

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN THỊ THÚY HÀ

**BÀI GIẢNG
ĐIỆN TỬ SỐ**

HÀ NỘI – 10.2013

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 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 và 1 phụ lục được bố cục như sau:

Chương 1: Hàm Boole và cổng logic.

Chương 2: Mạch logic tổ hợp.

Chương 3: Mạch logic tuần tự.

Chương 4: Bộ nhớ bán dẫn.

Phụ lục: Giới thiệu một số hệ đếm thông dụng

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	Arthmetic 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 khoá 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	FET có cực cửa cách ly bằng lớp oxit kim loại
MROM	Mask ROM	ROM được chế tạo bằng phương pháp che mặt nạ
MSB	Most Significant Bit	Bit có ý nghĩa lớn nhất
MSI	Medium Scale Integrated	Mức độ tích hợp trung bình
MUX	Multiplexer	Bộ ghép kênh
Nibble	4 bit	
NMOS	N – channel MOS	Transistor trường kênh dẫn N
PMOS	P – channel MOS	Transistor trường kênh dẫn P
PRE	Preset	Tái lập

RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
RAS	Row Address Select	Chọn địa chỉ hàng
RBI	Riple Blanking Input	Đầu vào xóa nối tiếp
RBO	Riple Blanking Output	Đầu ra xóa nối tiếp
ROM	Read Only Memory	Bộ nhớ chỉ đọc
RTL	Resistance Transistor Logic	Cổng logic dùng điện trở và transistor
SRAM	Static RAM	RAM tĩnh
SSI	Small Scale Integrated	Mức độ tích hợp trung bình
TTL	Transistor – Transistor Logic	Cổng logic dùng Transistor
VLSI	Very Large Scale Integrated	Mức độ tích hợp rất lớn

MỤC LỤC

LỜI NÓI ĐẦU	i
THUẬT NGỮ VIẾT TẮT	ii
MỤC LỤC	v
CHƯƠNG 1. HÀM BOOLE VÀ CÔNG LOGIC	1
GIỚI THIỆU CHUNG	1
1.1 ĐẠI SỐ BOOLE	1
1.1.1. Các định lý cơ bản.....	2
1.1.2 Các định luật cơ bản:.....	2
1.1.3. Ba quy tắc về đẳng thức :.....	2
1.2 CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM BOOLE.....	3
1.2.1 Bảng trạng thái.....	3
1.2.2 Phương pháp đại số.....	4
1.2.3 Phương pháp bảng Các nô (bảng Karnaugh hay phương pháp hình học.).....	6
1.3. CÁC PHƯƠNG PHÁP TỐI THIỂU HÓA (RÚT GỌN HÀM).....	8
1.3.1. Phương pháp đại số	8
1.3.2 Phương pháp bảng Các nô.....	9
1.4 CÔNG LOGIC VÀ CÁC THAM SỐ CHÍNH.....	12
1.4.1 Cổng logic cơ bản	13
1.4.2. Logic dương và logic âm.....	16
1.4.3. Một số cổng ghép thông dụng.....	16
1.4.4 Tính đa chức năng của cổng NAND, NOR.....	19
1.5. Các tham số chính.....	22
1.6. MỘT SỐ LƯU Ý KHI SỬ DỤNG IC SỐ.....	26
1.6.1. Sơ đồ chân và ký hiệu trên thân IC.....	26
1.6.2. Một số đặc điểm của IC họ TTL và CMOS.	27
1.6.3. Xử lý cổng thừa, lỗi vào thừa.....	27
TÓM TẮT	27
CÂU HỎI ÔN TẬP	28
CHƯƠNG 2: MẠCH LOGIC TỔ HỢP	33
GIỚI THIỆU CHUNG	33
2.1 KHÁI NIỆM CHUNG.....	33
2.1.1. Đặc điểm cơ bản của mạch tổ hợp.....	33
2.1.2. Phương pháp biểu diễn chức năng logic	34

2.2 PHÂN TÍCH MẠCH LOGIC TỔ HỢP	34
2.3 THIẾT KẾ MẠCH LOGIC TỔ HỢP	35
2.4. MẠCH MÃ HOÁ VÀ GIẢI MÃ	38
2.4.1 Một số loại mã nhị phân thông dụng	38
2.4.2. Các mạch mã hoá:	40
2.4.3. Các bộ giải mã	44
2.4.4. Các bộ biến mã.....	54
2.5. BỘ HỢP KÊNH VÀ PHÂN KÊNH	57
2.5.1 Bộ hợp kênh (MUX-Multiplexer)	57
2.5.2. Bộ phân kênh (Demultiplexer: DMUX)	60
2.5.3. Một số ứng dụng của bộ hợp kênh và phân kênh	62
2.6. MẠCH SỐ HỌC.	64
2.6.1. Mạch tổng	64
2.6.2. Mạch hiệu.	67
2.6.3. Bộ cộng, trừ theo bù 1 và bù 2.	69
2.7. MẠCH SO SÁNH.	70
2.7.1. Bộ so sánh.....	70
2.8. MẠCH TẠO VÀ KIỂM TRA CHẴN LẺ.....	73
2.8.1. Mã chẵn, lẻ.	73
4.8.2. Mạch tạo và kiểm tra chẵn/lẻ.	73
2.9. MẠCH TẠO MÃ VÀ GIẢI MÃ HAMMING	76
2.9.1. Tạo mã.....	76
2.9.2. Giải mã.....	79
2.10. ĐƠN VỊ SỐ HỌC VÀ LOGIC (ALU).....	80
TÓM TẮT	83
CÂU HỎI ÔN TẬP	84
CHƯƠNG 3. MẠCH LOGIC TUẦN TỰ.....	85
GIỚI THIỆU.....	85
NỘI DUNG	85
3.1. KHÁI NIỆM CHUNG VÀ MÔ HÌNH TOÁN HỌC.....	85
3.1.1. Khái niệm chung	85
3.1.2. Mô hình toán học	85
3.2. PHẦN TỬ NHỚ CỦA MẠCH TUẦN TỰ	86
3.2.1. Các loại Trơơ	86
3.2.2. Đầu vào không đồng bộ của trơơ.	95
3.2.3. Chuyển đổi giữa các loại trơơ.	96
3.3. ỨNG DỤNG CỦA TRƠƠ TRONG MẠCH ĐỊNH THỜI	104

3.3.1. Mạch điện của IC 555.....	104
3.3.2. Một vài ứng dụng của IC định thời 555.....	105
3.4. PHƯƠNG PHÁP MÔ TẢ MẠCH TUẦN TỰ.....	108
3.4.1. Bảng.....	109
3.4.2. Đồ hình trạng thái.....	110
3.5. PHÂN TÍCH MẠCH TUẦN TỰ.....	112
3.5.1. Các bước phân tích mạch tuần tự.....	112
3.5.2. Phân tích mạch tuần tự đồng bộ.....	113
3.5.3. Phân tích mạch tuần tự không đồng bộ.....	115
3.6. THIẾT KẾ MẠCH TUẦN TỰ.....	117
3.6.1. Các bước thiết kế mạch tuần tự đồng bộ.....	117
3.6.2. Các bước thiết kế mạch tuần tự không đồng bộ.....	118
3.6.3. Thiết kế mạch tuần tự từ đồ hình trạng thái.....	121
3.6.4. Ví dụ.....	123
3.6.3. Thiết kế mạch tuần tự từ bảng.....	128
3.7. MỘT SỐ VÍ DỤ KHÁC.....	131
3.7.1. Mạch tuần tự đồng bộ.....	131
3.7.2. Mạch tuần tự không đồng bộ.....	135
3.8. MỘT SỐ MẠCH TUẦN TỰ THÔNG DỤNG.....	140
3.8.1. Bộ đếm.....	140
3.8.2. Thiết kế bộ đếm.....	159
3.8.3. Thiết kế bộ đếm từ IC đếm.....	166
3.8.4. Bộ ghi dịch (Shift Register).....	170
TÓM TẮT.....	179
CÂU HỎI ÔN TẬP CHƯƠNG 3.....	180
CHƯƠNG 4: BỘ NHỚ BÁN DẪN.....	183
GIỚI THIỆU.....	183
4.1. KHÁI NIỆM CHUNG.....	183
4.1.1. Khái niệm.....	183
4.1.2. Những đặc trưng chính của bộ nhớ.....	183
4.1.3. Phân loại.....	184
4.1.4. Tổ chức của bộ nhớ.....	185
4.2. BỘ NHỚ CỐ ĐỊNH - ROM.....	186
4.2.1. Cấu trúc chung của ROM.....	186
4.2.2. MROM.....	192
4.2.3. PROM.....	193

4.3. BỘ NHỚ BẢN CỐ ĐỊNH	194
4.3.1. EPROM (Erasable PROM)	194
4.3.2. EEPROM (Electrically Erasable PROM)	195
4.4. RAM.....	196
4.4.1. Cấu trúc khối của RAM.....	196
4.4.2. Cấu tạo của DRAM	198
4.4.3. SRAM	199
4.5. Đĩa CỨNG SILICON- BỘ NHỚ FLASH	199
4.6. BỘ NHỚ CACHE	201
TÓM TẮT	201
CÂU HỎI ÔN TẬP	202
TÀI LIỆU THAM KHẢO	203
PHỤ LỤC	204
GIỚI THIỆU VỀ HỆ ĐẾM	204
A1. Hệ thập phân	204
A2. Hệ nhị phân.....	204
A3. Các phép tính trong hệ nhị phân.....	205
B. Hệ 8 (bát phân) và hệ 16 (thập lục phân).....	206
C. Chuyển đổi cơ số giữa các hệ đếm.....	207
C1. Chuyển đổi từ hệ cơ số 10 sang các hệ khác	207
C.2. Đổi một biểu diễn trong hệ bất kì sang hệ thập phân.....	208
C.3. Đổi các số từ hệ nhị phân sang hệ cơ số 8 và 16	209

CHƯƠNG 1. HÀM BOOLE VÀ 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 có, mệnh đề đúng hoặc sai; các đối tượng này được biểu diễn bằng biến logic. Khi trạng thái đối tượng là tồn tại thì biến logic biểu diễn có giá trị là 1 và ký hiệu là A , nếu không thì 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:

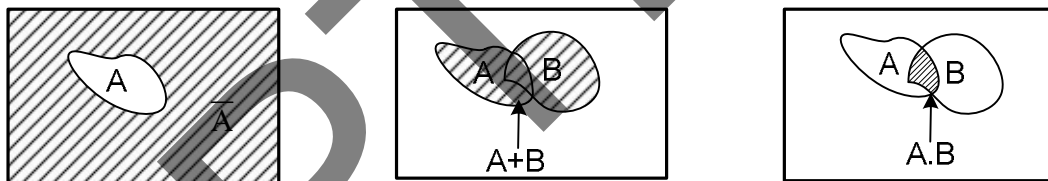
Phép phủ định logic đối với một biến A hay còn gọi là phép đảo. Khi nhận tác động của phép toán này, A sẽ nhận giá trị đảo với giá trị ban đầu và ký hiệu là \bar{A} .

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 là thì nhận giá trị 0. Ta nhận được trong tập hợp này hai hằng số 0 và 1.



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

Để thể hiện các hàm logic bằng mạch điện người ta sử dụng các cổng logic. Các cổng logic được xây dựng dựa trên cấu hình mạch chuyên biệt được gọi là họ mạch logic điển hình là : Mạch logic Điện trở - Transistor (RTL), Mạch logic Điốt – Transistor (DTL), Mạch logic Transistor – Transistor (TTL), CMOS...

Trong chương này sẽ trình bày các ký hiệu cổng logic chủ yếu và được dùng phổ biến hiện nay.

1.1 ĐẠI SỐ BOOLE

Có ba loại quan hệ logic cơ bản nhất là: ĐẢO, HOẶC, VÀ. Mạch điện thực hiện ba phép toán cơ bản là cổng NOT, OR và AND. Ngoài ba phép toán cơ bản trên còn có các phép toán logic khác như: NAND, NOR, XOR, XNOR...

Các tín hiệu vào còn được gọi là các biến logic vào, tín hiệu ra được gọi là hàm ra. Trong đại số logic, biến số và hàm số đều chỉ lấy hai giá trị là “0” và “1”. Mỗi biến số biểu thị

một điều kiện để sự kiện có thể phát sinh. Điều kiện đó chỉ có thể có hoặc không. Hàm số biểu thị bản thân sự kiện đó có phát sinh hay không phát sinh.

1.1.1. 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	$A.\bar{A} = 0$	$A + \bar{A} = 1$
4	Bất biến	$A.A = A$	$A + A = A$
5	Hấp thụ	$A + A.B = A$	$A.(A + B) = A$
6	Hoàn nguyên	$\bar{\bar{A}} = A$	
7	Định lý DeMorgan	$\overline{(A.B.C...)} = \bar{A} + \bar{B} + \bar{C}...$	$\overline{(A + B + C + ...)} = \bar{A}.\bar{B}.\bar{C}...$

Bảng 1.1. Một số định lý cơ bản trong đại số Boole

1.1.2 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 $A.B = A$

1.1.3. Ba quy tắc về đẳng thức :

1.1.3.1. 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: Ta có công thức $\overline{(A+B)} = \bar{A}.\bar{B}$. Dùng $F = A+C$ thay vào biến A, ta có:

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

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

1.1.3.3. 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.

1.2 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.

1.2.1 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**.

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 1-2) và có tên gọi là các **hạng tích** hay còn gọi là **minterm**.

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 1-2. Bảng trạng thái hàm 3 biến

Đặc điểm của bảng trạng thái:

+ Rõ ràng, trực quan. Sau khi xác định các giá trị biến vào có thể tìm được giá trị đầu ra nhờ bảng trạng thái. Do vậy, trong các sổ tay tra cứu đều giới thiệu bảng trạng thái để độc giả biết được chức năng logic của mạch.

+ Để giải quyết bài toán ở dạng logic thì sử dụng bảng trạng thái là hữu ích nhất. Do vậy, 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.

1.2.2 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 1-3 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 1-3. 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:

$$f(X_{n-1}, \dots, X_0) = \sum_{i=0}^{2^n-1} a_i m_i \quad (1.1)$$

hoặc bằng chỉ một dạng tích các tổng:

$$f(X_{n-1}, \dots, X_0) = \prod_{i=0}^{2^n-1} (a_i + M_i) \quad (1.2)$$

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

a) Biểu diễn hàm sau theo dạng minterm:

$F(A, B, C) = A + BC \rightarrow$ Đây là dạng minterm không đầy đủ. Muốn đưa về dạng chuẩn tắc (đủ biến) ta 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 1-3). 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:

$$\begin{aligned} F(A, B, C) &= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC \\ &\quad \begin{array}{ccccc} \downarrow\downarrow\downarrow & \downarrow\downarrow\downarrow & \downarrow\downarrow\downarrow & \downarrow\downarrow\downarrow & \downarrow\downarrow\downarrow \\ 111_2 & 110_2 & 101_2 & 100_2 & 011_2 \\ 7_{10} & 6_{10} & 5_{10} & 4_{10} & 3_{10} \end{array} \\ &\rightarrow F(A, B, C) = m_7 + m_6 + m_5 + m_4 + m_3 = \sum (3, 4, 5, 6, 7) \end{aligned}$$

b) Biểu diễn hàm sau theo dạng Maxterm:

$F(A, B, C) = A + BC = (A + B)(A + C) \rightarrow$ Đây là dạng Maxterm không đầy đủ. Muốn đưa về dạng chuẩn (đủ biến) ta 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, ta viết biểu thức thành dạng sau:

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

$$\begin{array}{ccccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ (0 & 0 & 0)_2 & (0 & 0 & 1)_2 & (0 & 1 & 0)_2 \\ 0_{10} & & 1_{10} & & 2_{10} & & & \end{array}$$

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

Nhận xét:

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 ta 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).

1.2.3 Phương pháp bảng Các ô (bảng Karnaugh hay phương pháp hình học.)

Tổ chức của bảng Các ô:

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), ta 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 1-4.

		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 $\overline{A}B C\overline{D}$
	01	m_4 $\overline{A}B\overline{C}D$	m_5 $\overline{A}B C D$	m_7 $A\overline{B}\overline{C}\overline{D}$	m_6 $A\overline{B} C\overline{D}$
	11	m_{12} $AB\overline{C}\overline{D}$	m_{13} $AB\overline{C}D$	m_{15} $AB C\overline{D}$	m_{14} $AB C D$
	10	m_8 $A\overline{B}\overline{C}\overline{D}$	m_9 $A\overline{B} C\overline{D}$	m_{11} $AB\overline{C}\overline{D}$	m_{10} $AB C\overline{D}$

		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 $\overline{A}B C$
	1	m_4 $A\overline{B}\overline{C}$	m_5 $A\overline{B}C$	m_7 $AB\overline{C}$	m_6 $AB C$

		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 $\overline{A}\overline{B}C D\overline{E}$	m_6 $\overline{A}B\overline{C}\overline{D}\overline{E}$	m_7 $\overline{A}B\overline{C}D\overline{E}$	m_5 $\overline{A}B C\overline{D}\overline{E}$	m_4 $\overline{A}B C D\overline{E}$
	01	m_8 $\overline{A}B\overline{C}\overline{D}E$	m_9 $\overline{A}B\overline{C}D E$	m_{11} $\overline{A}B C\overline{D}E$	m_{10} $\overline{A}B C D E$	m_{14} $\overline{A}B\overline{C}\overline{D}E$	m_{15} $\overline{A}B\overline{C}D E$	m_{13} $\overline{A}B C\overline{D}E$	m_{12} $\overline{A}B C D E$
	11	m_{24} $AB\overline{C}\overline{D}\overline{E}$	m_{25} $AB\overline{C}D\overline{E}$	m_{27} $AB C\overline{D}\overline{E}$	m_{26} $AB C D\overline{E}$	m_{30} $AB\overline{C}\overline{D}E$	m_{31} $AB\overline{C}D E$	m_{29} $AB C\overline{D}E$	m_{28} $AB C D E$
	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} $A\overline{B}C D\overline{E}$	m_{22} $A\overline{B}\overline{C}\overline{D}E$	m_{23} $A\overline{B}\overline{C}D E$	m_{21} $A\overline{B}C\overline{D}E$	m_{20} $A\overline{B}C D E$

Bảng 1-4. 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 \ AB	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	0	0	0	1

Bảng 1-5. 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.

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

Trong thực tế, khi viết một hàm logic dưới dạng nào đó, thì dạng có được không phải là dạng duy nhất. Thông thường nếu biểu thức càng đơn giản thì mạch điện cũng đơn giản. Khi thiết kế mạch phải đảm bảo sao cho các phần tử trong mạch phải tối thiểu nhất để tiết kiệm chi phí, do vậy, người thiết kế phải sử dụng các phương pháp để tối thiểu hóa hàm logic.

Có ba phương pháp tối thiểu hoá. Nếu số biến số tương đối ít ($n \leq 6$) khi đó dùng phương pháp hình vẽ, phương pháp này dùng bảng Các nô. Nếu số biến tương đối nhiều dùng phương pháp đại số hoặc dùng phương pháp Mc. Quine Cluskey.

1.3.1. 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, $A + \bar{A} = 1$, $A + A.B = A$ ta có:

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

Nhận xét:

Từ ví dụ trên ta 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.

1.3.2 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 kế 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 ta 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' ta 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, ta có hàm đã tối giản.

Đối với Maxterm:

1. Gộp các ô kế 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' ta 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, ta 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:

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

$\bar{A}C$ (grouping 01 and 11 in row 0)
 \bar{B} (grouping 00 and 01 in column 0)

Bảng 1-6. Bảng Các nô

+ Xây dựng bảng Các nô 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 1-6)

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

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

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

Rút gọn theo Maxterm

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

$\bar{B} + C$ (gộp hàng A=0)

$\bar{A} + \bar{B}$ (gộp cột B=1)

Bảng 1-7. Bảng Các nô

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

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

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

Bảng 1-8 trình bày một số cách gộp và giá trị của hàm theo minterm.

a) $B.\bar{D}$

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

b) $\bar{A}.B$

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

c) $\bar{B}.\bar{D}$

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

d) \bar{B}

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

e) \bar{C}

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

f) $\bar{A}.\bar{B} + A.B$

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

Bảng 1-8. 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 1-9. Ta 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.

AB \ CD	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 1-9

Sau khi rút gọn ta có biểu thức hàm như sau:

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

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 ta có thể biểu diễn được mã Hexa, nhưng các ký hiệu $A_{16}(1010)$, $B_{16}(1011)$, $C_{16}(1100)$, $D_{16}(1101)$, $E_{16}(1110)$, $E_{16}(1111)$ 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:

$$f(A, B, C, \dots) = \sum m_i + \sum_d m_j ; \quad (1.3)$$

Σ_d là ký hiệu của điều kiện tùy chọn;

Dạng chuẩn Maxterm:

$$F(A, B, C, \dots) = \prod M_i \prod_d M_j \quad (1.4)$$

\prod_d là ký hiệu của điều kiện tùy chọn;

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

Lập bảng Các nô 1-10.

Từ đó, tìm được hàm tối giản sau:

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

CD \ AB	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 1-10. Bảng Các nô tìm hàm F

1.4 CÔNG LOGIC VÀ CÁC THAM SỐ CHÍNH

Cổng logic cơ sở là mạch điện thực hiện ba phép tính cơ bản trong đại số logic, vậy sẽ có ba loại cổng logic cơ sở là AND, OR và NOT.

1.4.1 Cổng logic cơ bản

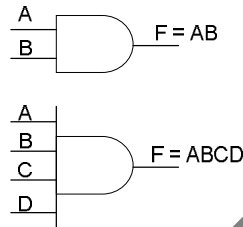
1.4.1.1 Cổng AND

Cổng AND thực hiện hàm logic

$$f = f(A, B) = A.B = AB \quad (1.5)$$

hoặc nhiều biến:

$$f(A, B, C, D, \dots) = A.B.C.D\dots = ABCD\dots \quad (1.6)$$



Hình 1-2. Ký hiệu cổng AND - theo tiêu chuẩn ANSI (American National Standards Institute), Viện tiêu chuẩn Quốc gia Hoa kỳ

Bảng trạng thái 1-11a, 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 1-11a, 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 1-11b). Cổng AND có n đầu vào sẽ có 2^n hạng tích (dòng) trong bảng trạng thái.

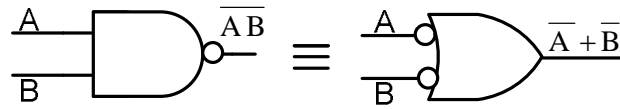
Thông qua định lý DeMorgan ta có thể biến đổi hàm ra của cổng NAND để tìm sự tương đương giữa cổng NAND và cổng OR

$$f = \overline{A.B} = \overline{A} + \overline{B} \quad (1.7)$$

hoặc đối với cổng nhiều đầu vào

$$f = \overline{A.B.C.D\dots} = \overline{A} + \overline{B} + \overline{C} + \overline{D} + \dots \quad (1.8)$$

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 1-3 trình bày sự tương đương trên.



Hình 1-3. Sơ đồ tương đương giữa cổng NAND và cổng OR logic âm

1.4.1.2. Cổng OR

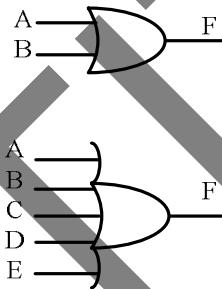
Cổng OR thực hiện hàm logic:

$$f(A, B) = A + B \quad (1.9)$$

hoặc với hàm nhiều biến:

$$f(A, B, C, D, \dots) = A + B + C + D + \dots \quad (1.10)$$

Ký hiệu của cổng OR được biểu diễn ở hình 1-4.



Hình 1-4. 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 1-12a, 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

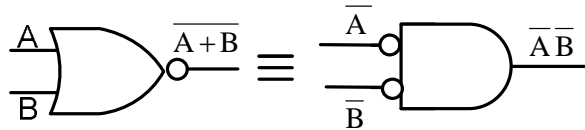
Bảng 1-12 a, b. Bảng trạng thái của cổng OR.

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

$$f = \overline{A + B} = \overline{A} \cdot \overline{B} \quad (1.11)$$

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 1-5 mô tả sự tương đương đã trình bày trên đây:



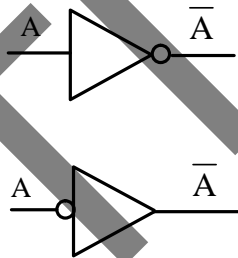
Hình 1-5. Sơ đồ tương đương giữa cổng NOR và cổng AND

1.4.1.3. Cổng NOT

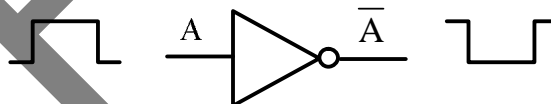
Cổng NOT thực hiện hàm logic:

$$f = \overline{A} \quad (1.12)$$

Ký hiệu của cổng NOT được chỉ ra trên hình 1-6.



Hình 1-6. Ký hiệu của cổng NOT theo tiêu chuẩn ANSI.



Hình 1-7. Nguyên lý hoạt động của cổng NOT

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

Hoạt động của cổng NOT được tóm tắt ở bảng 1-13a, b.

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

a) Theo giá trị logic

b) Theo mức logic

Bảng 1-13a, b. Bảng trạng thái của cổng NOT.

1.4.2. 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.

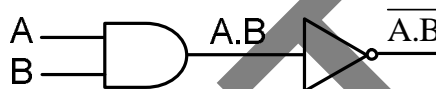
Logic âm là đảo của logic dương. Trong logic dương mức 1 có điện thế cao hơn mức 0. Đối với logic âm, ngược lại mức 0 có điện thế cao hơn mức 1.

1.4.3. 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.

1.4.3.1. Cổng NAND

Ghép nối tiếp một cổng AND với một cổng NOT ta được cổng NAND (Hình 1-8).

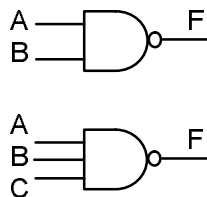


Hình 1-8. Sơ đồ cấu tạo cổng NAND

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

$$\begin{aligned} f &= \overline{A.B} = \overline{AB} \\ f &= \overline{A.B.C.D...} = \overline{ABCD...} \end{aligned} \quad (1.13)$$

Ký hiệu cổng NAND (hình 1-9) và bảng trạng thái (bảng 3-4).



Hình 1-9. 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

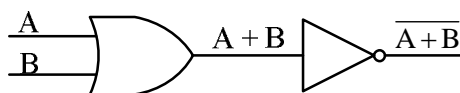
Bảng 1-14a,b. Bảng trạng thái của cổng NAND

1.4.3.2 Cổng NOR

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

Từ hình 1-10, có thể viết được hàm ra của cổng NOR 2 và nhiều đầu vào như sau:

$$f = \overline{A + B} \quad \text{hay} \quad f = \overline{A + B + C + \dots} \quad (1.14)$$



Hình 1-10. Sơ đồ cấu tạo cổng NOR

Ký hiệu của cổng NOR 2 đầu vào như chỉ ở hình 1-11.



Hình 1-11. Sơ đồ cấu tạo cổng NOR theo tiêu chuẩn ANSI.

Hoạt động của cổng NOR được giải thích bằng bảng trạng thái như chỉ ở bảng 1-15a,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

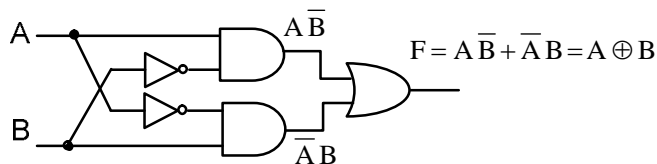
Bảng 1-15a, b. Bảng trạng thái của cổng NOR 2 đầu vào.

1.4.3.3 Cổng XOR

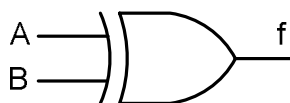
Cổng hoặc tuyệt đối (Exclusive OR) còn có một số tên gọi khác, cổng khác dấu, cổng cộng modun 2. Đây là một loại cổng ghép phức tạp hơn NAND và NOR. Biểu thức logic đầu ra của cổng là:

$$f = A.\bar{B} + \bar{A}.B = A \oplus B \quad (1.15)$$

Từ biểu thức đầu của (1.15) có mạch logic của cổng XOR như hình 1-12.



Hình 1-12 Sơ đồ của cổng XOR 2 đầu vào



Hình 1-13. Ký hiệu của cổng XOR 2 đầu vào theo tiêu chuẩn ANSI

Phần 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 1-13.

Bảng 1-16 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

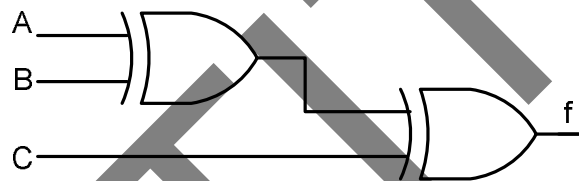
a) Bảng trạng thái

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

b) Bảng chức năng

Bảng 1-16. 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 1-14.



Hình 1-14. 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:

$$A \oplus B = B \oplus A \quad (1.16)$$

2. Luật kết hợp:

$$(A \oplus B) \oplus C = A \oplus (B \oplus C) \quad (1.17)$$

3. Luật phân phối:

$$A(B \oplus C) = A.B \oplus A.C \quad (1.18)$$

4. Các phép toán của biến và hằng số:

$$A \oplus 1 = \overline{A} \quad (1.19)$$

$$A \oplus 0 = A \quad (1.20)$$

$$A \oplus A = 0 \quad (1.21)$$

$$A \oplus \overline{A} = 1 \quad (1.22)$$

5. Luật đổi chỗ nhân quả

Nếu $A \oplus B = C$ thì

$$A \oplus C = B \quad \text{và} \quad B \oplus C = A \quad (1.23)$$

1.4.3.4 Cổng XNOR

Cổng XNOR còn gọi là cổng không hoặc tuyệt đối hay cổng đồng dấu.

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

Biểu thức logic đầu ra:

$$f = AB + \overline{A}\overline{B} \quad \text{hay} \quad f = \overline{A \oplus B} = A \sim B \quad (1.24)$$

Bảng 1-17. Bảng trạng thái và trạng thái của cổng XNOR 2 đầu vào

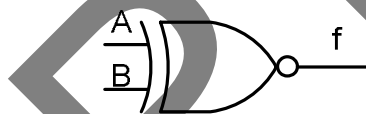
A	B	f
0	0	1
0	1	0
1	0	0
1	1	1

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

a) Bảng trạng thái

b) Bảng chức năng

Ký hiệu của cổng XNOR 2 đầu vào được trình bày trên hình 1-15.



Hình 1-15. 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 1-17.

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

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...

1.4.4 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

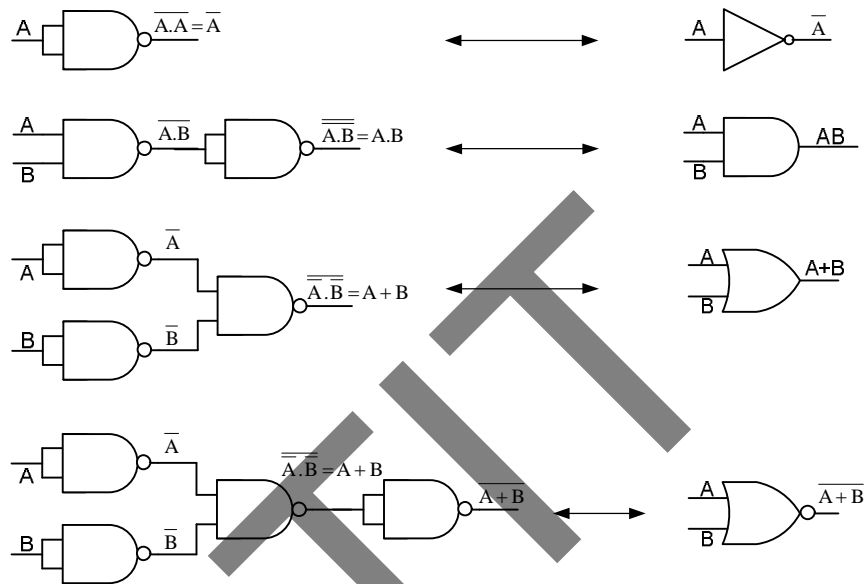
điểm này phải tìm ra cổng có thể tạo ra hệ hàm đầy đủ. Cổng NAND và cổng NOR có thể thỏa mãn điều kiện này.

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:

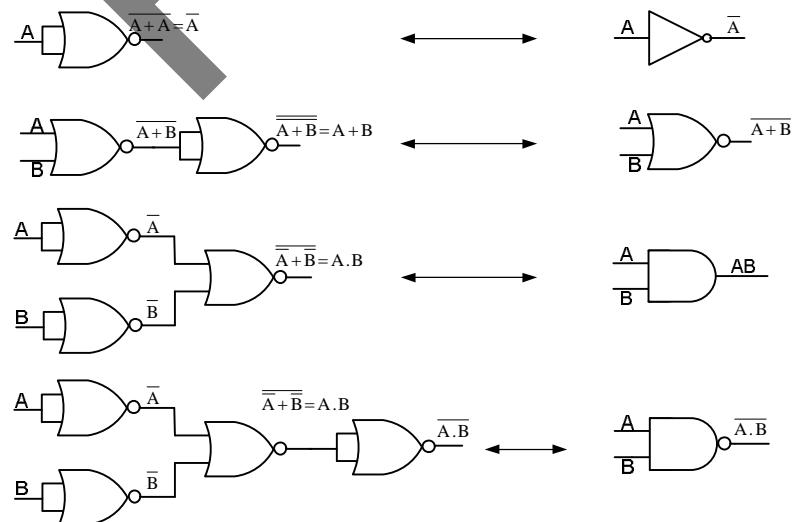
Từ cổng NAND có thể tạo ra các cổng NOT, AND, OR và NOR.

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



Hình 1-16. Tính đa chức năng của cổng NAND

Tính đa chức năng của cổng NOR:



Hình 1-17. Tính đa chức năng của cổng NOR

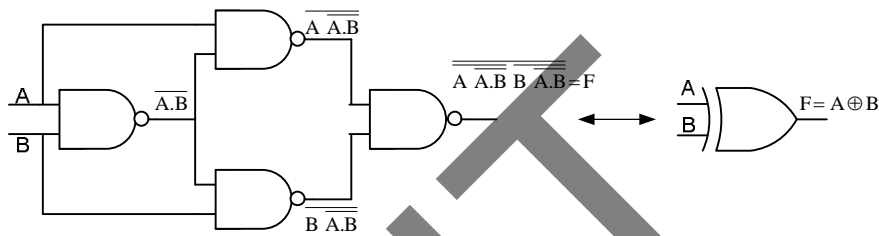
Từ hình 1-16 và 1-17, ta có thể kết luận là mọi mạch logic tổ hợp có thể xây dựng chỉ từ một loại cổng cơ bản là cổng NAND hoặc cổng NOR. Đây là một đặc điểm quan trọng trên quan điểm tính đồng nhất của công nghệ chế tạo, do vậy, giá thành chi phí giảm, độ tin cậy cao.

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

Giải:

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

Từ biểu thức biến đổi trên vẽ được sơ đồ logic hình 1-18.



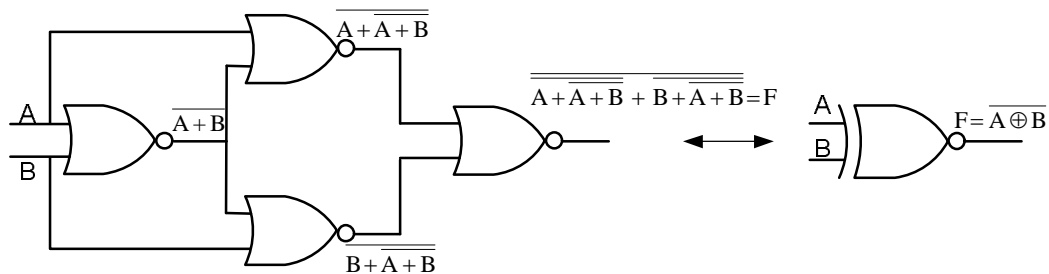
Hình 1-18. Mạch logic thể hiện hàm XOR toàn NAND

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

Giải:

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

Từ biểu thức biến đổi trên, vẽ được sơ đồ logic trên hình 1-19:

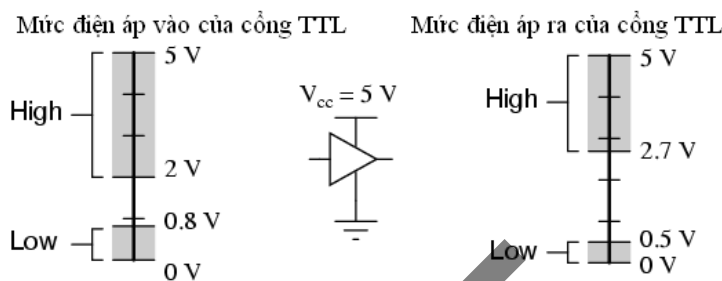


Hình 1-19. Mạch logic thể hiện hàm XNOR toàn NOR

1.5. Các tham số chính

1.5.1 Mức logic (Logic Levels)

Mức logic là mức điện thế trên đầu vào và đầu ra của cổng tương ứng với logic "1" và logic "0", nó phụ thuộc điện thế nguồn nuôi của cổng (V_{CC} đối với họ TTL (Transistor Logic) và V_{DD} đối với họ MOS (Metal Oxide Semiconductor)). Lưu ý rằng, nếu mức logic vào vượt quá điện thế nguồn nuôi có thể gây hư hỏng cho cổng.



Hình 1-20. Mức điện áp vào/ra của cổng TTL

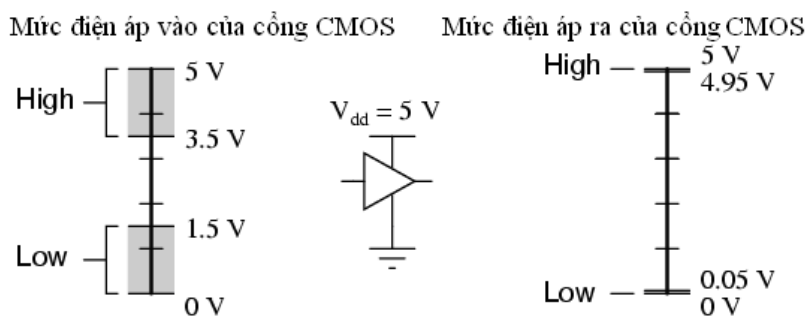
Thông thường giá trị của hai mức logic là giá trị danh định. Thực tế, giá trị này có thể biến đổi nhỏ do các thông số của các phần tử trong mạch hoặc do nguồn cấp biến đổi hoặc do nhiệt độ...

Mức logic là tham số quan trọng của cổng, có thể xác định được trạng thái logic vào/ra bằng cách dùng vôn kế.

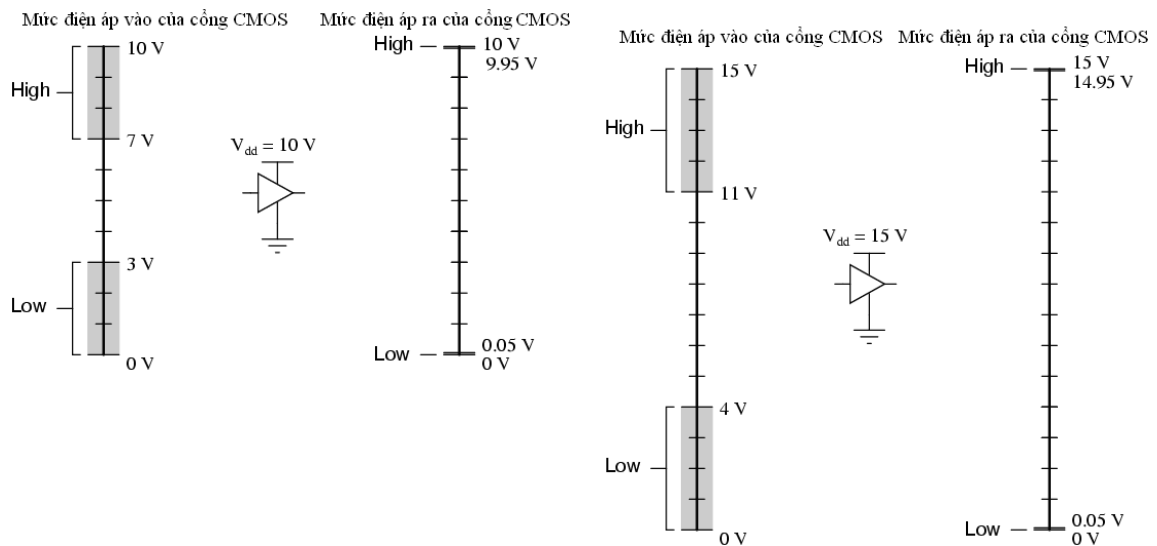
Thông thường các cổng TTL hoạt động ở mức nguồn nuôi là $5\text{ V} \pm 0,25\text{ V}$. Lý tưởng, TTL có mức tín hiệu cao là 5V; thấp là 0V. Tuy nhiên, trên thực tế các cổng TTL không đạt được điện áp lý tưởng đó và nó được thiết kế để chấp nhận một dải điện áp tương ứng với mức H (high) và L (low). Mức L ứng với điện áp từ $0\text{ V} \div 0,8\text{ V}$ và mức H ứng với điện áp từ $2\text{ V} \div 5\text{ V}$

Nếu điện áp của tín hiệu nằm trong dải từ 0,8V đến 2V thì tín hiệu ra sẽ không chắc chắn đúng (có thể ở mức H có thể ở mức L). Do vậy khi làm việc với mạch phải chắc chắn rằng không để khoảng điện áp này đưa đến đầu vào. (Hình 1-20).

Hình 1-21 mô tả mức điện áp vào/ra của họ cổng CMOS.



Hình 1-21. Mức điện áp vào/ra của cổng CMOS



Hình 1-22. Mức điện áp vào/ra của cổng CMOS trong trường hợp 10V và 15V

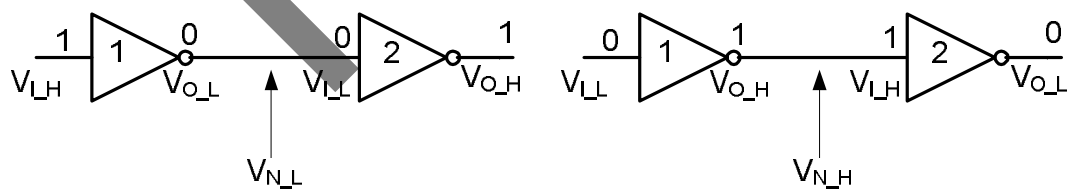
Không giống như TTL, mức nguồn nuôi được đặt cố định là 5V, CMOS có nguồn cung cấp cao hơn: 15V (một số mạch có nguồn là 18V). Do vậy, hình vẽ 1-22 giới thiệu mức điện áp vào/ra của CMOS ứng với trường hợp nguồn 10V và 15V.

1.5.2. Độ chống nhiễu (Noise Immunity)

Độ chống nhiễu (hay độ phòng vệ nhiễu) là mức nhiễu lớn nhất tác động tới đầu vào hoặc đầu ra của cổng mà chưa làm thay đổi trạng thái vốn có của nó.

Độ chống nhiễu là tiêu chuẩn đánh giá độ nhạy của mạch logic đối với tạp âm xung trên đầu vào vi mạch. Độ chống nhiễu khi đầu ra ở mức logic 0 và 1 là khác nhau.

Nếu gọi độ chống nhiễu khi đầu ra ở mức logic 1 là V_{N_H} (Noise_High); độ chống nhiễu khi đầu ra ở mức logic 0 là V_{N_L} (Noise_Low). Điện áp vào/ra tương ứng là V_I (In); V_O (Out) thì có thể phân ra hai trường hợp:



a) Tác động nhiễu khi mức ra ở trạng thái thấp b) Tác động nhiễu khi mức ra ở trạng thái cao

Hình 1-23a, b. Mô tả tác động nhiễu đến các cổng logic

+ Nhiễu mức cao: trong trường hợp các cổng vẫn hoạt động bình thường: nếu đầu ra cổng 1 lấy logic H (hình 1-23a) thì đầu ra cổng 2 phải có logic L. Khi tính tới tác động của nhiễu, ta có:

$$\begin{aligned}
 V_{O_Hmin} + V_{N_H} &\geq V_{I_Hmin} \\
 \Leftrightarrow V_{N_H} &\geq V_{I_Hmin} - V_{O_Hmin} \\
 \Leftrightarrow V_{N_H} &\geq V_{I_Hmin} - V_{ON}
 \end{aligned}
 \tag{1.25}$$

Với cổng TTL: $V_{N_H} \geq 2V - 2,7V = -0,7V$

Với cổng CMOS: $V_{N_H} \geq 3,5V - 4,95V = -1,45V$

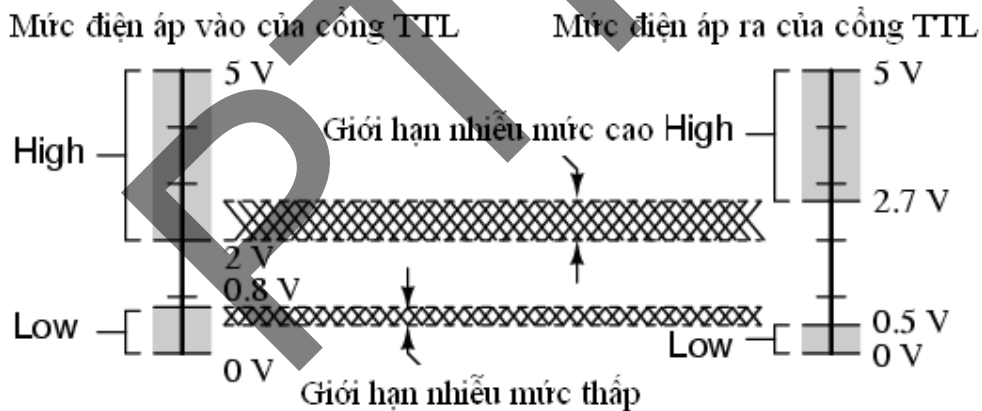
+ Nhiều mức thấp: đầu ra cổng 1 lấy logic L (hình 1-23b), tương tự có:

$$\begin{aligned}
 V_{O_Lmax} + V_{N_L} &\leq V_{I_Lmax} \\
 \Leftrightarrow V_{N_L} &\leq V_{I_Lmax} - V_{O_Lmax} \\
 \Leftrightarrow V_{N_L} &\leq V_{I_Lmax} - V_{OFF}
 \end{aligned}
 \tag{1.26}$$

Với cổng TTL: $V_{NL} \leq 0,8V - 0,5V = 0,3V$

Với cổng CMOS: $V_{NL} \leq 1,5V - 0,05V = 1,45V$

Qua sự tính toán trên có thể biểu diễn độ chống nhiễu bằng hình vẽ 1-24. Thấy rằng sự khác biệt giữa dải điện áp vào và ra chính là dải nhiễu của cổng. Ví dụ, đối với cổng TTL thì mức nhiễu thấp là sự sai khác giữa 0.8V và 0.5V bằng 0.3V, trong khi mức nhiễu cao là sự sai khác giữa 2.7V và 2V bằng 0.7V. Tương tự đối với cổng CMOS.



Hình 1-24. Độ chống nhiễu của họ cổng TTL

1.5.3. Khả năng mắc tải vào, ra (Fan in, Fan out)

Hệ số mắc tải cho biết khả năng nối được bao nhiêu đầu vào tới đầu ra của một cổng đã cho mà vẫn đảm bảo sự hoạt động tin cậy, đảm bảo tốc độ, giới hạn về nhiệt độ và các tham số khác.

Hệ số mắc tải phụ thuộc dòng ra (hay dòng phun) của cổng chịu tải và dòng vào (hay dòng hút) của các cổng tải ở cả hai trạng thái H, L.

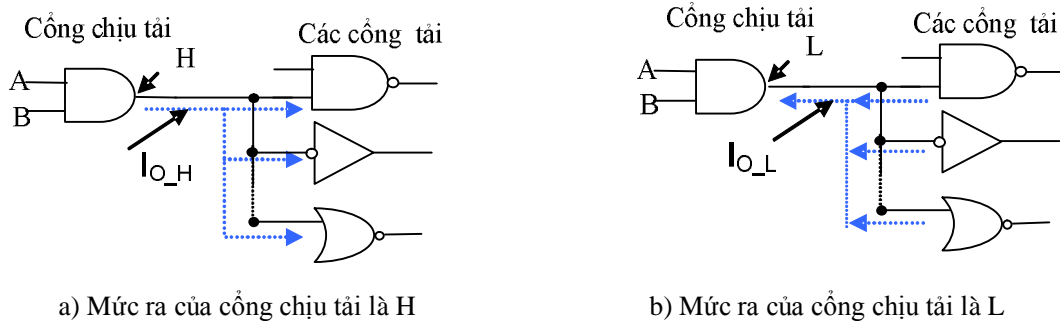
Do hạn chế bởi công suất ra tối đa của mạch, số lượng tải có thể nối với mạch logic là có hạn.

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

Đất chung: Các tải được nối với đầu ra của mạch và đất.

Nguồn chung: Các tải được nối với đầu ra của mạch và nguồn cung cấp.

Hình 1-25 mô tả về cách nối tải.



Hình 1-25a,b. Mô tả về hệ số mắc tải.

1.5.4. Công suất (Power)

Đối với các cổng logic có hai loại công suất chính: công suất tiêu tán và công suất điều khiển.

Công suất tiêu tán.

Đây là tiêu chuẩn để đánh giá lượng công suất tiêu thụ (tổn hao) trên các phần tử trong vi mạch. Công suất tiêu hao thường cỡ vài mW đối với một vi mạch số và là giá trị trung bình giữa công suất tiêu tán khi đầu ra ở mức 0, 1 (Các công suất này thường khác nhau).

Công suất tiêu tán càng nhỏ càng tốt và có ý nghĩa đặc biệt quan trọng trong các thiết bị xách tay hay các thiết bị dùng pin.

Công suất điều khiển.

Công suất điều khiển là công suất của tín hiệu điều khiển ở đầu vào sao cho mạch vẫn hoạt động tốt. Công suất điều khiển càng nhỏ càng tốt.

1.5.5. Trễ truyền đạt (Propagation Delay)

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

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

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

Có 2 loại trễ truyền đạt: Trễ xảy ra khi đầu ra thay đổi từ mức cao (High) xuống mức thấp (Low) và ngược lại.

Do cấu tạo của mạch logic, trễ giữa hai loại chuyển biến thường khác nhau. Chúng giống nhau về mức và gần nhau về giá trị nhưng không tương đương.

Độ rộng sườn trước t_{PHL} và độ rộng sườn sau t_{PLH} là khoảng thời gian để biên độ xung thay đổi trong khoảng từ 10% đến 90% giá trị biên độ cực đại.

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

Trễ truyền đạt trung bình được tính theo công thức:

$$t_{pd} = \frac{t_{PHL} + t_{PLH}}{2} \quad (1.27)$$

Khi mắc nối tiếp nhiều cổng logic thì trễ truyền đạt của toàn mạch sẽ bằng tổng trễ truyền đạt của mỗi cổng.

1.6. MỘT SỐ LƯU Ý KHI SỬ DỤNG IC SỐ

1.6.1. Sơ đồ chân và ký hiệu trên thân IC.

Với họ TTL:

LS: tốc độ cao dùng diode Schottky.

AS: tốc độ siêu cao dùng diode Schottky.

ALS: tốc độ cao, công suất thấp.

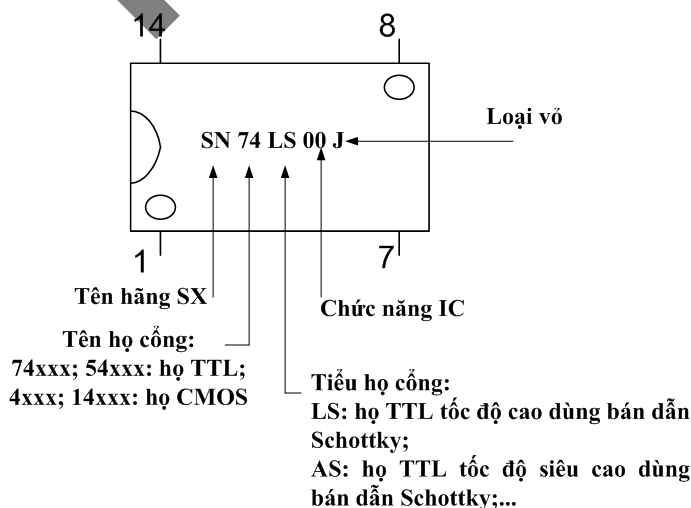
F: tốc độ siêu cao, thời gian trễ cực nhỏ.

Với họ CMOS:

HC: tốc độ cao (gấp 10 lần họ LS).

ACL khả năng chống nhiễu tốt, tốc độ nhịp cao, trễ nhỏ.

AHC: tốc độ cực cao (gấp 3 lần HC), công suất, dòng tiêu thụ nhỏ.



1.6.2. Một số đặc điểm của IC họ TTL và CMOS.

Đối với IC họ TTL

+ Tất cả các đầu vào của cổng TTL để hở sẽ hoạt động như mức logic 1 (do diode base-emitter của transistor của mạch vào không được phân cực thuận). Trường hợp này gọi là thả nổi đầu vào.

+ Khi không sử dụng một đầu vào nào đó của cổng thì phải nối nó với đất hoặc dương nguồn sao cho chức năng của cổng không bị thay đổi.

+ Không được nối trực tiếp hai đầu ra của hai cổng TTL với nhau. Trong trường hợp này phải sử dụng cổng collector để hở -OC (Open Collector).

Đối với IC họ CMOS.

+ Không được phép thả nổi các đầu vào không được sử dụng đến mà phải nối chúng với đất hoặc dương nguồn hoặc đầu khác sao cho chức năng của cổng không bị thay đổi.

+ Điện trở đầu vào cao gây nên hiện tượng tích tụ hạt tĩnh điện, dẫn đến phát sinh điện thế có thể đủ lớn để đánh thủng lớp điện môi mỏng giữa cực G và kênh dẫn. Do vậy người ta chế tạo lưới diode điện trở ở đầu vào nhằm bảo vệ transistor.

+ Điện trở đầu ra thường nhỏ nên tốc độ chuyển mạch tương đối nhanh.

1.6.3. Xử lý cổng thừa, lối vào thừa.

Xử lý cổng thừa:

Nối các lối vào của cổng thừa với đất hoặc dương nguồn sao cho lối ra có mức logic bằng 1. Vì lúc này công suất tiêu thụ của cổng đạt giá trị nhỏ nhất.

Xử lý lối vào thừa.

Nối các lối vào thừa với đất hoặc dương nguồn sao cho tính chất của cổng không bị thay đổi hoặc có thể nối chân thừa với một trong các chân đang sử dụng.

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.

Đồng thời chương này đã trình bày cấu trúc, nguyên lý và đặc điểm của cổng 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.

Có 2 loại vi mạch số phổ biến nhất : TTL và MOS. TTL là công nghệ điển hình trong nhóm công nghệ transistor bao gồm TTL, HTL, ECL..., MOS là công nghệ vi mạch sử dụng MOSFET, trong đó điển hình là MOS...

CÂU HỎI ÔN TẬP

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

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

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

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

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

- 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 Các nô?

- 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) $\overline{CD + \overline{CD}} \cdot \overline{AC + D}$

b) $\overline{\overline{ABC}} \cdot \overline{\overline{AB} + \overline{BC} + \overline{CA}}$

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. $\overline{A \oplus B} = \overline{A} \cdot \overline{B} + AB$

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

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

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

d) $A + AB$

10. Chứng minh đẳng thức

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

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) $\overline{CD + \overline{CD}} \cdot \overline{AC + D}$

c) $\overline{\overline{ABC}} \cdot \overline{\overline{AB} + \overline{BC} + \overline{CA}}$

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

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

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

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

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

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

16. Xây dựng bảng trạng thái cho hàm sau?

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

17. Cho hàm 3 biến :

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

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

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

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

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

18. Cho hàm logic sau:

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

a) Lập bảng Cárnô và rút gọn hàm F?

b) Lập bảng Cárnô và rút gọn hàm \overline{F} ?

19. Cho hàm logic sau:

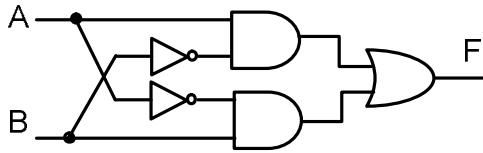
$$F(A, B, C, D) = \overline{(A \oplus B) \cdot (C + D)}$$

a) Đưa hàm F về dạng chuẩn tắc

b) Lập bảng Cárnô và rút gọn hàm F?

c) Lập bảng Cárnô và rút gọn hàm \overline{F} ?

20. 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?

21. Cho hàm logic:

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

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

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

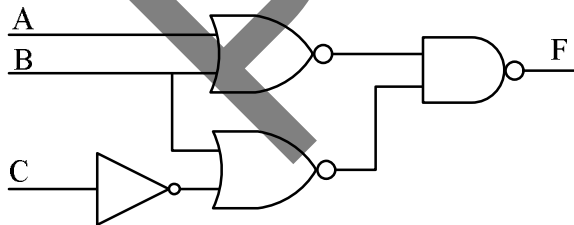
- a) $A \oplus B = B \oplus A$
- b) $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
- c) $A(B \oplus C) = A.B \oplus A.C$
- d) Nếu $A \oplus B = C$ thì $A \oplus C = B$ và $B \oplus C = A$

23. + Hãy chứng minh:

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

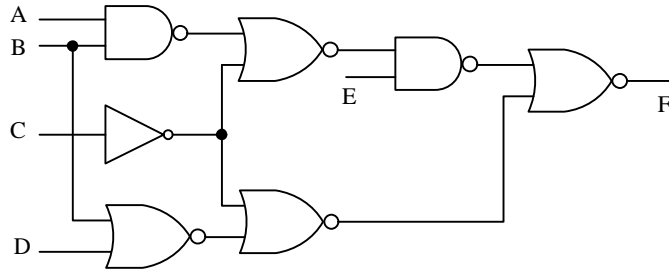
+ Hãy vẽ sơ đồ logic của hàm $F = A \oplus B \oplus C \oplus D$

24. Cho hình vẽ sau

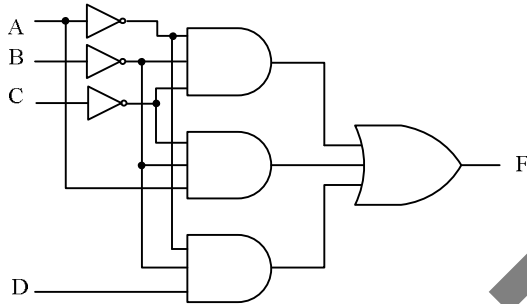


- a) Viết biểu thức hàm ra F.
 - b) Xây dựng bảng trạng thái.
 - c) Tối ưu hóa mạch.
25. Cho hàm logic $F = A.B + B.C + A.C$
- a) Viết lại biểu thức F theo cấu trúc toàn NAND.
 - b) Viết lại biểu thức F theo cấu trúc toàn NOR.
 - c) Vẽ mạch logic hàm F theo cấu trúc toàn NAND và toàn NOR.

