

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN THỊ THÚY HÀ

**BÀI GIẢNG
KỸ THUẬT SỐ**

HÀ NỘI – 12.2013

LỜI NÓI ĐẦU

Cùng với sự tiến bộ của khoa học và công nghệ, các thiết bị điện tử đang và sẽ tiếp tục được ứng dụng ngày càng rộng rãi và mang lại hiệu quả cao trong hầu hết các lĩnh vực kinh tế kỹ thuật cũng như đời sống xã hội.

Việc xử lý tín hiệu trong các thiết bị điện tử hiện đại đều dựa trên cơ sở nguyên lý số. Bởi vậy việc hiểu sâu sắc về điện tử số là điều không thể thiếu được đối với kỹ sư ngành Điện - Điện tử, Điện tử - Viễn thông, cũng như CNTT. Nhu cầu hiểu biết về Điện tử số không phải chỉ riêng đối với các kỹ sư các ngành nói trên mà còn cần thiết đối với nhiều cán bộ kỹ thuật các chuyên ngành khác có ứng dụng điện tử.

Bài giảng này giới thiệu một cách hệ thống các phần tử cơ bản trong các mạch điện tử số kết hợp với các mạch điển hình, giải thích các khái niệm cơ bản về cổng điện tử số, các phương pháp phân tích và thiết kế mạch logic cơ bản.

Bài giảng bao gồm các kiến thức cơ bản về mạch cổng logic, cơ sở đại số logic, mạch logic tổ hợp, các trigơ, mạch logic tuần tự, các mạch phát xung và tạo dạng xung, các bộ nhớ thông dụng. Bài giảng gồm 4 chương, trước và sau mỗi chương đều có phần giới thiệu và phần tóm tắt để giúp người học dễ nắm bắt kiến thức. Ngoài ra bài giảng còn có các câu hỏi ôn tập để người học kiểm tra mức độ nắm bắt kiến thức sau khi học mỗi chương. Trên cơ sở các kiến thức căn bản, bài giảng đã cố gắng tiếp cận các vấn đề hiện đại, đồng thời liên hệ với thực tế kỹ thuật.

Bài giảng gồm có 4 chương được bố cục như sau:

Chương 1: Hệ đếm.

Chương 2: Cổng logic

Chương 3: Mạch logic tổ hợp.

Chương 4: Mạch logic tuần tự.

Do thời gian có hạn nên bài giảng này không tránh khỏi thiếu sót, rất mong bạn đọc góp ý. Các ý kiến xin gửi về Bộ môn Kỹ thuật điện tử - Khoa Kỹ thuật Điện tử 1- Học viện Công nghệ Bưu chính viễn thông.

Xin trân trọng cảm ơn!

Tác giả.

THUẬT NGỮ VIẾT TẮT

ALU	Arithmetic Logic Unit	Đơn vị tính logic và số học
ANSI	American National Standards Institute	Viện tiêu chuẩn Quốc gia Hoa kỳ
BCD	Binary Coded Decimal	Số thập phân mã hóa theo nhị phân
Bit	Binary Digit	Cột số nhị phân (Đơn vị thông tin nhỏ nhất)
Bus	Một số đường dây dẫn mắc song song dùng cho việc truyền các tín hiệu địa chỉ, dữ liệu và điều khiển	
Byte	Một nhóm gồm 8 bit	
C, CLK	Clock	Xung đồng hồ (Xung nhịp)
Cache	Bộ nhớ trung gian	
CAS	Column Address Select	Chọn địa chỉ cột
CLR	Clear	Xóa
CMOS	Complementary Metal Oxide Semiconductor	Vật liệu bán dẫn gồm hai linh kiện NMOS và PMOS mắc tổ hợp với nhau
CPU	Central Processing Unit	Đơn vị xử lý trung tâm
Crumb	2 bit	
CS	Chip Select	Chọn chip
DDL	Diode-Diode Logic	Cổng logic chứa các diode
Deckle	10 bit	
DLL	Delay_Locked Loop	Vòng khóa pha trễ
DEMUX	DeMultiplexer	Bộ phân kênh
DRAM	Dynamic RAM	RAM động
DTL	Diode Transistor Logic	Cổng logic chứa các diode và transistor
Dynner	32 bit	
ECL	Emitter Couple Logic	Cổng logic ghép cực Emitter
EEPROM	Electrically Erasable ROM	ROM lập trình được và xóa được bằng điện
EPROM	Erasable ROM	ROM lập trình được và xóa được bằng tia cực tím
FET	Field Effect Transistor	Transistor hiệu ứng trường
H	High	Mức logic cao
IC	Integrated Circuit	Mạch tích hợp
IEEE	Institute of Electrical and Electronics Engineers	Viện kỹ thuật Điện và điện tử
ISP	In- System Programming	Lập trình trên hệ thống
L	Low	Mức logic thấp
Latch	Bộ chốt	
LCD	Liquid Crystal Display	Hiển thị tinh thể lỏng
LED	Light Emitting Diode	Điốt phát quang
LSB	Least Significant Bit	Bit có ý nghĩa bé nhất
Maxterm	Thừa số lớn nhất	
Minterm	Số hạng nhỏ nhất	

MOSFET	Metal Oxide Semiconductor FET	<i>FET có cực cửa cách ly bằng lớp ooxit kim loại</i>
MROM	Mask ROM	<i>ROM được chế tạo bằng phương pháp che mặt nạ</i>
MSB	Most Significant Bit	<i>Bit có ý nghĩa lớn nhất</i>
MSI	Medium Scale Integrated	<i>Mức độ tích hợp trung bình</i>
MUX	Multiplexer	<i>Bộ ghép kênh</i>
Nibble	4 bit	
NMOS	N – chanel MOS	<i>Transistor trường kênh dẫn N</i>
PMOS	P – chanel MOS	<i>Transistor trường kênh dẫn P</i>
PRE	Preset	<i>Tái lập</i>
RAM	Random Access Memory	<i>Bộ nhớ truy cập ngẫu nhiên</i>
RAS	Row Address Select	<i>Chọn địa chỉ hàng</i>
RBI	Riple Blanking Input	<i>Đầu vào xóa nối tiếp</i>
RBO	Riple Blanking Output	<i>Đầu ra xóa nối tiếp</i>
ROM	Read Only Memory	<i>Bộ nhớ chỉ đọc</i>
RTL	Resistance Transistor Logic	<i>Cổng logic dùng điện trở và transistor</i>
SRAM	Static RAM	<i>RAM tĩnh</i>
SSI	Small Scale Integrated	<i>Mức độ tích hợp trung bình</i>
TTL	Transistor – Transistor Logic	<i>Cổng logic dùng Transistor</i>
VLSI	Very Large Scale Integrated	<i>Mức độ tích hợp rất lớn</i>

MỤC LỤC

LỜI NÓI ĐẦU	i
THUẬT NGỮ VIẾT TẮT	ii
MỤC LỤC	iv
CHƯƠNG 1: HỆ ĐẾM.....	1
GIỚI THIỆU	1
1.1. BIỂU DIỄN SỐ.....	1
1.1.1 Hệ thập phân	1
1.1.2 Hệ nhị phân.....	2
1.1.3 Hệ 8 (bát phân) và hệ 16 (thập lục phân).....	4
1.2. CHUYỂN ĐỔI CƠ SỐ GIỮA CÁC HỆ ĐẾM	6
1.2.1. Chuyển đổi từ hệ cơ số 10 sang các hệ khác.....	6
1.2.2. Đổi một biểu diễn trong hệ bất kì sang hệ thập phân.....	8
1.2.3. Đổi các số từ hệ nhị phân sang hệ cơ số 8 và 16.....	8
1.3 MỘT SỐ PHÉP TOÁN.....	9
1.3.1. Số nhị phân có dấu	9
1.3.2 Các phép cộng và trừ số nhị phân có dấu	10
1.3.3 Phép nhân.	12
TÓM TẮT	12
CÂU HỎI ÔN TẬP.....	12
CHƯƠNG 2. CỔNG LOGIC	15
GIỚI THIỆU CHUNG	15
2.1 . CÁC HÀM CHUYỂN MẠCH CƠ BẢN.....	15
2.1.1. Hàm AND.....	15
2.1.2. Hàm OR.....	16
2.1.3. Hàm NOT.	16
2.2. MỘT SỐ ĐỊNH LÝ CƠ BẢN TRONG ĐẠI SỐ BOOLE.....	17
2.2.1. Các định lý cơ bản.....	17
2.2.2 Các định luật cơ bản:	17
2.2.3. Ba quy tắc về đẳng thức	17
2.3 CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM BOOLE	18
2.3.1 Bảng trạng thái.....	18
2.3.2 Phương pháp đại số.....	19
2.3.3 Phương pháp bảng Các nô (bảng Karnaugh hay phương pháp hình học).	21
2.3. CÁC PHƯƠNG PHÁP TỐI THIỂU HÓA (RÚT GỌN HÀM).....	23
2.3.1. Phương pháp đại số	23
2.3.2 Phương pháp bảng Các nô	23
2.3.3. Rút gọn hàm logic hàm tùy chọn.....	26
2.4 CỔNG LOGIC	27
2.4.1 Cổng logic cơ bản	27
2.4.2. Logic dương và logic âm.....	29
2.4.3. Một số cổng ghép thông dụng.....	30

2.4.4 Tính đa chức năng của cổng NAND, NOR.....	33
TÓM TẮT	35
CÂU HỎI ÔN TẬP	36
CHƯƠNG 3: MẠCH LOGIC TỔ HỢP	40
GIỚI THIỆU CHUNG	40
3.1 KHÁI NIỆM CHUNG.....	40
3.1.1. Đặc điểm cơ bản của mạch tổ hợp.....	40
3.1.2. Phương pháp biểu diễn chức năng logic	40
3.2 PHÂN TÍCH MẠCH LOGIC TỔ HỢP	41
3.3 THIẾT KẾ MẠCH LOGIC TỔ HỢP	42
3.4. MẠCH MÃ HOÁ VÀ GIẢI MÃ	44
3.4.1 .Một số loại mã thông dụng	44
3.4.2. Các mạch mã hoá:	46
3.4.3. Các bộ giải mã	50
3.4.4. Các bộ biến mã.....	54
3.5. BỘ HỢP KÊNH VÀ PHÂN KÊNH	56
3.5.1 Bộ hợp kênh (MUX-Multiplexer)	56
3.5.2. Bộ phân kênh (Demultiplexer: DMUX)	59
3.5.3. Một số ứng dụng của bộ ghép kênh và phân kênh.....	61
3.6. MẠCH SỐ HỌC	63
3.6.1. Mạch cộng.	63
3.6.2. Mạch trừ	66
3.6.3. Mạch cộng, trừ theo bù 1 và bù 2.....	68
3.6.4. Mạch so sánh.	69
3.7. MẠCH PHÁT HIỆN SAI.....	71
3.7.1. Mạch tạo và kiểm tra chẵn lẻ.	71
3.7.2 Mạch tạo mã và giải mã Hamming.....	73
3.8. ĐƠN VỊ SỐ HỌC VÀ LOGIC (ALU).	76
TÓM TẮT	77
CÂU HỎI ÔN TẬP	77
CHƯƠNG 4. MẠCH LOGIC TUẦN TỰ.....	79
GIỚI THIỆU.....	79
NỘI DUNG	79
4.1. KHÁI NIỆM CHUNG VÀ MÔ HÌNH TOÁN HỌC.....	79
4.1.1. Khái niệm chung	79
4.1.2. Mô hình toán học	79
4.2. PHẦN TỬ NHỚ CỦA MẠCH TUẦN TỰ	80
4.2.1. Các loại Trơơ	80
4.2.2. Đầu vào không đồng bộ của trơơ.	90
4.2.3. Chuyển đổi giữa các loại trơơ.	90
4.3. PHÂN TÍCH MẠCH TUẦN TỰ	97
4.3.1. Các bước phân tích mạch tuần tự đồng bộ.....	97
4.3.2. Các bước phân tích mạch tuần tự không đồng bộ	98
4.4. BỘ ĐẾM.....	98
4.4.1. Phân tích bộ đếm.	98
4.4.2. Thiết kế bộ đếm	117

4.4.3. Giới thiệu một số IC đếm	124
4.5. Bộ ghi dịch (Shift Register)	127
4.5.1. Giới thiệu chung:.....	127
4.5.2. Bộ ghi song song.....	128
4.5.3. Bộ ghi dịch nối tiếp.....	129
4.5.4. Ứng dụng của bộ ghi dịch.....	130
4.6. Thanh chốt dữ liệu (Latch).....	135
TÓM TẮT	137
CÂU HỎI ÔN TẬP CHƯƠNG 4.....	137
TÀI LIỆU THAM KHẢO	140

PTIT

CHƯƠNG 1: HỆ ĐẾM

GIỚI THIỆU

Khi nói đến số đếm, người ta thường nghĩ ngay đến hệ thập phân với 10 chữ số được ký hiệu từ 0 đến 9. Hệ thập phân là một trong nhiều hệ đếm. Thông thường người ta quen lấy số 10 làm gốc nhưng trên thực tế một số nguyên dương bất kỳ nào cũng có thể lấy làm gốc cho hệ đếm.

Máy tính hiện đại thường không sử dụng số thập phân, mà hay sử dụng số nhị phân với hai ký hiệu là 0 và 1. Khi biểu diễn các số nhị phân rất lớn, người ta thay nó bằng các số bát phân (Octal) và thập lục phân (Hexadecimal).

Trong chương này không chỉ trình bày các hệ thập phân, hệ nhị phân, hệ bát phân, hệ thập lục phân và còn nghiên cứu cách chuyển đổi giữa các hệ đếm, số nhị phân có dấu.

1.1. BIỂU DIỄN SỐ

Tính chất quan trọng nhất của một hệ thống số là sử dụng một dãy các ký tự để thể hiện một con số trong hệ. Giá trị của một số được thể hiện thông qua giá trị và vị trí của mỗi ký tự, vị trí này có trọng số tăng dần tính từ phải qua trái. Số ký tự được dùng gọi là cơ số của hệ và ký hiệu là r . Trọng số của một hệ đếm bất kỳ sẽ bằng r^i , với i là một số nguyên dương hoặc âm.

Trong kỹ thuật số có bốn hệ thống số quan trọng được sử dụng: hệ thập phân, hệ nhị phân, hệ bát phân (hệ tám) và hệ thập lục phân (hệ mười sáu).

Trong toán học, người ta gọi hệ đếm theo cơ số của chúng. Ví dụ: Hệ nhị phân = Hệ cơ số 2, Hệ thập phân = Hệ cơ số 10...

Dưới đây, trình bày một số hệ đếm thông dụng.

1.1.1 Hệ thập phân

Hệ thập phân có 10 ký hiệu từ 0 đến 9 nên còn gọi là hệ cơ số 10. Khi ghép các ký hiệu với nhau sẽ được một biểu diễn số.

Ví dụ: 1265,34 là biểu diễn số trong hệ thập phân:

$$1265,34 = 1 \times 10^3 + 2 \times 10^2 + 6 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2}$$

Trong đó: 10^n là **trọng số của hệ**; các hệ số nhân (1, 2, 6...) chính là **ký hiệu** của hệ.

Một số dương N bất kỳ trong hệ thập phân có thể khai triển thành:

$$N_{10} = \sum a_i 10^i \quad (1.1)$$

trong đó, N_{10} : biểu diễn bất kỳ theo hệ 10, a_i hệ số nhân có giá trị từ 0 đến 9.

Nếu phần nguyên có n chữ số thì $i = (n-1) \div 0$;

Nếu phần phân số có m chữ số thì $i = -1 \div -m$;

Nếu dùng r thay cho cơ số 10 thì biểu thức (1.1) có dạng tổng quát cho mọi hệ đếm.

Biểu diễn số tổng quát:

$$N_{10} = \sum_{i=-m}^{n-1} a_i r^i \quad (1.2)$$

Ưu điểm của hệ thập phân là tính tiện dụng nên nó được sử dụng trong đời sống hàng ngày. Đây là hệ mà con người dễ nhận biết nhất. Ngoài ra, nhờ có nhiều ký hiệu nên khả năng biểu diễn của hệ rất lớn, cách biểu diễn gọn, tốn ít thời gian viết và đọc.

Nhược điểm chính của hệ là do có nhiều ký hiệu nên việc thể hiện bằng thiết bị kỹ thuật sẽ khó khăn và phức tạp.

1.1.2 Hệ nhị phân

1.1.2.1. Tổ chức hệ nhị phân

Hệ nhị phân (*Binary number systems*) còn gọi là hệ cơ số hai, chỉ gồm hai ký hiệu 0 và 1, cơ số của hệ là 2, trọng số của hệ là 2^n . Hệ đếm này được sử dụng rộng rãi trong mạch số.

Trong hệ nhị phân, mỗi chữ số chỉ lấy 2 giá trị hoặc 0 hoặc 1 và được gọi tắt là "bit" (**B**inary **d**igit). Như vậy, bit là số nhị phân 1 chữ số. Số bit tạo thành độ dài biểu diễn của một số nhị phân.

- **Crumb, Tydbit, hoặc Tayste:** 2 bit.
- **Nibble, hoặc Nybble:** 4 bit.
- **Byte:** 8 bit.
- **Word:** (phụ thuộc vào từng hệ thống)

Các giá trị $2^{10} = 1024$ được gọi là 1Kbit, $2^{20} = 1048576$ - Mêga Bit ...

Bit tận cùng bên phải gọi là bit có trọng số bé nhất (LSB – **L**east **S**ignificant **B**it) và bit tận cùng bên trái gọi là bit có trọng số lớn nhất (MSB - **M**ost **S**ignificant **B**it).

Biểu diễn nhị phân dạng tổng quát :

$$N_2 = \sum_{i=-m}^{n-1} a_i 2^i \quad (1.3)$$

Trong đó, a_i là hệ số nhân của hệ có giá trị bằng 0 hoặc 1. Các chỉ số của hệ số đồng thời cũng bằng lũy thừa của trọng số tương ứng.

Ví dụ :

1	1	0.	0	0	→ số nhị phân phân số
2^2	2^1	2^0	2^{-1}	2^{-2}	→ trọng số tương ứng.

Ưu điểm chính của hệ nhị phân là chỉ có hai ký hiệu nên rất dễ thể hiện bằng các thiết bị cơ, điện. Các máy vi tính và các hệ thống số đều dựa trên cơ sở hoạt động nhị phân (2 trạng thái). Do đó, hệ nhị phân được xem là ngôn ngữ của các mạch logic, các thiết bị tính toán hiện đại - ngôn ngữ máy.

Nhược điểm của hệ là biểu diễn dài, do đó thời gian viết, đọc dài.

1.1.2.2. Các phép tính trong hệ nhị phân

a. Phép cộng

Qui tắc cộng hai số nhị phân giống như phép cộng trong hệ thập phân, tức là cộng các bit có cùng trọng số theo quy tắc sau.

Nguyên tắc cộng nhị phân là : $0 + 0 = 0$, $1 + 0 = 1$, $1 + 1 = 10$ ($10_2 = 2_{10}$).

Ví dụ:

$$\begin{array}{r} 1\ 0\ 1_2 \quad (5_{10}) \\ + 1\ 0\ 0_2 \quad (4_{10}) \\ \hline 1\ 0\ 0\ 1_2 \quad (9_{10}) \end{array} \quad \begin{array}{r} (13_{10}) \quad 1\ 1\ 0\ 1_2 \\ (11_{10}) \quad + 1\ 0\ 1\ 1_2 \\ \hline (24_{10}) \quad 1\ 1\ 0\ 0\ 0_2 \end{array} \quad \begin{array}{r} (4,375_{10}) \quad 1\ 0\ 0,0\ 1\ 1_2 \\ (3,750_{10}) \quad + \quad 1\ 1,1\ 1\ 0_2 \\ \hline (8,125_{10}) \quad 1\ 0\ 0\ 0,0\ 0\ 1_2 \end{array}$$

b. Phép trừ

Qui tắc trừ hai bit nhị phân cho nhau như sau :

$0 - 0 = 0$; $1 - 1 = 0$; $1 - 0 = 1$; $10 - 1 = 1$ (mượn 1)

Ví dụ:

$$\begin{array}{r} 1\ 1\ 0\ 1_2 \quad (13_{10}) \\ - 1\ 1\ 0_2 \quad (6_{10}) \\ \hline 0\ 1\ 1\ 1_2 \quad (7_{10}) \end{array} \quad \begin{array}{r} 25_{10} \quad 1\ 1\ 0\ 0\ 1_2 \\ (11_{10}) \quad - 1\ 0\ 1\ 1_2 \\ \hline (14_{10}) \quad 0\ 1\ 1\ 1\ 0_2 \end{array} \quad \begin{array}{r} (5,3125_{10}) \quad 1\ 0\ 1,0\ 1\ 0\ 1_2 \\ (2,8125_{10}) \quad - \quad 1\ 0,1\ 1\ 0\ 1_2 \\ \hline (2,5000_{10}) \quad 0\ 1\ 0,1\ 0\ 0\ 0_2 \end{array}$$

c. Phép nhân

Qui tắc nhân hai bit nhị phân như sau:

$0 \times 0 = 0$, $0 \times 1 = 0$, $1 \times 0 = 0$, $1 \times 1 = 1$

Phép nhân hai số nhị phân cũng được thực hiện giống như trong hệ thập phân.

Chú ý : Phép nhân có thể thay bằng phép dịch trái và cộng liên tiếp.

Ví dụ:

$$\begin{array}{r} 1\ 0\ 0\ 1_2 \quad (9_{10}) \\ \times \quad 1\ 1_2 \quad (3_{10}) \\ \hline 1\ 0\ 0\ 1 \\ + 1\ 0\ 0\ 1 \\ \hline 1\ 1\ 0\ 1\ 1_2 \quad (27_{10}) \end{array} \quad \begin{array}{r} (5,5_{10}) \quad 1\ 0\ 1,1_2 \\ (2_{10}) \quad \times \quad 1\ 0_2 \\ \hline 0\ 0\ 0\ 0 \\ + 1\ 0\ 1\ 1 \\ \hline (11_{10}) \quad 1\ 0\ 1\ 1,0 \end{array}$$

d. Phép chia

Phép chia nhị phân cũng tương tự như phép chia số thập phân.

Ví dụ:

$$\begin{array}{r} 1\ 0\ 0\ 1_2 \overline{) 1\ 1_2} \\ - 1\ 1 \\ \hline 0\ 0\ 1\ 1 \\ - 1\ 1 \\ \hline 0\ 0\ 0\ 0 \end{array}$$

Trong trường hợp số bị chia nhỏ hơn số chia, cách thực hiện giống như ví dụ trên, kết quả thương số chỉ có phần lẻ sau dấu phẩy, mỗi lần thêm một số 0 vào số bị chia cần ghi một số 0 vào thương số phía sau dấu phẩy cho tới khi số bị chia “lớn hơn” số chia. Phép tính này tương tự như trong hệ thập phân.

1.1.3 Hệ 8 (bát phân) và hệ 16 (thập lục phân)

1.1.3.1 Hệ 8 (Octal number systems)

a. Tổ chức của hệ.

Hệ 8 gồm 8 ký hiệu: 0, 1, 2, 3, 4, 5, 6 và 7 nên cơ số của hệ là 8. Hệ cơ số 8 có thể được biểu diễn thành 2^3 . Do đó, mỗi ký hiệu trong hệ 8 có thể thay thế bằng 3 bit trong hệ nhị phân.

Dạng biểu diễn tổng quát của hệ bát phân như sau:

$$N_8 = \sum_{i=-m}^{n-1} a_i 8^i \quad (1.4)$$

Trong đó, a là hệ số nhân lấy các giá trị từ 0 đến 7.

b. Các phép tính trong hệ 8.

Phép cộng.

Phép cộng trong hệ bát phân được thực hiện tương tự như trong hệ thập phân. Khi kết quả của việc cộng hai hoặc nhiều chữ số cùng trọng số lớn hơn hoặc bằng 8 phải nhớ lên chữ số có trọng số lớn hơn kế tiếp.

Ví dụ:

$$\begin{array}{r} \text{a)} \quad 127_8 \\ + 375_8 \\ \hline 524_8 \end{array}$$

$$\begin{array}{r} \text{b)} \quad 632_8 \\ + 543_8 \\ \hline 1405_8 \end{array}$$

Trong ví dụ a) tiến hành cộng như sau: $7 + 5 = 12_{10}$; trong hệ 8 không có số 12 nên phải chia 12 cho 8, số dư viết xuống tổng tương ứng với trọng số đó, thương số nhớ lên trọng số kế tiếp; tức là $12 : 8 = 1$ dư 4, số 4 được viết xuống tổng; tại trọng số kế tiếp $2 + 7 + 1(\text{nhớ}) = 10$; sau đó lấy $10 : 8 = 1$ dư 2, viết 2 xuống tổng và số 1 được nhớ lên trọng số kế tiếp; cuối cùng, lấy $1 + 3 + 1(\text{nhớ}) = 5$.

Phép trừ.

Phép trừ cũng được tiến hành như trong hệ thập phân. Khi mượn 1 ở có trọng số lớn hơn kế tiếp thì chỉ cần cộng thêm 8_{10} .

$$\begin{array}{r} \text{a)} \quad 623_8 \\ - 375_8 \\ \hline 226_8 \end{array}$$

$$\begin{array}{r} \text{b)} \quad 452, 5_8 \\ - 343, 7_8 \\ \hline 1046, 6_8 \end{array}$$

Trong ví dụ a) tiến hành trừ như sau: $3 + 8$ (mượn ở trọng số kế tiếp) $- 5 = 6$; tại trọng số kế tiếp $2 - 7 - 1 + 8$ (mượn) $= 2$; cuối cùng lấy $6 - 3 - 1 = 2$.

Thông thường, các phép tính trong hệ 8 ít được sử dụng.

1.1.3.2. Hệ 16

a. Tổ chức của hệ.

Hệ 16 hay hệ thập lục phân hay hệ Hexa (*Hexadecimal number systems*). Hệ gồm 16 ký hiệu là 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F nên còn gọi là hệ cơ số 16.

Trong đó, $A = 10_{10}$, $B = 11_{10}$, $C = 12_{10}$, $D = 13_{10}$, $E = 14_{10}$, $F = 15_{10}$.

Cơ số của hệ là 16, số 16 có thể được biểu diễn bằng 2^4 . Do vậy, có thể dùng một từ nhị phân 4 bit (từ 0000 đến 1111) để biểu thị các ký hiệu thập lục phân. Dạng biểu diễn tổng quát:

$$N_{16} = \sum_{i=-m}^{n-1} a_i 16^i \quad (1.5)$$

Trong đó, a là hệ số nhân lấy các giá trị từ 0 đến F.

b. Các phép tính trong hệ cơ số 16.

Phép cộng.

Khi tổng hai chữ số lớn hơn 15, lấy tổng chia cho 16. Số dư được viết xuống chữ số tổng và thương số được nhớ lên chữ số có trọng số lớn hơn kế tiếp. Nếu các chữ số là A, B, C, D, E, F thì trước hết, phải đổi chúng về giá trị thập phân tương ứng rồi mới tiến hành cộng.

Ví dụ:

$$\begin{array}{rcl} \text{a)} & 695_{16} & \\ & + 875_{16} & \\ \hline & F0A_{16} & \\ \text{b)} & 4A, 5_{16} & \\ & + 3B, 7_{16} & \\ \hline & 85, C_{16} & \end{array}$$

Trong ví dụ a) tiến hành cộng như sau: $5 + 5 = 10_{10} = A_{16}$; sau đó: $9 + 7 = 16$, trong hệ 16 không có số 16 nên phải chia 16 cho 16, số dư viết xuống tổng tương ứng với trọng số đó, thương số nhớ lên trọng số kế tiếp; tức là $16 : 16 = 1$ dư 0, số 0 được viết xuống tổng, số 1 được cộng vào trọng số kế tiếp; tại trọng số kế tiếp $6 + 8 + 1(\text{nhớ}) = 15_{10} = F_{16}$;

Phép trừ: Khi trừ một số bé hơn cho một số lớn hơn cũng mượn 1 ở cột kế tiếp bên trái, nghĩa là cộng thêm 16 rồi mới trừ.

$$\begin{array}{rcl} \text{a)} & E95_{16} & \\ & - 87C_{16} & \\ \hline & 619_{16} & \\ \text{b)} & 4A, 5_{16} & \\ & - 3B, 7_{16} & \\ \hline & 0E, E_{16} & \end{array}$$

Trong ví dụ a) tiến hành trừ như sau: $5 + 16$ (mượn ở trọng số kế tiếp) $- 12 (C_{16}) = 9$; tại trọng số kế tiếp $9 - 7 - 1 = 1$; cuối cùng lấy $14 (E_{16}) - 8 = 6$.

Phép nhân.

Muốn thực hiện phép nhân trong hệ 16 phải đổi các số trong mỗi thừa số về thập phân, nhân hai số với nhau. Sau đó, đổi kết quả về hệ 16.

Bảng 1-2 biểu diễn 16 số đầu tiên trong các hệ số đếm.

Hệ thập phân	Hệ nhị phân	Hệ bát phân	Hệ thập lục phân
0	0 0 0 0	0	0
1	0 0 0 1	1	1
2	0 0 1 0	2	2
3	0 0 1 1	3	3
4	0 1 0 0	4	4
5	0 1 0 1	5	5
6	0 1 1 0	6	6
7	0 1 1 1	7	7
8	1 0 0 0	10	8
9	1 0 0 1	11	9
10	1 0 1 0	12	A
11	1 0 1 1	13	B
12	1 1 0 0	14	C
13	1 1 0 1	15	D
14	1 1 1 0	16	E
15	1 1 1 1	17	F

Bảng 1-2. Biểu diễn số của 4 hệ đếm thường dùng.

1.2. CHUYỂN ĐỔI CƠ SỐ GIỮA CÁC HỆ ĐẾM

1.2.1. Chuyển đổi từ hệ cơ số 10 sang các hệ khác

Để thực hiện việc đổi một số thập phân đầy đủ sang các hệ khác phải chia ra hai phần: phần nguyên và phần số.

Đối với phần nguyên:

Ví dụ, đổi từ hệ thập phân sang hệ nhị phân:

Trong đẳng thức sau, vế trái là số thập phân, vế phải là số nhị phân:

$$\begin{aligned} N_{10} &= a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 \\ &= 2(a_n 2^{n-1} + a_{n-1} 2^{n-2} + \dots + a_1) + a_0 \end{aligned} \quad (1.6)$$

Do a_i có giá trị bằng 0 hoặc 1, nên có:

$$\begin{aligned} \frac{N_{10} - a_0}{2} &= a_n 2^{n-1} + a_{n-1} 2^{n-2} + \dots + a_1 \\ &= 2(a_n 2^{n-2} + a_{n-1} 2^{n-3} + \dots + a_2) + a_1 \end{aligned} \quad (1.7)$$

Từ biểu thức (1.6) và (1.7) nhận thấy:

Bit đầu tiên của số nhị phân là a_0 bằng số dư khi chia N_{10} cho 2. Bit tiếp theo của số nhị phân là a_1 bằng số dư khi chia thương số của phép chia trước cho 2.

Tương tự như vậy để tìm toàn bộ các bit của số nhị phân.

Đối với việc đổi từ hệ thập phân sang hệ 8 và 16 cũng thực hiện tương tự như vậy.

Tóm lại, để chuyển từ hệ 10 sang các hệ khác, chia liên tiếp phần nguyên của số thập phân cho cơ số của hệ cần chuyển đến, số dư sau mỗi lần chia viết đảo ngược trật tự là kết quả cần tìm. Phép chia dừng lại khi kết quả lần chia cuối cùng bằng 0.

Ví dụ 1: Đổi số 35_{10} sang số nhị phân.

35	2	=17	Dư 1	a_0
17	2	=8	Dư 1	a_1
8	2	=4	Dư 0	a_2
4	2	=2	Dư 0	a_3
2	2	=1	Dư 0	a_4
1	2	=0	Dư 1	a_5

Vậy : $35_{10} = 100011_2$

Ví dụ 2: Đổi số 35_{10} sang hệ 8.

35	8	=4	Dư 3	a_0
4	8	=0	Dư 4	a_1

Vậy : $35_{10} = 43_8$

Ví dụ 3: Đổi số 35_{10} sang hệ 16.

35	16	=2	Dư 3	a_0
2	16	=0	Dư 2	a_1

Vậy : $35_{10} = 23_{16}$

Đổi với phần phân số :

Ví dụ, đổi từ hệ thập phân sang hệ nhị phân:

Trong đẳng thức sau, vế trái là số thập phân, vế phải là số nhị phân:

$$N_{10} = a_{-1} 2^{-1} + a_{-2} 2^{-2} + \dots + a_{-m} 2^{-m} \quad (1.8)$$

Nhân 2 vế với 2, được:

$$2N_{10} = a_{-1} + (a_{-2} 2^{-1} + a_{-3} 2^{-2} + \dots + a_{-m} 2^{-m+1}) \quad (1.9)$$

a_{-1} trở thành phần nguyên của phần nguyên của vế phải. Phần phân số còn lại là:

$$2N_{10} - a_{-1} = a_{-2} + (a_{-3} 2^{-1} + a_{-4} 2^{-2} + \dots + a_{-m} 2^{-m+2}) \quad (1.10)$$

Nếu tiếp tục nhân 2 vế với 2, được a_{-2} là phần nguyên của vế phải (của tích số lần thứ 2):

$$2[2N_{10} - a_{-1}] = a_{-2} + (a_{-3} 2^{-1} + a_{-4} 2^{-2} + \dots + a_{-m} 2^{-m+2}) \quad (1.11)$$

Tương tự như vậy, tìm được toàn bộ các bit của số nhị phân.

Đối với việc đổi từ phần phân số của hệ thập phân sang hệ 8 và 16 cũng thực hiện tương tự như vậy.

Tóm lại, khi chuyển phần phân số, thực hiện như sau: nhân liên tiếp phần phân số của số thập phân với cơ số của hệ cần chuyển đến, phần nguyên thu được sau mỗi lần nhân, viết

tuần tự là kết quả cần tìm. Phép nhân dừng lại khi phần phân số triệt tiêu hoặc cho đến khi đạt được số bit nằm sau dấu phẩy theo yêu cầu (trong trường hợp phép nhân không hội tụ về 0).

Ví dụ 1: Đổi số $35,375_{10}$ sang số nhị phân.

Phần nguyên vừa thực hiện ở ví dụ a), do đó chỉ cần đổi phần phân số $0,375$.

0,375	x	2	= 0,75	Phần nguyên = 0	a ₁
0,75	x	2	= 1,5	Phần nguyên = 1	a ₂
0,5	x	2	= 1,0	Phần nguyên = 1	a ₃
0,0	x	2	= 0	Phần nguyên = 0	a ₄

Kết quả : $0,375_{10} = 0,0110_2$

Sử dụng phần nguyên đã có ở ví dụ 1) được : $35,375_{10} = 10011,0110_2$

Ví dụ 2: Đổi số $0,375_{10}$ sang hệ 8.

0,375	x	8	= 3,0	Phần nguyên = 3	a ₁
0,0	x	8	= 0	Phần nguyên = 0	a ₂

Kết quả : $0,375_{10} = 0,3_8$

Ví dụ 3: Đổi số $0,375_{10}$ sang hệ 16.

0,375	x	16	= 6,0	Phần nguyên = 6	a ₁
0,0	x	16	= 0	Phần nguyên = 0	a ₂

Kết quả : $0,375_{10} = 0,6_{16}$

1.2.2. *Đổi một biểu diễn trong hệ bất kì sang hệ thập phân*

Muốn thực hiện phép biến đổi, dùng công thức :

$$N_{10} = a_{n-1} \times r^{n-1} + \dots + a_0 \times r^0 + a_{-1} \times r^{-1} + \dots + a_{-m} \times r^{-m} \quad (1.12)$$

Thực hiện lấy tổng vế phải sẽ có kết quả cần tìm. Trong biểu thức trên, a_i và r là hệ số và cơ số hệ có biểu diễn.

$$\text{Ví dụ: } 10110_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 22_{10}$$

$$215_8 = 2 \times 8^2 + 1 \times 8^1 + 5 \times 8^0 = 141_{10}$$

$$76A_{16} = 7 \times 16^2 + 6 \times 16^1 + 10 \times 16^0 = 1898_{10}$$

1.2.3. *Đổi các số từ hệ nhị phân sang hệ cơ số 8 và 16*

Vì $8 = 2^3$ và $16 = 2^4$ nên chỉ cần dùng một số nhị phân 3 bit là đủ ghi 8 ký hiệu của hệ cơ số 8 và từ nhị phân 4 bit cho hệ cơ số 16. Do đó, muốn đổi một số nhị phân sang hệ cơ số 8 và 16 chia số nhị phân cần đổi, kể từ dấu phân số sang trái và phải thành từng nhóm 3 bit hoặc 4 bit. Sau đó, thay các nhóm bit đã phân bằng ký hiệu tương ứng của hệ cần đổi tới.

Ví dụ:

a. Đổi số $110111,0111_2$ sang số hệ cơ số 8

Tính từ dấu phân số, chia số này thành các nhóm 3 bit như sau :

110 111 , 011 100

↓ ↓ ↓ ↓

6 7 3 4

Kết quả: $110111,0111_2 = 67,34_8$ (đã thêm 2 số 0 phía sau dấu phẩy để tiện biến đổi).

b. Đổi số nhị phân $11110110,01101_2$ sang số hệ cơ số 16

Phân nhóm và thay thế như sau :

0001 1111 0110 0110 1000

↓ ↓ ↓ ↓ ↓

1 F 6 6 8

Kết quả: $11110110,01101_2 = 1F6,68_{16}$

1.3 MỘT SỐ PHÉP TOÁN

1.3.1. Số nhị phân có dấu

1.3.1.1 Biểu diễn số nhị phân có dấu

Có ba phương pháp thể hiện số nhị phân có dấu.

Số thập phân	Biểu diễn theo bit dấu	Biểu diễn theo bù 1	Biểu diễn theo bù 2
-7	1.111	1.000	1.001
-6	1.110	1.001	1.010
-5	1.101	1.010	1.011
-4	1.100	1.011	1.100
-3	1.011	1.100	1.101
-2	1.010	1.101	1.110
-1	1.001	1.110	1.111
0	0000		
+1	0.001	0.001	0.001
+2	0.010	0.010	0.010
+3	0.011	0.011	0.011
+4	0.100	0.100	0.100
+5	0.101	0.101	0.101
+6	0.110	0.110	0.110
+7	0.111	0.111	0.111

Bảng 1-3 là biểu diễn các số nhị phân có dấu.

a. Sử dụng một bit dấu.

Trong phương pháp này dùng một bit phụ, đứng trước các bit trị số để biểu diễn dấu, ‘0’ chỉ dấu dương (+), ‘1’ chỉ dấu âm (-).

Ví dụ: $+9_{10} = 0.000\ 1001_2$

$-9_{10} = 1.000\ 1001_2$

b. Sử dụng phép bù 1.

Số dương giữ nguyên trị số, bit dấu là 0; số âm: bit dấu là 1 và lấy bù 1 các bit trị số. Bù 1 được thực hiện bằng cách lấy đảo của các bit cần được lấy bù.

$$\begin{aligned}\text{Ví dụ: } +9_{10} &= 0.000\ 1001_2 \\ -9_{10} &= 1.111\ 0110_2 (\text{bù } 1)\end{aligned}$$

c. Sử dụng phép bù 2

Là phương pháp phổ biến nhất. Số dương thể hiện bằng số nhị phân không bù (bit dấu bằng 0), còn số âm được biểu diễn qua bù 2 (bit dấu bằng 1).

Bù 2 được thực hiện bằng cách lấy bù 1 cộng 1.

Có thể biểu diễn số âm theo phương pháp bù 2 xen kẽ: bắt đầu từ bit LSB, dịch về bên trái, giữ nguyên các bit cho đến gặp bit 1 đầu tiên và lấy bù các bit còn lại. Bit dấu giữ nguyên.

$$\text{Ví dụ: } +9_{10} = 0.000\ 1001_2 \quad ; \quad -9_{10} = 1.111\ 0111_2 (\text{bù } 2) .$$

Bảng 1-3 biểu diễn các số nhị phân có dấu.

1.3.2 Các phép cộng và trừ số nhị phân có dấu

Như đã nói ở trên, phép bù 1 và bù 2 thường được áp dụng để thực hiện các phép tính nhị phân với số có dấu.

1. Cộng và trừ các số theo biểu diễn bù 1

a. Phép cộng.

Hai số dương: cộng như cộng nhị phân thông thường, kể cả bit dấu.

$$\begin{array}{r} 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1_2 \quad (5_{10}) \\ +\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1_2 \quad (7_{10}) \\ \hline 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0_2 \quad (12_{10}) \end{array}$$

Hai số âm: biểu diễn chúng ở dạng bù 1 và cộng như cộng nhị phân, kể cả bit dấu. Bit tràn cộng vào kết quả. Chú ý, kết quả được viết dưới dạng bù 1.

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0_2 \quad (-5_{10}) \\ +\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0_2 \quad (-7_{10}) \\ \hline 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0_2 \\ \downarrow \\ \text{Bit tràn} \rightarrow +\ 1 \\ \hline 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \quad (-12_{10}) \end{array}$$

Hai số khác dấu và số âm lớn hơn: cộng số dương với bù 1 của số âm. Kết quả không có bit tràn và ở dạng bù 1.

$$\begin{array}{r} 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1_2 \quad (-10_{10}) \\ +\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1_2 \quad (+5_{10}) \\ \hline 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0 \quad (-5_{10}) \end{array}$$

Hai số khác dấu và số dương lớn hơn: cộng số dương với bù 1 của số âm. Bit tràn được cộng vào kết quả.

$$\begin{array}{r}
 0000\ 1010_2 \quad (+10_{10}) \\
 + 1111\ 1010_2 \quad (-5_{10}) \\
 \hline
 10000\ 0100_2 \\
 \downarrow \\
 \text{Bit tràn} \rightarrow + 1 \\
 \hline
 0000\ 0101 \quad (+5_{10})
 \end{array}$$

b. *Phép trừ.*

Để thực hiện phép trừ, lấy bù 1 của số trừ, sau đó thực hiện các bước như phép cộng.

2. Cộng và trừ nhị phân theo biểu diễn bù 2

a. *Phép cộng.*

Hai số dương: cộng như cộng nhị phân thông thường. Kết quả là dương.

$$\begin{array}{r}
 0000\ 1011_2 \quad (11_{10}) \\
 + 0000\ 0111_2 \quad (7_{10}) \\
 \hline
 0001\ 0010_2 \quad (18_{10})
 \end{array}$$

Hai số âm: lấy bù 2 của hai số hạng và cộng, kết quả ở dạng bù 2.

$$\begin{array}{r}
 1111\ 0101_2 \quad (-11_{10}) \\
 + 1111\ 1001_2 \quad (-7_{10}) \\
 \hline
 11110\ 1110_2 \\
 \downarrow \\
 \text{Bit tràn} \rightarrow \text{bỏ} \\
 \hline
 1110\ 1110 \quad (-18_{10})
 \end{array}$$

Hai số khác dấu và số dương lớn hơn: lấy số dương cộng với bù 2 của số âm. Kết quả bao gồm cả bit dấu, bit tràn bỏ đi.

$$\begin{array}{r}
 0000\ 1011_2 \quad (+11_{10}) \\
 + 1111\ 1001_2 \quad (-7_{10}) \\
 \hline
 10000\ 0100_2 \\
 \downarrow \\
 \text{Bit tràn} \rightarrow \text{bỏ} \\
 \hline
 0000\ 0100 \quad (+4_{10})
 \end{array}$$

Hai số khác dấu và số âm lớn hơn: số dương được cộng với bù 2 của số âm, kết quả ở dạng bù 2 của số dương tương ứng. Bit dấu là 1.

$$\begin{array}{r}
 1111\ 0101_2 \quad (-11_{10}) \\
 + 0000\ 0111_2 \quad (+7_{10}) \\
 \hline
 1111\ 1100_2 \quad (-3_{10})
 \end{array}$$

b. Phép trừ.

Phép trừ hai số có dấu là các trường hợp riêng của phép cộng. Ví dụ, khi lấy +9 trừ đi +6 là tương ứng với +9 cộng với -6.

1.3.3 Phép nhân.

Nhân hai số nhị phân có dấu cũng giống như nhân hai số nhị phân thông thường với quy tắc nhân là:

$$0 \times 0 = 1 \times 0 = 0 \times 1 = 0; 1 \times 1 = 1.$$

Dấu trong phép nhân được xác định như sau:

- Tích của hai số cùng dấu sẽ mang dấu dương.
- Tích của hai số khác dấu sẽ mang dấu âm.

Trong quá trình nhân, bit dấu của hai số được kiểm tra và dấu của kết quả được lưu lại trước khi thực hiện phép tính.

Thông thường, trong hệ thống số phép nhân nhị phân được thực hiện thông qua phép cộng và phép dịch trái liên tiếp.

Ví dụ:

$$\begin{array}{r} 110_2 \quad (6_{10}) \\ \times 11_2 \quad (3_{10}) \\ \hline 110 \\ 110 \\ \hline 10010_2 \quad (18_{10}) \end{array}$$

TÓM TẮT

Trong chương này chúng ta giới thiệu về một số hệ đếm thường được sử dụng trong hệ thống số: hệ nhị phân, hệ bát phân, hệ thập lục phân. Và phương pháp chuyển đổi giữa các hệ đếm đó. Trong chương 1 chúng ta cần nắm vững tổ chức của các hệ đếm và các phép tính số học trong các hệ đếm.

CÂU HỎI ÔN TẬP

1. Phân biệt các hệ đếm nhị phân, hệ thập phân, hệ 8 và hệ 16.
2. Hãy viết lại 16 trạng thái đầu tiên của hệ đếm nhị phân.
3. Đổi các số thập phân sau sang các hệ khác:
 - a) $N_{10} = 75$;
 - b) $N_{10} = 157$;
 - c) $N_{10} = 1976$;
 - d) $N_{10} = 2711$;
4. Đổi các số nhị phân sau sang hệ thập phân:

- a) $N_2 = 1011010$;
 b) $N_2 = 111000111$;
 c) $N_2 = 100001111$;
 d) $N_2 = 101010101$;
5. Đổi số nhị phân sau sang dạng bát phân:
- a) 0101 1111 0100 1110
 b) 1010 1100 1001 1000
 c) 1111 1010 1101 1001
 d) 1000 1101 1100 0011
6. Thực hiện phép tính hai số hệ 16 sau:
- a) $132,44_{16} + 215,02_{16}$.
 b) $13E_{16} + 2FD_{16}$.
 a) $3B9_{16} + 7A3_{16}$.
 a) $9B5_{16} + 6D8_{16}$.
7. Thực hiện phép tính hai số hệ 8 phân sau:
- a) $132,44_8 + 215,02_8$.
 b) $637_8 + 245_8$.
 c) $410_8 + 723_8$.
 d) $215_8 + 654_8$.
8. Thực hiện phép cộng hai số có dấu sau theo phương pháp bù 1:
- a) $0.101\ 1111_2 + 0.100\ 1110_2$
 b) $1.010\ 1100_2 + 1.001\ 1000_2$
 c) $1.111\ 1010_2 + 1.101\ 1001_2$
 d) $1.000\ 1101_2 + 1.100\ 0011_2$
9. Thực hiện phép cộng hai số có dấu sau theo phương pháp bù 2:
- a) $0.101\ 1111_2 + 0.100\ 1110_2$
 b) $1.010\ 1100_2 + 1.001\ 1000_2$
 c) $1.111\ 1010_2 + 1.101\ 1001_2$
 d) $1.000\ 1101_2 + 1.100\ 0011_2$
10. Hãy viết 4 số kế tiếp trong các dãy số sau:
- a) 11001_2 11010_2 11011_2
 b) 624_8 625_8 626_8

c) $9D_{16}$ $9E_{16}$ $9F_{16}$

11. Hãy chuyển đổi các số sau sang biểu diễn tương đương khác:

a) $Z_{16} = 24AE_{16} \rightarrow A_{10} \rightarrow A_2$

b) $Z_{16} = A6F2_{16} \rightarrow A_{10} \rightarrow A_2$

c) $A_{10} = 3118_{10} \rightarrow Z_{16} \rightarrow A_2$

d) $A_{10} = 9785_{10} \rightarrow Z_{16} \rightarrow A_2$

12. Hãy tính hiệu hai số nhị phân sau và kiểm tra lại kết quả ở dạng thập phân:

a) $11\ 0101 - 10\ 0101$

b) $1001\ 0110 - 10\ 0110$

PTIT

CHƯƠNG 2. CÔNG LOGIC

GIỚI THIỆU CHUNG

Đại số Boole (đại số logic) là một tập hợp các đối tượng có hai trạng thái: có hoặc không, mệnh đề đúng hoặc sai; các đối tượng này được biểu diễn bằng biến logic. Thông thường, khi trạng thái đối tượng là tồn tại thì biến logic biểu diễn có giá trị là 1 và ký hiệu là A , ngược lại biến logic của nó có giá trị là 0 và ký hiệu là \bar{A} .

Giữa các biến logic, người ta định nghĩa 3 phép toán cơ sở:

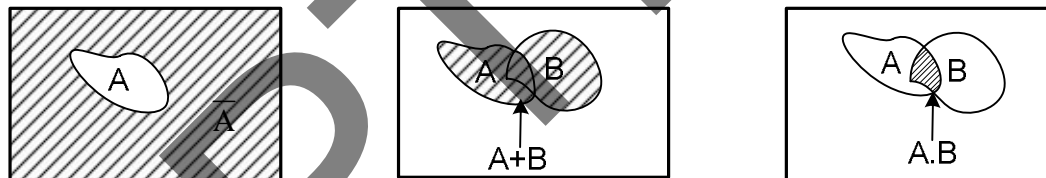
Phép phủ định logic đối với một biến A hay còn gọi là phép đảo. Khi nhận tác động của phép toán này, A sẽ nhận giá trị đảo với giá trị ban đầu và ký hiệu là \bar{A} .

Phép cộng logic (phép *hoặc*) được ký hiệu bằng dấu “+”. Ví dụ, $(A + B)$, mỗi biến được gọi là một số hạng và kết quả gọi là tổng.

Phép nhân logic (phép *và*) được ký hiệu bằng dấu “.”. Ví dụ, $(A . B)$, mỗi biến được gọi là một thừa số và kết quả gọi là tích.

Có thể dùng giản đồ Venn trong lý thuyết tập hợp để biểu diễn 3 phép toán logic trên.

Một trạng thái của đối tượng nào đó luôn có thì biến logic biểu diễn nó luôn có giá trị 1 ngược lại thì nhận giá trị 0. Nhận được trong tập hợp này hai hằng số 0 và 1.



Hình 2-1. Đồ thị Venn mô tả ba phép tính cơ bản

Sau đây, sẽ thảo luận chi tiết các vấn đề này.

2.1 . CÁC HÀM CHUYỂN MẠCH CƠ BẢN

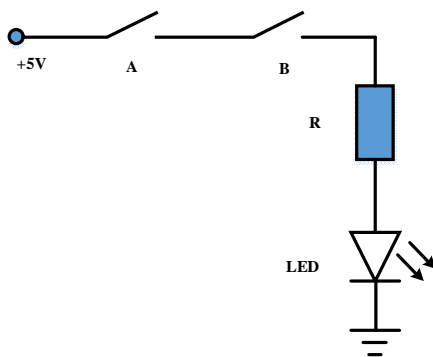
Đại số chuyển mạch hay còn được gọi là đại số Boole do nhà toán học Anh George Boole sáng lập và ông Shannon phát triển. Bắt nguồn từ các bài toán có mối quan hệ nhân quả, ông Boole đã đưa hệ nhị phân vào bài toán này để đưa hai giá trị 1 và 0 thay cho trạng thái đóng và ngắt của một chuyển mạch và được thể hiện bằng hàm toán học và được gọi là hàm chuyển mạch. Một hệ thống gồm các chuyển mạch được mắc song song hay nối tiếp sẽ biểu diễn được các hàm logic. Sau đây, sẽ đề cập đến một số hàm chuyển mạch cơ bản.

