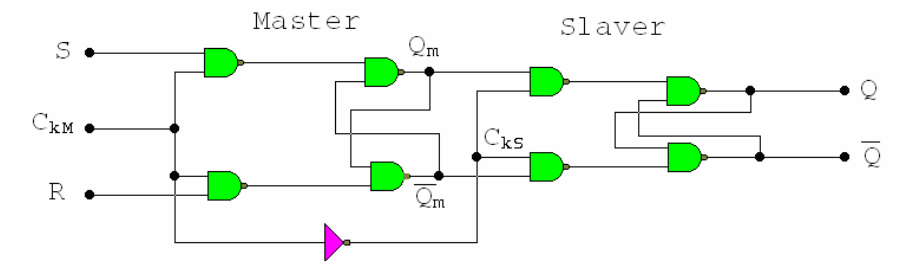
TRƯỜNG ĐẠI HỌC SƯ PHAM HUẾ

THs: PHAN VĂN ĐƯỜNG

GIÁO TRÌNH ĐIỆN TỬ

VI MẠCH – ĐIỆN TỬ SỐ



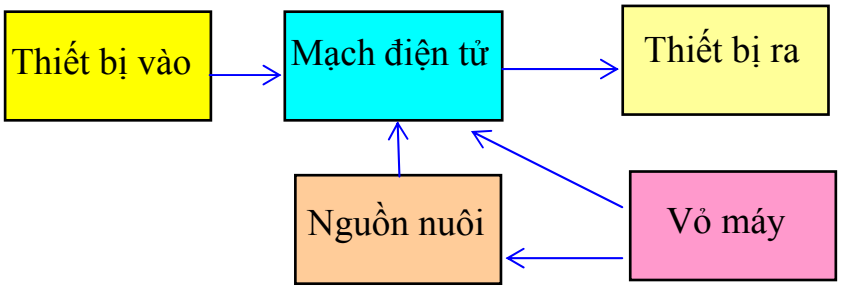
HUẾ 3-2008

CHƯƠNG 1 VI MẠCH (I.C.)

1.1. KHÁI NIỆM MỞ ĐẦU

Thiết bị điện tử là những dụng cụ, máy móc dùng các linh kiện điện tử giúp cho con người thực hiện một chức năng nào đó (máy tính, máy in, máy quét, máy thu hình...)

Một thiết bị điện tử thường có sơ đồ khối như hình sau (Hình 1.1)



Hình 1.1: *Sơ đồ khối một thiết bị điện tử*

Thiết bị vào: Biến đổi những tín hiệu không điện thành điện (đầu từ, bàn phím, camera, micro v.v...)

Thiết bị ra: Biến đổi các tín hiệu đã được gia công, xử lý thành những mục đích cần khống chế và điều khiển (đưa ra loa, đầu từ, hiển thị lên màn hình...)

Nguồn cung cấp: Cung cấp toàn bộ năng lượng cho máy hoạt động, nguồn cung cấp là nguồn điện một chiều được lọc rất kỷ và rất ổn định.

Vỏ máy: Bảo vệ thiết bị bên trong và để trang trí.

Mạch điện tử: Phần quan trọng nhất của thiết bị điện tử, đóng vai trò gia công và xử lý số liệu theo những mục đích và chương trình định trước. Việc gia công và xử lý này căn cứ vào đặc tính của từng phần tử của mạch, căn cứ vào những định luật ghép nối các phần tử với nhau. Bao gồm:

a/Linh kiện điện tử : Được chia làm hai loại

* Linh kiện tích cực: Đóng vai trò chính trong thiết bị gồm có: Transistor, Diode. Tín hiệu điện qua nó sẽ bị biến đổi.

*Linh kiện thụ động: Gồm có: Điện trở (R), tụ điện (C), cuộn cảm (L). Giúp cho các linh kiện tích cực hoạt động. Chỉ gia công số liệu chứ không xử lý số liệu.

b/Mạch điện:

Các linh kiện điện tử trên được liên kết với nhau theo các định luật nhất định để thực hiện các chức năng nhất định. Có nhiều định luật để nối các phần tử với nhau nhưng chỉ có hai nguyên lý làm việc chung :

*Nguyên lý tương tự (analog): Tín hiệu ở đầu vào và đầu ra đều biến thiên liên tục theo

thời gian.

*Nguyên lý số (digital): Tín hiệu ở đầu vào và đầu ra đều biến thiên rời rạc nhằm thực hiện các phép tính toán. Nguyên lý số tác động nhanh và có khả năng rộng lớn hơn nguyên lý tương tự. Tất cả các đại lượng đều có thể biến đổi thành rời rạc (ta gọi là số hóa).

Thiết bị điện tử có các yêu cầu sau:

a/ Kích thước nhỏ: Gọn, chiếm ít không gian, trọng lượng bé nhưng vẫn giữ nguyên tính năng.

b/ Độ tin cậy cao: Xác suất để mạch làm việc bình thường trong những điều kiện cho trước (không đồng nghĩa tuổi thọ với độ bền của thiết bị).

c/ Hiệu suất cao: Tiết kiệm năng lượng: $\frac{P_2}{P_1} \rightarrow 1$

P_2: Công suất ở tải.

P_1: Công suất nguồn cung cấp.

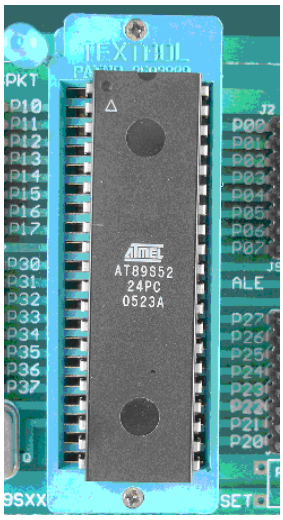
d/ Giá thành hạ.

Như vậy yêu cầu đầu tiên là giảm nhỏ kích thước của thiết bị đã đưa đến việc giảm nhỏ kích thước các linh kiện trong mạch. Điều này xuất hiện việc vi hình hóa (micro modun) mạch điện, dẫn đến việc chế tạo vi mạch.

1.2. ĐẠI CƯƠNG VỀ VI MẠCH

1.2.1. Cấu tạo

Vi mạch còn gọi là mạch tích hợp (integrated circuit), gọi tắt là IC. Có hình dang bên ngoài như hình 1.2.



Hình 1.2: *Hình dạng của vi mạch*

Đây là các mạch điện tử chứa các linh kiện tích cực (transistor, diode) và linh kiện thụ động (điện trở, tụ điện có điện dung bé), kích thước rất bé cỡ µm (hoặc nhỏ hơn) được kết nối với nhau theo công nghệ silicon. Tất cả các linh kiện của mạch được chế tạo đồng thời trên một đế (subtrate) làm bằng Silic. Vỏ ngoài của vi mạch thường làm bằng kim loại hoặc bằng