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



27. Cho hình vẽ sau:



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

CHƯƠNG 2: 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*. Chương 2 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 đó được xét đến ở chương 1. 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.
- Hợp 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.
- Các phép so sánh số 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.

2.1 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).

2.1.1. 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.

2.1.2. 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.

2.2 PHÂN TÍCH MẠCH LOGIC TỔ HỢP

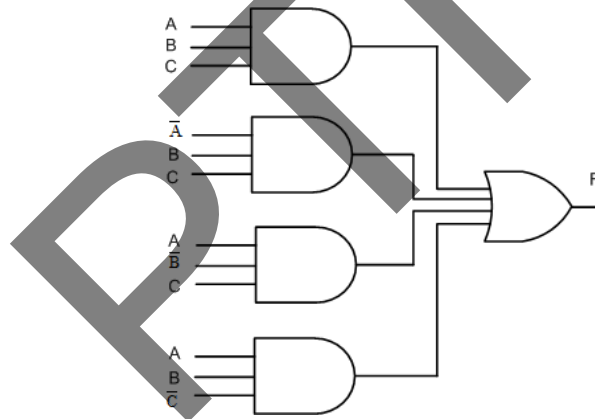
Phân tích mạch logic tổ hợp là đánh giá, phê phán một mạch đó. Trên cơ sở đó, có thể rút gọn, chuyển đổi dạng thực hiện của mạch điện để có được lời giải tối ưu theo một nghĩa nào đấy.

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:



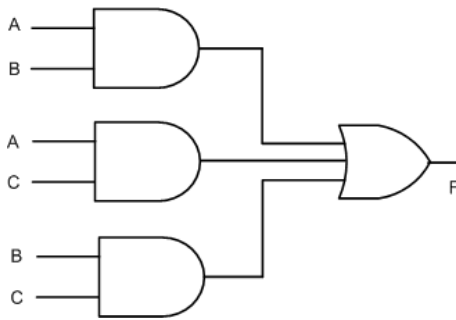
Hình 2. 1. Sơ đồ mạch logic

Viết biểu thức hàm và thực hiện rút gọn:

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

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

Từ đó vẽ được mạch sau:

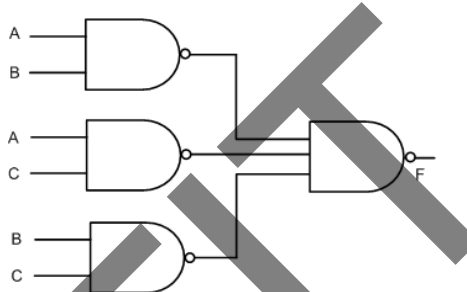


Hình 2. 2. Sơ đồ mạch logic sau khi rút gọn

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:

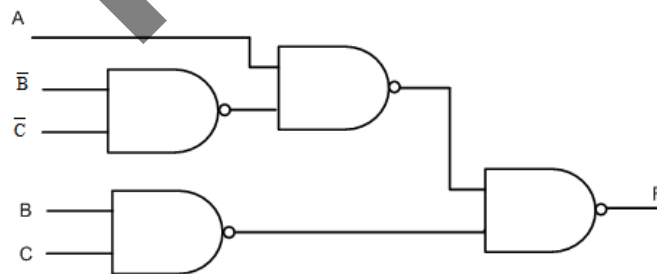


Hình 2. 3. Sơ đồ mạch logic sau khi tối ưu.

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:

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

Từ đó vẽ được mạch sau:



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

2.3 THIẾT KẾ MẠCH LOGIC TỔ HỢP

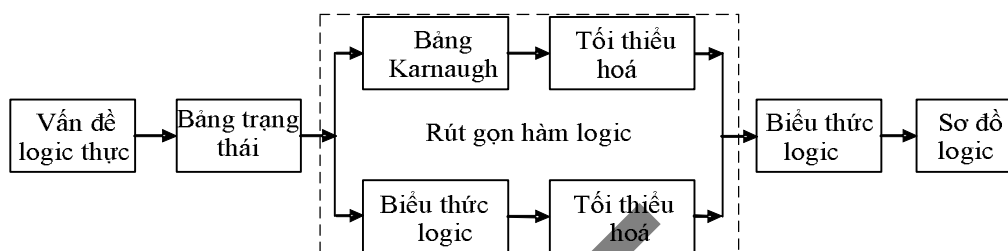
Quá trình thiết kế nói chung của mạch tổ hợp gồm các bước được mô tả trên hình 2-5.

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 2. 5. Các bước thiết kế mạch logic tổ hợp

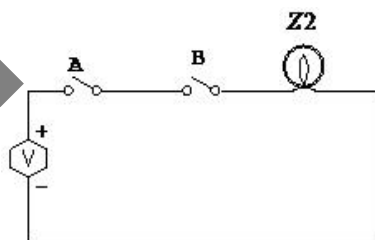
+ Bảng trạng thái:

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 có bảng trạng thái.

Từ một bảng chức năng có thể được các bảng trạng thái khác nhau, nếu thay giá trị logic khác nhau.

Ví dụ: Sơ đồ nguyên lý mạch điện hình 2-6.



Hình 2. 6. Mạch điện 2 chuyển mạch mắc nối tiếp với bóng điện

Chuyển mạch A	Chuyển mạch B	Bóng đèn F
Ngắt	Ngắt	Tắt
Ngắt	Đóng	Tắt
Đóng	Ngắt	Tắt
Đóng	Đóng	Sáng

Bảng 2-1. Bảng chức năng (trạng thái)

+ Biểu thức logic: Từ bảng chức năng 2-1 nếu thay đổi giá trị logic theo 4 cách khác nhau sẽ được các biểu thức logic khác nhau.

Trường hợp a: 0 biểu thị ngắt và tắt ; 1 biểu thị đóng và sáng

Từ đó, lập được bảng trạng thái 2-2.

Bảng 2-2. Bảng trạng thái.

Từ bảng trạng thái 2-2 suy ra biểu thức logic:

$$F = A.B$$

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

Trường hợp b: 0 biểu thị đóng và sáng

1 biểu thị ngắt và tắt

Biểu thức logic: $F = A + B$

Trường hợp c: 0 biểu thị ngắt và sáng

1 biểu thị đóng và tắt

Biểu thức logic: $F = \overline{A.B}$

Trường hợp d: 0 biểu thị đóng và sáng

1 biểu thị ngắt và tắt

Biểu thức logic: $F = \overline{A + B}$

Khi liệt kê bảng chức năng hoặc bảng trạng thái, có thể không liệt kê các tổ hợp trạng thái tín hiệu đầu vào không thể có hay bị cấm. Những tổ hợp này cũng có thể được liệt kê, nhưng tại đầu ra, ở trạng thái tương ứng ghi dấu chéo “X”, thường sử dụng các trạng thái đánh dấu chéo để tối thiểu hoá hàm logic.

+ Tối thiểu hàm logic:

Thiết kế sơ đồ mạch logic trực tiếp từ hàm số có được từ bảng trạng thái thường là rất phức tạp. Còn sau khi đã thực hiện tối thiểu hoá hàm logic, nói chung việc thiết kế thuận lợi hơn, không những chỉ dùng số linh kiện ít hơn, mà còn nâng cao độ tin cậy của mạch logic.

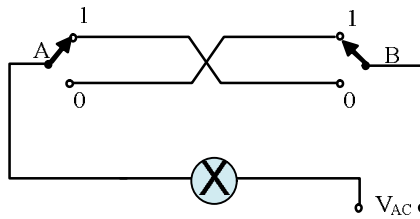
+ Vẽ sơ đồ logic:

Kết quả việc tối thiểu hoá là biểu thức logic OR – AND. Căn cứ vào việc chọn lựa loại cổng logic cụ thể, cần biến đổi biểu thức logic đó thành dạng phù hợp. Ví dụ, nếu chọn dùng cổng NAND phải có biểu thức dạng NAND hoặc dùng cổng NOR, NORAND phải có biểu thức tương ứng.

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:

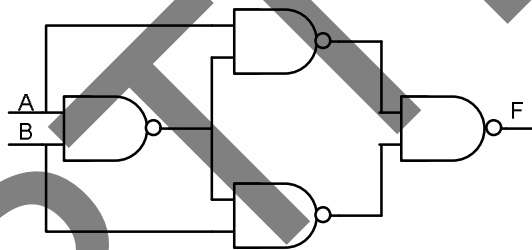
+ Nếu ký hiệu hai công tắc là hai biến A, B. Khi ở tầng 1 bật đèn và lên tầng 2 thì tắt đèn đi và ngược lại. Như vậy đèn chỉ có thể sáng ứng với hai tổ hợp chuyển mạch ở vị trí ngược nhau. Còn đèn tắt khi ở vị trí giống nhau. Hệ thống chiếu sáng trong có sơ đồ như hình 2-7.



Hình 2. 7. Mạch điện của hệ chiếu sáng

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

Bảng 2-3. Bảng trạng thái mô tả hoạt động của hệ chiếu sáng



Hình 2. 8. Sơ đồ logic thể hiện hàm F

Bảng trạng thái mô tả hoạt động của hệ như chỉ ở bảng 2-3.

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

hoặc

$$F = \overline{\overline{\bar{A}B} \cdot \overline{AB}}$$

Đây là hàm XOR. Hàm này có thể được thể hiện bằng nhiều kiểu mạch khác nhau. Hình 2.8 là một dạng sơ đồ thể hiện hàm F.

2.4. MẠCH MÃ HOÁ VÀ GIẢI MÃ

2.4.1 Một số loại mã nhị phân thông dụng

2.4.1.1. 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.

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 2-4. Cấu tạo mã BCD với các trọng số khác nhau.

Ngoài mã N-BCD còn có rất nhiều loại mã BCD với các trọng số khác nhau. Bảng 2-4 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.

Mã BCD 7421.

Mã BCD 5121.

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.

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 vì tổ hợp này dễ dàng được tạo thành do các nhiễu gây ra nên dễ xuất hiện lỗi kỹ thuật.

2.4.1.2. Các dạng mã nhị phân khác

Số thập phân	Số nhị phân	Mã dư 3	Mã Gray	Mã Gray Dư 3	Mã Johnson	Mã vòng
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 0	0 0 0 0 0	0 0 0 0 0 0 0 0 1
1	0 0 0 1	0 1 0 0	0 0 0 1	0 1 1 0	1 0 0 0 0	0 0 0 0 0 0 0 0 1 0

Số thập phân	Số nhị phân	Mã dư 3	Mã Gray	Mã Gray Dư 3	Mã Johnson	Mã vòng
2	0 0 1 0	0 1 0 1	0 0 1 1	0 1 1 1	1 1 0 0 0	0 0 0 0 0 0 0 1 0 0
3	0 0 1 1	0 1 1 0	0 0 1 0	0 1 0 1	1 1 1 0 0	0 0 0 0 0 0 1 0 0 0
4	0 1 0 0	0 1 1 1	0 1 1 0	0 1 0 0	1 1 1 1 0	0 0 0 0 0 1 0 0 0 0
5	0 1 0 1	1 0 0 0	0 1 1 1	1 1 0 0	1 1 1 1 1	0 0 0 0 1 0 0 0 0 0
6	0 1 1 0	1 0 0 1	0 1 0 1	1 1 0 1	0 1 1 1 1	0 0 0 1 0 0 0 0 0 0
7	0 1 1 1	1 0 1 0	0 1 0 0	1 1 1 1	0 0 1 1 1	0 0 1 0 0 0 0 0 0 0
8	1 0 0 0	1 0 1 1	1 1 0 0	1 1 1 0	0 0 0 1 1	0 1 0 0 0 0 0 0 0 0
9	1 0 0 1	1 1 0 0	1 1 0 1	1 0 1 0	0 0 0 0 1	1 0 0 0 0 0 0 0 0 0

Bảng 2-5. Cấu tạo của một số mã nhị phân thông dụng.

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.

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.

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.

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.

2.4.2. Các mạch mã hoá:

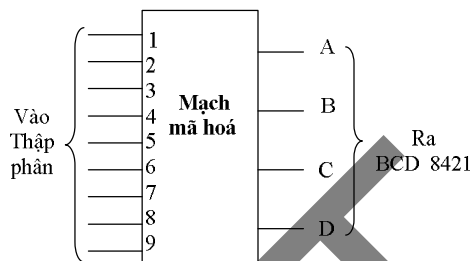
Nói một cách tổng quát, mã hoá là dùng văn tự, ký hiệu hay mã để biểu thị một đối tượng xác định. Các hệ thống điện tử số (như máy tính) chỉ tiếp nhận thông tin dưới dạng nhị phân, mã nhị phân có hai chữ số 0 và 1, vì vậy các số liệu, các lệnh cho máy tính thực hiện đều phải được biểu diễn dưới dạng 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.

2.4.2.1. Bộ mã hoá thập phân sang BCD8421:

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 $n = 4$. Mã nhị phân 4 bit có 16 tổ hợp (từ mã). Chỉ cần chọn 10 từ mã tùy ý trong số đó là đủ biểu thị 10 tín hiệu đầu vào. Vậy có rất nhiều phương án.



Hình 2. 9. Sơ đồ khối mạch mã hóa

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

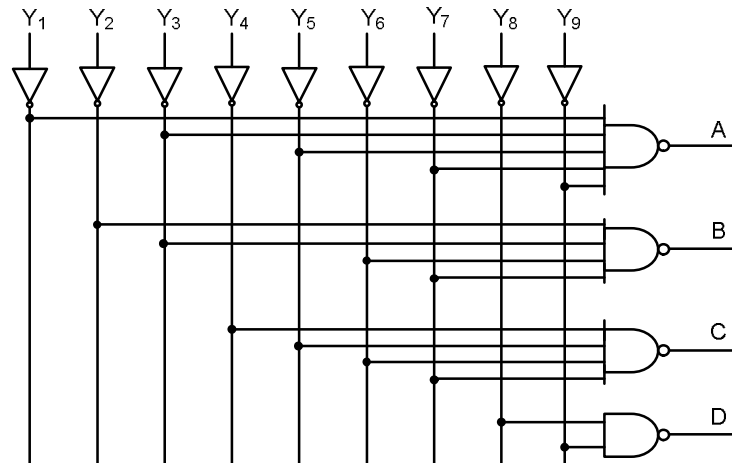
Số thập phân	D	C	B	A
0 (Y_0)	0	0	0	0
1 (Y_1)	0	0	0	1
2 (Y_2)	0	0	1	0
3 (Y_3)	0	0	1	1
4 (Y_4)	0	1	0	0
5 (Y_5)	0	1	0	1
6 (Y_6)	0	1	1	0
7 (Y_7)	0	1	1	1
8 (Y_8)	1	0	0	0
9 (Y_9)	1	0	0	1

Bảng 2-6. Bảng mã hoá BCD – 8421:

Từ bảng trạng thái 2-6, tìm được biểu thức đầu ra sau

$$\left. \begin{aligned}
 D &= Y_8 + Y_9 = \overline{Y_8} \cdot \overline{Y_9} \\
 C &= Y_4 + Y_5 + Y_6 + Y_7 = \overline{Y_4} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7} \\
 B &= Y_2 + Y_3 + Y_6 + Y_7 = \overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_6} \cdot \overline{Y_7} \\
 A &= Y_1 + Y_3 + Y_5 + Y_7 + Y_8 = \overline{Y_1} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_7} \cdot \overline{Y_8}
 \end{aligned} \right\} \quad (2.1)$$

Sơ đồ logic của bộ mã hoá BCD – 8421 được trình bày trên hình 2-6.



Hình 2. 10. Sơ đồ logic của bộ mã hoá nhị - thập phân.

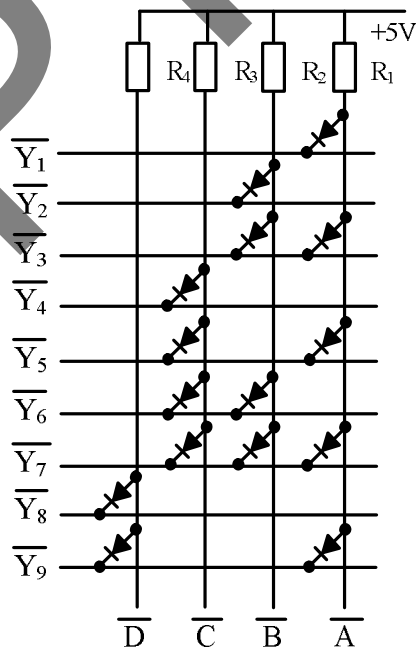
Từ hệ phương trình trên có thể viết lại như sau (dùng định lý DeMorgan) và dùng ma trận diode (cổng AND) để xây dựng mạch:

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

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

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

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



Hình 2. 11. Mạch mã hóa dùng diode

2.4.2.2. 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.

Xem xét nguyên lý hoạt động và quá trình thiết kế bộ mã hoá ưu tiên qua ví dụ sau: 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 2-7 bảng trạng thái bộ mã hoá ưu tiên.

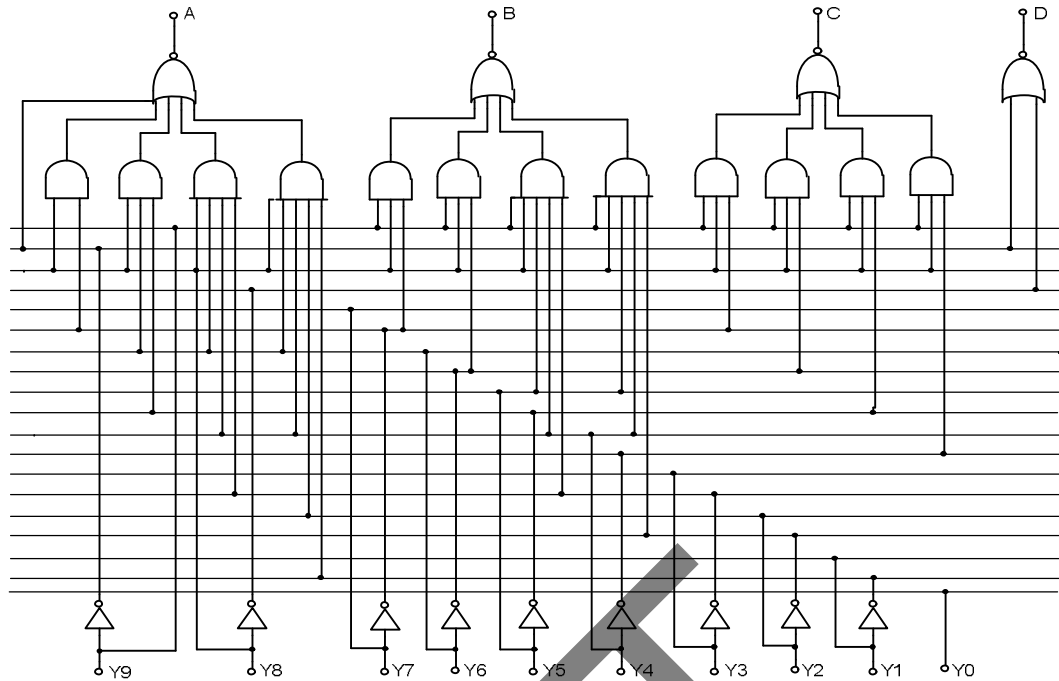
Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	D	C	B	A
1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	0	x	1	1	1	0
1	1	1	1	1	1	1	0	x	x	1	1	0	1
1	1	1	1	1	1	0	x	x	x	1	1	0	0
1	1	1	1	1	0	x	x	x	x	1	0	1	1
1	1	1	1	0	x	x	x	x	x	1	0	1	0
1	1	1	0	x	x	x	x	x	x	1	0	0	1
1	1	0	x	x	x	x	x	x	x	1	0	0	0
1	0	x	x	x	x	x	x	x	x	0	1	1	1
0	x	x	x	x	x	x	x	x	x	0	1	1	0

Bảng 2-7. Bảng trạng thái bộ mã hoá ưu tiên.

$$\left. \begin{aligned}
 \bar{D} &= \bar{Y}_9 + Y_9 \bar{Y}_8 = \bar{Y}_9 + \bar{Y}_8 & \rightarrow D &= \overline{\bar{Y}_9 + \bar{Y}_8} \\
 \bar{C} &= Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 \bar{Y}_6 + Y_9 Y_8 \bar{Y}_5 + Y_9 Y_8 \bar{Y}_4 & \rightarrow C &= \overline{Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 \bar{Y}_6 + Y_9 Y_8 \bar{Y}_5 + Y_9 Y_8 \bar{Y}_4} \\
 \bar{B} &= Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 \bar{Y}_6 + Y_9 Y_8 Y_5 \bar{Y}_4 \bar{Y}_3 + Y_9 Y_8 Y_5 Y_4 \bar{Y}_2 & \rightarrow B &= \overline{Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 \bar{Y}_6 + Y_9 Y_8 Y_5 Y_4 \bar{Y}_3 + Y_9 Y_8 Y_5 Y_4 \bar{Y}_2} \\
 \bar{A} &= \bar{Y}_9 + Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 Y_7 \bar{Y}_6 \bar{Y}_5 + Y_9 Y_8 Y_7 Y_6 \bar{Y}_5 Y_4 \bar{Y}_3 + Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 Y_3 \bar{Y}_2 \bar{Y}_1 & \rightarrow A &= \overline{\bar{Y}_9 + Y_9 Y_8 \bar{Y}_7 + Y_9 Y_8 Y_7 Y_6 \bar{Y}_5 + Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 \bar{Y}_3 + Y_9 Y_8 Y_7 Y_6 Y_5 Y_4 Y_3 \bar{Y}_2 \bar{Y}_1}
 \end{aligned} \right\} (2.2)$$

Theo bảng 2-7, 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 (2.2).

Sơ đồ logic hình 2.12 thoả mãn yêu cầu trên



Hình 2. 12. Bộ mã hoá ưu tiên

2.4.3. 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.

2.4.3.1. 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 2. 13. 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 2-13.

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 2-8.

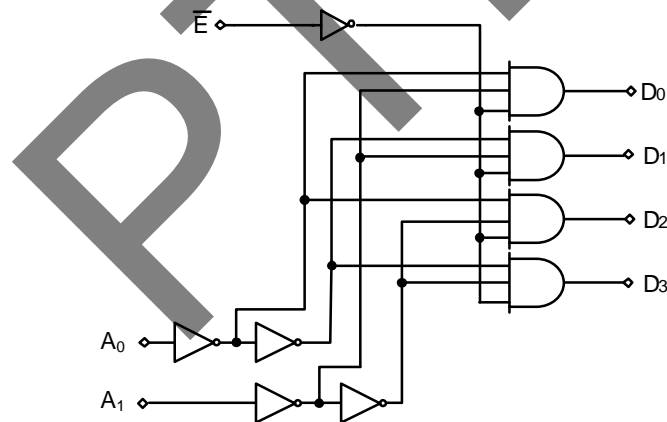
E	A ₁	A ₀	D ₀	D ₁	D ₂	D ₃
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Bảng 2-8. Bảng trạng thái

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

$$\left. \begin{aligned} D_0 &= E \cdot \overline{A_1} \cdot \overline{A_0} \\ D_1 &= E \cdot \overline{A_1} \cdot A_0 \\ D_2 &= E \cdot A_1 \cdot \overline{A_0} \\ D_3 &= E \cdot A_1 \cdot A_0 \end{aligned} \right\} \quad (2.3)$$

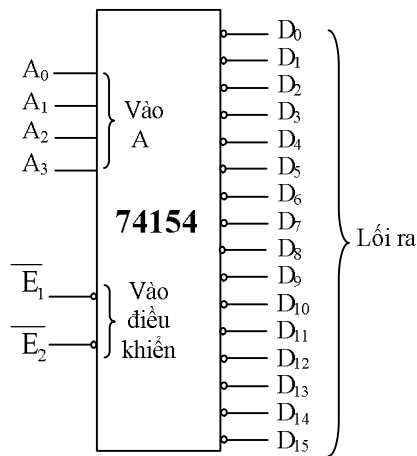
Từ biểu thức (2.3) vẽ được sơ đồ logic như hình 2.14.



Hình 2. 14. 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 2-10. Các đầu vào E₁, E₂, 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.



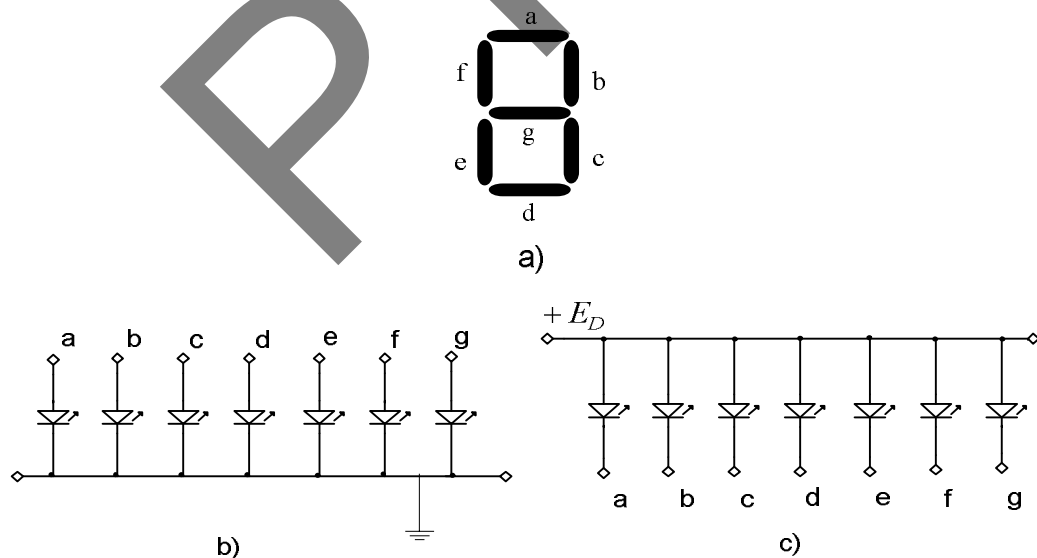
Hình 2. 15. Sơ đồ chân của IC 74154

2.4.3.2. 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 2. 16 a) Cấu trúc của chỉ thị số 7 đoạn, b) LED catốt chung, c) LED anốt chung

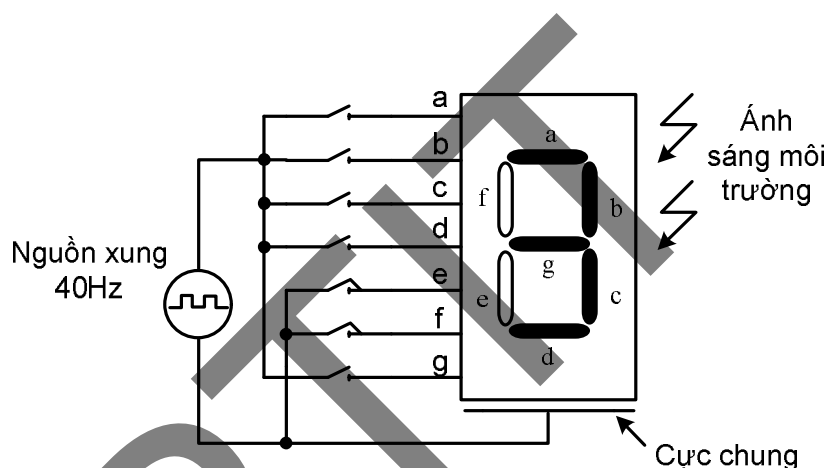
Hình 2.16 mô tả cấu trúc của đèn hiển thị số 7 đoạn sáng dùng diode phát quang, cách ký hiệu các đoạn bằng các chữ cái a, b, c, d, e, f, g. Bảy đoạn là 7 diode phát quang.

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.

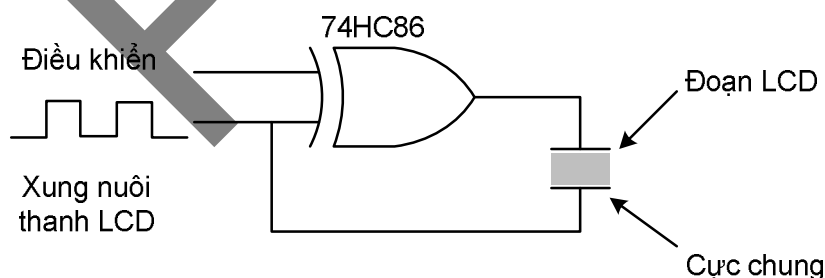
Các bộ giải mã có các đầu ra tích cực thấp, sử dụng LED anốt chung, đầu ra của bộ giải mã được nối với các catốt qua các điện trở hạn chế dòng.

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

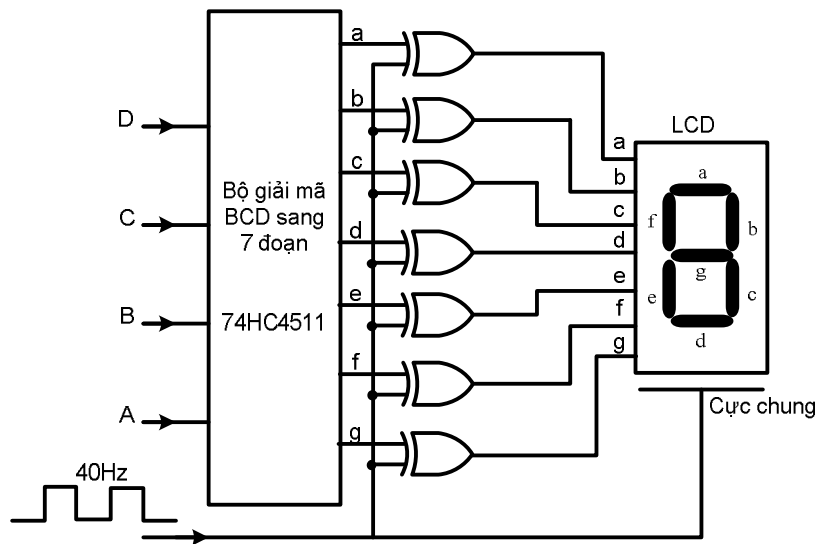
LCD chỉ hoạt động khi có ánh sáng phân cực chiếu vào. Để đạt được mục đích này, phía trước các đoạn cần có một kính phân cực ánh sáng. Các đoạn không hoạt động sẽ phản xạ đối với luồng ánh sáng phân cực này và do đó, không nhìn thấy chúng. Ngược lại các đoạn hoạt động không phản xạ nên bị đen lại. Ưu 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.



Hình 2. 17. Màn hình LCD và phương pháp kích thích sáng số 3 thập phân

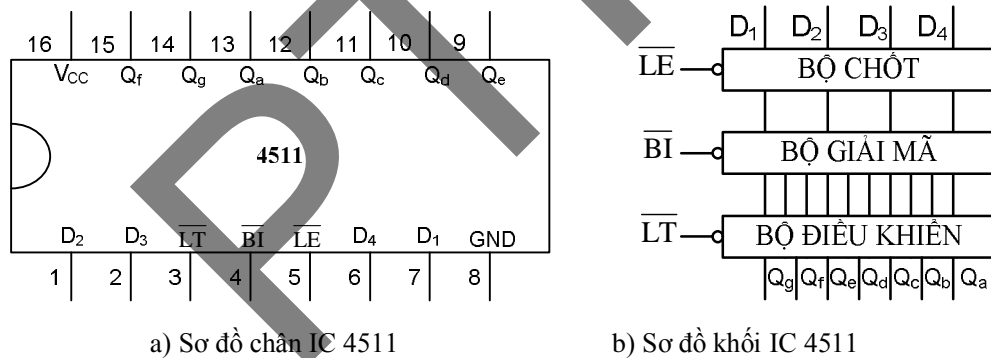


Hình 2. 18. Phương pháp kích thích 1 thanh LCD



Hình 2. 19. Mạch điện giải mã BCD sang 7 đoạn

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. Hình 2-12 mô tả một màn hình LCD gồm 7 đoạn giống LED với một điện cực chung với các tín hiệu vào từ a đến g tương ứng; khi các thanh a, b, c, d, g được kích thích với nguồn xung có biên độ từ 0 đến 5V, tần số 40Hz thì chúng sẽ phân xạ sáng, các thanh còn lại không được điện thế xoay chiều tác động nên vẫn tối. Do vậy, số 3 được hiển thị trên màn hình LCD.



Hình 2. 20 . Sơ đồ chân và sơ đồ khối của IC giải mã 4511.

Bảng chức năng của IC 4511 được chỉ ra như bảng 2-9.

Bằng cách tương tự, người ta có thể tạo ra màn LCD là các ma trận cỡ lớn, ví dụ 640(cột) x 480 (hàng) với 2400 mỗi nối với LCD (gọi là các điểm ảnh) đi qua các bộ lọc màu tạo ba màu cơ bản (đỏ, xanh lục và xanh lam) tạo màu cho từng điểm ảnh với một vi điện tử cỡ lớn VLSI điều khiển có thể quét màn hình LCD với tốc độ cao, thiết lập các mẫu ký số, ký tự và các ảnh video chất lượng cao.

Lối vào							Lối ra							Hiển thị
\overline{LE}	\overline{BI}	\overline{LT}	D ₄	D ₃	D ₂	D ₁	Q _a	Q _b	Q _c	Q _d	Q _e	Q _f	Q _g	
X	X	L	X	X	X	X	H	H	H	H	H	H	H	8
X	L	H	X	X	X	X	L	L	L	L	L	L	L	Đề trắng
L	H	H	L	L	L	L	H	H	H	H	H	H	L	0
L	H	H	L	L	L	H	L	H	H	L	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	H	2
L	H	H	L	L	H	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	H	L	H	H	L	H	H	5
L	H	H	L	H	H	L	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	L	Đề trắng
L	H	H	H	L	H	H	L	L	L	L	L	L	L	Đề trắng
L	H	H	H	H	L	L	L	L	L	L	L	L	L	Đề trắng
L	H	H	H	H	H	L	L	L	L	L	L	L	L	Đề trắng
L	H	H	H	H	H	H	L	L	L	L	L	L	L	Đề trắng
L	H	H	H	H	H	H	L	L	L	L	L	L	L	Đề trắng
H	H	H	X	X	X	X	Phụ thuộc vào mã BCD khi \overline{LE} chuyển trạng thái từ L đến H							

Bảng 2-9. Bảng chức năng của IC 4511

2.4.3.3. Thiết kế bộ giải mã BCD sang 7 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

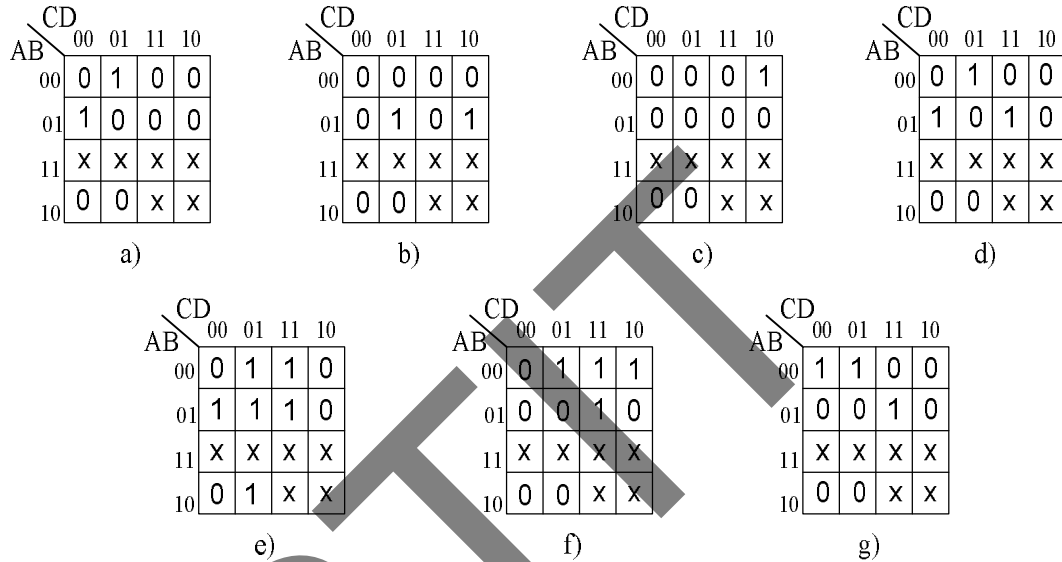
Bảng 2-10. Bảng trạng thái bộ giải mã

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 2-10 là bảng trạng thái bộ giải mã BCD sang bảy đoạn. Tối thiểu hoá dùng phương pháp bảng Các nô. Hình 2.21 trình bày các bảng Karnaugh.

Các ô “x” có thể tuỳ 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 2. 21. 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) $\bar{b} = \bar{B} + C.D + \bar{C}.\bar{D}$

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

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

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

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

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

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

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

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

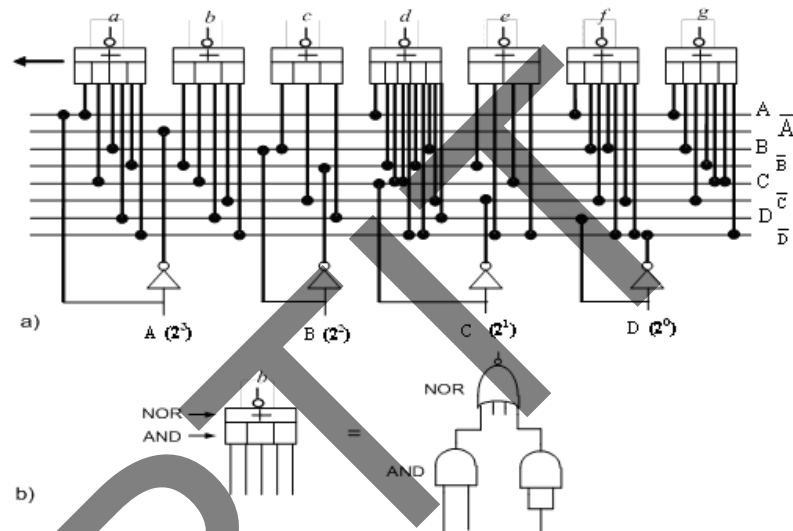
$$f = \overline{A + B \cdot \overline{D} + B \cdot \overline{C} + \overline{C} \cdot \overline{D}}$$

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

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

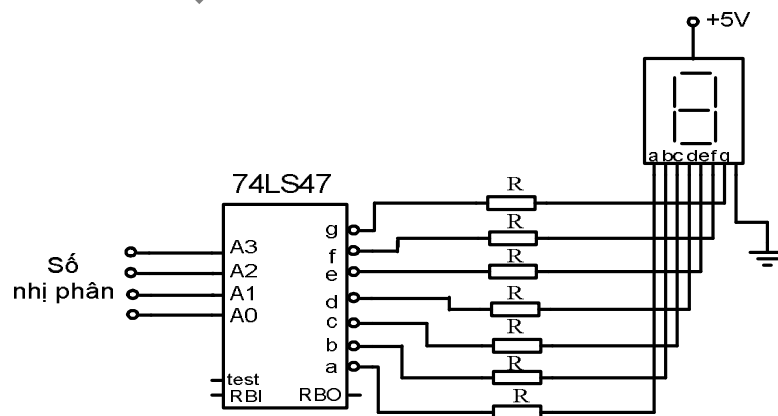
Xuất phát từ các hàm logic tối thiểu hoá sơ đồ logic được trình bày trên hình 2.22.

Bộ giải mã BCD sang 7 đoạn (loại 7447).



Hình 2. 22. a) Bộ giải mã BCD sang 7 đoạn; b) Một dạng ký hiệu của cổng NORAND

Vì mạch 7447 là mạch giải mã nhị phân sang 7 đoạn có đầu ra tác động thấp nên đèn chỉ thị 7 đoạn là LED có anốt chung.



Hình 2. 23. Sơ đồ nối 7447 và LED anốt chung

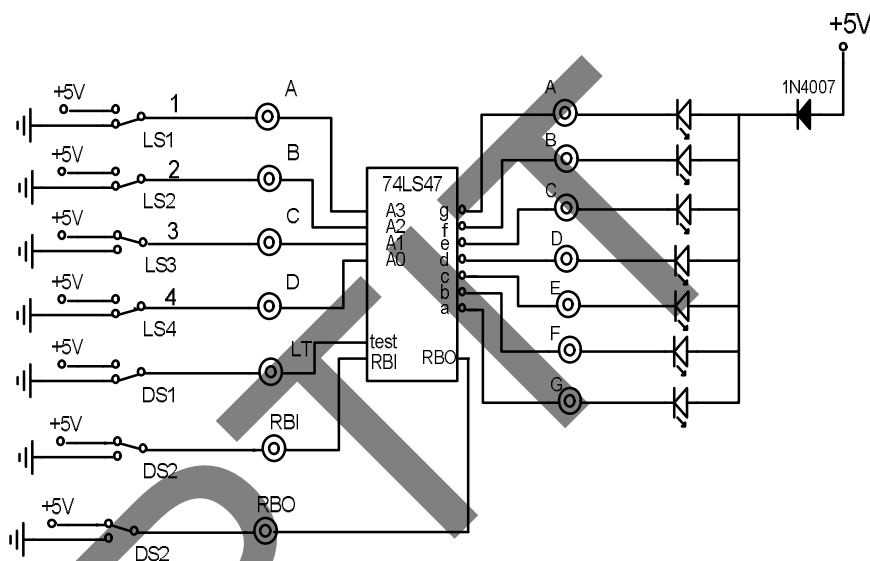
Sơ đồ nối mạch giữa 7447 và chỉ thị 7 đoạn được trình bày trên hình 2.23.

Trong đó các điện trở R làm nhiệm vụ bảo vệ cho các LED, $R = 300 \Omega$.

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) (Hình 2.24).



Hình 2. 24. Bộ giải mã BCD sang 7 đoạn.

Khi $RBO = 0$ thì LED tắt

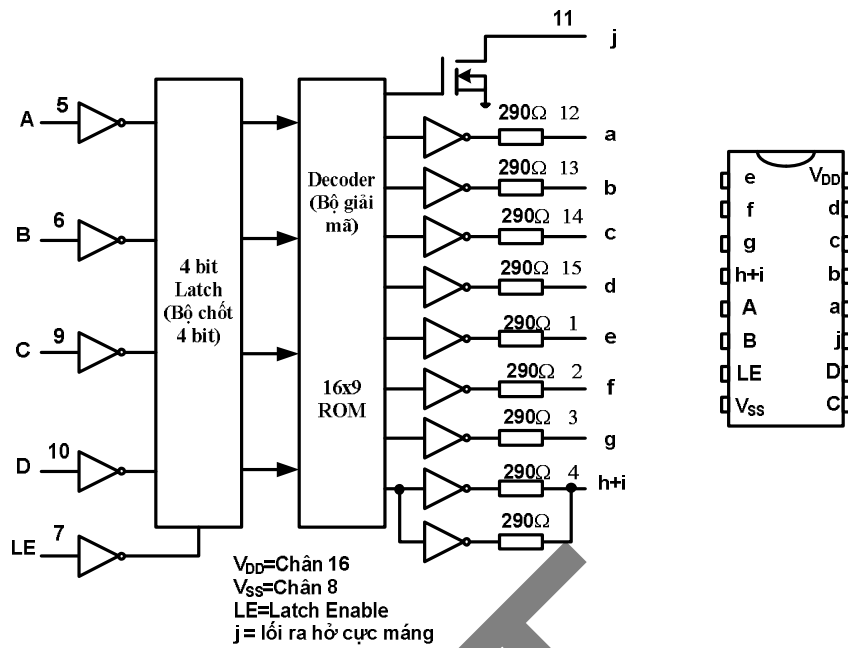
Khi $DCBA = 0000$ tương ứng hiển thị số 0. Nếu $RBI = 0$ thì LED tắt, còn các đầu vào ở giá trị khác, đầu ra hiển thị các chữ số khác 0 vẫn sáng bình thường.

Bộ giải mã nhị phân sang 7 đoạn (loại MC 14495).

Vì mạch MC 14495 là bộ giải mã nhị phân sang 7 đoạn hay còn gọi là bộ giải mã hexa sang 7 đoạn. Bộ giải mã này có các đầu ra tác động cao nên dùng đèn chỉ thị 7 đoạn có catốt chung.

Để bảo vệ cho các LED, người ta đã chế tạo các điện trở cùng loại 290Ω nối với các đầu ra trong vi mạch. Vì vậy khi dùng nguồn nuôi 5V không cần lắp thêm các điện trở R như khi dùng vi mạch 7447.

Sơ đồ được trình bày trên hình 2.25.



Hình 2. 25 Bộ giải mã nhị phân sang 7 đoạn (loại MC 14495)

Bảng 2-11 trình bày bảng trạng thái MC14495 và các hiển thị của LED.

Đầu vào				Đầu ra									Hiển thị
D	C	B	A	a	b	c	d	e	f	g	h+i	j	
0	0	0	0	1	1	1	1	1	1	0	0	Z cao	0
0	0	0	1	0	1	1	0	0	0	0	0	Z cao	1
0	0	1	0	1	1	0	1	1	0	1	0	Z cao	2
0	0	1	1	1	1	1	1	0	0	1	0	Z cao	3
0	1	0	0	0	1	1	0	0	1	1	0	Z cao	4
0	1	0	1	1	0	1	1	0	1	1	0	Z cao	5
0	1	1	0	1	0	1	1	1	1	1	0	Z cao	6
0	1	1	1	1	1	1	0	0	0	0	0	Z cao	7
1	0	0	0	1	1	1	1	1	1	1	0	Z cao	8
1	0	0	1	1	1	1	1	0	1	1	0	Z cao	9
1	0	1	0	1	1	1	0	1	1	1	1	Z cao	A
1	0	1	1	0	0	1	1	1	1	1	1	Z cao	B
1	1	0	0	1	0	0	1	1	1	0	1	Z cao	C
1	1	0	1	0	1	1	1	1	0	1	1	Z cao	D
1	1	1	0	1	0	0	1	1	1	1	1	Z cao	E
1	1	1	1	1	0	0	0	1	1	1	1	0	F

Bảng 2-11. Bảng trạng thái MC14495 (Giải mã hexa sang bảy đoạn)

2.4.4. 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 hợp kênh và phân kênh. Tuy nhiên trên thực tế có một số loại IC có sẵn để thực hiện chức năng này.

2.4.4.1. Bộ biến mã từ nhị phân sang Gray

Bảng 2-12 là bảng trạng thái mô tả mối quan hệ giữa mã nhị phân và mã Gray 3 bit.

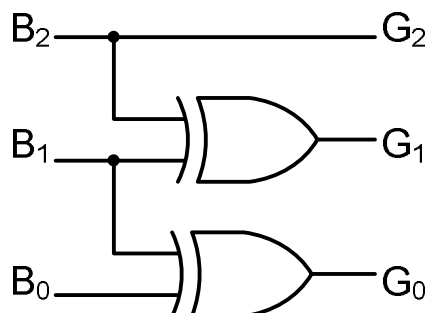
Thập phân	Nhị phân			Gray		
	B ₂	B ₁	B ₀	G ₂	G ₁	G ₀
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Bảng 2-12. Bảng trạng thái mạch biến mã

Từ bảng trạng thái 2-12, viết được biểu thức của hàm ra như sau:

$$\left. \begin{aligned} G_2 &= B_2 \\ G_1 &= B_2 \oplus B_1 \\ G_0 &= B_1 \oplus B_0 \end{aligned} \right\} \quad (2.4)$$

Từ đó, vẽ được sơ đồ logic như trên hình 2.26.



Hình 2. 26. Mạch điện thực hiện bộ biến mã từ nhị phân sang Gray

2.4.4.2. Bộ biến mã từ Gray sang nhị phân

Bảng 2-13 là bảng trạng thái mô tả mối quan hệ giữa mã nhị phân và mã Gray 3 bit.

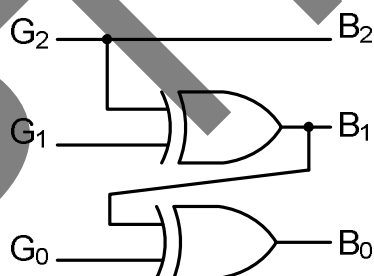
Thập phân	Gray			Nhị phân		
	G ₂	G ₁	G ₀	B ₂	B ₁	B ₀
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	1	0	1	0
3	0	1	0	0	1	1
4	1	1	0	1	0	0
5	1	1	1	1	0	1
6	1	0	1	1	1	0
7	1	0	0	1	1	1

Bảng 2-13. Bảng trạng thái mạch biến mã

Từ bảng trạng thái 2-13, viết được biểu thức của hàm ra như sau:

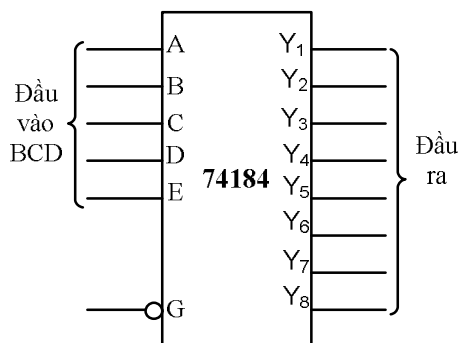
$$\left. \begin{aligned} B_2 &= G_2 \\ B_1 &= B_2 \oplus G_1 \\ B_0 &= B_1 \oplus G_0 \end{aligned} \right\} \quad (2.5)$$

Từ đó, vẽ được sơ đồ logic như trên hình 2.27.



Hình 2. 27. Mạch điện thực hiện bộ biến mã từ Gray sang nhị phân

2.4.4.3. Bộ biến mã từ BCD sang nhị phân dùng IC 74184



Hình 2. 28. Sơ đồ khối của IC 74184.

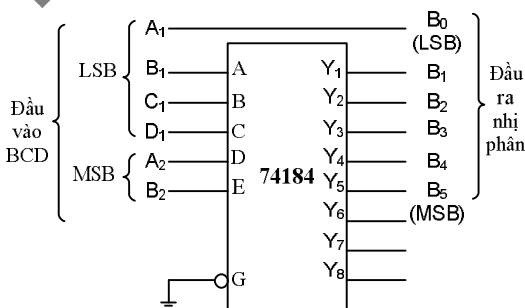
Hình 2.28 là sơ đồ khối của IC 74184.

Bảng chức năng của IC 74184 được cho trên bảng 2-14.

Số BCD	Các đầu vào						Các đầu ra				
	E	D	C	B	A	G	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁
0-1	0	0	0	0	0	0	0	0	0	0	0
2-3	0	0	0	0	1	0	0	0	0	0	1
2-5	0	0	0	1	0	0	0	0	0	1	0
6-7	0	0	0	1	1	0	0	0	0	1	1
8-9	0	0	1	0	0	0	0	0	1	0	0
10-11	0	1	0	0	0	0	0	0	1	0	1
12-13	0	1	0	0	1	0	0	0	1	1	0
12-15	0	1	0	1	0	0	0	0	1	1	1
16-17	0	1	0	1	1	0	0	1	0	0	0
18-19	0	1	1	0	0	0	0	1	0	0	1
20-21	1	0	0	0	0	0	0	1	0	1	0
22-23	1	0	0	0	1	0	0	1	0	1	1
22-25	1	0	0	1	0	0	0	1	1	0	0
26-27	1	0	0	1	1	0	0	1	1	0	1
28-29	1	0	1	0	0	0	0	1	1	1	0
30-31	1	1	0	0	0	0	0	1	1	1	1
32-33	1	1	0	0	1	0	1	0	0	0	0
32-35	1	1	0	1	0	0	1	0	0	0	1
36-37	1	1	0	1	1	0	1	0	0	1	0
38-39	1	1	1	0	0	0	1	0	0	1	1
Bất kỳ	x	x	x	x	x	1	1	1	1	1	1

Bảng 2-14. Bảng chức năng của IC 74184

Từ bảng 2-14, xây dựng được bộ biến mã từ BCD sang nhị phân như hình 2.29.



Hình 2. 29. Bộ biến mã từ BCD sang nhị phân

Các đầu vào BCD được đưa vào các chân từ A đến E. Bit có trọng số nhỏ nhất của số BCD được đưa thẳng tới đầu ra của nhị phân. Mạch này chấp nhận hai số BCD với một chữ số đầy đủ 4 bit: D₁ C₁ B₁ A₁ và hai bit có trọng số bé nhất của chữ số thứ hai BCD: B₂ A₂.

Điều này có nghĩa là các đầu vào BCD nằm trong phạm vi từ 00 đến 39 đều được biến thành số nhị phân tương ứng bởi mạch này. Các chân Y_6 , Y_7 và Y_8 không được dùng cho mạch này. Những chân này thường được dùng để tìm bù 9 và bù 10 cho số BCD.

Đối với mạch biến mã từ nhị phân sang BCD thường sử dụng IC 74185.

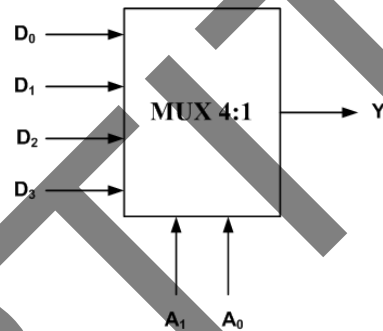
2.5. BỘ HỢP KÊNH VÀ PHÂN KÊNH

2.5.1 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). Chức năng logic cơ bản của bộ hợp 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 2.30 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 2. 30. Bộ hợp kênh 4 vào – 1 ra

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

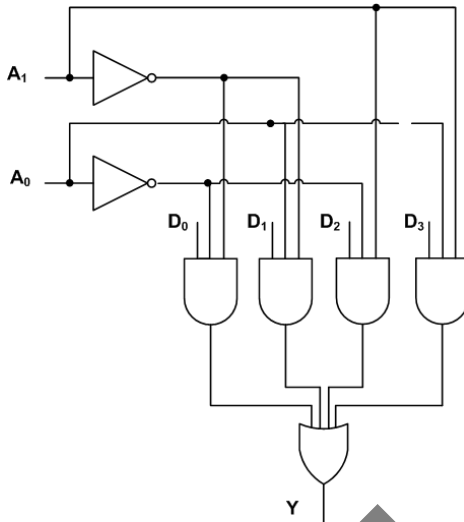
A_1	A_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Bảng 2-15. Bảng chức năng của bộ hợp kênh 4:1.

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

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

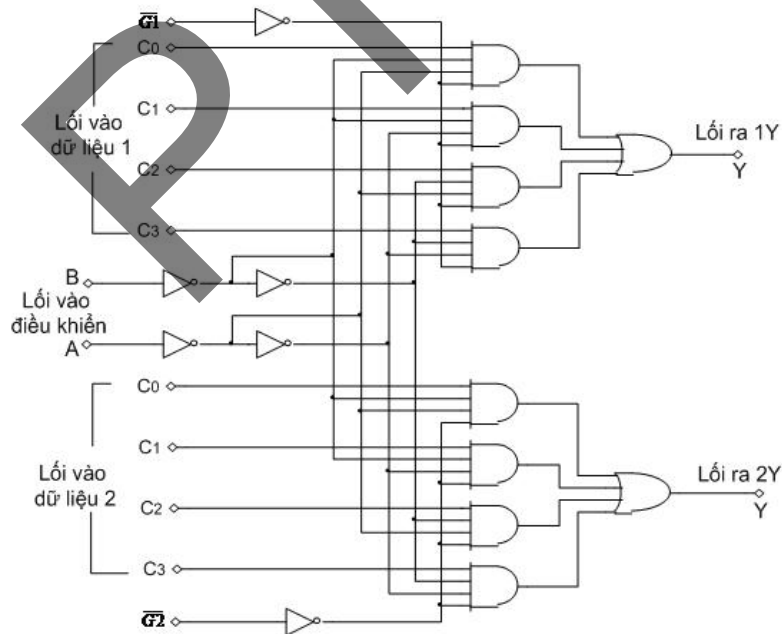
Mạch điện 2.31 thể hiện hàm Y.



Mạch hợp kênh 4:1

Hình 2. 31. Sơ đồ logic mạch hợp kênh 4:1.

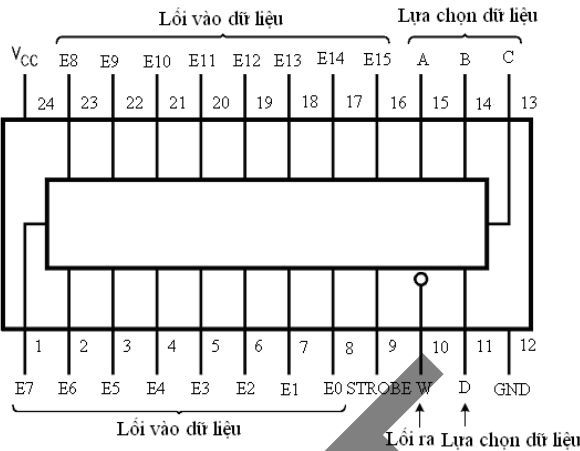
Trong thực tế người ta chế tạo các bộ hợp kênh có 4, 8 hoặc 16 đầu vào dữ liệu. Hình 2.32 trình bày sơ đồ logic của vi mạch 74LS153. Trong vi mạch gồm 2 bộ hợp kênh có 4 đường vào dữ liệu, kí hiệu là C_0, C_1, C_2, C_3 và một đường ra Y. Cả hai bộ hợp kênh đều có chung 2 đầu vào điều khiển A, B, mỗi bộ hợp 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): $0V$.



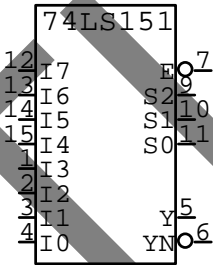
Hình 2. 32. Sơ đồ logic của bộ hợp kênh (4:1)

Vì mạch 74150 có ký hiệu logic trên hình 2.33 là bộ hợp 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 hợp kênh 8 đường có hai đầu ra Y và W là đảo của Y (hình 2.34).



Hình 2. 33. Ký hiệu logic của bộ hợp kênh 74150



Hình 2. 34. Ký hiệu logic của bộ hợp 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

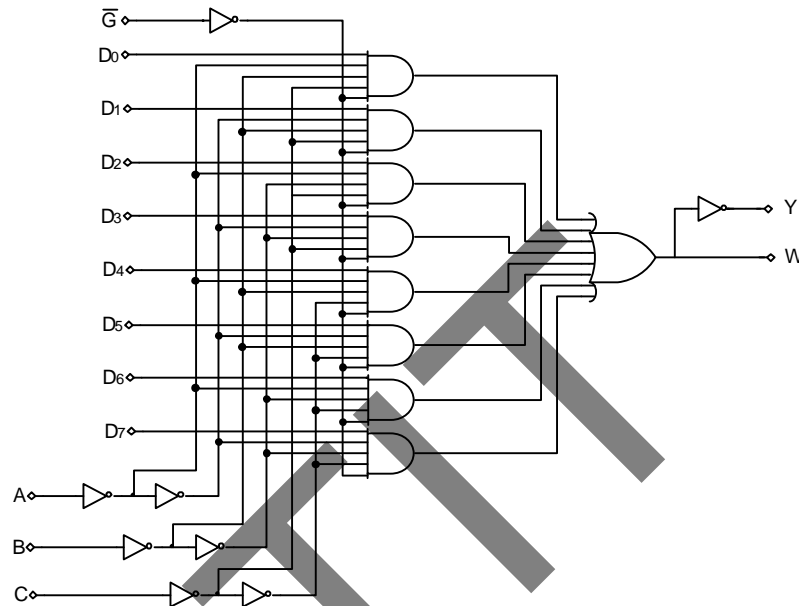
Bảng 2-13. Bảng chức năng của bộ hợp kênh 74151.