2.1.1. Hàm AND.

Hình 2-2 mô tả hàm AND. Hai chuyển mạch đấu nối tiếp với nhau và nối tiếp với điện trở R và LED. Khi có dòng chạy qua mạch thì LED sáng, vậy LED chỉ sáng khi cả hai chuyển mạch A, B cùng đóng. Hai chuyển mạch A, B là biến của hàm AND, trạng thái của LED là giá trị của hàm AND được ký hiệu là F .

Biểu thức sau mô tả mối quan hệ giữa hàm và biến của hàm AND.

$$F(A,B) = A \text{ AND } B = A.B = AB$$



Hình 2-2. Mạch điện mô tả hàm AND

Đối với hàm nhiều biến có biểu thức sau:

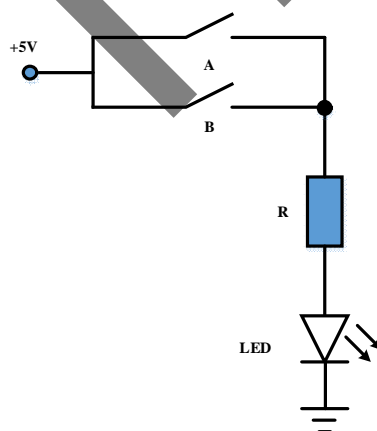
$$F(A,B,C,D...) = A.B.C.D...$$

2.1.2. Hàm OR.

Hình 2-3 mô tả hàm OR. Hai chuyển mạch đấu song song với nhau và nối tiếp với điện trở R và LED. Khi có dòng chạy qua mạch thì LED sáng, vậy LED chỉ tắt khi cả hai chuyển mạch A, B cùng mở. Hai chuyển mạch A và B là biến của hàm OR, trạng thái của LED là giá trị của hàm OR được ký hiệu là F.

Biểu thức sau mô tả mối quan hệ giữa hàm và biến của hàm OR.

$$F(A,B) = A \text{ OR } B = A+B$$



Hình 2-3. Mạch điện mô tả hàm OR

Đối với hàm nhiều biến có biểu thức sau:

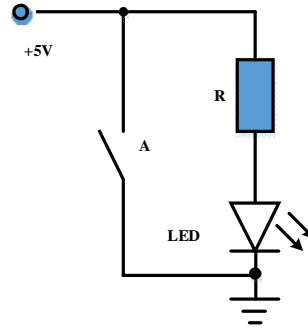
$$F(A,B,C,D...) = A+B+C+D+...$$

2.1.3. Hàm NOT.

Hình 2-4 mô tả hàm NOT. Chuyển mạch A đấu song song với LED. Khi có dòng chạy qua mạch thì LED sáng, vậy LED chỉ sáng khi chuyển mạch A ở trạng thái mở. Chuyển mạch A là biến của hàm NOT, trạng thái của LED là giá trị của hàm NOT được ký hiệu là F.

Biểu thức sau mô tả mối quan hệ giữa hàm và biến của hàm NOT.

$$F(A) = \text{NOT } A = \overline{A}$$



Hình 2-4. Mạch điện mô tả hàm NOT

Đối với hàm nhiều biến có biểu thức sau:

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

2.2. MỘT SỐ ĐỊNH LÝ CƠ BẢN TRONG ĐẠI SỐ BOOLE

2.2.1. Các định lý cơ bản.

Vì trong đại số logic chỉ có thể có hai hằng số 0 và 1 nên các biến logic cũng chỉ lấy một trong hai giá trị đó. Do đó, xuất hiện các định lý cơ bản sau:

STT	Tên gọi	Dạng tích	Dạng tổng
1	Đồng nhất	$A.1 = A$	$A + 0 = A$
2	Phần tử 0, 1	$A.0 = 0$	$A + 1 = 1$
3	Bù	$A.\overline{A} = 0$	$A + \overline{A} = 1$
4	Bất biến	$A.A = A$	$A + A = A$
5	Hấp thụ	$A + A.B = A$	$A.(A + B) = A$
6	Hoàn nguyên	$\overline{\overline{A}} = A$	
7	Định lý DeMorgan	$\overline{(A.B.C\dots)} = \overline{A} + \overline{B} + \overline{C}\dots$	$\overline{(A + B + C + \dots)} = \overline{A}.\overline{B}.\overline{C}\dots$

Bảng 2.1. Một số định lý cơ bản trong đại số Boole

2.2.2 Các định luật cơ bản:

- + Hoán vị: $A.B = B.A$, $A+B = B+A$
- + Kết hợp: $A.(B.C) = (A.B).C$, $A+(B+C) = (A+B)+C$
- + Phân phối: $A.(B+C) = A.B + A.C$; $(A+B).(A+C) = A + B.C$
- + Nhất quán: nếu $A + B = B$ thì $A.B = A$

2.2.3. Ba quy tắc về đẳng thức :

2.2.3.1. Quy tắc thay thế:

Trong bất kỳ đẳng thức logic nào nếu muốn thay một biến nào đó bằng một hàm số thì đẳng thức vẫn được thiết lập.

Quy tắc này có ứng dụng rất lớn trong việc biến đổi công thức đã biết để tạo ra công thức mới, mở rộng phạm vi ứng dụng của công thức đã biết.

Ví dụ: Có công thức $\overline{(A+B)} = \overline{A} \cdot \overline{B}$. Dùng $F = A+C$ thay vào biến A:

$$\overline{(A+C)+B} = \overline{A+C} \cdot \overline{B} = \overline{A} \cdot \overline{C} \cdot \overline{B} \text{ hay } \overline{A+B+C} = \overline{A} \cdot \overline{B} \cdot \overline{C}$$

2.2.3.2. Quy tắc tìm đảo của hàm số:

Phép đảo của hàm số được thực hiện bằng cách đổi dấu nhân thành dấu cộng và ngược lại; đổi 0 thành 1 và ngược lại; đổi biến nguyên thành biến đảo và ngược lại. Ngoài ra, những dấu đảo nào của hàm nhiều biến vẫn phải giữ nguyên, và tuân thủ theo quy tắc đổi “nhân trước, cộng sau”.

Ví dụ: $F = \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} \cdot E$ hàm đảo tương ứng là $\overline{F} = A + B + \overline{C} + \overline{D} + E$

2.2.3.3. Quy tắc đối ngẫu:

Hàm F và F' là đối ngẫu với nhau khi các dấu cộng và dấu nhân; số '0' và số '1' đổi chỗ cho nhau một cách tương ứng.

Ví dụ: $F = A \cdot (B + C)$ thì $F' = A + B \cdot C$

Do quy tắc đối ngẫu nên các định lý cơ bản có thể viết dưới 2 dạng đối ngẫu nhau là dạng tích và dạng tổng.

2.3 CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM BOOLE

Như đã nói ở trên, hàm logic được thể hiện bằng những biểu thức đại số như các môn toán học khác. Đây là phương pháp tổng quát nhất để biểu diễn hàm logic. Ngoài ra, một số phương pháp khác cũng được dùng để biểu diễn loại hàm này. Mỗi phương pháp đều có ưu điểm và ứng dụng riêng của nó. Dưới đây là nội dung của một số phương pháp thông dụng.

2.3.1 Bảng trạng thái

Bảng trạng thái liệt kê giá trị (trạng thái) mỗi biến theo từng cột và giá trị hàm theo một cột riêng (thường là bên phải bảng). Bảng trạng thái còn được gọi là **bảng sự thật** hay **bảng chân lý**.

m	A	B	C	f
m_0	0	0	0	0
m_1	0	0	1	1
m_2	0	1	0	1
m_3	0	1	1	0
m_4	1	0	0	1
m_5	1	0	1	0
m_6	1	1	0	0
m_7	1	1	1	1

Bảng 2-2. Bảng trạng thái hàm 3 biến

Đối với hàm n biến sẽ có 2^n tổ hợp độc lập. Các tổ hợp này được kí hiệu bằng chữ m_i , với $i = 0$ đến $2^n - 1$ (xem bảng 2-2) và có tên gọi là các **hạng tích** hay còn gọi là **minterm**.

Đặc điểm của bảng trạng thái:

+ Rõ ràng, trực quan. Sau khi xác định các giá trị biến vào thì có thể tìm được giá trị đầu ra nhờ bảng trạng thái.

+ Để giải quyết bài toán ở dạng logic thì sử dụng bảng trạng thái là hữu ích nhất. Nên trong quá trình thiết kế mạch số việc đầu tiên nên làm là lập bảng trạng thái.

Nhược điểm chủ yếu của bảng trạng thái là sẽ phức tạp nếu số biến quá nhiều, không thể dùng các công thức và định lý để tính toán.

2.3.2 Phương pháp đại số

Có 2 dạng biểu diễn là dạng *tuyến* (tổng các tích) và dạng *hội* (tích các tổng).

+ Dạng tuyến: Mỗi số hạng của tổng được gọi là một *hạng tích* hay *minterm* (đủ biến), và thường kí hiệu bằng chữ " m_i " (chỉ số i được tính trong hệ thập phân).

+ Dạng hội: Mỗi thừa số là *hạng tổng* hay *maxterm* (đủ biến), thường được kí hiệu bằng chữ " M_i ". Nếu trong tất cả mỗi hạng tích hay hạng tổng có đủ mặt các biến, thì dạng tổng các tích hay tích các tổng tương ứng được gọi là dạng *chuẩn*. Dạng chuẩn là duy nhất.

Bảng 2-3 là các m_i và M_i của hàm 2 biến và 3 biến.

Biến		Minterm (m_i)	Maxterm (M_i)
A	B		
0	0	$\bar{A} \bar{B} = m_0$	$A + B = M_1$
0	1	$\bar{A} B = m_1$	$A + \bar{B} = M_1$
1	0	$A \bar{B} = m_2$	$\bar{A} + B = M_2$
1	1	$A B = m_3$	$\bar{A} + \bar{B} = M_3$

a)

Biến			Minterm (m_i)	Maxterm (M_i)
A	B	C		
0	0	0	$\bar{A} \bar{B} \bar{C} = m_0$	$A + B + C = M_0$
0	0	1	$\bar{A} \bar{B} C = m_1$	$A + B + \bar{C} = M_1$
0	1	0	$\bar{A} B \bar{C} = m_2$	$A + \bar{B} + C = M_2$
0	1	1	$\bar{A} B C = m_3$	$A + \bar{B} + \bar{C} = M_3$
1	0	0	$A \bar{B} \bar{C} = m_4$	$\bar{A} + B + C = M_4$
1	0	1	$A \bar{B} C = m_5$	$\bar{A} + B + \bar{C} = M_5$
1	1	0	$A B \bar{C} = m_6$	$\bar{A} + \bar{B} + C = M_6$
1	1	1	$A B C = m_7$	$\bar{A} + \bar{B} + \bar{C} = M_7$

b)

Bảng 2-3. Cấu trúc của minterm và Maxterm 3 biến

Tổng quát, hàm logic n biến có thể biểu diễn chỉ bằng một dạng tổng các tích:

$$f(X_{n-1}, \dots, X_0) = \sum_{i=0}^{2^n-1} a_i m_i \quad (2.1)$$

hoặc bằng chỉ một dạng tích các tổng:

$$f(X_{n-1}, \dots, X_0) = \prod_{i=0}^{2^n-1} (a_i + M_i) \quad (2.2)$$

Ở đây, a_i chỉ lấy hai giá trị 0 hoặc 1. Đối với một hàm thì minterm và maxterm là bù của nhau.

a) Biểu diễn hàm sau theo dạng minterm:

Ví dụ cho hàm $F(A, B, C) = A + BC \rightarrow$ Đây là dạng minterm không đầy đủ. Muốn đưa về dạng chuẩn tắc (đủ biến) sử dụng một số định lý đã nêu để biến đổi.

$$\begin{aligned} F(A, B, C) &= A + BC = A(B + \bar{B})(C + \bar{C}) + (A + \bar{A})BC = \\ &= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + \bar{A}BC = \\ &= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC \end{aligned}$$

Đây là dạng chuẩn minterm.

Tuy nhiên, biểu diễn này khá dài nên mỗi một hạng tích được thay thế bằng ký hiệu m_i tương ứng (xem bảng 2-3). Lưu ý, nguyên biến (biến không đảo) được thay bằng số “1₂” và đảo biến được thay bằng số “0₂”. Như vậy, biểu thức có dạng:

$$\begin{array}{ccccccccc} F(A, B, C) & = & ABC & + & AB\bar{C} & + & A\bar{B}C & + & A\bar{B}\bar{C} & + & \bar{A}BC \\ & & \downarrow\downarrow\downarrow & & \downarrow\downarrow\downarrow & & \downarrow\downarrow\downarrow & & \downarrow\downarrow\downarrow & & \downarrow\downarrow\downarrow \\ & & 111_2 & & 110_2 & & 101_2 & & 100_2 & & 011_2 \\ & & 7_{10} & & 6_{10} & & 5_{10} & & 4_{10} & & 3_{10} \end{array}$$

$$\rightarrow F(A, B, C) = m_7 + m_6 + m_5 + m_4 + m_3 = \sum(3, 4, 5, 6, 7)$$

b) Biểu diễn hàm sau theo dạng Maxterm:

Ví dụ, cho hàm $F(A, B, C) = A + BC = (A + B)(A + C) \rightarrow$ Đây là dạng Maxterm không đầy đủ. Muốn đưa về dạng chuẩn (đủ biến) sử dụng một số định lý đã nêu để biến đổi.

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

Giống như minterm, người ta cũng biểu diễn hàm logic theo ký hiệu M_i . Trong đó nguyên biến được thay thế bằng số “0₂” và đảo biến thay bằng số “1₂”. Do đó, viết biểu thức thành dạng sau:

$$\begin{array}{ccccccccc} F(A, B, C) & = & (A + B + C) & (A + B + \bar{C}) & (A + \bar{B} + C) \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ & & (0 & 0 & 0)_2 & (0 & 0 & 1)_2 & (0 & 1 & 0)_2 \\ & & 0_{10} & & 1_{10} & & 2_{10} \end{array}$$

$$\rightarrow F(A, B, C) = M_0 \cdot M_1 \cdot M_2 = \prod(0, 1, 2)$$

Từ các ví dụ trên có một số nhận xét sau:

Đối với dạng minterm: m_i được gọi là số hạng nhỏ nhất. Số hạng nhỏ nhất có các tính chất sau:

- + đều bao gồm tất cả các biến của hàm trong một thừa số; mỗi biến số chỉ xuất hiện một lần dưới dạng thừa số hoặc là nguyên biến hoặc là đảo biến.

- + tích của hai số hạng nhỏ nhất bất kỳ luôn bằng 0

- + tổng của tất cả các số hạng nhỏ nhất luôn bằng 1

Đối với dạng Maxterm: M_i được gọi là thừa số lớn nhất. Thừa số lớn nhất có các tính chất sau:

- + đều bao gồm tất cả các biến của hàm;

- + mỗi biến số chỉ xuất hiện một lần dưới dạng tổng của thừa số hoặc là nguyên biến hoặc là đảo biến.

- + tổng của hai thừa số lớn nhất bất kỳ luôn bằng 1

- + tích của tất cả các thừa số luôn bằng 0

Ưu điểm của phương pháp đại số:

- + Dùng các ký hiệu logic biểu diễn mối quan hệ logic giữa các biến làm cho cách viết gọn, cách viết này có tính khái quát và trừu tượng cao.

- + Rất tiện sử dụng các công thức và định lý của đại số Boole để biến đổi.

- + Tiện cho việc sử dụng sơ đồ logic để thực hiện hàm số. Chỉ dùng các ký hiệu logic của mạch điện công tương ứng thay thế phép toán xét trong biểu thức hàm số thì được một sơ đồ logic.

Nhược điểm chính của phương pháp này là khó xác định giá trị hàm ứng với tổ hợp biến một cách trực tiếp đối với các hàm phức tạp (không trực quan như bảng trạng thái).

2.3.3 Phương pháp bảng Các nô (bảng Karnaugh hay phương pháp hình học).

Tổ chức của bảng Các nô:

Một hàm logic có n biến sẽ có 2^n ô (mỗi ô tương ứng với một minterm m_i của hàm). Các tổ hợp biến phải xếp theo thứ tự mã Gray nghĩa là các hạng tích trong hai ô kế cận chỉ khác nhau một biến. Các tổ hợp biến được viết theo một dòng (thường là phía trên) và một cột (thường là bên trái).

Tính tuần hoàn của bảng Các nô:

Không những các ô kế cận khác nhau một biến mà các ô đầu dòng và cuối dòng, đầu cột và cuối cột cũng chỉ khác nhau một biến (kể cả 4 góc vuông của bảng) nên các ô này cũng gọi là ô kế cận.

Cách ghi giá trị hàm trên bảng Các nô: Muốn thiết lập bảng Các nô của một hàm đã cho dưới dạng chuẩn tổng các tích (minterm), chỉ việc ghi giá trị 1 vào các ô ứng với hạng tích có mặt trong biểu diễn (ứng với $a_i = 1$), các ô còn lại sẽ lấy giá trị 0 hoặc được bỏ trống. Nếu hàm cho dưới dạng tích các tổng (Maxterm), cách làm cũng tương tự, các ô ứng với hạng tổng có trong biểu diễn lại lấy giá trị 0 (ứng với $a_i = 0$), và các ô khác lấy giá trị 1.

Cấu tạo bảng Các nô cho hàm 3 biến, 4 biến và 5 biến được cho tại bảng 2-4.

		BC			
		00	01	11	10
A	0	m_0 $\overline{A}\overline{B}\overline{C}$	m_1 $\overline{A}\overline{B}C$	m_3 $\overline{A}B\overline{C}$	m_2 $\overline{A}BC$
	1	m_4 $A\overline{B}\overline{C}$	m_5 $A\overline{B}C$	m_7 $AB\overline{C}$	m_6 ABC

		CD			
		00	01	11	10
AB	00	m_0 $\overline{A}\overline{B}\overline{C}\overline{D}$	m_1 $\overline{A}\overline{B}C\overline{D}$	m_3 $\overline{A}B\overline{C}\overline{D}$	m_2 $\overline{A}BC\overline{D}$
	01	m_4 $\overline{A}\overline{B}\overline{C}D$	m_5 $\overline{A}\overline{B}CD$	m_7 $\overline{A}BCD$	m_6 $\overline{A}BC\overline{D}$
11	11	m_{12} $AB\overline{C}\overline{D}$	m_{13} $AB\overline{C}D$	m_{15} $ABCD$	m_{14} $AB\overline{C}\overline{D}$
	10	m_8 $A\overline{B}\overline{C}\overline{D}$	m_9 $A\overline{B}C\overline{D}$	m_{11} $A\overline{B}CD$	m_{10} $A\overline{B}\overline{C}D$

		CDE							
		000	001	011	010	110	111	101	100
AB	00	m_0 $\overline{A}\overline{B}\overline{C}\overline{D}\overline{E}$	m_1 $\overline{A}\overline{B}\overline{C}D\overline{E}$	m_3 $\overline{A}\overline{B}C\overline{D}\overline{E}$	m_2 $\overline{A}\overline{B}CD\overline{E}$	m_6 $\overline{A}\overline{B}\overline{C}\overline{D}E$	m_7 $\overline{A}\overline{B}C\overline{D}E$	m_5 $\overline{A}\overline{B}CD\overline{E}$	m_4 $\overline{A}\overline{B}CD\overline{E}$
	01	m_8 $\overline{A}\overline{B}\overline{C}\overline{D}E$	m_9 $\overline{A}\overline{B}\overline{C}DE$	m_{11} $\overline{A}\overline{B}C\overline{D}E$	m_{10} $\overline{A}\overline{B}CD\overline{E}$	m_{14} $\overline{A}\overline{B}\overline{C}DE$	m_{15} $\overline{A}\overline{B}CDE$	m_{13} $\overline{A}\overline{B}CD\overline{E}$	m_{12} $\overline{A}\overline{B}CD\overline{E}$
11	11	m_{24} $AB\overline{C}\overline{D}\overline{E}$	m_{25} $AB\overline{C}D\overline{E}$	m_{27} $AB\overline{C}\overline{D}E$	m_{26} $AB\overline{C}DE$	m_{30} $AB\overline{C}\overline{D}E$	m_{31} $AB\overline{C}DE$	m_{29} $AB\overline{C}D\overline{E}$	m_{28} $AB\overline{C}D\overline{E}$
	10	m_{16} $A\overline{B}\overline{C}\overline{D}\overline{E}$	m_{17} $A\overline{B}\overline{C}D\overline{E}$	m_{19} $A\overline{B}C\overline{D}\overline{E}$	m_{18} $A\overline{B}CD\overline{E}$	m_{22} $A\overline{B}\overline{C}\overline{D}E$	m_{23} $A\overline{B}C\overline{D}E$	m_{21} $A\overline{B}CD\overline{E}$	m_{20} $A\overline{B}CD\overline{E}$

Bảng 2-4. Bảng Các nô cho hàm 3, 4, 5 biến

Ví dụ: Xây dựng bảng Các nô cho hàm logic sau:

$$F(A, B, C, D) = \Sigma(0, 1, 5, 7, 10, 14, 15)$$

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

Bảng 2-5. Bảng Các nô

Ưu điểm nổi bật nhất của bảng Các nô là tính kề nhau về logic của các số hạng nhỏ nhất (minterm), nó biểu thị rõ ràng thành sự liên kề hình học của các ô trong bảng. Do vậy, rất dễ dàng tối thiểu hóa hàm.

Nhược điểm là do có quá nhiều ô nên trong trường hợp nhiều biết việc tổ chức bảng rất phức tạp. Do đó, chỉ nên dùng bảng Các nô cho trường hợp hàm logic có số biến nhỏ hơn 6.

2.3. CÁC PHƯƠNG PHÁP TỐI THIỂU HÓA (RÚT GỌN HÀM)

2.3.1. Phương pháp đại số

Dựa vào các định lý đã học để đưa biểu thức về dạng tối giản.

Ví dụ: Hãy đưa hàm logic về dạng tối giản:

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

Lời giải:

Áp dụng định lý, $A + \bar{A} = 1$, $A + A.B = A$ được:

$$\begin{aligned} f &= AB + \bar{A}C + BC(A + \bar{A}) \\ &= AB + ABC + \bar{A}C + \bar{A}BC \\ &= AB + \bar{A}C \end{aligned}$$

Nhận xét:

Từ ví dụ trên nhận thấy: nếu trong tổng các tích, xuất hiện một biến và đảo của biến đó trong hai số hạng khác nhau, các thừa số còn lại trong hai số hạng đó tạo thành thừa số của một số hạng thứ ba thì số hạng thứ ba đó là thừa và có thể bỏ đi.

2.3.2 Phương pháp bảng Các nô

Phương pháp này thường được dùng để rút gọn các hàm có số biến không vượt quá 5.

Các bước tối thiểu hóa:

Đối với minterm:

1. Gộp các ô kề cận có giá trị '1' (hoặc '0') lại thành từng nhóm 2, 4, ..., 2^i ô. Số ô trong mỗi nhóm càng lớn kết quả thu được càng tối giản tức là nếu gộp được 2^n ô thì tối giản được n biến. Một ô có thể được gộp nhiều lần trong các nhóm khác nhau. Nếu gộp theo các ô có giá trị '0' sẽ thu được biểu thức bù của hàm.

2. Thay mỗi nhóm bằng một hạng tích mới, trong đó giữ lại các biến giống nhau theo dòng và cột.

3. Cộng các hạng tích mới lại, có hàm đã tối giản.

Đối với Maxterm:

1. Gộp các ô kề cận có giá trị '0' (hoặc '1') lại thành từng nhóm 2, 4, ..., 2^i ô. Số ô trong mỗi nhóm càng lớn kết quả thu được càng tối giản. Một ô có thể được gộp nhiều lần trong các nhóm khác nhau. Nếu gộp theo các ô có giá trị '1' sẽ thu được biểu thức bù của hàm.

2. Thay mỗi nhóm bằng một hạng tổng mới, trong đó giữ lại các biến giống nhau theo dòng và cột.

3. Nhân các hạng tổng mới lại, có hàm đã tối giản.

Ví dụ: Hãy dùng bảng Các nô để tối giản hàm : $f(A, B, C) = \sum(0, 1, 3, 4, 5)$

Lời giải:

A \ BC	BC				
	00	01	11	10	
0	1	1	1	0	$\overline{A}C$
1	1	1	0	0	
					\overline{B}

Bảng 2-6. Bảng Các ô

+ Xây dựng bảng Các ô tương ứng với hàm đã cho.

Rút gọn theo *minterm*

+ Gộp các ô có giá trị 1 kề cận lại với nhau thành hai nhóm (bảng 2-6)

Lời giải phải tìm : $f(A, B, C) = \overline{B} + \overline{A}C$

Nếu gộp các ô có giá trị 0 lại theo hai nhóm, thu được biểu thức hàm bù \overline{f} :

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

Rút gọn theo *Maxterm*

A \ BC	BC				
	00	01	11	10	
0	1	1	1	0	$\overline{B} + C$
1	1	1	0	0	
					$\overline{A} + \overline{B}$

Bảng 2-7. Bảng Các ô

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

Nếu gộp các ô có giá trị 1 lại theo hai nhóm, thu được biểu thức hàm bù \overline{f} :

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

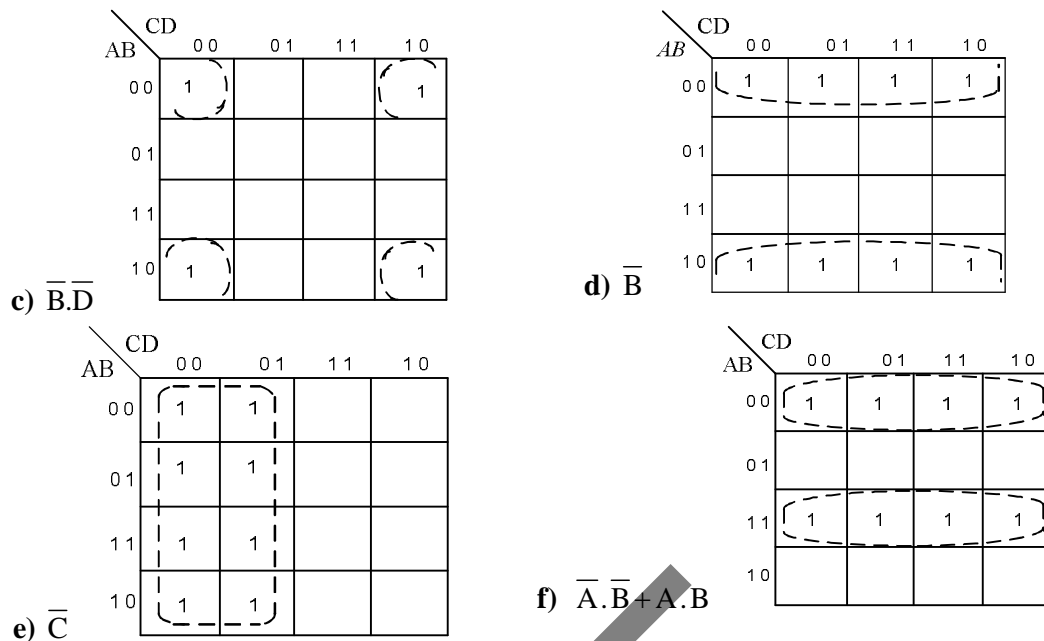
Bảng 2-8 trình bày một số cách gộp và giá trị của hàm theo *minterm*.

a) $B.\overline{D}$

AB \ CD	CD			
	00	01	11	10
00				
01	1			1
11	1			1
10				

b) $\overline{A}.B$

AB \ CD	CD			
	00	01	11	10
00				
01	1	1	1	1
11				
10				



Bảng 2-8. Bảng Các nô có 2^n ô được gộp.

Một số vấn đề cần lưu ý khi tiến hành rút gọn bằng bảng Các nô:

- Vòng gộp càng to càng tốt vì số biến được rút gọn càng nhiều.
- Mỗi vòng gộp bao gộp ít nhất một số hạng nhỏ nhất – minterm (hoặc một thừa số lớn nhất - Maxterm) không có trong vòng khác. Vòng nào bao gồm các số hạng đã có trong các vòng khác thì vòng đó là vòng thừa. Tuy nhiên, một số hạng có thể có mặt trong nhiều vòng khác nhau.
- Phải khoanh vòng sao cho toàn bộ số hạng nhỏ nhất - minterm (hoặc một thừa số lớn nhất - Maxterm) của hàm số đều nằm trong các vòng, không được để sót.

Ví dụ: Hãy dùng bảng Các nô để tối giản hàm :

$$f(A, B, C) = \sum(1, 4, 5, 6, 8, 12, 13, 15)$$

Lời giải:

Lập bảng Các nô ở bảng 2-9. Nhận thấy vòng $(m_4 + m_5 + m_{12} + m_{13})$ là lớn nhất nhưng các vòng khác đều đã chứa m_4, m_5, m_{12}, m_{13} nên vòng này là vòng thừa.

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

Bảng 2-9

Sau khi rút gọn, biểu thức hàm có dạng:

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

2.3.3. Rút gọn hàm logic hàm tùy chọn

Khái niệm hàm tùy chọn

Trên thực tế, tồn tại một số tổ hợp biến có giá trị không ảnh hưởng đến kết quả của hàm. Ví dụ: số BCD là số mã hóa 10 ký hiệu thập phân thành nhị phân 4 bit. Với 4 bit nhị phân có thể biểu diễn được mã Hexa, nhưng các ký hiệu $A_{16}(1010)$, $B_{16}(1011)$, $C_{16}(1100)$, $D_{16}(1101)$, $E_{16}(1110)$, $F_{16}(1111)$ lại không phù hợp với mã BCD.

Do vậy, khi lập bảng Các nô không quan tâm đến các giá trị này. Sáu giá trị này được gọi là các trạng thái tùy chọn (*don't care*). Các trạng thái này có thể có giá trị 1 hoặc 0, tùy thuộc vào mục đích người sử dụng và thông thường chúng được ký hiệu bằng chữ “x”.

Khi tiến hành tối thiểu bằng bảng Các nô: tùy theo yêu cầu, có thể tùy ý khoanh vòng qua điều kiện tùy chọn để hàm tối giản hơn.

Từ đó, có thể viết dạng tổng quát của hàm logic như sau:

Dạng chuẩn minterm:

$$f(A, B, C, \dots) = \sum m_i + \sum_d m_j; \quad (2.3)$$

\sum_d là ký hiệu của điều kiện tùy chọn;

Dạng chuẩn Maxterm:

$$F(A, B, C, \dots) = \prod M_i \prod_d M_j \quad (2.4)$$

\prod_d là ký hiệu của điều kiện tùy chọn;

Ví dụ: Tối thiểu hóa hàm $F(A, B, C, D) = \sum(0, 1, 2, 3, 6, 8) + \sum_d(10, 11, 12, 13, 14, 15)$

Lời giải:

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

Bảng 2-13. Bảng Các nô tìm hàm F

Lập bảng Các nô 2-13.

Từ đó, tìm được hàm tối giản sau:

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

2.4 CỔNG LOGIC

2.4.1 Cổng logic cơ bản

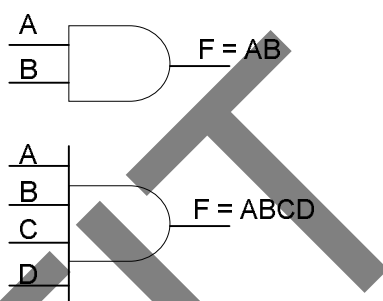
2.4.1.1 Cổng AND

Cổng AND thực hiện hàm logic

$$f = f(A, B) = A.B = AB \quad (2.5)$$

hoặc nhiều biến:

$$f(A, B, C, D, \dots) = A.B.C.D\dots = ABCD\dots \quad (2.6)$$



Hình 2-5. Ký hiệu cổng AND theo tiêu chuẩn ANSI (American National Standards Institute), Viện tiêu chuẩn Quốc gia Hoa kỳ

Bảng trạng thái 2-14a, b là nguyên lý hoạt động của cổng AND (2 đầu vào).

A	B	F = AB
0	0	0
0	1	0
1	0	0
1	1	1

a) Ghi theo giá trị logic

A	B	F = AB
L	L	L
L	H	L
H	L	L
H	H	H

b) Ghi theo mức logic

Bảng 2-14a,b. Bảng trạng thái mô tả hoạt động của cổng AND 2 đầu vào.

Các cổng logic được thực hiện bằng các cấu kiện bán dẫn như: diode, transistor, FET,... Để các phần tử này đóng mở được, tín hiệu tác động tới đầu vào của chúng phải có một mức điện áp thỏa mãn trong một dải giá trị nào đấy. Trong trường hợp này, chính xác hơn ta thay các giá trị logic bằng các mức điện áp tương ứng hay còn gọi là mức logic.

Theo qui ước, logic 1 được thay bằng mức điện thế cao, viết tắt là H (High) còn logic 0 được thay bằng mức điện thế thấp, viết tắt là L (Low) (bảng 2-14b). Cổng AND có n đầu vào sẽ có 2^n hạng tích (dòng) trong bảng trạng thái.

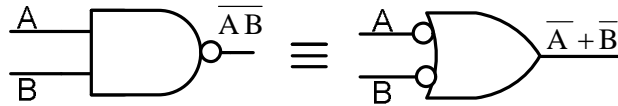
Thông qua định lý DeMorgan có thể biến đổi hàm ra của cổng NAND để tìm sự tương đương giữa cổng NAND và cổng OR

$$f = \overline{A.B} = \overline{A} + \overline{B} \quad (2.7)$$

hoặc đối với cổng nhiều đầu vào

$$f = \overline{A.B.C.D...} = \overline{A} + \overline{B} + \overline{C} + \overline{D} + ... \quad (2.8)$$

Nói cách khác, nếu tác động tới các đầu vào một cổng OR logic âm thì hàm ra của nó trùng với hàm ra của cổng NAND. Hình 2-6 trình bày sự tương đương trên.



Hình 2-6. Sơ đồ tương đương giữa cổng NAND và cổng OR logic âm

2.4.1.2. Cổng OR

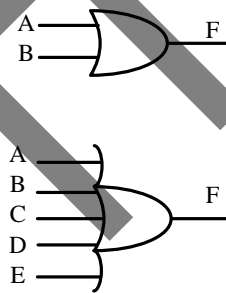
Cổng OR thực hiện hàm logic:

$$f(A, B) = A + B \quad (2.9)$$

hoặc với hàm nhiều biến:

$$f(A, B, C, D...) = A + B + C + D + ... \quad (2.10)$$

Ký hiệu của cổng OR được biểu diễn ở hình 2-7.



Hình 2-7. Ký hiệu của cổng OR theo tiêu chuẩn ANSI

Tương tự như cổng AND, nguyên lý hoạt động của cổng OR có thể được giải thích thông qua bảng trạng thái (Bảng 2-15a, b).

Một cổng OR có n đầu vào sẽ có 2^n hạng tích trong bảng trạng thái của nó.

A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

a) Theo giá trị logic

A	B	f
L	L	L
L	H	H
H	L	H
H	H	H

b) Theo mức điện thế

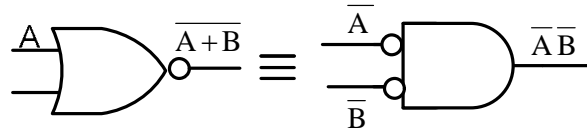
Bảng 2-15 a, b. Bảng trạng thái của cổng OR.

Cũng trên sơ đồ, định lý DeMorgan có thể tìm được mối quan hệ giữa cổng NOR và cổng AND.

$$f = \overline{\overline{A} + \overline{B}} = \overline{A}. \overline{B} \quad (2.11)$$

Khi tác động tới đầu vào cổng AND logic âm, thì hàm ra của nó tương đương với hàm ra của cổng NOR với logic dương.

Hình 2-8 mô tả sự tương đương đã trình bày trên đây:



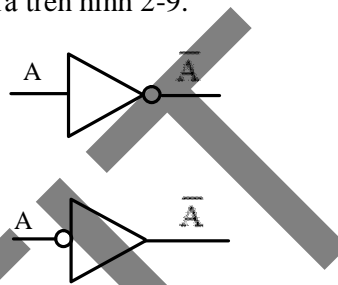
Hình 2-8. Sơ đồ tương đương giữa cổng NOR và cổng AND

2.4.1.3. Cổng NOT

Cổng NOT thực hiện hàm logic:

$$f = \bar{A} \quad (2.12)$$

Ký hiệu của cổng NOT được chỉ ra trên hình 2-9.



Hình 2-9. Ký hiệu của cổng NOT theo tiêu chuẩn ANSI.



Hình 2-10. Nguyên lý hoạt động của cổng NOT

Hoạt động của cổng NOT khá đơn giản, nếu đầu vào: $A = 0$ thì $\bar{A} = 1$, nếu $A = 1$ thì $\bar{A} = 0$

Hoạt động của cổng NOT được tóm tắt ở bảng 2-16a, b.

A	f
0	1
1	0

a) Theo giá trị logic

A	f
L	H
H	L

b) Theo mức logic

Bảng 2-16a, b. Bảng trạng thái của cổng NOT.

2.4.2. Logic dương và logic âm

Đối với cổng NOT đã khảo sát ở phần trên, việc đảo tín hiệu trước hay sau là như nhau: Dấu “tròn” ở đầu vào hoặc đầu ra của cổng chỉ ra là giá trị tác động có tích cực thấp. Khi không có dấu “tròn” này mức logic tương ứng sẽ là tích cực cao hay còn gọi là logic dương.

Logic dương là logic có điện thế mức 1 luôn lớn hơn điện thế mức 0.

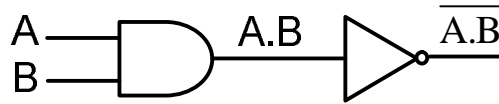
Logic âm là đảo của logic dương. Trong logic dương mức 1 có điện thế cao hơn mức 0. Đối với logic âm, ngược lại mức 0 có điện thế cao hơn mức 1.

2.4.3. Một số cổng ghép thông dụng

Khi ghép ba loại cổng logic cơ bản nhất sẽ thu được các mạch logic từ đơn giản đến phức tạp. Ở đây, chỉ xét một vài mạch ghép đơn giản nhưng rất thông dụng.

2.4.3.1. Cổng NAND

Ghép nối tiếp một cổng AND với một cổng NOT, được cổng NAND (Hình 2-11).

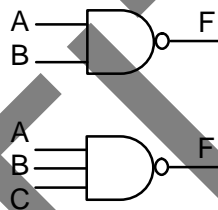


Hình 2-11. Sơ đồ cấu tạo cổng NAND

Hàm ra của cổng NAND 2 và nhiều biến vào như sau:

$$\begin{aligned} f &= \overline{A.B} = \overline{AB} \\ f &= \overline{A.B.C.D...} = \overline{ABCD...} \end{aligned} \quad (2.13)$$

Ký hiệu cổng NAND (hình 2-12) và bảng trạng thái (bảng 2-17).



Hình 2-12. Ký hiệu của cổng NAND theo tiêu chuẩn ANSI

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

A	B	F
L	L	H
L	H	H
H	L	H
H	H	L

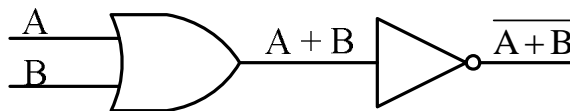
Bảng 2-17a,b. Bảng trạng thái của cổng NAND

3.1.3.2 Cổng NOR

Cổng NOR được thiết lập bằng cách nối tiếp một cổng OR với một cổng NOT.

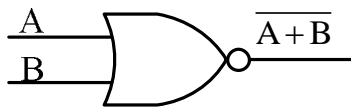
Từ hình 2-9, có thể viết được hàm ra của cổng NOR 2 và nhiều đầu vào như sau:

$$f = \overline{A + B} \text{ hay } f = \overline{A+B+C+...} \quad (2.14)$$



Hình 2-9. Sơ đồ cấu tạo cổng NOR

Ký hiệu của cổng NOR 2 đầu vào như chỉ ở hình 2-10a,b.



Hình 2-10. Ký hiệu cổng NOR theo tiêu chuẩn ANSI.

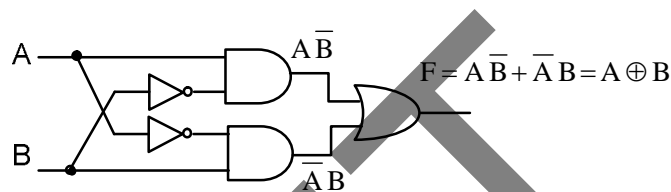
Hoạt động của cổng NOR được giải thích bằng bảng trạng thái như chỉ ở bảng 2-18a,b.

A	B	f
0	0	1
0	1	0
1	0	0
1	1	0

A	B	f
L	L	H
L	H	L
H	L	L
H	H	L

Bảng 2-18a, b. Bảng trạng thái của cổng NOR 2 đầu vào.

2.4.3.3 Cổng XOR



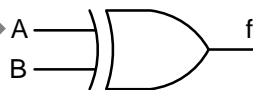
Hình 2-11 Sơ đồ của cổng XOR 2 đầu vào

Cổng XOR (Exelusive OR) còn có một số tên gọi khác, cổng khác dấu, cổng cộng modun 2. Đây là một loại cổng ghép phức tạp hơn NAND và NOR. Biểu thức logic đầu ra của cổng là:

$$f = A.\bar{B} + \bar{A}.B = A \oplus B \quad (2.15)$$

Từ biểu thức đầu của (2.15) có mạch logic của cổng XOR như hình 2-12.

Phân tử hợp thành của cổng XOR gồm cả ba loại cổng logic cơ sở AND, OR, NOT. Ký hiệu của cổng XOR 2 đầu vào được trình bày trên hình 2-16.



Hình 2-16. Ký hiệu của cổng XOR 2 đầu vào theo tiêu chuẩn ANSI

Bảng 2-19 là bảng trạng thái và bảng chức năng của cổng XOR 2 đầu vào.

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

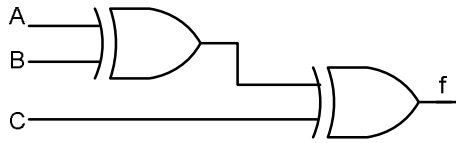
a) Bảng trạng thái

A	B	f
L	L	L
L	H	H
H	L	H
H	H	L

b) Bảng chức năng

Bảng 2-19. Bảng trạng thái và chức năng của cổng XOR

Hoạt động của cổng XOR nhiều đầu vào cũng tương tự như cổng 2 đầu vào, nghĩa là số bit 1 trên tất cả các đầu vào là một số lẻ, thì hàm ra có logic 1, ngược lại nếu cổng có số bit 1 trên tất cả các đầu vào là một số chẵn, thì hàm ra có logic 0. Có thể sử dụng cổng XOR 2 đầu vào để thực hiện hàm XOR nhiều đầu vào như hình 2-13.



Hình 2-13. Sơ đồ thực hiện hàm XOR 3 đầu vào

Từ biểu thức và bảng trạng thái của cổng XOR, có thể suy ra một số tính chất của hàm XOR như sau:

1. Luật giao hoán:

$$A \oplus B = B \oplus A \quad (2.16)$$

2. Luật kết hợp:

$$(A \oplus B) \oplus C = A \oplus (B \oplus C) \quad (2.17)$$

3. Luật phân phối:

$$A(B \oplus C) = A.B \oplus A.C \quad (2.18)$$

4. Các phép toán của biến và hằng số:

$$A \oplus 1 = \overline{A} \quad (2.19)$$

$$A \oplus 0 = A \quad (2.20)$$

$$A \oplus A = 0 \quad (2.21)$$

$$A \oplus \overline{A} = 1 \quad (2.22)$$

5. Luật đổi chỗ nhân quả

Nếu $A \oplus B = C$ thì

$$A \oplus C = B \quad \text{và} \quad B \oplus C = A \quad (2.23)$$

2.4.3.4 Cổng XNOR

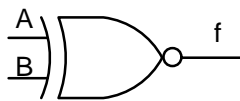
Cổng không hoặc tuyệt đối còn có tên gọi là cổng đồng dấu.

Cổng XNOR được tạo thành khi mắc nối tiếp cổng XOR và cổng NOT.

Biểu thức logic đầu ra:

$$f = AB + \overline{A}\overline{B} \quad \text{hay} \quad f = \overline{A \oplus B} = A \sim B \quad (2.24)$$

Ký hiệu của cổng XNOR 2 đầu vào được trình bày trên hình 2-14.



Hình 2-14. Ký hiệu của cổng XNOR 2 đầu vào theo tiêu chuẩn ANSI

Hoạt động của cổng XNOR 2 đầu vào được mô tả ở bảng trạng thái 2-20.

Có thể xây dựng XNOR nhiều đầu vào bằng cách tương tự như xây dựng XOR nhiều đầu vào

Bảng 2-20. Bảng trạng thái và trạng thái của cổng XNOR 2 đầu vào

A	B	f	A	B	f
0	0	1	L	L	H
0	1	0	L	H	L
1	0	0	H	L	L
1	1	1	H	H	H

a) Bảng trạng thái

b) Bảng chức năng

XOR và XNOR là hai loại cổng có rất nhiều ứng dụng trong kỹ thuật số. Chúng là phần tử chính hợp thành bộ cộng, trừ, so sánh hai số nhị phân v.v...

2.4.4 Tính đa chức năng của cổng NAND, NOR.

Theo tính chất của các cổng logic cơ bản, mọi hàm logic đều có thể thực hiện được nhờ cách kết hợp 3 hàm cơ sở AND, OR và NOT, tức là mọi cấu trúc mạch phức tạp đều có thể tổng hợp được từ các hệ hàm này.

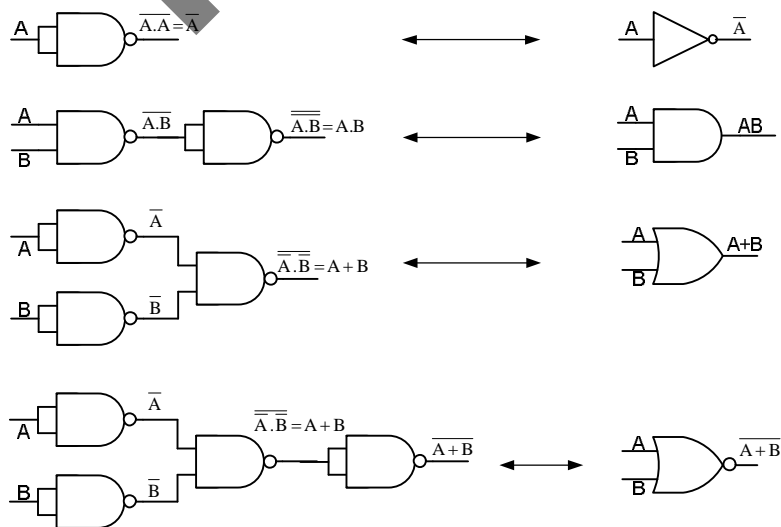
Tuy nhiên, do công nghệ chế tạo các cổng AND, OR và NOT có nhiều điểm khác nhau nên khó có thể thực hiện được trong các mạch tích hợp số. Do vậy, để khắc phục được nhược điểm này phải tìm ra cổng có thể tạo ra hệ hàm đầy đủ. Cổng NAND và cổng NOR có thể thỏa mãn điều kiện này.

Điều này có nghĩa là từ cổng NAND hoặc cổng NOR có thể tạo ra các cổng logic cơ bản khác.

Tính đa chức năng của cổng NAND:

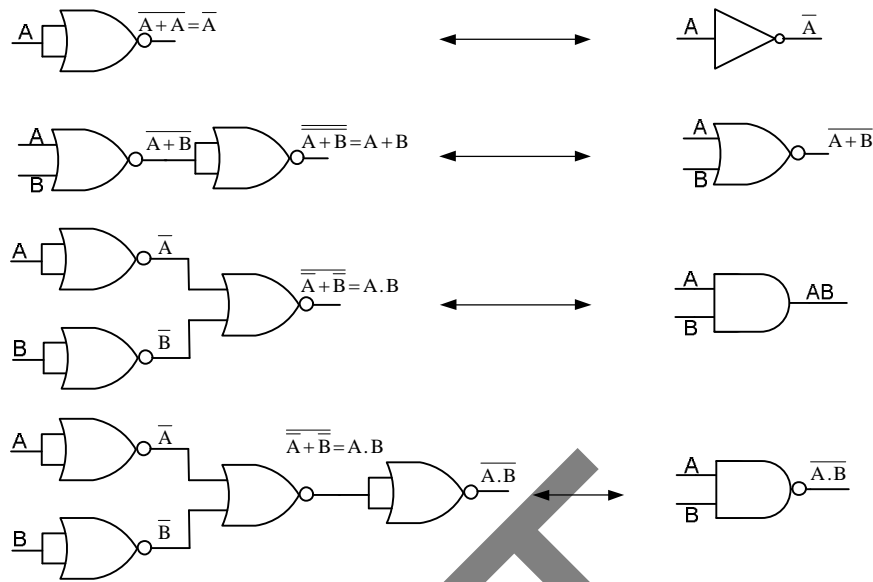
Từ cổng NAND có thể tạo ra các cổng NOT, AND, OR và NOR.

Để tạo được các cổng logic này dùng các định lý Boole để biến đổi.



Hình 2-15. Tính đa chức năng của cổng NAND

Tính đa chức năng của cổng NOR:



Hình 2-16. Tính đa chức năng của cổng NOR

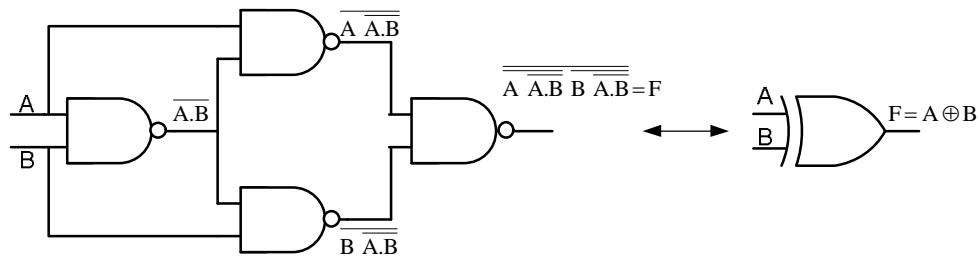
Từ hình 2-15 và 2-16, có thể kết luận là mọi mạch logic tổ hợp có thể xây dựng chỉ từ một loại cổng cơ bản là cổng NAND hoặc cổng NOR. Đây là một đặc điểm quan trọng trên quan điểm tính đồng nhất của công nghệ chế tạo, do vậy, giá thành chi phí giảm, độ tin cậy cao.

Ví dụ 1 : Cho hàm logic $F = A\bar{B} + \bar{A}B$, hãy xây dựng mạch về dạng toàn NAND.

Giải:

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

Từ biểu thức biến đổi trên vẽ được sơ đồ logic hình 2-17.



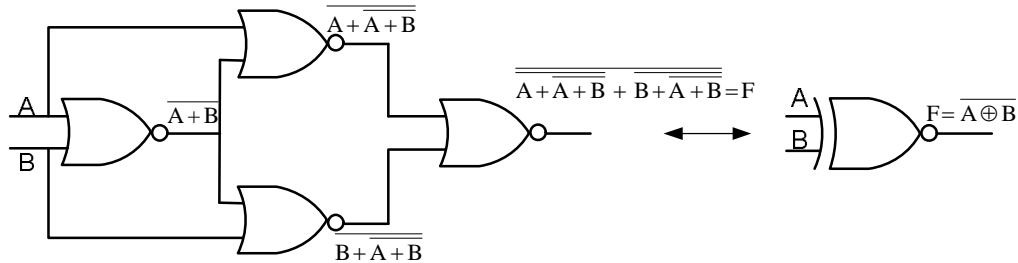
Hình 2-17. Mạch logic thể hiện hàm XOR toàn NAND

Ví dụ 2 : Cho hàm logic $F = AB + \bar{A}\bar{B}$, hãy xây dựng mạch về dạng toàn NOR.