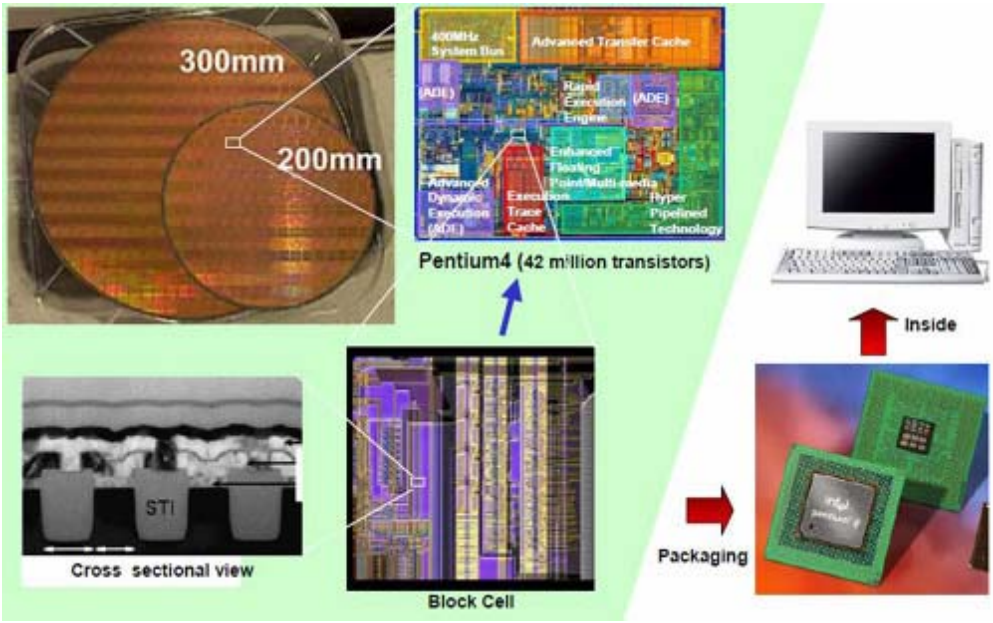
chất dẻo (plastic) Các linh kiện trong vi mạch không thể tách rời nhau. Mỗi vi mạch sẽ đảm nhiệm một chức năng điện tử nhất định nào đó (khuếch đại, giải mã, lập mã, bộ đếm, bộ nhớ...).

Có đến hàng triệu transistor trong một vi mạch, số lượng này ngày càng tăng do số lượng thông tin cần xử lý ngày càng nhiều. Mạch điện tử ngày càng phức tạp, gồm rất nhiều linh kiện điện tử được tích hợp lại. Hiện nay, công nghệ silicon đang tính tới những giới hạn của vi mạch tích hợp và các nhà nghiên cứu đang nỗ lực tìm ra một loại vật liệu mới có thể thay thế công nghệ silicon này.

Hệ thống trên một vi mạch (system-on-a-chip) SOC là một hệ thống điện tử được xây dựng trên một đế silicon. Ý tưởng ban đầu là tích hợp tất cả các linh kiện của một thiết bị điện tử (máy tăng âm, thu hình, máy tính…) lên trên một vi mạch đơn (hay còn gọi là một chip đơn). Hệ thống SOC này có thể bao gồm các khối chức năng số, tương tự, tín hiệu kết hợp (mixed-signal) và cả các khối tạo dao động. Một hệ thống điển hình bao gồm một loạt các mạch tích hợp cho phép thực hiện các nhiệm vụ khác nhau. Từ đó ta có mạch tích hợp khuếch đại, mạch lập mã, giải mã, xử lý, bộ nhớ…

Sự phát triển gần đây của công nghệ bán dẫn cho phép chúng ta tích hợp ngày càng nhiều thành phần vào một hệ thống trên một vi mạch SOC, có thể tích hợp thêm các khối như: bộ xử lý tín hiệu số, bộ mã hóa, giải mã,... tùy theo yêu cầu của từng ứng dụng cụ thể.

Hình 1.3 cho ta cấu trúc bên trong và hình dạng bên ngoài vi mạch Pentium IV



Hình 1.3: *Cấu trúc bên trong và hình dáng bên ngoài của vi mạch Pentium IV*
a/Cấu trúc bên trong, b/ Hình dạng bên ngoài, c/ Dùng trong máy điện toán cá nhân.

Vi mạch cần giải quyết các vấn đề sau:

1. Khoảng không gian mà số lượng các linh kiện điện tử chiếm chỗ:

Một máy tính điện tử cần dùng đến hàng triệu, hàng vài chục triệu bộ phận rời. Nếu không thực hiện bằng vi mạch, thì không những thể tích của nó sẽ quá lớn mà điện năng cung cấp cho nó cũng sẽ vô cùng phức tạp.

2. Độ tin cậy(reliability) của hệ thống điện tử: là độ đáng tin cậy trong hoạt động đúng theo tiêu chuẩn thiết kế. Độ tin cậy của một hệ thống tất nhiên phụ thuộc vào độ tin cậy của các thành phần cấu thành và các bộ phận nối tiếp giữa chúng. Hệ thống càng phức tạp, số bộ phận càng tăng và chỗ nối tiếp càng nhiều. Vì vậy, nếu dùng bộ phận rời cho các hệ thống phức tạp, độ tin cậy của nó sẽ giảm thấp. Một hệ thống như vậy sẽ rất dễ dàng hư hỏng.

3. Tuổi thọ trung bình t của một hệ thống điện tử gồm n thành phần sẽ là:

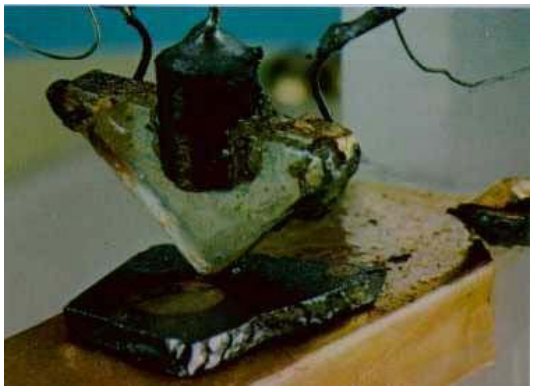
$$\frac{1}{t} = \frac{1}{t_1} + \frac{1}{t_2} + ... + \frac{1}{t_n}$$

Vậy nếu một transistor có tuổi thọ là 10^6 giờ, thì một máy tính gồm 500.000 transistor sẽ chỉ có tuổi thọ là 2 giờ.

Các thành phần trong vi mạch được chế tạo đồng thời và cùng phương pháp, nên tuổi thọ vi mạch xấp xỉ tuổi thọ một transistor Planar.

1.2.2.Lịch sử vi mạch

Năm 1947, John Bardeen và William Brattain của phòng thí nghiệm Bell (Bell Lab Hoa kỳ) phát minh ra Transistor tiếp điểm PCT (Point Contact Transistor), đây là một đột phá trong nỗ lực tìm ra thiết bị mới thay cho đèn điện tử tiêu tốn quá nhiều năng lượng. Dòng điện vào (bên trái hình tam giác) được truyền qua lớp dẫn điện (conversion layer) trên bề mặt bản Germanium và được khuyếch đại thành dòng ra (bên phải hình tam giác.)



Năm 1950, William Shockley cũng ở hãng Bell phát minh ra transistor kiểu tiếp hợp. Đây là mô hình đầu tiên của loại bipolar transitor sau này. Việc phát minh ra transistor là một bước tiến vĩ đại của kỹ thuật điện tử, mở đầu cho việc phát minh ra vi mạch. Sau đó, William Shockley rời Bell Labs, thành lập Shockley Semiconductor tại 391 đường San Antonio tại Mountain View California. Những nhân viên đầu tiên của ông có Gordon Moore và Robert

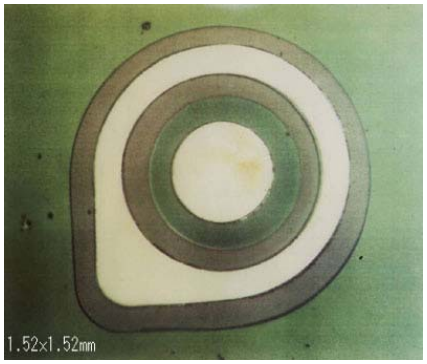
Noyce, là những người đã sáng lập ra Intel. Công ty bắt đầu phát triển các thiết bị silicon.