Từ bảng chức năng 2-13, 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:

$$Y = G \cdot \left(\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D_0 + A \cdot \bar{B} \cdot \bar{C} \cdot D_1 + \bar{A} \cdot B \cdot \bar{C} \cdot D_2 + A \cdot B \cdot \bar{C} \cdot D_3 + \right. \\ \left. + \bar{A} \cdot \bar{B} \cdot C \cdot D_4 + A \cdot \bar{B} \cdot C \cdot D_5 + \bar{A} \cdot B \cdot C \cdot D_6 + A \cdot B \cdot C \cdot D_7 \right) \quad (2.6)$$

$$W = \bar{Y}$$

Từ biểu thức logic 2.6 có sơ đồ logic của bộ hợp kênh trên hình 2.35.

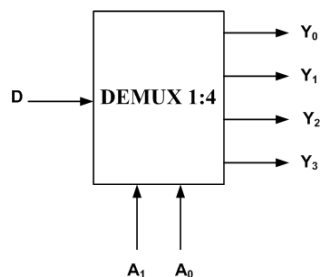


Hình 2. 35. Sơ đồ logic của bộ hợp kênh 8 đường 74151

2.5.2. 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 hợp 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 hợp kênh, bộ phân kênh cũng có các đường vào điều khiển (n đường) chọn đầu ra (2^n đầu ra).

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



Sơ đồ khối DEMUX 1:4

Hình 2. 36. Sơ đồ khối của bộ phân kênh 1:4.

Sơ đồ khối mô tả chức năng của bộ phân kênh một đầu vào, 4 đầu ra được trình bày trên hình 2.36. Từ sơ đồ khối hình 2.36 lập được bảng trạng thái.

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 2-14. 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 2-14.

Từ bảng trạng thái 2-14, viết được biểu thức logic của bộ phân kênh như sau:

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

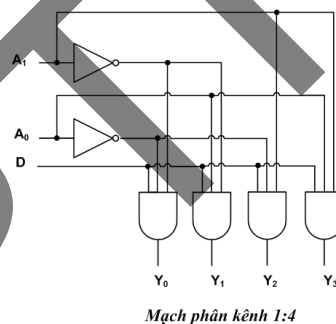
$$Y_1 = A \cdot \bar{B} \cdot D$$

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

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

(2.7)

Từ biểu thức logic, có thể xây dựng sơ đồ logic của bộ phân kênh:



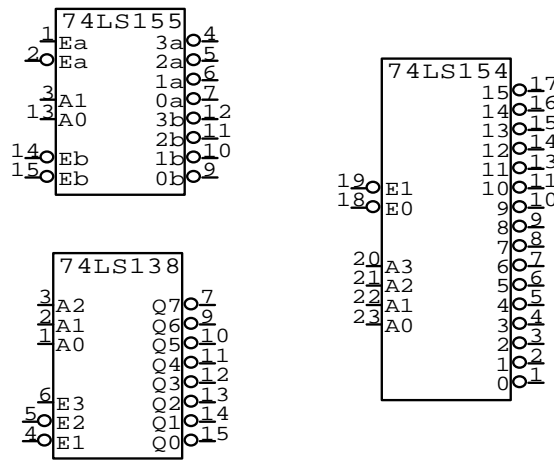
Hình 2. 37. Sơ đồ logic mạch phân kênh 1:4.

Hình 2.37 là các ký hiệu của các vi mạch phân kênh thường gặp trong thực tế.

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.

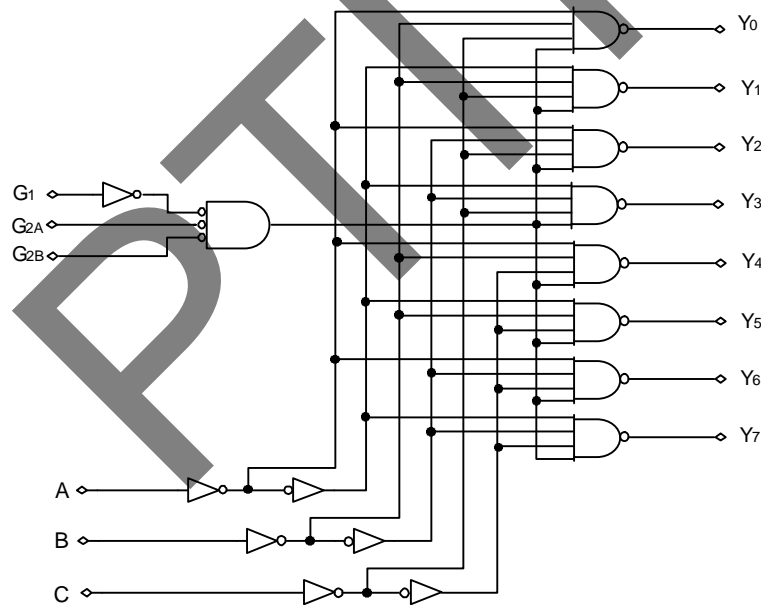
IC 74LS138 là bộ giải mã địa chỉ, đồng thời có thể dùng làm bộ phân kênh một đường vào, 8 đường ra dữ liệu (hình 2.38).



Hình 2. 38. Ký hiệu logic của các bộ phân kênh

Hình 2-39 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 2. 39. Sơ đồ logic của mạch giải mã, phân kênh 74LS138

2.5.3. Một số ứng dụng của bộ hợp kênh và phân kênh

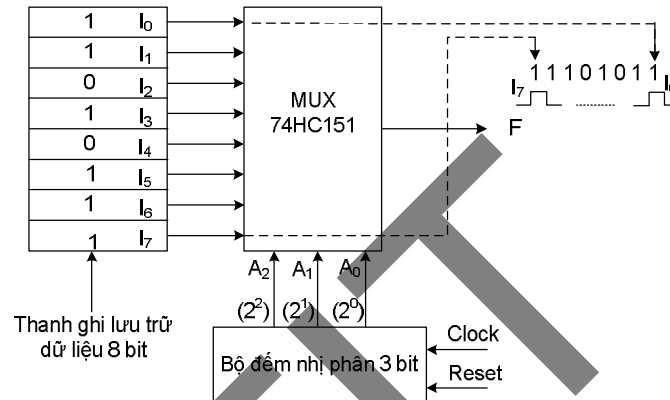
2.5.3.1. Định tuyến dữ liệu

Có nhiều dữ liệu được định hướng tới một đích duy nhất, khi đó sử dụng bộ hợp 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.

2.5.3.2. Chuyển đổi luồng dữ liệu từ song song sang 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 hợp kênh-MUX.

Ví dụ: Hình 2.40 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 2. 40. Chuyển dữ liệu song song 8 bit sang dãy nối tiếp 8 bit sau 8 xung nhịp

2.5.3.3. Tạo hàm logic

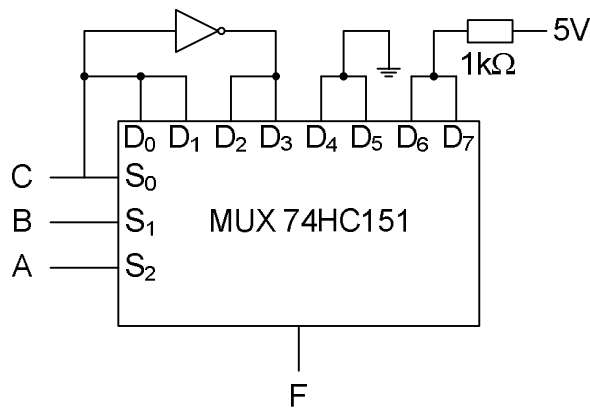
Sử dụng MUX tạo hàm logic trực tiếp từ bảng trạng thái không cần rút gọn, với mục đích này các đầu vào chọn (đầu vào địa chỉ) là các biến logic, mỗi đầu vào dữ liệu được nối thường xuyên với mức cao (logic 1) hay mức thấp (logic 0) tùy theo bảng trạng thái.

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

Từ biểu thức trên, lập được bảng trạng thái 2-15.

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 2-15. Bảng trạng thái



Hình 2. 41. 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 chọn B, C là đầu vào địa chỉ thì A sẽ là đầu vào dữ liệu, hoặc 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 2.41.

2.6. MẠCH SỐ HỌC.

2.6.1. Mạch tổng.

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 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 mạch cộng song song 4 bit từ mạch cộng toàn phần (đầy đủ) 1 bit.

2.6.1.1. Mạch bán tổng (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.

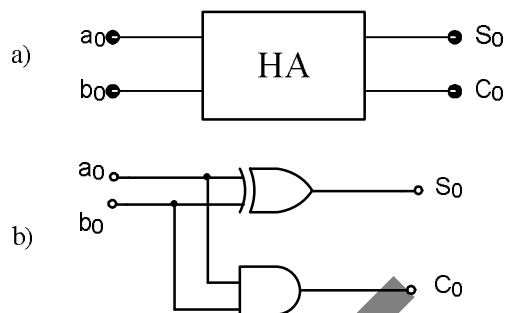
Mạch có 2 đầu ra: S_0 là kết quả phép cộng ở cột có trọng số nhỏ nhất.

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 2-16.

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 2-16. Bảng trạng thái của mạch HA



Hình 2. 42. a) Sơ đồ khối của HA, b) Sơ đồ logic

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

$$\begin{aligned} S_0 &= a_0 \oplus b_0 \\ C_0 &= a_0 . b_0 \end{aligned} \quad (2.8)$$

Sơ đồ khối và sơ đồ logic của HA được trình bày trên hình 2.42.

2.6.1.2. Mạch tổng 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 2-17.

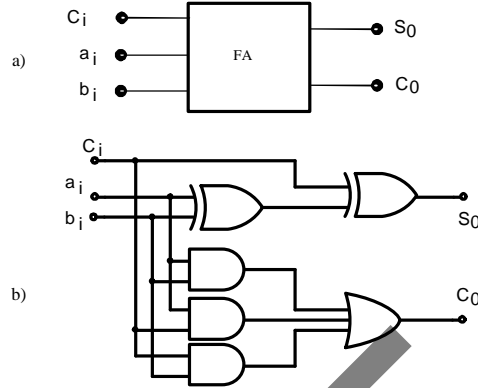
a_i	b_i	C_i	S_0	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Bảng 2-17. Bảng trạng thái của bộ FA

Từ bảng trạng thái suy ra hàm logic của FA:

$$\begin{aligned} S_0 &= a_i \oplus b_i \oplus C_i \\ C_0 &= a_i \cdot b_i + a_i C_i + b_i C_i \end{aligned} \quad (2.9)$$

Sơ đồ khối và mạch logic trình bày trên hình 2.43.



Hình 2. 43. 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.

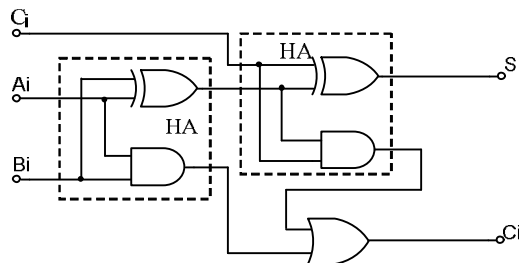
$$\begin{aligned} S_0 &= \overline{a_i} \cdot \overline{b_i} C_i + \overline{a_i} \cdot b_i \cdot \overline{C_i} + a_i \cdot \overline{b_i} \cdot \overline{C_i} + a_i \cdot b_i \cdot C_i \\ &= \overline{a_i} (\overline{b_i} C_i + b_i \cdot \overline{C_i}) + a_i (\overline{b_i} \cdot \overline{C_i} + b_i \cdot C_i) \\ &= a_i \oplus b_i \oplus C_i \end{aligned} \quad (2.10)$$

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

Có thể tính C_0 theo cách khác:

$$\begin{aligned} C_0 &= (\overline{a_i} \cdot b_i \cdot C_i + a_i \cdot \overline{b_i} \cdot C_i) + (a_i \cdot b_i \cdot \overline{C_i} + a_i \cdot b_i \cdot C_i) \\ &= C_i (\overline{a_i} \cdot b_i + a_i \cdot \overline{b_i}) + a_i \cdot b_i = C_i (a_i \oplus b_i) + a_i \cdot b_i \\ C_0 &= C_i (a_i \oplus b_i) + a_i \cdot b_i \end{aligned} \quad (2.11)$$

Từ công thức (2.11) và (2.12) có thể xây dựng mạch FA từ hai mạch HA. Sơ đồ logic của FA từ hai HA được trình bày trên hình 2.44.



Hình 2. 44. Sơ đồ mạch FA.

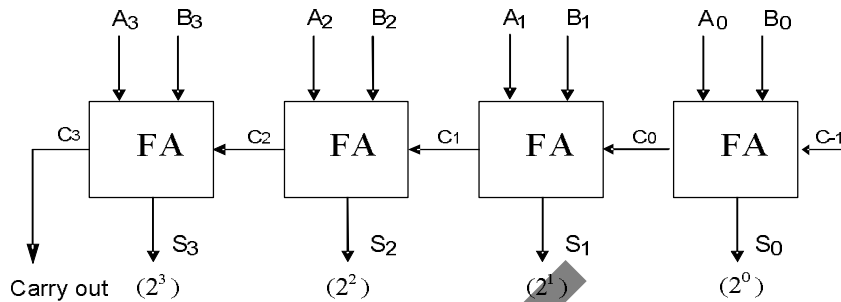
Nếu $C_1 = 0$ thì FA trở thành HA.

2.6.1.3. Bộ cộng song song 4 bit.

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

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

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



Hình 2. 45. Bộ cộng song song 4 bit.

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 2.45.

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 của bộ cộng loại này: Tín hiệu nhớ ở đầu ra (Carry Out) được tạo thành sau khi đi qua bốn bộ cộng FA_0, \dots, FA_3 . Nếu mỗi bộ cộng FA 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).

2.6.2. Mạch hiệu.

2.6.2.1. Mạch bán hiệu (HS-Half Subtract)

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 2-18.

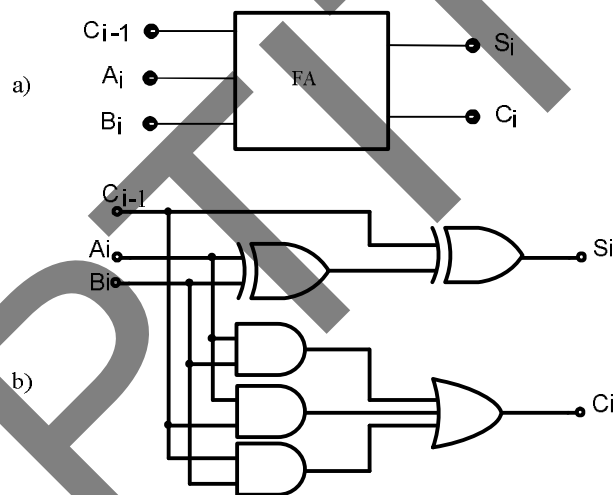
a_0	b_0	S_0	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Bảng 2-18. Bảng trạng thái của mạch HS

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

$$\begin{aligned} S_0 &= a_0 \oplus b_0 \\ B_0 &= \overline{a_0} \cdot b_0 \end{aligned} \quad (2.12)$$

Sơ đồ khối và sơ đồ logic của HS được trình bày trên hình 2.46.



Hình 2. 46. Sơ đồ mạch FA.

2.6.2.2. Mạch hiệu toàn phần

Mạch hiệu 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 2-19 là bảng trạng thái của bộ trừ đầy đủ.

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

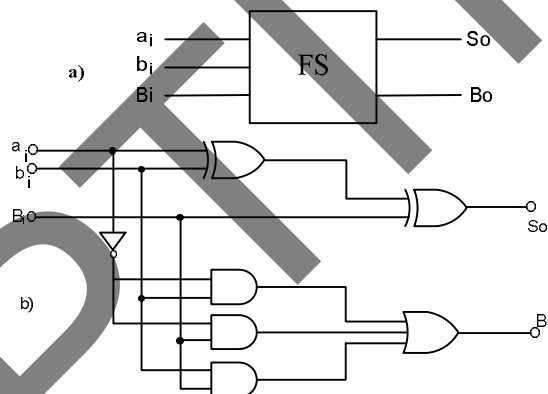
Bảng 2-19. Bảng trạng thái của bộ trừ

Hàm logic:

$$\begin{aligned} S_0 &= a_i \oplus b_i \oplus B_i \\ B_0 &= \overline{a_i} \cdot b_i + \overline{a_i} \cdot B_i + b_i \cdot B_i \end{aligned} \quad (2.13)$$

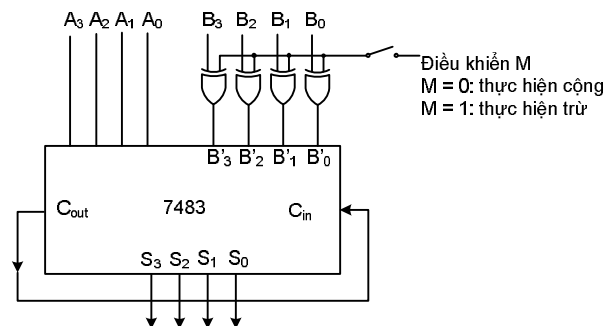
Sơ đồ khối và sơ đồ logic được trình bày trên hình 2.47.

Nếu $B_i = 0$ có mạch trừ bán phần.



Hình 2.47. Sơ đồ của FS: a) Sơ đồ khối; b) Sơ đồ logic.

2.6.3. Bộ cộng, trừ theo bù 1 và bù 2.

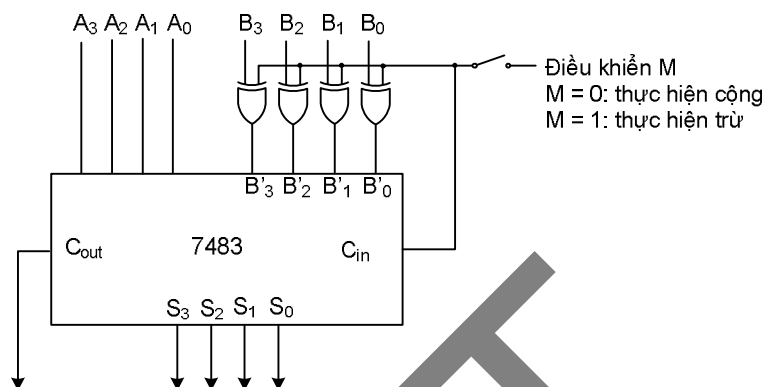


Hình 2.48. Mạch cộng/trừ theo bù 1

Cộng/ trừ theo bù 1 có nghĩa là số dấu trừ biến thành dấu cộng thông qua phép bù 1. Khi thực hiện cộng, nếu xuất hiện bit tràn thì phải cộng vào kết quả để được đáp số chính xác.

Cộng/ trừ theo bù 2 có nghĩa là số dấu trừ biến thành dấu cộng thông qua phép bù 2. Khi thực hiện cộng, nếu xuất hiện bit tràn thì phải bỏ bit đó đi.

Hình 2.48 và 2.49 giới thiệu mạch cộng/trừ theo bù 1 và 2 sử dụng IC 7483 (cộng 2 số nhị phân 4 bit) và IC 7486 (XOR).



Hình 2. 49. Mạch cộng/trừ theo bù 2.

2.7. 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.

2.7.1. Bộ so sánh.

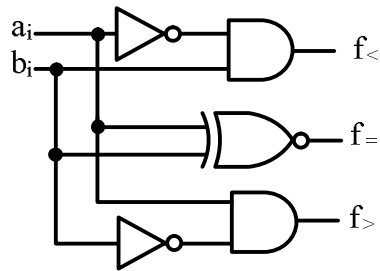
2.7.2.1. Bộ so sánh 1 bit.

Từ bảng trạng thái 2-20 có biểu thức ra:

$$\begin{aligned} f_{<} &= \overline{a_i} \cdot b_i \\ f_{=} &= a_i \oplus b_i \\ f_{>} &= a_i \cdot \overline{b_i} \end{aligned} \quad (2.14)$$

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 2-20. Bảng trạng thái của mạch so sánh



Hình 2. 50. Mạch điện bộ so sánh 1 bit

2.7.2.2. Bộ 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:

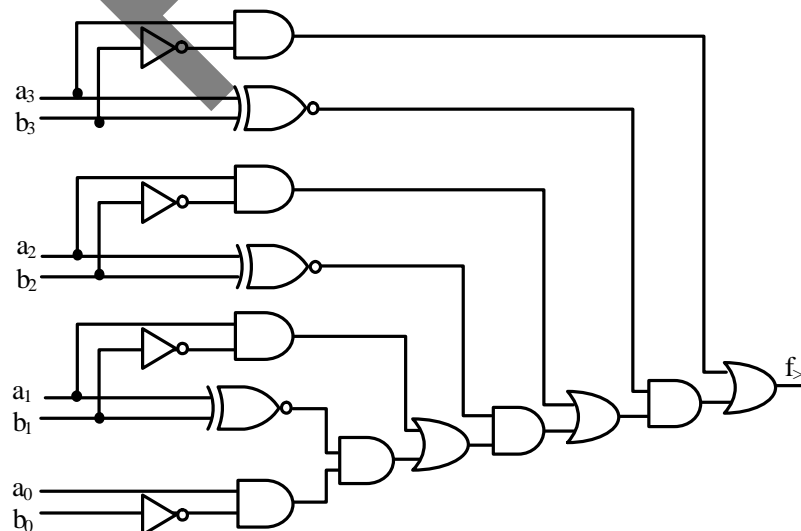
$$f_{>} = a_3 \cdot \overline{b_3} + \overline{a_3 \oplus b_3} \cdot a_2 \cdot \overline{b_2} + \overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot a_1 \cdot \overline{b_1} + \overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot \overline{a_1 \oplus b_1} \cdot a_0 \cdot \overline{b_0}$$

Từ biểu thức trên, vẽ được sơ đồ logic 2.51.

Một trong những bộ so sánh thông dụng hiện nay là 7485, 74LS85, 74HC85 - IC này so sánh 2 số nhị phân 4 bit. Bảng 2-21 là bảng chức năng của IC 74HC85.

Hình 2.52 là ký hiệu logic của IC 74HC85.

IC 74HC85 có 8 đường vào dữ liệu là 2 số nhị phân 4 bit $A_3A_2A_1A_0$ và $B_3B_2B_1B_0$ và 3 đầu ra tích cực mức cao thể hiện 3 kết quả của phép so sánh (hình 2-48). Ngoài ra IC này còn có 3 đầu vào nối tăng để mở rộng phép so sánh.

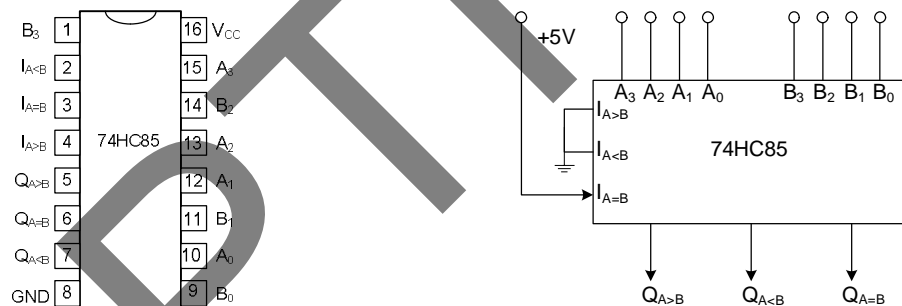


Hình 2. 51. Mạch điện bộ so sánh lớn hơn 4 bit

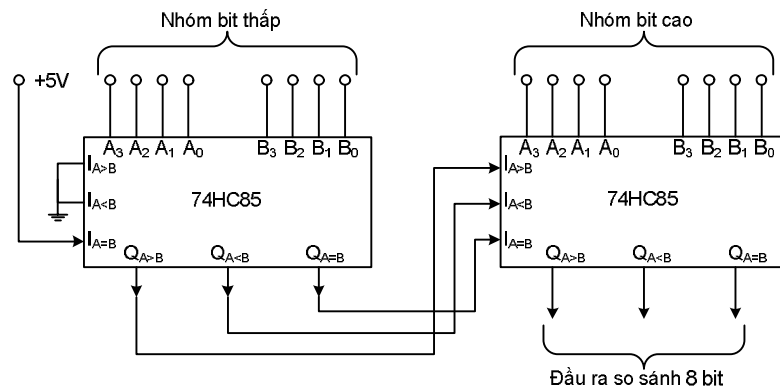
Đầu vào so sánh				Đầu vào nối tiếp tầng			Đầu ra		
A_3B_3	A_2B_2	A_1B_1	A_0B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$Q_{A>B}$	$Q_{A<B}$	$Q_{A=B}$
$A_3>B_3$	x	x	x	x	x	x	1	0	0
$A_3<B_3$	x	x	x	x	x	x	0	1	0
$A_3=B_3$	$A_2>B_2$	x	x	x	x	x	1	0	0
$A_3=B_3$	$A_2<B_2$	x	x	x	x	x	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	x	x	x	x	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	x	x	x	x	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	x	x	x	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	x	x	x	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	0	0	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	1	0	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	0	1	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	0	0	1	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	1	0	0	0	0

Bảng 2-21. Bảng trạng thái của IC so sánh 4 bit 74HC85

Ví dụ muốn thực hiện so sánh 2 số nhị phân 8 bit phải nối 2 IC 7485. Hình 2.52 và 2.53 giới thiệu ký hiệu và cách đấu nối IC.



Hình 2. 52. Ký hiệu logic của IC 74HC85 và cách nối IC 74HC85 khi so sánh 2 số nhị phân 4 bit



Hình 2. 53. Cách mắc IC 74HC85 khi so sánh 2 số nhị phân 8 bit

2.8. MẠCH TẠO VÀ KIỂM TRA CHẴN LẼ.

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 chúng ta 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).

2.8.1. 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.

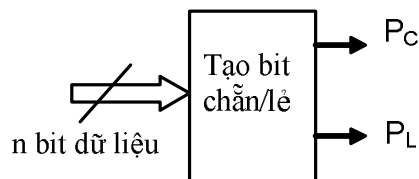
Bảng 2-22 là một ví dụ về mã chẵn/lẻ.

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
1000	1000	1	1000	0
1001	1001	0	1001	1

Bảng 2-22. Mã BCD 8421 chẵn / lẻ

4.8.2. Mạch tạo và kiểm tra chẵn/lẻ.

Hình 2.54 là sơ đồ khối của mạch tạo bit chẵn/lẻ.



Hình 2. 54. Sơ đồ khối của mạch tạo bit chẵn/lẻ

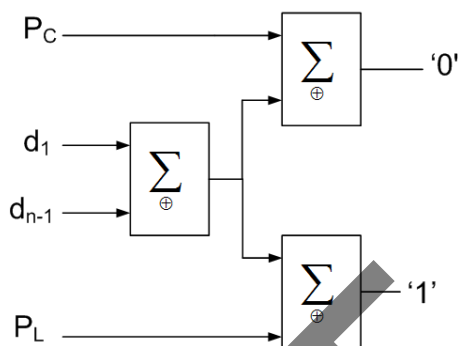
Biểu thức kiểm tra tính chẵn:

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

Biểu thức kiểm tra tính lẻ:

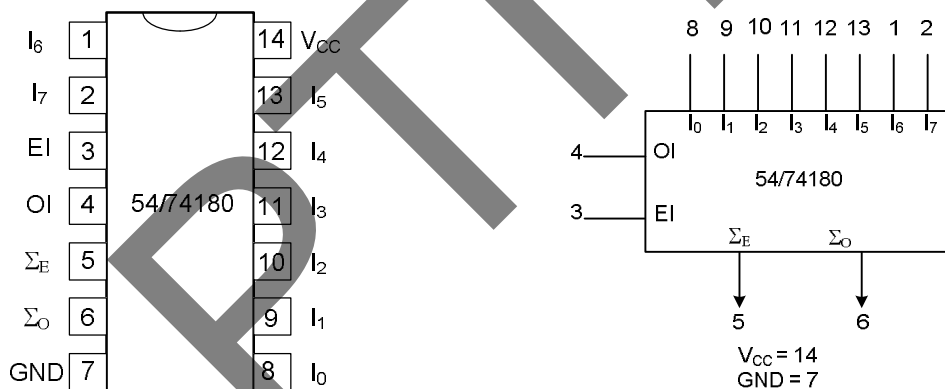
$$P_L \oplus d_1 \oplus d_2 \oplus \dots \oplus d_n = 1$$

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



Hình 2. 55. 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 2. 56. Sơ đồ chân của IC 54/74180 và sSơ đồ ký hiệu của IC 54/74180

Sau đây là bảng mô tả chân của IC

Chân	Chức năng
$I_0 \div I_7$	Dữ liệu vào
OI (Odd Input)	Đầu vào lẻ
EI (Even Input)	Đầu vào chẵn
Σ_O (Odd Parity Output)	Đầu ra lẻ
Σ_E (Even Parity Output)	Đầu ra chẵn

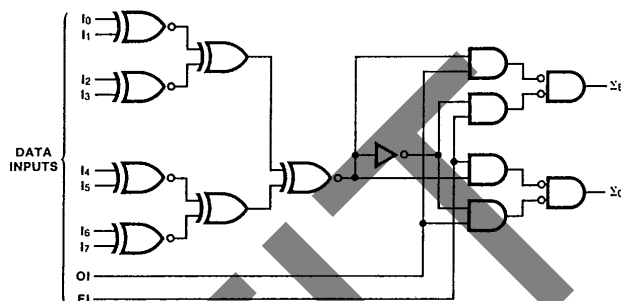
Bảng 2-26. Bảng mô tả chân IC

Bảng chức năng của IC:

Tính chẵn/lẻ của các đầu vào từ 0 ÷ 7	Đầu vào		Đầu ra	
	Đầu vào chẵn	Đầu vào lẻ	Đầu ra chẵn	Đầu ra lẻ
Chẵn	H	L	H	L
Lẻ	H	L	L	H
Chẵn	L	H	L	H
Lẻ	L	H	H	L
x	H	H	L	L
x	L	L	H	H

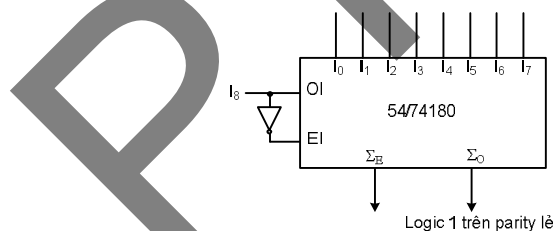
Bảng 2-27. Bảng chức năng của IC

Sơ đồ logic của IC



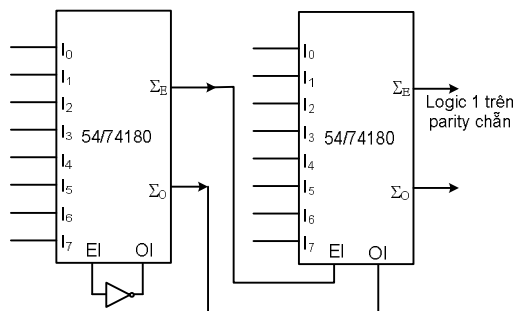
Hình 2. 57. Sơ đồ logic của IC 74180

Ví dụ 1: Tạo bộ kiểm tra tính lẻ 9 bit dùng 74180.



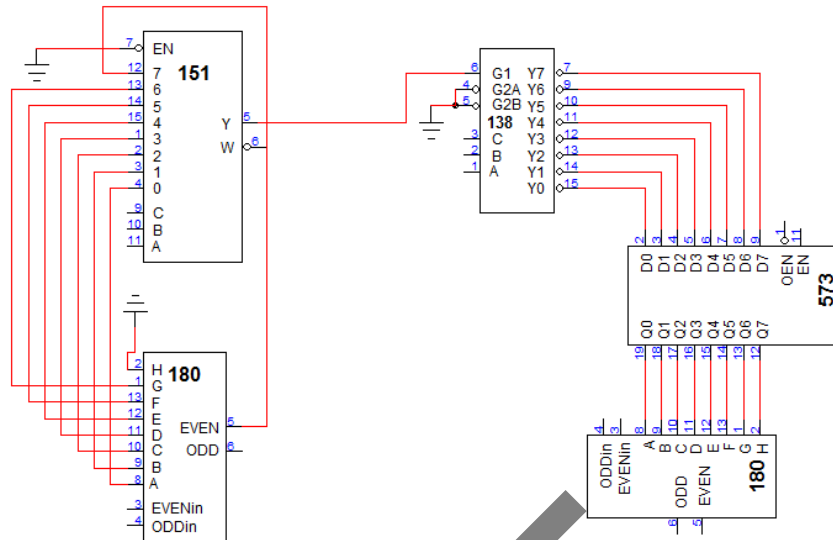
Hình 2. 58. Mạch kiểm tra parity lẻ 9 bit

Ví dụ 2: Tạo bộ kiểm tra tính chẵn 16 bit dùng 74180



Hình 2. 59. Mạch kiểm tra parity chẵn 16 bit

Ví dụ 3: Sơ đồ khối của mạch tạo và kiểm tra tính chẵn 8 bit:



Hình 2. 60. Sơ đồ khối của mạch tạo và kiểm tra tính chẵn

2.9. 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

$$D + P + 1 \leq 2^P \quad (2.15)$$

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.

2.9.1. 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 (2.15) 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 $P_8P_4P_2P_1$
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	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 2-28. 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 $\Rightarrow D = 6 \Rightarrow P = 4$ thỏa mãn công thức (2.15).

10	9	8	7	6	5	4	3	2	1
1	0	P_8	1	1	0	P_4	1	P_2	P_1

Tìm P_1 :

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

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

$$P_1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0 \rightarrow P_1 = 0$$

Tìm P_2 :

$$P_2 \oplus \text{vị trí 3} \oplus \text{vị trí 6} \oplus \text{vị trí 7} \oplus \text{vị trí 10} = 0.$$

$$P_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 = 0.$$

$$P_2 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0 \rightarrow P_2 = 1$$

Tìm P_4 :

$$P_4 \oplus \text{vị trí 5} \oplus \text{vị trí 6} \oplus \text{vị trí 7} = 0.$$

$$P_4 \oplus D_2 \oplus D_3 \oplus D_4 = 0.$$

$$P_4 \oplus 0 \oplus 1 \oplus 1 = 0 \rightarrow P_4 = 0$$

Tìm P_8 :

$$P_8 \oplus \text{vị trí 9} \oplus \text{vị trí 10} = 0.$$

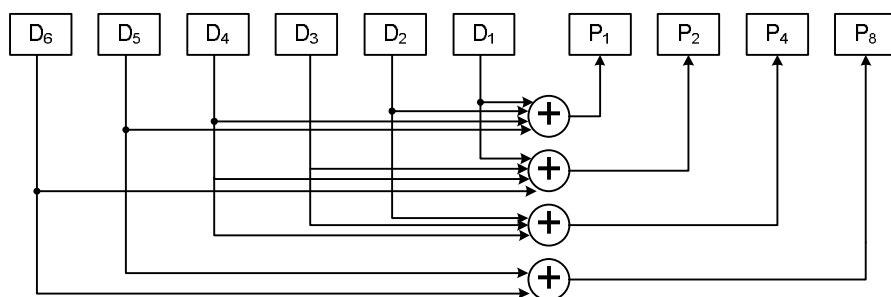
$$P_8 \oplus D_5 \oplus D_6 = 0.$$

$$P_8 \oplus 0 \oplus 1 = 0 \rightarrow P_8 = 1$$

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 2.61.



Hình 2. 61. Mạch tạo mã Hamming

2.9.2. 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 từ mã nhận được bị 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:

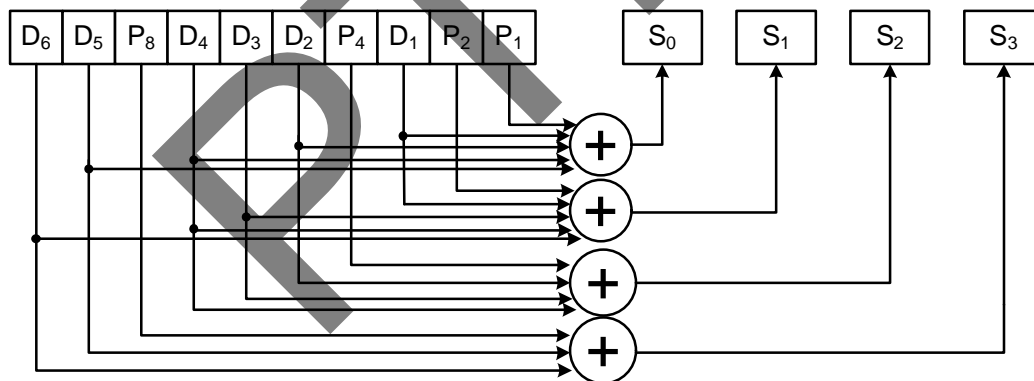
$$S_3 = P_8 \oplus D_5 \oplus D_6$$

$$S_2 = P_4 \oplus D_2 \oplus D_3 \oplus D_4$$

$$S_1 = P_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6$$

$$S_0 = P_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5$$

Từ đó, vẽ được mạch giải mã như hình 2.62.



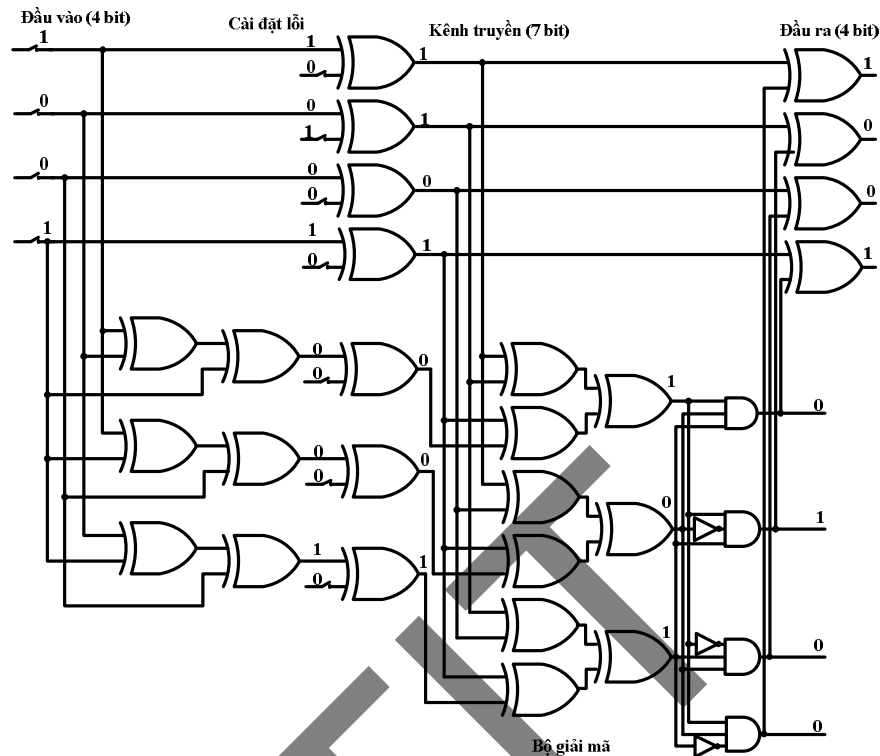
Hình 2. 62. Mạch giải mã Hamming

Sau đây giới thiệu mạch tạo mã và giải mã Hamming 4 bit (Hình 2.63).

Mạch này mô tả cách sử dụng mã Hamming 4 bit để phát hiện lỗi và sửa lỗi. Nó sử dụng 4 chuyển mạch đầu vào để thiết lập giá trị của 4 bit cho đầu vào bộ mã hóa, đồng thời mạch cũng cho ra giá trị của 3 bit chẵn/lẻ, sau đó các bit chẵn/lẻ này được truyền cùng 4 bit giá trị.

Bảy chuyển mạch và bảy cổng XOR cho phép chèn lỗi vào các dữ liệu đã được truyền đi. Nếu chuyển mạch ngắt thì bit dữ liệu tương ứng được truyền đi một cách bình thường. Nếu

chuyển mạch đóng tương đương với một bit lỗi được truyền (cổng XOR tiến hành đảo bit tương ứng).



Hình 2. 63. Mạch tạo mã và giải mã Hamming 4 bit

Đầu tiên, mạch đặt tất cả 7 chuyển mạch lỗi về vị trí ngắt. Sau đó lựa chọn vài giá trị của mạch mã hóa với chuyển mạch đóng và quan sát các bit chẵn/lẻ cách hoạt động của bộ giải mã. Tiếp theo, thay đổi các chuyển mạch đánh lỗi để chèn các bit lỗi vào dữ liệu và quan sát lại cách hoạt động của bộ giải mã. Điều gì sẽ xảy ra nếu bạn kích hoạt nhiều hơn một chuyển mạch đánh lỗi?

Khối giải mã nhận được 7 bit mã và khôi phục lại thông tin ban đầu (4 bit dữ liệu).

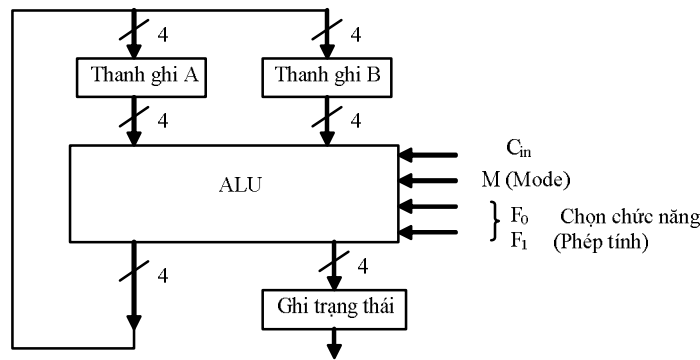
2.10. ĐƠN VỊ SỐ HỌC VÀ LOGIC (ALU).

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 hợp 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 2.64.



Hình 2. 64. Sơ đồ khối của bộ ALU 4 bit

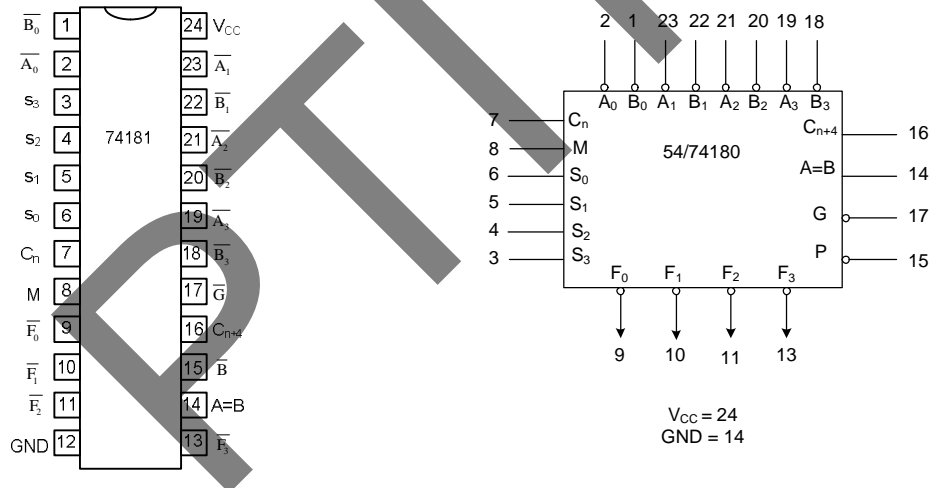
M là đầu vào chọn phép tính số học hay logic.

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 thanh ghi.

Hình 2.65 giới thiệu sơ đồ chân và sơ đồ ký hiệu của 74181ALU



Hình 2. 65. Sơ đồ chân và ký hiệu của IC 74181

Bảng chức năng của IC 74181.

Khi đầu vào điều khiển chọn Mod (M) ở mức cao thì tất cả các đầu vào nhớ bị ngăn cấm và thiết bị chỉ hoạt động trong chế độ logic. Khi M ở mức thấp thì các đầu vào nhớ được kích hoạt và thiết bị hoạt động ở chế độ số học với hai số 4 bit. Thiết bị ghép nối các đầu vào nhớ của từng tầng bằng cách sử dụng đầu ra C_{n+4} hoặc sử dụng tín hiệu P (Carry Propagate) và G (Carry Generate). Các lỗi P và G không ảnh hưởng đến đầu vào nhớ. Khi không quan tâm đến tần số, có thể sử dụng chế độ nhớ nối tiếp bằng cách nối đầu ra nhớ (C_{n+4}) với đầu vào nhớ (C_n) của tầng tiếp theo.

Đầu vào lựa chọn chế độ hoạt động				Các đầu vào và đầu ra hoạt động ở mức tích cực cao	
S_3	S_2	S_1	S_0	Logic (M=H)	Số học (M=L) ($C_n=H$)
L	L	L	L	\overline{A}	A
L	L	L	H	$\overline{A+B}$	$A+B$
L	L	H	L	$\overline{A.B}$	$A+\overline{B}$
L	L	H	H	0	Trừ 1 (của bù 2)
L	H	L	L	$\overline{A.B}$	A cộng $A.\overline{B}$
L	H	L	H	\overline{B}	$(A+B)$ cộng $A.\overline{B}$
L	H	H	L	$A \oplus B$	A trừ B trừ 1
L	H	H	H	$A.\overline{B}$	$(A.B)$ trừ 1
H	L	L	L	$\overline{A+B}$	A cộng $A.B$
H	L	L	H	$\overline{A \oplus B}$	A cộng B
H	L	H	L	\overline{B}	$(A+\overline{B})$ cộng $A.B$
H	L	H	H	$A.B$	$A.B$ trừ 1
H	H	L	L	1	A cộng A^*
H	H	L	H	$A+\overline{B}$	$(A+B)$ cộng A
H	H	H	L	$A+B$	$(A+\overline{B})$ cộng A
H	H	H	H	A	A trừ 1

Bảng 2-29. Bảng chức năng của IC 74181 khi M ở mức logic thấp

A^* : từng bit được dịch sang vị trí có trọng số cao hơn

Đầu ra $A=B$ ở mức cao khi cả 4 đầu ra \overline{F} ở mức cao và nó còn được dùng để báo hiệu logic tương đương khi thiết bị hoạt động ở chế độ trừ. Đầu ra $A=B$ là cổng collector để hở và có thể nối với các đầu ra $A=B$ khác bằng cổng AND để thực hiện so sánh khi nhiều hơn 4 bit. Tín hiệu $A=B$ có thể được sử dụng cùng với tín hiệu C_{n+4} để chỉ thị $A>B$ hoặc $A<B$.

Bảng chức năng mô tả hoạt động của khối số học mà không có đầu vào nhớ. Đầu vào nhớ được đính kèm theo từng chế độ hoạt động. Ví dụ, khi lựa chọn mã LHHL để phát ra trạng thái A trừ B trừ 1 (bù 2) mà không có đầu vào nhớ và phát ra tín hiệu A trừ B khi có đầu vào nhớ.

Do phép trừ được biểu diễn bởi phép bù (bù 1), đầu ra nhớ chính là lỗi mượn, do vậy tín hiệu nhớ được phát ra khi không bị dưới mức và không được phát ra khi dưới mức.

Thiết bị được sử dụng cả hai trường hợp khi đầu vào tích cực thấp và tích cực cao tương ứng với đầu ra tích cực thấp và tích cực cao.

Đầu vào lựa chọn chế độ hoạt động				Các đầu vào và đầu ra hoạt động ở mức tích cực thấp	
S_3	S_2	S_1	S_0	Khối logic (M=H)	Khối số học (M=L) ($C_n = L$)
L	L	L	L	\overline{A}	A trừ 1
L	L	L	H	$\overline{A.B}$	AB trừ 1
L	L	H	L	$\overline{A+B}$	$A.\overline{B}$ trừ 1
L	L	H	H	1	trừ 1 (của bù 2)
L	H	L	L	$\overline{A+B}$	A cộng ($A+\overline{B}$)
L	H	L	H	\overline{B}	A. B cộng ($A+\overline{B}$)
L	H	H	L	$\overline{A \oplus B}$	A trừ B trừ 1
L	H	H	H	$\overline{A+B}$	$A+\overline{B}$
H	L	L	L	$A.\overline{B}$	A cộng ($A+B$)
H	L	L	H	$A \oplus B$	A cộng B
H	L	H	L	B	$A\overline{B}$ cộng ($A+B$)
H	L	H	H	$A+B$	$A+B$
H	H	L	L	0	$A+A^*$
H	H	L	H	$A.\overline{B}$	A . B cộng A
H	H	H	L	$A.B$	$A.\overline{B}$ cộng A
H	H	H	H	A	A

Bảng 2-30. Bảng chức năng của IC 74181 khi M ở mức logic cao

TÓM TẮT

Trong chương này, 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 2. Trọng tâm của chương là nắm vững đặc điểm mạch tổ hợp và phương pháp chung khi thiết kế, phân tích mạch tổ hợp. Vì vậy, chương đã giới thiệu một cách chọn lọc bộ mã hoá, bộ giải mã, bộ hợp 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.

Khi phân tích mạch tổ hợp đã cho, có thể viết ra hàm logic đầu ra cho từng cấp của sơ đồ, rồi tiến hành tối thiểu hoá hàm logic đó để biểu thị rõ mối quan hệ giữa đầu ra với đầu vào.

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 hợp 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 hợp kênh để thực hiện hàm logic $f = A..B.C + \overline{A}.\overline{B}$
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 3. MẠCH LOGIC TUẦN TỰ

GIỚI THIỆU.

Chương 2 đã nghiên cứu về phép phân tích và thiết kế các mạch logic tổ hợp. Mặc dù rất qua trọng nhưng nó chỉ là một phần của các hệ thống kỹ thuật số. Một phần qua 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 2.

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

3.1. KHÁI NIỆM CHUNG VÀ MÔ HÌNH TOÁN HỌC

3.1.1. Khái niệm chung

Trong chương này, 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.

3.1.2. 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.

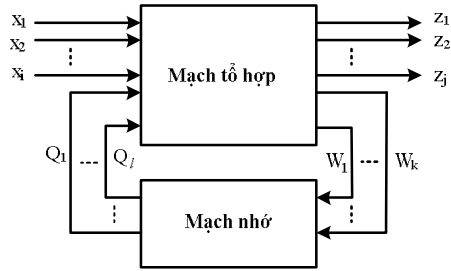
Q - tập trạng thái trong trước đó của mạch.

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 3-1. Sơ đồ khối mạch tuần tự

Trong phương trình toán học của mạch tuần tự 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.

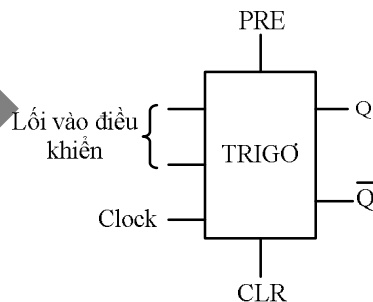
$Q(n)$: là trạng thái bên trong trước đó.

Để hiểu rõ hơn về mạch tuần tự, xét các phần tử có trong mạch. Mạch logic tổ hợp đã được xét ở chương 2 còn phần tử nhớ chính là các trigơ.

3.2. PHẦN TỬ NHỚ CỦA MẠCH TUẦN TỰ

3.2.1. 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.



Hình 3-2. Sơ đồ tổng quát của một trigơ

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.

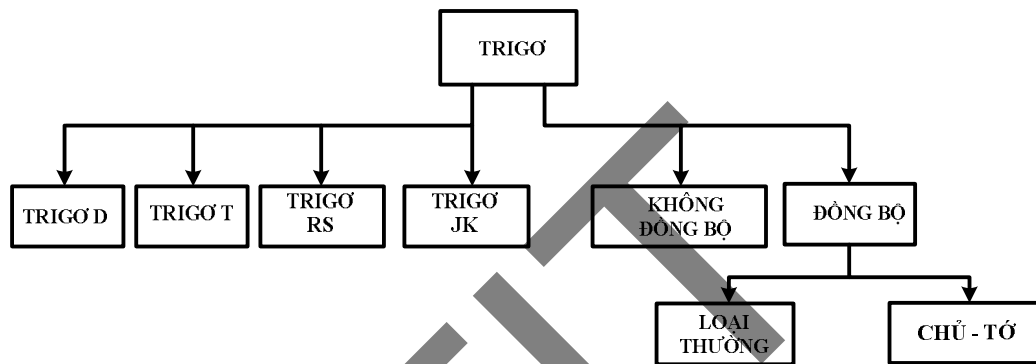
Trigơ có từ 1 đến một vài lối điều khiển, có hai đầu ra luôn luôn ngược nhau là Q và \bar{Q} . Tùy từng loại trigơ có thể có thêm các đầu vào lập (PRESET) và đầu vào xoá (CLEAR). Ngoài ra, trigơ còn có đầu vào đồng bộ (CLOCK). Hình 3-2 là sơ đồ khối tổng quát của trigơ.

Phân loại:

Theo chức năng làm việc của các đầu vào điều khiển: hiện nay thường sử dụng loại trigơ 1 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 3-3.

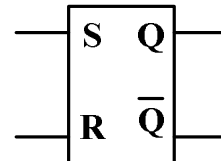


Hình 3-3. Sơ đồ phân loại trigơ

3.2.1.1. Trigơ RS

Trigơ RS không đồng bộ

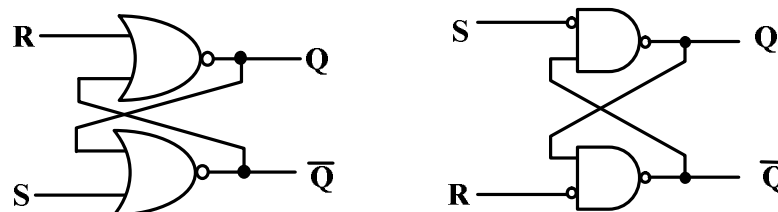
Trigơ RS là loại có hai đầu vào điều khiển S, R. Chân S gọi là đầu vào "lập" (SET) và R được gọi là đầu vào "xoá" (RESET).



Hình 3-4 là ký hiệu của trigơ RS trong các sơ đồ logic.

Hình 3-4. Sơ đồ ký hiệu trigơ RS không đồng bộ

Hình 3-5 là sơ đồ nguyên lý của trigơ RS và RS đồng bộ.



Hình 3-5. Sơ đồ nguyên lý của trigơ RS không đồng bộ cổng NOR và cổng NAND

Nguyên lý hoạt động của trigơ RS không đồng bộ cổng NAND

Nguyên lý hoạt động của trigơ RS không đồng bộ cổng thông qua bảng trạng thái 3-1.

\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 3-1. Bảng trạng thái của trigơ RS cổng NAND

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).

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.

Căn cứ vào bảng Các nô hình 3-2, tìm được phương trình đặc trưng:

$$\begin{cases} Q^{n+1} = S + \bar{R} \cdot Q^n \\ \bar{S} \cdot \bar{R} = 1 \end{cases} \quad (3.1)$$

Hình 3-6

Bảng Các nô của Q^{n+1}

$\bar{S} \bar{R}$	00	01	11	10
Q^n				
0	X	1	0	0
1	X	1	1	0

Bảng trạng thái của trigơ RS dùng cổng NOR được cho ở bảng 3-2.

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 3-2. Bảng trạng thái của trigơ RS

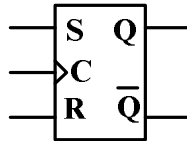
C	S	R	Q^{n+1}	Mod hoạt động
0	x	x	Q^n	Nhớ
1	0	0	Q^n	Nhớ
1	0	1	0	Xoá
1	1	0	1	Lập
1	1	1	x	Cấm

Bảng 3-3. Bảng trạng thái của trigơ RS đồng bộ cổng NAND

Đặc điểm của Trơ cơ bản:

- Ưu điểm: Mạch đơn giản, có thể nhớ 1 bit, là cơ sở để cấu trúc các Trơ 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.

Trơ RS đồng bộ



Hình 3-7a). Sơ đồ ký hiệu trơ RS đồng bộ

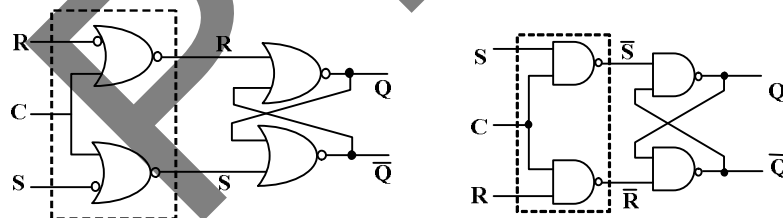
Sơ đồ ký hiệu của trơ RS đồng bộ được chỉ ra trên hình 3-7a).

Để khắc phục nhược điểm của loại trơ 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 trơ RS đồng bộ được trình bày ở hình 3-7b).

Nguyên lý hoạt động của trơ RS đồng bộ cổng NAND:

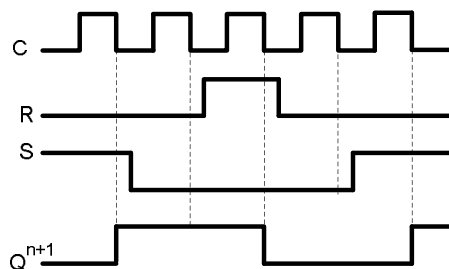
Khi $C = 0$ các cổng vào bị ngắt, trơ RS cơ bản không tiếp thu tín hiệu vào, mạch được duy trì trạng thái cũ. Khi $C = 1$ các cổng vào thông, mạch sẵn sàng tiếp thu tín hiệu vào R, S.

Nguyên lý hoạt động trơ RS đồng bộ cũng giống như trơ RS cơ bản, chúng chỉ khác là chỉ tiếp nhận tín hiệu đầu vào R, S khi $C = 1$. Vì lý do đó nên gọi là trơ RS đồng bộ. Bảng trạng thái được chỉ ra trên bảng 3-3.



Hình 3-7b. Sơ đồ nguyên lý của trơ RS đồng bộ cổng NOR và cổng NAND

Đồ thị thời gian dạng xung được trình bày trên hình 3-8.



Hình 3-8. Đồ thị thời gian dạng xung của trơ 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.

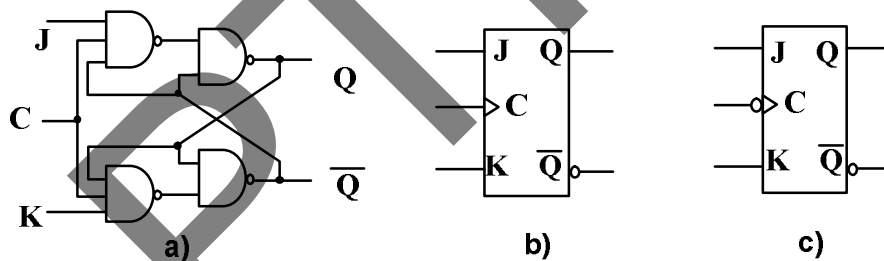
3.2.1.2. Trgơ JK

Trgơ JK là loại trigơ có hai đầu vào điều khiển J, K. Trgơ 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. Trgơ có thể lập hay xoá trong khoảng thời gian ứng với sườn âm hoặc sườn dương của xung đồng bộ C. Trgơ JK thuộc loại đồng bộ.

