm có giá trị bằng 0, và số lần hàm bằng 0 chính là số tổng của biểu thức, trong mỗi một tổng, các biến có giá trị bằng 0 được giữ nguyên, còn các biến có giá trị bằng 1 lấy phủ định.

Hàm trong bảng trên viết dưới dạng tích các tổng: $Z = (A + \overline{B})(\overline{A} + \overline{B})$

Ví dụ 3.4. Xét ví dụ điển hình và tổng quát hơn để mô tả biến và hàm logic: Một mạch điện hình 3.1, gồm một nguồn cấp cho một bóng đèn qua hai công tắc mắc nối tiếp, bóng đèn chỉ sáng khi cả 2 công tắc đều đóng.



Hình 3.1. Mạch cấp nguồn cho đèn.

Gọi A là biến mô tả nguồn điện, khi cấp nguồn ứng với logic 1 và khi không cấp nguồn ứng với logic 0.

Gọi B và C là biến mô tả công tắc, công tắc đóng ứng với logic 1 và hở ứng với logic 0.

Y là hàm chỉ trạng thái bóng đèn, đèn sáng ứng với logic 1 và tối ứng với logic 0.

- Quan hệ giữa hàm Y và các biến A, B, C dạng bảng như sau:

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

Nghĩa là đèn LED chỉ sáng khi được cấp nguồn và các công tắc đều đóng.

-Viết dưới dạng đại số:

+ Tổng các tích: $Y = ABC$

+ Tích các tổng:

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

3.2.3. Một số quy tắc và định luật cơ bản trong đại số logic

3.3.1. Quy tắc với các hằng số

Phép VÀ (AND): gọi A, B là biến logic

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Phép HOẶC (OR): gọi A, B là biến logic

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Phép PHỦ ĐỊNH (NOT):

A	\bar{A}
0	1
1	0

3.3.2. Quy tắc đối với hàm số 1 hoặc nhiều biến thường gặp

	Phép OR	Phép AND
Với hàm số một biến	$A + 1 = 1$	$A \cdot 0 = 0$
	$A + 0 = A$	$A \cdot 1 = A$
	$A + \bar{A} = 1$	$A \cdot \bar{A} = 0$
	$A = A + A + A + \dots$	$A = A \cdot A \cdot A \dots$
	$=$	
Định lý giao hoán	$A + B = B + A$	$A \cdot B = B \cdot A$

Định lý kết hợp	$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
Định lý phân phối	$A(B+C) = A \cdot B + A \cdot C$	$A+B \cdot C = (A+B) \cdot (A+C)$
Định lý hấp thụ	$A + A \cdot B = A$	$A \cdot (A + B) = A$
Định lý DeMorgan	$\overline{A + B} = \overline{A} \cdot \overline{B}$	$\overline{A \cdot B} = \overline{A} + \overline{B}$

Trong đó A, B, C là biến logic

Việc chứng minh các mệnh đề trên có thể thực hiện theo hai phương pháp :

- Phương pháp đại số (logic), nghĩa là biến đổi một vế cho bằng với vế kia.
- Sử dụng kết logic của cả hai vế và so sánh bằng cách lập bảng sự thật.

Ví dụ chứng minh định lý hấp thụ $A + A \cdot B = A$

+ Phương pháp đại số:

$$\text{Vế trái} = A + A \cdot B = A(1 + B) = A = \text{Vế phải}$$

+ Phương pháp so sánh kết quả logic:

A	B	AB	Vế trái	Vế phải
0	0	0	0	0
0	1	0	0	0
1	0	0	1	1
1	1	1	1	1

Từ bảng kết quả, ta thấy giá trị logic của vế trái luôn giống vế phải trong mọi trường hợp hay định lý đã được chứng minh.

3.3. CÁC CỘNG LOGIC

Mạch số được xây dựng từ các mạch điện tử nhỏ gọi là các cỗng logic (logic gate), mỗi cỗng logic biểu diễn một hàm logic cơ bản. Các cỗng logic hay còn gọi là các mạch logic cơ bản.

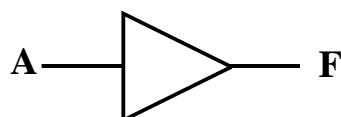
3.3.1. Cỗng ĐÚNG (TRUE hay Buffer (đệm) hay Transfer (truyền qua))

Bảng sự thật:

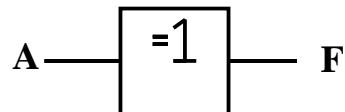
A	F = A
0	0
1	1

Ký hiệu trên mạch điện tử số:

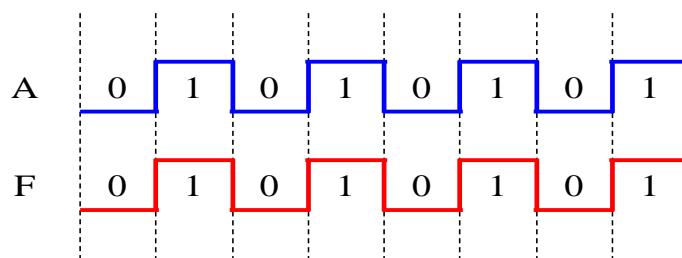
Ký hiệu chuẩn
(Chuẩn IEEE Std 91/91a)



Kiểu chữ nhật
(Chuẩn IEC 60617-12)



Ví dụ biểu diễn tín hiệu dưới dạng sóng

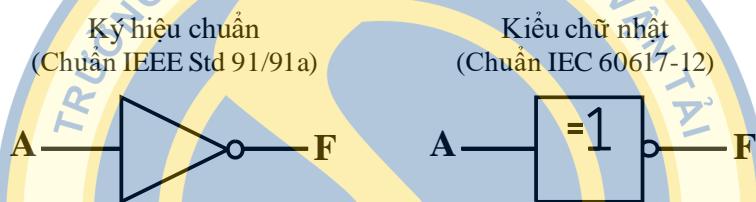


3.3.2. Cỗng ĐẢO (NOT)

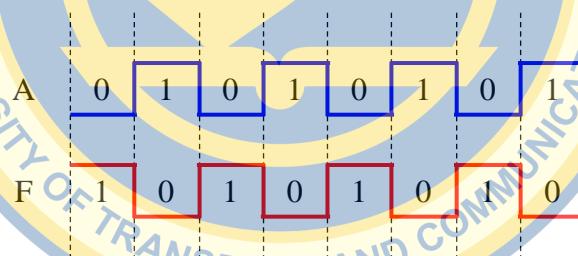
Bảng sự thật:

A	$F = \bar{A}$
0	1
1	0

Ký hiệu trên mạch điện tử số:



Ví dụ dạng tín hiệu đầu vào và đầu ra:



3.3.3. Cỗng HOẶC (OR)

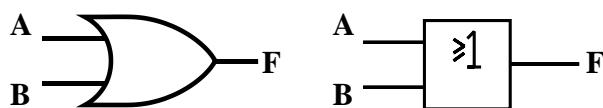
Hàm logic $F = A + B$

Bảng sự thật:

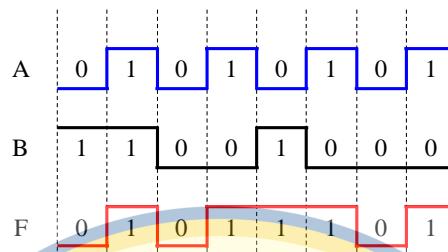
A	B	$F = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Ký hiệu trên mạch điện tử số:

Ký hiệu chuẩn
(Chuẩn IEEE Std 91/91a) Kiểu chữ nhật
(Chuẩn IEC 60617-12)



Ví dụ dạng tín hiệu đầu vào và đầu ra:



3.3.4. Cổng VÀ (AND)

Hàm logic $F = A \cdot B$

Bảng sự thật:

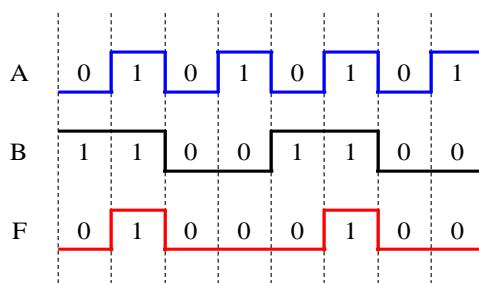
A	B	$F = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Ký hiệu trên mạch điện tử số:

Ký hiệu chuẩn
(Chuẩn IEEE Std 91/91a) Kiểu chữ nhật
(Chuẩn IEC 60617-12)



Ví dụ dạng tín hiệu đầu vào và đầu ra:



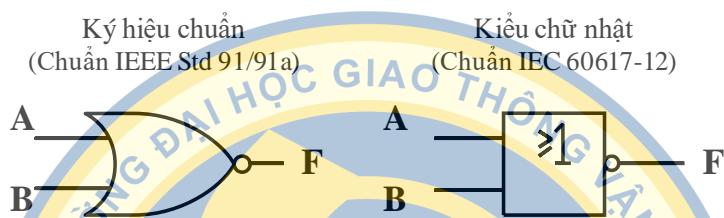
3.3.5. Cỗng HOẶC-ĐẢO (NOR: OR+NOT)

Hàm logic $F = \overline{A+B}$

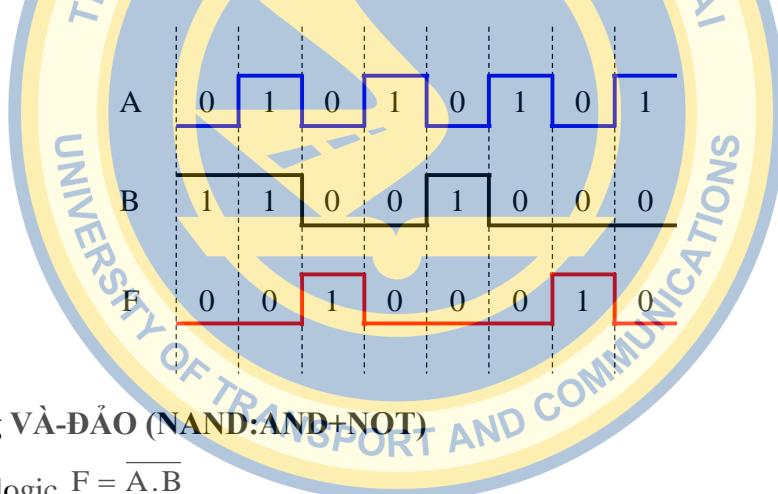
Bảng sự thật:

A	B	$F = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Ký hiệu trên mạch điện tử số:



Ví dụ dạng tín hiệu đầu vào và đầu ra:



3.3.6. Cỗng VÀ-ĐẢO (NAND: AND+NOT)

Hàm logic $F = \overline{A \cdot B}$

Bảng sự thật:

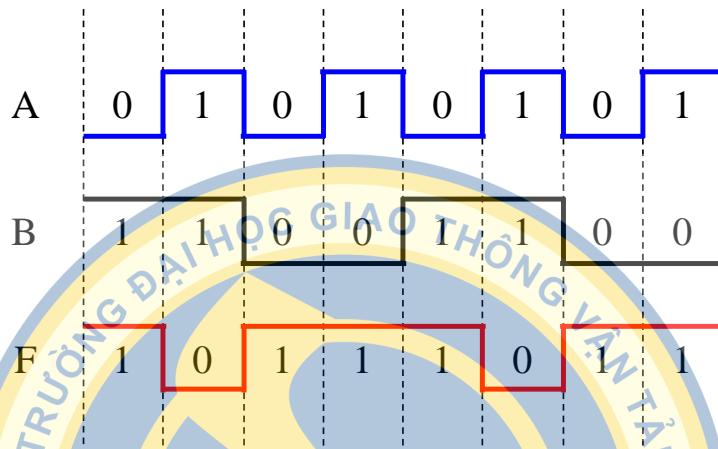
A	B	$F = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

Ký hiệu trên mạch điện tử số:

Ký hiệu chuẩn
(Chuẩn IEEE Std 91/91a) Kiểu chữ nhật
(Chuẩn IEC 60617-12)



Ví dụ dạng tín hiệu đầu vào và đầu ra:



3.3.7. Cổng HOẶC-LOẠI TRÙ (XOR: EXCLUSIVE-OR)

Cổng XOR còn được gọi là cổng EX-OR, là cổng có logic:

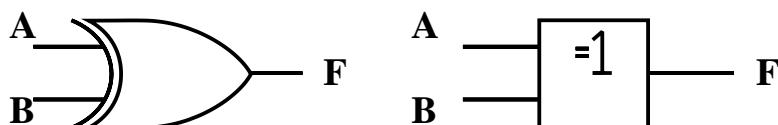
$$F = A \oplus B = \overline{AB} + A\overline{B}$$

Bảng sự thật:

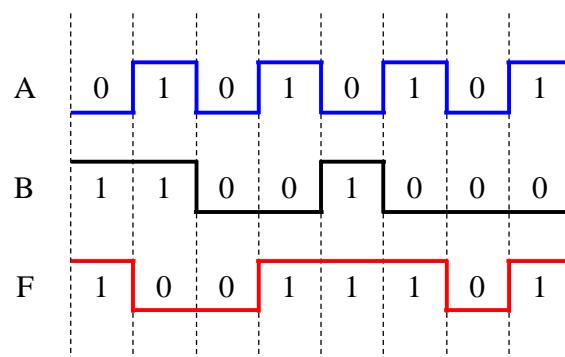
A	B	$F = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Ký hiệu trên mạch điện tử số:

Ký hiệu chuẩn
(Chuẩn IEEE Std 91/91a) Kiểu chữ nhật
(Chuẩn IEC 60617-12)



Ví dụ dạng tín hiệu đầu vào và đầu ra:



3.3.8. Cổng NOR-LOẠI TRÙ (XNOR: EXCLUSIVE-NOR)

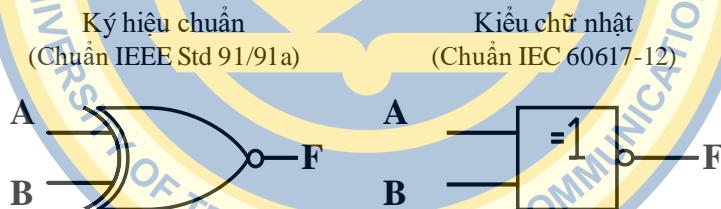
Cổng NOR còn được gọi là cổng EX-NOR, là cổng có logic:

$$F = \overline{A \oplus B} = \overline{AB} + AB$$

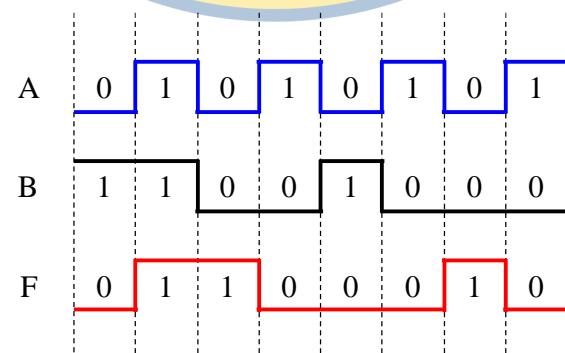
Bảng sự thật:

A	B	$F = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Ký hiệu trên mạch điện tử số:



Ví dụ dạng tín hiệu đầu vào và đầu ra:



3.3.9. Cỗng NGỤ Ý (IMPLY)

Hàm logic “ngụ ý” $F = \overline{A} + B$, đây là hàm quan trọng để xây dựng hệ thống mạch điện tử số kiểu mới sử dụng memristor.

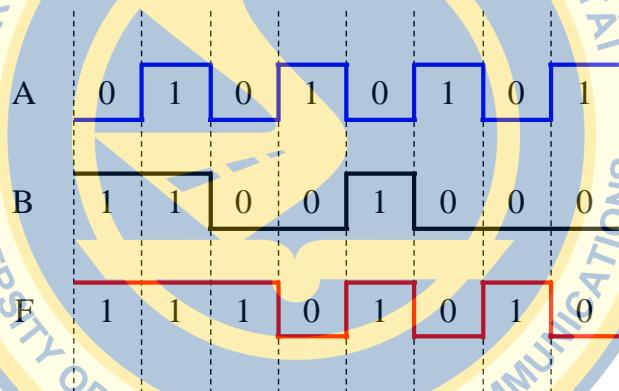
Bảng sự thật:

A	B	$F = \overline{A} + B$
0	0	1
0	1	1
1	0	0
1	1	1

Ký hiệu trên mạch điện tử số:



Ví dụ dạng tín hiệu đầu vào và đầu ra:



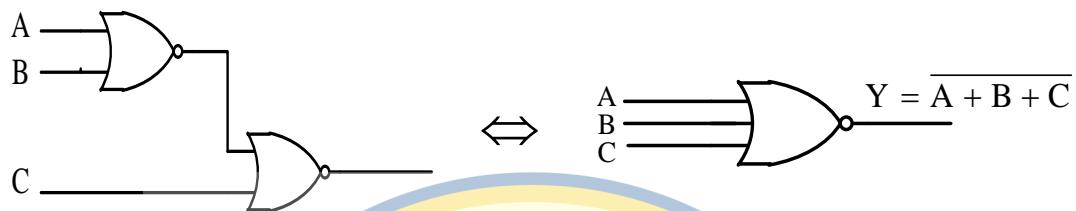
Chú ý:

- Giống như điện trở, tụ điện, cuộn cảm, diode, và transistor trong mạch điện tử tương tự, các cỗng logic trình bày trên được coi là các linh kiện cơ bản trong điện tử số, qua đó các mạch điện tử số từ đơn giản đến phức tạp có thể xây dựng dựa trên các phần tử cơ bản trên. Trong đó cỗng NAND và NOR được gọi là các cỗng đa năng thích hợp cho việc sử dụng trong kiến trúc mạch bù sê đề cập trong chương 4.

- Hiện nay, chuẩn IEEE Std 91/91a là chuẩn được sử dụng phổ biến để biểu diễn cỗng logic số, trong giáo trình này chúng tôi cũng biểu diễn mạch điện tử số với ký hiệu cỗng logic theo chuẩn này.

- Các cỗng OR, NAND, AND, NOR, XOR, và XNOR có thể mở rộng cho n đầu vào nhiều đầu vào khi sử dụng luật giao hoán và kết hợp.

Ví dụ:



Bảng tổng hợp các cổng logic thường dùng:

Cổng	Ký hiệu	Hàm ra	Bảng sự thật															
NOT		$F = \overline{A}$	<table border="1"> <tr> <td>A</td><td>F</td></tr> <tr> <td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td></tr> </table>	A	F	0	1	1	0									
A	F																	
0	1																	
1	0																	
OR		$F = A + B$	<table border="1"> <tr> <td>A</td><td>B</td><td>F</td></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
AND		$F = A \cdot B$	<table border="1"> <tr> <td>A</td><td>B</td><td>F</td></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
NOR		$F = \overline{A + B}$	<table border="1"> <tr> <td>A</td><td>B</td><td>F</td></tr> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

NAND		$F = \overline{AB}$	<table border="1" data-bbox="1049 242 1330 460"> <tr><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
XOR		$F = \overline{A}\overline{B} + \overline{A}B \\ = A \oplus B$	<table border="1" data-bbox="1049 511 1330 729"> <tr><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
XNOR		$F = \overline{AB} + \overline{A}\overline{B} \\ = A \oplus B$	<table border="1" data-bbox="1049 781 1330 999"> <tr><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

3.4. CHUYỂN ĐỔI CÔNG ĐA DỤNG THÀNH CÁC CÔNG CƠ BẢN

Các công logic cơ bản khác đều có thể xây dựng từ một công duy nhất NAND hoặc NOR, vì vậy NAND và NOR được gọi là các công đa dụng:

3.4.1. Xây dựng từ NAND

1. Công NOT

Ta có, $F = \overline{A}$, theo định lý của đại số logic ta có $A = AA$, nên $F = \overline{AA}$.

Vì vậy nếu ngắn mạch hai đầu vào thì NAND sẽ biến thành NOT



2. Công OR

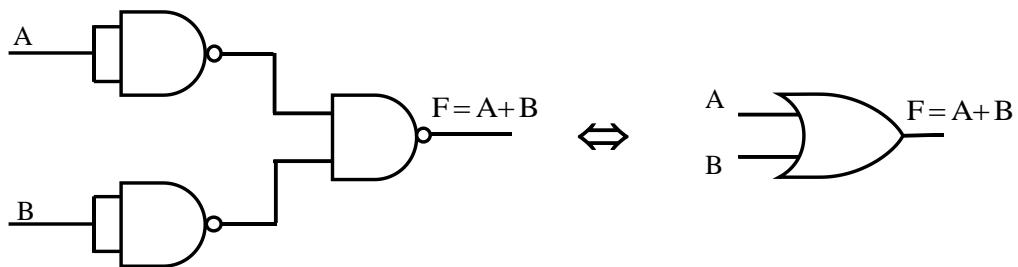
Ta có, $F = A + B$

Hàm tương đương với $F = \overline{\overline{A} + \overline{B}}$,

Theo định lý DeMorgan:

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

OR có thể tương đương với mạch chỉ sử dụng NAND



3. Cổng AND

Ta có, $F = AB$

Hàm tương đương với $F = \overline{\overline{AB}}$,

AND có thể tương đương với mạch chỉ sử dụng NAND

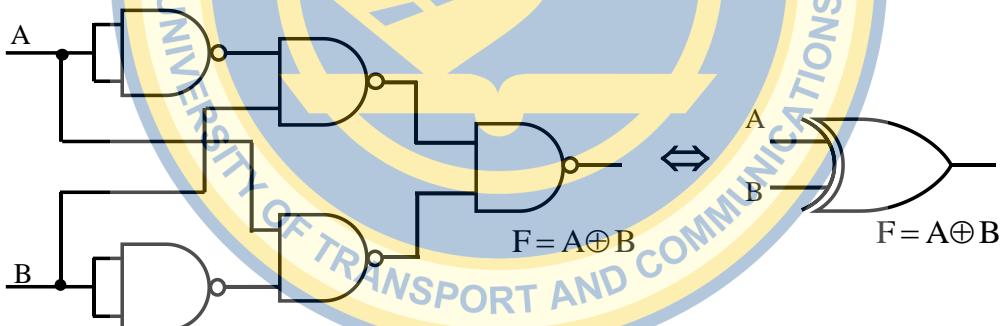


4. Cổng XOR

Ta có, $F = A \oplus B = \overline{AB} + AB$

Hàm tương đương với $F = \overline{A}B + A\overline{B} = \overline{\overline{A}\overline{B}} + \overline{A}\overline{B} = \overline{\overline{ABA}} = \overline{\overline{(AAB)(ABA)}} = (AAB)(ABA)$

Cổng XOR có thể tương đương với mạch chỉ sử dụng NAND:

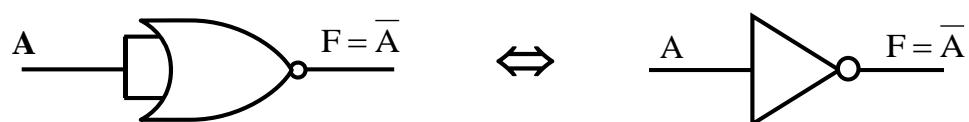


3.4.2. Xây dựng từ NOR:

1. Cổng NOT

Ta có, $F = \overline{A}$, theo định lý của đại số logic ta có $A = A + A$, nên $F = \overline{A + A}$.

Vì vậy nếu ngắn mạch hai đầu vào thì NOR sẽ biến thành NOT

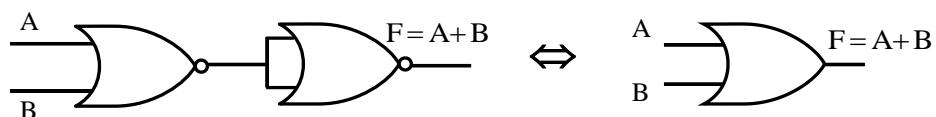


2. Cổng OR

Ta có, $F = A + B$,

Hàm tương đương với $F = \overline{\overline{A} + \overline{B}}$,

OR có thể tương đương với mạch chỉ sử dụng NOR

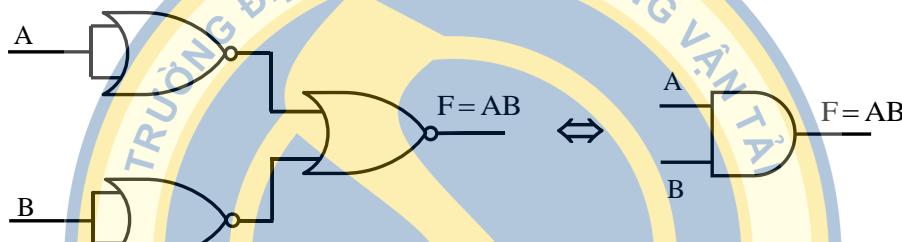


3. Cổng AND

Ta có, $F = AB$,

Hàm tương đương với $F = \overline{AB} = \overline{A} + \overline{B}$,

AND có thể tương đương với mạch chỉ sử dụng NOR



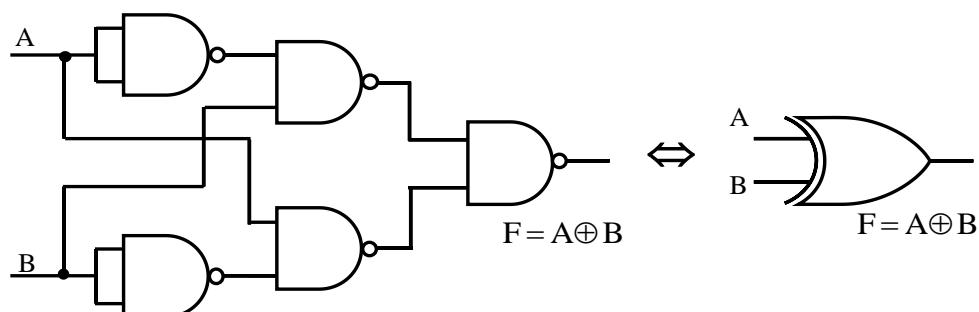
4. Cổng XOR

Ta có, $F = A \oplus B = \overline{AB} + A\overline{B}$

Hàm tương đương với:

$$\begin{aligned} F &= \overline{A}B + A\overline{B} = \overline{A}\overline{B} + A\overline{B} = \overline{A} + \overline{B} + A + \overline{B} = A + \overline{B} + \overline{A} + B = A + \overline{BB} + \overline{AA} + B \\ &= (\overline{A} + BB) + (\overline{AA} + B) \end{aligned}$$

Cổng XOR có thể tương đương với mạch chỉ sử dụng NOR:



Ôn tập chương 3

Lý thuyết

- Nêu một số quy tắc và định luật cơ bản trong đại số logic.
- Phân biệt các loại công logic: ký hiệu, bảng chân lý, giản đồ xung.
- Nêu các phương pháp đại số để biểu diễn một hàm logic bất kỳ.

Bài tập

1. Xây dựng bảng sự thật, từ biểu thức sau:

- a) $F = AC + AB$ c) $F = A(\bar{B} + \bar{C})$
b) $F = AB(B + C + \bar{D})$ d) $F = (A + B + C)AB$

2. Chuyển sang kiểu PoS các hàm sau

- a) $F = \bar{ABC} + \bar{ABC} + \bar{ABC} + \bar{ABC}$
b) $F = ABC + \bar{ABC} + \bar{ABC} + \bar{ABC} + \bar{ABC} + \bar{ABC} + \bar{ABC}$
c) $F = ABDCD + AB(\bar{CD}) + (\bar{AB})CD$
d) $F = AB + A(B + C) + B(B + C)$
e) $F = A + AB + \bar{ABC}$
f) $F = ABDCD + AB(\bar{CD}) + (\bar{AB})CD$

3. Xây dựng hàm logic dưới dạng đại số và bảng với hàm Y có bốn biến A, B, C, D cho các trường hợp sau:

Hàm nhận giá trị 1 khi có hai biến nhận giá trị 0, các trường hợp khác hàm nhận giá trị 0.

Hàm nhận giá trị 1 khi có nhiều hơn 3 biến nhận giá trị 1, các trường hợp khác hàm nhận giá trị 0.

Hàm nhận giá trị 0 khi có nhiều hơn hai biến nhận giá trị 1, các trường hợp khác hàm nhận giá trị 1.

4. Chuyển sang kiểu SoP và PoS cho hàm logic có giá trị như sau

(a)

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

(b)

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

(c)

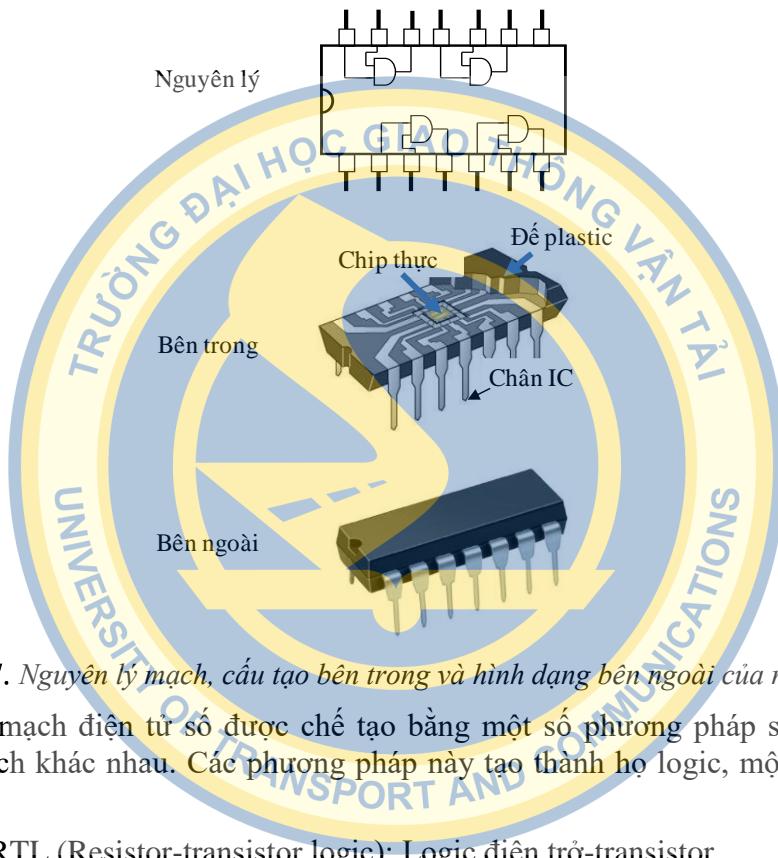
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Chương 4

VI MẠCH SỐ

4.1. GIỚI THIỆU

Các công logic và mạch điện tử số được chế tạo như các mạch tích hợp (IC: Integrated circuit) sử dụng các linh kiện: điện trở, tụ điện, diode, transistor kết nối thành mạch điện tử. Chip này được gắn trên phiến (wafer) silic và đóng gói để cách ly với môi trường để không làm thay đổi tính chất điện của bán dẫn cũng như bảo vệ về cơ khí cho mạch bên trong. Đầu ra, đầu vào, đường cấp nguồn, đất,...được nối với các chân bên ngoài IC.



Hình 4.1. Nguyên lý mạch, cấu tạo bên trong và hình dạng bên ngoài của một IC số.

– Vì mạch điện tử số được chế tạo bằng một số phương pháp sản xuất kiểu thiết kế mạch khác nhau. Các phương pháp này tạo thành họ logic, một số họ logic phổ biến:

- + RTL (Resistor-transistor logic): Logic điện trở-transistor.
- + DTL (Diode-transistor logic): Logic diode-transistor.
- + TTL (Transistor-transistor logic): Logic transistor-transistor.
- + ECL (Emitter-coupled logic): Logic kết hợp emitter.
- + PMOS (P-type Metal oxide semiconductor): Logic bán dẫn-oxit-kim loại kênh P.
- + NMOS (N-type Metal oxide semiconductor): Logic bán dẫn-oxit-kim loại kênh N.

- + CMOS (Complementary metal oxide semiconductor): Logic bán dẫn-oxit-kim loại bù.
- + Bi-CMOS (Bipolar-CMOS): CMOS lưỡng cực.

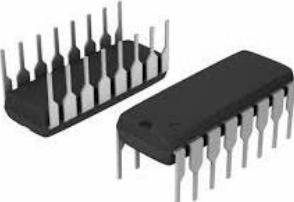
Họ logic RTL và DTL được chế tạo trong thời kỳ đầu của ngành công nghiệp vi mạch số, hiện nay ít khi được sử dụng trong các thiết kế mới. Thay vào đó là các hệ TTL, CMOS, Bi-CMOS.

– Phân loại theo độ tích hợp: Theo tiêu chí này vi mạch được phân loại dựa trên số transistor hay cổng logic trên một đơn vị diện tích, trong đó đơn vị diện tích không được quy định một cách chính xác, nhưng tùy theo nhà sản xuất nó nằm trong khoảng vài mm² đến khoảng 450 mm². Độ tích hợp tuân theo định luật Moore (số transistor trên một đơn vị diện tích tăng gấp đôi theo năm).

Bảng 4.1. Phân loại IC theo độ tích hợp.

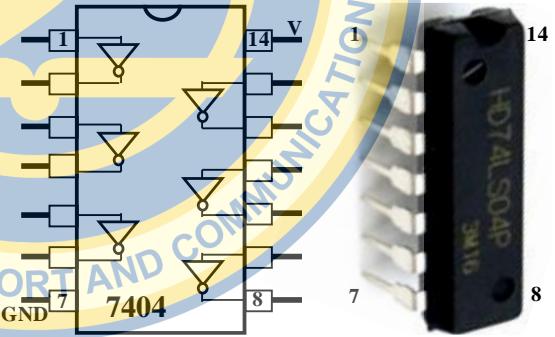
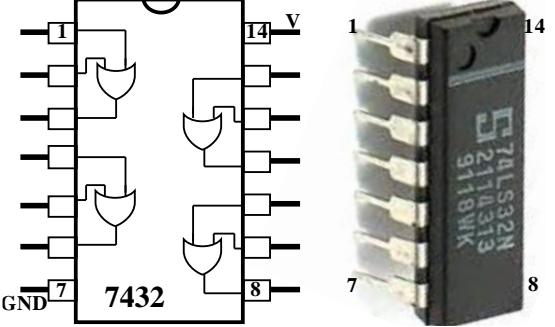
Loại vi mạch	Số transistor	Số cổng	Ứng dụng điển hình
Tích hợp cỡ nhỏ (SSI: small scale integration)	1 – 10	1 – 12	Công logic số cơ bản
Tích hợp cỡ vừa (MSI: Medium scale integration)	10 – 500	12 – 99	<ul style="list-style-type: none"> Mạch tổ hợp Mạch dãy Vi xử lý
Tích hợp cỡ lớn (LSI: Large scale integration)	500 – 20.000	100 – 9999	
Tích hợp cỡ lớn (VLSI: Very large scale integration)	20.000 – 1.000.000	10.000 – 99.999	<ul style="list-style-type: none"> Vi xử lý Vi điều khiển
Tích hợp cỡ cực lớn (ULSI: Ultra large-scale integration)	> 1.000.000	> 100.000	

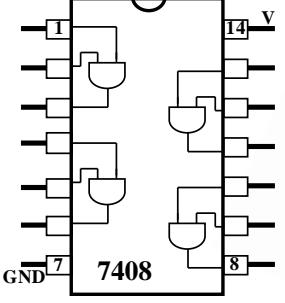
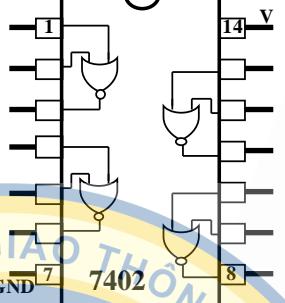
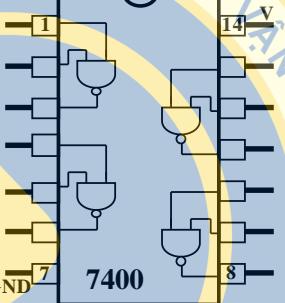
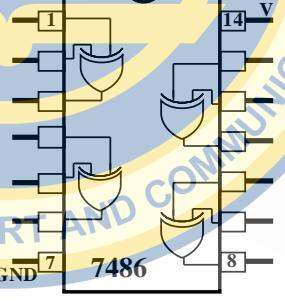
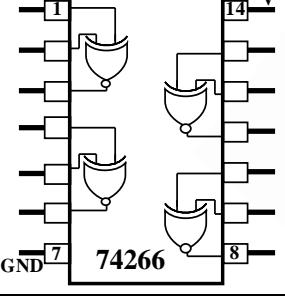
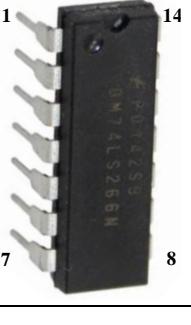
– Phân loại theo kiểu cách đóng vỏ:

Kiểu cách đóng vỏ	Ví dụ	Cách thi công mạch
DIP (Dual In Line Package) Hai hàng chân		Hàn thường (xuyên lỗ): tạo thành các lỗ trên mạch in PCB để đặt và gắn IC.

LCC (Leadless Chip Carrier) Không có chân nối ra		
PQFP (Plastic Quad Flat Package): Kiểu bốn hàng chân		Hàn dán: IC được đặt và gắn trực tiếp trên bề mặt PCB (không cần lỗ).
PGA (Pin Grid Array): Nhiều hàng chân		

– Các IC số logic kiểu đóng gói hai hàng chân thường gặp:

Công	Ký hiệu	IC số logic theo kiểu đóng gói hai hàng chân
NOT		
OR		

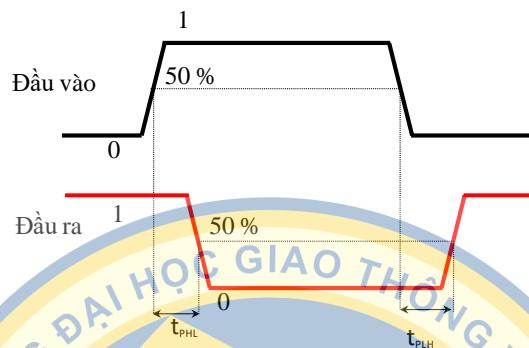
AND			
NOR			
NAND			
XOR			
XNOR			

4.2. CÁC ĐẶC TÍNH CỦA VI MẠCH SỐ

Các họ vi mạch số khác nhau thường được đánh giá bằng cách so sánh đặc tính các cổng cơ bản mỗi họ IC. Các đặc tính quan trọng của họ vi mạch số là:

4.2.1. Trễ truyền đạt

Dạng sóng mô tả ở chương 3 là dạng lý tưởng, nghĩa là tín hiệu đầu ra gần như ngay lập tức thay đổi theo tín hiệu đầu vào. Thực tế, trễ truyền đạt là khoảng thời gian để đầu ra của mạch có đáp ứng đồng với sự thay đổi mức logic của đầu vào. Thông số này điều chỉnh tốc độ hoạt động của một mạch logic.



Hình 4.2. Xác định thời gian trễ truyền đạt.

Có hai khoảng thời gian trễ do truyền được xác định:

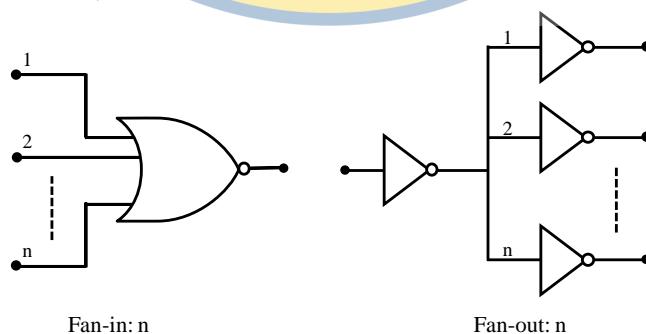
- t_{PLH} : là thời gian trễ khi tín hiệu chuyển từ logic 0 (Low) đến logic 1 (High).
- t_{PHL} : là thời gian trễ khi tín hiệu chuyển từ logic 1 (High) đến logic 0 (Low).

Lưu ý rằng t_{PHL} là khoảng thời gian trễ khi đầu ra chuyển từ cao xuống thấp, đo giữa thời điểm xung vào và xung ra thay đổi trạng thái được 50%. Còn t_{PLH} là khoảng trễ đáp ứng lại đầu ra khi nó chuyển từ thấp lên cao. Nhìn chung, t_{PLH} và t_{PHL} không nhất thiết bằng nhau, và cả hai thay đổi tùy thuộc điều kiện tái.

Tham số thời gian trễ trung bình t thường được dùng để đánh giá độ trễ truyền đạt của IC số, nó phụ thuộc chủ yếu vào thời gian chuyển mạch của từng linh kiện chế tạo chip như BJT, MOSFET.

$$t = \frac{t_{PLH} + t_{PHL}}{2} \quad (4.1)$$

4.2.2. Hệ số tải đầu vào, ra



Hình 4.3. Mô hình hệ số tải vào Fan-in và hệ số tải ra Fan-out.

- Hệ số tải đầu vào (Fan-in): là số đầu vào lớn nhất của mạch mà vẫn đảm bảo mạch làm việc tin cậy. Ví dụ, số tải lớn nhất đầu vào là 5 nối đến một cổng logic mà vẫn đảm bảo mạch hoạt động tin cậy, khi đó hệ số tải là 5.
- Hệ số tải đầu ra (Fan-out): là số tải có thể nối được với đầu ra của mạch mà vẫn đảm bảo được sự hoạt động tin cậy, Ví dụ, số tải lớn nhất đầu ra nối đến 6 cổng logic mà vẫn đảm bảo mạch hoạt động tin cậy, khi đó hệ số tải là 6.

$$\text{Fan-out}_L = \frac{I_{OL}(\text{max})}{I_{IL}(\text{max})}$$

$$\text{Fan-out}_H = \frac{I_{OH}(\text{max})}{I_{IH}(\text{max})}$$

Để xác định hệ số tải ra, ta cần biết khả năng kích thích dòng của đầu ra (I_{OL} max và I_{OH} max) và yêu cầu dòng mỗi đầu vào (I_{IL} và I_{IH}). Thông số này được đưa ra trong bảng tra cứu dữ liệu IC. Tải đầu ra là giá trị nhỏ hơn nếu Fan-out_L khác Fan-out_H nhau.

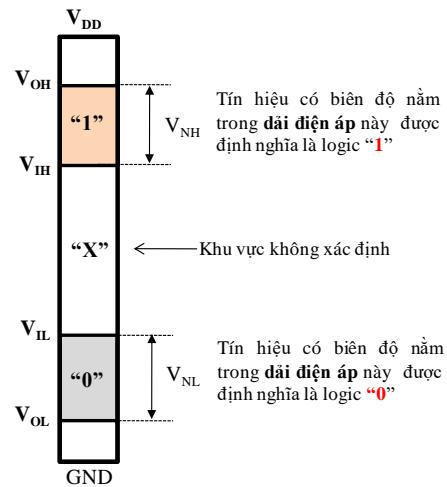
4.2.3. Tham số dòng điện và điện áp

Tham số dòng điện:

- Dòng điện đầu vào mức cao (I_{IH}): là dòng điện đến đầu vào khi điện áp mức cao ứng với logic 1 đưa đến.
- Dòng điện đầu vào mức thấp (I_{IL}): là dòng điện đưa đến đầu vào khi mức điện áp thấp ứng với logic 0 đưa đến.
- Dòng điện đầu ra mức cao (I_{OH}): là dòng điện tại đầu ra khi đầu ra có mức điện áp cao ứng với logic 1.
- Dòng điện đầu ra mức thấp (I_{OL}): là dòng điện tại đầu ra khi đầu ra có mức điện áp thấp ứng với logic 0.

Tham số điện áp:

- Điện áp đầu vào mức cao (V_{IH}): mức điện áp nhỏ nhất cho logic 1 tại đầu vào.
- Điện áp đầu vào mức thấp (V_{IL}): mức điện áp lớn nhất cho logic 0 tại đầu vào.
- Điện áp đầu ra mức cao (V_{OH}): mức điện áp nhỏ nhất cho logic 1 tại đầu ra.
- Điện áp đầu ra mức thấp (V_{OL}): mức điện áp lớn nhất cho logic 0 tại đầu ra.



Hình 4.4. Mô tả về tham số điện áp và lè nhiễu của IC số.

Chú ý: dải điện áp từ V_{IL} đến V_{IH} là **khu vực không xác định** hay khoảng bảo vệ, để tránh nhầm giữa logic 0 và 1.

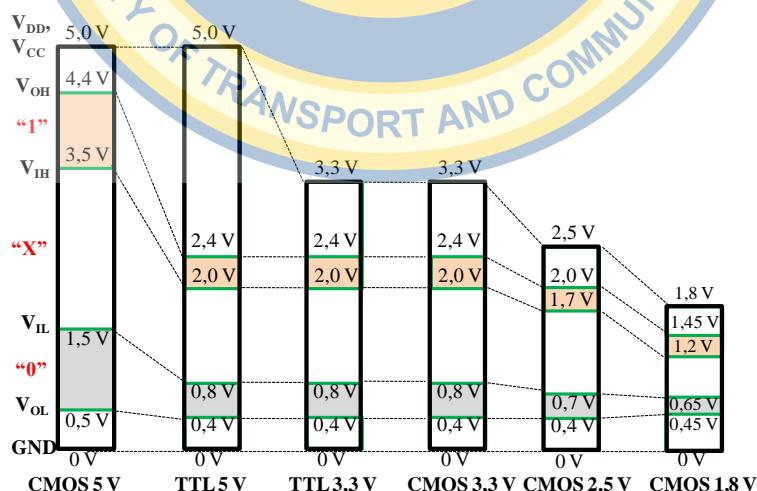
4.2.4. Tính miền nhiễu

Tính miền nhiễu là khả năng chống nhiễu của IC và được xác định bởi lè nhiễu. Lè nhiễu là sự chênh lệch giữa V_{OH} với V_{IH} và V_{OL} với V_{IL} nên ta có hai đại lượng lè nhiễu:

- Lè nhiễu mức cao: $V_{NH} = V_{OH} - V_{IH}$
- Lè nhiễu mức thấp: $V_{NL} = V_{IL} - V_{OL}$

Hình 4.5 mô tả phương pháp xác định lè nhiễu cao và thấp. Từ sơ đồ này có thể thấy tín hiệu số 0 hay 1 được xác định trong một dải tín hiệu chứ không phải một giá trị nhất định như tín hiệu tương tự, nghĩa là khi gấp nhiều cộng vào tín hiệu thực, khả năng giải mã ra các giá trị logic ít bị mất đi, đây chính là một trong những ưu thế của tín hiệu số trong bao toàn thông tin cũng như truyền dẫn.

4.2.5. Nguồn cấp



Hình 4.5. Nguồn cấp, mức điện áp, lè nhiễu của một số họ vi mạch điển hình.

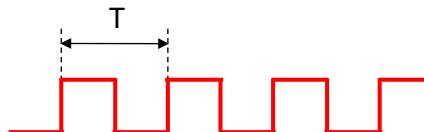
Nguồn cấp cho IC số là nguồn 1 chiều. Các họ logic khác nhau thì thông số này cũng khác nhau. Các giá trị điển hình: 5 V; 3,3 V; 2,5 V; 1,8 V,.. Người thiết kế cần xem xét thông số này để chọn nguồn cấp phù hợp. Xu hướng hiện nay nguồn cấp càng ngày càng giảm do yêu cầu về giảm công suất tiêu thụ, đặc biệt là cho các ứng dụng di động.

4.2.6. Tần số làm việc

Tần số làm việc F được tính là tần số tín hiệu vào, nói lên tốc độ làm việc của vi mạch, tần số làm việc tỉ lệ với tốc độ xử lý tín hiệu và công suất tiêu tán.

$$F = \frac{1}{T}$$

T : Chu kỳ tín hiệu



4.2.7. Công suất tiêu tán

Công suất tiêu tán P gồm 2 thành phần công suất tĩnh và công suất động:

$$P = P_Q + P_D \quad (4.2)$$

+ P_D : Công suất tĩnh, sinh ra bởi dòng rò

$$P_D = VI_r \quad (4.3)$$

Trong đó:

- V là nguồn cung cấp cho IC, với TTL ký hiệu là V_{CC} , với CMOS ký hiệu là V_{DD} .