Giải:

$$\begin{aligned}
 F &= A\bar{B} + \bar{A}B = \overline{\overline{A\bar{B} + \bar{A}B}} = \overline{\overline{A\bar{B}} \cdot \overline{\bar{A}B}} = \overline{\overline{A} + B \cdot A + \bar{A} + \bar{B}} = \overline{\overline{A} + \bar{A} + B + \bar{B}} = \overline{1 + 1} = \overline{0} = 1 \\
 &= \overline{\overline{A} + \bar{A} + B + \bar{B}} = \overline{\overline{A} + \bar{A}} + \overline{B + \bar{B}} = \overline{1} + \overline{1} = 0 + 0 = 0 \\
 &= \overline{\overline{A} + \bar{A}} + \overline{B + \bar{B}} = \overline{1} + \overline{1} = 0 + 0 = 0
 \end{aligned}$$

Từ biểu thức biến đổi trên vẽ được sơ đồ logic trên hình 3-18:



Hình 2-18. Mạch logic thể hiện hàm XNOR toàn NOR

TÓM TẮT

Phép toán logic của Leibniz được George Boole bổ sung và thực hiện trong thế kỷ 19 và được gọi là đại số Boole. Hiện nay đại số Boole có giá trị kỹ thuật trong nhiều lĩnh vực như:

- + Các mạch điều khiển điện tử;
- + Các mạch số điện tử;
- + Điều khiển cơ khí;
- + Thiết bị thủy lực...

Shanon đã chứng minh rằng đại số Boole phù hợp để giải quyết các vấn đề kỹ thuật mạch. Từ đó, có thể xây dựng được các hàm logic cơ bản, làm tiền đề cho việc thiết kế các vi mạch cỡ lớn.

Đối với bất kỳ mạch thiết kế logic nào điều căn bản là phải thiết kế một sản phẩm đáp ứng các yêu cầu: giá thấp nhất, yêu cầu không gian thấp nhất, tốc độ hoạt động tối đa, các linh kiện có sẵn, dễ dàng kết nối các linh kiện, dễ dàng thiết kế. Để đạt được các yêu cầu đó cần phải có một vài công cụ để tối thiểu hóa các biểu thức logic. Một số phương pháp được sử dụng phổ biến là phương pháp đại số Boole, bảng Các nô.

Chương 2 cũng đã trình bày cấu trúc, nguyên lý và đặc điểm của cổng logic thường dùng. Xuất phát từ thực tế mạch điện đã vi mạch hoá, nên trọng tâm chú ý nghiên cứu của chúng ta là các cổng được vi mạch hoá.

CÂU HỎI ÔN TẬP

1. Chuyển các hàm logic sau sang dạng chuẩn của minterm?

- a) $A.B.(CD + \overline{A.B})$
- b) $A.(B + \overline{A.C}).(\overline{A + B.C})$
- c) $\overline{A.B.A.C.A.D} + \overline{B.C.B.C} + \overline{C.D}$
- d) $(A + \overline{B}).(\overline{B + C}).(\overline{C + D})$

2. Chuyển các hàm logic sau sang dạng chuẩn của maxterm?

- a) $A.B.(CD + \overline{A.B})$
- b) $A.(B + \overline{A.C}).(\overline{A + B.C})$
- c) $\overline{A.B.A.C.A.D} + \overline{B.C.B.C} + \overline{C.D}$
- d) $(A + \overline{B}).(\overline{B + C}).(\overline{C + D})$

3. Rút gọn hàm sau theo phương pháp dùng bảng Cánhô?

- a) $F(A, B, C, D) = \Pi(1, 4, 6, 9, 10, 11, 14, 15).$
- b) $F(A, B, C, D) = \Sigma(3, 7, 8, 9, 10, 12).$

4. Rút gọn hàm sau theo phương pháp dùng bảng Cánhô?

- a) $F(A, B, C, D) = \Sigma(3, 6, 8, 9, 11, 12) + \Sigma_d(0, 1, 2, 13, 14, 15).$
- b) $F(A, B, C, D) = \Sigma(0, 1, 4, 9, 12, 13) + \Sigma_d(2, 3, 6, 10, 11, 14).$

5. Rút gọn biểu thức sau bằng phương pháp đại số:

- a) $A\overline{B} + BD + CDE + D\overline{A}$
- b) $(\overline{A} + \overline{B} + \overline{C})(\overline{A} + B + C)(A + \overline{B} + C)(A + B + \overline{C})$

6. Rút gọn hàm sau theo phương pháp đại số?

- a) $\overline{CD} + \overline{CD} . \overline{AC} + \overline{D}$
- b) $\overline{ABC} . \overline{AB} + \overline{BC} + \overline{CA}$

7. Phân tích ý nghĩa của việc tối ưu hoá mạch điện của các họ cổng logic? Cho ví dụ minh hoạ?

8. Chứng minh các đẳng thức:

- a. $\overline{A \oplus B} = \overline{A} \overline{B} + AB$
- b. $AB(A \oplus B \oplus C) = ABC$
- c. $A \oplus B \oplus C = \overline{A} \oplus \overline{B} \oplus \overline{C}$

9. Rút gọn:

- a) $\overline{A\overline{B}} + \overline{A}B$

b) $\overline{\overline{A} \overline{B} + A B}$

c) $A B + A \overline{B}$

d) $A + AB$

10. Chứng minh đẳng thức

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

b) $A \oplus B = \overline{A} \oplus \overline{B}$

c) $\overline{A \oplus B} = \overline{A} \oplus \overline{B}$

d) $\overline{A \oplus B} = A \oplus \overline{B}$

11. Chứng minh đẳng thức

a) $\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = AB + AC + BC$

b) $AB + \overline{A}C + BC = AB + \overline{A}C$

12. Rút gọn

a) $AB + BCD + \overline{A}C + \overline{B}C$

b) $\overline{CD} + \overline{C}\overline{D} . \overline{A}C + \overline{D}$

c) $\overline{A}\overline{B}\overline{C} . \overline{A}B + \overline{B}C + \overline{C}A$

d) $\overline{A}C + AB + \overline{B}C + BCDE$

13. Rút gọn biểu thức sau bằng phương pháp đại số:

a) $AB(A \oplus B \oplus C)$

b) $\overline{A} \overline{B} + \overline{A}B + \overline{A} \overline{B} + AB$

c) $\overline{A} \overline{B} + AC + \overline{B}C + \overline{B} \overline{C} \overline{D} + B \overline{C}E + \overline{B}CF$

14. Thể hiện hàm sau bằng mạch cổng NAND 2 lối vào?

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

15. Thể hiện hàm sau bằng mạch cổng NOR 2 lối vào?

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

16. Cho hàm 3 biến :

$$F_1(A, B, C) = A.B.C + \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C}$$

$$F_2(A, B, C) = \overline{A}.B.\overline{C} + A.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + \overline{A}.\overline{B}.C$$

a) Lập bảng trạng thái và bảng Cánhô cho từng hàm?

b) Tìm mối quan hệ nếu có giữa hai hàm?

c) Xác định hàm $F(A, B, C) = F_1.F_2 + \overline{F_1}.\overline{F_2}$

17. Cho hàm logic sau:

$$F(A, B, C, D) = \Sigma(0, 3, 5, 6, 9, 10, 12, 15)$$

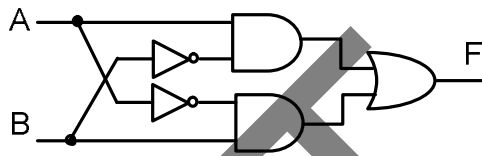
- a) Lập bảng Cánhô và rút gọn hàm F ?
- b) Lập bảng Cánhô và rút gọn hàm \bar{F} ?

18. Cho hàm logic sau:

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

- a) Đưa hàm F về dạng chuẩn tắc
- b) Lập bảng Cánhô và rút gọn hàm F ?
- c) Lập bảng Cánhô và rút gọn hàm \bar{F} ?

19. Cho mạch điện như hình vẽ:



- a) Thiết lập bảng trạng thái mô tả hoạt động của mạch?
- b) Vẽ đồ thị dạng xung tại đầu ra khi dạng xung vào cho tùy chọn?

20. Cho hàm logic:

$$f(A, B, C, D) = \Sigma(0, 2, 5, 6, 7, 8, 10, 13, 15)$$

- a) Viết biểu thức tối giản của hàm
- b) Thực hiện hàm bằng 1 mạch tối ưu toàn NOR 2 lối vào

21. Hãy chứng minh tính chất sau của hàm XOR:

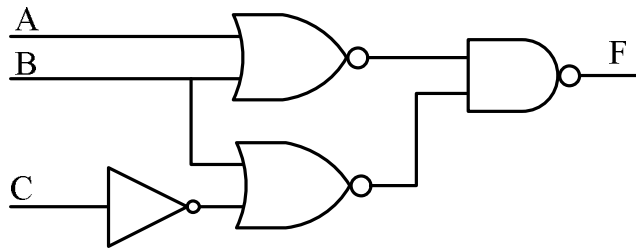
- a) $A \oplus B = B \oplus A$
- b) $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
- c) $A(B \oplus C) = A.B \oplus A.C$
- d) Nếu $A \oplus B = C$ thì $A \oplus C = B$ và $B \oplus C = A$

22. + Hãy chứng minh:

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

+ Hãy vẽ sơ đồ logic của hàm $F = A \oplus B \oplus C \oplus D$

23. Cho hình vẽ sau

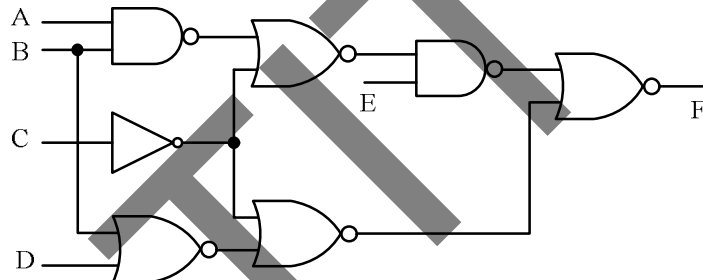


- a) Viết biểu thức hàm ra F.
- b) Xây dựng bảng trạng thái.
- c) Tối ưu hóa mạch.

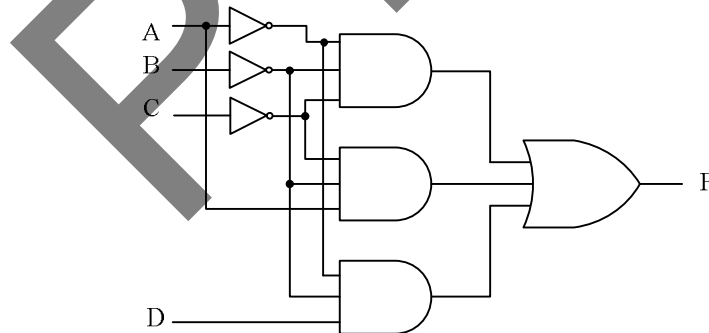
24. Cho hàm logic $F = A.B + B.C + A.C$

- a) Viết lại biểu thức F theo cấu trúc toàn NAND.
- b) Viết lại biểu thức F theo cấu trúc toàn NOR.
- c) Vẽ mạch logic hàm F theo cấu trúc toàn NAND và toàn NOR.

25. Viết biểu thức hàm ra F của mạch điện sau và lập bảng trạng thái tương ứng:



26. Cho hình vẽ sau:



- a) Viết biểu thức hàm F.
- b) Lập bảng trạng thái.
- c) Tối ưu mạch về dạng toàn NAND.

CHƯƠNG 3: MẠCH LOGIC TỔ HỢP

GIỚI THIỆU CHUNG

Các hàm logic được thực hiện nhờ các hệ vật lý gọi là các *hệ logic* hay là các *mạch logic*. Trong chương 3 chúng ta đề cập đến các mạch logic tổ hợp, tức là các mạch mà tín hiệu ở đầu ra chỉ phụ thuộc vào tín hiệu ở đầu vào của mạch tại thời điểm đang xét. Nói cách khác, các tín hiệu ra không phụ thuộc vào "lịch sử" của tín hiệu vào trước đó, nghĩa là các hệ này làm việc theo nguyên tắc không có nhớ. Hoạt động của các mạch tổ hợp được mô tả bằng các bảng trạng thái hoặc bằng các hàm chuyển mạch logic đặc trưng cho quan hệ giữa các đại lượng vào và ra của hệ thống. Về mặt cấu trúc, các mạch tổ hợp không chứa một thiết bị hoặc một phần tử nhớ thông tin nào cả.

Trong chương này đề cập đến các mạch điện cụ thể thực hiện các chức năng khác nhau của hệ thống số. Các mạch điện này được thiết kế dựa trên các cổng logic tổ hợp. Các cổng logic này được tích hợp trong một IC cỡ vừa (MSI) có chứa khoảng vài chục tới vài trăm các cổng logic cơ sở. Những linh kiện này được chế tạo nhằm thực hiện một số các hoạt động thu nhận, truyền tải, biến đổi các dữ liệu thông qua tín hiệu nhị phân, xử lý chúng theo một phương thức nào đó.

Phần đầu của chương giới thiệu cách phân tích và thiết kế các mạch logic tổ hợp đơn giản.

Phần tiếp theo giới thiệu một số mạch tổ hợp thông dụng trong các hệ thống số:

- Mã hoá và giải mã các luồng dữ liệu nhị phân.
- Ghép kênh và phân kênh để chọn hoặc chia tách các luồng số nhị phân theo những yêu cầu nhất định để định tuyến cho chúng trong việc truyền dẫn thông tin.
- Các mạch cộng, trừ, so sánh số nhị phân để đánh giá định tính và định lượng trọng số của các số nhị phân.
- Mạch tạo và kiểm tra tính chẵn lẻ, mạch tạo và giải mã Hamming.
- Đơn vị số học và logic (ALU).

3.1 KHÁI NIỆM CHUNG

Căn cứ vào đặc điểm và chức năng logic, các mạch số được chia thành 2 loại chính: mạch tổ hợp và mạch tuần tự (mạch tuần tự được trình bày ở chương sau).

3.1.1. Đặc điểm cơ bản của mạch tổ hợp

Trong mạch số, mạch tổ hợp là mạch mà trị số ổn định của tín hiệu đầu ra ở thời điểm đang xét chỉ phụ thuộc vào tổ hợp các giá trị tín hiệu đầu vào. Đặc điểm cấu trúc mạch tổ hợp là được cấu trúc nên từ các cổng logic. Vậy các mạch điện cổng ở chương 2 và các mạch logic ở chương 3 đều là các mạch tổ hợp.

3.1.2. Phương pháp biểu diễn chức năng logic

Các phương pháp thường dùng để biểu diễn chức năng logic của mạch tổ hợp là hàm số logic, bảng trạng thái, sử dụng logic, bảng Cac nô (Karnaugh), cũng có khi biểu thị bằng đồ thị thời gian dạng xung.

Đối với vi mạch cỡ nhỏ (SSI) thường biểu diễn bằng hàm logic. Đối với vi mạch cỡ vừa (MSI) thường biểu diễn bằng bảng trạng thái.

3.2 PHÂN TÍCH MẠCH LOGIC TỔ HỢP

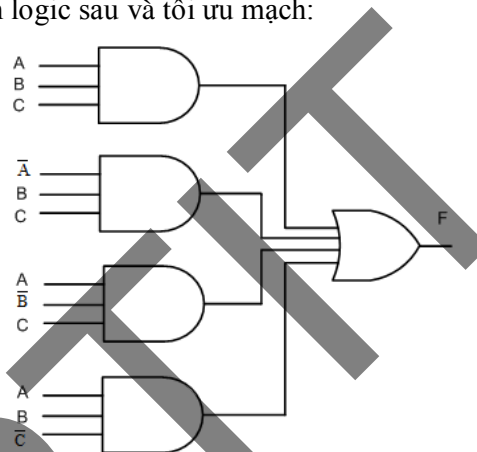
Phân tích mạch logic tổ hợp là đánh giá một mạch điện. Trên cơ sở đó, có thể rút gọn, chuyển đổi dạng thực hiện của mạch điện để có được lời giải tối ưu theo một nghĩa nào đấy.

Mạch tổ hợp có thể bao gồm hai hay nhiều tầng, mức độ phức tạp của của mạch cũng rất khác nhau.

Nếu mạch đơn giản thì tiến hành lập bảng trạng thái, viết biểu thức, rút gọn, tối ưu (nếu cần) và cuối cùng vẽ lại mạch điện.

Nếu mạch phức tạp thì tiến hành phân đoạn mạch để viết biểu thức, sau đó rút gọn, tối ưu (nếu cần) và cuối cùng vẽ lại mạch điện.

Ví dụ: Phân tích mạch logic sau và tối ưu mạch:



Hình 3. 1. Sơ đồ mạch logic

Viết biểu thức hàm và thực hiện rút gọn:

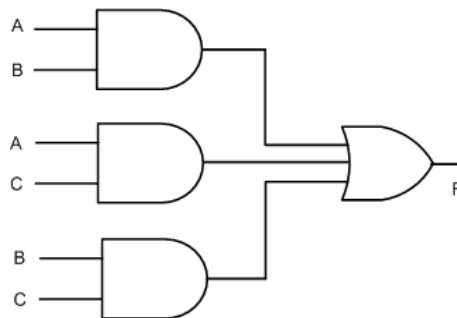
$$F = ABC + \bar{A}BC + A\bar{B}C + ABC$$

$$\Rightarrow F = \bar{A}BC + ABC + A\bar{B}C + ABC + ABC\bar{C} + ABC$$

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

$$= BC + AC + AB$$

Từ đó vẽ được mạch sau:

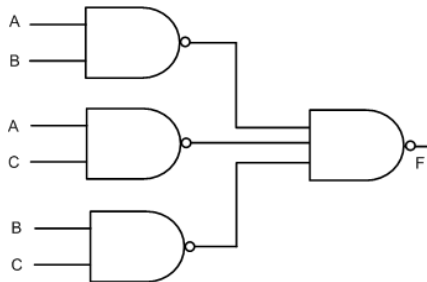


Hình 3. 2. Sơ đồ mạch logic sau khi rút gọn

Thực hiện tối ưu về dạng toàn NAND:

$$\Rightarrow F = \overline{\overline{AB + AC + BC}} = \overline{\overline{AB} \cdot \overline{AC} \cdot \overline{BC}}$$

Từ đó vẽ được mạch sau:

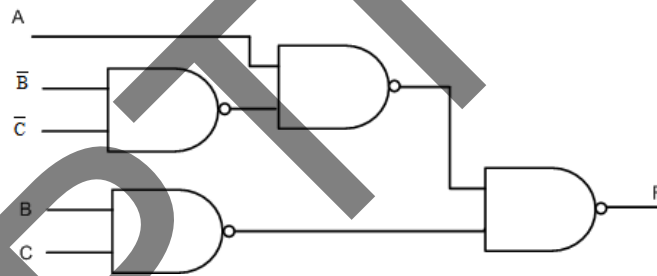


Hình 3. 3. Sơ đồ mạch logic sau khi tối ưu.

Sơ đồ hình 3.3 chưa thực sự tối ưu vì vẫn sử dụng hai loại cổng NAND (NAND 2 lối vào và NAND 3 lối vào), do vậy phải tối ưu về dạng NAND 2 lối vào:

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

Từ đó vẽ được mạch sau:



Hình 3. 4. Sơ đồ mạch logic sử dụng cổng NAND 2 lối vào.

3.3 THIẾT KẾ MẠCH LOGIC TỔ HỢP

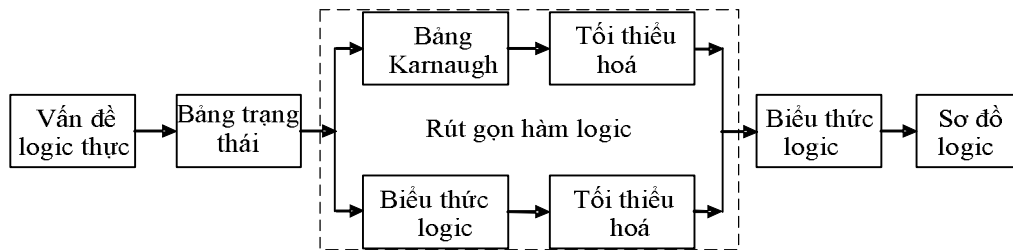
Quá trình thiết kế nói chung của mạch tổ hợp gồm các bước được mô tả trên hình 3.5.

Phương pháp thiết kế logic các mạch tổ hợp là các bước cơ bản tìm ra sơ đồ mạch điện logic từ các yêu cầu nhiệm vụ đã cho.

Các bước của quá trình thiết kế có thể rút gọn trong bốn bước chính:

+ Phân tích yêu cầu

Yêu cầu nhiệm vụ thiết kế của vấn đề logic thực có thể là những yêu cầu trình bày dưới dạng văn bản, cũng có thể là một bài toán logic cụ thể. Nhiệm vụ phân tích là xác định cái nào là biến số đầu vào, cái nào là hàm số đầu ra và mối quan hệ logic giữa hàm và biến.



Hình 3. 5. Các bước thiết kế mạch logic tổ hợp

+ Bảng trạng thái:

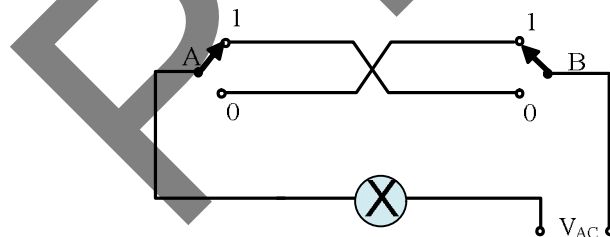
Đầu tiên, từ các yêu cầu cụ thể liệt kê thành bảng biểu diễn quan hệ tương ứng giữa trạng thái tín hiệu đầu vào và trạng thái hàm số đầu ra. Đó là bảng kê chức năng logic gọi tắt là bảng chức năng (hay là bảng trạng thái).

Tiếp theo thay các giá trị logic cho trạng thái, tức là dùng các ký hiệu 0 và 1 thay cho các trạng thái tương ứng của đầu vào và đầu ra. Kết quả được bảng trạng thái.

Ví dụ : Một ngôi nhà hai tầng. Người ta lắp hai chuyển mạch hai chiều tại hai tầng, sao cho ở tầng nào cũng có thể bật hoặc tắt đèn. Hãy thiết kế một mạch logic mô phỏng hệ thống đó?

Lời giải:

+ Nếu ký hiệu hai công tắc là hai biến A, B. Khi ở tầng 1 bật đèn và lên tầng 2 thì tắt đèn đi và ngược lại. Như vậy đèn chỉ có thể sáng ứng với hai tổ hợp chuyển mạch ở vị trí ngược nhau. Còn đèn tắt khi ở vị trí giống nhau. Hệ thống chiếu sáng trong có sơ đồ như hình 3.6.

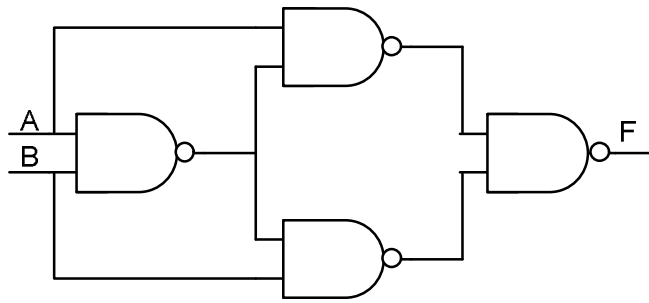


Hình 3. 6. Mạch điện của hệ chiếu sáng

Bảng trạng thái mô tả hoạt động của hệ như chỉ ở bảng 3-3.

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

Bảng 3-3. Bảng trạng thái mô tả hoạt động của hệ chiếu sáng



Hình 3. 7. Sơ đồ logic thể hiện hàm F

Biểu thức của hàm là: $F = \overline{A}B + A\overline{B} = A \oplus B$

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

Đây là hàm XOR. Hàm này có thể được thể hiện bằng nhiều kiểu mạch khác nhau. Hình 3.7 là một dạng sơ đồ thể hiện hàm F.

3.4. MẠCH MÃ HOÁ VÀ GIẢI MÃ

3.4.1 .Một số loại mã thông dụng

3.4.1.1 Các dạng mã nhị thập phân (BCD-Binary Coded Decimal)

Số BCD có vai trò rất quan trọng trong máy tính. Khi đưa số thập phân vào máy tính thì phải chuyển số thập phân đó thành số nhị phân và khi hiển thị phải chuyển số nhị phân thành thập phân. Số BCD thực hiện nhiệm vụ chuyển 10 ký hiệu thập phân thành cụm số nhị phân 4 bit (1 decade). Từ số 10 trở lên thì mỗi ký hiệu số được biểu thị ít nhất bằng 2 decade nhị phân.

Ví dụ: $9 = 1001$; $10 = 0001\ 0000$.

Có rất nhiều cách mã hóa 10 ký hiệu thập phân thành 4 bit nhị phân, nhưng trong phần này chỉ giới thiệu một số mã BCD thông dụng.

Mã BCD tự nhiên (N-BCD: Nature BCD) hay gọi là mã BCD 8421.

Trong mã N-BCD, các chữ số thập phân được nhị phân hoá theo trọng số như nhau 2^3 , 2^2 , 2^1 , 2^0 nên có 6 tổ hợp dư, ứng với các số thập phân 10, 11, 12, 13, 14 và 15. Sự xuất hiện các tổ hợp này trong bản tin được gọi là lỗi dư.

Ngoài mã N-BCD còn có rất nhiều loại mã BCD với các trọng số khác nhau. Bảng 1-4 giới thiệu một số loại mã BCD thường gặp với các chỉ số đi theo mã là các trọng số BCD ở các vị trí tương ứng.

- a) Mã BCD 7421.
- b) Mã BCD 5121.
- c) Mã BCD 2421 (mã Aiken). Mã Aiken được sắp xếp đối xứng, tức là sự phủ định của tất cả các vị trí của một từ mã nhị phân ở một vị trí nào đó sẽ có một từ mã nằm đối xứng có phần bù tương ứng.

Ví dụ:

$$7_{10} = 0\ 1\ 1\ 1_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \text{ với BCD 8421}$$

$$= 1\ 0\ 0\ 0_2 = 1 \times 7 + 0 \times 4 + 0 \times 2 + 0 \times 1 \text{ với BCD } 7421$$

$$= 1\ 0\ 1\ 0_2 = 1 \times 5 + 0 \times 1 + 1 \times 2 + 0 \times 1 \text{ với BCD } 5121$$

$$= 1\ 1\ 0\ 1_2 = 1 \times 2 + 1 \times 4 + 0 \times 2 + 1 \times 1 \text{ với BCD } 2421$$

Ưu điểm:

- Mã BCD có trọng số không thay đổi;
- Được sắp xếp theo quy luật.
- Dễ nhớ.

Nhược điểm:

- Xuất hiện từ mã 0000 và 1111. Tổ hợp này dễ dàng được tạo thành do các nhiễu gây ra nên dễ xuất hiện lỗi kỹ thuật.

Số thập phân	Trọng số của mã BCD				
	8421	7421	5121	2421	4221
0	0 0 0 0	0 0 0 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	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	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	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0	0 1 1 1	0 1 0 0	1 0 0 0
5	0 1 0 1	0 1 0 1	1 0 0 0	1 0 1 1	0 1 1 1
6	0 1 1 0	0 1 1 0	1 0 0 1	1 1 0 0	1 1 0 0
7	0 1 1 1	1 0 0 0	1 0 1 0	1 1 0 1	1 1 0 1
8	1 0 0 0	1 0 0 1	1 0 1 1	1 1 1 0	1 1 1 0
9	1 0 0 1	1 0 1 0	1 1 1 1	1 1 1 1	1 1 1 1

Bảng 3-4. Cấu tạo mã BCD với các trọng số khác nhau.

3.4.1.2. Các dạng mã nhị phân khác

a. Mã dư 3

Do trọng số nhị phân của mỗi vị trí biểu diễn thập phân là tự nhiên nên máy tính có thể thực hiện trực tiếp các phép tính cộng, trừ, nhân, chia theo mã NBCD. Tuy nhiên nhược điểm chính của mã là tồn tại tổ hợp toàn Zero (0), gây khó khăn trong việc đồng bộ khi truyền dẫn tín hiệu.

Vì vậy, người ta sử dụng mã Dư-3 được hình thành từ mã NBCD bằng cách cộng thêm 3_{10} vào mỗi tổ hợp mã. Như vậy, mã không bao gồm tổ hợp toàn Zero. Mã Dư-3 chủ yếu được dùng để truyền dẫn tín hiệu mà không dùng cho việc tính toán trực tiếp.

b. Mã Gray.

Mã Gray còn được gọi là mã cách 1, là loại mã mà các tổ hợp mã kế nhau chỉ khác nhau duy nhất 1 bit. Loại mã này không có tính trọng số. Do đó, giá trị thập phân đã được mã hóa chỉ được giải mã thông qua bảng mã mà không thể tính theo tổng trọng số như đối với mã BCD.

Mã Gray có thể được tổ chức theo nhiều bit. Bởi vậy, có thể đếm theo mã Gray.

Cũng tương tự như mã BCD, ngoài mã Gray chính còn có mã Gray dư-3.

c. Mã Johnson (vòng xoắn).

Mã Johnson sử dụng 5 bit nhị phân để biểu diễn 10 ký hiệu thập phân. Mã này có số bit 1 tăng dần từ trái qua phải cho đến khi đầy, sau đó giảm dần bit 1.

d. Mã vòng

Mã vòng sử dụng 10 bit nhị phân để biểu diễn 10 ký hiệu thập phân với các trọng số 9876543210. Mỗi tổ hợp mã chỉ bao gồm một bit 1 chạy vòng từ phải qua trái.

Số thập phân	Số nhị phân	Mã dư 3	Mã Gray	Mã Gray Dư 3	Mã Johnson	Mã vòng
0	0000	0011	0000	0010	00000	0000000001
1	0001	0100	0001	0110	10000	0000000010
2	0010	0101	0011	0111	11000	0000000100
3	0011	0110	0010	0101	11100	0000001000
4	0100	0111	0110	0100	11110	0000010000
5	0101	1000	0111	1100	11111	0000100000
6	0110	1001	0101	1101	01111	0001000000
7	0111	1010	0100	1111	00111	0010000000
8	1000	1011	1100	1110	00011	0100000000
9	1001	1100	1101	1010	00001	1000000000

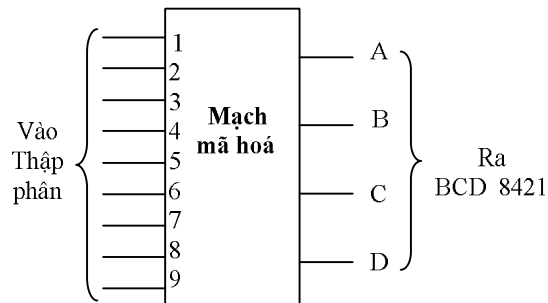
Bảng 3-5. Cấu tạo của một số mã nhị phân thông dụng.

3.4.2. Các mạch mã hoá:

Nói một cách tổng quát, mã hoá là dùng văn tự, ký hiệu hay mã để biểu thị một đối tượng xác định. Các hệ thống điện tử số (như máy tính) chỉ tiếp nhận thông tin dưới dạng nhị phân, mã nhị phân có hai chữ số 0 và 1, vì vậy các số liệu, các lệnh cho máy tính thực hiện đều phải được biểu diễn dưới dạng nhị phân. Sau khi đã được máy tính xử lý cần thiết phải giải mã để đưa thông tin cho con người (hệ thập phân). Mã nhị phân n bit có 2^n trạng thái, có thể biểu thị 2^n tín hiệu. Vậy để tiến hành mã hoá N tín hiệu, cần sử dụng n bit sao cho $2^n \geq N$.

Bộ mã hoá là mạch điện thao tác mã hoá, có nhiều bộ mã hoá khác nhau, bộ mã hoá nhị phân, bộ mã hoá nhị - thập phân, bộ mã hoá ưu tiên v.v.

3.4.2.1. Bộ mã hoá thập phân sang BCD8421:



Hình 3. 8. Sơ đồ khối mạch mã hóa

Bộ mã hoá nhị - thập phân là mạch điện chuyển mã hệ thập phân bao gồm 10 chữ số, đầu ra là nhóm mã số nhị phân gọi là mã nhị phân BCD (Binary Coded Decimal). Căn cứ vào công thức $2^n \geq N = 10$ nên chọn $n = 3$. Mã nhị phân 4 bit có 16 tổ hợp (từ mã). Chỉ cần chọn 10 từ mã tuỳ ý trong số đó là đủ biểu thị 10 tín hiệu đầu vào, vậy có rất nhiều phương án tạo ra bộ mã hóa BCD.

Dưới đây là bảng mã hoá BCD – 8421 rất thường dùng:

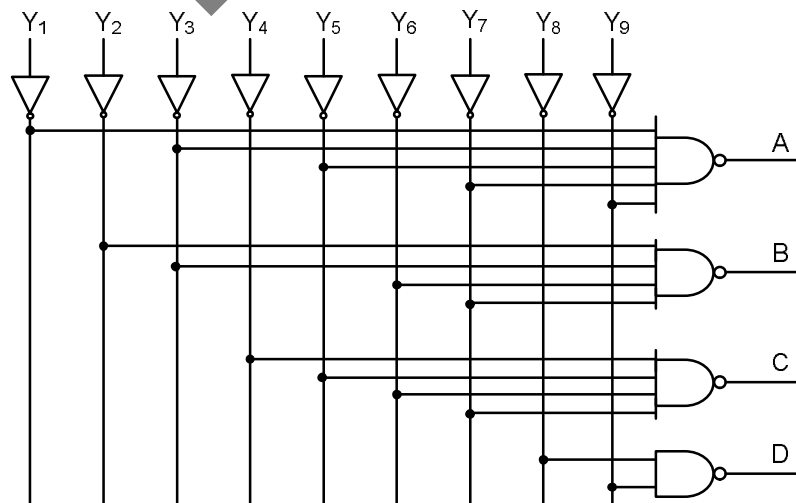
Số thập phân	D	C	B	A
0 (Y_0)	0	0	0	0
1 (Y_1)	0	0	0	1
2 (Y_2)	0	0	1	0
3 (Y_3)	0	0	1	1
4 (Y_4)	0	1	0	0
5 (Y_5)	0	1	0	1
6 (Y_6)	0	1	1	0
7 (Y_7)	0	1	1	1
8 (Y_8)	1	0	0	0
9 (Y_9)	1	0	0	1

Bảng 3-3. Bảng mã hoá BCD – 8421:

Từ bảng trạng thái 3-4, tìm được biểu thức đầu ra sau

$$\left. \begin{aligned} D &= Y_8 + Y_9 = \overline{Y_8} \cdot \overline{Y_9} \\ C &= Y_4 + Y_5 + Y_6 + Y_7 = \overline{Y_4} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7} \\ B &= Y_2 + Y_3 + Y_6 + Y_7 = \overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_6} \cdot \overline{Y_7} \\ A &= Y_1 + Y_3 + Y_5 + Y_7 + Y_8 = \overline{Y_1} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_7} \cdot \overline{Y_8} \end{aligned} \right\} \quad (3.1)$$

Sơ đồ logic của bộ mã hoá BCD – 8421 được trình bày trên hình 3.9.



Hình 3. 9. Sơ đồ logic của bộ mã hoá nhị - thập phân.

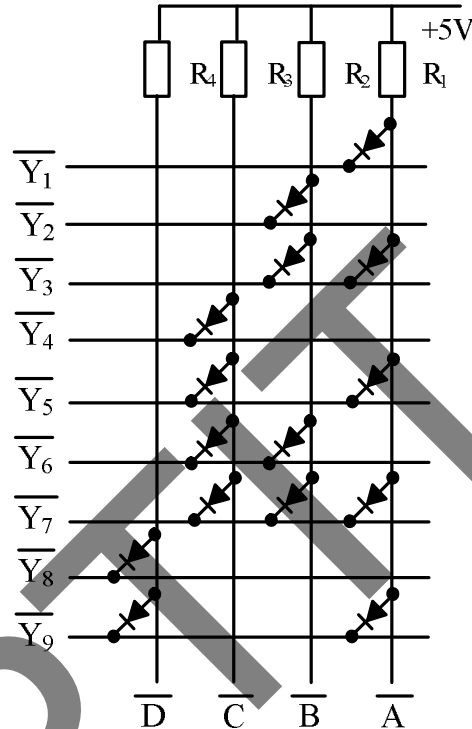
Từ hệ phương trình trên có thể viết lại như sau (dùng định lý DeMorgan) và dùng ma trận diode (cổng AND) để xây dựng mạch (hình 3.10).

$$\overline{D} = \overline{Y_8 + Y_9} = \overline{Y_8} \cdot \overline{Y_9}$$

$$\overline{C} = \overline{Y_4 + Y_5 + Y_6 + Y_7} = \overline{Y_4} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7}$$

$$\overline{B} = \overline{Y_2 + Y_3 + Y_6 + Y_7} = \overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_6} \cdot \overline{Y_7}$$

$$\overline{A} = \overline{Y_1 + Y_3 + Y_5 + Y_7 + Y_9} = \overline{Y_1} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_7} \cdot \overline{Y_9}$$



Hình 3. 10. Mạch mã hóa dùng diode

3.4.2.2. Bộ mã hoá ưu tiên

Trong các bộ mã hoá thông thường, tín hiệu đầu vào tồn tại độc lập (không có tình huống có 2 tín hiệu trở lên đồng thời tác động). Bộ mã hoá ưu tiên thì khác, có thể có nhiều tín hiệu đồng thời đưa đến, nhưng mạch điện chỉ tiến hành mã hoá tín hiệu điện đầu vào nào có mức ưu tiên cao nhất ở thời điểm đó.

Chúng ta xem xét nguyên lý hoạt động và quá trình thiết kế bộ mã hoá ưu tiên qua ví dụ sau:

Hãy thiết kế một mạch logic để mã hoá nhị phân đối với 10 tín hiệu vào Y_0, Y_1, \dots, Y_9 sao cho mức độ ưu tiên cao nhất giảm dần từ Y_9 đến Y_0 . Nếu có nhiều tín hiệu đồng thời xuất hiện ở đầu vào thì tín hiệu nào có mức ưu tiên cao nhất trong số đó mới được mã hoá, giả thiết cả tín hiệu đầu vào và tín hiệu đầu ra đều tích cực ở mức thấp.

Theo yêu cầu trên, căn cứ công thức $2^n \geq N = 10$, vậy dùng mã nhị phân $n = 4$ bit.

Bảng 3-5 bảng trạng thái bộ mã hoá ưu tiên.

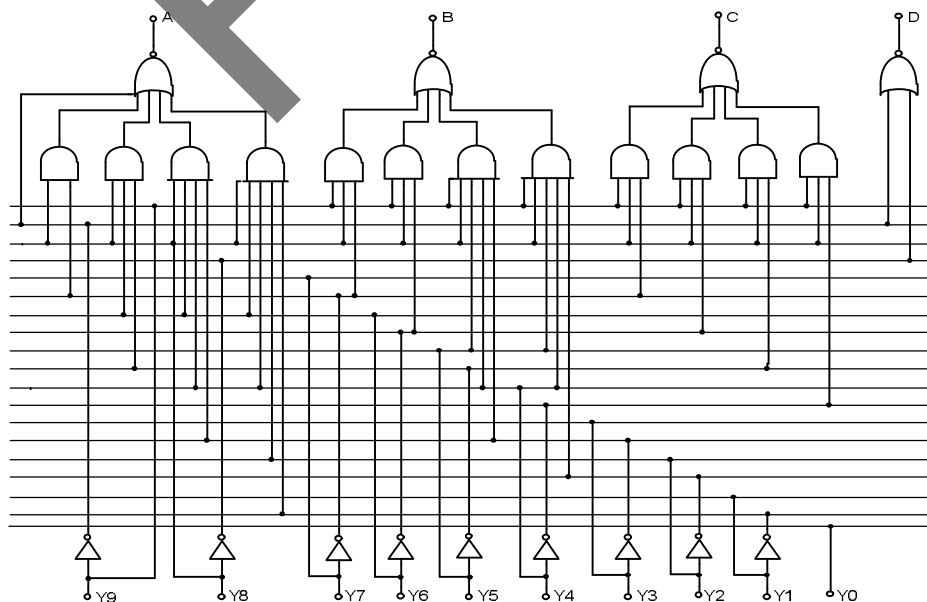
Y ₉	Y ₈	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀	D	C	B	A
1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	0	x	1	1	1	0
1	1	1	1	1	1	1	0	x	x	1	1	0	1
1	1	1	1	1	1	0	x	x	x	1	1	0	0
1	1	1	1	1	0	x	x	x	x	1	0	1	1
1	1	1	1	0	x	x	x	x	x	1	0	1	0
1	1	1	0	x	x	x	x	x	x	1	0	0	1
1	1	0	x	x	x	x	x	x	x	1	0	0	0
1	0	x	x	x	x	x	x	x	x	0	1	1	1
0	x	x	x	x	x	x	x	x	x	0	1	1	0

Bảng 3-5- Bảng trạng thái bộ mã hoá ưu tiên.

$$\begin{aligned}
 \bar{D} &= \bar{Y}_9 + Y_9 \bar{Y}_8 = \bar{Y}_9 + \bar{Y}_8 & \rightarrow D &= \overline{\bar{Y}_9 + \bar{Y}_8} \\
 \bar{C} &= Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 \bar{Y}_6 + Y_9 Y_8 \bar{Y}_5 + Y_9 Y_8 \bar{Y}_4 & \rightarrow C &= \overline{Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 \bar{Y}_6 + Y_9 Y_8 \bar{Y}_5 + Y_9 Y_8 \bar{Y}_4} \\
 \bar{B} &= Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 \bar{Y}_6 + Y_9 Y_8 Y_5 Y_4 \bar{Y}_3 + Y_9 Y_8 Y_5 Y_4 \bar{Y}_2 & & \\
 &\rightarrow B = \overline{Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 \bar{Y}_6 + Y_9 Y_8 Y_5 Y_4 \bar{Y}_3 + Y_9 Y_8 Y_5 Y_4 \bar{Y}_2} \\
 \bar{A} &= \bar{Y}_9 + Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 Y_7 Y_6 \bar{Y}_5 + Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 \bar{Y}_3 + Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 \bar{Y}_1 \rightarrow \\
 &\rightarrow A = \overline{\bar{Y}_9 + Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 Y_7 Y_6 \bar{Y}_5 + Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 \bar{Y}_3 + Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 \bar{Y}_1}
 \end{aligned} \quad (3.2)$$

Theo bảng 3-5, khi có nhiều tín hiệu cùng tác động, thì tín hiệu có mức ưu tiên cao nhất được mã hoá, các tín hiệu khác tương ứng x (dù là 1 hay 0) cũng không có tác dụng. Kết quả được các hàm logic đầu ra dưới dạng NORAND công thức (3.2).

Sơ đồ logic hình 3.11 thoả mãn yêu cầu trên



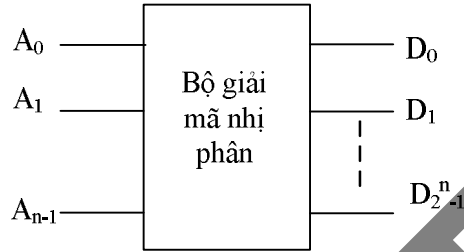
Hình 3. 11. Bộ mã hoá ưu tiên

3.4.3. Các bộ giải mã

Giải mã là một quá trình phiên dịch hàm đã được gán bằng một từ mã. Mạch điện thực hiện giải mã gọi là bộ giải mã. Bộ giải mã biến đổi từ mã thành tín hiệu ở đầu ra. Có nhiều bộ giải mã, nhưng chúng đều có nguyên tắc hoạt động và phương pháp thiết kế tương tự nhau. Sau đây chỉ xét các bộ giải mã thông dụng.

3.4.3.1. Bộ giải mã nhị phân

Bộ giải mã nhị phân còn có tên là bộ giải mã "1 từ n ", bộ giải mã địa chỉ hoặc bộ chọn địa chỉ nhị phân.



Hình 3. 12. Sơ đồ khối bộ giải mã nhị phân

Chức năng của bộ giải mã nhị phân là lựa chọn duy nhất một đầu ra (lấy giá trị 1 hoặc 0), khi tác động tới đầu vào một số nhị phân. Như vậy, nếu số nhị phân là n bit (n đầu vào) sẽ nhận diện được 2^n địa chỉ khác nhau (trên 2^n đầu ra). Nói khác đi, mạch chọn địa chỉ nhị phân là một mạch logic tổ hợp có n đầu vào và 2^n đầu ra, nếu tác động tới đầu vào một số nhị phân thì chỉ duy nhất một đầu ra được lựa chọn, lấy giá trị 1 (tích cực cao) hoặc 0 (tích cực thấp), các đầu ra còn lại đều không được lựa chọn, lấy giá trị 0 hoặc 1. Sơ đồ khối tổng quát của bộ chọn địa chỉ nhị phân như chỉ ở hình 3.12.

Ví dụ: Xây dựng mạch giải mã nhị phân 2 vào 4 ra và một đầu vào điều khiển E.

Lời giải: Lập bảng trạng thái 3-6.

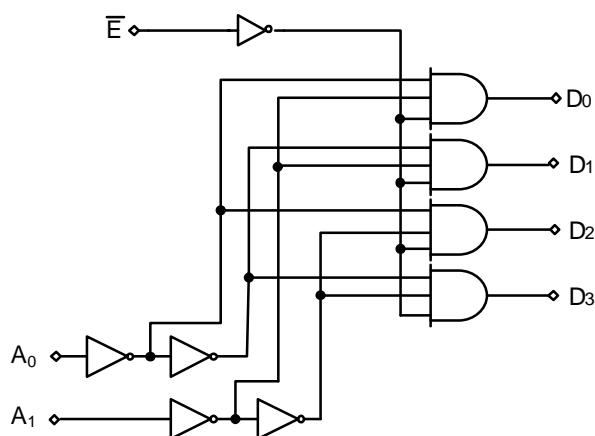
E	A ₁	A ₀	D ₀	D ₁	D ₂	D ₃
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Bảng 3-6. Bảng trạng thái

Từ bảng trạng thái viết được biểu thức hàm ra như sau:

$$\left. \begin{aligned} D_0 &= E \cdot \overline{A_1} \cdot \overline{A_0} \\ D_1 &= E \cdot \overline{A_1} \cdot A_0 \\ D_2 &= E \cdot A_1 \cdot \overline{A_0} \\ D_3 &= E \cdot A_1 \cdot A_0 \end{aligned} \right\} \quad (3.3)$$

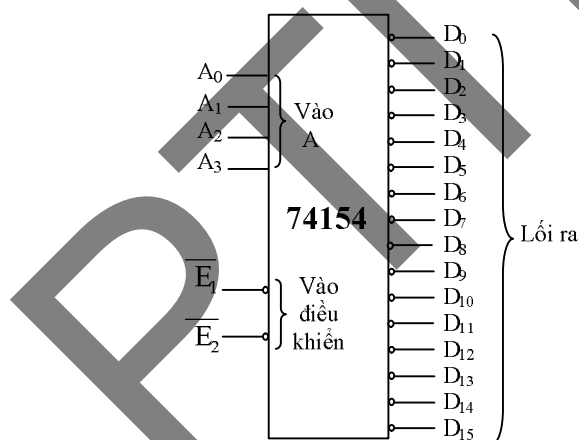
Từ biểu thức (3.3) vẽ được sơ đồ logic như hình 3.13.



Hình 3. 13. Sơ đồ logic mạch giải mã địa chỉ 2 vào – 4 ra.

IC 74154 là một bộ chọn địa chỉ nhị phân 4 vào 16 ra. Ký hiệu logic của nó được chỉ ra ở hình 3.14. Các đầu vào E_1 , E_2 , hoạt động theo tích cực thấp thường được sử dụng để mở rộng dung lượng hoặc thay đổi chức năng logic của bộ chọn địa chỉ.

Có thể mở rộng dung lượng bộ chọn địa chỉ nhị phân bằng cách ghép các IC có dung lượng nhỏ lại với nhau.



Hình 3. 14. Sơ đồ chân của IC 74154

3.4.3.2. Mạch giải mã 7 đoạn

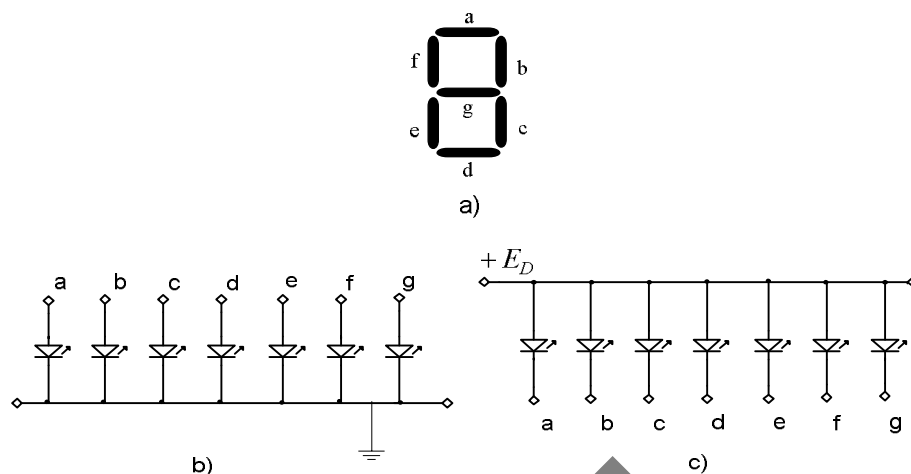
Mã nhị phân BCD được chuyển sang thập phân và hiển thị các số thập phân bằng 7 đoạn sáng. 7 đoạn sáng có thể là LED hoặc tinh thể lỏng (LCD), ứng với mỗi tổ hợp xác định các thanh sáng sẽ hiển thị một chữ số ở hệ đếm thập phân.

Đối với LED, mỗi đoạn là một diode phát quang, khi có dòng điện đi qua đủ lớn (từ 5mA đến 30mA) thì đoạn tương ứng sẽ sáng.

Ngoài 7 đoạn sáng chính, mỗi LED có thêm một diode để biểu thị dấu phân số khi cần thiết. LED có 2 loại chính: LED anốt chung và LED catốt chung.

Hình 3.15 mô tả cấu trúc của đèn hiển thị số 7 đoạn sáng dùng diode phát quang, cách ký hiệu các đoạn bằng các chữ cái a, b, c, d, e, f, g. Bảy đoạn là 7 diode phát quang.

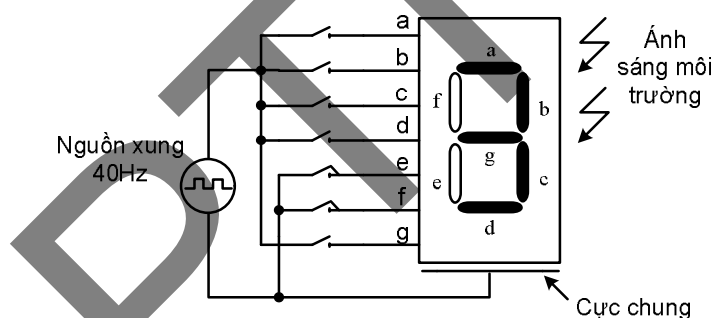
Đối với LED catốt chung, catốt được nối với nhau và nối đất. Còn LED anốt chung thì anốt được nối với nhau và nối với + 5V.