• 1/10/1956 William Shockley, John Bardeen và Walter Brattain được trao giải Nobel vật lý cho những nghiên cứu về chất bán dẫn và phát hiện tác động của transistor

• 1957 Một nhóm gồm 8 người đã rời Shockley Semiconductor Laboratory để thành lập Fairchild Semiconductors. Đây là công ty đầu tiên chỉ tập trung phát triển silicon.

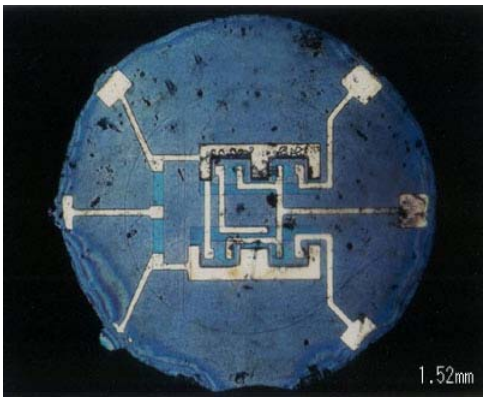
• 12 / 9/1958 Jack Saint Clair Kilby của công ty Texas Instruments (Hoa kỳ) phát minh ra mạch tích hợp (Integrated Circuit) đầu tiên, mở đầu cho thời kỳ hoàng kim của vi điện tử, là nền tảng của chip hiện đại ngày nay. Điểm quan trọng trong phát minh của Kilby là ở ý tưởng về việc tích hợp tất cả các linh kiện điện tử của một mach điện tử trên một tấm silicon.

• Năm 1959, Jean Hoerni và Robert Noyce (công ty Fairchild, Mỹ), sau này là người đồng sáng lập hãng Intel, thành công trong việc chế tạo ra transistor trên một mặt phẳng silicon. Hình 1.4 là transistor với cả 3 cực: gốc (base), thu (colector) và phát (emiter) cùng nằm trên một mặt phẳng.



Hình 1.4: *Transistor với cả 3 cực: gốc, thu và phát cùng nằm trên một mặt phẳng.*

• Năm 1961, cũng chính Jean Hoerni và Robert Noyce đã tạo ra mạch flip-flop (với 4 transistor và 5 điện trở) trên mặt silicon (Hình 1.5).

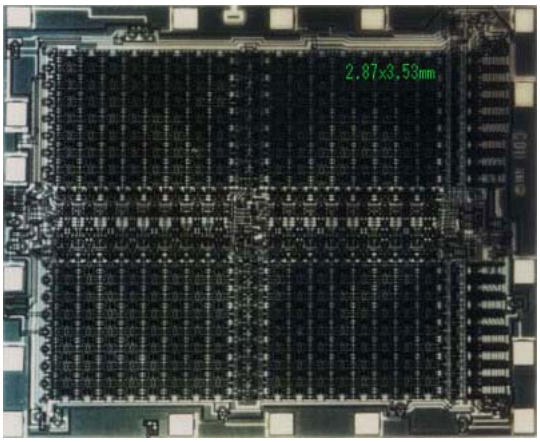


Hình 1.5: *Cấu trúc bên trong một vi mach flip-flop*

• 19/4/1965 Gordon Moore nhà sáng lập Intel với bài viết "Bổ sung thêm các thành phần

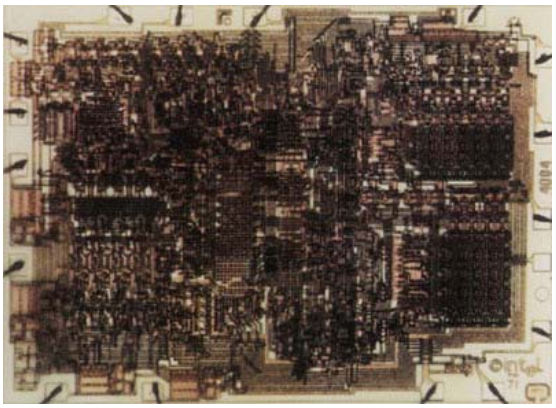
vào vi mạch" đăng trên tạp chí Electronics đã Công bố Định luật Moore, dự đoán số transistor trên chip mỗi năm sẽ tăng gấp đôi trong vòng 10 năm tới. Năm 1975 ông đã sửa lại là cứ 24 tháng thì số transistor lại tăng gấp đôi. Tới nay dự báo của ông vẫn còn đúng.

• Năm 1970, G.E.Smith và W.S.Boyle (AT&T Bell Lab., USA) tạo ra mạch CCD 8-bit. Cùng năm 1970, J.Karp và B.Regitz (công ty Intel, Mỹ) tạo ra mạch DRAM 1103 với trên 1000 tế bào nhớ. (Hình 1.6)



Hình 1.6: *DRAM 1103*

• Năm 1971, M.E.Hoff, S.Mazer, 嶋 正利, F.Faggin (công ty Intel, Mỹ) tạo ra bộ vi xử lý 4004 với trên 2.200 transistor (Hình 1.7).

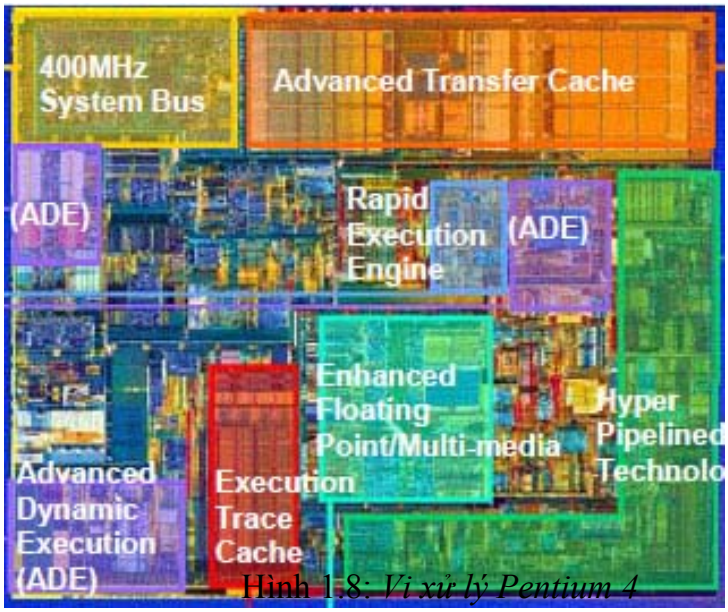


Hình 1.7: *Vi xử lý Intel 4004*

• Năm 2000 Kilby được giải Nobel Vật lý cho việc chế tạo ra mạch tích hợp. 42 năm sau khi công nghệ này ra đời.

• Năm 2004, công ty Intel (Mỹ) chế tạo chip Pentium 4 với trên 42 triệu transistor (Hình 1.8).

Cùng năm 2004, Intel tung ra chip Itanium 2 (9MB cache) phục vụ máy chủ, với số transistor lên tới 592 triệu con.



Hình 1.8: *Vi xử lý Pentium 4*

Hình 1.8: *Chip Pentium 4*

• Năm 2005, nhóm liên kết giữa IBM, SONY, SONY Computer Entertainment, và Toshiba giới thiệu chip CELL đa lõi (multicore), hoạt động ở tốc độ 4GHz.

Chưa đầy 50 năm kể từ ngày Kilby đề xuất ra ý tưởng về vi mạch, ngành công nghệ vi mạch đã đạt được những thành tựu rực rỡ. Sự tăng trưởng ở tốc độ chóng mặt của ngành công nghệ vi mạch là chìa khóa quan trọng bậc nhất trong cuộc cách mạng công nghệ thông tin hiện nay.