- I_r là dòng rò

- + P_D : Công suất động

Công suất động P_D phụ thuộc vào đặc tính của linh kiện làm IC, tần số tín hiệu số, thời gian tăng hạ sườn xung tín hiệu số:

$$P_D = V^2 \left((C_{pd} F_{in} n) + \sum_{j=1}^N (C_{L,j} F_{out,j}) \right) \quad (4.4)$$

Trong đó:

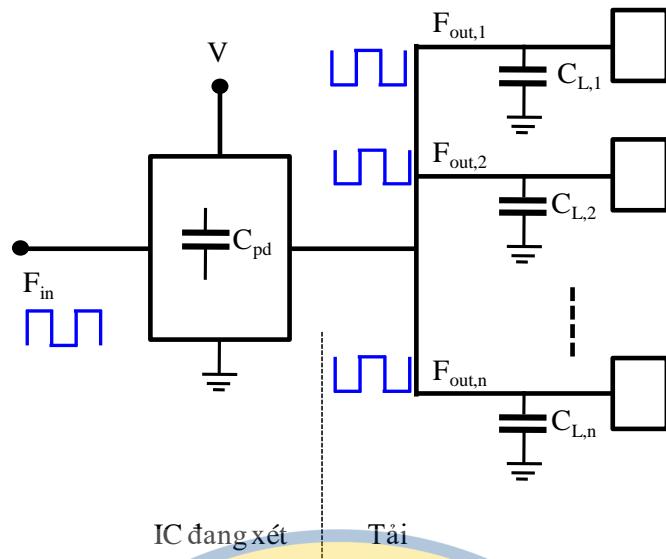
- C_{pd} : Điện dung tiêu tán công suất. Do điện dung ký sinh giữa các điện cực của diode hay transistor. Trong quá trình chuyển mạch, dòng qua sẽ làm các tụ này nạp và phóng điện, gây nên tiêu tán công suất.

- F_{in} : Tần số tín hiệu đưa vào.

- n : số bit chuyển trạng thái trong một thời điểm.

- $F_{out,j}$: Tần số tín hiệu đưa ra tải thứ j (tải ở đây hiểu là tầng nối sau của IC số đang xét)

- $C_{L,j}$: Điện dung tải thứ j (tải ở đây hiểu là tầng nối sau của IC số đang xét)



Hình 4.6. Mô hình các tham số ảnh hưởng đến công suất động của IC số.

Như vậy công suất tiêu tán được xác định bằng công thức:

$$P = VI_r + V^2 \left((C_{pd} F_{in} n) + \sum_{j=1}^N (C_{L,j} F_{out,j}) \right) \quad (4.5)$$

Khi xét 1 bit và IC số nối với 1 đầu ra và coi tần số ra bằng tần số vào:

$$P = VI_r + V^2 F (C_{pd} + C_L) \quad (4.6)$$

4.2.8. Dải nhiệt độ

Hầu hết các loại IC đều được tính toán để có thể hoạt động trong dải nhiệt độ khá rộng:

- + Ứng dụng trong thương mại: $0^{\circ}\text{C} \leftrightarrow +70^{\circ}\text{C}$.
- + Ứng dụng trong công nghiệp: $-40^{\circ}\text{C} \leftrightarrow +85^{\circ}\text{C}$.
- + Ứng dụng trong quân sự: $-55^{\circ}\text{C} \leftrightarrow +125^{\circ}\text{C}$

4.3. CÁC HỌ LOGIC SỬ DỤNG DIODE VÀ TRANSISTOR LUÔNG CỤC

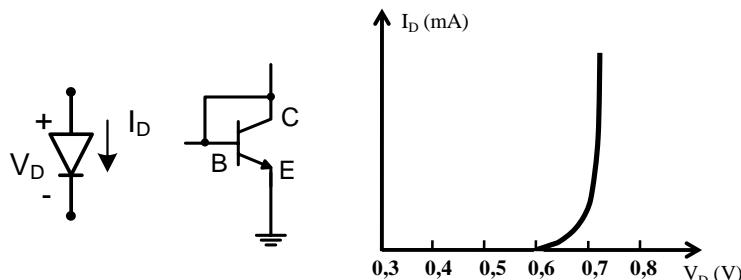
Năm họ logic sử dụng Diode hoặc Transistor là: RTL, DTL, TTL, ECL, IIL. Hai họ logic MOS và CMOS sử dụng Transistor hiệu ứng trường gọi tắt là MOSFET hay MOS.

4.3.1. Đặc tính diode

Diode được chế tạo một cách độc lập hoặc xây dựng từ transistor bằng cách ngắn mạch điện cực base-collector (khi đó diode được xem như tiếp giáp base-emitter của transistor).

Mối quan hệ giữa dòng I_D qua diode với điện áp thuận V_D (tương đương với V_{BE} nếu diode xây dựng từ transistor) được thể hiện ở hình 4.7.

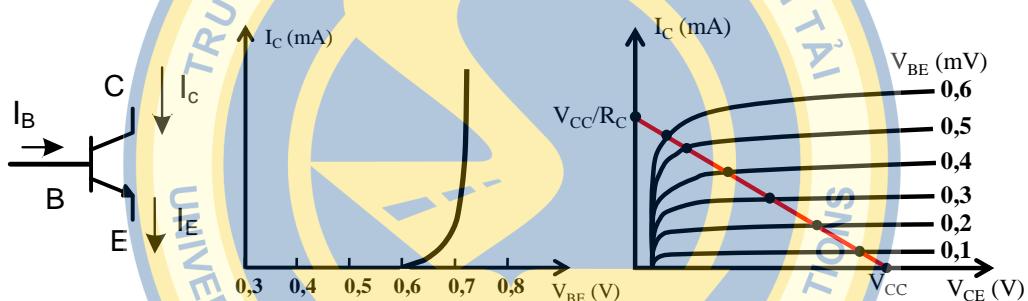
- Khi V_D (hay V_{BE}) < 0,6 V diode ở trạng thái ngắt.
 - Khi $0,6 V \leq V_D$ (hay $V_{BE} \leq 0,7 V$) dòng tăng mạnh.
 - Và khi V_D (hay $V_{BE} \geq 0,7 V$), dòng bão hòa, diode ở trạng thái đóng.
- Đặc tính đóng ngắt này của diode được sử dụng để thiết kế mạch logic số.



Hình 4.7. Đặc tuyến dòng áp của diode lý tưởng.

4.3.2. Đặc tính transistor lưỡng cực

Transistor lưỡng cực (BJT) gồm hai loại npn và pnp, có cấu trúc vật liệu bán dẫn hoặc silicon hoặc germanium. Hoạt động của BJT phụ thuộc vào dòng của hai loại hạt dẫn: điện tử và lỗ trống.



Hình 4.8. Đặc tuyến dòng áp của BJT.

Dường đặc tuyến vào và đặc tuyến ra của transistor npn biểu diễn hình 4.8:

- Khi $V_{BE} < 0,6 V$ transistor làm việc tại miền ngắt và hở mạch collector-emitter.
- Khi $0,6 V \leq V_{BE} \leq 0,7 V$, transistor làm việc ở miền khuếch đại và I_C được xác định theo công thức: $I_C = h_{FE} I_B$, trong đó h_{FE} là hệ số khuếch đại dòng một chiều.
- Và khi $V_{BE} \geq 0,7 V$, transistor làm việc ở miền bão hòa, và $I_B \geq I_{CS}/h_{FE}$

Bảng 4.2. Các tham số và ứng dụng của BJT.

V_{BE} (V)	V_{CE} (V)	Quan hệ dòng điện	Miền hoạt động	Trạng thái
< 0,6 V	Hở mạch	$I_B = I_C$	Miền ngắt	Ngắt (Off)
0,6 – 0,7 V	> 0,2 V	$I_C = h_{FE} \times I_B$	Miền khuếch đại	
0,7 – 0,8 V	0,2 V	$I_B \geq I_{CS}/h_{FE}$	Miền bão hòa	Đóng (On)

Ví dụ xét mạch như hình 4.9 với các giá trị: $R_C = 1k\Omega$; $R_B = 22k\Omega$; $V_{BE} = 0,7$ V; $h_{FE} = 50$; $V_{CC} = 5V$; Giả sử IC số có mức điện áp cao H = 5 V và mức điện áp thấp L = 0,2V.

+ Nếu điện áp vào $V_i = 0,2$ V (mức thấp), transistor ở chế độ ngắt, Collector-emitter hở mạch, khi đó điện áp tại đầu ra chính $V_o=H=5V$.

+ Khi điện áp vào $V_i = 5$ V (mức cao), $V_{BE} > 0,6$ V, dòng điện cực gốc I_B được xác định theo công thức:

$$I_B = \frac{V_i - V_{BE}}{R_B} = \frac{5V - 0,7V}{220k\Omega} = 0,195 \text{ mA}$$

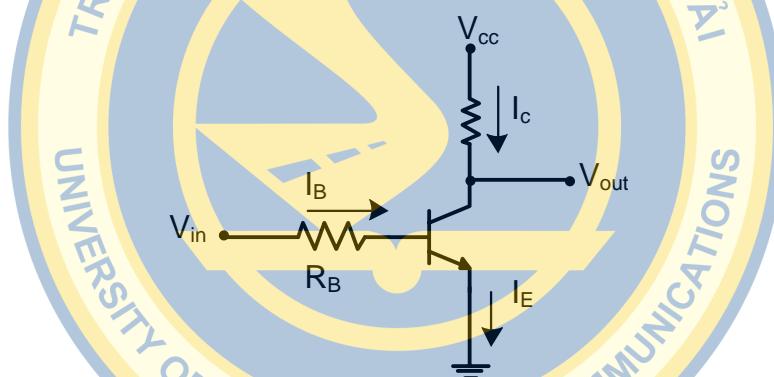
Xét $V_{CE} = 0,2$ V, dòng collector đạt giá trị lớn nhất:

$$I_{CS} = \frac{V_{CC} - V_{CE}}{R_C} = \frac{5V - 0,2V}{1k\Omega} = 4,8 \text{ mA}$$

Khi đó:

$$I_{CS}/h_{FE} = 4,8 \text{ mA} / 50 = 0,096 \text{ mA}$$

Như vậy $I_B = 0,195 \text{ mA} > I_{CS}/h_{FE} = 0,096 \text{ mA}$, do đó transistor ở chế độ bão hòa, tức là $V_o = V_{CE}$ (bão hòa) = 0,2 V = L.



Hình 4.9. Ví dụ về mạch transistor.

Kết quả phân tích cho thấy mạch măc hình 4.9 làm việc như một bộ đảo.

Transistor trong mạch đảo trên biểu diễn đầu ra chỉ có hai điều kiện hoạt động: Đầu ra là H khi đầu vào là L khi transistor ở chế độ ngắt. Nói cách khác, đầu ra là L khi đầu vào là H khi transistor ở chế độ bão hòa. Tổng hợp hai miền làm việc transistor làm việc ở chế độ chuyển mạch. Cũng giống như linh kiện khác, ta sử dụng chế độ này để chế tạo IC logic số.

4.3.3. Resistor-transistor logic (RTL)

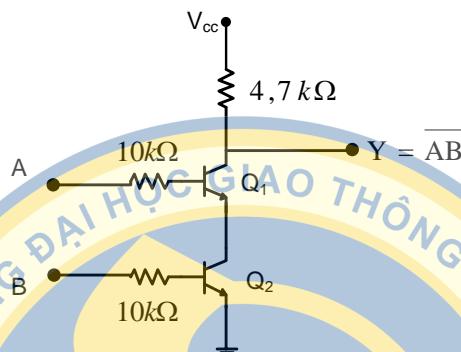
Trong khuôn khổ giáo trình Kỹ thuật Điện tử số, chúng tôi chỉ tập trung giới thiệu mạch nguyên lý của công đa năng NAND, các công khác như NOT, AND, OR, hay NOR,.. đọc giả có thể tìm hiểu thêm ở các tài liệu tham khảo khác.

Mạch cơ bản của công NAND trong họ logic số RTL như hình 4.10, mỗi đầu vào kết hợp một điện trở và một transistor. Cực collector của transistor Q₁ nối với điện trở đến nguồn cấp V_{cc}. Đầu ra được lấy ra trên emitter của Q₁ hay collector Q₂.

Nguyên lý hoạt động:

- Nếu các đầu vào ở mức cao (1 V đến 3,6 V), ứng với transistor ở chế độ bão hòa, do đó đầu ra nối với điểm đất GND, tức là ở mức thấp ($\leq 0,2$ V).
- Nếu các đầu vào ở mức thấp 0,2 V, tắt cả các transistor ở chế độ ngắt, do đó đầu ra nối với nguồn cấp, đầu ra của mạch ở mức cao.

Đây là điều kiện của công logic NAND.



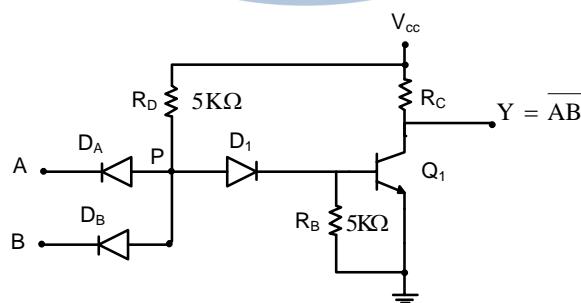
Hình 4.10. Mạch nguyên lý của công NAND trong họ RTL.

Đặc tính của họ RTL như sau:

- + Tốc độ làm việc chậm: độ trễ lên đến 500 ns, không thể hoạt động ở tần số quá 4 MHz.
- + Hệ số tải đầu ra 4-5 với trễ chuyển mạch 50 ns.
- + Khả năng miễn nhiễu kém.
- + Công suất tiêu tán trung bình cao.
- + Dễ ảnh hưởng bởi yếu tố nhiệt độ.

4.3.4. Diode transistor logic (DTL)

Hình 4.11 là sơ đồ nguyên lý công NAND trong họ logic diode-transistor (DTL). Mỗi đầu vào kết hợp với một diode D_A, D_B.



Hình 4.11. Mạch nguyên lý của công NAND trong họ DTL.

Nguyên lý hoạt động:

– Nếu đầu vào bất kỳ của cổng là thấp: 0,2 V, diode tương ứng phân cực thuận và dẫn dòng qua V_{cc} và R_D . Điện áp tại điểm P bằng điện áp vào 0,2 V cộng với điện áp rơi trên diode nên tổng là 0,9 V, chưa đủ để Q_1 dẫn (Để transistor dẫn, điện áp tại P lớn hơn 0,6 V (V_{BE}) qua Q_1 và điện áp rơi qua hai diode D_1, D_2 , tổng 1,8 V). Do đó, Q_1 duy trì ở trạng thái ngắt. Mạch collector-emitter hở mạch. Điện áp đầu ra tại collector transistor Y nối với nguồn qua R_C , tức là gần bằng V_{CC} (5 V), khi đó đầu ra Y ở mức cao 5 V.

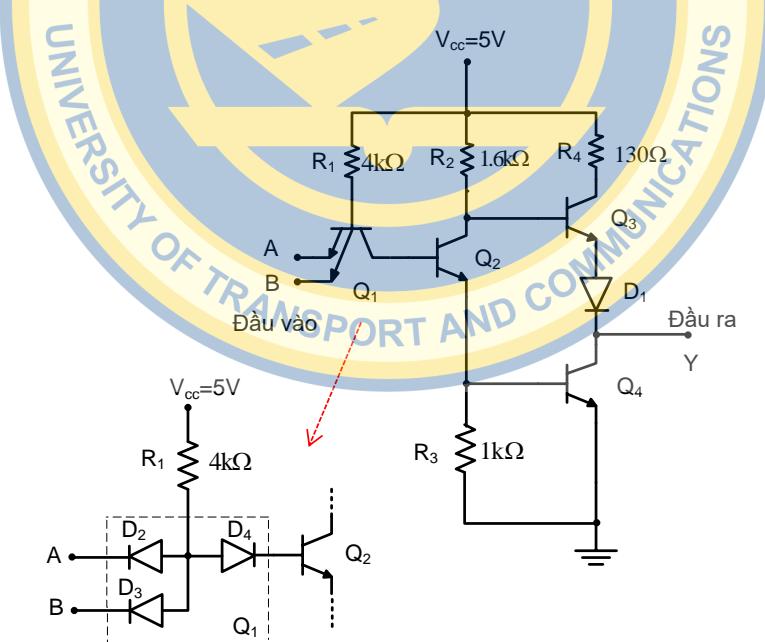
– Nếu các đầu vào cổng ở mức cao 5 V, các diode phân cực ngược, dòng điện sẽ qua R_D , D_1 , D_2 và cực gốc của transistor. Điện áp tại P lúc này là 2,1 V, Lúc này transistor làm việc ở miền bão hòa. Đầu ra $Y = V_{CE} = 0,2$ V, ứng với logic mức thấp.

Họ DTL có các đặc tính sau:

- + Trễ truyền đạt mạch DTL khoảng 30 ns.
- + Hệ số tái đầu ra cao do trở kháng đầu vào khoảng 8.
- + Hệ số tái đầu vào cao do trở kháng đầu vào khoảng 8.
- + Lê nhiễu cao.

4.3.5. Transistor-transistor logic (TTL)

Texas Instrument giới thiệu IC TTL vào năm 1964, chuẩn đầu tiên là series 54/74. Họ TTL là một trong những họ logic phổ biến hiện nay với tốc độ chuyển mạch cao, linh kiện TTL được dùng nhiều trong các mạch số phức tạp, làm giao diện cho các linh kiện đòi hỏi kích thích bằng dòng cao.



Hình 4.12. Mạch nguyên lý của cổng NAND trong họ TTL (hình trên) và mạch diode tương đương Q1 (hình dưới).

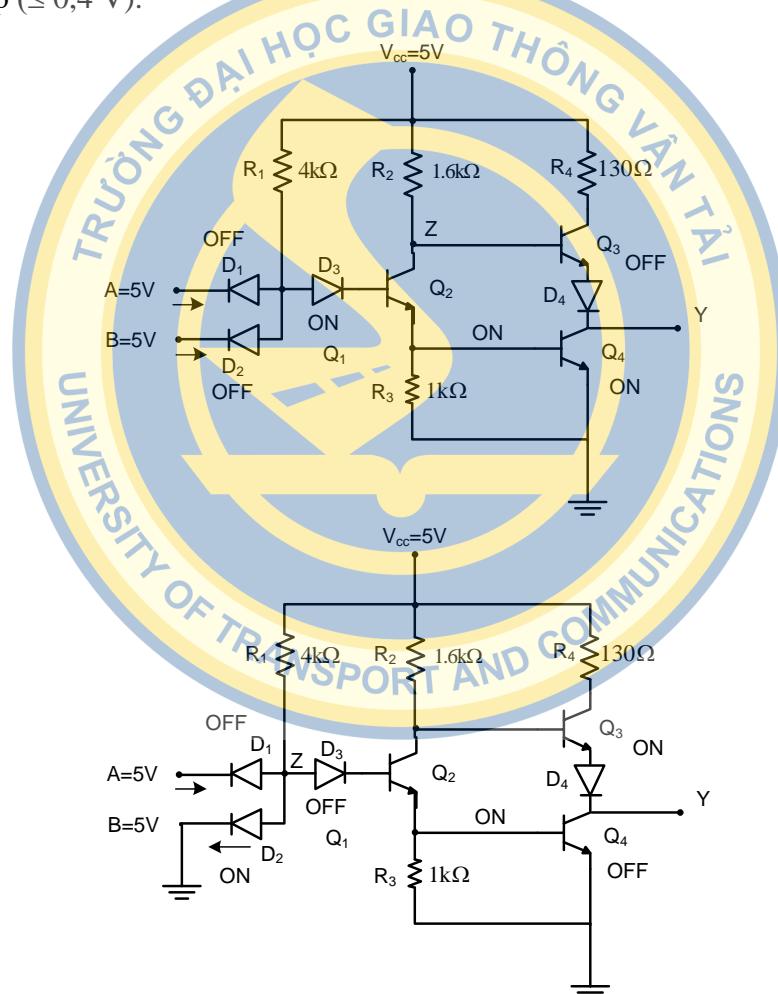
Công NAND trong TTL có sơ đồ điện hình như hình 4.12. Transistor Q₁ chế tạo có hai cực emitter, do đó có hai tiếp giáp E-B để mở Q₁. Transistor được sắp xếp theo kiểu totem-pole, Q₃ hoặc Q₄ sẽ dẫn điện phụ thuộc vào trạng thái logic của đầu vào. Diode tương đương thay cho transistor nhiều cực emitter Q₁ ở hình dưới của 4.12: các diode D₂ và D₃ biểu thị hai tiếp giáp E-B của Q₁ và D₄ là tiếp giáp cực C-B.

Nguyên lý hoạt động:

- Đầu ra ở mức thấp:

Khi cả hai đầu vào A, B đều ở mức cao (≥ 2 V). Mức điện áp này sẽ làm D₁, D₂ khóa, dòng qua R₁ và D₃ vào cực base của Q₂, làm Q₂ đóng. Dòng từ cực emitter Q₂ đến cực base của Q₄, làm Q₄ đóng. Mặt khác, thế tại điểm A lầy từ dòng collector Q₂ qua điện trở R₂ không đủ phân cực cho tiếp giáp E-B của Q₃ và diode D₄ làm Q₃ ngắt (nối D₄, ở đây để đảm bảo Q₃ ngắt hoàn toàn).

Vậy với Q₃ ngắt và Q₄ đóng, đầu ra cuối Y sẽ nối xuống điểm đất và đạt mức điện áp thấp ($\leq 0,4$ V).



Hình 4.13 Công NAND trong TTL với hai trạng thái đầu ra cao (hình trên) và thấp (hình dưới).

- Đầu ra ở cao:

Khi ít nhất một trong hai đầu vào ở mức thấp, đầu ra sẽ là mức cao. Ví dụ hình 4.13 mô tả khi đầu vào B ở mức thấp, điều này làm D₂ được phân cực thuận, do đó, dòng điện chạy từ nguồn V_{CC} qua R₁ và D₂, xuống đất. Điện áp thuận qua D₂ giữ điểm Z ở mức xấp xỉ 0,7 V, không đủ phân cực cho D₃ và tiếp giáp E-B của Q₂ (= 0,7V + 0,7 V), dẫn tới Q₂ ngắt, dẫn tới cực base của Q₄ không được phân cực làm Q₄, tức là Q₄ cũng ngắt. Mặt khác, do cũng không có dòng qua collector Q₂, thế lấy từ nguồn qua R₂ đủ để phân áp cho tiếp giáp B-E của Q₃, làm Q₃ đóng.

Vậy với Q₃ tắt và Q₄ mở, đầu ra cuối Y sẽ nối lên nguồn và đạt mức điện áp cao (≥ 2 V).

Các tham số cơ bản của họ TTL

Họ TTL gồm nhiều chuỗi sản phẩm như 74, 74LS, 74F,... một số tham số điển hình của họ TTL tổng hợp tại bảng 4.3:

Bảng 4.3. Tổng hợp các tham số cơ bản của họ logic TTL.

Tham số điển hình	74	74S	74LS	74AS	74ALS	74F
Tiêu chuẩn	Schottky tốc độ cao	Schottky công suất thấp	Schottky tiên tiến	Schottky công suất thấp tiên tiến	Nhanh	
Trễ truyền dẫn (ns)	9,0	3,0	9,5	1,7	4,0	3,0
Công suất tiêu tốn (mW)	10	20	2,0	8,0	1,2	6,0
Tốc độ (MHz)	35	125	45	200	70	100
V _{OH} (V)	2,4	2,7	2,7	2,5	2,5	2,5
V _{OL} (V)	0,4	0,5	0,5	0,5	0,5	0,5
V _{IH} (V)	2,0	2,0	2,0	2,0	2,0	2,0
V _{IL} (V)	0,8	0,8	0,8	0,8	0,8	0,8
I _{OH} (mA)	0,4	1,0	0,4	2,0	0,4	1,0
I _{OL} (mA)	16	20	8,0	20	8,0	20
I _{IH} (μ A)	40	50	20	20	20	20
I _{IL} (mA)	1,6	2,0	0,4	0,5	0,1	0,6

Lưu ý: với IC số cụ thể, cần tham khảo tài liệu kỹ thuật đi kèm để có tham số chính xác.

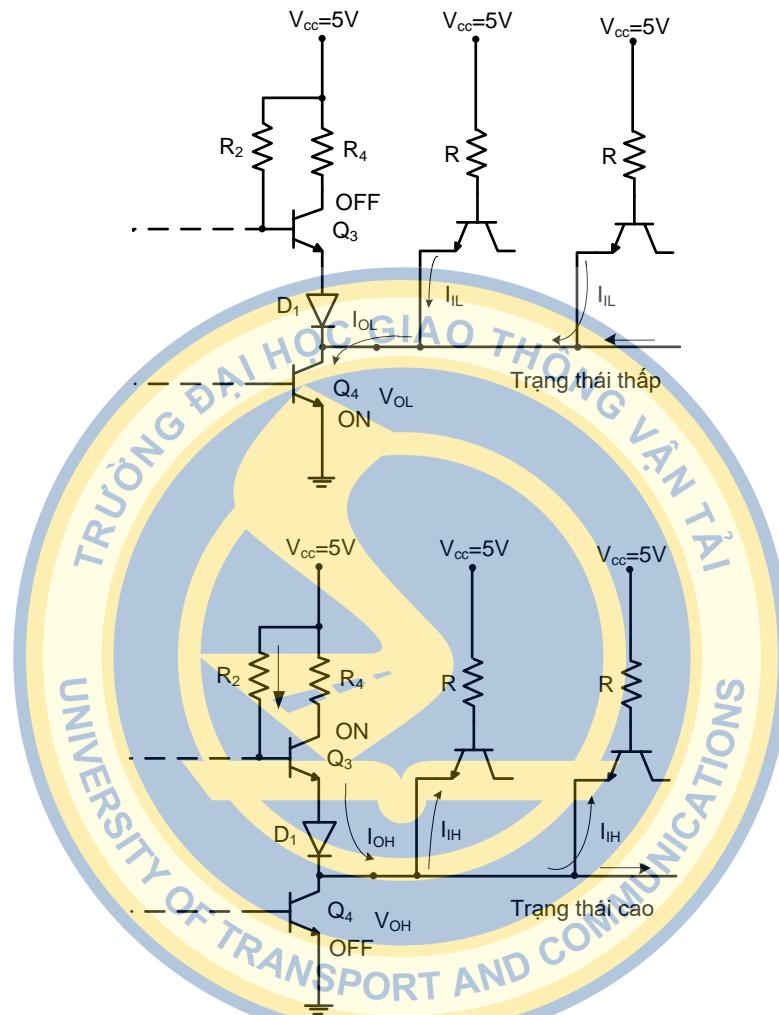
Một số đặc điểm khác cần chú ý khi sử dụng họ TTL

+ Giới hạn hệ số tải: Hình 4.14 mô tả khi IC TTL nối với tải ở đầu ra khi ở trạng thái thấp và cao.

Khi ở trạng thái thấp, transistor Q₄ đang mở và là bộ phận thu dòng phản ánh ngược trở lại từ tải I_{OL}. Giá trị I_{OL} bằng tổng các dòng I_{IL} từ mỗi đầu vào của tải. Tuy Q₄ mở, nhưng vẫn tồn tại giá trị điện trở C-E (dù rất nhỏ), vì vậy V_{OL} vẫn khác 0. Nếu nhiều tải được nối, tức dòng I_{OL} tăng, làm cho V_{OL} tăng theo, và có thể vượt quá giới hạn V_{OL}, khi đó sẽ không phân biệt được trạng thái logic 0.

Tương tự, nếu đầu ra ở trạng thái cao, Q_3 đóng vai cho cung cấp dòng I_{OH} (bằng với tổng dòng I_{IH} của các đầu vào TTL của tầng tải). Khi có nhiều tải cùng nối, làm I_{OH} lớn, dẫn tới điện áp tổng đi qua R_2 , tiếp giáp B-E của Q_3 và D_1 , là khá lớn, điều này lại làm V_{OH} giảm, và có thể dưới ngưỡng. Nghĩa là trạng thái logic bị mất.

Như vậy dù ở trạng thái cao hay thấp, số đầu ra kết nối cần nằm trong giới hạn cho phép, đây chính là lý do sinh ra hệ số tải ra fan-out.



Hình 4.14. Nối tải trong TTL khi đầu ra thấp (hình trên) và cao (hình dưới).

- Ví dụ 4.1:** (a) Xác định hệ số tải đầu ra 74LS00 cổng NAND.
 (b) Xác định số đầu vào 74LS00 có thể kích ngược lại đầu ra 7400?

Hướng dẫn:

- (a) Từ bảng 4.3, ta có tham số của họ 74LS:

$$I_{OH} = 0,4 \text{ mA}, \quad I_{OL} = 8 \text{ mA}$$

$$I_{IH} = 0,02 \text{ mA}, \quad I_{IL} = 0,4 \text{ mA}$$

$$\text{Hệ số tải đầu ra (trạng thái cao)} = \frac{I_{OH}}{I_{IH}} = \frac{0,4}{0,02} = 20$$

$$\text{Hệ số tải đầu ra (trạng thái thấp)} = \frac{I_{OL}}{I_{IL}} = \frac{8}{0,4} = 20$$

Do đó hệ số tải đầu ra 74LS00 là 20.

(b) Từ bảng 4.3, ta có tham số của họ 74:

$$I_{OH} = 0,4 \text{ mA}, I_{OL} = 16 \text{ mA}$$

Ở trạng thái cao, số cỗng 74LS00 có thể kích thích bởi 7400 là:

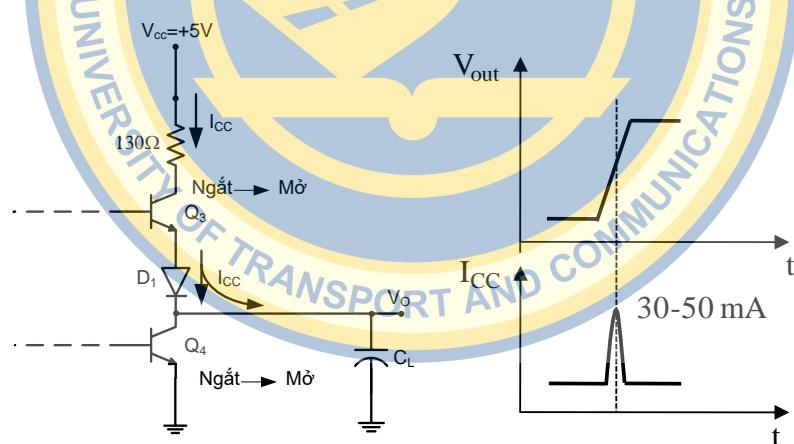
$$= \frac{I_{OH,7400}}{I_{IH,74LS00}} = \frac{0,4}{0,02} = 20$$

Ở trạng thái thấp, số cỗng 74LS00 có thể kích thích bởi 7400 là:

$$= \frac{I_{OL,7400}}{I_{IL,74LS00}} = \frac{16}{0,4} = 40$$

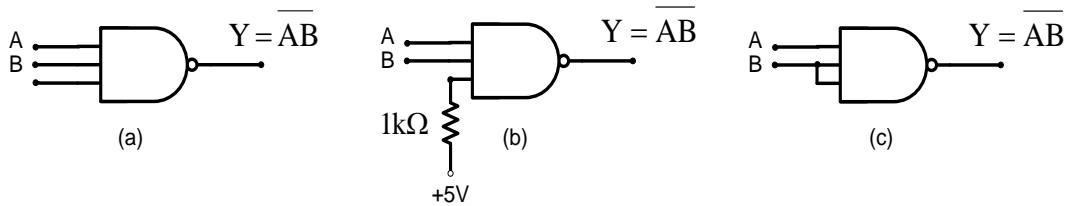
Như vậy, số cỗng 74LS00 có thể được kích thích bởi 7400 là min (20,40) = 20.

+ Xung dòng: mạch logic TTL chịu tác động của các xung đột biến hoặc xung nhiễu sinh ra trong cấu trúc totem-pole của đầu ra. Khi đầu ra chuyển trạng thái từ thấp lên cao (hình 4.15), hai transistor Q₃ và Q₄ thay đổi trạng thái: Q₃ đóng thành mở, Q₄ mở thành đóng. Q₄ thay đổi từ trạng thái bão hòa nên thời gian chuyển trạng thái dài hơn Q₃. Vì vậy, có một khoảng thời gian ngắn cả Q₃, Q₄ đều dẫn điện và sinh ra một dòng đột biến bị hút từ nguồn +5V. Để dòng này không ảnh hưởng đến mạch, cần nối tụ giữa nguồn và điểm đất để hạn chế dòng xung này.



Hình 4.15. Hiện tượng xung dòng khi thay đổi trạng thái trong TTL.

+ Đầu vào không sử dụng: trong thực tế thiết kế mạch số, có thể có đầu vào trong IC TTL không được sử dụng hết, khi đó ta có thể để tự do (no connection), hay nguồn nuôi qua điện trở khoảng 1 kΩ, hay nối đầu vào không sử dụng đó với một đầu vào nào đó đang sử dụng. Ví dụ hình 4.16 mô tả các cách xử lý đầu vào không sử dụng của cỗng NAND 3 đầu vào khi chỉ sử dụng 2.



Hình 4.16. Các cách xử lý đầu vào không sử dụng. a, Đè nguyên không nối. b, Nối đến nguồn cung cấp qua điện trở. c, Nối với một đầu vào sử dụng.

4.4. CÁC HỌ LOGIC SỬ DỤNG TRANSISTOR HIỆU ÚNG TRƯỜNG

4.4.1. Đặc tính transistor hiệu ứng trường

Bảng 4.4 tóm hợp các trường hợp kích thích điện áp điều khiển và kết quả đầu ra của transistor hiệu ứng trường bán dẫn-ô xít-kim loại MOSFET (Metal-Oxide Semiconductor Field-Effect Transistor), thường được viết ngắn hơn FET.

- FET kênh n (nFET): Khi $V_{GS} = 0$, không hình thành được kênh dẫn nguồn giữa cực D và cực S, FET ngắt. Khi V_{GS} bằng nguồn cấp, điện áp đạt ngưỡng (V_T), thời điểm này hình thành kênh dẫn giữa cực D và cực S, hay FET đóng.
- FET kênh p (pFET): ngược lại khi $V_{GS} = 0$, kênh dẫn giữa cực D và cực S bão hòa, hay FET đóng, khi V_{GS} bằng nguồn cấp, kênh dẫn giữa cực D và cực S bị giải phóng, hay FET ngắt.

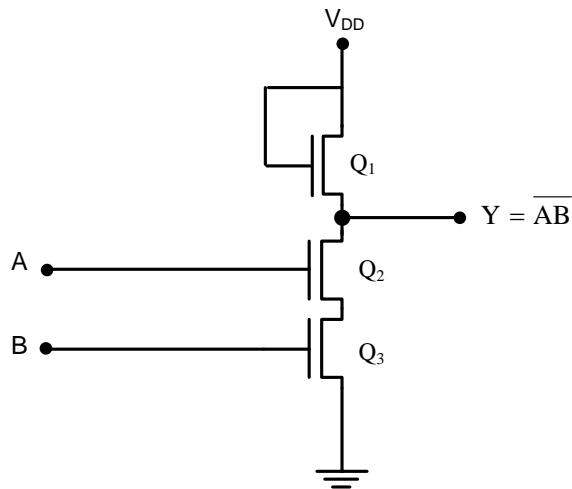
Bảng 4.4. FET trong chế độ xung.

	$G=mức thấp=0$	$G=mức cao=1$	
nFET	D G S	D Ngắt (OFF) S	D Đóng (ON) S
pFET	D G S	D Đóng (ON) S	D Ngắt (OFF) S

Mạch số dùng MOSFET được phân thành 2 nhóm: (1) họ MOS gồm PMOS sử dụng pFET và NMOS sử dụng nFET và (2) CMOS, dùng cả kênh n và kênh p.

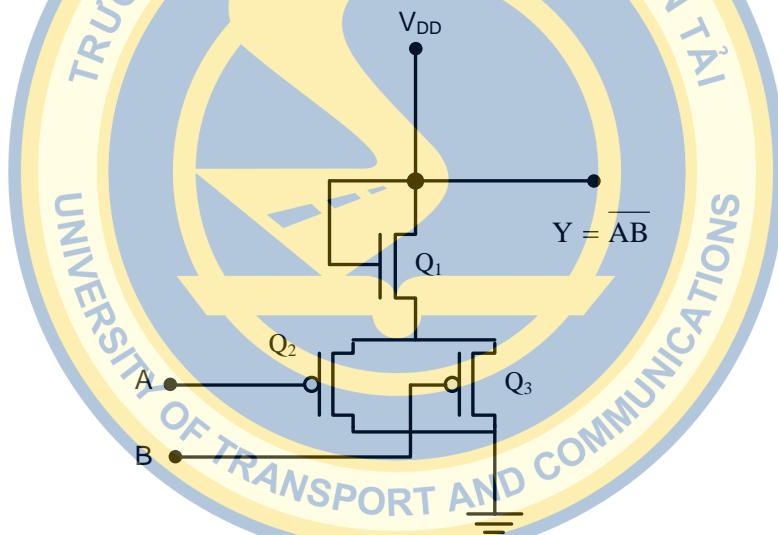
4.4.2. Họ MOS

- **nMOS:** Hình 4.17 vẽ sơ đồ công NAND trong nMOS, trong đó Q₁ hoạt động như điện trở, Q₂ và Q₃ mắc nối tiếp và được điều khiển bởi mức điện áp tại đầu vào A và B.



Hình 4.17. Cấu tạo công NAND trong nMOS.

- + Nếu A hoặc B hoặc cả hai ở mức thấp 0 V (logic 0), nFET ngắn, đầu ra Y lấy áp từ nguồn V_{DD} qua Q₁, nhận khoảng điện áp ở mức logic cao ($\geq 3,5$ V).
- + Khi A và B ở mức cao 5 V (logic 1), cả Q₂ và Q₃ đều đóng, đầu ra Y nối xuống điểm đất, tức là nhận mức thấp (logic 0).



Hình 4.18. Cấu tạo công NAND trong pMOS.

- pMOS: Công NAND trong pMOS có sơ đồ nguyên lý hình 4.18, Q₁ và Q₂ mắc song song và được điều khiển bởi mức điện áp tại đầu vào A và B, Q₁ cũng làm việc như một điện trở.
- + Nếu A hoặc B hoặc cả hai ở mức thấp cao V (logic 0), pFET đóng, đầu ra Y lấy áp từ nguồn V_{DD} qua Q₁, nhận khoảng điện áp ở mức logic cao ($\geq 3,5$ V).
- + Khi A và B ở mức cao 5 V (logic 1), cả Q₂ và Q₃ đều ngắn, đầu ra Y nối xuống điểm đất, tương tự nMOS, Y sẽ nhận mức thấp (logic 0).

- Đặc điểm của logic MOS:

So với các họ logic lưỡng cực, họ logic nMOS và pMOS có tốc độ hoạt động chậm hơn, tiêu hao năng lượng ít hơn, có giới hạn nhiễu hẹp hơn, khoảng điện áp nguồn nuôi lớn hơn, hệ số tải lớn hơn.

+ Trễ đường truyền: thời gian trễ đường truyền CMOS chuẩn khoảng 25 ns-150 ns, thông thường khoảng 50 ns. Tham số này phụ thuộc vào điện áp nguồn cấp và các hệ số khác.

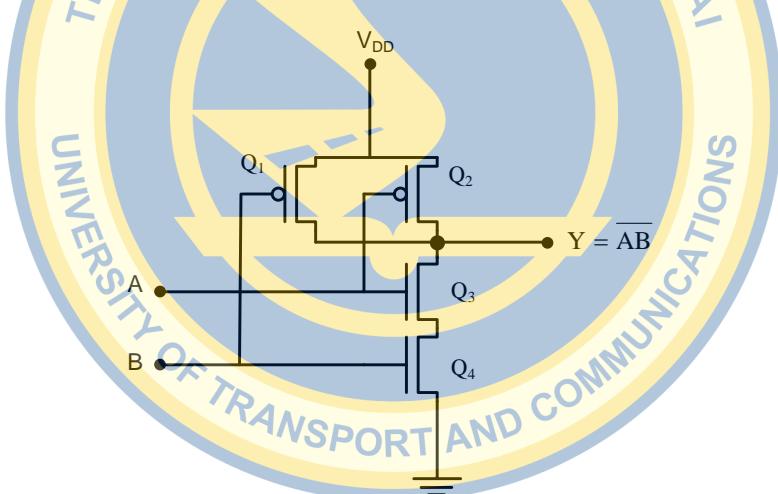
+ Giới hạn nhiễu: họ CMOS có giới hạn nhiễu cao nhất trong họ logic. Tùy thuộc vào điện áp cung cấp và thường bằng khoảng 45 % nguồn cấp V_{DD} . Giới hạn nhiễu là như nhau cho cả mức cao và mức thấp. Với nguồn cấp $V_{DD} = 5$ V, giới hạn nhiễu khoảng 2,25 V

+ Công suất tiêu tán: mạch logic MOS tiêu hao ít công suất.

Quy trình chế tạo đơn giản: logic MOS là họ logic đơn giản nhất do chỉ dùng một thành phần cơ bản của transistor pFET hoặc nFET, không đòi hỏi các thành phần khác như diode, điện trở. Do đó, công suất tiêu tán thấp, thích hợp với LSI.

+ Độ nhạy tĩnh điện: các họ logic MOS dễ bị ảnh hưởng do điện tích tĩnh điện. Toàn bộ điện tích tĩnh điện, gây nên **một điện trường** qua màng điện môi oxide mỏng, làm suy giảm tính chất cách điện hay có thể làm hỏng lớp này.

4.4.3. Họ CMOS



Hình 4.19. Cổng CMOS NAND.

- Nguyên lý hoạt động

CMOS là họ logic chế tạo sử dụng kiến trúc MOS bù (Complementary metal–oxide–semiconductor) nghĩa là cả pFET và nFET. Đây là họ logic được sử dụng phổ biến hiện nay trong mạch số, vi xử lý, máy tính số,..

Hình 4.19 là sơ đồ nguyên lý cổng NAND sử dụng công nghệ CMOS, pFET Q₁ và Q₂ mắc song song (thường gọi là mạch pull-up) và nFET Q₃ và Q₄ mắc nối tiếp (thường gọi là mạch pull-down). Nguyên lý hoạt động của cổng CMOS NAND:

A	B	Q ₁	Q ₂	Q ₃	Q ₄	Y
Thấp (logic 0)	Thấp (logic 0)	Đóng	Đóng	Ngắt	Ngắt	Cao (V _{DD}), hay logic 1
Thấp (logic 0)	Cao (logic 1)	Đóng	Ngắt	Đóng	Ngắt	Cao (V _{DD}), hay logic 1
Cao (logic 1)	Thấp (logic 0)	Ngắt	Đóng	Ngắt	Đóng	Cao (V _{DD}), hay logic 1
Cao (logic 1)	Cao (logic 1)	Ngắt	Ngắt	Đóng	Đóng	Thấp (GND), hay logic 0

– Đặc điểm của họ CMOS

+ Công suất tiêu thụ nhỏ hơn TTL và MOS, điều này không những giảm năng lượng tiêu thụ mà thiết kế và sản xuất IC cũng đơn giản và rẻ hơn. Đặc tính này rất thích hợp với các ứng dụng di động.

+ Khả năng miễn nhiễu: IC CMOS có khoảng lề nhiễu cao hơn so với TTL.

+ Kích thước nhỏ hơn, do vậy độ tích hợp lớn hơn.

+ Tuy nhiên, giống MOS, linh kiện CMOS có hạn chế là nhạy cảm với nhiễu điện từ hơn TTL.

– Các dòng (series) sản phẩm:

Giống như họ TTL, CMOS gồm nhiều dòng, và vì ra đời sau nên đa số họ CMOS cũng có cách đánh số từ 74 giống TTL:

+ Dòng 4000/14000:

Hai dòng này tương đương nhau, các IC trong series 4000/14000 có công suất tiêu hao thấp, có thể vận hành với nhiều giá trị điện áp nguồn 3-5 V, tuy nhiên tốc độ chậm hơn so với các series TTL và CMOS khác, hiện nay series 4000/14000 ít được dùng trong các thiết kế hiện đại, tích hợp nhiều tính năng.

+ Dòng 74C:

Dòng này tương thích chân và tương đương chức năng với thiết bị TTL cùng số hiệu. Đặc điểm hiệu suất của dòng 74C gần giống với dòng 4000.

+ Dòng 74HC/HCT (High speed: tốc độ cao)

Đây là phiên bản cải tiến của dòng 74C, có tốc độ chuyển mạch cao gấp nhiều lần 74LS của TTL, và dòng ra cao hơn nhiều so với 74C. Đa số các IC thuộc dòng 74HC/HCT đều tương thích chân và chức năng với IC TTL cùng số hiệu.

+ Dòng 74AC/ACT (Advanced: tiên tiến)

74AC/ACT tương đương tính năng với nhiều IC thuộc dòng TTL nhưng không tương thích chân với TTL. ACT có ưu điểm hơn so với dòng HC ở khả năng kháng nhiễu, trễ đường truyền, tốc độ xung nhịp tối đa.

+ Dòng 74AHC (Advanced high speed: tốc độ cao tiên tiến)

Là một dòng IC mới để thay thế dòng 74HC, dành cho các ứng dụng tốc độ cao, công suất tiêu hao thấp hơn, dòng kích thích thấp hơn.

+ Logic BiCMOS (Bipolar junction transistor CMOS).

Đây là một dòng lai giữa TTL và CMOS nhằm tận dụng lợi thế tốc độ nhanh của TTL và suất tiêu hao thấp của CMOS. Dòng BiCMOS tiêu hao công suất cực thấp và hoạt động ở tốc độ cực cao. Tuy nhiên công nghệ BiCMOS hiện vẫn đang được nghiên cứu và không được áp dụng cho các mạch số phức tạp như vi xử lý.

- Công nghệ điện áp thấp:

Các yêu cầu của sản phẩm điện tử ngày nay như: điện áp tiêu thụ thấp, nhỏ gọn, nhiều tính năng, tốc độ xử lý nhanh,... Để thỏa mãn yêu cầu này, các IC cần tăng về mật độ trên chip, qua đó tăng độ tích hợp nhiều mạch hơn trên chip, giảm thời gian truyền tín hiệu trong mạch. Như chỉ ra ở hình 4.5 trên, thay vì chuẩn 5 V, IC logic hoạt động từ điện áp nguồn chuẩn có xu hướng giảm dần 3,3; 2,5; 1,8 V,... Ví dụ, một số dòng sản phẩm hãng Texas Instruments: dòng 74LVC (Low-voltage CMOS), 74ALVC (Advanced low-voltage CMOS), 74 LVT (Low-voltage BiCMOS Technology).

4.5. GIAO DIỆN HỘ TTL VÀ CMOS

Như đã đề cập, mạch điện tử số hiện nay sử dụng chủ yếu hai công nghệ là TTL và CMOS, cho nên kết nối hai dòng này với các tham số dòng áp khác nhau là một vấn đề cần quan tâm khi thiết kế mạch. Mạch giao diện (interface circuit) là mạch nối giữa mạch kích thích (TTL hay CMOS) và tải (CMOS hay TTL) sao cho đảm bảo duy trì trạng thái logic ứng với đặc thù riêng của từng họ.

Bảng 4.5. Giá trị dòng của IC số TTL và CMOS dùng nguồn 5V.

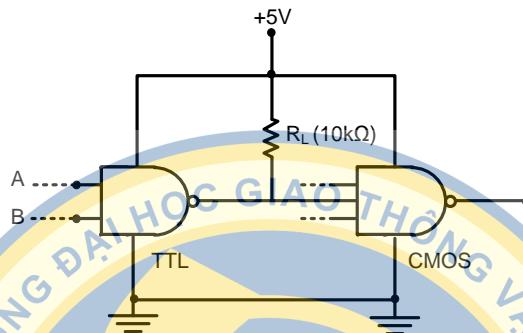
	I_{OH} (mA)	I_{OL} (mA)	I_{IH} (μA)	I_{IL} (mA)
74	0,4	16	40	1,6
74S	1,0	20	50	2,0
74LS	0,4	8,0	20	0,4
74AS	0,2	20	20	0,5
74ALS	0,4	8,0	20	0,1
74F	1,0	20	20	0,6
CMOS (4000B)	0,4	0,4	1,0	0,001
CMOS (74HC/HCT)	4,0	4,0	1,0	0,001
CMOS (74AC/ACT)	24	24	1,0	0,001
CMOS (74AHC/AHCT)	8,0	8,0	1,0	0,001

4.5.1. TTL kích thích CMOS

Trường hợp này TTL là mạch đang xét và CMOS là tải. Bảng trên cho thấy giá trị dòng vào CMOS rất nhỏ so với giá trị dòng ra của TTL bất kỳ. Do đó, TTL có thể đáp ứng các yêu cầu dòng vào của CMOS.