Hình 3. 15. a) Cấu trúc của chỉ thị số 7 đoạn, b) LED catốt chung, c) LED anốt chung

Các bộ giải mã có các đầu ra tích cực thấp, sử dụng LED anốt chung, đầu ra của bộ giải mã được nối với các catốt qua các điện trở hạn chế dòng.

Dụng cụ hiển thị bằng tinh thể lỏng, gọi tắt là LCD (Liquid Crystal Display).



Hình 3. 16. Màn hình LCD và phương pháp kích thích sáng số 3 thập phân

LCD chỉ hoạt động khi có ánh sáng phân cực chiếu vào. Để đạt được mục đích này, phía trước các đoạn cần có một kính phân cực ánh sáng. Các đoạn không hoạt động sẽ phản xạ đối với luồng ánh sáng phân cực này và do đó không nhìn thấy chúng. Ngược lại các đoạn hoạt động không phản xạ nên bị đen lại. Ưu điểm nổi bật của LCD là công suất tiêu thụ rất nhỏ. Đây là ưu điểm quan trọng đặc biệt với các thiết bị bằng pin nhỏ (như máy tính cầm, máy tính xách tay...). Tuy nhiên, nhược điểm là độ sáng yếu vì LCD tận dụng ánh sáng có sẵn (nguồn tự nhiên của môi trường hay nguồn sáng nhỏ chiếu hậu) và điều khiển phản xạ của ánh sáng này.

LCD được bố trí kiểu 7 đoạn như LED làm việc với điện thế xoay chiều (25Hz đến 60Hz) với dòng tiêu thụ rất nhỏ.

3.4.3.3. Thiết kế bộ giải mã BCD sang 7 đoạn

Các đầu vào D,C,B, A là mã BCD 8421. Trong đó 6 trạng thái từ 1010 ÷ 1111 không được sử dụng, đánh dấu chéo “X” để xử lý tối thiểu hoá. Tín hiệu đầu ra a ÷ g dùng để kích sáng LED tương ứng của hiển thị 7 thanh.

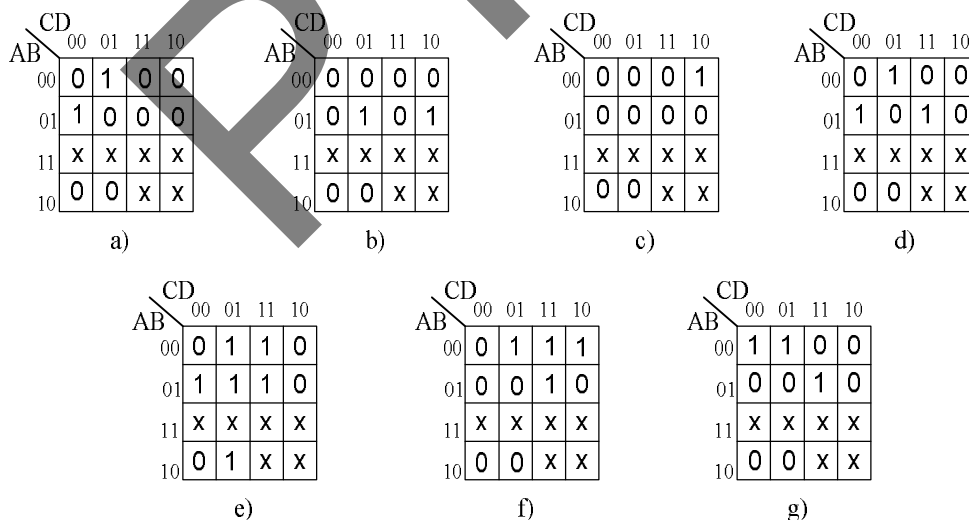
Ở đây xét mạch giải mã có các đầu ra tích cực ở mức thấp, dùng LED anốt chung.

Bảng 3-7 là bảng trạng thái bộ giải mã BCD sang bảy đoạn.

A	B	C	D	a	b	c	d	e	f	g	Số được hiển thị
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	1	0	0	9

Bảng 3-7. Bảng trạng thái bộ giải mã

Hình 3.17 trình bày các bảng Karnaugh. Các ô “x” có thể tùy chọn, trong khi tối thiểu hoá. Tối thiểu hoá dạng minterm, đối với các ô trong bảng Karnaugh có giá trị 0 để xác định hàm đảo:



Hình 3. 17. Bảng Karnaugh hàm logic đầu ra của các LED hiển thị a, b, c, d, e, f, g

(a) $\bar{a} = A + C + BD + \bar{B}.\bar{D}$ (dạng OR-AND)

$a = \overline{A + C + BD + \bar{B}.\bar{D}}$ (dạng NOR-AND)

(b) $\bar{b} = \bar{B} + C.D + \bar{C}.\bar{D}$

$$b = \overline{\overline{B} + C.D + \overline{C}.D}$$

$$(c) \quad \overline{c} = B + \overline{C} + D$$

$$c = \overline{B + \overline{C} + D}$$

$$(d) \quad \overline{d} = A + \overline{B}.D + \overline{B}.C + B.\overline{C}.D + C.\overline{D}$$

$$d = \overline{A + \overline{B}.D + \overline{B}.C + B.\overline{C}.D + C.\overline{D}}$$

$$(e) \quad \overline{e} = \overline{B}.D + C.\overline{D}$$

$$e = \overline{\overline{B}.D + C.\overline{D}}$$

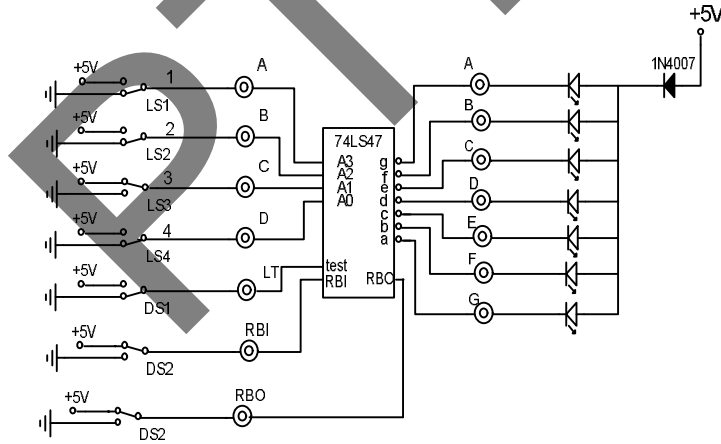
$$(f) \quad \overline{f} = A + B.\overline{D} + B.\overline{C} + \overline{C}.\overline{D}$$

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

$$(g) \quad \overline{g} = A + C.\overline{D} + \overline{B}.C + B.\overline{C}$$

$$g = \overline{A + C.\overline{D} + \overline{B}.C + B.\overline{C}}$$

Xuất phát từ các hàm logic tối thiểu hoá sơ đồ logic được trình bày trên hình 3.18.



Hình 3. 18. Bộ giải mã BCD sang 7 đoạn.

Xoá các số 0 không có nghĩa và xoá các số không cần thiết của LED hiển thị.

Việc hiển thị có thể cần nhiều chữ số, dẫn đến trường hợp phần lớn số chữ số không có tín hiệu trên đầu vào, lúc này chúng hiển thị giá trị 0. Điều này gây khó khăn cho người đọc và tiêu tốn năng lượng của nguồn nuôi. Hoặc đôi khi không cần hiển thị các số 0 cần thiết. Để thực hiện các mục đích trên, trong thiết kế người ta đưa vào các chân điều khiển LED ở trạng thái tắt.

Ví dụ đối với IC 7447 người ta thiết kế có 2 chân RBO (Ripple Blanking Output – Đầu ra xoá nối tiếp) và RBI (Ripple Blanking Input – đầu vào xoá nối tiếp).

Khi $RBO = 0$ thì LED tắt. Khi $DCBA = 0000$ tương ứng hiển thị số 0. Nếu $RBI = 0$ thì LED tắt, còn các đầu vào ở giá trị khác, đầu ra hiển thị các chữ số khác 0 vẫn sáng bình thường.

3.4.4. Các bộ biến mã

Có nhiều loại mã nhị phân được dùng trong các hệ thống kỹ thuật số. Một vài mã này là nhị phân được tạo mã sang BCD, BCD dư 3, Gray, Gray dư 3, bát phân hoặc thập lục phân... Thông thường người ta yêu cầu đổi từ mã này sang mã khác. Ví dụ đầu vào của hệ thống kỹ thuật số có thể là mã NBCD, đầu ra có thể là LED 7 đoạn, hệ thống này xử lý dữ liệu theo dạng nhị phân nên dữ liệu phải chuyển từ mã NBCD sang mã nhị phân.

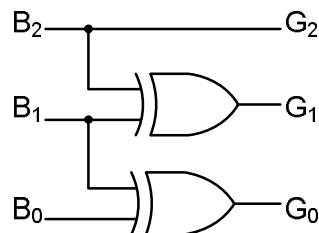
Các bộ biến mã có thể được thiết kế bằng cách sử dụng các cổng logic, các bộ ghép kênh và phân kênh. Tuy nhiên trên thực tế có một số loại IC có sẵn để thực hiện chức năng này.

3.4.4.1. Bộ biến mã từ nhị phân sang Gray

Bảng 3-9 là bảng trạng thái mô tả mối quan hệ giữa mã nhị phân và mã Gray 3 bit.

Thập phân	Nhị phân			Gray		
	B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Bảng 3-9. Bảng trạng thái mạch biến mã



Hình 3. 19. Mạch điện thực hiện bộ biến mã từ nhị phân sang Gray

Từ bảng trạng thái 3-9 viết được biểu thức của hàm ra như sau:

$$\left. \begin{aligned} G_2 &= B_2 \\ G_1 &= B_2 \oplus B_1 \\ G_0 &= B_1 \oplus B_0 \end{aligned} \right\} \quad (3.4)$$

Từ đó, vẽ được sơ đồ logic như trên hình 3.19.

3.4.4.2. Bộ biến mã từ Gray sang nhị phân

Bảng 3-10 là bảng trạng thái mô tả mối quan hệ giữa mã nhị phân và mã Gray 3 bit.

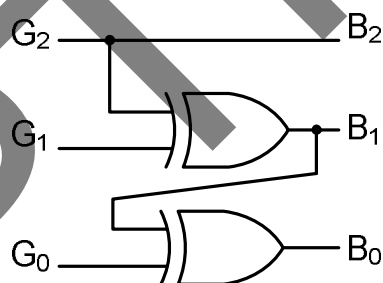
Thập phân	Gray			Nhị phân		
	G ₂	G ₁	G ₀	B ₂	B ₁	B ₀
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	1	0	1	0
3	0	1	0	0	1	1
4	1	1	0	1	0	0
5	1	1	1	1	0	1
6	1	0	1	1	1	0
7	1	0	0	1	1	1

Bảng 3-10. Bảng trạng thái mạch biến mã

Từ bảng trạng thái 3-10, viết được biểu thức của hàm ra như sau:

$$\left. \begin{aligned} B_2 &= G_2 \\ B_1 &= B_2 \oplus G_1 \\ B_0 &= B_1 \oplus G_0 \end{aligned} \right\} \quad (3.5)$$

Từ đó, vẽ được sơ đồ logic như trên hình 3.20



Hình 3. 20. Mạch điện thực hiện bộ biến mã từ Gray sang nhị phân

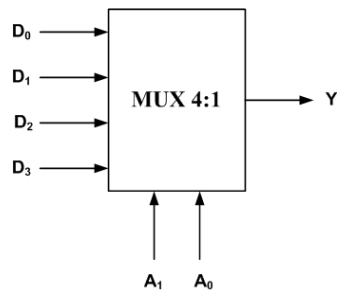
3.5. BỘ HỢP KÊNH VÀ PHÂN KÊNH

3.5.1 Bộ hợp kênh (MUX-Multiplexer)

Bộ hợp kênh còn gọi là bộ dồn kênh (hay bộ ghép kênh), nó cũng được gọi là bộ chọn dữ liệu (Data Selector). Bộ ghép kênh thực hiện dưới dạng một mạng các cổng NORAND (hay ORAND). Chức năng logic cơ bản của bộ ghép kênh là dưới sự điều khiển của tín hiệu chọn (n đầu vào điều khiển) thực hiện chọn ra kênh nào đó (trong số 2^n kênh đầu vào) để nối thông tín hiệu đầu vào được chọn đến đầu ra.

Để người dùng không bị nhầm lẫn trong việc xác định địa chỉ kênh, các nhà sản xuất vi mạch đã dùng các chỉ số kênh 0, 1, 2, ... trùng với giá trị thập phân của tổ hợp nhị phân tương ứng của các đầu vào điều khiển.

Hình 3.21 trình bày sơ đồ khối của bộ hợp kênh 4 đầu vào và 1 đầu ra dữ liệu.



Sơ đồ khối MUX 4:1

Hình 3. 21. Bộ hợp kênh 4 vào – 1 ra

Từ sơ đồ khối, xây dựng bảng trạng thái của MUX 4:1.

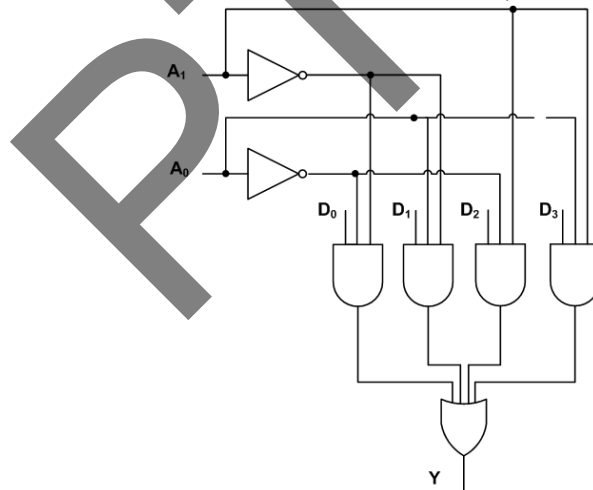
A ₁	A ₀	Y
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

Bảng 3-11. Bảng chức năng của bộ hợp kênh 4:1.

Từ bảng trạng thái, viết được biểu thức hàm ra:

$$Y = D_0 \cdot \overline{A_1} \cdot \overline{A_0} + D_1 \cdot \overline{A_1} \cdot A_0 + D_2 \cdot A_1 \cdot \overline{A_0} + D_3 \cdot A_1 \cdot A_0$$

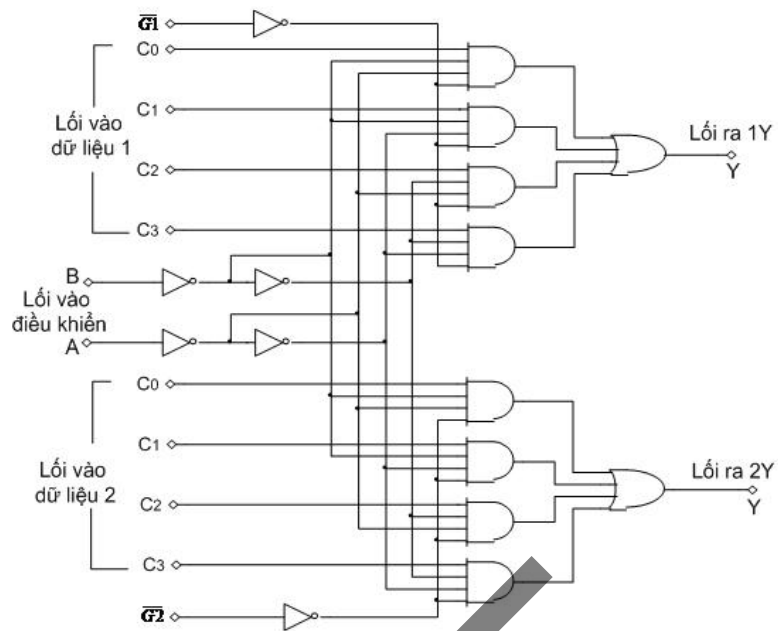
Mạch điện hình 3.22 thể hiện hàm Y.



Mạch hợp kênh 4:1

Hình 3. 22. Sơ đồ logic mạch hợp kênh 4:1

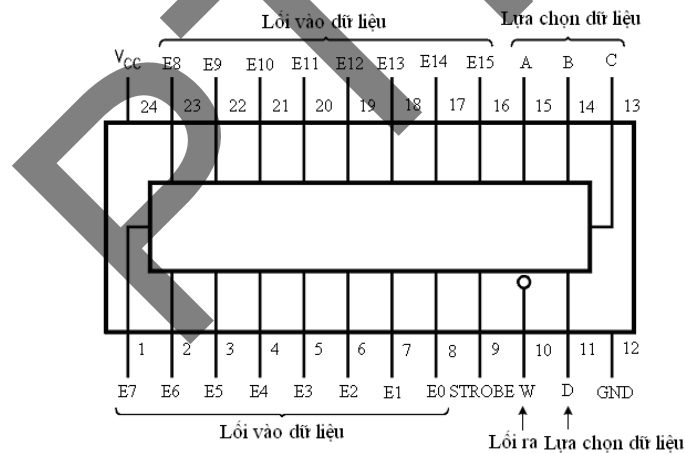
Trong thực tế người ta chế tạo các bộ ghép kênh có 4, 8 hoặc 16 đầu vào dữ liệu. Hình 3.23 trình bày sơ đồ logic của vi mạch 74LS153. Trong vi mạch gồm 2 bộ ghép kênh có 4 đường vào dữ liệu, kí hiệu là C₀, C₁, C₂, C₃ và một đường ra Y. Cả hai bộ ghép kênh đều có chung 2 đầu vào điều khiển A, B, mỗi bộ ghép kênh đều có đầu vào cho phép \overline{G} riêng. Mạch thuộc họ logic TTL, chân 16 là nguồn nuôi V_{CC}: + 5V, chân 8 là đất (GND): 0V.



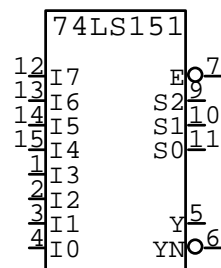
Hình 3. 23. Sơ đồ logic của bộ ghép kênh (4:1)

Vi mạch 74150 có ký hiệu logic trên hình 3.24 là bộ ghép kênh có 16 đường vào dữ liệu và 4 đường vào điều khiển chọn kênh A, B, C, D và một đầu vào cho phép \overline{G} . Đầu ra dữ liệu ký hiệu là W.

IC 74151 là bộ ghép kênh 8 đường có hai đầu ra Y và W là đảo của Y (hình 3.25).



Hình 3. 24. Ký hiệu logic của bộ ghép kênh 74150



Hình 3. 25. Ký hiệu logic của bộ ghép kênh 74151

Đầu vào điều khiển			Cho phép	Đầu ra	Đầu ra
C	B	A	\overline{G}	Y	$W = \overline{Y}$
L	L	L	L	D_0	\overline{D}_0
L	L	H	L	D_1	\overline{D}_1
L	H	L	L	D_2	\overline{D}_2
L	H	H	L	D_3	\overline{D}_3
H	L	L	L	D_4	\overline{D}_4
H	L	H	L	D_5	\overline{D}_5
H	H	L	L	D_6	\overline{D}_6
H	H	H	L	D_7	\overline{D}_7
x	x	x	H	L	H

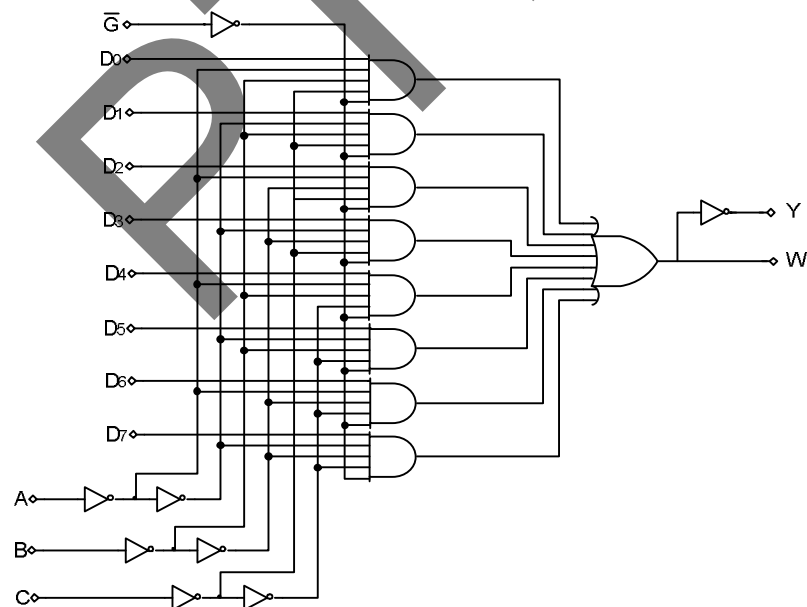
Bảng 3-12. Bảng chức năng của bộ ghép kênh 74151.

Từ bảng chức năng 3-12, suy ra bảng trạng thái với cách chọn mức thấp L là 0, mức cao H là 1 và có thể suy ra biểu thức logic sau:

$$Y = G \cdot \left(\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D_0 + A \cdot \overline{B} \cdot \overline{C} \cdot D_1 + \overline{A} \cdot B \cdot \overline{C} \cdot D_2 + A \cdot B \cdot \overline{C} \cdot D_3 + \right. \\ \left. + \overline{A} \cdot \overline{B} \cdot C \cdot D_4 + A \cdot \overline{B} \cdot C \cdot D_5 + \overline{A} \cdot B \cdot C \cdot D_6 + A \cdot B \cdot C \cdot D_7 \right) \quad (3.6)$$

$$W = \overline{Y}$$

Từ biểu thức logic (3.6) có sơ đồ logic của bộ ghép kênh trên hình 3.26.

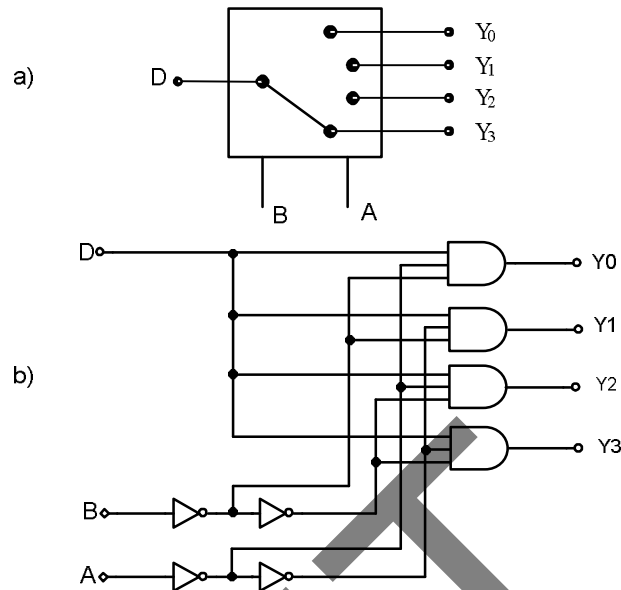


Hình 3. 26. Sơ đồ logic của bộ ghép kênh 8 đường 74151

3.5.2. Bộ phân kênh (Demultiplexer: DMUX)

Bộ phân kênh là một mạch logic tổ hợp có một đường vào và nhiều đường ra dữ liệu. Khác với bộ ghép kênh, bộ phân kênh làm chức năng chọn, truyền dữ liệu từ một đường vào

dữ liệu đến các đường ra riêng biệt. Cũng giống bộ ghép kênh, bộ phân kênh cũng có các đường vào điều khiển (n đường) chọn đầu ra (2^n đầu ra).



Hình 3. 27. a) Sơ đồ khối và b) Sơ đồ logic của bộ phân kênh 1 đầu vào và 4 đầu ra dữ liệu

Đầu vào được nối với đầu ra nào là tùy theo tổ hợp giá trị của các đầu vào điều khiển.

Sơ đồ khối mô tả chức năng của bộ phân kênh một đầu vào, 4 đầu ra được trình bày trên hình 3.27a.

Từ sơ đồ khối hình 3.27a, bộ phân kênh này có hai đầu vào điều khiển là A, B một đầu vào dữ liệu D và 4 đầu ra dữ liệu được ký hiệu là Y_0, Y_1, Y_2, Y_3 .

B	A	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

Bảng 3-13. Bảng trạng thái của bộ phân kênh 1 vào 4 ra.

Có thể lập bảng trạng thái biểu diễn sự hoạt động của bộ phân kênh trên như bảng 3-13.

Từ bảng trạng thái 3-13, viết được biểu thức logic của bộ phân kênh như sau:

$$Y_0 = \bar{A} \cdot \bar{B} \cdot D$$

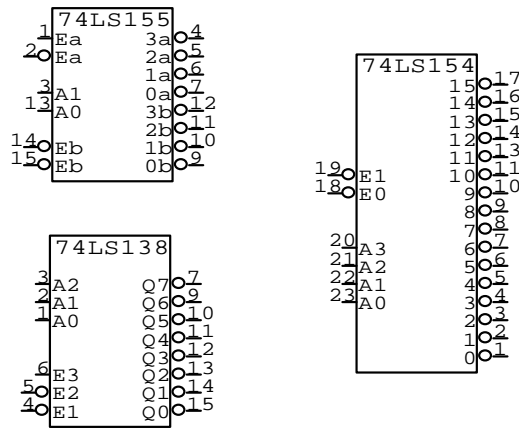
$$Y_1 = A \cdot \bar{B} \cdot D$$

$$Y_2 = \bar{A} \cdot B \cdot D$$

$$Y_3 = A \cdot B \cdot D$$

(3.7)

Từ biểu thức logic, có thể xây dựng sơ đồ logic của bộ phân kênh như hình 3.27b. Hình 3.28 là các ký hiệu của các vi mạch phân kênh thường gặp trong thực tế.



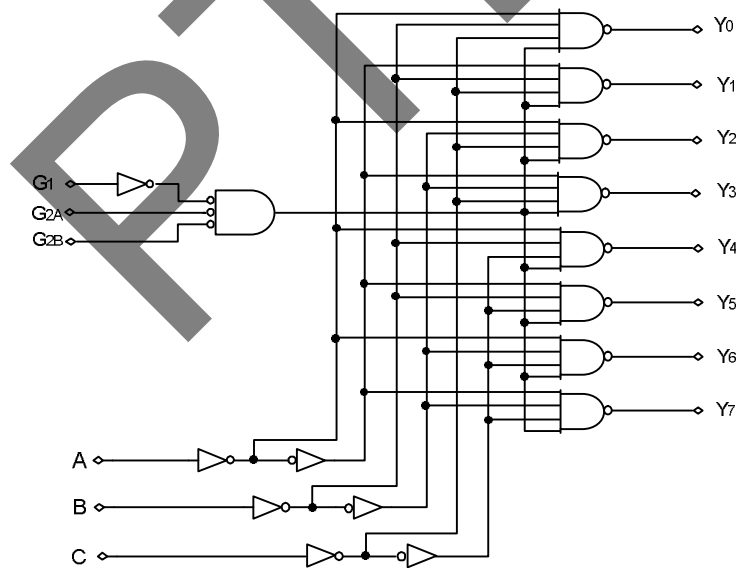
Hình 3. 28. Ký hiệu logic của các bộ phân kênh

IC 74LS155 gồm 2 bộ phân kênh một đầu vào, 4 đầu ra dữ liệu. Hai đầu vào điều khiển chọn kênh A, B được dùng chung cho cả hai bộ phân kênh.

IC 74LS154 là bộ giải mã địa chỉ 4 bit nhị phân.

IC 74LS138 là bộ giải mã địa chỉ, đồng thời có thể dùng làm bộ phân kênh một đường vào, 8 đường ra dữ liệu. Hình 3.29 trình bày sơ đồ logic của bộ phân kênh 74LS138.

Mạch có thể dùng như bộ phân kênh một đường vào, 8 đường ra dữ liệu Y_0, Y_1, \dots, Y_7 . Khi dùng IC 74LS138 làm bộ phân kênh thì các đầu vào cho phép G_1, G_{2A}, G_{2B} dùng làm đầu vào dữ liệu.



Hình 3. 29. Sơ đồ logic của mạch giải mã, phân kênh 74LS138

3.5.3. Một số ứng dụng của bộ ghép kênh và phân kênh

3.5.3.1. Định tuyến dữ liệu

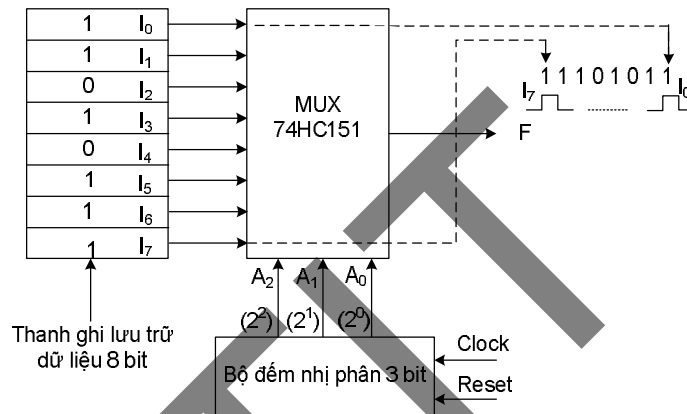
Có nhiều dữ liệu được định hướng tới một đích duy nhất, khi đó sử dụng bộ ghép kênh sẽ cho phép chọn dữ liệu nào (định tuyến đầu vào) hướng tới đích, các dữ liệu không được chọn sẽ bị cấm không tới được đích.

3.5.3.2. Chuyển đổi luồng dữ liệu từ song song nối tiếp và ngược lại

Một luồng dữ liệu số song song có tính chất mọi bit của nó xuất hiện đồng thời, luôn có ưu thế về tốc độ xử lý nhanh nhưng khi truyền trên khoảng cách xa sẽ tốn nhiều đường truyền nên thông thường nó được chuyển đổi thành dữ liệu kiểu nối tiếp (với tính chất các bit xuất hiện tuần tự) trước khi đi đến đường truyền nhờ bộ ghép kênh-MUX.

Ví dụ:

Hình 3.30 cho phép thực hiện biến đổi 8 bit dữ liệu 8 bit song song thành một dãy nối tiếp theo trật tự xác định nhờ các tín hiệu chọn $A_2A_1A_0$ luân chuyển tuần tự từ trạng thái 000 đến trạng thái theo một chu kỳ xác định nhờ khối tạo xung nhịp (clock) tạo ra xung có chu kỳ là T_B .



Hình 3. 30. Chuyển dữ liệu song song 8 bit sang dãy nối tiếp 8 bit sau 8 xung nhịp

3.5.3.3. Tạo hàm logic

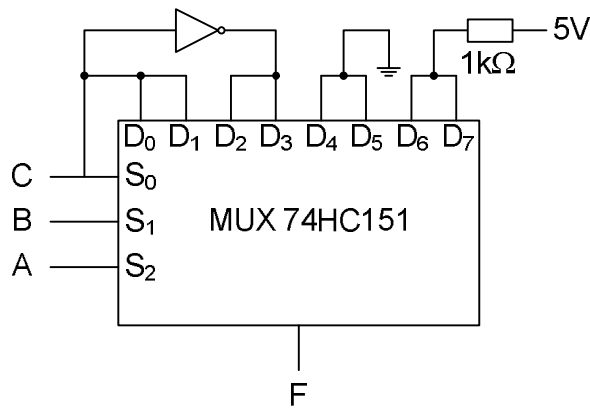
Sử dụng MUX tạo hàm logic trực tiếp từ bảng trạng thái không cần rút gọn, với mục đích này các đầu vào chọn (đầu vào địa chỉ) là các biến logic, mỗi đầu vào dữ liệu được nối thường xuyên với mức cao (logic 1) hay mức thấp (logic 0) tùy theo bảng trạng thái.

Ví dụ: $F(A, B, C) = \sum(1, 2, 6, 7)$.

Từ biểu thức trên, lập được bảng trạng thái 3-15.

Đầu vào chọn			Đầu ra
$A_{(A2)}$	$B_{(A1)}$	$C_{(A0)}$	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Bảng 3-15. Bảng trạng thái



Hình 3. 31. Sơ đồ logic thực hiện hàm F

Có nhiều cách chọn đầu vào địa chỉ và đầu vào dữ liệu: Nếu chọn A, B là đầu vào địa chỉ thì C sẽ là đầu vào dữ liệu, hoặc ta chọn B, C là đầu vào địa chỉ thì A sẽ là đầu vào dữ liệu, hoặc ta chọn A, C là đầu vào địa chỉ thì B sẽ là đầu vào dữ liệu (tùy theo người sử dụng). Ví dụ ở đây chọn A, B là đầu vào địa chỉ và C là đầu vào dữ liệu thì được:

+ $AB = 00$ thì $F = C$;

+ $AB = 01$ thì $F = \overline{C}$;

+ $AB = 10$ thì $F = 0$;

+ $AB = 11$ thì $F = 1$;

Từ các biểu thức trên, vẽ được sơ đồ mạch thực hiện như hình 3.31.

3.6. MẠCH SỐ HỌC.

3.6.1. Mạch cộng.

Giả sử có hai số nhị phân n bit A và B, trong đó:

$$A = A_{n-1} A_{n-2} \dots A_0 \quad \text{và} \quad B = B_{n-1} B_{n-2} \dots B_0$$

Để xây dựng mạch thực hiện phép cộng hai số cần nắm vững nguyên tắc bộ cộng đầy đủ 1 bit, trên cơ sở đó xây dựng bộ cộng song song n bit. Cụ thể trong phần này sẽ giới thiệu phương pháp xây dựng bộ cộng song song 4 bit từ bộ cộng toàn phần (đầy đủ) 1 bit.

3.6.1.1. Mạch cộng 1 bit.

a. Mạch cộng bán phần (Half Adder – HA)

Mạch bán tổng có 2 đầu vào:

a_0 là bit có trọng số nhỏ nhất của số A.

b_0 là bit có trọng số nhỏ nhất của số B.

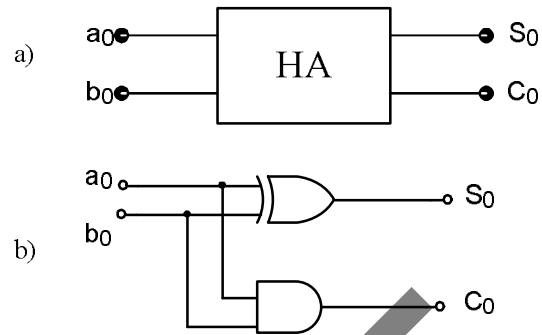
Mạch có 2 đầu ra: S_0 là kết quả phép cộng ở cột có trọng số nhỏ nhất.

C_0 là giá trị nhớ sang cột có trọng số cao hơn kế tiếp.

Nguyên tắc hoạt động của mạch bán tổng được diễn tả bởi bảng trạng thái 3-16.

a_0	b_0	S_0	C_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Bảng 3-16. Bảng trạng thái của mạch HA



Hình 3.32. a) Sơ đồ khối của HA, b) Sơ đồ logic

Từ bảng trạng thái, suy ra biểu thức logic của HA:

$$\begin{aligned} S_0 &= a_0 \oplus b_0 \\ C_0 &= a_0 \cdot b_0 \end{aligned} \quad (3.8)$$

Sơ đồ khối và sơ đồ logic của HA được trình bày trên hình 3.32.

b. Mạch cộng toàn phần (Full Adder - FA)

Mạch có 3 đầu vào: a_i là chữ số cột thứ i của số A.

b_i là chữ số cột thứ i của số B.

C_i là bit nhớ của trọng số nhỏ hơn liền kề chuyển đến

Mạch có 2 đầu ra: S_0 là kết quả phép cộng ở cột thứ i .

C_0 là bit nhớ sang trọng số lớn hơn kế tiếp.

Bảng trạng thái của FA được trình bày ở bảng 3-17.

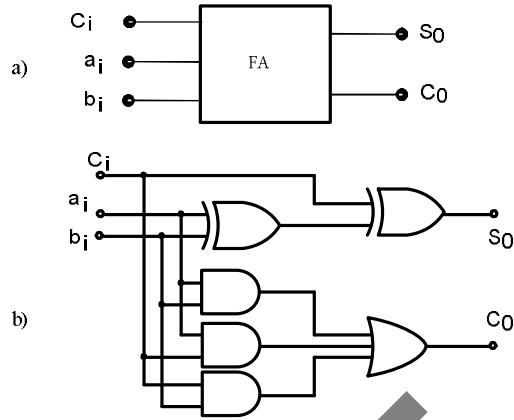
a_i	b_i	C_i	S_0	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Bảng 3-17. Bảng trạng thái của bộ FA

Từ bảng trạng thái suy ra hàm logic của FA:

$$\begin{aligned} S_0 &= a_i \oplus b_i \oplus C_i \\ C_0 &= a_i \cdot b_i + a_i \cdot C_i + b_i \cdot C_i \end{aligned} \quad (3.9)$$

Sơ đồ khối và mạch logic trình bày trên hình 3.33.



Hình 3. 33. a) Sơ đồ khối của FA, b) Mạch logic.

Từ bảng trạng thái của FA dễ dàng suy ra biểu thức logic bằng cách lấy tổng chuẩn của các tích, tức là lấy tổng các Mintex mà hàm bằng 1.

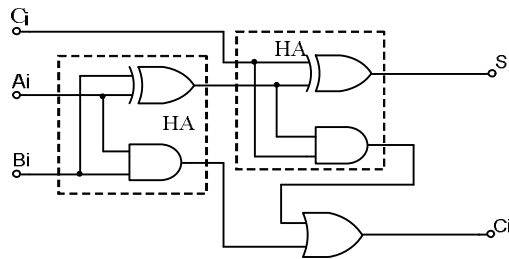
$$\begin{aligned} S_0 &= \overline{a_i} \cdot \overline{b_i} \cdot C_i + \overline{a_i} \cdot b_i \cdot \overline{C_i} + a_i \cdot \overline{b_i} \cdot \overline{C_i} + a_i \cdot b_i \cdot C_i \\ &= \overline{a_i} (\overline{b_i} C_i + b_i \cdot \overline{C_i}) + a_i (\overline{b_i} \cdot \overline{C_i} + b_i \cdot C_i) \\ &= a_i \oplus b_i \oplus C_i \end{aligned} \quad (3.10)$$

$$C_0 = \overline{a_i} \cdot b_i \cdot C_i + a_i \cdot \overline{b_i} \cdot C_i + a_i \cdot b_i \cdot \overline{C_i} + a_i \cdot b_i \cdot C_i$$

Có thể tính \$C_0\$ theo cách khác:

$$\begin{aligned} C_0 &= (\overline{a_i} \cdot b_i \cdot C_i + a_i \cdot \overline{b_i} \cdot C_i) + (a_i \cdot b_i \cdot \overline{C_i} + a_i \cdot b_i \cdot C_i) \\ &= C_i (\overline{a_i} \cdot b_i + a_i \cdot \overline{b_i}) + a_i \cdot b_i = C_i (a_i \oplus b_i) + a_i \cdot b_i \\ C_0 &= C_i (a_i \oplus b_i) + a_i \cdot b_i \end{aligned} \quad (3.11)$$

Từ công thức (3.11) và (3.12) có thể xây dựng mạch FA từ hai mạch HA. Sơ đồ logic của FA từ hai HA được trình bày trên hình 3.34.



Hình 3. 34. Sơ đồ mạch FA.

Nếu \$C_i = 0\$ thì FA trở thành HA.

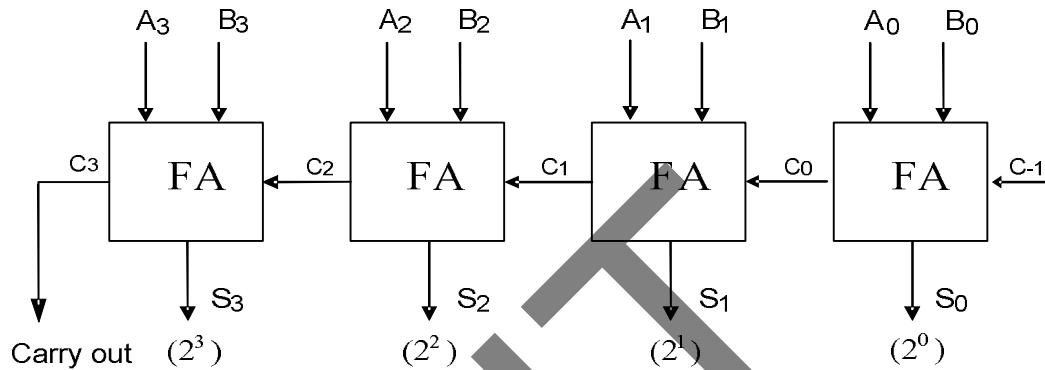
3.6.1.2. Mạch cộng song song 4 bit.

Giả sử có hai số nhị phân 4 bit A và B. Trong đó:

$$A = A_3 \cdot A_2 \cdot A_1 \cdot A_0$$

$$B = B_3 \cdot B_2 \cdot B_1 \cdot B_0$$

Với A_0, B_0 là cột có trọng số bé nhất của 2 số A và B (2^0). A_3, B_3 là cột có trọng số lớn nhất của 2 số A và B (2^3). Từ bộ cộng nhị phân 1 bit có thể đưa ra mạch của bộ cộng song song 4 bit thực hiện phép cộng hai số A và B như hình 3.35.



Hình 3. 35. Bộ cộng song song 4 bit.

Ví dụ: Có hai số A và B với $A = 1111$, $B = 1001$. Phép cộng hai số được thực hiện như sau:

A =	1	1	1	1	
B =	1	0	0	1	
C =	1	1	1	1	
	= C_3	C_2	C_1	C_0	C_{-1}
Σ =	1	1	0	0	0

Nhược điểm:

Tín hiệu nhớ ở đầu ra (Carry Out) được tạo thành sau khi đi qua bốn bộ cộng FA_0, \dots, FA_3 . Nếu mỗi bộ cộng FA là hai mạch cộng bán phần thì Carry Out sẽ xuất hiện ở đầu ra sau một thời gian $T = 8$ lần thời gian quá độ với thời gian quá độ là thời gian trễ chuyển mạch của một tầng logic. Do vậy thời gian trễ rất lớn, đặc biệt khi số bit của mỗi số hạng tăng lên đủ lớn. Để khắc phục nhược điểm đó người ta dùng bộ cộng nhớ nhanh hay bộ cộng nhớ nhìn trước (Fast carry hay Carry look Ahead).

3.6.2. Mạch trừ.

2.6.2.1. Mạch bán hiệu (HS-Half Subtractor)

Mạch bán hiệu có 2 đầu vào:

a_0 là bit có trọng số nhỏ nhất của số A.

b_0 là bit có trọng số nhỏ nhất của số B.

Mạch có 2 đầu ra:

S_0 (*Subtract Output*) là kết quả phép trừ ở cột có trọng số nhỏ nhất.

B_0 (*Borrow Out*) là giá trị “mượn” của cột có trọng số cao hơn kế tiếp.

Nguyên tắc hoạt động của mạch bán hiệu được diễn tả bởi bảng trạng thái 3-18.

a_0	b_0	S_0	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Bảng 3-18. Bảng trạng thái của mạch HS

Từ bảng trạng thái, suy ra biểu thức logic của HS:

$$\begin{aligned} S_0 &= a_0 \oplus b_0 \\ B_0 &= \overline{a_0} \cdot b_0 \end{aligned} \quad (3.12)$$

Sơ đồ khối và sơ đồ logic của HS được trình bày trên hình 3.36.

Error! Objects cannot be created from editing field codes.

Hình 3. 36. Sơ đồ mạch HS.

3.6.2.2. Mạch trừ toàn phần (Full Subtractor-FS)

a_i	b_i	B_i	S_0	B_0
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

Bảng 3-19. Bảng trạng thái của mạch trừ.

Mạch trừ toàn phần có 3 đầu vào:

a_i là chữ số cột thứ i của số A .

b_i là chữ số cột thứ i của số B .

B_i là bit “mượn” nhớ từ cột trước đưa tới

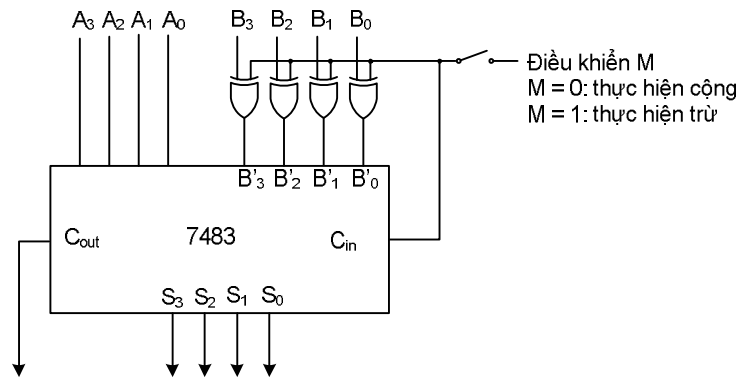
Mạch có 2 đầu ra:

S_0 (*Subtract Output*) kết quả phép trừ ở cột thứ i .

B_0 (*Borrow Out*) là đầu ra “mượn”.

Bảng 3-19 là bảng trạng thái của bộ trừ đầy đủ.

Hàm logic:



Hình 3. 39. Mạch cộng/trừ theo bù 2

3.6.4. Mạch so sánh.

Trong các hệ thống số, đặc biệt là trong máy tính, thường thực hiện việc so sánh hai số. Hai số cần so sánh có thể là các số nhị phân, có thể là các ký tự đã mã hoá nhị phân. Mạch so sánh có thể hoạt động theo kiểu nối tiếp hoặc theo kiểu song song. Trong phần này, sẽ nghiên cứu bộ so sánh theo kiểu song song.

3.6.4.1. Mạch so sánh bằng nhau.

a. Mạch so sánh bằng nhau 1 bit.

a_i	b_i	g_i
0	0	1
0	1	0
1	0	0
1	1	1

Bảng 3-20

Xét 2 bit a_i và b_i , gọi g_i là kết quả so sánh. Từ đó là có bảng trạng thái 3-20.

Từ bảng trạng thái 3-20, viết được biểu thức hàm ra:

$$g_i = \overline{a_i} \cdot \overline{b_i} + a_i \cdot b_i = a_i \oplus b_i \quad (3.14)$$

Error! Objects cannot be created from editing field codes.

Hình 3. 40. Sơ đồ logic hàm ra của bộ so sánh bằng 1 bit

b. Mạch so sánh bằng nhau 4 bit.

So sánh hai số nhị phân 4 bit $A = a_3a_2a_1a_0$ với $B = b_3b_2b_1b_0$. Vậy hai số A và B bằng nhau khi $a_3 = b_3, a_2 = b_2, a_1 = b_1, a_0 = b_0$.

Biểu thức đầu ra tương ứng là:

$$G = g_3 \cdot g_2 \cdot g_1 \cdot g_0 \quad (3.15)$$

$$\begin{aligned} g_3 &= \overline{a_3 \oplus b_3} \\ g_2 &= \overline{a_2 \oplus b_2} \\ g_1 &= \overline{a_1 \oplus b_1} \\ g_0 &= \overline{a_0 \oplus b_0} \end{aligned}$$

với

3.6.4.2. Mạch so sánh lớn hơn

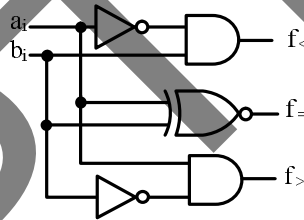
a. Mạch so sánh lớn hơn 1 bit.

a_i	b_i	$f_{<}$	$f_{=}$	$f_{>}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Bảng 3-21. Bảng trạng thái của mạch so sánh

Từ bảng trạng thái 3-21, có biểu thức ra:

$$\begin{aligned} f_{<} &= \overline{a_i} \cdot b_i \\ f_{=} &= \overline{a_i \oplus b_i} \\ f_{>} &= a_i \cdot \overline{b_i} \end{aligned} \quad (3.16)$$



Hình 3. 41. Mạch điện bộ so sánh 1 bit

b. Mạch so sánh 4 bit (So sánh lớn hơn).

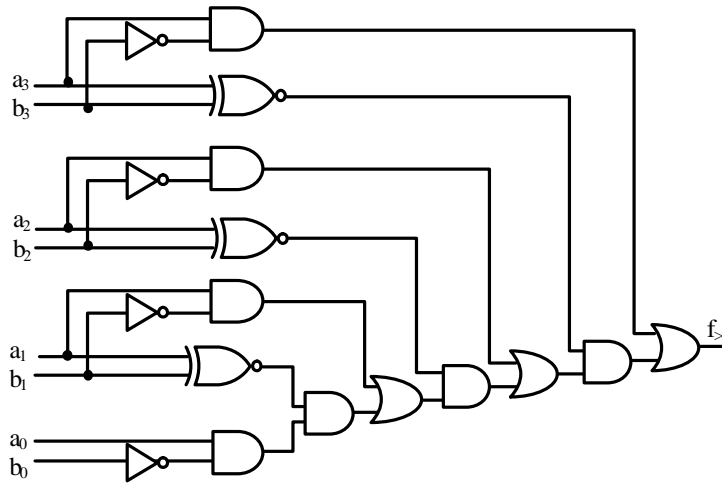
So sánh hai số nhị phân 4 bit $A = a_3a_2a_1a_0$ với $B = b_3b_2b_1b_0$. Số A lớn hơn số B khi:

$a_3 > b_3$ hoặc $a_3 = b_3$ và $a_2 > b_2$ hoặc $a_3 = b_3$ và $a_2 = b_2$ và $a_1 > b_1$ hoặc $a_3 = b_3$ và $a_2 = b_2$ và $a_1 = b_1$ và $a_0 > b_0$.

Từ đó, có biểu thức hàm ra là:

$$f_{>} = a_3 \cdot \overline{b_3} + \overline{a_3 \oplus b_3} \cdot a_2 \cdot \overline{b_2} + \overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot a_1 \cdot \overline{b_1} + \overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot \overline{a_1 \oplus b_1} \cdot a_0 \cdot \overline{b_0}$$

Từ biểu thức trên, vẽ được sơ đồ logic 3.42.



Hình 3. 42. Mạch điện bộ so sánh lớn hơn 4 bit

Một trong những bộ so sánh thông dụng hiện nay là 7485, 74LS85, 74HC85 - IC này so sánh 2 số nhị phân 4 bit.

3.7. MẠCH PHÁT HIỆN SAI

3.7.1. Mạch tạo và kiểm tra chẵn lẻ.

Có nhiều phương pháp mã hoá dữ liệu để phát hiện lỗi và sửa lỗi khi truyền dữ liệu. Phương pháp đơn giản nhất là thêm một bit vào dữ liệu được truyền đi sao cho số chữ số 1 trong dữ liệu luôn là chẵn hoặc lẻ. Bit thêm vào đó được gọi là bit chẵn/lẻ.

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

Xây dựng sơ đồ tạo được bit chẵn, lẻ để thêm vào n bit dữ liệu.

Xây dựng sơ đồ kiểm tra hệ xem đó là hệ chẵn hay lẻ với $(n + 1)$ bit ở đầu vào (n bit dữ liệu, 1 bit chẵn/lẻ).

3.7.1.1 Mã chẵn, lẻ.

Mã chẵn và mã lẻ là hai loại mã có khả năng phát hiện lỗi hay dùng nhất. Để thiết lập loại mã này, chỉ cần thêm một bit chẵn/ lẻ (bit parity) vào tổ hợp mã đã cho, nếu tổng số bit 1 trong từ mã (bit tin tức + bit chẵn/lẻ) là chẵn thì được mã chẵn và ngược lại được mã lẻ.

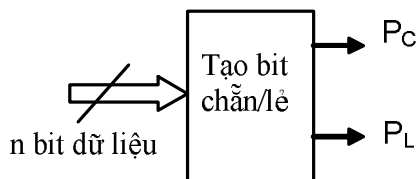
Bảng 3-23 là một ví dụ về mã chẵn/lẻ.