1.2.3.Vỏ ngoài của vi mạch

Hiện nay do chưa thể chế tạo được một số linh kiện có trị số lớn trong vi mạch (tụ điện có điện dung lớn, cuộn cảm...). Do mức độ tích hợp ngày càng lớn, vi mạch cần kết nối với các linh kiện, các thiết bị khác nên vi mạch có nhiều chân ra ngoài để nối với các linh kiện, các thiết bị này. Có nhiều kiểu vỏ bọc khác nhau làm bằng kim loại, gốm (ceramic) hoặc chất dẻo (plastic), hiện nay phổ biến các kiểu sau:

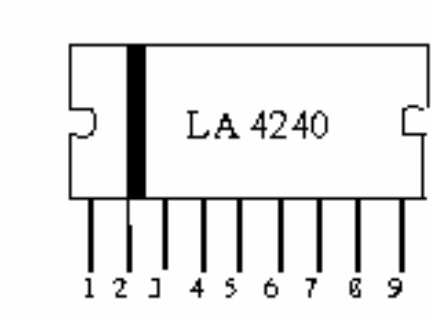
Loại SIP *(Single in Package)**:***

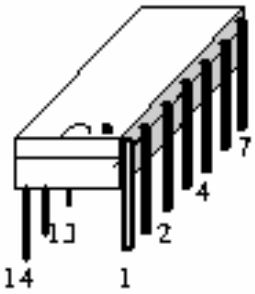
Có hình chữ nhật, chân ra chia đều trên một hàng. Chân số 1 được nhận biết nhờ một đường vạch ngang hoặc một chấm (Hình 1.9a)

Loại DIP (Dual in Package):

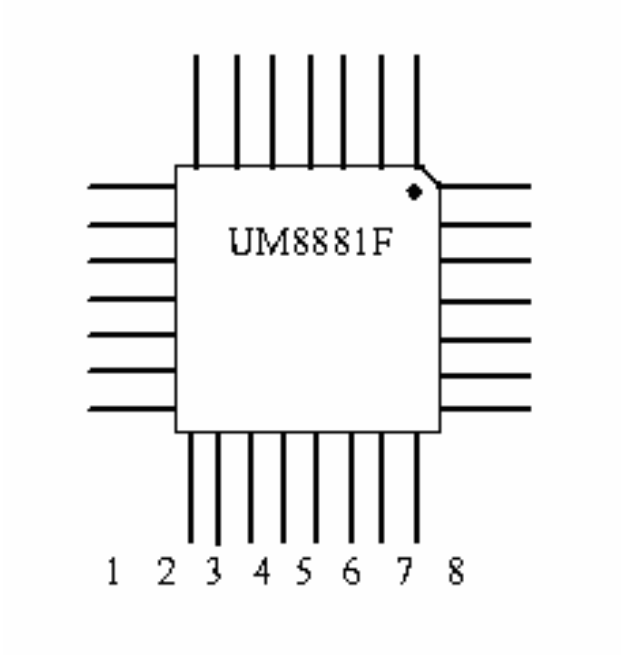
Chân vi mạch được chia làm hai hàng song song (Hình 1.9b)

Loại QFP (Quad Flat Package): Dạng hình vuông, có 4 hàng chân ra được bố trí chung quanh vi mạch. Thường gặp ở các vi mạch cực lớn như các Vi xử lý. (Hình1.9c)





Hình 1.9a: *Vỏ vi mạch loại SIP* ***Hình 1.9b:*** *Vỏ vi mạch loại DIP*



Hình 1.9c: *Vỏ vi mạch loại QFP*

1.2.4. Phân loại vi mạch

a/Dựa trên quan điểm thiết kế vi mạch: Người ta phân loại dựa trên mức độ tích hợp các phần tử trong vi mạch. Chia làm:

Vi mạch cở nhỏ SSI (Small Scale Intergration): Chứa vài chục Transistor hoặc vài cổng logic. Ra đời từ đầu thập niên 60 (mạch khuếch đại, mạch lật...)

Vi mạch cở vừa MSI (Medium Scale Intergration): Chứa vài chục cổng logic hoặc hàng trăm transistor. Ra đời giữa thập niên 60 (bộ giải mã,thanh ghi, bộ đếm...)

Vi mạch cở lớn LSI (Large Scale Intergration) : Chứa vài trăm cổng logic hoặc hàng ngàn transítor. Ra đời đầu thập niên 70 (các vi xử lý 4 hoặc 8 bit, cửa ghép nối vào ra...)

Vi mạch cực lớn VLSI (Very Large Scale Intergration): Chứa vài ngàn cổng logic hoặc hàng vạn transistor. Ra đời cuối thập niên 70 (các vi xữ lý 16 hoặc 32 bit ...)

Vi mạch ULSI (Ultra Large Scale Intergration): Chứa vài trăm ngàn cổng hoặc vài triệu transistor. Ra đời đầu thập niên 90 cho đến nay.

Bảng 1.1: *Mức độ tích hợp trong các vi mạch*

Loại vi mạch	Số lượng chức năng	Số lượng Transistor	Diện tích bề mặt của mỗi vi mạch
SSI	2 ÷ 20	100	3 mm^2
MSI	20 ÷ 100	500	8 mm^2
LSI	100 ÷ 50.000	100.000	20 mm^2
VLSI	50.000 ÷ 100.000	250.000	40 mm^2
ULSI	100.000 ÷ 400.000	1.000.000 ÷ 4.000.000	70 mm^2 ÷ 150 mm^2

b/Dựa trên quan điểm sử dụng: Tùy theo bản chất của tín hiệu vào và ra người ta chia làm 3 loại sau:

Vi mạch tuyến tính (IC Analog): Tín hiệu vào và ra có biên độ biến thiên liên tục theo thời gian. Còn gọi là vi mạch tương tự, vi mạch thuật toán.

Vi mạch số (IC Digital): Biên độ tín hiệu vào và ra có giá trị gián đoạn (thường ở hai mức điện áp). Còn gọi là vi mạch logic.

Vi mạch chuyển đổi: Là cầu nối giữa 2 loại trên gồm:

*ADC (Analog Digital Converter): Tín hiệu vào lên tục, tín hiệu ra gián đoạn.

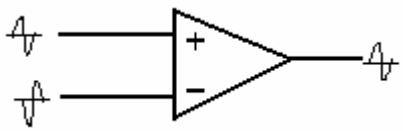
*DAC (Digital Analog Converter): Tín hiệu vào gián đoạn, tín hiệu ra liên tục.

1.3.VI MẠCH TUYẾN TÍNH

Vi mạch tuyến tính là những mạch tổ hợp mà điện áp ra là một hàm liên tục đối với điện áp vào.

Vi mạch tuyến tính còn được gọi là vi mạch khuếch đại thuật toán (operational amplifier), vi mạch tương tự.

1.3.1. Ký hiệu : Vi mạch tuyến tính có ký hiệu như hình 1.10



Hình 1.10: *Ký hiệu của vi mạch tuyến tính*

Vi mạch tuyến tính có hai đầu vào và một đầu ra gồm:

Đầu vào đảo (-) : Tín hiệu vào và ra ngược pha nhau 180 độ

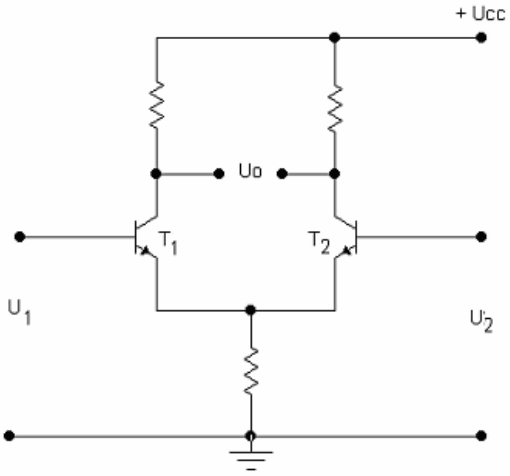
Đầu vào thuận (+) : Tín hiệu vào và ra cùng pha nhau

Một mạch khuếch đại tuyến tính lý tưởng có những đặc tính sau :

- Điện trở vào vô cùng lớn $R_v = \infty$
- Điên trở ra bằng không $R_r = 0$
- Hệ số khuếch đại vô cùng lớn $K_u = \infty$
- Dãi tần khuếch đại vô cùng lớn
- Cân bằng một cách lý tưởng : Nếu $U_v = 0$ thì $U_r = 0$
- Các thông số không bị biến đổi theo nhiệt độ và độ ẩm

1.3.2. Mạch khuếch đại vi sai

Sơ đồ cơ bản của vi mạch tuyến tính là khuếch đại vi sai. Đây là mạch khuếch đại dùng hai transistor mắc theo kiểu liên kết emitter (Hình 1.11).