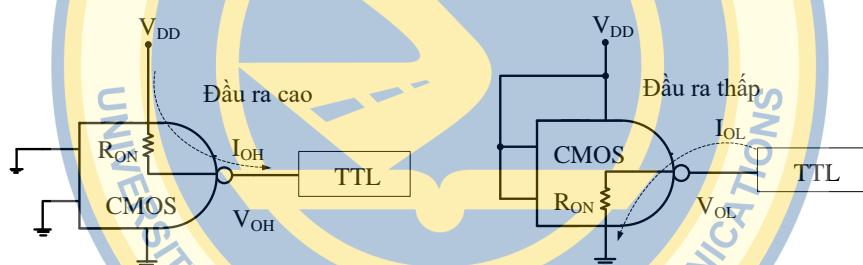
Ở khía cạnh điện áp, $V_{OH(min)}$ của mỗi dòng TTL rất nhỏ so với $V_{IH(min)}$ của các dòng 4000B, 74HC, 74AC, do đó, cần phải nâng điện áp đầu ra của TTL.

Giải pháp chung cho vấn đề này thể hiện ở hình 4.20: Đầu ra TTL nối đến 5 V bằng điện trở kéo lên R_L . Do có điện trở kéo lên R_L sẽ làm điện áp đầu ra TTL tăng xấp xỉ 5 V ở trạng thái cao, tương đương với mức logic chuẩn của CMOS.



Hình 4.20. Mạch TTL kích CMOS làm việc ở điện áp tiêu chuẩn.

4.5.2. CMOS kích thích TTL



Hình 4.21. Mạch CMOS kích TTL.

Trường hợp này CMOS là mạch đang xét và TTL là tải (hình 4.21).

CMOS kích thích TTL ở trạng thái cao: Đầu ra giống như nguồn áp với trờ kháng nguồn R_{ON} là điện trở tương đương khi mở của pFET (xem cấu tạo mạch CMOS hình 4.19). Từ phân tích này và các bảng tham số bảng trên, có thể thấy đầu ra CMOS có thể cung cấp đủ điện áp (V_{OH}) và dòng (I_{OH}) để đáp ứng yêu cầu đầu vào TTL trạng thái cao.

CMOS kích thích TTL ở trạng thái thấp: đầu ra làm việc như bộ phận nhận dòng qua của điện trở R_{ON} (là khi mở của nFET) xuống đất. Từ bảng tham số cho thấy đầu vào TTL có dòng I_{OL} khoảng từ $100 \mu\text{A}$ - 2 mA . Họ 4000B có I_{OL} thấp, không đủ dòng để kích đầu vào nào của họ 74 hay 74AS. Nhưng, các họ 74HC và 74HCT có thể thu nhận tối đa 4 mA , do vậy đủ để kích thích một tải TTL thuộc họ bất kỳ. Vì vậy cần chọn loại IC hợp lý cho trường hợp này.

Ôn tập chương 4

Lý thuyết

1. Phân loại họ vi mạch số phổ biến?
2. Nêu các đặc tính của vi mạch số?
3. Trình bày đặc điểm cơ bản họ logic CMOS?
4. Phương pháp kết nối giữa họ TTL và CMOS?

Bài tập

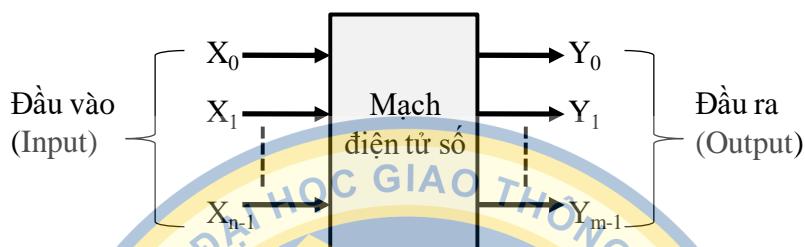
1. Tính hệ số tải ra fan-out cho các dòng TTL:
 - a. 74
 - b. 74LS
 - c. 74AS
 - d. 74ALS
 - e. 74F
2. Tính hệ số tải ra fan-out cho các dòng CMOS:
 - a. 4000B
 - b. 74HC/HCT
 - c. 74AC/ACT
 - d. 74AHC/AHCT
3. Thiết kế và phân tích mạch NOT sử dụng các họ logic:
 - a. RTL
 - b. DTL
 - c. TTL
 - d. MOS
 - e. CMOS
4. Thiết kế và phân tích mạch AND sử dụng các họ logic:
 - a. RTL
 - b. DTL
 - c. TTL
 - d. MOS
 - e. CMOS
5. Thiết kế và phân tích mạch OR sử dụng các họ logic:
 - a. RTL
 - b. DTL
 - c. TTL
 - d. MOS
 - e. CMOS
6. Thiết kế mạch NOR sử dụng các họ logic:
 - a. RTL
 - b. DTL
 - c. TTL
 - d. MOS
 - e. CMOS

Chương 5

PHƯƠNG PHÁP THIẾT KẾ MẠCH ĐIỆN TỬ SỐ

5.1. TỔNG QUAN

Mạch điện tử số có thể mô hình hóa như hình 5.1, bao gồm n đầu vào là các biến logic X_0, X_1, \dots, X_{n-1} , và m đầu ra là các hàm logic Y_0, Y_1, \dots, Y_{m-1} , quan hệ giữa đầu ra và đầu vào chính là mạch điện tử số cần thiết kế. Khi tín hiệu ra chỉ phụ thuộc tín hiệu vào, mạch gọi là mạch điện tử số tổ hợp, còn khi tín hiệu ra phụ thuộc tín hiệu vào và trạng thái bên trong, mạch gọi là mạch điện tử số dãy.



Hình 5.1. Mô hình toán học của mạch điện tử số.

Bài toán thiết kế mạch từ mô hình, đến số hóa, tối thiểu các quan hệ logic, và sơ đồ mạch nguyên lý. Mạch được thực hiện bằng cách kết nối các linh kiện cơ bản NOT, AND, OR, NAND, NOR, XOR, ...được thương mại hóa dưới dạng các IC số (xem mục 4.1). Ở mức độ cao hơn, mạch số sau khi thiết kế ở mức logic, chuyển tiếp sang mức linh kiện transistor, điện trở, diode,.. và sản xuất thành các IC số chức năng riêng tương ứng với hàm đã thiết kế.

Nội dung chương 5 trình bày các bước cơ bản phân tích và thiết kế mạch điện tử số kiểu tổ hợp: sử dụng đại số logic, bảng Karnaugh để tối thiểu hóa hàm logic; nguyên tắc và quy định về mạch điện tử số; phân tích kiểm tra thiết kế; và thiết kế chỉ sử dụng các công logic đa năng NAND hay NOR.

5.2. CÁC BƯỚC THIẾT KẾ MẠCH ĐIỆN TỬ SỐ

Các bước thiết kế mạch điện tử số được mô tả ở hình 5.2

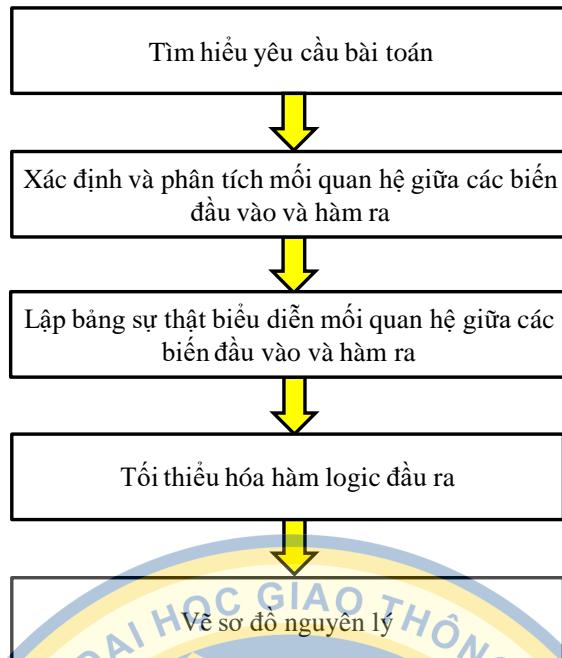
5.2.1. Phân tích bài toán

Đầu vào bao gồm:

- Tín hiệu từ các cảm biến với mức ngưỡng nhất định hay chuyển đổi ADC.
- Tín hiệu từ khói khác đưa tới khi mạch là một module nhỏ trong hệ thống.
- Tín hiệu giả định.

Đầu ra có thể:

- Tín hiệu điều khiển cơ cấu chấp hành như động cơ, loa, đèn,...
- Tín hiệu đến khói khác đưa tới khi mạch là một module nhỏ trong hệ thống.
- Tín hiệu ra giả định.

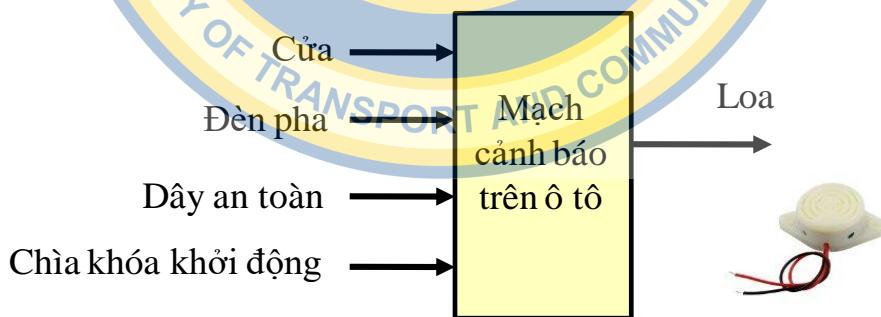


Hình 5.2. Các bước thiết kế mạch điện tử số.

Như phân tích ở chương 3, mạch logic số làm việc ở chế độ nhị phân. Đặc tính này cho phép sử dụng đại số logic như một công cụ toán học để mô tả mối quan hệ giữa đầu vào và đầu ra của mạch giống như một phương trình đại số.

Ví dụ 5.1: Thiết kế mạch điện tử số nhằm cảnh báo cho ô tô với yêu cầu là đưa ra cảnh báo khi:

- Khi cửa xe mở và chìa khóa được chuyển sang vị trí khởi động.
- Khi đèn pha chưa được chuyển sang vị trí khởi động và đèn pha bật.
- Khi dây an toàn không được khóa và chìa khóa được chuyển sang vị trí khởi động.



Hình 5.3. Sơ đồ mạch cảnh báo trên ô tô.

Bài toán được mô hình hóa ở hình 5.3, gồm 04 biến đầu vào:

- + Cửa, ký hiệu là A, khi cửa mở A = 1, khi cửa đóng A = 0.
- + Đèn pha, ký hiệu là B, khi đèn pha bật B = 1, khi đèn pha tắt B = 0.

+ Dây an toàn, ký hiệu là C, khi dây an toàn được thắt B = 1, khi dây an toàn không được thắt B = 0.

+ Chìa khóa khởi động, ký hiệu là D, khi chìa khóa khởi động được bật sang vị trí khởi động D = 1, khi chìa khóa khởi động không được bật sang vị trí khởi động D = 0.

Gọi tín hiệu cảnh báo đầu ra là F, khi có tín hiệu cảnh báo F = 1, khi không có tín hiệu cảnh báo F = 0.

Theo yêu cầu đặt ra, ta có bảng sự thật như sau:

Bảng 5.1. Bảng sự thật mạch cảnh báo trên ô tô.

Đầu vào				Đầu ra
Cửa xe (A)	Đèn pha (B)	Dây an toàn (C)	Khóa khởi động (D)	Cảnh báo (F)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1
1	1	1	1	1

Mỗi quan hệ đầu vào đầu ra của mạch còn có thể viết

- Dưới dạng tổng các tích (SoP):

$$F = \overline{ABC}D + \overline{A}B\overline{CD} + \overline{A}B\overline{CD} + \overline{A}BC\overline{D} + A\overline{BC}D + A\overline{B}CD + AB\overline{CD} + AB\overline{CD} + ABC\overline{D} + ABCD$$

- Hay dạng tích các tổng (PoS):

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

5.2.2. Tối thiểu hóa

Tối thiểu hóa nhằm giảm số biến và số tổ hợp các biến trong hàm logic, tối thiểu hóa hàm logic là bước quan trọng trước khi xây dựng mạch điện tử số, vì giúp giảm số linh kiện thực hiện mạch qua đó làm giảm giá thành, công suất tiêu thụ đồng thời tăng thời gian làm việc và tăng độ tin cậy của mạch.

5.2.2.1. Phương pháp đại số

Gồm 2 bước cơ bản:

- Biểu thức ban đầu được biến thành dạng tổng các tích bằng cách áp dụng định lý Demorgan và nhân các số hạng nhiều lần.

- Khi biểu thức ban đầu đã ở dạng tổng các tích, tìm thừa số chung trong các tích và đặt ra ngoài, nếu có thể cố gắng làm sao loại bỏ bớt một hay nhiều số hạng bằng việc đặt thừa số chung.

Ví dụ 5.2: Tối thiểu hóa hàm logic từ ví dụ 5.1

$$F = \overline{ABC}D + \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + +\overline{A}BC\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}CD + AB\overline{C}\overline{D} + AB\overline{C}\overline{D} + ABC\overline{D} + ABCD$$

Nhóm lại các tổ hợp:

$$\begin{aligned} F &= \overline{ABC}D + \overline{A}\overline{B}\overline{C}\overline{D} + AB\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + AB\overline{C}\overline{D} + \overline{A}\overline{B}CD + AB\overline{C}\overline{D} + A\overline{B}CD + ABCD = \\ &= \overline{AC}D(\overline{B} + B) + ACD(\overline{B} + \overline{B}) + \overline{BC}\overline{D}(\overline{A} + A) + B\overline{C}\overline{D}(\overline{A} + A) + ACD(\overline{B} + B) \\ &= \overline{AC}D + A\overline{C}D + \overline{BC}\overline{D} + B\overline{C}\overline{D} + ACD \\ &= \overline{CD}(\overline{A} + A) + B\overline{D}(\overline{C} + C) + ACD \\ &= \overline{CD} + B\overline{D} + ACD \end{aligned}$$

Ví dụ 5.3: Đơn giản mạch số

$$Z = ABC + AB\overline{(A\overline{C})}$$

Hướng dẫn:

Áp dụng định lý Demorgan và biến đổi:

$$\begin{aligned} Z &= ABC + A\overline{B}(\overline{A} + \overline{C}) = ABC + A\overline{B}(A + C) \\ &= ABC + A\overline{B}A + A\overline{B}C = ABC + A\overline{B}C + A\overline{B} = AC(B + \overline{B}) + A\overline{B} = AC + A\overline{B} \end{aligned}$$

Ví dụ 5.4: Rút gọn biểu thức $Z = (A + B)(A + B + D)\overline{D}$

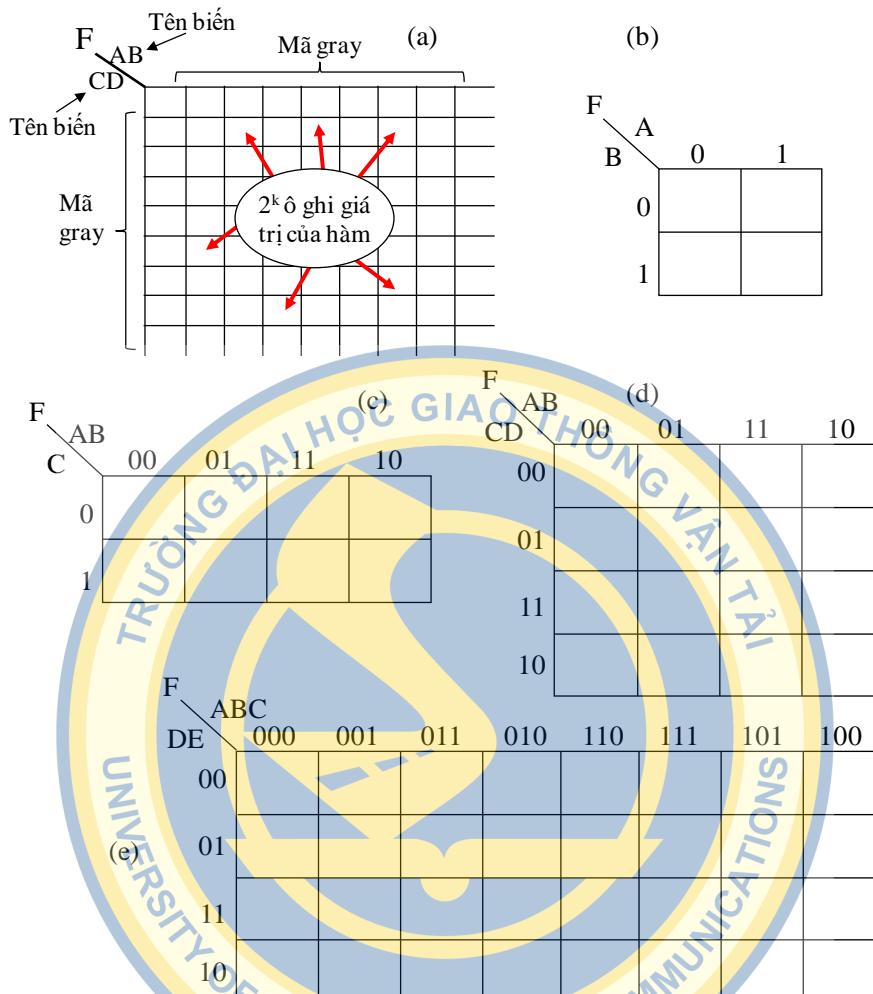
Hướng dẫn:

Có thể biến đổi biểu thức thành dạng tổng các tích, áp dụng định lý đại số logic, kết quả là:

$$\begin{aligned} Z &= (\overline{A} + B)(A + B + D)\overline{D} = (\overline{A} + B)(A\overline{D} + B\overline{D} + D\overline{D}) = (\overline{A} + B)(A\overline{D} + B\overline{D} + 0) \\ &= \overline{A}A\overline{D} + \overline{A}B\overline{D} + BA\overline{D} + BB\overline{D} = 0 + \overline{A}B\overline{D} + BA\overline{D} + BB\overline{D} = \\ &= \overline{A}B\overline{D} + AB\overline{D} + B\overline{D} = B\overline{D}(\overline{A} + A) + B\overline{D} = \\ &= B\overline{D} + B\overline{D} = B\overline{D} \end{aligned}$$

5.2.2.2. Phương pháp bảng Karnaugh

Bảng Karnaugh, hay còn gọi là bảng K, được phát minh bởi nhà khoa học Maurice Karnaugh vào năm 1953. Bảng Karnaugh là công cụ trực quan giúp tối thiểu hóa hàm logic một cách thuận lợi khi số biến không nhiều.



Hình 5.4. Cấu tạo bảng Karnaugh tổng quát (a) và cấu tạo chi tiết ứng với số bit n là (b) 2, và (c) 3 và (d) 4 và (e) 5.

– Lập bảng Karnaugh: Bảng K là sự kết hợp của bảng sự thật và mã gray thành bảng hai chiều như mô tả ở hình 5.4. Có n biến được chia đều hay gần đều ra hàng và cột, sẽ tạo thành 2^n ô, tương ứng với 2^n tổ hợp của biến.

– Tối thiểu hóa hàm logic bằng phương pháp bảng Karnaugh:

	Tổng các tích SoP	Tích các tổng PoS
+Thiết lập bảng	Ghi các giá trị 1 vào ô tương ứng với hạng tích có trong hàm biểu diễn, các ô còn lại lấy giá trị 0.	Ghi các giá trị 0 vào các ô ứng với hạng tổng có trong hàm biểu diễn, các ô còn lại lấy giá trị 1.

+Tối thiểu hóa	<ul style="list-style-type: none"> - Nhóm 2^k, với $k = 0,1,2\dots$, ô giá trị “1” đối xứng nhau qua trực tạo thành bảng, như vậy số lượng ô “1” trong một nhóm luôn là: 1, 2, 4, 8, 16,... - Để đảm bảo tối thiểu, quá trình nhóm cần thỏa mãn: số ô “1” trong mỗi nhóm là lớn nhất; mỗi ô “1” có thể tham gia nhiều nhóm, nhưng số nhóm phủ hết ô “1” phải là nhỏ nhất. - Do các tổ hợp đối xứng trong mã gray chỉ khác nhau một biến. Nên trong mỗi nhóm, các bit khác nhau này sẽ bị triệt tiêu. Hàm tối thiểu chính là tổng các tích lấy từ mỗi nhóm đã tối thiểu này. 	<ul style="list-style-type: none"> - Nhóm 2^k, với $k = 0,1,2\dots$, ô giá trị “0” đối xứng nhau qua trực tạo thành bảng, như vậy số lượng ô “0” trong một nhóm luôn là: 1, 2, 4, 8, 16,... - Để đảm bảo tối thiểu, quá trình nhóm cần thỏa mãn: số ô “0” trong mỗi nhóm là lớn nhất; mỗi ô “0” có thể tham gia nhiều nhóm, nhưng số nhóm phủ hết ô “0” phải là nhỏ nhất. - Do các tổ hợp đối xứng trong mã gray chỉ khác nhau một biến. Nên trong mỗi nhóm, các bit khác nhau này sẽ bị triệt tiêu. Hàm tối thiểu chính là tích các tổng lấy từ mỗi nhóm đã tối thiểu này.
----------------	---	---

Chú ý:

- Trường hợp “Không xác định” hay “X”: nói cách khác, sẽ có một vài tổ hợp tín hiệu vào ta không xác định được tín hiệu ra ở mức cao hay mức thấp. Khi đó, biểu thức đầu ra đơn giản nhất, “X” có thể coi là 1 (trường hợp tổng các tích) hay 0 (trường hợp tích các tổng) để tối thiểu hóa. Tuy nhiên “X” chỉ có thể ghép với ô (trường hợp tổng các tích) hay 0 (trường hợp tích các tổng) để tạo thành nhóm mà không thể ghép các “X” với nhau.
- Để thuận tiện cho việc nhận diện các nhóm, thường dùng các ký hiệu như hình vuông, hình tròn, cung, với nét liền hay nét đứt để đánh dấu các nhóm:



Ví dụ 5.5: Tối thiểu hóa hàm logic ở ví dụ 5.1

+ Ở dạng **tổng các tích SoP**:

$$F = \overline{\overline{A}\overline{B}\overline{C}\overline{D}} + \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + AB\overline{C}\overline{D} + AB\overline{C}\overline{D} + ABC\overline{D} + ABCD$$

Thiết lập bảng K và tối thiểu hóa, từ phương trình hay từ bảng 5.1 biểu diễn sang bảng K và tối thiểu hóa:

F	AB	CD	00	01	11	10
00	0	1	1	0		
01	1	1	1	1		
11	0	0	1	1		
10	0	1	1	0		

↓

F	AB	CD	00	01	11	10
00	0	1	1	0		
01	1	1	1	1		
11	0	0	1	1		
10	0	1	1	0		

$\overline{ABC}D + \overline{A}\overline{B}\overline{C}D + A\overline{B}\overline{C}D + A\overline{B}\overline{C}\overline{D}$
 $= \overline{CD}(\overline{AB} + \overline{A}B + AB + A\overline{B}) = \overline{CD}$

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
UNIVERSITY OF TRANSPORT AND COMMUNICATIONS

Như vậy: $F = AD + B\overline{D} + \overline{CD}$

+ Với F dưới dạng tích các tổng PoS:

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

F

	AB	CD	00	01	11	10
00	0	1	1	0		
01	1	1	1	1		
11	0	0	1	1	1	1
10	0	1	1	1	0	0

A + $\overline{C} + \overline{D} =$

$= B + D$

Ta được: $F = (A + \overline{C} + \overline{D})(B + D)$

Ví dụ 5.6: Một số dạng nhóm thường gặp với hàm 4 biến kiểu tích các tổng SoP:

	F	AB	CD	00	01	11	10
00				1	1	1	0
01				1	1	1	0
11				1	1	1	0
10				1	1	1	0

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

	F	AB	CD	00	01	11	10
00				1	0	0	1
01				1	0	0	0
11				0	0	0	0
10				1	0	0	1

$$F = \overline{ABC} + \overline{DB}$$

	F	AB	CD	00	01	11	10
00				1	1	0	0
01				0	1	1	0
11				1	0	0	1
10				1	0	0	1

$$F = \overline{ACD} + \overline{BCD} + \overline{BD}$$

	F	AB	CD	00	01	11	10
00				1	0	0	1
01				1	0	0	1
11				1	0	0	1
10				1	0	0	1

$$F = \overline{B}$$

	F	AB	CD	00	01	11	10
00				0	X	0	X
01				0	0	X	X
11				X	1	0	0
10				1	X	0	0

$$F = \overline{AC}$$

	F	AB	CD	00	01	11	10
00				1	1	1	1
01				1	1	1	1
11				1	1	1	1
10				1	1	1	1

$$F = "1"$$

(X là trạng thái không xác định)

Ví dụ 5.7: Tối thiểu hóa hàm sau:

$$f(A, B, C) = \sum 0, 3, 4, 7 \quad N=2, 6$$

Hướng dẫn:

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

Và trạng thái không xác định “X” ứng với hai tổ hợp: $\overline{A}B\overline{C}$, $A\overline{B}C$

Lập bảng Karnaugh:

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

Ta được:

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

Ví dụ 5.8: Tối thiểu hóa hàm sau:

$$f(A, B, C, D) = \sum 3, 4, 5, 6, 7, 12, 13$$

Hướng dẫn:

$$f(A, B, C, D) = \overline{AB}CD + \overline{ABC}\overline{D} + \overline{ABC}D + \overline{AB}\overline{CD} + \overline{AB}CD + ABC\overline{D} + AB\overline{CD} + AB\overline{C}D$$

Lập bảng Karnaugh

		CD		00	01	11	10
		AB	00	0	0	1	0
0	1	00	1	1	1	1	
		01	1	1	1	1	
11	10	11	1	1	0	0	
		10	0	0	0	0	

$$\text{Ta được: } f(A, B, C, D) = \overline{A}B + B\overline{C} + \overline{A}CD$$

Ví dụ 5.9: Tối thiểu hóa hàm sau:

$$f(A, B, C, D) = \sum 4, 5, 6, 7, 12, 13, 14, 15$$

Hướng dẫn:

Lập bảng Karnaugh

		CD	00	01	11	10
	AB	00	0	0	0	0
	01	1	1	1	1	
	11	1	1	1	1	
	10	0	0	0	0	

$$f(A, B, C, D) = B$$

Ví dụ 5.10: Tối thiểu hóa hàm có giá trị như bảng sự thật sau:

Đầu vào			Đầu ra
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	X
1	0	0	X
1	0	1	1
1	1	0	1
1	1	1	1

Hướng dẫn:

Từ bảng sự thật ta có bảng Karnaugh:

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

Từ bảng K ta có biểu thức đầu ra:

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

Ví dụ 5.11: Tối thiểu hóa hàm:

$$f(A, B, C, D) = \prod_{\{0, 2, 8, 10, 11, 14, 15\}}$$

Hướng dẫn:

$$f(A, B, C, D) = (\bar{A} + \bar{B} + \bar{C} + \bar{D})(\bar{A} + \bar{B} + C + \bar{D})(A + \bar{B} + \bar{C} + \bar{D})(A + \bar{B} + C + \bar{D}) \\ (A + \bar{B} + C + D)(A + B + C + \bar{D})(A + B + C + D)$$

Lập bảng Karnaugh, ta có:

	CD 00	01	11	10
AB 00	0	1	1	0
01	1	1	1	1
11	1	1	0	0
0	0	1	0	0

$$f(A, B, C, D) = (\bar{A} + \bar{C})(B + \bar{D})$$

5.2.3. Sơ đồ mạch

Nguyên tắc vẽ mạch điện tử số cũng tuân theo các nguyên tắc chung của vẽ mạch điện tử và vẽ kỹ thuật hiện đại như khung tên, kích thước bản vẽ, sử dụng hình vẽ tổng quan, hình vẽ chi tiết, tạo các lớp linh kiện màu khác nhau,... Bên cạnh đó, mạch điện tử số có một số quy định đặc thù sau:

- Để thuận tiện cho việc vẽ và đọc, mạch nguyên lý không vẽ điểm đất (GND).
- Từ hàm logic đã tối thiểu, vẽ sơ đồ mạch sử dụng các công cơ bản như trình bày ở mục 3.2, chương 3.
- Đường nối mạch:

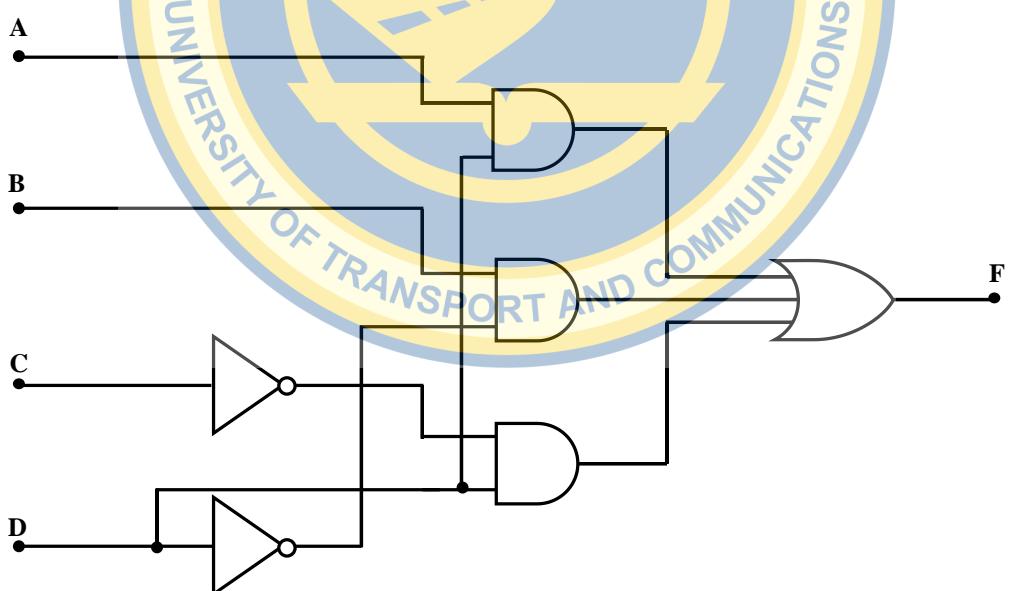
Kiểu cách cũ	Không cắt nhau	Cắt nhau
Kiểu cách mới hiện nay: thuận lợi cho việc thực thi trong các chương trình thiết kế mạch (CAD).		

– Ngoài ra, còn có một số ký hiệu thường gặp khi vẽ hay phân tích mạch:

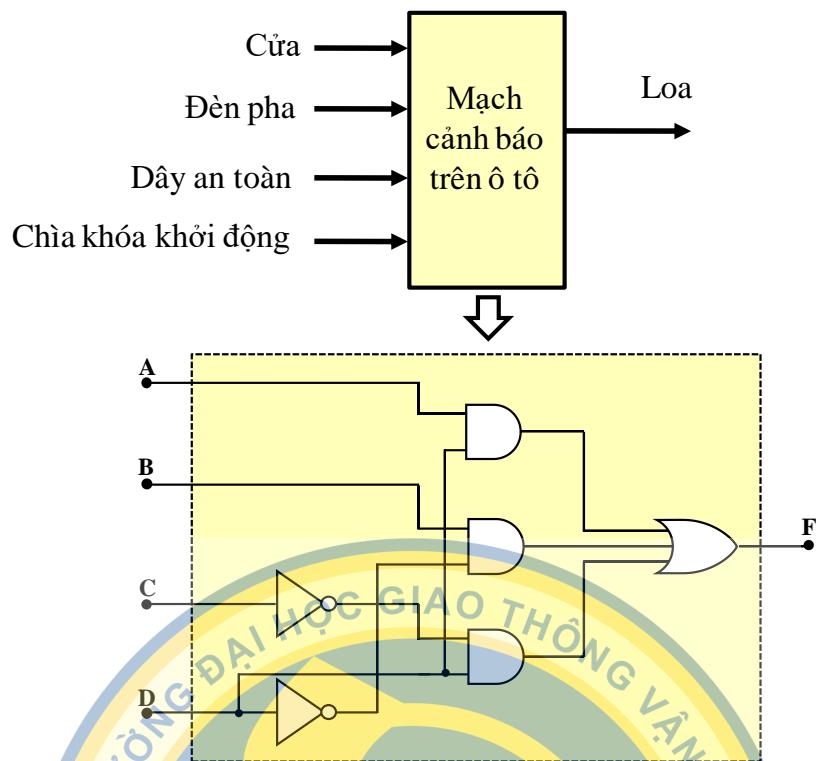
Ký hiệu	Ý nghĩa	Ví dụ
	Tích cực ở mức cao	
	Tích cực ở mức thấp	
	Tích cực ở sườn dương của xung nhịp	
	Tích cực ở sườn âm của xung nhịp	

Ví dụ 5.12. Vẽ sơ đồ mạch với thiết kế ở ví dụ 5.1:

+ Dạng SoP: $F = AD + B\bar{D} + \bar{C}D$



Lưu ý: để hình vẽ rõ ràng, nên thực hiện cách vẽ như sau: bố trí linh kiện trước và cùng một loại linh kiện đặt thẳng hàng, sau đó mới tiến hành đi dây nối các linh kiện.



5.2.4. Kiểm tra thiết kế

Có thể kiểm tra thiết kế bằng việc thay các biến đầu vào từ bảng sự thật, sau đó tính toán tín hiệu đầu ra dựa vào mạch, và so sánh với đầu ra của bảng sự thật.

Ví dụ giá trị logic của đầu ra và dạng sóng tương ứng với đầu vào của mạch cảnh báo cho ô tô hình như hình dưới đây. Ta thấy, tại mọi thời điểm giá trị logic đầu ra thu được trùng khớp với bảng sự thật phân tích ở bảng 5.1, ví dụ 1, chứng tỏ mạch thiết kế đúng theo yêu cầu.

	A	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1	Cửa
Dữ liệu vào	B	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	Đèn pha
	C	0 0 1 1 0 0 1 1	0 0 1 1 0 0 1 1	Dây an toàn
	D	0 1 0 1 0 1 0 1	0 1 0 1 0 1 0 1	Khóa khởi động
Dữ liệu ra	F	0 1 0 0 1 1 1 0	1 0 1 1 1 1 1 1	Loa cảnh báo

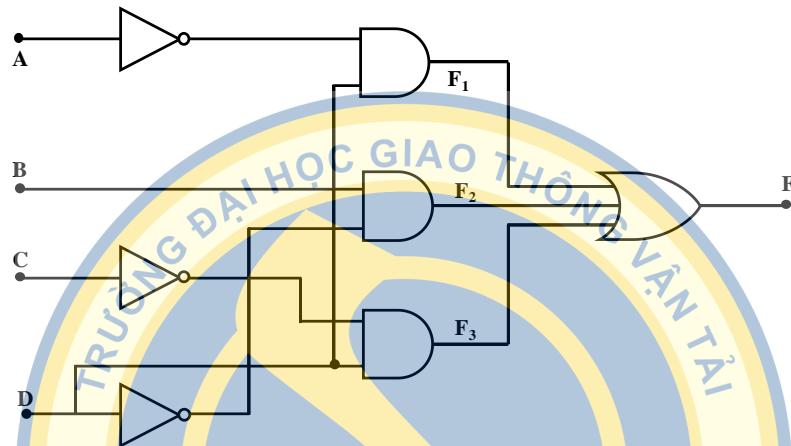
5.3. PHÂN TÍCH MẠCH TỔ HỢP

Phân tích mạch tổ hợp là quá trình ngược lại với thiết kế: từ mạch nguyên lý, tìm ra quan hệ hàm ra và biến logic đầu vào, bảng sự thật, từ đó xác định được mô hình bài toán hay chuyên sang dạng thiết kế sử dụng các công logic khác.

Nội dung phân tích mạch tổ hợp gồm:

- Từ sơ đồ mạch xác định đầu vào/ra.
- Xác định phương trình hàm ra.
- Viết bảng sự thật.

Ví dụ 5.13 Phân tích mạch có sơ đồ như sau:



Hướng dẫn:

Từ mạch đã thấy mạch có 4 đầu vào A, B, C, D và 1 đầu ra F.

Xác định hàm ra: ta có $F = F_1 + F_2 + F_3$

$$F_1 = \overline{AD}; F_2 = \overline{BD}; F_3 = \overline{CD}$$

$$\text{Vậy, } F = \overline{A}D + \overline{B}\overline{D} + \overline{C}\overline{D}$$

Từ hàm logic này, ta có bảng sự thật của mạch như sau:

Đầu vào				Đầu ra
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1

0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

5.4. THIẾT KẾ SỬ DỤNG CÁC CỘNG ĐA DỤNG

Thông thường các hàm logic sẽ cần dùng cổng NOT, AND và OR, tức là cần phải dùng hai loại IC khác nhau khi thi công. Tuy nhiên, các cổng logic đa năng như NAND, NOR hay Imply có thể thực hiện được tất cả các mối quan hệ như đã trình bày trong mục 3.4, chương 3. Nghĩa là có thể thiết kế và thi công mạch điện tử số chỉ sử dụng một trong các cổng đa năng này. Có thể nói đây là chìa khóa rất quan trọng trong công nghệ sản xuất IC độ tích hợp cao VLSI, vì nó làm giảm độ phức tạp khi chế tạo, qua đó giảm chi phí sản phẩm. Chẳng hạn, nguyên lý thiết kế IC CMOS trình bày ở chương 4 cũng được dựa trên kiểu cổng NAND hay NOR.

5.4.1. Sử dụng cổng NAND

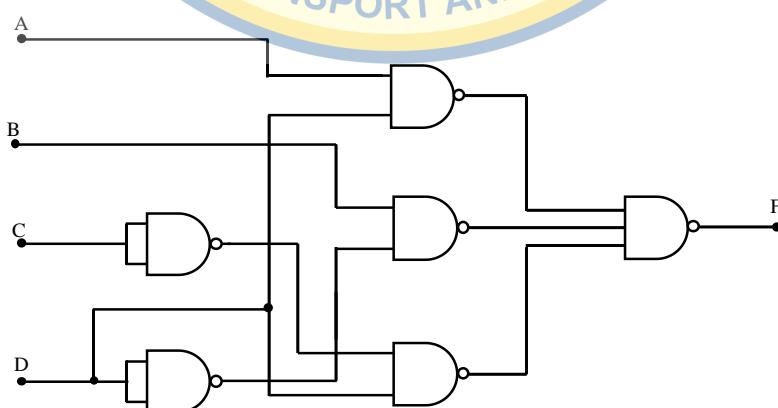
Để thực hiện chức năng logic bởi cổng NAND, sử dụng định lý DeMorgan và các định lý, tiên đề của đại số logic, chuyển hàm về dạng các tích đảo, khi đó quan hệ giữa các biến có dạng NAND và ta có thể dễ dàng thực hiện việc vẽ mạch.

Ví dụ 5.14 Vẽ sơ đồ mạch chỉ dùng NAND với hàm đã tối ưu ở ví dụ 5.1

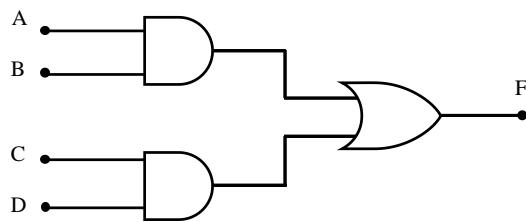
Hướng dẫn:

$$\begin{aligned}
 F &= AD + B\bar{D} + \bar{C}D \\
 &= \overline{\overline{AD}} + \overline{B\bar{D}} + \overline{\overline{\bar{C}D}} = (\overline{AD})(\overline{B\bar{D}})(\overline{\bar{C}D}) = (\overline{AD})(\overline{B}\overline{D})(\overline{\bar{C}D}) \\
 &= (\overline{AD})(\overline{B}\overline{D})(\overline{\bar{C}D})
 \end{aligned}$$

Sơ đồ chỉ dùng NAND:



Ví dụ 5.15: Chuyển đổi thiết kế mạch có sơ đồ sau sang mạch sử dụng toàn NAND



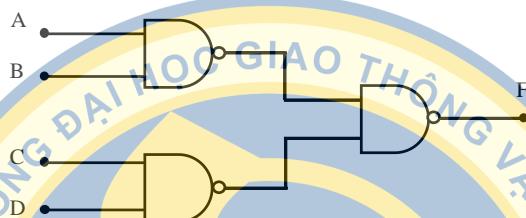
Hướng dẫn:

Từ sơ đồ, ta có $F = AB + CD$

Sử dụng đại số Boolean và định lý Demorgan biến đổi:

$$F = AB + CD = \overline{(\overline{AB} + \overline{CD})} = \overline{(\overline{AB})(\overline{CD})}$$

Từ đó mạch sử dụng toàn NAND có dạng:



5.4.2. Sử dụng cổng NOR

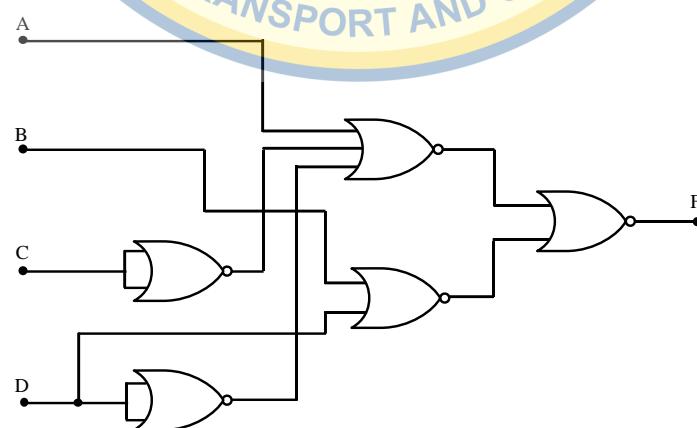
Để thực hiện chức năng logic bởi cổng NOR, sử dụng định lý DeMorgan và các định lý, tiên đề của đại số logic, chuyển hàm về dạng các tổng đảo, khi đó quan hệ giữa các biến có dạng NOR và ta có thể dễ dàng thực hiện việc vẽ mạch.

Ví dụ 5.16: Vẽ sơ đồ mạch chỉ dùng NOR với hàm đã tối ưu ở ví dụ 5.1

Hàm kiểu PoS, sau khi tối thiểu ở ví dụ 5.1 có dạng $F = (A + \bar{C} + \bar{D})(B + D)$

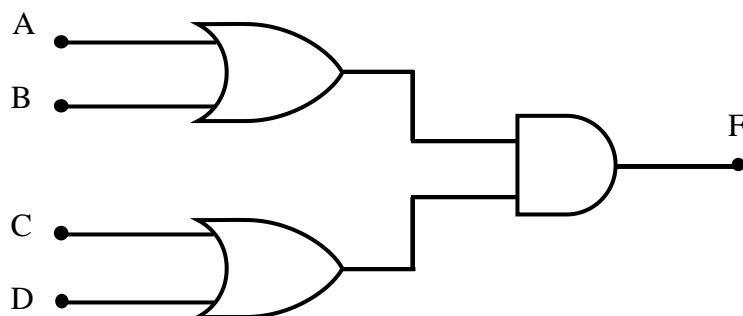
Sử dụng định lý DeMorgan ta được:

$$\begin{aligned} F &= (A + \bar{C} + \bar{D})(B + D) = (A + \bar{C} + \bar{D}) + (B + D) \\ &= ((A + \bar{C}) + (\bar{C} + \bar{D})) + (B + D) \end{aligned}$$



Ví dụ 5.17

Chuyển đổi thiết kế mạch có sơ đồ sau sang mạch sử dụng toàn NOR



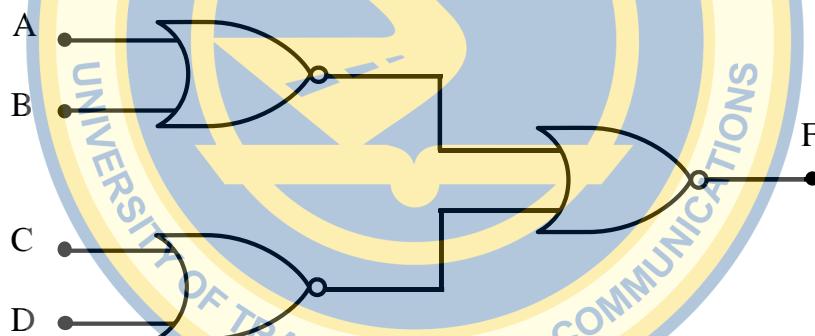
Hướng dẫn:

Từ sơ đồ mạch, ta có $F = (A+B)(C+D)$ (có dạng OR-AND)

Áp dụng đại số Boolean và định lý Demorgan để biến đổi.

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

Mạch sử dụng toàn NOR có dạng:



Một vài ví dụ thêm:

Ví dụ 5.18

Thiết kế mạch logic có ba đầu vào A, B, C. Đầu ra sẽ ở mức cao chỉ khi đa số đầu vào ở mức cao.

Hướng dẫn:

Bước 1: Lập bảng sự thật

Trên cơ sở yêu cầu của bài toán, đầu ra Z bằng 1 khi có hai hay ba đầu vào bằng 1. Trường hợp khác thì đầu ra sẽ bằng 0.

Đầu vào			Đầu ra
A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

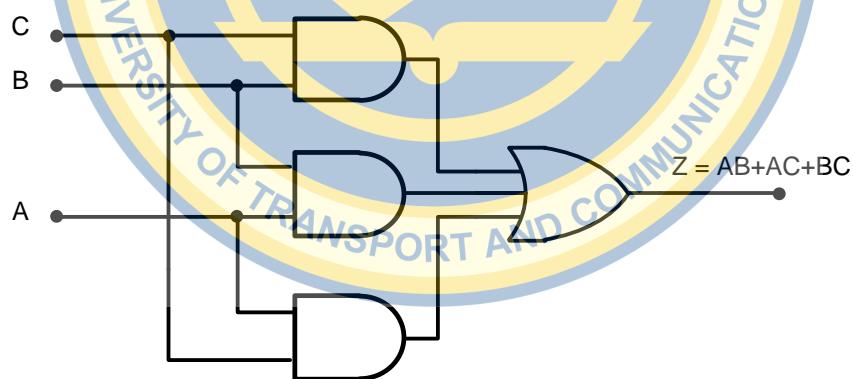
Bước 2: Viết biểu thức đầu ra

$$Z = \overline{ABC} + A\overline{BC} + AB\overline{C} + ABC$$

Bước 3: Tối thiểu hóa biểu thức đầu ra (sử dụng phương pháp đại số hay bảng Karnaugh)

$$Z = AB + AC + BC$$

Bước 5: Xây dựng mạch từ biểu thức đã rút gọn



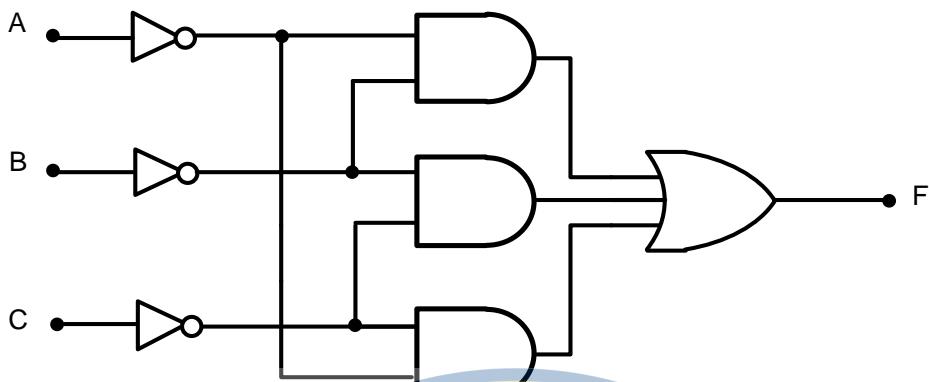
Ví dụ 5.19

Xây dựng mạch thực hiện hàm: $F = \overline{BC} + \overline{AC} + \overline{AB}$ sử dụng:

- Các cổng logic cơ bản.
- Chỉ sử dụng cổng NAND.
- Chỉ sử dụng cổng NOR.