Sự hoạt động của trigơ JK được trình bày bằng bảng trạng thái 3-4

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	\bar{Q}^n	Thay đổi trạng thái theo mỗi xung nhịp

Bảng 3-4. Bảng trạng thái của trigơ JK đồng bộ



Hình 3-9. 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.

Từ bảng trạng thái, lập bảng Các nô để tìm phương trình đặc trưng: Coi 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	0	1	1
1	1	0	0	1

Trong bảng trên, các ô chứa 1 được圈 ra để tìm phương trình đặc trưng. Các ô 1 ở hàng JK=01 và JK=10 được圈 ra cùng một nhóm và gán cho phương trình $J \cdot \bar{Q}^n$. Các ô 1 ở hàng JK=00 và JK=11 được圈 ra cùng một nhóm và gán cho phương trình $\bar{K} \cdot Q^n$.

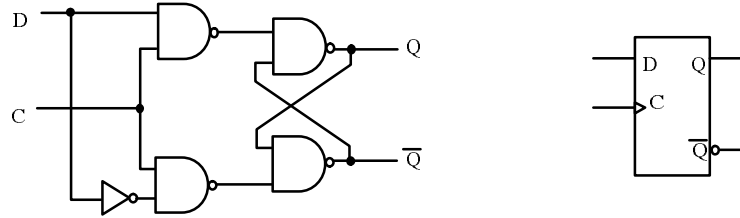
Hình 3-10. Bảng Các nô tìm phương trình đặc trưng của trigơ JK

Từ hình 3-10, có phương trình:

$$Q^{n+1} = J \cdot \overline{Q^n} + \overline{K} \cdot Q^n \quad (3.2)$$

Sơ đồ nguyên lý và sơ đồ ký hiệu của trigơ JK được trình bày ở hình 3-9.

3.2.1.3. Triger D



Hình 3-11. Sơ đồ nguyên lý và sơ đồ ký hiệu của trigơ D đồng bộ

Triger D có sơ đồ nguyên lý và sơ đồ kí hiệu trên hình 3-11 có 1 đầu vào D và 2 đầu ra Q và \overline{Q} .

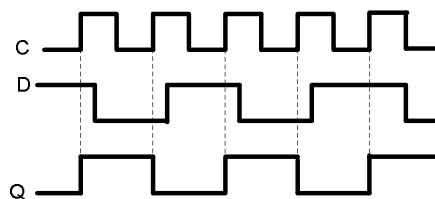
Bảng 3-11. Bảng trạng thái của trigơ D

Q^n	D	Q^{n+1}
0	0	0
0	1	1
1	0	0
1	1	1
Bảng 3-11. Bảng trạng thái trigơ D		

Từ bảng trạng thái 3-11, tìm được phương trình đặc trưng của trigơ D:

$$Q^{n+1} = D \quad (3.3)$$

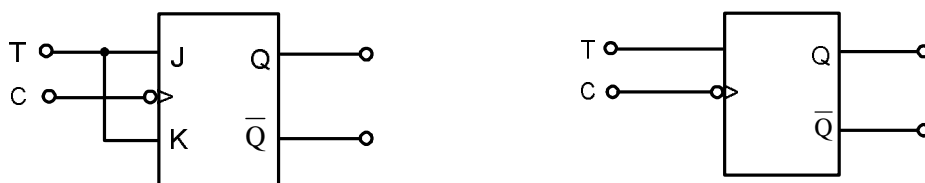
với điều kiện đã xuất hiện sườn dương C, nếu điều kiện này không thỏa mãn, trigơ giữ nguyên trạng thái cũ. Đồ thị thời gian của dạng xung của trigơ D được trình bày trên hình 3-11.



Hình 3-11. Dạng xung của Triger D

3.2.1.4. Triger T (Toggle – lật)

Triger 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 3-12.



Hình 3-12. Kí hiệu logic của Trơ T

Nếu cho $J = K = T$, trơ JK chuyển thành trơ T. Phương trình đặc trưng

$$Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n = T\bar{Q}^n + \bar{T}Q^n$$

$$Q^{n+1} = T \oplus Q^n \quad (3.4)$$

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.

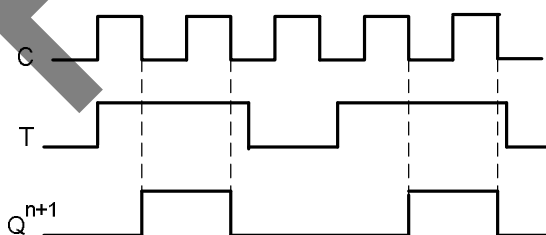
$T = 1$ có sườn âm của C thì $Q^{n+1} = \bar{Q}^n$

Bảng 3-12: Bảng trạng thái của Trơ T

Q^n	T	Q^{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Bảng 3-12: Bảng trạng thái của Trơ T

Đồ thị thời gian dạng sóng của Trơ T được chỉ ra trên hình 3-13.



Hình 3-13: Đồ thị thời gian dạng sóng của Trơ T

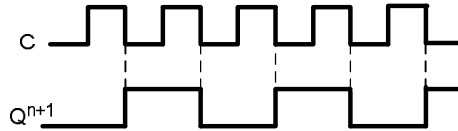
3.2.1.3. Trơ T'

Trơ T' là mạch điện chỉ có chức năng chuyển đổi trạng thái trong điều kiện định thời của C. Trơ T' là Trơ T mà $T = 1$. Phương trình đặc trưng:

$$Q^{n+1} = T \oplus Q^n = 1 \oplus Q^n = \bar{Q}^n \quad (3.5)$$

với điều kiện xuất hiện sườn âm của C.

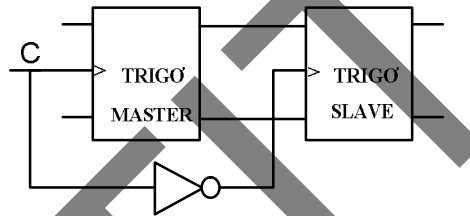
Đồ thị thời gian dạng sóng của Trơ T' được trình bày trên hình 3-14.



Hình 3-14. Đồ thị thời gian dạng sóng của Trơ T'

Nhận xét: Từ các bảng trạng thái của các trơ trên thấy rằng: Các trơ 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 thoả mãn điều kiện $Q^k = Q$. Còn trơ T và trơ 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 trơ JK hoặc là '1' đối với trơ T. Như vậy, trơ D, trơ RS có thể làm việc ở cả hai chế độ: đồng bộ và không đồng bộ còn trơ T và trơ JK chỉ có thể làm việc ở chế độ đồng bộ.

3.2.1.3. Các loại trơ Chủ- tớ (MS-Master- Slave).



Hình 3-15. Cấu trúc của trơ MS

Do các loại trơ 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 trơ MS khắc phục được nhược điểm này. Đầu ra của trơ MS thay đổi tại sườn dương và sườn âm của xung nhịp, nên cấu trúc của nó gồm 2 trơ 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 trơ hoạt động. Về nguyên tắc hoạt động của loại trơ MS (RS-MS, JK-MS, D-MS, T-MS) hoàn toàn giống như các loại trơ thông thường (RS, JK, D, T).

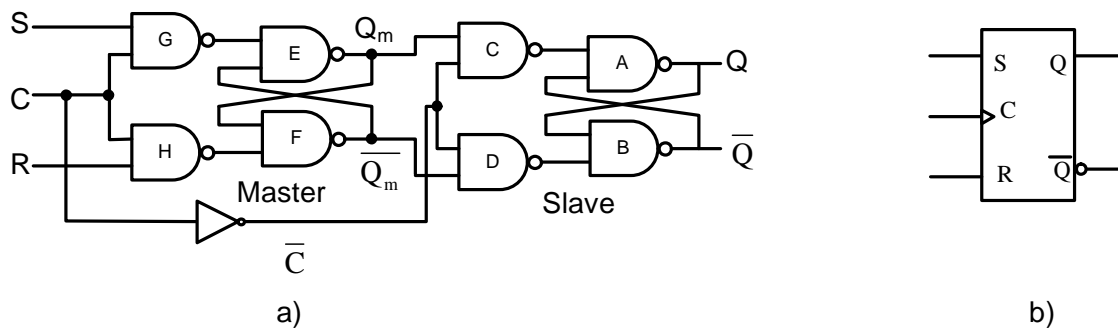
Cấu trúc chung của một trơ MS được minh hoạ ở hình 3-15.

a) Trơ RS Master- Slave

Mạch này giải quyết triệt để vấn đề trực tiếp điều khiển, đó là nhược điểm của các loại trơ trên.

Cấu trúc mạch và nguyên lí hoạt động.

Hình 3-16 là sơ đồ của trơ RS Master Slave gồm 2 trơ RS cơ bản ghép liên tiếp với nhau, một là trơ RS Master (trơ chủ), một là trơ RS Slave (trơ tớ), xung đồng hồ cung cấp cho chúng là đảo nhau (qua mạch đảo).



Hình 3-16: Trơ RS Master- Slave. Sơ đồ logic (a) ; kí hiệu logic (b)

Nguyên lí hoạt động:

+ Khi $C = 0$ cổng G, H ngắt nên trơ 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.

$C = 1$, trơ 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 $\bar{C} = 0$, trơ 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, Trơ Master ngắt, khi đó \bar{C} đột biến lên 1, Trơ Slave tiếp nhận tín hiệu đã được trơ Master ghi nhớ từ thời gian $C = 1$, nghĩa là trơ Slave chuyển đổi trạng thái theo biểu thức logic:

$$Q^{n+1} = S + \bar{R}.Q^n \quad (3.6)$$

Với điều kiện đã xuất hiện sườn âm của xung đồng hồ C.

Như vậy trơ 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) Trơ JK Master - Slave

Cấu trúc mạch và nguyên lí hoạt động.

Loại trigơ RS master Slave được trình bày trên hình 3-16 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 $\overline{Q}_m = 1$.

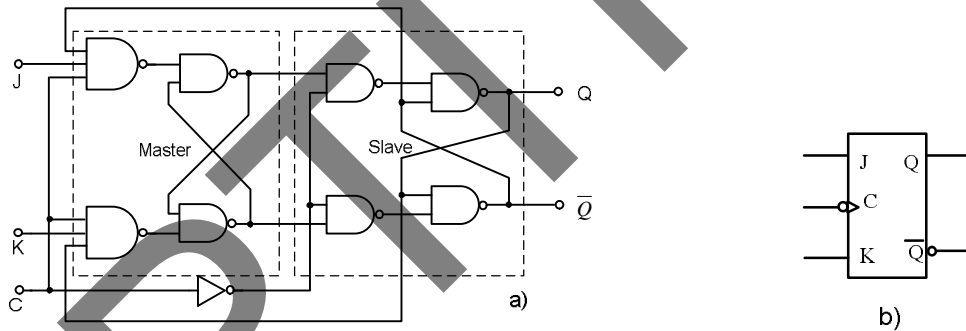
Chú ý một điểm sau: Khi $C = 1$, trigơ Master Slave, Q và \overline{Q} không đổi trạng thái và là đảo của nhau. Do đó chỉ cần đem mức các đầu ra Q và \overline{Q} đưa đến đầu vào của G và H thì có thể khắc phục được tình trạng cả Q_m và \overline{Q}_m đều bằng 1.

Để giải quyết vấn đề trên người ta cải tiến trigơ RS Master Slave theo sơ đồ được trình bày trên hình 3-17, 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, thấy nguyên tắc hoạt động giống như trigơ RS Master Slave, chỉ khác sự tương đương sau của tín hiệu đầu vào:

$$S = J \cdot \overline{Q}^n \quad (3.7)$$

$$R = K \cdot Q^n \quad (3.8)$$



Hình 3-17: Trgơ JK Master Slave Sơ đồ logic (a) Kí hiệu logic (b)

Áp dụng công thức (3.6) tìm được:

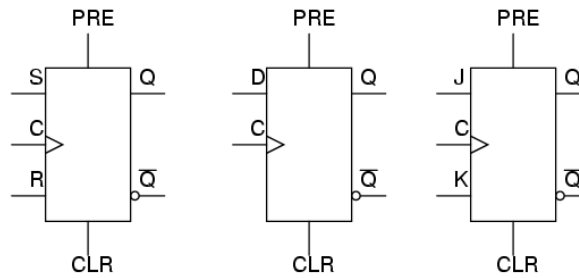
$$Q^{n+1} = S + \overline{R} \cdot Q^n = J \cdot \overline{Q}^n + \overline{K} \cdot Q^n \cdot Q^n = J \cdot \overline{Q}^n + (\overline{K} + Q^n) \cdot Q^n$$

$$Q^{n+1} = J \cdot \overline{Q}^n + \overline{K} \cdot Q^n \quad (3.9)$$

Với điều kiện đã xuất hiện sườn âm của C, công thức (3.9) là phương trình đặc trưng của trigơ JK.

3.2.2. Đầu vào không đồng bộ của trigơ.

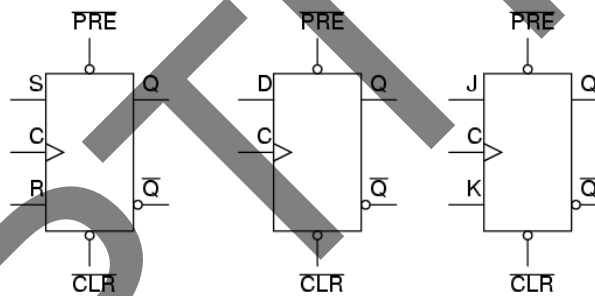
Các đầu vào dữ liệu thông thường của trigơ như D, S, R, J hoặc K là những đầu vào đồng bộ vì chúng có ảnh hưởng lên các đầu ra khi có tác động của xung Clock. Trong thực tế các trigơ còn có thêm 2 đầu vào không đồng bộ, các lối này tác động trực tiếp lên các đầu ra mà không phụ thuộc vào xung Clock. Các đầu vào này thường được ký hiệu là: PRE (lập) và CLR (R - xóa) hoặc \overline{PRE} và \overline{CLR} (\overline{R}). Xem hình 3-18 và 3-19.



Hình 3-18. Đầu vào không đồng bộ ở mức tích cực cao

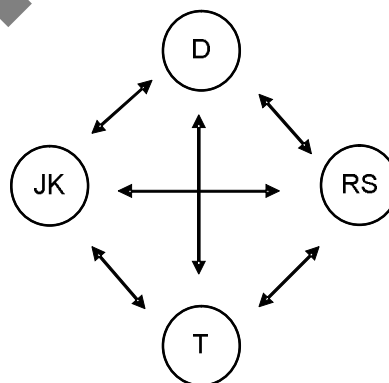
Khi đầu vào PRE (preset) được kích hoạt thì trigơ sẽ ở trạng thái lập ($Q = 1$ và $\bar{Q} = 0$), lúc này trigơ không phụ thuộc vào các đầu vào đồng bộ hay xung Clock. Khi lỗi CLR (clear) được kích hoạt thì trigơ sẽ ở trạng thái xóa ($Q = 0$ và $\bar{Q} = 1$), lúc này trigơ cũng không phụ thuộc vào các 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 \bar{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 3-19. Đầu vào không đồng bộ ở mức tích cực thấp

3.2.3. Chuyển đổi giữa các loại trigơ.



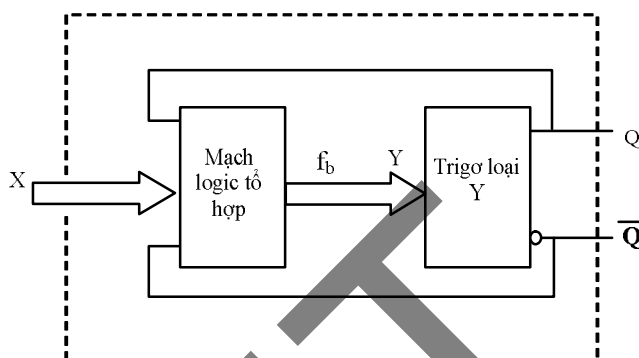
Hình 3-20 . Các khả năng chuyển đổi trigơ

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.

3.2.3.1. Phương pháp chuyển đổi giữa các loại trigơ.

1. Nguyên tắc chung



Hình 3-21. 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 3-21.

Từ sơ đồ hình 3-21 có thể suy ra cách 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 3-13:

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 3-13. 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.

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:

$$\begin{cases} Q^{n+1} = S + \overline{R}.Q^n \\ R.S = 0 \end{cases} \quad (3.10)$$

còn phương trình đặc trưng của trigơ JK có dạng:

$$Q^{n+1} = J.\overline{Q}^n + \overline{K}.Q^n \quad (3.11)$$

So sánh (3.10) và (3.11) tìm được:

$$S = J.\overline{Q}^n$$

$$R = K$$

Tuy nhiên, điều kiện ràng buộc $R.S = 0$ nên khi $J = K = 1$; $Q^n = 0$ thì

$$\begin{cases} R = K = 1 \\ S = J.\overline{Q}^n = 1 \end{cases}$$

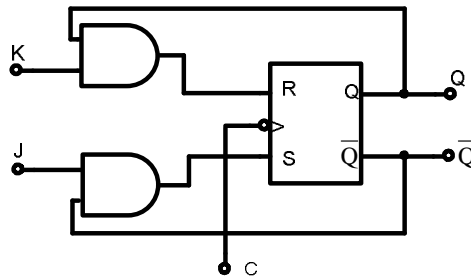
Sẽ không thỏa mãn điều kiện $R.S = 0$.

Có thể biến đổi lại:

$$\begin{aligned} Q^{n+1} &= J.\overline{Q}^n + \overline{K}.Q^n + Q^n.\overline{Q}^n = J.\overline{Q}^n + (\overline{K} + \overline{Q}^n)Q^n \\ Q^{n+1} &= J.\overline{Q}^n + \overline{K.Q^n}.Q^n \end{aligned} \quad (3.12)$$

So sánh (3-10) và (3-12) tìm được hàm kích:

$$\begin{cases} S = J.Q^n \\ R = K.Q^n \end{cases} \quad (3.13)$$



Hình 3-22. Mạch điện chuyển đổi từ RS sang JK

Hình 3-22 là mạch điện chuyển đổi từ RS \rightarrow JK (thỏa mãn điều kiện R.S = 0).

+ Dùng bảng kích thích.

$Q^n \backslash JK$	00	01	11	10
0	0	0	1	1
1	x	0	0	x

a) Đối với S

$Q^n \backslash JK$	00	01	11	10
0	x	x	0	0
1	0	1	1	0

b) Đối với R

Hình 3-23. Bảng Các nô xác định sự phụ thuộc của S, R vào Q, J, K

Từ bảng hàm kích thích 3-13. Nếu xem Q^n , J, K là biến và S, R là hàm, xây dựng được hai bảng Các nô như hình 3-23. Từ bảng Các nô, tìm được kết quả giống công thức (3-13).

$$\begin{cases} S = J.Q^n \\ R = K.Q^n \end{cases}$$

Tương tự, tìm được hàm logic chuyển đổi:

RS \rightarrow D

$$\begin{cases} R = \bar{D} \\ S = D \end{cases} \quad (3.14)$$

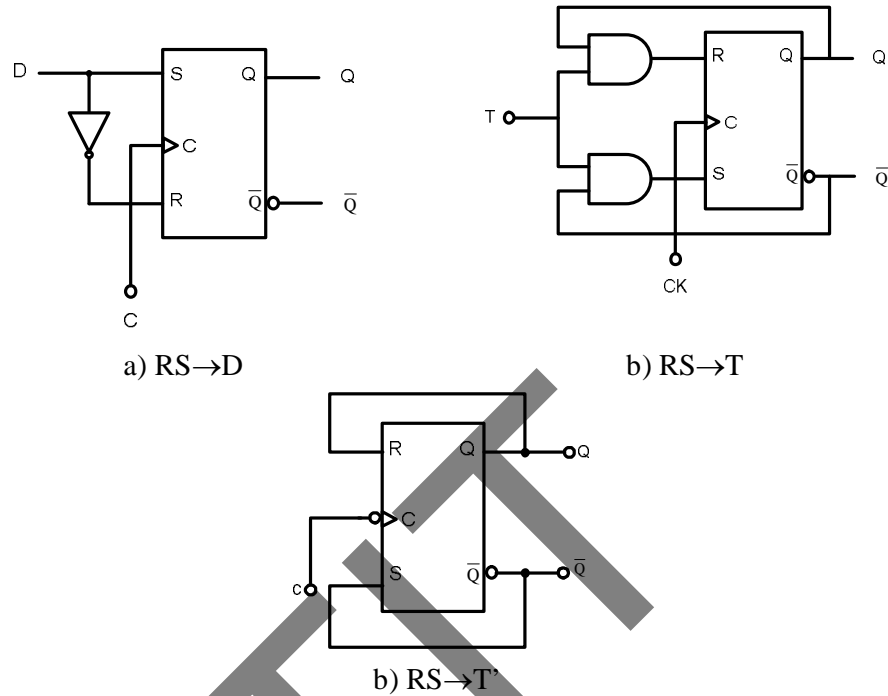
RS \rightarrow T

$$\begin{cases} R = T.Q^n \\ S = T.Q^n \end{cases} \quad (3.15)$$

RS \rightarrow T'

$$\begin{cases} R = Q^n \\ S = \overline{Q^n} \end{cases} \quad (3.16)$$

Mạch chuyển đổi được trình bày trên hình (3-24)



Hình 3-24. Mạch chuyển đổi từ trigger RS trở thành a) D ; b) T ; c) T'

b) Chuyển đổi từ trigger JK sang RS, D, T, T'

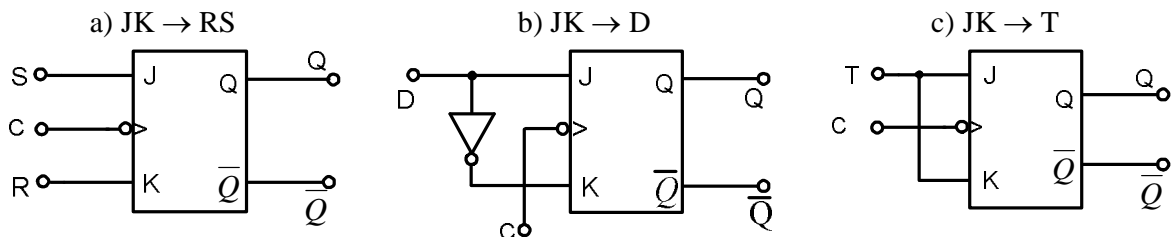
Bằng phương pháp tương tự như phần trước tìm được các biểu thức và sơ đồ:

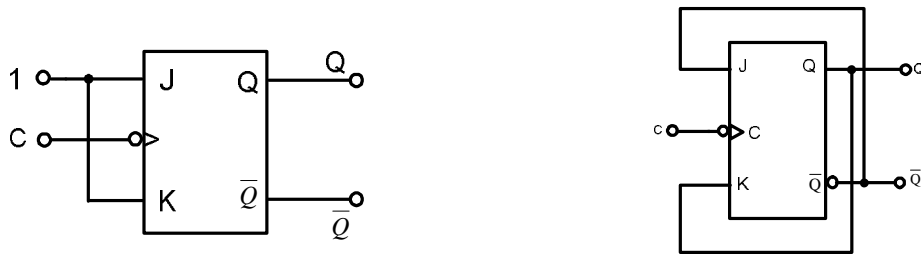
+ Từ trigger JK sang RS.

Biểu thức:

$$\begin{cases} J = S \\ K = R \end{cases} \quad (3.17)$$

Sơ đồ 3-25a)





d) $JK \rightarrow T'$

Hình 3-23. Chuyển đổi từ trigơ JK sang a) RS ; b) D ; c) T ; d) T'

+ Từ JK sang D

Biểu thức:

$$\left. \begin{array}{l} J = D \\ K = \overline{D} \end{array} \right\} \quad (3.18)$$

Sơ đồ hình 3-25b).

+ Từ JK sang T

Biểu thức:

$$\left. \begin{array}{l} J = T \\ K = T \end{array} \right\} \quad (3.19)$$

Sơ đồ hình 3-25c).

+ Từ JK sang T'

Biểu thức:

$$\left. \begin{array}{l} J = T = 1 \\ K = T = 1 \end{array} \right\} \quad (3.20)$$

Hoặc

$$\left. \begin{array}{l} J = \overline{Q^n} \\ K = Q^n \end{array} \right\} \quad (3.21)$$

Sơ đồ hình 3-25d).

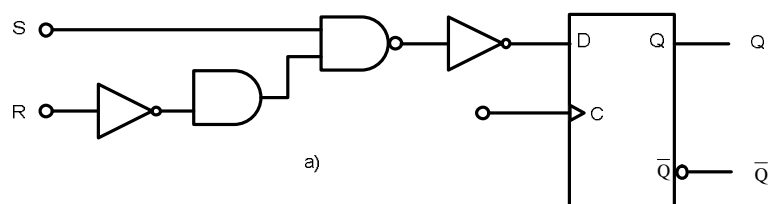
c) Chuyển từ D sang các loại

+ Từ D sang RS

Cân bằng hai phương trình đặc trưng. Tìm được biểu thức:

$$D = S + \overline{R}Q^n \quad (3.22)$$

Sơ đồ hình 3-26a)

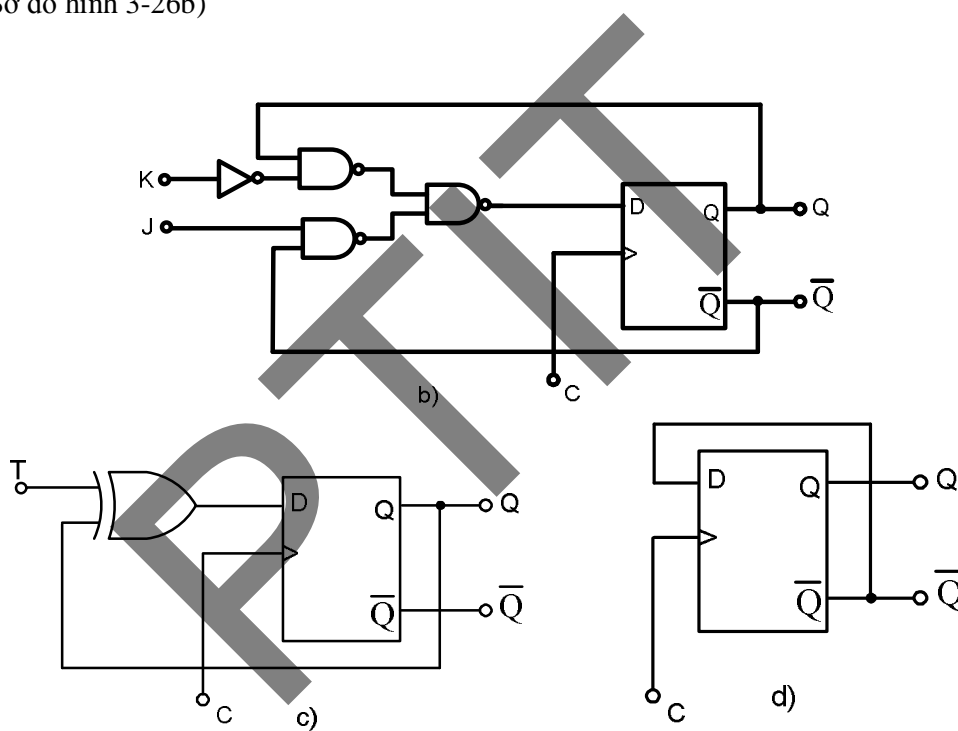


+ Từ D sang JK

Biểu thức:

$$D = J\overline{Q^n} + \overline{K}.Q^n \quad (3.23)$$

Sơ đồ hình 3-26b)



Hình 3-26. Mạch điện biến đổi từ D sang a) RS ; b) JK ; c) T ; d) T'

+ Từ D sang T

Biểu thức:

$$D = T \oplus Q^n \quad (3.24)$$

Sơ đồ hình 3-26c)

+ Từ D sang T'

Biểu thức:

$$D = T \oplus Q^n$$

$$T = 1 \rightarrow D = \overline{Q^n} \quad (3.25)$$

Sơ đồ hình 3-26d)

d) Chuyển từ T sang RS, JK, D.

+ Từ T sang RS

Biểu thức:

$$T = S.\overline{Q^n} + R.Q^n \quad (3.26)$$

Sơ đồ hình 3-27a).

+ Từ J sang JK:

Biểu thức:

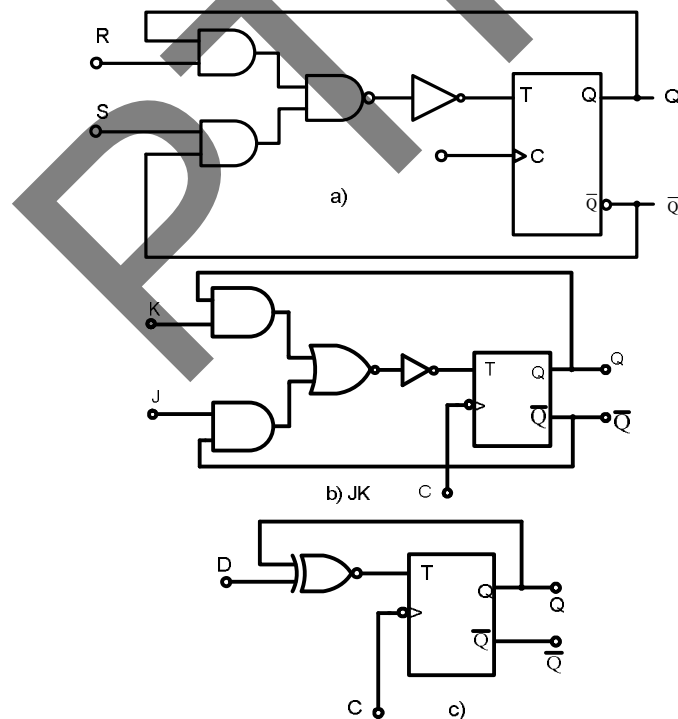
$$T = J\overline{Q^n} + \overline{K}.Q^n \quad (3.27)$$

Sơ đồ hình 3-27a) 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 3-27c).



Hình 3-27. Mạch điện chuyển từ T sang a) RS ; b) JK ; c) D

3.3. ỨNG DỤNG CỦA TRIGƠ TRONG MẠCH ĐỊNH THỜI.

3.3.1. Mạch điện của IC 555.

Bộ định thời 555 được sử dụng rất rộng rãi trong các bộ dao động đa hài, đa hài đợi, và các bộ so sánh v.v... Hình 3-28 là sơ đồ khối nguyên lý của IC định thời này, trong đó chức năng của các chân được chỉ ra trong bảng sau:

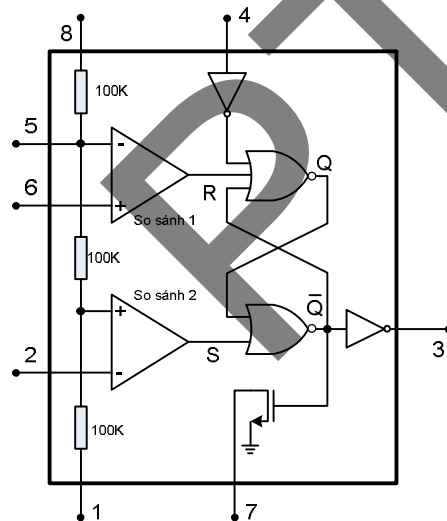
Chân	Chức năng	Chân	Chức năng
1	Đất - GND	5	Điện áp điều khiển
2	Chân kích thích	6	Chân ngưỡng
3	Đầu ra	7	Đầu phóng điện
4	Xoá - Reset	8	Nguồn – Vcc

Bảng 3-14. Bảng mô tả chức năng của các chân trong IC

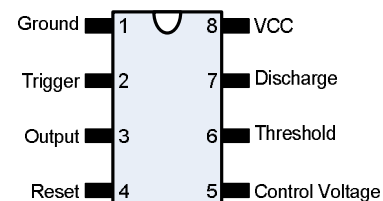
Sơ đồ nguyên lý của IC 555, gồm một mạch phân áp với 3 điện trở R (100kΩ) nối với chân 8. Ở đây chân 8 bao gồm cả nguồn nuôi của các bộ so sánh và các cổng logic trong mạch.

Điện áp +V_{CC} nối với chân 8 có giá trị từ +5V ÷ +25V tùy theo mức biên độ của xung ở đầu ra.

Mạch gồm hai bộ so sánh (1) và (2). Điện áp đầu vào đảo của bộ so sánh thứ nhất có mức điện áp bằng (2/3)V_{CC} điện áp ở đầu vào thuận của bộ so sánh thứ hai bằng (1/3)V_{CC}.



Hình 3-27. Sơ đồ khối nguyên lý của IC LMC 555



Hình 3-28. Sơ đồ chân của IC LMC555

Đầu ra của hai bộ so sánh được nối với đầu vào của Trigrơ RS, Trigrơ RS sử dụng trong mạch này dùng các cổng NOR, do đó mức tích cực là mức cao. Chân 4 được nối với đầu vào của một cổng NOR thông qua một cổng NOT có tác dụng điều khiển hoạt động của trigrơ, điện áp chân 4 ở mức cao (mức “1”) trigrơ hoạt động bình thường, còn điện áp chân 4 ở mức

thấp (mức “0”) cảm trigơ hoạt động. Đầu ra \overline{Q} của trigơ RS được đưa qua cổng NOT tới đầu ra ở chân 3, đồng thời \overline{Q} nối với transistor T để tạo đầu phóng điện.

Bảng chức năng của IC 555 được chỉ ra trên bảng 3-15.

TH	$\overline{\text{TRIG}}$	$\overline{\text{R}}$	OUT	DIS
X	X	L	L	Thông
$> \frac{2}{3} E_c$	$> \frac{1}{3} E_c$	H	L	Thông
$< \frac{2}{3} E_c$	$> \frac{1}{3} E_c$	H	Không đổi	Không đổi
X	$> \frac{1}{3} E_c$	H	H	Ngắt

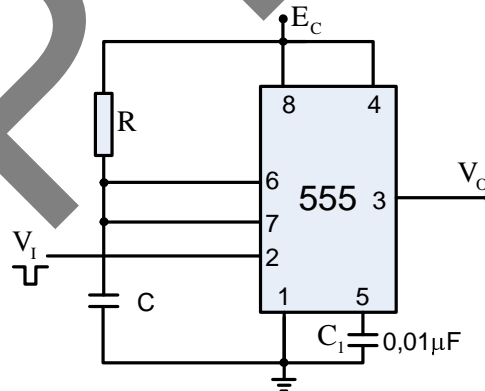
Bảng 3-15. Bảng chức năng của IC 555

3.3.2. Một vài ứng dụng của IC định thời 555

3.3.2.1. Mạch đa hài đợi

Mạch đa hài đợi có một trạng thái ổn định và một trạng thái tạm ổn định. Khi có tác dụng của xung ngoài, mạch có thể chuyển đổi từ trạng thái ổn định sang trạng thái tạm ổn định. Sau khi duy trì một thời gian, mạch sẽ tự động quay lại trạng thái ổn định. Thời gian tạm ổn định phụ thuộc vào các thông số của mạch mà không phụ thuộc vào xung kích. Mạch đa hài được ứng dụng trong các mạch định thời, tạo dạng xung, trễ v.v..

Hình 3-29 trình bày sơ đồ khối của mạch đa hài đợi dùng IC - 555.



Hình 3-29: Mạch đa hài đợi dùng IC - 555

Trong sơ đồ hình 3-29, tụ C_1 tác dụng cùng điện trở R (trong sơ đồ nguyên lý hình 3-27) tạo thành mạch lọc thông thấp.

Giả sử lúc đầu tụ C chưa tích điện nên $V_C = 0$. Khi đóng mạch nguồn điện, tụ C tích điện, mạch tích điện từ $+E_C$ qua R, qua C xuống đất. Tụ C tích điện, điện áp V_C tăng lên, cho đến khi $V_C \geq (2/3)V_{CC}$, điện áp ra của bộ so sánh (1) R = 0, dẫn đến $\overline{Q} = 1$. Vì $\overline{Q} = 1$ nên

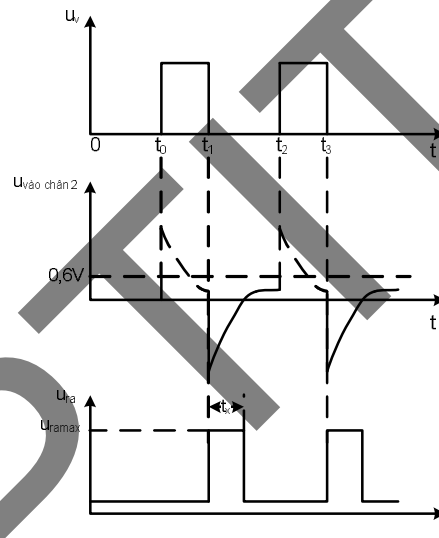
cho transistor T mở, tụ C phóng điện qua R, qua transistor xuống đất. Tụ C phóng điện đến khi $V_C = 0$. Đây là trạng thái ổn định bền (trạng thái đợi) của mạch.

Khi có xung vuông đưa tới đầu vào, nếu biên độ xung vào đủ lớn, điện áp ra của bộ so sánh (2) $S = 1$, do đó $Q = 1$ và $\overline{Q} = 0$, bắt đầu thời gian kéo dài xung ở đầu ra. Vì $\overline{Q} = 0$ nên T cấm, tụ C tích điện, mạch tích điện từ $+V_{CC}$, qua R, qua C xuống đất, tụ C nạp điện với hằng số thời gian:

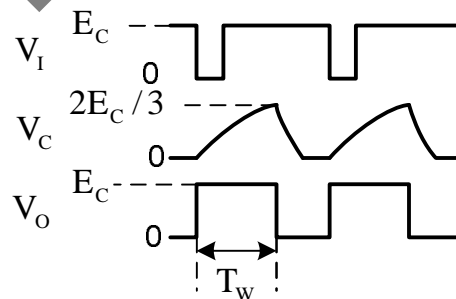
$$\tau_n = R.C \quad (3.28)$$

Tụ C tích điện, điện áp U_C tăng, cho đến khi $V_C \geq (2/3)V_{CC}$, đối với bộ so sánh (1) điện áp ra $R = 0$, dẫn đến $\overline{Q} = 1$, chấm dứt thời gian kéo dài xung ở đầu ra. Vì $\overline{Q} = 1$, T mở, tụ C phóng điện, qua R, qua T xuống đất, cho đến khi $V_C = 0$, trở lại trạng thái ban đầu.

Giải đồ điện áp - thời gian minh họa hoạt động của mạch đa hài đợi dùng IC - 555 được trình bày trên hình 3-30.



Hình 3-30. Giải đồ điện áp - thời gian của mạch đa hài đợi dùng IC - 555



Hình 3-31. Mạch đa hài đợi dùng IC 555 và dạng sóng

Hình 3-30 là sơ đồ nguyên lý và giải đồ thời gian của mạch đa hài đợi dùng IC 555, trong đó RC là mạch định thời. Độ kéo dài xung đầu ra được xác định bằng công thức

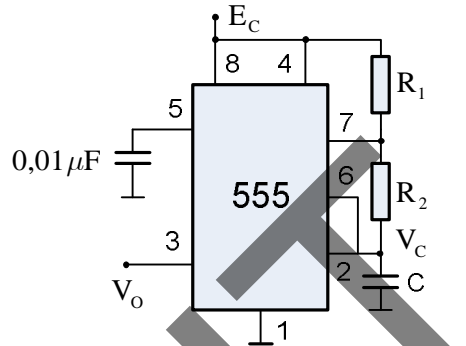
$$T_w = RC \ln 3 \approx 1,1RC \quad (3.29)$$

Mạch dao động đa hài đợi này yêu cầu độ rộng xung đầu vào nhỏ hơn độ rộng xung đầu ra, nếu nó lớn hơn thì yêu cầu dùng thêm mạch vi phân ở đầu vào.

3.3.2.2. Mạch đa hài

Một bộ dao động đa hài là một bộ dao động để tạo ra dạng xung. Nó có hai trạng thái chuẩn mà không yêu cầu sự kích hoạt từ bên ngoài. Bộ này thường được dùng làm xung điều khiển cho các mạch tuần tự.

Sơ đồ nguyên lý mạch đa hài dùng IC - 555 được trình bày trên hình 3-32.



Hình 3-32. Mạch đa hài dùng IC - 555

Lúc đầu đóng mạch nguồn nuôi một chiều, tụ C chưa kịp nạp điện, điện áp trên tụ $U_C = 0$, do đó điện áp ra của bộ so sánh (1) $R = 0$, điện áp ra của bộ so sánh (2) $S = 1$, dẫn đến $Q = 1$, $\overline{Q} = 0$ và T cấm. Tụ C tích điện, mạch tích điện từ $+V_{CC}$, qua R_1 , R_2 , qua C xuống đất, tụ C nạp điện với hằng số thời gian: $\tau_{nap} = (R_1 + R_2) \cdot C$. Tụ C nạp điện, điện áp trên tụ C tăng lên, cho đến khi $V_C \geq (2/3)V_{CC}$, điện áp ra của bộ so sánh (1) $R = 1$, $\overline{Q} = 1$, dẫn đến transistor T thông, tụ C phóng điện, mạch phóng điện từ cực dương của tụ C, qua R_2 , qua T xuống đất đến cực âm của tụ C. Tụ C phóng điện đến khi điện áp trên tụ $V_C \leq (1/3)V_{CC}$, điện áp ra của bộ so sánh (2) $R = 0$, dẫn đến $Q = 1$, $\overline{Q} = 0$. Do $\overline{Q} = 0$ nên transistor T bị cấm, tụ C lại được nạp điện, quá trình được lặp lại như cũ. Hiện tượng này tiếp diễn liên tục và tuần hoàn.

Thời gian nạp và phóng của tụ được tính theo công thức (3.30) và (3.31).

+ Thời gian nạp:

$$t_{nap} = \tau_{nap} \ln 2.$$

$$t_{nap} = 0,7(R_1 + R_2)C \quad (3.30)$$

+ Thời gian phóng:

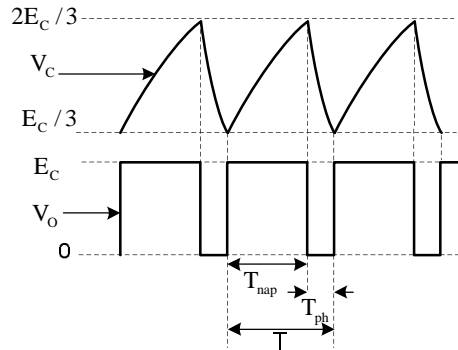
$$t_{ph} = \tau_{ph} \ln 2 \approx 0,7R_2C \quad (3.31)$$

+ Chu kỳ của xung ở đầu ra:

$$T = t_{\text{nap}} + t_{\text{ph}} = 0,7(R_1 + 2R_2)C \quad (3.32)$$

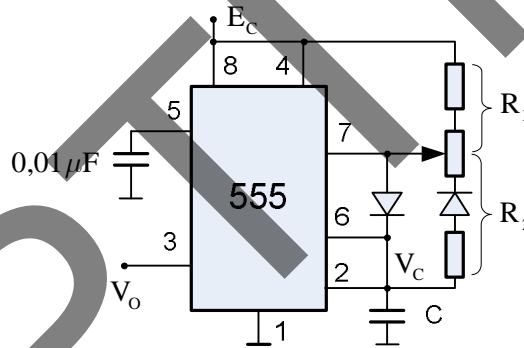
Dạng tín hiệu ở các cực của mạch được trình bày trên hình 3-33.

Do thời gian phóng, nạp không bằng nhau, nên xung vuông ra không đối xứng



Hình 3-33. Dạng điện áp tại các chân của mạch đa hài dùng IC - 555.

$$f = \frac{1}{T} = \frac{1,43}{(R_1 + 2R_2)C} \quad (3.33)$$



Hình 3-34. Mạch đa hài điều chỉnh được độ lấp đầy xung dùng IC 555

Xung đầu ra có độ lấp đầy phụ thuộc vào cả điện trở R_1 và R_2 và không thể tạo ra xung vuông với độ lấp đầy bằng 50% thông qua việc thay đổi giá trị R_1 và R_2 . Để có được xung vuông với độ lấp đầy bằng 50%, người ta sử dụng mạch có thêm 2 diode khi đó trở phóng và nạp điện cho tụ có thể thay đổi độc lập và tạo ra xung mong muốn.

Hình 3-34 là sơ đồ nguyên lý của mạch đa hài dùng IC 555 mà độ lấp đầy có thể thay đổi được.

3.4. PHƯƠNG PHÁP MÔ TẢ MẠCH TUẦN TỰ.

Thiết bị được thiết kế phải được mô tả bằng lời hay một số hình thức khác. Công việc đầu tiên của người thiết kế là phải phiên dịch các dữ kiện đó thành 1 hình thức mô tả hoạt động của thiết bị cần phải thiết kế một cách trung thực và duy nhất. Nói cách khác là phải hình thức hoá dữ liệu ban đầu.

Có hai cách hình thức hoá thường dùng đó là dùng bảng và đồ hình trạng thái.

3.4.1. Bảng

a. Bảng chuyển đổi trạng thái.

Bảng chuyển đổi trạng thái bao gồm các hàng và các cột, các hàng ghi các trạng thái trong, các cột ghi các giá trị của tín hiệu vào. Các ô ghi giá trị các trạng thái trong kế tiếp mà mạch sẽ chuyển đến ứng với các giá trị ở hàng và cột. Bảng chuyển đổi trạng thái được mô tả ở bảng 3-16.

		Tín hiệu vào				
<div>V S</div>		V ₁	V ₂	V _n	
Trạng thái trong	S ₁	Q ₁ ⁿ⁺¹	Q ₂ ⁿ⁺¹		Q _i ⁿ⁺¹ →	Trạng thái kế tiếp Q ⁿ⁺¹
	S ₂					
	:					
	S _n					

Bảng 3-16. Bảng chuyển đổi trạng thái

b. Bảng tín hiệu ra.

Các hàng của bảng ghi các trạng thái trong, các cột ghi các tín hiệu vào. Các ô ghi giá trị của tín hiệu ra tương ứng. Bảng tín hiệu ra được mô tả ở bảng 3-17.

		Tín hiệu vào				
<div>V S</div>		V ₁	V ₂	V _n	
Trạng thái trong	S ₁	Z ₁	Z ₂		Z _j	→ Tín hiệu ra
	S ₂					
	⋮					
	S _n					

Bảng 3-17. Bảng tín hiệu ra

Có thể gộp hai bảng chuyển đổi trạng thái và bảng tín hiệu ra thành một bảng chung gọi là bảng chuyển đổi trạng thái / ra. Lúc đó trên các ô ghi các giá trị của trạng thái kế tiếp và tín hiệu ra (Q^{n+1} / Z) tương ứng với trạng thái hiện tại và tín hiệu vào.

Bảng chuyển đổi trạng thái và tín hiệu ra được mô tả ở bảng 3-18.

		Tín hiệu vào				
<div> <div>V</div> <div>S</div> </div>		V ₁	V ₂	V _n	
Trạng thái trong	S ₁	Q ₁ ⁿ⁺¹ / Z	Q ₂ ⁿ⁺¹ / Z		Q _i ⁿ⁺¹ / Z →	Trạng thái kế tiếp Q ⁿ⁺¹ và tín hiệu ra Z
	S ₂					
	:					
	S _n					

Bảng 3-18. Bảng chuyển đổi trạng thái và tín hiệu ra

3.4.2. Đồ hình trạng thái.

Đồ hình trạng thái là hình vẽ phản ánh quy luật chuyển đổi trạng thái và tình trạng các giá trị ở đầu vào và đầu ra tương ứng của mạch tuần tự.

Đồ hình trạng thái là một đồ hình có hướng gồm hai tập:

M - Tập các đỉnh và K - Tập các cung có hướng.

a. Đối với mô hình Mealy thực hiện ánh xạ.

Tập các trạng thái trong là tập các đỉnh M; Tập các tín hiệu vào / ra là tập các cung K.

Trên cung có hướng đi từ trạng thái trong Q_i đến trạng thái trong Q_j ghi tín hiệu vào/ra tương ứng.

b. Đối với mô hình Moore.

Vì tín hiệu ra chỉ phụ thuộc vào trạng thái trong của mạch mà không phụ thuộc vào tín hiệu vào cho nên thực hiện ánh xạ:

Tập các trạng thái trong, tín hiệu ra là tập các đỉnh M.

Tập các tín hiệu vào là tập các cung K.

c. Chuyển đổi giữa hai mô hình Mealy và Moore.

Mealy và Moore là hai mô hình toán học của mạch tuần tự, nó là hai cách biểu diễn khác nhau của mạch tuần tự. Giữa hai cách biểu diễn này là ánh xạ 1-1 và luôn tồn tại thuật toán để chuyển từ mô hình này sang mô hình kia và ngược lại.

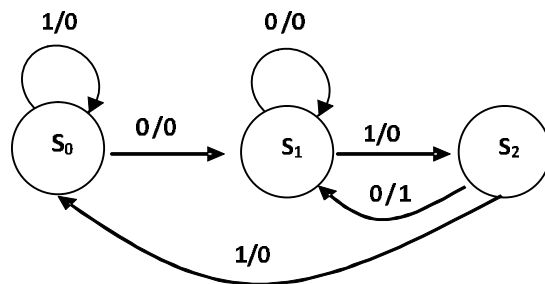
Chuyển đổi từ mô hình Mealy sang mô hình Moore.

Quá trình chuyển đổi được thực hiện theo các bước sau:

Ứng với mỗi cặp (Q^{n+1} / Z) của mô hình Mealy quy định thành một trạng thái tương ứng với Q của mô hình Moore.

Lập bảng chuyển đổi trạng thái cho mô hình Moore: ứng với mỗi trạng thái Q thì có một tín hiệu ra tương ứng.

Ví dụ: Mạch tuần tự được mô tả bởi mô hình Mealy có đồ hình trạng thái, bảng chuyển đổi trạng thái, tín hiệu ra như ở hình 3-35.



Hình 3-35a). Đồ hình trạng thái

Bước 1. Từ bảng chuyển đổi trạng thái / tín hiệu ra, gán các trạng thái trong Q_j của mô hình Moore như sau:

$$S_1 / 0 = Q_0; \quad S_0 / 0 = Q_1; \quad S_2 / 0 = Q_2; \quad S_1 / 1 = Q_3;$$

V \ S	0	1
S_0	$S_1 / 0$	$S_0 / 0$
S_1	$S_1 / 0$	$S_2 / 0$
S_2	$S_1 / 1$	$S_0 / 0$

Hình 3-35b). Bảng chuyển đổi trạng thái / tín hiệu ra

Bước 2: Lập bảng chuyển đổi trạng thái cho mô hình Moore: Gán tín hiệu ra cho mỗi trạng thái Q . Tín hiệu ra Z chính là Z của cặp (S, R) của mô hình Mealy. Các bước này được mô tả ở bảng 3-19.

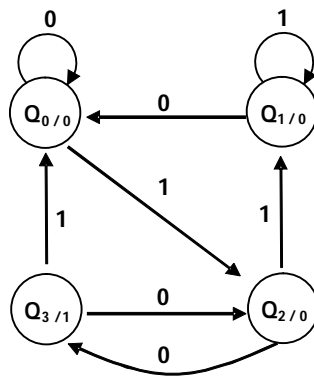
Mealy	Moore	
Trạng thái / tín hiệu ra	Trạng thái	Tín hiệu ra
$S_1 / 0$	Q_0	0
$S_0 / 0$	Q_1	0
$S_2 / 0$	Q_2	0
$S_1 / 1$	Q_3	1

Bảng 3-19. Gán trạng thái, tín hiệu ra cho mô hình Moore

Từ đó lập được bảng chuyển đổi trạng thái sau:

Mealy	Moore			Tín hiệu ra
	Trạng thái hiện tại	Trạng thái kế tiếp		
		V = 0	V = 1	
S ₁ / 0	Q ₀	Q ₀	Q ₂	0
S ₀ / 0	Q ₁	Q ₀	Q ₁	0
S ₂ / 0	Q ₂	Q ₃	Q ₁	0
S ₁ / 1	Q ₃	Q ₂	Q ₀	1

Bảng chuyển đổi trạng thái



b) Đồ hình trạng thái

Hình 3-36. Mô hình Moore tương ứng với mô hình Mealy ở bảng 3-19.

Chuyển đổi từ mô hình Moore sang mô hình Mealy.

Việc chuyển từ mô hình Moore sang mô hình Mealy đơn giản hơn bởi vì chỉ cần ghi thêm bên cạnh mỗi ô trong bảng chuyển đổi của mô hình Moore tín hiệu ra tương ứng. Sau đó tiến hành tối thiểu hoá các trạng thái của mô hình Mealy nhận được.

Nhận xét: Mô hình Mealy và mô hình Moore là hai mô hình dùng để biểu diễn mạch tuần tự, chúng luôn luôn tương đương với nhau.

Mô hình Mealy có hàm ra phụ thuộc vào tín hiệu vào và trạng thái trong của mạch. Mô hình Moore có hàm ra chỉ phụ thuộc vào trạng thái trong của mạch.

Nhìn chung mô hình Mealy có số trạng thái trong ít hơn so với mô hình Moore, nhưng mô hình Moore lại có hệ hàm ra đơn giản hơn mô hình Mealy.

Trên thực tế cả hai mô hình này đều được dùng để mô tả mạch tuần tự. Tùy từng trường hợp mà sử dụng loại mô hình nào để thiết kế mạch có sơ đồ đơn giản hơn, kinh tế hơn.

3.5. PHÂN TÍCH MẠCH TUẦN TỰ

3.5.1. Các bước phân tích mạch tuần tự

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 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. Đối với 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 mạch tuần tự 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.

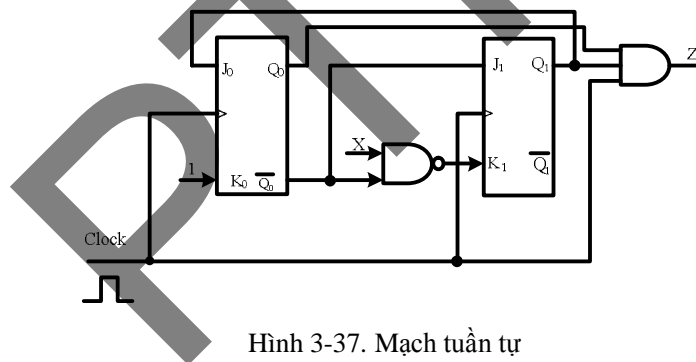
- Chức năng của mạch: Dựa vào đồ hình trạng thái xác định được chức năng của mạch

3.5.2. Phân tích mạch tuần tự đồng bộ

Phân tích mạch tuần tự đồng bộ có sơ đồ được biểu diễn như hình 3-37.

Bước 1: Sơ đồ trên có hai đầu vào là tín hiệu X và xung nhịp Clock. Có một tín hiệu Z ra, mạch sử dụng hai phần tử nhớ là hai trigơ JK (Q_0 và Q_1).

Bước 2: Xác định đầu vào, đầu ra và số trạng thái trong của mạch.



Hình 3-37. Mạch tuần tự

Mạch này có thể được biểu diễn bằng một “hộp đen” có hai đầu vào và một đầu ra. Do mạch được cấu tạo bằng hai trigơ nên số trạng thái có thể có của mạch là 4. Cụ thể là:

$Q_1Q_0 = 00, 01, 10$ và 11 .

Bước 3: Xác định phương trình hàm ra và hàm kích cho trigơ.

Từ sơ đồ trên tìm được:

+ Phương trình hàm ra: $Z = C Q_1 Q_0$

+ Phương trình hàm kích:

$J_0 = Q_1; K_0 = 1$

$$J_1 = \overline{Q_0} ; K_1 = \overline{X \overline{Q_0}} = \overline{X} + Q_0$$

Bước 4. Bảng chuyển đổi trạng thái.

Phương trình đặc trưng của trigơ JK là:

$$Q^{n+1} = J \overline{Q^n} + \overline{K} Q^n$$

Phương trình chuyển đổi trạng thái:

$$Q_0^{n+1} = J_0 \overline{Q_0^n} + \overline{K_0} Q_0^n = Q_1 \overline{Q_0^n}$$

$$Q_1^{n+1} = J_1 \overline{Q_1^n} + \overline{K_1} Q_1^n = \overline{Q_0^n} \overline{Q_1^n} + \overline{X + Q_0^n} Q_1^n = \overline{Q_0^n} \overline{Q_1^n} + X \overline{Q_0^n} Q_1^n$$

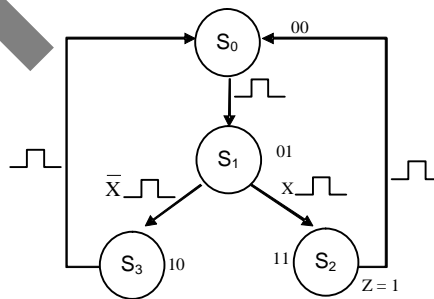
Từ các phương trình trên, lập được bảng chuyển đổi trạng thái (Bảng 3-20).

	Trạng thái hiện tại (Q^n)	Trạng thái kế tiếp (Q^{n+1})		Tín hiệu ra	
		X = 0	X = 1	X = 0	X = 1
	$Q_0 Q_1$	$Q_0 Q_1$	$Q_0 Q_1$	Z	Z
S_0	00	01	01	0	0
S_1	01	10	11	0	0
S_2	11	00	00	1	1
S_3	10	00	00	0	0

Bảng 3-20. Bảng chuyển đổi trạng thái

Bước 5: Đồ hình trạng thái.

Từ bảng chuyển đổi trạng thái trên, xây dựng được đồ hình trạng thái như hình 3-38 (mô hình Mealy). Đồ hình gồm 4 trạng thái trong S_0, S_1, S_2, S_3 . Các trigơ JK hoạt động tại sườn âm của xung nhịp. Nhìn vào đồ hình trạng thái, thấy ở trạng thái trong S_2 ($Q_0 Q_1 = 11$) khi có xung nhịp C thì mạch sẽ đưa ra tín hiệu $Z = 1$.



Hình 3-38. Đồ hình trạng thái

Bước 6: Chức năng của mạch: Trên đồ hình trạng thái, thấy có hai đường chuyển đổi trạng thái là $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$ và $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$. Theo đường $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$ thì tín hiệu ra $Z = 1$ sẽ được đưa ra cùng thời điểm có xung nhịp thứ 3. Theo đường $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$ thì không có tín hiệu ra ($Z = 0$). Do vậy, sẽ phân tích theo con đường thứ nhất $S_0 \rightarrow S_1$.