BCD 8421	BCD 8421 chẵn		BCD 8421 lẻ	
		P_C		P_L
0000	0000	0	0000	1
0001	0001	1	0001	0
0010	0010	1	0010	0
0011	0011	0	0011	1
0100	0100	1	0100	0
0101	0101	0	0101	1
0110	0110	0	0110	1
0111	0111	1	0111	0

BCD 8421	BCD 8421 chẵn		BCD 8421 lẻ	
	P_C		P_L	
1000	1000	1	1000	0
1001	1001	0	1001	1

Bảng 3-23. Mã BCD 8421 chẵn / lẻ

3.7.1.2. Mạch tạo và kiểm tra chẵn/lẻ.



Hình 3. 43. Sơ đồ khối của mạch tạo bit chẵn/lẻ

Hình 3.43 mô tả sơ đồ khối của mạch tạo bit chẵn/lẻ.

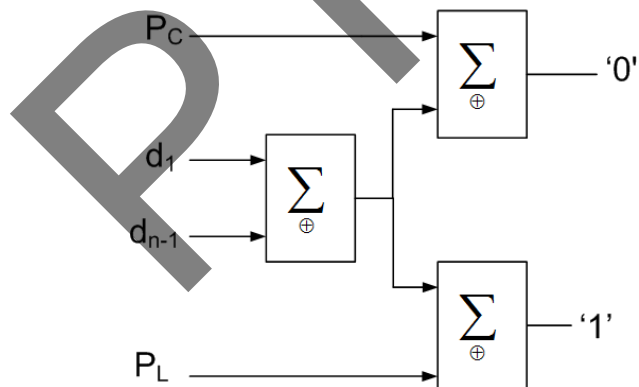
Biểu thức kiểm tra tính chẵn:

$$P_C \oplus d_1 \oplus d_2 \oplus \dots \oplus d_n = 0 \quad (3.17)$$

Biểu thức kiểm tra tính lẻ:

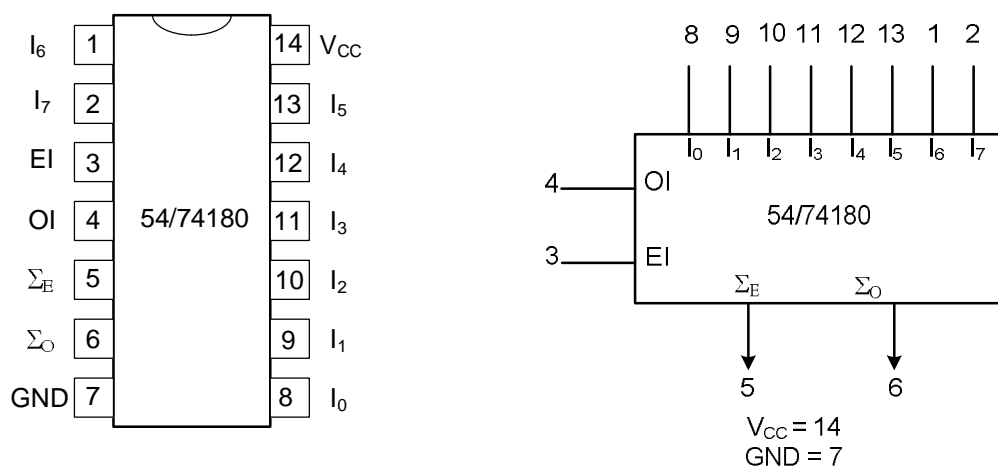
$$P_L \oplus d_1 \oplus d_2 \oplus \dots \oplus d_n = 1 \quad (3.18)$$

Từ đó, xây dựng được mạch kiểm tra tính chẵn/lẻ:



Hình 3. 44. Sơ đồ khối của mạch kiểm tra chẵn/lẻ

Do công dụng phổ biến của mạch tạo và kiểm tra chẵn/lẻ nên người ta đã thiết kế sẵn trên một chip MSI (74180) (hình 3.45).



Hình 3. 45. Sơ đồ chân và ký hiệu của IC 54/74180

3.7.2 Mạch tạo mã và giải mã Hamming

Mã Hamming là loại mã tuyến tính, mã này được R.W.Hamming đưa ra và sử dụng trong một số hệ thống thông tin. Mã này có khả năng sửa sai một lỗi. Mã Hamming có sơ đồ tạo mã và giải mã đơn giản.

Số bit kiểm tra P và số bit tin tức D phải thỏa mãn biểu thức

$$D + P + 1 \leq 2^P \quad (3.19)$$

Khi số bit tin tức tăng thì số bit kiểm tra cũng tăng, tuy nhiên tốc độ của số bit tin tức nhanh hơn nhiều so với sự tăng của số bit kiểm tra. Do vậy, khi số bit tin tức càng lớn thì hiệu quả sử dụng bit chẵn/lẻ càng cao, điều này dẫn đến việc sử dụng mã Hamming trở nên phổ biến.

3.7.2.1. Tạo mã

Để tạo mã Hamming, cần xác định giá trị các bit kiểm tra tính chẵn hoặc lẻ của từ mã. Vị trí các bit kiểm tra chẵn/lẻ là các bit có thứ tự 2^i tính từ bit có trọng số nhỏ nhất.

Ví dụ: với số bit tin tức là 5, áp dụng công thức (3.19) thì số bit kiểm tra chẵn/lẻ là 4. Cách sắp xếp vị trí các bit như sau:

9	8	7	6	5	4	3	2	1
D ₅	P ₈	D ₄	D ₃	D ₂	P ₄	D ₁	P ₂	P ₁

(*)

Các bit chẵn/lẻ có nhiệm vụ kiểm tra tính chẵn hoặc lẻ (tùy theo yêu cầu) của một số bit kể cả bản thân chúng trong tổ hợp mã.

Bit chẵn/lẻ P có nhiệm vụ kiểm tra tính chẵn lẻ tại các vị trí nó chiếm giữ có giá trị 1 (theo bảng nhị phân).

Hệ thập phân	Hệ nhị phân P ₈ P ₄ P ₂ P ₁
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0

Hệ thập phân	Hệ nhị phân $P_8P_4P_2P_1$
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
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

Bảng 3-24. Bảng trạng thái

Bit P_1 kiểm tra tính chẵn/lẻ tại các vị trí : 1, 3, 5, 7, 9, 11...

Bit P_2 kiểm tra tính chẵn/lẻ tại các vị trí : 2, 3, 6, 7, 10, 11...

Bit P_4 kiểm tra tính chẵn/lẻ tại các vị trí : 4, 5, 6, 7, 12, 13...

Bit P_8 kiểm tra tính chẵn/lẻ tại các vị trí : 8, 9, 10, 11...

Nếu xây dựng mã Hamming chẵn thì phải thỏa mãn biểu thức kiểm tra tính chẵn như sau:

$$d_1 \oplus d_2 \oplus d_3 \oplus \dots \oplus d_n = 0$$

Nếu xây dựng mã Hamming lẻ thì phải thỏa mãn biểu thức kiểm tra tính lẻ như sau:

$$d_1 \oplus d_2 \oplus d_3 \oplus \dots \oplus d_n = 1.$$

Tìm P_0 :

Do bit P_0 kiểm tra tính chẵn tại các vị trí : 1, 3, 5, 7, 9, 11... nên viết được biểu thức:

$$P_0 \oplus \text{vị trí 3} \oplus \text{vị trí 5} \oplus \text{vị trí 7} \oplus \text{vị trí 9} \oplus \dots = 0.$$

$$P_0 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus \dots = 0.$$

Thay giá trị các vị trí trong công thức (*) tìm được P_0 .

Tương tự đối với các P khác. Cuối cùng thay vào biểu thức (*) tìm được từ mã Hamming tương ứng.

Ví dụ: Cho từ mã thông tin là 101101, hãy xây dựng từ mã Hamming chẵn.

Lời giải:

$$101101 \Rightarrow D = 6 \Rightarrow P = 4 \text{ thỏa mãn công thức (3.19).}$$

$$\begin{array}{cccccccc} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ \hline \end{array}$$

1	0	P ₈	1	1	0	P ₄	1	P ₂	P ₁
---	---	----------------	---	---	---	----------------	---	----------------	----------------

Tìm P₁:

$$P_1 \oplus \text{vị trí 3} \oplus \text{vị trí 5} \oplus \text{vị trí 7} \oplus \text{vị trí 9} = 0.$$

$$P_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5 = 0.$$

$$P_1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0 \rightarrow P_1 = 0$$

Tìm P₂:

$$P_2 \oplus \text{vị trí 3} \oplus \text{vị trí 6} \oplus \text{vị trí 7} \oplus \text{vị trí 10} = 0.$$

$$P_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 = 0.$$

$$P_2 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0 \rightarrow P_2 = 1$$

Tìm P₄:

$$P_4 \oplus \text{vị trí 5} \oplus \text{vị trí 6} \oplus \text{vị trí 7} = 0.$$

$$P_4 \oplus D_2 \oplus D_3 \oplus D_4 = 0.$$

$$P_4 \oplus 0 \oplus 1 \oplus 1 = 0 \rightarrow P_4 = 0$$

Tìm P₈:

$$P_8 \oplus \text{vị trí 9} \oplus \text{vị trí 10} = 0.$$

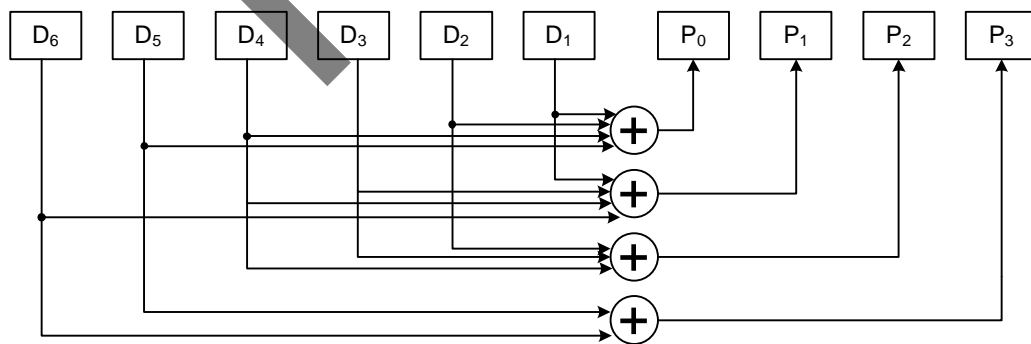
$$P_8 \oplus D_5 \oplus D_6 = 0.$$

$$P_8 \oplus 0 \oplus 1 = 0 \rightarrow P_8 = 1$$

Từ mã Hamming chẵn phát đi là:

1	0	1	1	1	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---

Từ đó, vẽ được mạch tạo mã như hình 3.46.



Hình 3. 46. Mạch tạo mã Hamming

3.7.2.2. Giải mã

Khi nhận được từ mã, tiến hành tính S_i . Nếu mã Hamming chẵn thì $S_i = 0$, nếu là Mã Hamming lẻ thì $S_i = 1$. Nếu kết quả nhận được không đúng thì tổ hợp giá trị của các S_i chính là vị trí lỗi (đối với mã chẵn) hoặc tổ hợp các giá trị đảo của S_i chính là vị trí lỗi (đối với mã lẻ).

Biểu thức của các S_i (đối với ví dụ trên) là:

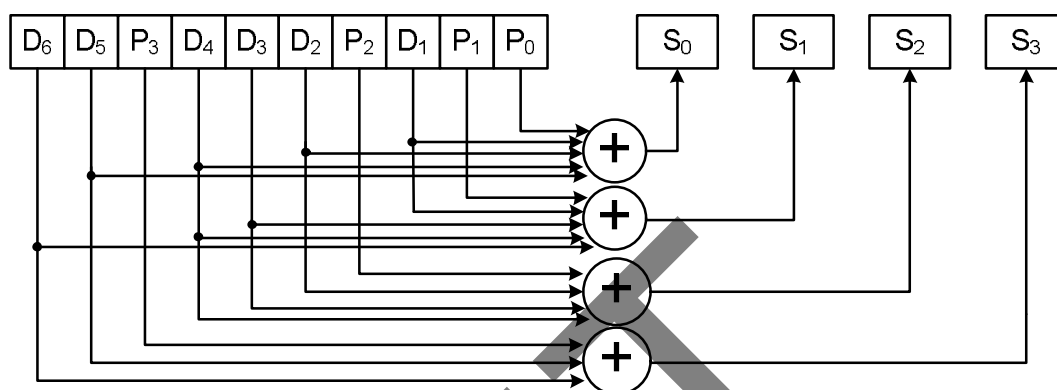
$$S_3 = P_3 \oplus D_5 \oplus D_6$$

$$S_2 = P_2 \oplus D_2 \oplus D_3 \oplus D_4$$

$$S_1 = P_1 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6$$

$$S_0 = P_0 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5$$

Từ đó, vẽ được mạch giải mã như hình 3.47.



Hình 3. 47. Mạch giải mã Hamming

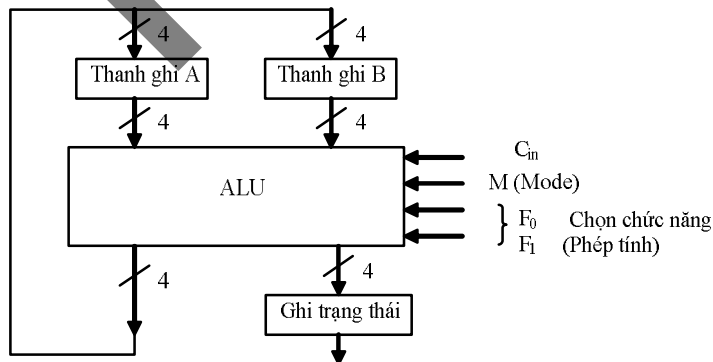
3.8. ĐƠN VỊ SỐ HỌC VÀ LOGIC (ALU).

Đơn vị số học và logic (Arithmetic – Logic Unit) là một thành phần cơ bản không thể thiếu được trong các máy tính. Nó bao gồm 2 khối chính là khối logic và khối số học và một khối ghép kênh.

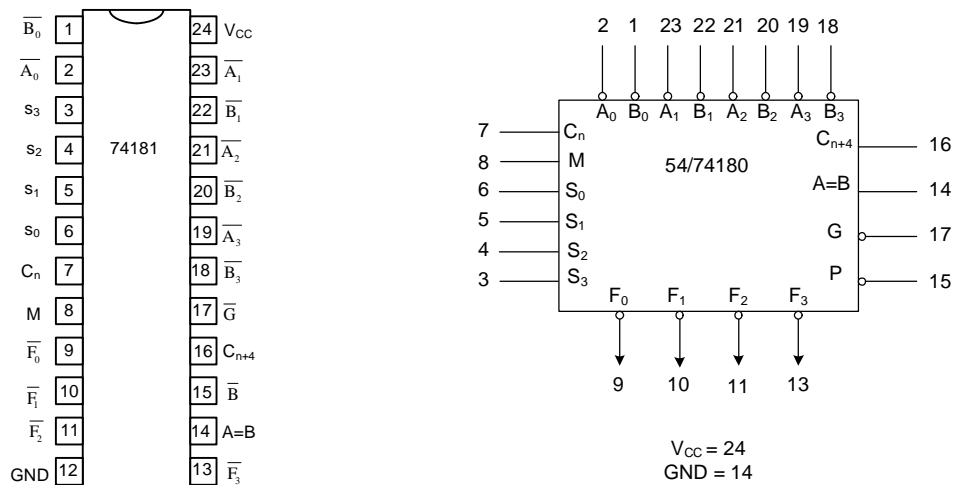
Khối logic: Thực hiện các phép tính logic như là AND, OR, NOT, XOR.

Khối số học: Thực hiện các phép tính số học như là: cộng, trừ, tăng 1, giảm 1.

Sơ đồ khối của 1 đơn vị số học – logic ALU 4 bit được mô tả ở hình 3.48.



Hình 3. 48. Sơ đồ khối của bộ ALU 4 bit



Hình 3. 49. Sơ đồ chân và ký hiệu của IC 74181

M là đầu vào chọn phép tính số học hay logic.

F₀, F₁ là hai đầu vào chọn chức năng.

Sau khi một phép tính số học hay logic được thực hiện thì kết quả sẽ được ghi lên 1 thanh ghi, ví dụ thanh ghi A. Kết quả này có thể được sử dụng để thực hiện phép tính sau.

Bộ ALU còn tạo ra các bit trạng thái chuyển đổi thành ghi.

Hình 3.49 giới thiệu sơ đồ chân và sơ đồ ký hiệu của ALU 74181

TÓM TẮT

Trong chương này, chúng ta đã giới thiệu mạch logic tổ hợp. Mạch tổ hợp do các phần tử logic cơ bản cấu trúc nên. Đặc điểm của mạch tổ hợp là tín hiệu đầu ra ở thời điểm bất kỳ nào cũng chỉ phụ thuộc vào tín hiệu ở đầu vào ở thời điểm đó mà không liên quan đến trạng thái vốn có của mạch.

Mạch tổ hợp rất phong phú, không thể xem xét hết trong chương 3. Trọng tâm của chúng ta là nắm vững đặc điểm mạch tổ hợp và phương pháp chung khi thiết kế, phân tích mạch tổ hợp. Vì vậy, chúng ta đã giới thiệu một cách chọn lọc bộ mã hoá, bộ giải mã, bộ ghép kênh, phân kênh, mạch cộng, trừ, mạch so sánh...trong quá trình đó, đã xem xét phương pháp phân tích và thiết kế mạch tổ hợp.

Việc tối thiểu hoá hàm logic rất quan trọng. Vì việc này làm cho mạch logic đơn giản, kinh tế. Chúng ta mong muốn mạch điện càng ít linh kiện càng tốt, số đầu vào của mạch cổng cũng không thể quá nhiều.

CÂU HỎI ÔN TẬP

1. Mạch logic tổ hợp là mạch thực hiện chức năng gì?
2. Bộ mã hoá ưu tiên khác với bộ mã hoá thông thường ở điểm nào?
3. Dụng cụ hiển thị 7-đoạn có thể hiển thị những ký tự gì?

4. Dụng cụ hiển thị 7-đoạn Anốt chung khác với loại Catốt chung ở điểm gì? Vẽ hình minh họa.
5. Thiết kế bộ ghép kênh 5 đường vào dữ liệu?
6. Dùng IC 74138 để thiết kế bộ phân kênh 1 vào 32 ra.
7. Thuật ngữ *parity* (tính chẵn lẻ) có nghĩa là gì? Nếu bộ tạo bit chẵn lẻ nhận một bit kiểm tra parity chẵn, nó yêu cầu nhận bit parity ở dạng gì?
8. Xây dựng mạch tạo và kiểm tra chẵn/ lẻ 7 bit?
9. Thiết kế mạch logic toàn NAND 2 lối vào để thực hiện việc lấy biểu quyết đa số: gồm 3 lối vào và 1 lối ra. Trạng thái lối ra luôn có mức theo đa số mức của tín hiệu vào.
10. Xây dựng mạch biến mã BCD 8421 thành nhị phân áp dụng cho số $(1001\ 0110\ 0101)_2 = 965_{10}$.
11. Xây dựng mạch biến mã nhị phân 4 bit thành mã Gray.
12. Xây dựng mạch tổ hợp có 4 lối vào và 1 lối ra. Hàm ra có mức logic 1 khi tín hiệu lối vào chia hết cho 3.
13. Khi cần so sánh 2 số nhị phân 16 bit cần dùng mấy IC 7485? Hãy vẽ sơ đồ thực hiện việc ghép nối đó.
14. Khi cần cộng 2 số nhị phân 32 bit cần dùng mấy IC 74283? Hãy vẽ sơ đồ thực hiện việc ghép nối đó.
15. Dùng bộ ghép kênh để thực hiện hàm logic $f = A.B.C + \overline{A.B}$
16. Xây dựng mạch biến mã BCD 8421 thành mã dư 3.
17. Xây dựng mạch biến mã mã dư 3 thành BCD 8421.
18. Xây dựng mạch logic kiểm tra tính lẻ có 3 đầu vào và 1 đầu ra. Chức năng của mạch là khi có lẻ số bit 1 ở đầu vào thì đầu ra có trạng thái 1.

CHƯƠNG 4. MẠCH LOGIC TUẦN TỰ

GIỚI THIỆU.

Chúng ta đã nghiên cứu về phép phân tích và thiết kế các mạch logic tổ hợp. Mặc dù rất quan trọng nhưng nó chỉ là một phần của các hệ thống kỹ thuật số. Một phần quan trọng của các hệ thống kỹ thuật số khác là phân tích và thiết kế mạch tuần tự. Tuy nhiên việc thiết kế các mạch tuần tự lại phụ thuộc vào việc thiết kế mạch tổ hợp đã được đề cập ở chương 3.

Có nhiều ứng dụng mà đầu ra số phải được tạo để phù hợp với tuần tự nhận được các tín hiệu vào. Yêu cầu này không thể được thỏa mãn bằng việc sử dụng hệ thống logic tổ hợp.

Những ứng dụng này yêu cầu đầu ra không chỉ phụ thuộc vào các điều kiện đầu vào hiện có mà còn phụ thuộc vào lịch sử của các đầu vào. Lịch sử được cung cấp bằng cách phản hồi từ đầu ra về lại đầu vào.

Mạch logic tuần tự không những phụ thuộc vào trạng thái các đầu vào và còn phụ thuộc vào trạng thái trong của nó. Mạch tuần tự được chia làm hai loại chính là mạch tuần tự không đồng bộ và mạch tuần tự đồng bộ.

Trong phần này sẽ giới thiệu về các phần tử nhớ của mạch tuần tự. Cách phân tích và thiết kế mạch tuần tự đơn giản và phức tạp.

NỘI DUNG

4.1. KHÁI NIỆM CHUNG VÀ MÔ HÌNH TOÁN HỌC

4.1.1. Khái niệm chung

Trong chương này, chúng ta sẽ nói đến hệ thống số được gọi là mạch logic tuần tự (hay còn gọi là mạch dãy - Sequential Circuit). Hoạt động của hệ này có tính chất kế tiếp nhau, tức là trạng thái hoạt động của mạch điện không những phụ thuộc trực tiếp đầu vào mà còn phụ thuộc vào trạng thái bên trong trước đó của chính nó. Nói cách khác các hệ thống này làm việc theo nguyên tắc có nhớ.

4.1.2. Mô hình toán học

Mạch tuần tự là mạch bao gồm mạch logic tổ hợp và mạch nhớ. Mạch nhớ là các trigơ. Đối với mạch tuần tự, đáp ứng ra của hệ thống mạch điện không chỉ phụ thuộc trực tiếp vào tín hiệu vào (X) mà còn phụ thuộc vào trạng thái nội (Q) của nó. Có thể mô tả sơ đồ khối tổng quát của mạch tuần tự.

Ở đây: X - tập tín hiệu vào.

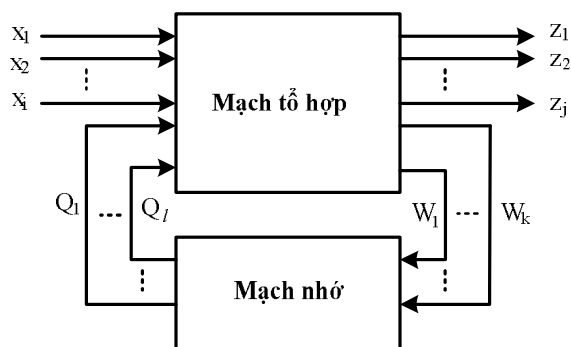
Q - tập trạng thái trong trước đó của mạch.

W - hàm kích.

Z - các hàm ra

Hoạt động của mạch tuần tự được mô tả bằng mối quan hệ toán học sau:

$$Z = f(Q, X)$$



Hình 4. 1. Sơ đồ khối mạch tuần tự

Trong phương trình toán học của mạch tuần tự thấy có hai thông tin. Đó là thông tin về trạng thái tiếp theo của mạch tuần tự và thông tin về tín hiệu ra của mạch. Hai thông tin này cùng phụ thuộc đồng thời vào trạng thái bên trong trước đó của mạch (Q) và tín hiệu tác động vào (X) của nó. Có thể viết lại biểu thức trên như sau:

$$Z = f [Q(n), X] ; \quad Q(n+1) = f [Q(n), X]$$

Trong đó: $Q(n+1)$: là trạng thái kế tiếp của mạch.

$Q(n)$: là trạng thái bên trong trước đó.

Để hiểu rõ hơn về mạch tuần tự, xét các phần tử có trong mạch. Mạch logic tổ hợp đã được xét ở chương 4 còn phần tử nhớ chính là các trigơ.

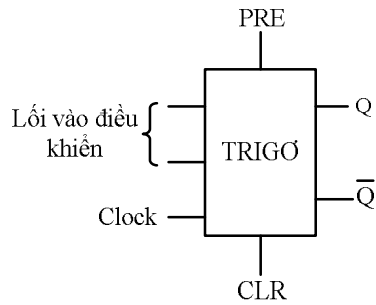
4.2. PHẦN TỬ NHỚ CỦA MẠCH TUẦN TỰ

4.2.1. Các loại Trigo

Định nghĩa: Trigo là phần tử có khả năng lưu trữ (nhớ) một trong hai trạng thái bằng ổn định tương ứng với hai mức logic 1 và 0. Trigo trong tiếng Anh còn gọi là Flip – Flop viết tắt là FF hay Latch.

Khi tác dụng một tín hiệu tới đầu vào có cực tính và biên độ thích hợp, trigơ có thể chuyển về một trong hai trạng thái cân bằng, và giữ nguyên trạng thái đó chừng nào chưa có tín hiệu tác dụng làm thay đổi trạng thái của nó. Trạng thái tiếp theo của trigơ không những phụ thuộc vào tín hiệu đầu vào mà còn phụ thuộc vào trạng thái hiện tại của nó. Như vậy nó có tính chất nhớ và nó được sử dụng làm các phần tử nhớ. Trigo được tạo thành từ các phần tử logic cơ bản.

Trigo có từ 1 đến một vài lối điều khiển, có hai đầu ra luôn luôn ngược nhau là Q và \bar{Q} . Tùy từng loại trigơ có thể có thêm các đầu vào lập (PRESET) và đầu vào xóa (CLEAR). Ngoài ra, trigơ còn có đầu vào đồng bộ (CLOCK). Hình 4.2 là sơ đồ khối tổng quát của trigơ.



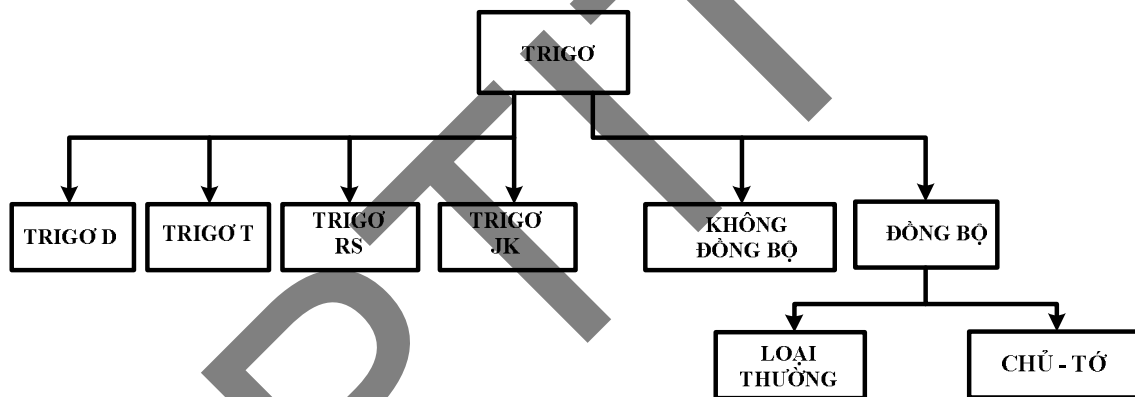
Hình 4. 2. Sơ đồ tổng quát của một trigơ

Phân loại:

Theo chức năng làm việc của của các đầu vào điều khiển: hiện nay thường sử dụng loại trigơ một đầu vào như trigơ D, T; loại hai đầu vào như trigơ RS, trigơ JK.

Theo phương thức hoạt động có hai loại: trigơ đồng bộ và trigơ không đồng bộ. Trong loại trigơ đồng bộ lại được chia làm hai loại: trigơ thường và trigơ chủ - tớ (Master- Slave).

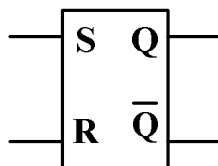
Sơ đồ khối của sự phân loại trigơ được cho ở hình 4.3.



Hình 4. 3. Sơ đồ phân loại trigơ

4.2.1.1. Trigơ RS

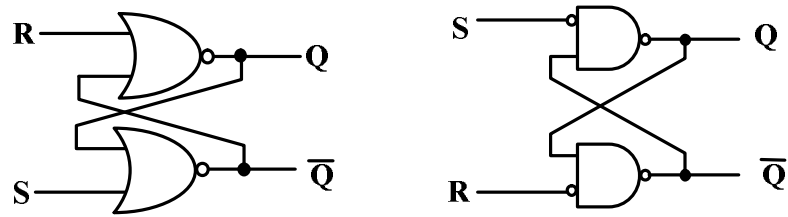
a) Trigơ RS không đồng bộ



Hình 4. 4. Sơ đồ ký hiệu trigơ RS không đồng bộ

Trigơ RS là loại có hai đầu vào điều khiển S, R. Chân S gọi là đầu vào "lập" (SET) và R được gọi là đầu vào "xóa" (RESET).

Hình 4.4 là ký hiệu của trigơ RS trong các sơ đồ logic. Hình 4.5 là sơ đồ nguyên lý của trigơ RS và RS đồng bộ.



Hình 4. 5. Sơ đồ nguyên lý của trigơ RS không đồng bộ cổng NOR và cổng NAND

b) Nguyên lý hoạt động của trigơ RS không đồng bộ cổng NAND

Khi không có tín hiệu, tức là $\bar{R} = \bar{S} = 1$, mạch có hai trạng thái ổn định $Q = 0$ và $\bar{Q} = 1$ hoặc $Q = 1$ và $\bar{Q} = 0$. Đầu ra của cổng này được nối trực tiếp với đầu vào của cổng kia, mạch có hồi tiếp dương, do đó mạch hoàn toàn duy trì trạng thái hiện có.

Giả sử trigơ có trạng thái $Q = 0$ và $\bar{Q} = 1$, đưa một xung âm vào đầu \bar{S} mạch chuyển nhanh sang trạng thái $Q = 1, \bar{Q} = 0$, và tự động duy trì ở trạng thái này. Vì thế xung âm đầu vào gọi là xung kích.

Giả sử trigơ có trạng thái $Q = 1, \bar{Q} = 0$, đưa một xung âm vào đầu \bar{R} mạch chuyển nhanh sang trạng thái $Q = 0, \bar{Q} = 1$.

Vì tín hiệu ở đầu vào \bar{S} có thể và chỉ có thể thiết lập trigơ ở trạng thái $Q = 1$, tín hiệu ở đầu vào \bar{R} có thể và chỉ có thể xóa trigơ vì trạng thái $Q = 0$, nên thường gọi \bar{S} là đầu vào đặt (Set) và đầu ra \bar{R} là đầu vào xóa (Reset).

\bar{S}	\bar{R}	Q^{n+1}	Mod hoạt động
0	0	x	Cấm
0	1	1	Lập
1	0	0	Xóa
1	1	Q^n	Nhớ

Bảng 4.1. Bảng trạng thái của trigơ RS cổng NAND

Mạch không cho phép đồng thời đưa tín hiệu vào cả \bar{R} và \bar{S} , tức là trạng thái $\bar{R} = \bar{S} = 0$ bị cấm. Vì nếu $\bar{R} = \bar{S} = 0$ thì Q và \bar{Q} đồng thời bằng 1, không phải là trạng thái ổn định của trigơ RS, và không thể xác định trạng thái trước của nó.

Chúng ta có thể xem Q^n, \bar{R}, \bar{S} là biến của hàm logic Q^{n+1} . Căn cứ vào bảng Các nô hình 4.6 tìm được phương trình đặc trưng:

$$\begin{cases} Q^{n+1} = S + \bar{R} \cdot Q^n \\ \bar{S} \cdot \bar{R} = 1 \end{cases} \quad (4.1)$$

Hình 4.6 trình bày bảng Các nô của Q^{n+1}

$\bar{S} \bar{R}$				
Q^n	00	01	11	10
0	X	1	0	0
1	X	1	1	0

Hình 4. 6. Bảng Các nô của Q^{n+1}

Bảng trạng thái của trigơ RS dùng cổng NOR được cho ở bảng 4.2.

S	R	Q^{n+1}	Mod hoạt động
0	0	Q^n	Nhớ
0	1	0	Xoá
1	0	1	Lập
1	1	x	Cấm

Bảng 4.2. Bảng trạng thái của trigơ RS

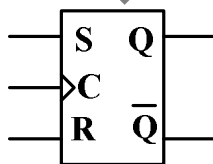
C	S	R	Q^{n+1}	Mod hoạt động
0	x	x	Q^n	Nhớ
1	0	0	Q^n	Nhớ
1	0	1	0	Xoá
1	1	0	1	Lập
1	1	1	x	Cấm

Bảng 4.3. Bảng trạng thái của trigơ RS đồng bộ cổng NAND

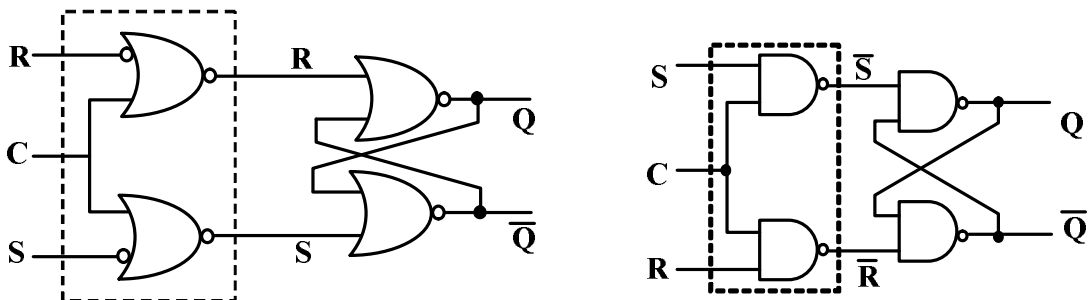
Đặc điểm của Trigơ cơ bản:

- Ưu điểm: Mạch đơn giản, có thể nhớ 1 bit, là cơ sở để cấu trúc các Trigơ hoàn hảo hơn.
- Nhược điểm: Tín hiệu trực tiếp điều khiển trạng thái đầu ra, ứng dụng bị hạn chế, tín hiệu vào ràng buộc lẫn nhau, không ở trạng thái cấm.

c) Trigơ RS đồng bộ



Hình 4. 7. Sơ đồ ký hiệu trigơ RS đồng bộ



Hình 4. 8. Sơ đồ nguyên lý của trigơ RS đồng bộ cổng NOR và cổng NAND

Sơ đồ ký hiệu của trigơ RS đồng bộ được chỉ ra trên hình 4.7.

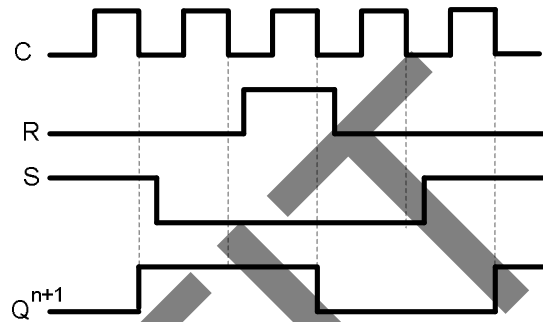
Để khắc phục nhược điểm của loại trigơ RS cơ bản là trực tiếp điều khiển, người ta thêm vào 2 cổng điều khiển và một tín hiệu điều khiển, đó chính là trigơ RS đồng bộ được trình bày ở hình 4.8.

Nguyên lí hoạt động của trigơ RS đồng bộ cổng NAND:

Khi $C = 0$ các cổng vào bị ngắt, trigơ RS cơ bản không tiếp thu tín hiệu vào, mạch được duy trì trạng thái cũ. Khi $C = 1$ các cổng vào thông, mạch sẵn sàng tiếp thu tín hiệu vào R, S.

Nguyên lí hoạt động trigơ RS đồng bộ cũng giống như trigơ RS cơ bản, chúng chỉ khác là chỉ tiếp nhận tín hiệu đầu vào R, S khi $C = 1$. Vì lí do đó nên gọi là trigơ RS đồng bộ. Bảng trạng thái được chỉ ra trên bảng 4.3.

Đồ thị thời gian dạng xung được trình bày trên hình 4.9.

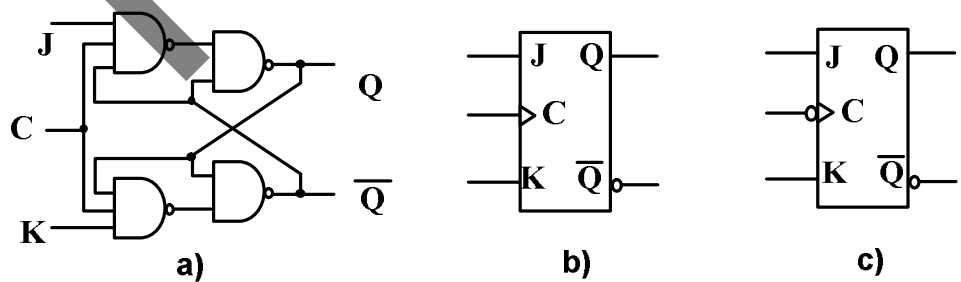


Hình 4. 9. Đồ thị thời gian dạng xung của trigơ RS

Ưu điểm: Điều khiển chọn mở mạch, trigơ chỉ tiếp thu tín hiệu vào khi $C = 1$.

Nhược điểm: Trong thời gian $C = 1$, tín hiệu vào vẫn trực tiếp điều khiển trạng thái đầu ra của trigơ.

4.2.1.2. Trigơ JK



Hình 4. 10 a) Sơ đồ nguyên lý của trigơ JK cổng NAND; b) Sơ đồ ký hiệu của trigơ JK cổng NAND; c) Sơ đồ ký hiệu của trigơ JK cổng NOR.

Trigơ JK là loại trigơ có hai đầu vào điều khiển J, K. Trigơ này có ưu điểm hơn trigơ RS là không còn tồn tại tổ hợp cấm bằng các đường hồi tiếp từ Q về chân R và từ \bar{Q} về S. Tuy nhiên, điểm đặc biệt là trigơ JK còn có thêm đầu vào đồng bộ C. Trigơ có thể lập hay xoá trong khoảng thời gian ứng với sườn âm hoặc sườn dương của xung đồng bộ C. Trigơ JK thuộc loại đồng bộ. Hình 4.10 là sơ đồ nguyên lý, sơ đồ ký hiệu của trigơ JK.

Sự hoạt động của trigơ JK được trình bày bằng bảng trạng thái 4.4.

C	J	K	Q^{n+1}	Mod hoạt động
0	x	x	Q^n	Nhớ (đối với loại trigơ JK dùng cổng NAND)
1	x	x	Q^n	Nhớ (đối với loại trigơ JK dùng cổng NOR)
Ck	0	0	Q^n	Nhớ
Ck	0	1	0	Xoá
Ck	1	0	1	Lập
Ck	1	1	$\overline{Q^n}$	Thay đổi trạng thái theo mỗi xung nhịp

Bảng 4.4. Bảng trạng thái của trigơ JK đồng bộ

Từ bảng trạng thái, lập bảng Các nô để tìm phương trình đặc trưng: J, K và Q^n là biến (bỏ qua clock) và Q^{n+1} là hàm:

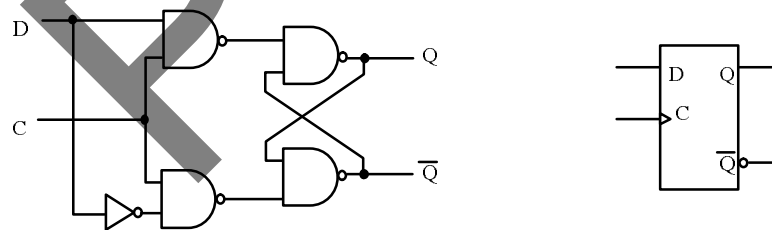
		JK			
Q^n	00	01	11	10	
	0	0	1	1	$J \cdot Q^n$
1	1	0	0	1	$\overline{K} \cdot Q^n$

Hình 4. 11. Bảng Các nô tìm phương trình đặc trưng của trigơ JK

Từ hình 4.11, có phương trình:

$$Q^{n+1} = J \cdot \overline{Q^n} + \overline{K} \cdot Q^n \quad (4.1)$$

4.2.1.3. Trigon D



Hình 4. 12. Sơ đồ nguyên lý và sơ đồ ký hiệu của trigơ D đồng bộ

Trigon D có sơ đồ nguyên lý và sơ đồ kí hiệu trên hình 4.10 có 1 đầu vào D và 2 đầu ra Q và \overline{Q} . Bảng 4.5 là bảng trạng thái của trigơ D

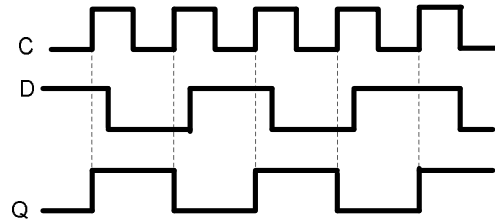
Q^n	D	Q^{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

Bảng 4.5. Bảng trạng thái trigơ D

Từ bảng trạng thái 4.5, tìm được phương trình đặc trưng của trigơ D:

$$Q^{n+1} = D \quad (4.2)$$

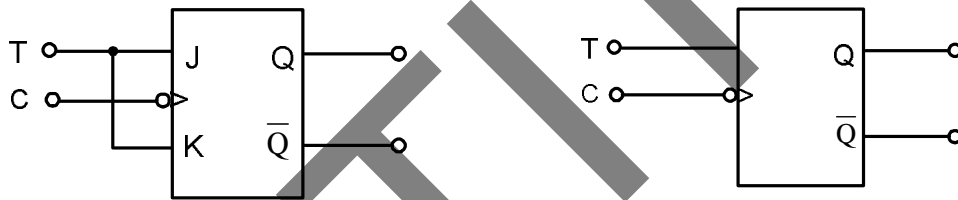
với điều kiện đã xuất hiện sườn dương C, nếu điều kiện này không thỏa mãn, trigơ giữ nguyên trạng thái cũ. Đồ thị thời gian của dạng xung của trigơ D được trình bày trên hình 4.13.



Hình 4. 13. Dạng xung của Trigơ D

4.2.1.4. Trigơ T (Toggle – lật)

Trigơ T là mạch điện có chức năng duy trì và chuyển đổi trạng thái tùy thuộc vào tín hiệu đầu vào T trong điều kiện định thời của C, kí hiệu logic được trình bày trên hình 4.14.



Hình 4. 14. Kí hiệu logic của Trigơ T

Nếu cho $J = K = T$, trigơ JK chuyển thành trigơ T. Phương trình đặc trưng:

$$Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n = T\bar{Q}^n + \bar{T}Q^n$$

$$Q^{n+1} = T \oplus Q^n \quad (4.3)$$

với điều kiện xuất hiện sườn âm C.

$T = 0$ dù có sườn âm của C thì $Q^{n+1} = Q^n$ vào duy trì trạng thái cũ.

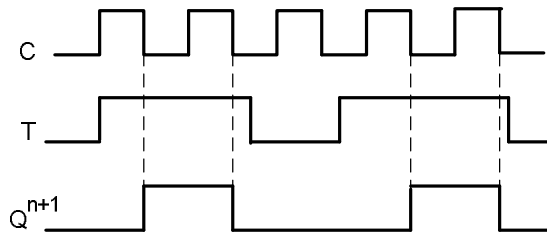
$T = 1$ có sườn âm của C thì $Q^{n+1} = \bar{Q}^n$

Bảng 4.6: Bảng trạng thái của Trigơ T

Q^n	T	Q^{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Bảng 4.6: Bảng trạng thái của Trigơ T

Đồ thị thời gian dạng sóng của Trigơ T được chỉ ra trên hình 4.15.



Hình 4. 15. Đồ thị thời gian dạng sóng của Triger T

4.2.1.4. Triger T'

Triger T' là mạch điện chỉ có chức năng chuyển đổi trạng thái trong điều kiện định thời của C. Triger T' là Triger T mà $T=1$. Phương trình đặc trưng:

$$Q^{n+1} = T \oplus Q^n = 1 \oplus Q^n = \overline{Q^n} \quad (4.4)$$

với điều kiện xuất hiện sườn âm của C.

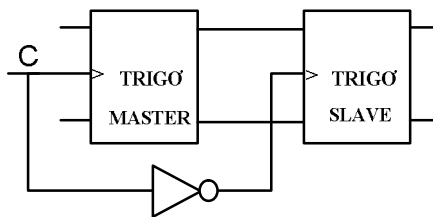
Đồ thị thời gian dạng sóng của Triger T' được trình bày trên hình 4.16.



Hình 4. 16. Đồ thị thời gian dạng sóng của Triger T'

Nhận xét: Các triger D và RS có thể làm việc được ở chế độ không đồng bộ vì mỗi tập tín hiệu vào điều khiển D, RS luôn luôn tồn tại ít nhất 1 trong 2 trạng thái ổn định. Trạng thái ổn định là trạng thái thỏa mãn điều kiện $Q^k = Q$. Còn triger T và triger JK không thể làm việc được ở chế độ không đồng bộ vì mạch sẽ rơi vào trạng thái dao động nếu như tập tín hiệu vào là '11' đối với triger JK hoặc là '1' đối với triger T. Như vậy, triger D, triger RS có thể làm việc ở cả hai chế độ: đồng bộ và không đồng bộ còn triger T và triger JK chỉ có thể làm việc ở chế độ đồng bộ.

4.2.1.4. Các loại triger Chủ-tớ (MS-Master- Slave).



Hình 4. 17. Cấu trúc của triger MS

Do các loại triger đồng bộ trên đều hoạt động tại sườn dương hay sườn âm của xung nhịp nên khi làm việc ở tần số cao thì đầu ra Q không đáp ứng kịp với sự thay đổi của xung nhịp, dẫn đến mạch hoạt động ở tình trạng không được tin cậy. Loại triger MS khắc phục được nhược điểm này. Đầu ra của triger MS thay đổi tại sườn dương và sườn âm của xung nhịp, nên cấu trúc của nó gồm 2 triger giống nhau nhưng cực tính điều khiển của xung Clock thì ngược nhau để đảm bảo sao cho tại mỗi sườn của xung sẽ có một triger hoạt động. Về nguyên tắc

hoạt động của loại trigơ MS (RS-MS, JK-MS, D-MS, T-MS) hoàn toàn giống như các loại trigơ thông thường (RS, JK, D, T).

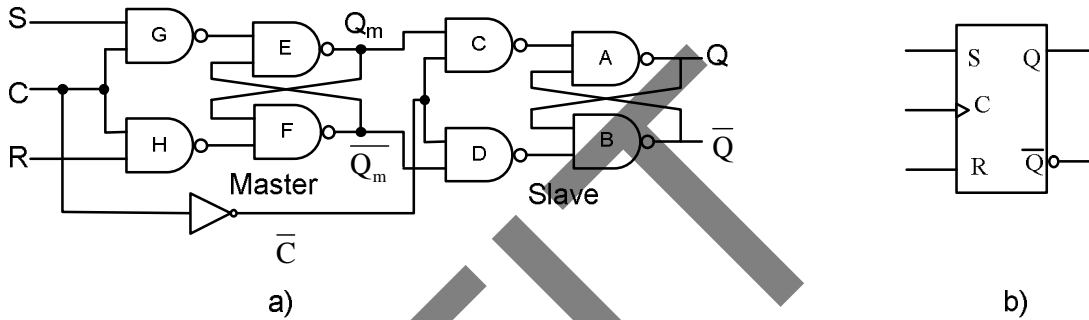
Cấu trúc chung của một trigơ MS được minh hoạ ở hình 4.17.

a) Trigơ RS Master- Slave

Mạch này giải quyết triệt để vấn đề trực tiếp điều khiển, đó là nhược điểm của các loại trigơ trên.

Cấu trúc mạch và nguyên lí hoạt động.

Hình 4.18 là sơ đồ của trigơ RS Master Slave gồm 2 trigơ RS cơ bản ghép liên tiếp với nhau, một là trigơ RS Master (trigơ chủ), một là trigơ RS Slave (trigơ tớ), xung đồng hồ cung cấp cho chúng là đảo nhau (qua mạch đảo).



Hình 4. 18. Trigơ RS Master- Slave. Sơ đồ logic (a) ; ký hiệu logic (b)

Nguyên lí hoạt động:

+ Khi $C = 0$ cổng G, H ngắt nên trigơ Master ngắt, $\bar{C} = 1$ cổng C, D thông, nó tiếp thu tín hiệu đầu ra Master do đó $Q = Q^m$, $\bar{Q} = \bar{Q}_m$

+ Sau đột biến sườn dương C.

$C = 1$, trigơ Master thông qua các cổng G, H tiếp nhận tín hiệu đầu vào, do đó:

$$Q_m^{n+1} = S + \bar{R}.Q_m^n$$

Với điều kiện S, R không đồng thời bằng không.

Khi $C = 1$ thì $\bar{C} = 0$, trigơ Slave ngắt đầu ra Q và \bar{Q} vẫn duy trì trạng thái cũ.

+ Khi có sườn âm của C.

Xung nhịp C đột biến xuống 0, Trigơ Master ngắt, khi đó \bar{C} đột biến lên 1, Trigơ Slave tiếp nhận tín hiệu đã được trigơ Master ghi nhớ từ thời gian $C = 1$, nghĩa là trigơ Slave chuyển đổi trạng thái theo biểu thức logic:

$$Q^{n+1} = S + \bar{R}.Q^n \quad (4.5)$$

Với điều kiện đã xuất hiện sườn âm của xung đồng hồ C.

Như vậy trigơ Master Slave đã giải quyết triệt để vấn đề trực tiếp điều khiển. Trạng thái đầu ra không chịu ảnh hưởng trực tiếp của các đầu vào R, S.

Đặc điểm cơ bản:

- Ưu điểm: Cấu trúc điều khiển Master Slave đã giải quyết vấn đề trực tiếp điều khiển, trong khi $C = 1$ tiếp thu tín hiệu, sườn âm của C kích thích chuyển trạng thái đầu ra.

- Nhược điểm: Vẫn còn ràng buộc giữa R và S khi $C = 1$.

b) Trigơ JK Master - Slave

Cấu trúc mạch và nguyên lí hoạt động.

Loại trigơ RS master Slave được trình bày trên hình 4.16 vẫn còn sự ràng buộc của R và S. Khi $R = S = 1$, các cổng G và H đều ở mức thấp, dẫn đến trạng thái cấm $Q_m = 1$ và $\bar{Q}_m = 1$.