Hướng dẫn:

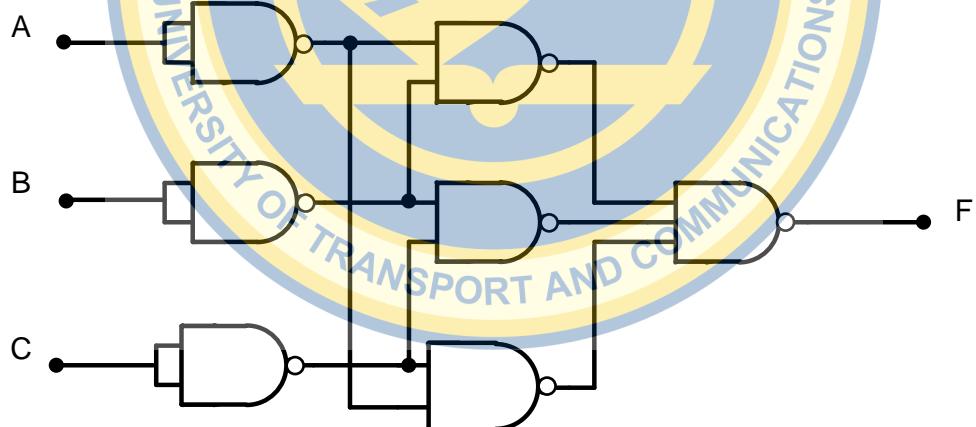
- Thực hiện mạch bởi các cổng AND, OR, NOT



- Thực hiện chuyển từng cổng sang cổng NAND

$$\begin{aligned}
 F &= \overline{BC} + \overline{AC} + \overline{AB} = \overline{\overline{BC}} + \overline{\overline{AC}} + \overline{\overline{AB}} = (\overline{BC})(\overline{AC})(\overline{AB}) \\
 &= (\overline{B}\overline{C})(\overline{A}\overline{C})(\overline{A}\overline{B})
 \end{aligned}$$

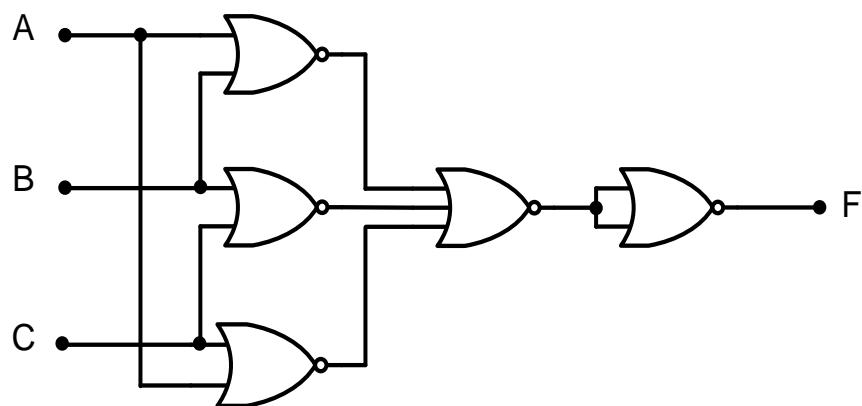
Sơ đồ:



- Chuyển đổi từng cổng sang cổng NOR

$$\begin{aligned}
 F &= \overline{BC} + \overline{AC} + \overline{AB} = (\overline{B} + \overline{C}) + (\overline{A} + \overline{C}) + (\overline{A} + \overline{B}) \\
 &= (\overline{B} + \overline{C}) + (\overline{A} + \overline{C}) + (\overline{A} + \overline{B})
 \end{aligned}$$

Sơ đồ:



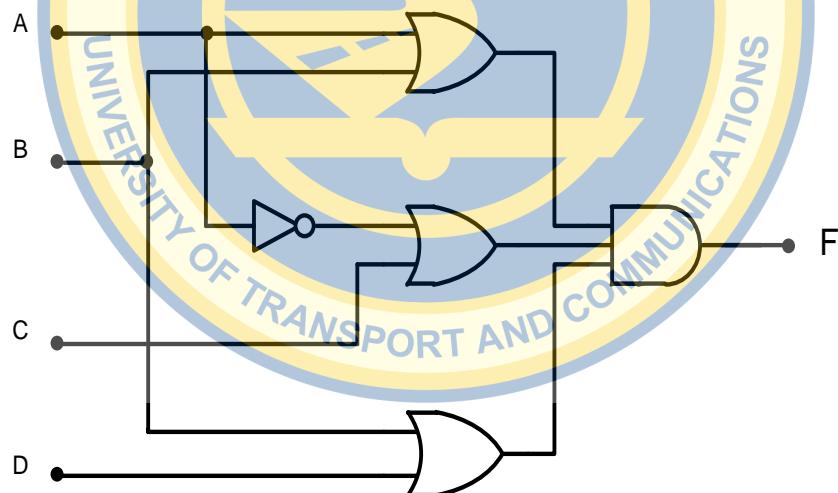
Ví dụ 5.20

Xây dựng mạch số với hàm: $F = (A+B)(\bar{A}+C)(B+D)$,

- (a) Sử dụng các cổng cơ bản.
- (b) Chỉ sử dụng cổng NAND.
- (c) Chỉ sử dụng cổng NOR

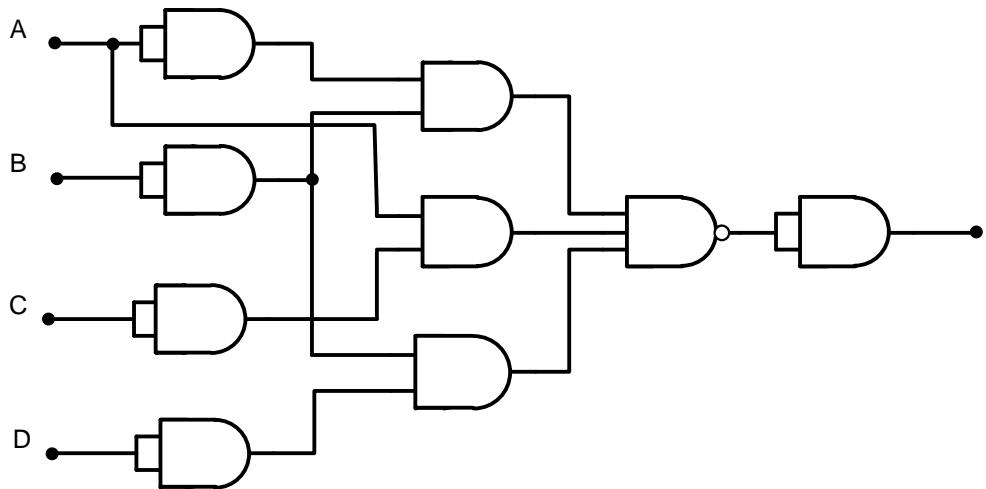
Hướng dẫn:

- (a) Sử dụng các cổng cơ bản:



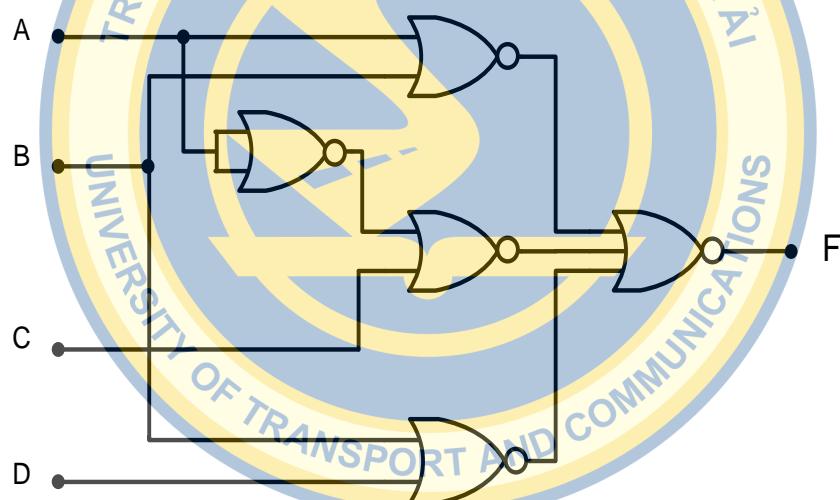
- (b) Thực hiện hàm chỉ sử dụng cổng NAND

$$\begin{aligned} F &= (A+B)(\bar{A}+C)(B+D) = (\overline{\overline{A}}+\overline{B})(\overline{\overline{A}}+\overline{C})(\overline{\overline{B}}+\overline{D}) \\ &= (\overline{AB})(\overline{AC})(\overline{BD}) = (\overline{AB})(\overline{AC})(\overline{BD}) \end{aligned}$$



(c) Thực hiện mạch chỉ sử dụng cổng NOR

$$\begin{aligned}
 F &= (A+B)(\bar{A}+\bar{C})(B+\bar{D}) = (\overline{A+B})(\overline{\bar{A}+\bar{C}})(\overline{B+\bar{D}}) \\
 &= \overline{(A+B)} + \overline{(\bar{A}+\bar{C})} + \overline{(B+\bar{D})} = \overline{(A+B)} + \overline{(AA+A\bar{C})} + \overline{(B+\bar{D})}
 \end{aligned}$$



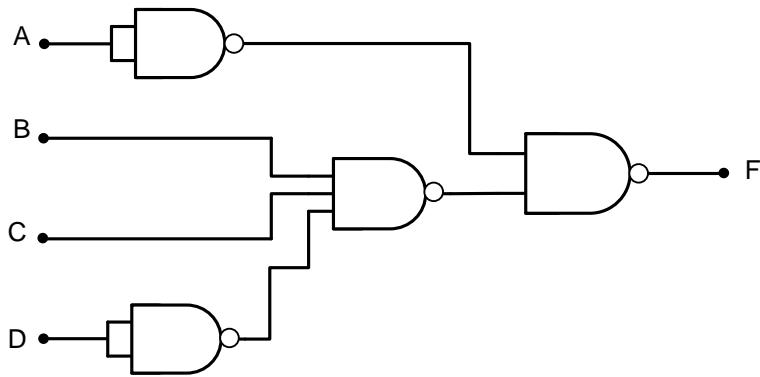
Ví dụ 5.21

- a, Thực hiện hàm $F = A + BC\bar{D}$ chỉ sử dụng cổng NAND
- b, Thực hiện hàm $F = (A + C)(A + \bar{D})(A + B + \bar{C})$ chỉ sử dụng cổng NOR.

Hướng dẫn

- a, Chỉ sử dụng cổng NAND

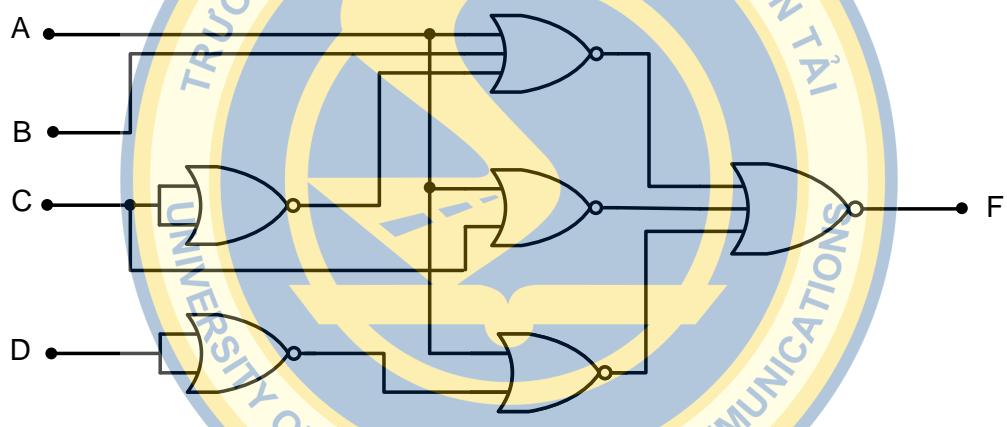
$$\text{Ta có: } F = A + BC\bar{D} = \overline{\overline{A} + BC\bar{D}} = \overline{\overline{A}} \overline{BC\bar{D}} = \overline{\overline{A}} \overline{BC} \overline{\overline{D}} = \overline{\overline{A}} \overline{BC} \overline{D}$$



b, Chỉ sử dụng cổng NOR

Ta có:

$$\begin{aligned}
 F &= (A+C)(A+\bar{D})(A+B+\bar{C}) = \overline{(A+C)} \cdot \overline{(A+\bar{D})} \cdot \overline{(A+B+\bar{C})} \\
 &= \overline{(A+C)} \cdot \overline{(A+\bar{D})} \cdot \overline{(A+B+\bar{C})} = (A+C) + (A+\bar{D}) + (A+B+\bar{C}) \\
 &= (A+C) + (A+DD) + (A+B+CC)
 \end{aligned}$$



Ôn tập chương 5

Lý thuyết

1. Nêu các bước cơ bản thực hiện thiết kế mạch điện tử số.
2. Nêu ý nghĩa của tối thiểu hóa trước khi xây dựng mạch điện tử số.
3. Trình bày các bước tối thiểu hóa dùng phương pháp đại số và phương pháp dùng bảng Karnaugh.
4. So sánh tối thiểu hóa dùng phương pháp đại số và phương pháp dùng bảng Karnaugh.

Bài tập

1. Tối thiểu hóa bằng phương pháp đại số và vẽ sơ đồ mạch, xác định tên và số lượng IC cần dùng để thi công mạch:

- a. $F = \bar{A}\bar{B} + \bar{A}\bar{B} + A\bar{B} + A\bar{B}$
- b. $F = ABC + \bar{A}BC + A\bar{B}C + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$
- c. $F = (A + B + C)(A + \bar{B} + \bar{C})(A + B + \bar{C})(A + \bar{B} + C)$
- d. $F = AB + A(B + C) + B(B + C)$

2. Tối thiểu hóa hàm sau sử dụng bảng Karnaugh:

- a. $F(A, B, C, D) = \sum 0, 1, 2, 3, 5, 7, 8, 9, 10, 11$
- b. $F(A, B, C, D) = \sum 0, 1, 3, 5, 7, 8, 9, 11, 14$

3. Tối thiểu hóa hàm sau sử dụng bảng Karnaugh:

- a. $F(A, B, C, D) = \prod 0, 1, 2, 4, 5, 6, 7, 14, 15$
- b. $f(A, B, C, D) = \prod 0, 1, 2, 4, 6, 9, 11, 12, 13, 15$

3. Sử dụng bảng Karnaugh để rút gọn biểu thức:

- a. $F = ABC + \bar{B}C + \bar{A}B$
- b. $F = \bar{C}(\bar{A}BD + D) + \bar{A}BC + \bar{C}D$

4. Thực hiện hàm sau chỉ sử dụng cổng NAND, xác định tên và số lượng IC cần dùng để thi công mạch:

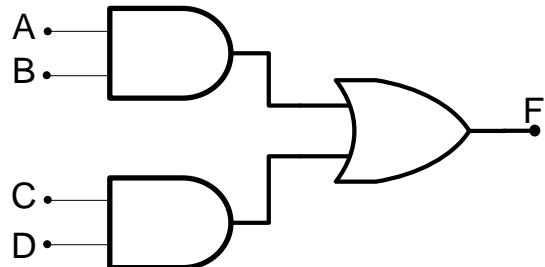
- a. $F = B(A + CD) + A\bar{C}$
- b. $F = \overline{\overline{AB}} + A + (\overline{B} + \overline{C})$

5. Thực hiện hàm sau chỉ sử dụng cổng NOR, xác định tên và số lượng IC cần dùng:

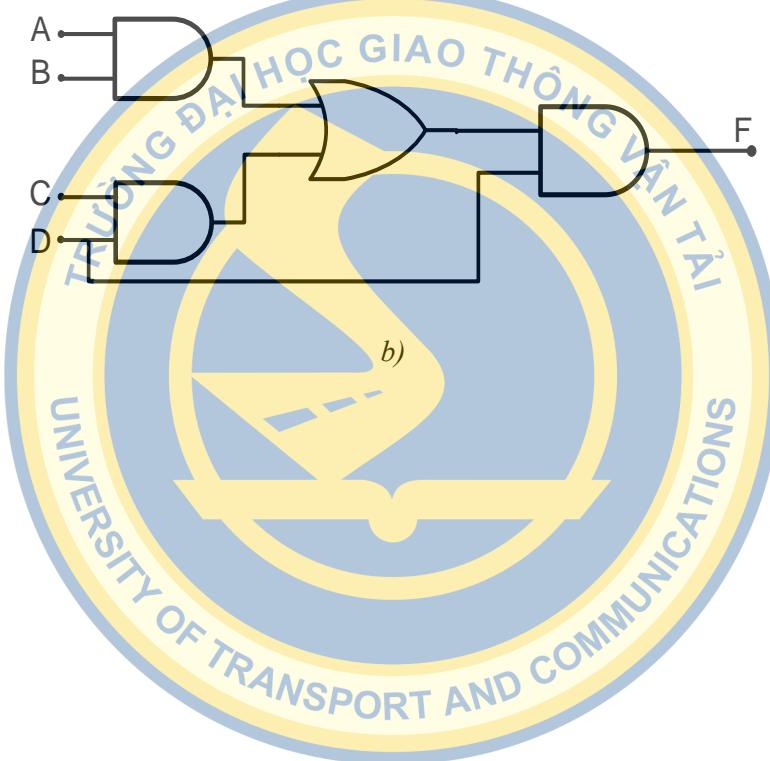
a. $F = \overline{ABC} + ABC + \overline{(A+B+C)}$

b. $F = \overline{AB} + AC + \overline{(B+C)}$

6. Xác định hàm logic đầu ra của mạch sau:



a)



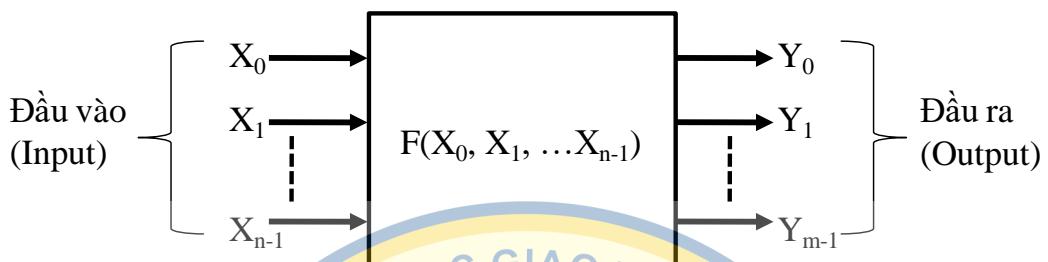
b)

Chương 6

MẠCH ĐIỆN TỬ SỐ TỔ HỢP THƯỜNG GẶP

6.1. GIỚI THIỆU

Mạch tổ hợp (hay còn gọi là mạch logic số tổ hợp) là mạch điện tử số có đầu ra (Y_0, Y_1, \dots, Y_{m-1}) chỉ phụ thuộc vào đầu vào (X_0, X_1, \dots, X_{n-1}).



Hình 6.1. Mô hình mạch số tổ hợp.

Để thiết kế mạch tổ hợp, bước quan trọng nhất là xác định mối quan hệ đầu vào/đầu ra: đây là các bước quan trọng do việc thiết kế hay việc làm cụ thể để hiểu chính xác cho việc lập bảng sự thật. Đôi khi người thiết kế phải sử dụng trực giác và kinh nghiệm của mình để giải thích chính xác vấn đề. Từ bảng sự thật, tối thiểu hóa hàm logic đầu ra, sử dụng các phương pháp đơn giản hóa như Phương pháp đại số, Bảng Karnaugh,... Thiết kế mạch thực tế cần đưa ra được: số cổng logic là ít nhất, thời gian truyền của tín hiệu qua mạch là nhỏ nhất, kết nối trong mạch là ít nhất, giới hạn về khả năng điều khiển mỗi cổng logic.

Chương này tập trung phân tích và thiết kế các mạch tổ hợp ứng dụng trong tính toán, chuyển mã, truyền số liệu như mạch cộng, trừ nhị phân; mạch giải mã, lập mã, mạch ghép kênh, phân kênh, mạch kiểm tra gói tin,...

6.2. BỘ CỘNG

Việc xử lý thông tin được thực hiện bởi máy tính số. Các phép toán học là các hàm cơ bản của một máy tính số. Phép cộng hai số nhị phân là một phép toán quan trọng, phép cộng đơn giản gồm:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

6.2.1 Bộ bán cộng 1 bit

Bộ bán cộng nhị phân (Half adder) 1 bit, có hai đầu vào và hai đầu ra. Gọi A, B là biến đầu vào và đầu ra là tổng S (sum) và số nhớ C (carry), bảng sự thật như sau:

Bảng 6.1. Bảng sự thật bộ bán cộng nhị phân 1 bit.

Các biến đầu vào		Các biến đầu ra	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Từ bảng sự thật 6.1 có thể thấy hàm ra S và C là cỗng XOR và AND. Biểu thức Boolean là:

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

$$C = AB$$

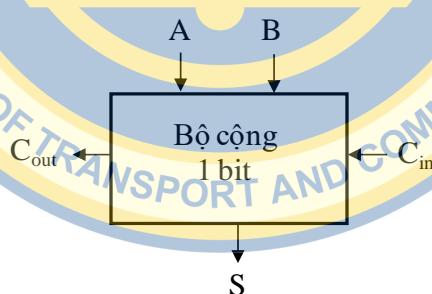
Hình 6.2 biểu diễn sơ đồ mạch bộ bán tổng:



Hình 6.2. Sơ đồ mạch bộ bán cộng nhị phân 1 bit.

6.2.2. Bộ cộng đầy đủ 1 bit

Mạch tổ hợp của bộ cộng đầy đủ (Full Adder) thực hiện phép cộng gồm 3 bit đầu vào: số thứ nhất (A), số thứ hai (B) và số nhớ trước đó đưa đến (C_{in}), đầu ra: tổng ra (S), số nhớ sau (C_{out}).



Bảng 6.2. Bảng sự thật bộ cộng nhị phân 1 bit.

Các biến đầu vào			Các biến đầu ra	
C_{in}	A	B	S	C_{out}
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

Sử dụng bảng Karnaugh để đơn giản biểu thức đầu ra S và C_{out} như sau:

		AB		00	01	11	10
		C_{in}	0	0	1	0	1
				1	0	1	0

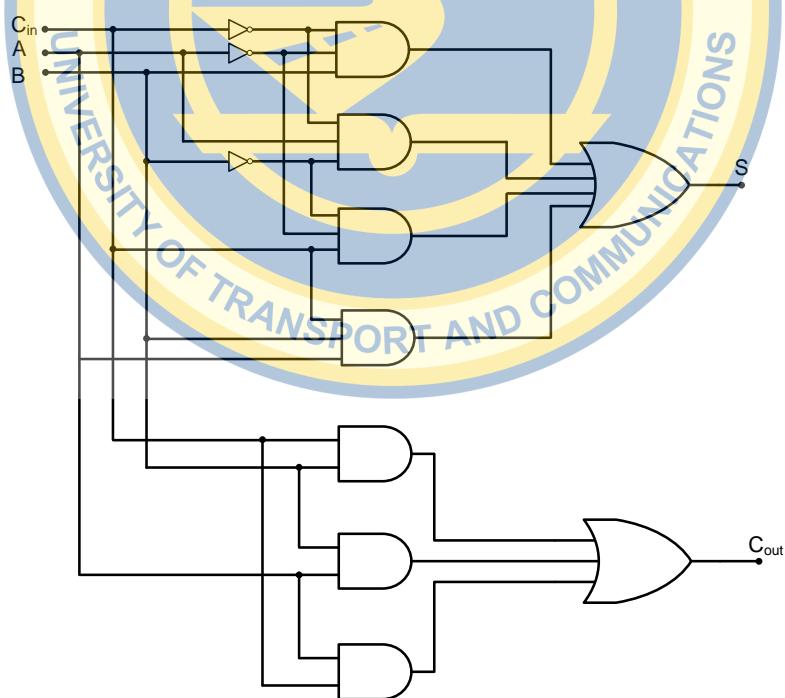
		AB		00	01	11	10
		C_{in}	0	0	0	1	0
				1	0	(1)	1

Đơn giản biểu thức Boolean của đầu ra ta được:

$$S = \bar{C}_{in} \bar{A}B + \bar{C}_{in} A\bar{B} + C_{in} \bar{A}\bar{B} + C_{in} AB$$

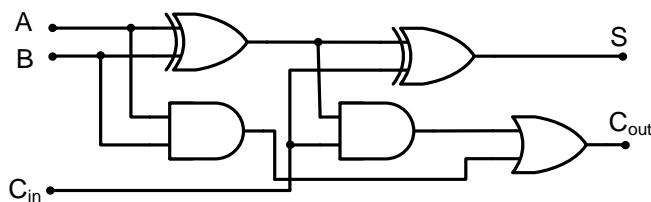
$$C_{out} = AB + BC_{in} + AC_{in}$$

Sơ đồ mạch logic như hình 6.3 với A, C_{in}, B là các biến đầu vào.



Hình 6.3. Sơ đồ mạch bộ cộng đầy đủ 1 bit.

Mạch trên có thể biến đổi từ bộ cộng bán tổng:

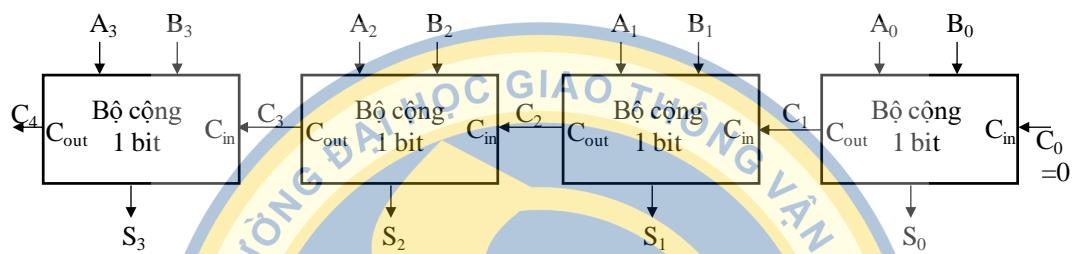


Hình 6.4. Sơ đồ mạch cộng đầy đủ xây dựng từ bộ bán cộng.

6.2.3. Bộ cộng nhiều bit

Bộ cộng nhiều bit được thực hiện bằng cách bắt đầu cộng từ bit LSB sau đó số nhớ được đưa lên bit có nghĩa cao hơn.

Ví dụ sơ đồ mạch cộng hai số 4 bit A₃A₂A₁A₀ và B₃B₂B₁B₀, ta có sơ đồ như sau:



6.3. MẠCH TRỪ

Tương tự như ở phép tính cộng, phép trừ hai số nhị phân gồm bốn phép tính cơ bản là:

$$\begin{aligned} 0 - 0 &= 0 \\ 0 - 1 &= 1 \text{ mượn } 1 \\ 1 - 0 &= 1 \\ 1 - 1 &= 0 \end{aligned}$$

6.3.1. Thiết kế bộ bán trừ

Bộ bán trừ (Half Subtractor) gồm hai đầu vào và hai đầu ra. Các biến đầu vào gồm số trừ và số bị trừ là A và B, đầu ra được hiệu là D và B_{out} (số mượn). Bảng sự thật như sau:

Bảng 6.3. Bảng sự thật của bộ bán trừ.

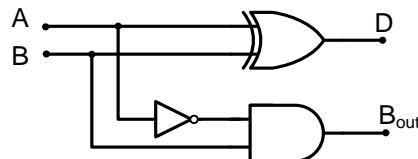
Các biến đầu vào		Các biến đầu ra	
A	B	D	B_{out}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Từ bảng sự thật ta có biểu thức Boolean cho hàm ra D và B như sau:

$$D = \overline{AB} + A\overline{B}$$

$$\text{và } B_{out} = \overline{AB}$$

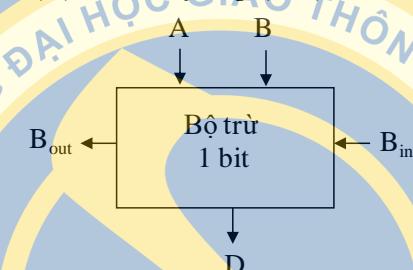
Hình 6.5 biểu diễn sơ đồ mạch bù bán trừ.



Hình 6.5. Sơ đồ mạch bù bán trừ.

6.3.2. Thiết kế bộ trừ đầy đủ

Mạch trừ nhị phân đầy đủ (Full Subtractor) gồm 3 bit vào: số bị trừ (A), số trừ (B) và số đã vay (B_{in}), tạo đầu ra (D) và số vay tiếp (B_{out}). Ta có bảng sự thật như bảng 6.4



Bảng 6.4. Bảng sự thật bộ trừ đầy đủ.

Đầu vào			Đầu ra	
A	B	B_{in}	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

A	B_{in}			
	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Bảng K cho hàm D

A	B_{in}			
	00	01	11	10
0	0	(1)	(1)	1
1	0	0	1	0

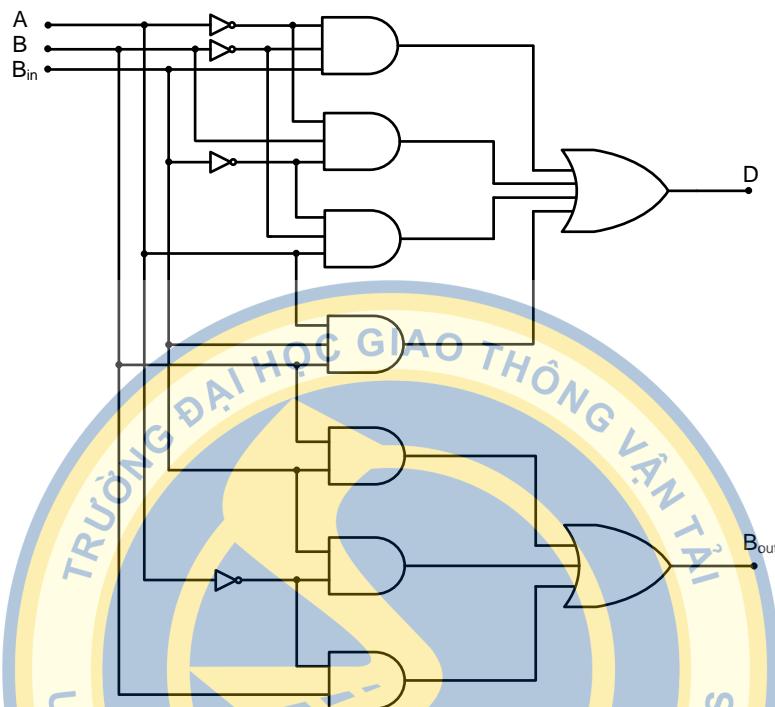
Bảng K cho hàm B_{out}

Tối giản biểu thức đầu ra ta được:

$$D = \overline{A}\overline{B}B_{in} + \overline{A}\overline{B}\overline{B}_{in} + A\overline{B}\overline{B}_{in} + AB\overline{B}_{in}$$

$$B_{out} = \overline{AB}_{in} + \overline{AB} + BB_{in}$$

Sơ đồ mạch được biểu diễn như hình 6.6

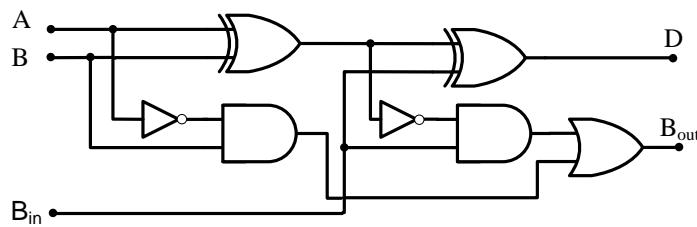


Hình 6.6. Sơ đồ mạch bộ trù đầy đủ 1 bit.

Mạch trù đầy đủ trên cũng có thể xây dựng từ mạch bán trù sử dụng biến đổi sau:

$$\begin{aligned} D &= \overline{A}\overline{B}B_{in} + \overline{A}\overline{B}\overline{B}_{in} + A\overline{B}\overline{B}_{in} + AB\overline{B}_{in} \\ &= A(\overline{B}B_{in} + \overline{B}\overline{B}_{in}) + A(\overline{B}\overline{B}_{in} + B\overline{B}_{in}) \\ &= A \oplus B \oplus B_{in} \end{aligned}$$

$$\begin{aligned} B_{out} &= \overline{AB}_{in} + \overline{AB} + BB_{in} = \overline{AB} + B_{in}(\overline{A} + \overline{B}) = \overline{AB} + B_{in}(\overline{AB} + \overline{AB} + AB + \overline{AB}) \\ &= \overline{AB} + B_{in}(\overline{AB} + AB) = \overline{AB} + ABB_{in} + B_{in}(AB + AB) \\ &= \overline{AB} + B_{in}(A \oplus B) \end{aligned}$$

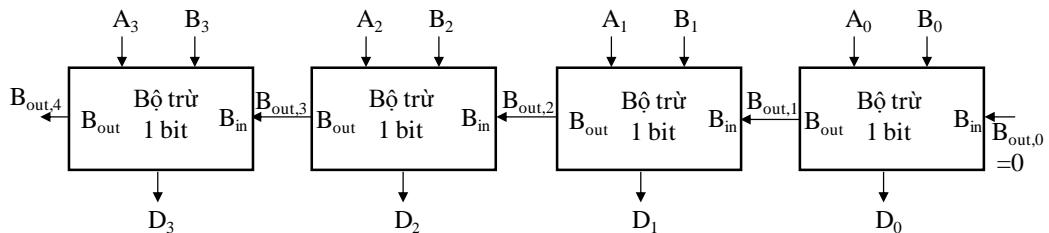


Hình 6.7. Sơ đồ mạch trù đầy đủ xây dựng từ bộ bán trù.

6.3.3. Bộ trừ nhiều bit

Bộ trừ nhiều bit được thực hiện bằng cách bắt đầu trừ từ bít LSB sau đó số vay được đưa lên bit có nghĩa cao hơn để trả.

Ví dụ sơ đồ thực hiện mạch trừ hai số 4 bit $A_3A_2A_1A_0$ và $B_3B_2B_1B_0$, ta có sơ đồ như sau:



6.4. MẠCH SO SÁNH

– Mạch so sánh (comparator) 1 bit

So sánh 2 số nhị phân 1 bit A và B, sẽ có ba khả năng đầu ra: $A=B$, $A>B$, $A<B$, với bảng sự thật:

Bảng 6.5. Bảng sự thật mạch so sánh 1 bit.

Đầu vào		Đầu ra		
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

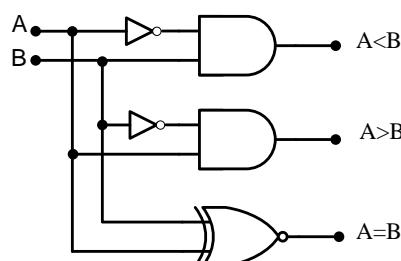
Từ bảng trên ta thấy:

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

$$\text{Hàm } (A > B) = A \cdot \overline{B}$$

$$\text{Hàm } (A < B) = \overline{A} \cdot B$$

Sơ đồ của bộ so sánh 1 bit:



Hình 6.8. Mạch so sánh nhị phân 1 bit.

– Mạch so sánh nhiều bit

Mạch so sánh nhiều bit cũng được xây dựng dựa trên mạch của 3 trường hợp trên. Ví dụ xét mạch so sánh 3 bit:

$$A = A_3A_2A_1$$

$$B = B_3B_2B_1$$

Khi đó ta thấy:

$$+ (A = B), \text{ khi: } (A_3 = B_3)(A_2 = B_2)(A_1 = B_1)$$

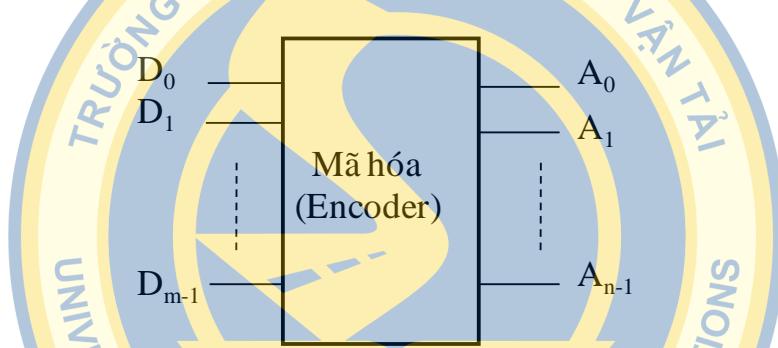
$$+ (A > B), \text{ khi } (A_3 > B_3) + (A_3 = B_3)(A_2 > B_2) + (A_3 = B_3)(A_2 = B_2)(A_1 > B_1)$$

$$+ (A < B), \text{ khi } (A_3 < B_3) + (A_3 = B_3)(A_2 < B_2) + (A_3 = B_3)(A_2 = B_2)(A_1 < B_1)$$

Mạch so sánh 3 bit này được xây dựng bằng cách kết hợp các hàm hoặc, các mạch kết quả 3 trường hợp so sánh trên.

6.5. MẠCH LẬP MÃ

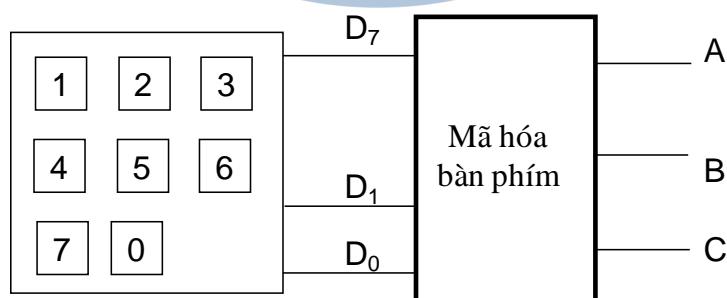
Mạch mã hóa (encoder) là mạch logic tổ hợp có 2^n (hoặc ít hơn) và n bit đầu ra.



Hình 6.9. Sơ đồ chung bộ mã hóa.

Ví dụ xét mạch lập mã nhị phân của bàn phím 8 số.

Gọi D₀ → D₇ biểu diễn ký của các số từ 0 → 7. Gọi A, B, C là các bit nhị phân đầu ra, ta có sơ đồ và bảng sự thật như sau:



Hình 6.10. Sơ đồ bộ mã hóa bàn phím.

Bảng 6.6. Bảng sự thật bộ mã hóa bàn phím 8 số.

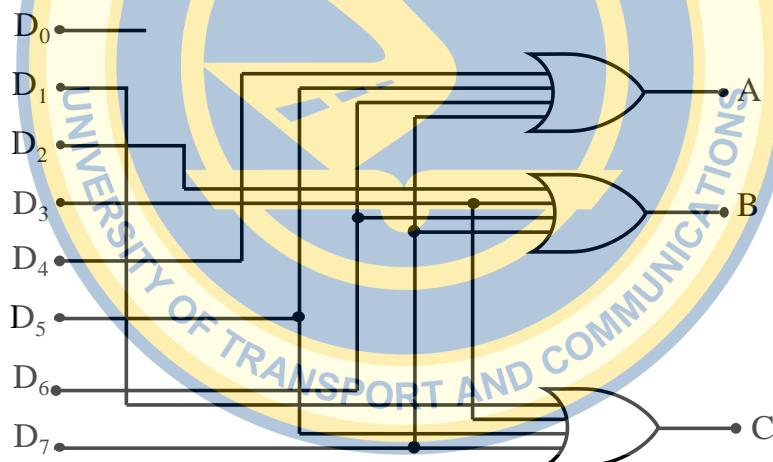
Đầu vào								Đầu ra		
D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Ta xác định được giá trị đầu ra và sơ đồ mạch:

$$A = D_4 + D_5 + D_6 + D_7$$

$$B = D_2 + D_3 + D_6 + D_7$$

$$C = D_1 + D_3 + D_5 + D_7$$



Hình 6.11. Sơ đồ mạch mã hóa bàn phím.

Mạch mã hóa ưu tiên

Trong mạch mã hóa này, việc ưu tiên được gán cho mỗi đầu vào khi có nhiều hơn một đầu vào cùng hoạt động (logic 1), đầu vào có mức ưu tiên cao nhất sẽ được mã hóa.

Ví dụ như xét mạch mã hóa trên, giả sử mạch ưu tiên cho ký số có trọng số cao. Khi đó, bảng sự thật mạch mã hóa ưu tiên như sau:

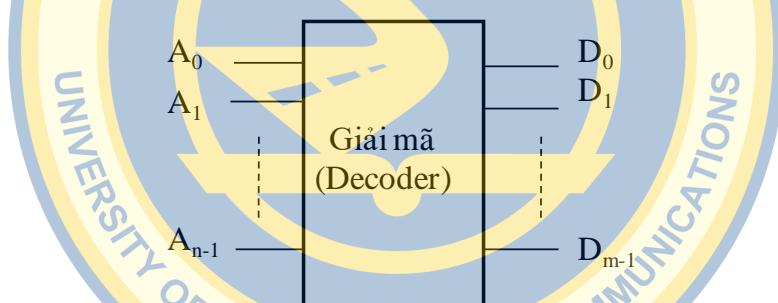
Bảng 6.7. Bảng sự thật mạch mã hóa ưu tiên 8 đầu vào

Đầu vào								Đầu ra		
D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	A	B	C
1	0	0	0	0	0	0	0	0	0	0
X	1	0	0	0	0	0	0	0	0	1
X	X	1	0	0	0	0	0	0	1	0
X	X	X	1	0	0	0	0	0	1	1
X	X	X	X	1	0	0	0	1	0	0
X	X	X	X	X	1	0	0	1	0	1
X	X	X	X	X	X	1	0	1	1	0
X	X	X	X	X	X	X	1	1	1	1

Như vậy D₇ có mức ưu tiên cao nhất (nghĩa là sẽ được phục vụ đầu tiên), giả sử cùng một thời điểm, các phím số 2,5,7 cùng được bấm, thì D₇ được mã hóa, các đầu ra khác D₀-D₆ sẽ bị cấm và đầu ra sẽ là 111.

6.6. MẠCH GIẢI MÃ

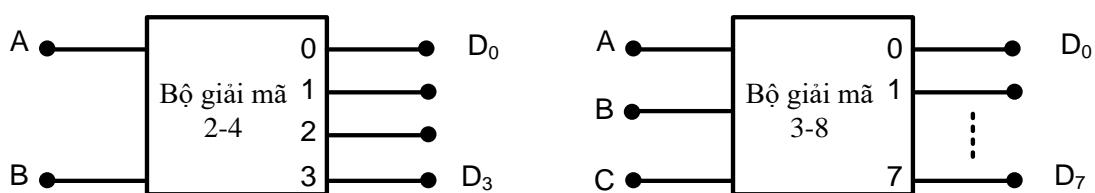
Ở chương 1, ta thấy rằng có nhiều loại mã được dùng trong các hệ thống số khác nhau, vì vậy cần có mạch chuyển mã khi có sự kết nối giữa hai mạch số khác nhau đó. Mạch giải mã (decoder) chuyển đổi n bit đầu vào thành m bit đầu ra ($\leq 2^n$).



Hình 6.12. Sơ đồ chung của mạch giải mã.

6.6.1. Mạch giải mã n đầu vào 2^n đầu ra

Mạch giải mã là mạch tổ hợp giải mã thông tin với n đầu vào đến 2^n đầu ra lớn nhất. Các mạch này được ứng dụng trong các mạch truy xuất bộ nhớ, thiết bị ngoại vi của vi xử lý. Hình 6.13 biểu diễn bộ giải mã $2 \rightarrow 4$, $3 \rightarrow 8$.



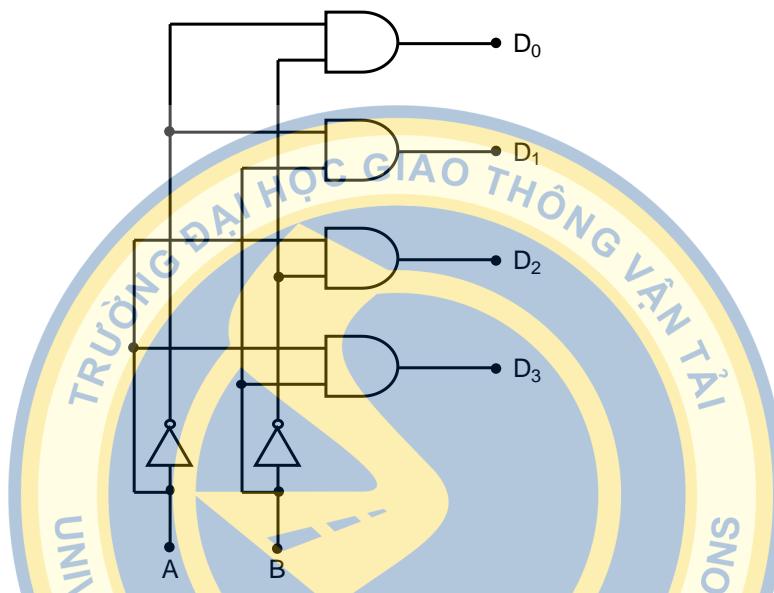
Hình 6.13. Mạch giải mã $2 \rightarrow 4$, $3 \rightarrow 8$.

– Mạch giải mã $2 \rightarrow 4$: với đầu vào là A, B và tám đầu ra là D₀, D₁, D₂, D₃, D₄.

Bảng 6.8. Bảng sự thật mạch giải mã $3 \rightarrow 8$.

Đầu vào		Đầu ra			
A	B	D ₀	D ₁	D ₂	D ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Ta có sơ đồ mạch:



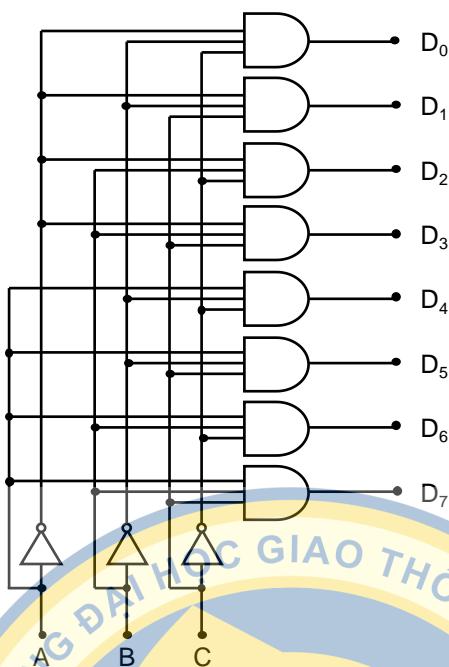
Hình 6.14. Sơ đồ mạch logic bộ giải mã $2 \rightarrow 4$.

– Mạch giải mã $3 \rightarrow 8$: với ba đầu vào là A, B, C và tám đầu ra là D₀, D₁, D₂, D₃, D₄, D₅, D₆, và D₇.

Bảng 6.9. Bảng sự thật mạch giải mã $3 \rightarrow 8$.

Các biến đầu vào			Các biến đầu ra							
A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Ta có sơ đồ mạch:



Hình 6.15. Sơ đồ mạch logic bộ giải mã $3 \rightarrow 8$.

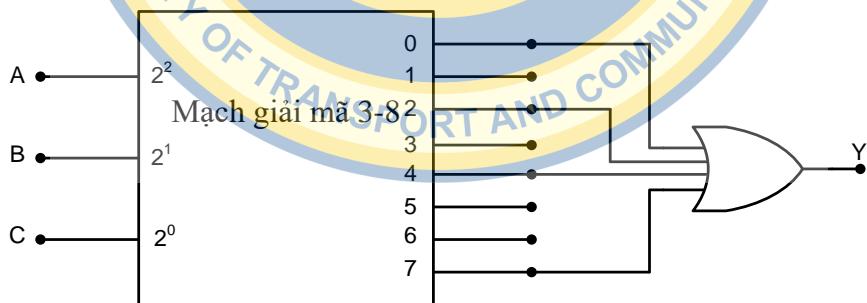
6.6.2. Thực hiện hàm logic sử dụng bộ giải mã

Quan hệ logic của mạch giải mã là các tích, cho nên trong một số trường hợp, có thể sử dụng tính chất này kết hợp với mạch OR để thiết kế mạch logic mà không cần sử dụng công logic cơ bản.

Ví dụ 6.1 Hàm logic có phương trình:

$$Y = A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + ABC$$

Được xây dựng từ Mạch giải mã $3 \rightarrow 8$, như hình sau:



Hình 6.16. Bộ giải mã thực hiện hàm logic trên.

Ví dụ 6.2: Xây dựng bộ cộng đầy đủ sử dụng mạch giải mã $3 \rightarrow 8$.

Mạch giải mã với công OR tại đầu ra có thể được sử dụng để xây dựng hàm Boolean. Mạch giải mã có số đầu vào bằng số biến trong hàm Boolean. Bảng sự thật 6.18 và hình 6.33 mạch thực hiện.

Bảng 6.10. Bảng sự thật bộ cộng nhị phân 1 bit và đầu ra mạch giải mã.