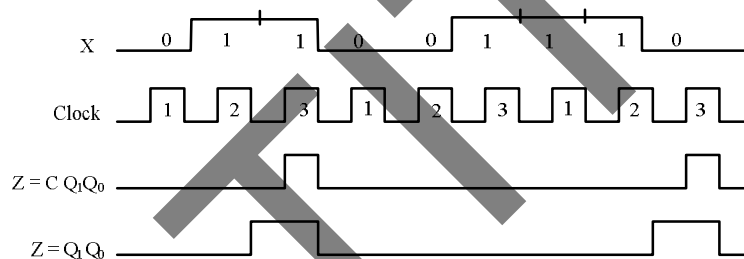
→ $S_2 \rightarrow S_0$: Sự chuyển đổi trạng thái đầu tiên từ $S_0 \rightarrow S_1$ chỉ nhờ tác động của xung nhịp mà không phụ thuộc vào trạng thái của X. Chuyển đổi trạng thái thứ hai từ $S_1 \rightarrow S_2$ nhờ tác động của xung nhịp và sự tác động của tín hiệu vào $X = 1$. Còn sự chuyển đổi trạng thái thứ ba từ $S_2 \rightarrow S_0$ chỉ nhờ tác động của xung nhịp mà không phụ thuộc vào tín hiệu vào.

Như vậy, mạch chỉ đưa ra tín hiệu ra $Z = 1$ khi đường chuyển đổi đi qua S_2 tức là mạch chỉ đưa ra tín hiệu ra $Z = 1$ khi dãy tín hiệu vào X có dạng 010, 011, 110 và 111. Có thể biểu diễn dãy tín hiệu vào để mạch có tín hiệu ra $Z = 1$ như sau:

0	1
↓	↓
$0 \leftarrow 1 \rightarrow 1$	$0 \leftarrow 1 \rightarrow 1$

Tóm lại, mạch cho ở sơ đồ trên có chức năng kiểm tra dãy tín hiệu vào X ở dạng chuỗi có độ dài bằng 3. Nếu chuỗi tín hiệu vào có dạng là 1 trong 4 dãy: 010, 011, 110 và 111 mạch sẽ cho tín hiệu ra $Z = 1$ tại thời điểm có xung nhịp thứ 3. Độ rộng của tín hiệu ra Z bằng độ rộng xung nhịp ($Z = C \cdot Q_1 \cdot Q_0$).

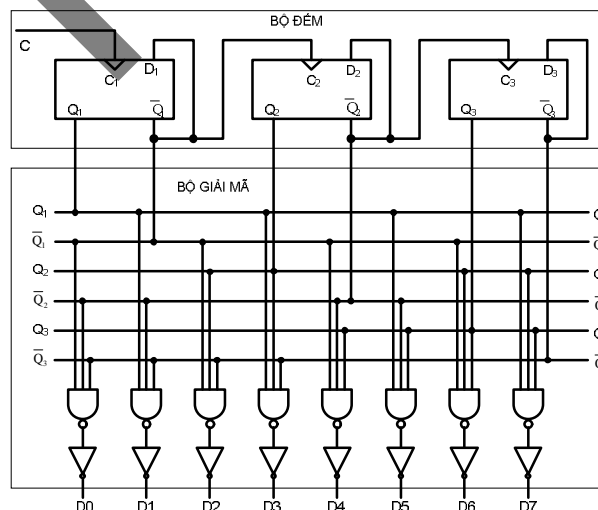
Hình 3-39 biểu diễn dạng xung ra của mạch hay còn gọi là giản đồ xung.



Hình 3-39. Dạng xung ra của mạch

3.7.3. Phân tích mạch tuần tự không đồng bộ.

Ví dụ: Hãy phân tích mạch tuần tự sau (hình 3-40) và cho biết chức năng của mạch.



Hình 3-40. Mạch tuần tự

Giải: Mạch hình 3-340 có hai phần chính: bộ đếm và bộ giải mã. Để phân tích mạch dễ dàng, phân tích từng phần.

Bước 1. Viết các phương trình:

+ Phương trình định thời: $C_1 = C$; $C_2 = \uparrow \overline{Q_1}$; $C_3 = \uparrow \overline{Q_3}$;

+ Phương trình kích: $D_1 = \overline{Q_1}$; $D_2 = \overline{Q_2}$; $D_3 = \overline{Q_3}$;

+ Phương trình chuyển đổi trạng thái:

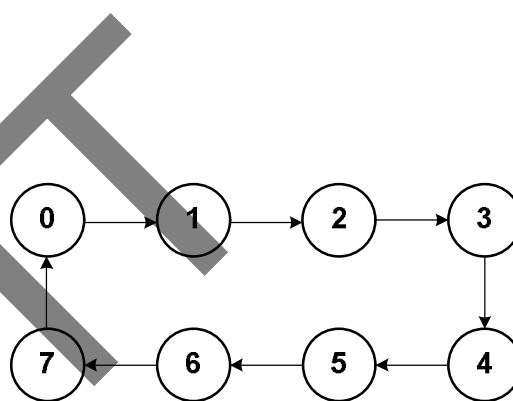
+ Phương trình đặc trưng của trigơ D: $Q^{n+1} = D$

+ $Q_1^{n+1} = \overline{Q_1}$; $Q_2^{n+1} = \overline{Q_2}$; $Q_3^{n+1} = \overline{Q_3}$;

Bước 2: Lập bảng trạng thái:

C	Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	0
6	1	1	0	1	1	1
7	1	1	1	0	0	0

Bảng 3-21. Bảng trạng thái



Hình 3-41. Đồ hình trạng thái

Bước 3: Từ bảng trạng thái 3-21, lập được đồ hình trạng thái như hình 3-41.

Bộ giải mã

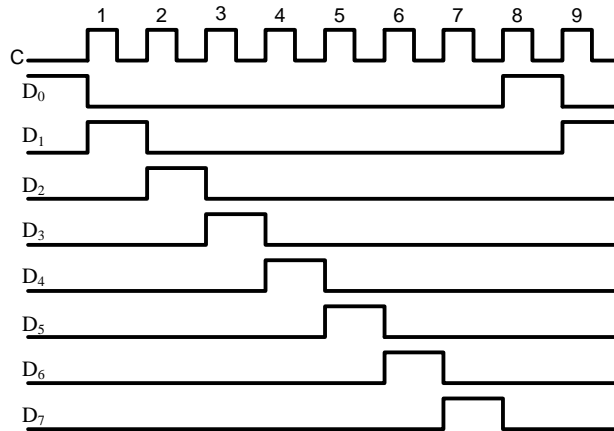
Tín hiệu ra từ bộ đếm được đưa vào bộ giải mã.

Có bảng trạng thái của bộ giải mã như bảng 3-22.

C	Q_3	Q_2	Q_1	Đầu ra
0	0	0	0	D0
1	0	0	1	D1
2	0	1	0	D2
3	0	1	1	D3
4	1	0	0	D4
5	1	0	1	D5
6	1	1	0	D6
7	1	1	1	D7

Bảng 3-22. Bảng trạng thái

Kết hợp cả hai phần phân tích, vẽ được dạng xung ra như hình 3-42.



Hình 3-42. Giản đồ xung của mạch

Nhận xét: Mạch tuần tự này chính là bộ tạo xung tuần tự (bộ phát xung nhịp). Mạch này dùng để tạo các xung tuần tự có thứ tự thời gian trước sau. Các xung này làm chuẩn thời gian cho sự đồng bộ của các bộ phận trong thiết bị. Xung Clock chuẩn được đưa vào bộ đếm, bộ giải mã sẽ biến đổi các trạng thái của bộ đếm thành xung tuần tự trên các đầu ra.

3.6. THIẾT KẾ MẠCH TUẦN TỰ.

3.6.1. Các bước thiết kế mạch tuần tự đồng bộ.

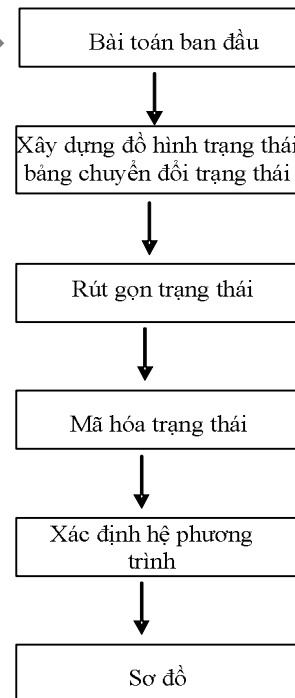
Các bước thiết kế được khái quát hóa trên hình 3-43.

Bước 1: Xác định bài toán, gán hàm và biến, tìm hiểu mối quan hệ giữa chúng.

Bước 2: Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái và hàm ra.

Bước 3: Rút gọn trạng thái (tối thiểu hoá trạng thái).

Việc tối thiểu hoá trạng thái chủ yếu dựa vào khái niệm trạng thái tương đương. Các trạng thái tương đương với nhau có thể được thay bằng một trạng thái chung đại diện cho chúng.



Hình 3-43. Lưu đồ thiết kế mạch tuần tự

Bước 4: Mã hoá trạng thái.

Số biến nhị phân dùng để mã hoá các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch. Nếu số lượng trạng thái trong là N , số biến nhị phân cần dùng là n thì n phải thoả mãn điều kiện: $n \geq \log_2 N$.

Có rất nhiều cách mã hoá khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Vấn đề là phải mã hoá sao cho sơ đồ mạch thực hiện là đơn giản nhất.

Bước 5: Xác định hệ phương trình của mạch. Có hai cách xác định:

+ Lập bảng chuyển đổi trạng thái và tín hiệu ra, từ đó xác định các phương trình kích cho các trigơ.

+ Dựa trực tiếp vào đồ hình trạng thái, viết hệ phương trình T_{on} , T_{off} của các trigơ và phương trình hàm ra.

Bước 6: Vẽ sơ đồ thực hiện.

3.6.2. Các bước thiết kế mạch tuần tự không đồng bộ

Bước 1: Xác định bài toán, gán hàm và biến, tìm hiểu mối quan hệ giữa chúng.

Bước 2: Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái và hàm ra.

Bước 3: Rút gọn trạng thái (tối thiểu hoá trạng thái).

Việc tối thiểu hoá trạng thái chủ yếu dựa vào khái niệm trạng thái tương đương. Các trạng thái tương đương với nhau có thể được thay bằng một trạng thái chung đại diện cho chúng.

Có nhiều phương pháp tối thiểu hoá trạng thái. Trong phần này, giới thiệu phương pháp tối thiểu hoá Caldwell. Cơ sở lý thuyết của việc tối thiểu hoá là dựa vào khái niệm các trạng thái tương đương.

Định nghĩa các trạng thái tương đương:

Trạng thái S_i được gọi là trạng thái tương đương với trạng thái S_j ($S_i \approx S_j$) khi và chỉ khi: nếu lấy S_i và S_j là hai trạng thái ban đầu thì với mọi dãy tín hiệu vào có thể chúng luôn cho dãy tín hiệu ra giống nhau.

Nếu có nhiều trạng thái tương đương với nhau từng đôi một thì chúng tương đương với nhau (tính chất bắc cầu). Để kiểm tra một nhóm các trạng thái xem chúng có tương đương với nhau không, có thể sử dụng bảng trạng thái và tín hiệu ra như sau:

Nhóm các trạng thái tương đương phải có những hàng trong bảng tín hiệu ra giống nhau.

Nhóm các trạng thái tương đương phải có những hàng trong bảng trạng thái ở cùng một cột (ứng với cùng một tổ hợp tín hiệu vào) là tương đương. Nghĩa là ứng với cùng một tổ hợp tín hiệu vào các trạng thái kế tiếp của chúng là tương đương.

Quy tắc Caldwell:

Những hàng (tương ứng với trạng thái trong) của bảng chuyển đổi trạng thái và tín hiệu ra sẽ được kết hợp với nhau và được biểu diễn bằng một hàng chung - đặc trưng (trạng thái đặc trưng) cho chúng nếu như chúng thoả mãn hai điều kiện sau:

- + Các hàng tương ứng trong ma trận ra giống nhau.
- + Trong ma trận ra, các hàng tương ứng phải thoả mãn 1 trong 3 điều sau:
 - Các hàng trong ma trận trạng thái giống nhau.
 - Các trạng thái ở trong cùng một cột nằm trong nhóm trạng thái được xét.
 - Các trạng thái ở trong cùng một cột là các trạng thái tương đương.

Sau khi đã thay thế các trạng thái tương đương bằng một trạng thái chung đặc trưng cho chúng, lặp lại các công việc tìm các trạng thái tương đương khác cho đến khi không thể tìm được các trạng thái tương đương nào nữa thì dừng lại. Số trạng thái trong bảng chuyển đổi trạng thái là tối thiểu.

Nhược điểm của phương pháp này là khi số trạng thái quá lớn thì công việc tối thiểu hoá mất nhiều thời gian.

Bước 4: Mã hoá trạng thái.

Số biến nhị phân dùng để mã hoá các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch. Nếu số lượng trạng thái trong là N , số biến nhị phân cần dùng là n thì phải thoả mãn điều kiện: $n \geq \log_2 N$.

Có rất nhiều cách mã hoá khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Vấn đề là phải mã hoá sao cho sơ đồ mạch thực hiện là đơn giản nhất.

Do mạch không đồng bộ hoạt động không có sự tác động của xung nhịp cho nên trong mạch thường có các hiện tượng chạy đua làm cho hoạt động của mạch bị sai, vì vậy khi mã hoá trạng thái phải tránh hiện tượng này.

Bước 5: Xác định hệ phương trình của mạch. Có hai cách xác định:

+ Lập bảng chuyển đổi trạng thái và tín hiệu ra, từ đó xác định các phương trình kích cho các trigơ.

+ Dựa trực tiếp vào đồ hình trạng thái, viết hệ phương trình T_{on} , T_{off} của các trigơ và phương trình hàm ra.

Cả hai cách này đều có dạng phương trình:

- + Phương trình của mạch chỉ dùng NAND.
- + Phương trình của mạch dùng trigơ RS không đồng bộ và các mạch NAND.
- + Phương trình của mạch dùng các loại trigơ khác.

Bước 6: Vẽ sơ đồ thực hiện.

Sau đây là nội dung của từng phương pháp.

Dựa vào các bước thiết kế mạch nêu trên, người ta đưa ra hai cách thiết kế mạch tuần tự: dùng bảng chuyển đổi trạng thái và dùng đồ hình trạng thái:

Cách 1: Dựa vào bảng chuyển đổi trạng thái.

a) Chỉ dùng các mạch NAND.

Ký hiệu : A, B, \dots, N là các biến nhị phân dùng để mã hoá các trạng thái trong của mạch.

$X_1, X_2 \dots X_m$ là các tín hiệu vào đã được mã hoá nhị phân.

$Z_1, Z_2 \dots Z_m$ là các tín hiệu ra đã được mã hoá nhị phân.

Dựa vào bảng chuyển đổi trạng thái xác định hệ phương trình:

$$A^k = f_A (A, B, \dots, N, X_1, X_2 \dots X_m)$$

$$B^k = f_B (A, B, \dots, N, X_1, X_2 \dots X_m)$$

.....

$$N^k = f_N (A, B, \dots, N, X_1, X_2 \dots X_m)$$

$$Z_1 = g_1 (A, B, \dots, N, X_1, X_2 \dots X_m)$$

$$Z_2 = g_2 (A, B, \dots, N, X_1, X_2 \dots X_m)$$

.....

$$Z_n = g_n (A, B, \dots, N, X_1, X_2 \dots X_m)$$

Tối thiểu hoá hệ hàm và viết phương trình ở dạng chỉ dùng NAND.

b) Mạch dùng trigơ RS và các mạch NAND.

Trong bảng trạng thái căn cứ vào sự thay đổi trạng thái của từng trigơ:

$A \Rightarrow A^k, B \Rightarrow B^k, \dots, N \Rightarrow N^k$, xác định được giá trị tương ứng của đầu vào kích R, S cho từng trigơ, từ đó viết được hệ phương trình:

$$\begin{aligned} R_A &= \Phi_1 (A, \dots, N, X_1, X_2 \dots X_m) \\ S_A &= \Phi_2 (A, \dots, N, X_1, X_2 \dots X_m) \end{aligned} \quad (3.34)$$

Tối thiểu hoá các hàm và viết phương trình ở dạng chỉ dùng NAND.

Tương tự với B, C, ... N cũng như vậy.

Xác định tín hiệu ra :

$$Z = \Phi (A, \dots, N, X_1, X_2 \dots X_m) \quad (3.35)$$

Tối thiểu hoá và viết phương trình ở dạng chỉ dùng NAND.

Tương tự đối với các loại trigơ khác.

Cách 2: Dựa trực tiếp vào đồ hình trạng thái.

Phương trình đầu vào kích (R, S) của trigơ A là:

$S_A = \text{tập hợp bật của } A + [(1)]$

$R_A = \text{tập hợp tắt của } A + [(0)]$

Hay

$$\begin{aligned} S_A &= T_{\text{on}A} + [(1)] \\ R_A &= T_{\text{off}A} + [(0)] \end{aligned} \quad (3.36)$$

Làm tương tự với các loại trigơ khác.

a) Chỉ dùng mạch NAND.

Phương trình đặc trưng của trigơ RS

$$Q^k = S + \bar{R}Q \Rightarrow A^k = S_A + \bar{R}_A A \quad (3.37)$$

Sau đó, phải tối thiểu hoá phương trình và viết dưới dạng chỉ dùng NAND. Đối với các trigơ khác cũng làm như vậy.

b) Dùng các trigơ RS không đồng bộ và các mạch NAND.

$$R_A = \Phi_{1A}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$S_A = \Phi_{2A}(A, \dots, N, X_1, X_2, \dots, X_m)$$

.....

$$R_N = \Phi_{1N}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$S_N = \Phi_{2N}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_1 = \Psi_1(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_2 = \Psi_2(A, \dots, N, X_1, X_2, \dots, X_m)$$

.....

$$Z_n = \Psi_n(A, \dots, N, X_1, X_2, \dots, X_m)$$

Tối thiểu hoá hệ phương trình.

c). Tương tự đối với các loại trigơ khác.

3.6.3. Thiết kế mạch tuần tự từ đồ hình trạng thái.

Giả thiết: Cho đồ hình trạng thái của mạch có tập tín hiệu vào V, tập tín hiệu ra R, tập trạng thái trong S (chưa mã hoá nhị phân).

Xác định: Hệ phương trình nhị phân của mạch (đã tối thiểu hoá). Trên cơ sở đó vẽ mạch điện.

3.6.3.1. Các bước thiết kế

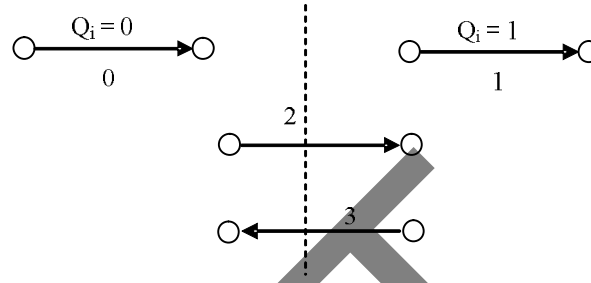
Mã hoá tín hiệu vào V, tín hiệu ra R, trạng thái trong S để chuyển thành mạch dạng nhị phân có các tập tín hiệu vào X, tín hiệu ra Y, trạng thái trong Q.

Xác định hệ phương trình tín hiệu ra: $Y_i = f_i(X, Q)$. Phương trình này được xác định trên các cung với mô hình kiểu Mealy, trên các đỉnh với mô hình kiểu Moore. Tối thiểu các hàm này.

Xác định hệ phương trình hàm kích cho các trigơ và tối thiểu hoá nó.

Sau đây giới thiệu thuật toán xác định phương trình đầu vào kích cho các trigơ từ đồ hình trạng thái.

Đối với trigơ Q_i bất kỳ sự thay đổi trạng thái từ Q_i^n đến Q_i^{n+1} chỉ có thể có 4 khả năng như hình 3-44.



Hình 3-44. Các cung biểu diễn sự thay đổi trạng thái từ Q_i^n đến Q_i^{n+1} của trigơ Q_i

Trong đó các cung biểu diễn sự thay đổi từ Q_i^n đến Q_i^{n+1} được ký hiệu như sau:

$0 \rightarrow 0$ là (0), $1 \rightarrow 1$ (là 1), $0 \rightarrow 1$ là (2), $1 \rightarrow 0$ là (3).

Từ quy ước có thuật toán sau:

a. Thuật toán xác định phương trình đầu vào kích cho trigơ Q_i loại D.

Phương trình đặc trưng của trigơ D : $Q_i^{n+1} = D_i$. Từ đó, tìm được:

$$\begin{aligned} D_i &= Q_i^{n+1} = \text{tuyển tất cả các cung đi tới đỉnh có } Q_i = 1. \\ &= \sum \text{các cung loại (2), kể cả khuyên tại đỉnh đó tức là cung loại 1} \\ &= \sum (1) \text{ và } (2) \end{aligned}$$

Tối thiểu hoá hàm D_i vừa tìm được rút ra phương trình đầu vào kích cho trigơ loại D.

b. Thuật toán xác định phương trình đầu vào kích cho trigơ T.

Phương trình đặc trưng của trigơ T: $Q_i^{n+1} = T_i \oplus Q_i \Rightarrow T_i = Q_i \oplus Q_i^{n+1} = Q_i'$

Trong đó Q_i' bằng 1 khi Q_i thay đổi trạng thái từ $0 \Rightarrow 1$ hoặc từ $1 \Rightarrow 0$.

Để tìm T_i thực hiện các bước sau:

- Điền sự thay đổi giá trị của Q_i vào các cung.
- $T_i = Q_i' = \sum \text{các cung có } Q_i \text{ thay đổi (cung loại 2, loại 3)} = \sum (2) \text{ và } (3).$

Tối thiểu hoá hàm T_i vừa tìm được rút ra phương trình kích cho trigơ T.

c. Thuật toán xác định phương trình đầu vào kích cho trigơ JK.

Phương trình đặc trưng của trigơ JK: $Q_i^{n+1} = J \overline{Q_i} + \overline{K} Q_i$

Xác định:

$T_{on} = \sum$ các cung mà Q_i được bật (Q_i thay đổi từ $0 \Rightarrow 1$ - cung loại 2) = $\sum (2)$. Đưa phương trình của T_{on} về dạng:

$$T_{on} = (T^*) \overline{Q_i} \Rightarrow J = T^* \quad (3.38)$$

$T_{off} = \sum$ các cung mà Q_i tắt (Q_i thay đổi từ $1 \Rightarrow 0$ - cung loại 3) = $\sum (3)$. Đưa phương trình của T_{off} về dạng:

$$T_{off} = (\overline{T^{**}}) Q_i \Rightarrow K = T^{**} \quad (3.39)$$

d. Thuật toán xác định phương trình đầu vào kích cho trigơ RS.

Phương trình đầu vào S của trigơ RS được xác định như sau:

$$S = T_{on} + [\text{Các cung loại (1)}]$$

$$R = T_{off} + [\text{Các cung loại (0)}]$$

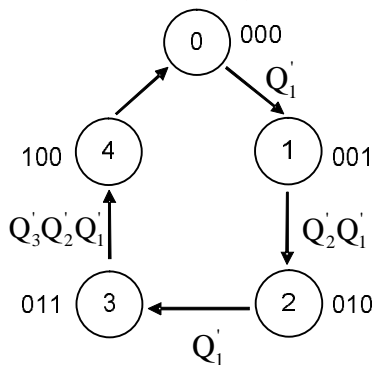
Các cung loại (1), các cung loại (0) để trong dấu [] ở biểu thức của S, R được lấy giá trị không xác định. Những giá trị này và những trạng thái không được sử dụng sẽ được dùng để tối thiểu hoá sao cho biểu thức nhận được là tối giản nhất.

3.6.4. Ví dụ

Để minh hoạ, xét ví dụ sau: Thiết kế bộ đếm đồng bộ có $M_d = 5$ với đồ hình trạng thái 3-45 và mã hoá trạng thái 3-23, dùng:

Trigơ D và các mạch AND; Trigơ T và các mạch AND; Trigơ JK và các mạch AND.

Giải:



Hình 3-45. Đồ hình trạng thái

$Q_2 Q_1$		Q_3			
		00	01	11	10
Q_3	0	0	1	3	2
	1	4	x	x	x

Bảng 3-23. Bảng mã hóa trạng thái

Bộ đếm M = 5 nên có 5 trạng thái 0, 1, 2, 3, 4. Để đơn giản, trên đồ hình không ghi các tín hiệu vào đếm và tín hiệu ra. Tín hiệu ra của bộ đếm chỉ xuất hiện khi bộ đếm đang ở trạng thái 4 và có tín hiệu vào đếm, lúc đó bộ đếm quay trở về trạng thái ban đầu 0 và cho ra tín hiệu ra.

Q_2Q_1 Q_3		00	01	11	10
0	1	2	4	3	
1	0	x	x	x	

Bảng 3-24. Bảng chuyển đổi trạng thái

Q_2Q_1 Q_3		00	01	11	10
		0	1	2	3
0	0	0	0	0	0
1	1	x	x	x	x

$$Z = Q_3$$

Bảng 3-25. Bảng Các nôm tìm hàm ra

Mạch có 5 trạng thái và do vậy được mã hoá ít nhất bằng 3 biến nhị phân tương ứng với 3 trigơ: $Q_3 Q_2 Q_1 (2^3 2^2 2^1)$ như trên bảng mã hoá trạng thái 3-23. Điền mã tương ứng vào các trạng thái trên đồ hình 3-45.

Từ bảng 3-25, phương trình tín hiệu ra Z:

$$Z = Q_3$$

Xác định các phương trình kích cho các trigơ :

a) Trigơ D.

Nhìn vào đồ hình trạng thái:

$Q_3 = 1$ tại đỉnh (4); $Q_2 = 1$ tại đỉnh (2), (3); $Q_1 = 1$ tại đỉnh (1), (3).

$D_3 = \sum$ Các cung đi đến đỉnh (4) = (3) = $\overline{Q_3} Q_2 Q_1$.

$D_2 = \sum$ Các cung đi đến đỉnh (2), (3) = (1) + (2) = $\overline{Q_3} \overline{Q_2} Q_1 + \overline{Q_3} Q_2 \overline{Q_1}$.

$D_1 = \sum$ Các cung đi đến đỉnh (1), (3) = (0) + (2) = $\overline{Q_3} \overline{Q_2} \overline{Q_1} + \overline{Q_3} Q_2 \overline{Q_1}$.

Từ đó, lập bảng Các nôm để tối thiểu hóa hàm D_i

Q_2Q_1 Q_3		00	01	11	10
		0	1	2	3
0	0	0	0	1	0
1	0	x	x	x	x

$$D_3 = Q_2 Q_1$$

Q_2Q_1 Q_3		00	01	11	10
		0	1	2	3
0	0	0	1	0	1
1	0	x	x	x	x

$$D_2 = \overline{Q_2} \cdot Q_1 + Q_2 \cdot \overline{Q_1} = Q_1 \oplus Q_2$$

Q_2Q_1		00	01	11	10
Q_3	0	1	0	0	1
	1	0	x	x	x

Bảng 3-26. Bảng Các nô tìm hàm kích

$$D_1 = \overline{Q_1} \cdot \overline{Q_3}$$

Dùng bảng Các nô 3-26, thu được kết quả:

$$D_3 = Q_2Q_1$$

$$D_2 = \overline{Q_2} Q_1 + Q_2 \overline{Q_1} = Q_2 \oplus Q_1$$

$$D_1 = \overline{Q_1} \overline{Q_3}$$

b) Xác định phương trình kích cho Triger T.

Điền sự thay đổi giá trị của Q_i (Q_i) vào các cung. Khi mạch đếm từ trạng thái (0) \Rightarrow (1) (nghĩa là từ 000 \Rightarrow 001) thì Q_3 thay đổi từ 0 \Rightarrow 1 nên ghi Q_3 lên cung đó. Khi mạch chuyển từ trạng thái (1) \Rightarrow (2) (tương ứng từ 001 \Rightarrow 010): Q_1 không thay đổi trạng thái (= 0), Q_2 thay đổi từ 0 \Rightarrow 1 và Q_3 thay đổi từ 1 \Rightarrow 0, nên ghi Q_2Q_3 lên cung từ (1) \Rightarrow (2). Tương tự như vậy có:

$$T_3 = Q_3 = \sum \text{các cung có } Q_3 \text{ thay đổi} = (3) + (4) = \overline{Q_3} \overline{Q_2} Q_1 + Q_3 \overline{Q_2} \overline{Q_1}$$

$$T_2 = Q_2 = \sum \text{các cung có } Q_2 \text{ thay đổi} = (1) + (3) = \overline{Q_3} \overline{Q_2} Q_1 + \overline{Q_3} Q_2 Q_1$$

$$T_1 = Q_1 = \sum \text{các cung có } Q_1 \text{ thay đổi} = (0) + (1) + (2) + (3) = \overline{Q_3} \overline{Q_2} \overline{Q_1} + \overline{Q_3} \overline{Q_2} Q_1 + \overline{Q_3} Q_2 \overline{Q_1} + \overline{Q_3} Q_2 Q_1$$

Q_2Q_1		00	01	11	10
Q_3	0	0	0	1	0
	1	1	x	x	x

$$T_3 = Q_3 + Q_2Q_1$$

Q_2Q_1		00	01	11	10
Q_3	0	0	1	1	0
	1	0	x	x	x

$$T_2 = Q_1$$

Q_2Q_1					
Q_3		00	01	11	10
0		1	1	1	1
1		0	x	x	x

Bảng 3-27. Bảng Các nô tìm hàm kích

$$T_1 = \overline{Q_3}$$

Lập bảng Các nô 3-27 cho các hàm trên thu được kết quả:

$$T_3 = Q_3 + Q_2Q_1 ; \quad T_2 = Q_1 ; \quad T_1 = \overline{Q_3}$$

c) Xác định phương trình kích cho Trigo JK.

Chú ý khi viết các biểu thức T_{on} , T_{off} của trigo thứ i cần phải đơn giản các biểu thức đó và đưa về dạng:

$$T_{on} = (T^*) \overline{Q_i} \Rightarrow \text{rút ra } J_i = T^*.$$

$$T_{off} = (\overline{T^{**}}) Q_i \Rightarrow \text{rút ra } K_i = T^{**}.$$

Viết các biểu thức T_{on} , T_{off} cho các trigo và từ đó xác định phương trình kích cho các trigo như sau:

$$T_{on1} = \sum \text{Các cung mà } Q_3 \text{ được bật (Chuyển từ } 0 \Rightarrow 1) = (3) = \overline{Q_3} Q_2 Q_1$$

$$T_{off1} = \sum \text{Các cung mà } Q_3 \text{ tắt (Chuyển từ } 1 \Rightarrow 0) = (4) = Q_3 \overline{Q_2} \overline{Q_1}$$

$$T_{on2} = \sum \text{Các cung mà } Q_2 \text{ được bật (Chuyển từ } 0 \Rightarrow 1) = (1) = \overline{Q_3} \overline{Q_2} Q_1$$

$$T_{off2} = \sum \text{Các cung mà } Q_2 \text{ tắt (Chuyển từ } 1 \Rightarrow 0) = (3) = \overline{Q_3} Q_2 Q_1$$

$$T_{on3} = \sum \text{Các cung mà } Q_1 \text{ được bật (Chuyển từ } 0 \Rightarrow 1) = (0) + (2) = \overline{Q_3} \overline{Q_1}$$

$$T_{off3} = \sum \text{Các cung mà } Q_1 \text{ tắt (Chuyển từ } 1 \Rightarrow 0) = (1) + (3) = \overline{Q_3} Q_1$$

Biểu diễn các hàm này trên bảng Các nô, sử dụng các trạng thái tùy chọn để tối thiểu hoá. Các trạng thái tùy chọn bao gồm 3 số không nằm trong phạm vi đếm 5, 6, 7. Ngoài ra còn một số trạng thái khác tùy vào từng bảng. Ví dụ, đối với bảng tính J_1 giá trị tùy chọn ngoài 3 số trên còn thêm ô có giá trị $Q_1 = 1$, bảng tính K_1 có thêm các ô có giá trị $Q_1 = 0$, tương tự như vậy với các bảng còn lại. Điều này có nghĩa là ngoài các giá trị tùy chọn đã cho còn có thêm các giá trị tùy chọn khác. Đối với phương trình $T_{on i}$ thêm điều kiện tùy chọn $Q_i = 1$, còn với phương trình $T_{off i}$ thêm điều kiện tùy chọn $Q_i = 0$. Từ đó, lập bảng Các nô để tìm hàm kích J, K.

Thu được kết quả từ bảng 3-28 như sau:

$$J_3 = Q_2 Q_1; \quad K_3 = 1; \quad J_2 = Q_1; \quad K_2 = Q_1; \quad J_1 = \overline{Q_3}; \quad K_1 = 1$$

The six Karnaugh maps are as follows:

- Top Left (J₃):** Shows 1s at (0, 11) and (1, 11). The equation is $J_3 = Q_2 Q_1$.
- Top Right (K₃):** Shows 1s at (0, 00), (0, 01), (1, 00), and (1, 01). The equation is $K_3 = 1$.
- Middle Left (J₂):** Shows 1s at (0, 01) and (1, 01). The equation is $J_2 = Q_1$.
- Middle Right (K₂):** Shows 1s at (0, 01) and (1, 01). The equation is $K_2 = Q_1$.
- Bottom Left (J₁):** Shows 1s at (0, 00), (0, 10), (1, 01), and (1, 10). The equation is $J_1 = \overline{Q_3}$.
- Bottom Right (K₁):** Shows 1s at (0, 01), (0, 11), (1, 01), and (1, 11). The equation is $K_1 = 1$.

Bảng 3-28. Bảng Các nô tìm hàm kích

d) Xác định phương trình kích cho Trơ RS.

$$S_3 = T_{on3} + [\text{Các cung loại (1)}] = (3) + [\phi]$$

$$R_3 = T_{off3} + [\text{Các cung loại (0)}] = (4) + [(0), (1), (2)]$$

$$S_2 = T_{on2} + [\text{Các cung loại (1)}] = (1) + [(2)]$$

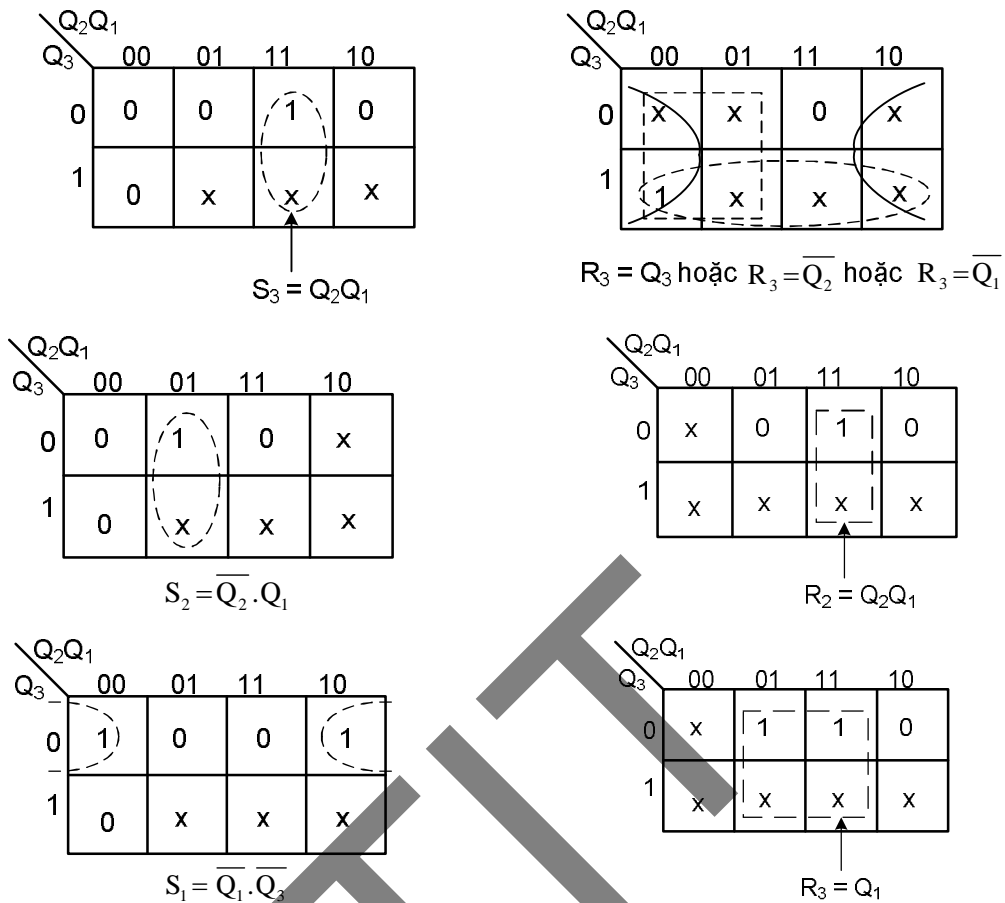
$$R_2 = T_{off2} + [\text{Các cung loại (0)}] = (3) + [(0), (4)]$$

$$S_1 = T_{on1} + [\text{Các cung loại (1)}] = (0) + (2) + [\phi]$$

$$R_1 = T_{off1} + [\text{Các cung loại (0)}] = (1) + (3) + [(4)]$$

Lưu ý: Các giá trị trong dấu [.] là các giá trị tùy chọn.

Biểu diễn các hàm này trên bảng Các nô và tối thiểu hoá chúng.



Bảng 3-29. Bảng Các nô tìm hàm kích

Sau khi rút gọn từ bảng 3-29 thu được kết quả sau:

$$S_3 = Q_2 Q_1 \quad ; \quad R_3 = Q_3 \text{ hoặc } R_3 = \overline{Q_2} \text{ hoặc } R_3 = \overline{Q_1}$$

$$S_2 = \overline{Q_2} \cdot Q_1 \quad ; \quad R_2 = Q_2 Q_1 ; S_1 = \overline{Q_1} \cdot \overline{Q_3} \quad ; \quad R_1 = Q_1$$

Từ các phương trình kích đã tìm, có thể vẽ được sơ đồ logic.

3.6.3. Thiết kế mạch tuần tự từ bảng.

Giả thiết: Cho bảng chuyển đổi trạng thái, bảng ra của mạch (chưa mã hoá nhị phân).

Xác định: Hệ phương trình nhị phân của mạch vào gồm hệ hàm ra, hệ hàm kích cho các trigơ. Trên cơ sở đó vẽ sơ đồ mạch.

Mã hoá tín hiệu vào V, tín hiệu ra R, trạng thái trong của mạch S để chuyển mạch ban đầu thành mạch nhị phân có tập tín hiệu vào X, tập tín hiệu ra Y, tập trạng thái trong Q.

Lập bảng chuyển đổi trạng thái, bảng ra của mạch nhị phân ứng với sự mã hoá trên.

Dựa vào bảng các đầu vào kích của các trigơ xác định các đầu vào kích cho các trigơ ứng với sự chuyển đổi trong bảng trạng thái.

Viết phương trình đầu vào kích cho từng Q_i của trigơ và các hàm ra rồi tối thiểu các hàm này. Trên cơ sở đó xây dựng mạch điện.

Ví dụ: Thiết kế mạch tuần tự đồng bộ dùng trigơ JK. Hệ có đầu vào là X, đầu ra là Z. Các đầu vào và ra đều là số nhị phân. Đầu ra $Z = 1$ nếu ở đầu vào X là dãy nhị phân xuất hiện theo quy luật $X = 0101$. Các trường hợp khác $Z = 0$.

S	S^{n+1}		Z	
	X=0	X=1		
$Q_3Q_2Q_1$	$Q_3Q_2Q_1$	$Q_3Q_2Q_1$	X=0	X=1
000	000	001	0	0
001	010	011	0	0
010	100	101	0	1
011	110	111	0	0
100	000	001	0	0
101	010	011	0	0
110	100	101	0	0
111	110	111	0	0
Bảng 3-30. Bảng chuyển đổi trạng thái				

Để xác định giá trị ở mỗi thời điểm của đầu ra không những phải biết giá trị đầu vào tại thời điểm đó mà còn phải biết các giá trị trước đó. Theo đề bài phát hiện dãy xung có 4 bit nên phải biết 3 bit trước đó. Điều này có nghĩa là cần 3 trigơ ($Q_3 Q_2 Q_1$). Trạng thái tiếp theo nhận được bằng cách dịch trái 1 bit.

Mã hóa trạng thái theo bảng 3-31:

S	S^n	S^{n+1}		Z	
		X=0	X=1	X=0	X=1
$Q_3Q_2Q_1$					
000	S_0	S_0	S_1	0	0
001	S_1	S_2	S_3	0	0
010	S_2	S_4	S_5	0	1
011	S_3	S_6	S_7	0	0
100	S_4	S_0	S_1	0	0
101	S_5	S_2	S_3	0	0
110	S_6	S_4	S_5	0	0
111	S_7	S_6	S_7	0	0
Bảng 3-31. Bảng mã hoá trạng thái					

Tiến hành tối thiểu hóa trạng thái:

Từ bảng chuyển đổi trạng thái đã được mã hóa trên, nhận thấy có một số trạng thái tương đương nhau, nên có thể loại bớt được một số trạng thái.

$$S_0 \approx S_4 \approx S_{04} ; S_1 \approx S_5 \approx S_{15} ; S_3 \approx S_7 \approx S_{37}$$

Từ đó lập được bảng chuyển đổi trạng thái 3-32:

S^n	S^{n+1}		Z	
	X=0	X=1	X=0	X=1
S_{04}	S_{04}	S_{15}	0	0
S_{15}	S_2	S_{37}	0	0
S_2	S_{04}	S_{15}	0	1
S_{37}	S_6	S_{37}	0	0
S_6	S_{04}	S_{15}	0	0

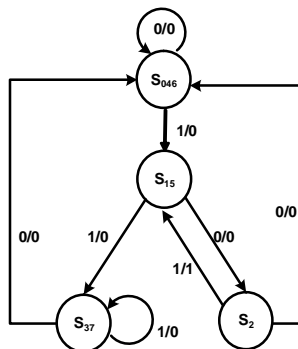
Bảng 3-32. Bảng chuyển đổi trạng thái sau khi tối thiểu hóa lần 1

Tiếp tục tiến hành tối thiểu hóa và lập được bảng 3-33:

S^n	S^{n+1}		Z	
	X=0	X=1	X=0	X=1
S_{046}	S_{046}	S_{15}	0	0
S_{15}	S_2	S_{37}	0	0
S_2	S_{046}	S_{15}	0	1
S_{37}	S_{046}	S_{37}	0	0

Bảng 3-33. Bảng chuyển đổi trạng thái sau khi tối thiểu hóa lần 2

Từ bảng chuyển đổi trạng thái lập được đồ hình trạng thái:



Hình 3-46. Đồ hình trạng thái.

Thực hiện mã hóa trạng thái như bảng 3-35.

S^n	Q_1Q_0
S_{046}	00
S_{15}	01
S_2	11
S_{37}	10
Bảng 3-35. Bảng mã hóa trạng thái	

Từ bảng 3-35, tính T_{on} , T_{off} để tìm phương trình hàm kích bằng cách tính T_{on} , T_{off} :

$$J_1 = Q_0$$

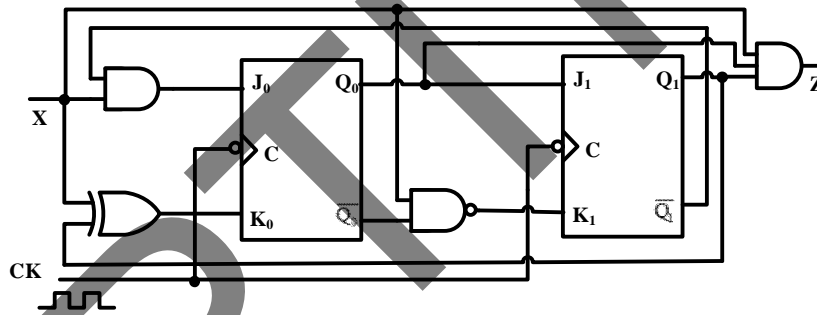
$$K_1 = \bar{X} + Q_0 = \overline{XQ_0}$$

$$J_0 = X\bar{Q}_1$$

$$K_0 = X \oplus Q_1$$

$$Z = XQ_1Q_0$$

Từ phương trình hàm kích vẽ được mạch sau:



Hình 3-47. Sơ đồ nguyên lý mạch tuần tự

3.7. MỘT SỐ VÍ DỤ KHÁC.

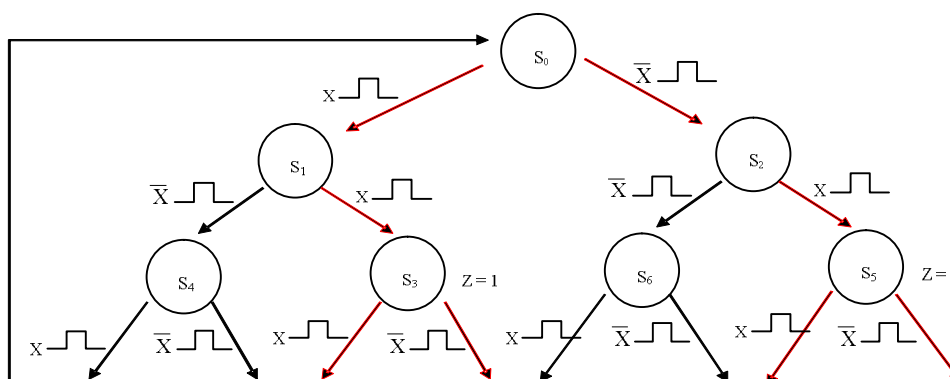
3.7.1. Mạch tuần tự đồng bộ.

Ví dụ 1: Thiết kế mạch tuần tự thực hiện nhiệm vụ kiểm tra dãy tín hiệu vào ở dạng nhị phân có độ dài bằng 3 được đưa vào liên tiếp trên đầu vào X. Nếu dãy tín hiệu vào có dạng là 010 hoặc 011 hoặc 110 hoặc 111 thì $Z = 1$. Các trường hợp khác $Z = 0$.

Giải:

Bước 1: Xác định bài toán. Mạch được thiết kế có nhiệm vụ phát hiện tín hiệu vào. Khi nhận được 1 trong các dãy tín hiệu trên thì mạch sẽ báo rằng đã nhận được.

Mạch phải thiết kế là mạch đồng bộ, nên sẽ có các đầu vào là X- tín hiệu vào, Ck- xung nhịp điều khiển, Z – tín hiệu ra.



Hình 3-48. Đồ hình trạng thái

Bước 2: Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái

Giả sử trạng thái ban đầu là S_0 :

Khi tín hiệu vào là X . Ck thì mạch sẽ chuyển tới trạng thái S_1 . Khi tín hiệu vào là \bar{X} . Ck mạch sẽ chuyển đến trạng thái S_2 .

Tương tự như vậy. Khi mạch ở trạng thái S_1 thì khi có tín hiệu X . Ck mạch chuyển đến trạng thái S_3 và chuyển đến trạng thái S_4 khi có tín hiệu \bar{X} . Ck. Tương tự, xây dựng được đồ hình 3-48.

Nếu mạch ở 1 trong 4 trạng thái S_3, S_4, S_5, S_6 : khi có tín hiệu vào X . Ck hoặc \bar{X} . Ck thì mạch sẽ chuyển về trạng thái ban đầu S_0 . Khi dãy tín hiệu vào là 110 hoặc 111 (ứng với đường chuyển đổi trạng thái là $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$) hay khi dãy tín hiệu vào là 010 hoặc 011 (ứng với đường chuyển đổi trạng thái là $S_0 \rightarrow S_2 \rightarrow S_5 \rightarrow S_0$) thì mạch sẽ cho tín hiệu ra $Z = 1$ tại thời điểm xung thứ 3. Với các đường chuyển đổi khác $Z = 0$.

Từ đồ hình trạng thái 3-48, xây dựng được bảng chuyển đổi trạng thái 3-36:

S^n	S^{n+1}		Z	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_2	S_1	0	0
S_1	S_4	S_3	0	0
S_2	S_6	S_5	0	0
S_3	S_0	S_0	1	1
S_4	S_0	S_0	0	0
S_5	S_0	S_0	1	1
S_6	S_0	S_0	0	0

Bảng 3-36. Bảng chuyển đổi trạng thái

Bước 3: Tối thiểu hoá trạng thái. Để có được sơ đồ mạch đơn giản phải tối thiểu hoá các trạng thái.

$\begin{array}{c} X \\ S \end{array}$	0	1
S_0	S_2 $Z = 0$	S_1 $Z = 0$
S_1	S_{46} $Z = 0$	S_{35} $Z = 0$
S_2	S_{46} $Z = 0$	S_{35} $Z = 0$
S_{35}	S_0 $Z = 1$	S_0 $Z = 1$
S_{46}	S_0 $Z = 0$	S_0 $Z = 0$

Bảng 3-37. Bảng chuyển đổi trạng thái sau khi gộp S_3 và S_5 ; S_4 và S_6

$\begin{array}{c} X \\ S \end{array}$	0	1
S_0	S_{12} $Z = 0$	S_{12} $Z = 0$
S_{12}	S_{46} $Z = 0$	S_{35} $Z = 0$
S_{35}	S_0 $Z = 1$	S_0 $Z = 1$
S_{46}	S_0 $Z = 0$	S_0 $Z = 0$

Bảng 3-38. Bảng chuyển đổi trạng thái sau khi gộp S_1 và S_2

Áp dụng quy tắc Caldwell cho bài toán trên thấy trạng thái S_4 tương đương với trạng thái S_6 ($S_4 \approx S_6$), S_3 tương đương với S_5 ($S_3 \approx S_5$). Thay thế các trạng thái tương đương bằng một trạng thái chung đặc trưng cho chúng. Ví dụ thay thế S_4, S_6 bằng S_{46} , thay thế S_3, S_5 bằng S_{35} . Từ đó lập được bảng chuyển đổi trạng thái 3-37 và 3-38:

Bước 4: Mã hóa trạng thái.

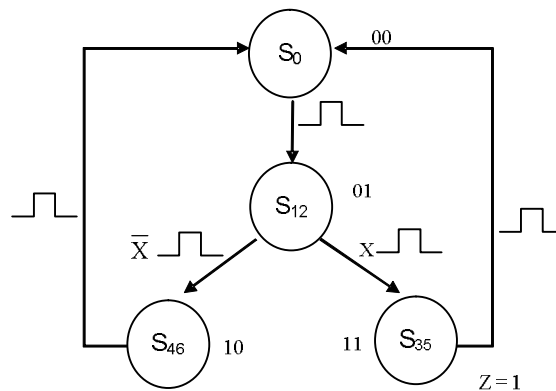
Sau khi gộp hai trạng thái S_1 và S_2 thành trạng thái chung S_{12} thì mạch chỉ còn 4 trạng thái $S_0, S_{12}, S_{35}, S_{46}$. Mã hoá 4 trạng thái này bằng hai biến nhị phân Q_1 và Q_0 (bảng 3-39). Từ đó, vẽ được đồ hình trạng thái 3-49.

Bước 5: Xác định hệ phương trình của mạch.

Có hai cách xác định hệ phương trình này.

Q_1	Q_0	Mã hoá S
0	0	S_0
0	1	S_{12}
1	1	S_{35}
1	0	S_{46}

Bảng 3-39. Bảng mã hoá trạng thái



Hình 3-49. Đồ hình trạng thái

Cách 1: Dựa vào bảng chuyển đổi trạng thái, lập bảng hàm kích 3-40 cho hai trigơ Q_0 và Q_1 .

Dùng bảng Các nô 3-41 để rút gọn, thu được kết quả sau:

$$J_1 = Q_0 ; K_1 = 1$$

$$J_0 = \overline{Q_1} ; K_0 = \overline{X} + Q_1 ; Z = Q_0 Q_1$$

Trạng thái hiện tại	Trạng thái kế tiếp		Các đầu vào của trigơ							
	X = 0	X = 1	X = 0		X = 1		X = 0		X = 1	
$Q_1 Q_0$	$Q_1 Q_0$	$Q_1 Q_0$	J_1	K_1	J_1	K_1	J_0	K_0	J_0	K_0
00	01 $Z = 0$	01 $Z = 0$	0	X	0	X	1	X	1	X
01	10 $Z = 0$	11 $Z = 0$	1	X	1	X	X	1	X	0
11	00 $Z = 1$	00 $Z = 1$	X	1	X	1	X	1	X	1
10	00 $Z = 0$	00 $Z = 0$	X	1	X	1	0	X	0	X

Bảng 3-40. Bảng hàm kích thích

$Q_1 Q_0$
 X

	00	01	11	10
0	0	1	x	x
1	0	1	x	x

$J_1 = Q_0$

$Q_1 Q_0$
 X

	00	01	11	10
0	x	x	1	1
1	x	x	1	1

$K_1 = 1$

$Q_1 Q_0$
 X

	00	01	11	10
0	1	x	x	0
1	1	x	x	0

$J_0 = \overline{Q_1}$

$Q_1 Q_0$
 X

	00	01	11	10
0	x	1	1	x
1	x	0	1	x

$K_0 = \overline{X} + Q_1$

Q_1Q_0		00	01	11	10
X	0	0	0	1	0
	1	0	0	1	0

Bảng 3-41. Bảng Các nô tìm hàm kích và hàm ra.

$$Z = Q_1Q_0$$

Cách 2: Dựa trực tiếp vào đồ hình trạng thái viết phương trình T_{on} , T_{off} của từng trigơ và phương trình tín hiệu ra.

Đối với trigơ JK :

$$T_{onQ} = T^* \bar{Q} \Rightarrow J_Q = T^*$$

$$T_{offQ} = \bar{T}^{**}Q \Rightarrow K_Q = T^{**}$$

Đối với trường hợp này có:

$$T_{onQ_1} = S_{12}X + S_{12}\bar{X} = S_{12} = \bar{Q}_1Q_0 \Rightarrow J_1 = Q_0$$

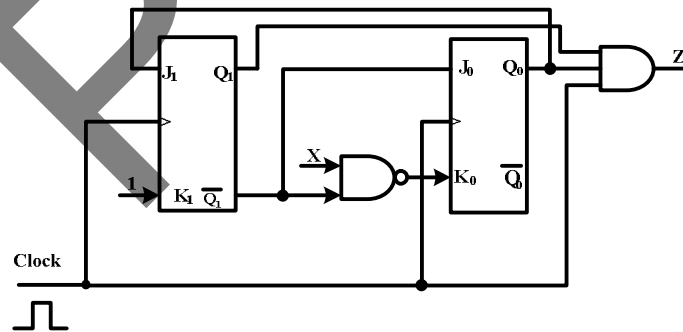
$$T_{offQ_1} = S_{35} + S_{46} = Q_1Q_0 + Q_1\bar{Q}_0 = Q_1 \Rightarrow K_1 = 1$$

$$T_{onQ_0} = S_0 = \bar{Q}_1\bar{Q}_0 \Rightarrow J_0 = \bar{Q}_1$$

$$T_{offQ_0} = S_{12}\bar{X} + S_{35} = \bar{Q}_1Q_0\bar{X} + Q_1Q_0 = Q_0(\bar{Q}_1\bar{X} + Q_1) \Rightarrow K_0 = \bar{Q}_1\bar{X} + Q_1 = \bar{X} + Q_1$$

Phương trình hàm ra $Z = Q_0Q_1C$

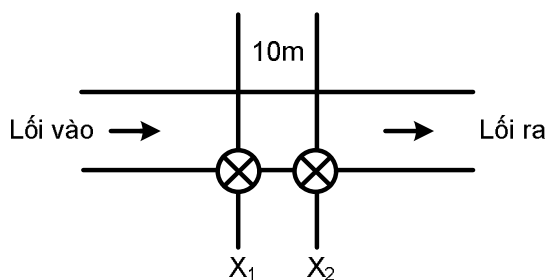
Bước 6: Sơ đồ mạch điện được thể hiện ở hình 3-50.



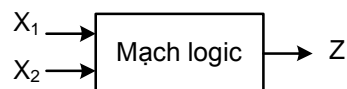
Hình 3-50. Mạch tuần tự kiểm tra dãy tín hiệu vào

3.7.2. Mạch tuần tự không đồng bộ

Một mạch tuần tự không đồng bộ được thiết kế để đếm số người vào thăm một viện bảo tàng. Mạch gồm hai đèn X_1 , X_2 được bố trí cách nhau 10 mét. Mạch được thiết kế sao cho mỗi lần chỉ đếm được một người.



Hình 3-51. Cách bố trí đèn ở viện bảo tàng



Hình 3-52 . Sơ đồ khối của mạch

Lời giải:

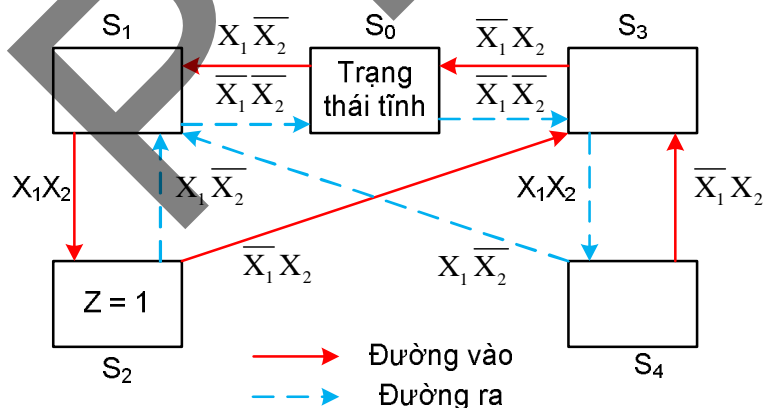
Khi có một người đi vào thì hai đèn sẽ bị chắn liên tiếp. Đầu tiên X_1 bị chắn, tiếp đến cả X_1 và X_2 cùng bị chắn, sau đó chỉ X_2 bị chắn. Lúc này, mạch cho ra tín hiệu $Z = 1$. Khi một người ra thì sẽ ngược lại. Đầu tiên đèn X_2 sẽ bị chắn, sau đó cả X_1 và X_2 cùng bị chắn và cuối cùng chỉ có X_1 bị chắn. Sơ đồ khối của mạch tạo tín hiệu đếm Z được mô tả bởi hình 3-52.

Hai đầu vào của mạch là $X_1 X_2$. Đầu ra Z được đưa tới đầu vào của bộ giải mã.

Quy ước: đèn bị chắn = X ; ngược lại = \bar{X} .

Đồ hình trạng thái được mô tả ở hình 3-53.

S_0 là trạng thái ban đầu của mạch. Nếu một người đi vào thì sự chuyển đổi của mạch sẽ là $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_0$. Nếu một người đi ra thì quá trình chuyển đổi trạng thái của mạch là $S_0 \rightarrow S_3 \rightarrow S_4 \rightarrow S_1 \rightarrow S_0$. Khi có một người ngấp ngừng sau đó lại quay ra ban đầu chắn đèn X_1 sau đó quay ra thì mạch sẽ chuyển đổi trạng thái $S_0 \rightarrow S_1 \rightarrow S_0$, lúc đó mạch sẽ không thực hiện đếm.



Hình 3-53. Đồ hình trạng thái

Tương ứng với đồ hình trạng thái trên, lập được bảng chuyển đổi trạng thái ở bảng 3-42.

