BỘ LAO ĐỘNG - THƯƠNG BINH VÀ XÃ HỘI

TỔNG CỤC DẠY NGHỀ

GIÁO TRÌNH

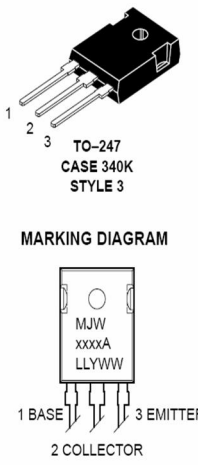
Tên mô đun: Kỹ thuật số

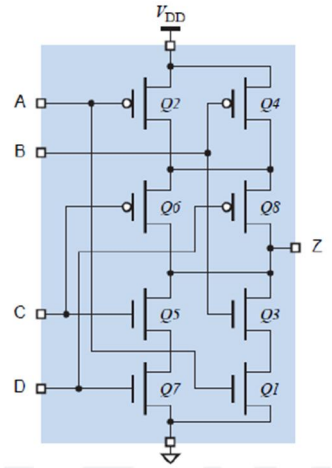
NGHỀ: ĐIỆN DÂN DỤNG

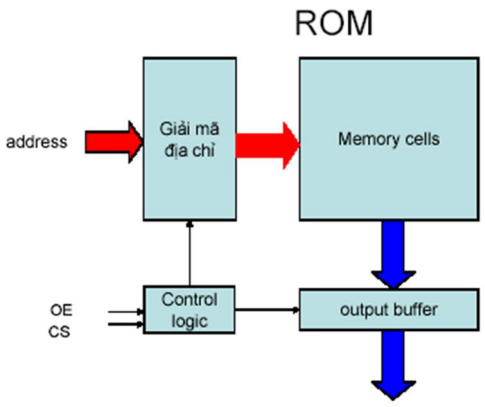
TRÌNH ĐỘ CAO ĐẲNG NGHỀ

(Ban hành kèm theo Quyết định số: 120 /QĐ-TCDN

Ngày 25 tháng 02 năm 2013 của Tổng cục trưởng Tổng cục Dạy nghề)







TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Hiện nay ở nước ta hầu hết các hoạt động của xã hội đều gắn với việc sử dụng điện năng. Điện không những được sử dụng ở thành phố mà còn được đưa về nông thôn, miền núi hoặc nhờ các trạm phát điện địa phương .

Cùng với sự phát triển của điện năng các thiết bị điện dân dụng ngày càng được "điện tử hóa" nhiều hơn. Chất lượng của các thiết bị điện dân dụng cũng như công nghiệp cũng không ngừng được cải tiến và nâng cao, cùng với sự phát triển của công nghệ mới các thiết bị đã được "số hóa" rất nhiều. Vì vậy đòi hỏi người làm việc trong các ngành, nghề và đặc biệt trong các nghề điện, điện tử không chỉ phải hiểu rõ về bản chất của các thiết bị điện thông qua việc hiểu rõ về cấu tạo, nguyên lý làm việc của những linh kiện điện tử, các mạch điện tử cơ bản và ứng dụng của chúng mà còn phải đi sâu tìm hiểu những công nghệ mới - công nghệ số.

Nội dung môn học "Kỹ thuật số" nhằm trang bị cho học viên những kiến thức cơ bản và những kỹ năng cần thiết về kỹ thuật số.

Trong chương trình đào tạo, mô - đun Kỹ thuật số là một trong những mô đun cơ sở phục vụ tốt cho các môn học, mô đun về chuyên ngành điện dân dụng và một số chuyên ngành khác như Điện công nghiệp, Điện tử công nghiệp cũng như dân dụng.

Những kiến thức mà giáo trình Kỹ thuật số cung cấp giúp cho người học học tốt hơn các môn học và mô đun: Thiết bị nhiệt gia dụng, thiết bị lạnh gia dụng, thiết bị tự động điều khiển dân dụng … trong chương trình đào tạo nghề Điện dân dụng.

Cấu trúc cơ bản của giáo trình bao gồm 12 bài:

Bài 1: Cơ sở kỹ thuật số;

Bài 2: Các phần tử lôgic cơ bản;

Bài 3: Các phần tử lôgic thông dụng;

Bài 4: Mạch mã hóa;

Bài 5: Mạch giải mã;

Bài 6: Mạch dồn kênh;

Bài 7: Mạch phân kênh;

Bài 8: Mạch lôgic tuần tự;

Bài 9: Mạch nhớ ROM;

Bài 10: Mạch nhớ RAM;

Bài 11: Mạch chuyển đổi A/D;

Bài 12: Mạch chuyển đổi D/A.

Trong quá trình biên soạn, nhóm tác giả đã tham khảo các tài liệu và giáo trình khác như ở phần cuối giáo trình đã thống kê.

Chúng tôi rất cảm ơn các cơ quan hữu quan của TCDN, BGH và các thày cô giáo trường CĐN Bách nghệ Hải Phòng cùng một số giáo viên có kinh nghiệm của các trường, cơ quan khác đã tạo điều kiện giúp đỡ cho nhóm tác giả hoàn thành giáo trình này.

Lần đầu được biên soạn và ban hành, giáo trình chắc chắn sẽ còn khiếm khuyết; rất mong các thày cô giáo và những cá nhân, tập thể của các trường đào tạo nghề và các cơ sở doanh nghiệp quan tâm đóng góp để giáo trình ngày càng hoàn thiện hơn, đáp ứng được mục tiêu đào tạo của mô đun nói riêng và ngành điện dân dụng cũng như các chuyên ngành kỹ thuật nói chung.

Mọi ý kiến đóng góp xin gửi về địa chỉ:

Trường Cao đẳng nghề Bách Nghệ Hải Phòng
Khoa Điện – Điện tử
Số 196/143 Đường Trường Chinh - Quận Kiến An - TP Hải Phòng
Email: khoadienbn@gmail.com

Hà Nội, ngày…..tháng…. năm 2013

Tham gia biên soạn

1. Chủ biên: PGS. TS. Phạm Ngọc Tiệp
2. Nguyễn Long Biên
3. Nguyễn Lâm Đông
4. Nguyễn Thế Chính

MỤC LỤC

TRANG

Lời giới thiệu ……
Bài 1: **Cơ sở kỹ thuật số** ……
1. Khái niệm tín hiệu số, tín hiệu tương tự
1.1. Tín hiệu tương tự
1.2 Tín hiệu số
2. Khái niệm các hệ đếm và mã
2.1. Các hệ đếm thông dụng.
2.2. Mã và các loại mã thông dụng.
3. Thực hiện các phép tính và chuyển đổi mã
3.1. Các phép tính trong kỹ thuật số.
3.2. Chuyển đổi giữa các hệ mã thông dụng.
4. Đại số logic
4.1. Khái niệm về đại số logic
4.2. Các hàm logic cơ bản
4.3. Biểu diễn hàm logic
4.4. Tối giản hàm logic
5. Thực hiện một số hàm logic cơ bản
Bài 2: **Các phần tử logic cơ bản**
1. Các cổng logic cơ bản
1.1. Cổng AND
1.2. Cổng OR
1.3. Cổng NOT
1.4. Cổng NAND
1.5. Cổng NOR
1.6. Cổng XOR
2. Một số ứng dụng cổng logic cơ bản
2.1 Mạch tạo xung vuông dùng các cổng NOT, NAND của IC 4049 và 7400.
2.2 Mạch đếm đến 8 dùng IC 74164
2.3 Mạch đếm đến 10 dùng IC 4017
3. Lắp ráp và cân chỉnh một số mạch dùng cổng logic cơ bản
3.1 Lắp mạch đếm 10 dùng IC 4017
3.2 Hiệu chỉnh và thay đổi hiệu ứng.
Bài 3: **Các phần tử logic thông dụng**
1. Mạch tạo thành cổng NAND
1.1. Tạo cổng NAND từ đi ốt - BJT
1.2. Tạo cổng NAND từ BJT - FET
2. Mạch tạo thành cổng NOR
2.1. Tạo cổng NOR từ đi ốt - BJT
2.2. Tạo cổng NOR từ BJT - FET ……
3. Mạch tạo thành cổng XOR
3.1. Tạo cổng XOR từ đi ốt - BJT

3.2. Tạo cổng XOR từ BJT - FET
4. Mạch so sánh
4.1. Sơ đồ nguyên lý mạch so sánh
4.2. Đặc điểm và nguyên lý làm việc ……
5. Mạch dùng cổng collector để hở
5.1. Sơ đồ nguyên lý
5.2. Đặc điểm và nguyên lý làm việc
6. Mạch dùng cổng ba trạng thái
6.1. Sơ đồ nguyên lý
6.2. Đặc điểm và nguyên lý làm việc
7. Thực hiện chuyển đổi từ cổng logic này sang cổng lô gic khác
7.1. Nguyên tắc chuyển đổi
7.2. Thực hành chuyển đổi
Bài 4: **Mạch mã hoá**
1. Khái niệm mạch mã hóa
1.1. Khái niệm
1.2. Nguyên tắc và các loại mạch mã
2. Mạch mã hóa 8/3 dùng các cổng logic cơ bản
2.1. Phương pháp mã 8/3
2.2. Mạch mã hóa 8/3 dùng cổng Logic cơ bản
2.3. Lắp ráp và cân chỉnh mạch mã hóa 8/3.
3. Mạch mã hóa 10/4 dùng IC TTL.
3.1. Phương pháp mã hóa 10/4
3.2. Mạch mã hóa 10/4 dùng IC TTL
3.3. Lắp ráp và cân chỉnh mạch mã hóa 10/4 dùng IC TTL.
Bài 5: **Mạch giải mã**
1. Khái niệm mạch giải mã
1.1. Khái niệm
1.2. Nguyên tắc xây dựng và các loại mạch giải mã
2. Mạch giải mã 2/4 dùng các cổng logic cơ bản
2.1. Phương pháp giải mã 2/4
2.2. Mạch giải mã 2/4 dùng cổng Logic cơ bản
3. Mạch giải mã 4/10 dùng IC TTL.
3.1. Phương pháp giải mã 4/10
3.2. Mạch giải mã 4/10 dùng IC TTL
4. Mạch giải mã từ BCD thành 7 đoạn (LED) hiển thị.
4.1. Phương pháp giải mã từ BCD sang hiển thị bằng LED 7 đoạn.
4.2. Sơ đồ mạch dùng IC TTL
4.3. Lắp ráp và cân chỉnh mạch
Bài 6: **Mạch dồn kênh**
1. Khái niệm mạch dồn kênh
1.1. Khái niệm
1.2. Nguyên tắc và các loại mạch dồn kênh
2. Mạch dồn kênh 4 vào 1 ra.
2.1. Phương pháp thực hiện

2.2. Mạch điện thực hiện.
3. Mạch dồn kênh 8 vào 1 ra.
3.1. Phương pháp thực hiện.
3.2. Mạch điện thực hiện.
3.3. Lắp ráp và cân chỉnh mạch dồn kênh 8 đường vào 1 đường ra.
4. Một số ứng dụng của mạch dồn kênh
4.1. Bộ chọn dữ liệu (data selector)
4.2. Biến đổi dạng thông tin vào song song thành ra nối tiếp
Bài 7: **Mạch phân kênh**
1. Khái niệm mạch phân kênh (DEMUX)
1.1. Khái niệm
1.2. Nguyên tắc và các loại mạch phân kênh
2. Mạch phân kênh 2 đầu ra
2.1. Phương pháp thực hiện mạch phân kênh 2 đầu ra.
2.2. Sơ đồ thực hiện dùng cổng logic cơ bản
2.3. Lắp sơ đồ, kiểm tra và cân chỉnh.
3. Mạch phân kênh 8 ra dùng IC CMOS
3.1. Phương pháp xây dựng.
3.2. Các sơ đồ mạch dùng IC CMOS
3.3. Lắp sơ đồ và cân chỉnh
Bài 8: **Mạch logic tuần tự**
1. Giới thiệu mạch logic tuần tự
1.1 Định nghĩa và phân loại
1.2. Mạch R - S Flip – Flop
1.3. Mạch J - K Flip – Flop
1.4. Mạch D Flip – Flop
1.5. Mạch T Flip – Flop
2. Một số mạch chuyển đổi và ứng dụng của FLIP-FLOP
2.1. Một số mạch chuyển đổi thông dụng.
2.2 Một số ứng dụng của FF
3. Mạch đếm (đếm thập phân, nhị phân)
3.1. Mạch đếm Mô đun
3.2. Mạch đếm không đồng bộ, đồng bộ.
3.3. Mạch đếm vòng
3.4. Lắp ráp và cân chỉnh một số mạch đếm thông dụng dùng FF.
4. Các mạch ghi dịch dữ liệu
4.1 Định nghĩa, cấu tạo, phân loại và ứng dụng
4.2 Thanh ghi dịch phải
4.3 Thanh ghi dich trái
4.4. Lắp ráp và cân chỉnh thanh ghi 4 bít dịch phải sử dụng FF.
Bài 9: **Mạch nhớ ROM**
1. Khái niệm chung về ROM
1.1. Khái niệm
1.2. Phân loại
2. Cấu trúc chung của ROM và các dang thường dùng

2.1. Sơ đồ khối của ROM
2.2. Các dạng thường thường gặp và ứng dụng
3. Lắp ráp và cân chỉnh ROM
3.1. Chuẩn bị vật tư và dụng cụ
3.2. Lắp mạch
3.3. Thực hành đo và cân chỉnh.
Bài 10: **Mạch nhớ RAM**
1. Khái niệm chung về RAM
1.1. Khái niệm
1.2. Phân loại
2. Cấu trúc chung của RAM
2.1. Cấu trúc chung của RAM
2.2. Các dạng RAM thường thường gặp và ứng dụng
3. Lắp ráp và cân chỉnh RAM
3.1. Chuẩn bị vật tư và dụng cụ
3.2. Lắp mạch
3.3. Thực hành đo và cân chỉnh.
Bài 11: **Mạch chuyển đổi A/D**
1. Khái niệm.
1.1. Khái niêm chung
1.2. Các tham số cơ bản
1.3. Nguyên lý hoạt động
2. Các phương pháp chuyển đổi
2.1. Chuyển đổi A/D theo phương pháp song song
2.2. Chuyển đổi A/D theo phương pháp nối tiếp theo mã nhị phân
3. Lắp ráp và cân chỉnh bộ A/D
3.1. Chuẩn bị vật tư và dụng cụ
3.2. Lắp mạch
3.3. Thực hành đo và cân chỉnh
Bài 12: **Mạch chuyển đổi D/A**
1. Khái niệm chung
1.1. Khái niệm, các tham số cơ bản
1.2. Nguyên lý hoạt động
2. Các phương pháp chuyển đổi
2.1 Chuyển đổi D/A theo phương pháp thang điện trở
2.2. Chuyển đổi D/A theo phương pháp mạch điện trở
2.3. Chuyển đổi D/A theo phương pháp mã hoá Shannom
3. Lắp ráp và cân chỉnh bộ A/D
3.1. Chuẩn bị vật tư và dụng cụ và lắp mạch
3.3. Thực hành đo và cân chỉnh

MÔ ĐUN: KỸ THUẬT SỐ
Mã mô đun: MĐ 26

Vị trí, tính chất, ý nghĩa, vai trò mô đun:

Trong các thiết bị điện - điện tử sử dụng trong dân dụng và công nghiệp hiện đại người ta thường dùng các mạch điện tử - số. Đặc trưng của mạch điện tử - số là các tín hiệu đầu vào và đầu ra các cổng logic, các mạch tích hợp, các bộ ghi dịch hoặc các bộ đếm, bộ chuyển đổi ADC hoặc DAC, bộ nhớ... đều là các tín hiệu số. Các mạch điện tử - số cũng được hoạt động đúng chức năn là do các mạch số cơ bản, tổ hợp và được đồng bộ nhờ các xung nhịp.

Việc nghiên cứu các khái niệm về tín hiệu số, mạch số, thiết kế và lắp ráp, sửa chữa, bảo dưỡng các mạch số là công việc hết sức cần thiết không thể thiếu của những người làm việc trong các lĩnh vực điện - điện tử nói chung và điện dân dụng nói riêng. Khi có kiến thức vững chắc và có tay nghề đã được rèn luyện để lắp ráp, sửa chữa các mạch số, góp phần không nhỏ vào sự vững vàng trong việc vận hành, bảo dưỡng và sửa chữa các thiết bị điện - điện tử dùng trong dân dụng và cả trong công nghiệp cũng như các thiết bị tự động điều khiển.

Mô đun này là một mô đun cơ sở chuyên môn nghề, được bố trí sau khi sinh viên học xong các môn học chung, các môn học/ mô đun: ATLĐ; Mạch điện; Vẽ điện; Vật liệu điện; Kỹ thuật điện tử cơ bản; Khí cụ điện hạ thế; Đo lường điện và không điện; Kỹ thuật xung.

Mục tiêu:

*Về kiến thức:
- Trình bày được cấu tạo và một số ứng dụng của cổng logic cơ bản.
- Trình bày được cấu trúc và sự hoạt động của các mạch FF, mã hoá, giải mã, dồn kênh, phân kênh, mạch đếm, ghi dịch, mạch chuyển đổi A/D, D/A, các bộ nhớ ROM và RAM.

* Về kỹ năng:
- Lắp ráp và cân chỉnh được các mạch ứng dụng cổng logic cơ bản.
- Lắp ráp và sửa chữa được các mạch: Mã hoá, giải mã, MUX, DEMUX, mạch đếm, ghi dịch, mạch chuyển đổi A/D, D/A; các bộ nhớ ROM và RAM đơn giản.

* Về thái độ:
- Có tính tư duy, sáng tạo và đảm bảo an toàn, vệ sinh công nghiệp.

Nội dung của mô đun:

Số TT	Tên các bài trong mô đun	Thời gian			
		Tổng số	Lý thuyết	Thực hành/ Bài tập	Kiểm tra*
1	Cơ sở kỹ thuật số	4	2	2	0
2	Các phần tử lôgic cơ bản	4	2	2	0
3	Các phần tử lôgic thông dụng	6	3	3	0
4	Mạch mã hóa	6	3	3	0
5	Mạch giải mã	6	3	2	1
6	Mạch dồn kênh	5	2	3	0
7	Mạch phân kênh	5	2	3	0
8	Mạch lôgic tuần tự	8	4	3	1
9	Mạch nhớ ROM	4	2	2	0
10	Mạch nhớ RAM	4	2	2	0
11	Mạch chuyển đổi A/D	4	2	2	0
12	Mạch chuyển đổi D/A	4	2	1	1
	Cộng:	**60**	**29**	**28**	**3**

BÀI 1
CƠ SỞ KỸ THUẬT SỐ
Mã bài: MĐ26.01

Giới thiệu:

Để có được kiến thức về kỹ thuật số, việc hiểu rõ khái niệm các loại tín hiệu số, mạch số và các kiến thức về đại số logic để hiểu được các khai sinệm về mã hóa. Bài học "Cơ sở kỹ thuật số" giới thiệu về các khái niệm của tín hiệu số, mạch số, khái niệm về hệ đếm, mã và những kiến thức cơ bản về đại số logic. Bài học là cơ sở hoàn hảo để tiếp thu các nội dung kiến thức tiếp theo của mô đun.

Mục tiêu:

- Phân biệt được tín hiệu số, tín hiệu tương tự, với các tín hiệu khác.
- Trình bày được khái niệm mã và hệ đếm.
- Thực hiện được các phép toán cộng, trừ, nhân, chia số nhị phân và cách chuyển đổi giữa các hệ đếm.
- Giải thích được các hàm số logic.
- Chủ động, sáng tạo trong quá trình học tập.

Nội dung chính:

1. Khái niệm tín hiệu số, tín hiệu tương tự

Mục tiêu:

Trình bày lại được khái niệm về tín hiệu tương tự và tín hiệu số; các mạch tương tự và tín hiệu số.

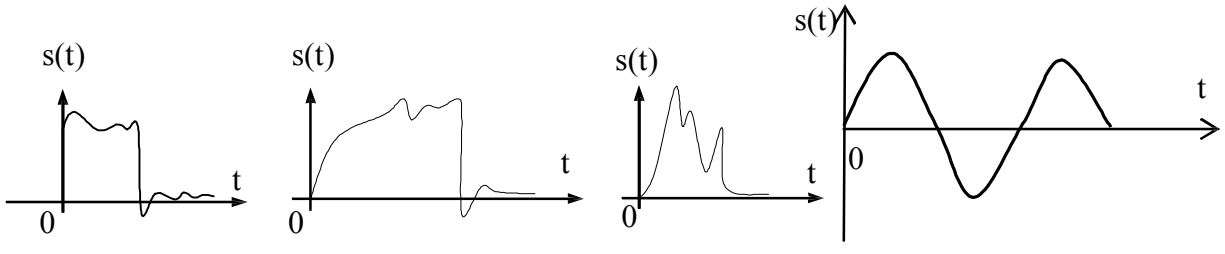
1.1. Tín hiệu tương tự:

TÝn hiÖu lµ biÓu hiÖn vËt lý cña tin tøc. Trong kü thuËt ®iÖn tö, tin tøc ®-îc biÕn ®æi thµnh c¸c dao ®éng ®iÖn tõ hoÆc ®iÖn tõ. Nh- vËy nãi c¸ch kh¸c tÝn hiÖu lµ c¸c dao ®éng ®iÖn - tõ cã chøa tin tøc. VÝ dô mirco biÕn ®æi tiÕng nãi thµnh mét dßng ®iÖn gÇn nh- liªn tôc theo thêi gian, gäi lµ tÝn hiÖu ©m tÇn. TÝn hiÖu ®iÖn tõ s¬ khai võa nãi trªn ta gäi chung lµ tÝn hiÖu s¬ cÊp.

Khi nghiên cứu tín hiệu người ta thường biểu diễn nó là một hàm của biến thời gian s(t) hoặc của biến tần số s(f). Tuy nhiên biểu diễn tín hiệu (điện áp hoặc dòng điện) là một hàm của biến thời gian s(t) là thuận lợi và thông dụng hơn cả.

1.1. Tín hiệu tương tự

Một tín hiệu biến thiên liên tục theo thời gian, người ta gọi là tín hiệu tương tự, tín hiệu tuyến tính hay tín hiệu analog (hình 1.1).



Hình 1.1. Tín hiệu tương tự

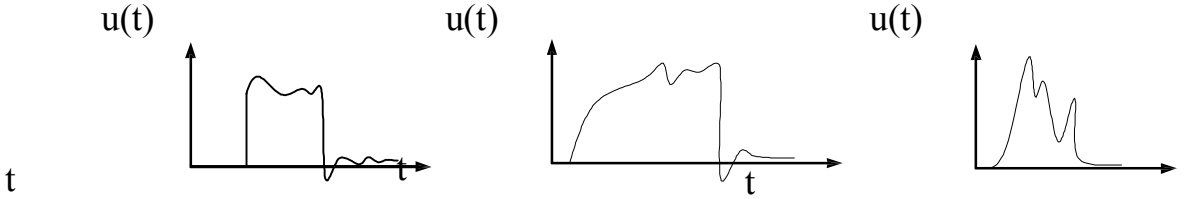
Trong điện tử, mạch điện để xử lý tín hiệu tương tự được gọi là mạch tương tự hay mạch analog.

1.2 Tín hiệu số

Tín hiệu xung điện là tín hiệu điện có giá trị biến đổi gián đoạn trong một khoảng thời gian rất ngắn có thể so sánh với quá trình quá độ của mạch điện.

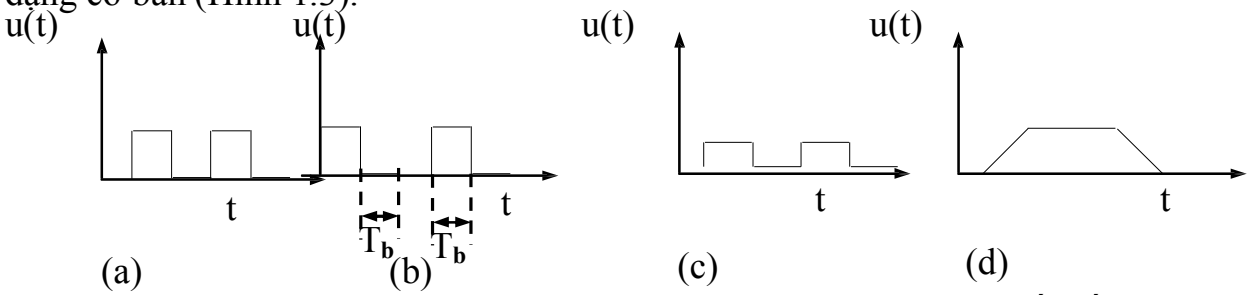
Xung điện trong kỹ thuật được chia làm 2 loại:

- Loại xung xuất hiện ngẫu nhiên trong mạch điện, ngoài mong muốn, được gọi là xung nhiễu, xung nhiễu thường có hình dạng bất kỳ (Hình 1.2).



Hình 1.2. Tín hiệu nhiễu

- Loại xung được tạo ra từ các mạch điện được thiết kế thường có một số dạng cơ bản (Hình 1.3).



Hình 1.3: Các dạng xung cơ bản của các mạch điện được thiết kế

Nếu ta biểu diễn tín hiệu xung điện áp là hàm u(t), trong đó t là biến thời gian thì tín hiệu có thể là tuần hoàn hoặc không tuần hoàn.

$$u(t) = u(t + nT); \quad n = 0, \pm 1, \pm 2 \ ... \qquad (1.1.)$$

Khi u(t) thoả mãn điều kiện (1.1) ở mọi thời điểm t thì u(t) là một tín hiệu tuần hoàn với chu kỳ T (ở đây T là khoảng thời gian nhỏ nhất để tín hiệu u(t) lặp lại quá trình trước đó cả về chiều và giá trị).

Nếu không tìm được một giá trị hữu hạn của T thoả mãn (1.1) tức là T tiến tới vô cùng ($T \to \infty$) thì u(t) sẽ là tín hiệu không tuần hoàn.

Trong điện tử, mạch điện để xử lý tín hiệu xung được gọi là mạch xung.

- Một dạng đặc biệt của tín hiệu xung, khi tín hiệu được biểu diễn bằng những xung vuông mà tất cả các khoảng thời gian có xung hay không có xung tín hiệu đều bằng một số nguyên lần khoảng thời gian T_b - hình 1.3.(b), người ta gọi đó là tín hiệu số hay tín hiệu digital và T_b được gọi là chu kỳ bít.

Trong điện tử, mạch điện để xử lý tín hiệu số được gọi là mạch số hay mạch digital.

2. Khái niệm mã và hệ đếm

Mục tiêu:

- Trình bày được khái niệm về một số hệ đếm thông dụng.
- Trình bày được khái niệm về mã và một số loại mã thông dụng.

2.1. Các hệ đếm thông dụng.

Thực tế hiện nay có 2 loại hệ đếm:

- Hệ đếm theo vị trí: Là hệ đếm mà trong đó giá trị số lượng của chữ số không chỉ phụ thuộc vào giá trị của nó, mà còn phụ thuộc vào vị trí đứng của nó trong con số (trọng số). Ví dụ: 1999; 2012 …
- Hệ đếm không theo vị trí: Là hệ đếm mà trong đó giá trị số lượng của chữ số không phụ thuộc vào vị trí của nó đứng trong con số. Đó là những số trong hệ La mã: I, II, III, ... , X, L, C, D, M

Ví dụ số 1087 trong hệ đếm theo vị trí, sẽ được viết trong hệ đếm không theo vị trí là MLXXXVII.

Thực tế hệ đếm theo vị trí là hệ đếm thông dụng nên ta chỉ xét các hệ đếm theo vị trí.

2.1.1. Hệ đếm cơ số 10 (hệ thập phân)

Hệ thập phân là hệ thống số rất quen thuộc trong đời sống của con người. Hệ đếm sử dụng 10 chữ số đầu tiên của dãy số tự nhiên 0, 1, 2, ..., 9 cùng với trọng số tương ứng của chúng để biểu diễn tất cả các số còn lại trong hệ đếm.

Để chỉ một số hệ đếm 10 (hệ đếm X) trong giáo trình sẽ dụng chỉ số 10 (X) phía dưới con số.

Dưới đây là vài ví dụ số thập phân:

$N_{10} = 1998_{10} = 1x10^3 + 9x10^2 + 9x10^1 + 8x10^0 = 1x1000 + 9x100 + 9x10 + 8x1$

$N_{10} = 3,14_{10} = 3x10^0 + 1x10^{-1} + 4x10^{-2} = 3/1 + 1/10 + 4/100$

2.1.2. Hệ đếm cơ số 2 (hệ nhị phân)

Hệ đếm nhị phân còn gọi là hệ đếm cơ số 2 là hệ đếm mà trong đó người ta chỉ sử dụng hai kí hiệu 0 và 1 để biểu diễn tất cả các số. Hai ký hiệu đó gọi chung là bit hoặc digit và nó đặc trưng cho mạch điện tử có hai trạng thái ổn định hay còn gọi là 2 trạng thái bền FLIP- FLOP (ký hiệu là FF).

Một nhóm 4 bít gọi là nibble.

Một nhóm 8 bít gọi là byte.

Nhóm nhiều bytes gọi là từ (word).

Xét số nhị phân 4 bít: $a_3 a_2 a_1 a_0$. Biểu diễn dưới dạng đa thức theo cơ số của nó là:

$a_3 a_2 a_1 a_0 = a_3.2^3 + a_2.2^2 + a_1.2^1 + a_0.2^0$

Trong đó:

- $2^0, 2^1, 2^2, 2^3$ (hay 1, 2, 4, 8) được gọi là các trọng số.
- a_0 được gọi là bit có trọng số nhỏ nhất, hay còn gọi bit có ý nghĩa nhỏ nhất (LSB: Least Significant Bit) .

- a_3 được gọi là bit có trọng số lớn nhất, hay còn gọi là bít có ý nghĩa lớn nhất (MSB: Most Significant Bit).

Ví dụ: $N_2 = 11011 = 1x2^4 + 1.2^3 + 0x2^2 + 1x2^1 + 1x2^0$

$= 1x16 + 1x8 + 0x\ 4 + 1x2 + 1x1$

Hoặc các số khác: 11100011; 1010101111; …

2.1.3. Hệ đếm cơ số 8 (hệ bát phân - Hệ Octa)

Hệ bát phân gồm tám số trong tập hợp $S_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$. Hay nói cách khác: Hệ bát phân là hệ đếm mà trong đó người ta dùng 8 chữ số đầu tiên của dãy số tự nhiên 0, 1, 2, 3, 4, 5, 6, 7 để biểu diễn tất cả các số trong tự nhiên.

Số N trong hệ bát phân:

$N_8 = (a_n a_{n-1} a_{n-2} . . . a_i . . . a_0 , a_{-1} a_{-2} . . . a_{-m})_8$ (với $a_i \in S_8$)

Có giá trị là:

$N_8 = a_n 8^n + a_{n-1} 8^{n-1} + a_{n-2} 8^{n-2} + . . + a_i 8^i . . . + a_0 8^0 + a_{-1} 8^{-1} + a_{-2} 8^{-2} + . . . + a_{-m} 8^{-m}$

Ví dụ: $N_8 = 1307,1_8 = 1x8^3 + 3x8^2 + 0x8^1 + 7x8^0 + 1x8^{-1} = 711,125_{10}$

2.1.4. Hệ đếm cơ số 16 (hệ thập lục phân - Hệ Hexa)

Hệ thập lục phân được dùng rất thuận tiện để con người giao tiếp với máy tính, hệ này gồm mười sáu số trong tập hợp S_{16}={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }; trong đó A tương đương với 10_{10}, B =11_{10}, ... , F=15_{10}.

Số N trong hệ thập lục phân:

$N_{16} = (a_n a_{n-1} a_{n-2} . . . a_i . . . a_0 , a_{-1} a_{-2} . . . a_{-m})_{16}$ (với $a_i \in S_{16}$)

Có giá trị là:

$N_{16} = a_n 16^n + a_{n-1} 16^{n-1} + a_{n-2} 16^{n-2} + . . + a_i 16^i . . . + a_0 16^0 + a_{-1} 16^{-1} + a_{-2} 16^{-2} + . . . + a_{-m} 16^{-m}$

Người ta thường dùng chữ H (hay h) sau con số để chỉ số thập lục phân.

Ví dụ: $N_{16} = 20EA,8H = 20EA,8_{16} = 2x16^3 + 0x16^2 + 14x16^1 + 10x16^0 + 8x16^{-1}$

$= 4330,5_{10}$

2.1.5. Biến đổi qua lại giữa các hệ thống số

Khi đã có nhiều hệ thống số, việc xác định giá trị tương đương của một số trong hệ này so với hệ kia là cần thiết. Trong phạm vi chương trình mô đun này, ta xét phương pháp biến đổi qua lại giữa các số trong bất cứ hệ nào sang hệ 10 và ngược lại.

1. Đổi một số từ hệ b sang hệ 10

Để đổi một số từ hệ b sang hệ 10 ta triển khai trực tiếp đa thức của b

Một số N trong hệ b:

$N_b = (a_n a_{n-1} a_{n-2} . . . a_i . . . a_0 , a_{-1} a_{-2} . . . a_{-m})_b$ với $a_i \in S_b$

Có giá trị tương đương trong hệ 10 là:

$N_b = a_n b^n + a_{n-1} b^{n-1} + . . . + a_i b^i + . . . + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + . . . + a_{-m} b^{-m}$.

Ví dụ:

* Đổi số $10110,11_2$ sang hệ 10

$10110,11_2 = 1x2^4 + 0 + 1x2^2 + 1x2 + 0 + 1x2^{-1} + 1x2^{-2} = 22,75_{10}$

* Đổi số $4BE,ADH$ sang hệ 10

$4BE,ADH = 4x16^2 + 11x16^1 + 14x16^0 + 10x16^{-1} + 13x16^{-2} = 1214,675_{10}$

2. Đổi một số từ hệ 10 sang hệ b

Đây là bài toán tìm một dãy ký hiệu cho số N hệ 10 viết trong hệ b.

Tổng quát, một số N cho ở hệ 10, viết sang hệ b có dạng:

$N_{10} = (a_n a_{n-1} \ldots a_0, a_{-1} a_{-2} \ldots a_{-m})_b = (a_n a_{n-1} \ldots a_0)_b + (0,a_{-1} a_{-2} \ldots a_{-m})_b$

Trong đó

$(a_n a_{n-1} \ldots a_0)_b$ = PE(N) là phần nguyên của N

và $(0,a_{-1} a_{-2} \ldots a_{-m})_b$ = PF(N) là phần lẻ của N

Phần nguyên và phần lẻ được biến đổi theo hai cách khác nhau:

- Phần nguyên:

Giá trị của phần nguyên xác định nhờ triển khai:

$PE(N) = a_n b^n + a_{n-1} b^{n-1} + \ldots + a_1 b^1 + a_0 b^0$

Hay có thể viết lại

$PE(N) = (a_n b^{n-1} + a_{n-1} b^{n-2} + \ldots + a_1)b + a_0$

Với cách viết này ta thấy nếu chia PE(N) cho b, ta được thương số là PE'(N) $= (a_n b^{n-1} + a_{n-1} b^{n-2} + \ldots + a_1)$ và số dư là a_0.

Vậy số dư của lần chia thứ nhất này chính là chữ số có trọng số nhỏ nhất (a_0) của phần nguyên.

Lặp lại bài toán chia PE'(N) cho b:

$PE'(N) = a_n b^{n-1} + a_{n-1} b^{n-2} + \ldots + a_1 = (a_n b^{n-2} + a_{n-1} b^{n-3} + \ldots + a_2)b + a_1$

Ta được số dư thứ hai, chính là chữ số có trọng số lớn hơn kế tiếp (a_1) và thương số là $PE''(N) = a_n b^{n-2} + a_{n-1} b^{n-3} + \ldots + a_2$.

Tiếp tục bài toán chia thương số có được cho b, cho đến khi thương số là 0 và được số dư của phép chia cuối cùng, đó chính là chữ số có trọng số lớn nhất (a_n)

- Phần lẻ:

Giá trị của phần lẻ xác định bởi:

$PF(N) = a_{-1} b^{-1} + a_{-2} b^{-2} + \ldots + a_{-m} b^{-m}$

Hay viết lại

$PF(N) = b^{-1} (a_{-1} + a_{-2} b^{-1} + \ldots + a_{-m} b^{-m+1})$

Nhân PF(N) với b, ta được :

$bPF(N) = a_{-1} + (a_{-2} b^{-1} + \ldots + a_{-m} b^{-m+1}) = a_{-1} + PF'(N)$.

Vậy lần nhân thứ nhất này ta được phần nguyên của phép nhân, chính là số mã có trọng số lớn nhất của phần lẻ (a_{-1}) (số a_{-1} này có thể vẫn là số 0).

PF'(N) là phần lẻ xuất hiện trong phép nhân.

Tiếp tục nhân PF'(N) với b, ta tìm được a_{-2} và phần lẻ PF"(N).

Lặp lại bài toán nhân phần lẻ với b cho đến khi kết quả có phần lẻ bằng không, ta sẽ tìm được dãy số ($a_{-1}a_{-2}...a_{-m}$).

Chú ý: Phần lẻ của số N khi đổi sang hệ b có thể gồm vô số số hạng (do kết quả của phép nhân luôn khác 0), điều này có nghĩa là ta không tìm được một số trong hệ b có giá trị đúng bằng phần lẻ của số thập phân, vậy tùy theo yêu cầu về độ chính xác khi chuyển đổi mà người ta lấy một số số hạng nhất định.

Ví dụ:

* Đổi $25,3_{10}$ sang hệ nhị phân

Phần nguyên: 25 : 2 = 12 dư 1 $\Rightarrow a_0 = 1$

12 : 2 = 6 dư 0 $\Rightarrow a_1 = 0$

6 : 2 = 3 dư 0 $\Rightarrow a_2 = 0$

3 : 2 = 1 dư 1 $\Rightarrow a_3 = 1$

1 : 2 = 0 dư 1 $\Rightarrow a_4 = 1$

Vậy PE(N) = 11001

Phần lẻ: 0,3 x 2 = 0,6 $\Rightarrow a_{-1} = 0$

0,6 x 2 = 1,2 $\Rightarrow a_{-2} = 1$

0,2 x 2 = 0,4 $\Rightarrow a_{-3} = 0$

0,4 x 2 = 0,8 $\Rightarrow a_{-4} = 0$

0,8 x 2 = 1,6 $\Rightarrow a_{-5} = 1 \ldots$

Nhận thấy kết quả của các bài toán nhân luôn khác không, do phần lẻ của lần nhân cuối cùng là 0,6, đã lặp lại kết quả của lần nhân thứ nhất, như vậy bài toán không thể kết thúc với kết quả đúng bằng 0,3 của hệ 10.

Giả sử bài toán yêu cầu lấy 5 số lẻ thì ta có thể dừng ở đây.

Vậy PF(N) = 0,01001.

Kết quả cuối cùng là:

$25,3_{10} = 11001,01001_2$

* Đổi $1376,85_{10}$ sang hệ thập lục phân

Phần nguyên: 1376 : 16 = 86 số dư = 0 $\Rightarrow a_0 = 0$

86 : 16 = 5 số dư = 6 $\Rightarrow a_1 = 6$

5 : 16 = 0 số dư = 5 $\Rightarrow a_2 = 5$

$1376_{10} = 560H$

Phần lẻ: 0,85 * 16 = 13,6 $\Rightarrow a_{-1} = 13_{10} = DH$

0,6 * 16 = 9,6 $\Rightarrow a_{-2} = 9$

0,6 * 16 = 9,6 $\Rightarrow a_{-3} = 9$

Đây là số thập phân tuần hoàn, nên nếu chỉ cần lấy 3 số lẻ: $0,85_{10} = 0,D99H$

Và kết quả cuối cùng:
$1376,85_{10}$ = 560,D99H

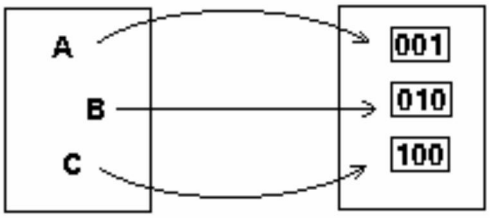
2.2. Mã và các loại mã thông dụng.

2.2.1. Khái niệm về mã

Trong đời sống hàng ngày, con người giao tiếp với nhau thông qua một hệ thống ngôn ngữ qui ước, nhưng trong máy tính chỉ xử lý các dữ liệu nhị phân. Do đó, một vấn đề đặt ra là làm thế nào tạo ra một giao diện dễ dàng giữa người và máy tính, nghĩa là máy tính thực hiện được những bài toán do con người đặt ra. Để thực hiện điều đó, người ta đặt ra vấn đề về mã hóa dữ liệu. Như vậy trong kỹ thuật số, mã hóa là quá trình biến đổi những ký hiệu quen thuộc của con người sang những ký hiệu quen thuộc với máy tính.

Nói chung, mã hóa là gán một ký hiệu cho một đối tượng để thuận tiện cho việc thực hiện một yêu cầu cụ thể nào đó.

Một cách toán học, mã hóa là một phép áp một đối một từ một **tập hợp nguồn** vào một tập hợp khác gọi là **tập hợp đích**.



Tập hợp nguồn có thể là tập hợp các số, các ký tự, dấu, các lệnh dùng trong truyền dữ liệu . . . và tập hợp đích thường là tập hợp chứa các tổ hợp thứ tự của các số nhị phân.

Một tổ hợp các số nhị phân tương ứng với một số được gọi là **từ mã**. Tập hợp các từ mã được tạo ra theo một qui luật cho ta một **bộ mã**. Việc chọn một bộ mã tùy vào mục đích sử dụng.

Ví dụ để biểu diễn các chữ và số, người ta có **mã ASCII** (American Standard Code for Information Interchange), **mã Baudot, EBCDIC** . . .. Trong truyền dữ liệu ta có **mã dò lỗi, dò và sửa lỗi, mật mã** . . ..

Vấn đề ngược lại mã hóa gọi là **giải mã**.

Cách biểu diễn các số trong các hệ khác nhau cũng có thể được xem là một hình thức mã hóa, đó là các mã thập phân, nhị phân, thập lục phân . . . và việc chuyển từ mã này sang mã khác cũng thuộc loại bài toán mã hóa.

2.2.2. Một số loại mã thông dụng

2.2.2.1. Mã BCD

1. Khái niệm

Việc sử dụng các số nhị phân để mã hóa các số thập phân gọi là các số BCD (**B**inary **C**ode **D**ecimal: Số thập phân được mã hóa bằng số nhị phân).

Mã BCD dùng các **từ mã nhị phân (số nhị phân 4 bit)** có giá trị tương đương thay thế cho **từng số hạng** trong số thập phân.

Ví dụ:

Số 625_{10} có mã BCD là 0110 0010 0101.

Mã BCD dùng rất thuận lợi: Mạch điện tử đọc các số BCD và hiển thị ra bằng đèn bảy đoạn (led hoặc LCD) hoàn toàn giống như con người đọc và viết ra số thập phân.

Bảng mã BCD của 10 số đầu tiên hệ thập phân như ở bảng 1.1.

Bảng 1.1: Mã BCD

Số hệ 10	Mã BCD	Số hệ 10	Mã BCD
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

2. Phân loại

Khi sử dụng số nhị phân 4 bit để mã hóa các số thập phân tương ứng với $2^4 = 16$ tổ hợp mã nhị phân phân biệt.

Do việc chọn 10 tổ hợp trong 16 tổ hợp để mã hóa các ký hiệu thập phân từ 0 đến 9 mà trong thực tế xuất hiện nhiều loại mã BCD khác nhau.

Mặc dù tồn tại nhiều loại mã BCD khác nhau, nhưng trong thực tế người ta chia làm hai loại chính: BCD có trọng số và BCD không có trọng số.

a. Mã BCD có trọng số:

Gồm có mã BCD tự nhiên, mã BCD số học.

- Mã BCD tự nhiên đó là loại mã mà trong đó các trọng số thường được sắp xếp theo thứ tự tăng dần.

Ví dụ: Mã BCD 8421, mã BCD 5421

- Mã BCD số học là loại mã mà trong đó có tổng các trọng số luôn luôn bằng 9.

Ví dụ: Loại mã: BCD 2421, BCD 5121, BCD 8 4-2-1

Mã BCD số học có đặc trưng: Để tìm từ mã thập phân của một số thập phân nào đó ta lấy bù (đảo) từ mã nhị phân của số bù 9 tương ứng.

Ví dụ: Mã BCD của 3 là 0011

Mà số 6 là bù 9 của 3: 6 là 1100

Lấy nghịch đảo ta có: 0011 → 3

Vậy, đặc trưng của mã BCD số học là có tính chất đối xứng qua một đường trung gian.

b. Mã BCD không có trọng số:

Là loại mã không cho phép phân tích thành đa thức theo cơ số của nó.

Ví dụ: Mã Gray, Mã Gray thừa 3.

Đặc trưng của mã Gray là loại bộ mã mà trong đó hai từ mã nhị phân đứng kế tiếp nhau bao giờ cũng chỉ khác nhau 1 bit.

Ví dụ: Mã Gray:

1 → 0001

2 → 0011

3 → 0010
4 → 0110

Còn đối với mã BCD 8421 như ở bảng 1.1.:

Các bảng dưới đây trình bày một số loại mã thông dụng:

Bảng 1.2: Các mã BCD tự nhiên.

BCD 8421				BCD 5421				BCD quá 3				Số thập phân
a_3	a_2	a_1	a_0	b_3	b_2	b_1	b_0	c_3	c_2	c_1	c_0	
0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	1	0	0	0	1	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1	0	1	2
0	0	1	1	0	0	1	1	0	1	1	0	3
0	1	0	0	0	1	0	0	0	1	1	1	4
0	1	0	1	1	0	0	0	1	0	0	0	5
0	1	1	0	1	0	0	1	1	0	0	1	6
0	1	1	1	1	0	1	0	1	0	1	0	7
1	0	0	0	1	0	1	1	1	0	1	1	8
1	0	0	1	1	1	0	0	1	1	0	0	9

Bảng 1.3: Các mã BCD số học

BCD 2421				BCD 5121				BCD 84-2-1				Số thập phân
a_3	a_2	a_1	a_0	b_3	B_2	b_1	b_0	c_3	c_2	c_1	c_0	
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	1	0	2
0	0	1	1	0	0	1	1	0	1	0	1	3
0	1	0	0	0	1	0	0	0	1	0	0	4
1	0	1	1	1	0	0	0	1	0	1	1	5
1	1	0	0	1	1	0	0	1	0	1	0	6
1	1	0	1	1	1	0	1	1	0	0	1	7
1	1	1	0	1	1	1	0	1	0	0	0	8
1	1	1	1	1	1	1	1	1	1	1	1	9

Bảng 1.4: BCD tự nhiên và mã Gray.

BCD 8421				BCD quá 3				Mã Gray				Gray quá 3				Số thập phân
a_3	a_2	a_1	a_0	c_3	c_2	c_1	c_0	G_3	G_2	G_1	G_0	g_3	g_2	g_1	g_0	
0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0
0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	0	1
0	0	1	0	0	1	0	1	0	0	1	1	0	1	1	1	2
0	0	1	1	0	1	1	0	0	0	1	0	0	1	0	1	3
0	1	0	0	0	1	1	1	0	1	1	0	0	1	0	0	4
0	1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	5
0	1	1	0	1	0	0	1	0	1	0	1	1	1	0	1	6
0	1	1	1	1	0	1	0	0	1	0	0	1	1	1	1	7
1	0	0	0	1	0	1	1	1	1	0	0	1	1	1	0	8
1	0	0	1	1	1	0	0	1	1	0	1	1	0	1	0	9

Do việc xuất hiện số BCD nên có hai cách nhập dữ liệu vào máy tính: nhập số nhị phân, nhập bằng mã BCD.

Để nhập số BCD thập phân hai chữ số thì máy tính chia số thập phân thành các đềcác và mỗi đềcác được biểu diễn bằng số BCD tương ứng.

Ví dụ: 11 (thập phân) có thể được nhập vào máy tính theo 2 cách:

- Số nhị phân: 1011
- Mã BCD : 0001 0001

2.2.2.2. Mã Gray

Mã **Gray** hay còn gọi là mã **cách khoảng đơn vị**.

Nếu quan sát thông tin ra từ một máy đếm đang đếm các sự kiện tăng dần từng đơn vị, ta sẽ được các số nhị phân dần dần thay đổi. Tại thời điểm đang quan sát có thể có những lỗi rất quan trọng. Ví dụ giữa hai số 7(0111) và 8 (1000), tất cả các phần tử nhị phân đều phải thay đổi trong quá trình đếm. Thực tế, sự giao hoán này không bắt buộc xảy ra đồng thời, ta có thể có các trạng thái liên tiếp sau: 0111 → 0110 → 0100 → 0000 → 1000

Trong một quan sát ngắn các kết quả thấy được khác nhau.

Để tránh hiện tượng này, người ta cần mã hóa mỗi số hạng sao cho hai số liên tiếp nhau chỉ khác nhau một phần tử nhị phân (1 bit) gọi là mã **cách khoảng đơn vị** hay mã **Gray**.

Tính kề nhau của các tổ hợp mã Gray (tức các mã liên tiếp chỉ khác nhau một bit) được dùng rất có hiệu quả để **rút gọn hàm logic tới mức tối giản**.

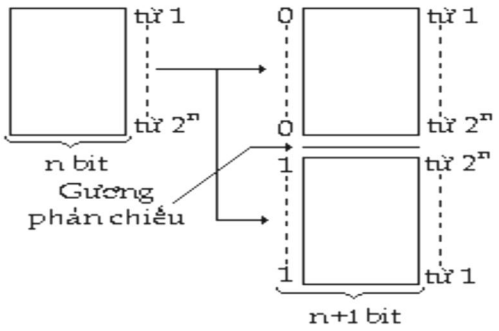
Ngoài ra, mã Gray còn được gọi là mã **phản chiếu** (do tính đối xứng của các số hạng trong tập hợp mã, giống như phản chiếu qua gương)

Người ta có thể thiết lập mã Gray bằng cách dựa vào tính đối xứng này:

- Giả sử ta đã có tập hợp 2^n từ mã của số n bit thì có thể suy ra tập hợp 2^{n+1} từ mã của số (n+1) bit bằng cách:
- Viết ra 2^n từ mã theo thứ tự từ nhỏ đến lớn

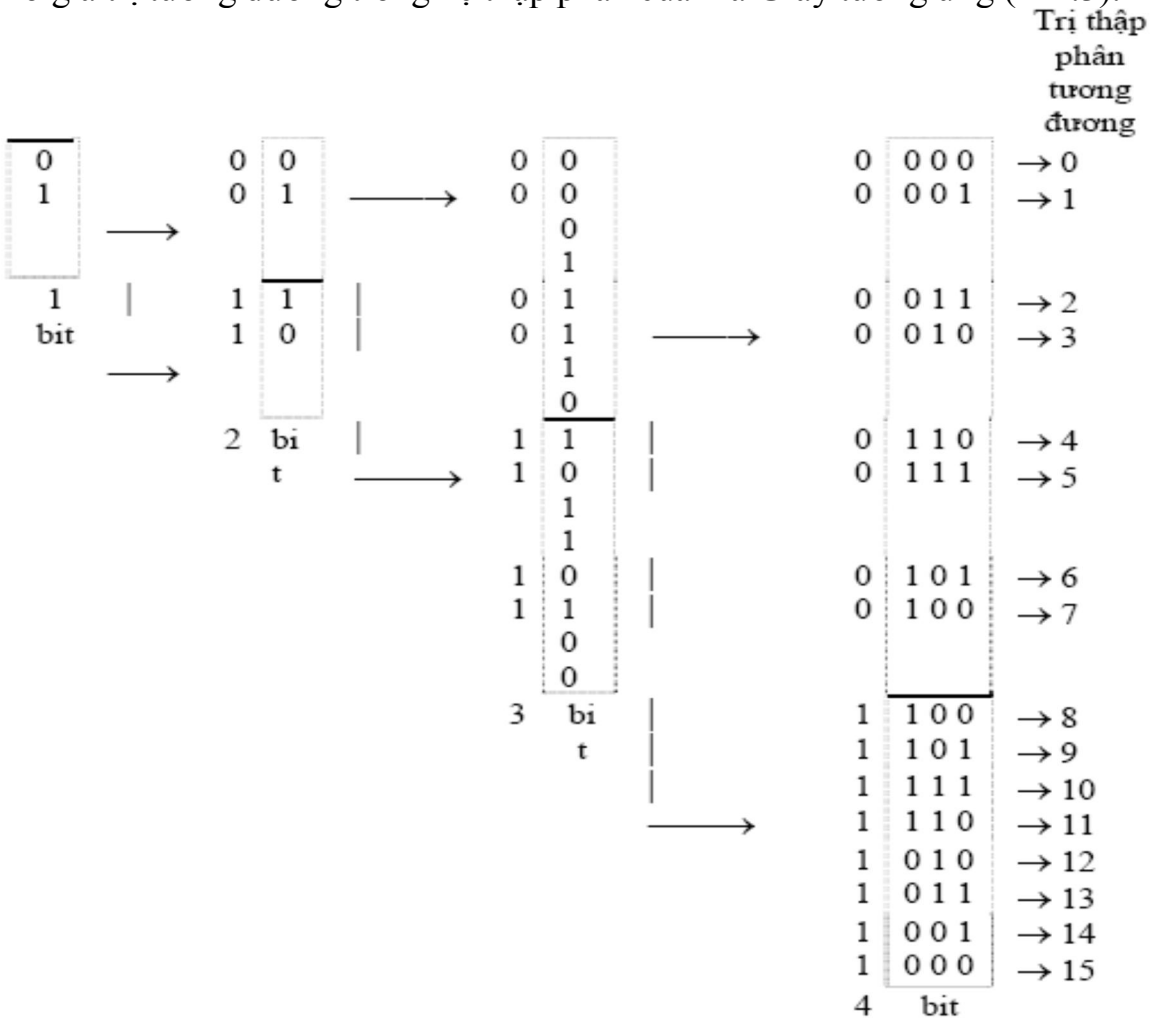
- Thêm số 0 vào trước tất cả các từ mã đã có để được một phần của tập hợp từ mã mới

- Phần thứ hai của tập hợp gồm các từ mã giống như phần thứ nhất nhưng trình bày theo thứ tự ngược lại (giống như phản chiếu qua gương) và phía trước thêm vào số 1 thay vì số 0



Để thiết lập mã Gray của số nhiều bit ta có thể thực hiện các bước liên tiếp từ tập hợp đầu tiên của số một bit (gồm hai bit 0, 1).

Dưới đây là các bước tạo mã Gray của số 4 bit. Cột bên phải của bảng mã 4 bit cho giá trị tương đương trong hệ thập phân của mã Gray tương ứng (H 1.3).



Nhận xét các bảng mã của các số Gray (1 bit, 2 bit, 3 bit và 4 bit) ta thấy các số gần nhau luôn luôn khác nhau một bit, ngoài ra, trong từng bộ mã, các số đối xứng nhau qua gương cũng khác nhau một bit.

3. Thực hiện các phép tính và chuyển đổi mã

Mục tiêu:

- Trình bày được các phép tính nhị phân

- Chuyển đổi qua lại được một số hệ mã thông dụng.

3.1. Các phép tính trong kỹ thuật số.

3.1.1. Các phép tính trên số nhị phân:

a. Phép cộng

Để cộng hai số nhị phân người ta dựa trên quy tắc cộng như sau:

0 + 0 = 0 nhớ 0; 0 + 1 = 1 nhớ 0; 1 + 0 = 1 nhớ 0; 1 + 1 = 0 nhớ 1

Ví dụ:

3	⇒ Số nhị phân	0011	
+		+	
2	⇒ Số nhị phân	0010	
5		0101	⇒ Số thập phân: 5

b. Phép trừ

Để trừ hai số nhị phân người ta dựa trên quy tắc trừ như sau:

0 - 0 = 0 mượn 0

0 - 1 = 1 mượn 1

1 - 0 = 1 mượn 0

1 - 1 = 0 mượn 0

Ví dụ:

7	→	0111	
- 5	→	- 0101	
2	→	0010	$= 1.2^2 + 0.2^1 + 1.2^0 = 2$

c. Phép nhân

Để nhân hai số nhị phân người ta dựa trên quy tắc nhân như sau:

0 . 0 = 0

0 . 1 = 0

1 . 0 = 0

1 . 1 = 1

Ví dụ:

7	→	0111	
x 5	→	x 0101	
35		0111	
		0000	
		0111	
		0000	
		0100011	$= 1.2^5 + 1.2^1 + 1.2^0 = 35$

d. Phép chia

Để chia hai số nhị phân người ta dựa trên quy tắc chia như sau:

0 : 0 = 0

1 : 1 = 1

Ví dụ: 10 | 5 → 1010 | 101

2 — 101 — 10 = 2

00

0

3.1.2. Các phép tính trên số BCD:

a. Phép cộng

Số thập phân là 128 thì:

- Số nhị phân là: 10000000
- Số BCD là: 0001 0010 1000

Do số BCD chỉ có từ 0 đến 9 nên đối với những số thập phân lớn hơn, nó chia số thập phân thành nhiều đề - các, mỗi đề - các được biểu diễn bằng số BCD tương ứng.

+ 5 →	0101	+ 7 →	0111		
3 →	0011	5 →	0101		
8	1000	12	1100		
			+		
		Số hiệu chỉnh →	0110		
			0001	0010	
			1	2	

b. Phép trừ

Có thể tiến hành trừ trực tiếp từng đề - các theo nguyên tắc trừ số nhị phân, hoặc cộng lần lượt số bị trừ với số bù 1 và bù 2 của số trừ.

A - B

7 →	0111	
- 5 →	- 0101	
2	0010	

A + Bù 1 ; 2

0111	
+ 1010	← Bù 1 của 5
10001	
+ 1	← Bù 2 của 5
0010	

Bù 1 là bit 0 thành 1, bit 1 thành 0.

Bù 2 là bù 1 cộng thêm 1.

3.2. Chuyển đổi giữa các hệ mã thông dụng.

3.2.1. Chuyển mã BCD 8421 sang mã Gray:

Để chuyển mã BCD 8421 sang mã Gray người ta thực hiện bằng cách: các bit 0, 1 đứng sau bit 0 (ở mã BCD 8421) khi chuyển sang mã Gray thì được giữ nguyên; còn các bit 0, 1 đứng sau bit 1 (ở mã BCD 8421) khi chuyển sang mã Gray thì được đảo ngược lại, nghĩa là từ bit 1 thành bit 0 và bit 0 thành bit 1.

Có thể thấy quy luật chuyển đổi như trên bảng 1.4 mục 2.2.2.1.