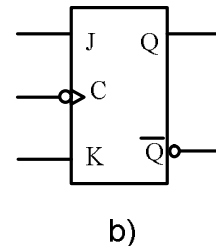
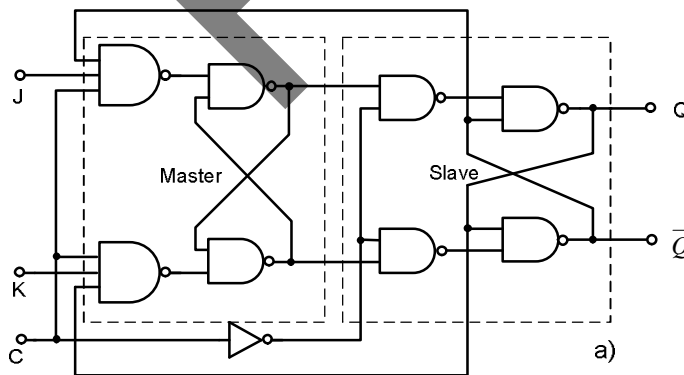
Chú ý một điểm sau: Khi $C = 1$, trigơ Master Slave, Q và \bar{Q} không đổi trạng thái và là đảo của nhau. Do đó chỉ cần đem mức các đầu ra Q và \bar{Q} đưa đến đầu vào của G và H thì có thể khắc phục được tình trạng cả Q_m và \bar{Q}_m đều bằng 1.

Để giải quyết vấn đề trên người ta cải tiến trigơ RS Master Slave theo sơ đồ được trình bày trên hình 4.17, và không dùng tên đầu vào RS mà gọi là J, K. Trigơ JK Master Slave và gọi tắt là trigơ JK.

Theo trình bày trên về sự cải tiến của trigơ JK, nguyên tắc hoạt động giống như trigơ RS Master Slave, chỉ khác sự tương đương sau của tín hiệu đầu vào:

$$S = J \cdot \bar{Q}^n \quad (4.6)$$

$$R = K \cdot Q^n \quad (4.7)$$



Hình 4. 19. Trigơ JK Master Slave Sơ đồ logic (a) Kí hiệu logic (b)

Thay công thức (4.7) và (4.8) vào công thức (4.6) được phương trình:

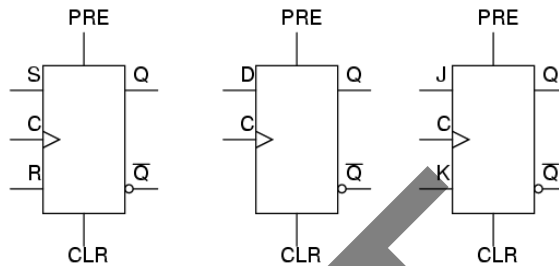
$$Q^{n+1} = S + \bar{R} \cdot Q^n = J \cdot \bar{Q}^n + \overline{K \cdot Q^n} \cdot Q^n = J \cdot \bar{Q}^n + (\bar{K} + \bar{Q}^n) \cdot Q^n$$

$$Q^{n+1} = J \cdot \bar{Q}^n + \bar{K} \cdot Q^n \quad (4.8)$$

Với điều kiện đã xuất hiện sườn âm của C, công thức (4.9) là phương trình đặc trưng của trigơ JK.

4.2.2. Đầu vào không đồng bộ của trigơ.

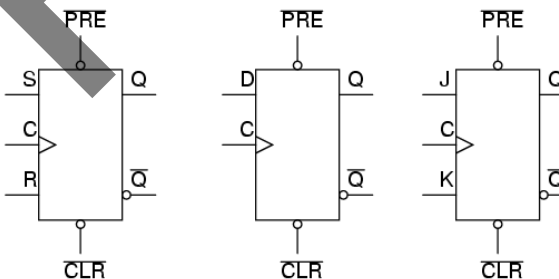
Các đầu vào dữ liệu thông thường của trigơ như D, S, R, J hoặc K là những đầu vào đồng bộ vì chúng có ảnh hưởng lên các đầu ra khi có tác động của xung Clock. Trong thực tế các trigơ còn có thêm 2 đầu vào không đồng bộ, các lỗi này tác động trực tiếp lên các đầu ra mà không phụ thuộc vào xung Clock. Các đầu vào này thường được ký hiệu là: PRE (lập) và CLR (R -xóa) hoặc $\overline{\text{PRE}}$ và $\overline{\text{CLR}}$ ($\overline{\text{R}}$). Xem hình 4.20 và 4.21.



Hình 4. 20. Đầu vào không đồng bộ ở mức tích cực cao

Khi đầu vào PRE (preset) được kích hoạt thì trigơ sẽ ở trạng thái lập ($Q = 1$ và $\overline{Q} = 0$), lúc này trigơ không phụ thuộc vào các đầu vào đồng bộ hay xung Clock. Khi lỗi CLR (clear) được kích hoạt thì trigơ sẽ ở trạng thái xóa ($Q = 0$ và $\overline{Q} = 1$), lúc này trigơ cũng không phụ thuộc vào các đầu vào đồng bộ hay xung Clock. Nếu cả hai lỗi PRE và CLR cùng được kích hoạt thì trạng thái đầu ra sẽ không phù hợp, lúc này Q và \overline{Q} có cùng một trạng thái.

Các đầu vào không đồng bộ cũng như các đầu vào đồng bộ có thể được thiết kế ở mức tích cực cao hoặc thấp. Nếu chúng hoạt động ở mức tích cực thấp thì sẽ có dấu tròn đảo ở ký hiệu logic giống như ký hiệu sườn âm của xung Clock.

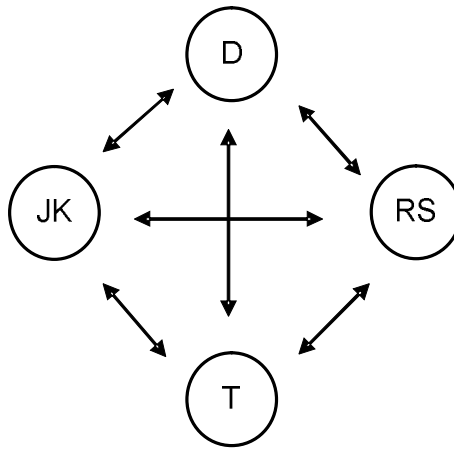


Hình 4. 21. Đầu vào không đồng bộ ở mức tích cực thấp

4.2.3. Chuyển đổi giữa các loại trigơ.

Trong thiết bị số có thể cần nhiều loại trigơ khác nhau. Tuy nhiên các IC trong công nghiệp thường chỉ sản xuất một số loại nhất định phổ biến là trigơ D, JK. Mặt khác trong một IC thường có 2 đến 4 trigơ cùng loại, có thể gây ra sự dư thừa loại này nhưng lại thiếu loại khác. Do đó, việc chuyển đổi giữa các loại trigơ là cần thiết. Tìm phương pháp để chuyển đổi giữa các loại trigơ.

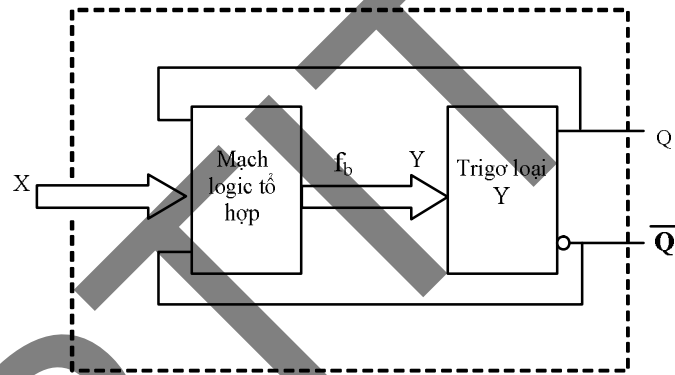
Với 4 loại trigơ trên thì có 12 khả năng chuyển đổi sang nhau.



Hình 4. 22. Các khả năng chuyển đổi trigơ

4.2.3.1. Phương pháp chuyển đổi giữa các loại trigơ.

Nguyên tắc chung



Hình 4. 23. Sơ đồ khối của phương pháp chuyển đổi trigơ

Chuyển đổi giữa các loại trigơ là việc biến chức năng của một trigơ đã cho thành một trigơ khác loại. Để thực hiện được công việc này bắt buộc phải bổ sung thêm một mạch liên hợp phụ. Sơ đồ khối thể hiện một trigơ đã chuyển đổi được trình bày trên hình 4.23.

Phương pháp chuyển đổi là tìm quan hệ hàm ra của mạch kích thích phụ f_b cũng chính là hàm kích thích của trigơ đã cho với đầu ra của nó và kích thích vào.

Việc chuyển đổi giữa các loại trigơ có thể thực hiện theo một trong hai phương pháp sau:

- Dùng phương trình đặc trưng của trigơ đã cho, kết hợp với phép biến đổi logic cần thiết để đưa nó về phương trình đặc trưng của trigơ cần tìm. Trên cơ sở phương trình đặc trưng, viết ra hàm kích thích và vẽ sơ đồ tương ứng của trigơ.

Phương pháp này tương đối nhanh, nhưng cần kỹ năng biến đổi hàm logic.

- Dùng bảng kích thích của trigơ đã cho và trigơ cần tìm để xác định hàm kích thích sau đó vẽ sơ đồ logic của trigơ cần tìm.

Như đã biết, lập bảng trạng thái cho các trigơ có nghĩa là tìm trạng thái kế tiếp của trigơ khi đã biết tín hiệu vào, còn bảng hàm kích là bảng tìm hàm kích (tín hiệu vào) khi đã biết

trạng thái ra. Nói cách khác, lập bảng hàm kích là quá trình ngược lại với việc lập bảng trạng thái.

Từ bảng trạng thái của các trigơ, lập được bảng hàm kích 4.7:

Q^n	Q^{n+1}	RS	JK	D	T
0	0	X0	0X	0	0
0	1	01	1X	1	1
1	0	10	X1	0	1
1	1	0X	X0	1	0

Bảng 4.7. Bảng hàm kích của các loại trigơ

Sau đây, xét một số ví dụ xây dựng các trigơ từ các trigơ cho trước thường hay được sử dụng trong thực tế.

4.2.3.2. Chuyển đổi giữa các loại trigơ

a) Chuyển đổi từ trigơ RS sang JK, D, T, T'.

Chuyển đổi từ trigơ RS sang JK. Để chuyển đổi cần tìm mối quan hệ:

$$S = f_1(Q, J, K); \quad R = f_2(Q, J, K)$$

Điều này có thể thực hiện bằng hai cách:

+ **Dùng phương trình đặc trưng**

Phương trình đặc trưng của trigơ RS có dạng:

$$\begin{cases} Q^{n+1} = S + \bar{R}.Q^n \\ R.S = 0 \end{cases} \quad (4.9)$$

còn phương trình đặc trưng của trigơ JK có dạng:

$$Q^{n+1} = J.\bar{Q}^n + \bar{K}.Q^n \quad (4.10)$$

So sánh (4.10) và (4.11) tìm được:

$$S = J.\bar{Q}^n$$

$$R = K$$

Tuy nhiên, điều kiện ràng buộc $R.S = 0$ nên khi $J = K = 1$; $Q^n = 0$ thì

$$\begin{cases} R = K = 1 \\ S = J.\bar{Q}^n = 1 \end{cases}$$

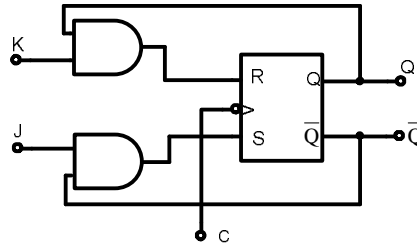
Sẽ không thỏa mãn điều kiện $R.S = 0$.

Có thể biến đổi lại:

$$\begin{aligned} Q^{n+1} &= J.\bar{Q}^n + \bar{K}.Q^n + Q^n.\bar{Q}^n = J.\bar{Q}^n + (\bar{K} + \bar{Q}^n).Q^n \\ Q^{n+1} &= J.\bar{Q}^n + \bar{K}.Q^n.Q^n \end{aligned} \quad (4.11)$$

So sánh (4.10) và (4.12), tìm được hàm kích:

$$\begin{cases} S = J.\overline{Q}^n \\ R = K.Q^n \end{cases} \quad (4.12)$$



Hình 4. 24. Mạch điện chuyển đổi từ RS sang JK

Hình 4.24 là mạch điện chuyển đổi từ RS \rightarrow JK (thỏa mãn điều kiện R.S = 0).

+ Dùng bảng kích thích.

JK	00	01	11	10
Q^n 0	0	0	1	1
Q^n 1	x	0	0	x

a) Đối với S

JK	00	01	11	10
Q^n 0	x	x	0	0
Q^n 1	0	1	1	0

b) Đối với R

Hình 4. 25. Bảng Các nô xác định sự phụ thuộc của S, R vào Q, J, K

Từ bảng hàm kích thích 4.7. Nếu xem Q^n , J, K là biến và S, R là hàm lập được hai bảng Các nô như hình 4.25. Từ bảng Các nô, tìm được kết quả giống công thức (4.13).

$$\begin{cases} S = J.\overline{Q}^n \\ R = K.Q^n \end{cases}$$

Tương tự, tìm được hàm logic chuyển đổi:

RS \rightarrow D

$$\begin{cases} R = \overline{D} \\ S = D \end{cases} \quad (4.13)$$

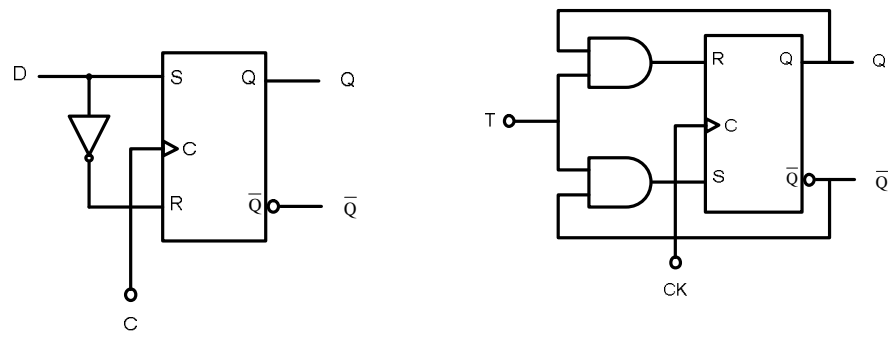
RS \rightarrow T

$$\begin{cases} R = T.Q^n \\ S = T.\overline{Q}^n \end{cases} \quad (4.14)$$

RS \rightarrow T'

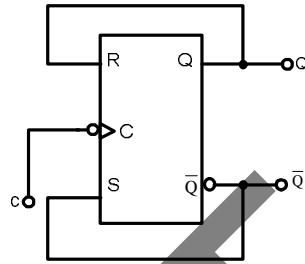
$$\begin{cases} R = Q^n \\ S = \overline{Q}^n \end{cases} \quad (4.15)$$

Mạch chuyển đổi được trình bày trên hình 4.26.



a) RS→D

b) RS→T



b) RS→T'

Hình 4. 26. Mạch chuyển đổi từ trigơ RS trở thành a) D ; b) T ; c) T'

b) Chuyển đổi từ trigơ JK sang RS, D, T, T'.

Bằng phương pháp tương tự như phần trước, thu được các biểu thức và sơ đồ cần tìm:

+ Từ trigơ JK sang RS.

Biểu thức:

$$\left. \begin{array}{l} J = S \\ K = R \end{array} \right\} \quad (4.16)$$

→ Sơ đồ 4.27a)

+ Từ JK sang D

Biểu thức:

$$\left. \begin{array}{l} J = D \\ K = \bar{D} \end{array} \right\} \quad (4.17)$$

→ Sơ đồ hình 4.27b).

+ Từ JK sang T

Biểu thức:

$$\left. \begin{array}{l} J = T \\ K = T \end{array} \right\} \quad (4.18)$$

→ Sơ đồ hình 4.27c).

+ Từ JK sang T'

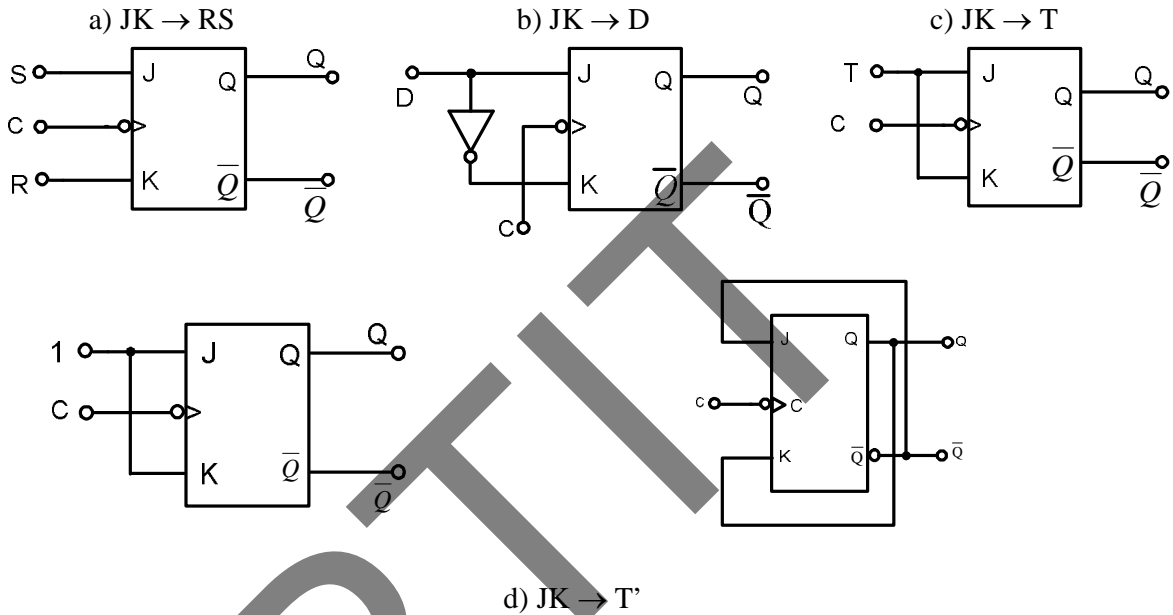
Biểu thức:

$$\left. \begin{array}{l} J = T = 1 \\ K = T = 1 \end{array} \right\} \quad (4.19)$$

Hoặc

$$\left. \begin{array}{l} J = \overline{Q^n} \\ K = Q^n \end{array} \right\} \quad (4.20)$$

→ Sơ đồ hình 4.27d).



Hình 4. 27. Chuyển đổi từ trigger JK sang a) RS ; b) D ; c) T ; d) T'

c) Chuyển từ D sang các loại

+ Từ D sang RS

Cân bằng hai phương trình đặc trưng. Tìm được biểu thức:

$$D = S + \overline{R}Q^n \quad (4.21)$$

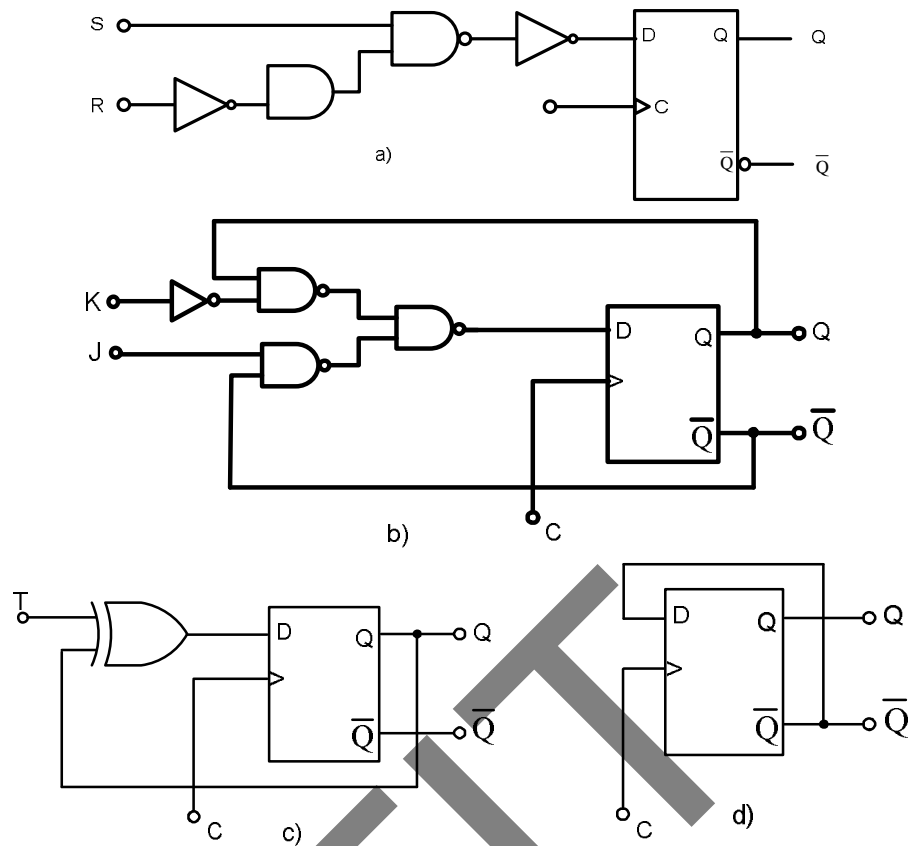
→ Sơ đồ hình 4.28a)

+ Từ D sang JK

Biểu thức:

$$D = J\overline{Q^n} + \overline{K}.Q^n \quad (4.22)$$

→ Sơ đồ hình 4.28b)



Hình 4. 28. Mạch điện biến đổi từ D sang a) RS ; b) JK ; c) T ; d) T'

+ Từ D sang T

Biểu thức:

$$D = T \oplus Q^n \quad (4.23)$$

→ Sơ đồ hình 4.28c)

+ Từ D sang T'

Biểu thức:

$$D = T \oplus Q^n$$

$$T = 1 \rightarrow D = \overline{Q^n} \quad (4.24)$$

→ Sơ đồ hình 4.28d)

d) Chuyển từ T sang RS, JK, D

+ Từ T sang RS

Biểu thức:

$$T = \overline{S.Q^n} + R.Q^n \quad (4.25)$$

→ Sơ đồ hình 4.29a).

+ Từ J sang JK:

Biểu thức:

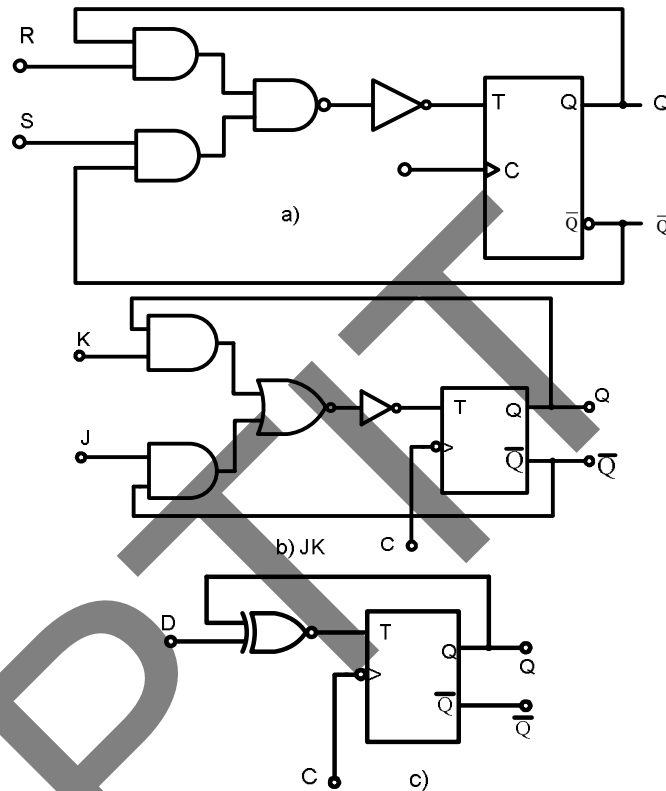
$$T = \overline{J}Q^n + \overline{K}.Q^n \quad (4.26)$$

Sơ đồ hình 4.29a) chỉ cần thay $R = K$ và $S = J$.

+ Từ T sang D:

Biểu thức: $T = D \oplus Q^n$

Mạch điện được thể hiện hình 4.29c).



Hình 4. 29. Mạch điện chuyển từ T sang a) RS ; b) JK ; c) D

4.3. PHÂN TÍCH MẠCH TUẦN TỰ.

4.3.1. Các bước phân tích mạch tuần tự đồng bộ

Bài toán phân tích là bài toán xác định chức năng của một mạch cho trước. Khi tiến hành phân tích cần tuân theo các bước sau:

- Sơ đồ mạch: Từ sơ đồ cho trước cần xác định chức năng từng phần tử cơ bản của sơ đồ, mối quan hệ giữa các phần tử đó.

- Xác định các đầu vào và ra, số trạng thái trong của mạch: Coi mạch như một hộp đen cần phải xác định các đầu vào và ra của mạch, đặc điểm của các đầu vào, đầu ra. Để xác định được số trạng thái trong của mạch cần phải xác định xem mạch được xây dựng từ bao nhiêu phần tử nhớ (trigơ JK) từ đó xác định được số trạng thái trong có thể có của mạch.

Gọi số trigơ là n thì số trạng thái có thể có của mạch là 2^n .

+ Xác định các phương trình:

- Xác định phương trình định thời của các trigơ.
- Xác định phương trình hàm ra, phương trình hàm kích của các trigơ.
- + Lập bảng chuyển đổi trạng thái. Bảng này biểu diễn mối quan hệ trạng thái kế tiếp, tín hiệu ra nhị phân với trạng thái hiện tại và các tín hiệu vào tương ứng.

Dựa vào phương trình đặc trưng của trigơ để thiết lập phương trình chuyển đổi trạng thái và từ đó xác định được trạng thái kế tiếp và tín hiệu ra tương ứng với tín hiệu vào và trạng thái hiện tại của mạch.

+ Đồ hình trạng thái: Từ bảng trạng thái xây dựng đồ hình trạng thái và tín hiệu ra của mạch.

+ Vẽ giản đồ dạng xung.

+ Chức năng của mạch: Dựa vào đồ hình trạng thái xác định được chức năng của mạch.

4.3.2. Các bước phân tích mạch tuần tự không đồng bộ

Cũng tương tự mạch tuần tự đồng bộ mạch tuần tự không đồng bộ có các bước phân tích như sau:

- + Xác định các phương trình:
 - Xác định phương trình định thời của các trigơ.
 - Xác định phương trình hàm ra, phương trình hàm kích của các trigơ.
- + Lập bảng chuyển đổi trạng thái. Bảng này biểu diễn mối quan hệ trạng thái kế tiếp, tín hiệu ra nhị phân với trạng thái hiện tại và các tín hiệu vào tương ứng.

Dựa vào phương trình đặc trưng của trigơ để thiết lập phương trình chuyển đổi trạng thái và từ đó xác định được trạng thái kế tiếp và tín hiệu ra tương ứng với tín hiệu vào và trạng thái hiện tại của mạch. Do mạch tuần tự không đồng bộ có điều kiện kích sườn xung Clock khác nhau nên khi thiết lập bảng phải chú ý đến điều kiện kích sườn xung Clock. Chỉ khi nào đúng sườn xung Clock điều khiển thì phương trình chuyển đổi trạng thái được thiết lập nếu không trạng thái trong của bộ đếm được giữ nguyên.

+ Đồ hình trạng thái: Từ bảng trạng thái xây dựng đồ hình trạng thái và tín hiệu ra của mạch.

+ Vẽ giản đồ dạng xung (đồ thị thời gian). Vẽ giản đồ dạng xung cần lưu ý: trigơ chỉ chuyển đổi trạng thái tương ứng với sườn kích của xung đồng hồ xuất hiện.

+ Chức năng của mạch: Dựa vào đồ hình trạng thái xác định được chức năng của mạch.

4.4. BỘ ĐẾM

4.4.1. Phân tích bộ đếm.

Bộ đếm là loại mạch tuần tự đơn giản, được xây dựng từ các phần tử nhớ là các trigơ và các mạch logic tổ hợp. Bộ đếm hoạt động theo chế độ tuần hoàn, có một đầu vào và một đầu ra chính. Kết thúc vòng tuần hoàn bộ đếm lại quay về trạng thái ban đầu và hàm ra lấy giá trị 1. Ngoài các đầu vào và đầu ra chính còn có các đầu vào và đầu ra phụ. Các đầu vào phụ có thể dùng để lập, xóa trạng thái trong của bộ đếm về một giá trị ban đầu theo mong muốn, hoặc

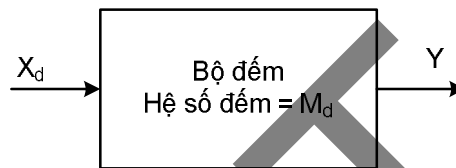
lập trình để tạo mod đếm, hoặc thay đổi tính năng của bộ đếm. Đầu ra phụ cũng có thể bao gồm nhiều loại khác nhau, các đầu ra từ Q_0 đến Q_{n-1} chính là trạng thái của bộ đếm. Khi cần có thể dẫn tín hiệu này qua một bộ giải mã để hiển thị kết quả đếm.

Phần này sẽ đưa ra những đặc điểm cơ bản nhất của bộ đếm và các phương pháp thiết kế bộ đếm.

4.4.1.1. Định nghĩa và phân loại bộ đếm

a . Định nghĩa.

Bộ đếm là một mạch tuần tự tuần hoàn có một đầu vào đếm và một đầu ra, mạch có số trạng thái trong bằng chính hệ số đếm (ký hiệu là M_d). Dưới tác dụng của tín hiệu vào đếm, mạch sẽ chuyển từ trạng thái trong này đến một trạng thái trong khác theo một thứ tự nhất định. Cứ sau M_d tín hiệu vào đếm mạch lại trở về trạng thái xuất phát ban đầu.

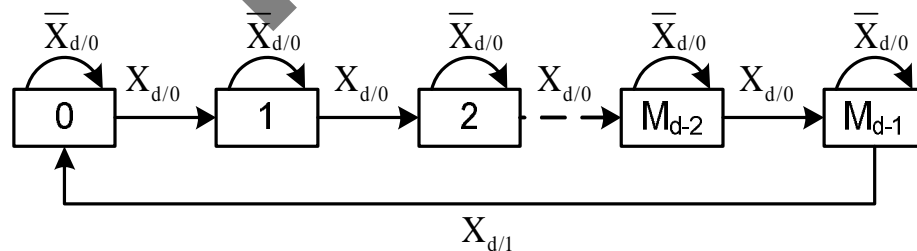


Hình 4. 30. Sơ đồ khối của bộ đếm

Các bộ đếm là phần tử cơ bản của các hệ thống số, chúng được sử dụng để đếm thời gian, chia tần số, điều khiển các mạch khác. Bộ đếm được dùng rất nhiều trong các dụng cụ đo lường chỉ thị số, các máy tính điện tử. Bất kỳ hệ thống số hiện đại nào đều sử dụng các bộ đếm. Để xây dựng các bộ đếm, người ta có thể dùng mã nhị phân, hoặc các mã khác như mã Gray, mã N BCD, mã vòng...

Sơ đồ khối được mô tả như ở hình 4. 30.

Đồ hình trạng thái của bộ đếm có hệ số đếm bằng M_d được mô tả ở hình 4.31. Khi không có tín hiệu vào đếm (\bar{X}_d) mạch giữ nguyên trạng thái cũ, khi có tín hiệu đếm (X_d) thì mạch sẽ chuyển đến trạng thái kế tiếp.



Hình 4. 31. Đồ hình trạng thái của bộ đếm M_d

Tính chất tuần hoàn của bộ đếm thể hiện ở chỗ: sau M_d tín hiệu vào X_d thì mạch lại quay trở về trạng thái xuất phát ban đầu.

Tín hiệu ra của bộ đếm chỉ xuất hiện ($Y = 1$) duy nhất trong trường hợp: bộ đếm đang ở trạng thái $M_d - 1$ và có tín hiệu vào X_d . Khi đó bộ đếm sẽ chuyển về trạng thái 0.

Trong trường hợp cần hiển thị trạng thái của bộ đếm thì phải dùng thêm mạch giải mã.

b. Phân loại bộ đếm.

Để phân loại bộ đếm có nhiều cách khác nhau:

+ Căn cứ vào sự khác biệt tình huống chuyển đổi trạng thái của trigơ trong bộ đếm, người ta phân thành hai loại: Bộ đếm đồng bộ (Synchronous), bộ đếm không đồng bộ (Asynchronous).

Trong bộ đếm đồng bộ (hay còn gọi là bộ đếm song song), các trigơ đều chịu tác dụng điều khiển của một xung đồng hồ duy nhất, đó là xung điều khiển đầu vào. Nói cách khác, sự chuyển đổi trạng thái của chúng là đồng bộ. Trong bộ đếm không đồng bộ (bộ đếm nối tiếp), có trigơ chịu tác dụng điều khiển trực tiếp của xung đếm đầu vào, nhưng cũng có trigơ chịu tác dụng điều khiển của xung ở đầu ra của trigơ khác. Vậy sự chuyển đổi trạng thái là không cùng một lúc, tức là không đồng bộ.

+ Căn cứ vào sự khác biệt của hệ số đếm của bộ đếm, người ta phân thành các loại: bộ đếm nhị phân, bộ đếm thập phân, bộ đếm N phân (hay gọi là bộ đếm bất kỳ). Nếu gọi n là số vị trí trong mã số nhị phân (cũng là số trigơ trong bộ đếm), gọi N là số trạng thái tích cực (cũng là số tổ hợp mã hóa được dùng khi lập mã), đối với bộ đếm nhị phân $N = 2^n$, đối với bộ đếm thập phân $N = 10$. Bộ đếm nhị phân và bộ đếm thập phân là trường hợp riêng của bộ đếm N phân. Thường gọi N là dung lượng bộ đếm hoặc độ dài đếm của bộ đếm.

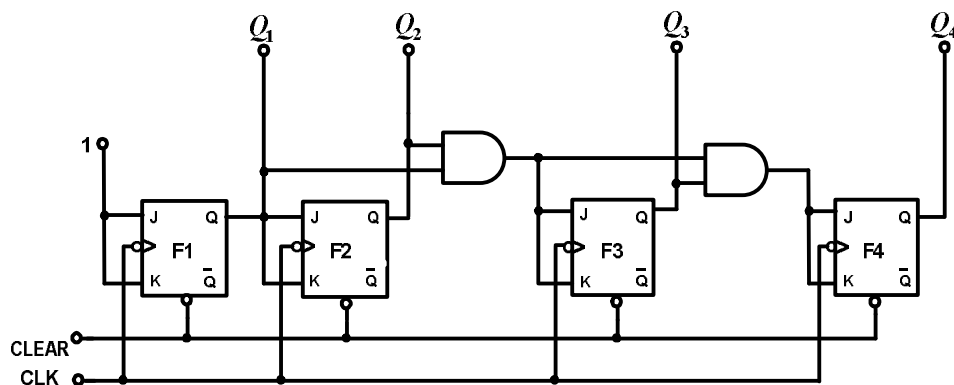
+ Căn cứ vào xung đếm đầu vào, mà số đếm của bộ đếm tăng hay giảm mà người ta phân thành 3 loại: Bộ đếm thuận (Up counter) hay còn gọi là bộ đếm tiến, bộ đếm nghịch (Down counter) hay còn gọi là bộ đếm lùi, bộ đếm thuận nghịch.

+ Phân loại theo khả năng lập trình: người ta phân thành: Bộ đếm có khả năng lập trình (Programmable counter), bộ đếm không có khả năng lập trình.

Bộ đếm có khả năng lập trình, là bộ đếm có thể sử dụng với các hệ số đếm khác nhau, tùy thuộc vào tín hiệu điều khiển đưa vào nó. Do vậy, bộ đếm này đa năng hơn các bộ đếm có một hệ số đếm cố định và ngày càng được sử dụng rộng rãi.

4.4.1.2 Bộ đếm đồng bộ

a. Bộ đếm nhị phân đồng bộ.



Hình 4. 32. Bộ đếm thuận nhị phân đồng bộ 4 bit

Bộ đếm nhị phân đồng bộ nói chung cấu trúc bằng trigơ T hoặc các trigơ loại khác nhưng mắc thành trigơ T. Hình 4.32 trình bày bộ đếm thuận nhị phân đồng bộ.

Nguyên lý làm việc:

Viết các phương trình:

Xung Clock của 4 trigơ đều là xung đếm đầu vào:

$$C_1 = C_2 = C_3 = C_4 = CLK \quad (4.27)$$

Phương trình kích:

$$\left. \begin{aligned} T_1 &= 1 \\ T_2 &= Q_1^n \\ T_3 &= Q_1^n Q_2^n \\ T_4 &= Q_1^n Q_2^n Q_3^n \end{aligned} \right\} \quad (4.28)$$

Phương trình đặc trưng của trigơ T:

$$Q^{n+1} = T\overline{Q}^n + \overline{T}Q^n$$

Thay phương trình kích vào phương trình đặc trưng, tìm được:

$$\left. \begin{aligned} Q_1^{n+1} &= T_1 \cdot \overline{Q_1^n} + \overline{T_1} \cdot Q_1^n = \overline{Q_1^n} \\ Q_2^{n+1} &= T_2 \cdot \overline{Q_2^n} + \overline{T_2} \cdot Q_2^n = Q_1^n \cdot \overline{Q_2^n} + \overline{Q_1^n} Q_2^n \\ Q_3^{n+1} &= T_3 \cdot \overline{Q_3^n} + \overline{T_3} \cdot Q_3^n = Q_1^n \cdot Q_2^n \cdot \overline{Q_3^n} + \overline{Q_1^n \cdot Q_2^n} Q_3^n \\ Q_4^{n+1} &= T_4 \cdot \overline{Q_4^n} + \overline{T_4} \cdot Q_4^n = Q_1^n \cdot Q_2^n \cdot Q_3^n \cdot \overline{Q_4^n} + \overline{Q_1^n \cdot Q_2^n \cdot Q_3^n} Q_4^n \end{aligned} \right\} \quad (4.29)$$

Khi có xung xóa Clear = 0 tất cả $Q_4^n Q_3^n Q_2^n Q_1^n = 0000$;

Clear = 1 mạch ở chế độ đếm.

Dựa vào phương trình chuyển đổi trạng thái, tính được bảng kết quả 4.8.

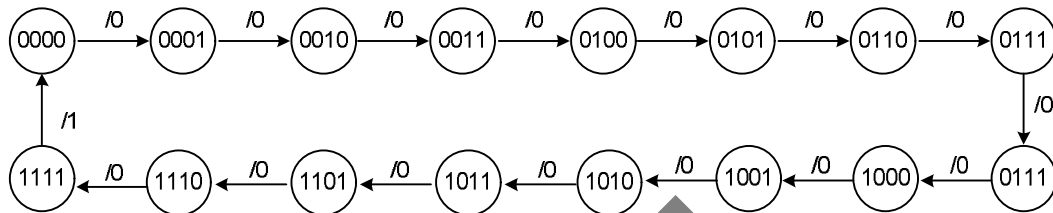
Trong đó các trạng thái kế tiếp của bộ đếm là: $Q_4^{n+1} Q_3^{n+1} Q_2^{n+1} Q_1^{n+1}$.

Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0

Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

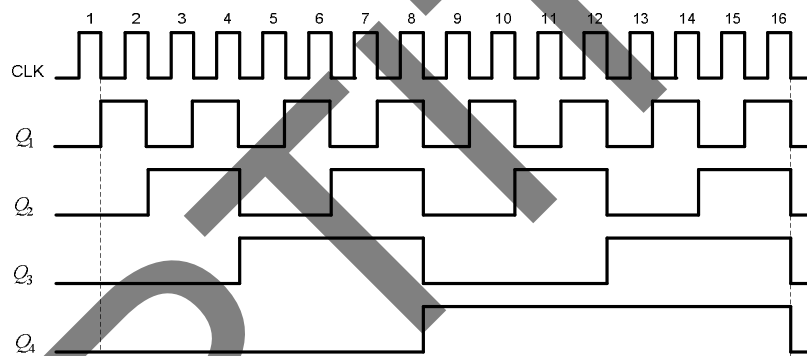
Bảng 4.8. Bảng trạng thái của bộ đếm

Từ bảng trạng thái, vẽ được đồ hình trạng thái như hình 4.33.



Hình 4. 33. Đồ hình trạng thái của bộ đếm nhị phân

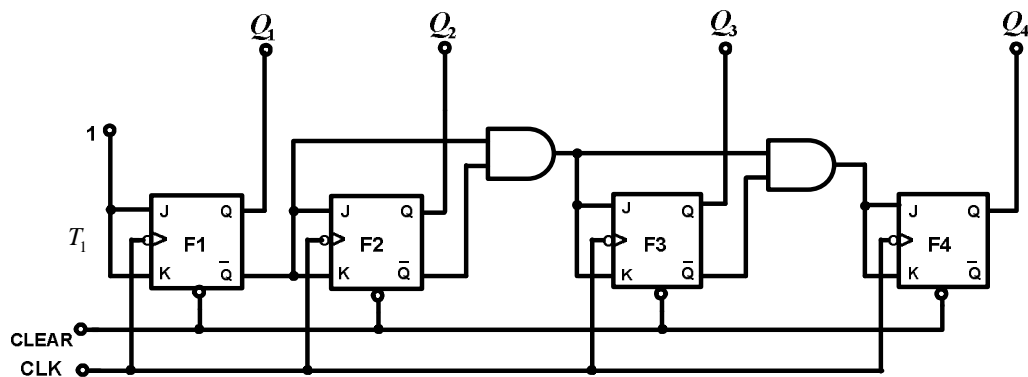
Dạng sóng của bộ đếm thuận nhị phân đồng bộ được trình bày ở hình 4.34.



Hình 4. 34. Dạng sóng bộ đếm thuận nhị phân đồng bộ

b. Bộ đếm nghịch nhị phân đồng bộ.

Sơ đồ được trình bày trên hình 4.35.



Hình 4. 35. Bộ đếm nghịch nhị phân đồng bộ

Xung Clock của 4 trigơ đều là xung đếm đầu vào:

$$C_1 = C_2 = C_3 = C_4 = \text{CLK}$$

Phương trình kích:

$$\left. \begin{aligned} T_1 &= 1 \\ T_2 &= \overline{Q_1^n} \\ T_3 &= \overline{Q_1^n} \cdot \overline{Q_2^n} \\ T_4 &= \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot \overline{Q_3^n} \end{aligned} \right\} \quad (4.30)$$

Phương trình đặc trưng của trigger T:

$$Q^{n+1} = T\overline{Q^n} + \overline{T}Q^n$$

Thay phương trình kích vào phương trình đặc trưng, tìm được:

$$\left. \begin{aligned} Q_1^{n+1} &= T_1 \cdot \overline{Q_1^n} + \overline{T_1} \cdot Q_1^n = \overline{Q_1^n} \\ Q_2^{n+1} &= T_2 \cdot \overline{Q_2^n} + \overline{T_2} \cdot Q_2^n = \overline{Q_1^n} \cdot \overline{Q_2^n} + Q_1^n \cdot Q_2^n \\ Q_3^{n+1} &= T_3 \cdot \overline{Q_3^n} + \overline{T_3} \cdot Q_3^n = \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot \overline{Q_3^n} + (Q_1^n + Q_2^n) \cdot Q_3^n \\ Q_4^{n+1} &= T_4 \cdot \overline{Q_4^n} + \overline{T_4} \cdot Q_4^n = \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot \overline{Q_3^n} \cdot \overline{Q_4^n} + (Q_1^n + Q_2^n + Q_3^n) \cdot Q_4^n \end{aligned} \right\} \quad (4.31)$$

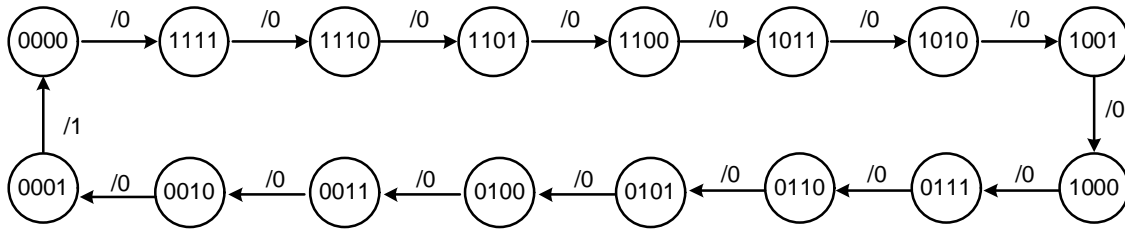
Khi có xung xóa Clear = 0 tất cả $Q_4^n Q_3^n Q_2^n Q_1^n = 0000$;

Tương tự như phần trên, dùng các phương pháp phân tích cơ bản, lập được bảng chuyển trạng thái 4.9.

Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	0	1	1	0	1
1	1	0	1	1	1	0	0
1	1	0	0	1	0	1	1
1	0	1	1	1	0	1	0
1	0	1	0	1	0	0	1
1	0	0	1	1	0	0	0
1	0	0	0	0	1	1	1
0	1	1	1	0	1	1	0
0	1	1	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0

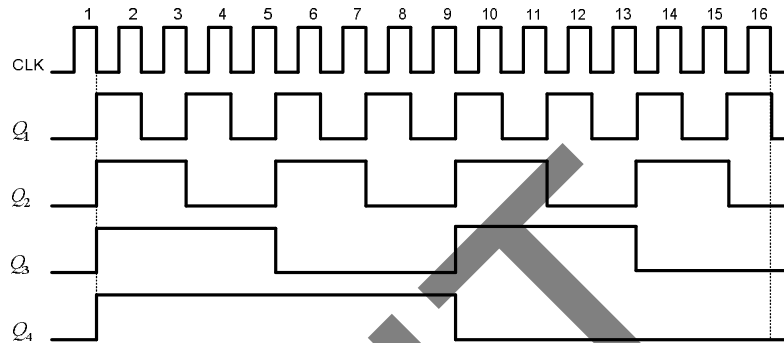
Bảng 4.9. Bảng trạng thái bộ đếm nghịch nhị phân đồng bộ

Từ bảng trạng thái, vẽ được đồ hình trạng thái như hình 4.36.



Hình 4. 36. Đồ hình trạng thái của bộ đếm nhị phân

Dạng sóng của bộ đếm nghịch được trình bày trên hình 4.37.

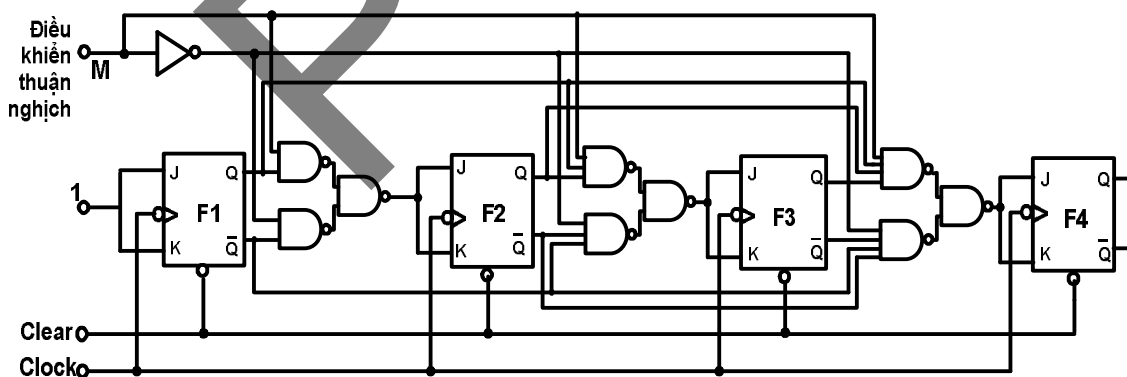


Hình 4. 37. Đồ thị dạng xung của bộ đếm nghịch đồng bộ

Nhận xét: So sánh hình 4.32 và hình 4.35, sự khác biệt của bộ đếm nghịch với bộ đếm thuận là đầu ra \bar{Q} của trigger cung cấp tín hiệu chuyển vị.

c. Bộ đếm thuận nghịch nhị phân đồng bộ.

Sơ đồ được trình bày trên hình 4.38.



Hình 4. 38. Bộ đếm thuận nghịch nhị phân đồng bộ.

Mạch có đầu vào điều khiển đếm thuận, đếm nghịch.

$M = 1$ mạch đếm thuận.

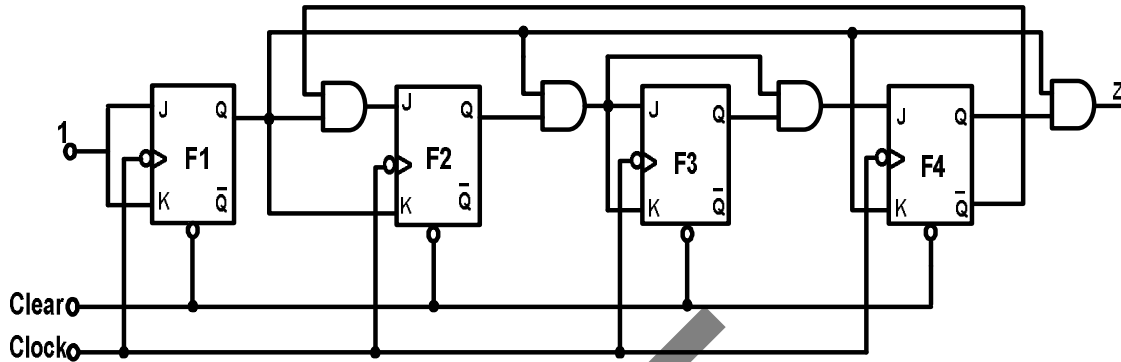
$M = 0$ mạch đếm nghịch

Đây thực chất là 2 bộ đếm thuận, nghịch gộp lại với nhau, có thêm một số cổng logic làm nhiệm vụ điều khiển.

d. Bộ đếm thập phân đồng bộ.

Bộ đếm thập phân là bộ đếm theo mã nhị- thập phân. Vì tương ứng với nhiều kiểu mã hóa của mã nhị- thập phân, nên bộ đếm cũng có nhiều bộ đếm thập phân khác nhau. Ở đây chỉ giới thiệu bộ đếm thập phân dùng mã nhị- thập phân 8421 hay NBCD.

Sơ đồ được trình bày trên hình 4.39.



Hình 4. 39. Bộ đếm thập phân đồng bộ

Mạch gồm 4 trigơ JK và một số cổng làm nhiệm vụ kích và chuyển vị (nhớ) đầu ra Z, xung đầu vào là CLK.

Nguyên lý hoạt động:

Viết phương trình.

+ Phương trình định thời

$$C_1 = C_2 = C_3 = C_4 = CLK \quad (4.32)$$

+ Phương trình hàm ra:

$$Z = Q_4^n \cdot Q_1^n \quad (4.33)$$

+ Phương trình kích:

$$\left. \begin{aligned} J_1 &= K_1 = 1 \\ J_2 &= \overline{Q_4^n} \cdot Q_1^n ; & K_2 &= Q_1^n \\ J_3 &= K_3 = Q_2^n \cdot Q_1^n \\ J_4 &= Q_3^n \cdot Q_2^n \cdot Q_1^n ; & K_4 &= Q_1^n \end{aligned} \right\} \quad (4.34)$$

+ Phương trình chuyển đổi trạng thái:

Tìm phương trình chuyển đổi trạng thái bằng cách thay phương trình kích vào phương trình đặc trưng của trigơ JK:

Phương trình đặc trưng của trigơ JK:

$$Q^{n+1} = J \cdot \overline{Q^n} + \overline{K} \cdot Q^n$$

Phương trình chuyển đổi trạng thái như sau:

$$\left. \begin{aligned}
Q_1^{n+1} &= J_1 \cdot \overline{Q_1^n} + \overline{K_1} \cdot Q_1^n = \overline{Q_1^n} \\
Q_2^{n+1} &= J_2 \cdot \overline{Q_2^n} + \overline{K_2} \cdot Q_2^n = Q_1^n \cdot \overline{Q_2^n} \cdot \overline{Q_4^n} + \overline{Q_1^n} \cdot Q_2^n \\
Q_3^{n+1} &= J_3 \cdot \overline{Q_3^n} + \overline{K_3} \cdot Q_3^n = Q_1^n \cdot Q_2^n \cdot \overline{Q_3^n} + \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot Q_3^n \\
Q_4^{n+1} &= J_4 \cdot \overline{Q_4^n} + \overline{K_4} \cdot Q_4^n = Q_1^n \cdot Q_2^n \cdot Q_3^n \cdot \overline{Q_4^n} + \overline{Q_1^n} \cdot Q_4^n
\end{aligned} \right\} \quad (4.35)$$

Lập bảng trạng thái:

Bắt đầu từ giá trị $Q_4^n Q_3^n Q_2^n Q_1^n = 0000$ thay vào phương trình trạng thái và phương trình đầu ra, được kết quả ở bảng 4.10.