Hình1.11: *Tầng khuếch đại vi sai*

Hai transistor sử dụng phải có các thông số hoàn toàn giống nhau, trở kháng vào Z_v vô cùng lớn.

Mạch khuếch đại vi sai có đặc điểm rất quan trọng là tín hiệu ra tỉ lệ với hiệu số của hai tín hiệu vào:

$U_r = K(U_1 - U_2)$ (1.1)

K là hệ số khuếch đại của mạch khuếch đại vi sai.

Từ (1.1) ta thấy: bất kỳ một thăng giáng nào của điện áp tín hiệu vào chung cho cả hai lối vào sẽ bị khữ lẫn nhau và không làm ảnh hưởng đến lối ra.

Thực tế, tín hiệu ra của mạch không những chỉ phụ thuộc vào hiệu số giữa hai tín hiệu vào, nó còn phụ thuộc vào mức trung bình của hai tín hiệu đó:

$$U_{ra} = \frac{U_1 + U_2}{2}$$

Như vậy (1.1) có thể viết: $$U_{ra} = K(U_1 - U_2) + \frac{K'}{2}(U_1 + U_2)$$

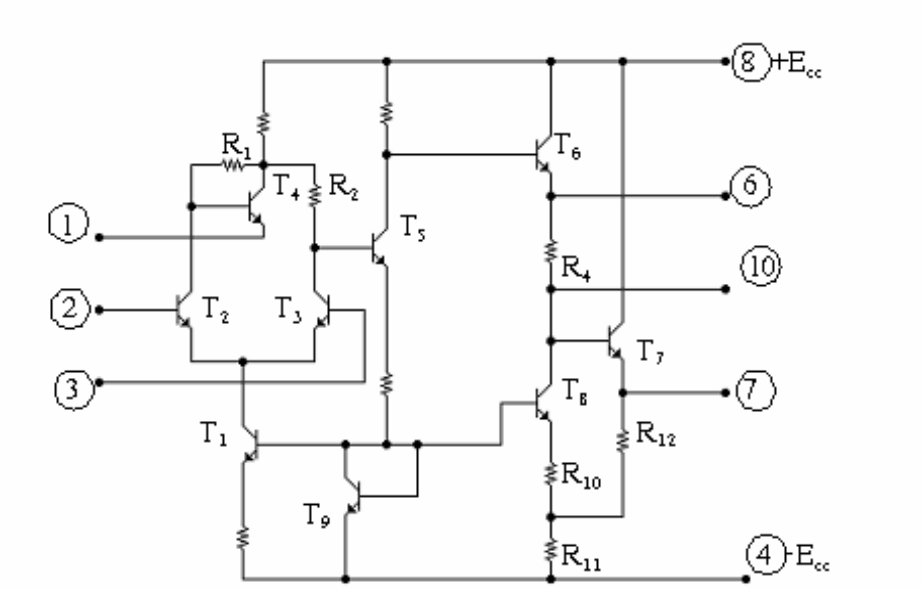
K᾽ là hệ số khuếch đại trung bình

1.3.3 Sơ đồ nguyên lý một vi mạch tuyến tính:

Vi mạch tuyến tính có cấu trúc bên trong tuỳ thuộc vào nhiệm vụ và yêu cầu kỹ thuật của vi mạch. Để xét cấu trúc của vi mạch thuật toán, ta có thể xét hai vi mạch tuyến tính thông dụng: 702 và 709.

a/Vi mạch khuếch đại thuật toán loại 702:

Tuỳ theo hãng sản xuất nó có nhiều tên gọi khác nhau: µA702L, SN52702N... Sơ đồ nguyên lý như hình 1.12



Hình 1.12: *Sơ đồ nguyên lý của vi mạch thuật toán loại 702*

Tầng thứ nhất: Gồm T_2 và T_3 các gánh của chúng là R_1, R_2 đây là tầng khuếch đại vi sai với nguồn dòng là T_1, và T_9. T_9 có nhiệm vụ bù nhiệt.

Tầng thứ hai: Bao gồm T_4 và T_5, điện áp ra của tầng này lấy từ cực C của T_5. T_4 điều chỉnh gánh của tầng một bằng cách rẽ dòng các điện trở gánh R_1 và R_2 nhiều hay ít. Khi điện áp ở cực B của T_4 tăng (ứng với điện áp Base của T_5 giảm) làm cho dòng điện cực thu tăng theo, đưa đến điện áp ở cực thu T_4 giảm xuống, nó giảm tương đương với việc tăng điện trở R_2 do đó làm tăng hệ số khuếch đại toàn bộ. Emitơ của T_4 được đấu đất.

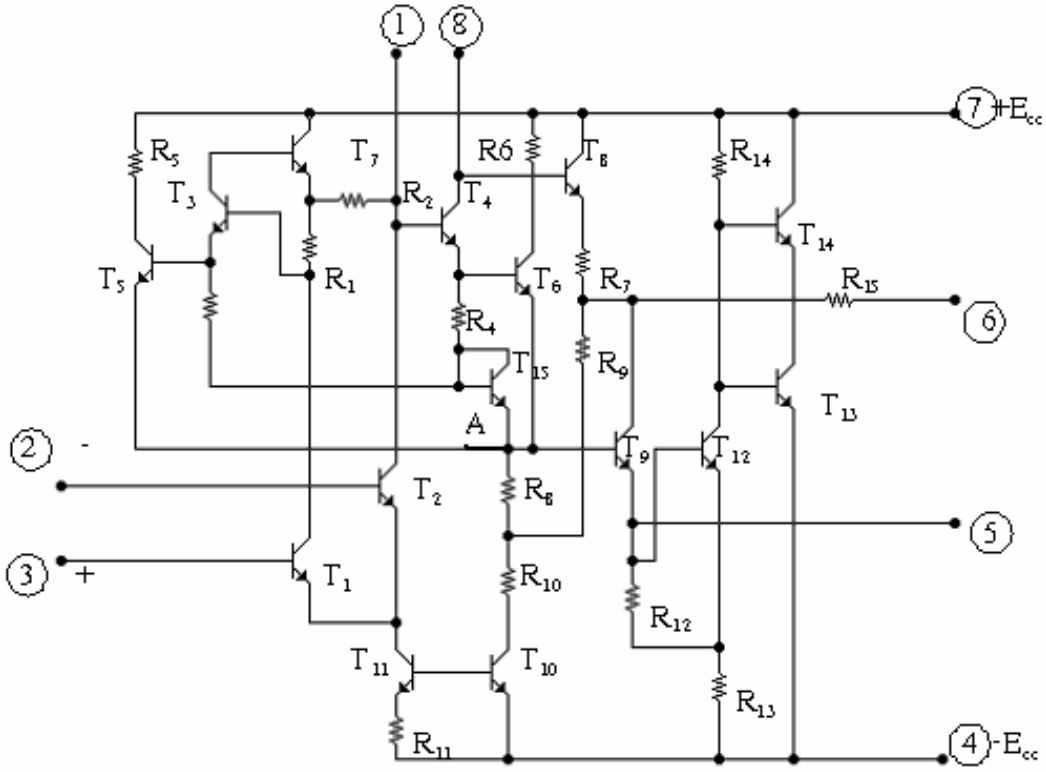
Tầng thứ ba: Bao gồm 3 transsistorr T_6, T_7 và T_8, trong đó T_6 làm mạch định mức điện áp một chiều, nghĩa là nó làm cho điện áp một chiều ở đầu ra (chân 7) bằng điện áp đất, T_8 làm nguồn dòng điện. Khi chưa có tín hiệu, điện áp của T_7 ở gần mức đất. Điện trở R_{11} nằm trong mạch cực C của T_7, nên đưa điện áp hồi tiếp dương về cực B của T_7, làm cho hệ số khuếch đại của T_7 tuy là mắc theo C chung nhưng vẫn lớn rất.