Mã BCD 8421				Mã Gray			
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

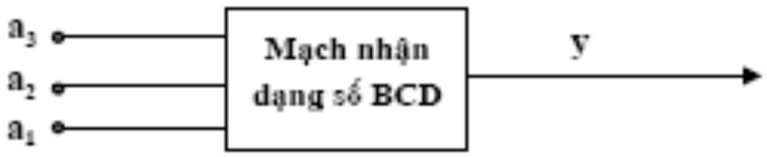
3.2.2. Chuyển mã Gray sang mã BCD 8421:

Để chuyển mã Gray sang mã BCD 8421 người ta thực hiện bằng cách: các bit 0, 1 đứng ngay sau bit 0 (ở mã Gray) khi chuyển sang mã BCD 8421 thì được giữ nguyên; còn các bit 0, 1 đứng ngay sau bit 1 (ở mã Gray) khi chuyển sang mã BCD 8421 thì được đảo ngược lại, nghĩa là từ bit 1 thành bit 0 và bit 0 thành bit 1; các bit kế tiếp lấy giá trị bit ngay trước đó làm cơ sở để chuyển đổi.

Ví dụ: Mã Gray có giá trị là **0111**; khi chuyển sang mã BCD 8421 thì bit thứ 2 giữ nguyên là 1; bít thứ 3 vì sau bit 1 nên đảo thành 0; bít thứ 4 vì đứng sau bit 3 đã đảo thành 0 nên giữ nguyên giá trị là 1; kết quả cho ta mã BCD 8421 của số đã cho là **0101**.

Tương tự: 1110 → 1011; 1111 → 1010 ...

3.2.3. Mạch nhận dạng số BCD 8421:



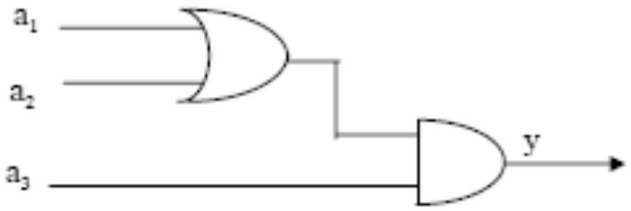
+ y = 1 → $a_3 a_2 a_1 a_0$ không phải số BCD 8421

+ y = 0 → $a_3 a_2 a_1 a_0$ là số BCD 8421

Nghĩa là: Để nhận dạng một số nhị phân 4 bit không phải là một số BCD 8421 thì đầu ra y = 1. Từ bảng mã BCD 8421 ta thấy: Nếu là số BCD 8421 thì khi bit a_3 bằng 1, bit a_1 hoặc a_2 không thể bằng 1. Vì vậy khi bit a_3 bằng 1 và bit a_1 hoặc a_2 bằng 1 thì số đã cho không phải là số BCD 8421. Từ đó:

Hàm số logic: $y = a_3 (a_1 + a_2) = a_3 a_1 + a_3 a_2$

Sơ đồ logic thực hiện:



4. Đại số logic

Mục tiêu:

Trình bày được các hàm logic cơ bản, các phương pháp biểu diễn hàm logic; tối giản (tối thiểu hóa) được các hàm logic không qua phức tạp.

4.1. Khái niệm về đại số logic

4.1.1. Khái niệm:

Trong mạch số, các tín hiệu thường cho ở hai mức điện áp 0 V và 5 V. Những linh kiện điện tử dùng trong mạch số làm việc ở chế độ khóa; nghĩa là chúng chỉ ở một trong hai trạng thái (tắt hoặc thông).

Do vậy để mô tả mạch số người ta dùng hệ nhị phân (Binary). Hai trạng thái trong mạch được mã hoá tương ứng là "1" hoặc "0". Hệ nhị phân thể hiện được trạng thái vật lý mà hệ thập phân không thể hiện được.

Đại số Bun (Boole) nghiên cứu mối liên hệ (các phép tính cơ bản) giữa các biến trạng thái (biến logic) chỉ nhận một trong hai giá trị "0" hoặc"1" và kết quả nghiên cứu là một hàm trạng thái (hàm logic) cũng chỉ nhận giá trị "0" hoặc"1".

Môn đại số mang tên người sáng lập ra nó - Đại số Bun (Boole) còn được gọi là đại số logic.

4.1.2. Một số định nghĩa:

- Biến logic: Đại lượng biểu diễn bằng ký hiệu nào đó chỉ lấy giá trị "1" hoặc "0".

- Hàm logic: Biểu diễn nhóm các biến logic liên hệ với nhau thông qua các phép toán logic, một hàm logic cho dù là đơn giản hay phức tạp cũng chỉ nhận giá trị hoặc là "1" hoặc là "0".

4.1.3 Các phép toán cơ bản của đại số Bun

Bởi vì các đại lượng chỉ có hai trạng thái nên đại số Bun rất khác đại số thường và dễ tính toán hơn. Ở đại số Bun không có phân số, số thập phân, số ảo, số phức, căn số… mà chỉ thực hiện chủ yếu 3 phép tính toán cơ bản sau:

- Phép OR (Hoặc; cộng)
- Phép AND (Và; nhân)
- Phép phủ định NOT

Các phép tính trên áp dụng cho logic 0 và 1.

4.2. Các hàm logic cơ bản

4.2.1. Hàm NOT (đảo, bù): $\boldsymbol{Y = \overline{A}}$

Bảng sự thật

A	$\boldsymbol{Y = \overline{A}}$
0	1
1	0

4.2.2. Hàm AND [tích logic, toán tử (.)] : **Y = A.B**

Bảng sự thật

A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

Nhận xét: Tính chất của hàm AND có thể được phát biểu như sau:
- Hàm AND của 2 (hay nhiều) biến chỉ có giá trị 1 khi tất cả các biến đều bằng 1. Hoặc:
- Hàm AND của 2 (hay nhiều) biến có giá trị 0 khi có một biến bằng 0.

4.2.3. Hàm OR [tổng logic, toán tử (+)] : Y = A + B

Bảng sự thật

A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Nhận xét: Tính chất của hàm OR có thể được phát biểu như sau:
- Hàm OR của 2 (hay nhiều) biến chỉ có giá trị 0 khi tất cả các biến đều bằng 0. Hoặc:
- Hàm OR của 2 (hay nhiều) biến có giá trị 1 khi có một biến bằng 1.

4.2.4.Hàm EX-OR (OR loại trừ) $Y = A \oplus B$

Bảng sự thật

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

Nhận xét: Một số tính chất của hàm EX - OR:
- Hàm EX - OR của 2 biến chỉ có giá trị 1 khi hai biến khác nhau và ngược lại. Tính chất này được dùng để so sánh 2 biến.
- Hàm EX - OR của 2 biến cho phép thực hiện cộng hai số nhị phân 1 bit mà không quan tâm tới số nhớ.
- Từ kết quả của hàm EX-OR 2 biến ta suy ra bảng sự thật cho hàm 3 biến

Y = A ⊕ B ⊕ C

A	B	C	Y = A ⊕ B ⊕ C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Trong trường hợp 3 biến (và suy rộng ra cho nhiều biến), hàm EX - OR có giá trị 1 khi số biến bằng 1 là số lẻ. Tính chất này được dùng để nhận dạng một chuỗi dữ liệu có số bit 1 là chẵn hay lẻ trong thiết kế mạch phát chẵn lẻ.

4.2.5. Tính chất của các hàm logic cơ bản:

4.2.5.1. Tính chất cơ bản:

– Có một phần tử trung tính duy nhất cho mỗi toán tử (+) và (.):
A + 0 = A ; 0 là phần tử trung tính của hàm OR
A . 1 = A ; 1 là phần tử trung tính của hàm AND
- Tính giao hoán:
A + B = B + A A .
B = B . A
- Tính phối hợp:
(A + B) + C = A + (B + C) = A + B + C
(A . B) . C = A . (B . C) = A . B . C
- Tính phân bố:
+ Phân bố đối với phép nhân: A . (B + C) = A . B + A . C
+ Phân bố đối với phép cộng: A + (B . C) = (A + B) . (A + C)
Phân bố đối với phép cộng là một tính chất đặc biệt của phép toán logic
- Không có phép tính lũy thừa và thừa số:
A + A + ... + A = A . A . . . . . A = A
- Tính bù:

$$A = \bar{\bar{A}}$$
$$A + \bar{A} = 1$$
$$A.\bar{A} = 0$$

4.2.5.2. Tính song đối (duality):

Tất cả biểu thức logic vẫn đúng khi thay phép toán (+) bởi phép (.) và 0 bởi 1 hay ngược lại. Điều này có thể chứng minh dễ dàng cho tất cả biểu thức ở trên.

Ví dụ : $A + B = B + A \Leftrightarrow A.B = B.A$

$A + \bar{A}.B = A + B \Leftrightarrow A.(\bar{A} + B) = A.B$

$A + 1 = 1 \Leftrightarrow A.0 = 0$

4.2.5.3. Định lý De Morgan

Định lý De Morgan được phát biểu bởi hai biểu thức:

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

Định lý De Morgan cho phép biến đổi qua lại giữa hai phép cộng và nhân nhờ vào phép đảo.

Định lý De Morgan được chứng minh bằng cách lập bảng sự thật cho tất cả trường hợp có thể có của các biến A, B, C với các hàm AND, OR và NOT của chúng.

Định lý De Morgan cho thấy các hàm logic không độc lập với nhau, chúng có thể biến đổi qua lại, sự biến đổi này cần có sự tham gia của hàm NOT. Kết quả là ta có thể dùng hàm (AND và NOT) hoặc (OR và NOT) để diễn tả tất cả các hàm.

Ví dụ:

Cho hàm số: $$Y = A.B + B.C + \overline{A}.C$$

- Chỉ dùng hàm AND và NOT để diễn tả hàm đã cho.

Chỉ cần đảo hàm Y hai lần, ta được kết quả:

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

- Nếu chỉ dùng hàm OR và NOT để diễn tả hàm trên làm như sau:

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

4.3. Biểu diễn hàm logic

Quy ước: Khi nghiên cứu một hệ thống logic, cần xác định qui ước logic. Qui ước này không được thay đổi trong suốt quá trình nghiên cứu. Người ta dùng 2 mức điện thế thấp và cao để gán cho 2 trạng thái logic 1 và 0. Qui ước **logic dương** gán điện thế thấp cho logic 0 và điện thế cao cho logic 1. Qui ước **logic âm** thì ngược lại.

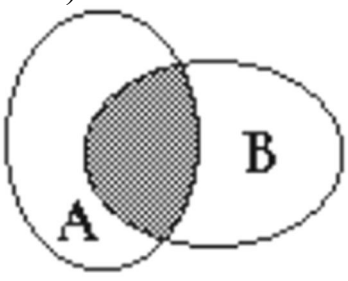
Trên thực tế thông thường chỉ sử dụng **Logic dương**.

Hàm logic thường được biểu diễn dưới các dạng sau:

4.3.1. Giản đồ Venn

Còn gọi là giản đồ Euler, đặc biệt dùng trong lãnh vực tập hợp. Mỗi biến logic chia không gian ra 2 vùng không gian con, một vùng trong đó giá trị biến là đúng (hay=1), và vùng còn lại là vùng phụ trong đó giá trị biến là sai (hay=0).

Ví dụ: Phần giao nhau của hai tập hợp con A và B (gạch chéo) biểu diễn tập hợp trong đó A và B là đúng (A AND B).



4.3.2. Bảng sự thật (Bảng chân lý)

Nếu hàm có n biến, bảng sự thật có n+1 cột và 2^n + 1 hàng. Hàng đầu tiên chỉ tên biến và hàm, các hàng còn lại trình bày các tổ hợp của n biến trong 2^n tổ hợp có thể có. Các cột đầu ghi giá trị của biến, cột cuối cùng ghi giá trị của hàm tương ứng với tổ hợp biến trên cùng hàng (gọi là trị riêng của hàm).

Ví dụ: Hàm OR của 2 biến A, B: f(A,B) = (A OR B) có bảng sự thật tương ứng.

A	**B**	**f(A,B) = A OR B**
0	0	0
0	1	1
1	0	1
1	1	1

4.3.3. Bảng (Bìa) Các - nô (Các - nô)

Đây là cách biểu diễn khác của bảng sự thật trong đó mỗi hàng của bảng sự thật được thay thế bởi một ô mà tọa độ (gồm hàng và cột) xác định bởi tổ hợp đã cho của biến.

Bảng Các - nô của n biến gồm 2^n ô. Giá trị của hàm được ghi tại mỗi ô của bảng. Bảng Các - nô rất thuận tiện để đơn giản hàm logic bằng cách nhóm các ô lại với nhau.

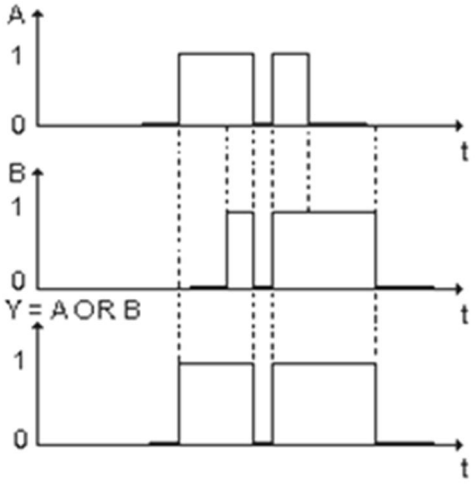
Ví dụ: Hàm OR ở trên được diễn tả bởi bảng Các - nô sau đây

A \ B	**0**	**1**
0	0	1
1	1	1

4.3.4. Giản đồ thời gian:

Dùng để diễn tả quan hệ giữa các hàm và biến theo thời gian, đồng thời với quan hệ logic.

Ví dụ: Giản đồ thời gian của hàm OR của 2 biến A và B, tại những thời điểm có một (hoặc 2) biến có giá trị 1 thì hàm có trị 1 và hàm chỉ có trị 0 tại những thời điểm mà cả 2 biến đều bằng 0.



4.4. Tối giản (tối thiểu hóa) hàm logic

Để thực hiện một hàm logic bằng mạch điện tử, người ta luôn luôn nghĩ đến việc sử dụng lượng linh kiện ít nhất. Muốn vậy, hàm logic phải ở dạng tối giản, nên vấn đề rút gọn hàm logic là **bước đầu tiên phải thực hiện** trong quá trình thiết kế. Có 3 phương pháp rút gọn hàm logic:

- Phương pháp đại số
- Phương pháp dùng bảng Các - nô
- Phương pháp Quine Mc. Cluskey

Trong nội dung giáo trình này chúng ta sẽ xét 2 phương pháp thông dụng là phương pháp đại số và phương pháp dùng bảng Các - nô.

4.4.1 Các phương pháp tối giản hàm logic

4.4.1.1. Phương pháp đại số

Phương pháp này bao gồm việc áp dụng các tính chất của hàm logic cơ bản. Một số đẳng thức thường sử dụng được nhóm lại như sau:

(1) $AB + \overline{A}B = B$ $(A+B).(\overline{A}+B) = B$ (1')

(2) $A + AB = A$ $A.(A+B) = A$ (2')

(3) $A + \overline{A}B = A + B$ $A.(\overline{A}+B) = A.B$ (3')

Chứng minh các đẳng thức 1, 2, 3:

(1) $AB + \overline{A}B = B(A+\overline{A}) = B.1 = B$

(2) $A + AB = A(1+B) = A$

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

Các đẳng thức (1'), (2'), (3') là song đối của (1), (2), (3).

Các qui tắc rút gọn:

- Qui tắc 1: Nhờ các đẳng thức trên nhóm các số hạng lại.

Ví dụ: Rút gọn biểu thức: $ABC + AB\bar{C} + A\bar{B}CD$

Theo (1) $ABC + AB\bar{C} = AB$

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

Theo (3) $B + \bar{B}CD = B + CD$

Và kết quả cuối cùng: $ABC + AB\overline{C} + A\overline{B}CD = A(B + CD)$

- Qui tắc 2: Ta có thể thêm một số hạng đã có trong biểu thức logic vào biểu thức mà không làm thay đổi biểu thức.

Ví dụ: Rút gọn biểu thức: $ABC + \bar{A}BC + A\bar{B}C + AB\bar{C}$

Thêm ABC vào để được:

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

Theo (1) các nhóm trong dấu ngoặc rút gọn thành: BC + AC + AB

Vậy:

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

- Qui tắc 3: Có thể bỏ số hạng chứa các biến đã có trong số hạng khác

Ví dụ 1: Rút gọn biểu thức: $AB + \bar{B}C + AC$

Biểu thức không đổi nếu ta nhân một số hạng trong biểu thức với 1, ví

dụ $(B + \overline{B})$: $AB + \overline{B}C + AC = AB + \overline{B}C + (B + \overline{B})AC$
Triển khai số hạng cuối cùng của vế phải, ta được:

$$AB + \overline{B}C + ABC + A\overline{B}C$$

Thừa số chung: $AB(1 + C) + \overline{B}C(1 + A) = AB + \overline{B}C$
Tóm lại: $AB + \overline{B}C + AC = AB + \overline{B}C.$
Trong bài toán này ta đã đơn giản được số hạng AC.

Ví dụ 2: Rút gọn biểu thức: $(A + B).(\overline{B} + C).(A + C)$
Biểu thức không đổi nếu ta thêm vào một thừa số bất kỳ một số có giá trị bằng 0, ví dụ $B.\overline{B}$:

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

Theo (2') (A+B).(A +B+C) = (A+B) và $(\overline{B} + C).(A + \overline{B} + C) = (\overline{B} + C)$
Vậy: $(A + B).(\overline{B} + C).(A + C) = (A + B).(\overline{B} + C)$
Trong bài toán này ta đã bỏ được thừa số (A+C)

4.4.1.2. Dùng bảng Các - nô

Dùng bảng Các - nô cho phép rút gọn dễ dàng các hàm logic chứa từ 3 tới 6 biến.

a. Nguyên tắc

Xét hai tổ hợp biến AB và $A\overline{B}$, hai tổ hợp này chỉ khác nhau một bit, ta gọi chúng là hai tổ hợp kề nhau.
Ta có: AB + $A\overline{B}$ = A , biến B đã được đơn giản .

Phương pháp của bảng Các - nô dựa vào việc nhóm các tổ hợp kề nhau trên bảng để đơn giản biến có giá trị khác nhau trong các tổ hợp này.
Công việc rút gọn hàm được thực hiện theo bốn bước:

- Vẽ bảng Các - nô theo số biến của hàm
- Chuyển hàm cần đơn giản vào bảng Các - nô
- Gom các ô chứa các tổ hợp kề nhau lại thành các nhóm sao cho có thể rút gọn hàm tới mức tối giản
- Viết kết quả hàm rút gọn từ các nhóm đã gom được.

b. Vẽ bảng Các - nô

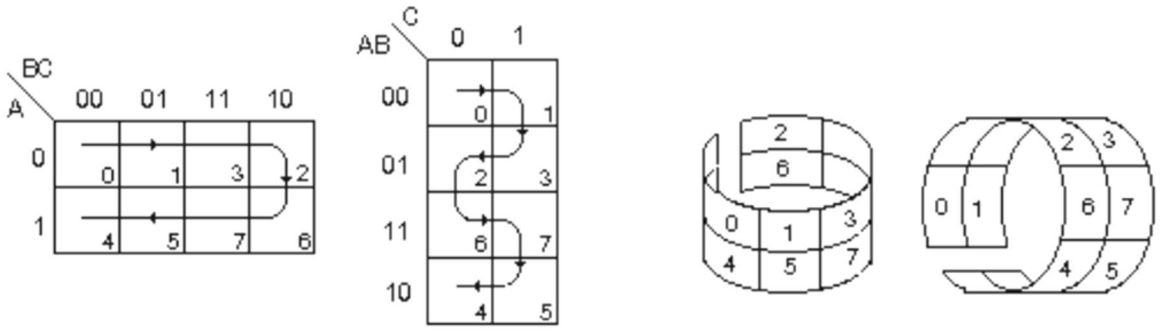
Bảng Các - nô thực chất là một dạng khác của bảng sự thật, trong đó mỗi ô của bảng tương đương với một hàng trong bảng sự thật.

Để vẽ bảng Các - nô cho n biến, người ta chia số biến ra làm đôi, phân nửa dùng để tạo $2^{n/2}$ cột, phân nửa còn lại tạo $2^{n/2}$ hàng (nếu n là số lẻ, người ta có thể cho số lượng biến trên cột lớn hơn số lượng biến cho hàng hay ngược lại cũng được). Như vậy, với một hàm có n biến, bảng Các - nô gồm 2^n ô, mỗi ô tương ứng với tổ hợp biến này. Các ô trong bảng được sắp đặt sao cho hai ô kề nhau chỉ khác nhau một đơn vị nhị phân (khác nhau

một bit), điều này cho thấy rất thuận tiện nếu chúng ta dùng mã Gray. Chính sự sắp đặt này cho phép ta đơn giản bằng cách nhóm các ô kề nhau lại.

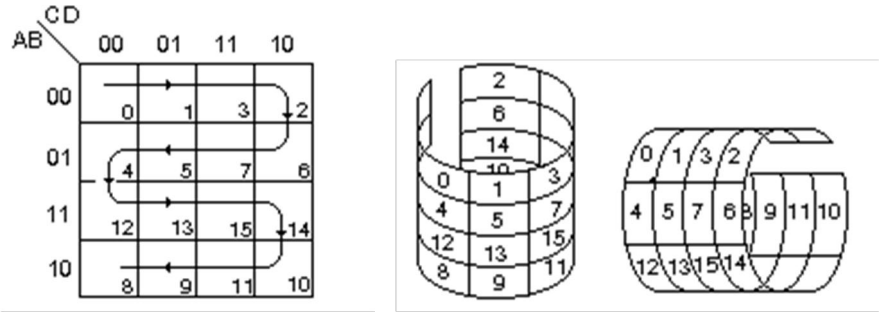
Với 2 biến AB, sự sắp đặt sẽ theo thứ tự: AB = 00, 01, 11, 10 (đây là thứ tự mã Gray, nhưng để cho dễ ta dùng số nhị phân tương ứng để đọc thứ tự này: 0, 1, 3, 2).

Ví dụ : Bảng Các - nô cho hàm 3 biến (A = MSB, và C = LSB)



Với 3 biến ABC, ta được: ABC = 000, 001, 011, 010, 110, 111, 101, 100 (số nhị phân tương ứng: 0, 1, 3, 2, 6, 7, 5, 4)

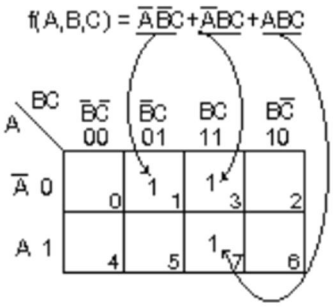
Lưu ý là ta có thể thiết lập bảng Các - nô theo chiều nằm ngang hay theo chiều đứng. Do các tổ hợp ở các bìa trái và phải kề nhau nên ta có thể coi bảng có dạng hình trụ thẳng đứng và các tổ hợp ở bìa trên và dưới cũng kề nhau nên ta có thể coi bảng có dạng hình trụ trục nằm ngang. Và 4 tổ hợp biến ở 4 góc cũng là các tổ hợp kề nhau.
Hình dưới là bảng Các - nô cho 4 biến.



c. Chuyển hàm logic vào bảng Các - nô.

Trong mỗi ô của bảng ta đưa vào giá trị của hàm tương ứng với tổ hợp biến, để đơn giản chúng ta có thể chỉ ghi các trị 1 mà bỏ qua các trị 0 của hàm. Ta có các trường hợp sau:
Từ hàm viết dưới dạng tổng chuẩn:

Thí dụ 1 : f(A,B,C) = $\overline{A}.\overline{B}.C + \overline{A}.B.C + A.B.C$



(H 2.5)

♦ **Nếu hàm không phải là dạng chuẩn**, ta phải đưa về dạng chuẩn bằng cách thêm vào các số hạng sao cho hàm vẫn không đổi nhưng các số hạng chứa đủ các biến.

Thí dụ 2 : $Y = \overline{A}\,BC + AB\,\overline{D} + A\,\overline{B}\,C + A\,\overline{C}\,D$

Hàm này gồm 4 biến, nên để đưa về dạng tổng chuẩn ta làm như sau:

$Y = \overline{A}\,BC(D+\overline{D}) + AB\,\overline{D}\,(C+\overline{C}) + A\,\overline{B}\,C(D+\overline{D}) + A\,\overline{C}\,D(B+\overline{B})$

$Y = \overline{A}\,BCD + \overline{A}\,BC\,\overline{D} + ABC\,\overline{D} + AB\,\overline{C}\,\overline{D} + A\,\overline{B}\,CD + A\,\overline{B}\,C\,\overline{D} + AB\,\overline{C}\,D + A\,\overline{B}\,\overline{C}\,D$

Và Hàm Y được đưa vào bảng Các - nô như sau:

AB \ CD	$\overline{C}\,\overline{D}$ 00	$\overline{C}D$ 01	CD 11	$C\overline{D}$ 10
$\overline{A}\,\overline{B}$ 00				
$\overline{A}B$ 01			1	1
AB 11	1	1		1
$A\overline{B}$ 10		1	1	1

- Từ dạng số thứ nhất, với các trọng lượng tương ứng A = 4, B = 2, C = 1

Ví dụ 3 : f(A,B,C) = Σ(1,3,7). Hàm số sẽ lấy giá trị 1 trong các ô 1,3 và 7.

– **Từ dạng tích chuẩn**: Ta lấy hàm đảo để có dạng tổng chuẩn và ghi trị 0 vào các ô tương ứng với tổ hợp biến trong tổng chuẩn này. Các ô còn lại chứa số 1.

Ví dụ 4 :

$\boldsymbol{Y = f(A,B,C)}$

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

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

Và bảng Các - nô tương ứng :

A \ BC	$\overline{B}\,\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 (0)	1 (1)	1 (3)	0 (2)
A 1	0 (4)	0 (5)	1 (7)	0 (6)

- Từ dạng số thứ hai:
Ví dụ 5 : f(A,B,C) = ∠(0,2,4,5,6)

Hàm sẽ lấy các trị 0 ở các ô 0, 2, 4, 5, 6. Dĩ nhiên là ta phải ghi các giá trị 1 trong các ô còn lại (hình vẽ trên).

4.4.2 Bài tập ứng dụng
Bài 1:
Cho hàm logic F(A, B, C) thỏa tính chất: F(A,B,C) = 1 nếu có một và chỉ một biến bằng 1.

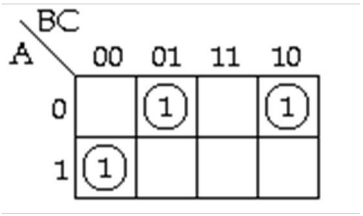
a- Lập bảng sự thật cho hàm F.
b- Rút gọn hàm F.
c- Diễn tả hàm F chỉ dùng hàm AND và NOT.

Giải

a. Dựa vào điều kiện của bài toán ta có bảng sự thật của hàm F:

A	B	C	F(A,B,C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0

b. Rút gọn hàm F bằng bảng Các - nô:
Bảng Các - nô:



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

c. Diễn tả hàm F chỉ dùng hàm AND và NOT:
Dùng định lý De Morgan, lấy đảo 2 lần hàm F:

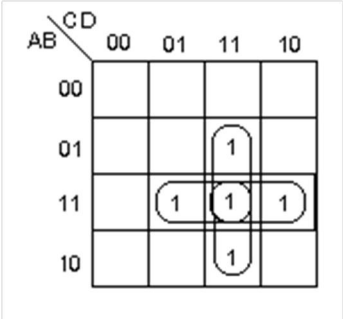
$$\boldsymbol{F(A,B.C) = \overline{\overline{F(A,B.C)}} = \overline{\overline{\bar{A}.\bar{B}.C + \bar{A}.B.\bar{C} + A.\bar{B}.\bar{C}}} = \overline{\overline{\bar{A}.\bar{B}.C}.\overline{\bar{A}.B.\bar{C}}.\overline{A.\bar{B}.\bar{C}}}}$$

Bài 2:
Cho hàm logic F(A, B, C, D) thỏa tính chất: F(A,B,C,D) = 1 khi có ít nhất 3 biến bằng 1.

a- Rút gọn hàm F.
b- Diễn tả hàm F chỉ dùng hàm OR và NOT

Giải:

a- Rút gọn hàm F
Ta có thể đưa hàm vào trong bảng Các - nô mà không cần vẽ bảng sự thật.
Ta đưa số 1 vào tất cả các ô có chứa 3 trị 1 của các biến trở lên



Và kết quả của hàm rút gọn là:
F(A,B,C,D) = ABC + ABD + ACD + BCD

b- Diễn tả hàm F chỉ dùng hàm OR và NOT
Dùng định lý De Morgan cho từng số hạng trong tổng
Viết lại hàm F:

$$F(A,B,C,D) = \overline{\overline{ABC}} + \overline{\overline{ABD}} + \overline{\overline{ACD}} + \overline{\overline{BCD}}$$

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

BÀI TẬP

1. Đổi các số thập phân dưới đây sang hệ nhị phân và hệ thập lục phân:
12; 24; 192; 2079; 15492; 0,25; 0,375; 0,376; 17,150; 192,1875
2. Đổi sang hệ thập phân và mã BCD các số nhị phân sau đây:
1011; 10110; 101,1; 0,1101; 0,001; 110,01; 1011011; 10101101011
3. Đổi các số thập lục phân dưới đây sang hệ 10 và hệ 8:
FF; 1A; 789; 0,13; ABCD,EF
4. Đổi các số nhị phân dưới đây sang hệ 8 và hệ 16:
a/ 111001001,001110001
b/ 10101110001,00011010101
c/1010101011001100,1010110010101
d/ 1111011100001,01010111001
5. Mã hóa số thập phân dưới đây dùng mã BCD:
12; 192; 2079; 15436; 0,375; 17,250
6. Diễn tả mỗi mệnh đề dưới đây bằng một biểu thức logic:
    a. Tất cả các biến A,B,C,D đều bằng 1.
    b. Tất cả các biến A,B,C,D đều bằng 0.
    c. Ít nhất 1 trong các biến X, Y, Z, T bằng 1.
    d. Ít nhất 1 trong các biến X, Y, Z, T bằng 0.
    e. Các biến A,B,C,D lần lượt có giá trị 0, 1, 1, 0.

7. Tính đảo của các hàm sau:

a. $f1 = (A + B)(\bar{A} + \bar{B})$

b. $f2 = (A + \bar{B} + \bar{C})(B + \bar{C} + D)(\bar{A} + C + D)$

c. $f3 = A(C + D) + (\bar{A} + C)(\bar{B} + C + D)$

d. $f4 = (AB + C)(BC + D) + \bar{A}BC + \bar{C}D$

e. $f5 = \bar{A}.\bar{B}.C + \bar{A}.B.\bar{C} + A(B.C + \bar{B}.\bar{C})$

8. Chứng minh bằng đại số các biểu thức sau:

a. $\overline{A.B + \bar{A}.\bar{B}} = \bar{A}.B + A.\bar{B}$

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

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

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

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

9. Rút gọn các hàm dưới đây bằng phương pháp đại số (A = MSB)

a. $f1 = ABC + A\bar{B}C + AB\bar{C}D$

b. $f2 = (A + BC) + \bar{A}(\bar{B} + \bar{C})(AD + C)$

c. $f3 = (A + B + C).(A + B + \bar{C}).(\bar{A} + B + C).(\bar{A} + B + \bar{C})$

d. *f4(A,B,C,D) = Σ(0,3,4,7,8,9,14,15)*

e. $f5 = \bar{A}B + AC + BC$

f. $f6 = (A + \bar{C})(B + C)(A + B)$

10. Dùng bảng Các - nô rút gọn các hàm sau: (A = MSB)

a. f(A,B,C) = Σ(1,3,4)

b. f(A,B,C) = Σ (1,3,7)

c. f(A,B,C) = Σ (0,3,4,6,7)

d. f(A,B,C) = Σ (1,3,4) . Các tổ hợp biến 6,7 cho hàm không xác định

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

f. f(A,B,C,D) = Σ (5,7,13,15)

g. f(A,B,C,D) = Σ (0,4,8,12)

h. f(A,B,C,D) = Σ (0,2,8,10)

i. f(A,B,C,D) = Σ (0,2,5,6,9,11,13,14)

j. f(A,B,C,D) = ∏(0,1,5,9,10,15)

k. f(A,B,C,D) = ∏ (0,5,9,10) với các tổ hợp biến (2,3,8,15) cho hàm không xác định.

l. f(A,B,C,D,E) = Σ (2,7,9,11,12,13,15,18,22,24,25,27,28,29,31)

m. f(A,B,C,D.E) = Σ (0,2,8,10,13,15,16,18,24,25,26,29,31) với các tổ hợp biến (7,9,14,30) cho hàm không xác định

n. f(A,B,C,D,E,F) = Σ (2,3,6,7,8,9,12,13,14,17,24,25,28,29,30,40,41,44,45,46,56,57,59,60,61,63)

o. f(A,B,C,D,E,F) = Σ (9,11,13,15,16,18,20,22,25,27,29,31,32,34,36,38,41,43,45,47,48,50,52,54)

BÀI 2
CÁC PHẦN TỬ LOGIC CƠ BẢN
Mã bài: MĐ26.02

Giới thiệu:

Cổng logic là tên gọi chung của các mạch điện tử có chức năng thực hiện các hàm logic. Cổng logic có thể được chế tạo bằng các công nghệ khác nhau (Lưỡng cực, MOS), có thể được tổ hợp bằng các linh kiện rời nhưng thường được chế tạo bởi công nghệ tích hợp IC (Integrated circuit).

Bài học này giới thiệu các loại cổng cơ bản, một số ứng dụng của các cổng logic cơ bản và lắp ráp, cân chỉnh một số mạch logic cơ bản thông dụng.

Mục tiêu:

- Trình bày được cấu tạo, nguyên lý hoạt động của các cổng logic cơ bản
- Trình bày được các ứng dụng của cổng logic cơ bản.
- Lắp ráp và cân chỉnh được các mạch dùng cổng logic cơ bản
- Chủ động, sáng tạo và đảm bảo an toàn trong quá trình làm việc.

Nội dung chính:

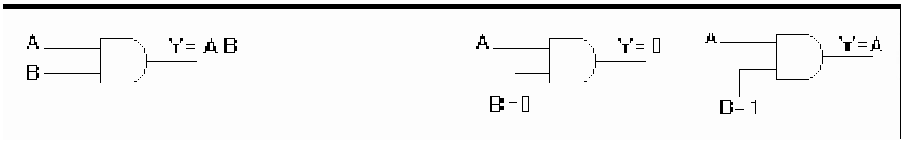
1. Các cổng logic cơ bản

Mục tiêu:

Trình bày được biểu thức logic, bảng sự thật và ký hiệu của các cổng logic cơ bản.

1.1. Cổng AND

- Dùng thực hiện hàm AND 2 hay nhiều biến.
- Cổng AND có số đầu vào tùy thuộc số biến và một đầu ra. Đầu ra của cổng là hàm AND của các biến đầu vào.
- Ký hiệu cổng AND 2 đầu vào cho 2 biến



(a) (b)

A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Hoặc

A	B	Y=A.B
x	0	0
x	1	A

Nhận xét:

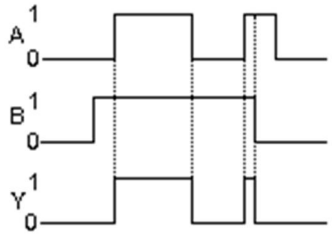
- Đầu ra cổng AND chỉ ở mức cao khi tất cả đầu vào lên mức cao.
- Khi có một đầu vào = 0, đầu ra = 0 bất chấp các đầu vào còn lại.

- Khi có một đầu vào =1, đầu ra = AND của các đầu vào còn lại.

Vậy với cổng AND 2 đầu vào ta có thể dùng 1 đầu vào làm đầu kiểm soát, khi đầu kiểm soát = 1, **cổng mở** cho phép tín hiệu logic ở đầu vào còn lại qua cổng và khi đầu kiểm soát = 0, **cổng đóng** , đầu ra luôn bằng 0, bất chấp đầu vào còn lại.

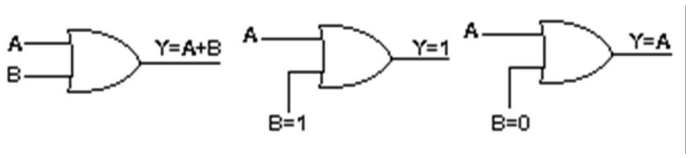
Với cổng AND có nhiều đầu vào hơn, khi có một đầu vào được đưa lên mức cao thì đầu ra bằng AND của các biến ở các đầu vào còn lại.

Hình dưới đây là giản đồ thời gian của cổng AND hai đầu vào. Trên giản đồ, đầu ra Y chỉ lên mức 1 khi cả A và B đều ở mức 1.



1.2. Cổng OR

- Dùng để thực hiện hàm OR 2 hay nhiều biến.
- Cổng OR có số đầu vào tùy thuộc số biến và một đầu ra.
- Ký hiệu cổng OR 2 đầu vào



- Bảng sự thật

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Hoặc

A	B	Y=A+B
x	1	1
x	0	A

Nhận xét:

- Đầu ra cổng OR chỉ ở mức thấp khi cả 2 đầu vào xuống mức thấp.
- Khi có một đầu vào =1, đầu ra = 1 bất chấp đầu vào còn lại.
- Khi có một đầu vào =0, đầu ra = OR các đầu vào còn lại.

Vậy với cổng OR 2 đầu vào ta có thể dùng 1 đầu vào làm đầu kiểm soát, khi đầu kiểm soát = 0, **cổng mở**, cho phép tín hiệu logic ở đầu vào còn lại qua cổng và khi đầu kiểm soát = 1, **cổng đóng**, đầu ra luôn bằng 1.

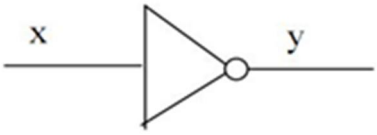
Với cổng OR nhiều đầu vào hơn, khi có một đầu vào được đưa xuống mức thấp thì đầu ra bằng OR của các biến ở các đầu vào còn lại.

1.3. Cổng NOT

- Còn gọi là cổng đảo (Inverter), dùng để thực hiện hàm đảo $\boldsymbol{Y = \overline{A}}$

Cổng NOT là cổng logic có 1 đầu vào và một đầu ra, với ký hiệu và bảng trạng

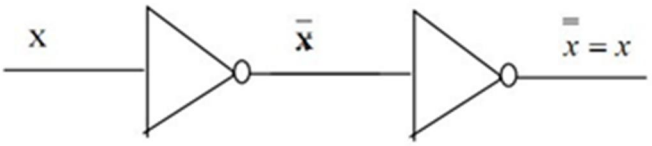
thái hoạt động như hình vẽ:



Bảng trạng thái:

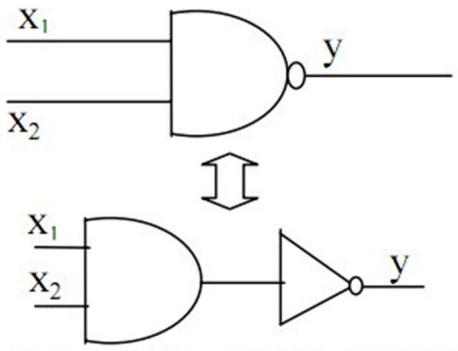
x	y
0	1
1	0

Cổng đảo giữ chức năng như một cổng đệm, nhưng người ta gọi là đệm đảo vì tín hiệu đầu ra ngược pha với tín hiệu đầu vào.
Ghép hai cổng đảo ta được cổng không đảo.



Sử dụng 2 cổng ĐẢO tạo ra cổng ĐỆM

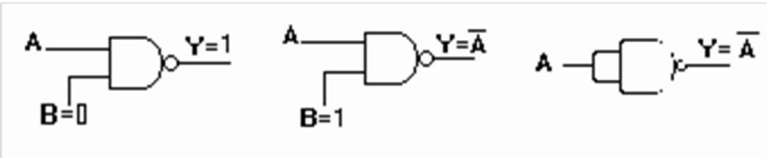
1.4. Cổng NAND
- Là kết hợp của cổng AND và cổng NOT, thực hiện hàm $\boldsymbol{Y = \overline{A.B}}$
(Ở đây chỉ xét cổng NAND 2 đầu vào, trường hợp nhiều đầu vào ta thay bằng hàm nhiều biến).
- Ký hiệu của cổng NAND (Gồm AND và NOT, cổng NOT thu gọn lại một vòng tròn)



x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

Cổng NAND: Ký hiệu, sơ đồ logic tương đương và bảng trạng thái

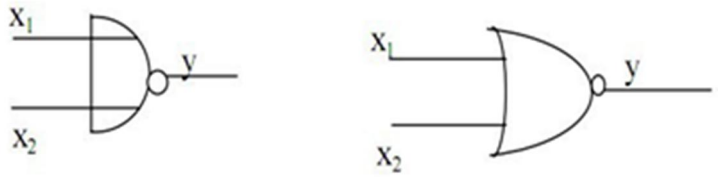
- Tương tự như cổng AND, ở cổng NAND ta có thể dùng 1 đầu vào làm đầu kiểm soát. Khi đầu kiểm soát = 1, cổng mở cho phép tín hiệu logic ở đầu vào còn lại qua cổng và bị đảo, khi đầu kiểm soát = 0, cổng đóng, đầu ra luôn bằng 1.
- Khi nối tất cả đầu vào của cổng NAND lại với nhau, nó hoạt động như một cổng đảo



1.5. Cổng NOR

Là sự kết hợp của cổng OR và cổng NOT, thực hiện hàm $\boldsymbol{Y = \overline{A + B}}$

Là cổng thực hiện chức năng của phép toán cổng đảo logic, là cổng có 2, đầu vào và 1 đầu ra có ký hiệu như hình vẽ:

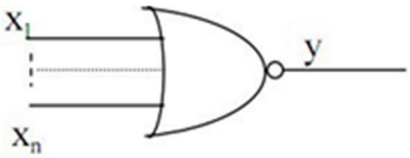


Ký hiệu Châu Âu Ký hiệu theo Mỹ, Nhật, Úc

Bảng trạng thái mô tả hoạt động của cổng NOR:

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

Xét trường hợp tổng quát cho cổng NOR có n đầu vào:

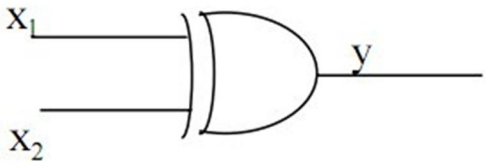


$$y_{NOR} = \begin{cases} 0 & \exists x_i = 1 \\ 1 & \forall x_i = 0 \end{cases} \quad (i = \overline{1,n})$$

Vậy đặc điểm của cổng NOR là: Tín hiệu đầu ra chỉ bằng 1 khi tất cả các đầu vào đều bằng 0, tín hiệu đầu ra sẽ bằng 0 khi có ít nhất 1 đầu vào bằng 1.

1.6. Cổng XOR (**Cổng EX-OR)**

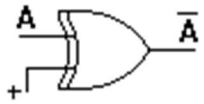
- Dùng để thực hiện hàm EX-OR: $Y = A \oplus B = \overline{A}B + A\overline{B}$
- Cổng EX-OR chỉ có 2 đầu vào và 1 đầu ra. Ký hiệu và bảng trạng thái làm việc như ở hình (a).



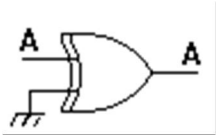
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

(a)

- Một tính chất rất quan trọng của cổng EX-OR:
+ Tương đương với một cổng đảo khi có một đầu vào nối lên mức cao, (H.b)
+ Tương đương với một cổng đệm khi có một đầu vào nối xuống mức thấp, (Hc)



(b)



(c)

2. Một số ứng dụng cổng logic cơ bản

Mục tiêu:

Trình bày được một số ứng dụng cơ bản của các cổng logic đã học; trên cơ sở đó xây dựng được một số sơ đồ ứng dụng thông dụng khác.

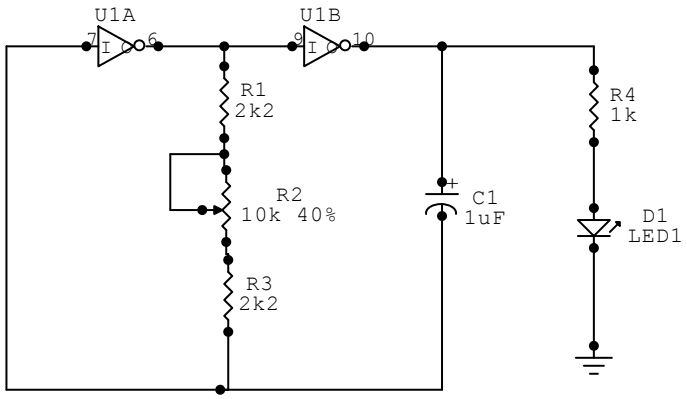
2.1 Mạch tạo xung vuông dùng các cổng NOT, NAND của IC 4049 và 7400.

2.1.1. Mạch tạo xung vuông trên IC 4049

Thực hiện vẽ mạch như hình sau bằng cách sử dụng:

- 2 cổng NOT trên IC 4049.
- Các điện trở, biến trở, tụ điện có giá trị như trên sơ đồ nguyên lý.
- 1 LED chỉ thị.

Sơ đồ nguyên lý:

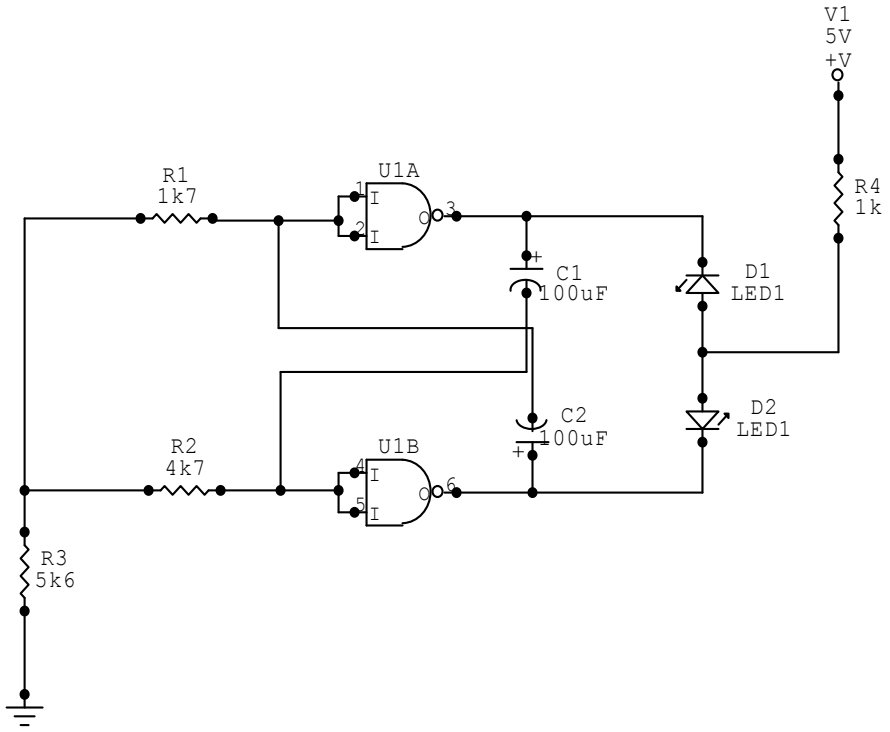


2.1.2. Mạch tạo xung vuông trên IC 7400.

Thực hiện vẽ mạch như hình sau bằng cách sử dụng:

- 2 cổng NAND trên IC 7400.
- Các điện trở, biến trở, tụ điện có giá trị như trên sơ đồ nguyên lý.
- 2 LED chỉ thị.

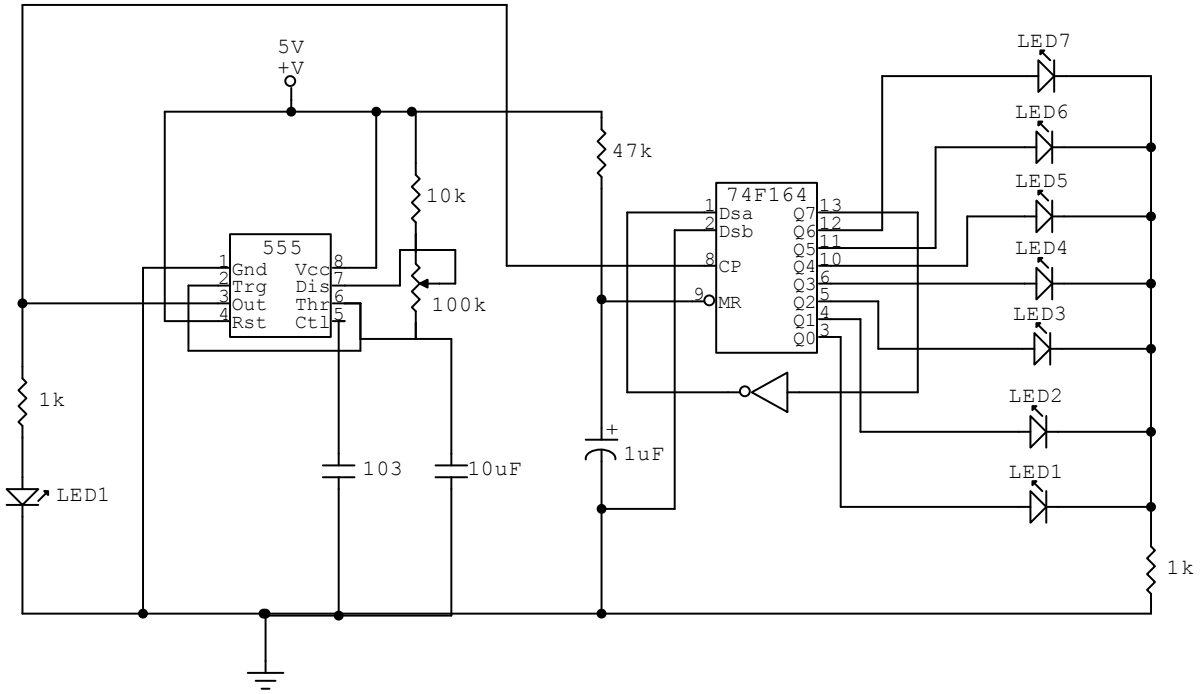
Sơ đồ nguyên lý:



Mạch có thể được mô phỏng nhờ Circuitmaker hoặc Proteur ...

2.2 Mạch đếm đến 8 dùng IC 74164

Yêu cầu của bài này là sinh viên phải xem trước phần lý thuyết về IC74164 (sơ đồ chân và nguyên lý hoạt động).



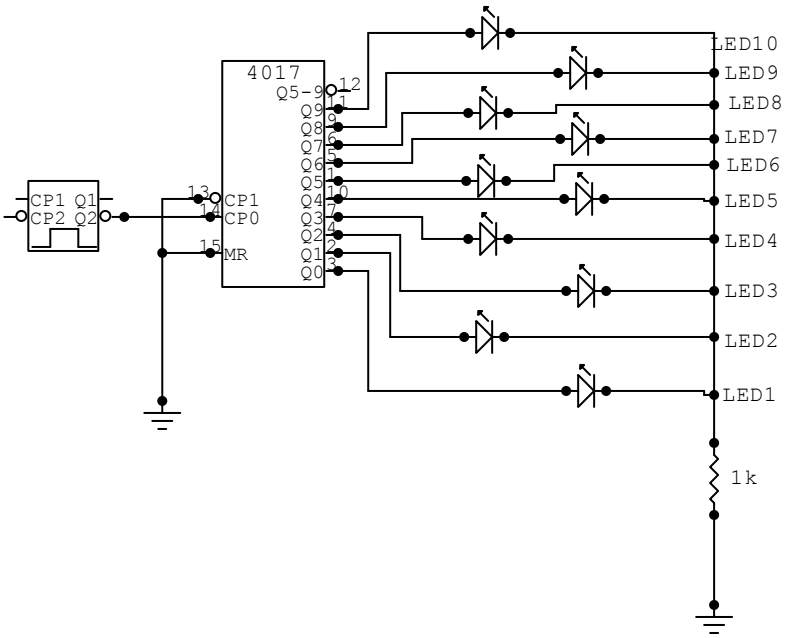
Sinh viên hãy vẽ mạch điện theo sơ đồ nguyên lý, sử dụng:

- 1 IC 74164 [Digital by Number/74xxx/74164]
- 1 Mạch tạo xung [Dùng IC 555]
- 1 "Inverter" [Digital/Buffer đến Inverter/ Inverter]
- 9 Logic Display [LED chỉ thị]
- Các điện trở, biến trở, tụ điện như trên sơ đồ nguyên lý.

Mạch có thể được mô phỏng nhờ Circuitmaker hoặc Proteur ...

2.3 Mạch đếm đến 10 dùng IC 4017

Yêu cầu của bài này là sinh viên phải xem trước phần lý thuyết về IC4017 (sơ đồ chân và nguyên lý hoạt động).



Sinh viên hãy dùng vẽ mạch điện dùng:

- 1 IC 4017 [Digital by Number/40xx/4017]
- 1 Mạch tạo xung [Instruments/Digital/Pulser] (p)
- 11 Logic Display [LED chỉ thị]
- Các điện trở, biến trở, tụ điện như trên sơ đồ nguyên lý (cho mạch tạo xung vuông như trên).

Mạch có thể được mô phỏng nhờ Circuitmaker, Proteur hoặc Orcad ...

3. Lắp ráp và cân chỉnh một số mạch dùng cổng logic cơ bản

Mục tiêu:

- Lắp ráp được mạch đếm 10 dùng IC 4017 trên bo mạch đa năng hoặc mạch in chế tạo sẵn.
- Biết cân chỉnh các tham số để mạch hoạt động theo yêu cầu.

3.1 Lắp mạch đếm 10 dùng IC 4017

3.1.1 Giới thiệu sơ đồ nguyên lý

Sơ đồ nguyên lý như ở phần 2.3.

3.1.2 Vật tư cần thiết:

Tổng hợp từ sơ đồ nguyên lý.

3.1.3 Thực hành lắp ráp:

3.1.3.1 Lắp trên bo mạch đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiệc trước khi dùng làm phần tử kết nối trong mạch.

- Bố trí các linh kiện hợp lý

- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.

- Mối hàn phải ngấu, bóng.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đếm.

- Mạch phải hoạt động khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ bên ngoài.

3.1.3.2 Lắp trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp thiết kế (không có đoạn bị đứt).

- Có sơ đồ lắp ráp kèm theo.

- Khi lắp phải cắm đúng vị trí và chiều các linh kiện theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, gọn không gây chạm chập trên mạch in.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đếm.

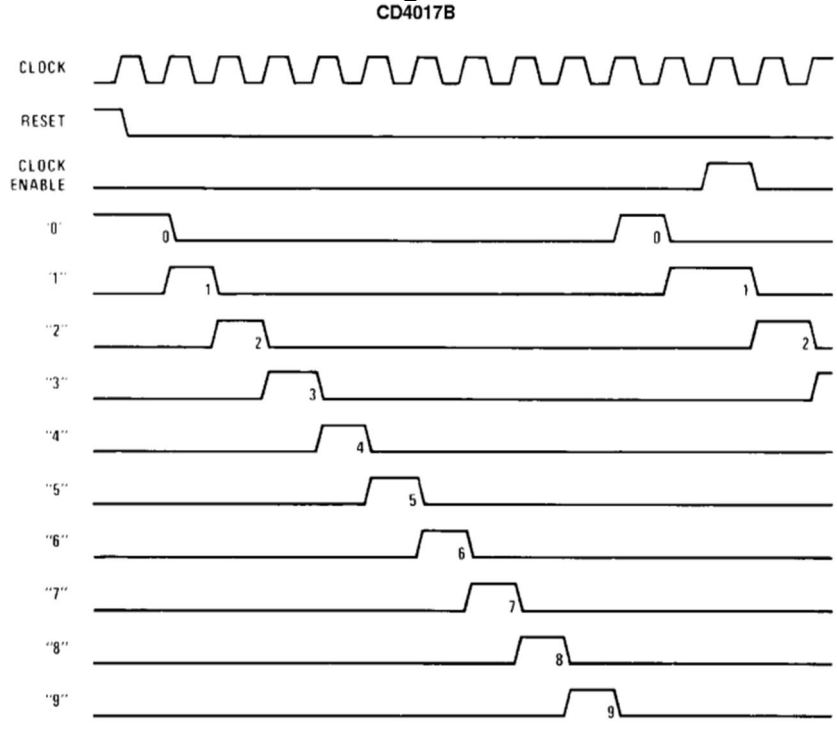
- Mạch phải hoạt động khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ bên ngoài.

3.2 Hiệu chỉnh và thay đổi hiệu ứng.

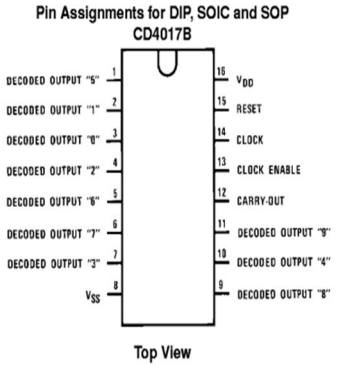
Dựa vào biểu đồ thời gian mô tả hoạt động và sơ đồ chân của IC 4017 để thay đổi các hiệu ứng:

- Mạch đếm liên tục

- Đếm đến 10 thì dừng



Biểu đồ thời gian



Sơ đồ chân

BÀI 3
CÁC PHẦN TỬ LOGIC THÔNG DỤNG
Mã bài: MĐ26.03

Giới thiệu:

Cổng logic là tên gọi chung của các mạch điện tử có chức năng thực hiện các hàm logic. Cổng logic có thể được chế tạo bằng các công nghệ khác nhau (Lưỡng cực, MOS), có thể được tổ hợp bằng các linh kiện rời nhưng thường được chế tạo bởi công nghệ tích hợp IC (Integrated circuit).

Bài học này giới thiệu các mạch tạo thành các cổng logic cơ bản từ các linh kiện bán dẫn rời hoặc tổ hợp; việc chuyển đổi giữa các loại cổng logic thông dụng.