Trạng thái hiện tại	Trạng thái kế tiếp và tín hiệu ra			
	$X_1 \ X_2$ 0 0	$X_1 \ X_2$ 0 1	$X_1 \ X_2$ 1 1	$X_1 \ X_2$ 1 0
S_0	$S_0 / Z = 0$	$S_3 / Z = 0$	$S_2 / Z = 1$	$S_1 / Z = 0$
S_1	$S_0 / Z = 0$	$S_3 / Z = 0$	$S_2 / Z = 1$	$S_1 / Z = 0$
S_2	$S_0 / Z = 0$	$S_3 / Z = 0$	$S_2 / Z = 1$	$S_1 / Z = 0$
S_3	$S_0 / Z = 0$	$S_3 / Z = 0$	$S_4 / Z = 0$	$S_1 / Z = 0$
S_4	$S_0 / Z = 0$	$S_3 / Z = 0$	$S_4 / Z = 0$	$S_1 / Z = 0$

Bảng 3-42. Bảng chuyển đổi trạng thái

Bảng có 5 hàng ứng với 5 trạng thái hiện tại có thể xuất hiện và 4 cột, mỗi cột ứng với một tổ hợp giá trị có thể của X_1, X_2 . Mỗi ô của bảng biểu diễn trạng thái kế tiếp và tín hiệu ra tương ứng với trạng thái hiện tại và giá trị của tín hiệu vào X_1, X_2 .

Trong bảng chuyển đổi trạng thái, những ô được khoanh tròn là những ô có trạng thái kế tiếp bằng trạng thái hiện tại. Những trạng thái đó là những trạng thái ổn định. Điều kiện cho trạng thái ổn định là $S^k = S$.

Trên bảng có những ô trống. Những ô này tương ứng với các tổ hợp tín hiệu không xuất hiện ở đầu vào. Những ô này có thể điền giá trị tùy chọn để tối thiểu hoá hệ phương trình của mạch.

Tiến hành tối thiểu hoá:

Có thể gán trạng thái kế tiếp và tín hiệu ra vào các ô trống sao cho hàng có ô trống có thể kết hợp với các hàng khác.

Ở bảng chuyển đổi trạng thái các hàng S_0, S_1, S_2 , và S_3, S_4 có các trạng thái kế tiếp và tín hiệu ra tương ứng là giống nhau nếu như gán:

- ô trống của hàng đầu tiên (ứng với S_0) là $S_2 / Z = 1$,
- ô trống của hàng thứ hai là $S_3 / Z = 0$,
- ô trống của hàng thứ tư là $S_1 / Z = 0$,
- ô trống của hàng thứ ba và thứ năm là $S_0 / Z = 0$,

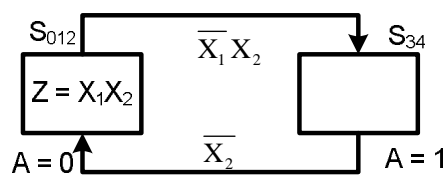
Nói cách khác, nếu thiếu trạng thái nào trong các hàng của bảng trạng thái thì có thể gán trạng thái còn thiếu đó vào chỗ trống của bảng để tiến hành rút gọn.

Khi đó bảng chuyển đổi trạng thái được rút gọn lại như bảng 3-43.

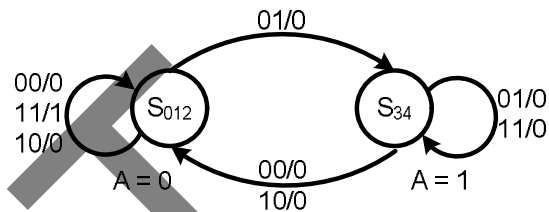
Mạch chỉ có hai trạng thái nên để mã hoá chỉ cần sử dụng một biến nhị phân A. Để mã hoá trạng thái S_{012} thì $A = 0$, S_{34} thì $A = 1$. Tín hiệu ra $Z = 1$ ở trạng thái S_{012} khi $X_1 X_2 = 11$.

Trạng thái hiện tại	Trạng thái kế tiếp và tín hiệu ra			
	$X_1 \ X_2$ 0 0	$X_1 \ X_2$ 0 1	$X_1 \ X_2$ 1 1	$X_1 \ X_2$ 1 0
S_{012}	$S_{012} / Z = 0$	$S_{34} / Z = 0$	$S_{012} / Z = 1$	$S_{012} / Z = 0$
S_{34}	$S_{012} / Z = 0$	$S_{34} / Z = 0$	$S_{012} / Z = 0$	$S_{012} / Z = 0$

Bảng 3-43 . Bảng chuyển đổi trạng thái và hàm ra rút gọn



Hình 3-54. Đồ hình trạng thái rút gọn



Hình 3-55. Đồ hình trạng thái tương đương

Từ đó, xây dựng được đồ hình trạng thái sau khi rút gọn ở hình 3-54.

Để tìm hàm kích cho đơn giản, vẽ đồ hình trạng thái tương đương với các trạng thái vào/ra (hình 3-55)

Dùng trigơ RS để thiết kế (dựa vào bảng hàm kích của trigơ RS).

Cung	Q^n	Q^{n+1}	R	S
0	0	0	X	0
T_{on}	0	1	0	1
T_{off}	1	0	1	0
1	1	1	0	X

Bảng 3-44. Bảng hàm kích của trigơ RS

Có phương trình đầu vào kích (R, S) của trigơ là:

$$S = \text{tập hợp bật của } Q + [(1)] = T_{on} + [(1)];$$

Tập hợp bật của Q (T_{on}) là các cung mà Q chuyển từ 0 \rightarrow 1.

$$R = \text{tập hợp tắt của } Q + [(0)] = T_{off} + [(0)] ;$$

Tập hợp tắt của Q (T_{off}) là các cung mà Q chuyển từ 1 \rightarrow 0.

Các cung $[(0)]$, $[(1)]$ được lấy giá trị không xác định (x) và được dùng để tối thiểu hoá.

Từ đó, lập bảng Các nô để tối thiểu hóa hàm kích:

$X_1 X_2$					
		00	01	11	10
A		0	1	0	0
0		0	1	0	0
1		0	x	x	0

$$S_A = \overline{X_1} X_2$$

$X_1 X_2$					
		00	01	11	10
A		x	0	x	x
0		x	0	x	x
1		1	0	0	1

$$R_A = \overline{X_2}$$

Bảng 3-45. Bảng Các nô tìm hàm kích

Từ bảng 3-45, tìm được hàm kích cho trigơ RS:

$$S_A = \overline{X_1} X_2 ; \quad R_A = \overline{X_2}$$

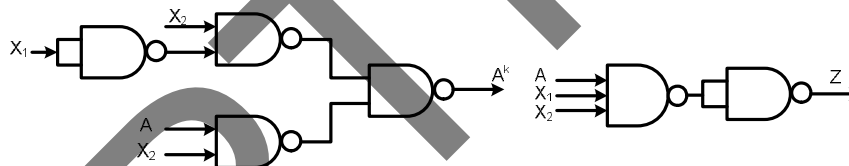
Phương trình đặc trưng của trigơ RS: $Q^{n+1} = S_A + \overline{R_A} Q_A$

Thay giá trị của R_A , S_A vào biểu thức thu được mạch toàn NAND:

$$A^k = \overline{X_1} X_2 + X_2 A = \overline{\overline{\overline{\overline{X_1} X_2}} + \overline{\overline{\overline{\overline{X_2} A}}}} = \overline{\overline{\overline{\overline{X_1} X_2}} \cdot \overline{\overline{\overline{\overline{X_2} A}}}} = \overline{\overline{\overline{\overline{X_1} X_2}} \cdot \overline{\overline{\overline{\overline{X_2} A}}}}$$

Phương trình ra: $Z = \overline{A} X_1 X_2$

Từ đó vẽ được sơ đồ mạch dùng NAND (hình 3-49).

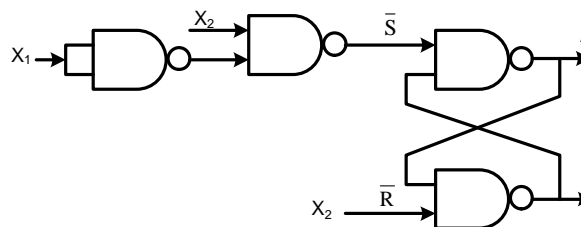


Hình 3-56. Sơ đồ mạch chỉ dùng NAND

Nếu thiết kế mạch dùng trigơ RS và các mạch NAND có:

$$S_A = \overline{X_1} X_2 \quad R_A = \overline{X_2}$$

Và mạch được biểu diễn ở hình 3-57.



Hình 3-57. Sơ đồ mạch dùng trigơ RS

3.8. MỘT SỐ MẠCH TUẦN TỰ THÔNG DỤNG

3.8.1. Bộ đếm.

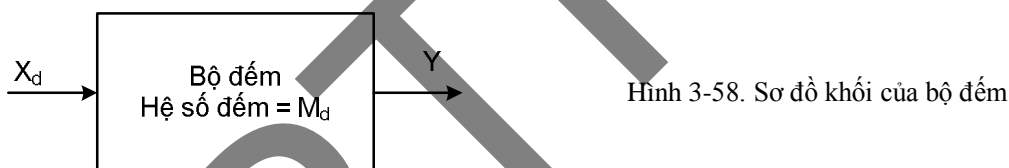
Bộ đếm là loại mạch tuần tự đơn giản, được xây dựng từ các phần tử nhớ là các trigơ và các mạch logic tổ hợp. Bộ đếm hoạt động theo chế độ tuần hoàn, có một đầu vào và một đầu ra chính. Kết thúc vòng tuần hoàn bộ đếm lại quay về trạng thái ban đầu và hàm ra lấy giá trị 1. Ngoài các đầu vào và đầu ra 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.

3.8.1.1. Định nghĩa và phân loại bộ đếm

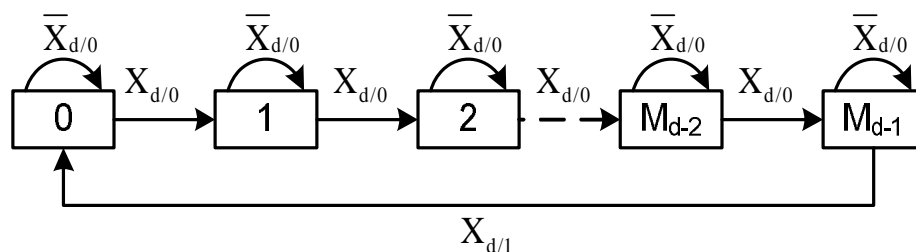
a . Định nghĩa.

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.



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 3- 58.



Đồ hình trạng thái của bộ đếm có hệ số đếm bằng M_d được mô tả ở hình 3-59. Khi không có tín hiệu vào đếm ($\overline{X_d}$) mạch giữ nguyên trạng thái cũ, khi có tín hiệu đếm (X_d) thì mạch sẽ chuyển đến trạng thái kế tiếp.

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.

3.8.1.2 Phương pháp cơ bản phân tích chức năng logic của bộ đếm.

Để phân tích mạch, người ta đưa ra một quy trình phân tích gồm 4 bước sau:

- + Viết phương trình.

Căn cứ vào mạch điện đã cho, viết phương trình định thời, phương trình đầu ra, phương trình kích, các công thức logic của tín hiệu đầu ra và tín hiệu đầu vào.

Tìm phương trình trạng thái.

Thay phương trình kích vào phương trình đặc trưng của trigơ tương ứng, sẽ tìm được phương trình trạng thái của mạch điện, cũng tức là phương trình trạng thái tiếp theo của các trigơ.

+ Lập bảng chuyển đổi trạng thái.

Đưa tất cả các tổ hợp có thể của trạng thái hiện tại và tín hiệu đầu vào của phương trình trạng thái và phương trình đầu ra, tính toán để tìm ra trạng thái tiếp theo và tín hiệu đầu ra tương ứng.

+ Vẽ đồ hình trạng thái và giản đồ xung.

+ Vẽ đồ thị thời gian cần lưu ý: trigơ chỉ chuyển đổi trạng thái tương ứng với sườn kích của xung đồng hồ xuất hiện.

Quy trình 4 bước trên là chung, không bắt buộc tuân theo máy móc, nên vận dụng linh hoạt trong các trường hợp cụ thể.

3.8.1.3 Bộ đếm đồng bộ

a. Bộ đếm nhị phân đồng bộ

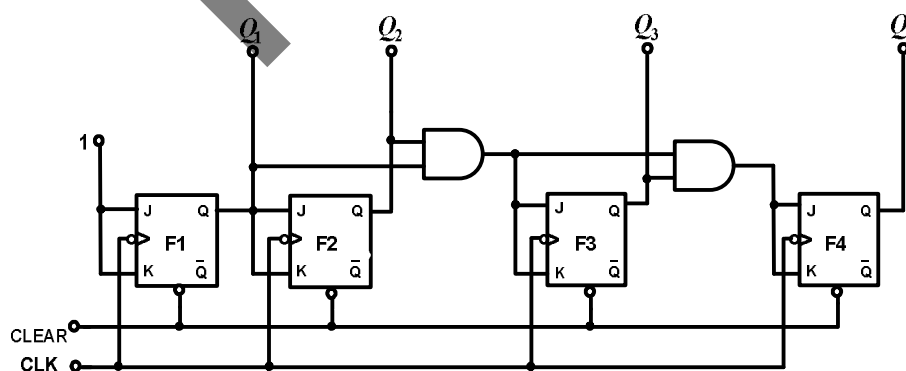
Bộ đếm nhị phân đồng bộ nói chung cấu trúc bằng trigơ T hoặc các trigơ loại khác nhưng mắc thành trigơ T. Hình 3-60 trình bày bộ đếm thuận nhị phân đồng bộ.

Nguyên lý làm việc:

Viết các phương trình:

Xung Clock của 4 trigơ đều là xung đếm đầu vào:

$$C_1 = C_2 = C_3 = C_4 = CLK \quad (3.40)$$



Hình 3-60. Bộ đếm thuận nhị phân đồng bộ 4 bit

Phương trình kích:

$$\left. \begin{aligned} T_1 &= 1 \\ T_2 &= Q_1^n \\ T_3 &= Q_1^n Q_2^n \\ T_4 &= Q_1^n Q_2^n Q_3^n \end{aligned} \right\} \quad (3.41)$$

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, có:

$$\left. \begin{aligned} Q_1^{n+1} &= T_1 \cdot \overline{Q_1^n} + \overline{T_1} \cdot Q_1^n = \overline{Q_1^n} \\ Q_2^{n+1} &= T_2 \cdot \overline{Q_2^n} + \overline{T_2} \cdot Q_2^n = Q_1^n \cdot \overline{Q_2^n} + \overline{Q_1^n} Q_2^n \\ Q_3^{n+1} &= T_3 \cdot \overline{Q_3^n} + \overline{T_3} \cdot Q_3^n = Q_1^n \cdot Q_2^n \cdot \overline{Q_3^n} + \overline{Q_1^n \cdot Q_2^n} Q_3^n \\ Q_4^{n+1} &= T_4 \cdot \overline{Q_4^n} + \overline{T_4} \cdot Q_4^n = Q_1^n \cdot Q_2^n \cdot Q_3^n \cdot \overline{Q_4^n} + \overline{Q_1^n \cdot Q_2^n \cdot Q_3^n} Q_4^n \end{aligned} \right\} \quad (3.42)$$

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.

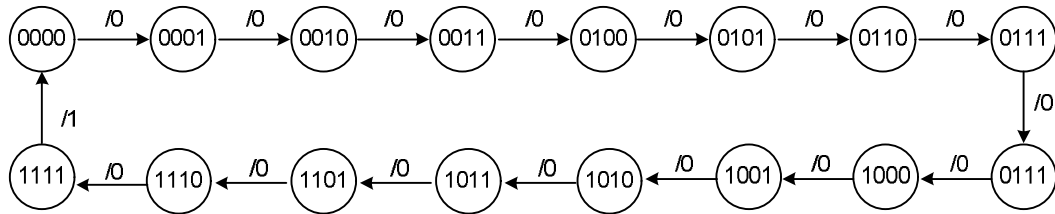
Dựa vào phương trình chuyển đổi trạng thái, tính được bảng kết quả 3-46.

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
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

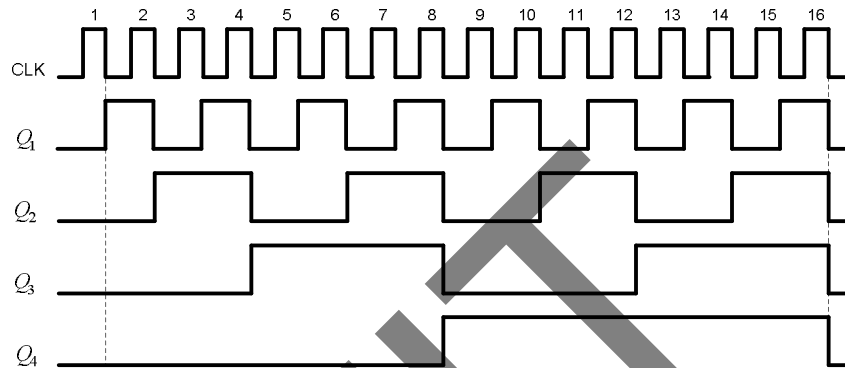
Bảng 3-46. Bảng trạng thái của bộ đếm

Từ bảng trạng thái, vẽ được đồ hình trạng thái như hình 3-61.



Hình 3-61. Đồ hình trạng thái của bộ đếm nhị phân

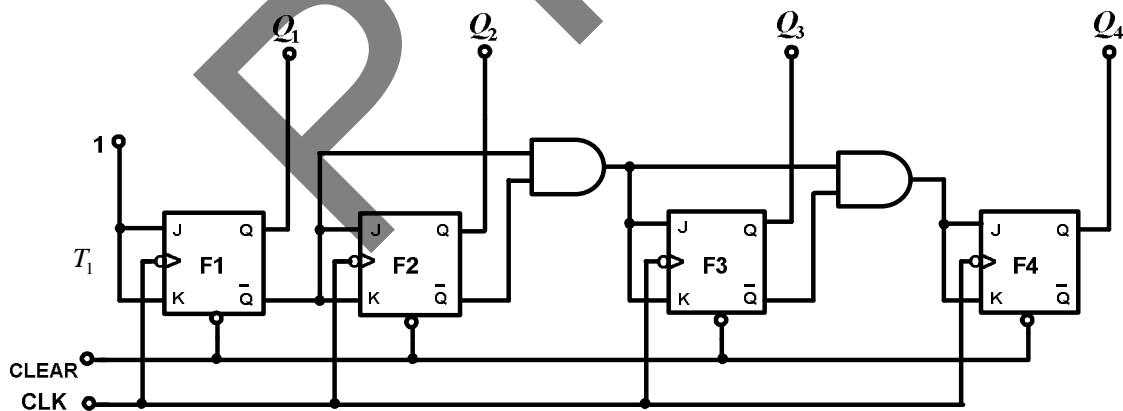
Dạng sóng của bộ đếm thuận nhị phân đồng bộ được trình bày ở hình 3-62.



Hình 3-62. Dạng sóng bộ đếm thuận nhị phân đồng bộ

b. Bộ đếm nghịch nhị phân đồng bộ

Sơ đồ được trình bày trên hình 3-63.



Hình 3-63. 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:

$$\left. \begin{aligned} T_1 &= 1 \\ T_2 &= \overline{Q_1^n} \\ T_3 &= \overline{Q_1^n} \cdot \overline{Q_2^n} \\ T_4 &= \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot \overline{Q_3^n} \end{aligned} \right\} \quad (3.43)$$

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, có:

$$\left. \begin{aligned} Q_1^{n+1} &= T_1 \cdot \overline{Q_1^n} + \overline{T_1} \cdot Q_1^n = \overline{Q_1^n} \\ Q_2^{n+1} &= T_2 \cdot \overline{Q_2^n} + \overline{T_2} \cdot Q_2^n = \overline{Q_1^n} \cdot \overline{Q_2^n} + Q_1^n Q_2^n \\ Q_3^{n+1} &= T_3 \cdot \overline{Q_3^n} + \overline{T_3} \cdot Q_3^n = \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot \overline{Q_3^n} + (Q_1^n + Q_2^n) \cdot Q_3^n \\ Q_4^{n+1} &= T_4 \cdot \overline{Q_4^n} + \overline{T_4} \cdot Q_4^n = \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot \overline{Q_3^n} \cdot \overline{Q_4^n} + (Q_1^n + Q_2^n + Q_3^n) \cdot Q_4^n \end{aligned} \right\} \quad (3.44)$$

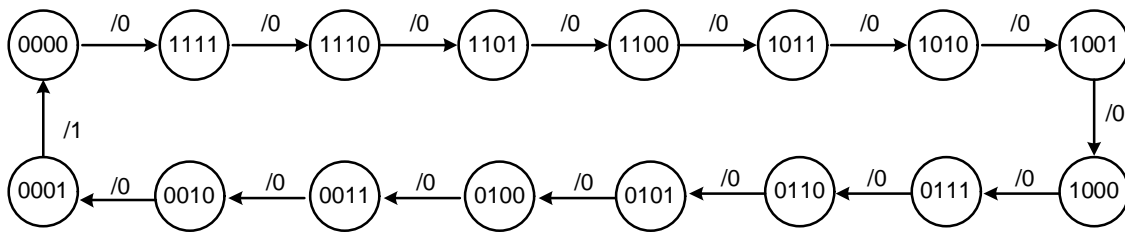
Khi có xung xóa Clear = 0 tất cả $Q_4^n Q_3^n Q_2^n Q_1^n = 0000$;

Tương tự như phần trên, dùng các phương pháp phân tích cơ bản, có bảng chuyển trạng thái 3-47.

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

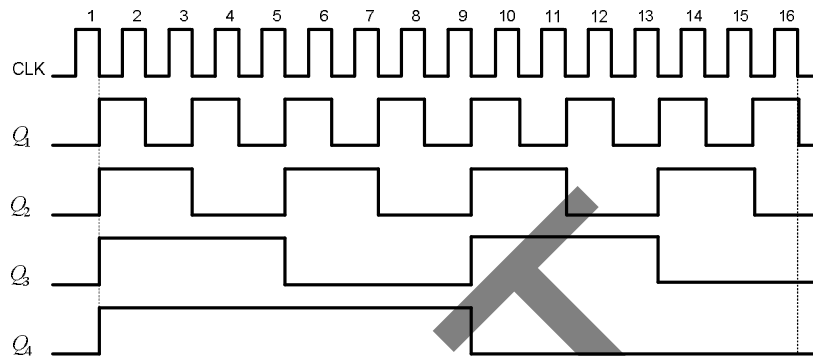
Bảng 3-47. Bảng trạng thái bộ đếm nghịch nhị phân đồng bộ

Từ bảng trạng thái, vẽ được đồ hình trạng thái như hình 3-64.



Hình 3-64. Đồ hình trạng thái của bộ đếm nhị phân

Dạng sóng của bộ đếm nghịch được trình bày trên hình 3-65.

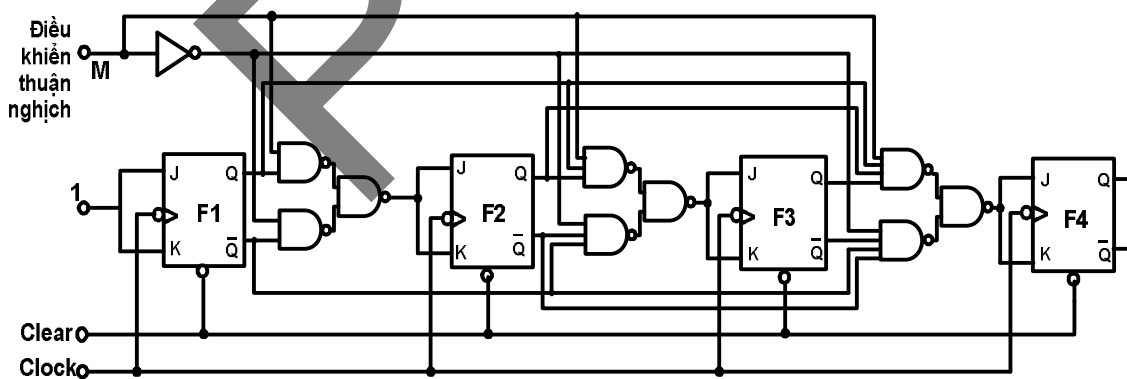


Hình 3-65. Đồ thị dạng xung của bộ đếm nghịch đồng bộ

Nhận xét: So sánh hình 3-60 và hình 3-63, thấy sự khác biệt của bộ đếm nghịch với bộ đếm thuận là đầu ra \bar{Q} của trigơ cùng cấp tín hiệu chuyển vị.

c. Bộ đếm thuận nghịch nhị phân đồng bộ.

Sơ đồ được trình bày trên hình 3-66.



Hình 3-66. Bộ đếm thuận nghịch nhị phân đồng bộ.

Mạch có đầu vào điều khiển đếm thuận, đếm nghịch.

$M = 1$ mạch đếm thuận.

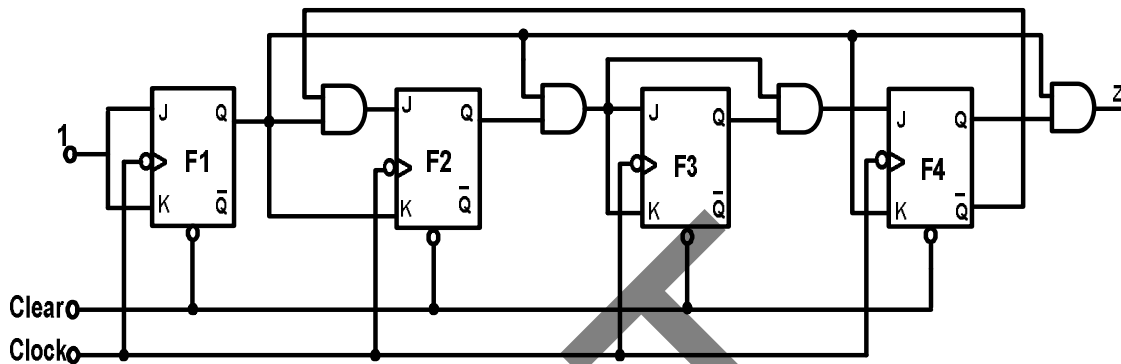
$M = 0$ mạch đếm nghịch

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 3-67.



Hình 3-67. Bộ đếm thập phân đồng bộ

Mạch gồm 4 trigơ JK và một số cổng logic 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

$$C_1 = C_2 = C_3 = C_4 = \text{CLK} \quad (3.45)$$

+ Phương trình hàm ra:

$$Z = Q_4^n \cdot Q_1^n \quad (3.46)$$

+ Phương trình kích:

$$\left. \begin{aligned} J_1 &= K_1 = 1 \\ J_2 &= \overline{Q_4^n} \cdot Q_1^n ; & K_2 &= Q_1^n \\ J_3 &= K_3 = Q_2^n \cdot Q_1^n \\ J_4 &= Q_3^n \cdot Q_2^n \cdot Q_1^n ; & K_4 &= Q_1^n \end{aligned} \right\} \quad (3.47)$$

+ 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:

$$\left. \begin{aligned} Q_1^{n+1} &= J_1 \cdot \overline{Q_1^n} + \overline{K_1} \cdot Q_1^n = \overline{Q_1^n} \\ Q_2^{n+1} &= J_2 \cdot \overline{Q_2^n} + \overline{K_2} \cdot Q_2^n = Q_1^n \cdot \overline{Q_2^n} + \overline{Q_1^n} \cdot Q_2^n \\ Q_3^{n+1} &= J_3 \cdot \overline{Q_3^n} + \overline{K_3} \cdot Q_3^n = Q_1^n \cdot Q_2^n \cdot \overline{Q_3^n} + \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot Q_3^n \\ Q_4^{n+1} &= J_4 \cdot \overline{Q_4^n} + \overline{K_4} \cdot Q_4^n = Q_1^n \cdot Q_2^n \cdot Q_3^n \cdot \overline{Q_4^n} + \overline{Q_1^n} \cdot \overline{Q_2^n} \cdot \overline{Q_3^n} \cdot Q_4^n \end{aligned} \right\} \quad (3.48)$$

Lập bảng trạng thái:

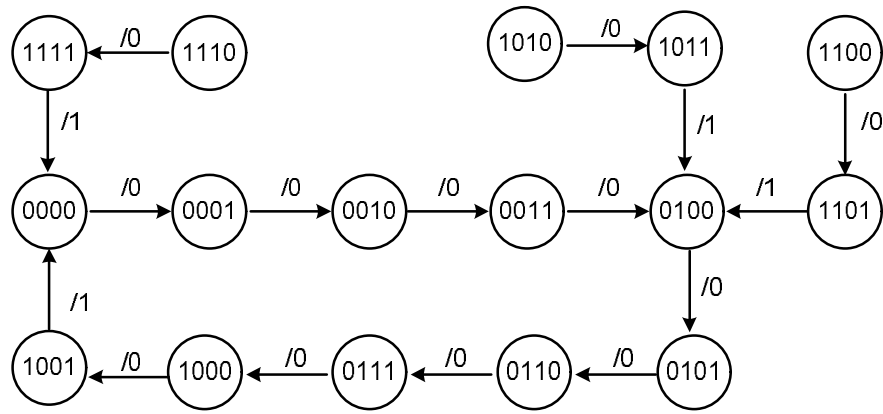
Bắt đầu từ giá trị $Q_4^n Q_3^n Q_2^n Q_1^n = 0000$ thay vào phương trình trạng thái và phương trình đầu ra, được kết quả ở bảng 3-47.

Giả sử bộ đếm hình 3-67 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 3-47. Bảng trạng thái của bộ đếm thập phân đồng bộ

Đồ hình trạng thái của mạch được thể hiện trên hình 3-68.

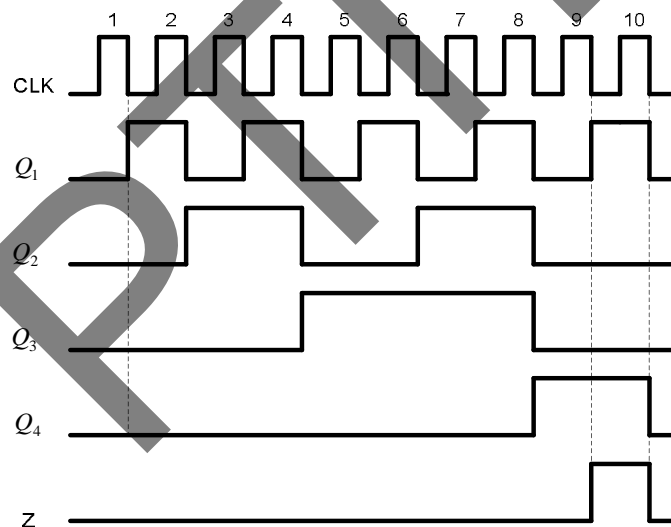


Hình 3-68. Đồ hình trạng thái

Lưu ý:

Trong bảng trạng thái 3-47 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 tức là mạch có thể tự khởi động.

Dạng sóng của bộ đếm thập phân đồng bộ được trình bày ở hình 3-69.



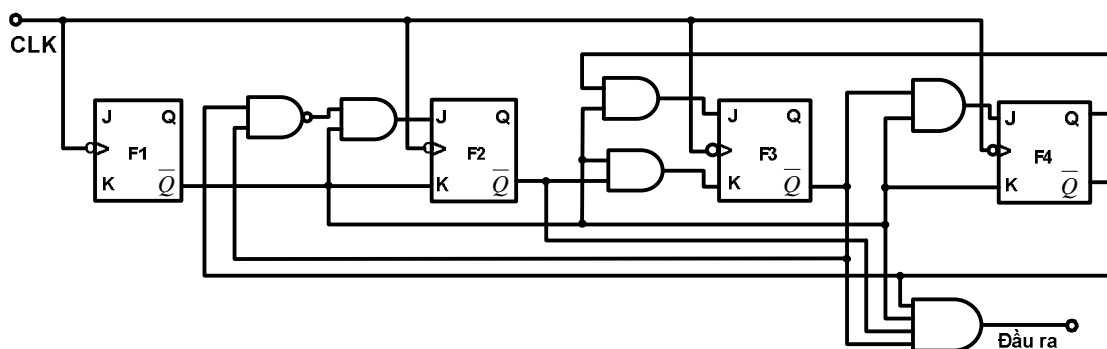
Hình 3-69. Dạng sóng bộ đếm thập phân đồng bộ

e. Bộ đếm nghịch thập phân đồng bộ.

Mạch điện được trình bày trên hình 3-70.

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.



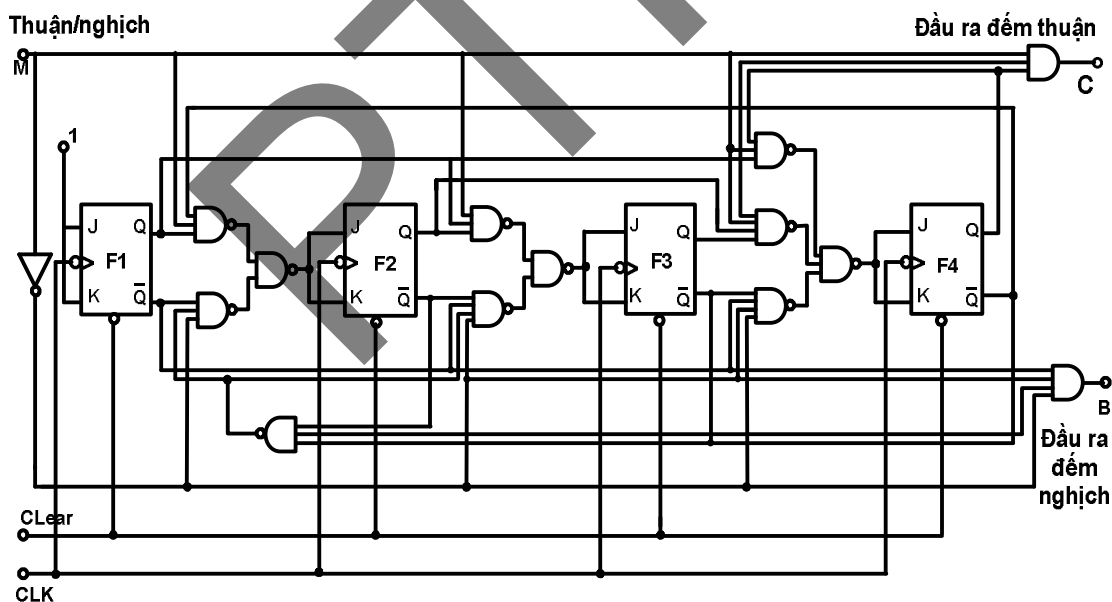
Hình 3-70. Bộ đếm nghịch thập phân đồng bộ

Cách thứ nhất là mắc dây chuyền 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 công chuyển vị (công 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.

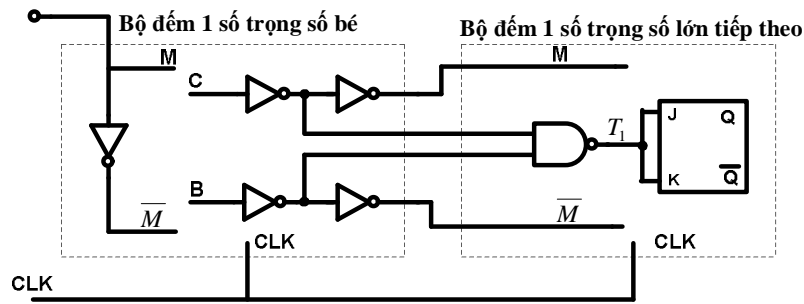
Sơ đồ được trình bày trên hình 3-71



Hình 3-71. 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 3-72.



Hình 3-72. 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.

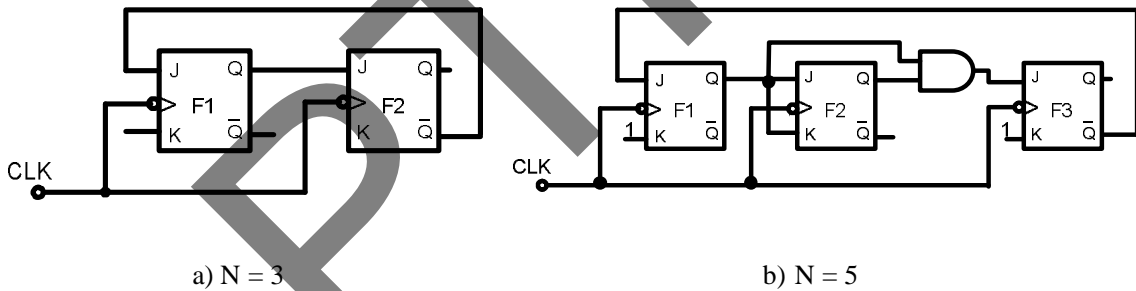
Khi bộ đếm có trọng số bé $B = 1$ thì bộ đếm trọng số lớn $\overline{M} = 1$, $T_1 = 1$ nó sẽ đếm nghịch.

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.

g. 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 3-73.



Hình 3-73. Bộ đếm đồng bộ N phân

3.8.1.4 Bộ đếm không đồng bộ

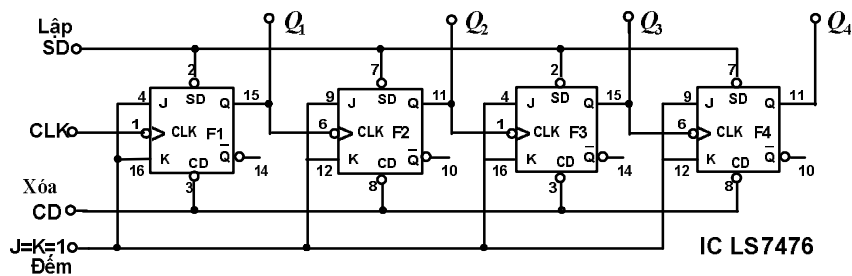
a. Bộ đếm nhị phân không đồng bộ

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 3-74.



Hình 3-74. Sơ đồ bộ đếm thuận nhị phân không đồng bộ

- Phương trình định thời:

$$CLK_1 = \downarrow CLK; CLK_1 = \downarrow Q_1; CLK_1 = \downarrow Q_2; CLK_1 = \downarrow Q_3; \quad (3.49)$$

- Phương trình trạng thái:

$$\left. \begin{aligned} Q_1^{n+1} &= \overline{Q_1^n} (\downarrow CLK) \\ Q_2^{n+1} &= \overline{Q_2^n} (\downarrow Q_1) \\ Q_3^{n+1} &= \overline{Q_3^n} (\downarrow Q_2) \\ Q_4^{n+1} &= \overline{Q_4^n} (\downarrow Q_3) \end{aligned} \right\} \quad (3.50)$$

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, có kết quả như bảng 3-48.

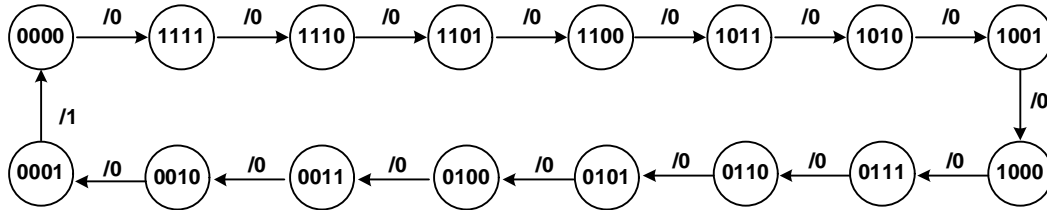
Q_4^n	Q_3^n	Q_2^n	Q_1^n	Q_4^{n+1}	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Điều kiện sườn âm
0	0	0	0	0	0	0	1	CLK_1
0	0	0	1	0	0	1	0	CLK_1, CLK_2
0	0	1	0	0	0	1	1	CLK_1
0	0	1	1	0	1	0	0	CLK_1, CLK_2, CLK_3
0	1	0	0	0	1	0	1	CLK_1
0	1	0	1	0	1	1	0	CLK_1, CLK_2
0	1	1	0	0	1	1	1	CLK_1
0	1	1	1	1	0	0	0	$CLK_1, CLK_2, CLK_3, CLK_4$
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 3-48. Bảng trạng thái của bộ đếm nhị phân không đồng bộ

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 $CD = 1$, $SD = 1$

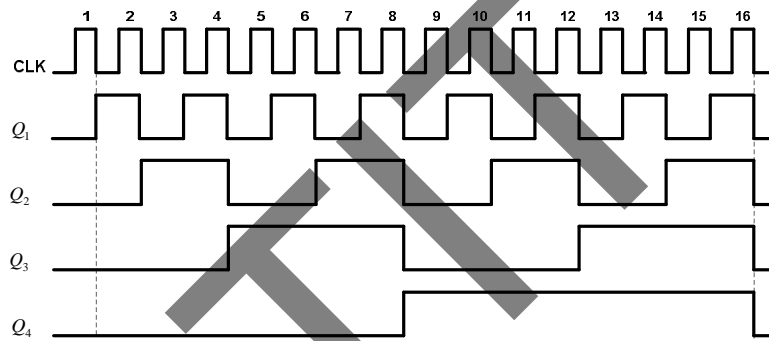
Để trigơ JK trở thành trigơ T' đầu vào đếm $J = K = 1$.

Hình 3-75 là đồ hình trạng thái của bộ đếm.



Hình 3-73. Đồ hình trạng thái của bộ đếm nhị phân

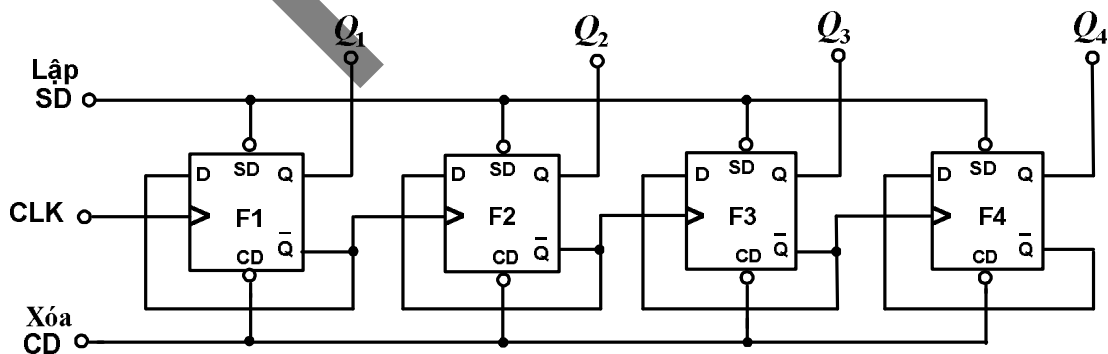
Dạng sóng của bộ đếm được trình bày ở hình 3-76.



Hình 3-76. 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 3-77).

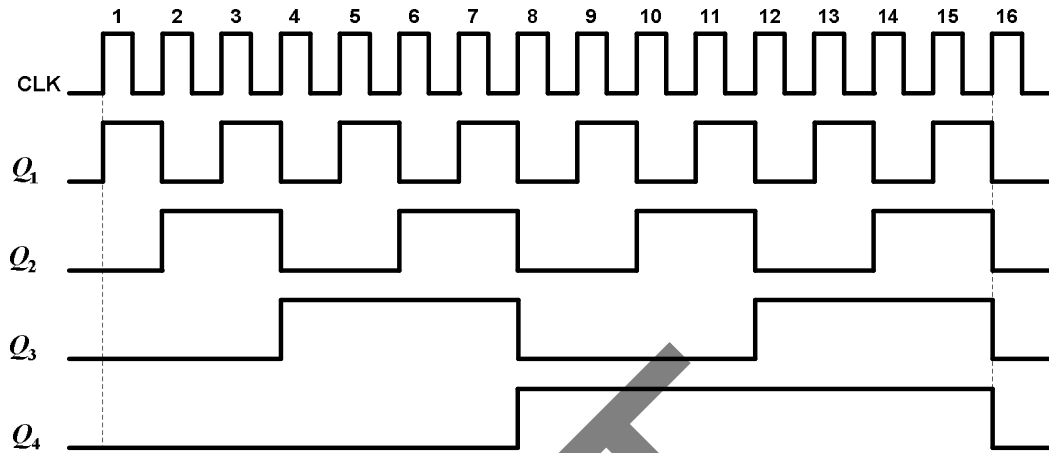


Hình 3-77. Sơ đồ đếm thuận nhị phân không đồng bộ kích bằng sườn dương

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 \bar{Q} của trigơ có trọng số bé hơn liền kề.

Nguyên tắc hoạt động của sơ đồ này cũng giống như sơ đồ dùng trigger JK kích bằng sườn âm, cần chú ý điều kiện định thời kích bằng sườn dương.

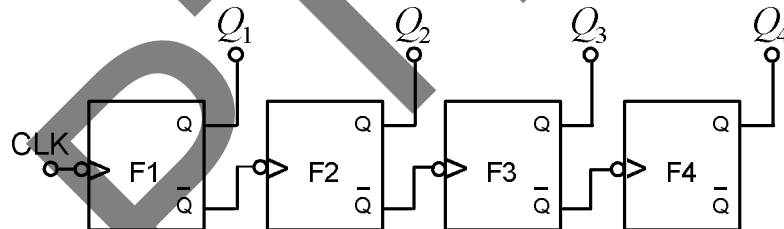
Dạng sóng của bộ đếm kiểu này được trình bày ở hình 3-78, chú ý là sườn xuống của Q là sườn lên của \overline{Q} .



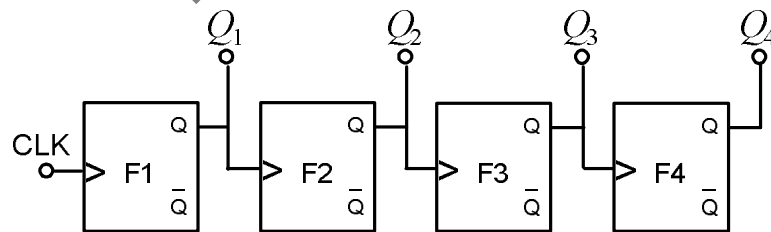
Hình 3-78. Dạng sóng của bộ đếm thuận nhị phân không đồng bộ kích bằng sườn dương

c. Bộ đếm nghịch nhị phân không đồng bộ

Sơ đồ bộ đếm này được trình bày ở hình 3-79 và 3-80 sử dụng các trigger mắc thành trigger T'.



Hình 3-79. Bộ đếm nghịch nhị phân không đồng bộ kích bằng sườn âm.



Hình 3-80. Bộ đếm nghịch nhị phân không đồng bộ kích bằng sườn dương.

- Phương trình định thời:

$$CLK_1 = \downarrow CLK; CLK_2 = \downarrow \overline{Q_1}; CLK_3 = \downarrow \overline{Q_2}; CLK_4 = \downarrow \overline{Q_3}; \quad (3.51)$$

Trigger T' trong sơ đồ trên sẽ lật trạng thái mỗi khi xuất hiện sườn âm của xung clock.

- Phương trình chuyển đổi trạng thái:

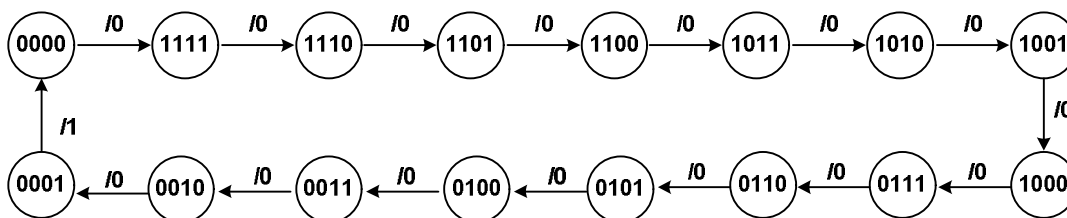
$$\left. \begin{aligned} Q_1^{n+1} &= \overline{Q_1^n} \quad (\downarrow \text{CLK}) \\ Q_2^{n+1} &= \overline{Q_2^n} \quad (\downarrow \overline{Q_1}) \\ Q_3^{n+1} &= \overline{Q_3^n} \quad (\downarrow \overline{Q_2}) \\ Q_4^{n+1} &= \overline{Q_4^n} \quad (\downarrow \overline{Q_3}) \end{aligned} \right\} \quad (3.52)$$

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, có kết quả như bảng 3-49.

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

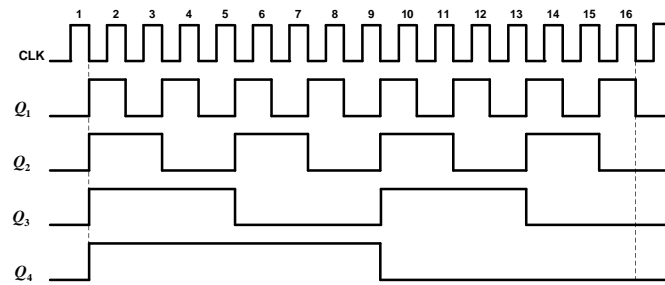
Bảng 3-49. Bảng trạng thái của bộ đếm nghịch

Hình 3-81 là đồ hình trạng thái của bộ đếm.

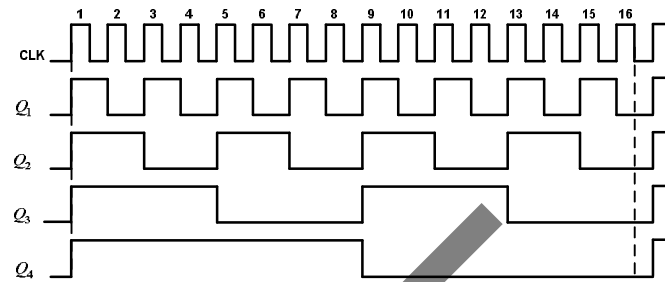


Hình 3-81. Đồ hình trạng thái của bộ đếm nhị phân

Dạng sóng của bộ đếm nghịch nhị phân không đồng bộ trình bày ở hình 3-82.



a) Kích bằng sườn âm



b) Kích bằng sườn dương

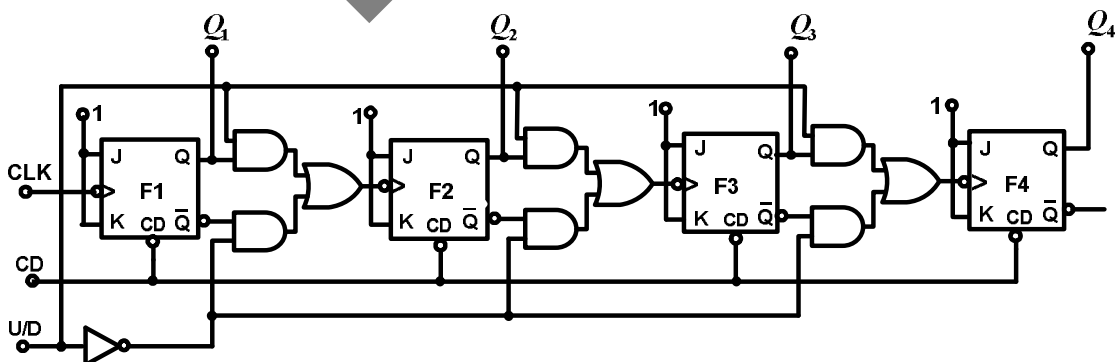
Hình 3-82. Dạng sóng của bộ đếm nghịch nhị phân không đồng bộ

d. Bộ đếm thuận nghịch nhị phân không đồng bộ

Sơ đồ được trình bày ở hình 3-83.

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 3-74. Đếm nghịch: Khi đầu vào UP/ Down = 0, đầu ra \bar{Q} của trigơ trước nối với đầu vào CLK của trigơ tiếp sau, mạch giống như sơ đồ hình 3-79.

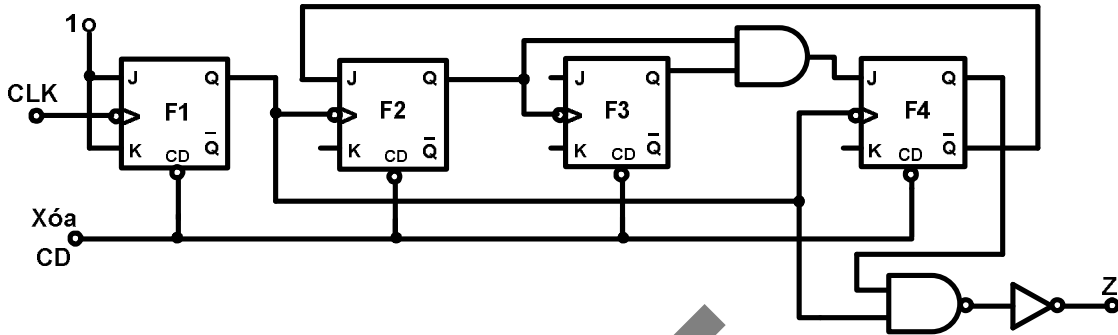


Hình 3-83. Bộ đếm thuận nghịch nhị phân không đồng bộ

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 3-84.



Hình 3-84. 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:

$$\left. \begin{aligned} C_1 &= \downarrow \text{CLK}; \\ C_2 &= C_4 = \downarrow Q_1 \\ C_3 &= \downarrow Q_2 \end{aligned} \right\} \quad (3.53)$$

Phương trình đầu ra:

$$Z = Q_4^n \cdot Q_1^n \quad (3.54)$$

Phương trình kích:

$$\left. \begin{aligned} J_1 &= K_1 = 1 \\ J_2 &= \overline{Q_4^n}; \quad K_2 = 1 \\ J_3 &= K_3 = 1 \\ J_4 &= Q_3^n \cdot Q_2^n; \quad K_4 = 1 \end{aligned} \right\} \quad (3.55)$$

Theo qui ước đầu vào để trống tương ứng với mức logic 1.

Tìm phương trình chuyển đổi trạng thái:

Thay các giá trị (3.55) vào phương trình đặc trưng của trigơ JK được:

$$\left. \begin{aligned} Q_1^{n+1} &= J_1 \cdot \overline{Q_1^n} + \overline{K_1} \cdot Q_1^n = \overline{Q_1^n} & (\downarrow \text{CLK}) \\ Q_2^{n+1} &= J_2 \cdot \overline{Q_2^n} + \overline{K_2} \cdot Q_2^n = \overline{Q_2^n} \cdot \overline{Q_4^n} & (\downarrow Q_1) \\ Q_3^{n+1} &= J_3 \cdot \overline{Q_3^n} + \overline{K_3} \cdot Q_3^n = \overline{Q_3^n} & (\downarrow Q_2) \\ Q_4^{n+1} &= J_4 \cdot \overline{Q_4^n} + \overline{K_4} \cdot Q_4^n = \overline{Q_2^n} \cdot \overline{Q_3^n} \cdot \overline{Q_1^n} & (\downarrow Q_1) \end{aligned} \right\} \quad (3.56)$$

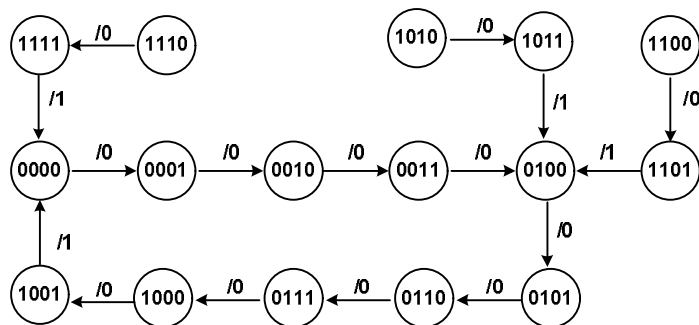
Lập bảng trạng thái:

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 lập được bảng 3-50.

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 3-50. Bảng trạng thái của bộ đếm thập phân không đồng bộ

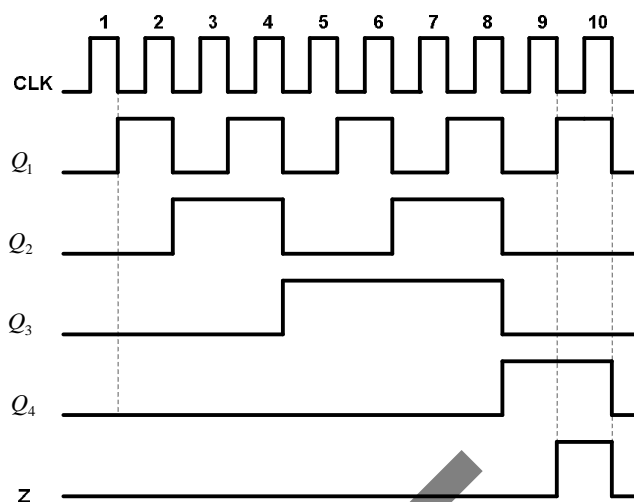
Đồ hình trạng thái của mạch được thể hiện trên hình 3-83.



Hình 3-83. Đồ hình trạng thái

Căn cứ vào đồ hình trạng thái, kết luận rằng bộ đếm này có khả năng tự khởi động.

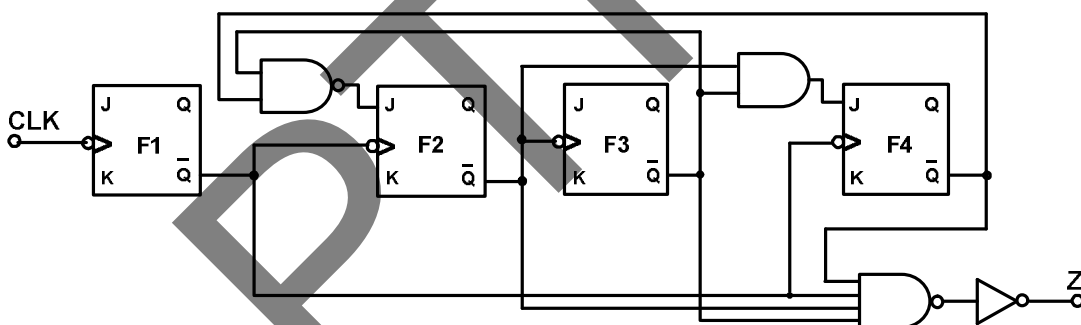
Dạng sóng của bộ đếm thuận thập phân đồng bộ được trình bày ở hình 3-86.



Hình 3-86. 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 3-87.



Hình 3-87. Bộ đếm nghịch thập phân không đồng bộ

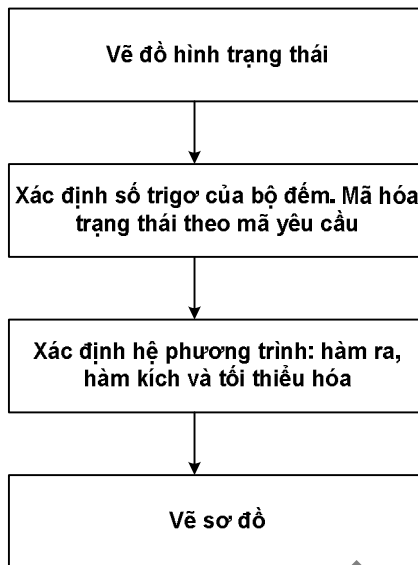
Cách phân tích cũng tương tự như ở mục trên.

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.

3.8.2. Thiết kế bộ đếm

3.8.2.1. Các bước thiết kế bộ đếm

Hình 3-88 là lưu đồ thiết kế bộ đếm.