b/Vi mạch khuếch đại thuật toán loại 709:

Có sơ đồ nguyên lý như hình 1.13. Tuỳ theo hãng sản xuất nó có nhiều tên gọi khác nhau: MA 709, K1YT 53, SN 72709.... Mạch thuật toán 709 bao gồm 4 tầng chính.

***Tầng thứ nhất*:** Gồm có T_1 và T_2 là mạch khuếch đại vi sai T_{11} và T_{10} được mắc theo kiểu thiên áp dùng diode để làm nguồn dòng điện cho T_1 và T_2.

***Tầng thứ hai*:** Gồm cặp T_3, T_5 và T_4, T_6, đây cũng là tầng khuếch đại vi sai mắc theo kiểu phức hợp để tăng trở kháng vào và hệ số khuếch đại của tầng. T_{15} làm nhiệm vụ bù nhiệt cho mạch thiên áp tầng hai. Các transistor T_4 và T_6 còn hợp với R_8, R_{10} làm thành một mạch hồi tiếp âm từ cực C của T_2 về ổn định nguồn dòng điện T_{11}. Nguồn dòng điện này còn được ổn định thêm bằng mạch hồi tiếp âm từ đầu ra qua R_{15}, R_9 , R_{10} đến T_{10}. T_{15} được đấu với điểm A coi như điểm giữa. Điện thế của điểm A được ổn định bằng các mạch T_9, R_7, R_9 và T_{12}, T_{13}, R_{15}, R_9. Các transistor T_3, T, T_7 còn có tác dụng ổn định điện áp nguồn cho tầng vi sai, T_7 làm mạch lọc nguồn.



Hình 1.13: *Sơ đồ nguyên lý của vi mạch thuật toán loại 709*

***Tầng thứ ba*:** Gồm hai Transistor T_8 và T_9 trong đó T_8 mắc theo kiểu C chung và T_9 mắc theo kiểu B chung. Đây là mạch định mức điện áp để đưa mức tín hiệu (thành phần 1 chiều) xuống gần bằng - E, có như thế tín hiệu mới có thể tiếp tục khuếch đại bằng T_{12}, T_{13} và T_{14}.

***Tầng thứ tư*:** Là tầng cuối, gồm T_{12}, T_{13} và T_{14}. Đây là tầng có hồi tiếp âm sâu bằng R_{15} ,

R_{17} và T_9, hệ số khuếch đại của tầng này khoảng bằng 3 và ít phụ thuộc vào nhiệt độ .

Mạch bù tần số ở đầu vào được nối giữa cực B và C của T_4, mạch bù tạo ra một hồi tiếp âm, nó gồm một điện trở mắc nối tiếp với một tụ điện.

Mạch bù đầu ra được nối giữa C của T_9 và E của T_{13}, T_{14}.

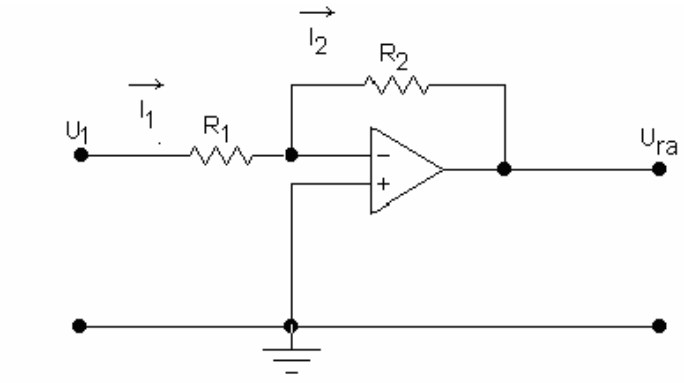
Tóm lại, qua cách phân tích sơ đồ nguyên lý của mạch khuếch đại thuật toán ở trên, đã cho ta biết nguyên lý cấu tạo chung bên trong của một mạch khuếch đại thuật toán. Tuy vậy, khi sử dụng các mạch khuếch đại thuật toán, ta không cần phải biết sơ đồ nguyên lý bên trong của mạch mà chỉ cần biết các tham số của nó gồm: sơ đồ chân, công suất, điện áp cung cấp …Các tham số này do nơi sản xuất cung cấp và ghi vào sách số liệu (data book) hoặc sổ tay hướng dẫn (handbook).

1.3.4. Các cách mắc cơ bản của vi mạch tuyến tính:

Vi mạch tuyến tính có 2 cách mắc cơ bản:

a/ Cách mắc đảo

Tín hiệu được đưa vào đầu vào đảo (-). R_2 là điện trở hồi tiếp (feedback), nó đưa một phần năng lượng từ đầu ra trở lại đầu vào. R_1 là điện trở tín hiệu. Đầu vào thuận nối đất (Hình 1.14)



Hình 1.14: *Cách mắc đảo*

Ta tìm hệ số khuếch đại điện áp của vi mạch:

Từ sơ đồ nguyên lý ta có thể viết:

$$I_1 = \frac{U_1 - U_v}{R_1} \quad \text{và} \quad I_2 = \frac{U_v - U_r}{R_2}$$

Coi mạch là lý tưởng :

$$\left.\begin{array}{l} R_V \sim \infty \\ I_v \sim 0 \end{array}\right\} \Rightarrow \quad I_1 \sim I_2$$

Do đó ta có thể viết: $\frac{U_1 - U_v}{R_1} = \frac{U_v - U_r}{R_2}$ (1.2)

Ngoài ra, xem hệ số khuếch đại là lý tưởng:

$K = \infty$, $U_v \sim 0$ Do đó (1.2) trở thành:

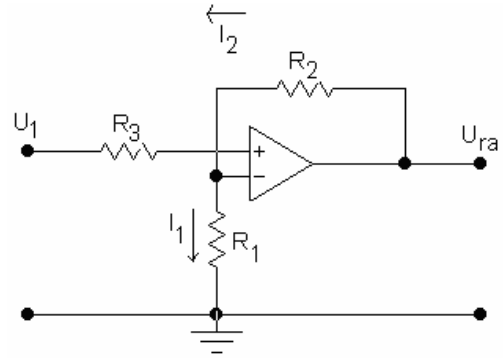
$$\frac{U_1}{R_1} = -\frac{U_{ra}}{R_2} \text{ hay } \frac{U_{ra}}{U_1} = -\frac{R_2}{R_1}$$

Vậy hệ số khuếch đại của mạch: $K_u = -\frac{R_2}{R_1}$

Dấu trừ cho biết đầu vào và đầu ra ngược pha nhau.

b/ Cách mắc thuận

Tín hiệu đưa vào đầu vào không đảo (+). Điện áp từ đầu ra đưa trở lại đầu vào đảo qua bộ chia thế gồm R_1 và R_2 (Hình 1.15) . Tín hiệu vào và ra cùng pha nhau.