Mục tiêu:

- Mô tả được cấu trúc phân bố của các cổng logic.
- Xây dựng được sơ đồ nguyên lý sử dụng các linh kiện bán dẫn để tạo các cổng logic.
- Thực hiện chuyển đổi từ cổng logic này sang cổng logic khác.
- Có tư duy, tác phong công nghiệp

Nội dung chính:

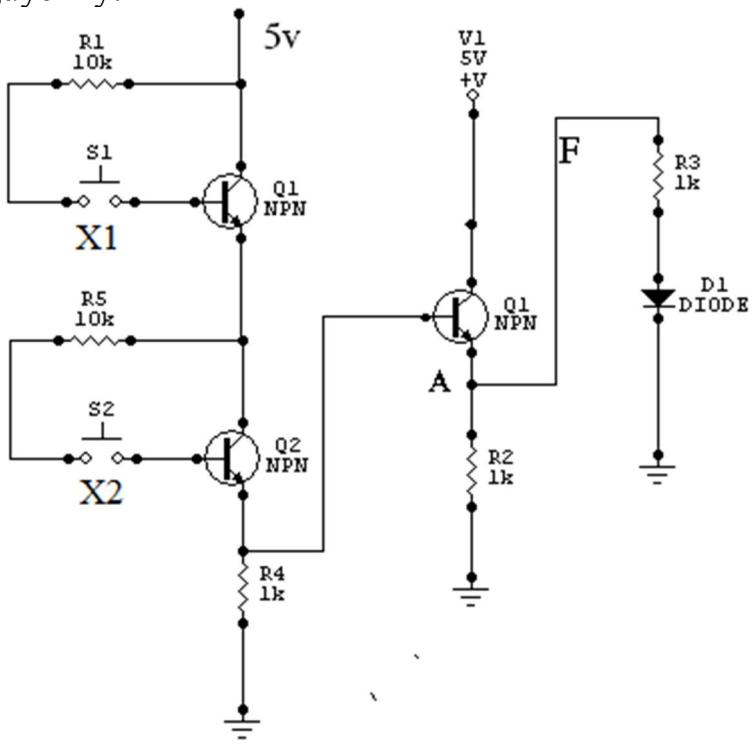
1. Mạch tạo thành cổng NAND

Mục tiêu:

Trình bày được sơ đồ nguyên lý, nguyên lý hoạt động của các mạch dùng đi - ốt, BJT, FET để tạo nên cổng NAND.

1.1. Tạo cổng NAND từ đi ốt - BJT

Sơ đồ nguyên lý:



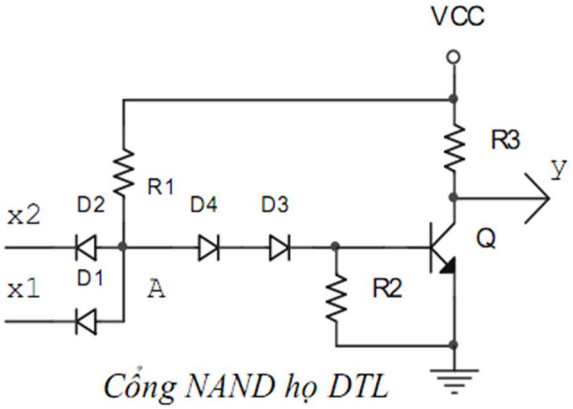
Sinh viên hãy vẽ mạch điện dùng:

- 3 BJT loại NPN như trên sơ đồ (hoặc PNP)
- 2 nút ấn S1, S2.
- 1 Logic Display [LED chỉ thị]
- Các điện trở có giá trị như trên sơ đồ nguyên lý.

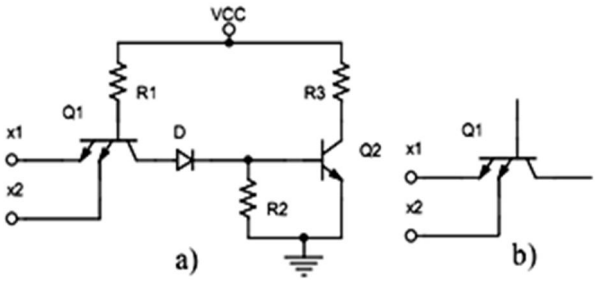
Nguyên lý hoạt động:

Khi hở S1 và S2 (X0 = 0; X2 = 0) hoặc khi chỉ nhấn một trong số S1 hoặc S2 (chỉ X1 = 0 hoặc X2 = 0) thì D1 không sáng (F = 0). Chỉ khi ấn đồng thời S1 và S2 (X1 = 1; X2 = 1) thì D1 mới sáng (F = 1).

Sơ đồ khác:



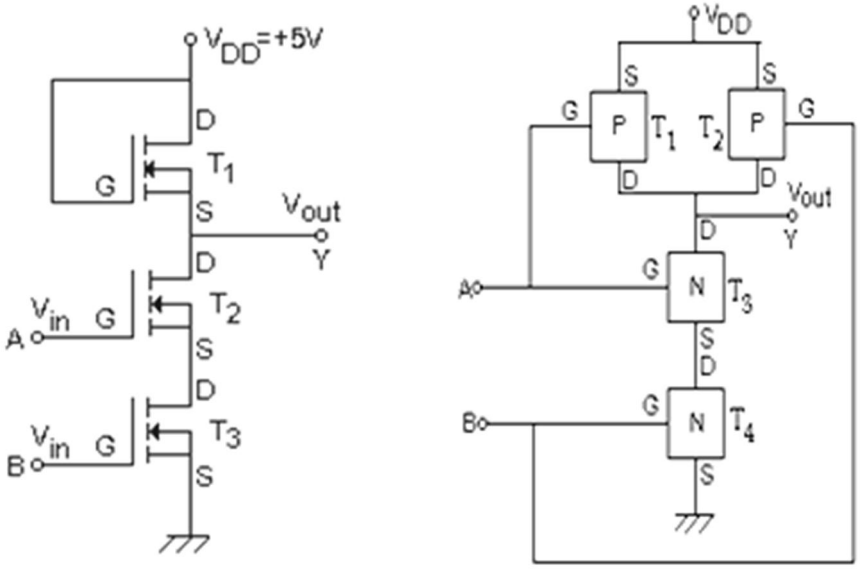
Cổng NAND họ DTL



Cổng NAND họ TTL

a. Sơ đồ mạch, b.Transistor 2 tiếp giáp

1.2. Tạo cổng NAND từ BJT - FET



a) **b)**

Cổng NAD họ NMOS như ở hình a); họ CMOS như ở hình b).

Nguyên lý hoạt động của chúng cơ bản giống nhau.

+ Cổng NAND họ NMOS:

- Khi 2 đầu vào nối lên mức cao, T_2 và T_3 dẫn, đầu ra xuống mức thấp.
- Khi có 1 đầu vào nối xuống mức thấp, một trong 2 transistor T_2 hoặc T_3 tắt, đầu ra lên mức cao.

Đó chính là kết quả của cổng NAND 2 đầu vào.

+ Cổng NAND họ CMOS:

- Khi 2 đầu vào nối lên mức cao, T1 và T2 tắt, T3 và T4 dẫn, đầu ra xuống mức thấp.

- Khi có 1 đầu vào nối xuống mức thấp, một trong 2 transistor T3 hoặc T4 tắt, một trong 2 transistor T1 hoặc T2 dẫn, đầu ra lên mức cao.

Đó chính là kết quả của cổng NAND 2 đầu vào.

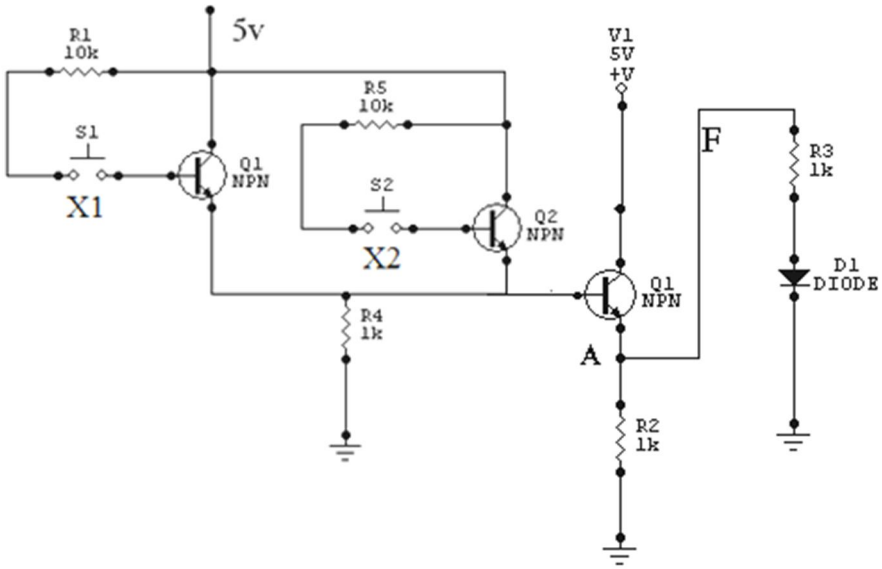
2. Mạch tạo thành cổng NOR

Mục tiêu:

Trình bày được sơ đồ nguyên lý, nguyên lý hoạt động của các mạch dùng đi - ốt, BJT, FET để tạo nên cổng NOR.

2.1. Tạo cổng NOR từ đi ốt - BJT

Sơ đồ nguyên lý:



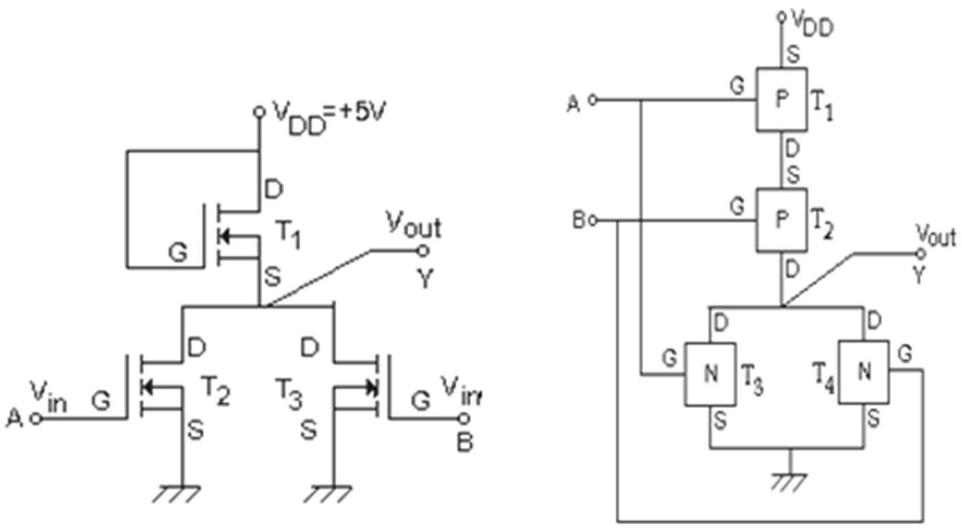
Sinh viên hãy vẽ mạch điện dùng:

- 3 BJT loại NPN như trên sơ đồ (hoặc PNP)
- 2 nút ấn S1, S2.
- 1 Logic Display [LED chỉ thị]
- Các điện trở có giá trị như trên sơ đồ nguyên lý.

Nguyên lý hoạt động:

Khi hở S1 và S2 (X1 = 0; X2 = 0) thì D1 không sáng (F = 0). Khi chỉ nhấn S1 hoặc S2, hoặc đồng thời S1 và S2 (X1 = 0; X2 = 1 hoặc X1 = 1; X2 = 0; hoặc X1 = 1; X2 = 1) thì D1 sáng (F = 1).

2.2. Tạo cổng NOR từ BJT - FET



Cổng NAD họ NMOS như ở hình a); họ CMOS như ở hình b).

Nguyên lý hoạt động của chúng cơ bản giống nhau.

+ Cổng NOR họ NMOS:

- Khi 2 đầu vào nối xuống mức thấp, T2 và T3 tắt, đầu ra lên mức cao.

- Khi có 1 đầu vào nối lên mức cao, một trong 2 transistor T2 hoặc T3 dẫn, đầu ra xuống mức thấp.

Đó chính là kết quả của cổng NOR 2 đầu vào.

+ Cổng NOR họ CMOS:

- Khi 2 đầu vào nối xuống mức thấp, T1 và T2 dẫn, T3 và T4 tắt, đầu ra lên mức cao.

- Khi có 1 đầu vào nối lên mức cao, một trong 2 transistor T3 hoặc T4 dẫn, một trong 2 transistor T1 hoặc T2 tắt, đầu ra xuống mức thấp.

Đó chính là kết quả của cổng NAND 2 đầu vào.

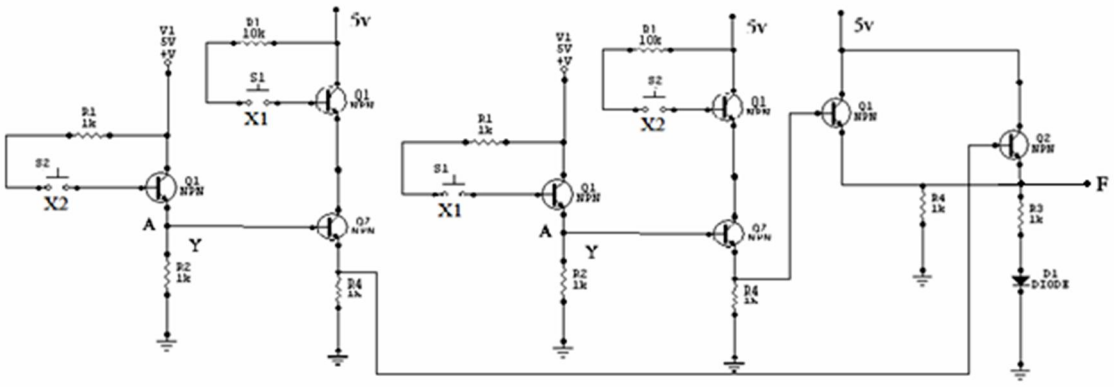
3. Mạch tạo thành cổng XOR (EXOR)

Mục tiêu:

Trình bày được sơ đồ nguyên lý, nguyên lý hoạt động của các mạch dùng đi - ốt, BJT, FET để tạo nên cổng XOR.

3.1. Tạo cổng XOR từ đi ốt - BJT

Sơ đồ nguyên lý thực hiện:



Sinh viên hãy vẽ mạch điện dùng:

- 8 BJT loại NPN như trên sơ đồ (hoặc PNP)
- 2 nút ấn S1, S2.
- 1 Logic Display [LED chỉ thị]
- Các điện trở có giá trị như trên sơ đồ nguyên lý.

Nguyên lý hoạt động:

Khi cùng hở S1 và S2 hoặc cùng nhấn (X1 = 0; X2 = 0) thì D1 không sáng (F = 0). Khi chỉ ấn S1 hoặc S2, hoặc đồng thời S1 và S2 (X1 = 0; X2 = 1 hoặc X1 = 1; X2 = 0; hoặc X1 = 1; X2 = 1) thì D1 sáng (F = 1).

3.2. Tạo cổng XOR từ BJT - FET

4. Mạch so sánh

Mục tiêu:

Trình bày được sơ đồ nguyên lý, nguyên lý hoạt động và đặc điểm của các mạch so sánh.

4.1. Sơ đồ nguyên lý mạch so sánh

4.1.1 Mạch so sánh 2 số 1 bit

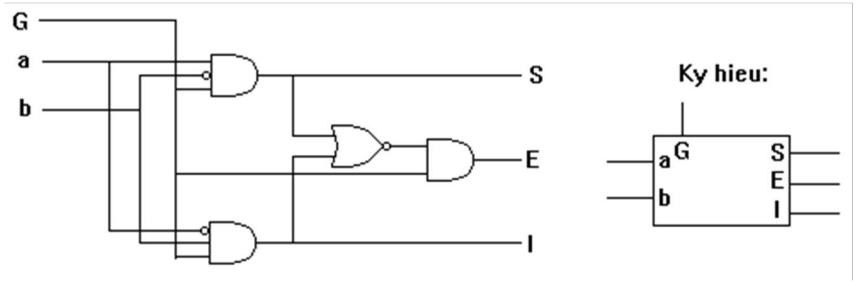
Bảng sự thật của mạch so sánh một bit có đầu vào cho phép (nối mạch) G :

G	a	b	S (a>b)	I (a<b)	E (a=b)
0	x	x	0	0	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	0	0	1

Hàm logic thực hiện:

$$S = G.(a.\bar{b})$$
$$I = G.(b.\bar{a})$$
$$E = G.(\overline{a \oplus b})$$

Mạch điện logic thực hiện:

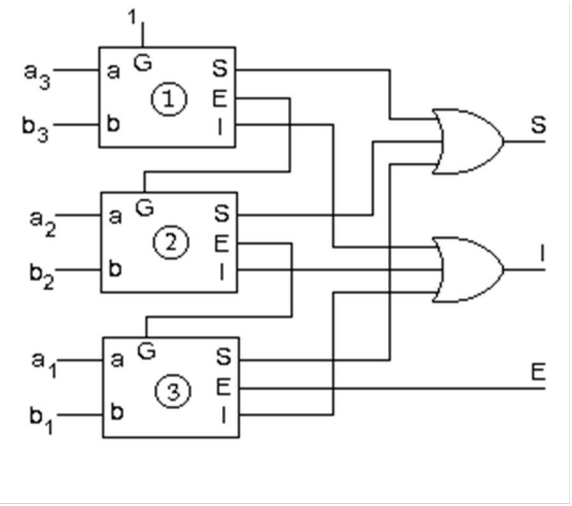


Từ mạch so sánh 1 bit ta có thể mở rộng để so sánh nhiều bit.

4.1.2 Mạch so sánh 2 số nhiều bit:

Để tạo mạch so sánh 2 số nhiều bit, trình tự làm giống như với 2 số 1 bit. Nghĩa là ta phải từ yêu cầu so sánh, lập ra bảng trạng thái làm việc, viết hàm logic và sơ đò logic thực hiện.

Dưới đây là sơ đồ logic của mạch so sánh 2 số 3 bit.



4.2. Đặc điểm và nguyên lý làm việc

Để so sánh 2 số nhiều bit, trước tiên người ta so sánh 2 bit cao nhất (MSB), kết quả lớn hoặc nhỏ hơn do 2 bit này quyết định, nếu 2 bit MSB bằng nhau người ta so sánh 2 bit có trọng số thấp hơn tiếp theo và kết quả được quyết định theo cách tương tự như ở 2 bit MSB. . . Sự so sánh được lặp lại cho đến bit LSB để được kết cuối cùng.

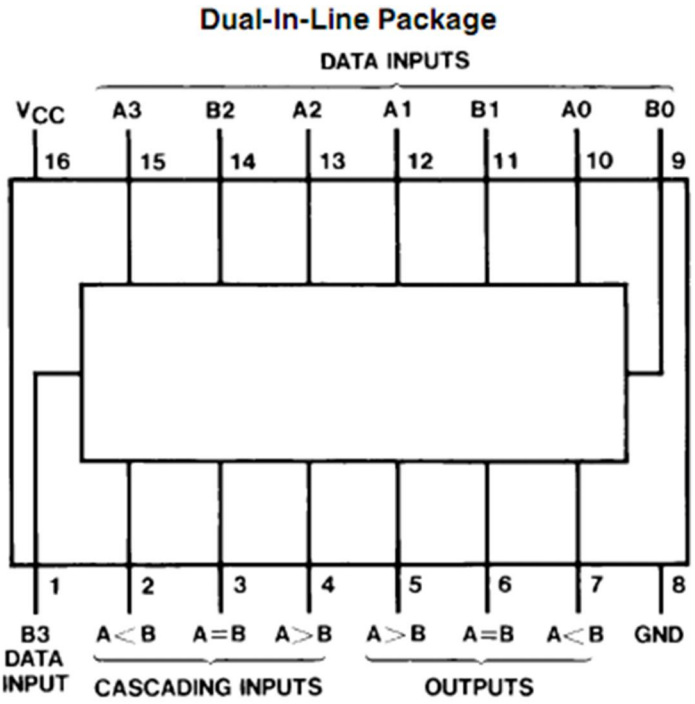
Từ sơ đồ logic thực hiện ở mục 4.1.2 ta có nguyên lý làm việc của mạch như sau:

- IC1 so sánh 2 bit cao (a3 & b3) nên đầu vào cho phép được đưa lên mức cao, nếu kết quả bằng nhau, đầu ra E của nó lên cao, cho phép IC2 so sánh, nếu kết quả lại bằng nhau, đầu ra E của IC2 lên cao cho phép IC3 so sánh, kết quả bằng nhau cuối cùng chỉ bởi đầu ra E của IC3.

- Các đầu vào cổng OR nhận tín hiệu từ các đầu ra S (hoặc I) sẽ cho kết quả lớn hơn (hoặc nhỏ hơn) tùy vào kết quả so sánh ở bất cứ bit nào. Thật vậy khi có một kết quả lớn hơn (hoặc nhỏ hơn) thì S (hoặc I) ở một IC lên cao, các đầu ra E và I (hoặc S) của các IC khác bằng 0, đây là điều kiện mở cổng OR để cho kết quả so sánh xuất hiện ở một trong các cổng OR này.

Trên thị trường có sẵn loại IC so sánh 4 bit 7485 có đầu nối mạch để mở rộng việc so sánh cho số nhiều bit hơn.

Sơ đồ đấu nối và bảng trạng thái làm việc của IC 7485 như ở dưới đây:



Bảng trạng thái làm việc của IC 7485

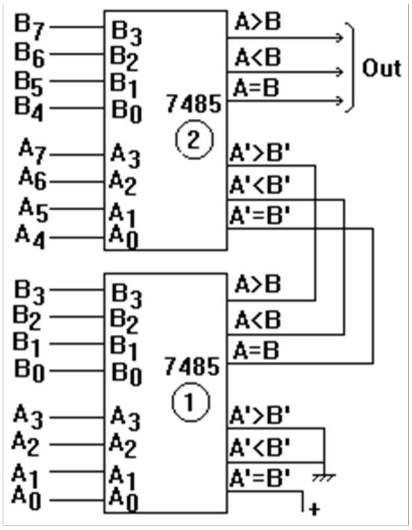
Trạng thái	Đầu vào so sánh				Đầu vào cho phép			Đầu ra		
	A3; B3	A2; B2	A1; B1	A0; B0	A' > B'	A' < B'	A' = B'	A > B	A < B	A = B
1	$A_3>B_3$	X	X	X	X	X	X	1	0	0
2	$A_3<B_3$	X	X	X	X	X	X	0	1	0
3	$A_3=B_3$	$A_2>B_2$	X	X	X	X	X	1	0	0
4	$A_3=B_3$	$A_2<B_2$	X	X	X	X	X	0	1	0
5	$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	X	X	X	X	1	0	0
6	$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	X	X	X	X	0	1	0
7	$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	X	X	X	1	0	0
8	$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	X	X	X	0	1	0
9	$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	0	1	0	0	1
10	$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	0	0	1	0	0
11	$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	1	0	0	1	0

Dựa vào bảng sự thật, ta thấy:

- Khi dùng IC 7485 để so sánh 2 số 4 bit ta phải giữ đầu vào nối mạch A'=B' ở mức cao, hai đầu vào nối mạch còn lại ở mức thấp, như vậy IC mới thể hiện được kết quả của trạng thái 9.

- Khi so sánh 2 số nhiều bit hơn ta phải dùng nhiều IC 7485 và nối đầu ra của IC so sánh bit thấp vào đầu vào nối mạch tương ứng của các IC so sánh các bit cao hơn và IC so sánh các bit thấp nhất có đầu vào nối mạch được mắc như khi dùng riêng lẻ. Để đọc được kết quả so sánh ta phải quan tâm tới các trạng thái 9, 10 và 11 trong bảng sự thật.

Cách mắc 2 IC 7485 để so sánh 2 số nhị phân 8 bit:



Ví dụ :

a. So sánh hai số A7 . . . .A0 = 10101111 và B7 . . . . B0 = 10110001

IC 2 so sánh các bit cao A7 . . .A4 = 1010 và B7 . . .B4 = 1011 có A7 = B7, A6 = B6, A5 = B5 và A4< B4 cho đầu ra A< B = 1 bất chấp trạng thái của các đầu vào nối mạch (trạng thái 8). Điều này có nghĩa là khi IC so sánh bit cao thấy có kết quả khác nhau giữa 2 số bit cao thì không quan tâm tới kết quả của bit thấp.

b. So sánh hai số A7 . . . .A0 = 10101111 và B7 . . . . B0 = 10101001

Trong trường hợp này kết quả hai số bit cao bằng nhau nên IC 2 nhìn vào đầu vào nối mạch để xem kết quả so sánh của IC1 (so sánh bit thấp), A3A2A1A0 = 1111 > B3B2B1B0 = 1001 nên đầu ra A > B = 1 để chỉ kết quả so sánh của 2 số 8 bit (trạng thái 10).

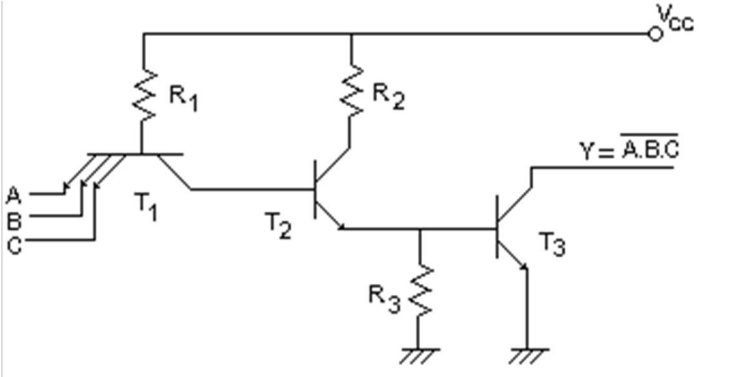
5. Mạch dùng cổng collector để hở

Mục tiêu:

Trình bày được sơ đồ nguyên lý, nguyên lý hoạt động và đặc điểm của các mạch dùng cổng collector để hở.

5.1. Sơ đồ nguyên lý

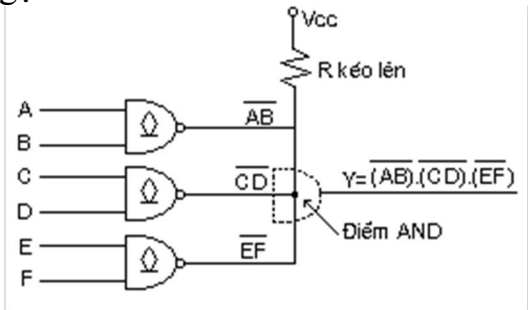
Xét sơ đồ nguyên lý của một cổng NAND 3 đầu vào có đầu ra có collector để hở.



5.2. Đặc điểm và nguyên lý làm việc

Đầu ra cực thu để hở có một số lợi điểm sau:

- Cho phép kết nối các đầu ra của nhiều cổng khác nhau, nhưng khi sử dụng phải mắc một điện trở từ đầu ra lên nguồn Vcc, gọi là điện trở kéo lên, trị số của điện trở này có thể được chọn lớn hay nhỏ tùy theo yêu cầu có lợi về mặt công suất hay tốc độ làm việc. Điểm nối chung của các đầu ra có tác dụng như một cổng AND nên ta gọi là điểm AND.



- Người ta cũng chế tạo các IC đầu ra có cực thu để hở cho phép điện trở kéo lên mắc vào nguồn điện thế cao, dùng cho các tải đặc biệt hoặc dùng tạo sự giao tiếp giữa họ TTL với CMOS dùng nguồn cao.

Nguyên lý làm việc của mạch giống như với một cổng NAND 3 đầu vào bình thường; chỉ khác là đầu ra phải nối với mọt điện trở phụ (điện trở kéo lên) và nối tới nguồn phụ để lấy được tín hiệu logic đầu ra.

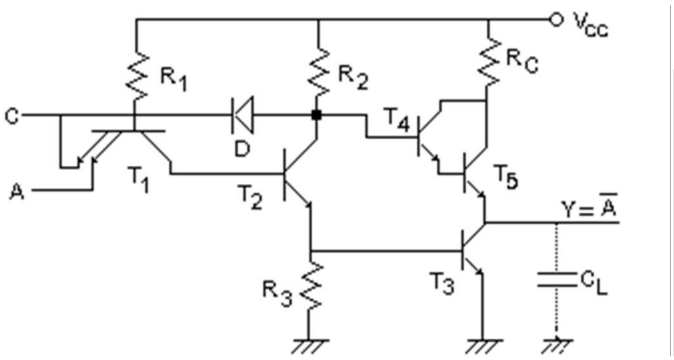
6. Mạch dùng cổng 3 trạng thái

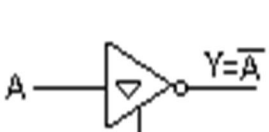
Mục tiêu:

Trình bày được sơ đồ nguyên lý, nguyên lý hoạt động và đặc điểm của các mạch dùng cổng 3 trạng thái.

6.1. Sơ đồ nguyên lý

Xét sơ đồ nguyên lý một cổng đảo có đầu ra 3 trạng thái. Ký hiệu của cổng đảo đầu ra 3 trạng thái, có đầu điều khiển C tác động mức cao và bảng sự thật:





C	A	Y
1	0	1
1	1	0
0	X	Z cao

Bảng sự thật

6.2. Đặc điểm và nguyên lý làm việc

Đặc điểm:

T4 & T5 được mắc Darlington để cấp dòng ra lớn cho tải. Diod D nối vào đầu vào C để điều khiển.

Nguyên lý hoạt động của mạch.

Hoạt động của mạch giải thích như sau:

- Khi C=1, Diod D tắt dẫn, mạch hoạt động như một cổng đảo
- Khi C=0, Diod D dẫn, cực thu T2 bị ghim áp ở mức thấp nên T3, T4 & T5 đều tắt, đầu ra mạch ở trạng thái tổng trở cao.

7. Thực hiện chuyển đổi từ cổng logic này sang cổng logic khác

Mục tiêu:

Trình bày được nguyên tắc chuyển đổi và chuyển đổi được giữa một số cổng logic thông dụng.

7.1 . Nguyên tắc chuyển đổi

Chúng ta đã thấy tất cả các hàm logic có thể được thay thế bởi 2 hàm duy nhất là hàm AND (hoặc OR) kết hợp với hàm NOT. Các cổng logic có chức năng thực hiện hàm logic, như vậy chúng ta chỉ cần dùng 2 cổng AND (hoặc OR) và NOT để thực hiện tất cả các cổng logic. Tuy nhiên, vì cổng NOT cũng có thể tạo ra từ cổng NAND (hoặc NOR). Như vậy, tất cả các hàm logic có thể được thực hiện bởi một cổng duy nhất, đó là cổng NAND (hoặc NOR). Hàm ý này cho phép chúng ta biến đổi qua lại giữa các cổng với nhau.

Quan sát Định lý De Morgan chúng ta rút ra qui tắc biến đổi qua lại giữa các cổng AND, NOT và OR , NOT như sau:

"Chỉ cần thêm các cổng đảo ở đầu vào và đầu ra khi biến đổi từ AND sang OR hoặc ngược lại".

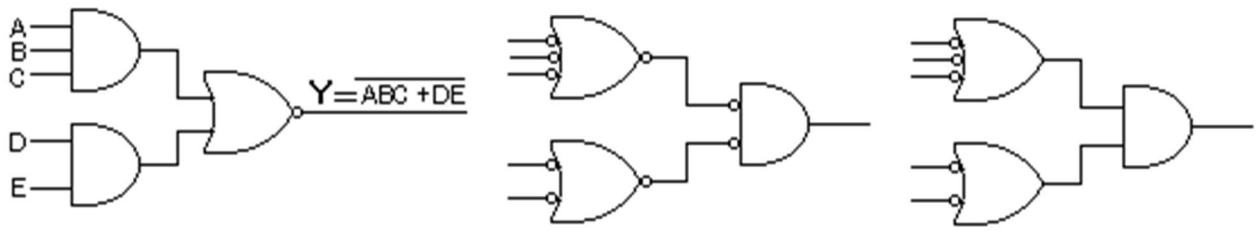
Dĩ nhiên nếu ở các đầu đã có đảo rồi thì đảo này sẽ mất đi.

7.2. Thực hành chuyển đổi

Bài thứ 1:

Ba mạch dưới đây tương đương nhau:

Hình b) có thể thực hiện nhiệm vụ của hình a) chỉ nhờ cách đổi AND - OR thêm các đảo ở các đầu vào và ra theo đúng luật De Morgan. Từ hình b) đổi sang hình c) ta bỏ 2 cổng đảo nối từ đầu ra cổng NOR đến đầu vào cổng AND



a) **b)** **c)**

Kiểm tra lại tính đúng đắn của các tương đương trên bằng cách đấu thực tế dùng các IC 7404; 7411; 7408; 7427; 7433; theo suy luận lý thuyết.

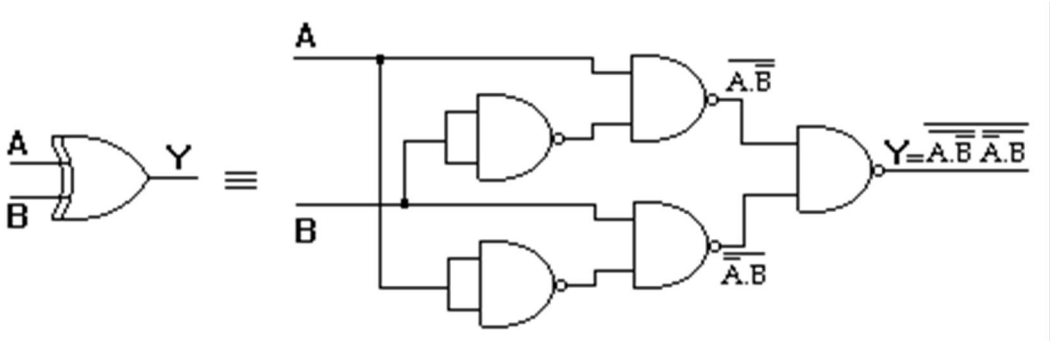
Bài thứ 2:

Thực hiện nhiệm vụ của cổng cổng EX-OR bằng cách chỉ dùng toàn cổng NAND.

Dùng định lý De-Morgan, biểu thức hàm EX-OR viết lại:

$$Y = A\bar{B} + \bar{A}B = \overline{\overline{A\bar{B}}.\overline{\bar{A}B}}$$

Mạch tương đương như ở dưới đây:



Kiểm tra lại tính đúng đắn của 2 sơ đồ trên bằng IC 7486 và 7400 đấu thực tế các cổng theo suy luận lý thuyết.

BÀI 4
MẠCH MÃ HÓA
Mã bài: MĐ26.04

Giới thiệu:

Mã hóa là gán các ký hiệu cho các đối tượng trong một tập hợp để thuận tiện cho việc thực hiện một yêu cầu cụ thể nào đó. Ví dụ mã BCD gán số nhị phân 4 bit cho từng số mã của số thập phân (từ 0 đến 9) để thuận tiện cho máy đọc một số có nhiều số mã; mã Gray dùng tiện lợi trong việc tối giản các hàm logic . . .. Mạch chuyển từ mã này sang mã khác gọi là mạch chuyển mã, cũng được xếp vào loại mạch mã hóa. Ví dụ mạch chuyển số nhị phân 4 bit sang số Gray là một mạch chuyển mã.

Bài học góp một phần vào việc thực hiện mục tiêu đào tạo của mô đun.

Mục tiêu:

- Trình bày được đặc tính, cấu trúc, nguyên lý làm việc và ứng dụng của mạch mã hoá.
- Phân biệt được sự khác nhau giữa mạch mã hoá và giải mã.
- Có tư duy, tác phong công nghiệp

Nội dung chính:

1. Khái niệm mạch mã hoá

Mục tiêu:

Trình bày được nguyên tắc mã hóa và một số loại mạch mã hóa.

1.1 . Khái niệm

Mạch mã hoá (ENCODER) là mạch có nhiệm vụ biến đổi những ký hiệu quen thuộc với con người sang những ký hiệu không quen thuộc con người.

1.2. Nguyên tắc và các loại mạch mã

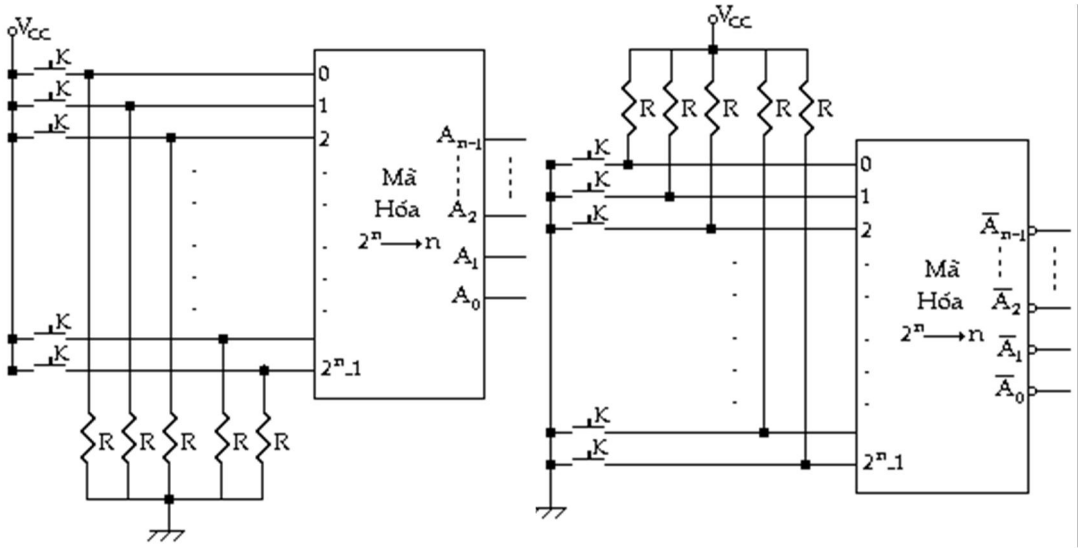
Một số nhị phân n bit cho 2^n tổ hợp số khác nhau. Vậy ta có thể dùng số n bit để mã cho 2^n đầu vào khác nhau, khi có một đầu vào được chọn bằng cách đưa nó lên mức tác động, ở đầu ra sẽ chỉ báo số nhị phân tương ứng. Đó là mạch mã hóa 2^n đường vào sang N đường đầu ra.

Dưới đây là mô hình một mạch mã hóa 2^n đường sang N đường.

Hình (a) là mạch có đầu vào và ra tác động cao: Khi các đầu vào đều ở mức thấp, mạch chưa hoạt động, các đầu ra đều ở mức thấp. Khi có một đầu vào được tác động bằng cách ấn khóa K tương ứng để đưa đầu vào đó lên mức cao, các đầu ra sẽ cho số nhị phân tương ứng.

- Hình (b) là mạch có đầu vào và ra tác động thấp. Hoạt động tương tự như mạch trên nhưng có mức tác động ngược lại. Trong mô hình (b) ký hiệu dấu o ở đầu ra để chỉ mức tác động thấp, còn ở đầu vào không có dấu o vì là mạch thật.

Trong trường hợp đầu ra có mức tác động thấp, muốn đọc đúng số nhị phân ở đầu ra, ta phải đảo các bit để đọc.



(a) **(b)**

Dĩ nhiên, người ta cũng có thể thiết kế theo kiểu đầu vào tác động thấp và đầu ra tác động cao hay ngược lại. Trên thực tế, ta có thể có bất cứ loại đầu vào hay ra tác động theo bất cứ kiểu nào (mức cao hay thấp).

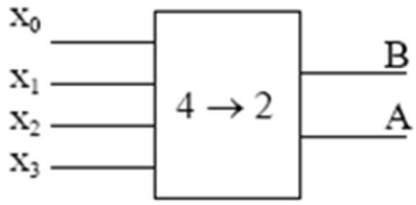
Chúng ta có thể có các loại mã nhị phân, thập phân.

Ngoài ra, để tránh trường hợp mạch cho ra một mã sai khi người sử dụng vô tình (hay cố ý) tác động đồng thời vào hai hay nhiều đầu vào, người ta thiết kế các mạch mã hóa ưu tiên: là mạch chỉ cho ra một mã duy nhất có tính ưu tiên khi có nhiều đầu vào cùng được tác động.

Mạch mã hoá ưu tiên

Trong các mạch mã hóa thường, tín hiệu đầu vào tồn tại độc lập tức là không có tình huống có 2 tín hiệu trở lên đồng thời tác động ở mức logic 1 (nếu ta chọn mức tích cực ở đầu vào là mức logic 1), do đó cần phải đặt ra vấn đề ưu tiên.

Vấn đề ưu tiên: Khi có nhiều tín hiệu đồng thời tác động, tín hiệu nào có mức ưu tiên cao hơn ở thời điểm đang xét sẽ tác động, tức là nếu đầu vào có độ ưu tiên cao hơn bằng 1 trong khi những đầu vào có độ ưu tiên thấp hơn nếu bằng 1 thì mạch sẽ tạo ra từ mã nhị phân ứng với đầu vào có mức độ ưu tiên cao nhất. Xét mạch mã hoá ưu tiên 4 → 2 (4 đầu vào, 2 đầu ra):



Bảng trạng thái mô tả hoạt động của mạch

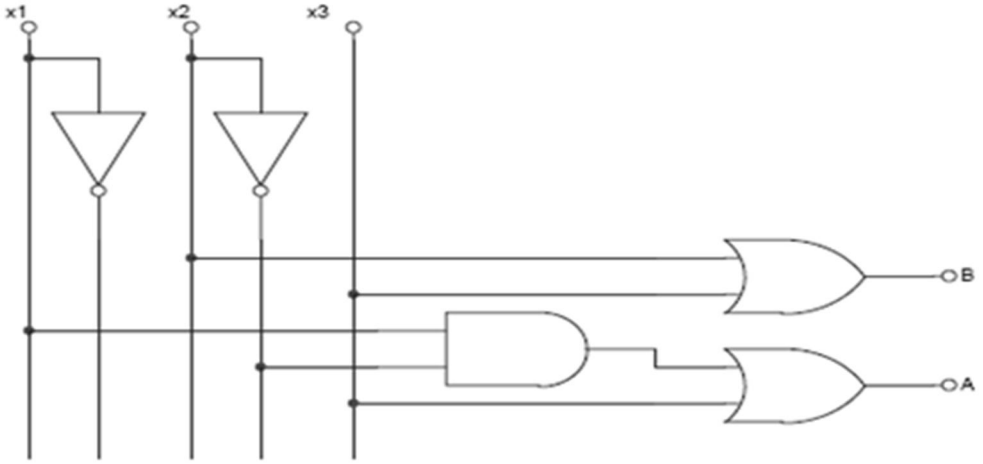
x_0	x_1	x_2	x_3	B	A
1	0	0	0	0	0
x	1	0	0	0	1
x	x	1	0	1	0
x	x	x	1	1	1

Phương trình tối giản :

$$A = x_1.\overline{x_2}.\overline{x_3} + x_3 = x_1.\overline{x_2} + x_3$$

$$B = x_2.\overline{x_3} + x_3 = x_2 + x_3$$

Sơ đồ logic:



Một số vi mạch mã hóa thông dụng: 74LS147, 74LS148.

2. Mạch mã hoá 8 đầu vào thành 3 đầu ra (mã hóa 8/3) dùng cổng logic cơ bản

Mục tiêu:

Trình bày được phương pháp và xây dựng được mạch mã hóa 8/3.

Lắp ráp và cân chỉnh được mạch mã hóa 8/3.

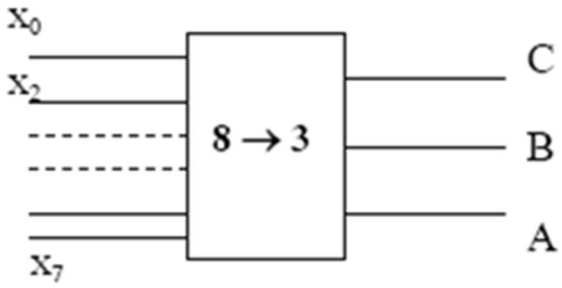
2.1. Phương pháp mã 8/3

Từ 8 bit nhị phân đầu vào dùng mạch mã hóa để chuyển thành 3 đầu ra thập phân (hoặc gọi là bát phân) theo thứ tự:

- Vẽ sơ đồ khối.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic.
- Vẽ sơ đồ logic thực hiện.

2.2. Mạch mã hóa 8/3 dùng cổng Logic cơ bản

Xét mạch mã hóa nhị phân từ 8 sang 3 (8 đầu vào và 3 đầu ra). Sơ đồ khối của mạch:

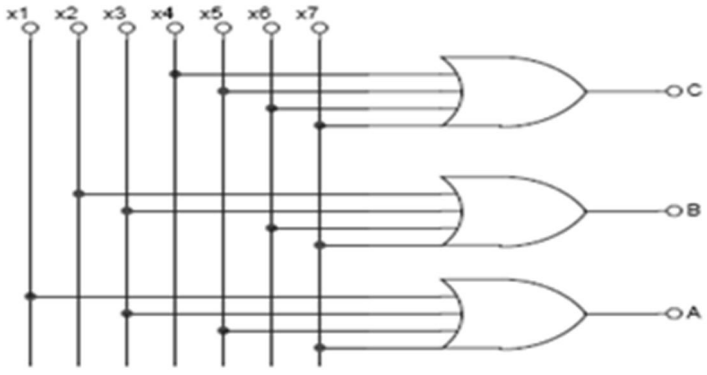


Trong đó:
- x0, x1,. . ., x7 là các đầu vào tín hiệu.
- A, B, C là các đầu ra.

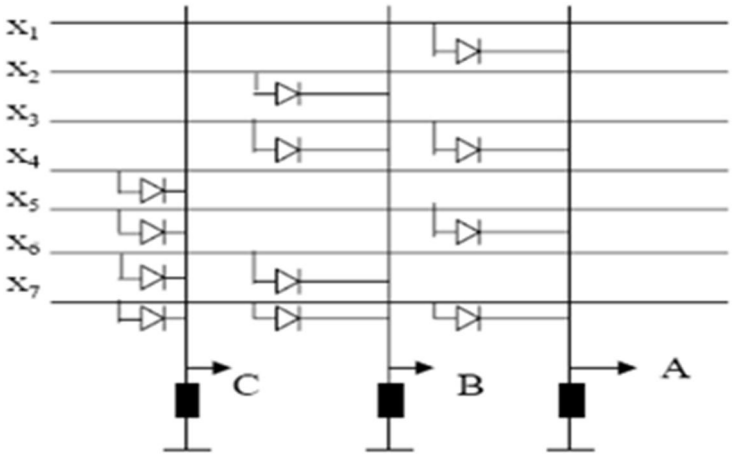
Mạch mã hóa nhị phân thực hiện biến đổi tín hiệu đầu vào thành một từ mã nhị phân tương ứng ở đầu ra, cụ thể như sau:
0 → 000 3 → 011 6 → 100
1 → 001 4 → 100 7 → 111
2 → 010 5 → 101
Chọn mức tác động (tích cực) ở đầu vào là mức logic 1, ta có bảng trạng thái mô tả hoạt động của mạch :

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	C	B	A
1	0	0	0	0	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0	0	1
0	0	**1**	0	0	0	0	0	0	1	0
0	0	0	**1**	0	0	0	0	0	1	1
0	0	0	0	**1**	0	0	0	1	0	0
0	0	0	0	0	**1**	0	0	1	0	1
0	0	0	0	0	0	**1**	0	1	1	0
0	0	0	0	0	0	0	**1**	1	1	1

Giải thích bảng trạng thái: Khi một đầu vào ở trạng thái tích cực (mức logic 1) và các đầu vào còn lại không được tích cực (mức logic 0) thì đầu ra xuất hiện từ mã tương ứng. Cụ thể là: khi đầu vào x0=1 và các đầu vào còn lại bằng 0 thì từ mã ở đầu ra là 000, khi đầu vào x1=1 và các đầu vào còn lại bằng 0 thì từ mã nhị phân ở đầu ra là 001, .. v.v. Phương trình logic tối giản:
A = x1 + x3 + x5 + x7 , B = x2 + x3 + x6 + x7 , C= x4 + x5 + x6 + x7
Sơ đồ logic:



Biểu diễn bằng cổng logic dùng Diode:



Nếu chúng ta chọn mức tác động tích cực ở đầu vào là mức logic 0, bảng trạng thái mô tả hoạt động của mạch lúc này như sau:

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	C	B	A
0	1	1	1	1	1	1	1	0	0	0
1	**0**	1	1	1	1	1	1	0	0	1
1	1	**0**	1	1	1	1	1	0	1	0
1	1	1	**0**	1	1	1	1	0	1	1
1	1	1	1	**0**	1	1	1	1	0	0
1	1	1	1	1	**0**	1	1	1	0	1
1	1	1	1	1	1	**0**	1	1	1	0
1	1	1	1	1	1	1	**0**	1	1	1

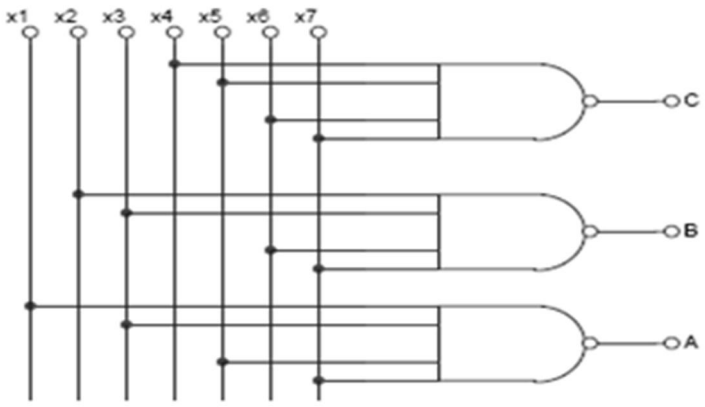
Phương trình logic tối giản :

$$A = \overline{x}_1 + \overline{x}_3 + \overline{x}_5 + \overline{x}_7 = \overline{x_1 x_3 x_5 x_7}$$

$$B = \overline{x}_2 + \overline{x}_3 + \overline{x}_6 + \overline{x}_7 = \overline{x_2 x_3 x_6 x_7}$$

$$C = \overline{x}_4 + \overline{x}_5 + \overline{x}_6 + \overline{x}_7 = \overline{x_4 x_5 x_6 x_7}$$

Sơ đồ mạch thực hiện:



2.3. Lắp ráp và cân chỉnh mạch mã hóa 8/3.

2.3.1 Giới thiệu sơ đồ nguyên lý

Sơ đồ nguyên lý như ở phần 2.2. Khi lắp ráp theo sơ đồ dùng các cổng NAND, các cổng NAND 4 đầu vào được thay bằng các cặp cổng NAND tương ứng khi sử dụng IC 74LS03.

2.3.2 Vật tư cần thiết:

Tổng hợp từ sơ đồ nguyên lý.

2.3.3 Thực hành lắp ráp:

2.3.3.1 Lắp mạch mã hóa 8/3 dùng đi ốt trên bo mạch đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiệc trước khi dùng làm phần tử kết nối trong mạch.
- Bố trí các linh kiện hợp lý
- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.
- Mối hàn phải ngấu, bóng.
- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đếm.
- Mạch phải hoạt động khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ bên ngoài.

2.3.3.2 Lắp mạch mã hóa 8/3 dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp thiết kế (không có đoạn bị đứt).
- Có sơ đồ lắp ráp kèm theo.
- Khi lắp phải cắm đúng vị trí và chiều các linh kiện theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, gọn không gây chạm chập trên mạch in.
- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đếm.
- Mạch phải hoạt động khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ bên ngoài.

2.3.2 Hiệu chỉnh mạch.

Dựa vào sơ đồ nguyên lý để kiểm tra khả năng hoạt động của mạch.

3. Mạch mã hoá 10 đầu vào thành 4 đầu ra (mã hóa 10/4) dùng IC TTL

Mục tiêu:

Trình bày được phương pháp và xây dựng được mạch mã hóa 10/4.

Lắp ráp và cân chỉnh được mạch mã hóa 10/4.

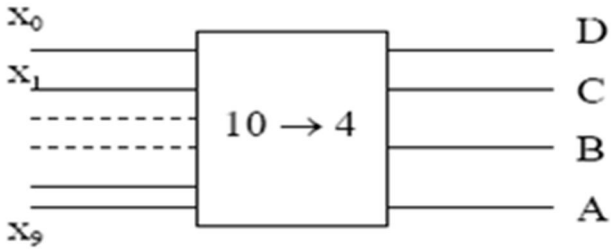
3.1. Phương pháp mã hóa 10/4

Từ 10 bit nhị phân đầu vào dùng mạch mã hóa để chuyển thành 4 đầu ra thập phân theo thứ tự:

- Vẽ sơ đồ khối.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic.
- Vẽ sơ đồ logic thực hiện.

3.2 . Mạch mã hóa 10/4 dùng IC TTL

Xét mạch mã hóa nhị phân từ 10 sang 4. Sơ đồ khối của mạch:



4

Bảng trạng thái mô tả hoạt động của mạch :

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	D	C	B	A
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0	0	0	0	0	1
0	0	**1**	0	0	0	0	0	0	0	0	0	1	0
0	0	0	**1**	0	0	0	0	0	0	0	0	1	1
0	0	0	0	**1**	0	0	0	0	0	0	1	0	0
0	0	0	0	0	**1**	0	0	0	0	0	1	0	1
0	0	0	0	0	0	**1**	0	0	0	0	1	1	0
0	0	0	0	0	0	0	**1**	0	0	0	1	1	1
0	0	0	0	0	0	0	0	**1**	0	1	0	0	0
0	0	0	0	0	0	0	0	0	**1**	1	0	0	1

Phương trình logic đã tối giản:

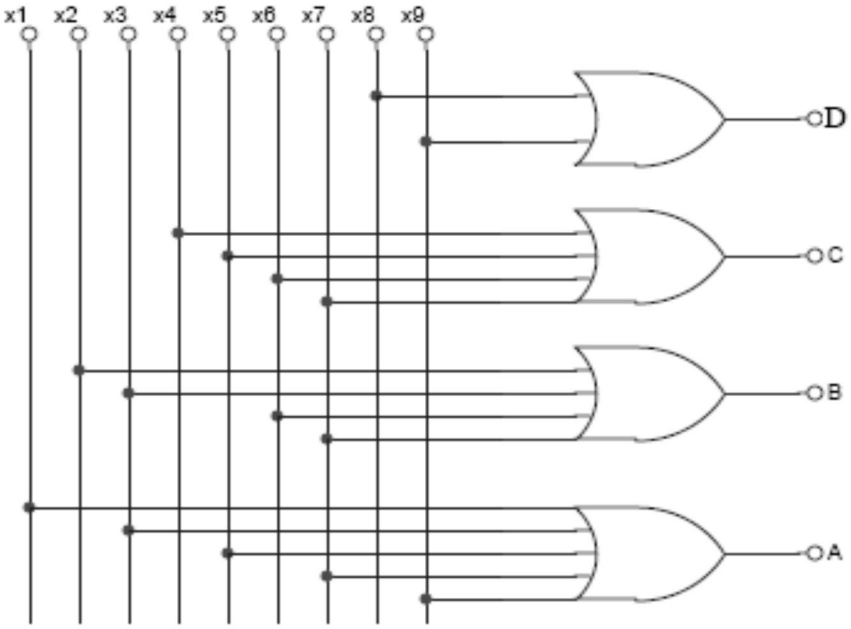
A = x1 + x3 + x5 + x7 + x9

B = x2 + x3 + x6 + x7

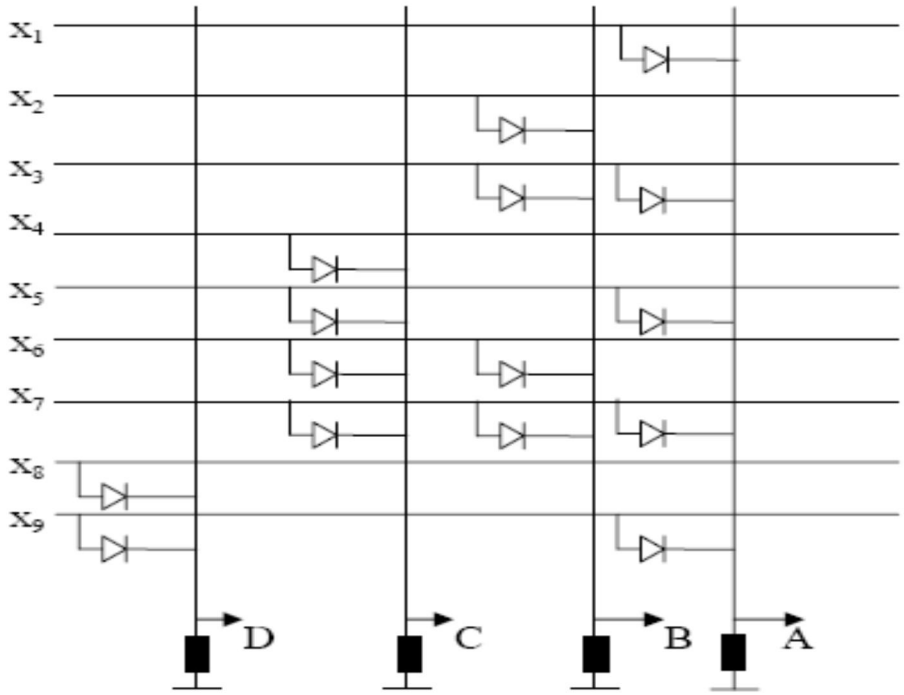
C = x4 + x5 + x6 + x7

D = x8 + x9

Biểu diễn bằng sơ đồ logic



Biểu diễn bằng cổng logic dùng Diode :



3.3. Lắp ráp và cân chỉnh mạch mã hóa 10/4 dùng IC TTL.

3.3.1 Giới thiệu sơ đồ nguyên lý

Sơ đồ nguyên lý như ở phần 3.2.

Khi lắp ráp, các cổng OR 4, 5 đầu vào có thể thay thế bằng các cặp cổng OR 2 đầu vào tương ứng khi sử dụng IC 74LS32.

3.3.2 Vật tư cần thiết:

Tổng hợp từ sơ đồ nguyên lý.

3.3.3 Thực hành lắp ráp:

3.3.3.1 Lắp mạch mã hóa dùng đi ốt trên bo mạch đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiệc trước khi dùng làm phần tử kết nối trong mạch.
- Bố trí các linh kiện hợp lý
- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.
- Mối hàn phải ngấu, bóng.
- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).
- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

3.3.3.2 Lắp mạch mã hóa dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp đã được thiết kế trước (không có đoạn mạch in nào được thiết kế là liên mà lại bị đứt trên bo mạch in chuẩn bị cho lắp ráp).

- Có sơ đồ lắp ráp kèm theo.

- Khi lắp phải cắm đúng vị trí và chiều các linh kiện (đặc biệt là các linh kiên bán dẫn) theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, bóng, gọn, không gây chạm chập trên mạch in.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

3.3.2 Hiệu chỉnh mạch.

Dựa vào sơ đồ nguyên lý để kiểm tra khả năng hoạt động của mạch.