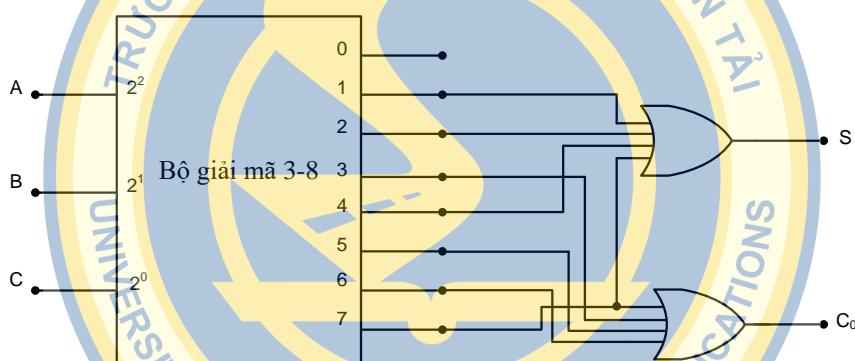
Tương ứng đầu ra của mạch giải mã $3 \rightarrow 8$	Đầu vào bộ cộng			Đầu ra bộ cộng	
D	A	B	C	S	C_0
D ₀	0	0	0	0	0
D ₁	0	0	1	1	0
D ₂	0	1	0	1	0
D ₃	0	1	1	0	1
D ₄	1	0	0	1	0
D ₅	1	0	1	0	1
D ₆	1	1	0	0	1
D ₇	1	1	1	1	1

Từ bảng trên, ta có:

$$S = D_1 + D_2 + D_4 + D_7$$

$$C_0 = D_3 + D_5 + D_6 + D_7$$

Vậy bộ cộng nhị phân 1 bit được xây dựng từ mạch giải mã $3 \rightarrow 8$, như hình sau:



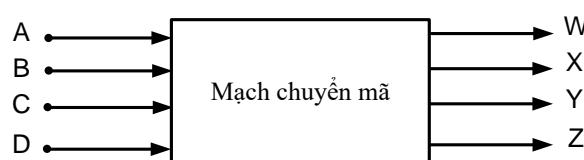
Hình 6.17. Xây dựng mạch cộng nhị phân 1 bit bộ giải mã.

6.6.3. Mạch giải mã n đầu vào n đầu ra

Đây là mạch giải mã thường gặp với các mã như nhị phân, gray, dư 3,... các trường hợp này số bit đầu vào của mạch sẽ bằng với số bit đầu ra.

– **Ví dụ 6.1:**

Thiết kế mạch chuyển mã từ Nhị phân 4 bit sang mã Gray.



Hình 6.18. Sơ đồ khối mạch chuyển mã nhị phân \rightarrow gray.

Hướng dẫn:

Hình 6.18 là sơ đồ khối của mạch

Từ các tổ hợp mã của mã Nhị phân 4 bit và Gray 4 bit (xem bảng 1.8, chương 1), ta có bảng sự thật sau:

Bảng 6.11. Bảng sự thật mạch chuyển mã Nhị phân \rightarrow Gray 4 bit.

Nhị Phân (Đầu vào)				Gray (Đầu ra)			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	0	1	0	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	0

AB	CD		W	
	00	01		11
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Bảng K cho W

AB	CD		X	
	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

Bảng K cho X

	CD	00	01	11	10
AB	00	0	0	1	1
	00	1	1	0	0
	01	1	1	0	0
	11	1	1	0	0
	10	0	0	1	1

Bảng K cho Y

	CD	00	01	11	10
AB	00	0	1	0	1
	00	0	1	0	1
	01	0	1	0	1
	11	0	1	0	1
	10	0	1	0	1

Bảng K cho Z

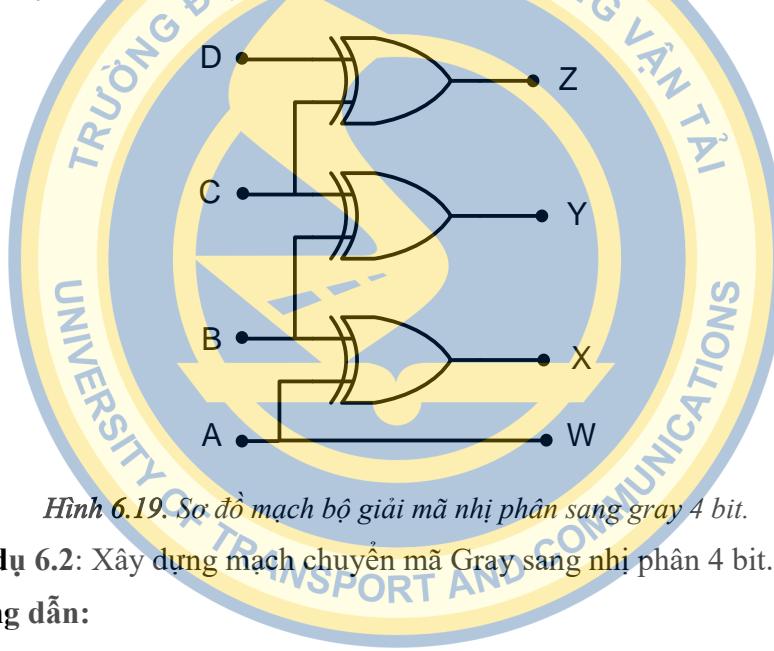
$$X = \bar{A}B + A\bar{B} = A \oplus B$$

$$W = A$$

$$Y = BC + \bar{B}C = B \oplus C$$

$$Z = \bar{C}D + CD = C \oplus D$$

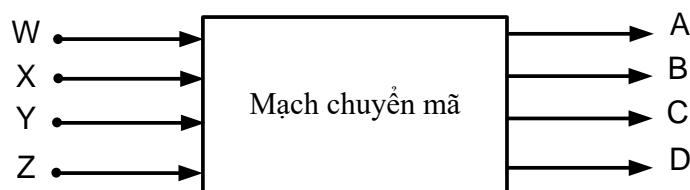
Sơ đồ mạch:



Hình 6.19. Sơ đồ mạch bộ giải mã nhị phân sang gray 4 bit.

– **Ví dụ 6.2:** Xây dựng mạch chuyển mã Gray sang nhị phân 4 bit.

Hướng dẫn:



Hình 6.20. Sơ đồ khái niệm của một mạch chuyển mã.

Từ các tổ hợp mã của mã Gray và mã Nhị phân 4 bit (xem bảng 1.8, chương 1), ta có bảng sự thật sau với đầu vào là W, X, Y, Z và đầu ra là A, B, C, D:

Bảng 6.12. Bảng sự thật mạch chuyển mã Gray \rightarrow Nhị phân 4 bit.

Gray (Đầu vào)				Nhị Phân (Đầu ra)			
W	X	Y	Z	A	B	C	D
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	0	1	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	1	0
1	0	0	1	1	1	1	1
1	0	0	0	1	1	1	1

Ta có bảng karnaugh dưới đây,

YZ	00	01	11	10
WX	00	0	0	0
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Bảng K của A

YZ	00	01	11	10
WX	00	0	0	0
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

Bảng K của B

YZ	00	01	11	10
WX	00	0	1	1
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

Bảng K của C

YZ	00	01	11	10
WX	00	0	1	0
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

Bảng K của D

Biểu thức đại số logic đầu ra là:

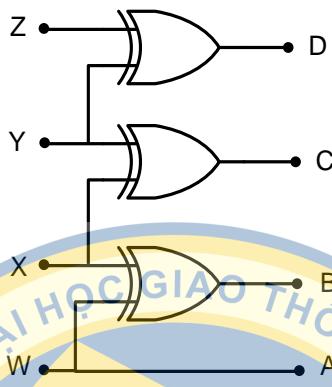
$$A = W$$

$$B = \overline{W}X + W\overline{X} = W \oplus X$$

$$C = \overline{W}XY + \overline{W}\overline{X}Y + W\overline{X}Y + W\overline{X}\overline{Y} = W \oplus X \oplus Y$$

$$D = WXYZ + WXYZ = W \oplus X \oplus Y \oplus Z$$

Sơ đồ mạch như sau:

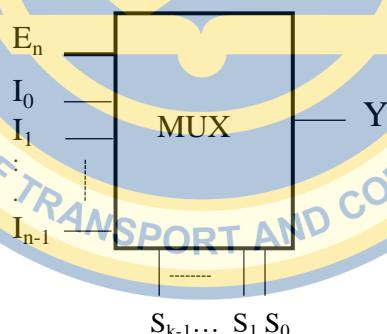


Hình 6.21. Sơ đồ mạch bộ giải mã gray → nhị phân 4 bit.

6.7. MẠCH GHÉP KÊNH VÀ PHÂN KÊNH

6.7.1. Mạch ghép kênh

Mạch ghép kênh (Multiplexer) viết tắt là MUX, là một chuyển mạch số hay mạch chọn dữ liệu, có n đầu vào, m đường chọn địa chỉ (S), 1 đầu cho phép (E_n) và một đầu ra Y (có thể có thêm đầu ra đảo).



Hình 6.22. Sơ đồ tổng quát mạch MUX.

Tuỳ theo giá trị của n đầu vào địa chỉ mà đầu ra sẽ bằng một trong những giá trị ở đầu vào I_j. Cụ thể là nếu giá trị thập phân của tổ hợp (S_{n-1}...S₁S₀) bằng j thì Y = I_j. Cũng giống như mạch đồng bộ, mạch MUX được thêm chân cho phép (Enable), E_n = 1, mạch hoạt động bình thường và khi E_n=0 mạch bị cấm hay không hoạt động.

MUX được dùng như một phần tử vạn năng để xây dựng những mạch tổ hợp khác, cụ thể là: tạo hàm logic, bộ chuyển đổi song song/nối tiếp, giải mã địa chỉ,....

– MUX 2 đầu vào

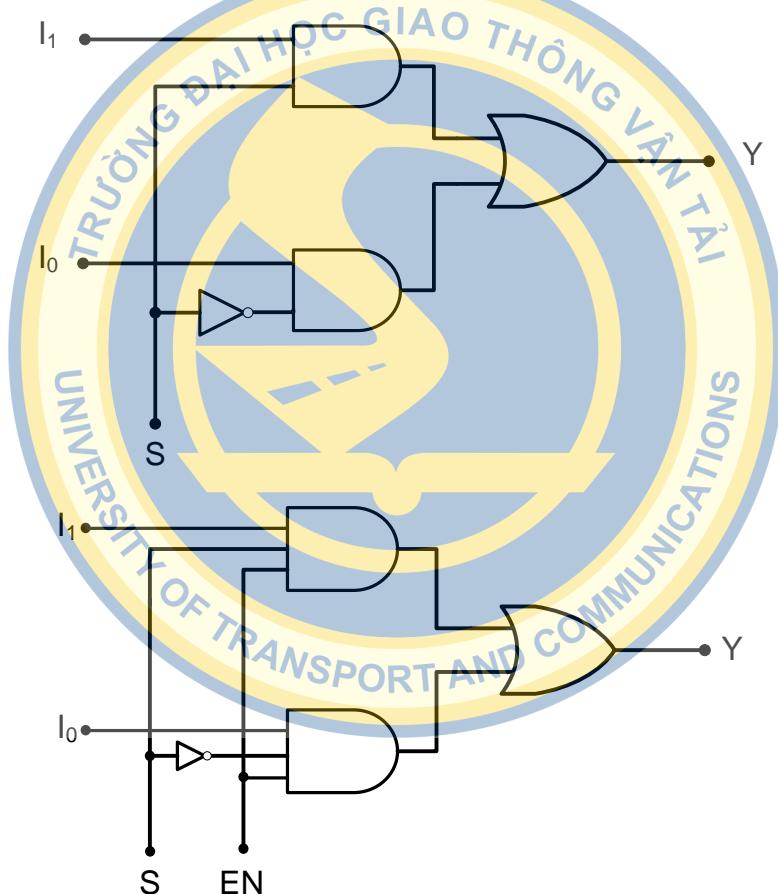
Bảng sự thật:

Đầu vào		Đầu ra
EN	S	Y
0	X	0
1	0	I ₀
1	1	I ₁

Ta được:

$$Y = I_0 \bar{S} + I_1 S$$

Sơ đồ mạch:



Hình 6.23. Sơ đồ mạch ghép kênh 2-1 có và không có chân EN.

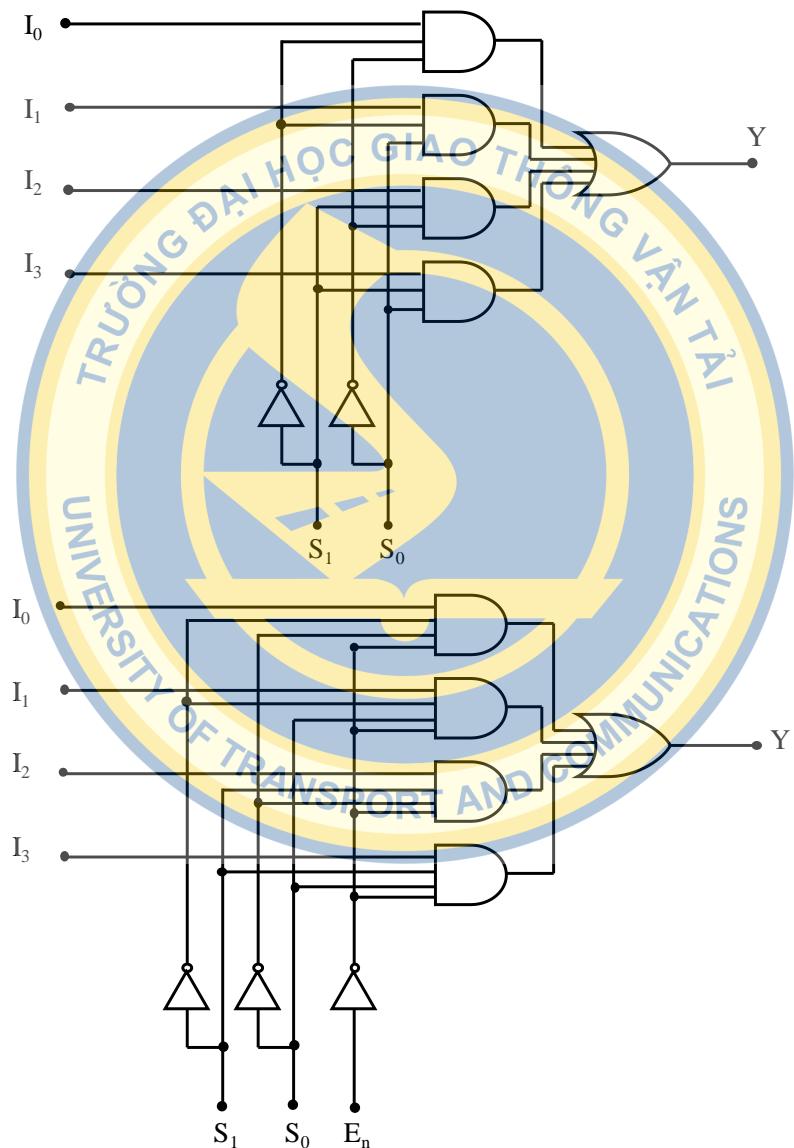
– MUX 4 đầu vào

Bảng sự thật:

Đầu vào			Đầu ra
EN	S ₁	S ₀	Y
1	X	X	0
0	0	0	I ₀
0	0	1	I ₁
0	1	0	I ₂
0	1	1	I ₃

$$Y = I_0 \bar{S}_1 \bar{S}_0 + I_1 \bar{S}_1 S_0 + I_2 S_1 \bar{S}_0 + I_3 S_1 S_0$$

Sơ đồ mạch:



Hình 6.24. Sơ đồ mạch ghép kênh có (hình trên) và không có En (hình dưới).

6.6.2. Thực hiện hàm logic sử dụng bộ ghép kênh

Cũng giống như mạch giải mã, MUX có thể sử dụng để tạo mạch logic.

Ví dụ sử dụng MUX 8 \rightarrow 1 thiết kế mạch có hàm

$$f(A, B, C) = \sum 2, 4, 7$$

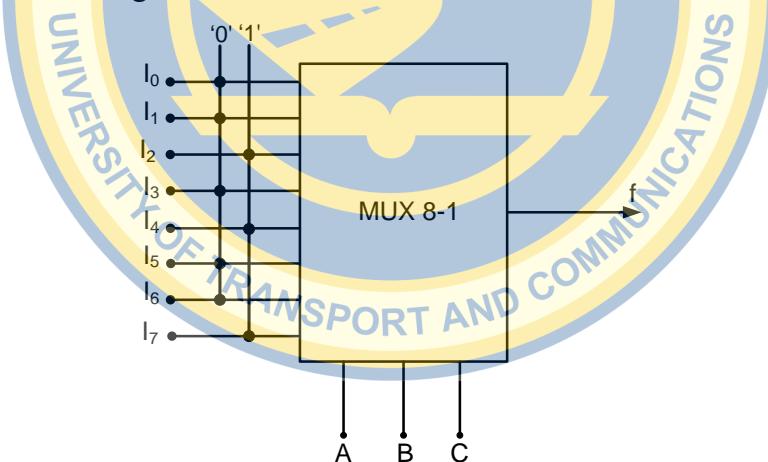
Với các biến A, B, C, tổng trên có thể viết:

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

Bảng sự thật của hàm tương ứng với đầu ra MUX 8 đầu vào:

Đầu ra MUX	A	B	C	f (A,B,C)
I ₀	0	0	0	0
I ₁	0	0	1	0
I ₂	0	1	0	1
I ₃	0	1	1	0
I ₄	1	0	0	1
I ₅	1	0	1	0
I ₆	1	1	0	0
I ₇	1	1	1	1

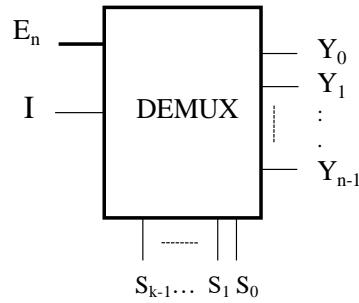
Từ bảng so sánh này, nếu đặt I₀ = I₁ = I₃ = I₅ = I₆ = 0 và I₂ = I₄ = I₇ = 1, thì đầu ra của MUX có chức năng của hàm trên, ta có sơ đồ mạch:



Hình 6.25. Sơ đồ mạch của hàm trên thực hiện bằng MUX.

6.7.2. Mạch phân kênh

Mạch phân kênh (Demultiplexer) viết tắt là DEMUX, là một chuyển mạch số hay mạch chọn dữ liệu, có 1 đầu vào, m đường chọn địa chỉ (S), 1 đầu cho phép (E_n) và n đầu ra (Mạch giải mã nêu trên chính là trường hợp đặc biệt của bộ phân kênh khi không có đầu vào).



Hình 6.26. Sơ đồ tóm tắt mạch DEMUX.

Tuỳ theo giá trị của n đầu vào chọn địa chỉ mà đầu ra sẽ bằng một trong những giá trị ở đầu vào I. Cụ thể là nếu giá trị thập phân của tổ hợp ($S_{n-1} \dots S_1 S_0$) bằng j thì $Y_j = I$. Cũng giống như mạch MUX, mạch DEMUX được thêm chân cho phép (Enable), $E_n = 1$, mạch hoạt động bình thường và khi $E_n=0$ mạch bị cấm hay không hoạt động.

Ví dụ: mạch DEMUX 4 đầu ra

Bảng sự thật:

Đầu vào		Đầu ra			
S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

Ta có:

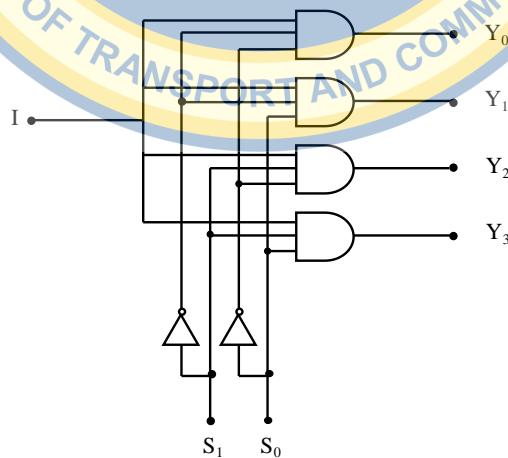
$$Y_0 = I \overline{S_1} \overline{S_0}$$

$$Y_1 = I S_1 \overline{S_0}$$

$$Y_2 = I \overline{S_1} S_0$$

$$Y_3 = I S_1 S_0$$

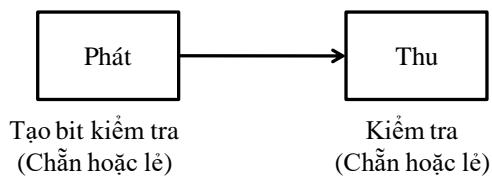
Sơ đồ mạch:



Hình 6.27. Sơ đồ mạch DEMUX 4 đầu ra.

6.8. MẠCH TẠO VÀ KIỂM TRA CHẴN VÀ LẺ

Phương pháp tạo và kiểm tra chẵn hay lẻ là một phương pháp đơn giản nhưng hiệu quả để kiểm tra xem đường truyền có lỗi hay không. Dựa vào cấu trúc bản tin tạo phía phát, phía thu sẽ kiểm tra.



Hình 6.28. Tạo và kiểm tra chẵn và lẻ trong truyền dữ liệu.

Có hai hệ thống là tạo và kiểm tra bit chẵn và tạo và kiểm tra bit lẻ.

6.8.1. Mạch tạo bit kiểm tra chẵn/lẻ

-Tạo bit kiểm tra chẵn: phía phát sẽ thêm vào 1 bit gọi là bit chẵn (P_e) sao cho tổng số bit “1” truyền đi là một số chẵn:

-Tạo bit kiểm tra lẻ: phía phát sẽ thêm vào 1 bit gọi là bit lẻ (P_o) sao cho tổng số bit “1” truyền đi là một số lẻ:

Ví dụ xét hệ thống truyền dữ liệu 4 bit, gọi tổ hợp bit bản tin được thiết kế là $D_3D_2D_1D_0$, bảng sau biểu diễn giá trị bit cần thêm vào cho hệ thống kiểm tra chẵn và lẻ.

Bảng 6.12. Bảng giá trị thêm vào để kiểm tra chẵn và lẻ với 4 bit truyền.

Bản tin 4 bit $D_3D_2D_1D_0$	Bit chẵn (P_e)	Bản tin 4 bit $D_3D_2D_1D_0$	Bit lẻ (P_o)
0000	0	0000	1
0001	1	0001	0
0010	1	0010	0
0011	0	0011	1
0100	1	0100	0
0101	0	0101	1
0110	0	0110	1
0111	1	0111	0
1000	1	1000	0
1001	0	1001	1
1010	0	1010	1
1011	1	1011	0
1100	0	1100	1
1101	1	1101	0
1110	1	1110	0
1111	0	1111	1
Chuỗi bit truyền đi thực		Chuỗi bit truyền đi thực	

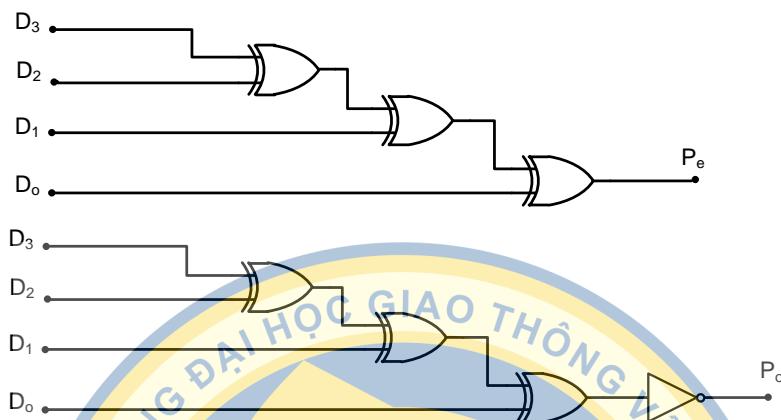
Từ bảng ta có phương trình hàm logic (dùng bảng Karnaugh):

$$P_e = D_3 \oplus D_2 \oplus D_1 \oplus D_0$$

và

$$P_o = \overline{D_3 \oplus D_2 \oplus D_1 \oplus D_0}$$

Sơ đồ mạch như hình sau:



Hình 6.29. Mạch tạo bit chẵn (hình trên) và lẻ (hình dưới).

6.8.2. Mạch kiểm tra chẵn lẻ

-Kiểm tra chẵn: phía thu sẽ kiểm tra nếu tổng số bit “1” nhận được là một số chẵn thì quá trình truyền dẫn không có lỗi;

-Kiểm tra lẻ: phía thu sẽ kiểm tra nếu tổng số bit “1” nhận được là một số lẻ thì quá trình truyền dẫn không có lỗi;

Ví dụ xét hệ thống nhận dữ liệu 4 bit, gọi tổ hợp bit bản tin nhận được là $D_2D_1D_0P$, gọi C_e ($C_e = 1$ nếu truyền đúng, $C_e = 0$ nếu truyền sai) và C_o ($C_o = 1$ nếu truyền đúng, $C_o = 0$ nếu truyền sai) là kết quả kiểm tra sử dụng hệ thống kiểm tra chẵn và lẻ tương ứng, bảng sau biểu diễn giá trị kiểm tra.

Bảng 6.19. Bảng giá trị kiểm tra chẵn và lẻ với 4 bit nhận

Bản tin 4 bit $D_2D_1D_0P$	Kiểm tra chẵn (C_e)	Bản tin 4 bit $D_2D_1D_0P$	Kiểm tra lẻ (C_o)
0 0 0 0	1	0 0 0 0	0
0 0 0 1	0	0 0 0 1	1
0 0 1 0	0	0 0 1 0	1
0 0 1 1	1	0 0 1 1	0
0 1 0 0	0	0 1 0 0	1
0 1 0 1	1	0 1 0 1	0
0 1 1 0	1	0 1 1 0	0
0 1 1 1	0	0 1 1 1	1

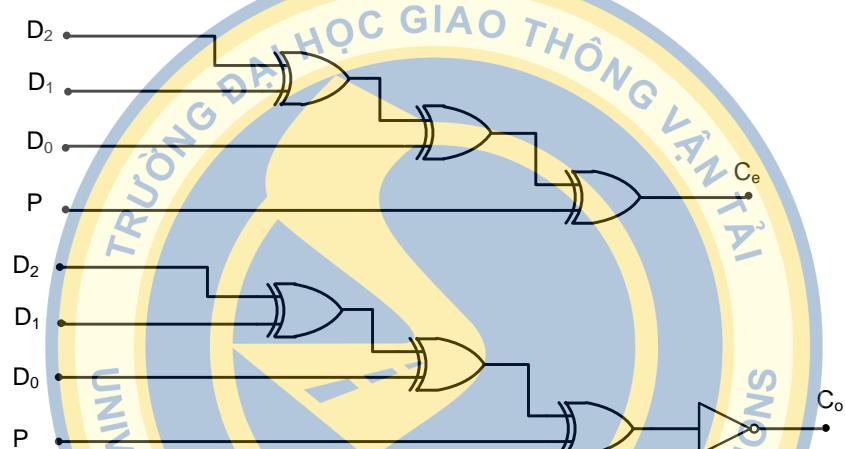
1 0 0 0	0	1 0 0 0	1
1 0 0 1	1	1 0 0 1	0
1 0 1 0	1	1 0 1 0	0
1 0 1 1	0	1 0 1 1	1
1 1 0 0	1	1 1 0 0	0
1 1 0 1	0	1 1 0 1	1
1 1 1 0	0	1 1 1 0	1
1 1 1 1	1	1 1 1 1	0

Từ bảng sự thật, ta có phương trình hàm logic:

$$C_e = D_2 \oplus D_1 \oplus D_0 \oplus P$$

$$C_o = \overline{D_2 \oplus D_1 \oplus D_0 \oplus P}$$

Mạch biểu diễn như sau:



Hình 6.30. Mạch kiểm tra chẵn (hình trên) và lẻ (hình dưới).

Ôn tập chương 6

Lý thuyết

1. Nêu khái niệm mạch mã hóa, mạch giải mã, vẽ sơ đồ khối từng loại mạch.
2. Nêu khái niệm mạch ghép kênh, mạch phân kênh, vẽ sơ đồ khối từng loại mạch.
3. Ưu, nhược điểm của việc thiết kế mạch tổ hợp với mạch ghép kênh và phân kênh?
4. Mạch mã hóa ưu tiên có gì khác biệt so với mạch mã hóa thông thường?

Bài tập

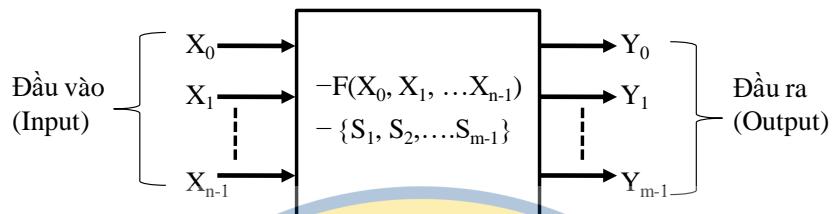
1. Xây dựng mạch cộng nhị phân 8 bit.
2. Xây dựng mạch cộng nhị phân 4 bit.
3. Xây dựng mạch trừ nhị phân 3 bit.
4. Xây dựng mạch cộng nhị phân 3 bit sử dụng NOR.
5. Xây dựng mạch cộng nhị phân 3 bit sử dụng NAND.
6. Xây dựng mạch trừ nhị phân 4 bit.
7. Xây dựng mạch trừ nhị phân 3 bit sử dụng NOR.
8. Xây dựng mạch trừ nhị phân 3 bit sử dụng NAND.
9. Thiết kế mạch so sánh hai số nhị phân 4 bit.
10. Xây dựng mạch giải mã từ mã nhị phân \rightarrow gray 3 bit.
11. Thiết kế mạch giải mã từ mã Gray \rightarrow dư 3, 4 bit.
12. Thiết kế mạch giải mã từ mã Gray \rightarrow nhị phân, 3 bit.
13. Thiết kế mạch giải mã từ mã dư 3, 4 bit \rightarrow gray
14. Thiết kế mạch giải mã từ mã dư 3, 3 bit \rightarrow 7 vạch.
15. Thiết kế mạch giải mã từ mã gray, 3 bit \rightarrow 7 vạch.
16. Thiết kế mạch giải mã từ mã nhị phân, 3 bit \rightarrow 7 vạch.
17. Thiết kế mạch giải mã từ mã nhị phân, 4 bit \rightarrow 7 vạch.
18. Thiết kế mạch giải mã từ mã dư 3, 4 bit \rightarrow 7 vạch.
19. Xây dựng mạch giải mã từ mã gray, 4 bit \rightarrow 7 vạch.
20. Thiết kế mạch giải mã từ mã Johnson \rightarrow dư 3, 4 bit.
21. Sử dụng mạch giải mã, xây dựng mạch có hàm
 - a. $Y = A\bar{B}\bar{C} + AB\bar{C} + \bar{A}\bar{B}\bar{C} + ABC$
 - b. $Y = \sum(2,4,7)$
 - c. $Y = \prod(1,3,6)$
22. Thiết kế mạch MUX 8 đầu vào.
23. Thiết kế mạch DEMUX 8 đầu ra.
24. Thực hiện hàm logic $Y = A\bar{B} + \bar{A}B + AB$ sử dụng MUX.
25. Thiết kế tạo và kiểm tra chẵn với 5 bit truyền đi
26. Thiết kế tạo và kiểm tra lẻ với 5 bit truyền đi

CHƯƠNG 7

MẠCH ĐIỆN TỬ SỐ TUẦN TỰ

7.1. GIỚI THIỆU

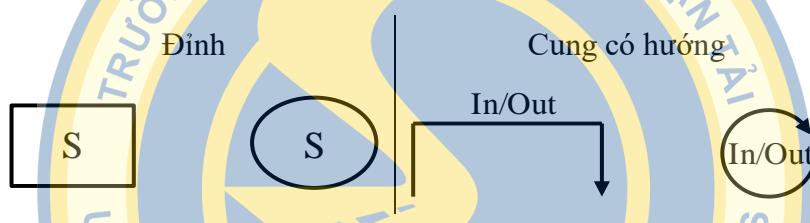
Mạch điện tử số tuần tự (hay còn gọi là mạch logic tuần tự, hay mạch dãy), có đầu ra phụ thuộc vào đầu vào (X_0, X_1, \dots, X_{n-1}) và cả trạng thái bên trong (S_0, S_1, \dots, S_{m-1}) của mạch.



Hình 7.1. Mô hình toán học của mạch tuần tự.

7.1.1. Biểu diễn mạch tuần tự

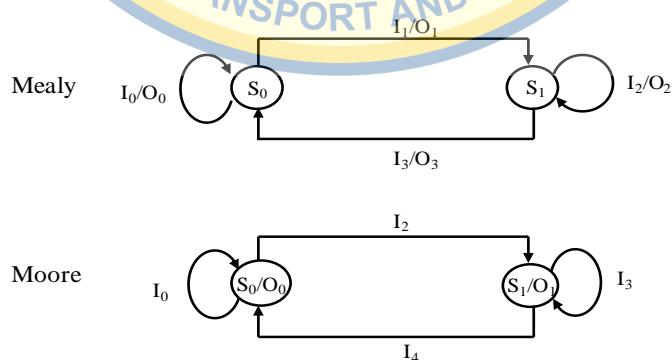
Mạch tuần tự thường được biểu diễn bằng đồ hình trạng thái:



– Theo mô hình Mealy: Tín hiệu ra phụ thuộc vào tín hiệu vào và trạng thái bên trong, khi đó:

- + Đỉnh biểu diễn trạng thái trong.
- + Cung biểu diễn tín hiệu vào/ra: trên cung đi từ đỉnh $S_i \rightarrow S_j$ ghi tín hiệu vào/ra tương ứng.

Ví dụ 7.1:



Hình 7.2. Biểu diễn mạch tuần tự theo mô hình Mealy và Moore.

- Theo mô hình Moore: Tín hiệu ra chỉ phụ thuộc vào trạng thái trong của mạch:
 - + Đinh trạng thái trong/tín hiệu ra.
 - + Cung biểu diễn tín hiệu vào.

7.1.2. Phân loại mạch tuần tự

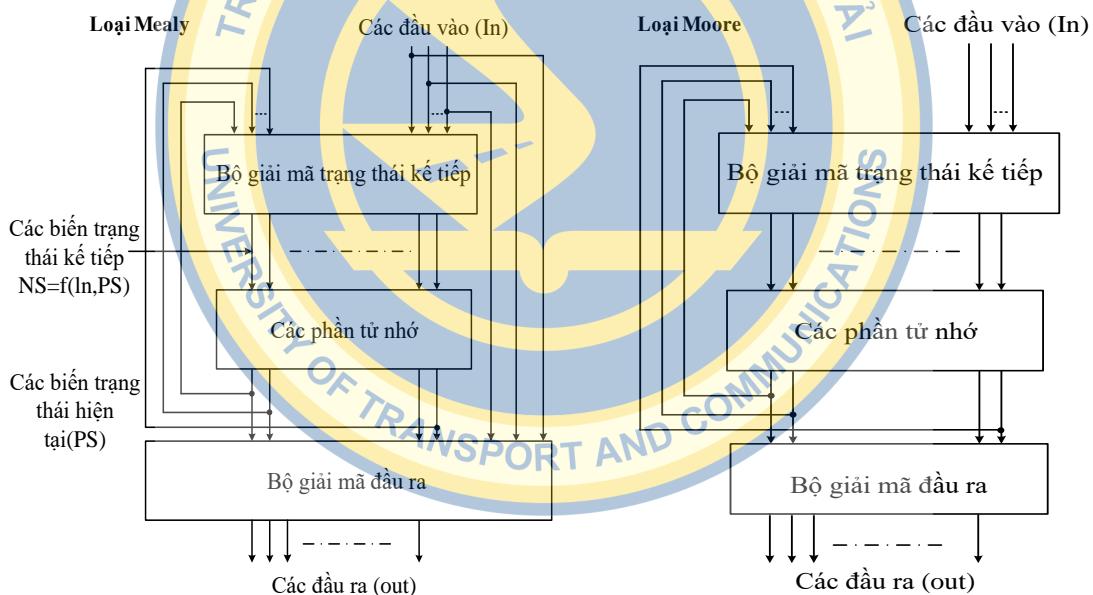
- Theo cách làm việc:

Mạch tuần tự đồng bộ: xung nhịp clk được đưa vào các phần tử để điều khiển quá trình hoạt động, mạch chỉ hoạt động ứng với trạng thái tích cực của xung nhịp.

Mạch tuần tự không đồng bộ: mạch hoạt động không có xung nhịp clk, trạng thái trong của mạch thay đổi ở thời điểm bất kỳ khi tín hiệu vào thay đổi.

- Theo đồ hình trạng thái (còn gọi là mô hình máy, hình 7.3):

- + Theo mô hình máy Mealy: Đây là loại mạch tuần tự đầy đủ. Đặc tính cơ bản của mô hình máy Mealy có đầu ra là hàm của đầu vào và trạng thái hiện tại của mạch. Trạng thái hiện tại của mạch được lưu trữ trong phần tử nhớ. Các biến trạng thái kế tiếp sẽ trở thành biến trạng thái hiện tại (được lưu trong bộ nhớ).
- + Theo mô hình máy Moore: Loại mạch này có đầu ra là hàm trạng thái hiện tại.

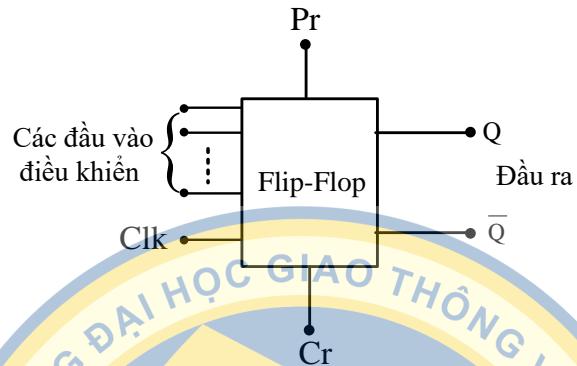


Hình 7.3. Sơ đồ cấu tạo mạch tuần tự Mealy và Moore.

7.2. FLIP-FLOP

Mạch Flip-Flop, ký hiệu là FF là một phần tử cơ bản của mạch số, có chức năng lưu trữ giá trị nhị phân. Mạch FF cùng với các công nghệ cơ bản dùng để xây dựng mạch dãy, vì vậy tìm hiểu về FF cần thiết cho phân tích và thiết kế mạch dãy. Sơ đồ tổng quan Flip-Flop biểu diễn như hình 7.4, bao gồm:

- + Một hay nhiều đầu vào điều khiển.
 - + Đầu vào đồng bộ Clock, ký hiệu Clk.
 - + Đầu vào thiết lập Pre-set, ký hiệu Pr.
 - + Đầu vào xóa Clear, ký hiệu Cr.
- + Hai đầu ra là trạng thái chính Q và trạng thái đảo \bar{Q} . Khi đầu ra Q là 1 hoặc 0 thì nó sẽ duy trì ở trạng thái đó cho đến khi xuất hiện kích thích từ đầu vào.



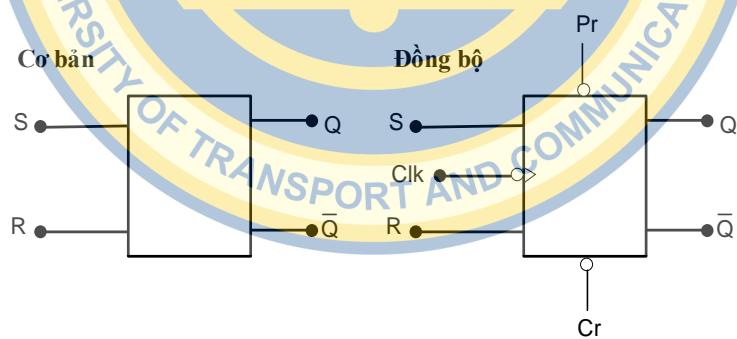
Hình 7.4. Sơ đồ cấu tạo chung của Flip-Flop.

7.2.1. Phân loại

Tùy thuộc vào đầu vào điều khiển và xung nhịp do có chuyển tiếp giữa hai trạng thái, flip-flop được phân thành 4 loại SR-FF, JK-FF, D-FF, T-FF.

Flip-Flop có thể được xây dựng từ các cổng logic cơ bản như NAND và NOR. Ở các trình bày sau đây, chúng tôi sử dụng cổng NAND để xây dựng các mạch FF.

7.2.2. S-R Flip-Flop



Hình 7.5. Ký hiệu SR-FF cơ bản và đầy đủ.

RS-FF cơ bản:

SR-FF là mạch FF có hai đầu vào điều khiển là Set (S) và Reset (R), Sơ đồ khói như hình 7.5. Gọi trạng thái hiện tại của FF là Q_n , và trạng thái tiếp theo là Q_{n+1} , Bảng 7.1 thể hiện mối quan hệ của đầu ra, đầu vào và các trạng thái của SR-FF cơ bản (không đồng bộ).

Bảng 7.1. Bảng sự thật của RS-FF.

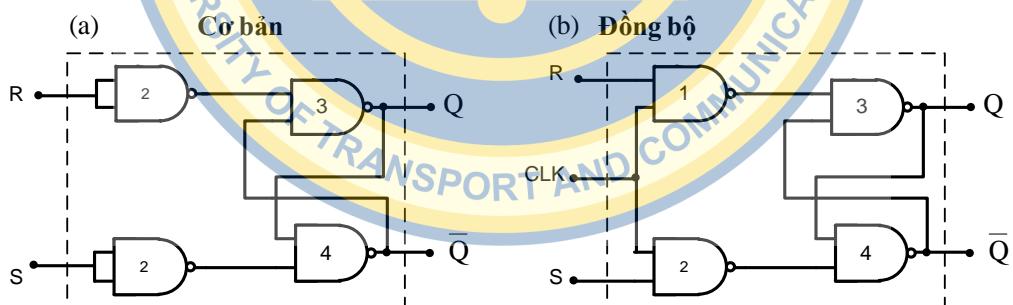
Đầu vào flip-flop		Đầu ra hiện tại	Đầu ra kế tiếp	Hoạt động
S	R	Q_n	Q_{n+1}	
0	0	0	0	Nhớ
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	1	Xóa
1	0	1	1	
1	1	0	X	
1	1	1	X	Cảm

		Q_{n+1}	RQ_n		
S	Q_n	00	01	11	10
0	0	0	1	0	0
1	1	1	1	X	X

Từ bảng Karnaugh trên, ta xác định được phương trình xác định trạng thái ra trong RS-FF:

$$Q_{n+1} = S + \overline{R}Q_n \quad (7.1)$$

SR-FF cơ bản cấu tạo từ cổng NAND có sơ đồ như hình 7.6 (a), có thể thấy là mạch hoàn toàn thỏa mãn quan hệ logic ở bảng 7.1.



Hình 7.6. SR-FF (a) cơ bản và (b) đồng bộ cấu tạo từ cổng NAND.

RS-FF đồng bộ

Hình 7.6 (b) là sơ đồ mạch RS-FF đồng bộ. Ở chế độ này, mạch sẽ không hoạt động (giữ nguyên trạng thái) khi xung nhịp không tích cực (logic 0), và sẽ hoạt động như SR-FF cơ bản nêu trên, khi xung nhịp tích cực (logic 1), và mỗi trạng thái thay đổi tương ứng với 1 xung. Cụ thể như sau:

+Trường hợp 1: Khi xung nhịp không tích cực (logic 0) hay $S = R = 0$, đầu ra của flip-flop duy trì ở trạng thái hiện tại, tức là $Q_n = Q_{n+1}$, như vậy FF đang ở trạng thái lưu dữ liệu.

+Trường hợp 2: Có xung nhịp ($Clk = 1$), và nếu $S = 0$, $R = 1$, đầu ra cổng NAND1 sẽ là 1, và đầu ra cổng NAND 2 sẽ là 0. Do đó đầu ra cổng NAND 4 sẽ là 1, $\bar{Q} = 1$, mức 1 này được đưa đến đầu vào cổng NAND 3 do đó cả hai đầu vào cổng NAND 3 đều là 1, đầu ra cổng NAND 3 sẽ là 0, $Q = 0$.

+Trường hợp 3: Có xung nhịp ($Clk = 1$), và $S = 1$, $R = 0$, đầu ra cổng NAND 2 sẽ là 1, và đầu ra cổng NAND 1 sẽ là 0. Mức 0 này sẽ làm cho đầu ra $Q = 1$. Khi đó đầu vào cổng NAND 4 có hai đầu vào ở mức 1 nên đầu ra $\bar{Q} = 0$.

+Trường hợp 4: Có xung nhịp ($Clk = 1$), và $S = 1$, $R = 1$, đầu ra hai cổng NAND 1 và NAND 2 đều là 0. Đầu vào cổng NAND 3 và 4 đều là 0 nên đầu ra hai cổng là 1, $Q = 1$ và $\bar{Q} = 1$, vi phạm quy luật logic của FF là hai đầu ra đảo nhau, vì vậy trường hợp này là cấm.

Bảng 7.2 tóm hợp lại các trường hợp của RS-FF khi có xung nhịp.

Bảng 7.2. Bảng sự thật của RS-FF.

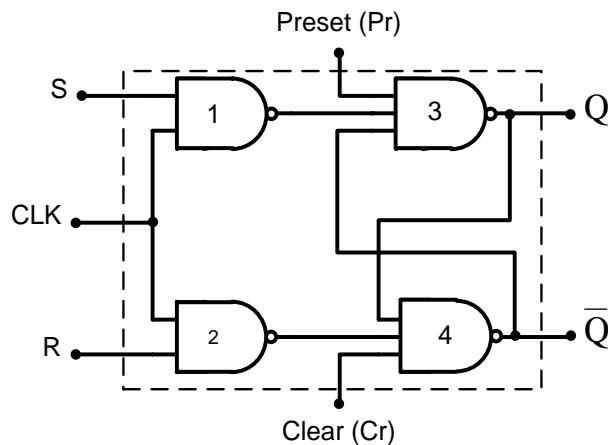
Đầu vào flip-flop		Đầu ra hiện tại		Đầu ra kế tiếp		Hoạt động
Clk	S	R	Q_n	Q_{n+1}	\bar{Q}_{n+1}	
0	X	X	0	0	1	Nhớ
	X	X	1	1	0	
	0	0	0	0	1	
	0	0	1	1	0	
	0	1	0	0	1	Xóa
	0	1	1	0	1	
	1	0	0	1	0	
	1	0	1	1	0	Lập trình
1	1	1	0	X	X	
	1	1	1	X	X	Cấm

RS-FF đặt trước giá trị

Trong thực tế nhiều ứng dụng yêu cầu khởi tạo ở giá trị 0 hay 1. Điều này được thực hiện bởi các đầu Preset (Pr, làm cho $Q = 0$) và Clear (Cr, làm cho $Q = 1$). Các đầu vào này có thể đưa vào ở thời điểm bất kỳ giữa các xung nhịp và không đồng bộ với xung nhịp. SR-FF với Pr và Cr sử dụng NAND được xây dựng ở hình 7.7:

+ Đặt logic 0: thì $Pr = 1$ và $Cr = 0$, đầu ra cổng NAND 4 sẽ là 1, $\bar{Q} = 1$, tức là $Q = 0$, FF bị xóa, hay đặt logic 0.

+ Đặt logic 1: thì $Pr = 0$ và $Cr = 1$, đầu ra cổng NAND 3 sẽ là 1, $Q = 1$, tức là FF được đặt logic 1.

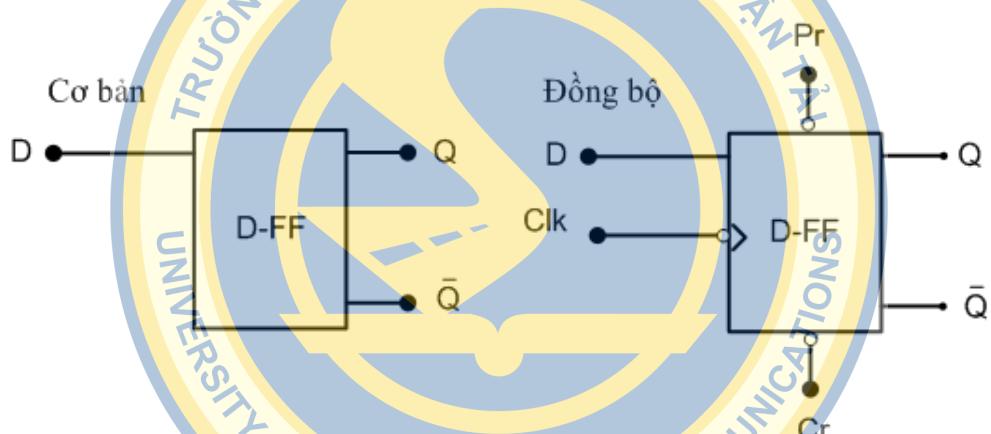


Hình 7.7. SR-FF với Preset và Clear.