Hình 1.15: *Cách mắc thuận*

Xem mạch là lý tưởng và chứng minh tương tự như trên, ta có thể viết:

$$K = \frac{U_r}{U_1} = \frac{R_1 + R_2}{R_1} = 1 + \frac{R_1}{R_2}$$

Khi cho $R_2 >> R_1$ thì hệ số khuếch đại của hai trường hợp trên là giống nhau và bằng:

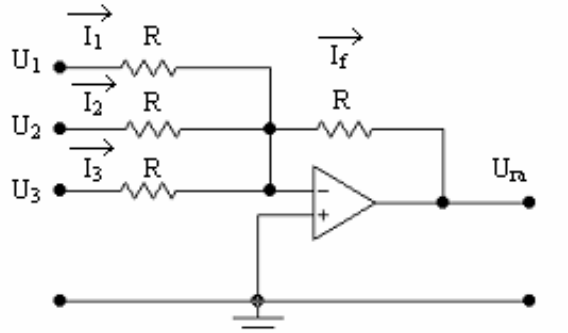
$$K_u = \frac{R_2}{R_1}$$

Tuy nhiên cách mắc đảo ổn định hơn vì có hồi tiếp âm.

1.3.5. Ứng dụng của Vi mạch thuật toán để thực hiện các phép tính cơ bản:

Sơ đồ thực hiện các phép tính cơ bản:

a/Sơ đồ thực hiện phép cộng: (Hình 1.16):



Hình 1.16: *Sơ đồ thực hiện phép cộng*

Coi dòng vào bằng 0 ta có: $I_1 + I_2 + I_3 = I_f$

Mặt khác, khi $U_v = 0$ (Coi hệ số khuếch đại của mạch là lớn vô cùng), đẳng thức trên có thể viết:

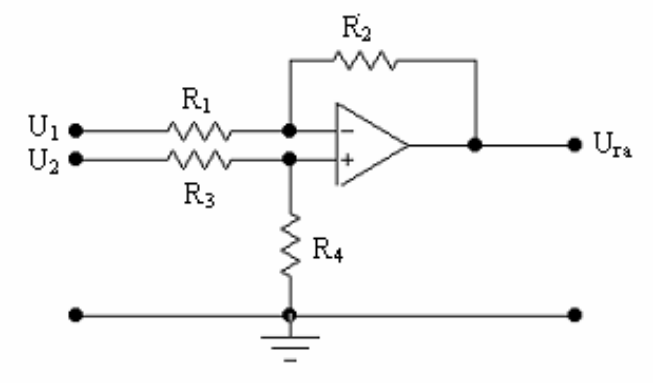
$$\frac{U_1}{R} + \frac{U_2}{R} + \frac{U_3}{R} = -\frac{U_{ra}}{R}$$

Hay : $U_{ra} = -(U_1 + U_2 + U_3)$.

Nghĩa là điện áp ra bằng tổng điện áp vào.

b/Sơ đồ thực hiện phép trừ: (Hình 1.17):

Tín hiệu được đưa vào cả hai lối vào đảo và thuận:



Hình1.17: *Sơ đồ thực hiện phép trừ*

Áp dụng các công thức tính hệ số khuếch đại trong trường hợp mắc đảo và thuận, ta có:

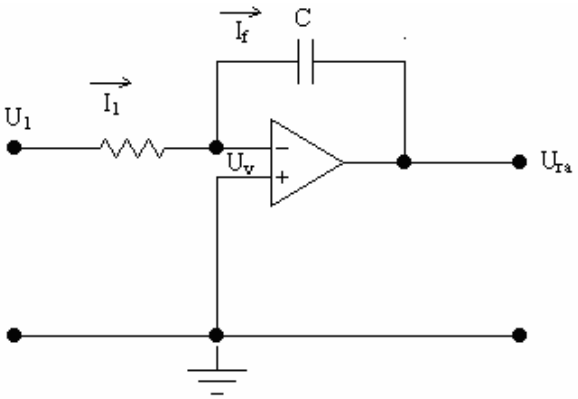
$$U_{ra} = U_2 \frac{R_4}{R_3 + R_4} \cdot \frac{R_1 + R_2}{R_1} - U_1 \frac{R_2}{R_1}$$

Nếu chọn $R_1 = R_2$, $R_3 = R_4$ thì đẳng thức trên trở thành:

$$U_{ra} = (U_2 - U_1)\frac{R_4}{R_1}$$

Nghĩa là điện áp ra tỷ lệ với hiệu điện áp vào.

c/Sơ đồ lấy tích phân: (Hình 1.18):



Hình 1.18: *Sơ đồ lấy tích phân*

Coi dòng vào bằng 0 ta có:

$$I_1 = I_f \quad (1.3)$$

Mặt khác coi $U_v = 0$ ta có:

$$I_1 = \frac{U_1}{R} \quad (1.4)$$

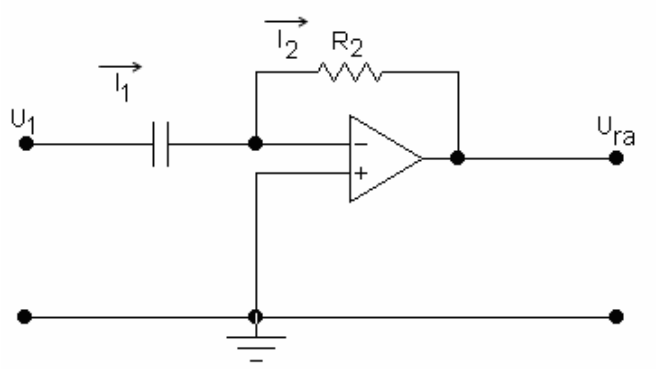
Mà: $$I_f = -C\frac{dU_{ra}}{dt} \quad (1.5)$$

Thay (1.3), (1.4) vào (1.5) ta có:

$$\frac{U_1}{R} = -C\frac{dU_{ra}}{dt}$$

$$dU_{ra} = -\frac{1}{CR}U_1 dt \Rightarrow U_{ra} = -\frac{1}{CR}\int U_1 dt$$

d/Sơ đồ lấy vi phân: (Hình 1.19)



Hình 1.19: *Sơ đồ lấy vi phân*

Ta có: $I_1 = I_f$ nhưng $I_1 = C\frac{dU_1}{dt}$

Vì $I_f = -\frac{U_{ra}}{R}$ nên

$$C\frac{dU_1}{dt} = -\frac{U_{ra}}{R} \text{ hay } U_{ra} = -CR\frac{dU_1}{dt}$$

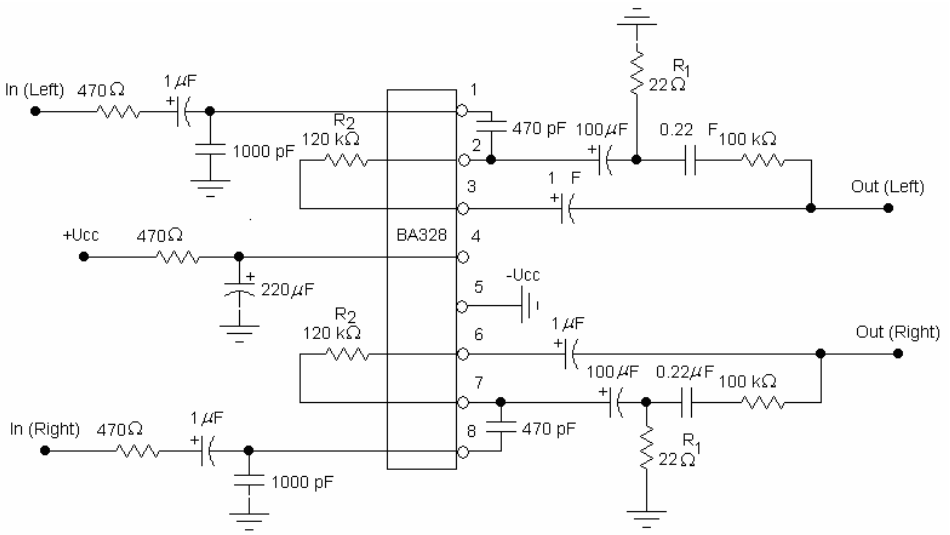
1.3.6. Ứng dụng vào các bộ khuếch đại

Vi mạch thuật toán được sử dụng rộng rãi trong các mạch khuếch đại.

a/Khuếch đại điện áp âm tần

Vi mạch BA328 được dùng khá phổ biến để khuếch đại điện áp âm tần (Hình 1.20).