Tìm và thay thế các IC số có số lượng đầu vào khác nhau, kiểm tra và so sánh kết quả.

BÀI 5
MẠCH GIẢI MÃ
Mã bài: MĐ26.05

Giới thiệu:

Giải mã là việc mạch làm nhiệm vụ biến đổi những ký hiệu không quen thuộc với con người sang những ký hiệu quen thuộc với con người. Ví dụ giải mã BCD là việc chuyển các số nhị phân 4 bit cho từng số mã thành các số thập phân (từ 0 đến 9) để thuận tiện cho việc giao tiếp với con người.

Bài học góp một phần vào việc thực hiện mục tiêu đào tạo của mô đun.

Mục tiêu:

- Mô tả được cấu trúc của mạch giải mã
- Trình bày được đặc tính và nguyên lý của mạch giải mã
- Phân biệt được sự khác nhau giữa mạch mã hoá và giải mã.
- Trình bày được ứng dụng của mạch giải mã.
- Có tư duy, tác phong trong công nghiệp

Nội dung chính:

1. Khái niệm mạch giải mã

Mục tiêu:

Mô tả được cấu trúc của mạch giải mã

1.2 . Khái niệm

Mạch giải mã (DECODER) là mạch làm nhiệm vụ biến đổi những ký hiệu không quen thuộc với con người sang những ký hiệu quen thuộc với con người.

1.2. Nguyên tắc xây dựng và các loại mạch giải mã

Nguyên tắc chung là: Từ N giá trị thập phân đầu vào dùng mạch giải mã để chuyển thành 2^n đầu ra nhị phân, theo thứ tự:

- Vẽ sơ đồ khối.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic (tối thiểu hóa nếu cần).
- Vẽ sơ đồ logic thực hiện.

Mạch giải mã có thể có các loại:

+ Theo quan hệ giữa đầu vào và đầu ra có: Mạch giải mã mà $N = 2^n$; $N \neq 2^n$.

+ Theo phương pháp hiển thị kết quả đầu ra: Giải mã nhị phân và giải mã hiển thị LED 7 thanh.

2. Mạch giải mã 2 đầu vào thành 4 đầu ra dùng cổng logic căn bản

Mục tiêu:

Trình bày được phương pháp và nguyên lý làm việc của mạch điện giải mã 2/4 dùng cổng logic cơ bản.

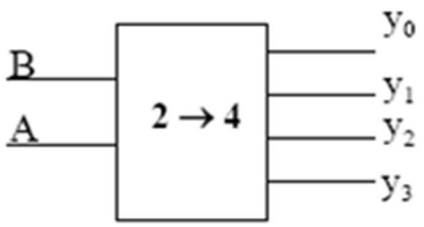
2.1. Phương pháp giải mã

Từ 2 giá trị thập phân đầu vào dùng mạch giải mã để chuyển thành 4 đầu ra nhị phân, theo thứ tự:

- Vẽ sơ đồ khối.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic (tối thiểu hóa nếu cần).
- Vẽ sơ đồ logic thực hiện.

2.2. Mạch điện giải mã dùng cổng Logic cơ bản

Xét mạch giải mã nhị phân 2→4 (2 đầu vào, 4 đầu ra) như trên hình vẽ 4.11. Chọn mức tích cực ở đầu ra là mức logic 1.



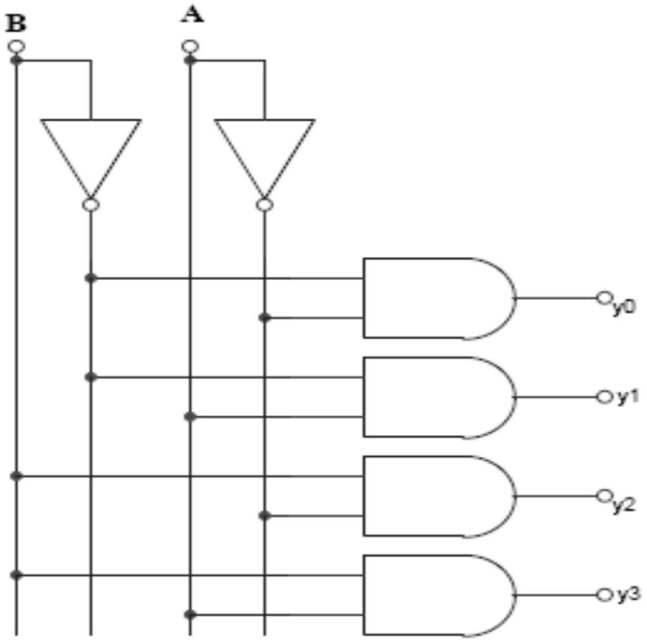
Bảng trạng thái mô tả hoạt động của mạch

B	A	y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

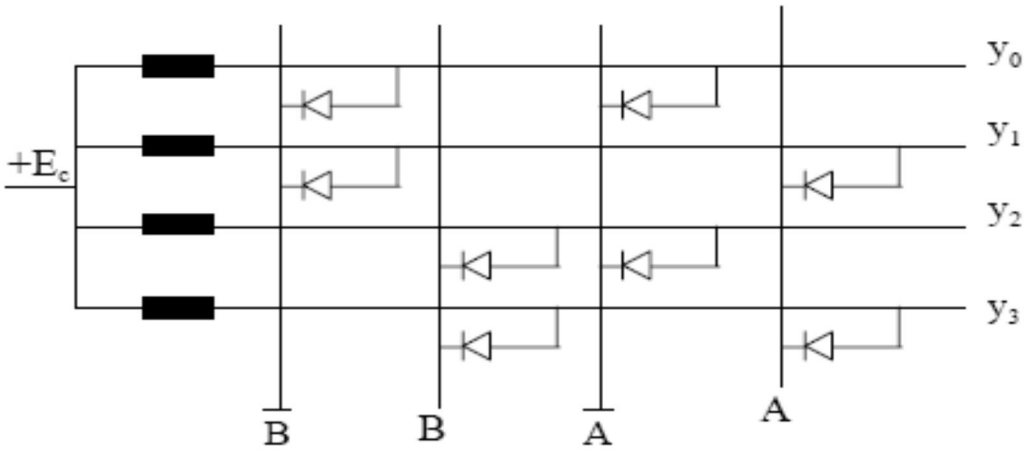
Phương trình logic tối giản :

$$y_0 = \overline{B}.\overline{A} \quad y_1 = \overline{B}.A$$
$$y_2 = B.\overline{A} \quad y_3 = A.B$$

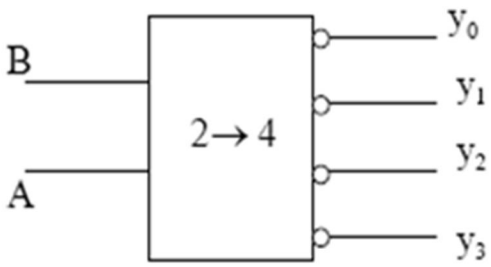
Sơ đồ logic:



Biểu diễn bằng cổng logic dùng Diode.



Trường hợp chọn mức tích cực ở đầu ra là mức logic 0 (mức logic thấp L):



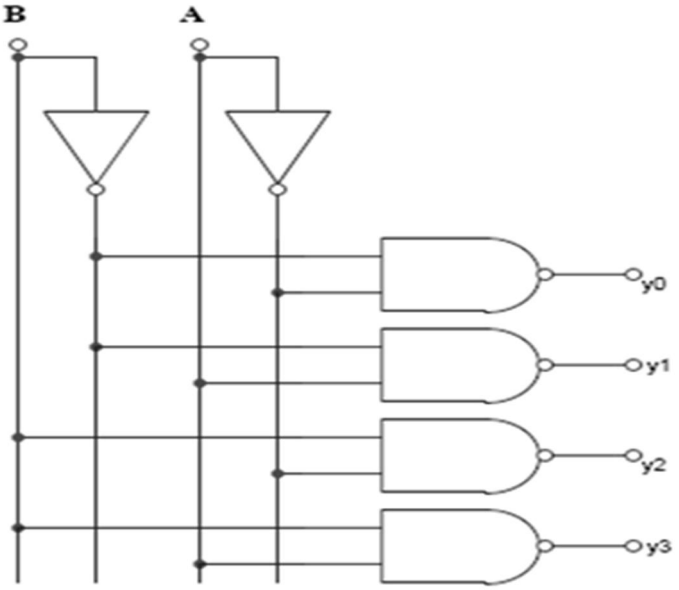
Bảng trạng thái mô tả hoạt động của mạch

B	A	y_0	y_1	y_2	y_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Phương trình logic:

$$y_0 = B + A = \overline{\overline{B}.\overline{A}}$$
$$y_1 = B + \overline{A} = \overline{\overline{B}.A}$$
$$y_2 = \overline{B} + A = \overline{B.\overline{A}}$$
$$y_3 = \overline{B} + A = \overline{B.A}$$

Sơ đồ logic:



3. Mạch giải mã 4 đầu vào thành 10 đầu ra dùng IC TTL

Mục tiêu:

Trình bày được phương pháp và nguyên lý làm việc của mạch điện giải mã 4/10 dùng cổng logic cơ bản.

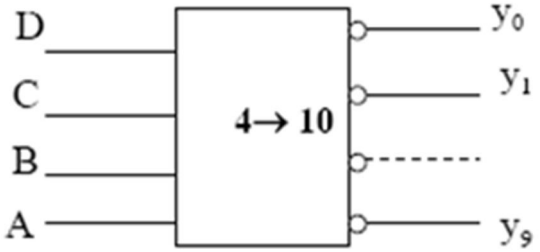
3.1. Phương pháp giải mã

Từ 4 giá trị thập phân đầu vào dùng mạch giải mã để chuyển thành 10 đầu ra nhị phân, theo thứ tự:

- Vẽ sơ đồ khối.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic (tối thiểu hóa nếu cần).
- Vẽ sơ đồ logic thực hiện.

3.2. Sơ đồ mạch dùng IC TTL

Sơ đồ khối của mạch giải mã 4/10



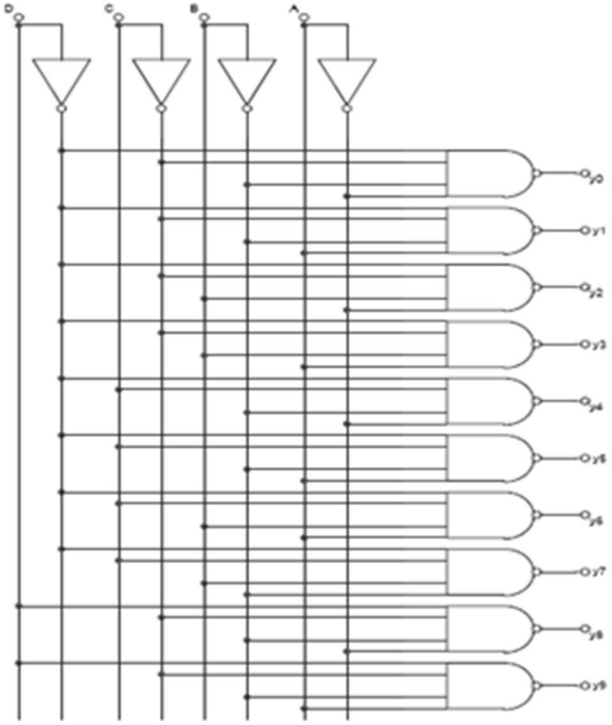
Chọn mức tích cực ở đầu ra là mức logic 1, lúc đó bảng trạng thái hoạt động của mạch như sau:

D	C	B	A	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
0	0	0	0	**1**	0	0	0	0	0	0	0	0	0
0	0	0	1	0	**1**	0	0	0	0	0	0	0	0
0	0	1	0	0	0	**1**	0	0	0	0	0	0	0
0	0	1	1	0	0	0	**1**	0	0	0	0	0	0
0	1	0	0	0	0	0	0	**1**	0	0	0	0	0
0	1	0	1	0	0	0	0	0	**1**	0	0	0	0
0	1	1	0	0	0	0	0	0	0	**1**	0	0	0
0	1	1	1	0	0	0	0	0	0	0	**1**	0	0
1	0	0	0	0	0	0	0	0	0	0	0	**1**	0
1	0	0	1	0	0	0	0	0	0	0	0	0	**1**

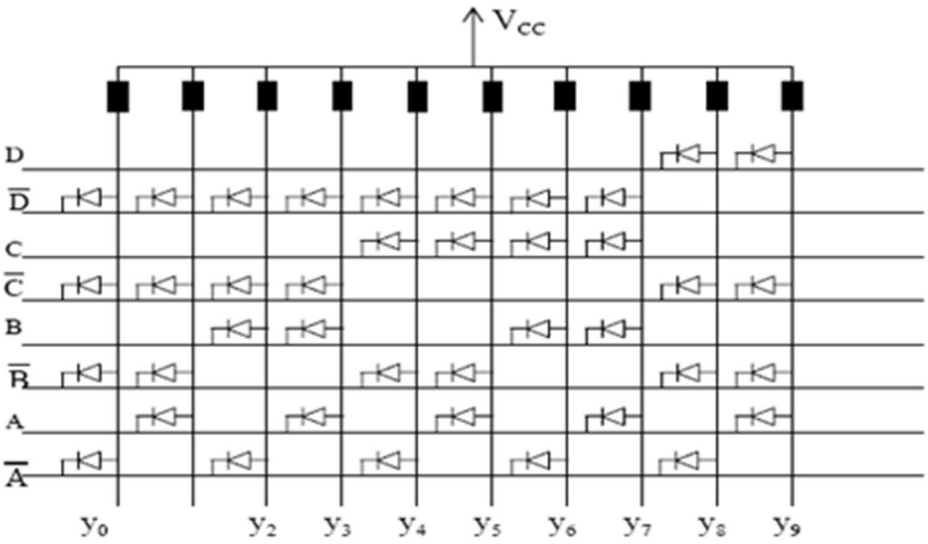
Phương trình logic:

$$y_0 = \overline{D}\overline{C}\overline{B}\overline{A} \qquad y_1 = \overline{D}\overline{C}\overline{B}A \qquad y_2 = \overline{D}\overline{C}B\overline{A} \qquad y_3 = \overline{D}\overline{C}BA$$
$$y_4 = \overline{D}C\overline{B}\overline{A} \qquad y_5 = \overline{D}C\overline{B}A \qquad y_6 = \overline{D}CB\overline{A} \qquad y_7 = \overline{D}CBA$$
$$y_8 = D\overline{C}\overline{B}\overline{A} \qquad y_9 = D\overline{C}\overline{B}A$$

Sơ đồ thực hiện mạch giải mã:



Sơ đồ thực hiện bằng cổng logic



Sơ đồ thực hiện bằng diode

4. Bộ giải mã từ BCD thành 7 đoạn (LED) hiển thị

Mục tiêu:

Trình bày được phương pháp và nguyên lý làm việc của mạch điện giải mã 4/10 hiển thị bằng LED 7 thanh dùng cổng logic cơ bản.

4.1. Phương pháp giải mã từ BCD sang hiển thị bằng LED 7 đoạn.

Từ 4 giá trị (4 bit) nhị phân đầu vào dùng mạch giải mã để chuyển thành 7 đầu ra logic cung cấp cho mạch hiển thị bằng LED 7 đoạn, theo thứ tự:

- Vẽ sơ đồ khối thực hiện.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic (tối thiểu hóa nếu cần).
- Vẽ sơ đồ logic thực hiện.

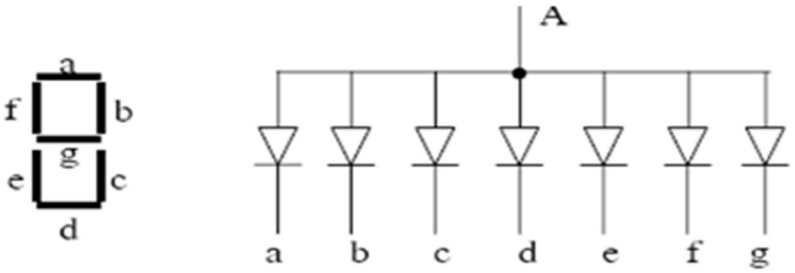
Nếu dùng IC giải mã, lựa chọn đúng IC tương ứng với loại LED 7 đoạn đang được sử dụng (a nốt chung hay ka tốt chung).

4.2. Sơ đồ mạch dùng IC TTL

4.2.1 Giới thiều về LED 7 đoạn:

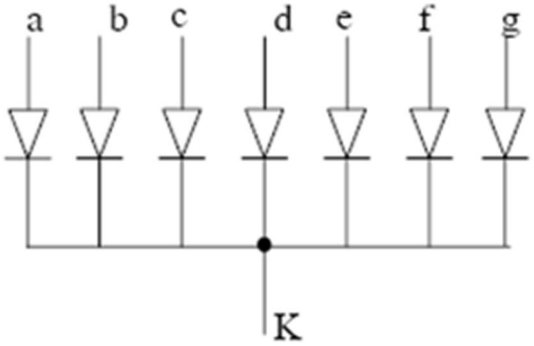
Đèn LED 7 đoạn, mỗi đoạn là 1 đèn LED. Tuỳ theo cách nối các Kathode hoặc các Anode của các LED trong đèn, mà người ta phân thành hai loại:

LED 7 đoạn loại Anode chung:



LED bảy đoạn loại Anode chung

LED 7 đoạn loại Kathode chung :

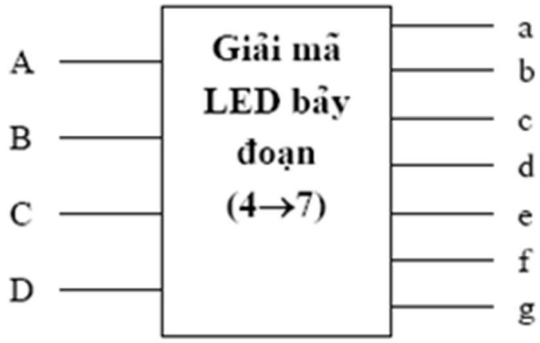


LED bảy đoạn loại Kathode chung

4.2.2 Mạch giải mã LED 7 đoạn:

Ứng với mỗi loại LED khác nhau ta có một mạch giải mã riêng.

Sơ đồ khối của mạch giải mã LED 7 đoạn như sau:



Sơ đồ khối mạch giải mã LED bảy đoạn

a. Xét đèn LED 7 đoạn loại Anode chung:

Đối với LED bảy đoạn loại anode chung, vì các anode của các đoạn led được nối chung với nhau và đưa lên mức logic 1 (5V), nên muốn đoạn led nào tắt ta nối kathode tương ứng lên mức logic 1 (5V) và ngược lại muốn đoạn led nào sáng ta nối kathode tương ứng xuống mass (mức logic 0).

Ví dụ: Để hiển thị số 0 ta nối kathode của đèn g lên mức logic 1 để đèn g tắt, và nối các kathode của đèn a, b, c, d, e, f xuống mass nên ta thấy số 0.

Lúc đó bảng trạng thái mô tả hoạt động của mạch giải mã LED bảy đoạn loại Anode chung như sau:

D	B	C	A	a	b	c	d	e	f	g	Số 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
1	0	1	0	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X

Dùng bảng Các - nô để tối thiểu hóa mạch trên. Phương trình tối thiểu hóa có thể viết ở dạng chính tắc 1 (tổng của các tích số) hoặc dạng chính tắc 2 (tích của các tổng số):

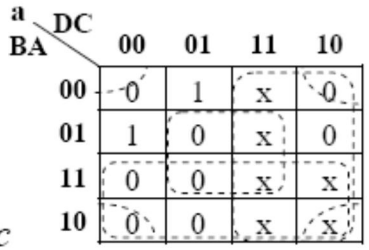
Phương trình logic của ngõ ra **a**:

Dạng chính tắc 2:

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

Dạng chính tắc 1:

$$\mathbf{a} = C\overline{B}A + \overline{D}C\overline{B}A$$



Lưu ý: Trên bảng Karnaugh chúng ta đã thực hiện tối thiểu hóa theo dạng chính tắc 2.

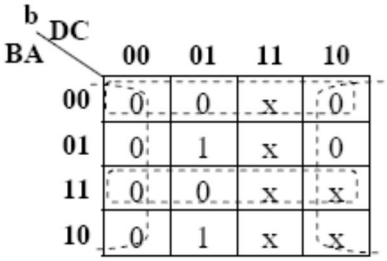
Phương trình logic của ngõ ra **b**:

Dạng chính tắc 2:

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

Dạng chính tắc 1:

$$\mathbf{b} = C\overline{B}A + CB\overline{A} = C(A \oplus B)$$



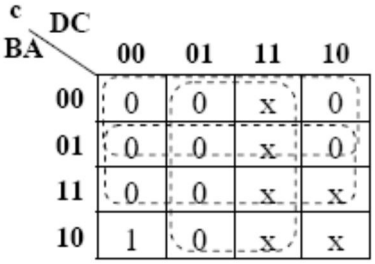
Phương trình logic của ngõ ra **c**:

Dạng chính tắc 2:

$$\mathbf{c} = B\overline{A}\overline{C}$$

Dạng chính tắc 1:

$$\mathbf{c} = \overline{D}\overline{C}B\overline{A}$$



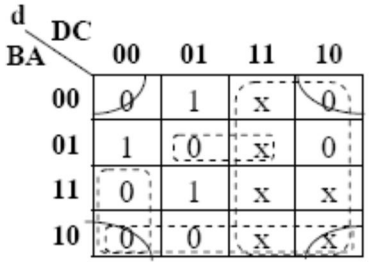
Phương trình logic của ngõ ra **d**:

Dạng chính tắc 2:

$$\mathbf{d} = \overline{D}(\overline{A}+B+\overline{C})(\overline{B}+C+D)(A+\overline{B})(A+C)$$
$$= \overline{A}\overline{B}C\overline{D} + ABC\overline{D} + A\overline{B}\overline{C}\overline{D}$$

Dạng chính tắc 1:

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



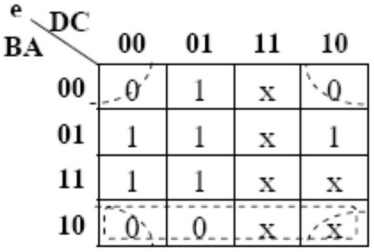
Phương trình logic của ngõ ra **e**:

Dạng chính tắc 2:

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

Dạng chính tắc 1:

$$\mathbf{e} = C\overline{B} + A$$



Phương trình logic của ngõ ra **f**:

Dạng chính tắc 2:

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

Dạng chính tắc 1:

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

f / BA \ DC	00	01	11	10
00	0	0	x	0
01	1	0	x	0
11	1	1	x	x
10	1	0	x	x

Phương trình logic của ngõ ra **g**:

Dạng chính tắc 2:

$$\mathbf{g} = \overline{D}(A+\overline{B})(\overline{C}+B)(\overline{B}+C)$$
$$= \overline{B}\overline{C}\overline{D} + \overline{D}CBA$$

Dạng chính tắc 1:

$$\mathbf{g} = \overline{D}CBA + \overline{D}\overline{C}\overline{B}$$

g / BA \ DC	00	01	11	10
00	1	0	x	0
01	1	0	x	0
11	0	1	x	x
10	0	0	x	x

b. Xét mạch giải mã đèn led 7 đoạn loại Kathode chung:

Chọn mức tích cực ở đầu ra là mức logic 1. Vì Kathode của các đoạn led được nối chung và được nối xuống mức logic 0 (0V-mass) nên muốn đoạn led nào tắt ta đưa Anode tương ứng xuống mức logic 0 (0V-mass).

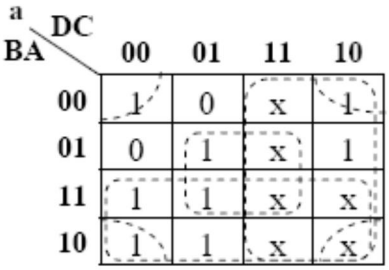
Ví dụ: Để hiển thị số 0 ta nối Anode của đoạn led g xuống mức logic 0 để đoạn g tắt, đồng thời các kathode của đoạn a, b, c, d, e, f được nối lên nguồn nên các đoạn này sẽ sáng do đó ta thấy số 0.

Lúc đó bảng trạng thái mô tả hoạt động của mạch như sau:

D	B	C	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

Tương tự như trường hợp trên, ta cũng dùng bảng Các - nô để tối thiểu hóa hàm mạch và đi tìm phương trình logic tối giản các đầu ra của các đoạn led: *(Lưu ý trong những sơ đồ Các - nô sau ta thực hiện tối thiểu hóa theo chính tắc 1)*

Phương trình logic của ngõ ra **a**:



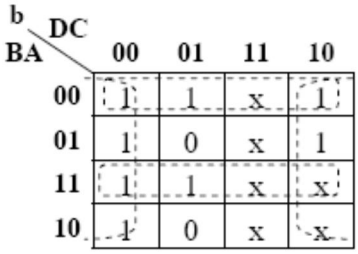
Dạng chính tắc 1:

$$\mathbf{a} = D + B + \overline{A}\,\overline{C} + AC$$

Dạng chính tắc 2:

$$\mathbf{a} = (\overline{A} + B + C + D)(A + B + \overline{C})$$
$$= AD + B + AC + \overline{A}\,\overline{C}$$

Phương trình logic của ngõ ra **b**:



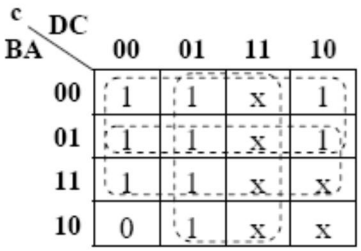
Dạng chính tắc 1:

$$\mathbf{b} = \overline{C} + BA + \overline{B}\,\overline{A} = \overline{C} + \overline{A \oplus B}$$

Dạng chính tắc 2:

$$\mathbf{b} = (\overline{C} + B + \overline{A})(\overline{C} + \overline{B} + A)$$
$$= \overline{C} + AB + \overline{A}\,\overline{B} = \overline{C} + \overline{A \oplus B}$$

Phương trình logic của ngõ ra **c**:



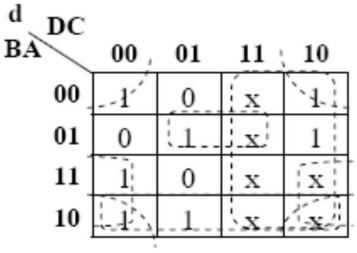
Dạng chính tắc 1:

$$\mathbf{c} = \overline{B} + A + C$$

Dạng chính tắc 2:

$$\mathbf{c} = C + \overline{B} + A$$

Phương trình logic của ngõ ra **d**:



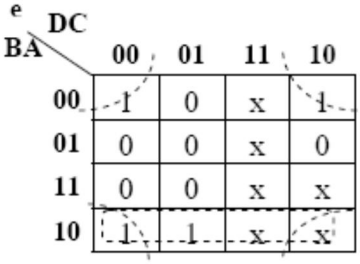
Dạng chính tắc 1:

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

Dạng chính tắc 2:

$$\mathbf{d} = (A + B + \overline{C})(\overline{A} + \overline{B} + \overline{C})(\overline{A} + B + C + D)$$
$$= (\overline{C} + A\overline{B} + \overline{A}B)(\overline{A} + B + C + D)$$
$$= (C + A \oplus B)(\overline{A} + B + C + D)$$

Phương trình logic của ngõ ra **e**:



Dạng chính tắc 1:

$$\mathbf{e} = \overline{A}.B + \overline{C}\,\overline{A}$$

Dạng chính tắc 2:

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

Phương trình logic của ngõ ra **f**:

Dạng chính tắc 1:

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

Dạng chính tắc 2:

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

f: BA \ DC	00	01	11	10
00	1	1	x	1
01	0	1	x	1
11	0	0	x	x
10	0	1	x	x

Phương trình logic của ngõ ra **g**:

Dạng chính tắc 1:

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

Dạng chính tắc 2:

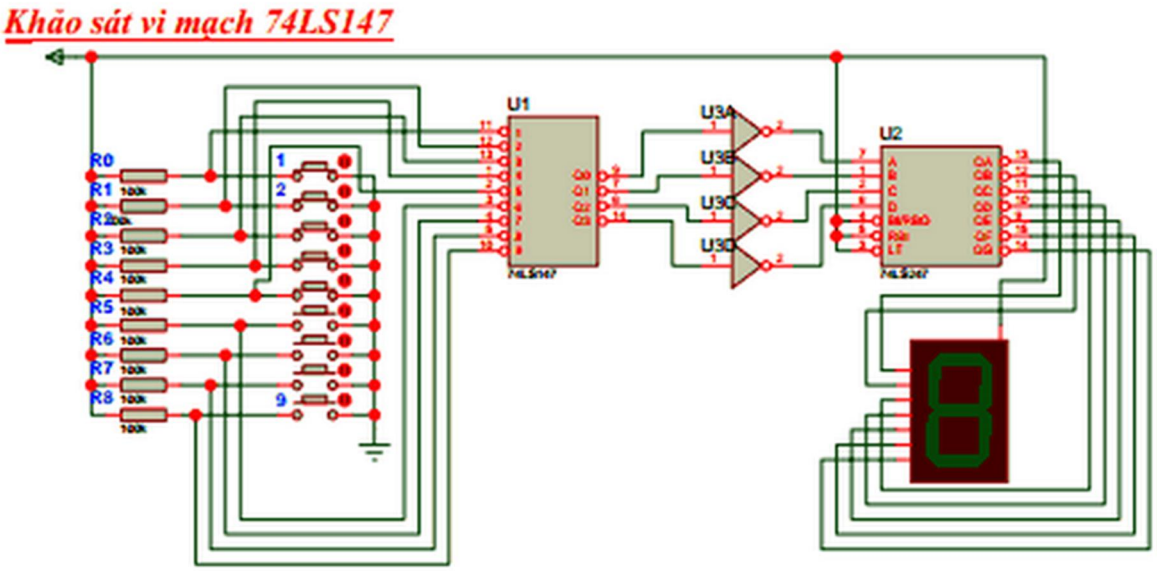
$$\mathbf{g} = (\overline{C}+\overline{B}+\overline{A})(B+C+D)$$

g: BA \ DC	00	01	11	10
00	0	1	x	1
01	0	1	x	1
11	1	0	x	x
10	1	1	x	x

4.3. Lắp ráp và cân chỉnh mạch

4.3.1. Lắp mạch giải mã LED 7 đoạn anốt chung trên IC giải mã 7447:

Lắp mạch giải mã kết hợp với IC giải mã nhị phân 74147 như trên sơ đồ nguyên lý sau:



4.3.1.1. Khảo sát sơ đồ chân của IC 7447

4.3.1.2. Khảo sát sơ đồ chân của LED 7 đoạn anốt chung.

4.3.1.3. Lắp mạch trên bo đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiệc trước khi dùng làm phần tử kết nối trong mạch.
- Bố trí các linh kiện hợp lý
- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.
- Mối hàn phải ngấu, bóng.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

4.3.1.4 Lắp mạch dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp đã được thiết kế trước (không có đoạn mạch in nào được thiết kế là liên mà lại bị đứt trên bo mạch in chuẩn bị cho lắp ráp).

- Có sơ đồ lắp ráp kèm theo.

- Khi lắp phải cắm đúng vị trí và chiều các linh kiện (đặc biệt là các linh kiên bán dẫn) theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, bóng, gọn, không gây chạm chập trên mạch in.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

BÀI TẬP

1. Thiết kế mạch mã hóa 32 đường sang 5 đường dùng IC 74148 và cổng logic.
2. Thiết kế mạch giải mã 4 đường sang 16 đường từ mạch giải mã 2 đường sang 4 đường có đầu vào cho phép.
3. Thiết kế mạch so sánh 4 bit từ mạch so sánh 1 bit
4. Thiết kế mạch chuyển từ mã Gray sang mã nhị phân
5. Thiết kế mạch chuyển từ mã BCD sang mã Excess-3 của các số từ 0 đến 9.
(Mã Excess-3 của 1 số có được từ trị nhị phân tương ứng cộng thêm 3, ví dụ mã số 0 là 0011, mã số 9 là 1100)
6. Dùng một mạch giải mã 3 sang 8 đường, 2 cổng NAND 3 đầu vào và 1 cổng AND 2 đầu vào thực hiện các hàm sau:
$F_1 = \Sigma(1,2,3)$; $F_2 = \Sigma(4,5,7)$; $F_3 = \Sigma(1,2,3,4,5,7)$

BÀI 6
MẠCH DỒN KÊNH
Mã bài: MĐ26.06

Giới thiệu:

Mạch chọn kênh còn gọi là mạch hợp kênh (ghép kênh, đa hợp) là mạch có chức năng chọn lần lượt 1 trong N kênh vào để đưa đến đầu ra duy nhất (đầu ra duy nhất đó gọi là đường truyền chung). Do đó, mạch chọn kênh còn gọi là mạch chuyển dữ liệu song song ở đầu vào thành dữ liệu nối tiếp ở đầu ra, được gọi là Multiplex (viết tắt là MUX).

Bài học góp một phần vào việc thực hiện mục tiêu đào tạo của mô đun.

Mục tiêu:

- Trình bày được cấu trúc và nguyên lý hoạt động của mạch dồn kênh.
- Lắp ráp, sửa chữa được mạch dồn kênh.
- Chủ động, sáng tạo và đảm bảo an toàn trong quá trình làm việc.

Nội dung chính:

1. Khái niệm mạch dồn kênh

Mục tiêu:

Trình bày được nguyên tắc và các loại mạch dồn kênh.

1.1 . Khái niệm

Mạch chọn kênh (MUX) là mạch để chọn một trong số N đầu vào để đưa tới 1 đầu ra duy nhất dựa trên tổ hợp 2^n của n đầu vào điều khiển.

1.2. Nguyên tắc và các loại mạch dồn kênh

Để đảm bảo thực hiện điều khiển theo yêu cầu thì số đầu vào điều khiển n phải đảm bảo điều kiện: $2^n \geq N$.

Theo tín hiệu đầu vào điều khiển ta có thể phân ra:

- Loại mạch MUX có $n \leq N$.
- Loại mạch MUX có $n = N$.

2. Cấu trúc mạch dồn kênh 4 đường vào 1 đường ra

Mục tiêu:

Trình bày được cấu trúc mạch dồn kênh 4 đường vào và 1 đường ra.

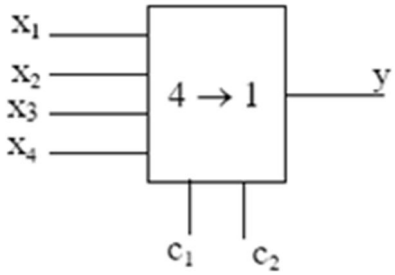
2.1. Phương pháp thực hiện

Từ 4 giá trị dữ liệu đầu vào dùng mạch MUX chuyển thành 1 đầu ra duy nhất theo các bộ mã điều khiển, theo thứ tự:

- Vẽ sơ đồ khối.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic (tối thiểu hóa nếu cần).
- Vẽ sơ đồ logic thực hiện.

2.2. Mạch điện thực hiện.

Xét mạch chọn kênh đơn giản có 4 đầu vào và 1 đầu ra như hình 4.23a.

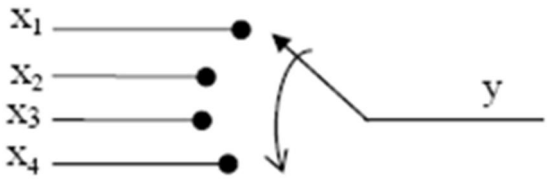


Mạch chọn kênh

Trong đó:

+ x_1, x_2, x_4: Các kênh dữ liệu vào.
+ Đầu ra y: Đường truyền chung.
+ c_1, c_2: Các đầu vào điều khiển

Vậy mạch này giống như 1 chuyển mạch:



Mạch chọn kênh

Để thay đổi lần lượt từ $x_1 \rightarrow x_4$ phải có điều khiển do đó đối với mạch chọn kênh để chọn lần lượt từ 1 trong 4 kênh vào cần có các đầu vào điều khiển c_1, c_2. Nếu có **N** kênh vào thì cần có **n** đầu vào điều khiển thỏa mãn quan hệ: $\mathbf{N=2^n}$. Nói cách khác: ***Số tổ hợp đầu vào điều khiển bằng số lượng các kênh vào***.

Việc chọn dữ liệu từ 1 trong 4 đầu vào để đưa đến đường truyền chung là tùy thuộc vào tổ hợp tín hiệu điều khiển tác động đến hai đầu vào điều khiển c_1, c_2.

+ $c_1 = c_2 = 0 \Rightarrow y = x_1$ (x_1 được nối tới đầu ra y).
+ $c_1 = 0, c_2 = 1 \Rightarrow y = x_2$ (x_2 được nối tới đầu ra y).
+ $c_1 = 1, c_2 = 0 \Rightarrow y = x_3$ (x_3 được nối tới đầu ra y).
+ $c_1 = 1, c_2 = 1 \Rightarrow y = x_4$ (x_4 được nối tới đầu ra y).

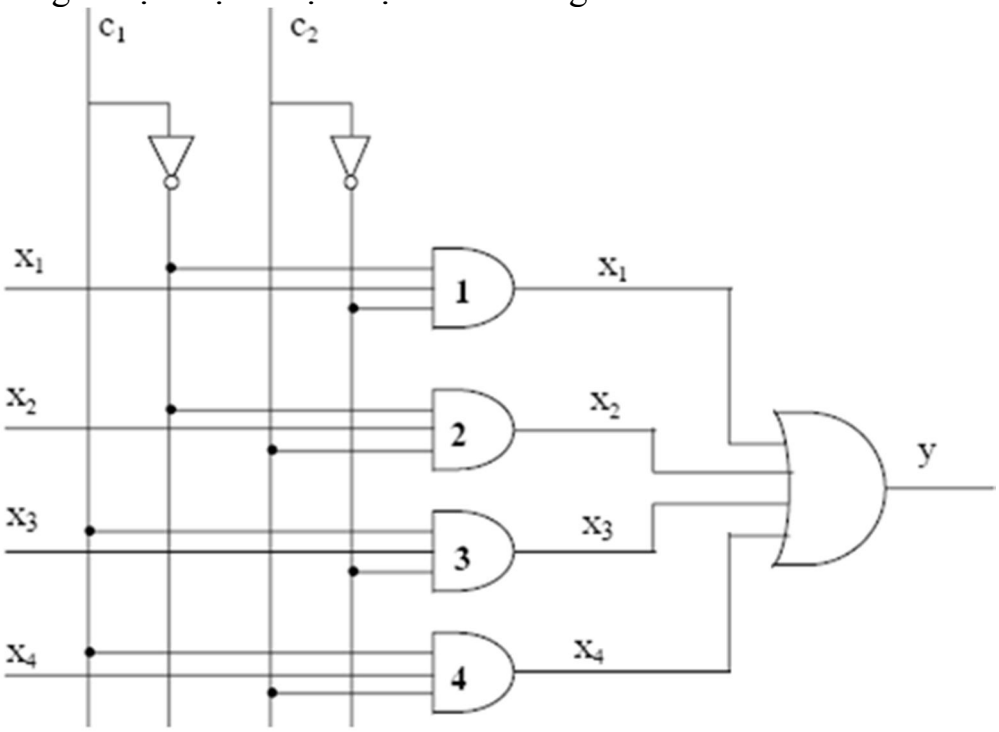
Vậy tín hiệu điều khiển phải liên tục để dữ liệu từ các kênh được liên tục đưa đến đầu ra. Từ đó ta lập được bảng trạng thái mô tả hoạt động của mạch chọn kênh như dưới đây:

c_1	c_2	y
0	0	x_1
0	1	X2
1	0	X3
1	1	X4

Phương trình (hàm) logic mô tả hoạt động của mạch :

$$y = \overline{c_1}\,\overline{c_2}.x_1 + \overline{c_1}\,c_2.x_2 + c_1\overline{c_2}.x_3 + c_1.c_2.x_4$$

Sơ đồ logic thực hiện nhiệm vụ của hàm logic trên:



Giải thích hoạt động của mạch:

$$+\ c_1 = c_2 = 0 \Rightarrow \overline{c_1} = \overline{c_2} = 1$$

$\Rightarrow$ cổng AND 1 có hai đầu vào điều khiển ở mức logic 1, cũng tương ứng với 1 đầu vào điều khiển ở mức logic 1 nên cổng AND 1 mở cho dữ liệu x_1 đưa vào.

$$+\ c_1 = 0,\ c_2 = 1 \Rightarrow \overline{c_1} = 1,\ c_2 = 0 \Rightarrow$$

cổng AND 2 có hai đầu vào điều khiển ở mức logic 1, cũng tương ứng với 1 đầu vào điều khiển ở mức logic 1 nên cổng AND 2 mở cho dữ liệu x_2 đưa vào.

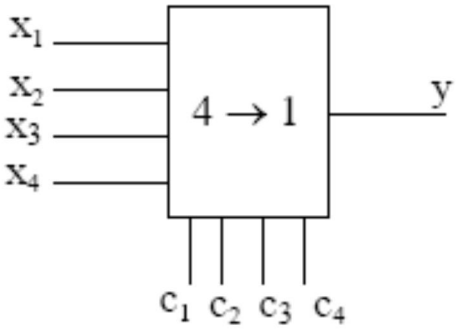
$$+\ c_1 = 1,\ c_2 = 0 \Rightarrow c_1 = 1,\ \overline{c_2} = 1 \Rightarrow$$

cổng AND 3 có hai đầu vào điều khiển ở mức logic 1, cũng tương ứng với 1 đầu vào điều khiển ở mức logic 1 nên cổng AND 3 mở cho dữ liệu x_3 đưa vào.

$$+\ c_1 = 1,\ c_2 = 1 \Rightarrow c_1 = c_2 = 1 \Rightarrow$$

cổng AND 4 có hai đầu vào điều khiển ở mức logic 1, cũng tương ứng với 1 đầu vào điều khiển ở mức logic 1 nên cổng AND 4 mở cho dữ liệu x_4 đưa vào.

Bây giờ, xét mạch chọn kênh có 4 đầu vào và 1 đầu ra, nhưng lại có 4 đầu điều khiển. Lúc này, ta không dựa vào tổ hợp tín hiệu tác động lên đầu vào điều khiển, mà chỉ xét đến mức tích cực ở đầu vào điều khiển. Ta sẽ chọn một trong hai mức logic 1 hoặc mức logic 0 làm mức tích cực, nếu 1 đầu vào trong số 4 đầu vào điều khiển tồn tại mức logic tích cực (mức 1 hoặc mức 0) thì kênh dữ liệu vào có cùng chỉ số với đầu vào điều khiển đó sẽ được kết nối với đầu ra. Trên hình 4.25 biểu diễn mạch chọn kênh với số lượng đầu vào điều khiển bằng số lượng kênh vào.



Nếu chọn mức tích cực của các đầu vào điều khiển là mức logic 1, ta có bảng trạng thái mô tả hoạt động của mạch như sau:

c_1	c_2	c_3	c_4	y
1	0	0	0	x_1
0	**1**	0	0	x_2
0	0	**1**	0	x_3
0	0	0	**1**	x_4

Phương trình logic:

$$y = c_1 . x_1 + c_2 . x_2 + c_3 . x_3 + c_4 . x_4$$

Ý nghĩa trong thực tế của mạch:

+ c_1, c_2, c_3, c_4 : Có thể hiểu là các địa chỉ (nguồn và đích).
+ x_1, x_2, x_3, x_4 : Thông tin cần truyền đi.

3. Cấu trúc mạch dồn kênh 8 đường vào 1 đường ra

Mục tiêu:

Trình bày được cấu trúc mạch dồn kênh 8 đường vào và 1 đường ra.

3.1. Phương pháp thực hiện

Từ 8 giá trị dữ liệu đầu vào dùng mạch MUX chuyển thành 1 đầu ra duy nhất theo các bộ mã điều khiển, theo thứ tự:

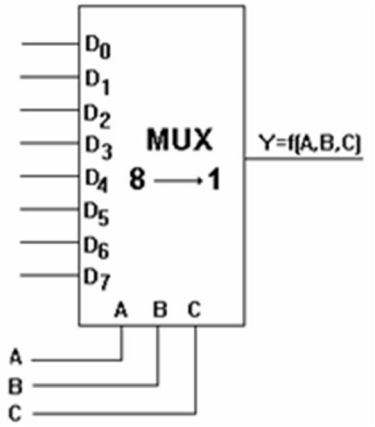
- Vẽ sơ đồ khối.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic (tối thiểu hóa nếu cần).

- Vẽ sơ đồ logic thực hiện.

3.2. Mạch điện thực hiện.

Theo đúng thứ tự thực hiện ta tạo một mạch điện thự hiện nhiệm vụ một mạch dồn kênh 8 → 1.

Hình dưới đây là ký hiệu một mạch dồn kênh 8 → 1 với 3 đầu vào điều khiển.



Giả sử bảng sự thật (hay bảng mô tả trạng thái làm việc) được quy ước như ở dưới đây:

Bảng sự thật:

A	B	C	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Khi đó, đầu ra Y quan hệ với các đầu vào dữ liệu và điều khiển theo hàm logic sau:

$$Y = \overline{A}.\overline{B}.\overline{C}D_0 + \overline{A}.\overline{B}.CD_1 + \overline{A}B\overline{C}D_2 + \overline{A}BCD_3 + A\overline{B}.\overline{C}D_4 + A\overline{B}CD_5 + AB\overline{C}D_6 + ABCD_7$$

3.3. Lắp ráp và cân chỉnh mạch dồn kênh 8 đường vào 1 đường ra.

4. Một số ứng dụng của mạch dồn kênh

Mục tiêu:

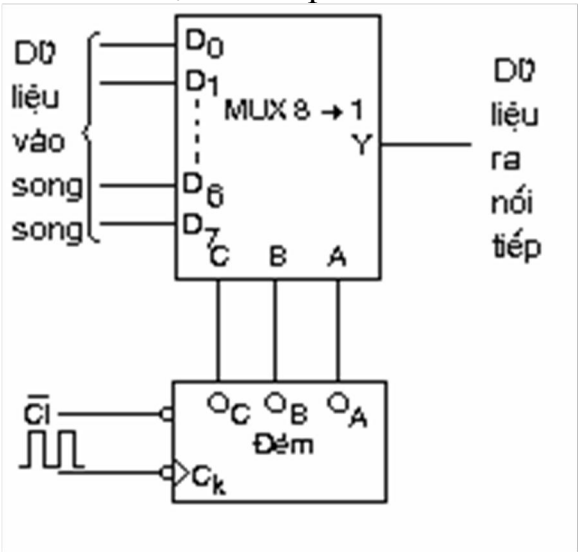
Trình bày được một số ứng dụng của mạch dồn kênh

4.1. Bộ chọn dữ liệu (data selector)

Dùng làm các mạch chọn dữ liệu như phần 2 vừa trình bày. Ngoài ra nó còn được dùng làm mạch biến chuỗi dữ liệu song song ở đầu vào thành nối tiếp ở đầu ra; mạch tạo chuỗi xung tuần hoàn; mạch tạo hàm logic …

4.2. Biến đổi dạng thông tin vào song song thành ra nối tiếp

Một mạch dồn kênh kết hợp với một mạch đếm sẽ biến chuỗi dữ liệu song song ở đầu vào thành chuỗi dữ liệu nối tiếp ở đầu ra.



4.3. Tạo chuỗi xung tuần hoàn :

Nếu cho dữ liệu vào tuần hoàn, dữ liệu ra nối tiếp cũng tuần hoàn, như vậy chỉ cần đặt trước các đầu vào thay đổi theo một chu kỳ nào đó ta sẽ được chuỗi xung tuần hoàn ở đầu ra.

Ngoài ra mạch dồn kênh có thể sử dụng độc lập hoặc kết hợp với cổng NOT để hình thành nên một mạch tạo hàm logic (Sinh viên có thể tham khảo thêm trong các tài liệu về kỹ thuật số).

Bài tập:

1. Cài đặt các hàm sau dùng bộ dồn kênh (multiplexer) 4 → 1 (Dùng thêm cổng logic nếu cần)

$$F_1 = A\overline{B} + \overline{A}B\overline{C} + \overline{B}C + AC$$

$$F_2 = A \oplus (\overline{B}C)$$

$$F_3 = \prod(1,3,6)$$

2. Thiết kế mạch MUX 4 → 1 từ các MUX 2 → 1

3. Dùng 2 MUX 2 → 1 để thực hiện 1 MUX 3 → 1 như sau:

AB = 00 chọn C

AB = 01 chọn D

AB =1X chọn E (Trường hợp này B không xác định).

4. Thực hiện hàm Z= AB +BC + CA

- Giải mã 3 sang 8 đường (dùng thêm cổng logic nếu cần).
- Đa hợp 4 → 1 (dùng thêm cổng logic nếu cần).

BÀI 7
MẠCH PHÂN KÊNH
Mã bài: MĐ26.07

Giới thiệu:

Mạch chọn kênh thực hiện chức năng ở đầu phát còn mạch phân kênh thực hiện chức năng ở đầu thu. Mạch phân kênh còn gọi là mạch tách kênh (phân đường, giải đa hợp), mạch này có nhiệm vụ tách nguồn dữ liệu khác nhau ở cùng một đầu vào để rẽ ra N đầu ra khác nhau. Do đó, mạch phân kênh còn gọi là mạch chuyển dữ liệu nối tiếp ở đầu vào thành dữ liệu song song ở đầu ra, được gọi là Demultiplex (viết tắt là DEMUX).

Mục tiêu:

- Trình bày được nguyên lư hoạt động và cấu trúc của mạch phân kênh.
- Lắp ráp, sửa chữa được mạch phân kênh.
- Chủ động, sáng tạo và đảm bảo an toàn trong quá trình làm việc.

Nội dung chính:

1. Khái niệm mạch phân kênh

Mục tiêu:

Trình bày được khái niệm về mạch phân kênh, nguyên tắc và phân loại mạch phân kênh.

1.1. Khái niệm

Mạch phân kênh còn gọi là mạch tách kênh (phân đường, giải đa hợp), mạch này có nhiệm vụ tách nguồn dữ liệu khác nhau ở cùng một đầu vào để rẽ ra N đầu ra khác nhau dựa vào bộ mã điều khiển của n đầu vào điều khiển.

Mạch giải đa hợp thực chất là mạch giải mã trong đó đầu vào cho phép trở thành đầu vào dữ liệu và đầu vào của tổ hợp số nhị phân trở thành đầu vào địa chỉ.

1.2. Nguyên tắc và các loại mạch phân kênh

Để đảm bảo thực hiện điều khiển theo yêu cầu thì số đầu vào điều khiển n phải đảm bảo điều kiện: $2^n \geq N$.

Theo tín hiệu đầu vào điều khiển ta có thể phân ra:

- Loại mạch DEMUX có $n \leq N$.
- Loại mạch DEMUX có $n = N$.

2. Cấu trúc mạch phân kênh 2 đầu ra dùng cổng logic cơ bản

Mục tiêu:

Trình bày được ccáu trúc mạch phân kênh 2 đầu ra dùng các cổng logic cơ bản và lắp ráp, kiểm tra, cân chỉnh được một mạch phân kênh 2 đầu ra.

2.1. Phương pháp thực hiện mạch phân kênh 2 đầu ra.

2.2. Sơ đồ thực hiện dùng cổng logic cơ bản

2.3. Lắp sơ đồ, kiểm tra và cân chỉnh.

3. Cấu trúc mạch phân kênh 8 đầu ra dùng IC CMOS

Mục tiêu:

Trình bày được ccáu trúc mạch phân kênh 8 đầu ra dùng các cổng logic cơ bản và lắp ráp, kiểm tra, cân chỉnh được một mạch phân kênh 8 đầu ra.

3.1. Phương pháp xây dựng.

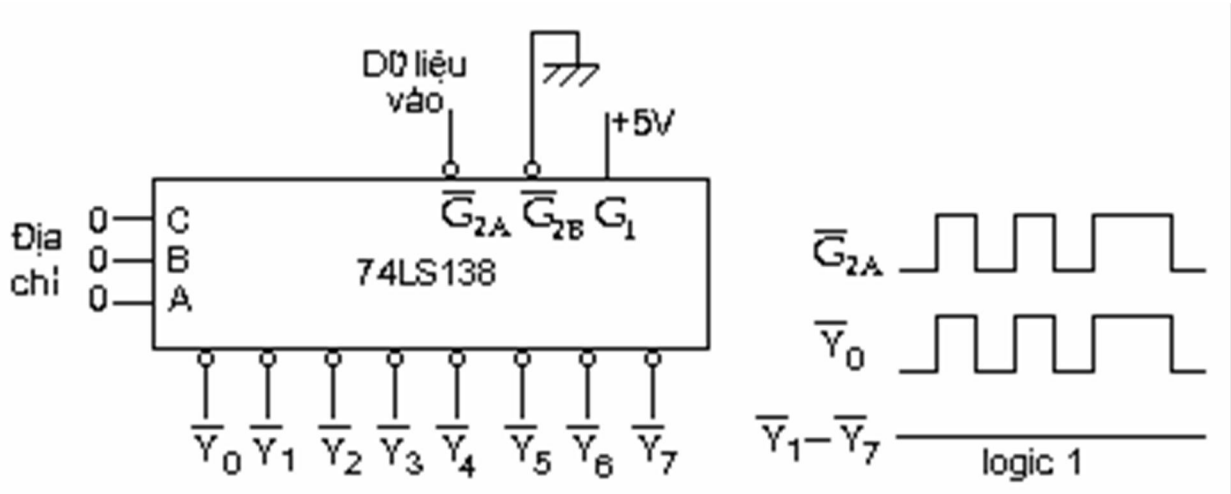
Từ yêu cầu với mạch DEMUX xây dựng mạch logic thực hiện, theo thứ tự:

- Vẽ sơ đồ khối.
- Xây dựng bảng trạng thái làm việc.
- Xây dựng hàm logic (tối thiểu hóa nếu cần).
- Vẽ sơ đồ logic thực hiện.

3.2. Các sơ đồ mạch dùng IC CMOS

Trên thị trường, người ta chế tạo mạch giải mã và giải đa hợp chung trong một IC, tùy theo điều kiện mà sử dụng. Ví dụ IC 74138 là IC Giải mã 3 sang 8 đường đồng thời là mạch giải đa hợp 1 → 8.

Khi sử dụng IC 74138 làm mạch giải đa hợp, người ta dùng một đầu vào cho phép làm đầu vào dữ liệu và các đầu vào số nhị phân làm đầu vào địa chỉ. Hình dưới đây IC 74138 dùng làm mạch phân kênh với đầu vào dữ liệu là $\overline{G}2A$; các đầu vào điều khiển là C, B, A; các đầu ra là $\overline{Y}\mathbf{0} \div \overline{Y}\mathbf{7}$.



3.3. Lắp sơ đồ và cân chỉnh

BÀI 8
MẠCH LOGIC TUẦN TỰ
Mã bài: MĐ26.08

Giới thiệu:

Trong chương trước, chúng ta đã khảo sát các loại mạch logic tổ hợp, đó là các mạch mà đầu ra của nó chỉ phụ thuộc vào các biến ở đầu vào mà không phụ thuộc vào trạng thái trước đó của mạch. Nói cách khác, đây là loại mạch không có khả năng nhớ, một chức năng quan trọng trong các hệ thống logic.

Chương này sẽ bàn về loại mạch logic thứ hai: mạch logic tuần tự.

- Mạch logic tuần tự là mạch có trạng thái đầu ra không những phụ thuộc vào tổ hợp các đầu vào mà còn phụ thuộc trạng thái đầu ra trước đó. Ta nói mạch tuần tự có tính nhớ. Mức logic tại đầu ra Q tại thời điểm (n +1) của mạch tuần tự là hàm logic của các biến đầu vào A, B, C . . . . và trạng thái của đầu ra Q tại thời điểm n trước đó.

$Q_{n+1} = f(Q_n, A, B, C \ldots)$

- Mạch tuần tự vận hành dưới tác động của xung đồng bộ (xung đồng hồ) và được chia làm 2 loại: **Đồng bộ** và **Không đồng bộ**.

Ở mạch **đồng bộ**, các phần tử của mạch chịu tác động đồng thời của xung đồng hồ (CK), còn ở mạch **không đồng bộ** thì không có điều kiện này.

Phần tử cơ bản cấu thành mạch logic tuần tự là các flipflop

Mục tiêu:

- Trình bày được đặc tính, cấu trúc và nguyên lý của các mạch logic tuần tự.
- Phân biệt được sự khác nhau giữa mạch logic tổ hợp và mạch tuần tự.
- Lắp ráp, sửa chữa được các mạch logic tuần tự.
- Chủ động, sáng tạo và đảm bảo an toàn trong quá trình làm việc.

Nội dung chính:

1. Giới thiệu mạch logic tuần tự

Mục tiêu:

Trình bày được khái niệm về các mạch FF thông dụng và mạch điện thực hiện các mạch FF đó.

1.1. Định nghĩa và phân loại

1.1.1. Định nghĩa

Mạch flipflop (FF) là mạch dao động đa hài lưỡng ổn (mạch tạo ra xung vuông và có hai trạng thái ổn định). Trạng thái của FF chỉ thay đổi khi có xung đồng hồ tác động.

Một FF thường có:

- Một hoặc hai đầu vào dữ liệu, một đầu vào xung C_K và có thể có các đầu vào với các chức năng khác.

- Hai đầu ra, thường được ký hiệu là Q (đầu ra chính) và $\overline{Q}$ (đầu ra phụ). Người ta thường dùng trạng thái của đầu ra chính để chỉ trạng thái của FF. Nếu hai đầu ra có trạng thái giống nhau ta nói FF ở **trạng thái cấm.**

Flipflop có thể được tạo nên từ mạch chốt (latch), điểm khác biệt giữa một mạch chốt và một FF là: FF chịu tác động của xung đồng hồ còn mạch chốt thì không.

Người ta gọi tên các FF bằng cách dựa vào tên các đầu vào dữ liệu của chúng.

1.1.2. Phân loại

Có hai cách phân loại:

- Phân loại theo tín hiệu điều khiển.
- Phân loại theo chức năng.

1.1.2.1. Phân loại theo tín hiệu điều khiển:

Theo tín hiệu điều khiển ta có:

- Loại không có tín hiệu điều khiển (còn gọi là không đồng bộ).
- Loại có tín hiệu điều khiển (còn gọi là đồng bộ).

Trong loại FF đồng bộ người ta còn chia ra: Loại đồng bộ mức „0“, mức „1“, sườn lên (hay sườn trước), sườn xuống (hay sườn sau) của xung đồng bộ.

- Loại làm việc theo kiểu chủ tớ (Master Slaver Flip Flop - MSFF).