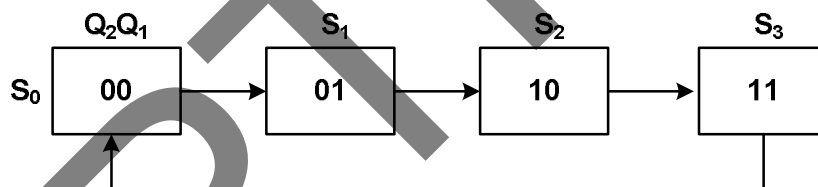
Hình 3-88. Lưu đồ thiết kế bộ đếm

3.10.2.2. 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 3-89.



Hình 3-89. 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 3-51 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 3-51. Bảng hàm kích

Tối thiểu hoá hàm kích của các trigơ, nhận được kết quả:

Đối với trigơ Q_1 :

$$R_1 = \overline{Q_2} Q_1 + Q_2 Q_1 = Q_1 (\overline{Q_2} + Q_2) = Q_1 ; \quad S_1 = \overline{Q_2} \overline{Q_1} + Q_2 \overline{Q_1} = \overline{Q_1} (\overline{Q_2} + Q_2) = \overline{Q_1}$$

$$J_1 = K_1 = 1;$$

$$T_1 = 1;$$

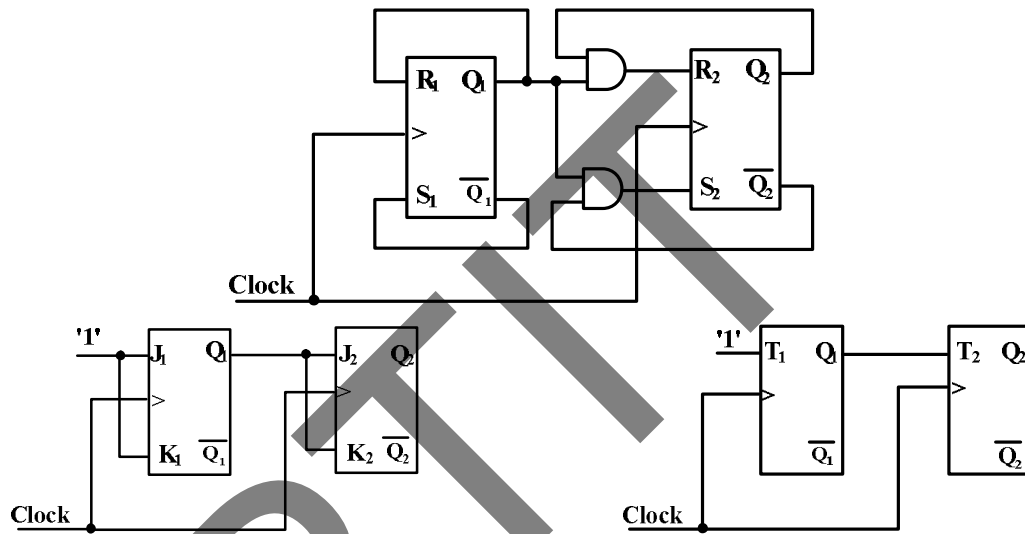
Đối với trigơ Q_2 :

$$R_2 = Q_2 Q_1; \quad S_2 = \overline{Q_2} Q_1$$

$$J_2 = K_2 = Q_1;$$

$$T_2 = Q_1;$$

Sơ đồ mạch điện như ở hình 3-90.

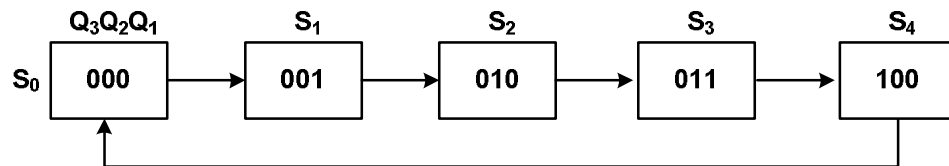


Hình 3-90. Bộ đếm Mod 4 dùng trigơ RS; JK; T

b. Bộ đếm N phân (hay bộ đếm có mod đếm bất kỳ)

Ví dụ: Thiết kế bộ đếm đồng bộ có $M_d = 3$.

Do $M_d = 5$ nên lập được đồ hình trạng thái như hình 3-91.



Hình 3-91. 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 3-52 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 3-52. Bảng hàm kích cho trigơ

Tối thiểu hoá hàm kích của các trigơ, nhận được kết quả:

$$J_1 = \overline{Q_3}; \quad K_1 = 1; \quad J_2 = K_2 = Q_2; \quad J_3 = Q_2 Q_1; \quad K_3 = 1;$$

Cách 2: Dùng bảng chuyển đổi trạng thái để tìm hàm kích:

		Q ₂ Q ₁			
		00	01	11	10
Q ₃	0	001	010	100	011
	1	000	x	x	x

Bảng 3-53. Bảng chuyển đổi trạng thái.

Từ bảng 3-53, tách ra làm 3 bảng chuyển đổi riêng biệt cho 3 trigơ để tìm phương trình chuyển đổi trạng thái của các Q₃ⁿ⁺¹, Q₂ⁿ⁺¹, Q₁ⁿ⁺¹.

		Q ₂ Q ₁			
		00	01	11	10
Q ₃	0	1	0	0	1
	1	0	x	x	x

Bảng 3-54. Bảng Các nô xác định Q₁ⁿ⁺¹

		Q ₂ Q ₁			
		00	01	11	10
Q ₃	0	0	1	0	1
	1	0	x	x	x

Bảng 3-55. Bảng Các nô xác định Q₂ⁿ⁺¹

Từ bảng 3-54 tìm được Q₁ⁿ⁺¹, từ Q₁ⁿ⁺¹ tìm được J₁, K₁:

$$Q_1^{n+1} = \overline{Q_3} \cdot \overline{Q_1}$$

$$Q_1^{n+1} = J_1 \cdot \overline{Q_1} + \overline{K_1} \cdot Q_1$$

$$\rightarrow J_1 = \overline{Q_3}; \quad K_1 = 1$$

Từ bảng 3-55 tìm được Q₂ⁿ⁺¹, J₂, K₂

$$--> J_2 = K_2 = Q_1$$

Bảng 3-56. Bảng Các nơ xác định Q_3^{n+1}

$$--> J_3 = Q_2 \cdot Q_1; \quad K_3 = 1$$

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

[illegible]

163

3.10.2.3. 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 \bar{Q} (với giả thiết xung Clock tích cực tại sườn âm ↓).

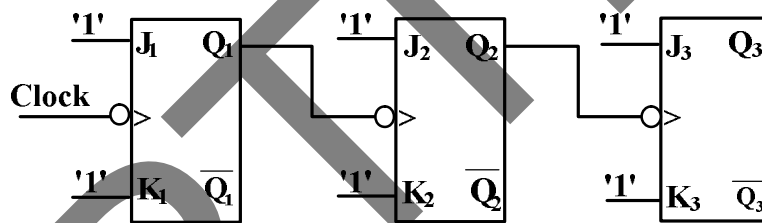
- Tín hiệu vào X_d luôn được đưa tới đầu vào xung nhịp của trigơ có trọng số nhỏ nhất.

Ví dụ đối với bộ đếm nhị phân không đồng bộ $M_d = 2^n$ dùng các trigơ $Q_1, Q_2 \dots Q_{n-1}$ với Q_1 là bit có trọng số nhỏ nhất, Q_{n-1} là bit có trọng số lớn nhất, có:

- Khi đếm tiến: $C_{Q1} = X; C_{Q2} = Q_1 \dots C_{Q_{n-1}} = C_{Q_{n-2}}$.

- Khi đếm lùi: $C_{Q1} = X; C_{Q2} = \bar{Q}_1 \dots C_{Q_{n-1}} = \bar{C}_{Q_{n-2}}$.

Sơ đồ của bộ đếm nhị phân không đồng bộ 3 bit ($M_d = 8$ - đếm tiến) dùng trigơ JK được cho ở hình 3-93.

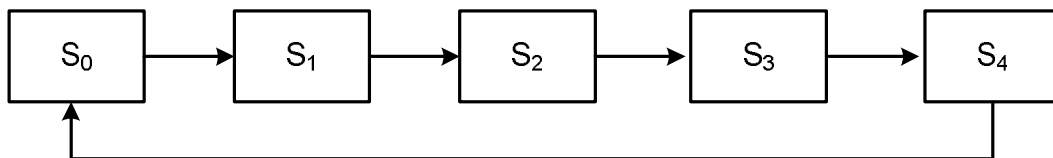


Hình 3-93. Bộ đếm nhị phân không đồng bộ 3 bit

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 3-94.



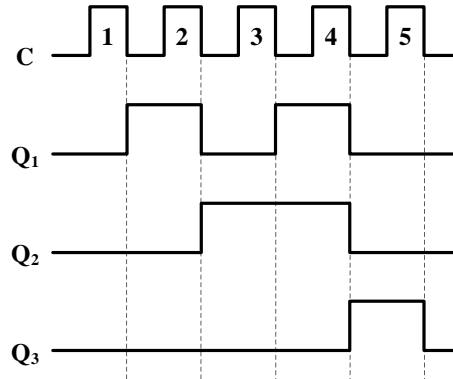
Hình 3-94. Đồ hình trạng thái bộ đếm mod 5 không đồng bộ

- 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 3-93.



Hình 3-93. 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 đều được dùng để tối thiểu hóa.

Q_2Q_1					
Q_3		00	01	11	10
	0	001	010	100	011
	1	000	x	x	x

Q_2Q_1					
Q_3		00	01	11	10
	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	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	0	0	1	0
	1	0	x	x	x

$$Q_3^{n+1} = \overline{Q_3} \cdot Q_2 \cdot Q_1$$

Bảng 3-58. Bảng chuyển đổi trạng thái và bảng tìm hàm kích

Sau khi tối thiểu hoá bằng bảng 3-58, nhận được hệ phương trình:

$$Q_1^{n+1} = \overline{Q_3} \cdot \overline{Q_1}$$

$$Q_2^{n+1} = \overline{Q_2}$$

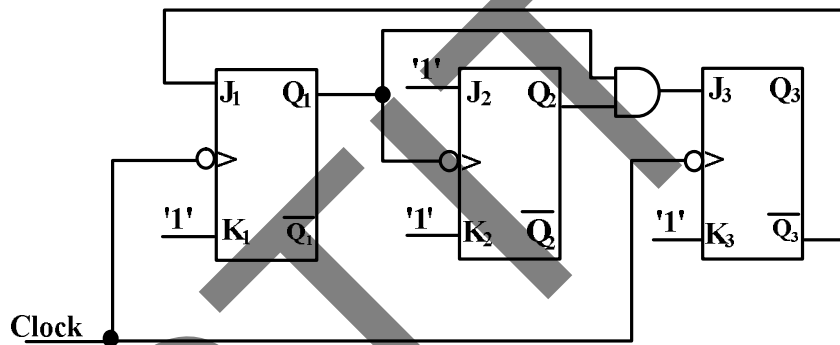
$$Q_1^{n+1} = \overline{Q_3} Q_2 Q_1$$

Kiểm tra khả năng tự khởi động bằng bảng 3-59.

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 3-59. Kiểm tra khả năng tự khởi động

Nhìn vào bảng 3-59, các trạng thái dư sau 1 số xung nhịp đều quay trở lại vòng đếm, do vậy bộ đếm này tự khởi động.



Hình 3-96. Bộ đếm Mod 5 không đồng bộ

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 3-96.

3.8.3. Thiết kế bộ đếm từ IC đếm

3.8.3.1. 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 3-60. 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 3-60. Một số loại IC thông dụng

3.8.3.2. 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
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 3-61. Một số loại IC thông dụng

Tất cả các IC giới thiệu trong bảng 3-61 đều hoạt động tại sườn âm của xung Clock. Các loại IC này đều có bốn trigơ MS.

a. IC 7490, 74390 và IC 7490

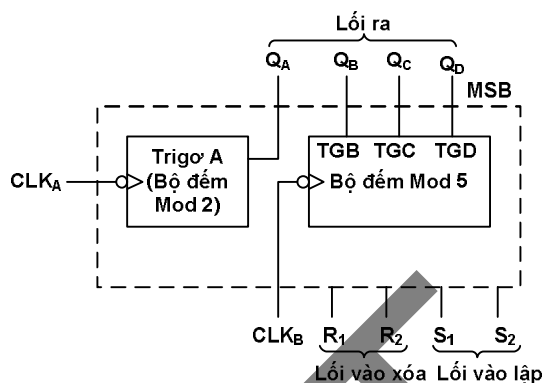
Hình 3-103 biểu thị cấu trúc cơ bản của IC 7490. Nó bao gồm 4 trigơ cung cấp bộ đếm gồm hai Mod đếm: Mod 2 và Mod 3. Các bộ đếm Mod 2 và Mod 5 có thể được sử dụng một cách độc lập. Trigơ A thực hiện đếm Mod 2, Trigơ B, C, D thực hiện đếm Mod 3.

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.



Hình 3-103. Cấu trúc cơ bản của IC 7490

Bảng 3-62 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 3-62. Bảng chức năng của IC7490

Bảng 3-63 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 3-63. Bảng chức năng của IC74290

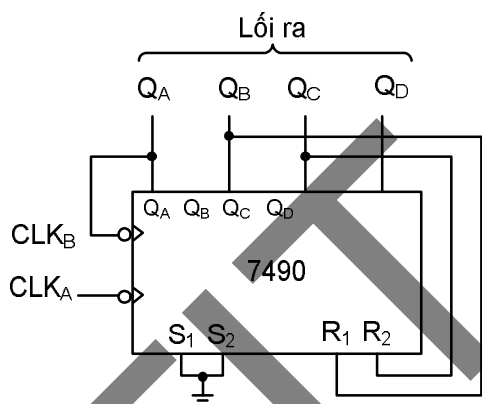
Để thực hiện đếm Mod 10 (BCD) thì đầu ra Q_A được nối với xung Clock B. Tín hiệu cần đếm được đưa vào xung clock A.

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 .

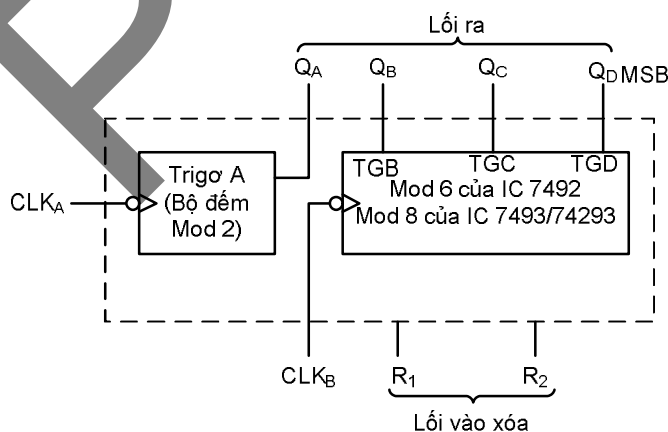
Để thực hiện được mạch chia này, phải nối bộ đếm về trạng thái chia hết cho 1010, sau đó nối đầu ra Q_B và Q_C với 2 chân Reset: R_1 và R_2 . Hình 3-104 minh họa bộ đếm này.



Hình 3-104. 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 3- 103.



Hình 3-103. 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 3-64. 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 3-64. Bảng trạng thái Mod 6 của IC

Bảng 3-65 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 3-65. 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.

3.8.4. Bộ ghi dịch (Shift Register)

3.8.4.1. 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 có bộ ghi dịch đa năng.

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 rời rạc hoặc được tích hợp trên các chip.

Bảng 3-66 giới thiệu một số IC TTL 54/74.

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

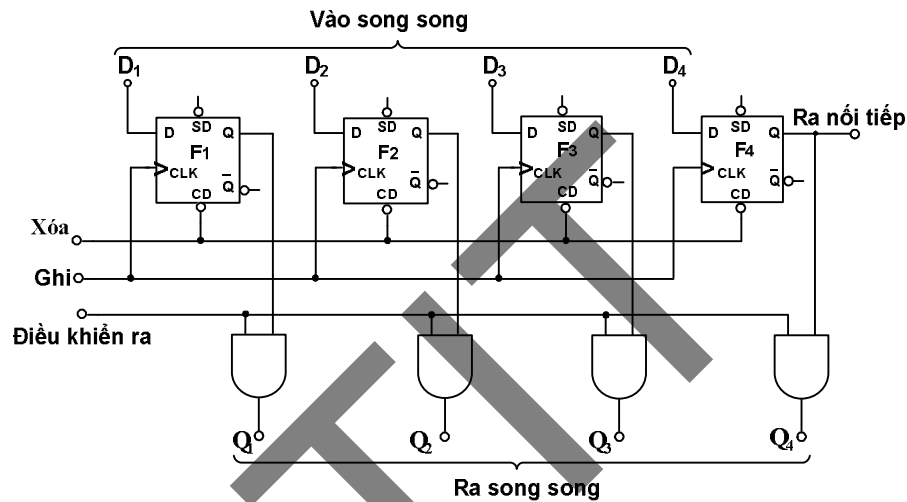
Tên IC	Mô tả
74199	8 bit - vào nối tiếp/song song, ra song song
74295A	4 bit 3 trạng thái - vào nối tiếp/song song, ra song song theo hai hướng
74395	4 bit 3 trạng thái - vào nối tiếp/song song, ra nối tiếp/song song

Bảng 3-66. Giới thiệu 1 số IC ghi dịch

3.8.4.2. Bộ ghi song song

Sơ đồ bộ ghi song song được trình bày trên hình 3-108.

Trong sơ đồ dùng 4 trigơ D và một mạch điều khiển ra dùng cổng AND 2 đầu vào.



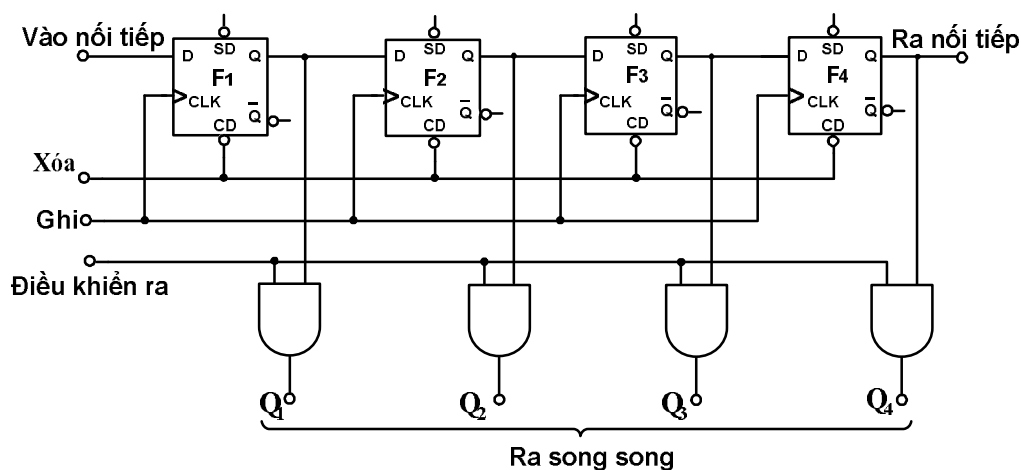
Hình 3-108. Sơ đồ logic bộ ghi song song 4 bit

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.

3.8.4.3. Bộ ghi nối tiếp

Bộ ghi dịch nối tiếp có thể dịch phải, dịch trái và cho ra song song hoặc ra nối tiếp. Hình 3-109 giới thiệu sơ đồ bộ ghi dịch vào nối tiếp dịch phải ra song song và ra nối tiếp



Hình 3-109. Bộ ghi nối tiếp dịch phải

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 3-67.

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 3-67. Bảng trạng thái của bộ ghi dịch

Giả sử muốn nhập số liệu D_1, D_2, D_3, D_4 , đây là mạch ghi dịch phải, do đó phải bit D_4 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 : $Q_1 Q_2 Q_3 Q_4 = D_1 D_2 D_3 D_4$; muốn lấy ra song song thì cho đầu “điều khiển ra” = 1.

+ Ra nối tiếp: Lấy ra ở Q_4 (trigơ cuối cùng).

Sau 4 xung nhịp đã nạp xong, D_4 xuất hiện ở đầu ra nối tiếp, vậy cần 3 xung nữa để $D_1 D_2 D_3$ được đưa ra Q_4 .

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.

3.8.4.3. 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:

$$\Delta t = N \times \frac{1}{f_c} \quad (3.57)$$

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

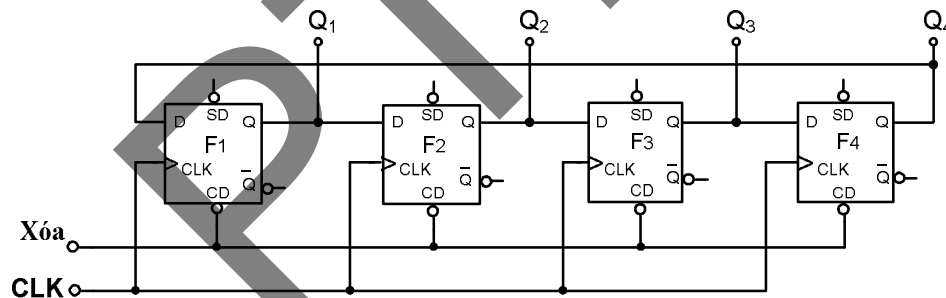
Nhờ bộ ghi dịch SIPO mà dữ liệu ở dạng nối tiếp được chuyển sang dạng song song.

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 3-112.



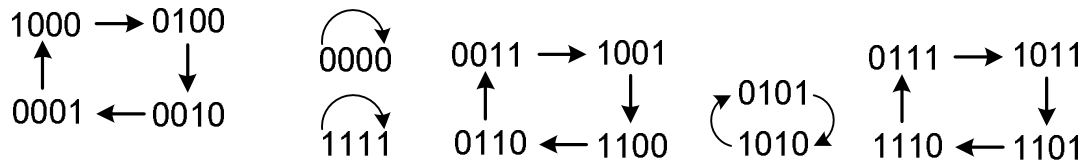
Hình 3-112. Bộ đếm vòng

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 3-113.

Trạng thái được dùng

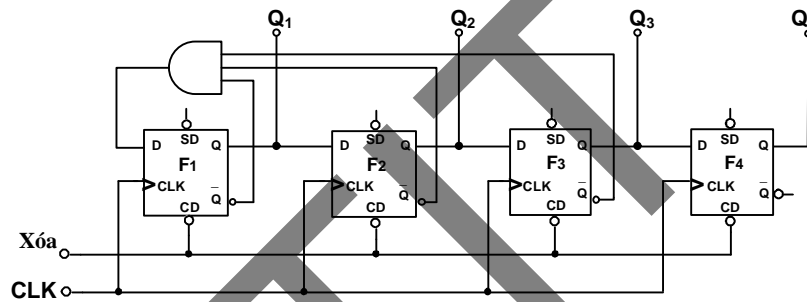
Trạng thái không được dùng



Hình 3-113. Đồ hình trạng thái bộ đếm vòng

Nhìn vào đồ hình trạng thái, 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, 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 3-114. Bộ đếm vòng 4 bit tự khởi động.

Hình 3-114 là bộ đếm vòng 4 bit tự khởi động.

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:

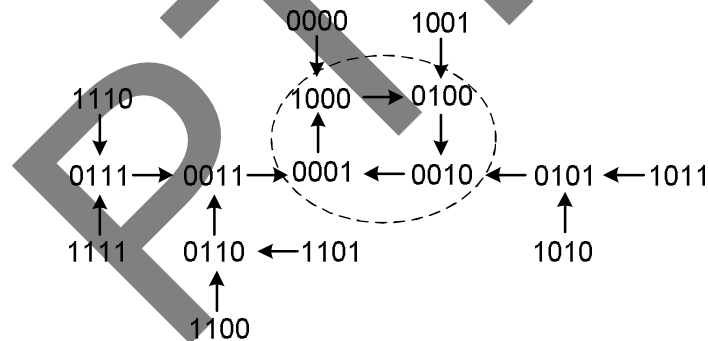
$$\left. \begin{aligned} Q_1^{n+1} &= \overline{Q_1^n} \overline{Q_2^n} \overline{Q_3^n} \\ Q_2^{n+1} &= Q_1^n \\ Q_3^{n+1} &= Q_2^n \\ Q_4^{n+1} &= Q_3^n \end{aligned} \right\}$$

Từ phương trình chuyển đổi trạng thái, lập được bảng trạng thái 3-68.

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 3-68. Bảng trạng thái bộ đếm vòng

Từ bảng trạng thái, vẽ được đồ hình trạng thái trên hình 3-113.



Hình 3-113. Đồ hình trạng thái bộ đếm vòng tự khởi động

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.

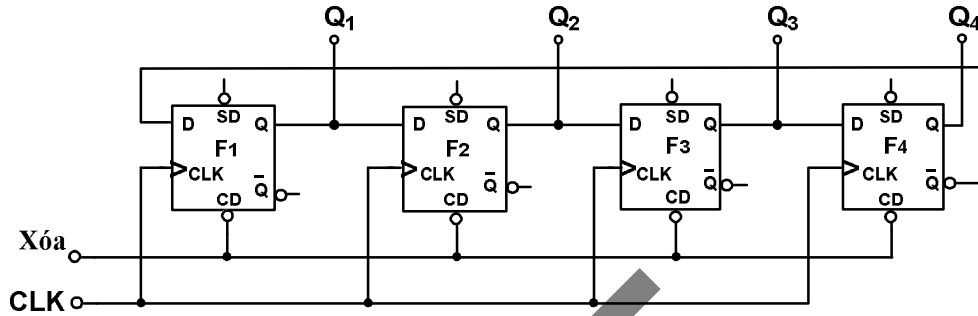
Nhược điểm: Hiệu suất sử dụng trạng thái thấp, số lượng trigơ bằng số lượng bit nhớ nên phải sử dụng nhiều trigơ.

c. 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:

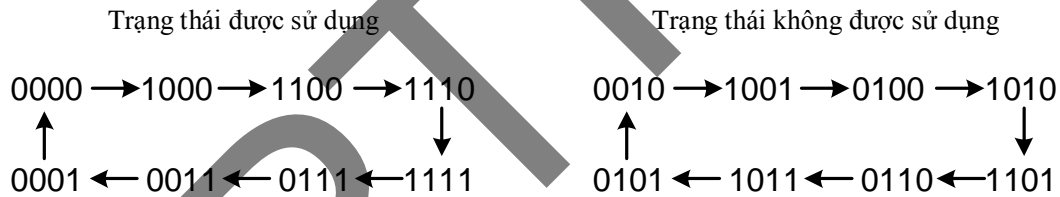
$$D_1 = \overline{Q_n}$$

Sơ đồ mạch của bộ đếm được chỉ ra trên hình 3-116.



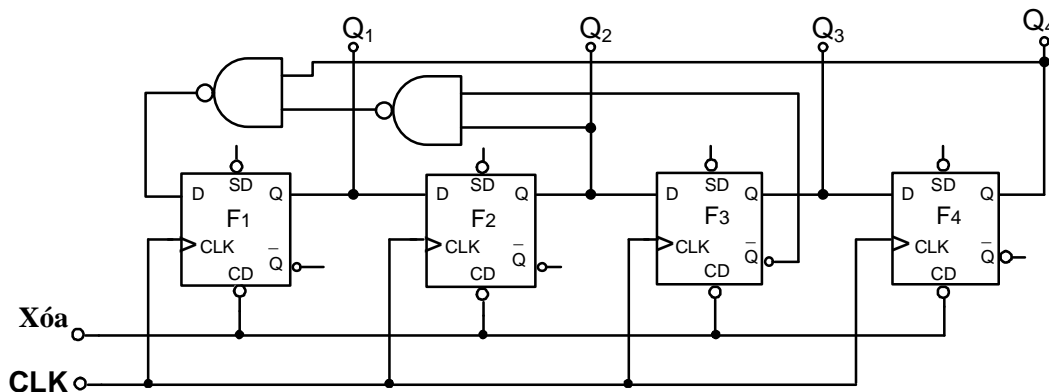
Hình 3-116. Bộ đếm vòng xoắn

Bằng cách phân tích, xây dựng được đồ hình trạng thái 3-117, 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 3-117. Đồ hình trạng thái của bộ đếm vòng xoắn

Do bộ đếm hình 3-116 không tự khởi động nên hình 3-118 giới thiệu bộ đếm vòng xoắn tự khởi động.



Hình 3-118. Bộ đếm vòng xoắn tự khởi động

Tiến hành phân tích mạch:

Phương trình kích:

$$\left. \begin{aligned} D_1 &= \overline{Q_2^n \cdot Q_3^n} \cdot Q_4^n = Q_2^n \cdot \overline{Q_3^n} + Q_4^n \\ D_2 &= Q_1^n \\ D_3 &= Q_2^n \\ D_4 &= Q_3^n \end{aligned} \right\}$$

Thay phương trình kích vào phương trình đặc trưng ($Q_i^{n+1} = D_i$) tìm được phương trình chuyển đổi trạng thái:

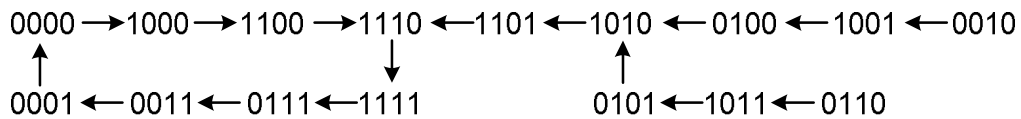
$$\left. \begin{aligned} Q_1^{n+1} &= Q_2^n \cdot \overline{Q_3^n} + Q_4^n \\ Q_2^{n+1} &= Q_1^n \\ Q_3^{n+1} &= Q_2^n \\ Q_4^{n+1} &= Q_3^n \end{aligned} \right\}$$

Từ phương trình chuyển đổi trạng thái, lập được bảng trạng thái như bảng 3-69.

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
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 3-69. Bảng trạng thái bộ đếm vòng

Từ bảng trạng thái, vẽ được đồ hình trạng thái 3-119.



Hình 3-119. Đồ hình trạng thái bộ đếm vòng xoắn tự khởi động

d. 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.

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.

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.

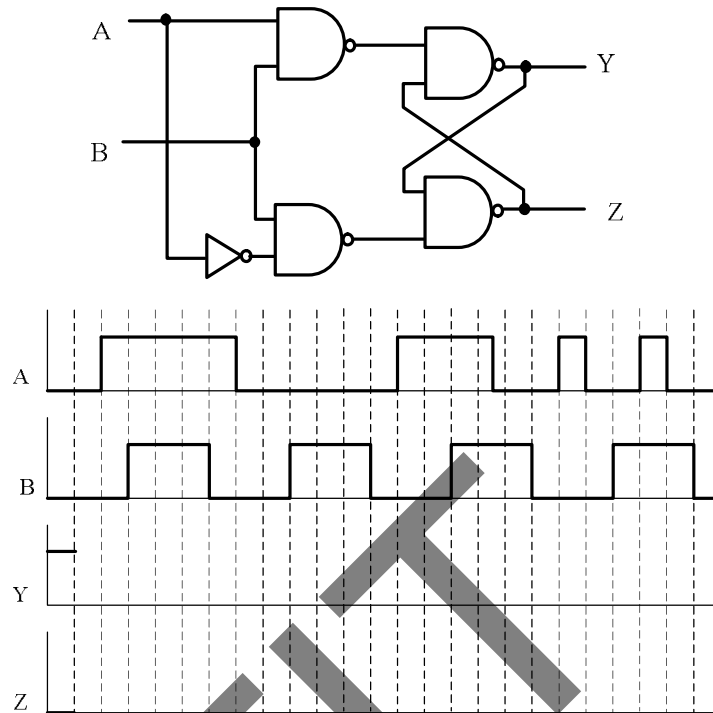
Chức năng logic và hình thức cấu trúc của trigơ là hai khái niệm khác nhau. Chức năng logic là quan hệ giữa trạng thái tiếp theo của đầu ra với trạng thái hiện tại của đầu ra và các tín hiệu đầu vào. Do chức năng logic khác nhau mà trigơ được phân thành các loại RS, D, T, JK. Còn do hình thức cấu trúc 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.

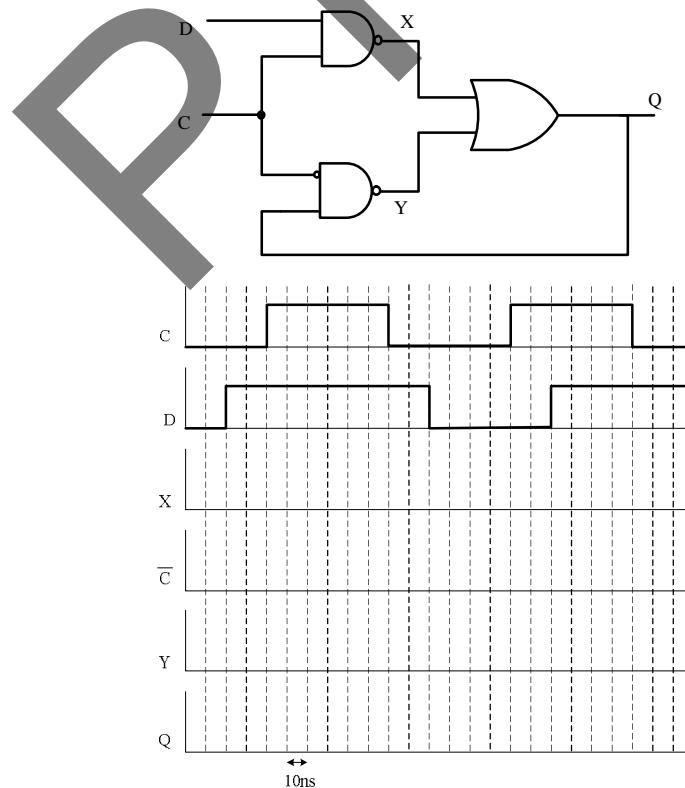
Mạch tuần tự cụ thể có rất nhiều chủng loại. Chương này chỉ giới thiệu một số loại mạch tuần tự điển hình: bộ đếm, bộ ghi dịch... Đồng thời với việc nắm vững cấu trúc, nguyên lý công tác và đặc điểm của các mạch tuần tự đó, chúng ta cũng phải nắm vững được đặc điểm chung của mạch tuần tự và phương pháp chung khi phân tích và thiết kế mạch tuần tự.

CÂU HỎI ÔN TẬP CHƯƠNG 3

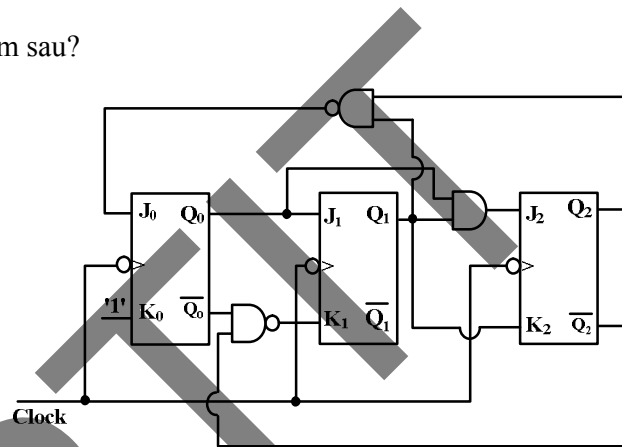
1. Mạch điện dưới đây là mạch của trigơ nào? Vẽ tiếp giản đồ xung ra của mạch?



2. 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).

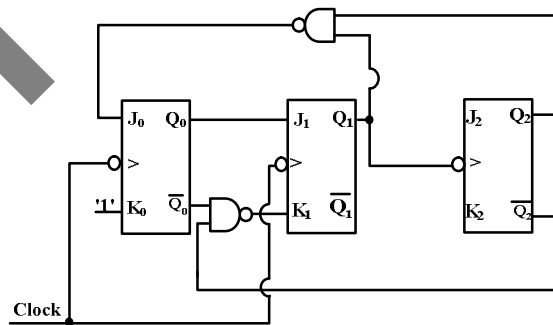


3. Một trigơ JK được ở chế độ lật. Nếu tần số Clock của nó là 1000 hz thì tần số tại đầu ra là bao nhiêu?
4. Một bộ đếm nhị phân 4 bit thì tần số tại đầu ra của bit có trọng số lớn nhất so với tần số xung nhịp là bao nhiêu?
5. Thiết kế bộ đếm thuận mã Gray 3 bit .
6. Thiết kế bộ đếm nghịch Mod 6.
7. 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

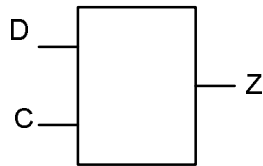
12. Cho bộ đếm sau. Cho biết đây là bộ đếm Mod mấy?



Hình vẽ bài 12

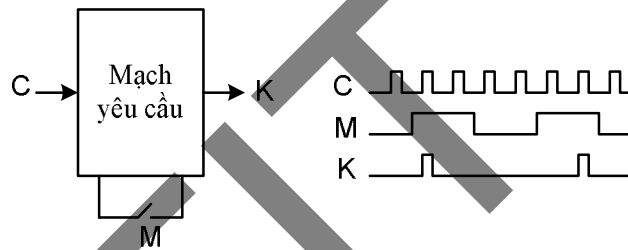
13. Dùng trigơ để thiết kế mạch kiểm tra dãy xung với giả thiết:

Dữ liệu nhị phân được đưa vào đầu D, mỗi bit đồng bộ với một xung đồng hồ trên đường C. Tín hiệu được đưa ra đầu Z khi mỗi lần chuỗi bit “1001” xuất hiện.



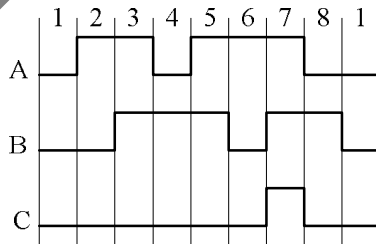
Hình vẽ bài 13

14. Thiết kế bộ đếm không đồng bộ $M = 9$?
15. Thiết kế bộ đếm đồng bộ $M = 12$?
16. Thiết kế bộ đếm nghịch thập phân đồng bộ?
17. Cho một dãy xung Clock có tần số cao đưa vào chân C và một công tắc M cho phép xung ra tại đầu ra K. Giả sử rằng có ít nhất một xung sẽ được đưa ra trong thời gian công tắc M hoạt động. Hãy thiết kế mạch theo nguyên lý hoạt động trên sơ đồ khối và giản đồ xung được cho ở hình vẽ sau:



Hình vẽ bài 17

18. Tín hiệu dưới dạng mã BCD được đưa liên tiếp trên đầu vào X (X là bit có trọng số lớn nhất). Giá trị thập phân của chữ số BCD đó được biểu diễn bằng n . Mỗi bit thông tin được đồng bộ với xung nhịp. Dùng trigơ để thiết kế mạch phát hiện sai khi mã trên đầu vào không phải mã BCD.
19. Dùng trigơ RS để thiết kế mạch tuần tự không đồng bộ hoạt động theo giản đồ xung sau:



Hình vẽ bài 19

20. Một phòng có hai cửa và một bóng đèn D. Hai công tắc X và Y được gắn trên tường gần mỗi cửa. Thiết kế mạch điều khiển sao cho có thể bật đèn khi vào phòng và tắt đèn khi ra khỏi phòng bởi các công tắc X và Y.

CHƯƠNG 4: BỘ NHỚ BÁN DẪN

GIỚI THIỆU

Bộ nhớ bán dẫn thay thế các loại bộ nhớ bằng vật liệu từ. Các tiến bộ mới của công nghệ bán dẫn trong thời gian gần đây đã cung cấp nhiều mạch nhớ loại MSI và LSI có độ tin cậy cao và giá thành hạ. Vào đầu thập kỷ 60 của thế kỷ 20, giá thành thương phẩm của một bit nhớ vào khoảng 2 USD. Đến nay (những năm đầu thế kỷ 21), giá thương phẩm của 128 Mbyte vào khoảng 20 USD. Như vậy giá thành thương phẩm của một bit nhớ sau khoảng 40 năm đã giảm đi khoảng 105.10^6 lần. Bộ nhớ bán dẫn điển hình có các tế bào nhớ sắp xếp theo hình chữ nhật, gắn trong khối hộp nhỏ bằng nhựa dạng DIP (Dual in line package). Tế bào nhớ cơ bản là một mạch trigơ, transistor hay mạch có khả năng tích trữ điện tích, tế bào nhớ này dùng để lưu trữ một bit tin.

Trong phần này giới thiệu một số bộ nhớ bán dẫn cơ bản.

4.1. KHÁI NIỆM CHUNG

4.1.1. Khái niệm

Bộ nhớ là một thiết bị có khả năng lưu trữ thông tin (nhị phân). Muốn sử dụng bộ nhớ, trước tiên, phải viết dữ liệu và các thông tin cần thiết vào nó, sau đó lúc cần thiết phải lấy dữ liệu đã viết trước đó để sử dụng. Thủ tục viết vào và đọc ra phải được kiểm soát chặt chẽ, tránh nhầm lẫn nhờ định vị chính xác từng vị trí ô nhớ và nội dung của nó theo một mã địa chỉ duy nhất.

4.1.2. Những đặc trưng chính của bộ nhớ

4.1.2.1. Dung lượng của bộ nhớ.

Dung lượng bộ nhớ là số bit thông tin tối đa có thể lưu giữ trong nó. Dung lượng cũng có thể biểu thị bằng số từ nhớ n bit. **Từ nhớ n bit** là số bit (n) thông tin mà có thể đọc hoặc viết đồng thời vào bộ nhớ. Ví dụ: Một bộ nhớ có dung lượng là 256 bit; nếu nó có cấu trúc để có thể truy cập cùng một lúc 8 bit thông tin, thì cũng có thể biểu thị dung lượng bộ nhớ là 32 từ nhớ $\times 8$ bit = 32 byte.

4.1.2.2. Cách truy cập thông tin.

Các bộ nhớ có thể có một trong hai cách truy cập thông tin.

Truy cập trực tiếp, hay còn gọi là truy cập ngẫu nhiên (random access). Ở cách này, không gian bộ nhớ được chia thành nhiều ô nhớ. Mỗi ô nhớ chứa được 1 từ nhớ n bit và có một địa chỉ xác định, mã hoá bằng số nhị phân k bit. Như vậy, người sử dụng có thể truy cập trực tiếp thông tin ở ô nhớ có địa chỉ nào đó trong bộ nhớ. Mỗi bộ nhớ có k bit địa chỉ sẽ có 2^k ô nhớ và có thể viết được 2^k từ nhớ n bit.

Truy cập liên tiếp (serial access) hay còn gọi là kiểu truy cập tuần tự. Các đĩa từ, băng từ, trống từ, thanh ghi dịch... có kiểu truy cập này. Các bit thông tin được đưa vào và lấy ra một cách tuần tự.

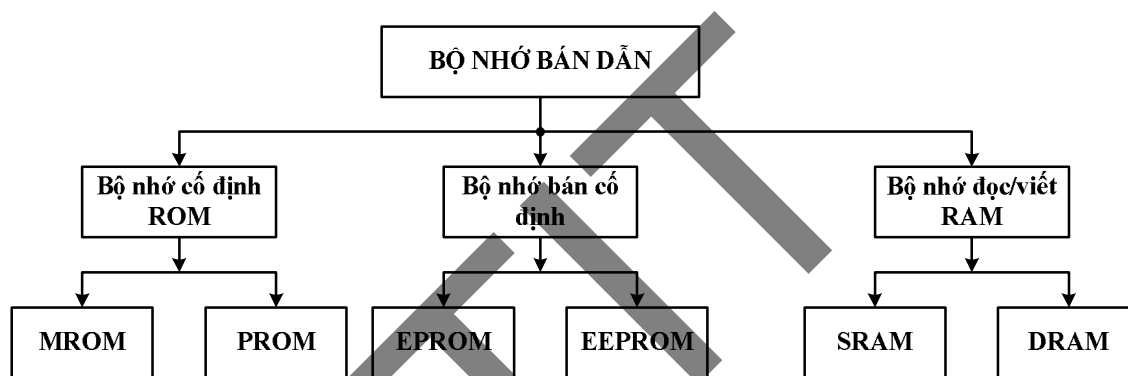
4.1.2.3. Tốc độ truy cập thông tin.

Đây là thông số rất quan trọng của bộ nhớ. Nó được đặc trưng bởi thời gian cần thiết để truy cập thông tin.

Thời gian truy cập thông tin ở các bộ nhớ truy cập kiểu trực tiếp gồm thời gian tìm địa chỉ của ô nhớ và thời gian đọc/viết thông tin trên đó. Thời gian truy cập thông tin phụ thuộc chủ yếu vào công nghệ chế tạo. Với công nghệ MOS thì thời gian truy cập khoảng 30 đến vài trăm ns.

Ở các bộ nhớ truy cập kiểu tuần tự, thời gian truy cập phụ thuộc vào vị trí của thông tin cần truy cập trong tập tin (file). Đối với các băng từ, đĩa từ thời gian truy cập của nó được định nghĩa là thời gian trung bình hoặc cực đại để truy cập một thông tin và nằm trong khoảng vài msec đến nhiều sec.

4.1.3. Phân loại



Hình 4-1. Sơ đồ phân loại bộ nhớ bán dẫn

Dựa trên thời gian viết và cách viết, có thể chia thành bộ nhớ cố định, bộ nhớ bán cố định và bộ nhớ đọc/viết được. Bộ nhớ có nội dung được viết sẵn một lần khi chế tạo được gọi là bộ nhớ cố định và được ký hiệu là ROM (Read Only Memory). Sau khi đã được viết (bằng mặt nạ-mask) từ nhà máy thì ROM loại này không viết lại được nữa đó chính là MROM. PROM là một dạng khác, các bit có thể được viết bằng thiết bị viết của người sử dụng trong một lần (Programmable ROM).

Bộ nhớ có thể đọc/ viết nhiều lần được gọi là RAM (Random Access Memory) gồm hai loại: bộ nhớ RAM tĩnh-SRAM (Static RAM) thường được xây dựng trên các mạch điện tử trigơ và RAM động-DRAM (Dynamic RAM) được xây dựng trên cơ sở nhớ các điện tích ở tụ điện; bộ nhớ này phải được hồi phục nội dung đều đặn, nếu không nội dung sẽ mất đi theo sự rò điện tích trên tụ. Giữa ROM và RAM có một lớp các bộ nhớ được gọi là EPROM (Erasable PROM), dữ liệu trong đó có thể xoá được bằng tia cực tím và viết lại được, EEPROM (Electric EPROM) có thể xoá được bằng dòng điện. Các loại này còn được gọi là bộ nhớ bán cố định. Các bộ nhớ DRAM thường thoả mãn những yêu cầu khi cần bộ nhớ có dung lượng lớn; trong khi đó khi cần có tốc độ truy xuất lớn thì phải dùng các bộ nhớ SRAM có giá thành đắt hơn. Nhưng cả hai loại này đều có nhược điểm là thuộc loại “bay hơi” (volatile), thông tin

sẽ bị mất đi khi nguồn nuôi bị ngắt. Do vậy, các chương trình dùng cho việc khởi động PC như BIOS thường phải nạp trên các bộ nhớ ROM.

4.1.4. Tổ chức của bộ nhớ

Bộ nhớ thường được tổ chức gồm nhiều vi mạch nhớ được ghép lại để có độ dài từ và tổng số từ cần thiết. Những chip nhớ được thiết kế sao cho có đầy đủ một số chức năng của bộ nhớ như:

Một ma trận nhớ gồm các ô nhớ, mỗi ô nhớ ứng với một bit nhớ.

Mạch logic giải mã địa chỉ ô nhớ.

Mạch logic cho phép đọc nội dung ô nhớ.

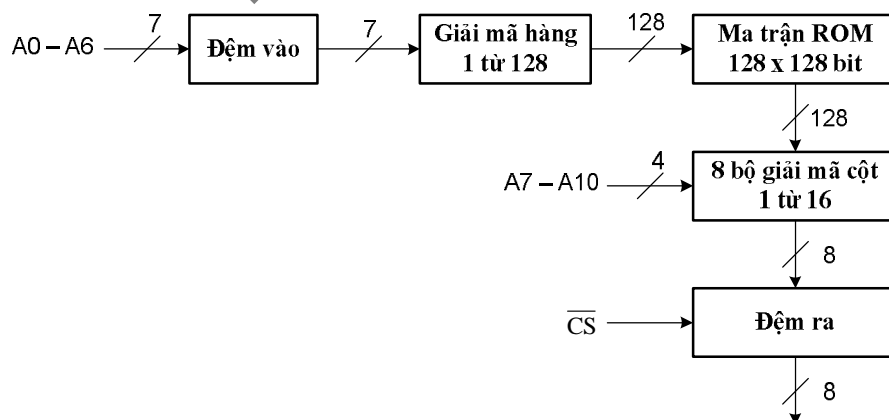
Mạch logic cho phép viết nội dung ô nhớ.

Các bộ đệm vào, bộ đệm ra và bộ mở rộng địa chỉ.

Cách tổ chức đơn giản nhất là tổ chức theo từ (word organized) với sự chọn tuyến tính. Một ma trận nhớ như vậy có độ dài của cột bằng số lượng từ (W) và độ dài của hàng bằng số lượng bit (B) trong một từ. Bộ chọn từ phải giải mã 1 từ (W), nghĩa là giải mã để có một đầu ra duy nhất cho một từ trong bộ nhớ. Phương pháp này có thời gian truy nhập ngắn nhưng cần một bộ giải mã lớn khi tổng số từ lớn, do đó làm tăng giá thành sản phẩm.

Kích thước của phần giải mã địa chỉ sẽ giảm đi khi tổ chức ma trận nhớ và phần logic chọn từ cho phép giải mã hai bước. Ma trận nhớ sử dụng giải mã hai bước ứng với từ vật lý và từ logic. Từ vật lý bao gồm số lượng bit trong một hàng của ma trận. Từ logic bao gồm số lượng bit tương ứng với một từ logic được nhận biết và gửi ra cùng một lúc. Cần hai bộ giải mã: một bộ giải mã hàng để chọn một từ vật lý và một bộ giải mã cột gồm có một vài mạch hợp kênh chọn một từ logic từ một từ vật lý đã chọn. Một từ vật lý được chia thành S từ logic. Bộ giải mã hàng là bộ giải mã chọn 1 từ W mà $B = W/S$ và bộ chọn cột chứa B bộ hợp kênh một đường từ S.

Ví dụ: sơ đồ ROM dung lượng 2048 x 8 (2048 từ, mỗi từ chứa 8 bit) tổ chức giải mã hai bước như hình 4- 2.



Hình 4-2. Ví dụ về bộ giải mã cho ma trận ROM 128 x 128

Ma trận nhớ là 128×128 , như vậy có $128 = 2^7$ từ vật lý. Một từ vật lý được chọn bởi 7 đường địa chỉ từ A0 đến A6. Bộ giải mã hàng chọn 1 hàng từ 128 hàng. Một từ vật lý được chia thành $128/8 = 16$ nhóm 8 bit. Nhóm thứ nhất chứa những bit có trọng số cao nhất của 16 từ logic. Nhóm thứ hai chứa các bit cao tiếp theo của 16 từ logic...Nhóm cuối cùng chứa những bit thấp nhất của 16 từ logic, do đó $S = 16$. Như vậy, những bộ giải mã cột gồm 8 bộ hợp kênh một đường từ 16 đường để cung cấp một từ logic ra 8 bit. Những địa chỉ từ A7 đến A10 điều khiển các bộ giải mã cột. Trường hợp đặc biệt khi số phần tử trong một từ vật lý bằng số bit trong một từ vật lý thì đó là bộ nhớ tổ chức theo bit có nghĩa là mỗi từ logic có độ dài 1 bit.

Các bộ đệm ra đảm bảo các mức logic mong muốn và cung cấp đủ dòng điện, ngoài ra nó còn có đầu ra collector hở hoặc 3 trạng thái cho phép nối chung đầu ra của một vài chip với nhau. Bộ đệm ra được điều khiển bởi một hay nhiều đầu vào như chọn mạch CS (Chip Select), cho phép mở CE (Chip Enable) hay cho phép mở đầu ba trạng thái OE (Output Enable).

4.2. BỘ NHỚ CỔ ĐỊNH - ROM

Để một máy tính điện tử tiến hành các hoạt động bình thường dù đơn giản hay phức tạp thì cần phải cung cấp cho máy trước tiên là các lệnh từng bước tiến hành. Tập hợp danh sách các lệnh được gọi là chương trình. Việc nhập một chương trình nào đó từ đĩa hay từ bàn phím vào máy tính phải tuân theo sự điều khiển của một chương trình đặc biệt : chương trình khởi động thường trú trong máy. Chương trình này được cài sẵn ở một bộ nhớ chỉ đọc (ROM). Bộ nhớ ROM có những thuộc tính sau:

- + Chỉ cho phép đọc nội dung đã được viết sẵn từ trước chứa trong nó. Nội dung này do người điều hành thiết kế lập trình sẵn và viết vào nó bằng một phương pháp đặc biệt.

- + Người sử dụng về nguyên tắc là không thể hoặc rất khó thay đổi nội dung thông tin đã viết nhớ trong ROM.

- + Nội dung được viết trong ROM có tính chất cố định, không bị mất đi theo thời gian hay do mất nguồn năng lượng cung cấp cho toàn bộ hệ thống số trong đó có ROM.

Do những thuộc tính trên nên ROM được dùng khi:

- + Chứa các chương trình vận hành máy tính, chương trình hỗ trợ việc lập trình hay các chương trình giám sát việc điều hành theo các bước nghiêm ngặt cố định.

- + Các chương trình cài đặt cho máy tính chuyên dụng chỉ thực hiện một nhóm các công việc giới hạn, được lập đi, lập lại có quy luật do công nghệ tự động hóa đòi hỏi (ví dụ: tự động điều khiển động cơ cho ô tô...).

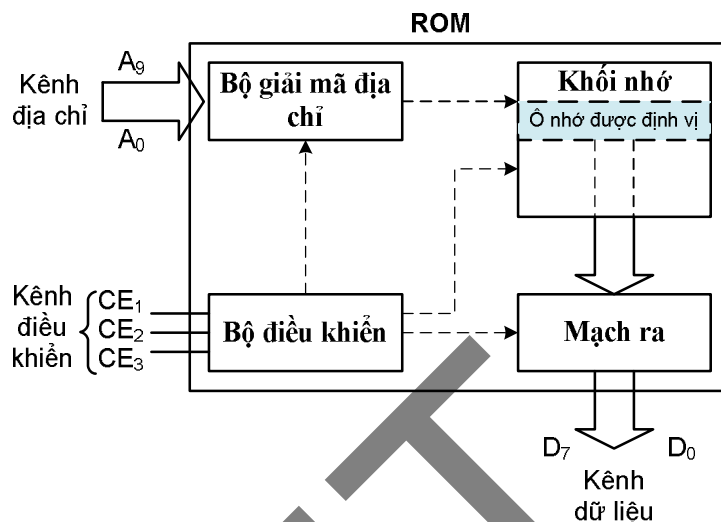
- + Lưu giữ các bảng biểu quan trọng được sử dụng thường xuyên như công cụ cho việc tính toán, xử lý của máy tính và các thiết bị liên quan.

4.2.1. Cấu trúc chung của ROM

ROM bao gồm 4 khối cơ bản:

- + Bộ nhớ chứa các ô nhớ và trong các ô nhớ là các từ nhớ.

- + Mạch điều khiển tiếp nhận các tín hiệu vào từ kênh điều khiển.
- + Bộ giải mã địa chỉ dùng để định vị ô nhớ.
- + Mạch ra dùng để đưa nội dung ô nhớ tới các thiết bị có liên quan cần tiếp nhận nội dung này.



Hình 4-3. Cấu trúc cơ bản của bộ nhớ ROM

4.2.1.1. Khối nhớ

Mỗi ô nhớ nhị phân có chức năng lưu giữ một trong hai trạng thái 0 hoặc 1. Tập hợp một nhóm các ô nhớ xếp theo hàng tạo thành một vị trí nhớ có khả năng lưu giữ một từ nhị phân 8 bit hoặc 16 bit... Kiểu sắp xếp này được gọi là cấu trúc bộ nhớ kiểu mảng (array) tuyến tính. Hình 4-4 biểu diễn một cấu trúc bộ nhớ kiểu mảng tuyến tính với dung lượng 4 x 3.

Trình tự đọc một từ nhớ như sau:

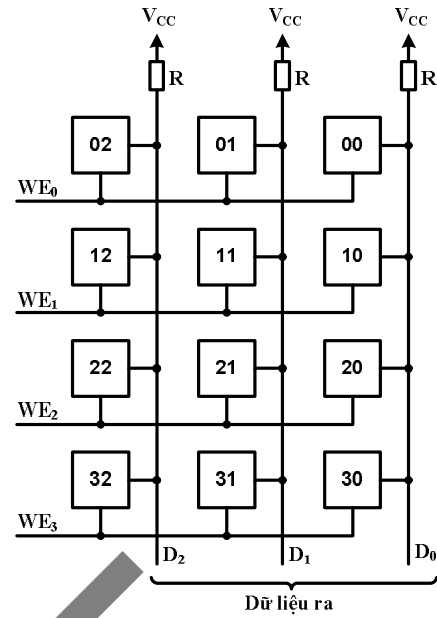
- + Kích hoạt đầu vào điều khiển WE (Word Enable) (tích cực ở mức thấp hay cao tùy theo cấu tạo của ô nhớ). Đây chính là đầu giải mã địa chỉ.

- + Các ô nhớ phải có đầu ra thuộc loại Collector hở hoặc đầu ra 3 trạng thái. Tất cả các ô nhớ trong một cột xác định có đầu ra nối chung với một đầu ra dữ liệu. Trong ví dụ này từ nhớ chỉ có 3 bit nên có 3 đầu ra $D_2D_1D_0$ tương ứng với 3 cột, 4 hàng tương ứng 4 từ được điều khiển theo các xung WE_0 WE_1 WE_2 và WE_3 .

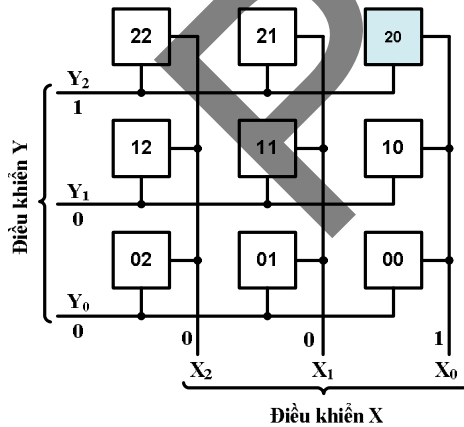
Điều này có nghĩa là mỗi một từ nhớ phải có một đầu điều khiển WE riêng. Đây chính là hạn chế của cấu trúc mảng tuyến tính vì khi số lượng địa chỉ tăng lên với các bộ nhớ dung lượng lớn thì cấu trúc bộ giải mã địa chỉ sẽ rất phức tạp.

Dạng cấu trúc thứ hai của bộ nhớ là tổ chức theo kiểu ma trận, thích hợp với các bộ nhớ có dung lượng lớn (hình 4-5).