Giả sử bộ đếm hình 4.39 là bộ đếm hàng đơn vị của hệ thập phân, mạch được chuyển đổi đến trạng thái 1001 (= 9)₁₀ tín hiệu nhớ trở thành mức cao, nhưng thật ra không tác động ngay mà đợi đến sườn âm của xung đồng hồ CLK thứ 10 xuất hiện, thì tín hiệu Z mới kích trigơ hàng chục của hệ thập phân chuyển đổi trạng thái, đồng thời bộ đếm đơn vị trở về 0 tức là mạch trở về trạng thái 0000.

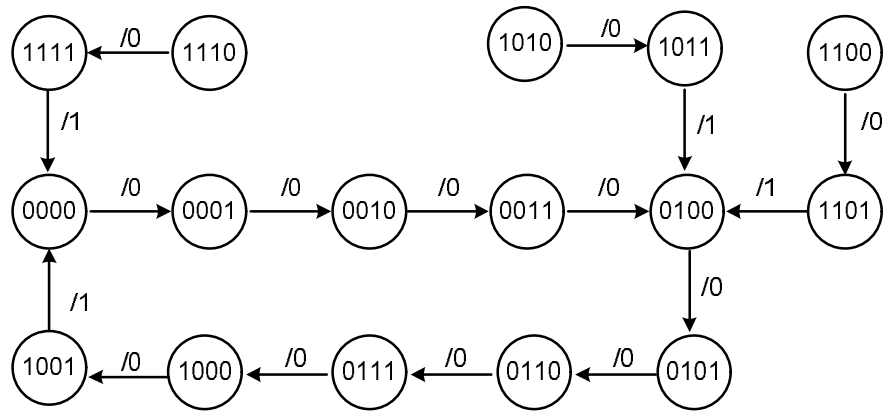
Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Z
0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	1	0
0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	0	1	0
0	1	0	1	0	1	1	0	0
0	1	1	0	0	1	1	1	0
0	1	1	1	1	0	0	0	0
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	0	1
1	0	1	0	1	0	1	1	0
1	0	1	1	0	1	0	0	1
1	1	0	0	1	1	0	1	0
1	1	0	1	0	1	0	0	1
1	1	1	0	1	1	1	1	0
1	1	1	1	0	0	0	0	1

Bảng 4.10. Bảng trạng thái của bộ đếm thập phân đồng bộ

Đồ hình trạng thái của mạch được thể hiện trên hình 4.40.

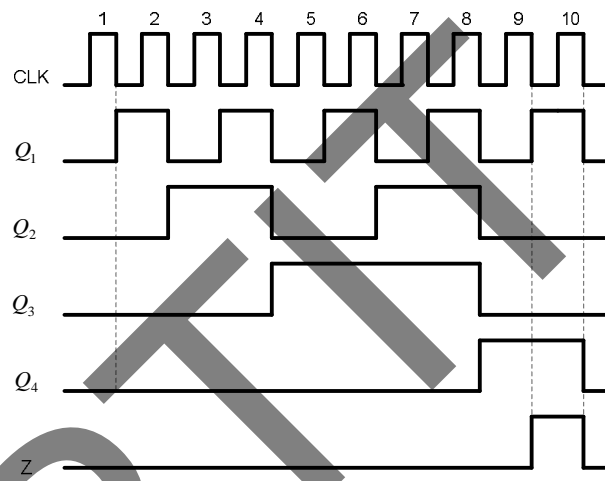
Lưu ý:

Trong bảng trạng thái 4.10 có 6 trạng thái từ 1010 ÷ 1111 là những trạng thái cấm. Nếu vì một lý do nào đó, ví dụ do nhiễu gây ra, thì mạch điện rơi vào trạng thái cấm, dưới tác dụng của xung clock mạch vẫn quay trở về trạng thái được sử dụng thì kết luận mạch có thể tự khởi động.



Hình 4. 40. Đồ hình trạng thái

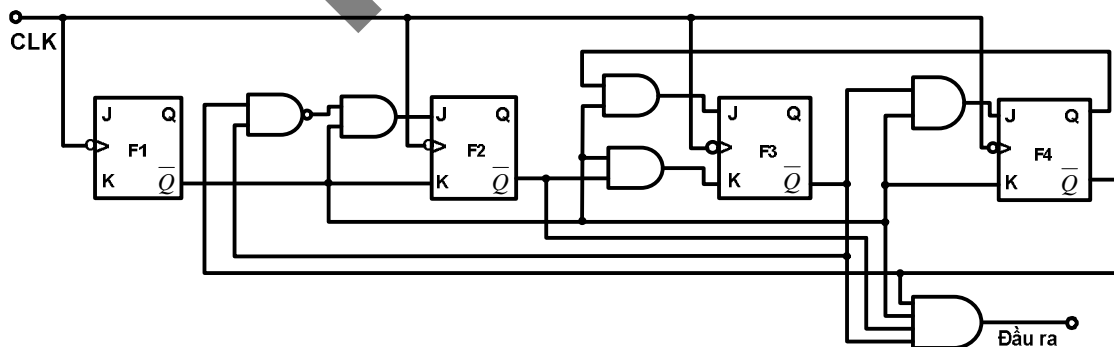
Dạng sóng của bộ đếm thập phân phân đồng bộ được trình bày ở hình 4.41.



Hình 4. 41. Dạng sóng bộ đếm thập phân phân đồng bộ

e. Bộ đếm nghịch thập phân đồng bộ.

Mạch điện được trình bày trên hình 4.42.



Hình 4. 42. Bộ đếm nghịch thập phân đồng bộ

Chúng ta dễ dàng phân tích sơ đồ theo phương pháp tương tự đã dùng ở trên.

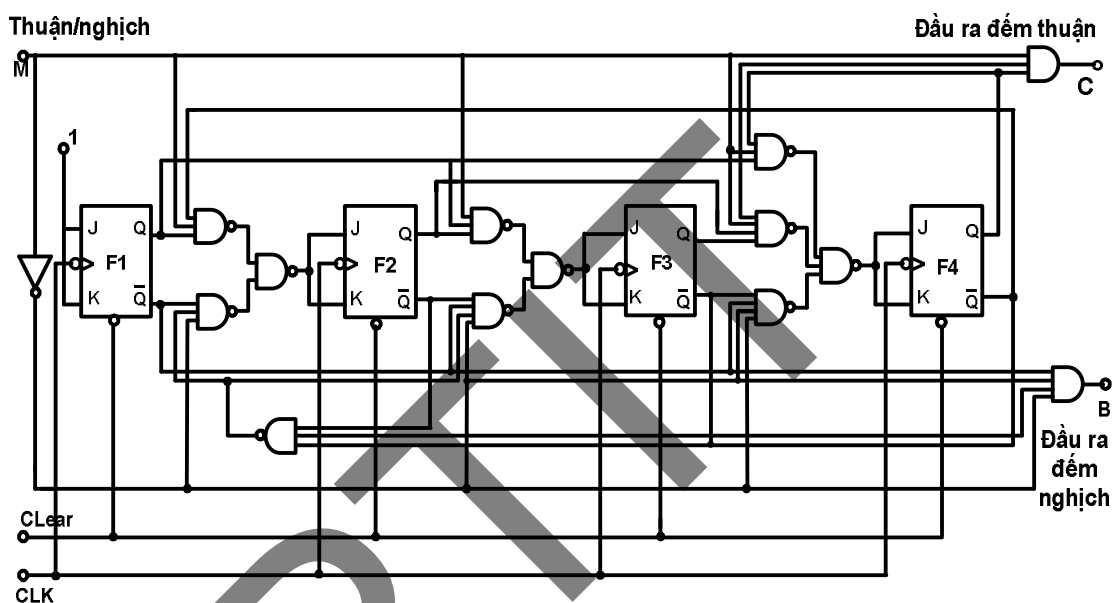
Có hai cách để cấu trúc bộ đếm thập phân đồng bộ nhiều chữ số từ bộ đếm thập phân đồng bộ một chữ số.

Cách thứ nhất là mắc dây chuyển các bộ đếm một chữ số, nối đầu ra của bộ đếm có trọng số bé với đầu vào của bộ đếm có trọng số lớn tiếp theo. Nguyên tắc hoạt động của bộ đếm 1 chữ số là đồng bộ, còn nguyên tắc làm việc của bộ đếm nhiều chữ số là không đồng bộ.

Cách thứ hai nối đầu ra của bộ đếm trọng số bé với tất cả các đầu vào đồng bộ của 4 trigơ của bộ đếm có trọng số lớn tiếp theo và cũng nối đầu vào cổng chuyển vị (cổng nhớ) với đầu ra của bộ đếm này, còn các xung đồng hồ của các bộ đếm đều là xung đếm đầu vào CLK. Trong cách thứ hai nguyên tắc hoạt động của toàn bộ bộ đếm nhiều chữ số cũng là đồng bộ.

f. Bộ đếm thuận nghịch thập phân đồng bộ.

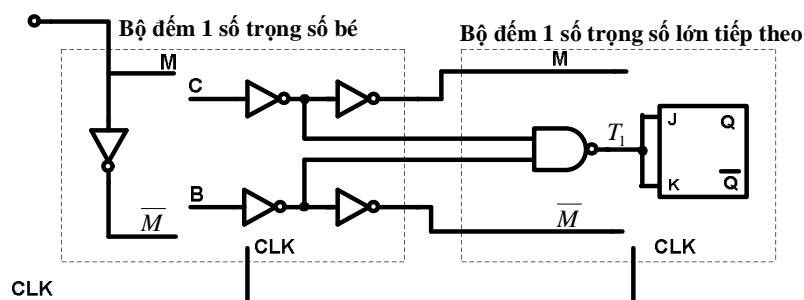
Sơ đồ được trình bày trên hình 4.43.



Hình 4. 43. Bộ đếm thuận nghịch thập phân đồng bộ

Khi tín hiệu điều khiển thuận nghịch $M = 1$, bộ đếm là thuận, $M = 0$ bộ đếm là nghịch.

Phương pháp ghép nối nhiều bộ đếm thuận nghịch thập phân đồng bộ được trình bày trên hình 4.44.



Hình 4. 44. Cách ghép nối nhiều bộ đếm thuận nghịch thập phân đồng bộ

Khi bộ đếm có trọng số bé $C = 1$ thì bộ đếm trọng số lớn $M = 1$, $T_1 = 1$ nó công tác ở chế độ đếm thuận.

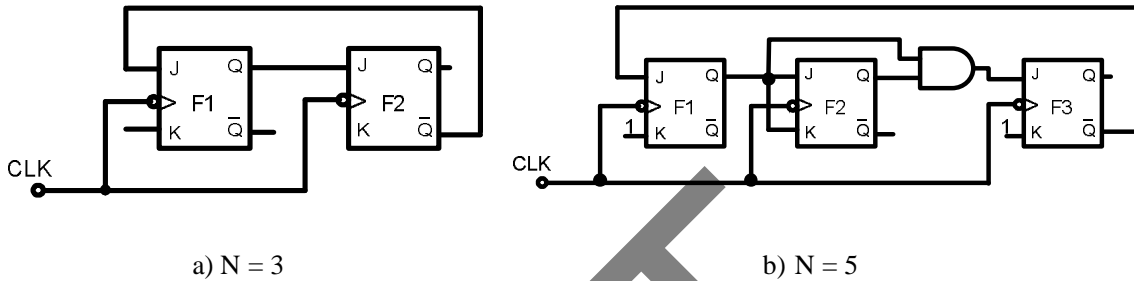
Khi bộ đếm có trọng số bé $B = 1$ thì bộ đếm trọng số lớn $\overline{M} = 1$, $T_1 = 1$ nó sẽ đếm nghịch.

Khi bộ đếm có trọng số bé $C = B = 0$ thì bộ đếm trọng số lớn $M = \overline{M} = T_1 = 0$, nó ngừng đếm.

e. Bộ đếm N phân đồng bộ.

Sử dụng trigơ có thể xây dựng bộ đếm với hệ số đếm N bất kì.

Một số sơ đồ được trình bày trên hình 4.45.



Hình 4. 45. Bộ đếm đồng bộ N phân

4.4.1.3 Bộ đếm không đồng bộ

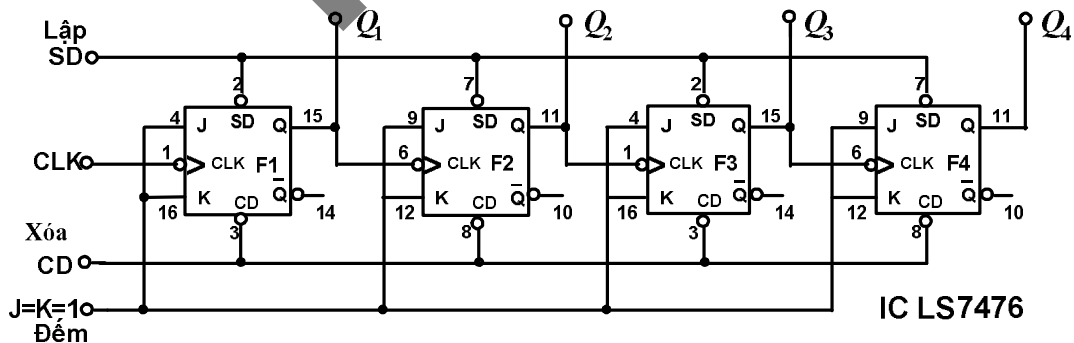
a. Bộ đếm nhị phân không đồng bộ.

Đếm nhị phân không đồng bộ còn được gọi là bộ đếm nối tiếp. Các trigơ được mắc thành trigơ T', mắc nối tiếp với nhau.

Đặc điểm của bộ đếm này là xung nhịp đếm không được đưa đồng thời vào tất cả các trigơ, mà chỉ được đưa vào đầu vào xung nhịp CLK của trigơ đầu tiên, các đầu vào xung nhịp của các trigơ tiếp theo đều được nối với đầu ra của các trigơ trước liền kề.

Các bộ đếm nhị phân không đồng bộ được phân thành: đếm thuận và đếm nghịch.

Sơ đồ bộ đếm thuận nhị phân không đồng bộ 4 bit được trình bày trên hình 4.46.



Hình 4. 46. Sơ đồ bộ đếm thuận nhị phân không đồng bộ

Nguyên lý hoạt động:

Viết các phương trình:

- Phương trình định thời:

$$\text{CLK}_1 = \downarrow \text{CLK}; \text{CLK}_1 = \downarrow Q_1; \text{CLK}_1 = \downarrow Q_2; \text{CLK}_1 = \downarrow Q_3; \quad (4.36)$$

Trigơ T' trong sơ đồ trên sẽ lật trạng thái mỗi khi xuất hiện sườn âm của xung clock.

Phương trình trạng thái:

$$\left. \begin{aligned} Q_1^{n+1} &= \overline{Q_1^n} \quad (\downarrow \text{CLK}) \\ Q_2^{n+1} &= \overline{Q_2^n} \quad (\downarrow Q_1) \\ Q_3^{n+1} &= \overline{Q_3^n} \quad (\downarrow Q_2) \\ Q_4^{n+1} &= \overline{Q_4^n} \quad (\downarrow Q_3) \end{aligned} \right\} \quad (4.37)$$

Tính toán:

Giả thiết trạng thái ban đầu $Q_4^n Q_3^n Q_2^n Q_1^n = 0000$; tuần tự thay vào phương trình trạng thái, lập được bảng 4.11.

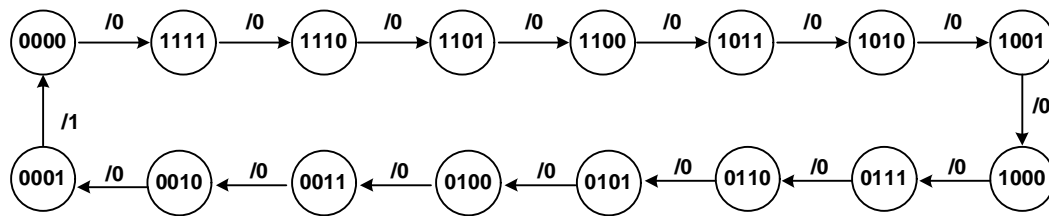
Trigơ ở sơ đồ trên có thể dùng vi mạch 7476 là trigơ JK, có hai đầu vào không đồng bộ: lập và xóa SD và CD đều tác dụng ở mức thấp, muốn xóa CD = 0, SD = 1, muốn đặt SD = 0, CD = 1, để bộ đếm làm việc ở chế độ đếm CD = 1, SD = 1

Để trigơ JK trở thành trigơ T' đầu vào đếm J = K = 1.

Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Điều kiện sườn âm
0	0	0	0	0	0	0	1	CLK ₁
0	0	0	1	0	0	1	0	CLK ₁ , CLK ₂
0	0	1	0	0	0	1	1	CLK ₁
0	0	1	1	0	1	0	0	CLK ₁ , CLK ₂ , CLK ₃
0	1	0	0	0	1	0	1	CLK ₁
0	1	0	1	0	1	1	0	CLK ₁ , CLK ₂
0	1	1	0	0	1	1	1	CLK ₁
0	1	1	1	1	0	0	0	CLK ₁ , CLK ₂ , CLK ₃ , CLK ₄
1	0	0	0	1	0	0	1	CLK ₁
1	0	0	1	1	0	1	0	CLK ₁ , CLK ₂
1	0	1	0	1	0	1	1	CLK ₁
1	0	1	1	1	1	0	0	CLK ₁ , CLK ₂ , CLK ₃
1	1	0	0	1	1	0	1	CLK ₁
1	1	0	1	1	1	1	0	CLK ₁ , CLK ₂
1	1	1	0	1	1	1	1	CLK ₁
1	1	1	1	0	0	0	0	CLK ₁ , CLK ₂ , CLK ₃ , CLK ₄

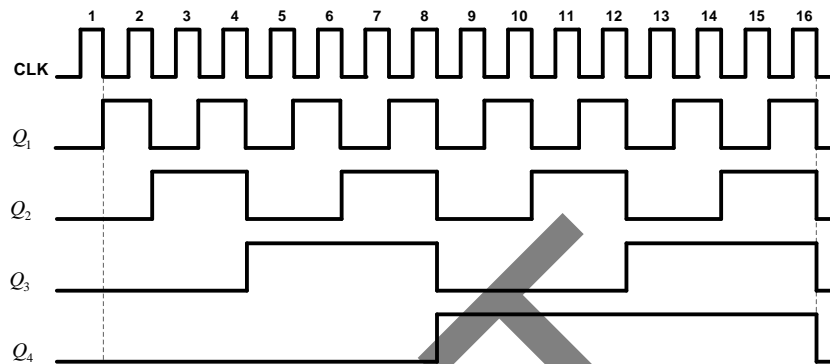
Bảng 4.11. Bảng trạng thái của bộ đếm nhị phân không đồng bộ

Hình 4.47 là đồ hình trạng thái của bộ đếm.



Hình 4. 47. Đồ hình trạng thái của bộ đếm nhị phân

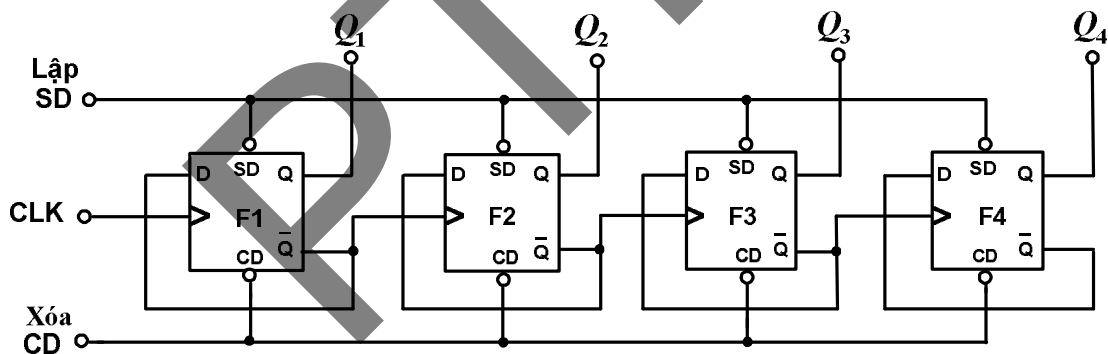
Dạng sóng của bộ đếm được trình bày ở hình 4.48.



Hình 4. 48. Dạng sóng bộ đếm thuận nhị phân không đồng bộ

b. Bộ đếm thuận nhị phân không đồng bộ kích bằng sườn dương.

Bộ đếm thuận nhị phân không đồng bộ kích bằng sườn dương (hình 4.49).

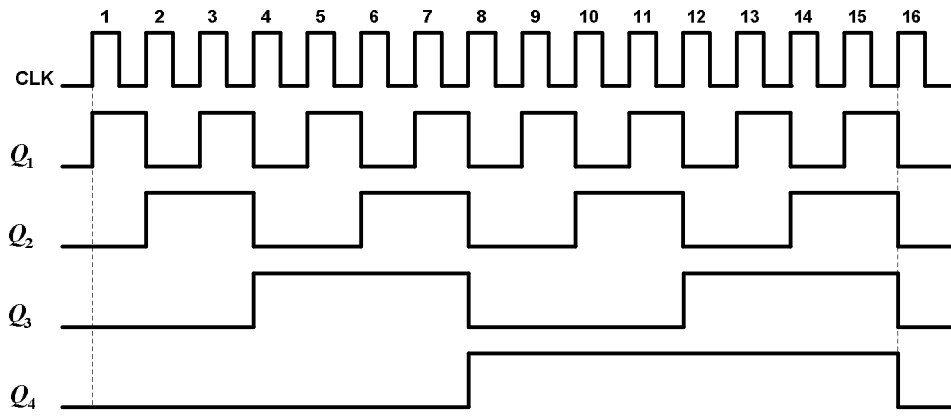


Hình 4. 49. Sơ đồ đếm thuận nhị phân không đồng bộ kích bằng sườn dương

Trong sơ đồ này, dùng 4 trigơ D kích bằng sườn dương được mắc thành trigơ T', đầu vào của các trigơ trừ trigơ đầu tiên được nối với đầu ra đảo \overline{Q} của trigơ có trọng số bé hơn liền kề.

Nguyên tắc hoạt động của sơ đồ này cũng giống như sơ đồ dùng trigơ JK kích bằng sườn âm, cần chú ý điều kiện định thời kích bằng sườn dương.

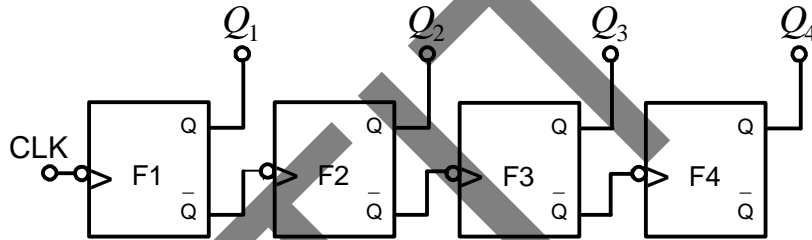
Dạng sóng của bộ đếm kiểu này được trình bày ở hình 4.50, chú ý là sườn xuống của Q là sườn lên của \overline{Q} .



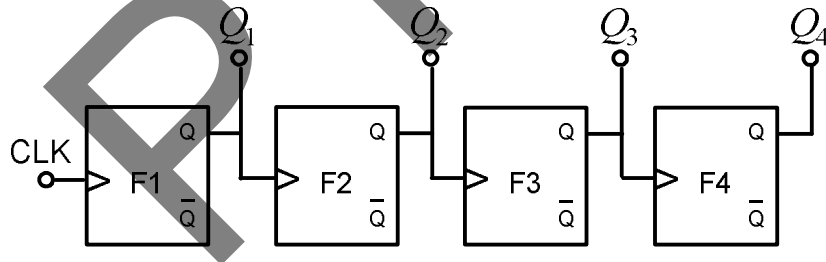
Hình 4. 50. Dạng sóng của bộ đếm thuận nhị phân không đồng bộ kích bằng sườn dương

c. Bộ đếm nghịch nhị phân không đồng bộ.

Sơ đồ bộ đếm này được trình bày ở hình 4.51 và 4.52 sử dụng các trigơ mắc thành trigơ T'.



Hình 4. 51. Bộ đếm nghịch nhị phân không đồng bộ kích bằng sườn âm.



Hình 4. 52. Bộ đếm nghịch nhị phân không đồng bộ kích bằng sườn dương.

Nguyên lý hoạt động của bộ đếm nghịch nhị phân không đồng bộ kích bằng sườn âm:

Viết các phương trình:

- Phương trình định thời:

$$CLK_1 = \downarrow CLK; CLK_1 = \downarrow \overline{Q_1}; CLK_1 = \downarrow \overline{Q_2}; CLK_1 = \downarrow \overline{Q_3}; \quad (4.38)$$

Trigơ T' trong sơ đồ trên sẽ lật trạng thái mỗi khi xuất hiện sườn âm của xung clock.

Phương trình chuyển đổi trạng thái:

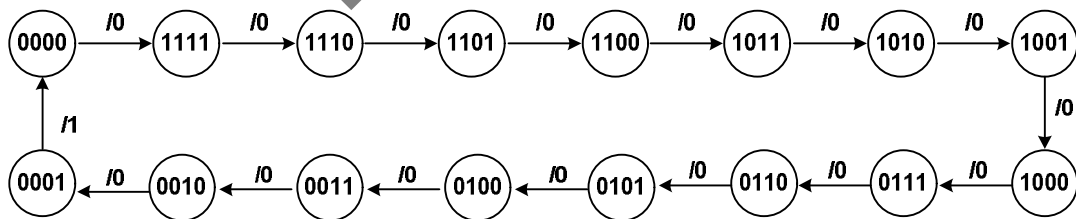
$$\left. \begin{aligned} Q_1^{n+1} &= \overline{Q_1^n} \ (\downarrow \text{CLK}) \\ Q_2^{n+1} &= \overline{Q_2^n} \ (\downarrow \overline{Q_1}) \\ Q_3^{n+1} &= \overline{Q_3^n} \ (\downarrow \overline{Q_2}) \\ Q_4^{n+1} &= \overline{Q_4^n} \ (\downarrow \overline{Q_3}) \end{aligned} \right\} \quad (4.39)$$

Giả thiết trạng thái ban đầu $Q_4^n Q_3^n Q_2^n Q_1^n = 0000$; tuần tự thay vào phương trình trạng thái, tìm được bảng 4.12.

Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	0	1	1	0	1
1	1	0	1	1	1	0	0
1	1	0	0	1	0	1	1
1	0	1	1	1	0	1	0
1	0	1	0	1	0	0	1
1	0	0	1	1	0	0	0
1	0	0	0	0	1	1	1
0	1	1	1	0	1	1	0
0	1	1	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0
0	0	0	0	1	1	1	1

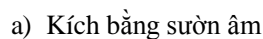
Bảng 4.12. Bảng trạng thái của bộ đếm nghịch

Hình 4.53 là đồ hình trạng thái của bộ đếm.



Hình 4. 53. Đồ hình trạng thái của bộ đếm nhị phân

Dạng sóng của bộ đếm nghịch nhị phân không đồng bộ trình bày ở hình 4.54.



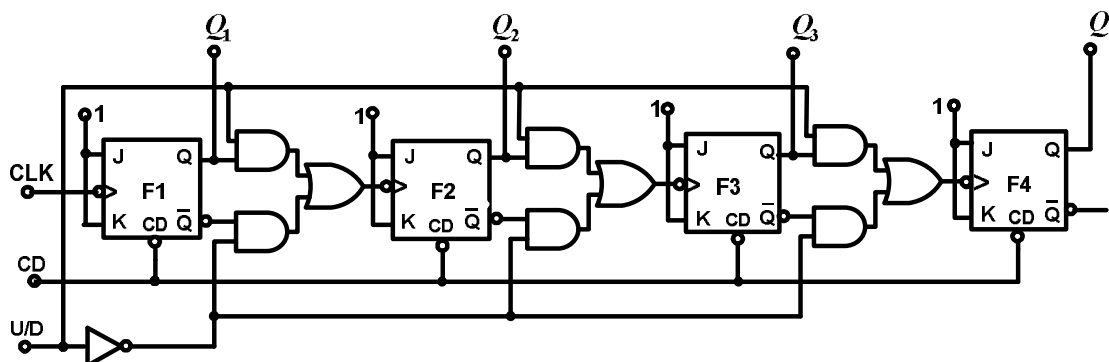
Hình 4. 54. Dạng sóng của bộ đếm nghịch nhị phân không đồng bộ

d. Bộ đếm thuận nghịch nhị phân không đồng bộ.

Sơ đồ được trình bày ở hình 4.55.

Thực chất đây là mạch ghép của bộ đếm thuận và nghịch nhị phân không đồng bộ. Sơ đồ có thêm đầu vào điều khiển UP/ Down và một số cổng để thực hiện đếm thuận hoặc đếm nghịch.

Đếm thuận: Khi đầu vào UP/ Down = 1, đầu ra Q của trigơ trước nối với đầu vào CLK của trigơ tiếp sau, mạch giống như sơ đồ hình 4.55. Đếm nghịch: Khi đầu vào UP/ Down = 0, đầu ra \overline{Q} của trigơ trước nối với đầu vào CLK của trigơ tiếp sau.

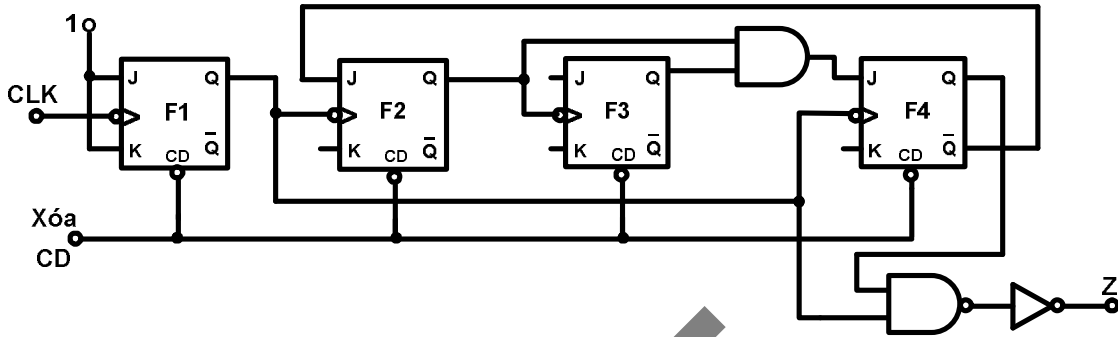


Hình 4. 55. Bộ đếm thuận nghịch nhị phân không đồng bộ

Đặc điểm của bộ đếm nhị phân không đồng bộ là cách ghép nối bộ đếm và cấu trúc bộ đếm đơn giản, nhưng có nhược điểm là tần số công tác thấp, xung nhiễu quá độ từ trạng thái mã hóa này sang trạng thái mã hóa khác lớn.

e. Bộ đếm thuận thập phân không đồng bộ.

Sơ đồ được trình bày ở hình 4.56.



Hình 4. 56. Bộ đếm thuận thập phân không đồng bộ

Sơ đồ gồm 4 trigơ JK, các cổng NAND, AND và NOT, CLK là xung đếm đầu vào, Z là tín hiệu chuyển vị (nhớ) đưa đến bộ đếm trọng số lớn hơn tiếp theo.

Nguyên lý hoạt động:

Phương trình định thời:

$$\left. \begin{aligned} C_1 &= \downarrow \text{CLK}; \\ C_2 &= C_4 = \downarrow Q_1 \\ C_3 &= \downarrow Q_2 \end{aligned} \right\} \quad (4.40)$$

Phương trình đầu ra:

$$Z = Q_4^n \cdot Q_1^n \quad (4.41)$$

Phương trình kích:

$$\left. \begin{aligned} J_1 &= K_1 = 1 \\ J_2 &= \overline{Q_4^n}; \quad K_2 = 1 \\ J_3 &= K_3 = 1 \\ J_4 &= Q_3^n \cdot Q_2^n; \quad K_4 = 1 \end{aligned} \right\} \quad (4.42)$$

Theo qui ước đầu vào để trống là nối với mức logic 1.

Tìm phương trình chuyển đổi trạng thái:

Thay các giá trị (4.43) vào phương trình đặc trưng của trigơ JK tìm được:

$$\left. \begin{aligned}
 Q_1^{n+1} &= J_1 \cdot \overline{Q_1^n} + \overline{K_1} \cdot Q_1^n = \overline{Q_1^n} & (\downarrow \text{CLK}) \\
 Q_2^{n+1} &= J_2 \cdot \overline{Q_2^n} + \overline{K_2} \cdot Q_2^n = \overline{Q_2^n} \cdot \overline{Q_4^n} & (\downarrow Q_1) \\
 Q_3^{n+1} &= J_3 \cdot \overline{Q_3^n} + \overline{K_3} \cdot Q_3^n = \overline{Q_3^n} & (\downarrow Q_2) \\
 Q_4^{n+1} &= J_4 \cdot \overline{Q_4^n} + \overline{K_4} \cdot Q_4^n = \overline{Q_2^n} \cdot \overline{Q_3^n} \cdot \overline{Q_1^n} & (\downarrow Q_1)
 \end{aligned} \right\} \quad (4.43)$$

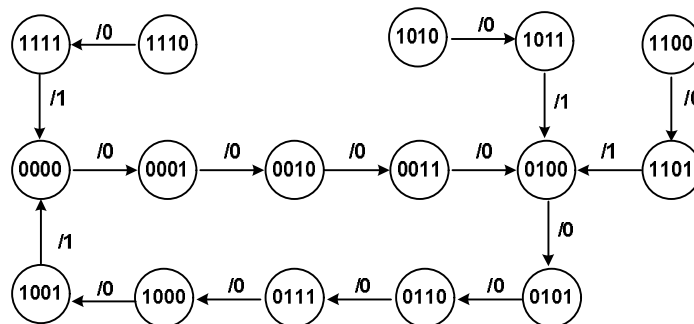
Lập bảng trạng thái:

Khi xung xóa (CD = 0) tất cả các đầu ra $Q_4^n Q_3^n Q_2^n Q_1^n = 0000$. Thay vào phương trình chuyển đổi trạng thái tìm được bảng 4.13.

Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Z	Điều kiện sườn âm
0	0	0	0	0	0	0	1	0	CLK ₁
0	0	0	1	0	0	1	0	0	CLK ₁ ; CLK ₂
0	0	1	0	0	0	1	1	0	CLK ₁
0	0	1	1	0	1	0	0	0	CLK ₁ ; CLK ₂ ; CLK ₃
0	1	0	0	0	1	0	1	0	CLK ₁
0	1	0	1	0	1	1	0	0	CLK ₁ ; CLK ₂
0	1	1	0	0	1	1	1	0	CLK ₁
0	1	1	1	1	0	0	0	0	CLK ₁ ; CLK ₂ ; CLK ₃ ; CLK ₄
1	0	0	0	1	0	0	1	0	CLK ₁
1	0	0	1	0	0	0	0	1	CLK ₁ ; CLK ₂
1	0	1	0	1	0	1	1	0	CLK ₁
1	0	1	1	0	1	0	0	1	CLK ₁ ; CLK ₂ ; CLK ₃ ; CLK ₄
1	1	0	0	1	1	0	1	0	CLK ₁
1	1	0	1	0	1	0	0	1	CLK ₁ ; CLK ₂ ; CLK ₄
1	1	1	0	1	1	1	1	0	CLK ₁
1	1	1	1	0	0	0	0	1	CLK ₁ ; CLK ₂ ; CLK ₃ ; CLK ₄

Bảng 4.13. Bảng trạng thái của bộ đếm thập phân không đồng bộ

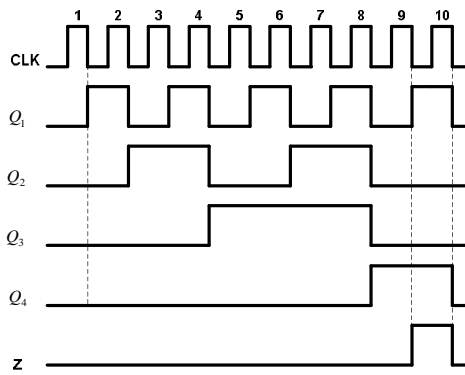
Đồ hình trạng thái của mạch được thể hiện trên hình 4.57.



Hình 4. 57. Đồ hình trạng thái

Căn cứ vào đồ hình trạng thái kết luận bộ đếm này có khả năng tự khởi động.

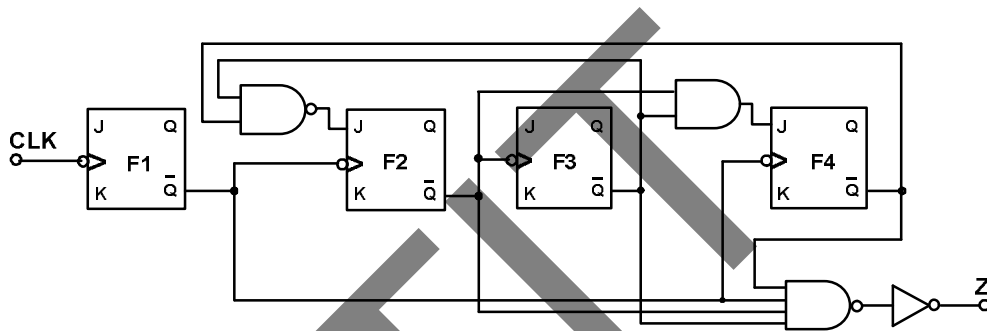
Dạng sóng của bộ đếm thuận thập phân đồng bộ được trình bày ở hình 4.58.



Hình 4. 58. Dạng sóng bộ đếm thuận thập phân không đồng bộ

f. Bộ đếm nghịch thập phân không đồng bộ.

Sơ đồ trình bày trên hình 4.59.



Hình 4. 59. Bộ đếm nghịch thập phân không đồng bộ

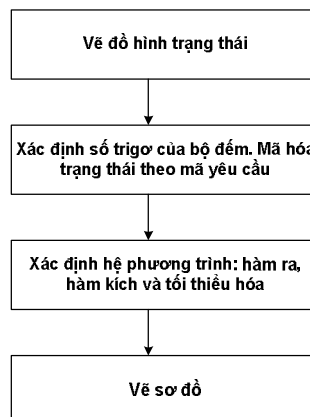
Cách phân tích cũng tương tự như ở mục trên.

Cách ghép nối các bộ đếm thập phân không đồng bộ 1 chữ số thành bộ đếm nhiều chữ số là ghép nối liên tiếp, đầu ra của bộ đếm có trọng số bé được nối với đầu vào CLK của bộ đếm có trọng số lớn kế tiếp.

4.4.2. Thiết kế bộ đếm

4.4.2.1. Các bước thiết kế bộ đếm

Hình 4.60 là lưu đồ thiết kế bộ đếm.



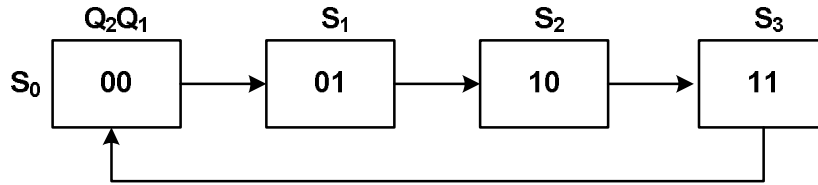
Hình 4. 60. Lưu đồ thiết kế bộ đếm

4.4.2.2. Bộ đếm đồng bộ.

a. Bộ đếm nhị phân.

Ví dụ: Thiết kế bộ đếm nhị phân đồng bộ có $M_d = 4$.

Do $M_d = 4$ nên lập được đồ hình trạng thái ở hình 4.61.



Hình 4. 61. Đồ hình trạng thái của bộ đếm

Từ đồ hình trạng thái, xác định được số trigơ cần dùng để thiết kế bộ đếm ($n = 2$) và mã hoá các trạng thái đó. Hai trigơ cần để mã hoá các thái là Q_1 và Q_2 . Dùng bảng hàm kích 4.14 để xác định các đầu vào kích cho các loại trigơ.

N		n + 1		Trigơ Q_2					Trigơ Q_1				
Q_2	Q_1	Q_2^k	Q_1^k	R_2	S_2	J_2	K_2	T_2	R_1	S_1	J_1	K_1	T_1
0	0	0	1	X	0	0	X	0	0	1	1	X	1
0	1	1	0	0	1	1	X	1	1	0	X	1	1
1	0	1	1	0	X	X	0	0	0	1	1	X	1
1	1	0	0	1	0	X	1	1	1	0	X	1	1

Bảng 4.14. Bảng hàm kích

Tối thiểu hoá hàm kích của các trigơ, nhận được kết quả:

Đối với trigơ Q_1 :

$$R_1 = \overline{Q_2} Q_1 + Q_2 Q_1 = Q_1 (\overline{Q_2} + Q_2) = Q_1 ; \quad S_1 = \overline{Q_2} \overline{Q_1} + Q_2 \overline{Q_1} = \overline{Q_1} (\overline{Q_2} + Q_2) = \overline{Q_1}$$

$$J_1 = K_1 = 1;$$

$$T_1 = 1;$$

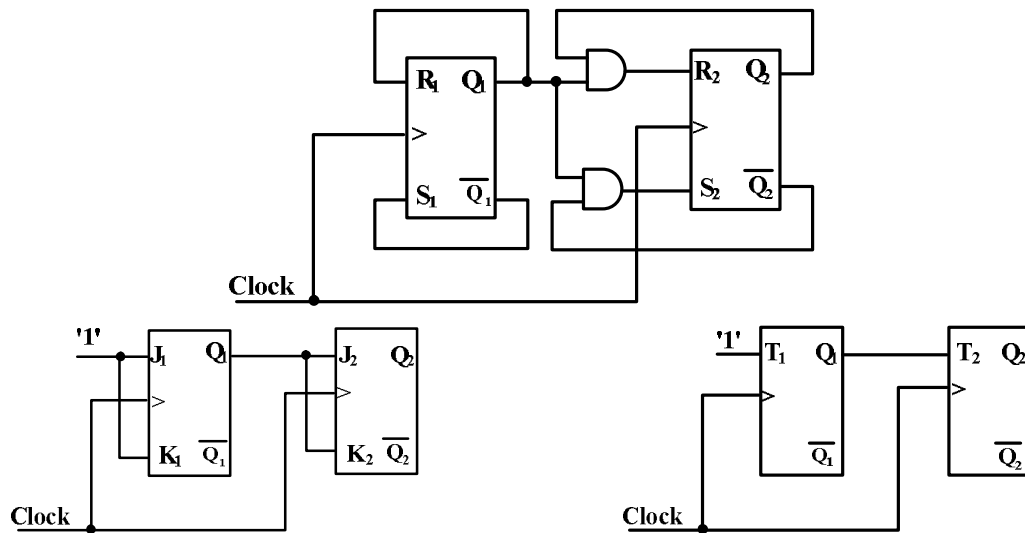
Đối với trigơ Q_2 :

$$R_2 = Q_2 Q_1; \quad S_2 = \overline{Q_2} Q_1$$

$$J_2 = K_2 = Q_1;$$

$$T_2 = Q_1;$$

Sơ đồ mạch điện như ở hình 4.62.

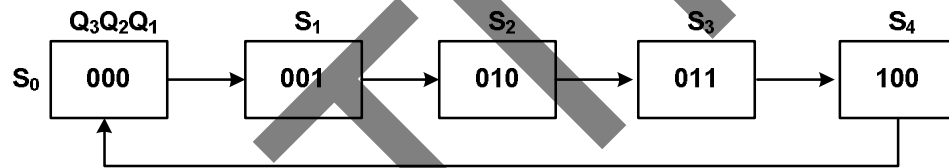


Hình 4. 62. Bộ đếm Mod 4 dùng trigơ RS; JK; T

b. Bộ đếm N phân (hay bộ đếm có mod đếm bất kỳ).

Ví dụ: Thiết kế bộ đếm đồng bộ có $M_d = 5$.

Do $M_d = 5$ nên lập được đồ hình trạng thái như hình 4.63.



Hình 4. 63. Đồ hình trạng thái của bộ đếm Mod 5

Từ đó xác định được số trigơ cần dùng để thiết kế bộ đếm ($n = 3$) và mã hoá các trạng thái đó. Ba trigơ cần để mã hoá các thái là Q_3 , Q_2 và Q_1 .

Cách 1: Dùng bảng hàm kích 4.15 để xác định các đầu vào kích cho các trigơ.

n			n + 1			Trigơ Q_3		Trigơ Q_2		Trigơ Q_1	
Q_3	Q_2	Q_1	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	J_3	K_3	J_2	K_2	J_1	K_1
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

Bảng 4.15. Bảng hàm kích cho trigơ

Tối thiểu hoá hàm kích của các trigơ, nhận được kết quả:

$$J_1 = \overline{Q_3} ; \quad K_1 = 1;$$

$$J_2 = K_2 = Q_2;$$

$$J_3 = Q_2 Q_1 ; \quad K_3 = 1;$$

Cách 2: Dùng bảng chuyển đổi trạng thái để tìm hàm kích:

Q_2Q_1 Q_3		00	01	11	10
		0	1	1	0
0	001	010	100	011	
1	000	x	x	x	

Bảng 4.16. Bảng chuyển đổi trạng thái.

Từ bảng 4.16, tách ra làm 3 bảng chuyển đổi riêng biệt cho 3 trigơ để tìm phương trình chuyển đổi trạng thái của các Q_3^{n+1} , Q_2^{n+1} , Q_1^{n+1} .

Q_2Q_1 Q_3		00	01	11	10
		0	1	1	0
0	1	0	0	1	
1	0	x	x	x	

Bảng 4.17. Bảng Các nô xác định Q_1^{n+1}

Q_2Q_1 Q_3		00	01	11	10
		0	1	1	0
0	0	1	0	1	
1	0	x	x	x	

Bảng 4.18. Bảng Các nô xác định Q_2^{n+1}

Từ bảng 4.17 tìm được Q_1^{n+1} , từ Q_1^{n+1} tìm được J_1, K_1 :

$$Q_1^{n+1} = \overline{Q_3} \cdot \overline{Q_1}$$

$$Q_1^{n+1} = J_1 \cdot \overline{Q_1} + \overline{K_1} \cdot Q_1$$

$$\rightarrow J_1 = \overline{Q_3}; K_1 = 1$$

Từ bảng 4.18 tìm được Q_2^{n+1} , J_2, K_2

$$Q_2^{n+1} = \overline{Q_2} \cdot Q_1 + Q_2 \cdot \overline{Q_1}$$

$$Q_2^{n+1} = J_2 \cdot \overline{Q_2} + \overline{K_2} \cdot Q_2$$

$$\rightarrow J_2 = K_2 = Q_1$$

Q_2Q_1 Q_3		00	01	11	10
		0	1	1	0
0	0	0	1	0	
1	0	x	x	x	

Bảng 4.19. Bảng Các nô xác định Q_3^{n+1}

Từ bảng 4.19 tìm được Q_3^{n+1} , J_3, K_3

$$Q_3^{n+1} = \overline{Q_3} \cdot Q_2 \cdot Q_1$$

$$Q_i^{n+1} = J_i \cdot \overline{Q_i} + \overline{K_i} \cdot Q_i$$

$$\rightarrow J_3 = Q_2 \cdot Q_1; K_3 = 1$$

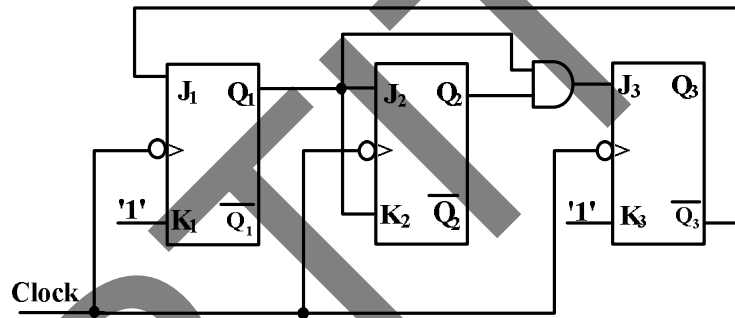
Kiểm tra khả năng tự khởi động bằng bảng 4.20.

n			n + 1		
Q ₃	Q ₂	Q ₁	Q ₃ ⁿ⁺¹	Q ₂ ⁿ⁺¹	Q ₁ ⁿ⁺¹
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	0

Bảng 4.20. Kiểm tra khả năng tự khởi động

Nhìn vào bảng trạng thái 4.20, trạng thái dư sau một số xung nhịp đều quay trở lại vòng đếm nên kết luận rằng bộ đếm này tự khởi động.

Sơ đồ mạch điện ở hình 4.64.



Hình 4. 64. Bộ đếm đồng bộ Mod 5

4.4.2.3. Bộ đếm không đồng bộ.

a. Bộ đếm nhị phân

Các bộ đếm này có sơ đồ rất đơn giản với đặc điểm:

- Chỉ dùng một loại trigơ T hoặc JK. Nếu dùng trigơ T thì đầu vào T luôn được nối với mức logic '1', nếu dùng trigơ JK thì J và K được nối với nhau và nối với mức '1'.

- Đầu ra của trigơ trước được nối với đầu vào xung nhịp của trigơ sau kế tiếp. Khi đếm tiến thì lấy ở đầu ra Q, khi đếm lùi thì lấy ở đầu ra \overline{Q} (với giả thiết xung Clock tích cực tại sườn âm ↓).

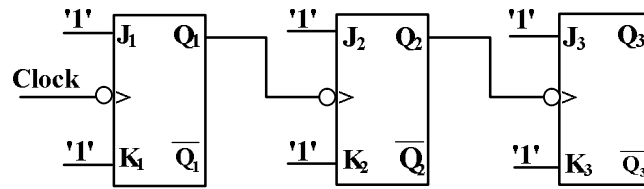
- Tín hiệu vào X_d luôn được đưa tới đầu vào xung nhịp của trigơ có trọng số nhỏ nhất.

Ví dụ đối với bộ đếm nhị phân không đồng bộ M_d = 2ⁿ dùng các trigơ Q₁, Q₂ ... Q_{n-1} với Q₁ là bit có trọng số nhỏ nhất, Q_{n-1} là bit có trọng số lớn nhất:

- Khi đếm tiến: C_{Q1} = X; C_{Q2} = Q₁ ... C_{Qn-1} = C_{Qn-2}.

- Khi đếm lùi: C_{Q1} = X; C_{Q2} = $\overline{Q_1}$... C_{Qn-1} = C _{$\overline{Q_{n-2}}$}

Sơ đồ của bộ đếm nhị phân không đồng bộ 3 bit ($M_d = 8$ - đếm tiến) dùng trigơ JK được cho ở hình 4.65.

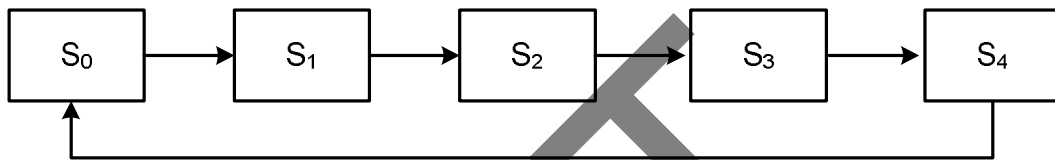


Hình 4. 65. Bộ đếm nhị phân không đồng bộ 3 bit

b. Bộ đếm N phân (hay bộ đếm có mod đếm bất kỳ).