1.2. Mạch R - S Flip – Flop (RSFF)

1.2.1. Khái niệm

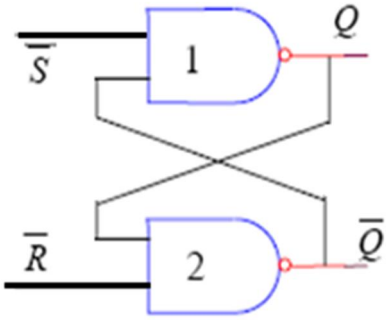
RSFF là một trong số các loại mạch logic tuần tự đơn giản, khi làm nhiệm vụ của một mạch chốt, nó có 2 đầu vào là S và R, hai đầu ra là Q và $\bar{Q}$. Khi làm nhiệm vụ một FF nó có thêm các đầu vào điều khiển (đồng bộ), xóa, Pr … để làm cơ sở xây dựng nên các mạch logic tuần tự theo yêu cầu.

RSFF có thể được xây dựng từ các cổng NAND, NOR (hoặc tương đương).

1.2.2. Mạch xây dựng từ cổng NAND

1.2.2.1. Mạch không đồng bộ:

Khi được xây dựng từ các cổng NAND, RSFF còn được gọi là mạch chốt tác động mức thấp (mức 0). Sơ đồ cấu trúc của mạch như sau:



$\bar{S}$	$\bar{R}$	Q
0	0	X
0	1	1
1	0	0
1	1	Q^0

RSFF không đồng bộ sử dụng cổng NAND và bảng trạng thái

Dựa vào bảng trạng thái của cổng NAND:

$$y = \begin{cases} 0 & \forall x_i = 1 \\ 1 & \exists x_i = 0 \end{cases}$$

Ta có:

- $\bar{S} = 0$, $\bar{R} = 1 \Rightarrow Q = 1$ hồi tiếp về cổng NAND 2 nên cổng NAND 2 có hai đầu vào bằng 1 vậy $\bar{Q} = 0$.

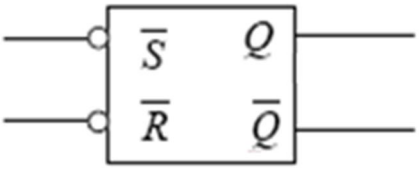
- $\overline{S} = 0$, $\overline{R} = 1 \Rightarrow \overline{Q} = 1$ hồi tiếp về cổng NAND 1 nên cổng NAND 1 có hai đầu vào bằng 1 vậy Q = 0.

- $\overline{S} = \overline{R} = 0 \Rightarrow \overline{Q} = Q = 1$ đây là trạng thái cấm.

- $\overline{S} = \overline{R} = 1$: Giả sử trạng thái trước đó có Q = 1, $\overline{Q} = 0 \Rightarrow$ hồi tiếp về cổng NAND 1 nên cổng NAND 1 có một đầu vào bằng 0 vậy Q = 1 ⇒ RSFF giữ nguyên trạng thái cũ.

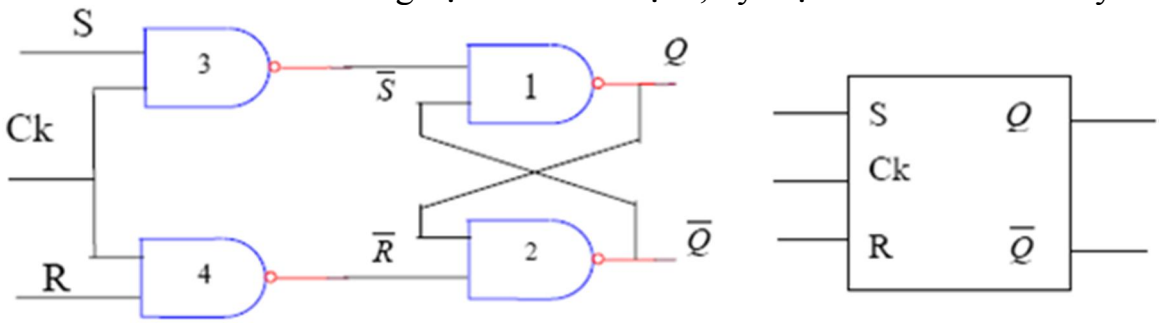
Như vậy gọi là FF không đồng bộ bởi vì chỉ cần một trong hai đầu vào S hay R thay đổi thì đầu ra cũng thay đổi theo.

Về mặt kí hiệu, RSFF không đồng bộ tác động mức thấp (0) được ký hiệu như sau:



1.2.2.2. Mạch đồng bộ:

Xét sơ đồ RSFF đồng bộ với sơ đồ mạch, ký hiệu như hình dưới đây:



RSFF đồng bộ: Sơ đồ logic và ký hiệu

Bảng trạng thái làm việc của mạch như sau:

Vào			**Ra**	
CK	**S**	**R**	$\mathbf{Q_{n+1}}$	$\overline{Q}_{\mathbf{n+1}}$
0	X	X	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$
1	0	0	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$
1	0	1	0	1
1	1	0	1	0
1	1	1	CẤM	CẤM

Trong đó: Ck là tín hiệu điều khiển đồng bộ hay tín hiệu đồng hồ (Clock). Khảo sát hoạt động của mạch:

- Ck = 0: cổng NAND 3 và 4 khóa không cho dữ liệu đưa vào. Vì cổng NAND 3 và 4 đều có ít nhất một đầu vào Ck = 0 ⇒ $\overline{S} = \overline{R} = 1 \Rightarrow \mathbf{Q_{n+1} = Q_n}$ (FF giữ nguyên trạng thái cũ).

- Ck = 1: cổng NAND 3 và 4 mở. Đầu ra Q sẽ thay đổi tùy thuộc vào trạng thái của S và R.

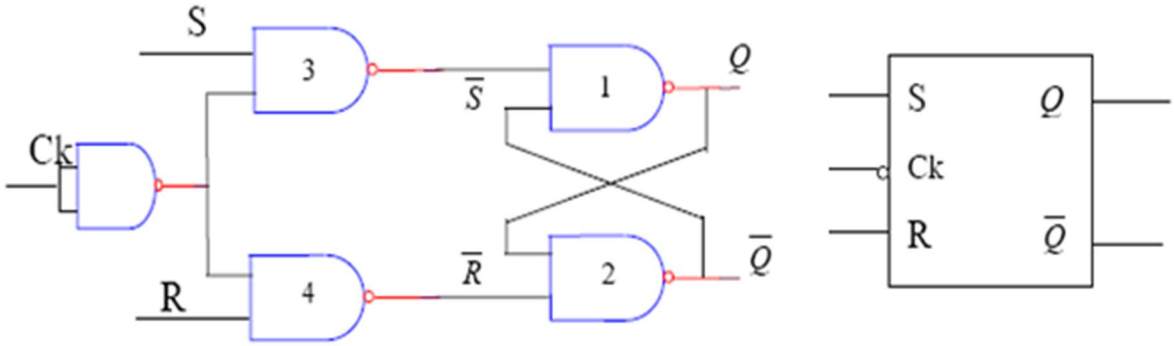
+ S = 0, R = 0 $\Rightarrow \overline{S} = \overline{R} = 1 \Rightarrow \mathbf{Q_{n+1}} = \mathbf{Q_n}$ (giữ nguyên trạng thái cũ).

+ S = 0, R = 1 $\Rightarrow \overline{S} = 1, \overline{R} = 0 \Rightarrow \mathbf{Q_{n+1}} = 0$

+ S = 1, R = 0 $\Rightarrow \overline{S} = 0, \overline{R} = 1 \Rightarrow \mathbf{Q_{n+1}} = 1$

+ S = R = 1 $\Rightarrow \overline{S} = \overline{R} = 0 \Rightarrow \mathbf{Q_{n+1}} = \mathbf{Q_n}$ = X (trạng thái cấm).

Trong trường hợp này Ck tác động mức 1. Trong trường hợp Ck tác động mức 0 thì ta mắc thêm cổng đảo như sau:



Như vậy, tùy thuộc vào mức tích cực của tín hiệu đồng bộ Ck, chúng ta có các loại tín hiệu điều khiển:

- Ck điều khiển theo mức 1 (Hình a.).
- Ck điều khiển theo mức 0 (Hình b.).
- Ck điều khiển theo sườn lên (sườn trước) (Hình c.).
- Ck điều khiển theo sườn xuống (sườn sau) (Hình d.).

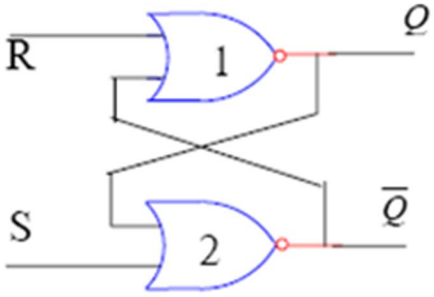
a.Mức 1 b.Mức 0 c.Sườn lên d.Sườn xuống

Các tín hiệu điều khiển Ck khác nhau

1.2.3. Mạch xây dựng từ cổng NOR

1.2.3.1. Mạch không đồng bộ:

Khi được xây dựng từ các cổng NOR, RSFF còn được gọi là mạch chốt tác động mức cao (mức 1). Sơ đồ cấu trúc của mạch như sau:



S	R	Q
0	0	Q^0
0	1	0
1	0	1
1	1	X

Dựa vào bảng trạng thái của cổng NOR, ta có:

- S = 0, R = 1 ⇒Q = 0, hồi tiếp về cổng NOR 2 nên cổng NOR 2 có hai đầu vào bằng 0 ⇒ $\overline{Q}$ = 1.

- S = 1, R = 0 ⇒ $\overline{Q}$ = 0, hồi tiếp về cổng NOR 1 nên cổng NOR 1 có hai đầu vào bằng 0 ⇒Q = 1

- Giả sử ban đầu: S = 0, R = 1 ⇒Q = 0 va $\overline{Q}$ = 1.

Nếu tín hiệu đầu vào thay đổi thành: S = 0, R = 0 ta có:

+ S = 0 và Q = 0 ⇒ $\overline{Q}$ = 1

+ R = 0 và $\overline{Q}$ = 1 ⇒Q = 0 ⇒RSFF giữ nguyên trạng thái cũ.

- Giả sử ban đầu: S = 1, R = 0 ⇒Q = 1 và $\overline{Q}$ = 0

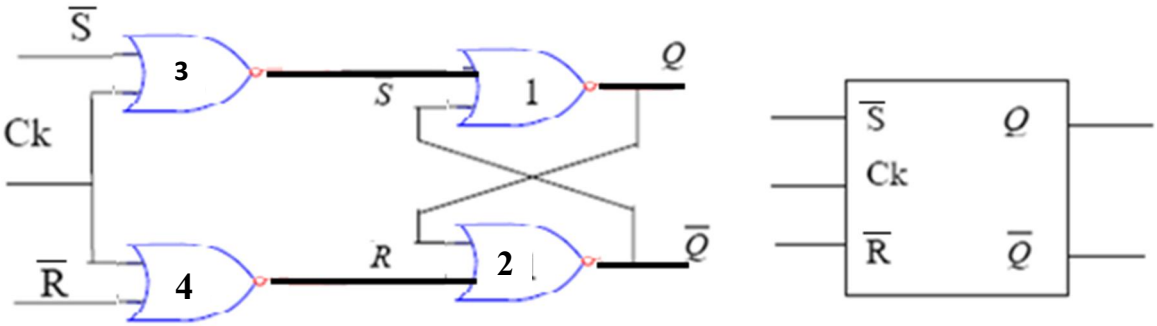
Nếu tín hiệu đầu vào thay đổi thành: S = 0, R = 0 ta có:

+ R = 0 và $\overline{Q}$ = 0 ⇒Q = 1

+ S = 0 và Q = 1 ⇒ $\overline{Q}$ = 0 ⇒RSFF giữ nguyên trạng thái cũ.

1.2.3.2. Mạch đồng bộ:

Xét sơ đồ RSFF đồng bộ với sơ đồ mạch, ký hiệu như hình dưới đây:



Bảng trạng thái làm việc của mạch như sau:

Vào			Ra	
CK	$\overline{S}$	$\overline{R}$	Q_{n+1}	$\overline{Q}_{n+1}$
0	X	X	Q_n	$\overline{Q}_n$
1	1	1	Q_n	$\overline{Q}_n$
1	0	1	1	0
1	1	0	0	1
1	0	0	CẤM	CẤM

Trong đó: Ck là tín hiệu điều khiển đồng bộ hay tín hiệu đồng hồ (Clock). Khảo sát hoạt động của mạch:

- Ck = 0: cổng NAND 3 và 4 khóa không cho dữ liệu đưa vào. Vì cổng NAND 3 và 4 đều có ít nhất một đầu vào Ck = 0 ⇒S = R = 1 ⇒$\mathbf{Q_{n+1}}$ = $\mathbf{Q_n}$ (FF giữ nguyên trạng thái cũ).

- Ck = 1: cổng NAND 3 và 4 mở. Đầu ra Q sẽ thay đổi tùy thuộc vào trạng thái của S và R.

+ $\bar{S}$ = 1, $\bar{R}$ = 1 ⇒ S = R = 0 ⇒ $\mathbf{Q_{n+1}} = \mathbf{Q_n}$ (giữ nguyên trạng thái cũ).

+ $\bar{S}$ = 0, $\bar{R}$ = 1 ⇒ S = 1, R = 0 ⇒ $\mathbf{Q_{n+1}} = 1$

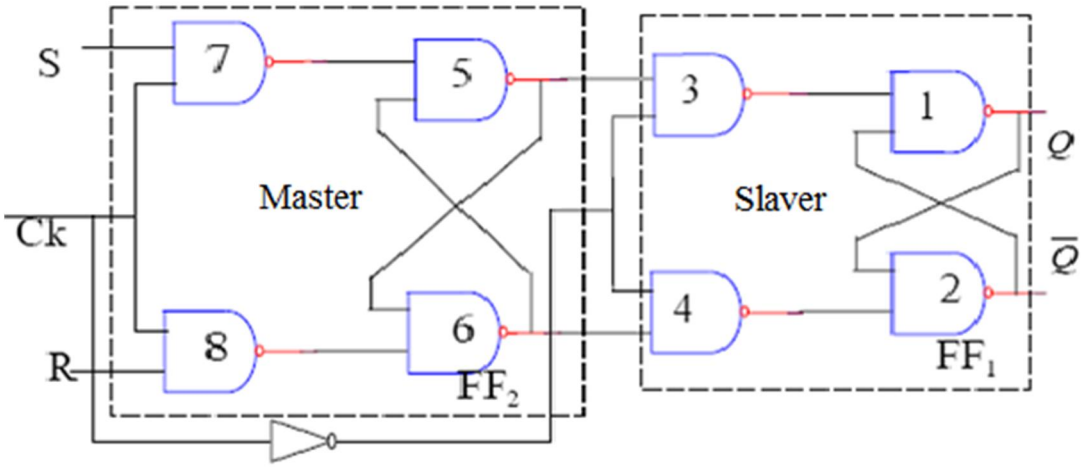
+ $\bar{S}$ = 1, $\bar{R}$ = 0 ⇒ S = 0, R = 1 ⇒ $\mathbf{Q_{n+1}} = 0$

+ $\bar{S} = \bar{R} = 0$ ⇒ S = R = 1 ⇒ $\mathbf{Q_{n+1}} = \mathbf{Q_n}$ = X (trạng thái cấm).

1.2.4. Mạch RSFF kiểu chủ tớ

Đối với RSFF này khi xung Ck tồn tại mức logic 1 dữ liệu sẽ được nhập vào FF, còn khi Ck tồn tại mức logic 0 thì dữ liệu chứa trong FF được xuất ra ngoài.

Về mặt cấu tạo bên trong RSFF kiểu chủ tớ (MSFF) gồm 2 FF: một FF thực hiện chức năng chủ (Master) và một FF thực hiện chức năng tớ (Slaver) như ở hình sau:



Hoạt động của FF điều khiển theo kiểu chủ/tớ:

+ Ck = 1: FF2 mở, dữ liệu được nhập vào FF2. Qua cổng đảo Ck = 0; FF1 khóa nên giữ nguyên trạng thái cũ trước đó.

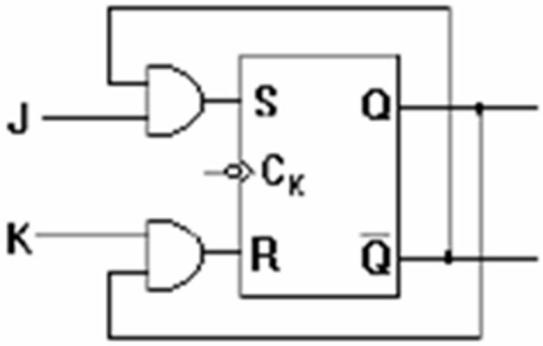
+ Ck = 0: FF2 khóa nên giữ nguyên trạng thái cũ trước đó. Qua cổng đảo Ck = 1; FF1 mở, dữ liệu được xuất ra ngoài.

Khi nghiên cứu chi tiết, ta thấy: RSFF kiểu chủ tớ đã thoát khỏi trạng thái cấm nhưng vẫn rơi vào trạng thái bất định, vì vậy người ta ít sử dụng RSFF kiểu chủ tớ trong trường hợp R = S = 1.

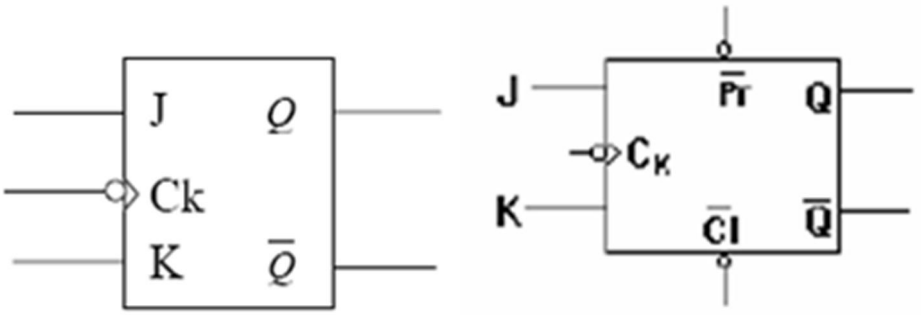
1.3. Mạch J - K Flip - Flop

1.3.1. Khái niệm

JKFF (còn được gọi là tri gơ vạn năng) là một FF được hình thành từ MSFF bằng cách: Từ MSFF người ta nơi 2 đường hồi tiếp trực tiếp từ đầu ra $\overline{Q}$ về đầu vào S và một đường từ đầu ra Q về đầu vào R. Như ở hình dưới đây:



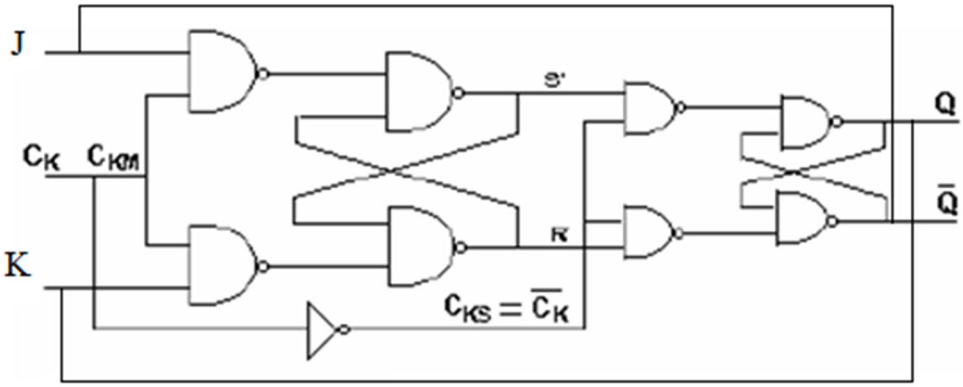
Ký hiệu của một JKF rút gọn (hình a.) và có đầy đủ các đầu vào xóa (Cl), đầu vào chương trình (Pr) tác động ở mức thấp (hình b.) như ở hình dưới đây:



a. b.

1.3.2. Mạch điện thực hiện.

Mạch điện của một JKFF xây dựng từ MSFF như ở hình dưới đây:



Bảng trạng thái làm việc của mạch như sau (để đơn giản ta bỏ qua tác động của các đầu vào Cl và Pr):

Vào			**Ra 1**		**Ra 2**	
CK	***J***	***K***	**S**	**R**	$\mathbf{Q_{n+1}}$	$\overline{Q}_{\mathbf{n+1}}$
0	X	X	**X**	**X**	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$
1	0	0	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$
1	0	1	0	1	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$
1	1	0	1	0	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$

1	1	1	$\overline{Q}_{\mathbf{n}}$	$\mathbf{Q_n}$	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$
0	0	0	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$
0	0	1	0	1	0	1
0	1	0	1	0	1	0
0	1	1	$\overline{Q}_{\mathbf{n}}$	$\mathbf{Q_n}$	$\overline{Q}_{\mathbf{n}}$	$\mathbf{Q_n}$

Từ bảng trạng thái làm việc ta thấy: Ở JKFF, dữ liệu đầu vào được đưa tới đầu ra sau một chu kỳ tác động của xung nhịp đầu vào ($1 \rightarrow 0$) và FF JK đã thoát khỏi trạng thái cấm khi J = K = 1 và thay vào đó là trạng thái đảo mức logic ở đầu ra (khi J = K = 1 thì $\mathbf{Q_{n+1}} = \overline{Q}_{\mathbf{n}}$). Người ta lợi dụng trạng thái đảo này để thiết kế mạch đếm.

Nhìn bảng trạng thái làm việc của JKFF ta thấy rõ: Muốn đưa dữ liệu đến đầu ra thì xung nhịp phải ở mức logic 0 (nghĩa là mức tích cực của xung nhịp phải là mức 0), nhưng từ nguyên lý hoạt động của MSFF thì muốn có dữ liệu đưa tới đầu ra trước đó xung nhịp phải ở mức 1 để dữ liệu được nhập vào RSFF chủ (M).

Do vậy ta có thể rút gọn bảng trạng thái làm việc của JKFF như sau:
Gọi J^n, K^n là trạng thái đầu vào vào DATA của J, K ở xung Ck thứ n.
Gọi Q^n, Q^{n+1} là trạng thái đầu ra Q ở xung Ck thứ n và thứ (n+1).
Lúc đó ta có bảng trạng thái mô tả hoạt động của JKFF:

J	K	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	$\overline{Q}^n$

Phương trình logic của JKFF:

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

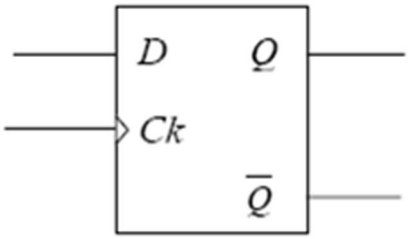
Nhận xét: JKFF là mạch điện có chức năng thiết lập trạng thái 0, trạng thái 1, chuyển đổi trạng thái và duy trì trạng thái đầu ra căn cứ vào các tín hiệu đầu vào J, K và xung nhịp đồng bộ Ck. Như vậy có thể nói JKFF là một FF rất vạn năng.

1.4. Mạch D Flip – Flop (DFF)

1.4.1. Khái niệm

DFF là một FF dùng để chốt dữ liệu, nó có một đầu vào là đầu vào dữ liệu D và các đầu ra theo quy ước của một FF (hai đầu ra liên hợp Q và $\overline{Q}$).

Ký hiệu của một DFF có xung nhịp tác động tích cực ở mức logic "1" như ở hình dưới đây:

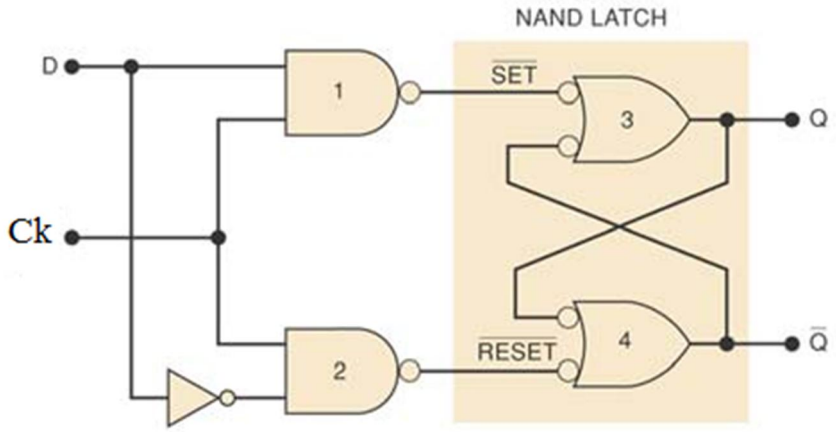


Trong đó: - D là đầu vào dữ liệu.

- Q, $\overline{Q}$ là các đầu ra.
- Ck: tín hiệu xung đồng bộ.

1.4.2. Mạch điện thực hiện.

DFF có thể thực hiện nhờ sơ đồ sau:



Hoặc có thể tạo ra DFF từ RSFF hoặc JKFF bằng cách nối một cổng đảo từ S qua R (hoặc từ J qua K). Khi này đầu vào S (hoặc J) sẽ trở thành đầu vào D.

Gọi D^n là trạng thái của đầu vào DATA D ở xung Ck thứ n.

Gọi Q^n, Q^{n+1} là trạng thái của đầu ra ở xung Ck thứ n và (n+1).

Lúc đó ta có bảng trạng thái của DFF như sau:

Bảng trạng thái:

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

Khai triển bảng này để tìm bảng đầu vào kích của DFF, ta có:

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

Bảng đầu vào kích của DFF:

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

Phương trình logic của DFF:

$$Q^{n+1} = D^n$$

Ứng dụng của DFF:

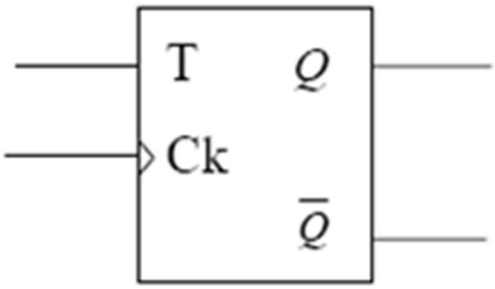
- Dùng DFF để chia tần số.
- Dùng DFF để lưu trữ dữ liệu để chế tạo các bộ nhớ và thanh ghi.
- Dùng DFF để chốt dữ liệu.

1.5. Mạch T Flip - Flop

1.5.1. Khái niệm

TFF còn được gọi là mạch lật (Togle), là một FF mà khi xung nhịp (đồng bộ) ở trạng thái tích cực thì mức logic đầu ra sẽ thay đổi mỗi khi có xung dữ liệu tích cực tác động ở đầu vào dữ liệu (T) của nó.

Ký hiệu của TFF như ở hình dưới đây:

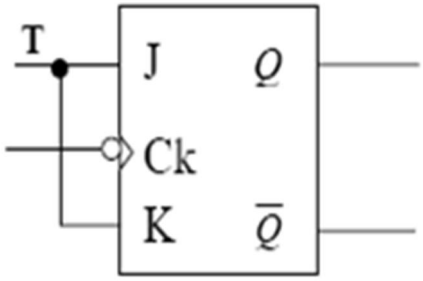


Trong đó:

- T: đầu vào dữ liệu
- Q, $\overline{Q}$: các đầu ra
- Ck: tín hiệu xung đồng bộ.

1.5.2. Mạch điện thực hiện.

Có thể tạo ra TFF từ JKFF bằng cách nối chung hai đầu vào J – K để tạo thành đầu vào T như ở hình dưới đây:



Gọi T^n là trạng thái của đầu vào DATA T ở xung Ck thứ n.

Gọi Q^n, Q^{n+1} là trạng thái của đầu ra ở xung Ck thứ n và (n+1).

Lúc đó ta có bảng trạng thái hoạt động khai triển của TFF như ở bảng dưới đây:

T^n	Q^{n+1}
0	Q^n
1	$\overline{Q}^n$

Từ bảng trạng thái này ta có nhận xét:

+ Khi T = 0: Mỗi khi có xung Ck tác động đầu ra Q duy trì trạng thái cũ trước đó.

+ Khi T = 1: Mỗi khi có xung Ck tác động đầu ra Q đảo trạng thái.

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

Từ bảng trạng thái khai triển của TFF ta tìm được bảng đầu vào kích của TFF như sau:

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

Phương trình logic của TFF:

$$Q^{n+1} = \overline{T^n}.Q^n + T^n.\overline{Q^n} \qquad \text{(dạng chính tắc 1)}$$

Hoặc: $$Q^{n+1} = (T^n + Q^n)(\overline{T^n} + \overline{Q^n}) \qquad \text{(dạng chính tắc 2).}$$

$$\Rightarrow Q^{n+1} = T^n \otimes Q^n$$

Tổng quát: Khi T = 1 TFF có thể sử dụng làm mạch đếm xung Ck hoặc mạch chia tần. Ghép nối tiếp n TFF với nhau sao cho đầu ra của TFF trước sẽ nối với đầu vào của TFF đứng sau (Cki +1 nối với Qi) và lúc bây giờ tất cả các đầu vào DATA T ở tất cả các TFF đều giữ mức logic 1, lúc đó tần số tín hiệu đầu ra sẽ là:

$$f_{Q^n} = \frac{f_{CK}}{2^n}$$

2. Mạch đếm (đếm thập phân, nhị phân)

Mục tiêu:

Trình bày cấu tạo, nguyên lý hoạt động các mạch đếm thông dụng.

2.1. Mạch đếm module

2.1.1. Khái niệm

Chức năng của mạch đếm là đếm số xung Ck đưa vào đầu vào hoặc thể hiện số trạng thái có thể có của các đầu ra.

Nếu xét khía cạnh tần số của tín hiệu thì mạch đếm có chức năng chia tần, nghĩa là tần số của tín hiệu ở đầu ra là kết quả của phép chia tần số của tín hiệu CK ở đầu vào cho số đếm của mạch.

Bộ đếm được xây dựng trên cơ sở các Flip - Flop (FF) ghép với nhau sao cho hoạt động theo một bảng trạng thái (qui luật) cho trước.

Số lượng FF sử dụng là số hàng của bộ đếm.

Bộ đếm còn được sử dụng để tạo ra một dãy địa chỉ của lệnh điều kiển, đếm số chu trình thực hiện phép tính, hoặc có thể dùng trong vấn đề thu và phát mã.

Có thể phân loại bộ đếm theo nhiều cách:

- Phân loại theo cơ sở các hệ đếm: Bộ đếm thập phân, bộ đếm nhị phân.

Trong đó bộ đếm nhị phân được chia làm hai loại:

+ Bộ đếm với dung lượng đếm là 2^n.

+ Bộ đếm với dung lượng đếm khác 2^n (đếm module M).

- Phân loại theo hướng đếm gồm: Mạch đếm lên (đếm tiến), mạch đếm xuống (đếm lùi), mạch đếm vòng.
- Phân loại mạch đếm theo tín hiệu chuyển: bộ đếm nối tiếp, bộ đếm song song, bộ đếm hỗn hợp.
- Phân loại dựa vào chức năng điều khiển:

+ Bộ đếm đồng bộ: Sự thay đổi đầu ra phụ thuộc vào tín hiệu điều kiển Ck.

+ Bộ đếm không đồng bộ.

Mặc dù có nhiều cách phân loại nhưng có ba tên gọi chính thường được sử dụng: Bộ đếm nối tiếp (không đồng bộ), Bộ đếm song song (đồng bộ), Bộ đếm hỗn hợp.

2.1.2. Mạch điện thực hiện.

Đối với các bộ đếm dù là nối tiếp hay song song; dù đếm lên hay đếm xuống, hoặc là đếm Modulo M (đếm lên/đếm xuống) đều có cách thiết kế chung và không phụ thuộc vào tín hiệu Ck tác động sườn lên, sườn xuống, mức 0 hay mức 1.

Các bước thực hiện :

- Từ yêu cầu thực tế xây dựng bảng trạng thái hoạt động của mạch.
- Dựa vào bảng đầu vào kích của FF tương ứng để xây dựng các bảng hàm giá trị của các đầu vào dữ liệu (DATA) theo đầu ra.
- Dùng các phương pháp tối thiểu để tối thiểu hóa các hàm logic trên.
- Thành lập sơ đồ logic theo hàm logic tối thiểu vừa tìm được.

2.2. Mạch đếm đồng bộ, không đồng bộ

2.2.1. Mạch đếm không đồng bộ (đếm nối tiếp)

2.2.1.1. Khái niệm

Bộ đếm nối tiếp là bộ đếm trong đó các TFF hoặc JKFF giữ chức năng của TFF được ghép nối tiếp với nhau và hoạt động theo một loại mã duy nhất là BCD 8421. Đối với loại bộ đếm này, các đầu ra thay đổi trạng thái không đồng thời với tín hiệu điều khiển Ck (tức không chịu sự điều khiển của tín hiệu điều khiển Ck) do đó mạch đếm nối tiếp còn gọi là mạch đếm không đồng bộ.

2.2.1.2. Phân loại

Theo hướng đếm và theo dung lượng đếm, mạch đếm nối tiếp được phân ra:

- Mạch đếm lên.
- Mạch đếm xuống.
- Mạch đếm lên /xuống.
- Mạch đếm module M.

2.2.1.3. Mạch đếm lên

Đây là bộ đếm có nội dung đếm tăng dần. Nguyên tắc ghép nối các TFF (hoặc JKFF thực hiện chức năng TFF) để tạo thành bộ đếm nối tiếp còn phụ thuộc vào tín hiệu điều khiển Ck. Có 2 trường hợp khác nhau:

- Tín hiệu Ck tác động sườn xuống: TFF hoặc JKFF được nghép nối với nhau theo qui luật sau:

$$Cki+1 = Qi$$

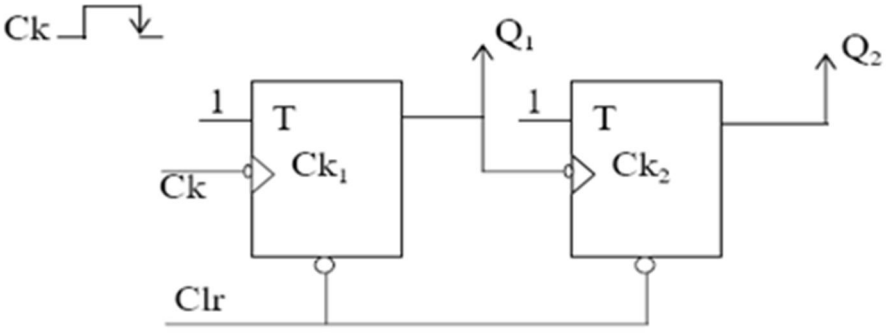
- Tín hiệu Ck tác động sường xuống: TFF hoặc JKFF được nghép nối với nhau theo qui luật sau:

$$\mathbf{Ck_{i+1}} = \overline{\mathbf{Q_i}}$$

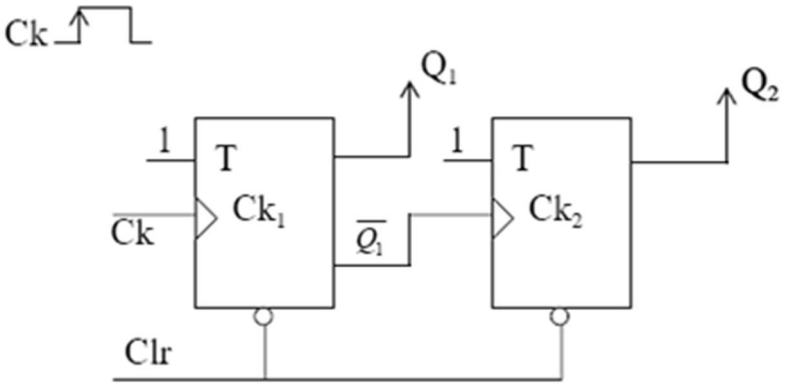
Trong đó T luôn luôn giữ ở mức logic 1 (T = 1) và đầu ra của TFF đứng trước nối với đầu vào Ck của TFF đứng sau.

Để minh họa chúng ta xét ví dụ về một mạch đếm nối tiếp, đếm 4, đếm lên, dùng TFF. Số lượng TFF cần dùng: $4 = 2^2 \Rightarrow$ dùng 2 TFF.

Trường hợp Ck tác động theo sườn xuống:

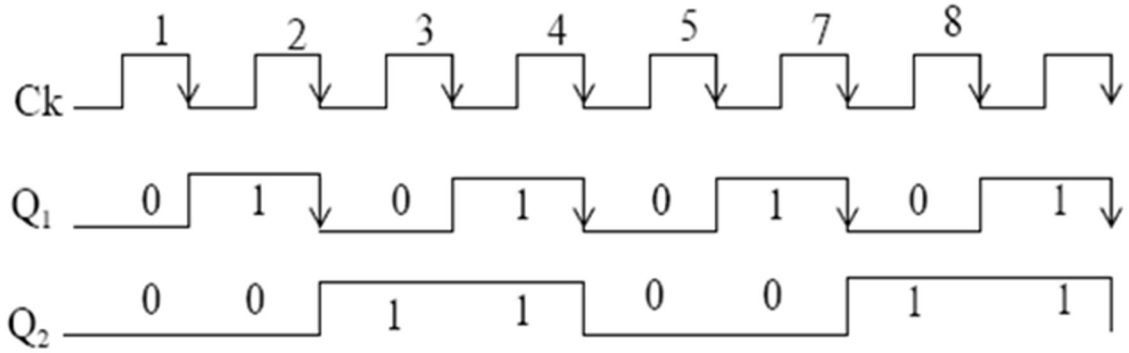


Trường hợp Ck tác động theo sườn lên:



Trong các sơ đồ mạch này Clr (Clear) là đầu vào xóa của TFF. Đầu vào Clr tác động mức thấp, khi Clr = 0 thì đầu ra Q của FF bị xóa về 0 (Q=0).

Giản đồ thời gian của mạch đếm lên có Ck tác động theo sườn xuống:

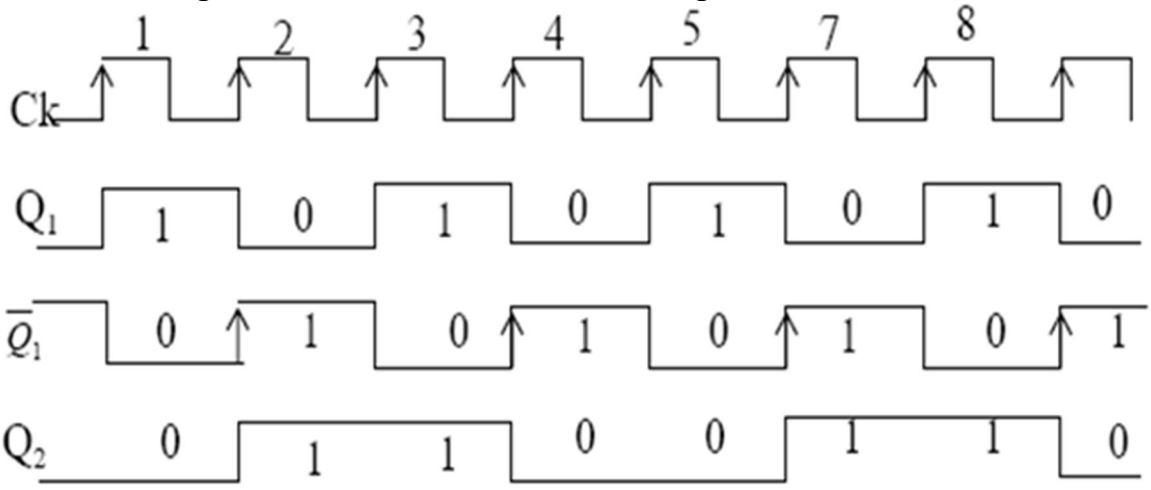


Giản đồ thời gian của mạch đếm lên có Ck tác động theo sườn xuống

Bảng trạng thái hoạt động của mạch đếm lên có Ck tác động theo sườn xuống:

Xung vào	Trạng thái hiện tại		Trạng thái kế tiếp	
Ck	Q_2	Q_1	Q_2	Q_1
1	0	0	0	1
2	0	1	1	0
3	1	0	1	1
4	1	1	0	0

Giản đồ thời gian mạch đếm lên có Ck tác động theo sườn lên:



Giản đồ thời gian mạch đếm lên có Ck tác động theo sườn lên

Bảng trạng thái hoạt động của mạch đếm lên có Ck tác động theo sườn lên:

Xung vào	Trạng thái hiện tại		Trạng thái kế tiếp	
Ck	Q_2	Q_1	Q_2	Q_1
1	0	1	1	0
2	1	0	1	1
3	1	1	0	0
4	0	0	0	1

2.2.1.4. Mạch đếm xuống

Đây là bộ đếm có nội dung đếm giảm dần. Nguyên tắc ghép các FF cũng phụ thuộc vào tín hiệu điều khiển Ck:

- Tín hiệu Ck tác động sườn xuống: TFF hoặc JKFF được nghép nối với nhau theo qui luật sau: $\mathbf{Ck_{i+1} = \overline{Q_i}}$

- Tín hiệu Ck tác động sườn xuống: TFF hoặc JKFF được nghép nối với nhau theo qui luật sau: $\mathbf{Ck_{i+1} = Q_i}$

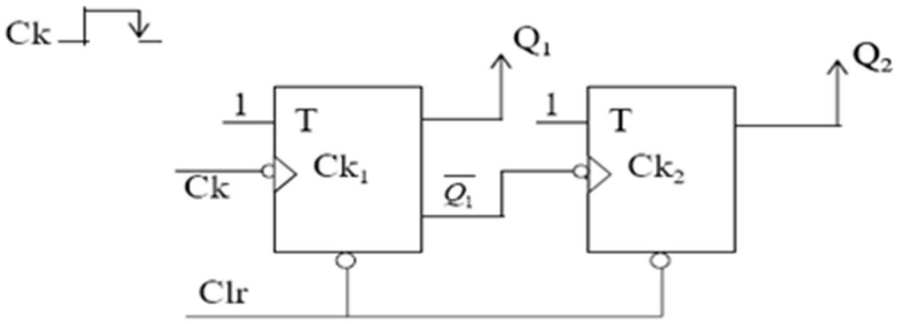
Trong đó T luôn luôn giữ ở mức logic 1 (T = 1) và đầu ra đảo của TFF đứng trước nối với đầu vào Ck của TFF đứng sau.

Ví dụ: Xét một mạch đếm 4, đếm xuống, đếm nối tiếp dùng TFF.

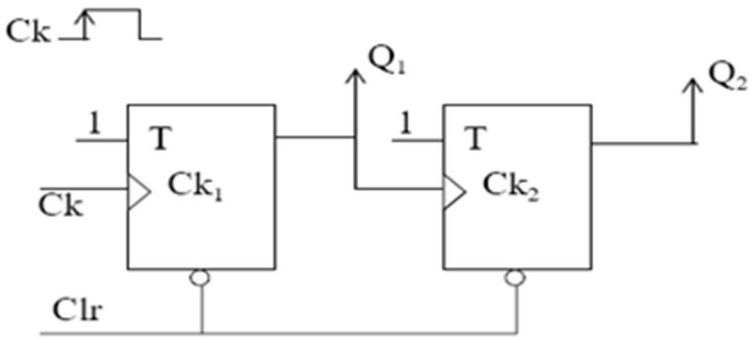
Số lượng TFF cần dùng: 4 = 22 ⇒ dùng 2 TFF.

Sơ đồ mạch thực hiện khi sử dụng Ck tác động sườn xuống và Ck tác động sườn lên.

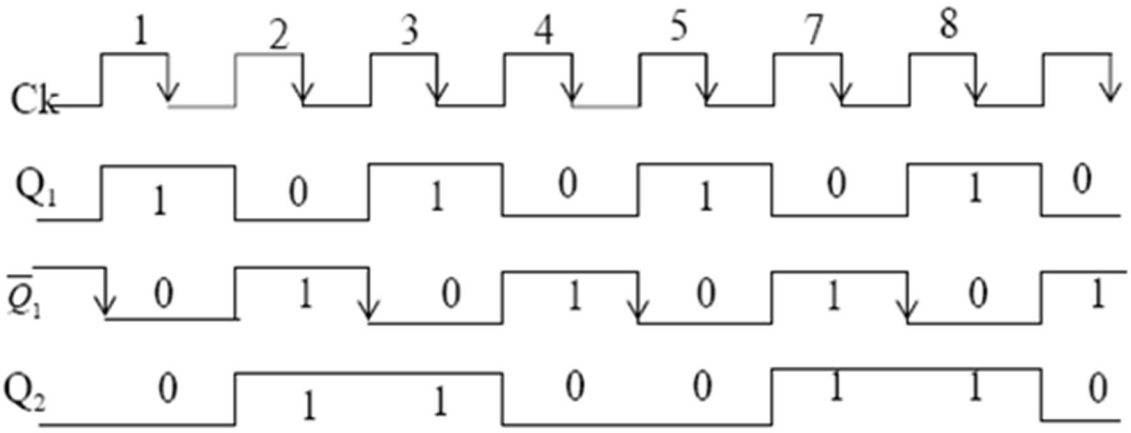
Trường hợp Ck tác động theo sườn xuống:



Trường hợp Ck tác động theo sườn lên:



Giản đồ thời gian của mạch đếm xuống có Ck tác động theo sườn xuống:

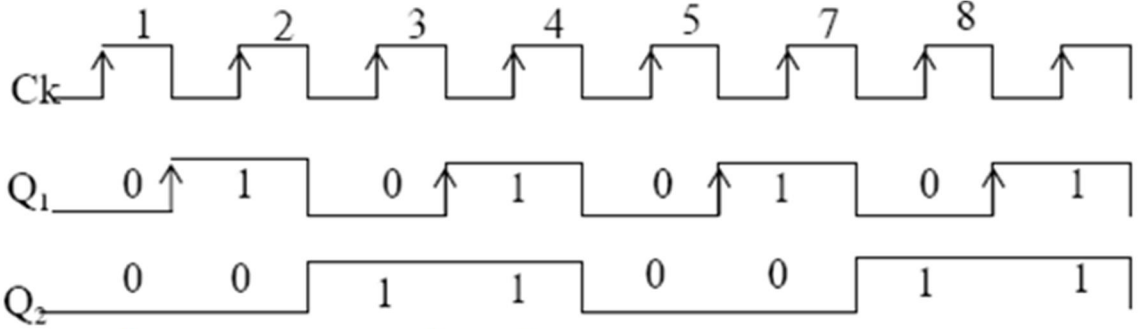


Giản đồ thời gian mạch đếm xuống có Ck tác động theo sườn xuống

Bảng trạng thái hoạt động của mạch đếm xuống có Ck tác động theo sườn xuống:

Xung vào	Trạng thái hiện tại		Trạng thái kế tiếp	
Ck	Q_2	Q_1	Q_2	Q_1
1	0	0	1	1
2	1	1	1	0
3	1	0	0	1
4	0	1	0	0

Giản đồ thời gian của mạch đếm xuống có Ck tác động theo sườn lên:



Giản đồ thời gian mạch đếm xuống có Ck tác động theo sườn lên

2.2.1.5. Mạch đếm lên, đếm xuống

Gọi X là tín hiệu điều khiển chiều đếm, ta quy ước:

+ Nếu X = 0 thì mạch đếm lên.

+ Nếu X = 1 thì đếm xuống.

Ta xét 2 trường hợp của tín hiệu Ck:

- Xét tín hiệu Ck tác động sườn xuống:

Lúc đó ta có phương trình logic: $Ck_{i+1} = \bar{X}.Q_i + X.\overline{Q_i} = X \oplus Q_i$

- Xét tín hiệu Ck tác động sườn lên:

Lúc đó ta có phương trình logic: $Ck_{i+1} = \bar{X}.\bar{Q}_i + X.Q_i = \overline{X \oplus Q_i}$

2.2.1.6. Mạch đếm module M

Đây là bộ đếm nối tiếp, theo mã BCD 8421, có dung lượng đếm khác 2^n.

Ví dụ: Xét mạch đếm 5, đếm lên, đếm nối tiếp.

Số lượng TFF cần dùng: Vì $2^2 = 4 < 5 < 8 = 2^3$ ⇒ dùng 3 TFF có đầu vào Clr. Vậy bộ đếm này sẽ có 3 đầu ra (chú ý: Số lượng FF tương ứng với số đầu ra).

Bảng trạng thái hoạt động của mạch:

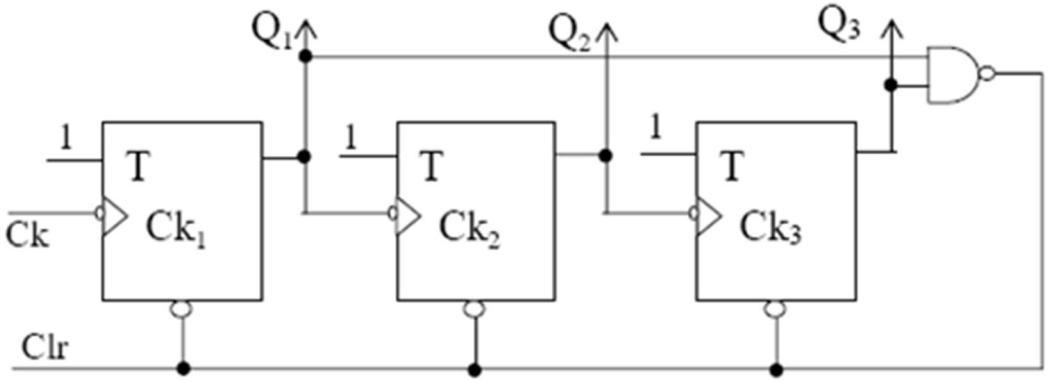
Xung vào	Trạng thái hiện tại			Trạng thái kế tiếp		
Ck	Q_3	Q_2	Q_1	Q_3	Q_2	Q_1
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	1
4	0	1	1	1	0	0
5	1	0	0	1/0	0	1/0

Nếu dùng 3 FF thì mạch có thể đếm được 8 trạng thái phân biệt (000 → 111 tương ứng 0→7). Do đó, để sử dụng mạch này thực hiện đếm 5, đếm lên, thì sau xung Ck thứ 5 ta tìm cách đưa tổ hợp 101 về 000 có nghĩa là mạch thực hiện việc đếm lại từ tổ hợp ban đầu. Như vậy, bộ đếm sẽ đếm từ 000 → 100 và quay về 000 trở lại, nói cách khác ta đã đếm được 5 trạng thái phân biệt. Để xóa bộ đếm về 000 ta phân tích: Do tổ hợp 101 có 2 đầu ra Q_1, Q_3 đồng thời bằng 1 (khác với các tổ hợp trước đó) → đây chính là dấu hiệu nhận biết để điều khiển xóa bộ đếm. Vì vậy để xóa bộ đếm về 000:

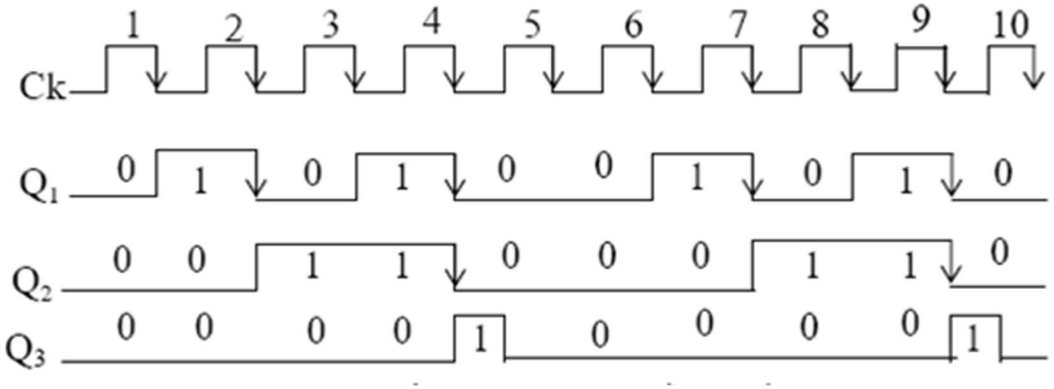
- Đối với FF có đầu vào Clr tác động mức 0 ta dùng cổng NAND 2 đầu vào.
- Đối với FF có đầu vào Clr tác động mức 1 ta dùng cổng AND 2 đầu vào.

Như vậy sơ đồ mạch đếm 5 là sơ đồ cải tiến từ mạch đếm 8 bằng cách mắc thêm phần tử cổng NAND (hoặc cổng AND) có hai đầu vào (tùy thuộc vào chân Clr tác động mức logic 0 hay mức logic 1) được nối đến đầu ra Q_1 và Q_3, và đầu ra của cổng NAND (hoặc AND) sẽ được nối đến đầu vào Clr của bộ đếm (cũng chính là đầu vào Clr của các FF).

Trong trường hợp Clr tác động mức thấp sơ đồ mạch thực hiện đếm 5 như sau:

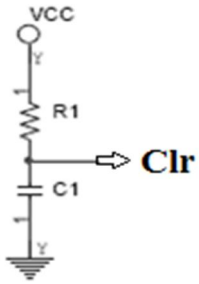


Giản đồ thời gian của mạch:



Chú ý:

Do trạng thái của đầu ra là không biết trước nên để mạch có thể đếm từ trạng thái ban đầu là 000 ta phải dùng thêm mạch xóa tự động ban đầu để xóa bộ đếm về 0 (còn gọi là mạch RESET ban đầu). Phương pháp thực hiện là dùng hai phần tử thụ động R và C như ở hình dưới đây:

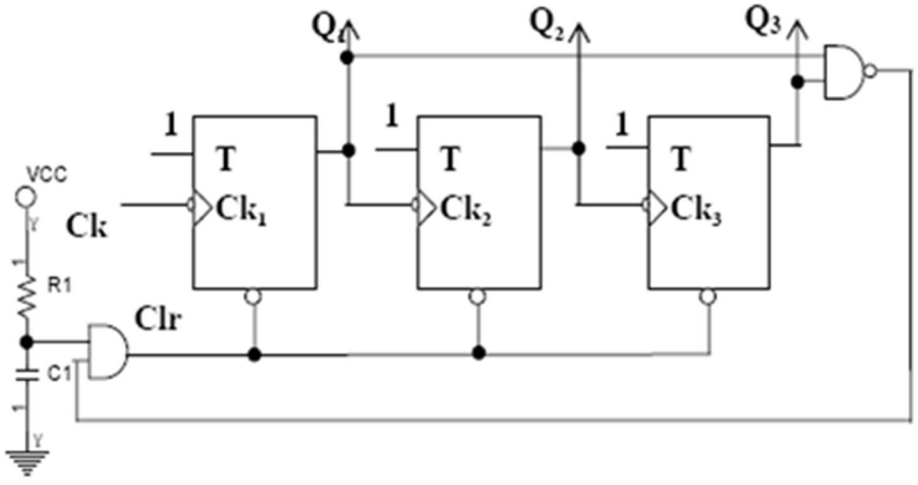


Mạch Reset mức 0

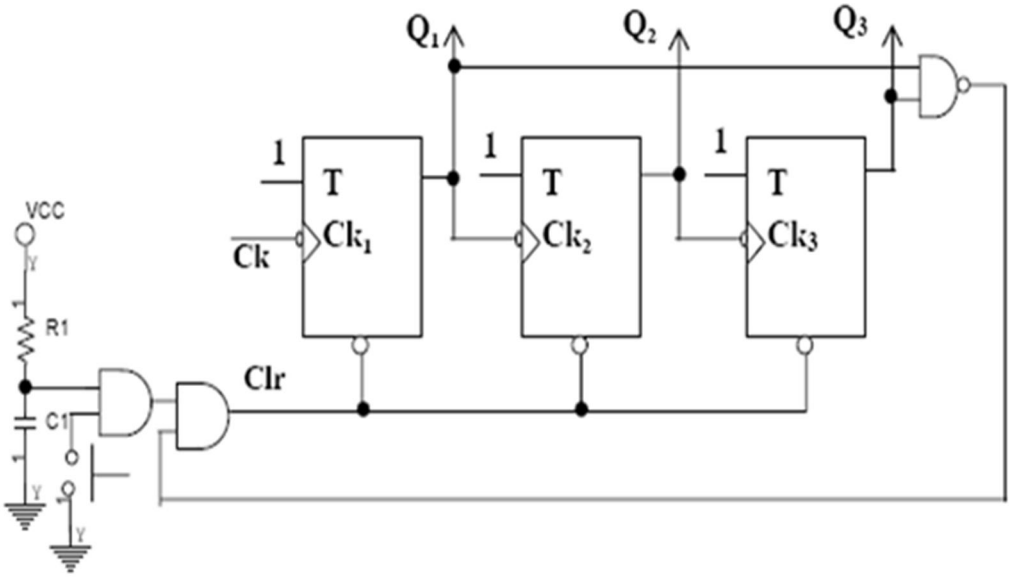
Mạch hoạt động như sau: Do tính chất điện áp trên tụ C_1 không đột biến được nên ban đầu mới cấp nguồn Vcc thì $V_C = 0 \Rightarrow$ đầu ra Clr = 0 và mạch có tác động Reset xóa bộ đếm, sau đó tụ C được nạp điện từ nguồn qua điện trở R với thời hằng nạp là $\tau = R_1C_1$ nên điện áp trên tụ tăng dần, cho đến khi tụ C_1 nạp đầy thì điện áp trên tụ xấp xỉ bằng Vcc $\Rightarrow$ đầu ra Clr = 1, mạch không còn tác dụng reset.

Chú ý khi thiết kế: Với một FF, ta biết được thời gian xóa (có trong Datasheet do nhà sản xuất cung cấp), do đó ta phải tính toán sao cho thời gian tụ C_1 nạp điện từ giá trị ban đầu đến giá trị điện áp ngưỡng phải lớn hơn thời gian xóa cho phép thì mới đảm bảo xóa được các FF.

Mạch cho phép xóa bộ đếm tự động:



Mạch cho phép xóa bộ đếm tự động và bằng tay:



Các bộ đếm nối tiếp có ưu điểm là đơn giản, dễ thiết kế. Nhưng chúng lại có nhước điểm là: Với dung lượng đếm lớn, số lượng FF sử dụng càng nhiều thì thời gian trễ tích lũy khá lớn. Nếu thời gian trễ tích lũy lớn hơn một chu kỳ tín hiệu xung kích thì lúc bấy giờ kết quả đếm sẽ sai. Do đó, để khắc phục nhược điểm này, người ta sử dụng bộ đếm song song.

2.2.2. Mạch đếm đồng bộ (đếm song song)

2.2.2.1. Khái niệm

Bộ đếm song song là bộ đếm trong đó các FF mắc song song với nhau và các đầu ra sẽ thay đổi trạng thái dưới sự điều khiển của tín hiệu Ck. Chính vì vậy mà người ta còn gọi bộ đếm song song là bộ đếm đồng bộ.