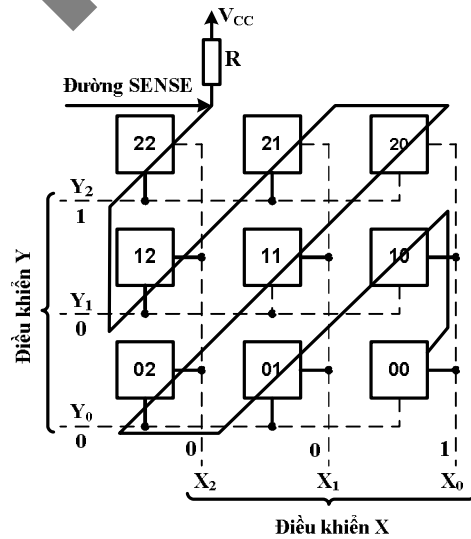
Hình 4-5a là kết cấu mảng ma trận (3 x 3) các ô nhớ. Ví dụ này đủ tổng quát cho một ma trận n hàng có ký hiệu từ Y_0 đến Y_{n-1} , m cột có ký hiệu từ X_0 đến X_{m-1} . Vậy bộ nhớ này có dung lượng là $(m \times n)$ bit nhớ. Mỗi ô nhớ của ma trận có hai đầu vào cho phép. Khi chúng đồng thời ở trạng thái tích cực thì ô nhớ mới xuất (đọc) được một bit lưu trong nó. Ví dụ để kích hoạt ô nhớ có ký hiệu là 20 thì các đầu X_0 và Y_2 phải ở trạng thái tích cực, ví dụ là tích cực cao ($X_0 = Y_2 = 1$), lúc này giá trị trên đầu cho phép phải là 2 tổ hợp: $X_2X_1X_0 = 001$ và $Y_2Y_1Y_0 = 100$. Với hai tổ hợp cho phép hàng Y_i và cột X_j này chỉ cho một ô nhớ duy nhất có địa chỉ (i, j) được chuyển lên trạng thái tích cực.



Hình 4-4. Cấu trúc bộ nhớ kiểu mảng tuyến tính



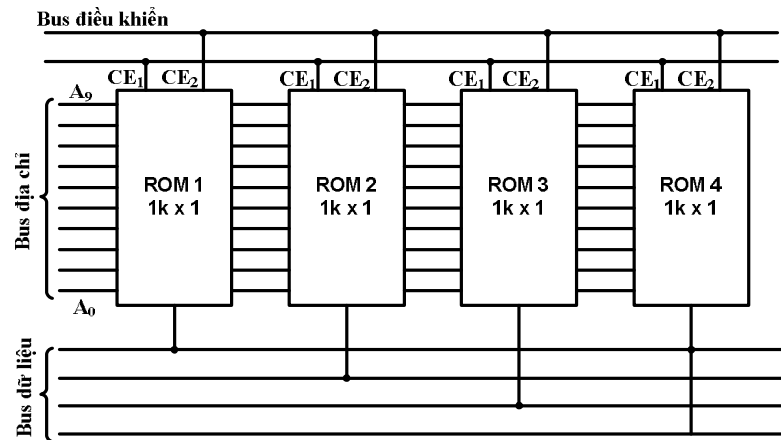
Hình 4-5a. Cấu trúc bộ nhớ kiểu ma trận hướng bit



Hình 4-5b. Tín hiệu điều khiển SENSE tìm đến từng điểm nút ma trận là một ô nhớ đơn lẻ

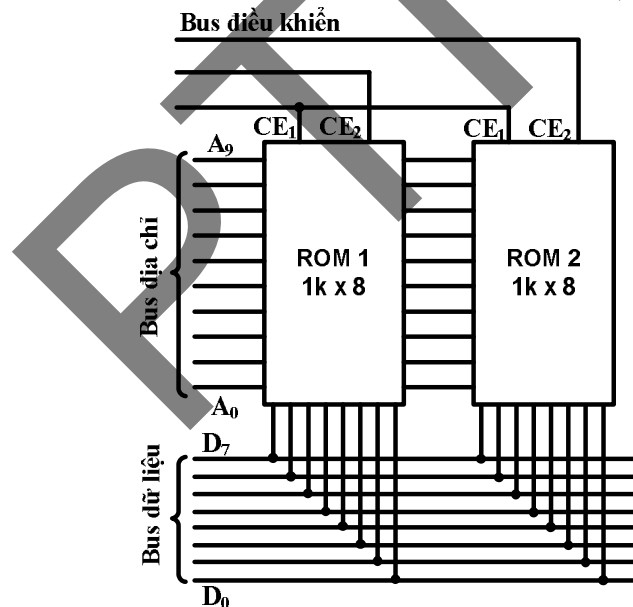
Hình 4-5b mô tả đường vẽ cắt mặt của tất cả các ô nhớ gọi là đường SENSE nhấn mạnh tới việc chỉ một dây trên được sử dụng. Dây này dùng để dò tới ô tích cực theo hướng bit và dữ liệu chứa trong một ô nhớ được sẽ được đọc ra. Tức là ô nhớ tích cực chỉ nằm trên dây

SENSE. Mỗi một ô nhớ phải là cổng 3 trạng thái hở collector (trạng thái 0, trạng thái 1 và trạng thái trở kháng cao).



Hình 4-6a. Mở rộng độ dài từ cho ROM từ 1k x 1 thành 1k x 4

Dung lượng của bộ nhớ trên hình 4-5b là 9 x 1 (nghĩa là có 9 từ nhớ, mỗi từ dài 1 bit). Muốn tạo ra một từ 4 bit, phải nối 4 mảng ma trận như vậy song song với nhau như hình 4-6a. Ở đây, mỗi mảng ma trận chứa 1024 ô nhớ, nên dung lượng tổng cộng là (1k x 4) bit. Hình 4-6b minh họa cách mở rộng dung lượng nhớ từ (1k x 8) bit thành (2k x 8) bit.

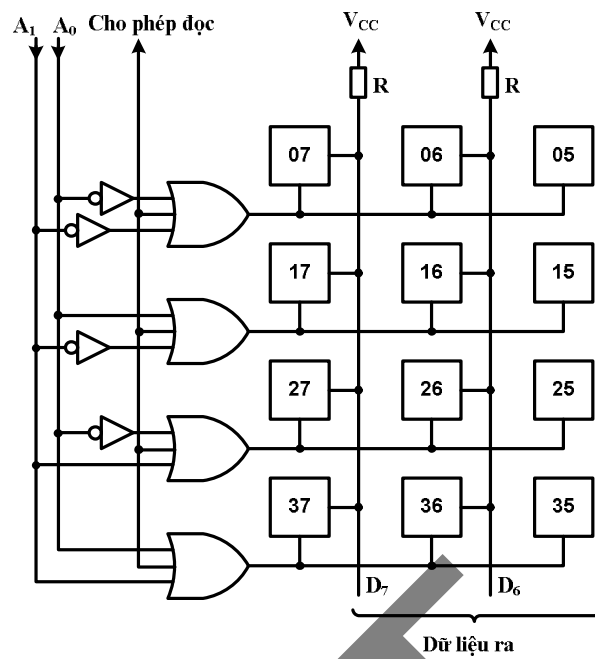


Hình 4-6b. Mở rộng dung lượng bộ nhớ ROM từ 1k x 8 thành 2k x 8

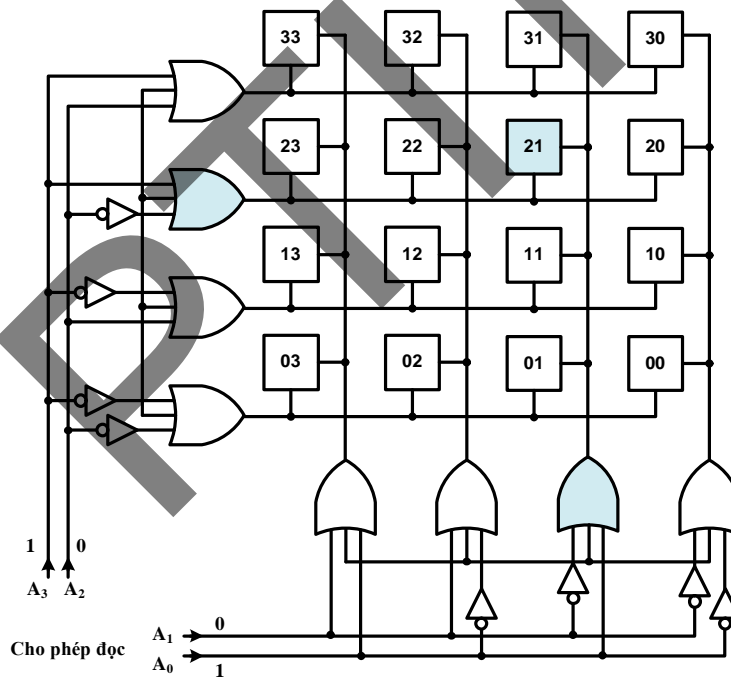
4.2.1.2. Bộ giải mã địa chỉ

Bộ giải mã địa chỉ là giao diện giữa kênh địa chỉ và khối nhớ. Nó có khả năng truyền rất nhiều địa chỉ trên một số ít đường truyền. Địa chỉ nhị phân phải được giải mã trước khi tác động tới mảng ô nhớ.

+ Trường hợp cấu trúc bộ nhớ dạng mảng tuyến tính:



Hình 4-7. Bộ giải mã ROM 2 vào 8 ra cho 32 ô nhớ (4 hàng x 8 cột)



Hình 4-8. Bộ giải mã địa chỉ cho mảng ma trận ô nhớ (4 x 4t) bit

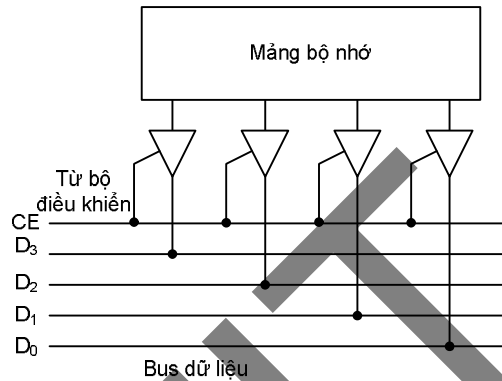
Ví dụ, với bộ giải mã địa chỉ 2 vào – 4 ra thì A_1, A_0 là các đầu vào địa chỉ, còn 4 đầu ra của bộ giải mã chính là các đầu cho phép WE (hình 4-7). Tức là ứng với một tổ hợp mã A_1A_0 thì chỉ có một đầu ra WE ở trạng thái tích cực.

+ Trường hợp bộ nhớ có cấu trúc dạng ma trận thì việc tổ chức mạch giải mã địa chỉ theo hàng và cột của ma trận đơn giản hơn nhiều. Ví dụ với bộ giải mã địa chỉ 4 vào – 16 ra : với việc chọn 2 bit địa chỉ A_1A_0 cho địa chỉ cột của ma trận và 2 bit còn lại A_3A_2 cho địa chỉ

hàng của ma trận. Như vậy chỉ cần 8 cổng NOT đã tạo ra được một địa chỉ hàng i cột j xác định tương ứng với một ô nhớ trong ma trận (hình 4-7). Ví dụ $A_3A_2A_1A_0 = 1001$, khi đó vị trí hàng 10, cột 01 (hàng 2 cột 1 trong biểu diễn thập phân của i, j), tức là ô 21 được chọn (chuyển lên trạng thái tích cực).

4.2.1.3. Mạch ra của bộ nhớ

Mạch ra có nhiệm vụ kết nối dữ liệu đã chọn với kênh dữ liệu vào lúc thích hợp. Trên hình 4-9, bộ đệm 3 trạng thái được sử dụng với mục đích ngắt kết nối giữa bộ nhớ với kênh dữ liệu khi bộ nhớ ở trạng thái không tích cực (tức là không làm việc), các cổng 3 trạng thái phải có hệ số tải cao đủ để ghép kênh dữ liệu với vô số tải khác cùng nối với kênh.



Hình 4-9. Cổng 3 trạng thái dùng làm mạch đầu ra của ROM. Tín hiệu CE từ bộ điều khiển cho phép hay không cho phép kết nối bộ nhớ với Bus dữ liệu

4.2.1.4. Mạch điều khiển trong ROM

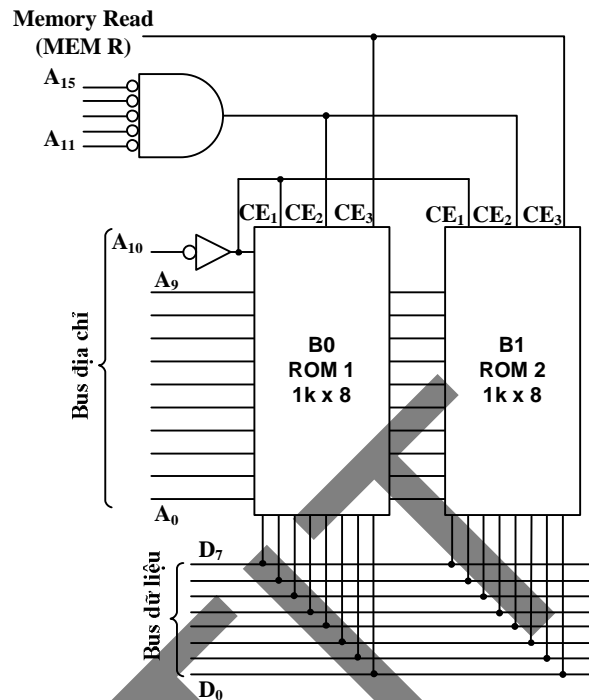
Mạch điều khiển trong ROM có chức năng khá đơn giản. Ví dụ hình 4-3 là cấu trúc bộ nhớ có 3 đầu điều khiển CE_1 , CE_2 và CE_3 (Chip Enable) hay còn gọi là CS (Chip Select). Khi cả 3 tín hiệu này đều tích cực thì ROM mới được phép hoạt động (cần một cổng AND hay cổng NAND 3 đầu vào tùy tính chất từng mạch). Tín hiệu CE này có thể thực hiện các nhiệm vụ sau:

- + Chọn chip cần đọc (khi bộ nhớ có nhiều chip ghép song song).
- + Định giờ cho phép kết nối ROM với kênh dữ liệu.
- + Tham gia vào tiến trình giải mã địa chỉ để mở rộng khả năng kiểm soát và chọn địa chỉ cho chính xác.

Ví dụ hình 4- 10 minh họa 3 chức năng điều khiển mà tín hiệu CE tham gia:

Bộ nhớ có chứa 2 chip nhớ loại 1k. Bus địa chỉ của máy tính có 16 bit ($A_0 \div A_{15}$) nên chúng dễ dàng xác định được 2k ô nhớ, các bit địa chỉ từ A_0 đến A_9 đã được nối song song và được giải mã đồng thời ở bên trong hai chip này. Tuy nhiên, chỉ có chip nào có cả ba đầu CE ở trạng thái tích cực thì mới cho phép xuất dữ liệu ra (ví dụ trong hình 4-9 bit A_{10} sẽ quyết định chip nào sẽ ở trạng thái tích cực). Chip B0 có khối địa chỉ nhị phân là 11 bit, từ 0 00000 00000 đến 0 11111 11111, chip B1 sẽ có các mã địa chỉ ô nhớ từ 1 00000 00000 đến 1 11111 11111. Hệ thống địa chỉ này có được cho hai chip B0 và B1 với điều kiện các bit địa chỉ có

trọng số cao hơn A_{10} (ví như $A_{11} \div A_{15}$) phải ở trạng thái không tích cực. Trong hình 4-10 cổng NOR đảm nhận nhiệm vụ này. CE_2 chỉ tích cực khi các địa chỉ thừa này bằng 0, còn nếu các địa chỉ này ở trạng thái tích cực thì CE_2 sẽ không tích cực, nghĩa là chip B0 và B1 sẽ không được chọn để tham gia quá trình xử lý.



Hình 4-10. Dùng đầu vào điều khiển CE để chọn chip

Với 16 bit địa chỉ thì có tối đa $2^6 \cdot 2^{10} = 64k$ hay 64 chip 1k được gán địa chỉ. Các bit từ A_{10} đến A_{15} được tổ hợp để chọn 1 trong 64 chip 1k này lên trạng thái tích cực để xuất dữ liệu, còn các địa chỉ từ A_0 đến A_9 dùng để chọn vị trí địa chỉ một ô nhớ cụ thể trong chip đã được chọn. Đầu điều khiển CE_3 dùng để định giờ hoạt động cho ROM, khi tín hiệu $MEM R$ xuất hiện thì dữ liệu đã được chọn được quyền xuất ra Bus dữ liệu.

Sau đây giới thiệu một số dạng ROM thường gặp.

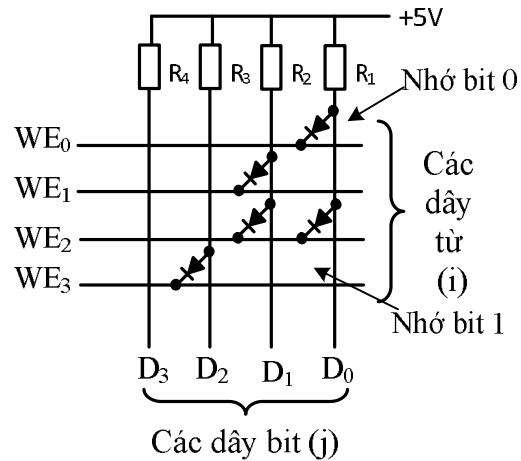
4.2.2. MROM

ROM lập trình theo kiểu mặt nạ được gọi là MROM. Nó được chế tạo trên một phiến silic theo một số bước xử lý như quang khắc và khêch tán để tạo ra những tiếp giáp bán dẫn có tính dẫn điện theo một chiều (như diode, transistor trường).

Người thiết kế định rõ chương trình muốn viết vào ROM, thông tin này được sử dụng để điều khiển quá trình làm mặt nạ. Hình 4-10 là một ví dụ đơn giản về sơ đồ MROM dùng diode.

Chỗ giao nhau giữa các dây từ (hàng) và các dây bit (cột) tạo nên một phần tử nhớ (ô nhớ).

Một diode được đặt tại đó (hình vẽ 4-11) sẽ cho phép lưu trữ số liệu “0”. Ngược lại những vị trí không có diode thì sẽ cho phép lưu trữ số liệu “1” (hoặc ngược lại có diode thì nhớ bit 1, không có thì nhớ bit 0-tùy tính chất mạch).

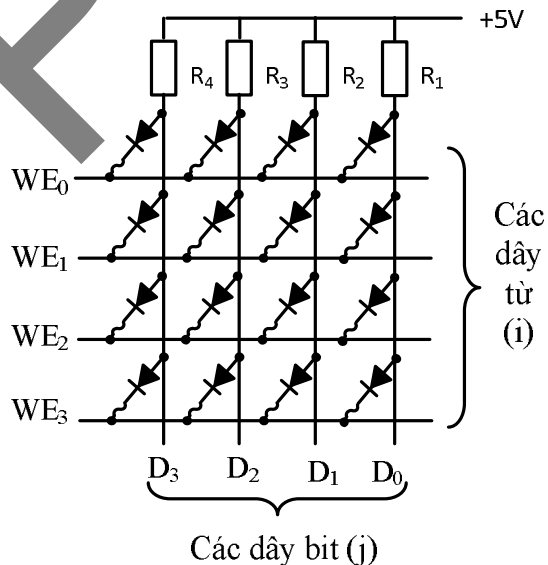


Hình 4-11. MROM dùng diode

Trong trường hợp hình 4-11, khi đọc một từ số liệu thứ i của ROM, bộ giải mã sẽ đặt dây từ đó xuống mức logic thấp, các dây còn lại ở mức cao. Do vậy chỉ những diode nối với dây này được phân cực thuận, do đó nó sẽ dẫn làm cho điện thế đầu ra trên các dây bit tương ứng ở mức logic thấp, các dây bit còn lại sẽ giữ ở mức cao.

Cả hai công nghệ MOS và lưỡng cực được dùng để chế tạo MROM. Thời gian truy nhập của bộ nhớ lưỡng cực khoảng từ 50 – 90 ns, bộ nhớ MOS lâu hơn khoảng 10 lần. Do đó ROM lưỡng cực nhanh hơn và có khả năng kích hoạt tốt hơn trong khi mạch nhớ MOS cùng dung lượng có kích thước nhỏ hơn và tiêu thụ năng lượng ít hơn. Tuy nhiên ảnh hưởng của nhiễu tĩnh điện lớn nên dễ làm mất hay sai lệch thông tin đã được viết nên cần có các biện pháp công nghệ chặt chẽ để đảm bảo an toàn thông tin được lưu trữ.

4.2.3. PROM



Hình 4-12. PROM dùng diode

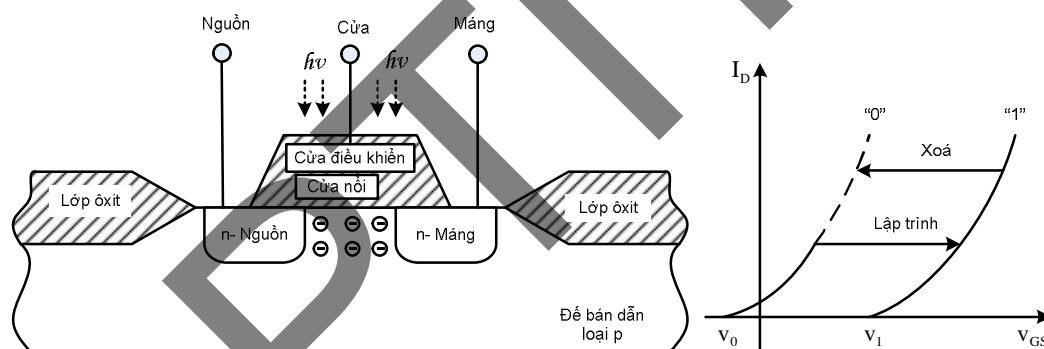
Để khắc phục nhược điểm của MROM là tính năng động kém do thời gian chế tạo phụ thuộc vào thời gian đặt hàng, hầu như không thay đổi được nội dung, giá thành tương đối cao nên người ta sử dụng loại ROM lập trình được – PROM.

PROM cũng gồm có các diode như ở MROM nhưng chúng có mặt đầy đủ tạo các vị trí giao nhau giữa dây từ và dây bit (hình 4-12). Mỗi diode được nối với một cầu chì. Bình thường khi chưa lập trình, các cầu chì còn nguyên vẹn, nội dung của PROM sẽ toàn là 0. Khi định vị đến một bit bằng cách đặt một xung điện ở đầu ra tương ứng, cầu chì sẽ bị đứt và bit này sẽ bằng 1. Bằng cách đó, có thể lập trình toàn bộ các bit trong PROM. Như vậy, việc lập trình đó có thể được thực hiện bởi người sử dụng chỉ một lần duy nhất, không thể sửa đổi được.

4.3. BỘ NHỚ BÁN CÓ ĐỊNH

4.3.1. EPROM (Erasable PROM)

Số liệu vào có thể được viết vào bằng xung điện nhưng được lưu giữ theo kiểu không bay hơi. Đó là loại ROM có thể lập trình được và xóa được. Hình 4- 12 chỉ ra cấu trúc của một transistor dùng để làm một ô nhớ gọi là FAMOST (Floating gate avalanche injection MOS transistor).



Hình 4-13. Cấu trúc của một EPROM

Trong ô nhớ dùng transistor này, cực cửa được nối với đường từ, cực máng được nối với đường bit và cực nguồn được nối với nguồn chuẩn được coi là nguồn cho mức logic 1. Khác với transistor MOS bình thường, transistor loại này còn có thêm một cửa gọi là *cửa nổi* (floating gate); đó là một vùng vật liệu được thêm vào vào giữa lớp cách điện cao như ở hình 4-13. Nếu cửa nổi không có điện tích thì nó không ảnh hưởng gì đến cực cửa điều khiển và transistor hoạt động như bình thường. Tức là khi dây từ được kích hoạt (cực cửa có điện thế dương) thì transistor dẫn, cực máng và nguồn được nối với nhau qua kênh dẫn và dây bit có mức logic 1. Nếu cửa nổi có các điện tử trong đó với điện tích âm thì chúng sẽ ngăn trường điều khiển của cửa cửa và dù dây từ được kích hoạt thì cũng không thể phát ra trường đủ mạnh với cực cửa điều khiển để làm thông transistor. Lúc này đường bit không được nối với nguồn chuẩn và ô nhớ coi như được giữ giá trị 0.

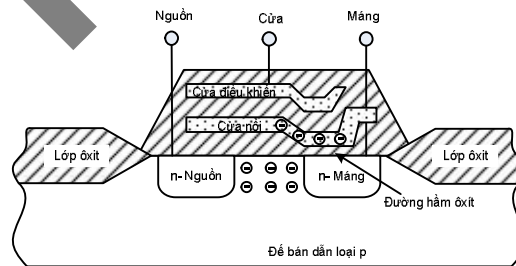
Việc nạp các điện tử vào vùng cửa nổi, tức là tạo ra các ô nhớ mang giá trị 0 được thực hiện bởi xung điện có độ dài cỡ 50 ms và độ lớn + 20 V đặt giữa cực cửa và cực máng. Lúc đó những điện tích mang năng lượng lớn sẽ đi qua lớp cách điện giữa đế và cửa nổi. Chúng tích tụ trong vùng cửa nổi và được giữ ở đây sau khi xung lập trình tắt. Đó là do cửa nổi được cách điện cao với xung quanh và các điện tử không còn đủ năng lượng sau khi lạnh đi, để có thể vượt ra ngoài lớp cách điện đó nữa. Chúng sẽ được giữ ở đây trong một thời gian rất dài (ít nhất là 10 năm).

Để xoá các thông tin, tức là làm mất các điện tích điện tử trong vùng cửa nổi, phải chiếu ánh sáng tử ngoại UV vào chip nhớ. Lúc này, những điện tử hấp thụ được năng lượng và sẽ nhảy lên các mức năng lượng cao và rời khỏi cửa nổi giống như cách mà chúng đã thâm nhập vào. Trong chip EPROM có một cửa sổ làm bằng thủy tinh thạch anh chỉ để cho ánh sáng tử ngoại đi qua khi cần xoá số liệu trong bộ nhớ.

4.3.2. EEPROM (Electrically Erasable PROM)

Cửa sổ thạch anh có giá thành khá đắt và không tiện lợi nên những năm gần đây xuất hiện các chip PROM có thể xoá số liệu bằng phương pháp điện. Cấu trúc của ô nhớ giống như hình 4-14.

Việc nạp các điện tử cho cửa nổi được thực hiện như cách ở EPROM. Bằng một xung điện tương đối dài, các điện tích mang năng lượng cao được phát ra trong đế sẽ thấm qua lớp cửa ôxit và tích tụ trong cửa nổi. Để xoá EEPROM, một lớp kênh màng mỏng ôxit giữa vùng cửa nổi trải xuống dưới đế và cực máng giữ vai trò quan trọng. Các lớp cách điện không thể là lý tưởng được, các điện tích có thể thấm qua lớp phân cách với một xác suất thấp. Xác suất này tăng lên khi bề dày của lớp giảm đi và điện thế giữa hai điện cực ở hai mặt lớp cách điện tăng lên. Muốn phóng các điện tích trong vùng cửa nổi một điện thế (-20 V) được đặt vào cực cửa điều khiển và cực máng. Lúc này các điện tử âm trong cửa nổi được chảy về cực máng qua kênh màng mỏng ôxit và số liệu lưu giữ được xoá đi. Điều lưu ý là phải làm sao cho dòng điện tích này chảy không quá lâu vì nếu không vùng cửa nổi này lại trở nên tích điện dương làm cho hoạt động của transistor không được trạng thái bình thường (mức nhớ 1).



Hình 4-14. Cấu trúc của EEPROM

Các chip ROM hiện nay có thời gian truy nhập từ 120 ns đến 150 ns dài hơn nhiều thời gian đó trong các chip nhớ RAM.

4.4. RAM

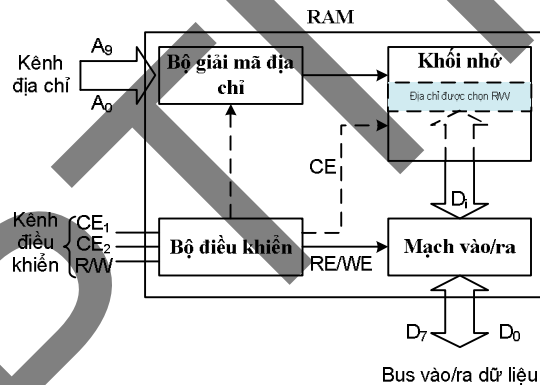
RAM có khả năng cho phép viết lưu trữ dữ liệu thông tin tạm thời trong một thời gian, sau đó lại đọc thông tin đó để tiếp tục xử lý khi cần thiết nên nó có tên là bộ nhớ đọc/viết. Một đặc tính quan trọng khác của RAM là các dữ liệu trong RAM chỉ có tính chất tạm thời, dễ bị xóa khi mất nguồn năng lượng cấp cho nó nên cần phải phòng bị cho nó một nguồn phụ nhằm tránh hiện tượng mất nguồn đột ngột, trong khi quá trình xử lý chưa kết thúc, thông tin trong RAM còn có ích.

4.4.1. Cấu trúc khối của RAM

Cũng như ROM, RAM cũng có 4 phần chính như mô tả trên hình 4-15. Điểm khác biệt là:

- + Mạch điều khiển của RAM phải có thêm đầu vào R/W điều khiển hai quá trình cơ bản trong thao tác của RAM: viết dữ liệu thông tin vào nó và quá trình xuất (đọc) thông tin đã viết.

- + Mạch đầu ra có khả năng kiểm soát hai chiều trước khi cho phép giao tiếp với kênh dữ liệu. Quá trình này tuân theo nguyên tắc: (đồng bộ với việc điều khiển R/W) khi bộ nhớ đang đọc thì không được viết và ngược lại; trạng thái thứ ba có thể chờ quyết định.



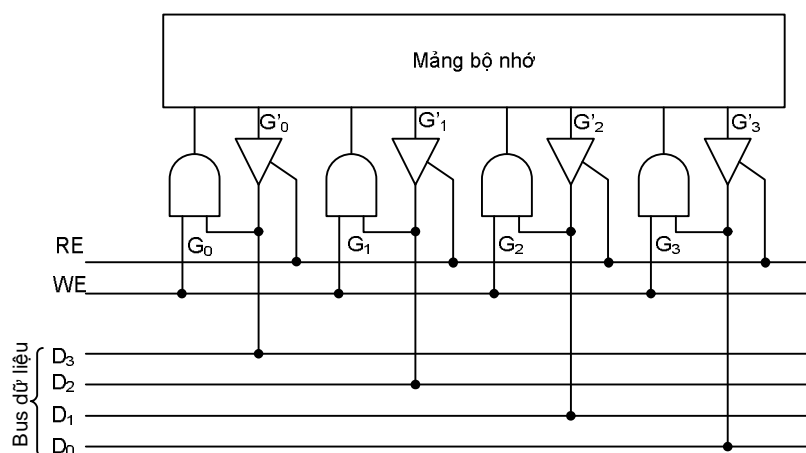
Hình 4-15. Cấu trúc 4 khối của một RAM có 8 bit dữ liệu và 8 bit địa chỉ

4.4.1.1. Mạch vào/ra của RAM

Mạch vào ra của RAM phức tạp hơn ROM. Hình 4-16 mô tả một dạng mạch vào/ra với bốn dữ liệu từ D_0 đến D_3 .

Các dữ liệu trên kênh dữ liệu G_0 , G_1 , G_2 và G_3 viết vào bộ nhớ khi tín hiệu cho phép viết $WE = 1$. Khi có tín hiệu điều khiển cho phép đọc $RE = 1$ thì các cổng G'_0 , G'_1 , G'_2 và G'_3 được điều khiển mở (đây là cổng 3 trạng thái) để xuất các dữ liệu ra kênh.

Một điều cần chú ý là các thao tác trên phải đồng bộ với việc chọn xong địa chỉ viết hay đọc trong bộ nhớ.



Hình 4-16. Mạch vào/ra 4 bit dữ liệu của RAM

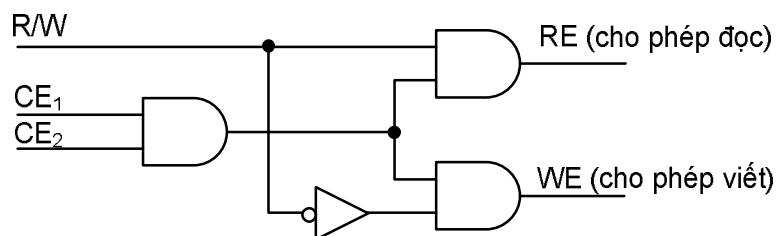
Cổng 3 trạng thái còn có khả năng đưa mức ra của G'_0 , G'_1 , G'_2 và G'_3 lên trạng thái trở kháng cao, do đó, bộ nhớ được đặt trong tình trạng chờ: không cho dữ liệu ra và cũng không thực hiện viết dữ liệu vào. Lúc này toàn bộ dữ liệu được cách ly ra khỏi kênh dữ liệu cho an toàn.

4.4.1.2. Mạch điều khiển của RAM

Hình 4-17 mô tả một dạng mạch đơn giản đưa tới RAM 3 chế độ hoạt động khác nhau từ 3 đầu vào điều khiển CE_1 , CE_2 và R/W . Ở mỗi thời điểm chỉ có một trong ba chế độ được thiết lập:

- + Khi ở chế độ đọc, xung R/W ở mức logic 1. Đồng thời các tín hiệu cho phép chọn CE_1 , CE_2 được kích hoạt ở mức 1 nên lúc này $RE = 1$, tức là chế độ đọc được thiết lập. Khi đó tín hiệu $\overline{R/W} = 0$ nên tín hiệu cho phép viết WE bằng 0 (cấm viết).

- + Khi ở chế độ viết, xung R/W ở mức logic 0, $\overline{R/W} = 1$, đồng thời các tín hiệu cho phép chọn CE_1 , CE_2 được kích hoạt ở mức 1 nên lúc này $WE = 1$, tức là nó ở chế độ viết. Khi đó tín hiệu $R/W = 0$ nên tín hiệu cho phép đọc RE bằng 0 (cấm đọc).

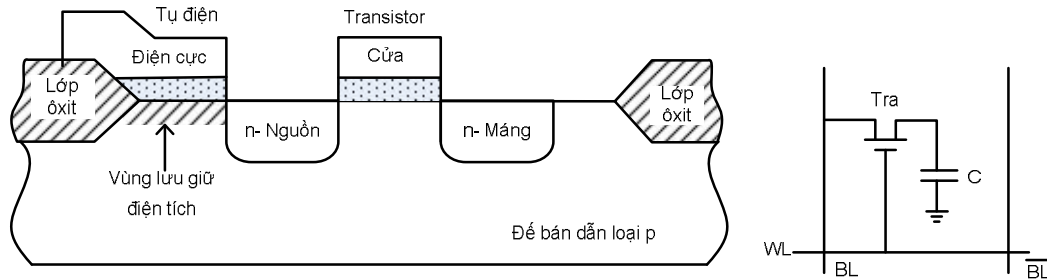


Hình 4-17. Mạch điều khiển RAM có 3 chế độ: đọc/viết/chờ

- + Tín hiệu tích cực đồng thời $CE_1 = CE_2 = 1$ ở cả hai chế độ đọc và viết phải được chuyển cùng lúc tới mảng ô nhớ nhằm thông báo việc xuất (khi đọc) hay việc nhập (khi viết) dữ liệu tới địa chỉ ô nhớ đã được mạch giải mã chọn.

+ Khi tín hiệu $CE_1 \cdot CE_2 = 0$ (có ít nhất một tín hiệu CE ở trạng thái không tích cực) thì mạch điều khiển ở hình 4-20 sẽ chuyển bộ nhớ sang chế độ chờ (Standby) bất chấp tín hiệu R/W có tích cực hay không, lúc này $RE = 0$ và $WE = 0$.

4.4.2. Cấu tạo của DRAM



Hình 4-18. Cấu tạo một ô nhớ của DRAM

Các ô nhớ được sắp xếp theo hàng và cột trong một ma trận nhớ. Địa chỉ ô nhớ được chia thành hai phần: địa chỉ hàng và cột. Hai địa chỉ này được đọc vào bộ đếm một cách lần lượt. Xử lý kiểu này được gọi là hợp kênh, lý do là để giảm kích thước bộ giải mã, tức là giảm kích thước và giá thành vi mạch. Quá trình dò kênh địa chỉ này được điều khiển bởi các tín hiệu RAS (Row Access Strobe) và CAS (Column Access Strobe).

Nếu \overline{RAS} ở mức tích cực thấp thì DRAM nhận được địa chỉ đặt vào nó và sử dụng như địa chỉ hàng.

Nếu \overline{CAS} ở mức tích cực thấp thì DRAM nhận được địa chỉ đặt vào nó và sử dụng như địa chỉ cột.

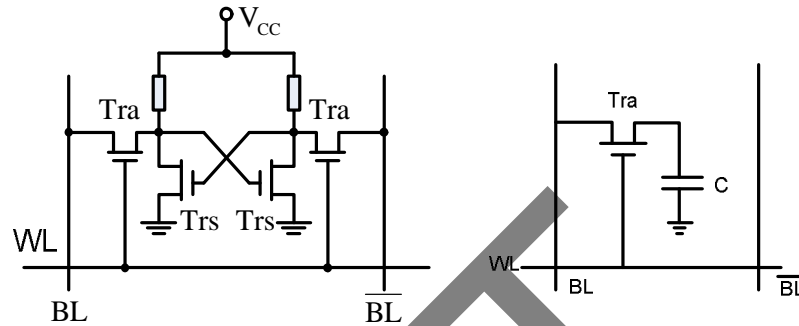
Một ô nhớ của DRAM gồm có một transistor trường MOS có trở đầu vào rất lớn và một tụ điện C là linh kiện lưu trữ một bit thông tin tương ứng với hai trạng thái có hoặc không có điện tích trên tụ.

Transistor hoạt động như một công tắc, cho phép nạp hay phóng điện tích của tụ khi thực hiện phép đọc hay viết. Cực cửa (Gate) của transistor được nối với dây hàng (còn gọi là dây từ-WL-Word Line) và cực máng (Drain) được nối với dây cột (còn được gọi là dây bit BL hoặc \overline{BL} -Bit Line), cực nguồn (Source) được nối với tụ điện. Điện áp nạp trên tụ tương đối nhỏ, vì thế cần sử dụng khuếch đại nhạy trong mạch nhớ.

Phần tử lưu giữ một bit dữ liệu là điện dung C, thường được tạo thành bằng điện dung tiếp giáp của MOSFET, có điện dung rất nhỏ, khoảng một vài pF. Do đó, thể tích của một tế bào DRAM bé hơn rất nhiều so với tế bào RAM tĩnh (SRAM). Nhờ thế, bộ nhớ DRAM có dung lượng nhớ rất lớn và giá thành thấp. Tuy nhiên, nhược điểm cơ bản của loại này là tụ điện không thể giữ được năng lượng đã nạp lâu dài. Trong DRAM luôn xảy ra sự tự phóng điện (dòng rò). Hiện tượng này làm điện áp trên tụ giảm dần theo thời gian được gọi là tự bay hơi. Vì vậy cứ sau 2 ms, nếu không được nạp bồi thì dữ liệu sẽ mất đi. Để duy trì dữ liệu cần phải định kỳ thực hiện nạp bồi. Quá trình này gọi là làm tươi (refresh) DRAM.

4.4.3. SRAM

Một ô nhớ của SRAM giữ thông tin bởi trạng thái của mạch trigơ. Thuật ngữ “tĩnh” chỉ ra rằng khi nguồn nuôi chưa bị cắt thì thông tin của ô nhớ vẫn được giữ nguyên. Khác với ô nhớ DRAM, ở đây ô nhớ trigơ cung cấp một tín hiệu số mạnh hơn nhiều vì đã có các transistor trong các ô nhớ, chúng có khả năng khuếch đại tín hiệu và do đó có thể cấp trực tiếp cho các đường bit. Trong DRAM, sự khuếch đại tín hiệu trong các bộ khuếch đại cần nhiều thời gian và do đó thời gian truy nhập dài hơn. Khi định địa chỉ trong các trigơ ở SRAM, các transistor bổ sung cho các trigơ, các bộ giải mã địa chỉ... cũng được đòi hỏi như ở DRAM.



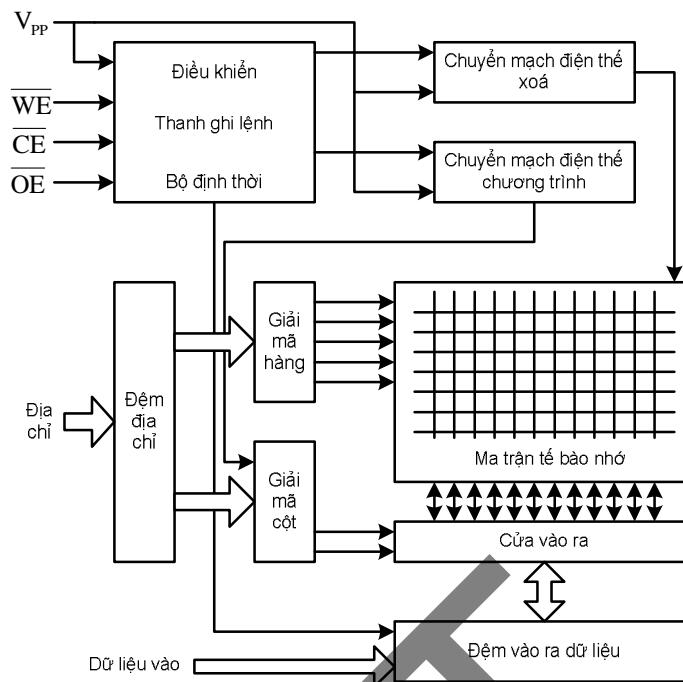
Hình 4-19. Cấu tạo một ô nhớ của SRAM và DRAM

Như trong DRAM, cực cửa của transistor được nối với đường từ và cực máng nối với cặp đường bit. Nếu số liệu được đọc từ ô nhớ, khi đó bộ giải mã hàng kích hoạt đường dây từ WL tương ứng. Hai transistor T dẫn và nối trigơ nhớ với cặp dây bit. Như vậy hai đầu ra Q và \bar{Q} được nối với các đường bit và các tín hiệu được truyền tới bộ khuếch đại ở cuối đường dây này. Vì điện thế chênh lệch lớn nên xử lý khuếch đại như vậy sẽ nhanh hơn trong DRAM (cỡ 10 ns hoặc ngắn hơn), do đó chip SRAM cần địa chỉ cột sớm hơn nếu thời gian truy nhập không được giảm. Như vậy SRAM không cần thực hiện phân kênh các địa chỉ hàng và cột. Sau khi số liệu ổn định, bộ giải mã cột chọn cột phù hợp và cho ra tín hiệu số liệu tới bộ đệm số liệu ra và tới mạch ra.

Viết số liệu được thực hiện theo cách ngược lại. Qua bộ đệm vào và bộ giải mã cột, số liệu viết được đặt vào bộ khuếch đại phù hợp. Cùng lúc đó bộ giải mã hàng kích hoạt đường dây từ và làm transistor T dẫn. Trigơ đưa số liệu được lưu trữ vào cặp dây bit. Tuy vậy, bộ khuếch đại nhạy hơn các transistor nên nó sẽ cấp cho các đường bit một tín hiệu phù hợp với số liệu viết. Do đó, trigơ sẽ chuyển trạng thái phù hợp với số liệu mới hoặc giữ giá trị đã được lưu trữ phụ thuộc vào việc số liệu viết trùng với số liệu đã lưu trữ hay không.

4.5. ĐĨA CỨNG SILICON- BỘ NHỚ FLASH

Trong những năm gần đây, một loại bộ nhớ không bay hơi mới đã xuất hiện trên thị trường, thường được sử dụng thay thế cho các ổ đĩa mềm và cứng trong những máy tính. Đó là bộ nhớ flash. Cấu trúc của chúng cơ bản như EEPROM, chỉ có lớp kênh ôxit ở các ô nhớ mỏng hơn. Do vậy chỉ cần điện thế cỡ 12 V là có thể cho phép thực hiện 10 000 chu trình xóa và lập trình. Bộ nhớ flash có thể hoạt động gần mềm dẻo như DRAM và SRAM nhưng lại không bị mất số liệu khi bị cắt điện. Hình 4-20 chỉ ra sơ đồ khối của nó.



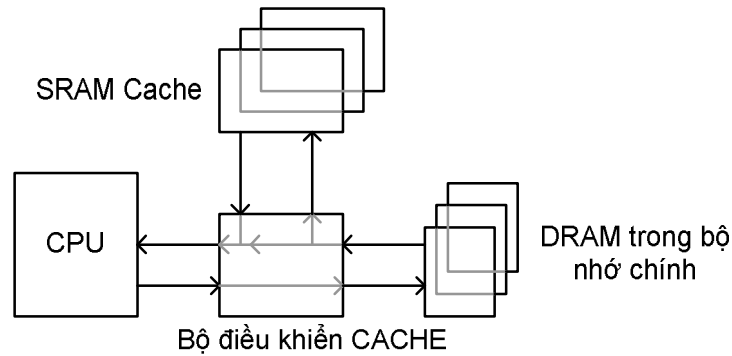
Hình 4-20. Sơ đồ bộ nhớ FLASH

Phần chính là mạng nhớ bao gồm các ô nhớ FAMOST như được mô tả ở mục trên. Giống như SRAM, bộ nhớ flash không dồn phân kênh địa chỉ. Các bộ giải mã hàng và cột chọn một đường từ và một hoặc nhiều cặp đường bit. Số liệu đọc được đưa ra ngoài bộ đệm số liệu I/O hoặc được viết vào ô nhớ đã được định địa chỉ bởi bộ đệm này qua cổng I/O. Xử lý đọc được thực hiện với điện thế MOS thông thường là 5V. Để lập trình một ô nhớ, đơn vị điều khiển flash đặt một xung điện thế ngắn cỡ 10 μ s và 12 V gây nên một sự chọc thủng thác lũ vào transistor nhớ để nạp vào cửa nổi. Một chip nhớ flash 1 Mb có thể được lập trình trong khoảng 2 sec, nhưng khác với EEPROM việc xoá được thực hiện từng chip một. Thời gian xoá cho toàn bộ bộ nhớ flash khoảng 1 sec. Xử lý đọc, lập trình và xoá được điều khiển bởi các lệnh có độ dài 2 byte được bộ xử lý viết vào các thanh viết lệnh của mạch điều khiển flash.

Mục đích sử dụng chính của bộ nhớ flash là để thay thế cho các ổ đĩa mềm và ổ đĩa cứng dung lượng nhỏ. Do nó là mạch tích hợp nên có ưu điểm là kích thước nhỏ và tiêu thụ năng lượng thấp, không bị ảnh hưởng của va đập. Các đĩa cứng chất rắn dựa trên cơ sở các bộ nhớ flash có lợi thế về công suất tiêu thụ cũng như giá thành có dung lượng tới vài Mbyte. Các card nhớ loại này có ưu điểm là không gặp phải vấn đề mất thông tin như trường hợp RAM CMOS khi pin Ni-Cd bị hỏng. Thời gian lưu trữ thông tin trong bộ nhớ flash ít nhất là 10 năm, thông thường là 100 năm, với khoảng thời gian này thì các đĩa mềm và cứng đã bị hỏng rồi.

Nhược điểm của bộ nhớ flash là chỉ có thể xoá theo kiểu lần lượt từng chip hoặc lần lượt từng trang.

4.6. BỘ NHỚ CACHE



Hình 4-21. Nguyên lý của Cache

Với các máy tính có tốc độ nhanh (trên 33MHz), cần phải xen các trạng thái đợi khi truy xuất dữ liệu tới các DRAM rẻ tiền nhưng có thời gian thâm nhập chậm (60-120ns). Điều này làm giảm hiệu suất của máy. Có thể giải quyết bằng cách dùng các SRAM có thời gian thâm nhập ngắn hơn (20-25 ns, thậm chí 12 ns) nhưng giá thành lại rất đắt. Bộ nhớ Cache kết hợp được các lợi điểm nhanh của SRAM và rẻ của DRAM. Giữa CPU và bộ nhớ chính bằng DRAM, người ta xen vào một bộ nhớ SRAM nhanh có dung lượng nhỏ bằng 1/10 hoặc 1/100 lần bộ nhớ chính gọi là cache; dưới sự điều khiển của mạch điều khiển cache, bộ nhớ này sẽ lưu trữ tạm thời các số liệu thường được gọi và cung cấp nó cho CPU trong thời gian ngắn.

Cache chứa các thông tin mới vừa được CPU sử dụng gần đây nhất. Khi CPU đọc số liệu nó sẽ đưa ra một địa chỉ tới bộ điều khiển cache. Sau đó một trong hai quá trình sau sẽ xảy ra:

Cache hit: nếu địa chỉ đó đã có sẵn trong RAM cache.

Cache miss: ngược lại, nếu địa chỉ đó không có sẵn trong RAM cache.

Như vậy, cache hit tỷ lệ với truy xuất thông tin có sẵn trong bộ nhớ cache SRAM, còn cache miss lại tỷ lệ với truy xuất thông tin có trong bộ nhớ chính là các DRAM.

TÓM TẮT

Trong chương này chúng ta trình bày nguyên lý cấu tạo, các tính năng cơ bản của các loại bộ nhớ bán dẫn: ROM, PROM, EPROM, EEPROM, SRAM, DRAM, FLASH, CACHE.

Các chip RAM không thích hợp cho các chương trình khởi động do các thông tin trên đó bị mất khi tắt nguồn. Do vậy phải dùng đến ROM, trong đó các số liệu cần lưu trữ được viết một lần theo cách không bay hơi để nhằm giữ được mãi.

Trong những năm gần đây, một loại bộ nhớ không bay hơi mới đã xuất hiện trên thị trường, thường được sử dụng thay thế cho các ổ đĩa mềm và cứng trong những máy tính. Đó là bộ nhớ flash. Cấu trúc của chúng cơ bản như EEPROM, chỉ có lớp kênh ôxít ở các ô nhớ mỏng hơn.

CÂU HỎI ÔN TẬP

1. Bộ nhớ ROM về cơ bản khác bộ nhớ RAM ở điểm gì?
2. Linh kiện lưu giữ bit thông tin của DRAM là linh kiện gì?
3. Linh kiện lưu giữ bit thông tin của SRAM là linh kiện gì?
4. Linh kiện lưu giữ bit thông tin của EPROM là linh kiện gì?
5. Trong EPROM, việc nạp các điện tích vào vùng cửa nổi có nghĩa là làm gì cho nó?
6. SRAM 6264: Cần bao nhiêu bit địa chỉ cho nó khi dung lượng là 2048 từ x 8 bit? Nó có thể nhớ bao nhiêu từ 32 byte? Nếu ghép song song 4 IC này thì dung lượng bộ nhớ tổng cộng là bao nhiêu?
7. Làm tươi bộ nhớ DRAM là gì? Tại sao phải làm tươi DRAM?
8. Một DRAM có dung lượng nhớ là 64 kbit thì cần bao nhiêu đầu vào/ra? Nó chứa được bao nhiêu từ nhị phân 8 bit? Vẽ sơ đồ khối của bộ nhớ?
9. Cho ROM có dung lượng 1k x 8. Hãy mở rộng dung lượng nhớ từ (1k x 8) bit thành (2k x 8) bit?
10. Cho bộ nhớ DRAM có dung lượng là 256 x 1 Kbit. Hãy mở rộng độ dài từ của bộ nhớ lên 1 Byte?

TÀI LIỆU THAM KHẢO

1. *Giáo trình Kỹ thuật số* - Trần Văn Minh, NXB Bru điện 2002.
2. *Cơ sở kỹ thuật điện tử số*, Đại học Thanh Hoa, Bắc Kinh, NXB Giáo dục 1996 .
3. *Kỹ thuật số*, Nguyễn Thúy Vân, NXB Khoa học và kỹ thuật 1994.
4. *Kỹ thuật điện tử số thực hành*, Bạch Gia Dương – Chữ Đức Trình, Nhà xuất bản đại học quốc gia Hà nội 2007.
5. *Giáo trình Kỹ thuật số*, Nguyễn Viêt Nguyễn, Nhà xuất bản giáo dục 2004.
6. *Mạch logic kỹ thuật số*, Nguyễn Minh Đức, Nhà xuất bản tổng hợp thành phố Hồ Chí Minh, 2004.
7. *Toán logic và kỹ thuật số*, Nguyễn Nam Quân - Khoa ĐHTC xuất bản – 2004.
8. *Lý thuyết mạch logic và Kỹ thuật số*, Nguyễn Xuân Quỳnh - NXB Bru điện – 1984.
9. *Cấu trúc máy vi tính*, Trần Quang Vinh, NXB Đại học Quốc gia Hà nội, 2005.
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.
14. *VHDL for Programmable Logic* by Kevin Skahill, Addison Wesley, 1996
15. *The Designer's Guide to VHDL* by Peter Ashenden, Morgan Kaufmann, 1996.
16. *Analysis and Design of Digital Systems with VHDL* by Dewey A., PWS Publishing, 1993.

PHỤ LỤC

GIỚI THIỆU VỀ HỆ ĐẾM

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 được 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...

A1. 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 ta 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.

Biểu diễn số tổng quát:

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

A2. 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.

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 :

$$N_2 = \sum_{i=-m}^{n-1} a_i 2^i$$

Trong đó, a 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 \rightarrow số nhị phân phân số

2^2 2^1 2^0 2^{-1} 2^{-2} \rightarrow trọng số tương ứng.

A3. Các phép tính trong hệ nhị phân

a. Phép cộng

Quy 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:

$$\begin{array}{r} 101_2 \quad (5_{10}) \\ + 100_2 \quad (4_{10}) \\ \hline 1001_2 \quad (9_{10}) \end{array} \quad \begin{array}{r} (13_{10}) \quad 1101_2 \\ (11_{10}) \quad + 1011_2 \\ \hline (24_{10}) \quad 11000_2 \end{array} \quad \begin{array}{r} (4,375_{10}) \quad 100,011_2 \\ (3,750_{10}) \quad + 11,110_2 \\ \hline (8,125_{10}) \quad 1000,001_2 \end{array}$$

b. Phép trừ

Quy 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:

$$\begin{array}{r} 1101_2 \quad (13_{10}) \\ - 110_2 \quad (6_{10}) \\ \hline 0111_2 \quad (7_{10}) \end{array} \quad \begin{array}{r} 25_{10} \quad 11001_2 \\ (11_{10}) \quad - 1011_2 \\ \hline (14_{10}) \quad 01110_2 \end{array} \quad \begin{array}{r} (5,3125_{10}) \quad 101,0101_2 \\ (2,8125_{10}) \quad - 10,1101_2 \\ \hline (2,5000_{10}) \quad 010,1000_2 \end{array}$$

c. Phép nhân

Quy 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:

$$\begin{array}{r} 1001_2 \quad (9_{10}) \\ \times 11_2 \quad (3_{10}) \\ \hline 1001 \\ + 1001 \\ \hline 11011_2 \quad (27_{10}) \end{array} \quad \begin{array}{r} (5,5_{10}) \quad 101,1_2 \\ (2_{10}) \quad \times 10_2 \\ \hline 0000 \\ + 1011 \\ \hline (11_{10}) \quad 1011,0 \end{array}$$

d. Phép chia

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

Ví dụ:

$$\begin{array}{r} 100'1_2 \quad | \quad 11_2 \\ - 11 \quad \quad \quad 11 \\ \hline 0011 \\ - 11 \\ \hline 0000 \end{array}$$

Trong trường hợp số bị chia nhỏ hơn số chia thì 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.

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.

B. Hệ 8 (bát phân) và hệ 16 (thập lục phân)

Hệ 8 (Octal number systems)

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:

$$N_8 = \sum_{i=-m}^{n-1} a_i 8^i$$

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

Hệ 16

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, ta 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:

$$N_{16} = \sum_{i=-m}^{n-1} a_i 16^i$$

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

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

Biểu diễn số của 4 hệ đếm thường dùng.

C. Chuyển đổi cơ số giữa các hệ đếm

C1. 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 ta phải chia ra hai phần: phần nguyên và phần số.

Đối với phần nguyên:

Ta 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

Ta có : $35_{10} = 100011_2$

Ví dụ 2: Đổi số 35_{10} sang hệ 8.

35	8	=4	Dư 3	a_0
4	8	=0	Dư 4	a_1

Ta có : $35_{10} = 43_8$

Ví dụ 3: Đổi số 35_{10} sang hệ 16.

$$\begin{array}{r|l} 35 & 16 = 2 \text{ Dư } 3 \\ 2 & 16 = 0 \text{ Dư } 2 \end{array} \quad \begin{array}{l} a_0 \\ a_1 \end{array}$$

Ta có : $35_{10} = 23_{16}$

Đổi với phần phân số :

Khi chuyển phần phân số ta 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 : Đổi số $35,375_{10}$ sang số nhị phân.

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

$$\begin{array}{r|l} 0,375 \times 2 = 0,75 & \text{Phần nguyên} = 0 \\ 0,75 \times 2 = 1,5 & \text{Phần nguyên} = 1 \\ 0,5 \times 2 = 1,0 & \text{Phần nguyên} = 1 \\ 0,0 \times 2 = 0 & \text{Phần nguyên} = 0 \end{array} \quad \begin{array}{l} a_{-1} \\ a_{-2} \\ a_{-3} \\ a_{-4} \end{array}$$

Kết quả : $0,375_{10} = 0,0110_2$

Sử dụng phần nguyên đã có ở ví dụ 1) ta có : $35,375_{10} = 100011,0110_2$

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

$$\begin{array}{r|l} 0,375 \times 8 = 3,0 & \text{Phần nguyên} = 3 \\ 0,0 \times 8 = 0 & \text{Phần nguyên} = 0 \end{array} \quad \begin{array}{l} a_{-1} \\ a_{-2} \end{array}$$

Kết quả : $0,375_{10} = 0,3_8$

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

$$\begin{array}{r|l} 0,375 \times 16 = 6,0 & \text{Phần nguyên} = 6 \\ 0,0 \times 16 = 0 & \text{Phần nguyên} = 0 \end{array} \quad \begin{array}{l} a_{-1} \\ a_{-2} \end{array}$$

Kết quả : $0,375_{10} = 0,6_{16}$

C.2. Đổi một biểu diễn trong hệ bất kì sang hệ thập phân

Muốn thực hiện phép biến đổi, ta dùng công thức :

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

Thực hiện lấy tổng về 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.

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

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

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

C.3. *Đổi các số từ hệ nhị phân sang hệ cơ số 8 và 16*

Vì $8 = 2^3$ và $16 = 2^4$ nên ta 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 ta 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, ta chia số này thành các nhóm 3 bit như sau :

110	111	,	011	100
↓	↓		↓	↓
6	7		3	4

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

b. Đổi số nhị phân $111110110,01101_2$ sang số hệ cơ số 16

Ta phân nhóm và thay thế như sau :

0001	1111	0110	0110	1000
↓	↓	↓	↓	↓
1	F	6	6	8

Kết quả: $111110110,01101_2 = 1F6,68_{16}$