7.2.3. D FLIP-FLOP

D-FF cơ bản:

D flip-flop chỉ có một đầu vào D (delay) hay đầu vào dữ liệu. Dữ liệu ở đầu vào được truyền đến đầu ra sau trễ một xung nhịp.



Hình 7.8. Ký hiệu D-FF cơ bản và đầy đủ.

Bảng sự thật của D-FF

Bảng 7.3. Bảng sự thật của D-FF.

Các đầu vào FF	Đầu ra hiện tại	Đầu ra kế tiếp	Hoạt động
D	Q_n	Q_{n+1}	
0	0	0	Nhớ
0	1	0	Xóa
1	0	1	Lập trình
1	1	1	Nhớ

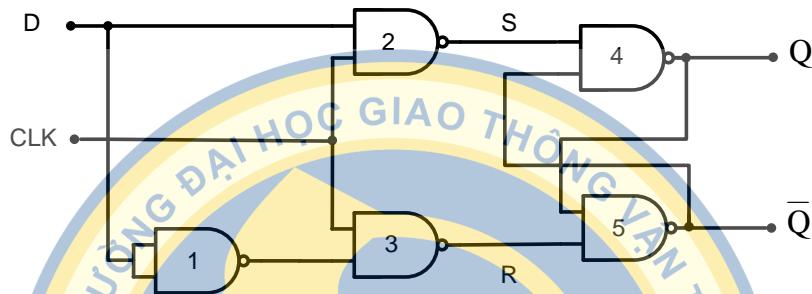
	Q_n	0	1
0	0	0	
1		1	1

Ta xác định được phương trình đặc tính của D-FF theo bảng Karnaugh

$$Q_{n+1} = D$$

D-FF đồng bộ

D-FF có thể được xây dựng từ SR-FF bằng cách kết hợp một bộ đảo giữa S và R. Sơ đồ cấu trúc D-FF như hình 7.9.



Hình 7.9. D-FF sử dụng cổng NAND.

Bảng 7.4. Bảng sự thật của RS-FF khi có Clk.

Đầu vào		Đầu ra hiện tại	Đầu ra kế tiếp	Hoạt động
Clk	D	Q_n	Q_{n+1}	
0	X	0	0	Nhớ
	X	1	1	
1	0	0	0	Nhớ
	0	1	0	Xóa
	1	0	1	Lập trình
	1	1	1	Nhớ

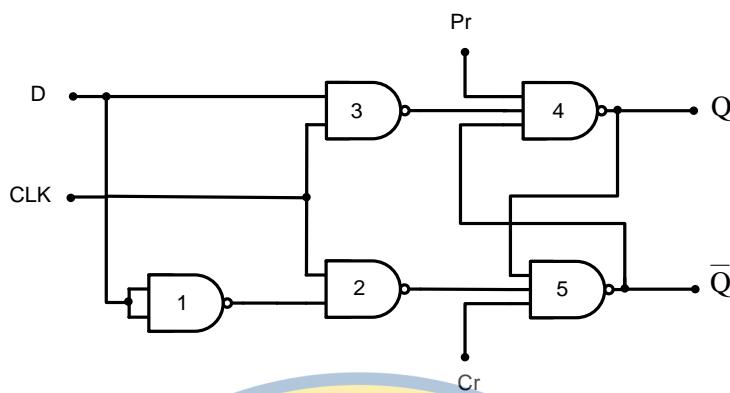
+Trường hợp 1: Khi đầu vào CLK = 0, giá trị của đầu vào D không ảnh hưởng.

+Trường hợp 2: Khi CLK = 1, và D = 1, cả hai đầu vào cổng NAND 2 là 1 nên đầu ra cổng NAND 2 sẽ là 0. Đầu ra cổng NAND 4, tức là $Q = 1$. Mặt khác, đầu ra cổng NAND 1 sẽ là 0, làm đầu ra cổng NAND 3 là 1, làm $\bar{Q} = 0$.

+Trường hợp 3: Nếu CLK = 1, D = 0, nên đầu ra cổng NAND 3 sẽ là 0, đầu ra cổng NAND 5 sẽ là 1, $\bar{Q} = 1$. Và đầu ra cổng NAND 2 sẽ là 1. Cổng NAND 4 có hai đầu vào là 1 nên đầu ra $Q = 0$.

D-FF đặt trước giá trị

Tương tự RS-FF, sơ đồ D-FF đặt trước giá trị có sơ đồ như hình 7.10



Hình 7.10. D-FF với Preset và Clear.

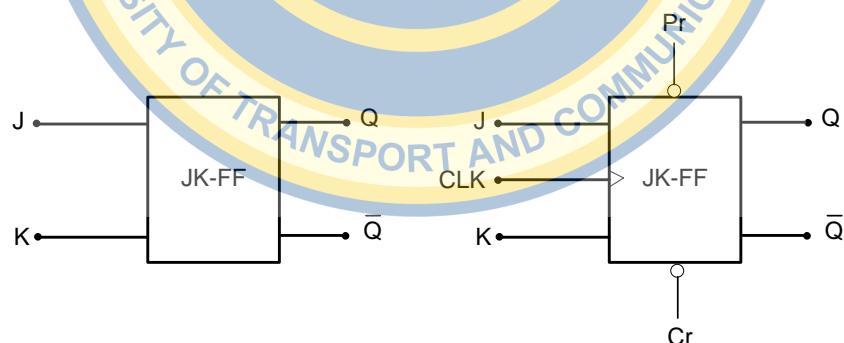
+ Đặt logic 0: thì $Pr = 1$ và $Cr = 0$, đầu ra cổng NAND 5 sẽ là 1, $\bar{Q} = 1$, tức là $Q = 0$, FF bị xóa, hay đặt logic 0.

+ Đặt logic 1: thì $Pr = 0$ và $Cr = 1$, đầu ra cổng NAND 4 sẽ là 1, $Q = 1$, tức là FF được đặt logic 1.

7.2.4. JK FLIP-FLOP

JK-FF cơ bản:

JK-FF có đặc tính giống SR-FF. Các đầu vào J và K cũng giống như đầu vào S và R. Chỉ có một sự khác biệt ở điều kiện chưa xác định của SR-FF. Khi $J=K=1$ thì FF sẽ lật trạng thái, nghĩa là đầu ra sẽ trạng thái đảo ở mọi thời điểm xung nhịp đi qua.



Hình 7.11. Ký hiệu JK-FF cơ bản và đầy đủ.

Mỗi quan hệ của tín hiệu ra với tín hiệu vào và trạng thái hiện tại thể hiện ở bảng 7.5.

Bảng 7.5. Bảng sự thật JK-FF.

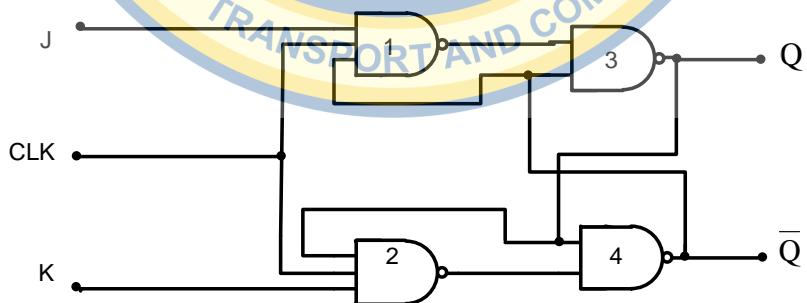
Đầu vào Flip-Flop		Đầu ra hiện tại	Đầu ra kế tiếp	Hoạt động
J	K	Q_n	Q_{n+1}	
0	0	0	0	Nhớ
0	0	1	1	
0	1	0	0	Xóa
0	1	1	0	
1	0	0	1	Lập trình
1	0	1	1	
1	1	0	1	
1	1	1	0	Xóa

		KQ_n	00	01	11	10
J	0	0	1	0	0	
1	1	1	1	0	1	

Từ bảng Karnaugh, ta được phương trình đặc tính của JK-FF:

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

JK-FF đồng bộ



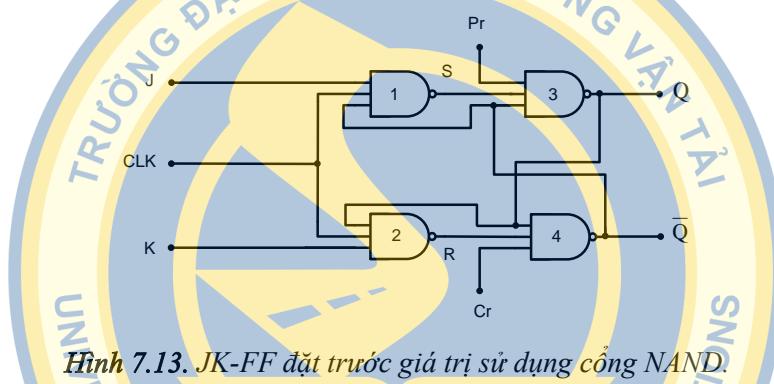
Hình 7.12. JK-FF đồng bộ sử dụng cổng NAND.

Tương tự, sau khi phân tích các trường hợp ta có bảng sự thật của JK-FF khi có xung đồng bộ

Bảng 7.6. Bảng sự thật JK-FF.

Đầu vào			Đầu ra hiện tại	Đầu ra kế tiếp	Hoạt động
Clk	J	K	Q_n	Q_{n+1}	
0	X	X	0	0	Nhớ
	X	X	1	1	
1	0	0	0	0	Xóa
	0	0	1	1	
	0	1	0	0	Lập trình
	0	1	1	0	
1	1	0	0	1	Xóa
1	1	0	1	1	
1	1	1	0	1	Xóa
1	1	1	1	0	Xóa

JK-FF đặt trước giá trị; hình 7.13



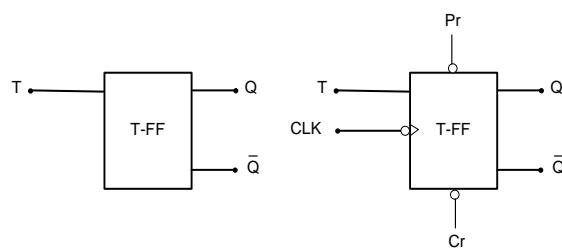
Hình 7.13. JK-FF đặt trước giá trị sử dụng cổng NAND.

+ Đặt logic 0: thì $Pr = 1$ và $Cr = 0$, đầu ra cổng NAND 4 sẽ là 1, $\bar{Q} = 1$, tức là $Q = 0$, FF bị xóa, hay đặt logic 0.

+ Đặt logic 1: thì $Pr = 0$ và $Cr = 1$, đầu ra cổng NAND 3 sẽ là 1, $Q = 1$, tức là FF được đặt logic 1.

7.2.5. T FLIP-FLOP

T-FF chỉ có một đầu vào, nó chỉ có hai trạng thái là trạng thái lật và trạng thái nhở. Do đó, T-FF thường được sử dụng để thiết kế bộ đếm và mạch dãy.

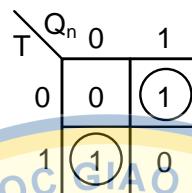


Hình 7.14. Ký hiệu T-FF cơ bản và đầy đủ.

Bảng sự thật của T-FF.

Bảng 7.7. Bảng sự thật T-FF.

Đầu vào	Đầu ra hiện tại	Đầu ra kế tiếp	Hoạt động
T	Q _n	Q _{n+1}	
0	0	0	Nhớ
0	1	1	
1	0	1	Lập trình
1	1	0	Xóa



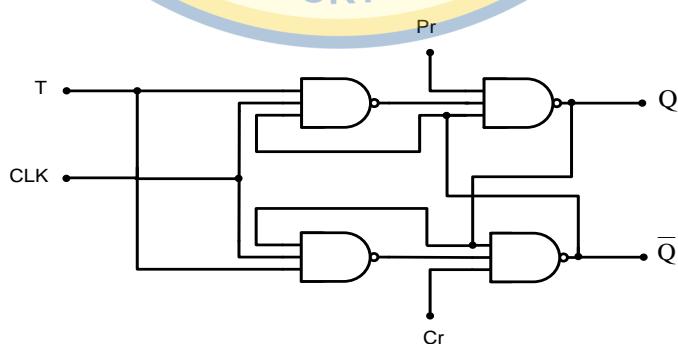
Sử dụng bảng Karnaugh, ta xác định được phương trình đặc tính:

$$Q_{n+1} = T\bar{Q}_n + \bar{T}Q_n$$

T-FF đồng bộ

Bảng 7.8. Bảng sự thật của T-FF trong chế độ đồng bộ.

Đầu vào	Đầu ra hiện tại	Đầu ra kế tiếp	Hoạt động	
Clk	T	Q _n		
0	X	0	0	Nhớ
	X	1	1	
1	0	0	0	Lập trình
	0	1	1	
	1	0	1	Xóa
	1	1	0	

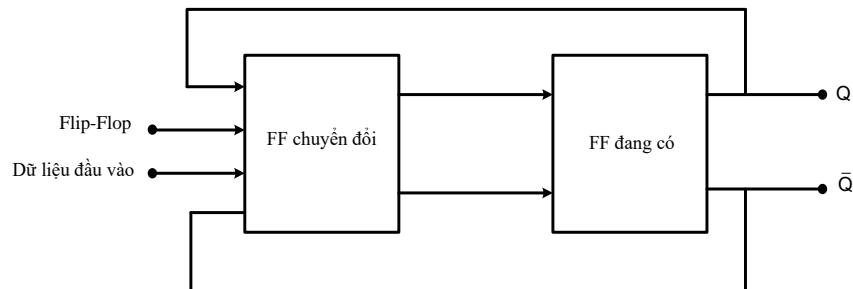


Hình 7.15. Sơ đồ mạch T-FF đồng bộ sử dụng NAND.

Tương tự với thiết kế và phân tích mạch đã trình bày với các FF khác, ta có sơ đồ mạch T-FF sử dụng NAND như hình 7.15, và quan hệ logic đầu vào ra như bảng 7.8.

7.2.6. CHUYỂN ĐỔI GIỮA CÁC FLIP-FLOP

Mô hình chuyển đổi các flip-flop biểu diễn như hình 7.16.



Hình 7.16. Mô hình chuyển đổi giữa các Flip-Flop.

Để thực hiện thiết kế mạch chuyển đổi cần kết hợp bảng chuyển đổi trạng thái, flip-flop, để lập bảng sự thật, dữ liệu đầu vào, dữ liệu đầu ra.

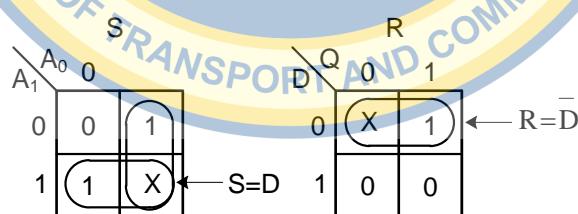
- Chuyển đổi SR-FF sang D-FF

Từ bảng chuyển đổi giữa các flip-flop ở bảng 7.11 của SR-FF và D-FF ta lập bảng sự thật như bảng 7.9.

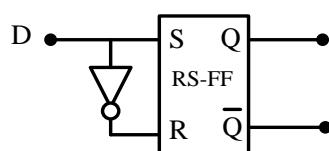
Bảng 7.9

Dữ liệu đầu vào FF	Đầu ra	Đầu vào SR-FF	
D	Q	S	R
0	0	0	X
1	0	1	0
0	1	0	1
1	1	X	0

Từ bảng trên ta lập bảng Karnaugh với đầu vào S và R như sau:



Từ đó ta vẽ mạch:



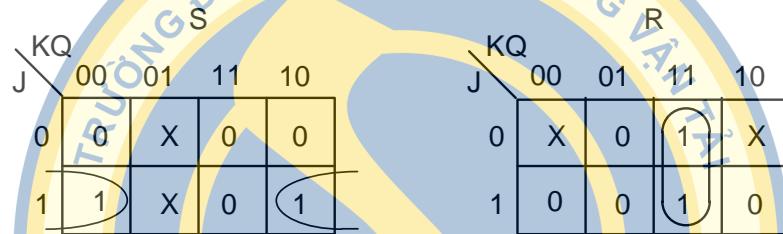
Hình 7.17. Chuyển SR-FF sang D-FF.

- Chuyển SR-FF sang JK-FF

Từ bảng chuyển đổi giữa các flip-flop bảng 7.11 của SR-FF và JK-FF ta lập bảng sự thật như bảng 7.10.

Bảng 7.10

Đầu vào FF		Đầu ra	Đầu vào SR-FF	
J	K	Q	S	R
0	0	0	0	X
0	1	0	0	X
1	0	0	1	0
1	1	0	1	0
0	1	1	0	1
1	1	1	0	1
0	0	1	X	0
1	0	1	X	0

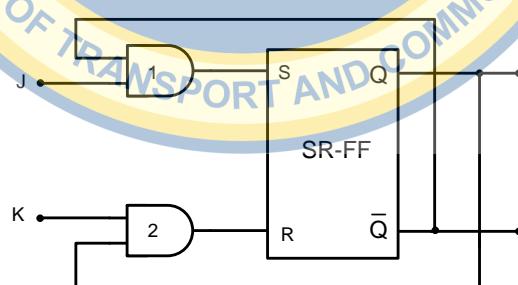


Từ bảng trên ta lập bảng Karnaugh xác định S và R, sơ đồ mạch như hình 7.19

$$S = J\bar{Q}$$

$$R = KQ$$

Từ đó ta vẽ mạch:



Hình 7.18. Sơ đồ mạch SR-FF sang JK-FF.

7.2.7. KÍCH HOẠT FLIP-FLOP

Flip-Flop là mạch làm việc với cơ chế đồng bộ. Có thể phân thành hai loại: kích hoạt mức và kích hoạt sườn xung.

7.2.7.1. Kích hoạt mức của Flip-Flop

Nếu flip-flop là Enable (cho phép) khi xung nhịp đưa vào là cao (High), và duy trì trong khoảng thời gian xung nhịp là cao, khi đó flip-flop là kích hoạt mức. Nếu flip-flop thay đổi trạng thái khi xung nhịp là dương thì flip-flop được kích hoạt mức dương, flip-flop thay đổi trạng thái khi xung nhịp là âm thì flip-flop được kích hoạt mức âm.

Hạn chế của kích hoạt mức là khi xung nhịp được kích hoạt, flip-flop thay đổi trạng thái một hay nhiều lần khi đầu vào thay đổi. Nếu các đầu vào không thay đổi thì đầu ra duy trì ổn định. Điều này có thể khắc phục bằng cách sử dụng master-slave flip-flop hoặc flip-flop kích hoạt bằng sườn xung.

7.2.7.2. Kích hoạt sườn xung của Flip-Flop

Hình 7.19 biểu diễn hai chuyển tiếp là sườn dương ($0 \rightarrow 1$) và sườn âm ($1 \rightarrow 0$). Việc flip-flop thay đổi trạng thái ứng với sườn dương hoặc sườn âm của xung nhịp được gọi là kích hoạt sườn xung.



Hình 7.19. Xung nhịp kích hoạt mạch.

7.2.8. Bảng kích hoạt của Flip-Flop

Từ bảng sự thật của flip-flop cho thấy nguyên lý hoạt động của flip-flop. Tuy nhiên trong thiết kế mạch dãy, cần xác định giá trị kích hoạt cho FF để được đầu ra mong muốn. Ví dụ, đầu ra SR-FF trước khi xuất hiện xung nhịp là $Q_n = 1$ và cần có đầu ra không thay đổi khi có xung nhịp.

Từ bảng đặc trưng của SR-FF, ta thấy điều kiện:

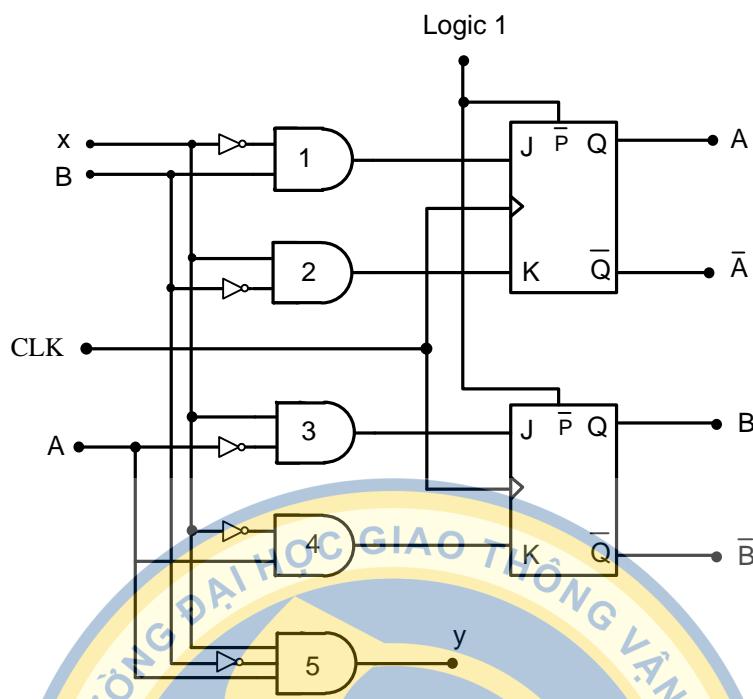
1. $S = R = 0$
2. $S = 1$ và $R = 0$

Bảng kích hoạt các flip-flop tổng hợp tại bảng 7.11.

Bảng 7.11. Bảng kích hoạt cho các flip-flop.

Trạng thái hiện tại (Q_n)	Trạng thái kế tiếp (Q_{n+1})	SR-FF		D-FF	JK-FF		T-FF
		S	R	D	J	K	T
0	0	0	X	0	0	X	0
0	1	1	0	1	1	X	1
1	0	0	1	0	X	1	1
1	1	X	0	1	X	0	0

7.3. PHÂN TÍCH MẠCH TUẦN TỰ



Hình 7.20. Ví dụ mạch tuần tự đồng bộ.

Hoạt động của mạch tuần tự được xác định bởi đầu vào, đầu ra, trạng thái của flip-flop. Đầu ra và trạng thái kế tiếp là hàm của đầu vào và trạng thái hiện tại. Phân tích mạch tuần tự bao gồm lập bảng hoặc đồ hình trạng thái của đầu vào, đầu ra, trạng thái trong. Viết phương trình Boolean mô tả hoạt động của mạch. Minh họa một mạch tuần tự đồng bộ như hình 7.20:

– Bảng trạng thái

Từ đầu vào, đầu ra và trạng thái của flip-flop có thể biểu diễn vào bảng trạng thái. Bảng trạng thái của mạch hình 7.20 được biểu diễn như bảng 7.12

Bảng 7.12. Bảng trạng thái.

Trạng thái hiện tại	Trạng thái kế tiếp		Đầu ra	
	x=0	x=1	x=0	x=1
AB	AB	AB	y	y
00	00	01	0	0
01	11	01	0	0
10	10	00	0	1
11	10	11	0	0

Phân tích mạch có thể bắt đầu từ trạng thái bất kỳ. Trong ví dụ này, ta phân tích trạng thái ban đầu là 00. Khi đó A=0, B = 0, từ sơ đồ khói, với x = 0, ta thấy cổng AND 1 và AND 2 tạo tín hiệu logic 0, do đó trạng thái kế tiếp không đổi. Ngoài ra, B-FF với cổng AND 3 và AND 4 cũng tạo tín hiệu logic 0 nên trạng thái kế tiếp không đổi. Khi có xung nhịp A-FF và B-FF ở trạng thái nhớ, trạng thái kế tiếp là 00. Tương tự, với A=0, B=0, với x=1, ta thấy cổng AND 1 tạo mức logic 0, cổng AND 2 tạo mức logic 1. Cùng điều kiện đó, cổng AND 3 tạo mức logic 0, cổng AND 4 tạo mức logic 1. Do đó, khi có xung nhịp, A-FF ở trạng thái xóa (Clear) và B-FF ở trạng thái thiết lập (set), tạo trạng thái kế tiếp là 01.

Tương tự, ta có các điều kiện khác nhau của bảng trạng thái. Khi trạng thái hiện tại là 01, A=0 và B=1. Từ sơ đồ khói, với x=0, ta thấy cổng 1 tạo mức logic 1 và cổng logic 2 tạo mức logic 0. Với B-FF, cả hai cổng 3 và 4 đều tạo mức logic 0. Do đó, trạng thái kế tiếp của B-FF không đổi. Khi có xung nhịp, A-FF ở trạng thái thiết lập (set), B-FF duy trì ở trạng thái nhớ, tạo trạng thái kế tiếp sẽ là 11. Tương tự, với A=0, B=1, với x=1, ta thấy cổng 1 và cổng 2 tạo mức logic 0. Cùng điều kiện đó, cổng 3 và cổng 4 tạo mức logic 0. Do đó khi có xung nhịp, cả A và B-FF duy trì ở trạng thái nhớ, tạo trạng thái kế tiếp là 01.

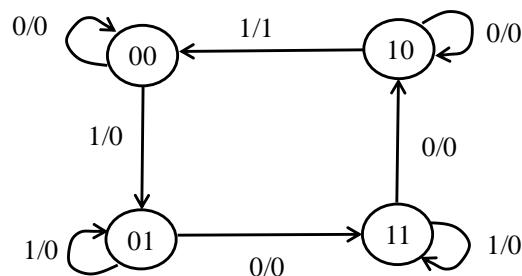
Khi trạng thái hiện tại là 10, A=1,B=0. Từ sơ đồ logic, với x=0 ta thấy cổng 1 và cổng 2 đều tạo mức logic 0. Với B-FF cổng 3 tạo mức logic 0 và cổng 4 tạo mức logic 1. Khi có xung nhịp, A-FF duy trì ở trạng thái nhớ và B là xóa (reset), tạo trạng thái kế tiếp là 10. Tương tự, với A=1, B=0, x=1, ta thấy cổng 1 tạo mức logic 0, cổng 2 tạo mức logic 1. Cùng điều kiện đó, cả cổng 3 và 4 đều tạo mức logic 0. Khi có xung nhịp A sẽ là xóa (reset) và B duy trì ở trạng thái nhớ, tạo trạng thái kế tiếp là 00.

Khi trạng thái hiện tại là 11, A=1, B=1. Từ sơ đồ logic, với x=0, ta thấy cổng 1 tạo mức logic 1 và cổng 2 tạo mức logic 0. Với B-FF, cổng 3 tạo mức logic 0 và cổng 4 tạo mức logic 1. Khi có xung nhịp, A-FF duy trì ở trạng thái nhớ, B sẽ là xóa (reset), tạo trạng thái kế tiếp là 10. Tương tự, với A=1, B=1, x=1, ta thấy cả cổng 1 và cổng 2 đều tạo mức logic 0. Cùng điều kiện đó, cả cổng 3 và cổng 4 tạo mức logic 0. Khi có xung nhịp, cả A-FF và B-FF duy trì ở trạng thái nhớ, tạo trạng thái kế tiếp là 11.

Tóm lại, một mạch tuần tự m-FF và n biến đầu vào sẽ có 2^m trạng thái. Đầu ra và trạng thái kế tiếp sẽ là 2^n tổ hợp.

- Sơ đồ trạng thái

Tất cả thông tin trong bảng trạng thái có thể biểu diễn dạng đồ hình trạng thái như hình 7.21.



Hình 7.21. Sơ đồ trạng thái của mạch hình 7.20.

– Phương trình trạng thái

Phương trình trạng thái là biểu thức đại số mô tả các điều kiện chuyển trạng thái của flip-flop. Phương trình này được xây dựng từ bảng trạng thái. Ví dụ, phương trình trạng thái A-FF xây dựng từ bảng 7.12. Từ cột trạng thái kế tiếp ta thấy A-FF chuyển đến trạng thái 1 bốn lần: khi $x=0$ và $AB=01$ hoặc 10 hoặc 11 hoặc khi $x=1$ và $AB=11$. Do đó, biểu thức đại số của phương trình trạng thái là:

$$A(t+1) = (\bar{A}\bar{B} + A\bar{B} + AB)\bar{x} + ABx$$

Tương tự, từ cột trạng thái kế tiếp ta thấy B-FF chuyển trạng thái đến trạng thái 1 bốn lần: khi $x=0$ và $AB=01$ hoặc khi $x=1$ và $AB=00$ hoặc 01 hoặc 11. Do đó biểu thức đại số của phương trình trạng thái là:

$$B(t+1) = \bar{A}\bar{B}\bar{x} + (\bar{A}\bar{B} + \bar{A}B + AB)x$$

Đơn giản biểu thức ta được:

$$A(t+1) = (B\bar{x})\bar{A} + (\bar{B}\bar{x})A$$

Nếu $B\bar{x} = J$ và $\bar{B}\bar{x} = K$ ta có: $A(t+1) = J\bar{A} + KA$

Tương tự, $B(t+1) = (\bar{A}x)\bar{B} + (A\bar{x})B$

Nếu $\bar{A}x = J$ và $A\bar{x} = K$ ta có: $B(t+1) = J\bar{B} + KB$

Đây chính là phương trình đặc trưng của JK-FF.

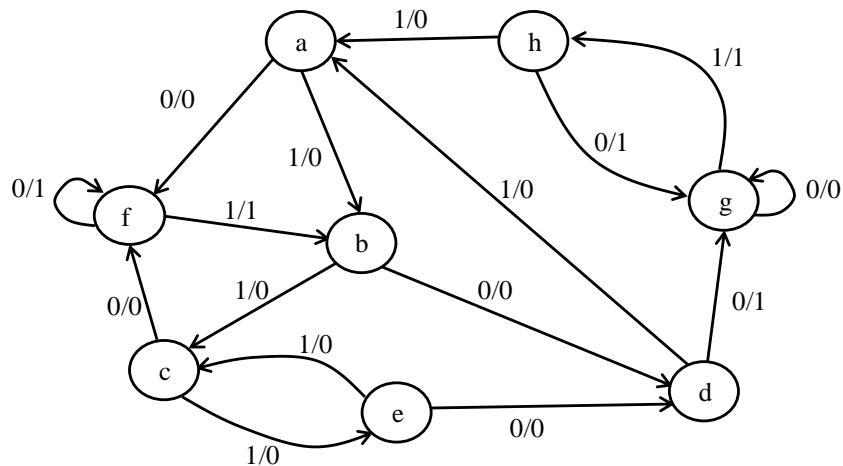
7.4. THIẾT KẾ MẠCH TUẦN TỰ

Các bước thực hiện thiết kế mạch tuần tự như sau:

1. Xây dựng đồ hình trạng thái: bài toán thường được mô tả theo sơ đồ trạng thái, nhưng nếu không thì cần phân tích các quan hệ của bài toán thiết kế, để tìm ra đồ hình trạng thái

2. Từ sơ đồ trạng thái ta lập bảng trạng thái, tối giản số trạng thái.
3. Mã hóa các trạng thái bởi các giá trị nhị phân. Từ đó, xác định số flip-flop, loại flip-flop cần dùng.
4. Từ bảng kích hoạt cho flip-flop, sơ đồ mạch, ta xác định được giá trị kích hoạt cho flip-flop chọn thiết kế mạch.
5. Sử dụng bảng Karnaugh hoặc các phương pháp khác để tối giản phương trình hàm kích cho flip-flop, hàm đầu ra.
6. Vẽ sơ đồ mạch.

Xét một ví dụ: thiết kế mạch tuần tự đồng bộ theo sơ đồ trạng thái hình 7.22



Hình 7.22. Đồ hình trạng thái.

Lập bảng trạng thái như bảng 7.13.

Bảng 7.13. Bảng trạng thái hình 7.22.

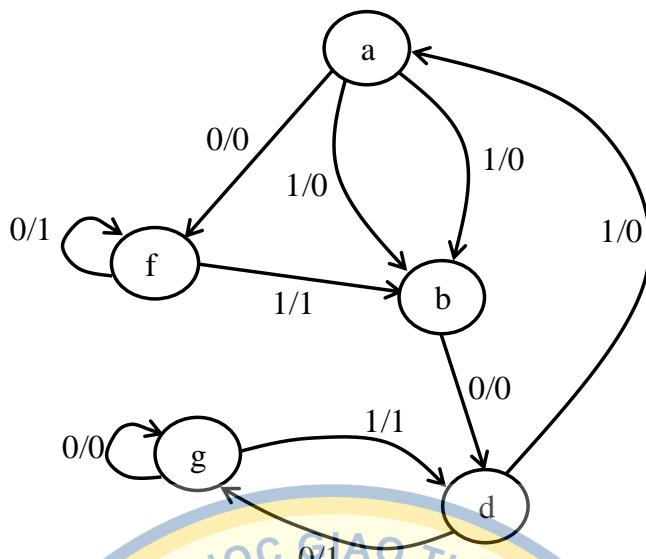
Trạng thái hiện tại	Trạng thái kế tiếp		Đầu ra	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

Từ bảng này, ta phân tích các trạng thái tương đương để có được bảng trạng thái sau khi rút gọn.

Bảng 7.14. Bảng trạng thái rút gọn.

Trạng thái hiện tại	Trạng thái kế tiếp		Đầu ra	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	a	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

Từ đây, ta có sơ đồ trạng thái rút gọn.



Hình 7.23. Sơ đồ trạng thái rút gọn

Tiếp theo ta mã hóa các trạng thái khác nhau theo các giá trị nhị phân. Có thể mã hóa theo nhiều cách khác nhau và ta chọn một trong các mã hóa đó như ở bảng sau.

Bảng 7.15. Bảng mã hóa.

Trạng thái	Mã hóa 1	Mã hóa 2	Mã hóa 3	Mã hóa 4
a	000	001	111	011
b	001	010	001	101
d	010	011	110	111
f	011	100	101	001
g	100	101	010	000

Bảng 7.16. Bảng trạng thái rút gọn với mã hóa 1.

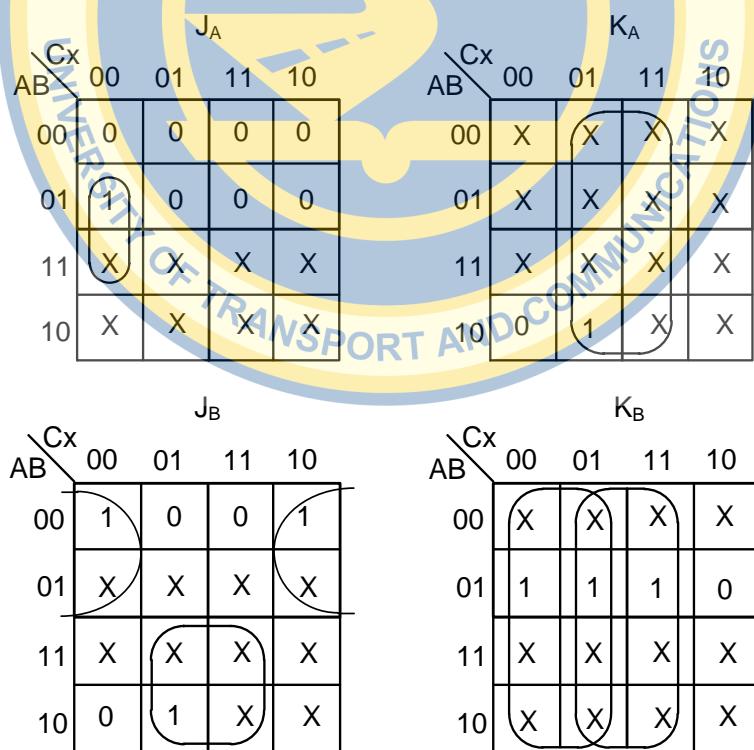
Trạng thái hiện tại	Trạng thái kế tiếp		Đầu ra	
	x = 0	x = 1	x = 0	x = 1
000	011	001	0	0
001	010	000	0	0
010	100	000	1	0
011	011	001	1	1
100	100	010	0	1

Đến đây ta lập bảng để xác định đầu vào, đầu ra của mạch như bảng 7.17

Bảng 7.17. Bảng xác định đầu vào, đầu ra

Trạng thái hiện tại			Đầu vào	Trạng thái kế tiếp			Đầu vào flip-flop						Đầu ra
A	B	C	x	A	B	C	J _A	K _A	J _B	K _B	J _C	K _C	y
0	0	0	0	0	1	1	0	X	1	X	1	X	0
0	0	0	1	0	0	1	0	X	0	X	1	X	0
0	0	1	0	0	1	0	0	X	1	X	X	1	0
0	0	1	1	0	0	0	0	X	0	X	X	1	0
0	1	0	0	1	0	0	1	X	X	1	0	X	1
0	1	0	1	0	0	0	0	X	X	1	0	X	0
0	1	1	0	0	1	1	0	X	X	0	X	0	1
0	1	1	1	0	0	1	0	X	X	1	X	0	1
1	0	0	0	1	0	0	X	0	0	X	0	X	0
1	0	0	1	0	1	0	X	1	1	X	0	X	1

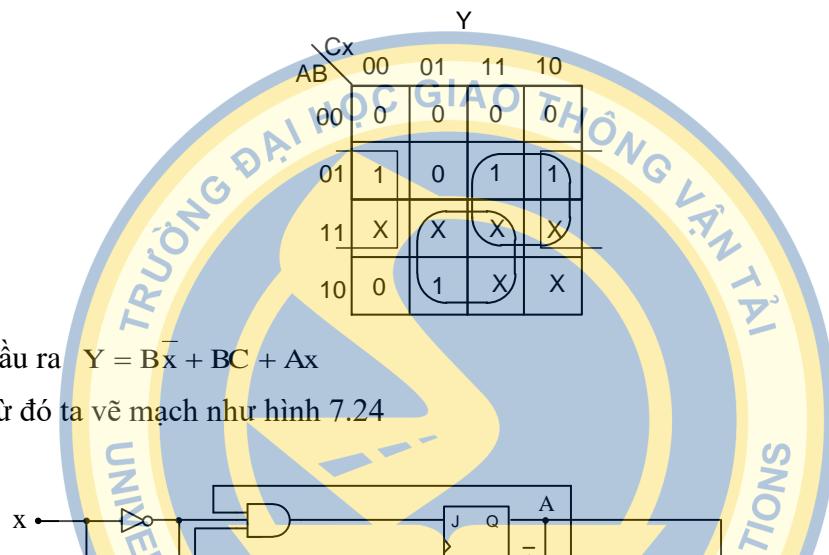
Lập bảng Karnaugh cho J_A, K_A, J_B, K_B, J_C và K_C.



		J _c				K _c					
		AB	00	01	11	10	AB	00	01	11	10
Cx	AB	00	1	1	X	x		X	X	1	1
	01	00	0	0	X	X		X	X	0	0
	11	01	X	X	X	X		X	X	X	X
	10	11	0	0	X	X		X	X	X	X
		10									

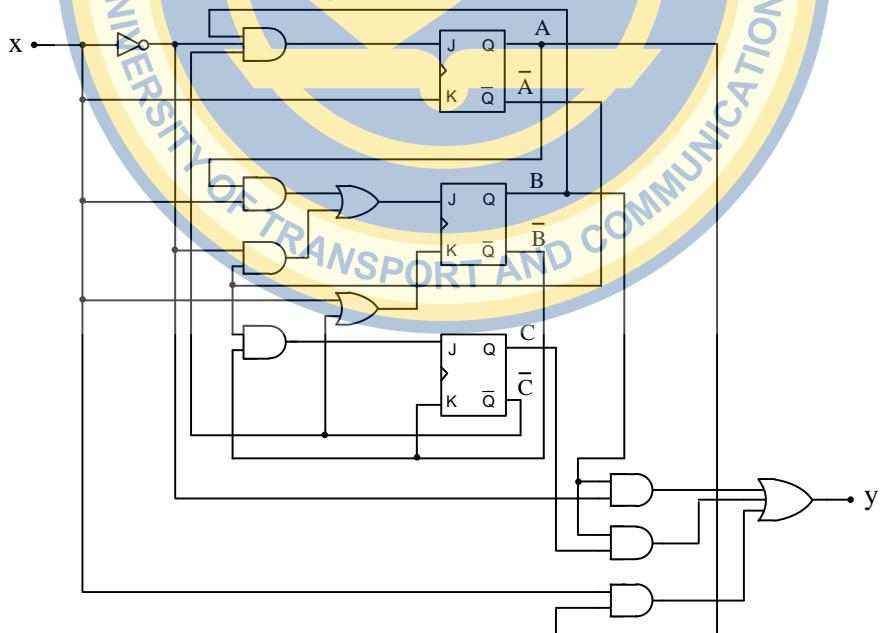
Từ bảng Karnaugh cho J_A, K_A, J_B, K_B, J_C, K_C ta có:

$$J_A = BC\bar{x} \text{ và } K_A = x; J_B = Ax + \bar{A}\bar{x} \text{ và } K_B = \bar{C} + x; J_C = \bar{A}\bar{B} \text{ và } K_C = \bar{B}$$



Đầu ra $Y = B\bar{x} + BC + Ax$

Từ đó ta vẽ mạch như hình 7.24



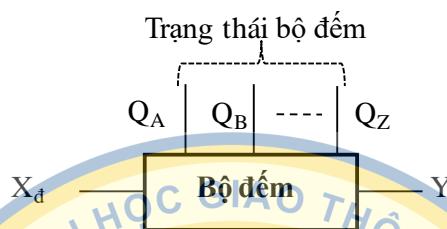
Hình 7.24. Sơ đồ mạch logic.

7.5. BỘ ĐÉM

7.5.1. GIỚI THIỆU

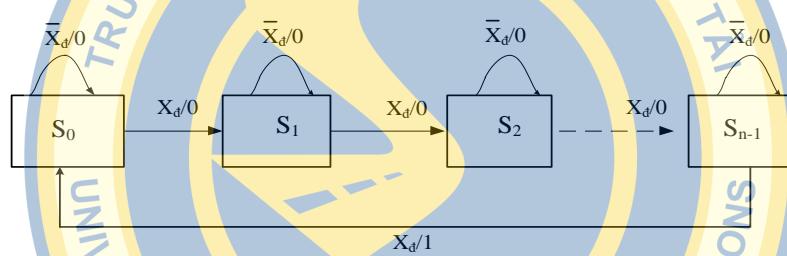
Bộ đếm là mạch tuần tự hoàn có một đầu vào đếm và một đầu ra, mạch có số trạng thái trong bằng hệ số đếm (K_d). Khi có tín hiệu vào đếm, mạch sẽ chuyển từ trạng thái trong này đến trạng thái trong khác theo một trình tự nhất định. Cứ sau K_d tín hiệu vào đếm, mạch trở về trạng thái ban đầu.

Bộ đếm thường được xây dựng từ một hay nhiều flip-flop mà trạng thái thay đổi theo trình tự khi xuất hiện xung nhịp. Các mạch đếm là thành phần cơ bản của các hệ thống số, được sử dụng để đếm thời gian, chia tần số, điều khiển các mạch khác...



Hình 7.25. Sơ đồ khái niệm bộ đếm.

7.5.1.1. Đồ hình trạng thái bộ đếm



Hình 7.26. Đồ hình trạng thái của bộ đếm.

Đồ hình trạng thái của bộ đếm có hệ số đếm bằng K_d mô tả như hình 7.26. Khi chưa có tín hiệu vào đếm (X_d), mạch giữ nguyên trạng thái. Khi có tín hiệu vào đếm, mạch sẽ chuyển đến trạng thái kế tiếp. Cứ sau K_d tín hiệu vào X_d mạch trở về trạng thái ban đầu.

7.5.1.2. Phân loại bộ đếm

Bộ đếm có thể được phân loại như sau:

- Theo nguyên tắc làm việc:
 - + Bộ đếm đồng bộ
 - + Bộ đếm không đồng bộ
- Theo K_d (hay còn gọi là số MOD: là tổng số trạng thái khác nhau mà bộ đếm đi qua trong mỗi vòng trọn vẹn trước khi quay về vị trí ban đầu)
 - + $K_d = 2^n$
 - + $K_d \neq 2^n$

- Phân loại theo hướng đếm
 - + Đếm tăng (hay còn gọi là đếm thuận)
 - + Đếm giảm (hay còn gọi là đếm nghịch)

7.5.2. Thiết kế bộ đếm đồng bộ

Các bước thực hiện như sau:

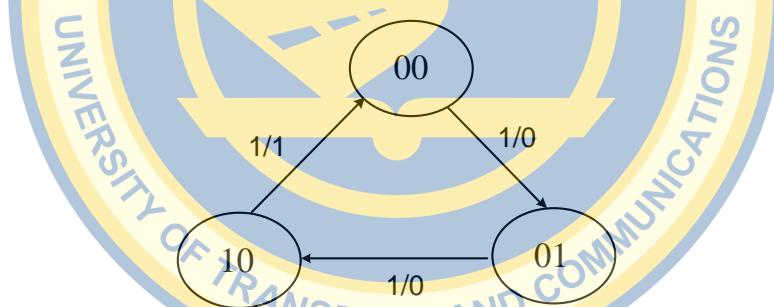
- Từ đầu bài, vẽ sơ đồ trạng thái mô tả hoạt động của bộ đếm.
- Từ bảng trạng thái, lập bảng trình tự đếm.
- Xác định số flip-flop cần dùng:
 - + Mã nhị phân và mã gray: $n \geq \log_2 K_d$
 - + Mã vòng: $n = K_d$
 - + Mã Johnson: $n = K_d/2$
- Xác định loại flip-flop dùng thiết kế bộ đếm. Lập bảng với trạng thái hiện tại, trạng thái kế tiếp, xác định đầu vào flip-flop dựa vào bảng 7.11 (xem mục 7.2.8).
 - Sử dụng bảng Karnaugh để tối thiểu hóa phương trình hàm ra của mạch.
 - Vẽ sơ đồ mạch gồm các flip-flop và các công logic.

7.5.2.1. Bộ đếm tăng

Với bộ đếm tăng, trạng thái tăng từ thấp đến cao.

Ví dụ 1: Thiết kế bộ đếm đồng bộ với $K_d = 3$, sử dụng T-FF.

Bộ đếm này sẽ có ba trạng thái, dùng 2 flip-flop. Sơ đồ trạng thái như hình 7.27.



Hình 7.27. Sơ đồ trạng thái bộ đếm $K_d = 3$.

Từ sơ đồ trạng thái, ta có bảng trạng thái bộ đếm và đầu vào kích cho các flip-flop loại T như bảng 7.18.

Bảng 7.18. Bảng trạng thái của bộ đếm và đầu vào kích hoạt cho các flip-flop.

Trạng thái hiện tại	Trạng thái kế tiếp	Đầu vào flip-flop	
$Q_A Q_B$	$Q_A Q_B$	T_A	T_B
0 0	0 1	0	1
0 1	1 0	1	1
1 0	0 0	1	0

Trạng thái 11 không sử dụng, ta coi như “X” để tối thiểu hóa.

Từ bảng karnaugh ta xác định được hàm đầu vào flip-flop.