Mạch đếm song song được sử dụng với bất kỳ FF loại nào và có thể đếm theo qui luật bất kỳ cho trước. Vì vậy, để thiết kế bộ đếm đồng bộ (song song) người ta dựa vào các bảng đầu vào kích của FF.

2.2.2.2. Mạch thực hiện

Đối với bộ đếm song song dù đếm lên hay đếm xuống, hoặc là đếm Module M (đếm lên/đếm xuống) đều có cách thiết kế chung và không phụ thuộc vào tín hiệu Ck tác động sườn lên, sườn xuống, mức 0 hay mức 1.

Các bước thực hiện :

- Từ yêu cầu thực tế xây dựng bảng trạng thái hoạt động của mạch.
- Dựa vào bảng đầu vào kích của FF tương ứng để xây dựng các bảng hàm giá trị của các đầu vào dữ liệu (DATA) theo đầu ra.
- Dùng các phương pháp tối thiểu để tối thiểu hóa các hàm logic trên.
- Thành lập sơ đồ logic thực hiện hàm logic đã tối thiểu.

Ví dụ: Thiết kế mạch đếm đồng bộ, đếm 5, đếm lên theo mã BCD 8421 dùng JKFF.

Trước hết xác định số JKFF cần dùng: Vì $2^2 = 4 < 5 < 8 = 2^3$ ⇒ dùng 3 JKFF ⇒ có 3 đầu ra Q_1, Q_2, Q_3.

Ta có bảng trạng thái mô tả hoạt động của mạch như sau:

Xung vào	Trạng thái hiện tại			Trạng thái kế tiếp		
Ck	Q_3	Q_2	Q_1	Q_3	Q_2	Q_1
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	1
4	0	1	1	1	0	0
5	1	0	0	0	0	0

Bảng đầu vào kích tổng hợp cho các FF như sau:

Q^n	Q^{n+1}	S^n	R^n	J^n	K^n	T^n	D^n
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	1	0
1	1	X	0	X	0	0	1

Từ đó ta suy ra bảng hàm giá trị của các đầu vào data theo các đầu ra:

Xung vào	Trạng thái hiện tại			Trạng thái kế tiếp						
	Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
1	0	0	0	0	0	1	0	X	0	X	1	X
2	0	0	1	0	1	0	0	X	1	X	X	1
3	0	1	0	0	1	1	0	X	X	0	1	X
4	0	1	1	1	0	0	1	X	X	1	X	1
5	1	0	0	0	0	0	X	1	0	X	0	X

Lập bảng Karnaugh để tối thiểu hóa ta được:

J_1 \ Q_3Q_2	00	01	11	10
Q_1 = 0	1	1	x	0
Q_1 = 1	x	x	x	x

$J_1 = Q_1$

K_1 \ Q_3Q_2	00	01	11	10
Q_1 = 0	x	x	x	x
Q_1 = 1	1	1	x	x

$K_1 = 1 = Q_1$

J_2 \ Q_3Q_2	00	01	11	10
Q_1 = 0	0	x	x	0
Q_1 = 1	1	x	x	x

$J_2 = Q_1$

K_2 \ Q_3Q_2	00	01	11	10
Q_1 = 0	x	0	x	0
Q_1 = 1	x	1	x	x

$K_2 = Q_1$

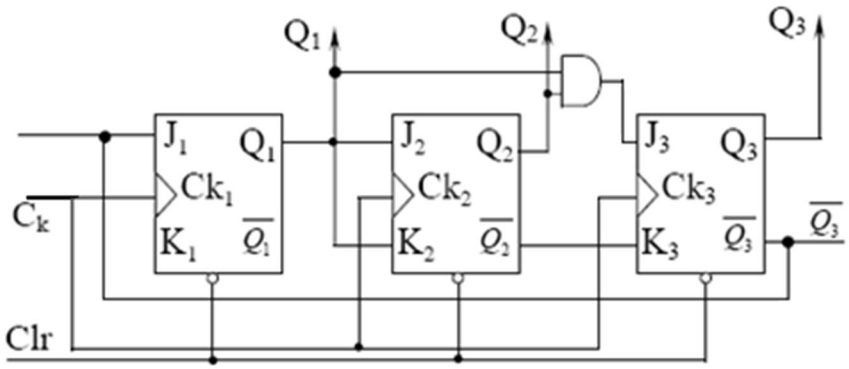
J_3 \ Q_3Q_2	00	01	11	10
Q_1 = 0	0	0	x	X
Q_1 = 1	0	1	x	x

$J_2 = Q_1Q_2$

K_3 \ Q_3Q_2	00	01	11	10
Q_1 = 0	x	0	x	0
Q_1 = 1	x	1	x	x

$K_3 = 1 = Q_3$

Lưu ý: Khi thiết kế tính toán ta dùng các phương pháp tối thiểu để đưa về phương trình logic tối giản. Nhưng trong thực tế thì đôi lúc không phải như vậy. Ví dụ: $K_3 = 1$, $K_3 = Q_3$ hay $K_3 = 2Q$ đều đúng, nhưng khi lắp ráp thực tế ta chọn $K_3 = 2Q$ để tránh dây nối dài gây nhiễu cho mạch. Sơ đồ logic:



Nguyên lý hoạt động của mạch như sau:

- Ban đầu dùng mạch xóa R-C để xóa: $Q_1 = Q_2 = Q_3 = 0$; các đầu vào $J_1 = K_1 = 1$; $J_2 = K_2 = Q_2$; $J_3 = 0$, $K_3 = 1$
- Khi Ck_1 ở mức tích cực theo quy định, trạng thái của các đầu ra đều thay đổi theo trạng thái đầu vào DATA trước đó.

$J_1 = K_1 = 1 \Rightarrow Q_1 = \overline{Q_1^0} = 1.$

$J_2 = K_2 = 1 \Rightarrow Q_2 = Q_2^0 = 0.$

$J_3 = 0, K_3 = 1 \Rightarrow Q_3 = 1$ bất chấp trạng thái trước đó là ở mức logic nào.

(Hoặc: $J_3 = 0, K_3 = 0 \Rightarrow Q_3 = Q_3^0 = 0) \Rightarrow Q_3Q_2Q_1 = 001$ khi đó

$j_1 = K_1 = \bar{Q}_3 = 1$; $J_2 = K_2 = Q_1 = 1$; $J_3 = Q_1.Q_2 = 0$; $K_3 = 1$ (Hoặc K3 = Q3 = 0).

- Khi Ck_2 ở mức tích cực theo quy định

$J_1 = K_1 = 1 \Rightarrow Q_1 = \overline{Q_1^1} = 0.$

$J_2 = K_2 = 1 \Rightarrow Q_2 = \overline{Q_2^1} = 1.$

$J_3 = 0, K_3 = 1 \Rightarrow Q_3 = 0.$

(Hoặc $J_3 = 0, K_3 = 0 \Rightarrow Q_3 = Q_3^1 = 0) \Rightarrow Q_3\,Q_2\,Q_1 = 010.$

Lúc đó: $J_1 = K_1 = \overline{Q_3} = 1$; $J_2 = K_2 = Q_1 = 0$; $J_3 = 0, K_3 = 1.$

(Hoặc $K_3 = \overline{Q_2} = 0$).

- Khi Ck_3 ở mức tích cực theo quy định

$J_1 = K_1 = 1 \Rightarrow Q_1 = \overline{Q_1^2} = 1.$

$J_2 = K_2 = 0 \Rightarrow Q_2 = Q_2^0 = 1.$

J3 = 0, K3 = 1 $\Rightarrow Q_3 = 0$ bất chấp trạng thái trước đó.

$\Rightarrow Q_3Q_2Q_1 = 011$; khi đó

$j_1 = K_1 = \bar{Q}_3 = 1$; $J_2 = K_2 = Q_1 = 1$; $J_3 = Q_1.Q_2 = 1$; $K_3 = 1$.

- Khi Ck_4 ở mức tích cực theo quy định

$J_1 = K_1 = 1 \quad \Rightarrow Q_1 = \overline{Q_1^3} = 0.$

$J_2 = K_2 = 1 \quad \Rightarrow Q_2 = \overline{Q_2^3} = 0.$

J3 = 0, K3 = 1 $\Rightarrow Q_3 = 1$ bất chấp trạng thái trước đó.

$\Rightarrow Q_3Q_2Q_1 = 100$ khi đó

$j_1 = K_1 = \bar{Q}_3 = 1$; $J_2 = K_2 = Q_1 = 0$; $J_3 = Q_1.Q_2 = 0$; $K_3 = 0$.

- Khi Ck_5 ở mức tích cực theo quy định

$J_1 = K_1 = 1 \quad \Rightarrow Q_1 = Q_1^4 = 0.$

$J_2 = K_2 = 1 \quad \Rightarrow Q_2 = Q_2^4 = 0.$

J3 = 0, K3 = 1 $\Rightarrow Q_3 = 0$ bất chấp trạng thái trước đó.

$\Rightarrow Q_3Q_2Q_1 = 000$ khi đó

$j_1 = K_1 = \bar{Q}_3 = 1$; $J_2 = K_2 = Q_1 = 0$; $J_3 = Q_1.Q_2 = 0$; $K_3 = 1$

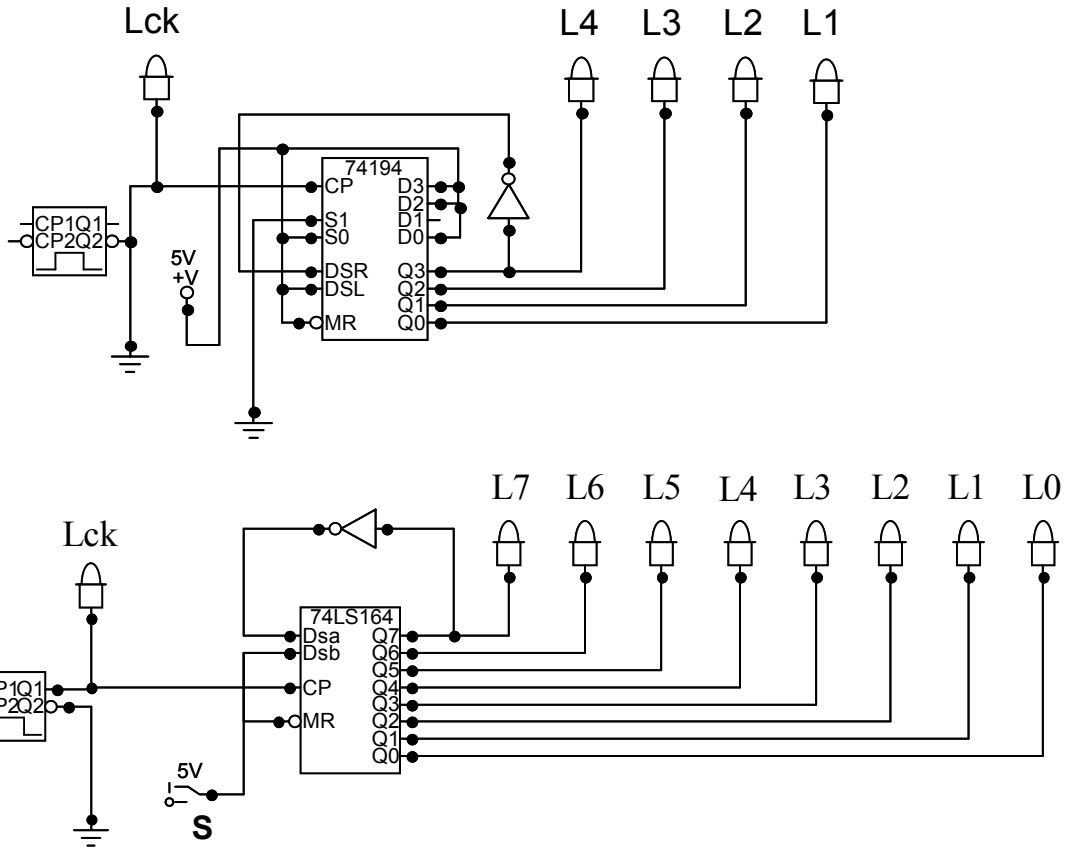
Mạch trở về trạng thái ban đầu.

2.2.3. Lắp ráp và cân chỉnh mạch đếm không đồng bộ và đồng bộ.

2.2.3.1 Lắp ráp và cân chỉnh mạch đếm không đồng bộ.

Lắp mạch đếm 4 bít dùng IC 74194; 8 bit dùng IC 74164 hiển thị LED.

Sơ đồ nguyên lý:



Trình tự thực hiện:

- Khảo sát sơ đồ chân của IC 74194
- Lắp mạch trên bo đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiệc trước khi dùng làm phần tử kết nối trong mạch.

- Bố trí các linh kiện hợp lý

- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.

- Mối hàn phải ngấu, bóng.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

➢ Lắp mạch dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp đã được thiết kế trước (không có đoạn mạch in nào được thiết kế là liên mà lại bị đứt trên bo mạch in chuẩn bị cho lắp ráp).

- Có sơ đồ lắp ráp kèm theo.

- Khi lắp phải cắm đúng vị trí và chiều các linh kiện (đặc biệt là các linh kiên bán dẫn) theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, bóng, gọn, không gây chạm chập trên mạch in.

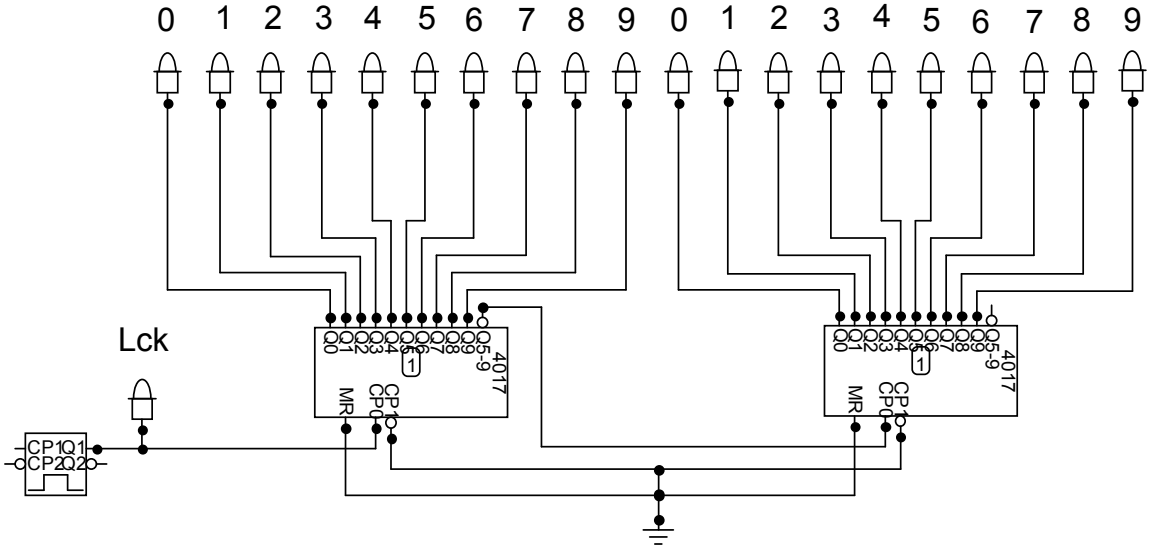
- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

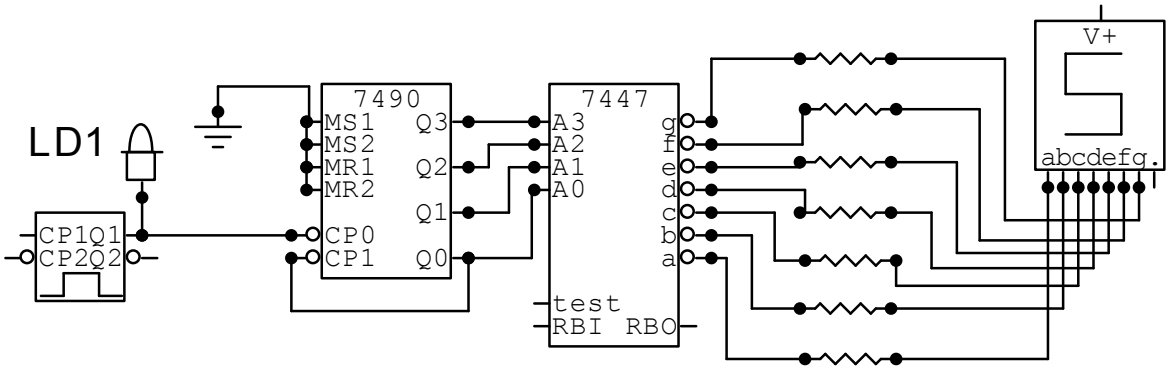
- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

2.2.3.2 Lắp ráp và cân chỉnh mạch đếm đồng bộ.

Lắp mạch đếm 100 dùng IC 4017 hiển thị LED và đếm 10 dùng IC 7490 + 7447 hiển thị LED 7 đoạn.

Sơ đồ nguyên lý:





Trình tự thực hiện:

- Khảo sát sơ đồ chân của IC 4017, 7490; 7447
- Lắp mạch trên bo đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiệc trước khi dùng làm phần tử kết nối trong mạch.

- Bố trí các linh kiện hợp lý

- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.

- Mối hàn phải ngấu, bóng.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

- Lắp mạch dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp đã được thiết kế trước (không có đoạn mạch in nào được thiết kế là liên mà lại bị đứt trên bo mạch in chuẩn bị cho lắp ráp).

- Có sơ đồ lắp ráp kèm theo.

- Khi lắp phải cắm đúng vị trí và chiều các linh kiện (đặc biệt là các linh kiên bán dẫn) theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, bóng, gọn, không gây chạm chập trên mạch in.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

2.3. Mạch đếm vòng

2.3.1. Khái niệm

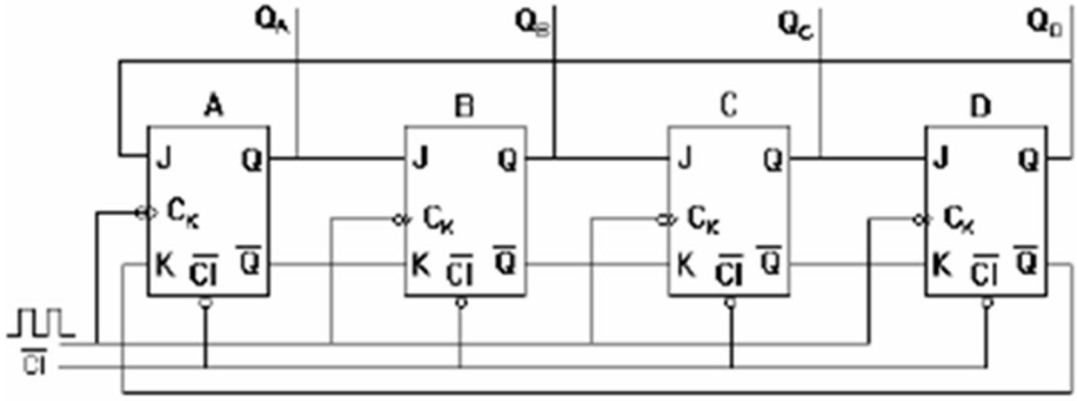
Mạch đếm vòng là một mạch đếm module M bất kỳ mà cứ hết module đó thì quá trình lại được lặp lại.

Thực chất mạch đếm vòng là một mạch ghi dịch, trong đó ta cho hồi tiếp từ một đầu ra nào đó về đầu vào để thực hiện một chu kỳ đếm. Tùy đường hồi tiếp mà ta có các chu kỳ đếm khác nhau.

2.3.2. Mạch điện thực hiện.

Ta xét một số sơ đồ mạch đếm vòng:

2.3.2.1. Hồi tiếp từ Q_D về J_A và $\overline{Q}_D$ về K_A



Đối với mạch này, sự đếm vòng chỉ thấy được khi có đặt trước đầu ra

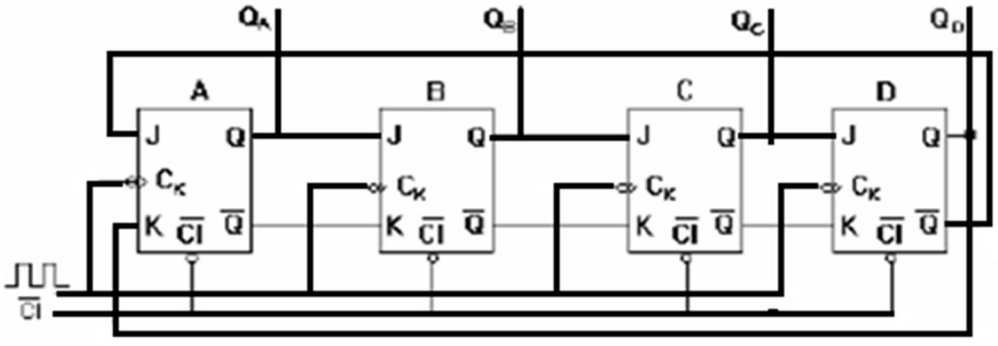
- Đặt trước Q_A =1, ta được kết quả như bảng dưới đây:

C_K	Q_D	Q_C	Q_B	Q_A	Số TP
Preset	0	0	0	1	1
1↓	0	0	1	0	2
2↓	0	1	0	0	4
3↓	1	0	0	0	8
4↓	0	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮

Nếu đặt trước $Q_A = Q_B = 1$ ta có bảng sau:

C_K	Q_D	Q_C	Q_B	Q_A	Số TP
Preset	0	0	**1**	1	**3**
1↓	0	**1**	1	0	**6**
2↓	**1**	1	0	0	**12**
3↓	1	0	0	**1**	**9**
4↓	0	0	**1**	1	**3**
⋮	⋮	⋮	⋮	⋮	⋮

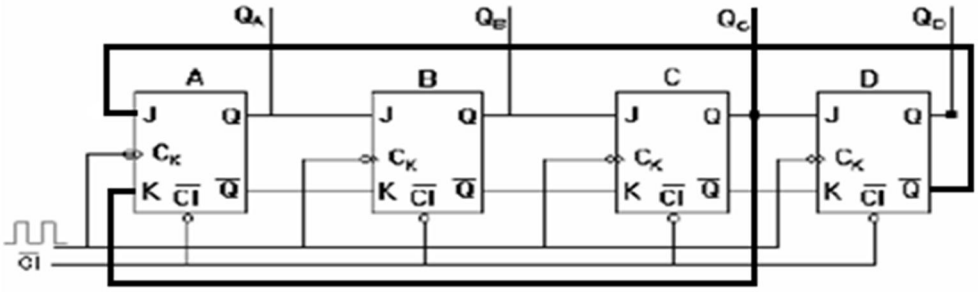
2.3.2.2. Hồi tiếp từ $\overline{Q}_D$ về J_A và Q_D về K_A :



Mạch này còn có tên là mạch đếm Johnson. Mạch có một chu kỳ đếm mặc nhiên mà không cần đặt trước. Nếu có đặt trước, mạch sẽ cho các chu kỳ khác nhau tùy vào tổ hợp đặt trước đó. Bảng dưới đây là chu kỳ đếm mặc nhiên khi không đặt trước đầu ra.

C_K	Q_D	Q_C	Q_B	Q_A	Số TP
Preset	0	0	0	0	0
1↓	0	0	0	1	1
2↓	0	0	1	1	3
3↓	0	1	1	1	7
4↓	1	1	1	1	15
5↓	1	1	1	0	14
6↓	1	1	0	0	12
7↓	1	0	0	0	8
8↓	0	0	0	0	0

2.3.2.3. Hồi tiếp từ $\overline{Q}_D$ về J_A và Q_C về K_A:



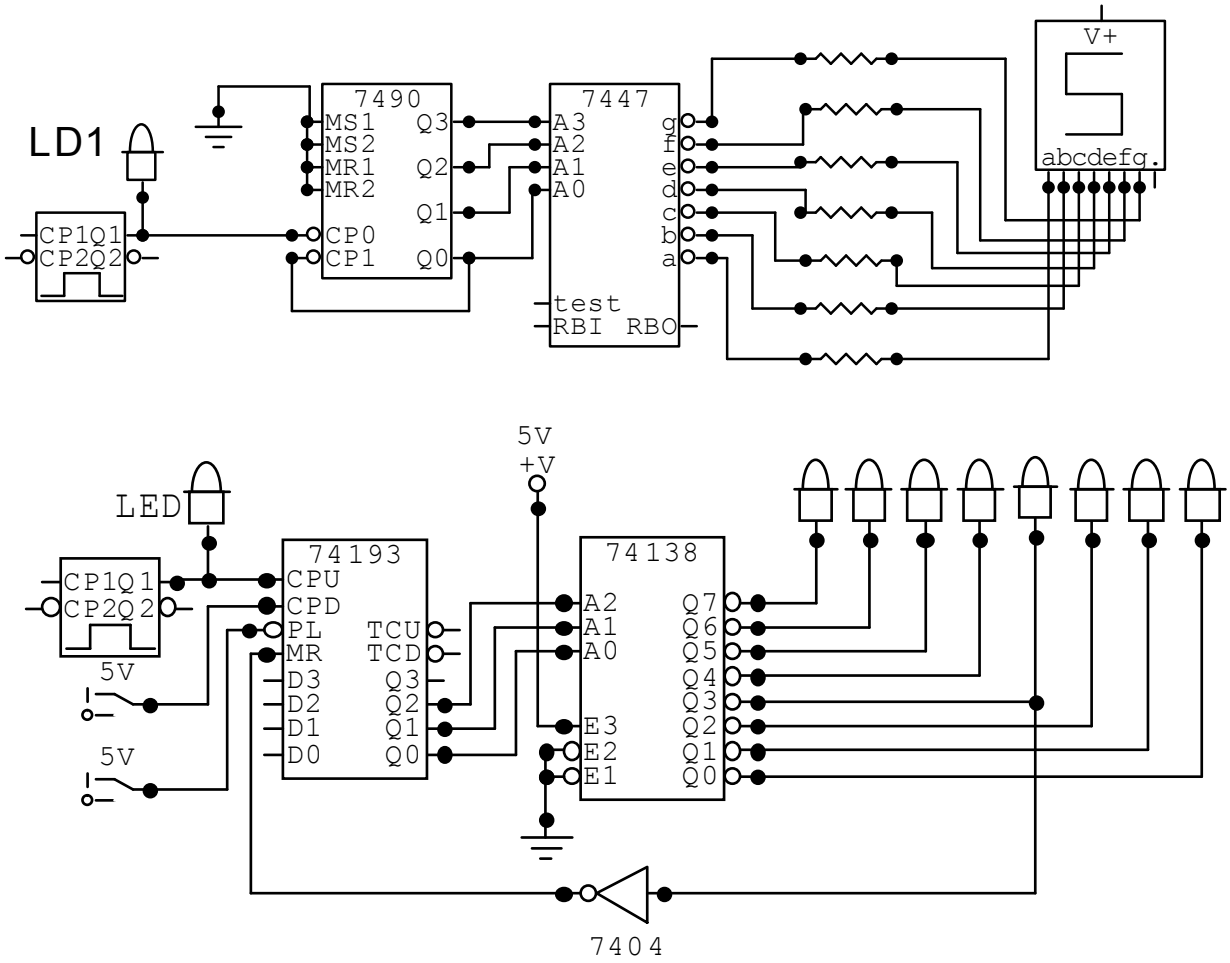
Trường hợp hồi tiếp này, bảng trạng thái làm việc cuả mạch như dưới đây:

C_K	Q_D	Q_C	Q_B	Q_A	Số TP
Preset	0	0	0	0	0
1↓	0	0	0	1	1
2↓	0	0	1	1	3
3↓	0	1	1	1	7
4↓	1	1	1	0	14
5↓	1	1	0	0	12
6↓	1	0	0	0	8
7↓	0	0	0	0	0

2.3.3. Lắp ráp mạch đếm vòng

2.3.3.1. Lắp ráp mạch lặp tuần tự:

Mục tiêu là giúp cho sinh viên hiểu và nắm được nguyên lý hoạt động của IC 74138 và IC 74193. Sơ đồ nguyên lý của mạch lặp:



Trình tự thực hiện:

- Khảo sát sơ đồ chân của IC 74138, 7404 và IC 74193
- Lắp mạch trên bo đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiệc trước khi dùng làm phần tử kết nối trong mạch.

- Bố trí các linh kiện hợp lý

- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.

- Mối hàn phải ngấu, bóng.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

- Lắp mạch dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp đã được thiết kế trước (không có đoạn mạch in nào được thiết kế là liên mà lại bị đứt trên bo mạch in chuẩn bị cho lắp ráp).

- Có sơ đồ lắp ráp kèm theo.

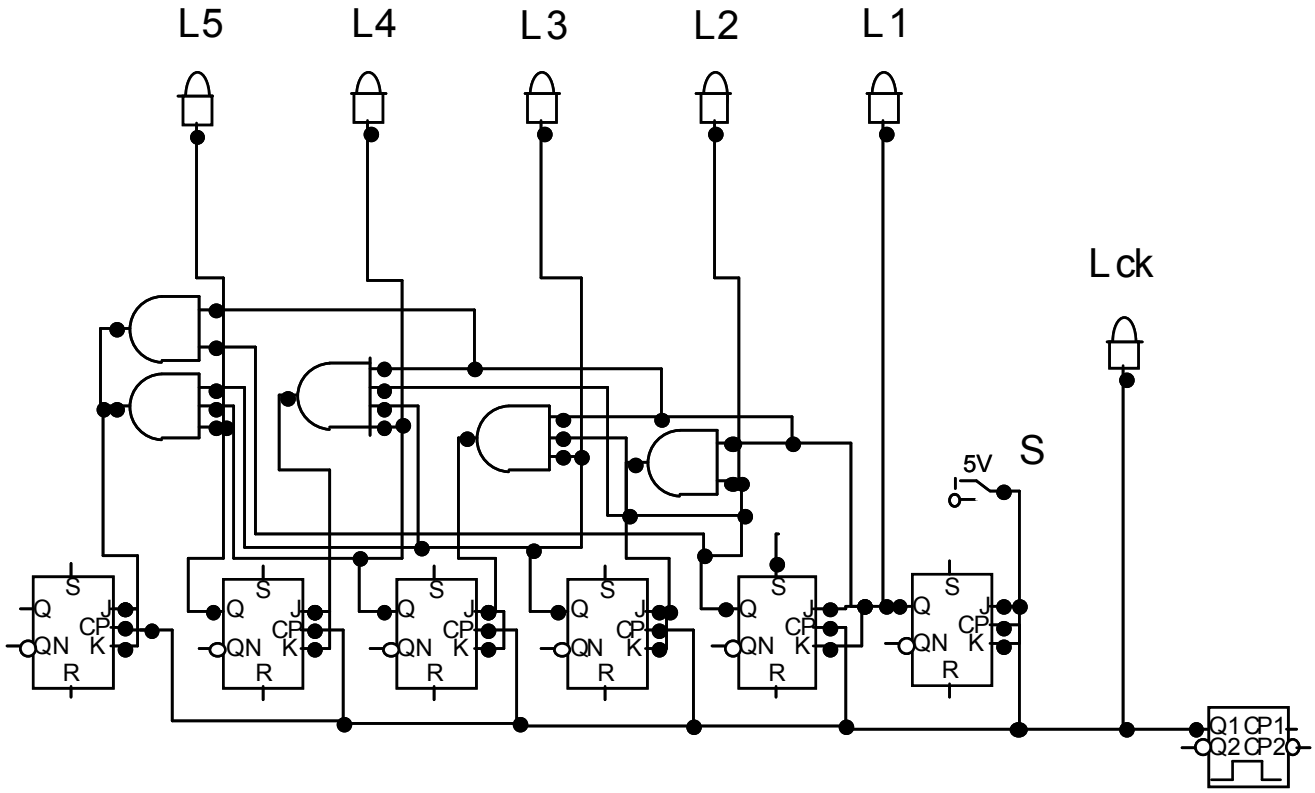
- Khi lắp phải cắm đúng vị trí và chiều các linh kiện (đặc biệt là các linh kiên bán dẫn) theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, bóng, gọn, không gây chạm chập trên mạch in.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

2.3.3.1. Lắp ráp mạch đếm đến 32 dùng IC 4027:

Sơ đồ nguyên lý:



Trình tự thực hiện:

- Khảo sát sơ đồ chân của IC 4027
- Lắp mạch trên bo đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiệc trước khi dùng làm phần tử kết nối trong mạch.

- Bố trí các linh kiện hợp lý

- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.

- Mối hàn phải ngấu, bóng.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

➢ Lắp mạch dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp đã được thiết kế trước (không có đoạn mạch in nào được thiết kế là liên mà lại bị đứt trên bo mạch in chuẩn bị cho lắp ráp).

- Có sơ đồ lắp ráp kèm theo.

- Khi lắp phải cắm đúng vị trí và chiều các linh kiện (đặc biệt là các linh kiên bán dẫn) theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, bóng, gọn, không gây chạm chập trên mạch in.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào (các xung đếm phát ra từ mạch tạo xung vuông đã chuẩn bị trước).

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và cấp xung đếm từ mạch phát xung đồng hồ (xung đếm) được chuẩn bị trước bên ngoài.

3. Các mạch ghi dịch dữ liệu

Mục tiêu:

- Trình bày được khái niệm, cấu tạo và nguyên lý hoạt động của các thanh ghi dịch trái, phải.

- Lắp ráp được thanh ghi 4 bít dịch phải dùng FF

4.1 Định nghĩa, cấu tạo, phân loại và ứng dụng

4.1.1. Định nghĩa

Thanh ghi dịch là một loại mạch logic tuần tự, được ứng dụng trong lưu trữ dữ liệu, trong đó thanh ghi được sử dụng như bộ nhớ tạm thời (lưu kết quả các phép tính) do khả năng lưu trữ của nó có hạn.

Còn bộ nhớ có khả năng lưu trữ các bit dữ liệu khá lớn. Về mặc cấu tạo bên trong nó được xây dựng trên cơ sở các thanh ghi (nhiều thanh ghi hợp thành bộ nhớ).

4.1.2. Cấu tạo

Thanh ghi được xây dựng trên cơ sở các DFF (hoặc các FF khác thực hiện chức năng của DFF) và trong đó mỗi DFF sẽ lưu trữ 1 bit dữ liệu.

Để tạo thanh ghi nhiều bit, người ta ghép nhiều DFF lại với nhau theo qui luật như sau:

- Đầu ra của DFF đứng trước được nối với đầu vào DATA của DFF sau **($D_{i+1} = Q_i$) ⇒ thanh ghi có khả năng dịch phải.**

- Hoặc đầu ra của DFF đứng sau được nối với đầu vào DATA của DFF đứng trước **($D_i = Q_{i+1}$) ⇒ thanh ghi có khả năng dịch trái.**

4.1.3. Phân loại

- Phân loại theo số bit dữ liệu lưu trữ: 4 bit, 5 bit, 8 bit, 16 bit, 32 bit. Đối với thanh ghi lớn 8 bit, người ta không dùng họ TTL mà dùng họ CMOS.

- Phân loại theo hướng dịch chuyển dữ liệu trong thanh ghi:
    + Thanh ghi dịch trái.
    + Thanh ghi dịch phải.
    + Thanh ghi vừa dịch phải vừa dịch trái.
- Phân loại theo đầu vào dữ liệu:
    + Đầu vào dữ liệu nối tiếp.
    + Đầu vào dữ liệu song song: Song song không đồng bộ, song song đồng bộ.
- Phân loại theo đầu ra:
    + Đầu ra nối tiếp.
    + Đầu ra song song.
    + Đầu ra vừa nối tiếp vừa song song.

4.1.4. Ứng dụng

Ghi dịch có khá nhiều ứng dụng:

- Một số nhị phân khi dịch trái 1 bit, giá trị được nhân lên gấp đôi và được chia hai khi dịch phải một bit.

Ví dụ số $1010.00 = 10_{10}$

- Khi dịch trái thành $10100.0 = 20_{10}$
- Khi dịch phải thành $101.000 = 5_{10}$.
- Trong máy tính thanh ghi (tên thường gọi của mạch ghi dịch) là nơi lưu tạm dữ liệu để thực hiện các phép tính, các lệnh cơ bản như quay, dịch ....
- Ngoài ra, mạch ghi dịch còn những ứng dụng khác như: tạo mạch đếm vòng, biến đổi dữ liệu nối tiếp ↔ song song, dùng thiết kế các mạch đèn trang trí, quang báo. . . ..

4.2 Thanh ghi dịch phải

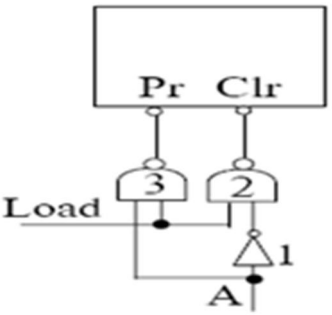
4.2.1. Khái niệm

4.2.1.1. Khái niệm

Là thanh ghi được xây dựng trên cơ sở các DFF (hoặc các FF khác thực hiện chức năng của DFF) và trong đó mỗi DFF sẽ lưu trữ 1 bit dữ liệu. Khi dữ liệu được đưa tới đầu vào thì chúng sẽ lần lượt thay thế các bít dữ liệu cũ và các bit dữ liệu cũ lần lượt được dịch chuyển trên các DFF từ trái qua phải theo thứ tự sắp xếp các DFF.

4.2.1.2. Cách nhập dữ liệu vào thanh ghi

- Nhập dữ liệu vào FF bằng chân Preset (Pr):



+ Khi Load = 0: Cổng NAND 3 và 2 khóa → đầu vào Pr = Clr = 1 → FF tự do → dữ liệu A không nhập vào được FF.

+ Khi Load = 1: Cổng NAND 2 và 3 mở. Lúc đó ta có: Pr = A, Clr = A.

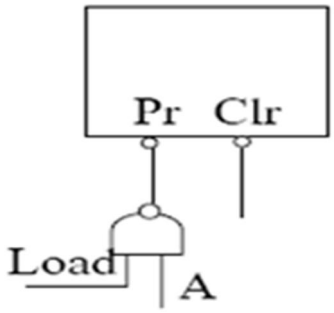
Nếu A = 0 $\Rightarrow$ Pr = 1, Clr = 0 $\Rightarrow$ Q = A = 0.

Nếu A = 1 $\Rightarrow$ Pr = 0, Clr = 1 $\Rightarrow$ Q = A = 1.

Vậy Q = A $\Rightarrow$ dữ liệu A được nhập vào FF.

Tuy nhiên, cách này phải dùng nhiều cổng logic không kinh tế và phải dùng chân Clr là chân xóa nên phải thiết kế đồng bộ.

Để khắc phục những nhược điểm đó ta dùng mạch như trên hình sau:



Chân Clr để trống tương đương với mức logic 1.

+ Khi Load = 0: cổng NAND khóa $\Rightarrow$ Pr = Clr =1 $\Rightarrow$ FF tự do. Dữ liệu không được nhập vào FF.

+ Khi Load = 1: cổng NAND mở $\Rightarrow$ Pr = A.

Giả sử ban đầu: Q = 0.

Nếu A = 0 $\Rightarrow$ Pr = 1, Clr = 1 $\Rightarrow$ Q = Q_0 = 0.

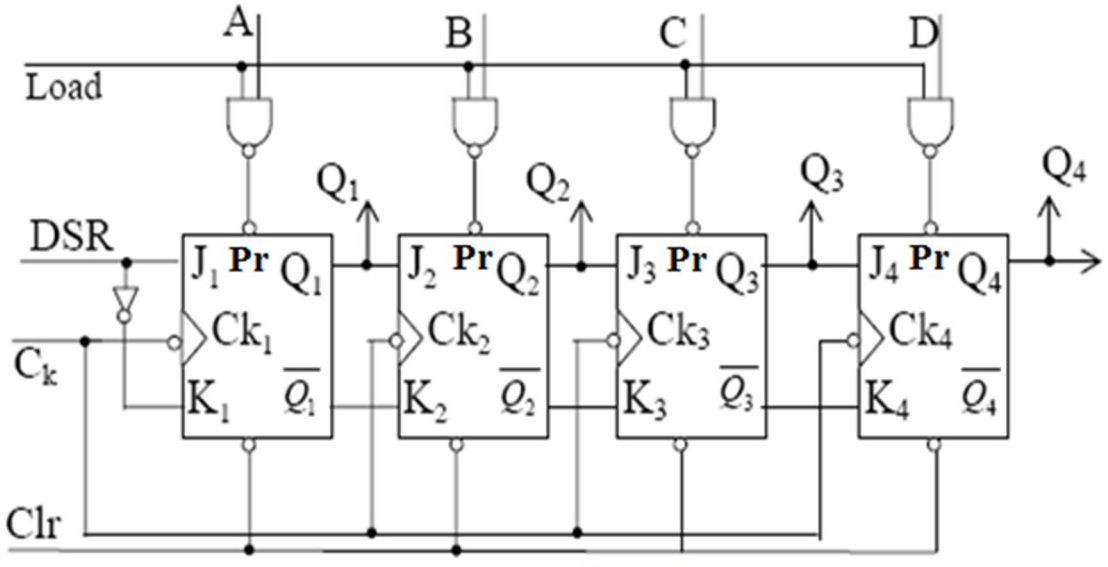
Nếu A = 1 $\Rightarrow$ Pr = 0, Clr = 1 $\Rightarrow$ Q = 1.

Vậy Q = A $\Rightarrow$ Dữ liệu A được nhập vào FF.

Chú ý: Phương pháp này đòi hỏi trước khi nhập phải xóa FF về 0.

4.2.2. Mạch điện điển hình

Xét một thanh 4 bit có khả năng dời phải.



Thanh ghi dịch phải 4 bit

Trong đó:

- DSR (Data Shift Right): Đầu vào Data nối tiếp (đầu vào dịch phải).
- Q_1, Q_2,Q_3, Q_4 : các đầu ra song song.

Để giải thích hoạt động của mạch, ta dựa vào bảng trạng thái của DFF.

Giả sử ban đầu: Đầu vào nhập Load = 1 → A, B, C, D được nhập vào thanh ghi dịch → Q_1 = A, Q_2 = B, Q_3 = C, Q_4 = D.

Hoạt động dịch phải của thanh ghi:

- Xét FF_1 : D = DSR_1, Q_1 = A.

  Nếu DSR_1 = 0 → Q = 0 ; nếu DSR_1 = 1 → Q = 1.

Kết luận: Sau một xung Ck tác động sườn xuống thì Q_1 = DSR_1.

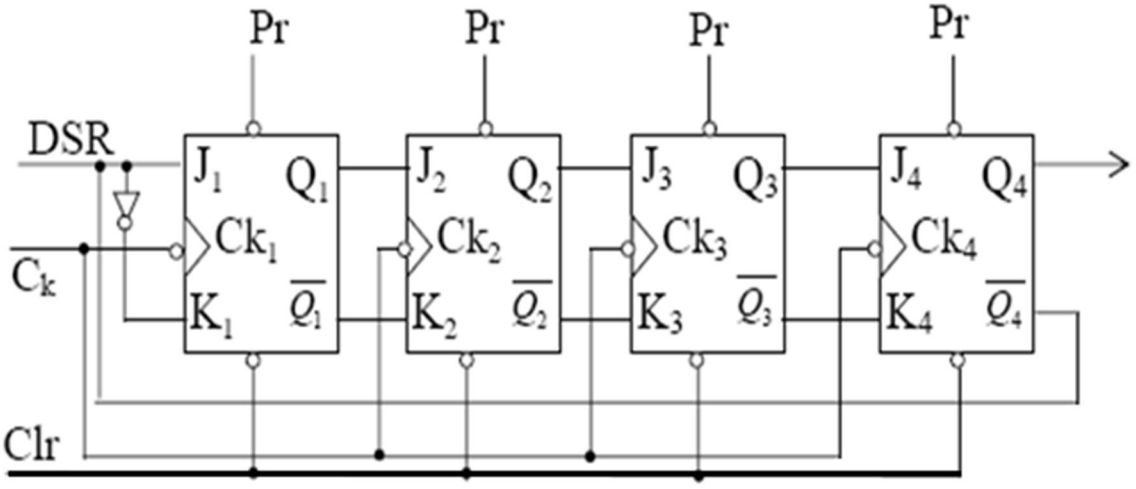
- Lúc đó FF_2, FF_3,FF_4 : Q_2 = A, Q_3 = B, Q_4 = C.

Tức là sau khi Ck tác động sườn xuống thì nội dung trong thanh ghi được dời sang phải 1 bit. Sau 4 xung, dữ liệu trong thanh ghi được xuất ra ngoài và nội dung DFF được thay thế bằng các dữ liệu từ đầu vào DATA nối tiếp DSR_1, DSR_2, DSR_3, DSR_4.

Ta có bảng trạng thái hoạt động của mạch:

Xung vào	Trạng thái hiện tại				Trạng thái kế			
	Q_1	Q_2	Q_3	Q_4	Q_1	Q_2	Q_3	Q_4
1	A	B	C	D	DSR_1	A	B	C
2	DSR_1	A	B	C	DSR_2	DSR_1	A	B
3	DSR_2	DSR_1	A	B	DSR_3	DSR_2	DSR_1	A
4	DSR_3	DSR_2	DSR_1	A	DSR_4	DSR_3	DSR_2	DSR_1

Trường hợp đầu ra Q bằng đầu vào dữ liệu nối tiếp DSR.



Ta có bảng trạng thái hoạt động của mạch như sau:

Xung vào	Trạng thái hiện tại				Trạng thái kế			
	Q_1	Q_2	Q_3	Q_4	Q_1	Q_2	Q_3	Q_4
1	0	0	0	0	1	0	0	0
2	1	0	0	0	1	1	0	0
3	1	1	0	0	1	1	1	0
4	1	1	1	0	1	1	1	1
5	1	1	1	1	0	1	1	1
6	0	1	1	1	0	0	1	1
7	0	0	1	1	0	0	0	1
8	0	0	0	1	0	0	0	0

Đây là mạch được ứng dụng nhiều trong thực tế.

4.3 Thanh ghi dịch trái

4.3.1. Khái niệm

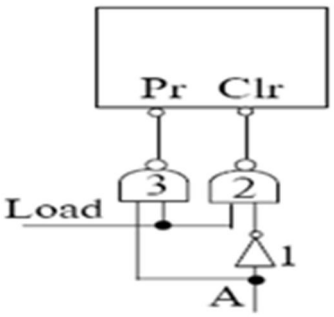
4.3.1.1. Khái niệm

Là thanh ghi được xây dựng trên cơ sở các DFF (hoặc các FF khác thực hiện chức năng của DFF) và trong đó mỗi DFF sẽ lưu trữ 1 bit dữ liệu. Khi dữ liệu được đưa tới đầu vào thì chúng sẽ lần lượt thay thế các bít dữ liệu cũ và các bit dữ liệu cũ lần lượt được dịch chuyển trên các DFF từ phải qua trái theo thứ tự sắp xếp các DFF.

4.3.1.2. Cách nhập dữ liệu vào thanh ghi

Cũng giống như thanh ghi dịch phải, thanh ghi dịch trái có thể nhập dữ liệu vào các FF theo các phương pháp sau:

- Nhập dữ liệu vào FF bằng chân Preset (Pr):



+ Khi Load = 0: Cổng NAND 3 và 2 khóa → đầu vào Pr = Clr = 1 → FF tự do → dữ liệu A không nhập vào được FF.

+ Khi Load = 1: Cổng NAND 2 và 3 mở. Lúc đó ta có: Pr = A, Clr = A.

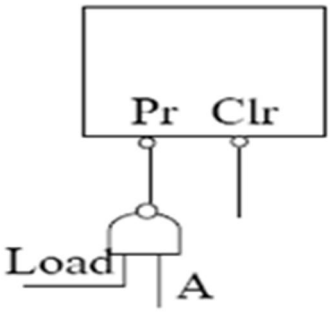
Nếu A = 0 $\Rightarrow$ Pr = 1, Clr = 0 $\Rightarrow$ Q = A = 0.

Nếu A = 1 $\Rightarrow$ Pr = 0, Clr = 1 $\Rightarrow$ Q = A = 1.

Vậy Q = A $\Rightarrow$ dữ liệu A được nhập vào FF.

Tuy nhiên, cách này phải dùng nhiều cổng logic không kinh tế và phải dùng chân Clr là chân xóa nên phải thiết kế đồng bộ.

Để khắc phục những nhược điểm đó ta dùng mạch như trên hình sau:



Chân Clr để trống tương đương với mức logic 1.

+ Khi Load = 0: cổng NAND khóa $\Rightarrow$ Pr = Clr =1 $\Rightarrow$ FF tự do. Dữ liệu không được nhập vào FF.

+ Khi Load = 1: cổng NAND mở $\Rightarrow$ Pr = A.

Giả sử ban đầu: Q = 0.

Nếu A = 0 $\Rightarrow$ Pr = 1, Clr = 1 $\Rightarrow$ Q = Q_0 = 0.

Nếu A = 1 $\Rightarrow$ Pr = 0, Clr = 1 $\Rightarrow$ Q = 1.

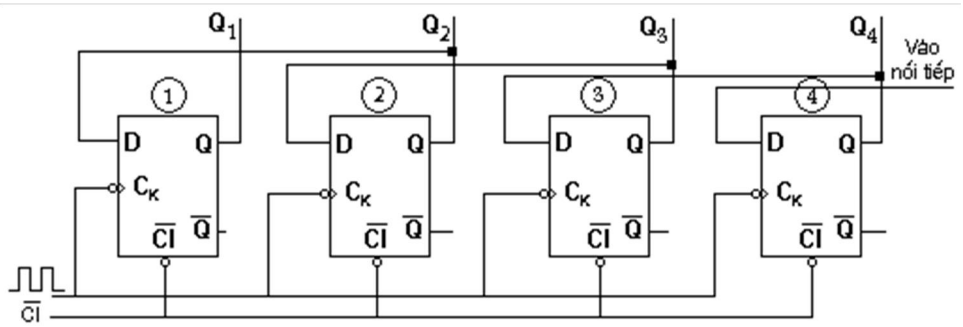
Vậy Q = A $\Rightarrow$ Dữ liệu A được nhập vào FF.

Phương pháp này cũng đòi hỏi trước khi nhập phải xóa FF về 0.

4.3.2. Mạch điện điển hình

Xét một thanh 4 bit có khả năng dời trái.

Để có mạch dịch trái, dữ liệu nối tiếp đưa vào đầu vào D của FF cuối cùng và các đầu ra của FF sau nối ngược trở lại đầu vào của FF trước.



Trong đó:

- D: Đầu vào Data nối tiếp (đầu vào dịch trái).
- Q_1, Q_2,Q_3, Q_4: các đầu ra song song.

Cho dữ liệu nối tiếp vào đầu vào D của FF 4, sau mỗi xung đồng hồ, dữ liệu truyền từ tầng sau ra tầng trước. Giả sử chuỗi dữ liệu giống như trên, trạng thái các đầu ra của các FF cho ở bảng sau:

Vào			Ra			
Cl	C_K	D_4	Q_1	Q_2	Q_3	Q_4
0	x	x	0	0	0	0
1	↓	1	0	0	0	1
1	↓	1	0	0	1	1
1	↓	1	0	1	1	1
1	↓	0	1	1	1	0
1	↓	0	1	1	0	0
1	↓	1	1	0	0	1
1	↓	0	0	0	1	0

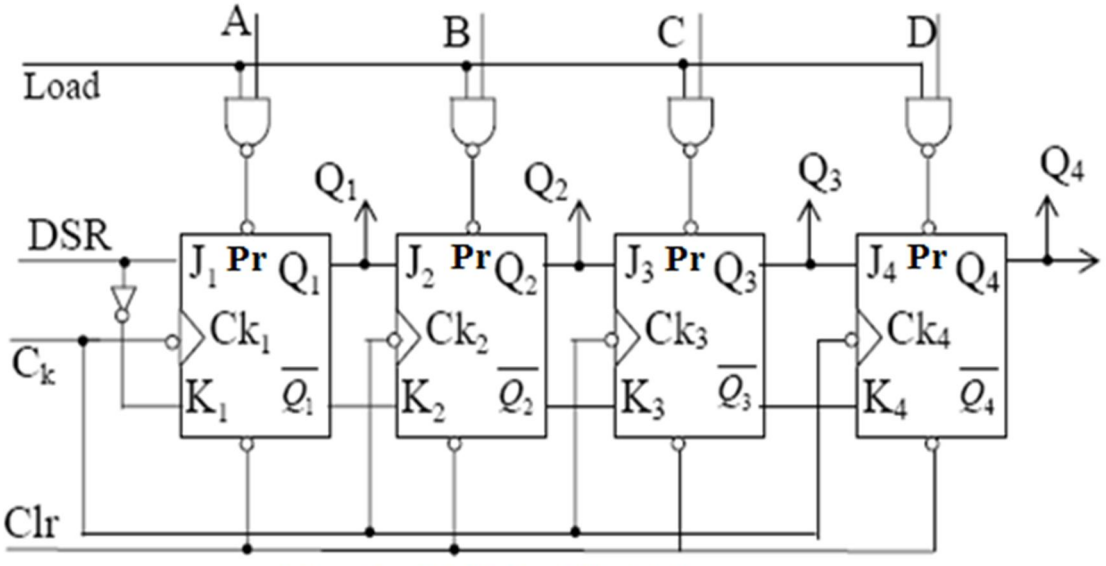
Trên thị trường hiện có khá nhiều loại IC ghi dịch, có đầy đủ các chức năng dịch phải trái, vào/ra nối tiếp, song song. Sau đây, chúng ta khảo sát 2 IC tiêu biểu:

- IC 74164: dịch phải 8 bit;
- IC 7495: 4 bit, dịch phải, trái, vào/ra nối tiếp/song song .

4.4. Lắp ráp và cân chỉnh thanh ghi 4 bít dịch phải sử dụng FF.

4.4.1. Lắp ráp và cân chỉnh thanh ghi 4 bít dịch phải sử dụng FF.

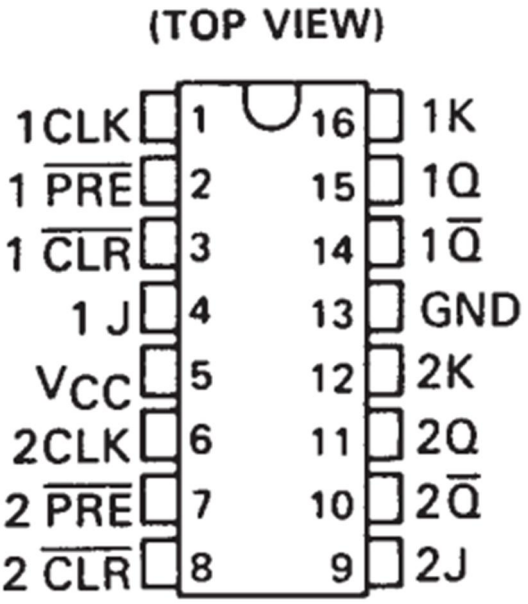
Mạch nguyên lý như ở hình dưới đây:

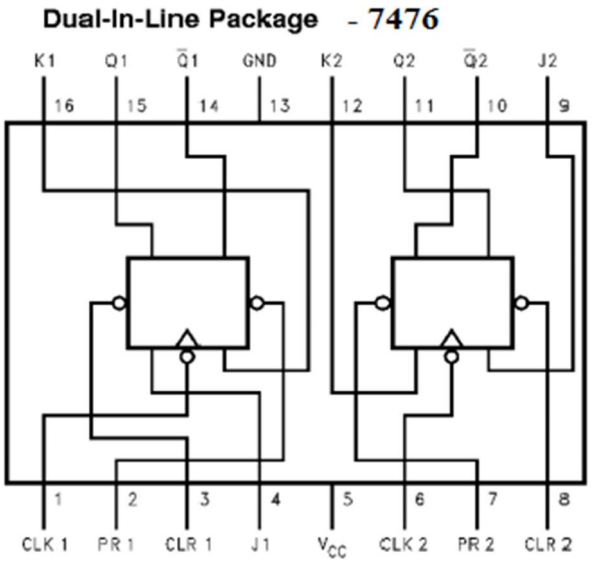


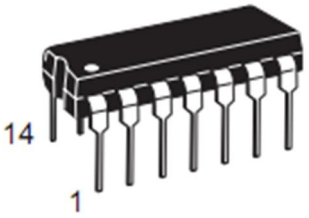
Thanh ghi dịch phải 4 bit

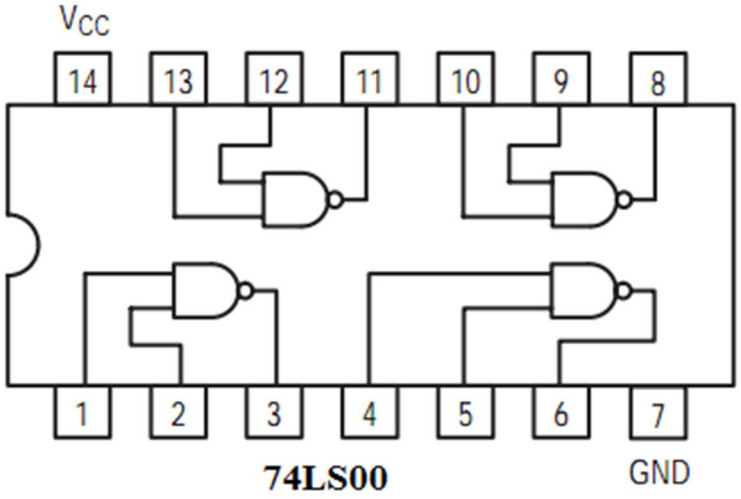
Mạch sử dụng 2 IC 7476 và 1 IC 7400

Sơ đồ chân và cấu trúc bên trong của IC 7476 và 7400:



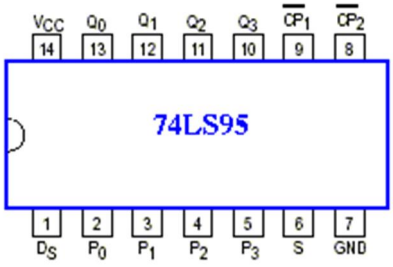


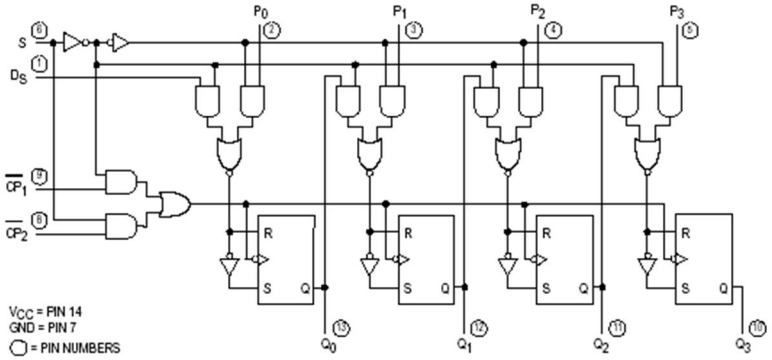




4.4.2. Lắp ráp và cân chỉnh thanh ghi 4 bít dịch phải sử dụng IC 74LS95

. Sơ đồ chân và cấu trúc bên trong của IC 74LS95 như sau:





4. Một số mạch chuyển đổi và ứng dụng của Flip Flop

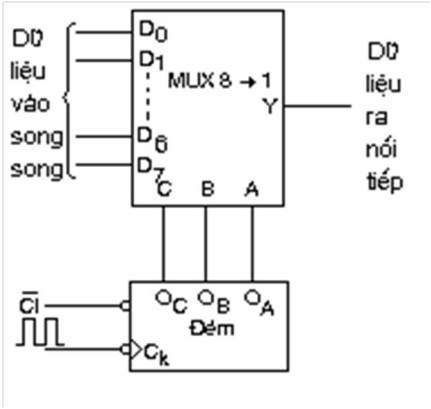
Mục tiêu:

Trình bày được một số mạch chuyển đổi cơ bản và một số ứng dụng của FF đã học.