Ví dụ: Thiết kế bộ đếm $M_d = 5$ không đồng bộ.

Từ yêu cầu bài toán, xây dựng đồ hình trạng thái như ở hình 4.66.



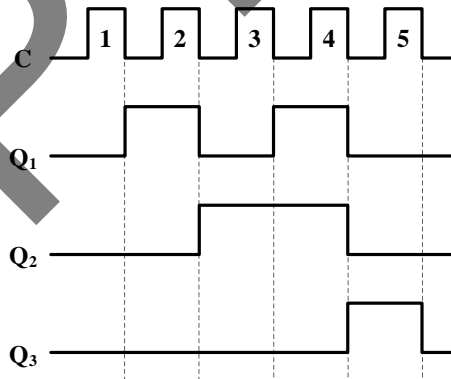
Hình 4. 66. Đồ hình trạng thái bộ đếm mod 5 không đồng bộ

- Chọn lựa mã hoá trạng thái

Có 5 trạng thái nên số trigơ bằng 3, chọn trigơ JK. Chọn mã BCD8421.

$S_0 = 000$; $S_1 = 001$; $S_2 = 010$; $S_3 = 011$; $S_4 = 100$.

- Chọn xung đồng hồ từ giản đồ xung 4.67.



Hình 4. 67. Giản đồ xung của bộ đếm mod 5

$C_1 = \downarrow C$; $C_2 = \downarrow Q_1$; $C_3 = \downarrow C$;

- Tìm hệ phương trình:

Khi xét trạng thái kế tiếp của các trigơ, ngoài những trạng thái không được dùng để mã hóa (từ $S_5 \div S_7 = 101 \div 111$) còn có các trạng thái không thỏa mãn điều kiện kích của xung Clock (nghĩa là không đúng sườn xung Clock hoặc chưa xuất hiện sườn xung Clock), các trạng thái này là tùy chọn và đều được dùng để tối thiểu hóa.

Q_2Q_1 Q_3		00	01	11	10
		0	1	0	1
	0	001	010	100	011
	1	000	x	x	x

Q_2Q_1 Q_3		00	01	11	10
		0	1	0	1
	0	1	0	0	1
	1	0	x	x	x

$$Q_1^{n+1} = \overline{Q_3} \cdot \overline{Q_1}$$

Q_2Q_1 Q_3		00	01	11	10
		0	1	0	1
	0	x	1	0	x
	1	x	x	x	x

$$Q_2^{n+1} = \overline{Q_2}$$

Q_2Q_1 Q_3		00	01	11	10
		0	1	0	1
	0	0	0	1	0
	1	0	x	x	x

$$Q_3^{n+1} = \overline{Q_3} \cdot Q_2 \cdot Q_1$$

Bảng 4.21. Bảng chuyển đổi trạng thái và bảng tìm hàm kích

Sau khi tối thiểu hoá bằng bảng 4.21, nhận được hệ phương trình:

$$Q_1^{n+1} = \overline{Q_3} \cdot \overline{Q_1}$$

$$Q_2^{n+1} = \overline{Q_2}$$

$$Q_3^{n+1} = \overline{Q_3} \cdot Q_2 \cdot Q_1$$

Kiểm tra khả năng tự khởi động bằng bảng 4.22.

n			n + 1		
Q_3	Q_2	Q_1	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	0

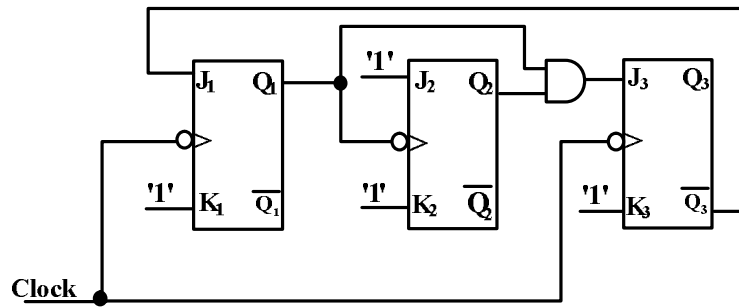
Bảng 4.22. Kiểm tra khả năng tự khởi động

Nhìn vào bảng 4.22, các trạng thái dư sau 1 số xung nhịp đều quay trở lại vòng đếm, do vậy bộ đếm này tự khởi động.

Từ hệ phương trình trạng thái, tìm được phương trình hàm kích bằng cách đồng nhất chúng với phương trình đặc trưng: $Q_i^{n+1} = J_i \overline{Q_i} + K_i Q_i$;

$$J_1 = \overline{Q_3} ; K_1 = 1 ; J_2 = K_2 = 1 ; J_3 = Q_1 Q_0 ; K_3 = 1 ;$$

Từ hệ phương trình hàm kích, vẽ được mạch điện của bộ đếm Mod 5 không đồng bộ như hình 4.68.



Hình 4. 68. Bộ đếm Mod 5 không đồng bộ

4.4.3. Giới thiệu một số IC đếm

4.4.3.1. IC đếm đồng bộ 54/74

Việc thiết kế các bộ đếm đồng bộ bằng cách sử dụng các trigơ đã được thảo luận trên đây. Các bộ đếm dành cho bất kỳ đồ hình trạng thái nào và bất kỳ trigơ nào đều có thể được thiết kế bằng các phương pháp này. Một vài bộ đếm (MSI) có sẵn và được cho trong bảng 4.23. Tất cả các IC này đều hoạt động tại sườn dương của xung Clock.

Tên IC	Mô tả	Đặc tính
74160	Bộ đếm thuận (UP) thập phân	Preset đồng bộ và Clear không đồng bộ
74161	Bộ đếm thuận (UP) nhị phân 4 bit	Preset đồng bộ và Clear không đồng bộ
74162	Bộ đếm thuận (UP) thập phân	Preset đồng bộ và Clear đồng bộ
74163	Bộ đếm thuận (UP) nhị phân 4 bit	Preset đồng bộ và Clear đồng bộ
74168	Bộ đếm thuận nghịch (UP/DOWN) thập phân	Preset đồng bộ và không Clear
74169	Bộ đếm thuận nghịch (UP/DOWN) nhị phân 4 bit	Preset đồng bộ và không Clear
74190	Bộ đếm thuận nghịch (UP/DOWN) thập phân	Preset đồng bộ và không Clear
74191	Bộ đếm thuận nghịch (UP/DOWN) nhị phân 4 bit	Preset đồng bộ và không Clear
74192	Bộ đếm thuận nghịch (UP/DOWN) thập phân	Preset đồng bộ và Clear
74193	Bộ đếm thuận nghịch (UP/DOWN) nhị phân 4 bit	Preset đồng bộ và Clear

Bảng 4.23. Một số loại IC thông dụng

4.4.3.2. IC đếm không đồng bộ 54/74

Tên IC	Mô tả	Đặc tính
7490	Bộ đếm BCD	Có đầu vào lập (Set) và xóa (Reset)
74290	Bộ đếm BCD	Có đầu vào xóa (Reset)
7492	Bộ đếm chia chẵn cho 12	Có đầu vào xóa (Reset)
7493	Bộ đếm nhị phân 4 bit	Có đầu vào xóa (Reset)
74293	Bộ đếm nhị phân 4 bit	Có đầu vào xóa (Reset)
74176	Bộ đếm BCD	Có đầu vào xóa (Reset); Có các đầu vào nạp dữ liệu
74196	Bộ đếm BCD	Có đầu vào xóa (Reset); Có các đầu vào nạp dữ liệu
74177	Bộ đếm nhị phân	Có đầu vào xóa (Reset); Có các đầu vào nạp dữ liệu
74197	Bộ đếm nhị phân	Có đầu vào xóa (Reset); Có các đầu vào nạp dữ liệu

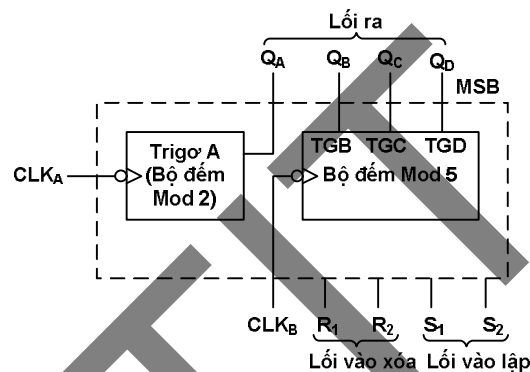
Tên IC	Mô tả	Đặc tính
74390	Hai bộ đếm thập phân	Có đầu vào xóa (Reset)
74393	Hai bộ đếm nhị phân 4 bit	Có đầu vào xóa (Reset);
74490	Hai bộ đếm BCD	Có đầu vào lập (Set) và xóa (Reset)

Bảng 4.24. Một số loại IC thông dụng

Tất cả các IC giới thiệu trong bảng 4.24 đều hoạt động tại sườn âm của xung Clock. Các loại IC này đều có bốn trigơ MS.

a. IC 7490, 74390 và IC 74490.

Hình 4.69 biểu thị cấu trúc cơ bản của IC 7490. Nó bao gồm 4 trigơ cung cấp bộ đếm gồm hai Mod đếm: Mod 2 và Mod 4. Các bộ đếm Mod 2 và Mod 5 có thể được sử dụng một cách độc lập. Trigơ A thực hiện đếm Mod 2, Trigơ B, C, D thực hiện đếm Mod 4.



Hình 4. 69. Cấu trúc cơ bản của IC 7490

Hai đầu vào xóa (Reset) là R_1 và R_2 , chúng hoạt động ở mức tích cực cao.

Hai đầu vào lập (Set) là S_1 và S_2 . Khi chúng ở trạng thái 1 thì bộ đếm được nạp giá trị 1001_2 .

IC 74390 gồm có hai bộ đếm BCD độc lập tương tự như 7490, có một đầu vào xóa (Reset) cho từng bộ đếm.

IC 74490 gồm có hai bộ đếm BCD độc lập. Mỗi phần có chứa 4 trigơ, mỗi phần đều có một đầu vào lập (S) và một đầu vào xóa (R) hoạt động ở mức tích cực cao.

Bảng 4.25 mô tả chức năng của IC 7490.

R_1	R_2	S_1	S_2	Q_D	Q_C	Q_B	Q_A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
L	X	L	X	Thực hiện đếm			
X	L	X	L	Thực hiện đếm			
L	X	X	L	Thực hiện đếm			
X	L	L	X	Thực hiện đếm			

Bảng 4.25. Bảng chức năng của IC7490

Bảng 4.26 mô tả chức năng của IC 74290.

$R_{0(1)}$	$R_{0(2)}$	$R_{9(1)}$	$R_{9(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
L	X	L	X	Thực hiện đếm			
X	L	X	L	Thực hiện đếm			
L	X	X	L	Thực hiện đếm			
X	L	L	X	Thực hiện đếm			

Bảng 4.26. Bảng chức năng của IC74290

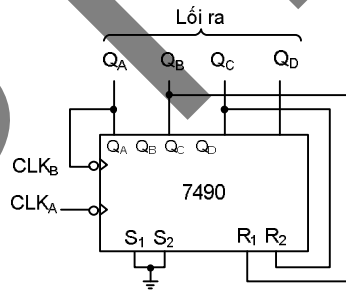
Để thực hiện đếm Mod 10 (BCD) thì đầu ra Q_A được nối với xung Clock B. Tín hiệu cần đếm được đưa vào xung clock A.

Ngoài ra bộ đếm này cũng có thể tạo ra các Mod khác nhau (nhỏ hơn 10) bằng phương pháp hồi tiếp về chân Reset.

Ví dụ: Thiết kế bộ đếm chia hết cho 6 bằng IC 7490.

Trước hết, chuyển 6_{10} sang nhị phân để tìm đầu ra cần xóa: 0110_2 . Như vậy, đầu ra cần xóa là Q_B và Q_C .

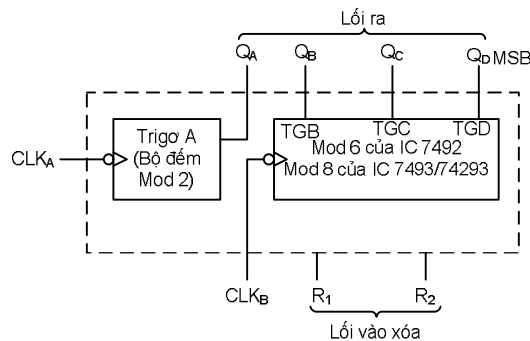
Để thực hiện được mạch chia này, phải nối bộ đếm về trạng thái chia hết cho 1010, sau đó nối đầu ra Q_B và Q_C với 2 chân Reset: R_1 và R_2 . Hình 4.70 minh họa bộ đếm này.



Hình 4. 70. Bộ đếm chia hết cho 6.

b. IC 7492, 7493, IC 74293, 74393.

Cấu trúc chung của các IC đếm này được cho ở hình 4. 71.



Hình 4. 71. Sơ đồ khối của IC

Hoạt động của những bộ đếm này giống như IC 7490, chỉ khác là không có các đầu vào lập và Mod 6 không đếm theo trình tự nhị phân. Trình tự này được cho trong bảng 4.27. Các IC này thường không dùng làm các bộ đếm mà dùng làm bộ chia tần.

CLK	Q _D	Q _C	Q _B
0	0	0	0
1	0	0	1
2	0	1	0
3	1	0	0
4	1	0	1
5	1	1	0

Bảng 4.27. Bảng trạng thái Mod 6 của IC

Bảng 4.28 mô tả chức năng của bộ đếm 7492/ 7493/ 74293.

R ₀₁	R ₀₂	Q _D	Q _C	Q _B	Q _A
H	H	L	L	L	L
L	X	Thực hiện đếm			
X	L	Thực hiện đếm			

Bảng 4.28. Bảng chức năng của IC

IC 74393 gồm hai bộ đếm nhị phân 4 bit với một đầu vào xóa (Reset) cho mỗi bộ đếm, chúng hoạt động ở mức tích cực cao.

4.5. Bộ ghi dịch (Shift Register)

4.5.1. Giới thiệu chung:

Bộ ghi dịch còn gọi là thanh ghi dịch là phần tử không thể thiếu được trong CPU, trong các hệ vi xử lý. Nó có khả năng ghi (nhớ) số liệu và dịch thông tin (sang phải hoặc sang trái).

Bộ ghi dịch được cấu tạo từ một dãy phần tử nhớ được mắc liên tiếp với nhau và một số các cổng logic cơ bản hỗ trợ. Muốn ghi và truyền một từ nhị phân n bit cần n phần tử nhớ (n trigơ). Trong các bộ ghi dịch thường dùng các trigơ đồng bộ như trigơ RS, T, JK, D.

Thông thường người ta hay dùng trigơ D hoặc các trigơ khác nhưng mắc theo kiểu trigơ D để tạo thành các bộ ghi.

Phân loại:

- Phân theo cách đưa tín hiệu vào và lấy tín hiệu ra:

Vào nối tiếp, ra song song – SIPO (Serial Input, Parallel Output): thông tin được đưa vào thanh ghi dịch tuần tự từng bit một, số liệu được đưa ra đồng thời tức là tất cả n trigơ của thanh ghi được đọc cùng một lúc.

Vào song song, ra song song – PIPO (Parallel Input, Parallel Output): thông tin được đưa vào và lấy ra đồng thời ở n trigơ.

Vào nối tiếp, ra nối tiếp – SISO (Serial Input, Serial Output): thông tin được đưa vào và lấy ra tuần tự từng bit một.

Vào song song, ra nối tiếp – PISO (Parallel Input, Serial Output): thông tin được đưa vào đồng thời cả n trigơ, lấy ra tuần tự từng bit một dưới sự điều khiển của xung nhịp.

- Phân theo hướng dịch:

Dịch phải, dịch trái, dịch hai hướng, dịch vòng

- Phân theo đầu vào:

Đầu vào đơn: mỗi trigơ trong bộ ghi dịch chỉ sử dụng một đầu vào điều khiển, ví dụ như trigơ D hay các trigơ khác mắc theo kiểu D.

Đầu vào đôi: các trigơ trong bộ ghi dịch sử dụng cả hai đầu vào điều khiển, ví dụ hai đầu vào điều khiển của trigơ JK hay trigơ RS.

- Phân theo đầu ra:

Đầu ra đơn: mỗi trigơ trong bộ ghi dịch chỉ có một đầu ra Q_i (hay $\overline{Q_i}$) được đưa ra chân của vi mạch.

Đầu ra đôi: cả hai đầu ra của trigơ Q_i và $\overline{Q_i}$ đều được đưa ra chân của vi mạch.

Nếu kết hợp một số chức năng trên được bộ ghi dịch đa năng.

Tên IC	Mô tả
7491, 7491A	8 bit - vào nối tiếp, ra song song
7494	4 bit - vào song song, ra nối tiếp
7495	4 bit - vào nối tiếp/song song, ra song song (dịch phải, dịch trái)
7496	5 bit - vào song song, ra song song; vào nối tiếp, ra nối tiếp
7499	4 bit - dịch theo hai hướng
74164	8 bit - vào nối tiếp, ra song song
74165	8 bit - vào nối tiếp/song song, ra nối tiếp
74166	8 bit - vào nối tiếp/song song, ra nối tiếp
74178, 74179	4 bit - dịch theo hai hướng
74194	4 bit - dịch theo hai hướng
74195	4 bit - vào nối tiếp/song song, ra song song
74198	8 bit - dịch theo hai hướng
74199	8 bit - vào nối tiếp/song song, ra song song
74295A	4 bit 3 trạng thái - vào nối tiếp/song song, ra song song theo hai hướng
74395	4 bit 3 trạng thái - vào nối tiếp/song song, ra nối tiếp/song song

Bảng 4.29. Giới thiệu 1 số IC ghi dịch

Ứng dụng của bộ ghi dịch:

Bộ ghi dịch được sử dụng rộng rãi để nhớ dữ liệu, chuyển dữ liệu từ song song thành nối tiếp và ngược lại. Bộ ghi dịch là thành phần không thể thiếu được trong CPU của các hệ vi xử lý, trong các cổng vào/ra có khả năng lập trình.

Bộ ghi dịch còn được dùng để thiết kế bộ đếm, tạo dãy tín hiệu nhị phân tuần hoàn...

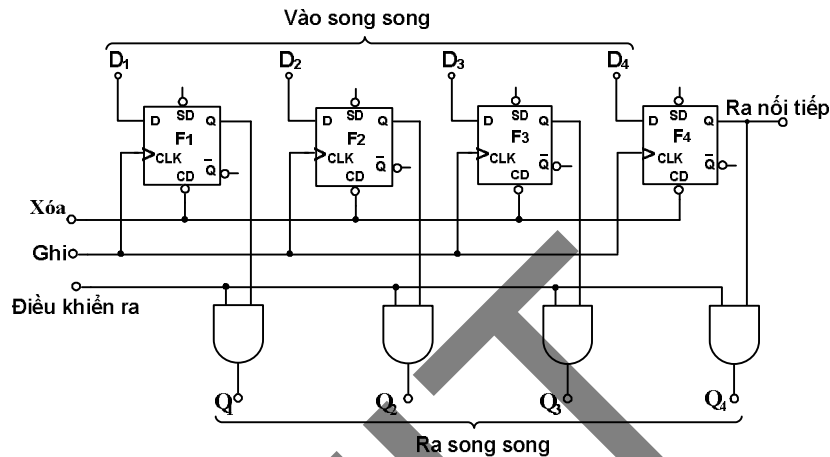
Các bộ ghi dịch có thể được thiết kế bằng cách sử dụng các trigơ rìi rạc hoặc được tích hợp trên các chip.

Bảng 4.29 giới thiệu một số IC TTL 54/74.

4.5.2. Bộ ghi song song

Sơ đồ bộ ghi song song được trình bày trên hình 4.72.

Trong sơ đồ dùng 4 trigơ D và một mạch điều khiển ra dùng cổng AND 2 đầu vào.

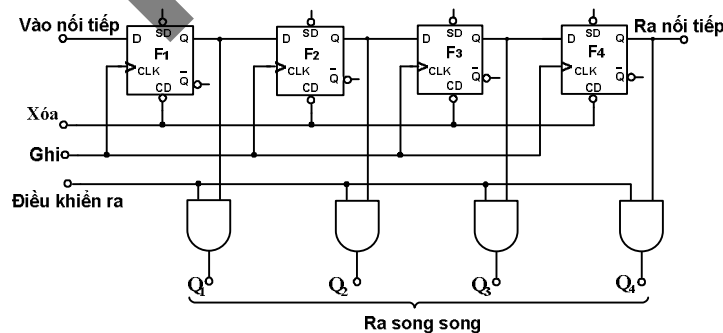


Hình 4. 72. Sơ đồ logic bộ ghi song song 4 bit

Hoạt động của sơ đồ như sau:

Đầu tiên dùng xung xóa $CD = 0$ để đầu ra $Q_1 Q_2 Q_3 Q_4 = 0 0 0 0$. Các số liệu cần ghi đưa vào D_1, D_2, D_3, D_4 . Khi có một xung điều khiển ghi đưa tới đầu vào CLK, dữ liệu được nạp vào bộ nhớ song song và cho đầu ra song song $Q_1 Q_2 Q_3 Q_4 = D_1 D_2 D_3 D_4$. Mỗi đầu ra Q được đưa tới 1 đầu vào của các cổng AND. Muốn cho dữ liệu tới các đầu ra, đầu vào “điều khiển ra” phải bằng 1.

4.5.3. Bộ ghi dịch nối tiếp



Hình 4. 73. Bộ ghi nối tiếp dịch phải

Bộ ghi dịch nối tiếp có thể dịch phải, dịch trái và cho ra song song hoặc ra nối tiếp. Hình 4.73 giới thiệu sơ đồ bộ ghi dịch vào nối tiếp dịch phải ra song song và ra nối tiếp

Đây là sơ đồ chỉ có đầu vào nối tiếp, còn đầu ra có cả ra song song và ra nối tiếp. Hoạt động của mạch được giải thích bằng bảng 4.30.

CLK	Q ₁	Q ₂	Q ₃	Q ₄
0	0	0	0	0
1	D ₄	0	0	0
2	D ₃	D ₄	0	0
3	D ₂	D ₃	D ₄	0
4	D ₁	D ₂	D ₃	D ₄
5	0	D ₁	D ₂	D ₃
6	0	0	D ₁	D ₂
7	0	0	0	D ₁

Bảng 4.30. Bảng trạng thái của bộ ghi dịch

Giả sử muốn nhập số liệu D₁, D₂, D₃, D₄, đây là mạch ghi dịch phải, do đó phải bit D₄ trước. Sau 4 xung nhịp thì cả 4 bit số liệu được nạp xong.

Cách lấy số liệu ra:

+ Ra song song: Sau 4 xung nhịp nạp xong số liệu, ở 4 đầu ra Q: Q₁ Q₂ Q₃ Q₄ = D₁ D₂ D₃ D₄; muốn lấy ra song song thì cho đầu “điều khiển ra” = 1.

+ Ra nối tiếp: Lấy ra ở Q₄ (trigơ cuối cùng).

Sau 4 xung nhịp đã nạp xong, D₄ xuất hiện ở đầu ra nối tiếp, vậy cần 3 xung nữa để D₁ D₂ D₃ được đưa ra Q₄.

Tóm lại, muốn ghi nối tiếp 4 bit cần 4 xung CLK và cho ra ở đầu ra song song. Còn để lấy số liệu ra nối tiếp cần thêm 3 xung nhịp nữa.

4.5.4. Ứng dụng của bộ ghi dịch

Ứng dụng cơ bản của các bộ ghi dịch là lưu trữ dữ liệu tạm thời và xử lý bit. Một số ứng dụng chung của bộ ghi dịch được trình bày sau đây:

a. Tạo trễ

Bộ ghi dịch SISO có thể được dùng để tạo thời gian trễ Δt cho các tín hiệu số và được cho bởi công thức:

$$\Delta t = N \times \frac{1}{f_c} \quad (4.44)$$

Trong đó N là số trigơ và f_c = là tần số xung Clock.

Như vậy, một xung đầu vào xuất hiện tại đầu ra sẽ bị trễ một khoảng thời gian Δt . Thời gian trễ này được tạo bởi tần số xung Clock hoặc số lượng trigơ có trong bộ ghi dịch.

b. Bộ chuyển dữ liệu từ nối tiếp sang song song

Nhờ bộ ghi dịch SIPO mà dữ liệu ở dạng nối tiếp được chuyển sang dạng song song.

c. Bộ chuyển dữ liệu từ song song sang nối tiếp

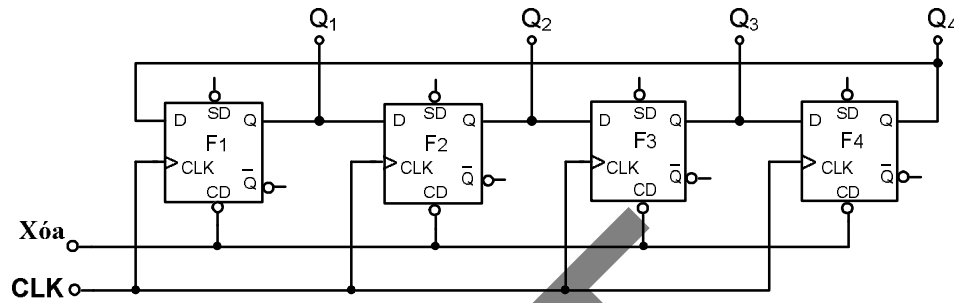
Nhờ bộ ghi dịch PISO mà dữ liệu ở dạng song song được chuyển sang dạng nối tiếp.

d. Bộ đếm vòng

Sơ đồ mạch được chỉ ra trên hình 4.74.

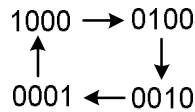
Mạch được phản hồi từ đầu ra về đầu vào thông qua biểu thức $D_1 = Q_n$, tức là nối đầu ra Q của trigơ thứ n với đầu vào D của trigơ đầu tiên tính từ trái sang phải. Lúc này trigơ tạo thành mạch vòng nên gọi là bộ đếm vòng; thực chất đây là bộ ghi dịch tự tuần hoàn.

Bằng phương pháp phân tích bộ đếm, có thể tìm được đồ hình trạng thái của bộ đếm vòng như hình 4.75.

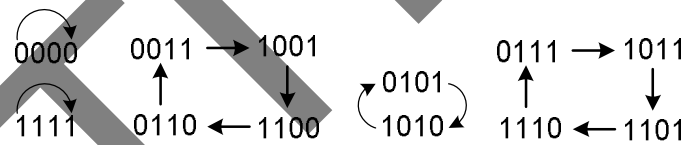


Hình 4. 74. Bộ đếm vòng

Trạng thái được dùng



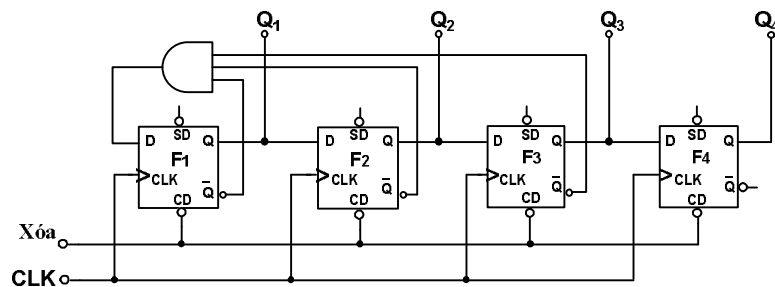
Trạng thái không được dùng



Hình 4. 75. Đồ hình trạng thái bộ đếm vòng

Nhìn vào đồ hình trạng thái, thấy rằng trong số đồ hình trên chỉ có một đồ hình thỏa mãn mã vòng (xem chương 1) là 1000, 0100, 0010, 0001. Do vậy, để mạch hoạt động chính xác, phải dùng xung kích (có thể sử dụng chân Preset) để đưa bộ đếm vào trạng thái sử dụng (ví dụ: 1000), sau đó mới tác động xung Clock.

Từ các đồ hình trạng thái này, thấy bộ đếm này không tự khởi động, tức là vì một lý do nào đó trạng thái của bộ đếm rơi vào các trạng thái không được sử dụng thì nó không thể trở về trạng thái được dùng, chỉ khi khởi động lại mạch mới có thể trở về trạng thái chính xác.



Hình 4. 76. Bộ đếm vòng 4 bit tự khởi động.

Hình 4.76 là bộ đếm vòng 4 bit tự khởi động.

Phân tích bộ đếm này giống như các bộ đếm đồng bộ.

Phương trình kích được viết như sau:

$$\left. \begin{aligned} D_1 &= \overline{Q_1^n} \overline{Q_2^n} \overline{Q_3^n} \\ D_2 &= Q_1^n \\ D_3 &= Q_2^n \\ D_4 &= Q_3^n \end{aligned} \right\}$$

Thay phương trình kích vào phương trình đặc trưng ($Q_i^{n+1} = D_i$) tìm được phương trình chuyển đổi trạng thái:

$$\left. \begin{aligned} Q_1^{n+1} &= \overline{Q_1^n} \overline{Q_2^n} \overline{Q_3^n} \\ Q_2^{n+1} &= Q_1^n \\ Q_3^{n+1} &= Q_2^n \\ Q_4^{n+1} &= Q_3^n \end{aligned} \right\}$$

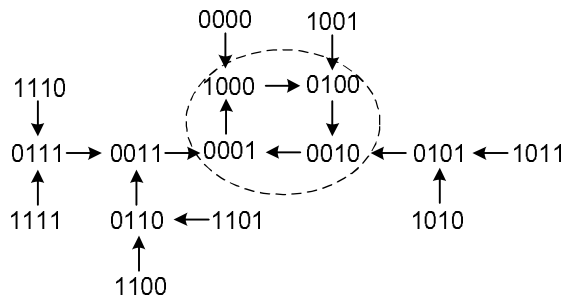
Từ phương trình chuyển đổi trạng thái, lập được bảng trạng thái như bảng 4.31.

Q_1^n	Q_2^n	Q_3^n	Q_4^n	Q_1^{n+1}	Q_2^{n+1}	Q_3^{n+1}	Q_4^{n+1}
0	0	0	0	1	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	1
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	0
1	1	1	0	0	1	1	1
1	1	1	1	0	1	1	1

Bảng 4.31. Bảng trạng thái bộ đếm vòng

Từ bảng trạng thái, vẽ được đồ hình trạng thái trên hình 4.77.

Ưu điểm của bộ đếm vòng: trong tất cả các trigơ chỉ có một trigơ có trạng thái 1, nên muốn lấy đầu ra của bộ đếm vòng tại các đầu ra Q mà không cần bộ giải mã. Khi cấp xung Clock liên tục thì các đầu ra Q và \overline{Q} của các trigơ sẽ cho ra các xung vuông luân lưu. Do đó, bộ đếm vòng còn được gọi là bộ phân phối xung vuông.



Hình 4. 77. Đồ hình trạng thái bộ đếm vòng tự khởi động

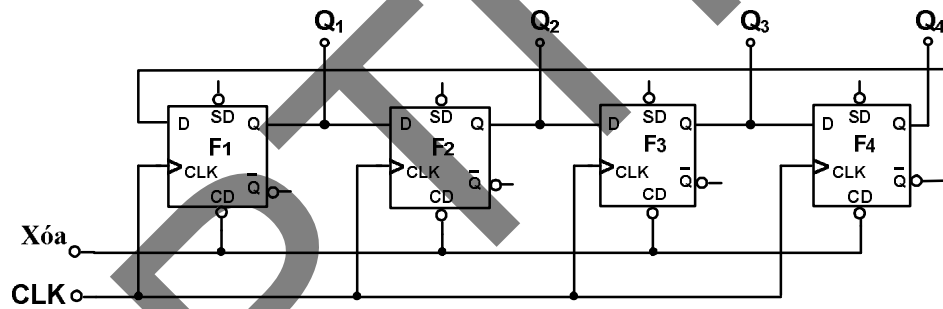
Nhược điểm: Hiệu suất sử dụng trạng thái thấp, số lượng trigger bằng số lượng bit nhớ nên phải sử dụng nhiều trigger.

e. Bộ đếm vòng xoắn

Đặc điểm của mã vòng xoắn (mã Johnson) là có số bit 1 trong từ mã tăng dần, sau đó lại giảm dần. Do đó, cấu trúc mạch của nó tuân theo biểu thức:

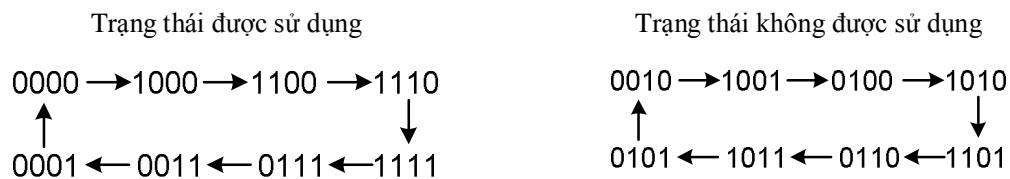
$$D_1 = \overline{Q_n}$$

Sơ đồ mạch của bộ đếm được chỉ ra trên hình 4.78.



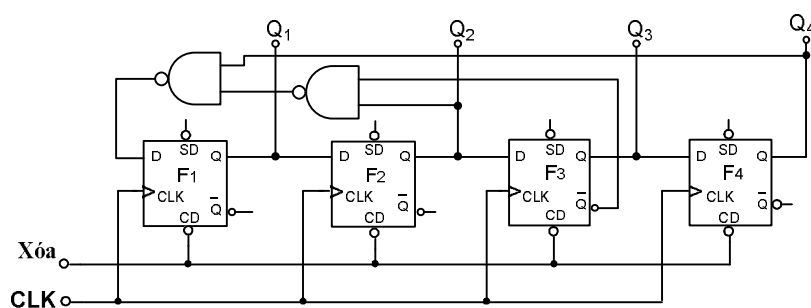
Hình 4. 78. Bộ đếm vòng xoắn

Bằng cách phân tích, xây dựng được đồ hình trạng thái 4.79, trong đó có 8 trạng thái không được sử dụng. Để mạch hoạt động đúng, trước khi đếm, phải reset bộ đếm về trạng thái 0000.



Hình 4. 79. Đồ hình trạng thái của bộ đếm vòng xoắn

Do bộ đếm hình 4.78 không tự khởi động nên hình 4.80 giới thiệu bộ đếm vòng xoắn tự khởi động.



Hình 4. 80. Bộ đếm vòng xoắn tự khởi động

Tiến hành phân tích mạch:

Phương trình kích:

$$\left. \begin{aligned} D_1 &= \overline{Q_2^n} \cdot \overline{Q_3^n} \cdot Q_4^n = \overline{Q_2^n} \cdot \overline{Q_3^n} + Q_4^n \\ D_2 &= Q_1^n \\ D_3 &= Q_2^n \\ D_4 &= Q_3^n \end{aligned} \right\}$$

Thay phương trình kích vào phương trình đặc trưng ($Q_i^{n+1} = D_i$) tìm được phương trình chuyển đổi trạng thái:

$$\left. \begin{aligned} Q_1^{n+1} &= \overline{Q_2^n} \cdot \overline{Q_3^n} + Q_4^n \\ Q_2^{n+1} &= Q_1^n \\ Q_3^{n+1} &= Q_2^n \\ Q_4^{n+1} &= Q_3^n \end{aligned} \right\}$$

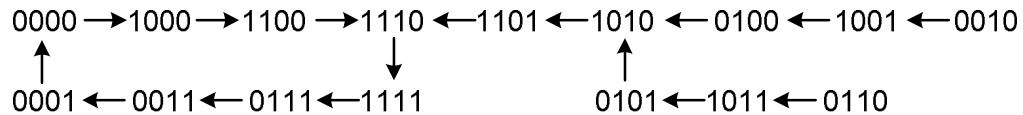
Từ phương trình chuyển đổi trạng thái, lập được bảng trạng thái như bảng 4.32.

Q_1^n	Q_2^n	Q_3^n	Q_4^n	Q_1^{n+1}	Q_2^{n+1}	Q_3^{n+1}	Q_4^{n+1}
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	1
0	0	1	1	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0
0	1	1	0	1	0	1	1
0	1	1	1	0	0	1	1
1	0	0	0	1	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	1	0	0	1	1	1	0

Q_1^n	Q_2^n	Q_3^n	Q_4^n	Q_1^{n+1}	Q_2^{n+1}	Q_3^{n+1}	Q_4^{n+1}
1	1	0	1	0	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1

Bảng 4.32. Bảng trạng thái bộ đếm vòng

Từ bảng trạng thái, vẽ được đồ hình trạng thái trên hình 4.79.



Hình 4. 81. Đồ hình trạng thái bộ đếm vòng xoắn tự khởi động

g. Bộ phát xung tuần tự

Bộ phát xung tuần tự có thể được dùng làm:

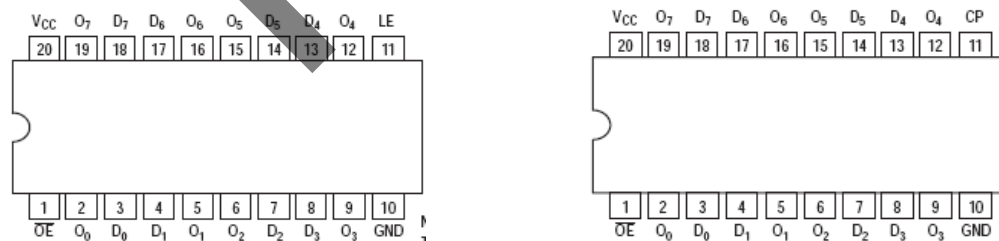
- + Bộ đếm.
- + Bộ phát xung giả ngẫu nhiên.
- + Bộ phát xung có chu kỳ được chọn sẵn và tuần tự.
- + Bộ phát xung mã...

4.6. Thanh chốt dữ liệu (Latch)

Thanh chốt dữ liệu là mạch logic số được dùng để lưu trữ trạng thái số (1 hoặc 0) trong bộ lưu trữ dữ liệu.

Thanh chốt dữ liệu thường được sử dụng trong các mạch giao tiếp Bus dữ liệu, các bộ phân kênh, hợp kênh, và trong các mạch điều khiển...

Hình 4.82 giới thiệu IC chốt dữ liệu 3 trạng thái: 74373 và 74374.



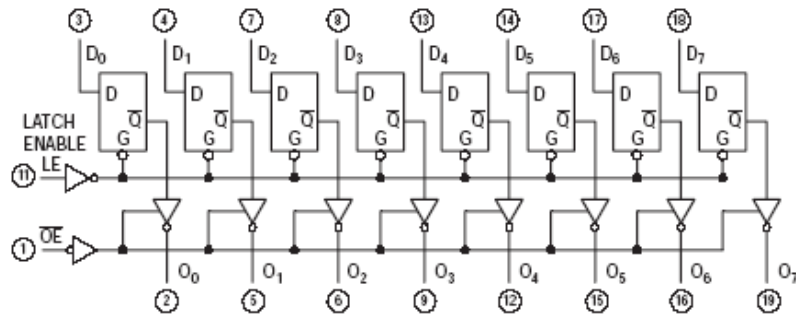
Hình 4. 82. Sơ đồ chân của IC 74373/74374

$D_0 \div D_7$ (Data)	Đầu vào dữ liệu
LE (Latch Enable)	Đầu vào điều khiển chốt dữ liệu (hoạt động ở mức cao)
CP (Clock Pulse)	Xung đồng hồ (hoạt động ở sườn dương)
\overline{OE} (Output Enable)	Điều khiển đầu ra (hoạt động ở mức thấp)
$O_0 \div O_7$ (Output)	Đầu ra dữ liệu

Bảng 4.33. Bảng mô tả chân của IC

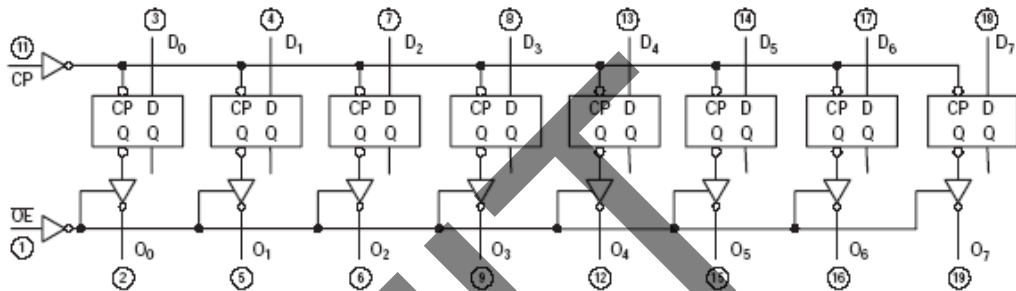
Hình 4.83 và 4.84. Sơ đồ logic của IC chốt

SN74LS373



Hình 4. 83. Sơ đồ logic của IC 74373

SN74LS374



Hình 4. 84. Sơ đồ logic của IC 74374

Bảng 4.34, 4.35 là bảng chức năng của IC

D_n	LE	\overline{OE}	O_n
H	H	L	H
L	H	L	L
X	L	L	Q_0
X	X	H	Z

Bảng 4.34. Bảng chức năng của IC 74373

D_n	LE	\overline{OE}	O_n
H	\uparrow	L	H
L	\uparrow	L	L
X	X	H	Z

Bảng 4.35. Bảng chức năng của IC 74374

Tám bộ chốt của IC 74373 là các bộ chốt theo mức, tức là khi tín hiệu điều khiển chốt LE ở mức cao thì các đầu ra $Q = D$; còn nếu nó ở mức thấp thì các đầu ra sẽ bị chốt theo mức tín hiệu trước đó.

Tám bộ lật của IC 74374 là các bộ lật theo kiểu D, chúng hoạt động tại sườn dương của xung Clock. Khi đúng sườn điều khiển thì các đầu ra Q được đặt theo mức của các đầu vào D.

Tín hiệu điều khiển đầu ra có thể được dùng để đặt 8 đầu ra hoặc ở trạng thái bình thường (mức 0 hoặc mức 1) hoặc ở trạng thái trở kháng cao. Trong trạng thái trở kháng cao các đầu ra là không tải và không thể điều khiển các đường Bus.

Tín hiệu điều khiển đầu ra không ảnh hưởng đến hoạt động bên trong của mạch chốt và mạch lật. Điều này có nghĩa là khi chưa có tín hiệu điều khiển đầu ra nhưng số liệu cũ vẫn được duy trì hoặc dữ liệu mới vẫn có thể được đưa vào.

TÓM TẮT

Khác với mạch logic tổ hợp, mạch logic tuần tự có tín hiệu đầu ra phụ thuộc không những tín hiệu đầu vào ở thời điểm xét mà cả vào trạng thái mạch điện sẵn có ở thời điểm đó. Đây là đặc điểm chức năng logic của mạch tuần tự. Để nhớ trạng thái mạch điện, mạch tuần tự phải có phần tử nhớ - đó là các trigơ.

Tính chất cơ bản của Trigơ

Trigơ là linh kiện logic cơ bản của mạch số. Trigơ có hai trạng thái ổn định, dưới tác dụng của tín hiệu bên ngoài có thể chuyển đổi từ trạng thái ổn định này sang trạng thái ổn định kia, nếu không có tác dụng tín hiệu bên ngoài thì nó duy trì mãi trạng thái ổn định vốn có. Vì thế, trigơ có thể được dùng làm phần tử nhớ của số nhị phân.

Quan hệ giữa chức năng logic và hình thức cấu trúc của trigơ

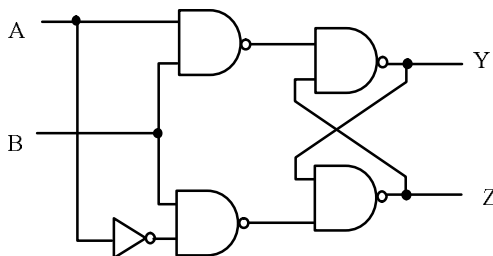
Chức năng logic và hình thức cấu trúc của trigơ là hai khái niệm khác nhau. Chức năng logic là quan hệ giữa trạng thái tiếp theo của đầu ra với trạng thái hiện tại của đầu ra và các tín hiệu đầu vào. Do chức năng logic khác nhau mà trigơ được phân thành các loại RS, D, T, JK. Còn do hình thức cấu trúc khác nhau mà trigơ lại được phân thành loại trigơ thường và loại trigơ chính phụ.

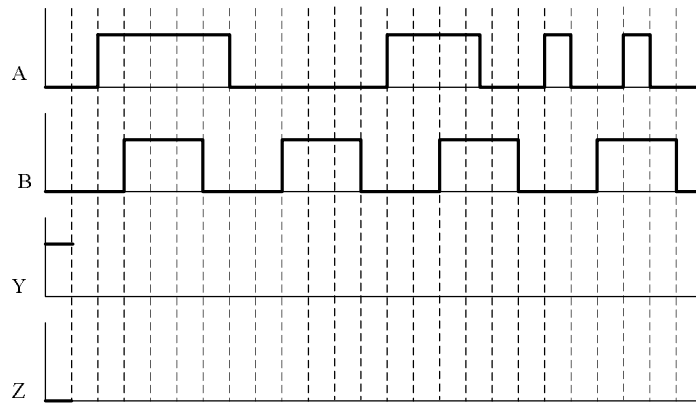
Một trigơ có chức năng logic xác định có thể thực hiện bằng các hình thức cấu trúc khác nhau. Ví dụ, các trigơ cấu trúc loại chính phụ và loại thường đều có thể thực hiện chức năng của một trigơ khác. Nghĩa là cùng một cấu trúc có thể đảm trách những chức năng khác nhau.

Mạch tuần tự cụ thể có rất nhiều chủng loại. Chương này chỉ giới thiệu một số loại mạch tuần tự điển hình: bộ đếm, bộ ghi dịch... Đồng thời với việc nắm vững cấu trúc, nguyên lý công tác và đặc điểm của các mạch tuần tự đó, chúng ta cũng phải nắm vững được đặc điểm chung của mạch tuần tự và phương pháp chung khi phân tích và thiết kế mạch tuần tự.

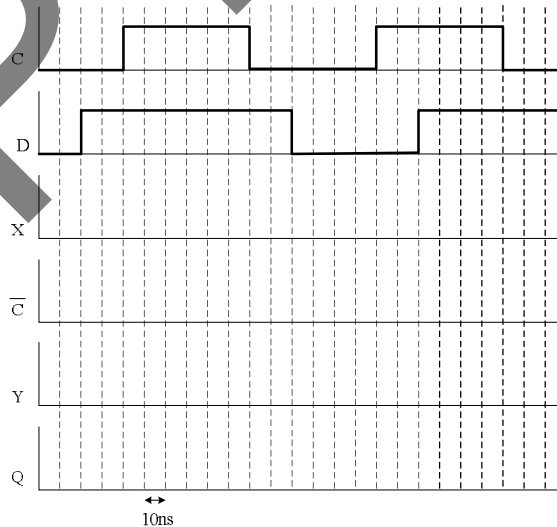
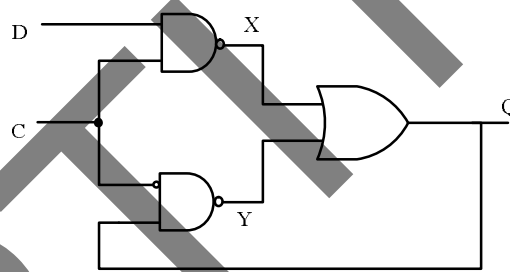
CÂU HỎI ÔN TẬP CHƯƠNG 4

1. Mạch điện dưới đây là mạch của trigơ nào? Vẽ tiếp giản đồ xung ra của mạch?



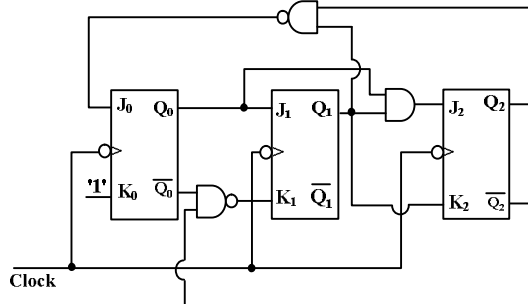


2. Một trigơ JK được ở chế độ lật. Nếu tần số Clock của nó là 1000 Hz thì tần số tại đầu ra là bao nhiêu?
3. Một bộ đếm nhị phân 4 bit thì tần số tại đầu ra của bit có trọng số lớn nhất so với tần số xung nhịp là bao nhiêu?
4. Hãy vẽ tiếp giản đồ xung sau. Mạch làm việc có thời gian trễ tín hiệu là $t_p = 10\text{ns}$ đối với mỗi cổng (tín hiệu \bar{C} trong giản đồ xung này không có thời gian trễ).



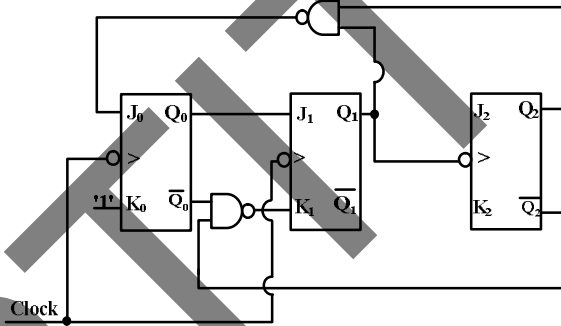
5. Thiết kế bộ đếm thuận mã Gray 3 bit.
6. Thiết kế bộ đếm nghịch Mod 6.
7. Thiết kế bộ đếm thuận Mod 6.
8. Thiết kế bộ đếm hiển thị giờ, phút, giây.

9. Thiết kế bộ đếm vòng 3 bit tự khởi động.
10. Thiết kế bộ tạo xung tuần tự có 6 nhịp xung, yêu cầu độ rộng xung nhịp bằng 4 lần chu kỳ xung Clock.
11. Phân tích bộ đếm sau?



Hình vẽ bài 11

12. Cho bộ đếm sau. Cho biết đây là bộ đếm Mod mấy?



Hình vẽ bài 12

13. Thiết kế bộ đếm không đồng bộ $M = 9$?
14. Thiết kế bộ đếm đồng bộ $M = 12$?
15. Thiết kế bộ đếm nghịch thập phân đồng bộ?

TÀI LIỆU THAM KHẢO

1. *Giáo trình Điện tử số*, Trần Thị Thúy Hà, Đỗ Mạnh Hà, NXB Bưu điện 2010.
2. *Giáo trình Kỹ thuật số*, Trần Văn Minh, NXB Bưu điện 2002.
3. *Cơ sở kỹ thuật điện tử số*, Đại học Thanh Hoa, Bắc Kinh, NXB Giáo dục 1996 .
4. *Kỹ thuật số*, Nguyễn Thúy Vân, NXB Khoa học và kỹ thuật 1994.
5. *Kỹ thuật điện tử số thực hành*, Bạch Gia Dương – Chữ Đức Trình, Nhà xuất bản đại học quốc gia Hà nội 2007.
6. *Giáo trình Kỹ thuật số*, Nguyễn Viết Nguyên, Nhà xuất bản giáo dục 2004.
7. *Mạch logic kỹ thuật số*, Nguyễn Minh Đức, Nhà xuất bản tổng hợp thành phố Hồ Chí Minh, 2004.
8. *Toán logic và kỹ thuật số*, Nguyễn Nam Quân - Khoa ĐHTC xuất bản – 2004.
9. *Lý thuyết mạch logic và Kỹ thuật số*, Nguyễn Xuân Quỳnh - NXB Bưu điện – 1984.
10. *Fundamentals of logic design*, fourth edition, Charles H. Roth, Prentice Hall 1991.
11. *Lessons in Electric Circuits, Volume No 4.Digital*, Tony R. Kuphaldt, Tái bản lần thứ 4.2007.
12. *Digital engineering design*, Richard F.Tinder, Prentice Hall 1991 .
13. *Digital design principles and practices*, John F.Wakerly, Prentice Hall 1990.