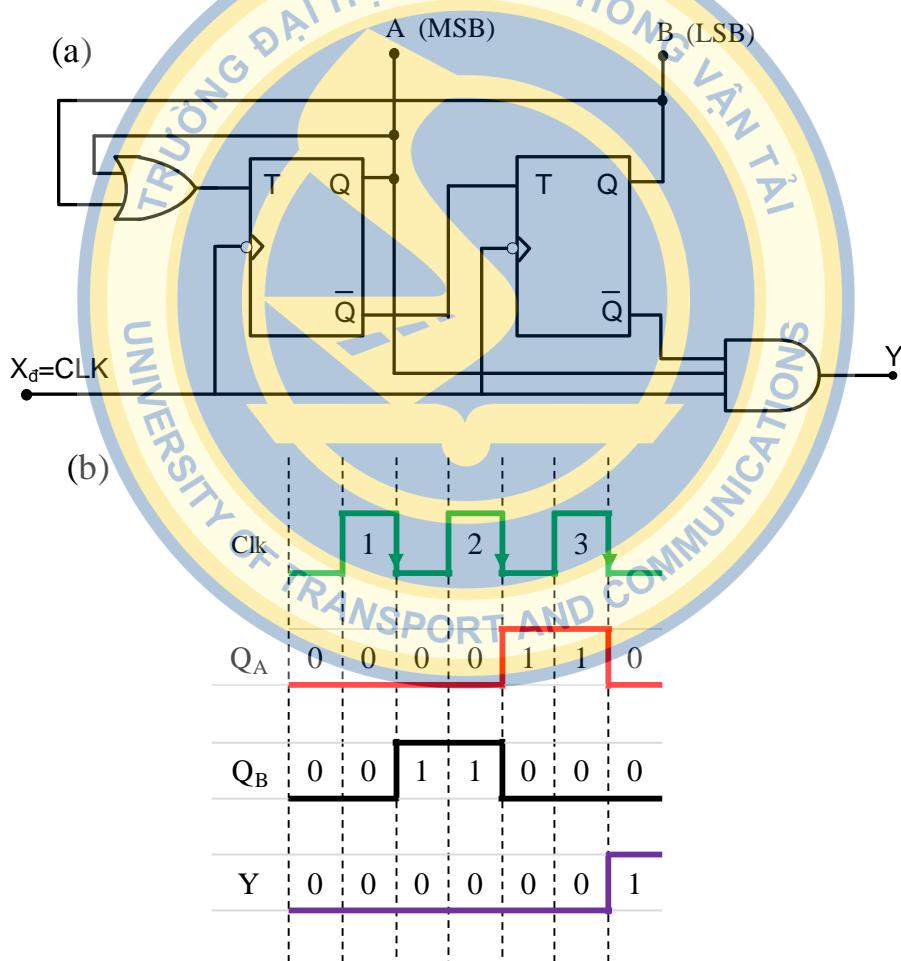
		T_A	T_B	
		Q_B	0	1
Q_A	0	0	0	1
	1	1	X	

		T_A	T_B	
		Q_B	0	1
Q_A	0	1	1	1
	1	0	X	

Từ bảng K ta có: $T_A = Q_A + Q_B$ và $T_B = \bar{Q}_A$

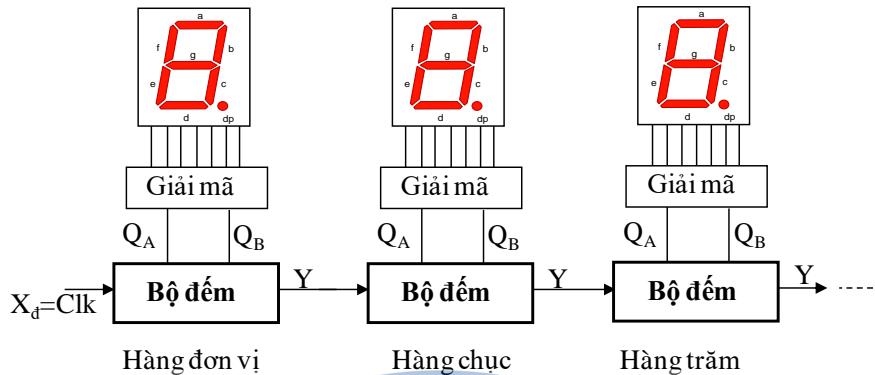
Đầu ra chuyển vị $Y = Q_A \bar{Q}_B X_d$

Từ đó ta vẽ được sơ đồ mạch với xung đếm tích cực sùn âm và giản đồ thời gian của bộ đếm như hình 7.28.



Hình 7.28. (a) Sơ đồ mạch, (b) Giản đồ thời gian của bộ đếm tăng đồng bộ $K_d = 3$.

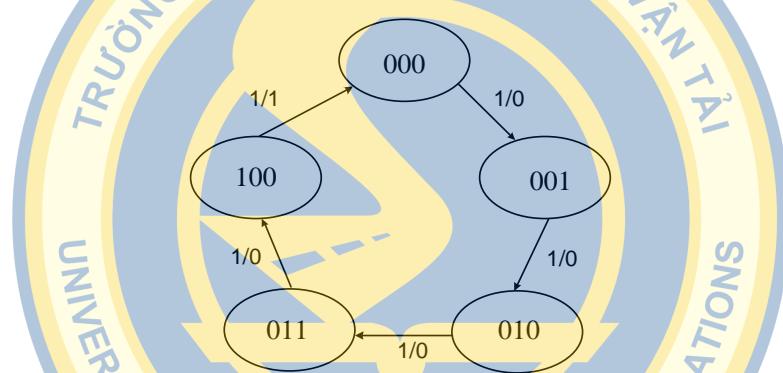
Bộ đếm bao gồm hàng đơn vị, chục, trăm...được xây dựng trên cơ sở của bộ đếm đơn vị trên, đầu ra chuyển vị của hàng đơn vị chính là đầu vào đếm của bộ đếm hàng chục,và đầu ra chuyển vị của hàng chục là đầu vào đếm của bộ đếm hàng trăm,...như sơ đồ sau.



Ví dụ 2: Thiết kế bộ đếm với $K_d = 5$, sử dụng JK-FF.

Bộ đếm với $K_d = 5$ có năm trạng thái nên cần dùng 3 flip-flop để thiết kế mạch.

Đồ hình trạng thái:



Hình 7.29. Sơ đồ trạng thái bộ đếm $K_d = 5$.

Từ sơ đồ trạng thái, ta có bảng trạng thái bộ đếm và đầu vào kích cho các flip-flop JK như bảng 7.19.

Bảng 7.19. Bảng trạng thái và đầu vào kích của flip-flop JK.

Trạng thái hiện tại	Trạng thái kế tiếp	Đầu vào flip-flop						
		J _A	K _A	J _B	K _B	J _C	K _C	
Q _A Q _B Q _C	Q _A Q _B Q _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	0 0 0	X	1	0	X	0	X	

Trạng thái 101, 110, và 111 không sử dụng, ta coi như “X” để tối thiểu hóa.

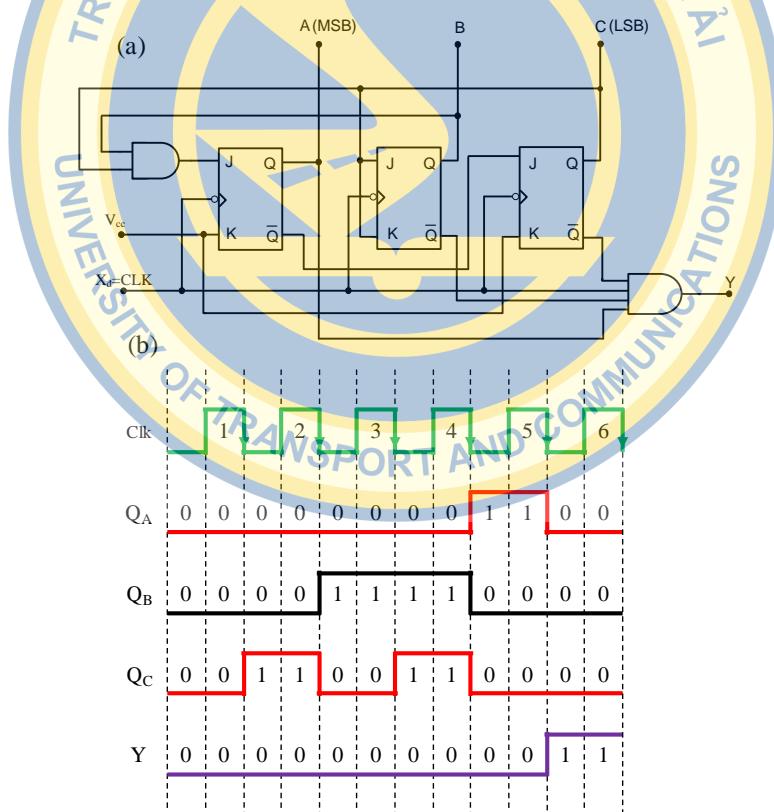
Tối thiểu hóa với bảng Karnaugh:

		J _A						K _A						J _B						K _B																			
		Q _B Q _C		00	01	11	10			Q _B Q _C		00	01	11	10			Q _B Q _C		00	01	11	10			Q _B Q _C		00	01	11	10								
		Q _A	Q _A	0	0	1	0			Q _A	Q _A	0	x	x	x			Q _A	Q _A	0	0	1	x			Q _A	Q _A	0	x	x	x			Q _A	Q _A	0	x	x	x
		0	1	x	x	x	x			0	1	x	x	x	x			0	1	x	x	x	x			0	1	x	x	x	x			0	1	x	x	x	x
		1	0	x	x	x	x			1	0	x	x	x	x			1	0	x	x	x	x			1	0	x	x	x	x			1	0	x	x	x	x
		J _C						K _C						J _C						K _C						J _C						K _C							
		Q _B	Q _C	00	01	11	10			Q _B	Q _C	00	01	11	10			Q _B	Q _C	00	01	11	10			Q _B	Q _C	00	01	11	10			Q _B	Q _C	00	01	11	10
		0	1	x	x	x	1			0	1	x	x	x	x			0	1	x	x	x	x			0	1	x	x	x	x			0	1	x	x	x	x
		1	0	x	x	x	x			1	0	x	x	x	x			1	0	x	x	x	x			1	0	x	x	x	x			1	0	x	x	x	x

$$J_A = Q_B Q_C; K_A = 1; J_B = Q_C, K_B = Q_B + Q_C; J_C = \bar{Q}_A; K_C = 1;$$

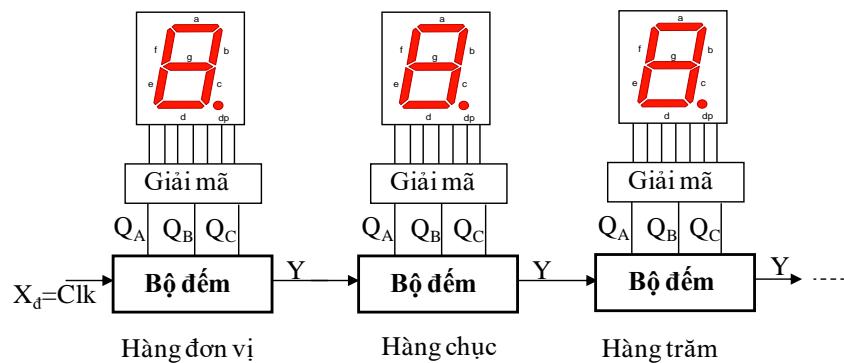
Đầu ra chuyển vị $Y = Q_A \bar{Q}_B \bar{Q}_C X_d$

$K_A = K_C = 1$, để các đầu vào này luôn đạt logic 1, ta nối với nguồn cấp. Sơ đồ mạch với xung đếm tích cực sườn âm:



Hình 7.30. Sơ đồ logic bộ đếm nhị phân $K_d = 5$.

Tương tự cũng có thể xây dựng bộ đếm gồm hàng đơn vị, chục, trăm...từ bộ đếm đơn vị trên như sau:

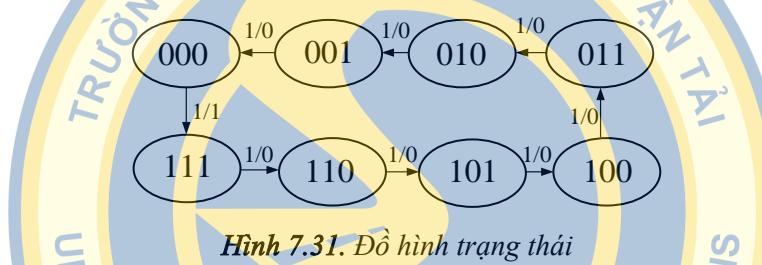


7.5.2.2. Bộ đếm giảm

Với bộ đếm giảm, trạng thái giảm từ cao xuống thấp.

Ví dụ: Thiết kế bộ đếm đồng bộ giảm với $K_d = 8$, sử dụng JK-FF.

Bộ đếm có 8 trạng thái nên cần dùng 3 flip-flop để thiết kế mạch. Sơ đồ trạng thái biểu diễn như hình sau:



Hình 7.31. Đồ họa trạng thái

Từ đồ họa trạng thái, ta có bảng trạng thái bộ đếm và đầu vào kích cho các flip-flop JK như bảng sau:

Bảng 7.20. Bảng trạng thái và đầu vào kích của flip-flop JK

Trạng thái hiện tại	Trạng thái kế tiếp	Đầu vào flip-flop						
		J _A	K _A	J _B	K _B	J _C	K _C	
Q _A Q _B Q _C	Q _A Q _B Q _C							
1 1 1	1 1 0	X	0	X	0	X	1	
1 1 0	1 0 1	X	0	X	1	1	X	
1 0 1	1 0 0	X	0	0	X	X	1	
1 0 0	0 1 1	X	1	1	X	1	X	
0 1 1	0 1 0	0	X	X	0	X	1	
0 1 0	0 0 1	0	X	X	1	1	X	
0 0 1	0 0 0	0	X	0	X	X	1	
0 0 0	1 1 1	1	X	1	X	1	X	

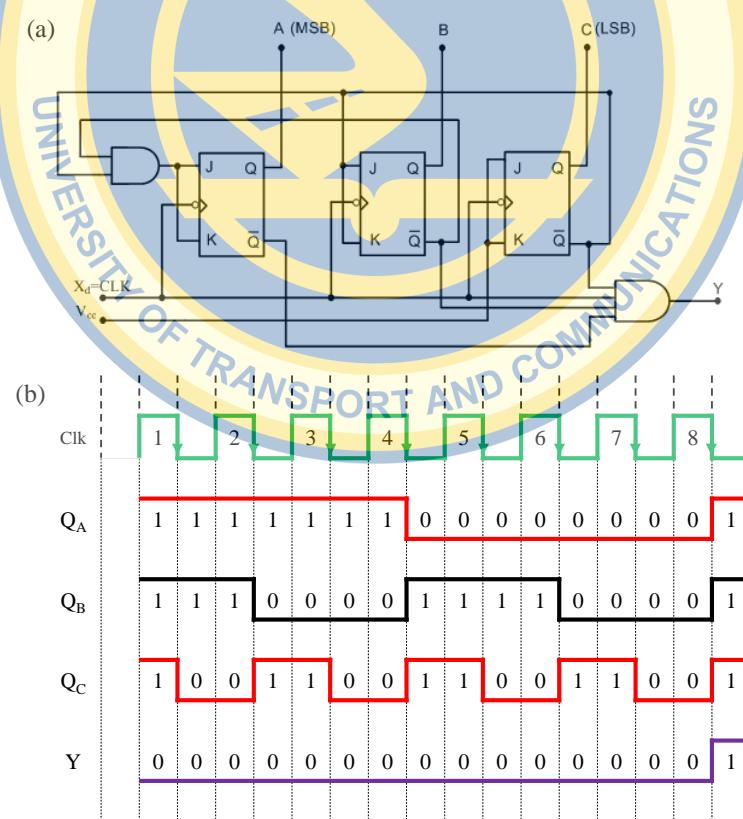
		J _A				K _A					
		Q _B Q _C	00	01	11	10	Q _B Q _C	00	01	11	10
Q _A	0		0	0	0	0		x	x	x	x
	1	x	x	x	x	1	0	0	0	0	
		J _B				K _B					
		Q _B Q _C	00	01	11	10	Q _B Q _C	00	01	11	10
Q _A	0	1	0	x	x	x	x	x	0	0	1
	1	1	0	x	x	x	x	x	0	0	1

		J _C				K _C					
		Q _B Q _C	00	01	11	10	Q _B Q _C	00	01	11	10
Q _A	0		x	x	1			1	1	1	x
	1	1	x	x	1		1	1	1	x	
		J _C				K _C					
		Q _B Q _C	00	01	11	10	Q _B Q _C	00	01	11	10
Q _A	0		0	x	x	1		1	1	1	x
	1	1	x	x	1		1	1	1	x	

$$J_A = \bar{Q}_B \bar{Q}_C; K_A = \bar{Q}_B \bar{Q}_C; J_B = \bar{Q}_C; K_B = \bar{Q}_C; J_C = 1; K_C = 1;$$

Đầu ra chuyển vị: $Y = \bar{Q}_A \bar{Q}_B \bar{Q}_C \cdot X_d$

Vẽ sơ đồ mạch



7.5.3. Bộ đếm không đồng bộ

So với bộ đếm đồng bộ, bộ đếm không đồng bộ thiết kế khá đơn giản, không cần phải xác định đầu vào kích cho các flip-flop, mà chúng được đặt trước giá trị.

7.5.3.1. Bộ đếm tăng

– Với $K_d = 2^n$, n là số flip-flop

Ta thực hiện việc xây dựng mạch theo nguyên tắc sau:

+ Nối nối tiếp các flip-flop,

+ Đầu vào mỗi flip-flop luôn được đặt trạng thái logic 1 bằng cách nối đến nguồn cấp (V_{DD} hay V_{cc}).

+ Đầu vào X_d chỉ nối đến CLK của flip-flop thứ nhất.

+ Từ flip-flop thứ hai trở đi CLK được nối với đầu ra Q của flip-flop trước đó.

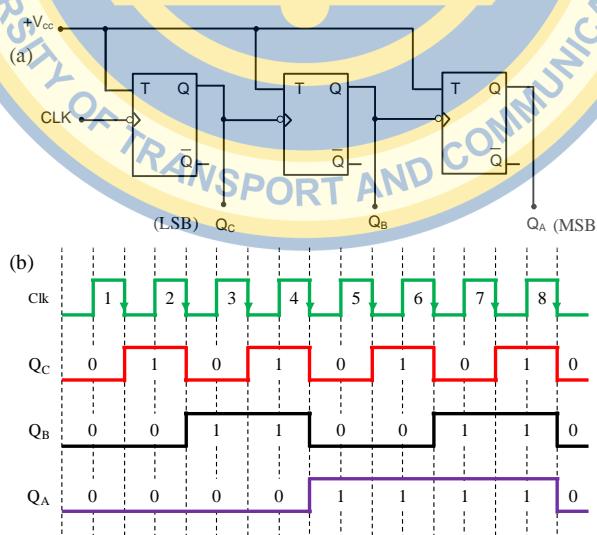
Ví dụ, xây dựng bộ đếm không đồng bộ $K_d = 8$, kích thích sườn âm.

Ta có, $n= 3$. Bảng 7.21 biểu diễn trình tự các trạng thái nhị phân khi xung nhịp đưa vào tuần tự.

Mạch được xây dựng như hình 7.33(a). Hình 7.33 (b) biểu diễn giản đồ thời gian của bộ đếm. Giá trị Q_C lật một lần trong mỗi chu kỳ xung nhịp, Q_B thay đổi sau sườn âm của tín hiệu Q_C , và Q_A thay đổi sau sườn âm của tín hiệu Q_B .

Bảng 7.21. Trình tự đếm bộ đếm tăng 3 bit.

Trạng thái bộ đếm	Q_A	Q_B	Q_C
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



Hình 7.33.(a) Sơ đồ mạch, (b) Gắn đồ thời gian của bộ đếm tăng không đồng bộ $K_d = 8$.

Nhân xét:

Từ giản đồ thời gian của bộ đếm không đồng bộ ta thấy, nếu chuỗi xung vào CLK có tần số f , thì tần số của Q_C sẽ là $f/2$, tần số Q_B bằng $f/4$, và tần số Q_A là $f/8$. Do đó, có thể sử dụng mạch đếm không đồng bộ làm mạch chia tần số, khi thay đầu vào đếm X_d bằng một nguồn có tần số f , nếu có n flip-flop được sử dụng trong mạch thì tần số lấy ra ở bit MSB sẽ là $f/2^n$.

– Với $K_d < 2^n$

Đây là loại bộ đếm không đồng bộ có số trạng thái đếm ít hơn ($< 2^n$) số trạng thái tối đa ($= 2^n$) mà n flop-flop hình thành. Như vậy mạch sẽ thay đổi từ 0 đến K_{d-1} ($< 2^n$) và lại quay về 0, mà không đến trạng thái tối đa 2^n . Một trong những phương pháp đơn giản để xây dựng mạch đếm kiểu này là kết nối trạng thái K_{d-1} với chân xóa CLR của tất cả các flip-flop, như vậy mạch sẽ bị xóa toàn bộ (hay đưa về trạng thái 0) ngay sau khi mạch đạt trạng thái K_{d-1} .

Ví dụ thiết kế bộ đếm không đồng bộ $K_d = 6$, với CLR tích cực thấp

Với $K_d = 6$, cần sử dụng $n = 3$, mạch sẽ thay đổi từ $000 \rightarrow 001 \rightarrow \dots \rightarrow 110 \rightarrow 000$ như bảng sau:

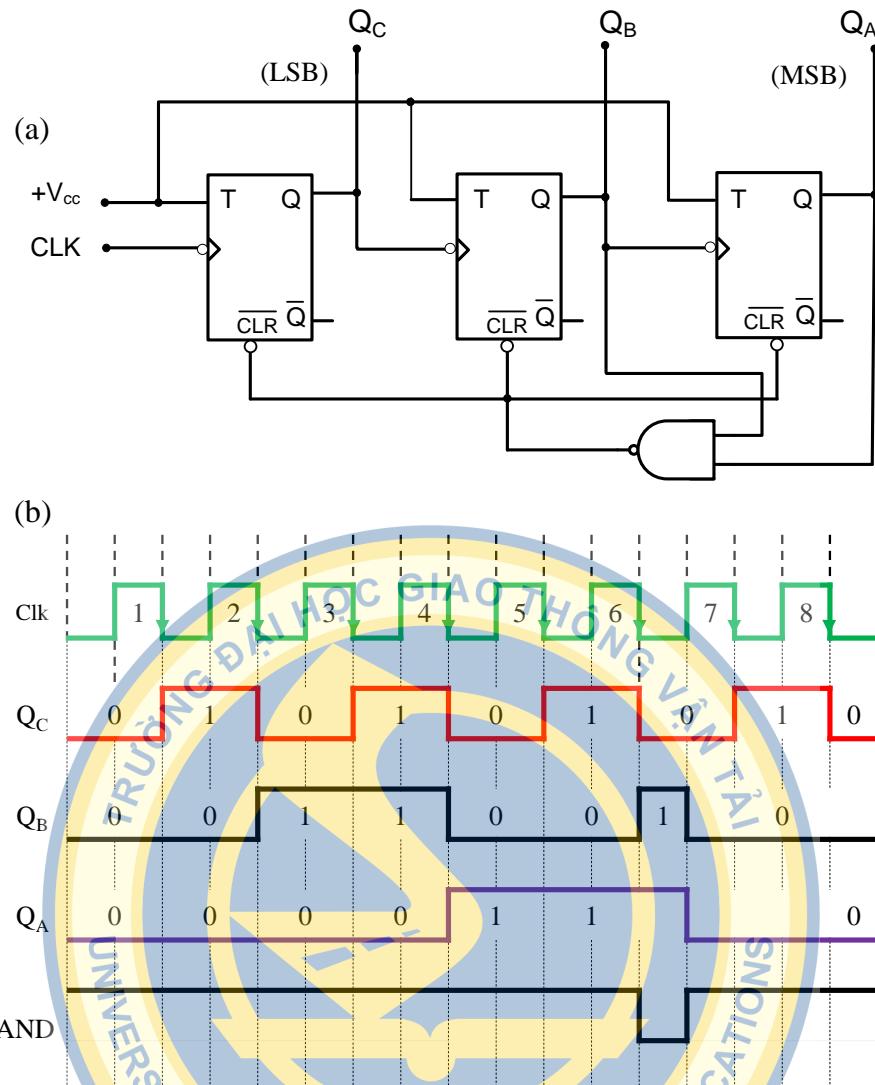
Bảng 7.22. Trình tự đếm bộ đếm.

Trạng thái bộ đếm	Q_A	Q_B	Q_C
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
0	0	0	0

Như vậy hàm xóa $= Q_A Q_B Q_C + Q_A Q_B Q_C = Q_A Q_B$, với $Q_A Q_B Q_C$ là trạng thái không xác định. CLR tích cực thấp, vậy mạch xóa chính là cổng NAND.

Mạch được xây dựng ở hình 7.34(a): Đầu ra cổng NAND nối đến đầu vào Clear từng flip-flop. Khi cổng NAND tạo mức cao ở đầu ra thì không ảnh hưởng đến bộ đếm. Đầu ra cổng NAND ở mức thấp chỉ khi $Q_A Q_B = 11$. Điều kiện này xảy ra ở xung nhịp thứ 6. Mức thấp tại đầu ra của cổng NAND sẽ xóa bộ đếm về trạng thái 000. Khi các flip-flop đã bị xóa, đầu ra của cổng NAND quay trở lại mức cao.

Từ hình 7.34 (b) ta thấy đầu ra Q_B là xung hẹp do thời gian xuất hiện trạng thái 110 rất ngắn trước khi bị xóa. Đầu ra Q_A có tần số bằng 1/6 tần số đầu vào, nên cũng có thể sử dụng làm bộ chia tần.



Hình 7.34. (a) Sơ đồ mạch, (b) Giải đồ thời gian của bộ đếm tăng không đồng bộ $K_d = 6$.

Nhân xét:

Mặc dù bộ đếm đạt đến trạng thái 110, và duy trì ở trạng thái này trong thời gian rất ngắn (cỡ ns, tùy theo độ trễ của các mạch logic và flip-flop) trước khi về trạng thái 000. Do đó ta có thể nói bộ đếm đếm thực chất từ 000 đến 101, bỏ qua trạng thái 110 và 111.

7.5.3.2. Bộ đếm giảm

Bộ đếm giảm sử dụng n flip-flop đếm từ số lớn nhất ($2^n - 1$) đến 0.

Ví dụ xây dựng bộ đếm giảm $K_d=8$.

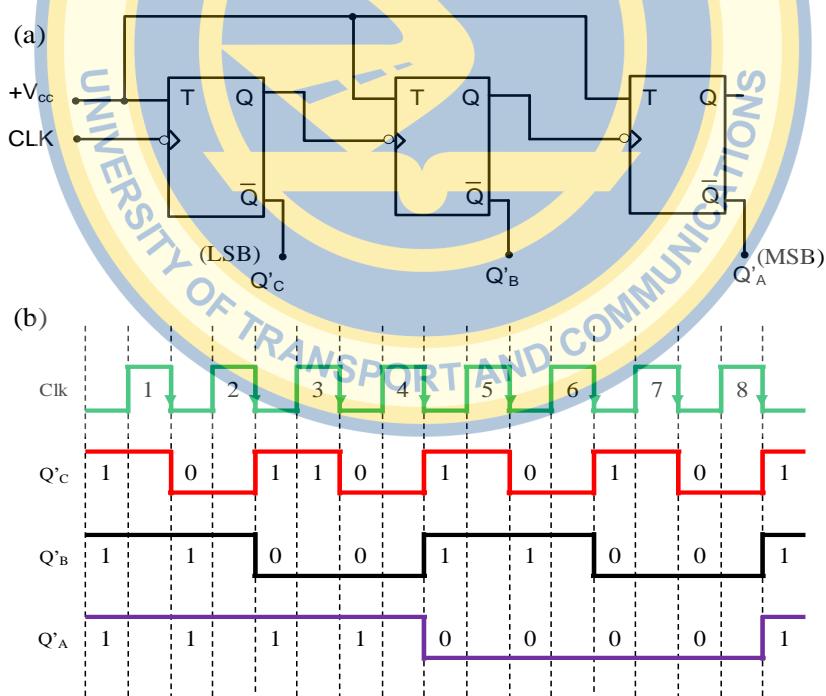
Ta có $n = 3$ bit, mạch gồm 8 trạng thái từ cao xuống thấp như bảng 7.23

Bảng 7.23. Trình tự đếm bộ đếm giảm 3 bit nhị phân.

Trình tự đếm	Q_A	Q_B	Q_C
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0

Ta có thể xây dựng mạch theo ba cách sau:

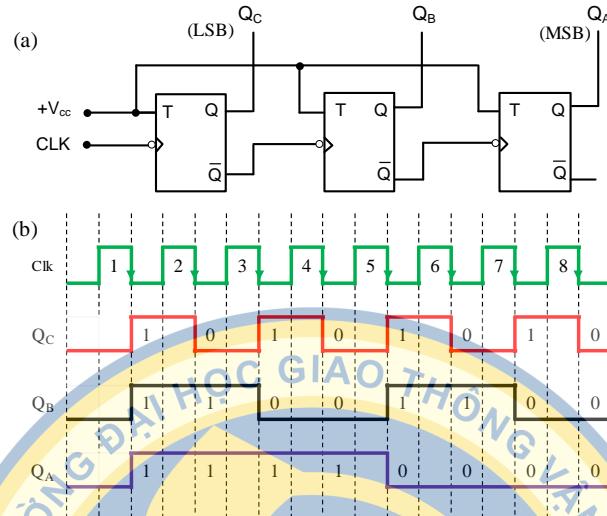
- *Mạch kích thích sườn âm, đầu ra đảo:* sơ đồ hình 7.35 với trạng thái của bộ đếm lấy từ đầu ra đảo của các flip-flop. Chuỗi đếm bắt đầu từ 111, khi có sườn âm của xung nhịp Q_C sẽ lật trạng thái, Q_B sẽ lật trạng thái ứng với sườn âm của xung nhịp đầu ra Q_C , và Q_A sẽ lật trạng thái ứng với sườn âm của xung nhịp đầu ra Q_B .



Hình 7.35. Bộ đếm không đồng bộ 3 bit, kích thích sườn âm, đầu ra đảo.

- *Mạch kích thích sùn âm, đầu ra thường*: Sơ đồ mạch và dạng sóng biểu diễn tương ứng ở hình 7.36(a) và 7.36(b). Ở loại mạch này, xung nhịp CLK của flip-flop thứ 2, thứ 3,... được điều khiển bởi đầu ra đảo (\bar{Q}) của flip-flop trước đó.

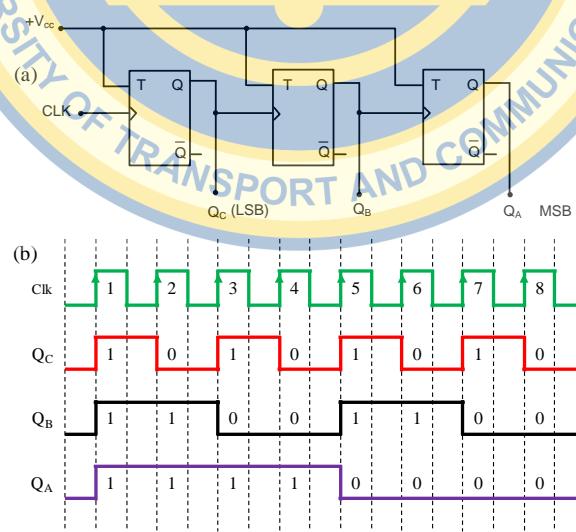
Bắt đầu từ 000, tại sùn âm xung nhịp thứ nhất, bộ đếm sẽ là 111. Tại sùn âm xung nhịp thứ 2 bộ đếm sẽ là 110, đến xung nhịp thứ 3 sẽ là 101,...



Hình 7.36. Bộ đếm giảm không đồng bộ 3 bit, kích thích sùn âm, đầu ra thường.

- *Mạch kích thích sùn dương, đầu ra thường*: Sơ đồ mạch và dạng sóng biểu diễn tương ứng ở hình 7.37(a) và 7.37(b), các flip-flop chuyển trạng thái ứng với sùn dương của xung nhịp.

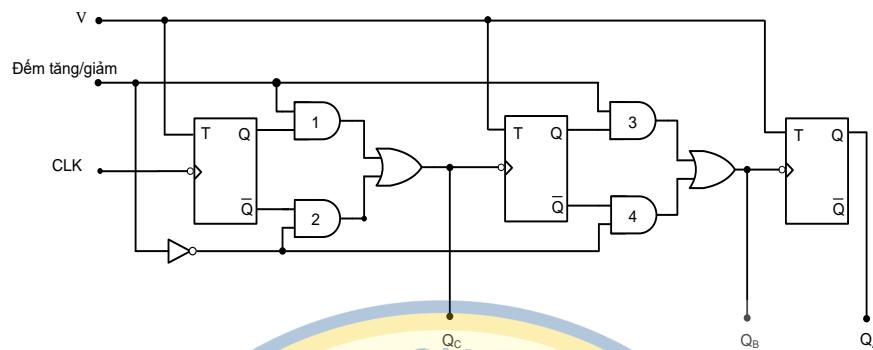
Khi có sùn dương của xung nhịp Q_C sẽ lật trạng thái, Q_B sẽ lật trạng thái ứng với sùn dương của xung nhịp đầu ra Q_C , và Q_A sẽ lật trạng thái ứng với sùn dương của xung nhịp đầu ra Q_B .



Hình 7.37. Bộ đếm giảm không đồng bộ 3 bit, kích thích sùn dương, đầu ra thường.

7.5.3.3. Bộ đếm tăng-giảm không đồng bộ

Ở phần trước chúng ta đã xét từng bộ đếm tăng, đếm giảm riêng biệt. Kết hợp hai bộ đếm này được bộ đếm tăng/giảm. Hoạt động của bộ đếm được điều khiển bởi đầu vào điều khiển tăng - giảm. Trình tự đếm được biểu diễn trong bảng 7.24.



Hình 7.38. Bộ đếm tăng giảm không đồng bộ 3 bit.

Bảng 7.24. Trình tự đếm của bộ đếm tăng-giảm 3 bit.

Trạng thái	Chế độ đếm tăng			Chế độ đếm giảm			
	Q _A	Q _B	Q _C	Trạng thái	Q _A	Q _B	Q _C
0	0	0	0	7	1	1	1
1	0	0	1	6	1	1	0
2	0	1	0	5	1	0	1
3	0	1	1	4	1	0	0
4	1	0	0	3	0	1	1
5	1	0	1	2	0	1	1
6	1	1	0	1	0	0	1
7	1	1	1	0	0	0	0

Từ sơ đồ mạch ta thấy có ba cổng logic cho mỗi tầng để chuyển từ chế độ đếm tăng sang đếm giảm. Các cổng logic cho phép đầu ra không đảo hoặc đầu ra đảo của flip-flop đến đầu vào Clock của flip-flop sau, phụ thuộc vào trạng thái đầu vào điều khiển đếm tăng/giảm.

- Khi đường điều khiển đếm tăng/giảm ở mức cao: Cổng AND 1 và 3 sẽ ở mức cao. Do đó, đầu ra của các cổng OR cũng ở mức cao, tín hiệu theo nhánh qua AND 1 và 3, mạch lúc này tương đương với mạch đếm tăng hình 7.34. Khi có xung nhịp, bộ đếm sẽ đếm tăng theo trình tự đếm từ 000 đến 111.

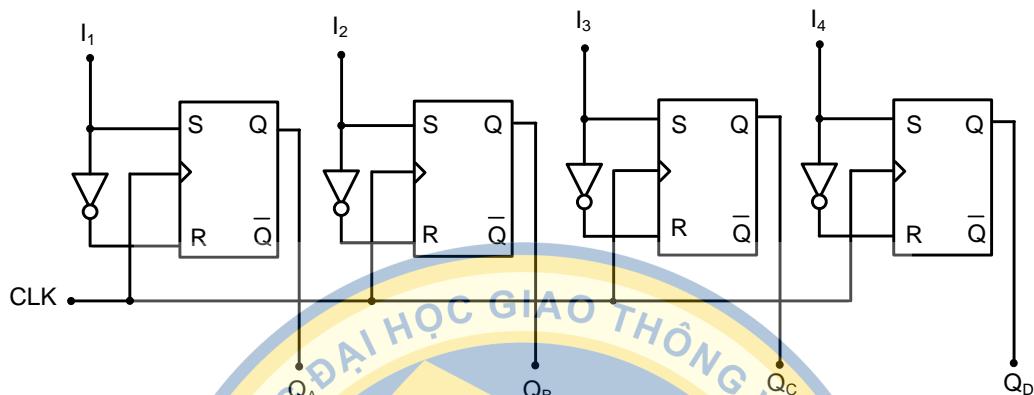
- Tương tự, khi đường điều khiển tăng/giảm ở mức thấp: Cổng AND 2 và 4 sẽ ở mức cao. Do đó, đầu ra của các cổng OR cũng ở mức cao, nhưng tín hiệu theo nhánh qua AND 2 và 4, mạch lúc này tương đương với mạch đếm giảm hình 7.35. Khi có xung nhịp, bộ đếm sẽ đếm giảm theo trình tự đếm từ 111 đến 000.

7.6. THANH GHI DỊCH

7.6.1. Giới thiệu

Thanh ghi dịch là mạch tuần tự tập hợp các flip-flop, có thể lưu trữ và dịch dữ liệu.

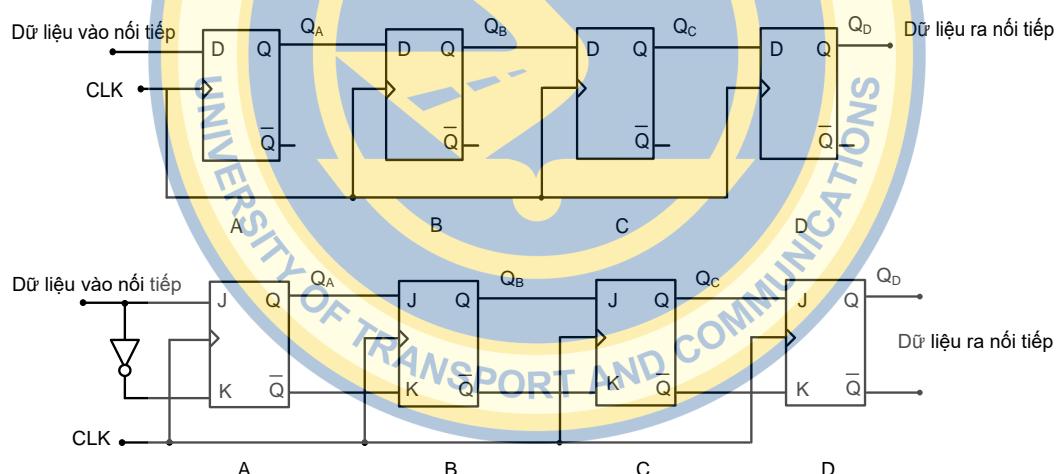
Thanh ghi có thể được thiết kế bởi các loại flip-flop khác nhau (SR-FF, JK-FF, D-FF), ví dụ hình 7.39 biếu diễn một dạng flip-flop được cấu thành từ 4 SR flip-flop, với đầu vào xung nhịp đồng bộ.



Hình 7.39. Thanh ghi 4 bit.

– *Thanh ghi dịch phải*

Thanh ghi dịch phải có thể được xây dựng từ JK-FF hay D-FF như hình 7.40.



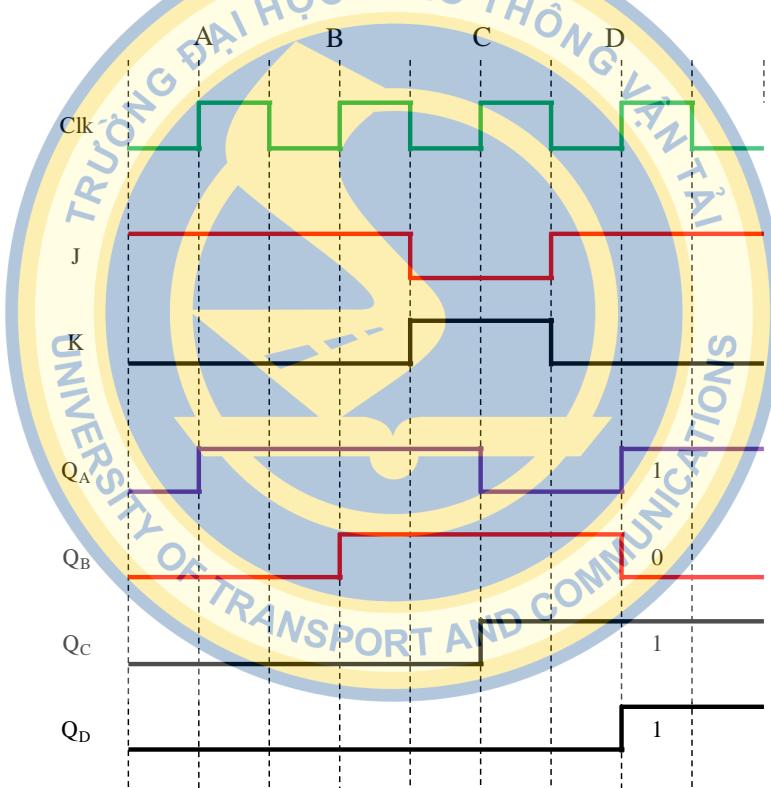
Hình 7.40. Thanh ghi dịch phải sử dụng D-FF và JK-FF.

Xung nhịp được đưa đồng thời đến các flip-flop. Khi có xung nhịp, mỗi flip-flop ở trạng thái set hoặc reset theo dữ liệu sẵn có ở các đầu vào tương ứng của từng flip-flop. Khi xuất hiện xung nhịp thứ nhất, bit dữ liệu đầu vào đưa đến A-FF. Ở thời điểm này, dữ liệu ở A-FF dịch chuyển sang B-FF và cứ tiếp tục như vậy đến các flip-flop sau. Mỗi xung nhịp, dữ liệu được lưu trong thanh ghi và dịch chuyển sang flip-flop bên phải.

Bảng 7.25. Hoạt động của thanh ghi dịch phải.

Xung nhịp	Q _A	Q _B	Q _C	Q _D	Đầu ra nối tiếp tại Q _D
Giá trị ban đầu	0	0	0	0	0
Sau xung nhịp thứ 1	1	0	0	0	0
Sau xung nhịp thứ 2	1	1	0	0	0
Sau xung nhịp thứ 3	0	1	1	0	0
Sau xung nhịp thứ 4	1	0	1	1	1

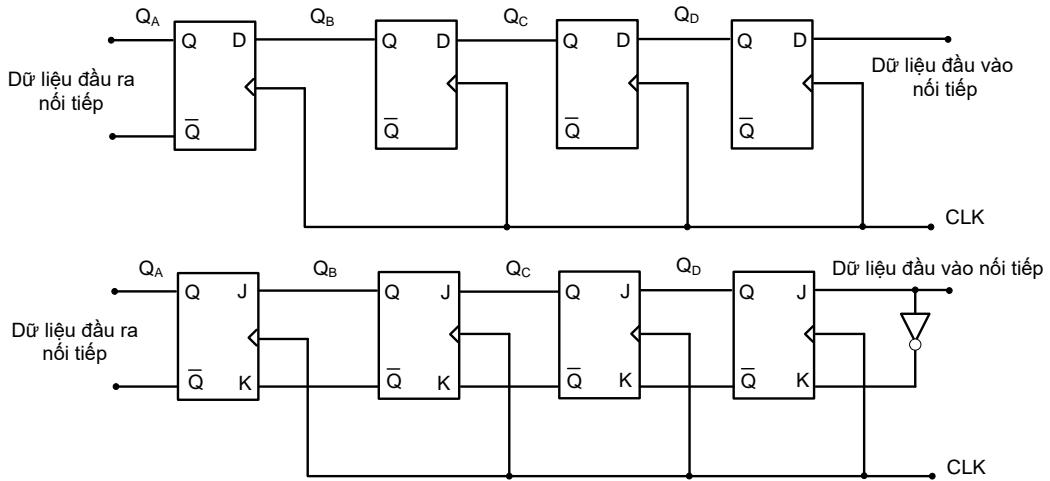
Ví dụ: xét tất cả các flip-flop là reset, đầu vào logic 1011 đưa đến đầu vào nối tiếp của A-FF. Dữ liệu sau bốn xung nhịp được mô tả như bảng 7.25. Dạng xung đầu vào bốn bit 1011 được biểu diễn như hình 7.41.



Hình 7.41. Dạng xung thanh ghi dịch phải 4 bit đầu vào nối tiếp.

– Thanh ghi dịch trái

Thanh ghi dịch trái cũng được xây dựng bởi JK-FF hoặc D-FF biểu diễn như hình 7.42. Nếu ta đưa 4 bit nhị phân 1110 vào thanh ghi. Nguyên tắc hoạt động được biểu diễn trong bảng 7.26.



Hình 7.42. Thanh ghi dịch trái sử dụng D-FF và sử dụng JK-FF.

Bảng 7.26. Bảng hoạt động của thanh ghi dịch trái.

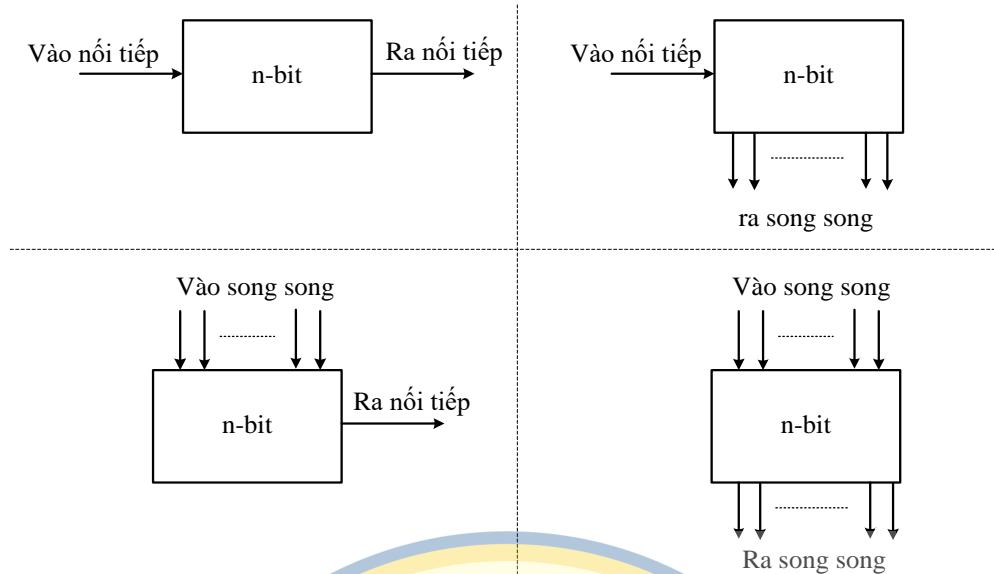
Xung nhịp	Q _A	Q _B	Q _C	Q _D	Đầu ra nối tiếp tại Q _A
Giá trị ban đầu	0	0	0	0	0
Sau xung nhịp thứ 1	0	0	0	0	0
Sau xung nhịp thứ 2	0	0	0	1	0
Sau xung nhịp thứ 3	0	0	1	1	0
Sau xung nhịp thứ 4	0	1	1	1	0

7.6.2. Ứng dụng trong lưu và đọc dữ liệu

Dữ liệu trong thanh ghi dịch có thể thực hiện dịch theo hai cách: dịch nối tiếp và dịch song song. Dịch nối tiếp là phương pháp dịch 1 bit khi có xung nhịp tuần tự, bắt đầu từ LSB hoặc MSB. Hay nói cách khác, trong phương pháp dịch song song, tất cả dữ liệu (đầu vào hay đầu ra) dịch đồng thời khi có xung nhịp.

Có hai cách dịch dữ liệu vào/ra thanh ghi (nối tiếp hay song song), nên có 4 loại thanh ghi cơ bản như hình 7.43.

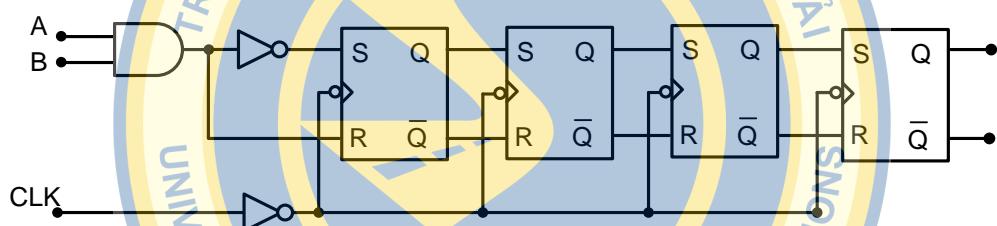
- Vào nối tiếp/ra nối tiếp (SISO- Serial in/serial out).
- Vào nối tiếp/ra song song (SIPO- Serial in/parallel out).
- Vào song song/ra nối tiếp (PISO- Parallel in/ serial out).
- Vào song song/ra song song (PIPO- Parallel in/ Parallel out)



Hình 7.43. Các phương pháp ghi và đọc dữ liệu của thanh ghi dịch.