4.1. Một số mạch chuyển đổi thông dụng.

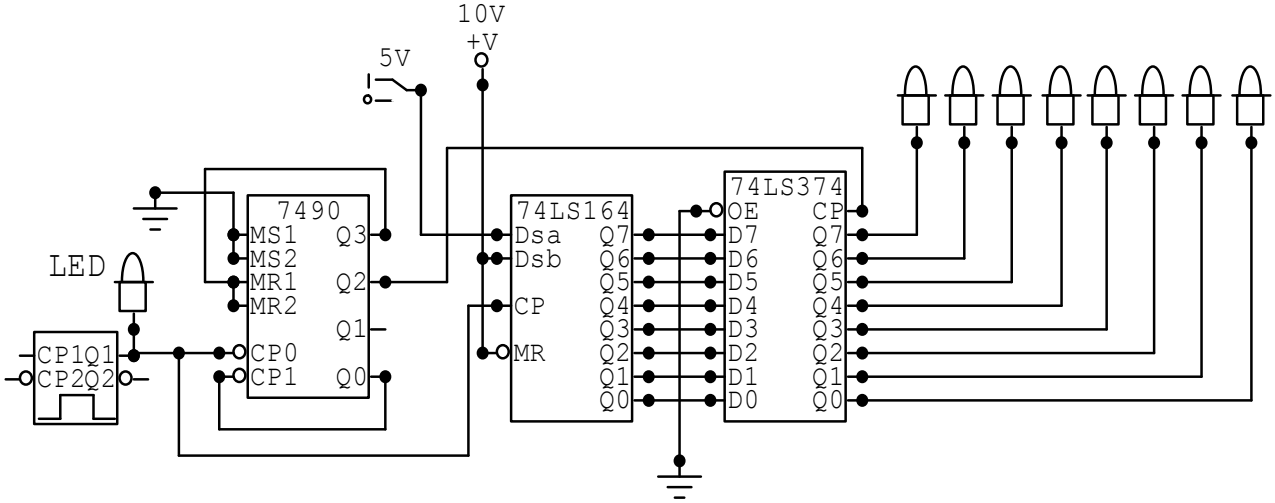
4.1.1. Mạch chuyển đổi dữ liệu song song thành nối tiếp:

Một mạch dồn kênh kết hợp với một mạch đếm sẽ biến chuỗi dữ liệu song song ở đầu vào thành chuỗi dữ liệu nối tiếp ở đầu ra.



4.1.2. Mạch chuyển đổi dữ liệu nối tiếp thành song song:

Sơ đồ nguyên lý của mạch chuyển đổi:



Trong đó IC 7490 đóng vai trò mạch đếm nhị - thập phân.

IC 74164 là một thanh ghi chuyển dịch 8 bit, nó hoạt động như sau:

Dữ liệu một trong hai đầu vào nối tiếp được chuyển thành một bit tương ứng với mỗi xung clock. Dữ liệu có thể lấy ra dưới dạng song song ở 8 đầu ra (hoặc dưới dạng nối tiếp ở một đầu ra đơn lẻ bất kỳ). Dữ liệu có thể đưa vào một trong hai đầu vào. Đầu vào còn lại được treo lên mức cao hoặc cấm xung nhịp.

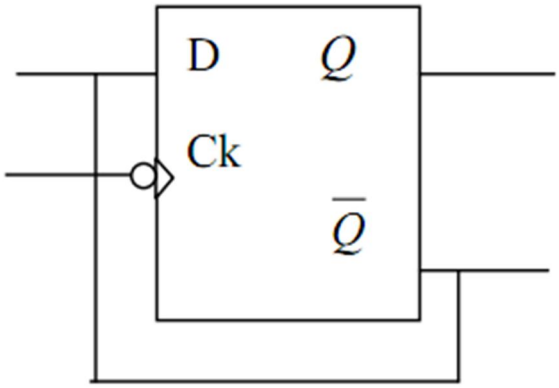
IC 74374 là một IC dịch chuyển có hoạt động như sau: Khi chân PL lên mức logic 1 thì toàn bộ tín hiệu ở các đầu vào được dịch đồng loạt ra đầu ra.

4.2 Một số ứng dụng của FF

4.2.1. Ứng dụng của DFF:

4.2.1.1. Dùng DFF để chia tần số.

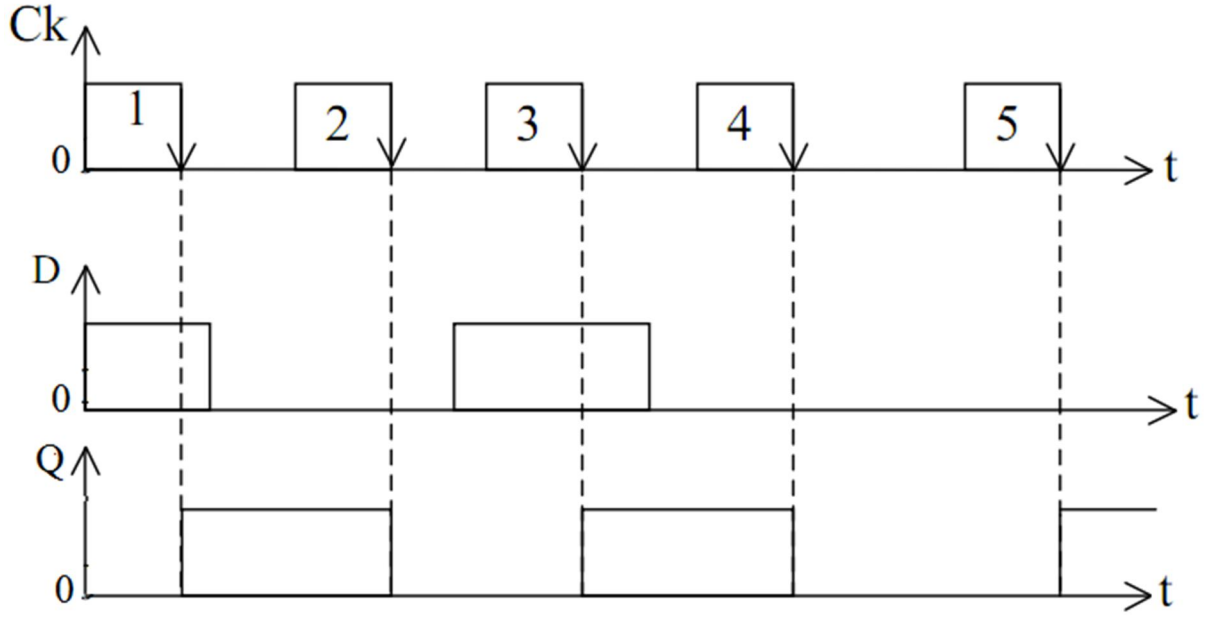
Sơ đồ mạch DFF thực hiện chức năng chia tần số.



Ở mạch này đầu ra $\bar{Q}$ được nối ngược trở về đầu vào D.

- Tín hiệu ra Q_0 đầu tiên luôn ở mức logic 0: $Q_0 = 0 \rightarrow \bar{Q}_0 = D1 = 1$
- Tín hiệu Ck (1) điều khiển theo sườn xuống nhìn tín hiệu D1 dưới mức logic 1.
  D1 = 1 → Q1 = 1 → $\bar{Q}_1 = D2 = 0$.
- Tín hiệu Ck (2) điều khiển theo sườn xuống nhìn tín hiệu D2 dưới mức logic 0.
  D2 = 0 → Q2 = 0 → $\bar{Q}2$ = D3 = 1.
- Tín hiệu Ck (3) điều khiển theo sườn xuống nhìn tín hiệu D3 dưới mức logic 1.
  D3 = 1 → Q3 = 0 → $\bar{Q}3$ = D4 = 0.
- Tín hiệu Ck(4) điều khiển theo sườn xuống nhìn tín hiệu D4 dưới mức logic 0.
  D4 = 0 → Q4 = 0 → $\bar{Q}4$ = D5 = 1 .....

  Biểu đồ thời gian của mạch:



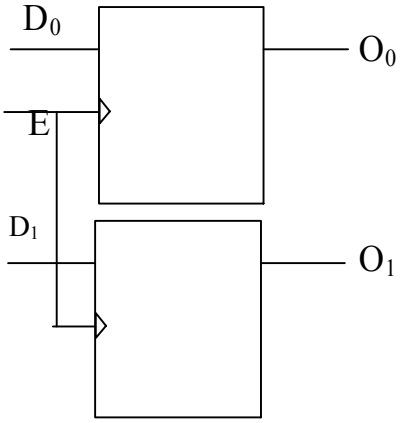
Nhận xét về tần số đầu ra:

$f_Q = \frac{f_{CK}}{2}$ → DFF giữ vai trò như mạch chia tần số.

- Dùng DFF để lưu trữ dữ liệu để chế tạo các bộ nhớ và thanh ghi.

4.2.1.2. Dùng DFF để chốt dữ liệu.

Sơ đồ mạch ứng dụng DFF để chốt dữ liệu.



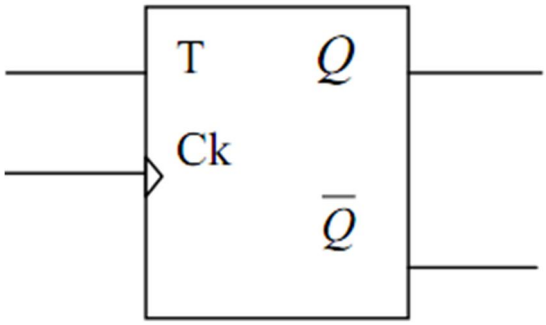
Hoạt động của mạch như sau:

- E = 1: $O_0 = D_0$, $O_1 = D_1$ nên tín hiệu được đưa đến các FF.
- E = 0: Mạch không tiếp nhận dữ liệu mới; $O_0 = D_0$, $O_1 = D_1$ → Mạch chốt dữ liệu trở lại.

4.2.2. Ứng dụng của TFF:

Như ta đã biết ở trên, TFF dùng để làm mạch đếm. Tuy vậy có thể dùng TFF để làm mạch chia tần số.

Sơ đồ mạch TFF thực hiện chức năng chia tần số.



Khi T = 1 thì TFF có thể thực hiện chức năng chia tần giống như với DFF. Khi số lượng TFF là 1 thì

$f_Q = \frac{f_{CK}}{2}$ → DFF giữ vai trò như mạch chia tần số.

Tổng quát: Khi ghép nối tiếp n TFF với nhau sao cho đầu ra của TFF trước được nối với đầu vào của TFF đứng sau (C_{ki+1} nối với Q_i) và lúc bây giờ tất cả các đầu vào DATA T ở tất cả các TFF đều giữ mức logic 1, lúc đó tần số tín hiệu đầu ra sẽ là:

$$f_{Q_n} = \frac{f_{CK}}{2^n}$$

Với Q_n là tín hiệu đầu ra của TFF thứ n; f_{CK} là tần số xung clock ở đầu vào đồng bộ TFF đầu tiên.

BÀI 9
MẠCH NHỚ ROM
Mã bài: MĐ26.09

Giới thiệu:

Tính ưu việt chủ yếu của các hệ thống số so với hệ thống tương tự là khả năng lưu trữ một lượng lớn thông tin số và dữ liệu trong những khoảng thời gian nhất định. Khả năng nhớ này là điều làm cho hệ thống số trở thành đa năng và có thể thích hợp với nhiều tình huống. Ví dụ trong một máy tính số, bộ nhớ trong chứa những lệnh mà theo đó máy tính có thể hoàn tất công việc của mình với sự tham gia ít nhất của con người. Bộ nhớ bán dẫn được sử dụng làm **bộ nhớ chính** trong các máy tính nhờ vào khả năng thỏa mãn tốc độ truy xuất dữ liệu của bộ xử lý trung tâm (CPU).

Ở chương trước, chúng ta đã thấy một nhóm các FF họp thành thanh ghi để lưu trữ và dịch chuyển thông tin. Các FF chính là các phần tử nhớ tốc độ cao được dùng rất nhiều trong việc điều hành bên trong máy tính, nơi mà dữ liệu dịch chuyển liên tục từ nơi này đến nơi khác.

Tiến bộ trong công nghệ chế tạo LSI và VLSI cho phép kết hợp một lượng lớn FF trong một chip tạo thành các bộ nhớ với các dạng khác nhau. Những bộ nhớ bán dẫn với công nghệ chế tạo transistor lưỡng cực (BJT) và MOS là những bộ nhớ nhanh nhất và giá thành của nó liên tục giảm khi các công nghệ LSI và VLSI ngày càng được cải tiến.

Bộ nhớ bán dẫn được dùng như là **bộ nhớ trong** chính của máy tính, nơi mà việc vận hành nhanh được xem như ưu tiên hàng đầu và cũng là nơi mà tất cả dữ liệu của chương trình lưu chuyển liên tục trong quá trình thực hiện một tác vụ do CPU yêu cầu.

Mặc dù bộ nhớ bán dẫn có tốc độ làm việc cao, rất phù hợp cho bộ nhớ trong, nhưng giá thành tính trên mỗi bit lưu trữ cao khiến cho nó không thể là loại thiết bị có tính chất lưu trữ khối, là loại thiết bị có khả năng lưu trữ hàng tỉ bit mà không cần cung cấp năng lượng và được dùng như là **bộ nhớ ngoài** (đĩa từ, băng từ, CD ROM . . .). Tốc độ xử lý dữ liệu ở bộ nhớ ngoài tương đối chậm nên khi máy tính làm việc thì dữ liệu từ bộ nhớ ngoài được chuyển vào bộ nhớ trong.

Bài học này nghiên cứu cấu tạo và tổ chức chung của các **bộ nhớ bán dẫn** và riêng cho ROM.

Mục tiêu:

- Trình bày được đặc tính, cấu trúc, nguyên lý hoạt động và ứng dụng của mạch nhớ.
- Lắp ráp bộ nhớ ROM
- Có tính tư duy, tác phong trong công nghiệp

Nội dung chính:

1. Khái niệm chung

Mục tiêu:

Trình bày được khái niệm và các phương pháp phân loại bộ nhớ bán dẫn nói chung và bộ nhớ ROM nói riêng.

1.1. Khái niệm

1.1.1. Thuật ngữ liên quan đến bộ nhớ

Để tìm hiểu cấu tạo, hoạt động của bộ nhớ chúng ta bắt đầu với một số thuật ngữ liên quan đến bộ nhớ

- Tế bào nhớ: là linh kiện hay một mạch điện tử dùng để lưu trữ một bit đơn (0 hay 1). Ví dụ của một tế bào nhớ bao gồm: mạch FF, tụ được tích điện, một điểm trên băng từ hay đĩa từ. . . .

- Từ nhớ: là một nhóm các bit (tế bào) trong bộ nhớ dùng biểu diễn các lệnh hay dữ liệu dưới dạng một số nhị phân. Ví dụ một thanh ghi 8 FF là một phần tử nhớ lưu trữ từ 8 bit. Kích thước của từ nhớ trong các máy tính hiện đại có chiều dài từ 4 đến 64 bit.

- Byte: từ 8 bit, đây là kích thước thường dùng của từ nhớ trong các PC.

- Dung lượng: chỉ số lượng bit có thể lưu trữ trong bộ nhớ. Ví dụ bộ nhớ có khả năng lưu trữ 4.096 từ nhớ 20 bit, dung lượng của nó là 4096 x 20, mỗi 1024 (=210) từ nhớ được gọi là "1K", như vậy 4096 x 20 = 4K x 20. Với dung lượng lớn hơn ta dùng "1M" hay 1meg để chỉ 220 = 1.048.576 từ nhớ.

- Địa chỉ: là số nhị phân dùng xác định vị trí của từ nhớ trong bộ nhớ. Mỗi từ nhớ được lưu trong bộ nhớ tại một địa chỉ duy nhất. Địa chỉ luôn luôn được biểu diễn bởi số nhị phân, tuy nhiên để thuận tiện người ta có thể dùng số hex hay thập phân, bát phân

- Tác vụ đọc (Read, còn gọi là fetch): một từ nhớ tại một vị trí nào đó trong bộ nhớ được truy xuất và chuyển sang một thiết bị khác.

- Tác vụ viết (ghi, Write, còn gọi là store): một từ mới được đặt vào một vị trí trong bộ nhớ, khi một từ mới được viết vào thì từ cũ mất đi.

- Thời gian truy xuất (access time): số đo tốc độ hoạt động của bộ nhớ, ký hiệu t_{ACC}, là thời gian cần để hoàn tất một tác vụ đọc. Đó là thời gian từ khi bộ nhớ nhận một địa chỉ mới cho tới lúc dữ liệu khả dụng ở đầu ra bộ nhớ.

- Bộ nhớ không vĩnh cữu (volatile): Bộ nhớ cần nguồn điện để lưu trữ thông tin. Khi ngắt điện, thông tin lưu trữ bị mất. Hầu hết bộ nhớ bán dẫn là loại không vĩnh cữu, trong khi bộ nhớ từ là loại vĩnh cữu (nonvolatile).

- Bộ nhớ truy xuất ngẫu nhiên (Random-Access Memory, RAM): Khi cần truy xuất một địa chỉ ta tới ngay địa chỉ đó. Vậy thời gian đọc hay viết dữ liệu vào các vị trí nhớ khác nhau trong bộ nhớ không tùy thuộc vào vị trí nhớ. Nói cách khác, thời gian truy xuất như nhau đối với mọi vị trí nhớ. Hầu hết bộ nhớ bán dẫn và nhẫn từ (bộ nhớ trong của máy tính trước khi bộ nhớ bán dẫn ra đời) là loại truy xuất ngẫu nhiên.

- Bộ nhớ truy xuất tuần tự (Sequential-Access Memory, SAM): Khi cần truy xuất một địa chỉ ta phải lướt qua các địa chỉ trước nó. Như vậy thời gian đọc và viết dữ liệu ở những vị trí khác nhau thì khác nhau. Những ví dụ của bộ nhớ này là băng từ, đĩa từ. Tốc độ làm việc của loại bộ nhớ này thường chậm so với bộ nhớ truy xuất ngẫu nhiên.

- Bộ nhớ đọc/viết (Read/Write Memory, RWM): Bộ nhớ có thể viết vào và đọc ra.

- Bộ nhớ chỉ đọc (Read-Only Memory, ROM): Là bộ nhớ mà tỉ lệ tác vụ đọc trên tác vụ ghi rất lớn. Về mặt kỹ thuật, một ROM có thể được ghi chỉ một lần ở nơi sản xuất và sau đó thông tin chỉ có thể được đọc ra từ bộ nhớ. Có loại ROM có thể được ghi nhiều lần nhưng tác vụ ghi phức tạp hơn là tác vụ đọc. ROM thuộc loại bộ nhớ vĩnh cửu và dữ liệu được lưu giữ khi đã cắt nguồn điện.

- Bộ nhớ tĩnh (Static Memory Devices): là bộ nhớ bán dẫn trong đó dữ liệu đã lưu trữ được duy trì cho đến khi nào còn nguồn nuôi.

- Bộ nhớ động (Dynamic Memory Devices): là bộ nhớ bán dẫn trong đó dữ liệu đã lưu trữ muốn tồn tại phải được ghi lại theo chu kỳ. Tác vụ ghi lại được gọi là làm tươi (refresh).

- Bộ nhớ trong (Internal Memory): Chỉ bộ nhớ chính của máy tính. Nó lưu trữ các lệnh và dữ liệu mà CPU dùng thường xuyên khi hoạt động.

- Bộ nhớ khối (Mass Memory): Còn gọi là bộ nhớ phụ, nó chứa một lượng thông tin rất lớn ở bên ngoài máy tính. Tốc độ truy xuất trên bộ nhớ này thường chậm và nó thuộc loại vĩnh cữu.

1.1.2. Các tác vụ và các nhóm chân của một IC nhớ

Mặc dù mỗi loại bộ nhớ có hoạt động bên trong khác nhau, nhưng chúng có chung một số nguyên tắc vận hành mà chúng ta có thể tìm hiểu sơ lược trước khi đi vào nghiên cứu từng loại bộ nhớ. Mỗi hệ thống nhớ luôn có một số yêu cầu ở các đầu vào và ra để hoàn thành một số tác vụ:

- Chọn địa chỉ trong bộ nhớ để truy xuất (đọc hoặc viết)
- Chọn tác vụ đọc hoặc viết để thực hiện
- Cung cấp dữ liệu để lưu vào bộ nhớ trong tác vụ viết
- Gửi dữ liệu ra từ bộ nhớ trong tác vụ đọc
- Cho phép (Enable) (hay Không, Disable) bộ nhớ đáp ứng (hay không) đối với lệnh đọc/ghi ở địa chỉ đã gọi đến.

Từ các tác vụ kể trên, có thể hình dung mỗi IC nhớ có một số đầu vào ra như sau:

- Đầu vào địa chỉ : mỗi vị trí nhớ xác định bởi một địa chỉ duy nhất, khi cần đọc dữ liệu ra hoặc ghi dữ liệu vào ta phải tác động vào chân địa chỉ của vị trí nhớ đó. Một IC có n chân địa chỉ sẽ có 2^n vị trí nhớ. Ký hiệu các chân địa chỉ là A_0 đến A_{n-1} Một IC có 10 chân địa chỉ sẽ có 1024 (1K) vị trí nhớ.

- Đầu vào/ra dữ liệu: Các chân dữ liệu là các đầu vào/ra, nghĩa là dữ liệu luôn được xử lý theo hai chiều. Thường thì dữ liệu vào/ra chung trên một chân nên các đầu này thuộc loại đầu ra 3 trạng thái. Số chân địa chỉ và dữ liệu của một IC xác định dung lượng nhớ của IC đó. Ví dụ một IC nhớ có 10 chân địa chỉ và 8 chân dữ liệu thì dung lượng nhớ của IC đó là 1Kx8 (8K bit hoặc 1K Byte).

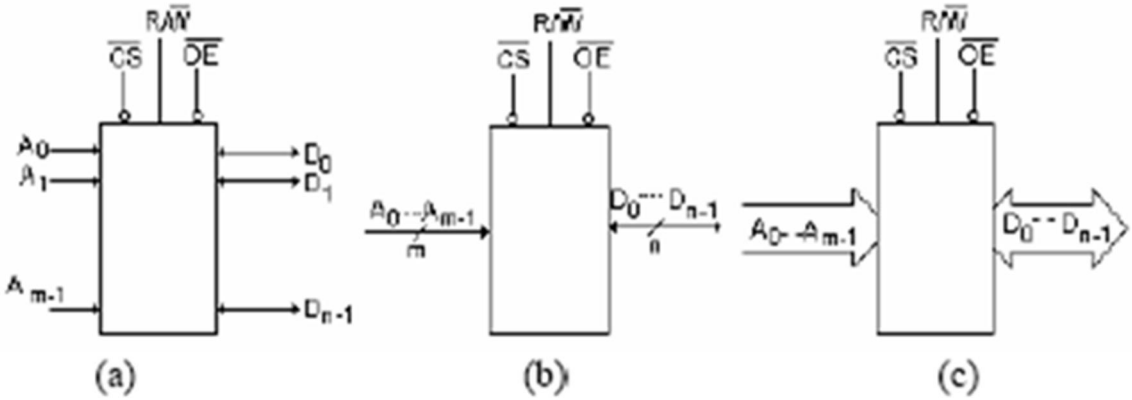
- Các đầu vào điều khiển: Mỗi khi IC nhớ được chọn hoặc có yêu cầu xuất nhập dữ liệu các chân tương ứng sẽ được tác động. Ta có thể kể ra một số đầu vào điều khiển:

* CS: Chip select - Chọn chip - Khi chân này xuống thấp IC được chọn
* CE: Chip Enable - Cho phép chip - Chức năng như chân CS
* OE: Output Enable - Cho phép xuất - Dùng khi đọc dữ liệu
* WR/: Read/Write - Đọc/Viết - Cho phép Đọc dữ liệu ra khi ở mức cao và Ghi dữ liệu vào khi ở mức thấp

* CAS: Column Address Strobe - Chốt địa chỉ cột

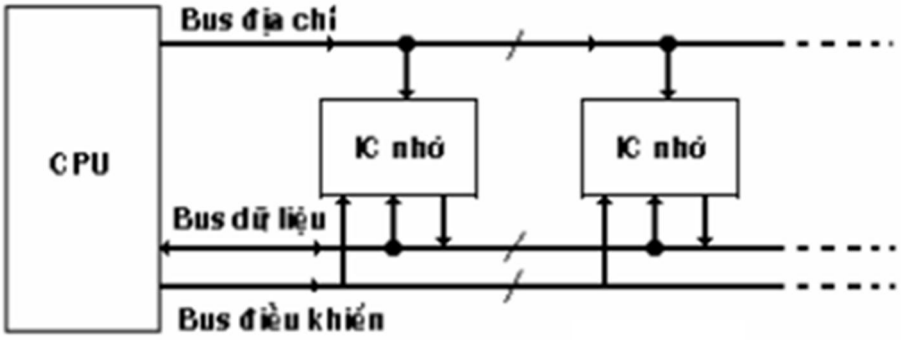
*RAS: Row Address Strobe - Chốt địa chỉ hàng.

Trong trường hợp chip nhớ có dung lượng lớn, để giảm kích thước của mạch giải mã địa chỉ bên trong IC, người ta chia số chân ra làm 2: địa chỉ hàng và địa chỉ cột. Như vậy phải dùng 2 mạch giải mã địa chỉ nhưng mỗi mạch nhỏ hơn rất nhiều. Ví dụ với 10 chân địa chỉ, thay vì dùng 1 mạch giải mã 10 đường sang 1024 đường, người ta dùng 2 mạch giải mã 5 đường sang 32 đường, hai mạch này rất đơn giản so với một mạch kia. Một vị trí nhớ bây giờ có 2 địa chỉ : hàng và cột, dĩ nhiên muốn truy xuất một vị trí nhớ phải có đủ 2 địa chỉ nhờ 2 tín hiệu RAS và CAS. Hình dưới đây cho thấy cách vẽ các nhóm chân của IC nhớ (m chân địa chỉ và n chân dữ liệu). Hình (b) và (c) vẽ các chân địa chỉ và dữ liệu dưới dạng các Bus. Hình (b) được dùng trong các sơ đồ chi tiết và hình (c) được dùng trong các sơ đồ khối.



1.2.3. Giao tiếp giữa IC nhớ và bộ xử lý trung tâm (CPU)

Trong hệ thống mọi hoạt động có liên quan đến IC nhớ đều do bộ xử lý trung tâm (Central Processing Unit, CPU) quản lý. Giao tiếp giữa IC nhớ và CPU được mô tả như sau:



Một tác vụ có liên quan đến bộ nhớ được CPU thực hiện theo các bước:

- Đặt địa chỉ quan hệ lên bus địa chỉ.
- Đặt tín hiệu điều khiển lên bus điều khiển.
- Dữ liệu khả dụng xuất hiện trên bus dữ liệu, sẵn sàng để ghi vào hoặc đọc ra.

1.2. Phân loại

Theo cấu trúc và đặc điểm của bộ nhớ, ta chia ra 3 loại bộ nhớ bán dẫn (khái niệm đã được nhắc tới trong mục 1.1.1):

- Bộ nhớ bán dẫn chỉ đọc (Read Only Memory, ROM)
- Bộ nhớ truy xuất ngẫu nhiên: (Random Access Memory, RAM)

Thật ra ROM và RAM đều là loại bộ nhớ truy xuất ngẫu nhiên, nhưng RAM được giữ tên gọi này. Để phân biệt chính xác ROM và RAM ta có thể gọi ROM là **bộ nhớ chết** (nonvolatile, vĩnh cữu) và RAM là **bộ nhớ sống** (volatile, không vĩnh cữu) hoặc nếu coi ROM là **bộ nhớ chỉ đọc** thì RAM là **bộ nhớ đọc được - viết được** (Read-Write Memory)

- Thiết bị logic lập trình được: (Programmable Logic Devices, PLD) có thể nói điểm khác biệt giữa PLD với ROM và RAM là qui mô tích hợp của PLD thường không lớn như ROM và RAM và các tác vụ của PLD thì có phần hạn chế.

2. Mạch nhớ ROM

Mục tiêu:

Trình bày được khái niệm, cấu trúc một số dạng ROM thường gặp trên thực tế.

2.1. Khái niệm về ROM

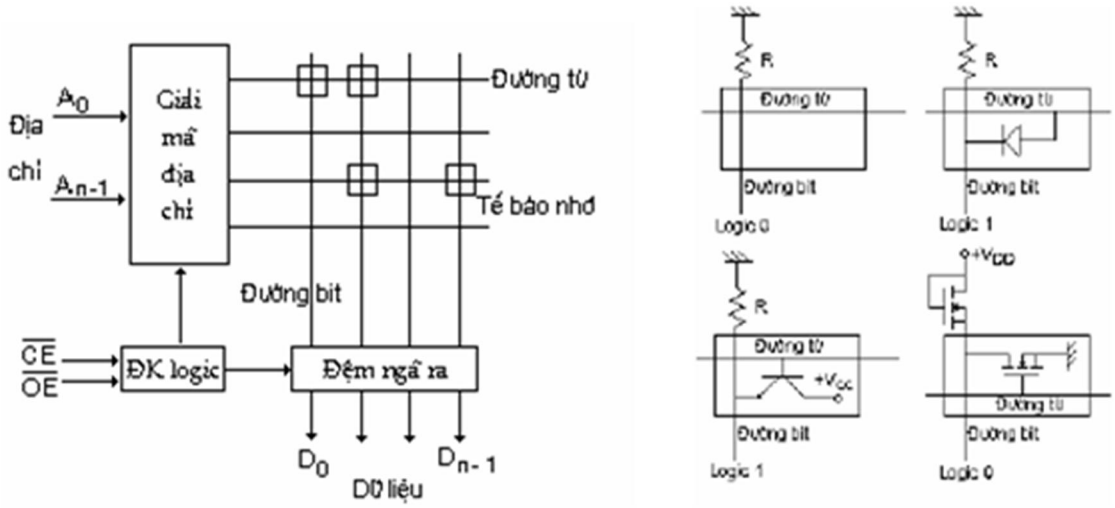
Là tên gọi của bộ nhớ chỉ đọc (Read-Only Memory, ROM). Mặc dù có tên gọi như thế nhưng chúng ta phải hiểu là khi sử dụng ROM, tác vụ đọc được thực hiện rất nhiều lần so với tác vụ ghi. Thậm chí có loại ROM chỉ ghi một lần khi xuất xưởng. Các tế bào nhớ hoặc từ nhớ trong ROM sắp xếp theo dạng ma trận mà mỗi phần tử chiếm một vị trí xác định bởi một địa chỉ cụ thể và nối với đầu ra một mạch giải mã địa chỉ bên trong IC. Nếu mỗi vị trí chứa một tế bào nhớ ta nói ROM có tổ chức bit và mỗi vị trí là một từ nhớ ta có tổ chức từ.

Ngoài ra, để giảm mức độ cồng kềnh của mạch giải mã, mỗi vị trí nhớ có thể được xác định bởi 2 đường địa chỉ : đường địa chỉ hàng và đường địa chỉ cột và trong bộ nhớ có 2 mạch giải mã nhưng mỗi mạch có số đầu vào bằng 1/2 số đường địa chỉ của cả bộ nhớ.

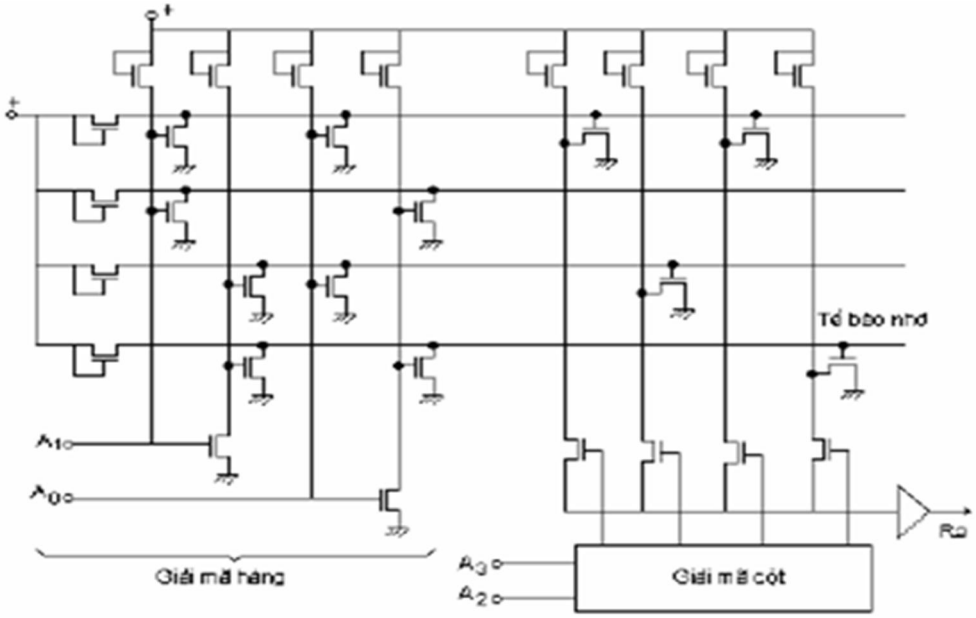
2.2. Cấu trúc chung của các dạng ROM thường gặp

2.2.1. ROM mặt nạ (Mask Programmed ROM, MROM)

Đây là loại ROM được chế tạo để thực hiện một công việc cụ thể như các bảng tính, bảng lượng giác, bảng logarit . . . ngay sau khi xuất xưởng. Nói cách khác, các tế bào nhớ trong ma trận nhớ đã được tạo ra theo một chương trình đã xác định trước bằng phương pháp mặt nạ: đưa vào các linh kiện điện tử nối từ **đường từ** qua **đường bít** để tạo ra một giá trị bit và để trống cho giá trị bit ngược lại. Dưới đây là mô hình của một MROM trong đó các ô vuông là nơi chứa (hay không) một linh kiện (diod, transistor BJT hay MOSFET) để tạo bit. Mỗi đầu ra của mạch giải mã địa chỉ gọi là đường từ và đường nối tế bào nhớ ra ngoài gọi là đường bit. Khi đường từ lên mức cao thì tế bào nhớ hoặc từ nhớ được chọn.



Nếu tế bào nhớ là Diod hoặc BJT thì sự hiện diện của linh kiện tương ứng với bit 1 (lúc này đường từ lên cao, Transsisstor hoặc diod dẫn, dòng điện qua điện trở tạo điện thế cao ở hai đầu điện trở) còn vị trí nhớ trống tương ứng với bit 0. Đối với loại linh kiện MOSFET thì ngược lại, nghĩa là sự hiện diện của linh kiện tương ứng với bit 0 còn vị trí nhớ trống tương ứng với bit 1 (muốn có kết quả như loại BJT thì thêm ở đầu ra các cổng đảo). Một ví dụ bộ nhớ MROM có dung lượng 16x1 với các mạch giải mã hàng và cột (các mạch giải mã 2 đường sang 4 đường của hàng và cột đều dùng Transistor MOS và có cùng cấu trúc).

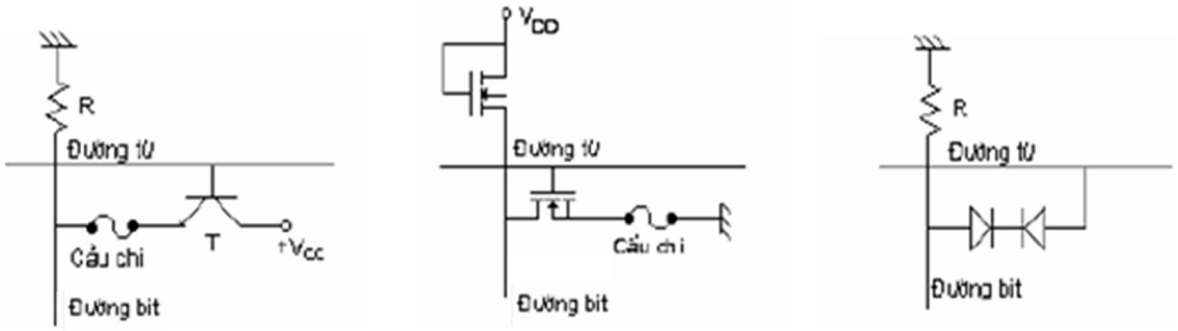


Trong thực tế, để đơn giản cho việc thực hiện, ở mỗi vị trí nhớ người ta đều cho vào một transistor MOS. Nhưng ở những vị trí ứng với bit 1 các transistor MOS được chế tạo với lớp SiO_2 dầy hơn làm tăng điện thế ngưỡng của nó lên, kết quả là transistor MOS này luôn luôn không dẫn điện, các transistor khác dẫn điện bình thường.

2.2.2. ROM lập trình được (Programmable ROM, PROM)

Có cấu tạo giống MROM nhưng ở mỗi vị trí nhớ đều có linh kiện nối với cầu chì. Như vậy khi xuất xưởng các ROM này đều chứa cùng một loại bit (gọi

là ROM trắng), lúc sử dụng người lập trình thay đổi các bit mong muốn bằng cách phá vỡ cầu chì ở các vị trí tương ứng với bit đó. Một khi cầu chì đã bị phá vỡ thì không thể nối lại được do đó loại ROM này cho phép lập trình một lần duy nhất để sử dụng, nếu bị lỗi không thể sửa chữa được.

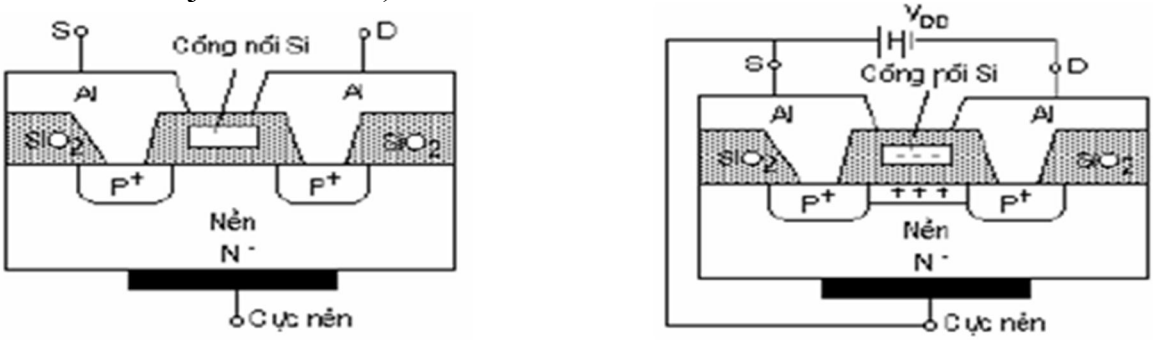


ROM lập trình được - PROM

Người ta có thể dùng 2 diod mắc ngược chiều nhau, mạch không dẫn điện, để tạo bit 0, khi lập trình thì một diod bị phá hỏng tạo mạch nối tắt, diod còn lại dẫn điện cho bit 1

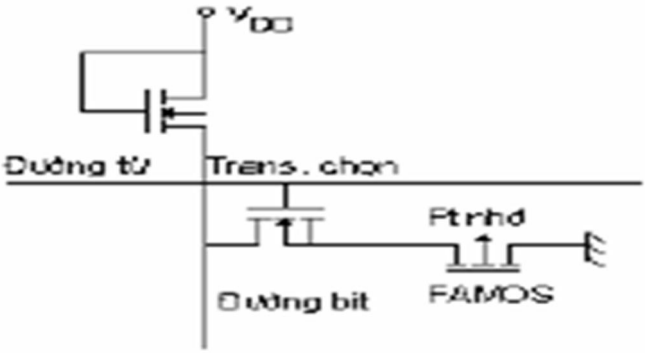
2.2.3. ROM lập trình được, xóa được bằng tia U.V. (Ultra Violet Erasable Programmable ROM, U.V. EPROM)

Đây là loại ROM rất tiện cho người sử dụng vì có thể dùng được nhiều lần bằng cách xóa và nạp lại. Cấu tạo của tế bào nhớ của U.V. EPROM dựa vào một transistor MOS có cấu tạo đặc biệt gọi là FAMOS (Floating Gate Avalanche Injection MOS)



Trên nền chất bán dẫn N pha tạp ít, tạo 2 vùng P pha tạp nhiều (P^+) nối ra ngoài cho 2 cực S (Source) và D (Drain). Trong lớp cách điện SiO_2 giữa 2 cực người ta cho vào một thỏi Silicon không nối với bên ngoài và được gọi là **cổng nổi**. Khi nguồn V_{DD}, phân cực ngược giữa cực nền và Drain còn nhỏ, transistor không dẫn, nhưng nếu tăng V_{DD} đủ lớn, hiện tượng thác lũ điện tử (avalanche) xảy ra, electron đủ năng lượng chui qua lớp cách điện tới bám vào cổng nổi. Do hiện tượng cảm ứng, một kênh dẫn P hình thành nối hai vùng bán dẫn P^+, transistor trở nên dẫn điện. Khi cắt nguồn, transistor tiếp tục dẫn điện vì electron không thể trở về để tái hợp với lỗ trống.

Để xóa EPROM, người ta chiếu tia U.V. vào các tế bào trong một khoảng thời gian xác định để electron trên cổng nổi nhận đủ năng lượng vượt qua lớp cách điện trở về vùng nền tái hợp với lỗ trống xóa kênh dẫn P và transistor trở về trạng thái không dẫn ban đầu.

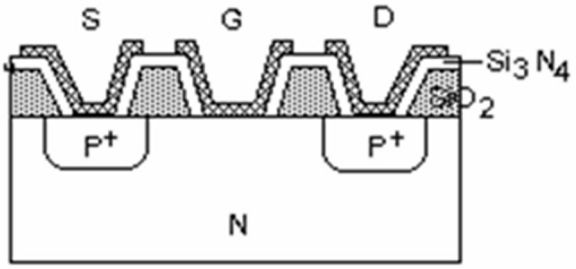


Mỗi tế bào nhớ EPROM gồm một transistor FAMOS nối tiếp với một transistor MOS khác mà ta gọi là transistor chọn, như vậy vai trò của FAMOS giống như là một cầu chì nhưng có thể phục hồi được.

Điểm bất tiện của U.V EPROM là cần thiết bị xóa đặc biệt phát tia U.V. và mỗi lần xóa tất cả tế bào nhớ trong một IC nhớ đều bị xóa. Như vậy người sử dụng phải nạp lại toàn bộ chương trình

2.2.4. ROM lập trình được và xóa được bằng xung điện (Electrically Erasable PROM, EEPROM hay Electrically Alterable PROM, EAPROM)

Đây là loại ROM lập trình được và xóa được nhờ xung điện và đặc biệt là có thể xóa để sửa trên từng byte. Các tế bào nhớ EEPROM sử dụng transistor MNOS (Metal Nitride Oxide Semiconductor) có cấu tạo như sau:



Giữa lớp kim loại nối ra các cực và lớp SiO_2 là một lớp mỏng chất Nitrua Silic (Si_3N_4) - từ 40nm đến 650nm - Dữ liệu được nạp bằng cách áp một điện thế dương giữa cực G và S (khoảng 20 đến 25V trong 100ms). Do sự khác biệt về độ dẫn điện, electron tích trên bề mặt giữa 2 lớp SiO_2 và Si_3N_4, các electron này tồn tại khi đã ngắt nguồn và làm thay đổi trạng thái dẫn điện của transistor. Bây giờ nếu áp một điện thế âm giữa cực G và S ta sẽ được một lớp điện tích trái dấu với trường hợp trước. Như vậy hai trạng thái khác nhau của Transistor có thể thiết lập được bởi hai điện thế ngược chiều nhau và như vậy các tế bào nhớ được ghi và xóa với 2 xung điện trái dấu nhau.

2.2.5. FLASH ROM

EPROM là loại nonvolatile, có tốc độ truy xuất nhanh (khoảng 120ns), mật độ tích hợp cao, giá thành rẻ tuy nhiên để xóa và nạp lại phải dùng thiết bị đặc biệt và lấy ra khỏi mạch. EEPROM cũng nonvolatile, cũng có tốc độ truy xuất nhanh, cho phép xóa và nạp lại ngay trong mạch trên từng byte nhưng có mật độ tích hợp thấp và giá thành cao hơn EPROM. Bộ nhớ FLASH ROM tận dụng được các ưu điểm của hai loại ROM nói trên, nghĩa là có tốc độ truy xuất nhanh, có mật độ tích hợp cao nhưng giá thành thấp.

Hầu hết các FLASH ROM sử dụng cách xóa đồng thời cả khối dữ liệu nhưng rất nhanh (hàng trăm ms so với 20 min của U.V. EPROM). Những FLASH ROM thế hệ mới cho phép xóa từng sector (512 byte) thậm chí từng vị trí nhớ mà không cần lấy IC ra khỏi mạch. FLASH ROM có thời gian ghi khoảng 10μs/byte so với 100 μs đối với EPROM và 5 ms đối với EEPROM.

3. Lắp ráp và cân chỉnh bộ nhớ ROM

Mục tiêu:

3.1. Chuẩn bị vật tư và dụng cụ
3.2. Lắp mạch
3.3. Thực hành đo và cân chỉnh.

BÀI 10
MẠCH NHỚ RAM
Mã bài: MĐ26.10

Giới thiệu:

Ở chương trước, chúng ta đã thấy một nhóm các FF họp thành thanh ghi để lưu trữ và dịch chuyển thông tin. Các FF chính là các phần tử nhớ tốc độ cao được dùng rất nhiều trong việc điều hành bên trong máy tính, nơi mà dữ liệu dịch chuyển liên tục từ nơi này đến nơi khác.

Bộ nhớ bán dẫn được dùng như là **bộ nhớ trong** chính của máy tính, nơi mà việc vận hành nhanh được xem như ưu tiên hàng đầu và cũng là nơi mà tất cả dữ liệu của chương trình lưu chuyển liên tục trong quá trình thực hiện một tác vụ do CPU yêu cầu.

Mặc dù bộ nhớ bán dẫn có tốc độ làm việc cao, rất phù hợp cho bộ nhớ trong, nhưng giá thành tính trên mỗi bit lưu trữ cao khiến cho nó không thể là loại thiết bị có tính chất lưu trữ khối, là loại thiết bị có khả năng lưu trữ hàng tỉ bit mà không cần cung cấp năng lượng và được dùng như là **bộ nhớ ngoài** (đĩa từ, băng từ, CD ROM . . .). Do các nhu cầu khác nhau mà các bộ nhớ cũng sẽ có yêu cầu khác nhau, có thể là sử dụng ROM, nhưng cũng có thể chỉ phải sử dụng RAM. Phần trên ta đã nghiên cứu cơ bản về ROM, bài học này nghiên cứu cấu tạo và tổ chức chung của RAM.

Mục tiêu:

- Trình bày được đặc tính, cấu trúc, hoạt động và ứng dụng của mạch nhớ RAM
- Lắp ráp được bộ nhớ RAM.
- Có tính tư duy, tác phong trong công nghiệp

Nội dung chính:

1. Khái niệm về RAM

Mục tiêu:

Trình bày được khái niệm và cách phân loại RAM thông dụng.

1.1. Khái niệm

Bộ nhớ truy xuất ngẫu nhiên (Random-Access Memory, RAM): Là bộ nhớ mà khi cần truy xuất một địa chỉ ta tới ngay địa chỉ đó. Vậy thời gian đọc hay viết dữ liệu vào các vị trí nhớ khác nhau trong bộ nhớ không tùy thuộc vào vị trí nhớ. Nói cách khác, thời gian truy xuất như nhau đối với mọi vị trí nhớ. Hầu hết bộ nhớ bán dẫn và nhẫn từ (bộ nhớ trong của máy tính trước khi bộ nhớ bán dẫn ra đời) là loại truy xuất ngẫu nhiên.

1.2. Phân loại

Theo tính chất hoạt động của RAM mà người ta chia thành hai loại RAM: RAM tĩnh và RAM động

- RAM tĩnh cấu tạo bởi các tế bào nhớ là các FF,

- RAM động lợi dụng các điện dung ký sinh giữa các cực của transistor MOS, trạng thái tích điện hay không của tụ tương ứng với hai bit 1 và 0. Do RAM động có mật độ tích hợp cao, dung lượng bộ nhớ thường rất lớn nên để định vị các phần tử nhớ người ta dùng phương pháp đa hợp địa chỉ, mỗi từ nhớ

được chọn khi có đủ hai địa chỉ hàng và cột được lần lượt tác động. Phương pháp này cho phép n đường địa chỉ truy xuất được 2^{2n} vị trí nhớ.

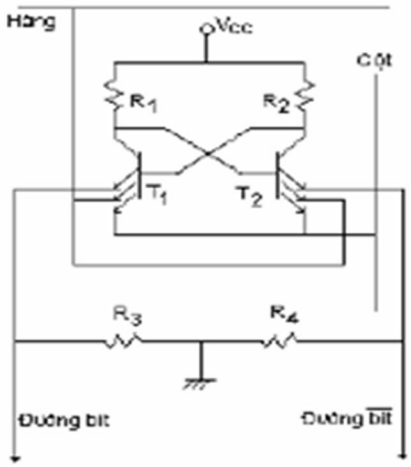
2. Cấu trúc chung của RAM

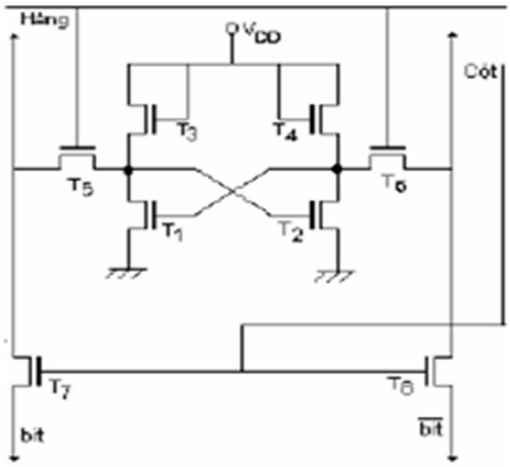
Mục tiêu:

Trình bày được cấu trúc chung của các dạng RAM thông dụng.

2.1. Cấu trúc của RAM tĩnh (Static RAM, SRAM)

Mỗi tế bào RAM tĩnh là một mạch FlipFlop dùng Transistor BJT hay MOS



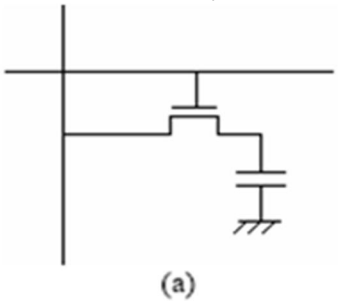


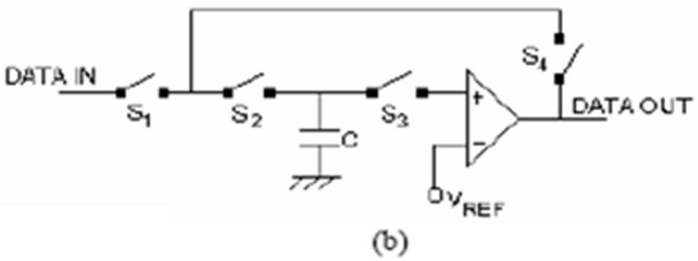
Hình trên là một tế bào nhớ RAM tĩnh dùng transistor BJT với 2 đường địa chỉ hàng và cột. Khi một trong hai đường địa chỉ hàng hoặc cột ở mức thấp các tế bào không được chọn vì cực E có điện thế thấp hai Transistor đều dẫn, mạch không hoạt động như một FF. Khi cả hai địa chỉ hàng và cột lên cao, mạch hoạt động như FF, hai trạng thái 1 và 0 của tế bào nhớ được đặc trưng bởi hai trạng thái khác nhau của 2 đường bit và $\overline{\text{bit}}$.

Giả sử khi T_1 dẫn thì T_2 tắt, đường bit có dòng điện chạy qua, tạo điện thế cao ở R3 trong khi đó đường bit không có dòng chạy qua nên ở R4 có điện thế thấp. Nếu ta qui ước trạng thái này tương ứng với bit 1 thì trạng thái ngược lại, là trạng thái T_1 tắt và T_2 dẫn, hiệu thế ở điện trở R3 thấp và ở R4 cao, sẽ là bit 0. R3 và R4 có tác dụng biến đổi dòng điện ra điện thế. Đối với tế bào nhớ dùng MOS, hai đường từ nối với T5, T6 và T7, T8 nên khi một trong hai đường từ ở mức thấp T1 và T2 bị cô lập khỏi mạch, tế bào nhớ không được chọn. Khi cả hai lên cao mạch hoạt động tương tự như trên. Trong mạch này R1 và R2 thay bởi T3 và T4 và không cần R3 và R4 như mạch dùng BJT.

2.2. RAM động (Dynamic RAM, DRAM)

Hình sau là một tế bào nhớ của DRAM





Hình (b) là một cách biểu diễn tế bào nhớ DRAM trong đó đơn giản một số chi tiết được dùng để mô tả các tác vụ viết và đọc tế bào nhớ này. Các khóa từ S_1 đến S_4 là các transistor MOS được điều khiển bởi các tín hiệu ra từ mạch giải mã địa chỉ và tín hiệu $R/\overline{W}$.

Để ghi dữ liệu vào tế bào, các khóa S_1 và S_2 đóng trong khi S_3 và S_4 mở. Bit 1 thực hiện việc nạp điện cho tụ C và bit 0 làm tụ C phóng điện. Sau đó các khóa sẽ mở để cô lập C với phần mạch còn lại. Một cách lý tưởng thì C sẽ duy trì trạng thái của nó vĩnh viễn nhưng thực tế luôn luôn có sự rỉ điện qua các khóa ngay cả khi chúng mở do đó C bị mất dần điện tích .

Để đọc dữ liệu các khóa S_2 , S_3 , S_4 đóng và S_1 mở, tụ C nối với một mạch so sánh với một điện thế tham chiếu để xác định trạng thái logic của nó. Điện thế ra mạch so sánh chính là dữ liệu được đọc ra. Do S_2 và S_4 đóng, dữ liệu ra được nối ngược lại tụ C để làm tươi nó. Nói cách khác, bit dữ liệu trong tế bào nhớ được làm tươi mỗi khi nó được đọc.

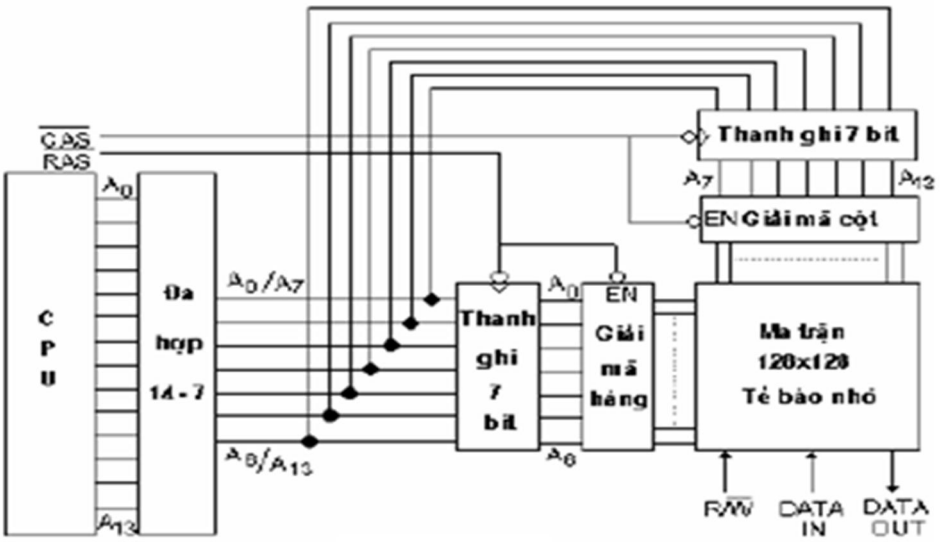
Sử dụng DRAM, được một thuận lợi là dung lượng nhớ khá lớn nhưng phải có một số mạch phụ trợ:

- Mạch đa hợp địa chỉ vì DRAM luôn sử dụng địa chỉ hàng và cột.
- Mạch làm tươi để phục hồi dữ liệu có thể bị mất sau một khoảng thời gian ngắn nào đó.

2.2.1. Đa hợp địa chỉ

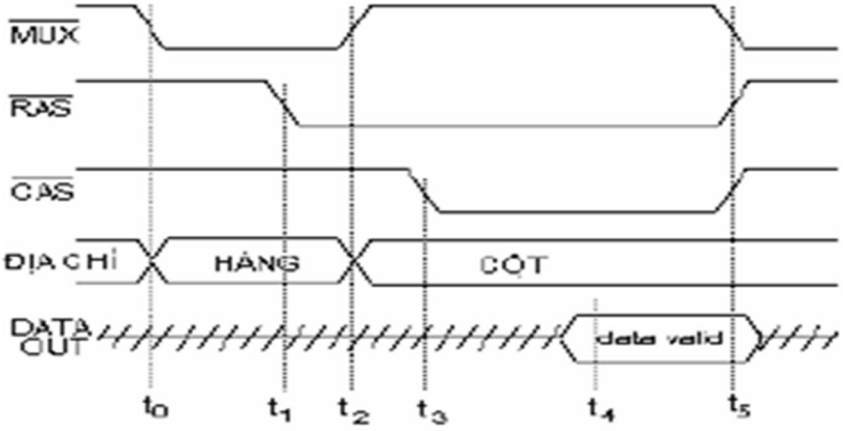
Như đã nói trên, do dung lượng của DRAM rất lớn nên phải dùng phương pháp đa hợp để chọn một vị trí nhớ trong DRAM. Mỗi vị trí nhớ sẽ được chọn bởi 2 địa chỉ hàng và cột lần lượt xuất hiện ở đầu vào địa chỉ.

Ví dụ với DRAM có dung lượng 16Kx1, thay vì phải dùng 14 đường địa chỉ ta chỉ cần dùng 7 đường và mạch đa hợp 14 → 7 (7 đa hợp 2→1) để chọn 7 trong 14 đường địa chỉ ra từ CPU (H 7.21). Bộ nhớ có cấu trúc là một ma trận 128x128 tế bào nhớ, sắp xếp thành 128 hàng và 128 cột, có một đầu vào và một đầu ra dữ liệu, một đầu vào $R/\overline{W}$. Hai mạch chốt địa chỉ (hàng và cột) là các thanh ghi 7 bit có đầu vào nối với đầu ra mạch đa hợp và đầu ra nối với các mạch giải mã hàng và cột. Các tín hiệu $\overline{RAS}$ và $\overline{CAS}$ dùng làm xung đồng hồ cho mạch chốt và tín hiệu Enable cho mạch giải mã. Như vậy 14 bit địa chỉ từ CPU sẽ lần lượt được chốt vào các thanh ghi hàng và cột bởi các tín hiệu $\overline{RAS}$ và $\overline{CAS}$ rồi được giải mã để chọn tế bào nhớ. Vận hành của hệ thống sẽ được thấy rõ hơn khi xét các giản đồ thời gian của DRAM.



2.2.2. Giản đồ thời gian của DRAM

Hình dưới là giản đồ thời gian đọc và viết tiêu biểu của DRAM (Hai giản đồ này chỉ khác nhau về thời lượng nhưng có chung một dạng nên ta chỉ vẽ một)



Giản đồ cho thấy tác động của tín hiệu $\overline{MUX}$ và các tín hiệu $\overline{RAS}$ và $\overline{CAS}$. Khi $\overline{MUX}$ ở mức thấp mạch đa hợp cho ra địa chỉ hàng ($A_0 \ldots A_6$) và được chốt vào thanh ghi khi tín hiệu $\overline{RAS}$ xuống thấp. Khi $\overline{MUX}$ ở mức cao mạch đa hợp cho ra địa chỉ cột ($A_7 \ldots A_{13}$) và được chốt vào thanh ghi khi tín hiệu $\overline{CAS}$ xuống thấp. Khi cả địa chỉ hàng và cột đã được giải mã, dữ liệu tại địa chỉ đó xuất hiện trên bus dữ liệu để đọc ra hoặc ghi vào (khả dụng)

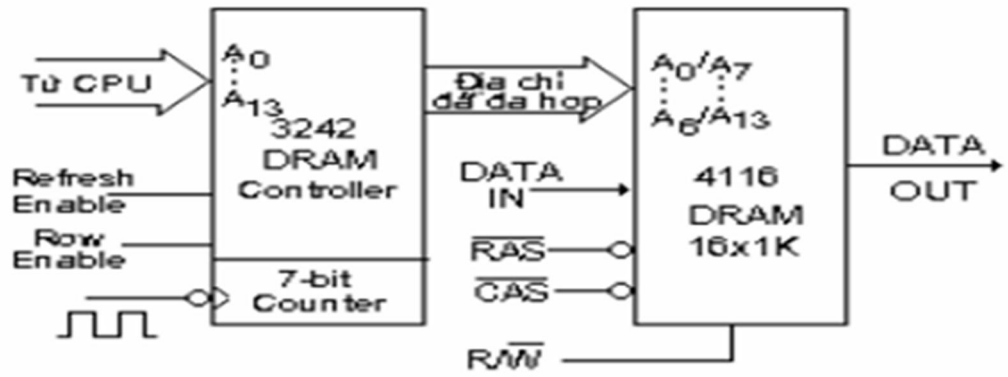
2.2.3. Làm tươi DRAM

DRAM phải được làm tươi với chu kỳ khoảng 2ms để duy trì dữ liệu. Trong phần trước ta đã thấy tế bào nhớ DRAM được làm tươi ngay khi tác vụ đọc được thực hiện. Lấy ví dụ với DRAM có dung lượng 16Kx1 (16.384 tế bào) nói trên, chu kỳ làm tươi là 2 ms cho 16.384 tế bào nhớ nên thời gian đọc mỗi tế bào nhớ phải là 2 ms/16.384 = 122 ns. Đây là thời gian rất nhỏ không đủ để đọc một tế bào nhớ trong điều kiện vận hành bình thường. Vì lý do này các hãng chế tạo đã thiết kế các chip DRAM sao cho ***mỗi khi tác vụ đọc được thực hiện đối với một tế bào nhớ, tất cả các tế bào nhớ trên cùng một hàng sẽ được làm tươi***. Điều này làm giảm một lượng rất lớn tác vụ đọc phải thực hiện để làm tươi tế bào nhớ. Trở lại ví dụ trên, tác vụ đọc để làm tươi phải thực hiện cho 128 hàng trong 2 ms. Tuy nhiên để vừa vận hành trong điều kiện bình thường vừa

phải thực hiện chức năng làm tươi người ta phải dùng thêm mạch phụ trợ, gọi là điều khiển DRAM (DRAM controller). IC 3242 của hảng Intel thiết kế để sử dụng cho DRAM 16K.

Đầu ra 3242 là địa chỉ 7 bit đã được đa hợp và nối vào đầu vào địa chỉ của DRAM. Một mạch đếm 7 bit kích bởi xung đồng hồ riêng để cấp địa chỉ hàng cho DRAM trong suốt thời gian làm tươi. 3242 cũng lấy địa chỉ 14 bit từ CPU đa hợp nó với địa chỉ hàng và cột đã được dùng khi CPU thực hiện tác vụ đọc hay viết. Mức logic áp dụng cho các đầu REFRESH ENABLE và ROW ENABLE xác định 7 bit nào của địa chỉ xuất hiện ở đầu ra mạch controller cho bởi bảng

REFRESH ENABLE	ROW ENABLE	CONTROLLER OUTPUT
HIGH	X	Refresh address (Từ mạch đếm)
LOW	HIGH	Địa chỉ hàng (A_0 ... A_6 từ CPU)
LOW	LOW	Địa chỉ cột (A_7 ... A_{13} từ CPU)



3. Lắp ráp và cân chỉnh bộ nhớ RAM

Mục tiêu:

3.1. Chuẩn bị vật tư và dụng cụ

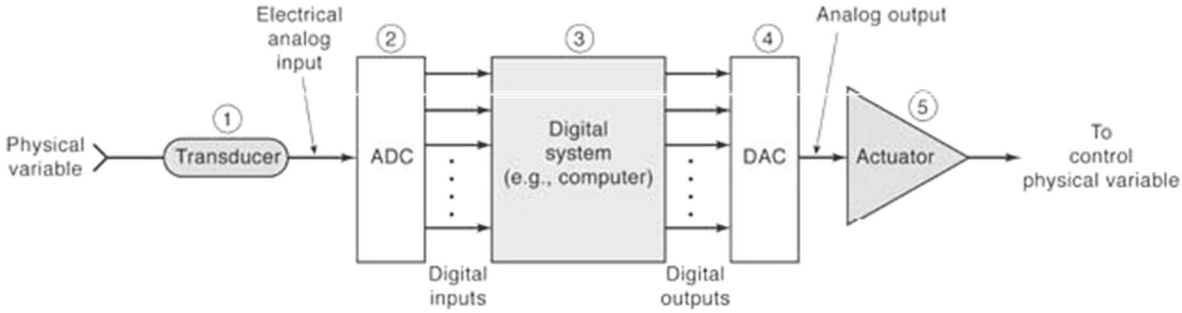
3.2. Lắp mạch

3.3. Thực hành đo và cân chỉnh.

BÀI 11
MẠCH CHUYỂN ĐỔI A/D
Mã bài: MĐ26.11

Giới thiệu:

Có thể nói sự biến đổi qua lại giữa các tín hiệu từ dạng tương tự sang dạng số là cần thiết trong các quá trình thực hiện giao tiếp giữa con người, tự nhiên với máy trong thực tế lao động sản xuất và nghiên cứu hàng ngày, có thể biểu diễn một cách tương đối theo sơ đồ sau:



Hệ thống số chỉ xử lý các tín hiệu số, mà tín hiệu trong tự nhiên là tín hiệu tương tự: Nên khâu đầu tiên trong quá trình kết nối tương tự số chính là mạch chuyển đổi tín hiệu từ tương tự sang số (A/DC).

Mục tiêu:

- Trình bày được khái niệm, nguyên lý cơ bản của mạch chuyển đổi A/D.
- Lắp ráp, sửa chữa được mạch chuyển đổi A/D.
- Có tính tư duy, tác phong trong công nghiệp

Nội dung chính:

1. Khái niệm chung

Mục tiêu:

Trình bày được khái niệm và giải thích được sơ đồ khối của mạch chuyển đổi A/D.

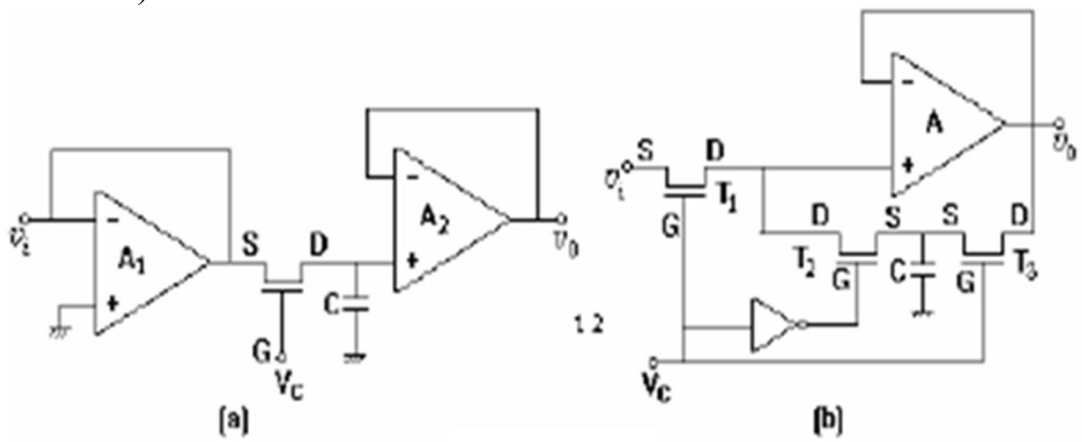
1.1. Khái niêm chung

Mạch chuyển đổi A/D là mạch dùng để chuyển đổi các tín hiệu từ dạng tương tự (đầu ra của các thiết bị tương tự) thành dạng tín hiệu số, thuận tiện cho việc xử lý của các mạch số hay hệ thống số.

Để biến đổi một tín hiệu tương tự sang tín hiệu số, người ta không thể biến đổi mọi giá trị của tín hiệu tương tự mà chỉ có thể biến đổi một số gía trị cụ thể bằng cách **lấy mẫu** tín hiệu đó theo một chu kỳ xác định nhờ một tín hiệu có dạng xung. Ngoài ra, mạch biến đổi cần một khoảng thời gian cụ thể (khoảng 1µs - 1ms) do đó **cần giữ mức tín hiệu** biến đổi trong khoảng thời gian này để mạch có thể thực hiện việc biến đổi chính xác. Đó là nhiệm vụ của mạch lấy mẫu và giữ.

Hình dưới đây là một dạng mạch lấy mẫu và giữ cơ bản: Điện thế tương tự cần biến đổi được lấy mẫu trong thời gian rất ngắn do tụ nạp điện nhanh qua

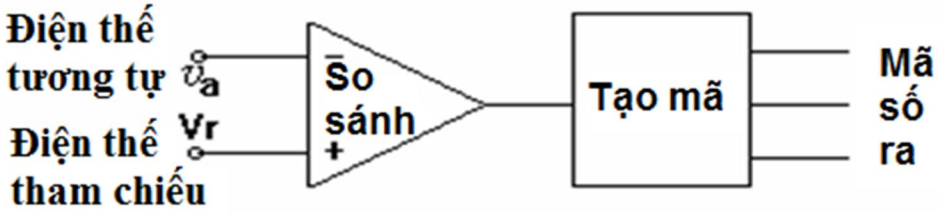
tổng trở ra thấp của OP-AMP khi các transistor dẫn và giữ giá trị này trong khoảng thời gian transistor dừng (tụ phóng rất chậm qua tổng trở vào rất lớn của OP-AMP)



1.2. Sơ đồ khối

Mạch biến đổi ADC gồm bộ phận trung tâm là một mạch so sánh. Điện thế tương tự chưa biết v_a áp vào một đầu vào của mạch so sánh, còn đầu vào kia nối đến một điện thế tham chiếu thay đổi theo thời gian Vr(t). Khi chuyển đổi điện thế tham chiếu tăng theo thời gian cho đến khi bằng hoặc gần bằng với điện thế tương tự (với một sai số nguyên lượng hóa). Lúc đó mạch tạo mã số ra có giá trị ứng với điện thế vào chưa biết. Vậy nhiệm vụ của mạch tạo mã số là thử một bộ số nhị phân sao cho hiệu số giữa v_a và trị nguyên lượng hóa sau cùng nhỏ hơn 1/2 LSB

$$\Rightarrow v_a - (V_{FS} / 2^n - 1)(B)_2 | < 1/2 \text{ LSB}$$



2. Các tham số cơ bản

Mục tiêu:

Trình bày được các tham số cơ bản đối với một mạch chuyển đổi A/D.

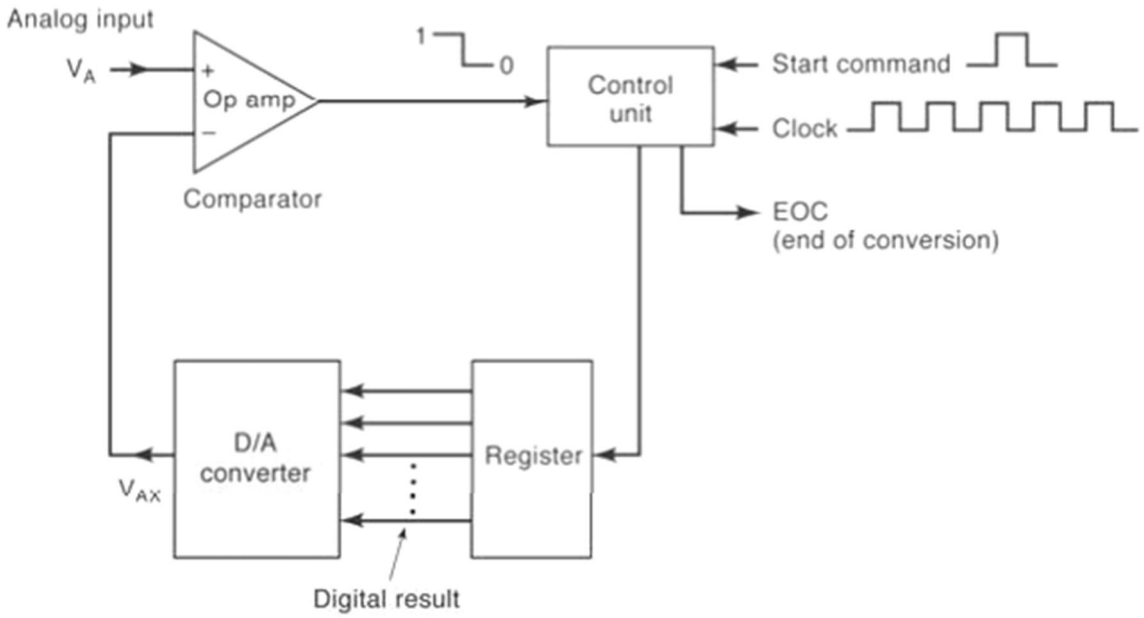
ADC – miêu tả giá trị analog đầu vào bằng giá trị số nhị phân.
ADC phức tạp và tốn nhiều thời gian biến đổi hơn DAC
Một số ADC sử dụng bộ DAC là một phần của nó
Một opamp được sử dụng làm bộ so sánh trong ADC

3. Nguyên lý hoạt động

Mục tiêu:

Vẽ sơ đồ khối tổng quát và trình bày được nguyên lý hoạt động của mạch ADC.

Về cơ bản sơ đồ khối của mạch ADC như sau:



Trong đó: - Op-amp giữ vai trò mạch so sánh.

- D/A converter cùng với thanh ghi (register) tạo ra điện thế tham chiếu cho khối so sánh.

Nguyên lý hoạt động của sơ đồ khối trên như sau:

- Lệnh START bắt đầu quá trình biến đổi
- Control unit thay đổi giá trị nhị phân trong thanh ghi
- Giá trị nhị phân trong thanh ghi được biến đổi thành giá trị nhị phân V_{AX}
- Bộ so sánh so sánh V_{AX} với V_A. Khi $V_{AX} < V_A$, đầu ra bộ so sánh ở mức cao. Khi $V_{AX} > V_A$, đầu ra có mức thấp, quá trình biến đổi kết thúc, giá trị nhị phân nằm trong thanh ghi. Bộ phận điều khiển sẽ phát ra tín hiệu end-of-conversion signal - EOC.

4. Các phương pháp chuyển đổi

Mục tiêu:

Trình bày được sơ đồ khối và nguyên lý hoạt động của một số phương pháp chuyển đổi A/D.

4.1. Chuyển đổi A/D theo phương pháp song song (parallel hay flash conversion)

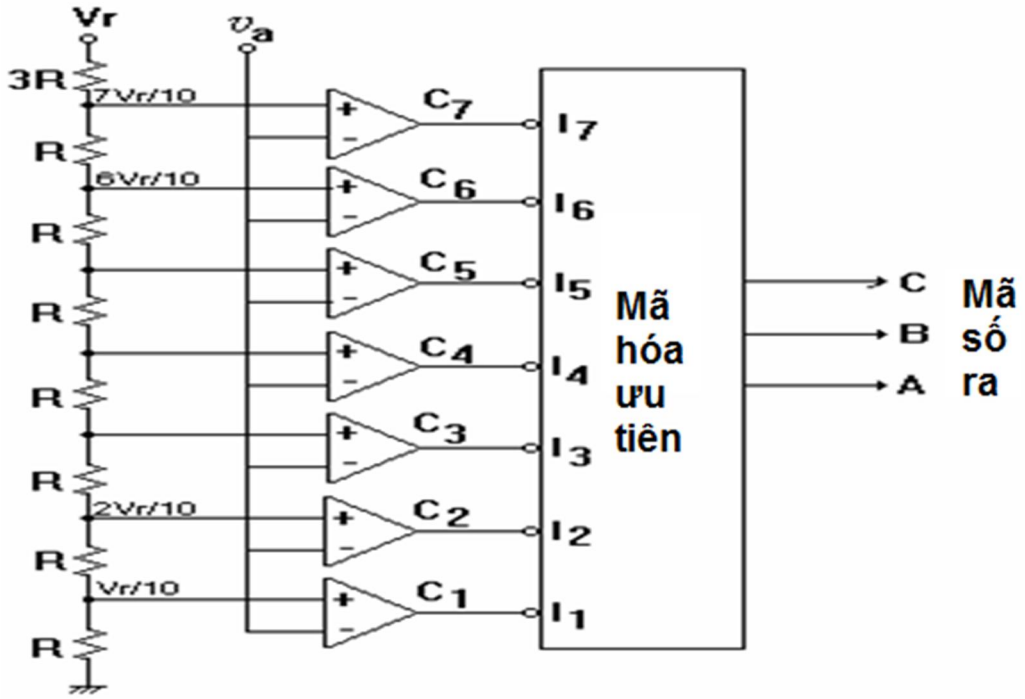
Đây là mạch đổi có tốc độ chuyển đổi rất nhanh, có thể đạt vài triệu lần trong một giây, áp dụng vào việc chuyển đổi tín hiệu hình trong kỹ thuật video. Ví dụ để có mạch đổi 3 bit, người ta dùng 7 mạch so sánh ở đầu vào và một mạch mã hóa ưu tiên để tạo mã số nhị phân ở đầu ra:

- Khi v_a < Vr /10, các đầu ra mạch so sánh đều lên cao khiến mã số ra là 000

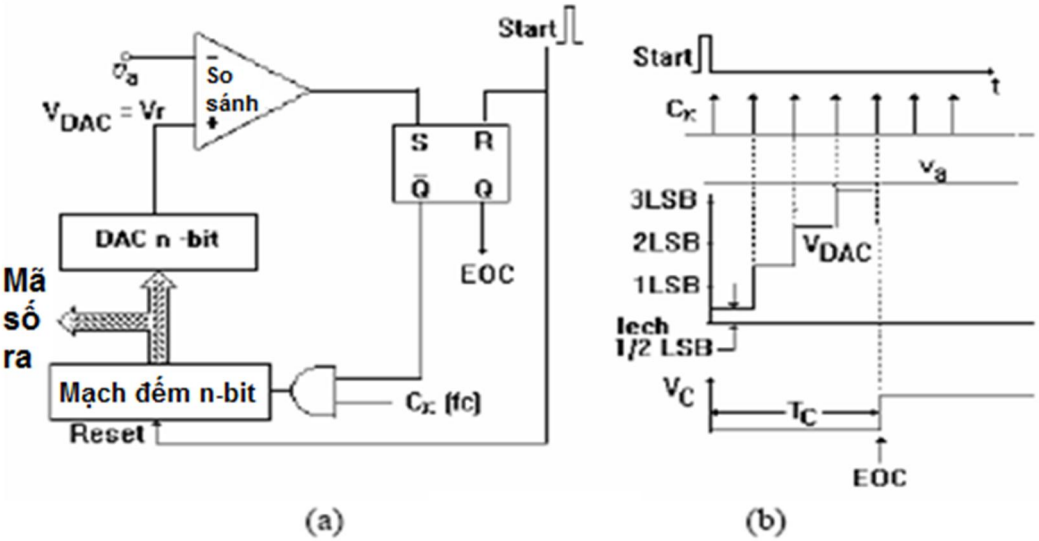
- Khi Vr /10 <v_a < 2Vr /10, đầu ra mạch so sánh 1 xuống thấp khiến mã số ra là 001

- Khi 2Vr /10 <v_a < 3Vr /10, đầu ra mạch so sánh 2 xuống thấp khiến mã số ra là 010

Cứ như thế, ta thấy mã số ra tỷ lệ với điện thế tương tự vào



4.2. Chuyển đổi A/D theo phương pháp dùng điện thế tham chiếu nấc thang



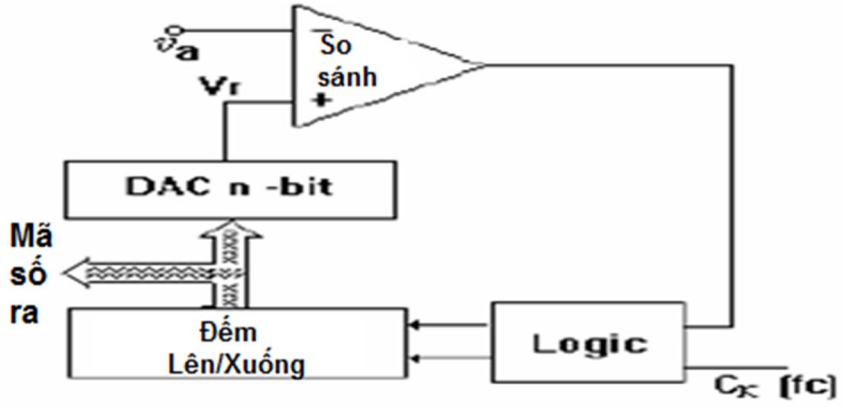
Một cách đơn giản để tạo điện thế tham chiếu có dạng nấc thang là dùng một mạch DAC mà số nhị phân vào được lấy từ mạch đếm lên. Khi có xung bắt đầu FlipFlop và mạch đếm được đặt về 0 nên đầu ra Q của FF lên 1, mở cổng AND cho xung C_K vào mạch đếm. Đầu ra mạch đếm tăng dần theo dạng nấc

thang (V_{DAC}), đây chính là điện thế tham chiếu, khi Vr còn nhỏ hơn v_a, đầu ra mạch so sánh còn ở mức thấp và Q vẫn tiếp tục ở mức cao, nhưng khi Vr vừa vượt v_a đầu ra mạch so sánh lên cao khiến Q xuống thấp, đóng cổng AND không cho xung C_K qua và mạch đếm dừng. Đồng thời đầu ra Q lên cao báo kết thúc sự chuyển đổi. Số đếm ở mạch đếm chính là số nhị phân tương ứng với điện thế vào.

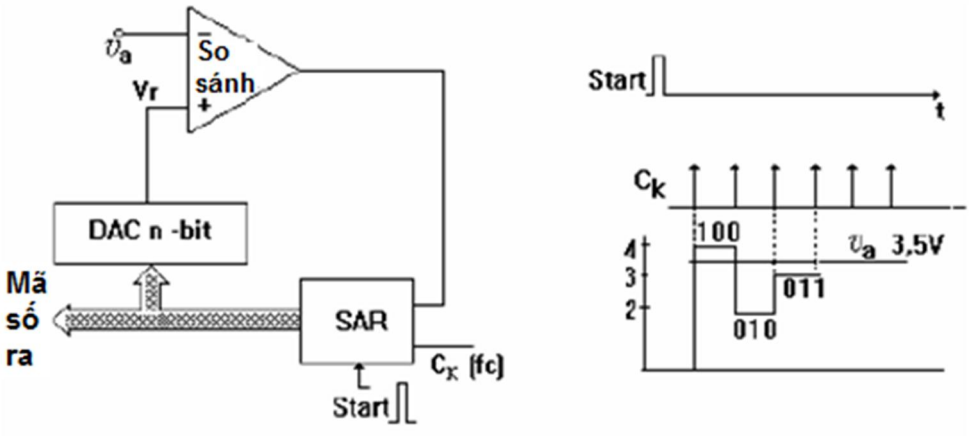
Gọi thời gian chuyển đổi là t_c. Thời gian chuyển đổi tùy thuộc điện thế cần chuyển đổi. Thời gian lâu nhất ứng với điện thế vào bằng trị lớn nhất:

$$t_c(max) = 2^n / f_{CK} = 2^n . T_{CK}$$

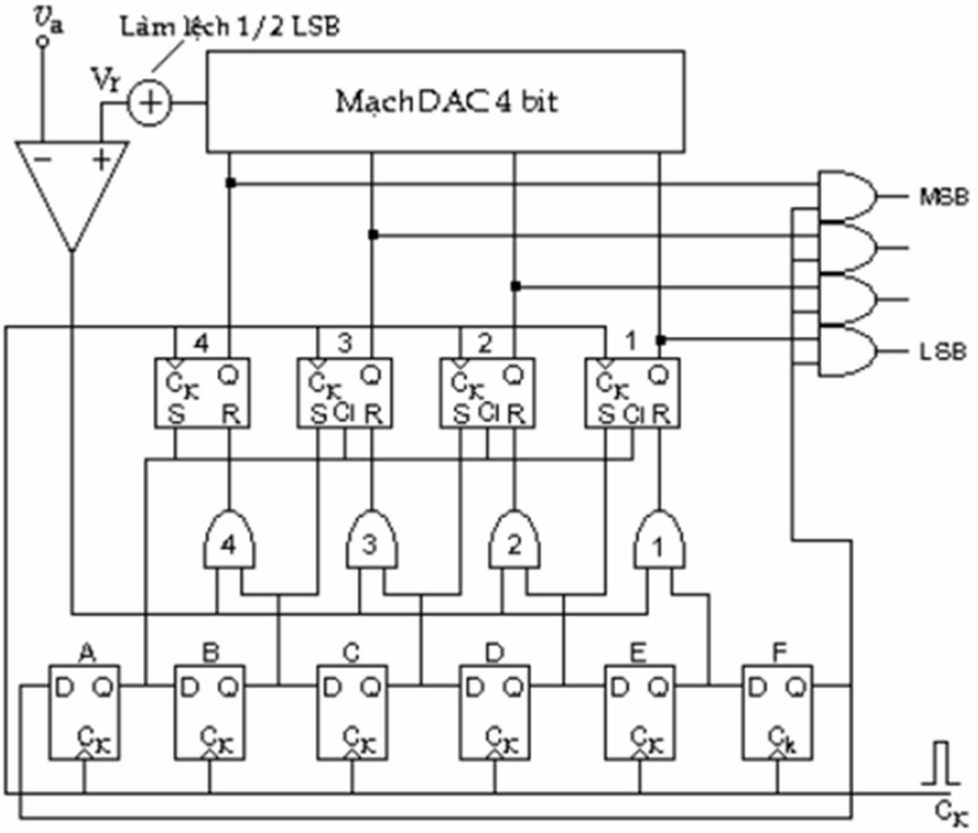
Mạch đổi này có tốc độ chậm. Một cách cải tiến là thay mạch đếm lên bởi một mạch đếm lên/xuống. Nếu đầu ra mạch so sánh cho thấy Vr nhỏ hơn v_a, mạch Logic sẽ điều khiển đếm lên và ngược lai thì mạch sẽ đếm xuống. Nếu v_a không đổi Vr sẽ dao động quanh trị v_a với hai trị số khác nhau 1 LSB



4.3. Chuyển đổi A/D nối tiếp dùng phương pháp lấy gần đúng kế tiếp



Mạch đổi lấy gần đúng kế tiếp dùng cách tạo điện thế tham chiếu một cách có hiệu quả hơn khiến việc chuyển đổi ra mã số n bit chỉ tốn n chu kỳ xung C_K. Mạch này bao gồm: một mạch so sánh, một mạch ghi dịch đặc biệt (SAR) và một mạch DAC.



Mạch SAR (Hình trên) là mạch ghi dịch có kết hợp điều khiển Logic. Mạch gồm 6 FF D mắc thành chuỗi, đầu ra FF cuối (F) hồi tiếp về FF đầu (A) , khối điều khiển gồm 4 cổng AND và 4 FF RS có đầu vào tác động mức cao, các đầu ra Q của các FF RS được đưa vào mạch DAC để tạo điện thế tương tự Vr (dùng so sánh với điện thế ra từ mạch lấy mẫu và giữ v_a), đồng thời đây cũng là mã số ra khi sự biến đổi đã kết thúc.

Nguyên lý hoạt động: Lúc có xung bắt đầu, mạch SAR được đặt về 0. Đầu ra DAC được làm lệch 1/2 LSB để tạo đặc tính chuyển đổi như đã nói trong phần trước, kế đó SAR đưa bit MSB lên cao (bằng cách preset FF A), các bit khác bằng 0, số này được đưa vào mạch DAC để tạo điện thế tham chiếu Vr để so sánh với v_a. Tùy theo kết quả so sánh, nếu Vr > v_a thì đầu ra mạch so sánh ở mức cao khiến SAR bỏ đi bit MSB khi có xung C_K kế tiếp xuất hiện, còn nếu Vr < v_a thì đầu ra mạch so sánh ở mức thấp, khiến SAR giữ bit MSB lại (FF RS 4 giữ nguyên trạng thái) đồng thời đưa bit có nghĩa kế tiếp lên cao (do FF 3 được set từ giá trị 1 ở đầu ra FF B, trị 1 này được chuyển từ FF A sang). Mạch so sánh tiếp tục làm việc và kết quả sẽ được quyết định theo cùng cách thức như đối với bit MSB.... Tiếp tục như vậy cho đến bit cuối cùng của SAR, lúc đó v_a gần Vr nhất và ta được kết quả chuyển đổi trong thời gian tối đa là n chu kỳ xung đồng hồ. Mạch chuyển đổi chấm dứt khi đầu ra FF F lên mức cao cho phép mở các đệm để cho mã số ra.

5. Lắp ráp, sửa chữa mạch chuyển đổi A/D

Mục tiêu:

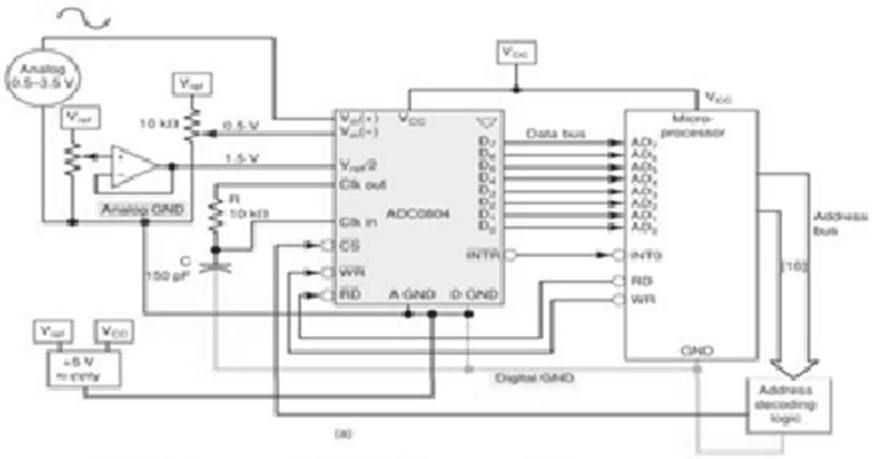
Lắp ráp hàn chỉnh một mạch chuyển đổi A/D.

5.1. Chuẩn bị vật tư và dụng cụ

- Linh kiện kiểm tra xác định theo sơ đồ nguyên lý.
- Chuẩn bị máy phát cao tần, máy hiện sóng; Đồng hồ vạn năng số, mỏ hàn; Bo mạch in đa năng (có thể chuẩn bị mạch in) …

5.2. Lắp mạch

Sơ đồ nguyên lý mạch A/DC 8 bit theo nguyên lý SAC dùng IC ADC0804



Trình tự thực hiện:

- Khảo sát sơ đồ chân của IC ADC0804
- Lắp mạch trên bo đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiếc trước khi dùng làm phần tử kết nối trong mạch.
- Bố trí các linh kiện hợp lý
- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.
- Mối hàn phải ngấu, bóng.
- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào.
- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và tín hiệu tương tự ở đầu vào.

- Lắp mạch dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp đã được thiết kế trước (không có đoạn mạch in nào được thiết kế là liên mà lại bị đứt trên bo mạch in chuẩn bị cho lắp ráp).
- Có sơ đồ lắp ráp kèm theo.
- Khi lắp phải cắm đúng vị trí và chiều các linh kiện theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, bóng, gọn, không gây chạm chập trên mạch in.
- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào.
- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và tín hiệu tương tự ở đầu vào.

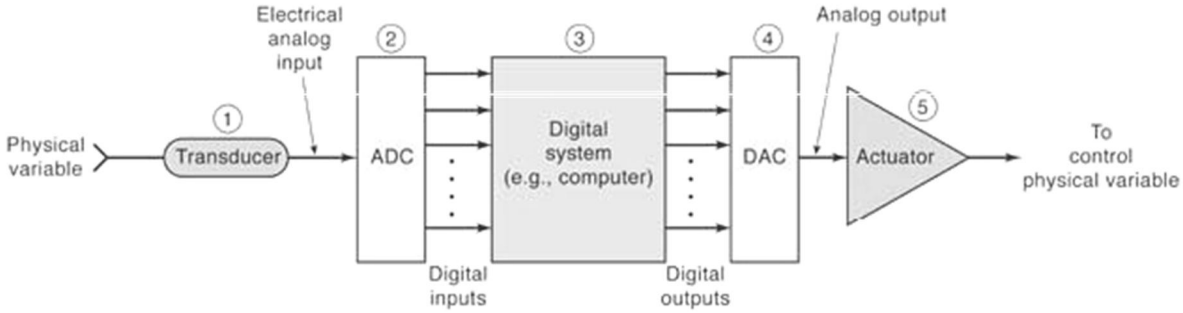
5.3. Thực hành đo và cân chỉnh

5.4. Sửa chữa một số mạch A/D thường gặp.

BÀI 12
MẠCH CHUYỂN ĐỔI D/A
Mã bài: MĐ26.12

Giới thiệu:

Có thể nói sự biến đổi qua lại giữa các tín hiệu từ dạng tương tự sang dạng số là cần thiết trong các quá trình thực hiện giao tiếp giữa con người, tự nhiên với máy trong thực tế lao động sản xuất và nghiên cứu hàng ngày, có thể biểu diễn một cách tương đối theo sơ đồ sau:



Kết quả từ các hệ thống số là các đại lượng số: Cần thiết phải đổi các tín hiệu số đó thành tín hiệu tương tự để có thể tác động vào các hệ thống vật lý và thể hiện ra bên ngoài (ví dụ tái tạo âm thanh hay hình ảnh) hay dùng vào việc điều khiển sau đó (ví dụ dùng điện thế tương tự để điều khiển vận tốc động cơ).

Mục tiêu:

- Trình bày được khái niệm, nguyên lý cơ bản của mạch chuyển đổi D/A.
- Lắp ráp, sửa chữa được mạch chuyển đổi D/A.
- Có tính tư duy, tác phong trong công nghiệp

Nội dung chính:

1. Khái niệm chung

Mục tiêu:

Trình bày được khái niệm về mạch chuyển đổi D/A.

Để làm việc được với mạch chuyển đổi D/A ta phải nắm chắc được các khái niệm:

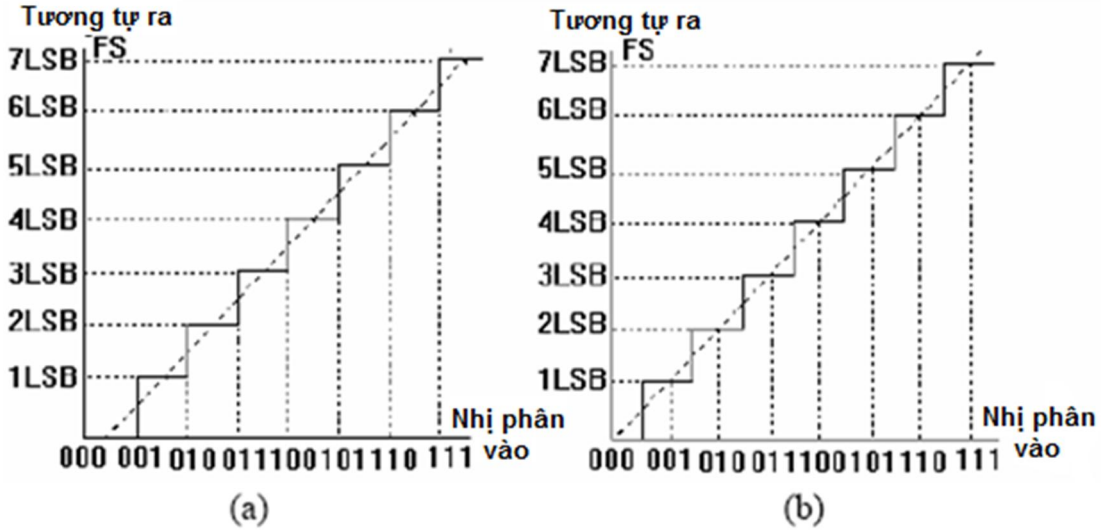
1.1. Bit có ý nghĩa thấp nhất (LSB) và bit có ý nghĩa cao nhất (MSB)

Qua các mạch biến đổi DAC kể trên ta thấy vị trí khác nhau của các bit trong số nhị phân cho giá trị biến đổi khác nhau, nói cách khác trị biến đổi của một bit tùy thuộc vào trọng lượng của bit đó.

Nếu ta gọi trị toàn giai là V_{FS} thì bit LSB có giá trị là: $LSB = V_{FS} / (2^n - 1)$

và bit $MSB = V_{FS} .2^{n-1} / (2^n - 1)$

Dưới đây là đặc tuyến chuyển đổi của một số nhị phân 3 bit



Hình (a) là đặc tuyến lý tưởng, tuy nhiên, trong thực tế để đường trung bình của đặc tính chuyển đổi đi qua điểm 0 điện thế tương tự ra được làm lệch (1/2)LSB - Hình (b). Như vậy điện thế tương tự ra được xem như thay đổi ở ngay giữa hai mã số nhị phân vào kế nhau. Ví dụ khi mã số nhị phân vào là 000 thì điện thế tương tự ra là 0 và điện thế tương tự ra sẽ lên nấc kế 000+(1/2)LSB rồi nấc kế tiếp ở 001+(1/2)LSB.v.v....Trị tương tự ra ứng với 001 gọi tắt là 1LSB và trị toàn giai V_{FS} = 7LSB tương ứng với số 111

1.2. Sai số nguyên lượng hóa (quantization error)

Trong sự biến đổi, ta thấy ứng với một giá trị nhị phân vào, ta có một khoảng điện thế tương tự ra. Như vậy có một sai số trong biến đổi gọi là sai số nguyên lượng hóa và =(1/2)LSB

1.3. Độ phân giải (resolution)

Độ phân giải được hiểu là giá trị thay đổi nhỏ nhất của tín hiệu tương tự ra có thể có khi số nhị phân vào thay đổi. Độ phân giải còn được gọi là trị bước (step size) và bằng trọng lượng bit LSB.

Số nhị phân n bit có 2^n giá trị và 2^n - 1 bước

Hiệu thế tương tự ra xác định bởi $v_0 = k.(B)_2$

Trong đó k chính là độ phân giải và $(B)_2$ là số nhị phân

Người ta thường tính phần trăm phân giải: $\%res = (k / V_{FS})100\ \%$

Với số nhị phân n bit $\%res = [1 / (2^n - 1)]100\ \%$

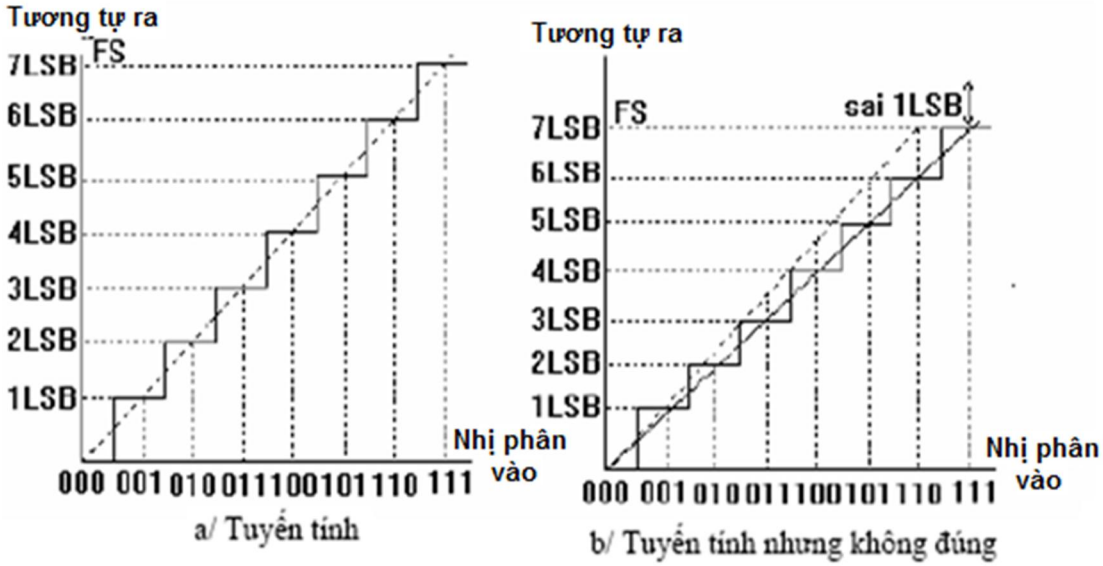
Các nhà sản xuất thường dùng số bit của số nhị phân có thể được biến đổi để chỉ độ phân giải. Số bit càng lớn thì độ phân giải càng cao (finer resolution).

1.4. Độ tuyến tính (linearity)

Khi điện thế tương tự ra thay đổi đều với số nhị phân vào ta nói mạch biến đổi có tính tuyến tính

1.5. Độ đúng (accuracy)

Độ đúng (còn gọi là độ chính xác) tuyệt đối của một DAC là hiệu số giữa điện thế tương tự ra và điện thế ra lý thuyết tương ứng với mã số nhị phân vào. Hai số nhị phân kế nhau phải cho ra hai điện thế tương tự khác nhau đúng 1LSB, nếu không mạch có thể tuyến tính nhưng không đúng.



2. Các tham số cơ bản

Mục tiêu:

Trình bày được các tham số cơ bản của mạch chuyển đổi D/A.

Các bộ DAC có thể được tích hợp vào trong những IC, một số thông số cơ bản của nó như sau:

- Resolution: Bước nhảy của bộ DAC - Bước nhảy của bộ biến đổi D/A được định nghĩa là khoảng thay đổi nhỏ nhất của đầu ra khi có sự thay đổi giá trị đầu vào. Một bộ biến đổi D/A N bit: Số mức đầu ra khác nhau là 2^N, số bước nhảy sẽ là 2^{N-1}. Số bước nhảy quy ước là K thì $K = \frac{Vref}{2^{N-1}}$
- Accuracy: Sai số sai số của bộ DAC
- Offset error: Đầu ra của DAC khi tất cả đầu vào bằng 0
- Settling time: Thời gian yêu cầu để DAC thực hiện biến đổi khi đầu vào chuyển đổi từ trạng thái **all 0** đến trạng thái **all 1.**

3. Các phương pháp chuyển đổi

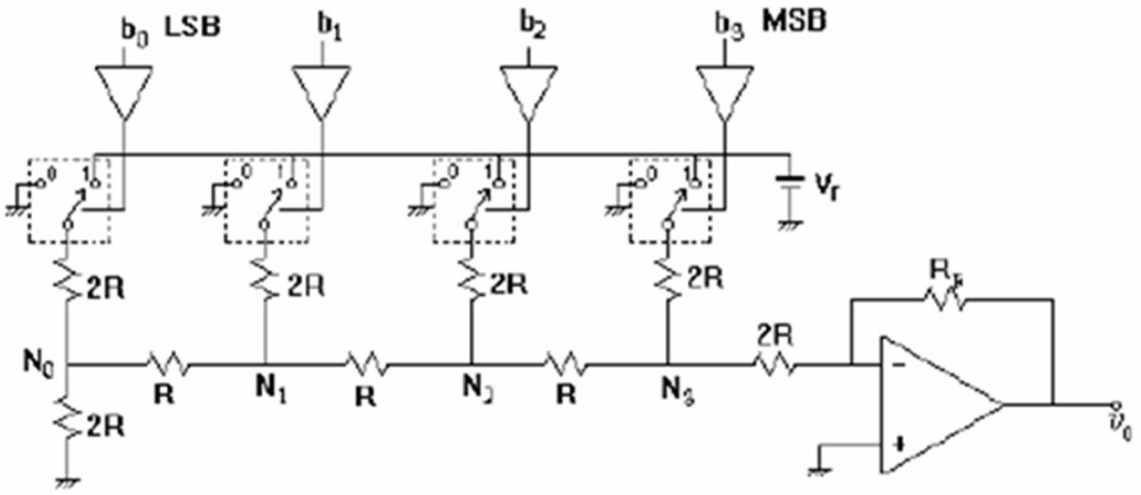
Mục tiêu:

Trình bày được các phương pháp chuyển đổi D/A.

3.1. Chuyển đổi D/A theo phương pháp thang điện trở

Mạch còn có tên gọi khác là phương pháp dùng mạng điện trở hình thang. Nội dung của phương pháp như sau:

Sơ đồ nguyên lý bộ D/AC 4 bit thực hiện như ở hình dưới đây.



Cho $R_F = 2R$ và lần lượt

Cho $b_3 = 1$ các bit khác = 0, ta được: $v_0 = -8(Vr/24)$

Cho $b_2 = 1$ các bit khác = 0, ta được: $v_0 = -4(Vr/24)$

Cho $b_1 = 1$ các bit khác = 0, ta được: $v_0 = -2(Vr/24)$

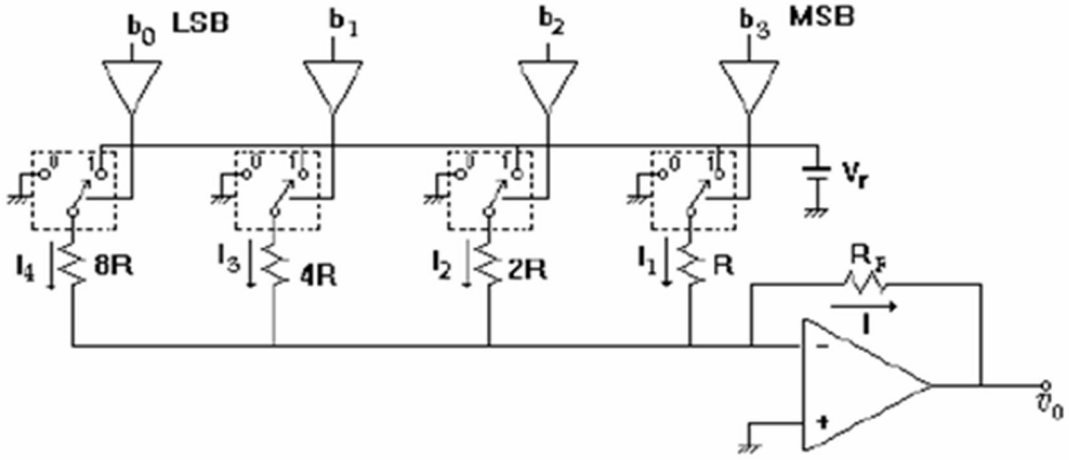
Cho $b_0 = 1$ các bit khác = 0, ta được: $v_0 = -(Vr/24)$

Ta thấy v_0 tỉ lệ với giá trị B của tổ hợp bit

$B = (b_3 b_2 b_1 b_0)_2 \Rightarrow v_0 = -B(Vr/24)$

3.2. Chuyển đổi D/A theo phương pháp mạch điện trở

Còn được gọi là phương pháp dùng mạng điện trở có trọng số khác nhau. Sơ đồ nguyên lý của mạch điện thực hiện một bộ chuyển đổi D/A 4 bít và nguyên lý hoạt động của nó như sau:



Ở mạch trên, nếu thay OP-AMP bởi một điện trở tải, ta có tín hiệu ra là dòng điện. Như vậy OP-AMP giữ vai trò biến dòng điện ra thành điện thế ra, đồng thời nó là một mạch cộng

Ta có $v_0 = -R_F.I = -(2^3 b_3 + 2^2 b_2 + 2b_1 + b_0)Vr.R_F/2^3R$

$= -(2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + ...+ 2b_1 + b_0)Vr.R_F/2^{n-1}.R$

Nếu $R_F = R$ thì:

$v_0 = -(2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + ...+ 2b_1 + b_0)Vr. /2^{n-1}$.

Ví dụ:

1- Khi số nhị phân là 0000 thì $v_0 = 0$

1111 thì $v_0 = -15Vr / 8$

2 - Với Vr = 5V ; $R = R_F = 1k\Omega$

Ta có kết quả chuyển đổi như sau:

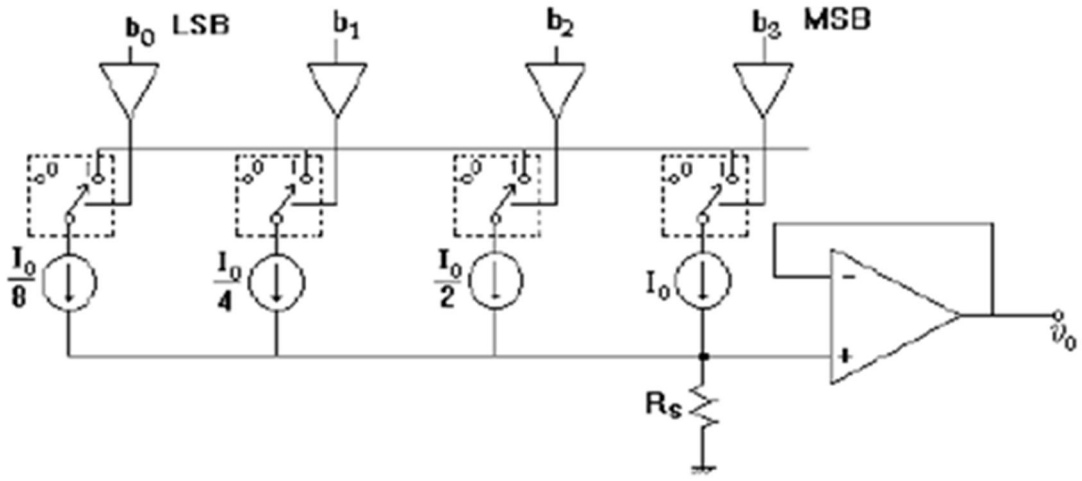
b_3	b_2	b_1	b_0	v_0 (V)
0	0	0	0	0
0	0	0	1	-0,625 ← LSB
0	0	1	0	-1,250
0	0	1	1	-1,875
0	1	0	0	-2.500
0	1	0	1	-3,125
0	1	1	0	-3,750
0	1	1	1	-4,375
1	0	0	0	-5,000
1	0	0	1	-5,625
1	0	1	0	-6,250
1	0	1	1	-6,875
1	1	0	0	-7,500
1	1	0	1	-8,125
1	1	1	0	-8,750
1	1	1	1	-9,375 ← Full Scale (V_{FS})

Mạch có một số hạn chế:

- Sự chính xác tùy thuộc vào điện trở và mức độ ổn định của nguồn tham chiếu Vr
- Với số nhị phân nhiều bit thì cần các điện trở có giá trị rất lớn, khó thực hiện.

3.3. Chuyển đổi D/A theo phương pháp dùng nguồn dòng có trọng số khác nhau.

Sơ đồ nguyên lý của mạch điện thực hiện một bộ chuyển đổi D/A 4 bít và nguyên lý hoạt động của nó như sau:



4. Lắp ráp, sửa chữa mạch chuyển đổi D/A

Mục tiêu:

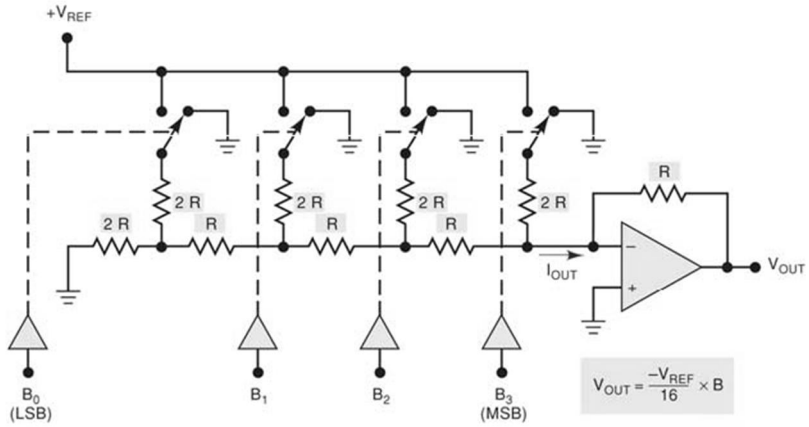
Lắp được mạch chuyển đổi D/A theo sơ đồ nguyên lý; đo và cân chỉnh được mạch chuyển đổi vừa lắp đúng yêu cầu kỹ thuật.

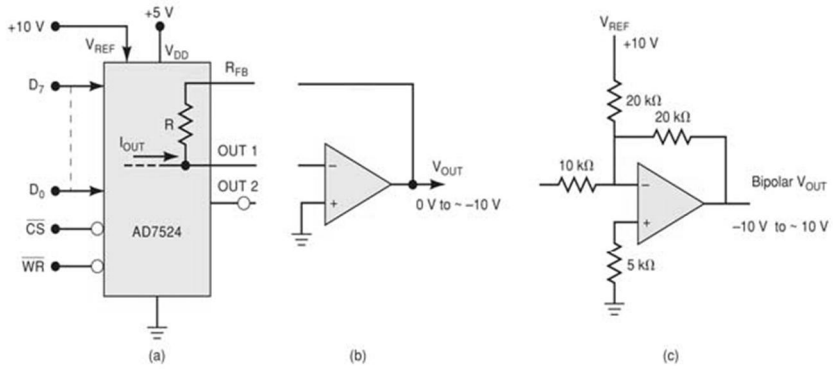
4.1. Chuẩn bị vật tư và dụng cụ

- Linh kiện kiểm tra xác định theo sơ đồ nguyên lý.
- Chuẩn bị máy phát cao tần, máy hiện sóng; Đồng hồ vạn năng số, mỏ hàn; Bo mạch in đa năng (có thể chuẩn bị mạch in) …

4.2. Lắp mạch

Sơ đồ nguyên lý mạch D/AC 8 bit theo phương pháp mạng điện trở có trọng số khác nhau dùng IC DAC loại AD7524





Trình tự thực hiện:

- Khảo sát sơ đồ chân của IC AD7524
- Lắp mạch trên bo đa năng:

- Yêu cầu chuẩn bị các linh kiện, dây nối được vệ sinh và tráng thiếc trước khi dùng làm phần tử kết nối trong mạch.

- Bố trí các linh kiện hợp lý

- Các đường dây nối trong mạch phải sóng, đẹp, không chồng chéo, dễ quan sát khi hiệu chỉnh và sửa chữa.

- Mối hàn phải ngấu, bóng.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào.

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và tín hiệu số ở đầu vào.

- Lắp mạch dùng IC trên bo mạch in chuẩn bị sẵn:

- Yêu cầu mạch in chuẩn bị sẵn (theo các phần mềm thiết kế mạch in đã có) phải sạch, các đường mạch in phải liền theo đúng sơ đồ lắp ráp đã được thiết kế trước (không có đoạn mạch in nào được thiết kế là liên mà lại bị đứt trên bo mạch in chuẩn bị cho lắp ráp).

- Có sơ đồ lắp ráp kèm theo.

- Khi lắp phải cắm đúng vị trí và chiều các linh kiện theo đúng sơ đồ lắp ráp, mối hàn phải ngấu, bóng, gọn, không gây chạm chập trên mạch in.

- Phải biết tiến hành kiểm tra nguội mạch để đảm bảo không gây chạm, chập, hở mạch hoặc các lỗi khác trước khi cấp nguồn và tín hiệu đầu vào.

- Mạch phải đảm bảo hoạt động đúng yêu cầu khi cấp nguồn đúng định mức và tín hiệu số ở đầu vào.

4.3. Thực hành đo và cân chỉnh

4.4. Sửa chữa một số mạch D/A thường gặp.

Tài liệu tham khảo:

- Giáo trình kỹ thuật số: Sách dùng cho các trường đào tạo hệ trung học chuyên nghiệp – Nguyễn Viết Nguyên – NXB Giáo dục - 2002
- Nguyễn Thúy Vân – Kỹ thuật số - NXB Khoa học và kỹ thuật – 1996
- Hồ Tấn Mẫn – Hệ thống kỹ thuật số nhập môn – NXB Đà Nẵng – 2002
- Giáo trình kỹ thuật số - ĐH SPKT TP. HCM
- Kỹ thuật xung số - NXB Khoa học và Kỹ thuật 2004
- Th.s Đặng Ngọc Khoa - Bài giảng Kỹ thuật số - Khoa Điện – Điện tử - ĐHCN TP Hồ Chí Minh.