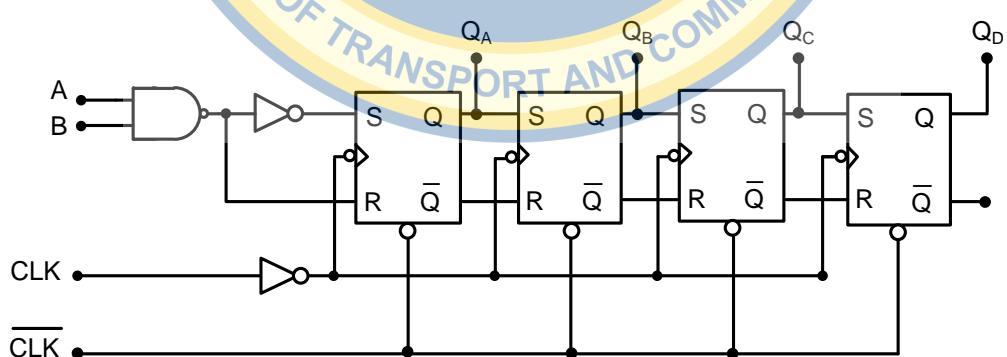
– **Thanh ghi dịch 4 bit vào nối tiếp/ra nối tiếp**

Có 8 SR-FF đưa đến đầu vào nối tiếp cũng như đầu ra nối tiếp. Xung nhịp đầu vào được kích hoạt là sườn âm xung nhịp.



Hình 7.44. Thanh ghi dịch 4 bit.

– **Thanh ghi dịch 4 bit vào nối tiếp/ra song song**



Hình 7.45. Sơ đồ logic thanh ghi dịch 4 bit.

Có 4 SR-FF, kích hoạt bởi sườn âm của xung nhịp. Sơ đồ hình 7.45 cũng giống hình 7.44, chỉ có hai sự khác biệt: (1) mỗi flip-flop có một đầu vào Clear không đồng bộ, và (2) có sẵn đầu ra ở mỗi flip-flop- 4 bit được lưu trong thanh ghi ở đầu ra là đồng thời.

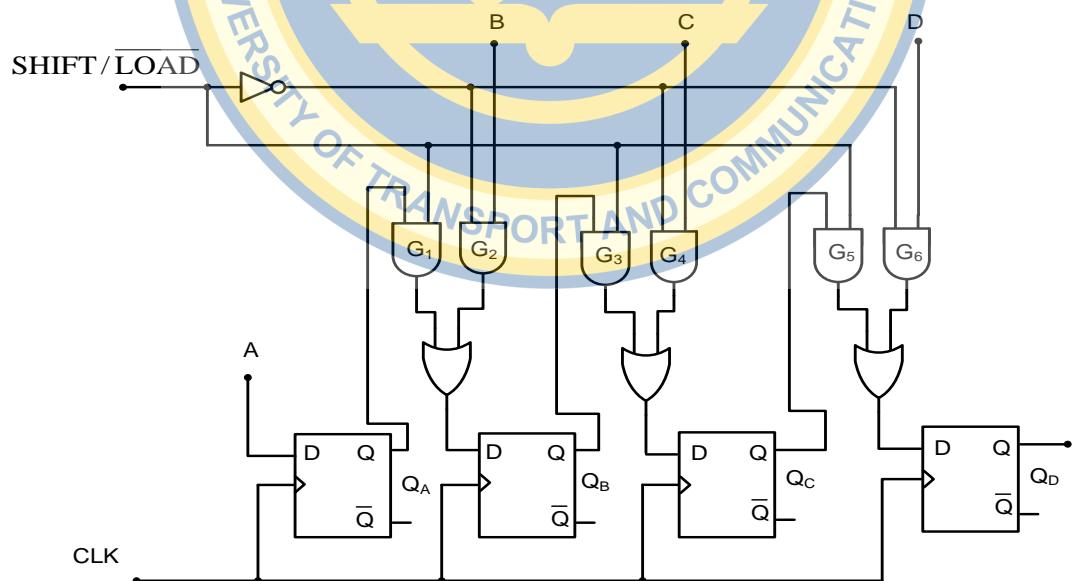
Với các đầu vào nối tiếp A và B, nếu dữ liệu vào nối tiếp đưa đến B thì A có thể sử dụng như một đường điều khiển.

Khi A ở mức cao: cổng NAND được kích hoạt và dữ liệu vào nối tiếp qua cổng NAND đảo. Dữ liệu đầu vào dịch nối tiếp vào thanh ghi.

Khi A ở mức thấp: đầu ra cổng NAND ở mức cao, dòng dữ liệu vào bị chặn và đến xung nhịp tiếp theo sẽ dịch bit 0 vào flip-flop đầu tiên, mỗi xung dương sẽ dịch bit 0 khác vào thanh ghi.

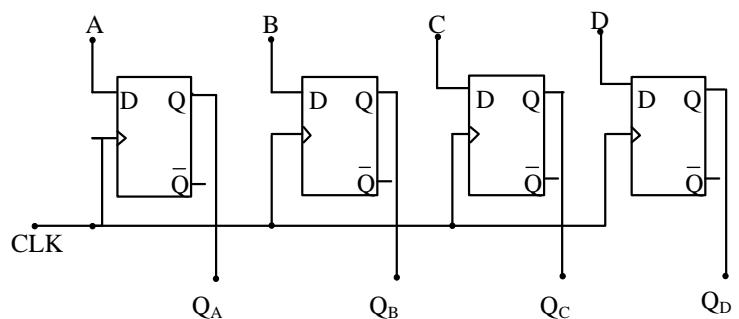
– Thanh ghi dịch 4 bit vào song song/ra nối tiếp

Thanh ghi 4 bit vào song song/ra nối tiếp minh họa như hình 7.46. A,B,C,D là dữ liệu đầu vào song song và (SHIFT/ LOAD) là đầu vào điều khiển các đầu vào dịch dữ liệu song song hay nối tiếp. Cổng OR cho phép dữ liệu vào dịch song song hay không, phụ thuộc cổng các AND được kích hoạt bởi đầu vào SHIFT/ LOAD . Khi SHIFT / LOAD ở mức cao, các cổng AND G₁, G₃, G₅ được kích hoạt, cho phép bit dữ liệu dịch phải từ tầng này sang tầng kế tiếp. Khi SHIFT/ LOAD ở mức thấp, cổng NAND G₂, G₄, G₆ được kích hoạt, cho phép bit dữ liệu đầu vào dịch song song. Khi có xung nhịp, D = 1 các flip-flop sẽ là set; D = 0 các flip-flop sẽ là reset, do đó lưu trữ đồng thời 4 bit.



Hình 7.46. Thanh ghi dịch 4 bit vào song song/ra nối tiếp.

– *Thanh ghi dịch 4 bit vào song song/ ra song song*



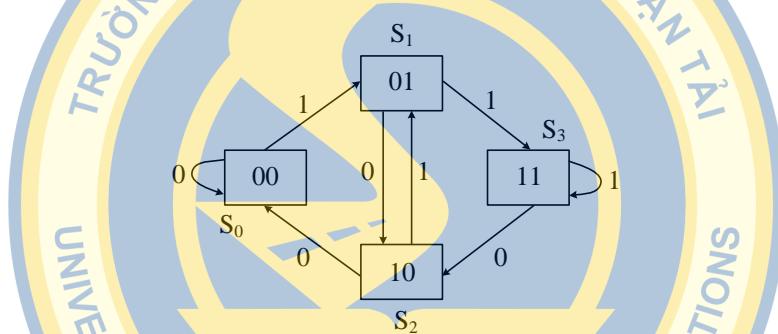
Hình 7.47. Thanh ghi dịch 4 bit vào song song/ ra song song.

Các đầu vào A,B,C,D nối trực tiếp đến đầu vào D của các flip-flop. Khi xung nhịp chuyển trạng thái, các đầu vào này đưa đến thanh ghi và đầu ra tương ứng Q_A , Q_B , Q_C , Q_D .

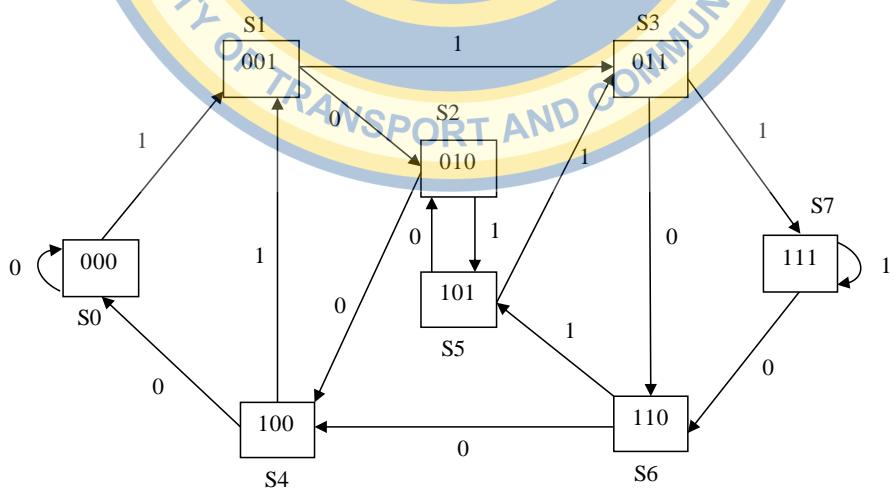
7.6.3. Ứng dụng trong thiết kế mạch tuần tự

7.6.3.1. Đồ hình trang thái

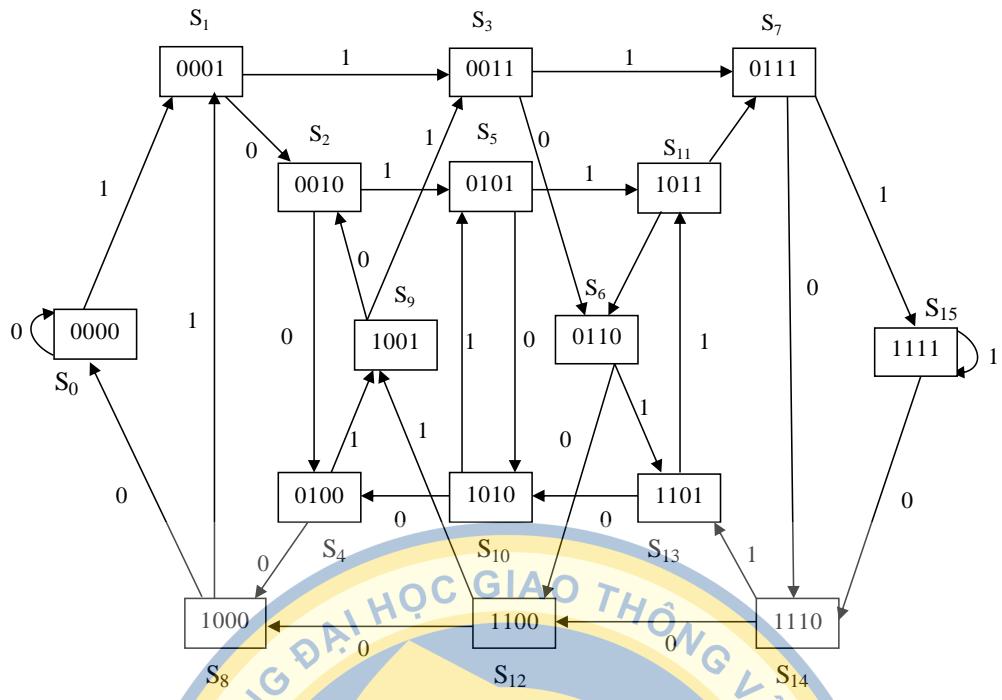
Đồ hình trang thái của thanh ghi dịch theo mô hình De Bruijn như sau:



Hình 7.48. Đồ hình trạng thái thanh ghi dịch 2 bit.



Hình 7.49. Đồ hình trạng thái thanh ghi dịch 3 bit.

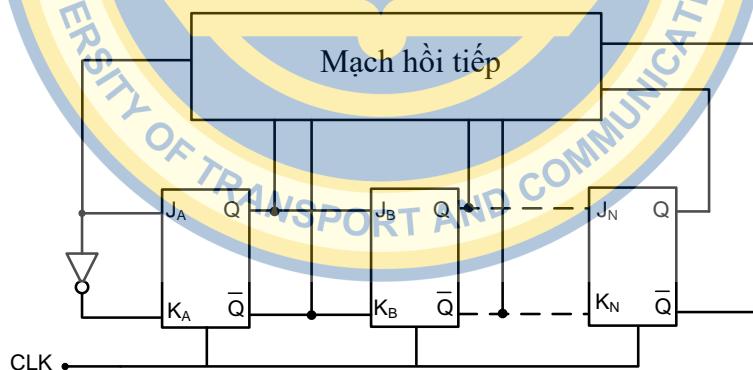


Hình 7.50. Đồ hình trạng thái thanh ghi dịch 4 bit.

7.6.3.2. Thiết kế bộ đếm dùng thanh ghi dịch

Từ đồ hình Bruijn của thanh ghi dịch ứng với giá trị hồi tiếp sẽ xác định được trạng thái kế tiếp. Bài toán thiết kế bộ đếm dùng thanh ghi dịch sẽ trở thành bài toán thiết kế hàm hồi tiếp cho thanh ghi dịch với các trạng thái là chu trình khép kín và số trạng thái bằng hệ số đếm.

Sơ đồ khởi



Hình 7.51. Bộ đếm dùng thanh ghi dịch.

Các bước thiết kế bộ đếm dùng thanh ghi dịch

- Bước 1: Xác định số flip-flop (số bit) của thanh ghi dịch $n \geq \log_2 K_d$
- Bước 2: Lựa chọn chu trình chuyển trạng thái bộ đếm theo đồ hình Bruijn
- Bước 3: Xác định hàm hồi tiếp và tối thiểu hóa hàm này.

- Bước 4: Xây dựng sơ đồ mạch.

Ví dụ: Thiết kế bộ đếm $K_d = 8$ dùng thanh ghi dịch.

- Bước 1: Xác định số bit $n = \log_2 8 = 3$

- Bước 2: Chọn chu trình chuyển biến trạng thái, căn cứ vào đồ hình trạng thái của thanh ghi dịch ta lựa chọn một chu trình có 8 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: Xác định hàm hồi tiếp, ký hiệu các flip-flop lần lượt là CBA, ta có bảng mã hóa trạng thái và hàm hồi tiếp như sau:

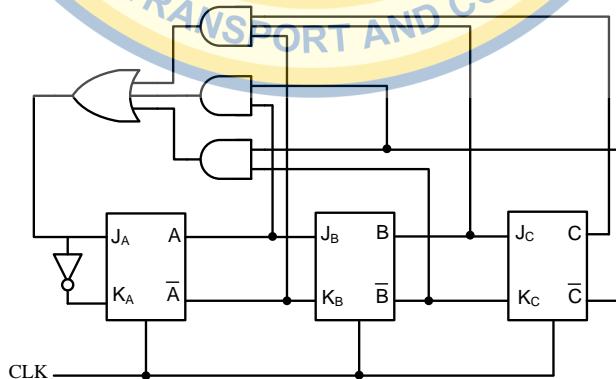
S	C	B	A	f_{ht}
S_0	0	0	0	1
S_1	0	0	1	1
S_3	0	1	1	1
S_7	1	1	1	0
S_6	1	1	0	1
S_5	1	0	1	0
S_2	0	1	0	0
S_4	1	0	0	0

Tối thiểu hóa f_{ht}:

BA		00	01	11	10
C		0	(1)	(1)	1
0	1	0	0	0	0
1	0	0	0	0	1

$$\text{Ta được: } f_{ht} = \overline{CB} + \overline{CA} + CBA$$

- Bước 4: Xây dựng sơ đồ

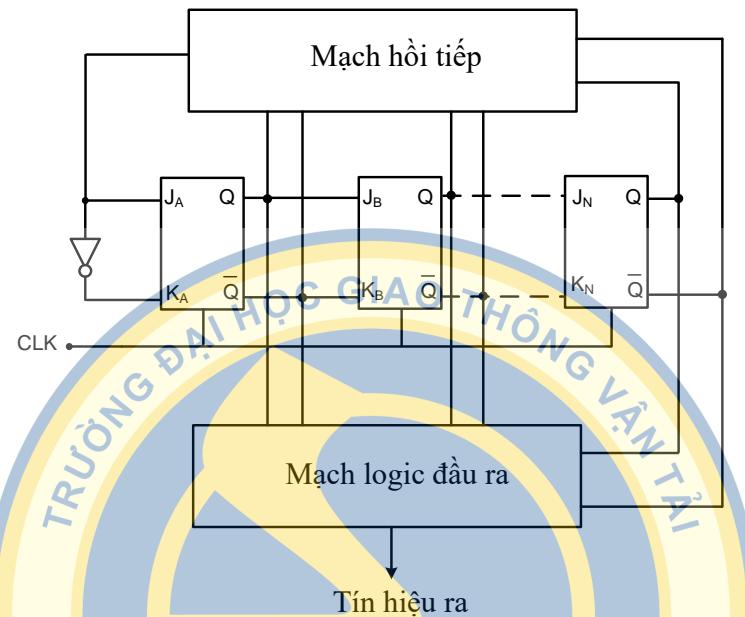


Hình 7.52. Sơ đồ mạch đếm sử dụng thanh ghi dịch.

7.6.3.3. Mạch tạo tín hiệu tuần hoàn

Thanh ghi dịch có thể dùng để tạo chuỗi tín hiệu tuần hoàn theo yêu cầu cho trước. 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. Có thể thấy rằng mạch thiết kế chính là bộ đếm có $K_d = L$ kết hợp với một mạch đầu ra cho phép lấy dãy tín hiệu tuần hoàn.

Ta có sơ đồ khối của mạch như sau:



Hình 7.53. Mạch tạo tín hiệu tuần hoàn từ thanh ghi dịch.

Các bước thiết kế **mạch tạo dãy tín hiệu tuần hoàn** từ thanh ghi dịch

- Bước 1: Xác định số bit n của thanh ghi dịch $n \geq \log_2 K_d$
- Bước 2: Lựa chọn chu trình chuyển trạng thái bộ đếm theo đồ họa Bruijn
- Bước 3: Xác định hàm hồi tiếp và tối thiểu hóa hàm này. Tối thiểu hóa L
- Bước 4: Xây dựng sơ đồ mạch

Ví dụ: Thiết kế mạch tạo chuỗi tín hiệu tuần hoàn từ thanh ghi dịch:

$$L = 0-1-0-1-1-1-0-1.$$

- Bước 1: Tín hiệu có chiều dài $L = 8$ nên $n = \log_2 8 = 3$
- Bước 2: Chọn chu trình chuyển biến 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 hóa

Ký hiệu các flip-flop lần lượt là CBA, ta có bảng mã hóa trạng thái và hàm hồi tiếp như sau:

S	C	B	A	f_{ht}	L
S ₀	0	0	0	1	0
S ₁	0	0	1	1	1
S ₃	0	1	1	1	0
S ₇	1	1	1	0	1
S ₆	1	1	0	1	1
S ₅	1	0	1	0	1
S ₂	0	1	0	0	0
S ₄	1	0	0	0	1

Tối thiểu hóa f_{ht}:

		BA	00	01	11	10
		C	0	1	1	1
		0	1	1	1	6
		1	0	0	0	1

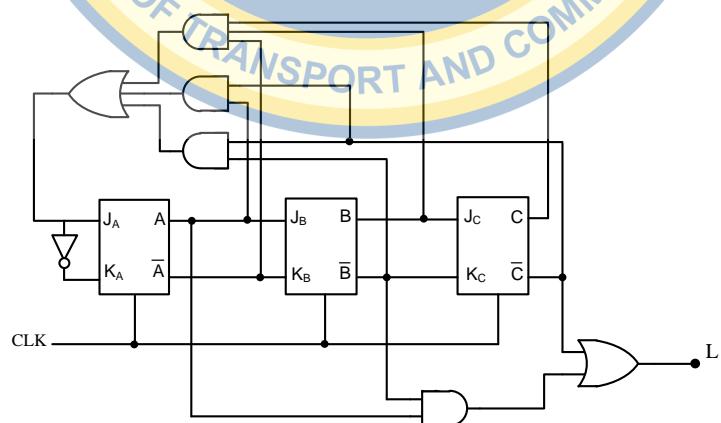
$$\text{Ta được: } f_{ht} = \overline{CB} + \overline{CA} + CBA$$

Tối thiểu hóa L:

		BA	00	01	11	10
		C	0	0	0	0
		0	0	1	0	0
		1	1	1	1	1

$$\text{Ta được: } L = C + \overline{BA}$$

Bước 4: Xây dựng sơ đồ



Hình 7.54. Sơ đồ mạch tạo tín hiệu tuần hoàn.

Ôn tập chương 7

Lý thuyết

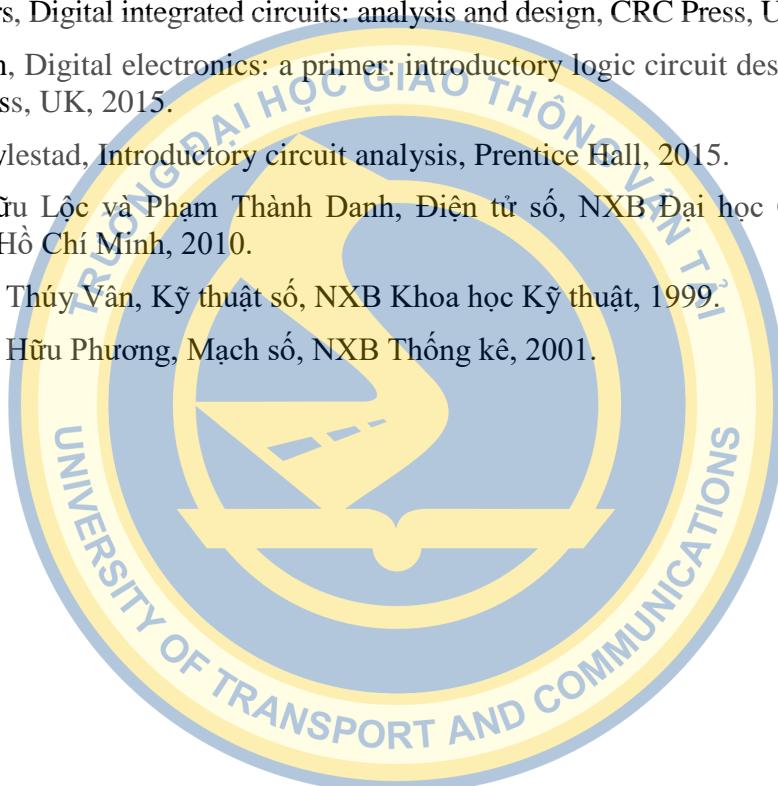
1. Trình bày khái niệm và phân loại mạch điện tử số tuần tự?
2. Nêu khái niệm và phân loại mạch flip-flop?
3. Nêu các bước thiết kế mạch điện tử số tuần tự?
4. Nêu khái niệm, phân loại và các bước thiết kế bộ đếm?
5. Trình bày khái niệm, phân loại và ứng dụng của thanh ghi dịch?

Bài tập

1. Thiết kế mạch chuyển D-FF sang JK-FF.
2. Thiết kế mạch chuyển JK-FF sang D-FF.
3. Thiết kế mạch chuyển JK-FF sang T-FF.
4. Thiết kế bộ đếm đồng bộ tăng, mã nhị phân có $K_d = 10$ dùng T-FF, xây dựng mạch đếm gồm hàng đơn vị, hàng chục, và hàng trăm từ bộ đếm thiết kế này.
5. Thiết kế bộ đếm đồng bộ tăng, mã nhị phân có $K_d = 9$ dùng JK-FF.
6. Thiết kế bộ đếm đồng bộ tăng, mã nhị phân có $K_d = 11$ dùng D-FF.
7. Thiết kế bộ đếm đồng bộ tăng, mã Gray có $K_d = 6$ dùng D-FF.
8. Thiết kế bộ đếm đồng bộ giảm, mã nhị phân có $K_d = 5$ dùng D-FF.
9. Thiết kế bộ đếm đồng bộ giảm, mã nhị phân, $K_d=10$ dùng JK-FF, xây dựng bộ đếm gồm hàng trăm, chục và hàng đơn vị từ bộ đếm này.
10. Thiết kế bộ đếm đồng bộ giảm, mã nhị phân 3 bit dùng T-FF.
11. Thiết kế bộ đếm đồng bộ giảm, mã nhị phân có $K_d = 10$ dùng JK-FF.
12. Thiết kế bộ đếm đồng bộ giảm, mã nhị phân có $K_d = 12$ dùng D-FF.
13. Thiết kế bộ đếm tăng, không đồng bộ, $K_d = 4$.
14. Thiết kế bộ đếm tăng, không đồng bộ, $K_d = 8$.
15. Thiết kế bộ đếm tăng, không đồng bộ, $K_d = 16$.
16. Thiết kế bộ đếm giảm, không đồng bộ, $K_d = 5$.
17. Thiết kế bộ đếm giảm, không đồng bộ, $K_d = 8$.
18. Thiết kế bộ đếm giảm, không đồng bộ, $K_d = 16$.
19. Cho một tín hiệu số, có tần số $f_1 = 18$ kHz. Thiết kế bộ chia tần (dùng bộ đếm) để nhận được tín hiệu có tần số $f_2 = 2$ kHz.
21. Thiết kế bộ đếm dùng thanh ghi dịch có $K_d = 6$.
22. Thiết kế bộ đếm dùng thanh ghi dịch có $K_d = 12$.
23. Thiết kế mạch tạo dãy tín hiệu tuần hoàn từ thanh ghi dịch: 10110111.
24. Thiết kế mạch tạo dãy tín hiệu tuần hoàn từ thanh ghi dịch: 11110011.

TÀI LIỆU THAM KHẢO

1. M. Rafiquzzman, Fundamentals of digital logic and microcomputer design, Wiley-Interscience, USA, 2005.
2. A. K. Maini, Digital electronics: principles, devices and applications, John Wiley & Sons, Ltd., England, 2007.
3. R.J. Tocci, N. Widmer, G. Moss, Digital systems: principles and applications, Prentice Hall, Ohio, USA, 2010.
4. IEEE standard 91a-1991: IEEE Graphic Symbols for Logic Functions IEEE publisher, USA, 1991.
5. S. Bowling, Understanding A/D converter performance specifications, Microchip Technology Inc, 2000.
6. J. E. Ayers, Digital integrated circuits: analysis and design, CRC Press, USA, 2009.
7. M. Nixon, Digital electronics: a primer: introductory logic circuit design, Imperial College Press, UK, 2015.
8. R. L. Boylestad, Introductory circuit analysis, Prentice Hall, 2015.
9. Phạm Hữu Lộc và Phạm Thành Danh, Điện tử số, NXB Đại học Công nghiệp Thành phố Hồ Chí Minh, 2010.
10. Nguyễn Thúy Vân, Kỹ thuật số, NXB Khoa học Kỹ thuật, 1999.
11. Nguyễn Hữu Phương, Mạch số, NXB Thống kê, 2001.



PHỤ LỤC 1: BẢNG MÃ ASCII 8 BIT

Hệ 10	Hệ 16	Hệ 2	Ký tự	Ý nghĩa
0	0	0	NUL	Null char
1	1	1	SOH	Start of Heading
2	2	10	STX	Start of Text
3	3	11	ETX	End of Text
4	4	100	EOT	End of Transmission
5	5	101	ENQ	Enquiry
6	6	110	ACK	Acknowledgment
7	7	111	BEL	Bell
8	8	1000	BS	Back Space
9	9	1001	HT	Horizontal Tab
10	0A	1010	LF	Line Feed
11	0B	1011	VT	Vertical Tab
12	0C	1100	FF	Form Feed
13	0D	1101	CR	Carriage Return
14	0E	1110	SO	Shift Out / X-On
15	0F	1111	SI	Shift In / X-Off
16	10	10000	DLE	Data Line Escape
17	11	10001	DC1	Device Control 1 (oft. XON)
18	12	10010	DC2	Device Control 2
19	13	10011	DC3	Device Control 3 (oft. XOFF)
20	14	10100	DC4	Device Control 4
21	15	10101	NAK	Negative Acknowledgement
22	16	10110	SYN	Synchronous Idle
23	17	10111	ETB	End of Transmit Block
24	18	11000	CAN	Cancel
25	19	11001	EM	End of Medium

26	1A	11010	SUB	Substitute
27	1B	11011	ESC	Escape
28	1C	11100	FS	File Separator
29	1D	11101	GS	Group Separator
30	1E	11110	RS	Record Separator
31	1F	11111	US	Unit Separator
32	20	100000		Space
33	21	100001	!	Exclamation mark
34	22	100010	"	Double quotes (or speech marks)
35	23	100011	#	Number
36	24	100100	\$	Dollar
37	25	100101	%	Procenttecken
38	26	100110	&	Ampersand
39	27	100111	'	Single quote
40	28	101000	(Open parenthesis (or open bracket)
41	29	101001)	Close parenthesis (or close bracket)
42	2A	101010	*	Asterisk
43	2B	101011	+	Plus
44	2C	101100	,	Comma
45	2D	101101	-	Hyphen
46	2E	101110	.	Period, dot or full stop
47	2F	101111	/	Slash or divide
48	30	110000	0	Zero
49	31	110001	1	One
50	32	110010	2	Two
51	33	110011	3	Three
52	34	110100	4	Four
53	35	110101	5	Five

54	36	110110	6	Six
55	37	110111	7	Seven
56	38	111000	8	Eight
57	39	111001	9	Nine
58	3A	111010	:	Colon
59	3B	111011	;	Semicolon
60	3C	111100	<	Less than (or open angled bracket)
61	3D	111101	=	Equals
62	3E	111110	>	Greater than (or close angled bracket)
63	3F	111111	?	Question mark
64	40	1000000	@	At symbol
65	41	1000001	A	Uppercase A
66	42	1000010	B	Uppercase B
67	43	1000011	C	Uppercase C
68	44	1000100	D	Uppercase D
69	45	1000101	E	Uppercase E
70	46	1000110	F	Uppercase F
71	47	1000111	G	Uppercase G
72	48	1001000	H	Uppercase H
73	49	1001001	I	Uppercase I
74	4A	1001010	J	Uppercase J
75	4B	1001011	K	Uppercase K
76	4C	1001100	L	Uppercase L
77	4D	1001101	M	Uppercase M
78	4E	1001110	N	Uppercase N
79	4F	1001111	O	Uppercase O
80	50	1010000	P	Uppercase P

81	51	1010001	Q	Uppercase Q
82	52	1010010	R	Uppercase R
83	53	1010011	S	Uppercase S
84	54	1010100	T	Uppercase T
85	55	1010101	U	Uppercase U
86	56	1010110	V	Uppercase V
87	57	1010111	W	Uppercase W
88	58	1011000	X	Uppercase X
89	59	1011001	Y	Uppercase Y
90	5A	1011010	Z	Uppercase Z
91	5B	1011011	[Opening bracket
92	5C	1011100	\	Backslash
93	5D	1011101]	Closing bracket
94	5E	1011110	^	Caret - circumflex
95	5F	1011111	_	Underscore
96	60	1100000	`	Grave accent
97	61	1100001	a	Lowercase a
98	62	1100010	b	Lowercase b
99	63	1100011	c	Lowercase c
100	64	1100100	d	Lowercase d
101	65	1100101	e	Lowercase e
102	66	1100110	f	Lowercase f
103	67	1100111	g	Lowercase g
104	68	1101000	h	Lowercase h
105	69	1101001	i	Lowercase i

106	6A	1101010	j	Lowercase j
107	6B	1101011	k	Lowercase k
108	6C	1101100	l	Lowercase l
109	6D	1101101	m	Lowercase m
110	6E	1101110	n	Lowercase n
111	6F	1101111	o	Lowercase o
112	70	1110000	p	Lowercase p
113	71	1110001	q	Lowercase q
114	72	1110010	r	Lowercase r
115	73	1110011	s	Lowercase s
116	74	1110100	t	Lowercase t
117	75	1110101	u	Lowercase u
118	76	1110110	v	Lowercase v
119	77	1110111	w	Lowercase w
120	78	1111000	x	Lowercase x
121	79	1111001	y	Lowercase y
122	7A	1111010	z	Lowercase z
123	7B	1111011	{	Opening brace
124	7C	1111100		Vertical bar
125	7D	1111101	}	Closing brace
126	7E	1111110	~	Equivalency sign - tilde
127	7F	1111111		Delete
128	80	10000000	€	Euro sign
129	81	10000001	,	
130	82	10000010	,	Single low-9 quotation mark
131	83	10000011	f	Latin small letter f with hook

132	84	10000100	,,	Double low-9 quotation mark
133	85	10000101	...	Horizontal ellipsis
134	86	10000110	†	Dagger
135	87	10000111	‡	Double dagger
136	88	10001000	^	Modifier letter circumflex accent
137	89	10001001	%o	Per mille sign
138	8A	10001010	Š	Latin capital letter S with caron
139	8B	10001011	<	Single left-pointing angle quotation
140	8C	10001100	Œ	Latin capital ligature OE
141	8D	10001101		
142	8E	10001110	Ž	Latin captial letter Z with caron
143	8F	10001111		
144	90	10010000		
145	91	10010001	‘	Left single quotation mark
146	92	10010010	’	Right single quotation mark
147	93	10010011	“	Left double quotation mark
148	94	10010100	”	Right double quotation mark
149	95	10010101	•	Bullet
150	96	10010110	–	En dash
151	97	10010111	—	Em dash
152	98	10011000	~	Small tilde
153	99	10011001	™	Trade mark sign
154	9A	10011010	š	Latin small letter S with caron
155	9B	10011011	>	Single right-pointing angle quotation mark
156	9C	10011100	œ	Latin small ligature oe
157	9D	10011101		

158	9E	10011110	ž	Latin small letter z with caron
159	9F	10011111	Ŷ	Latin capital letter Y with diaeresis
160	A0	10100000		Non-breaking space
161	A1	10100001	¡	Inverted exclamation mark
162	A2	10100010	¢	Cent sign
163	A3	10100011	£	Pound sign
164	A4	10100100	¤	Currency sign
165	A5	10100101	¥	Yen sign
166	A6	10100110	⋮	Pipe, Broken vertical bar
167	A7	10100111	§	Section sign
168	A8	10101000	„	Spacing diaeresis - umlaut
169	A9	10101001	©	Copyright sign
170	AA	10101010	ª	Feminine ordinal indicator
171	AB	10101011	«	Left double angle quotes
172	AC	10101100	¬	Not sign
173	AD	10101101		Soft hyphen
174	AE	10101110	®	Registered trade mark sign
175	AF	10101111	‐	Spacing macron - overline
176	B0	10110000	°	Degree sign
177	B1	10110001	±	Plus-or-minus sign
178	B2	10110010	²	Superscript two - squared
179	B3	10110011	³	Superscript three - cubed
180	B4	10110100	ˊ	Acute accent - spacing acute
181	B5	10110101	µ	Micro sign
182	B6	10110110	¶	Pilcrow sign - paragraph sign
183	B7	10110111	·	Middle dot - Georgian comma

184	B8	10111000	,	Spacing cedilla
185	B9	10111001	¹	Superscript one
186	BA	10111010	^o	Masculine ordinal indicator
187	BB	10111011	»	Right double angle quotes
188	BC	10111100	$\frac{1}{4}$	Fraction one quarter
189	BD	10111101	$\frac{1}{2}$	Fraction one half
190	BE	10111110	$\frac{3}{4}$	Fraction three quarters
191	BF	10111111	<i>i</i>	Inverted question mark
192	C0	11000000	À	Latin capital letter A with grave
193	C1	11000001	Á	Latin capital letter A with acute
194	C2	11000010	Ã	Latin capital letter A with circumflex
195	C3	11000011	Ã	Latin capital letter A with tilde
196	C4	11000100	Ä	Latin capital letter A with diaeresis
197	C5	11000101	Å	Latin capital letter A with ring above
198	C6	11000110	Æ	Latin capital letter AE
199	C7	11000111	Ç	Latin capital letter C with cedilla
200	C8	11001000	È	Latin capital letter E with grave
201	C9	11001001	É	Latin capital letter E with acute
202	CA	11001010	Ê	Latin capital letter E with circumflex
203	CB	11001011	Ë	Latin capital letter E with diaeresis
204	CC	11001100	Ì	Latin capital letter I with grave
205	CD	11001101	Í	Latin capital letter I with acute
206	CE	11001110	Î	Latin capital letter I with circumflex
207	CF	11001111	Ï	Latin capital letter I with diaeresis
208	D0	11010000	Đ	Latin capital letter ETH
209	D1	11010001	Ñ	Latin capital letter N with tilde
210	D2	11010010	Ò	Latin capital letter O with grave

211	D3	11010011	Ó	Latin capital letter O with acute
212	D4	11010100	Ô	Latin capital letter O with circumflex
213	D5	11010101	Õ	Latin capital letter O with tilde
214	D6	11010110	Ö	Latin capital letter O with diaeresis
215	D7	11010111	×	Multiplication sign
216	D8	11011000	Ø	Latin capital letter O with slash
217	D9	11011001	Ù	Latin capital letter U with grave
218	DA	11011010	Ú	Latin capital letter U with acute
219	DB	11011011	Û	Latin capital letter U with circumflex
220	DC	11011100	Ü	Latin capital letter U with diaeresis
221	DD	11011101	Ý	Latin capital letter Y with acute
222	DE	11011110	Þ	Latin capital letter THORN
223	DF	11011111	ß	Latin small letter sharp s - ess-zed
224	E0	11100000	à	Latin small letter a with grave
225	E1	11100001	á	Latin small letter a with acute
226	E2	11100010	â	Latin small letter a with circumflex
227	E3	11100011	ã	Latin small letter a with tilde
228	E4	11100100	ä	Latin small letter a with diaeresis
229	E5	11100101	å	Latin small letter a with ring above
230	E6	11100110	æ	Latin small letter ae
231	E7	11100111	ç	Latin small letter c with cedilla
232	E8	11101000	è	Latin small letter e with grave
233	E9	11101001	é	Latin small letter e with acute
234	EA	11101010	ê	Latin small letter e with circumflex
235	EB	11101011	ë	Latin small letter e with diaeresis
236	EC	11101100	ì	Latin small letter i with grave
237	ED	11101101	í	Latin small letter i with acute

238	EE	11101110	î	Latin small letter i with circumflex
239	EF	11101111	ï	Latin small letter i with diaeresis
240	F0	11110000	ð	Latin small letter eth
241	F1	11110001	ñ	Latin small letter n with tilde
242	F2	11110010	ò	Latin small letter o with grave
243	F3	11110011	ó	Latin small letter o with acute
244	F4	11110100	ô	Latin small letter o with circumflex
245	F5	11110101	õ	Latin small letter o with tilde
246	F6	11110110	ö	Latin small letter o with diaeresis
247	F7	11110111	÷	Division sign
248	F8	11111000	ø	Latin small letter o with slash
249	F9	11111001	ù	Latin small letter u with grave
250	FA	11111010	ú	Latin small letter u with acute
251	FB	11111011	û	Latin small letter u with circumflex
252	FC	11111100	ü	Latin small letter u with diaeresis
253	FD	11111101	ý	Latin small letter y with acute
254	FE	11111110	þ	Latin small letter thorn
255	FF	11111111	ÿ	Latin small letter y with diaeresis

PHỤ LỤC 2: DANH MỤC CÁC IC SỐ 74 THÔNG DỤNG

Tên	Số mạch/chip	Loại
7400	4	quad 2-input NAND gate
7401	4	quad 2-input NAND gate with open collector outputs
7402	4	quad 2-input NOR gate
7403	4	quad 2-input NAND gate with open collector outputs
7404	6	hex inverter
7405	6	hex inverter with open collector outputs
7407	6	hex buffer/driver with 30 V open collector outputs
7408	4	quad 2-input AND gate
7409	4	quad 2-input AND gate with open collector outputs
7410	3	triple 3-input NAND gate
7411	3	triple 3-input AND gate
7412	3	triple 3-input NAND gate with open collector outputs
7413	2	dual Schmitt trigger 4-input NAND gate
7414	6	hex Schmitt trigger inverter
7415	3	triple 3-input AND gate with open collector outputs
7416	6	hex inverter buffer/driver with 15 V open collector outputs
7417	6	hex buffer/driver with 15 V open collector outputs
7418	2	dual 4-input NAND gate with Schmitt trigger inputs
7419	6	hex Schmitt trigger inverter
7420	2	dual 4-input NAND gate
7421	2	dual 4-input AND gate
7422	2	dual 4-input NAND gate with open collector outputs

Tên	Số mạch/chip	Loại
7423	2	expandable dual 4-input NOR gate with strobe
7424	4	quad 2-input NAND gate gates with Schmitt trigger line-receiver inputs.
7425	2	dual 4-input NOR gate
7426	4	quad 2-input NAND gate with 15 V open collector outputs
7427	3	triple 3-input NOR gate
7428	4	quad 2-input NOR buffer
7430	1	8-input NAND gate
7431	6	hex delay elements
7432	4	quad 2-input OR gate
7433	4	quad 2-input NOR buffer with open collector outputs
7434	6	hex noninverters
7435	6	hex noninverters with open-collector outputs
7436	4	quad 2-input NOR gate
7437	4	quad 2-input NAND buffer
7438	4	quad 2-input NAND buffer with open collector outputs
7439	4	quad 2-input NAND buffer with open collector outputs, input and output terminals flipped, otherwise functionally identical to 7438
7440	2	dual 4-input NAND buffer
7441	1	BCD to decimal decoder/Nixie tube driver
7442	1	BCD to decimal decoder
7443	1	excess-3 to decimal decoder
7444	1	excess-3-Gray code to decimal decoder
7445	1	BCD to decimal decoder/driver

Tên	Số mạch/chip	Loại
7446	1	BCD to 7-segment display decoder/driver with 30 V open collector outputs
7447	1	BCD to 7-segment decoder/driver with 15 V open collector outputs
7448	1	BCD to 7-segment decoder/driver with internal pullups
7449	1	BCD to 7-segment decoder/driver with open collector outputs
7450	2	dual 2-wide 2-input AND-OR-invert gate
7451	2	dual 2-wide 2-input AND-OR-invert gate
7452	1	expandable 4-wide 2-input AND-OR gate
7453	1	expandable 4-wide 2-input AND-OR-invert gate
7454	1	3-2-2-3-input AND-OR-invert gate
7455	1	2-wide 4-input AND-OR-invert gate
7456	1	50:1 frequency divider
7457	1	60:1 frequency divider
7458	1	2-input & 3-input AND-OR gate
7459	1	2-input & 3-input AND-OR-invert gate
7460	2	dual 4-input expander
7461	3	triple 3-input expander
7462	1	3-2-2-3-input AND-OR expander
7463	6	hex current sensing interface gates
7464	1	4-2-3-2-input AND-OR-invert gate
7465	1	4-2-3-2 input AND-OR-invert gate with open collector output
7468	2	dual 4-bit decade counters
7469	2	dual 4-bit binary counters

Tên	Số mạch/chip	Loại
7470	1	AND-gated positive edge triggered J-K flip-flop with asynchronous preset and clear
74H71	1	AND-or-gated J-K master-slave flip-flop with preset
74L71	1	AND-gated R-S master-slave flip-flop with preset and clear
7472	1	AND gated J-K master-slave flip-flop with asynchronous preset and clear
7473	2	dual J-K flip-flop with asynchronous clear
7474	2	dual D positive edge triggered flip-flop with asynchronous preset and clear
7475	1	4-bit bistable latch
7476	2	dual J-K flip-flop with preset and clear
7477	1	4-bit bistable latch
74H78	2	dual positive pulse triggered J-K flip-flop with preset, common clock, and common clear
74L78	2	dual positive pulse triggered J-K flip-flop with preset, common clock, and common clear
74LS78	2	dual negative edge triggered J-K flip-flop with preset, common clock, and common clear
7479	2	dual D flip-flop
7480	1	gated full adder
7482	1	2-bit binary full adder
7483	1	4-bit binary full adder
7485	1	4-bit magnitude comparator
7486	4	quad 2-input XOR gate
7490	1	decade counter
7491	1	8-bit shift register, serial In, serial out, gated input

Tên	Số mạch/chip	Loại
7492	1	divide-by-12 counter
7493	1	4-bit binary counter
7494	1	4-bit shift register, dual asynchronous presets
7495	1	4-bit shift register, parallel in, parallel out, serial input
7496	1	5-bit parallel-in/parallel-out shift register, asynchronous preset
7497	1	synchronous 6-bit binary rate multiplier
7498	1	4-bit data selector/storage register
7499	1	4-bit bidirectional universal shift register
74100	2	dual 4-bit bistable latch
74104	1	J-K master-slave flip-flop
74106	2	dual J-K negative-edge-triggered flip-flop with preset and clear
74107	2	dual J-K flip-flop with clear
74107a	2	dual J-K negative-edge-triggered flip-flop with clear
74108	2	dual J-K negative-edge-triggered flip-flop with preset, common clear, and common clock
74110	1	AND-gated J-K master-slave flip-flop with data lockout
74112	2	dual J-K negative-edge-triggered flip-flop with clear and preset
74114	2	dual J-K negative-edge-triggered flip-flop with preset, common clock and clear
74122	1	retriggerable monostable multivibrator with clear
74124	2	dual voltage-controlled oscillator
74125	4	quad bus buffer with three-state outputs, negative enable
74128	4	quad 2-input NOR line driver

Tên	Số mạch/chip	Loại
74132	4	quad 2-input NAND Schmitt trigger
74133	1	13-input NAND gate
74134	1	12-input NAND gate with three-state output
74135	4	quad exclusive-or/NOR gate
74136	4	quad 2-input XOR gate with open collector outputs
74138	1	3 to 8-line decoder/demultiplexer
74139	2	dual 2 to 4-line decoder/demultiplexer
74144	1	decade counter/latch/decoder/7-segment driver, 15 V open collector outputs
74145	1	BCD to decimal decoder/driver
74147	1	10-line to 4-line priority encoder
74148	1	8-line to 3-line priority encoder
74150	1	16-line to 1-line data selector/multiplexer
74152	1	8-line to 1-line data selector/multiplexers
74153	2	dual 4-line to 1-line data selector/multiplexer
74154	1	4-line to 16-line decoder/demultiplexer
74157	4	quad 2-line to 1-line data selector/multiplexer, noninverting
74159	1	4-line to 16-line decoder/demultiplexer with open collector outputs
74161	1	synchronous 4-bit binary counter with asynchronous clear
74162	1	synchronous 4-bit decade counter with synchronous clear
74163	1	synchronous 4-bit binary counter with synchronous clear
74164	1	8-bit parallel-out serial shift register with asynchronous clear

Tên	Số mạch/chip	Loại
74165	1	8-bit serial shift register, parallel load, complementary outputs
74166	1	parallel-load 8-bit shift register
74167	1	synchronous decade rate multiplier
74168	1	synchronous 4-bit up/down decade counter
74169	1	synchronous 4-bit up/down binary counter
74170	1	4 by 4 register file with open collector outputs
74172	1	16-bit multiple port register file with three-state outputs
74173	4	quad D flip-flop with three-state outputs and asynchronous clear
74178	1	4-bit parallel-access shift register
74180	1	9-bit odd/even parity bit generator and checker
74181	1	4-bit arithmetic logic unit and function generator
74183	2	dual carry-save full adder
74184	1	BCD to binary converter
74185	1	6-bit binary to BCD converter
74190	1	synchronous up/down decade counter
74194	1	4-bit bidirectional universal shift register
74195	1	4-bit parallel-access shift register
74196	1	presetable decade counter/latch
74197	1	presetable binary counter/latch
74198	1	8-bit bidirectional universal shift register
74239	2	dual 2-of-4 decoder/demultiplexer, active high outputs
74247	1	BCD to 7-segment decoder/driver with 15 V open collector outputs
74248	1	BCD to 7-segment decoder/driver with Internal Pull-up

Tên	Số mạch/chip	Loại
		outputs
74283	1	4-bit binary full adder
74293	1	4-bit binary counter
74347	1	BCD-to-7 segment decoders/drivers open collector outputs, low voltage version of 7447
74348	1	8 to 3-line priority encoder with three-state outputs
74484	1	BCD-to-binary converter
74485	1	binary-to-BCD converter
74748	1	8 to 3-line priority encoder
74934	1	=ADC0829 ADC, see corresponding NSC datasheet
74937	1	=ADC3511 ADC, see corresponding NSC datasheet
74938	1	=ADC3711 ADC, see corresponding NSC datasheet
74948	1	=ADC0816 ADC, see corresponding NSC datasheet
74949	1	=ADC0808 ADC, see corresponding NSC datasheet

PHỤ LỤC 3: MỘT SỐ TRANG WEB TRA CỨU IC SỐ

<http://www.alldatasheet.com/>

<http://www.datasheetcatalog.com/>

<http://www.ic-datasheet-pdf.com/>

<http://www.ti.com>

<http://www.onsemi.com/>

<http://toshiba.semicon-storage.com/>

<http://am.renesas.com/products/security/boardid/datasheets/index.jsp>