BA328 thuộc loại khuếch đại hai kênh (stereo). Điện trở R_1 R_2 quyết định hệ số khuếch đại của Vi mạch. Thường ta giữ nguyên R_2 chỉ thay đổi R_1 để có hệ số khuếch đại theo ý muốn. Nếu ta tăng hệ số khuếch đại lên quá mức sẽ gây ra méo tín hiệu.



Hình 1.20: *Khuếch đại điện áp âm tần dùng vi mạch khuếch đại thuật toán BA328*

R_3 và tụ C tạo thành một vòng hồi tiếp âm điện áp để sửa đổi đường đặc trưng tần số. Thay đổi trị số của điện trở R_3 ta có âm thanh lợi trầm hoặc lợi bổng theo ý muốn. R_3 càng thấp, hồi tiếp âm càng sâu, mạch càng lợi trầm.

Tín hiệu cần khuếch đại có biên độ rất bé được đưa vào chân 8. Tín hiệu sau khi khuếch đại được lấy ra ở chân 6.

Điện áp cung cấp cho cả hai kênh được đưa vào chân 4 ($+V_{cc}$) và chân 5 ($-V_{cc}$). Chân 5 đồng thời cũng là đất của mạch.

Nguồn cung cấp 9 VDC được lọc bằng R và C_1.

Do độ nhạy cao và tín hiệu đầu vào rất bé, ta phải dùng dây có giáp bọc để dẫn tín hiệu vào và ra nhằm loại bỏ nhiễu.

b/Khuếch đại công suất âm tần

Để khuếch đại công suất âm tần ta có thể dùng vi mạch LA4440. Vi mạch LA4440 thuộc loại khuếch đại hai kênh, có sơ đồ khối như hình 1.21. Tuy nhiên vi mạch LA4440 có thể dùng để khuếch đại công suất âm tần một kênh.

Khi khuếch đại hai kênh (stereo), công suất ra danh định 6W x 2. Khi sử dụng để khuếch đại một kênh (mono), công suất ra danh định lên đến gần 20 W. Nguồn cung cấp 12V đến 15V.

Chân 1và chân 7: Đầu vào thứ hai của LA4440.

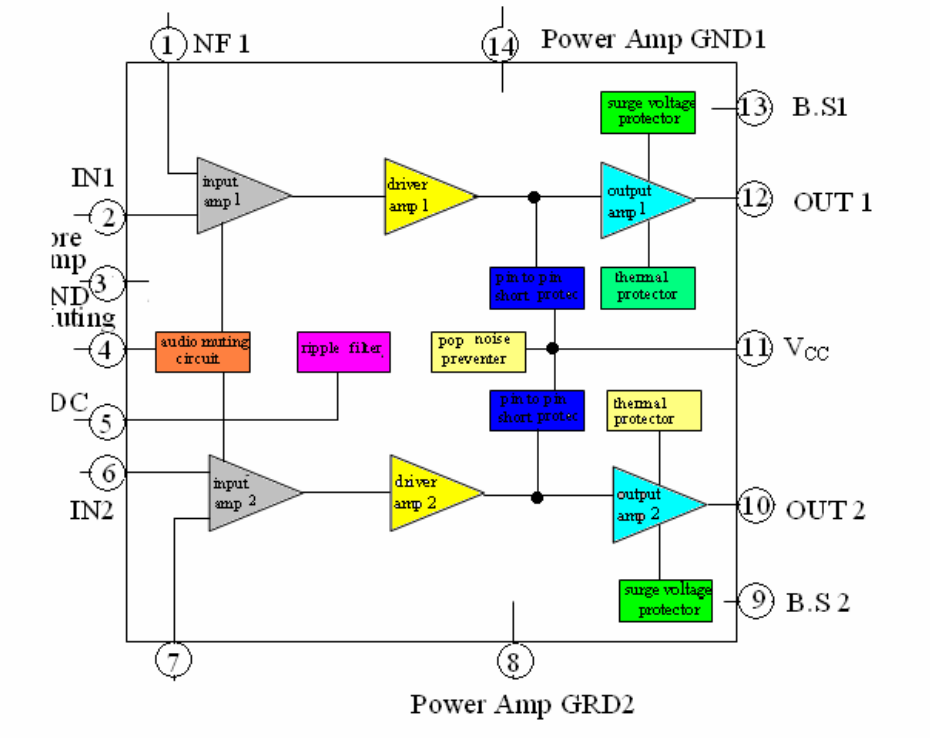
Chân 2 và chân 6: Đầu vào của LA4440.Tín hiệu âm tần cần khuếch đại được đưa vào

Chân 4: Đầu ra Mute

Chân 5: lọc Ripple filter

Chân 8, 14: Điểm mass của vi mạch

Chân 9 và 13: Bảo vệ quá áp

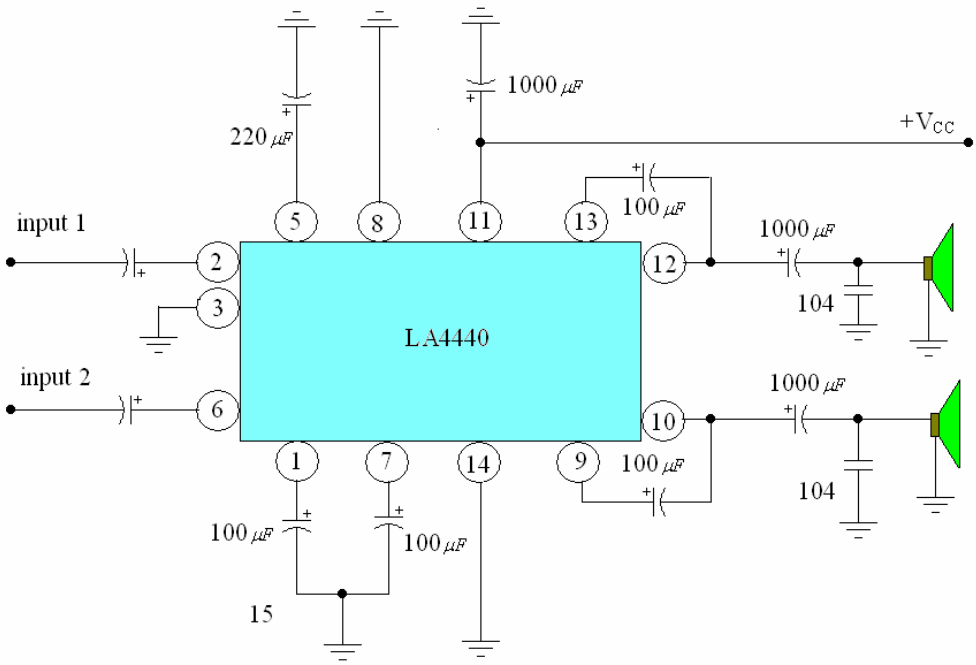


Hình 1.21: *Sơ đồ khối cấu trúc bên trong LA4440*

Chân 10 và 12: Đầu ra của vi mạch

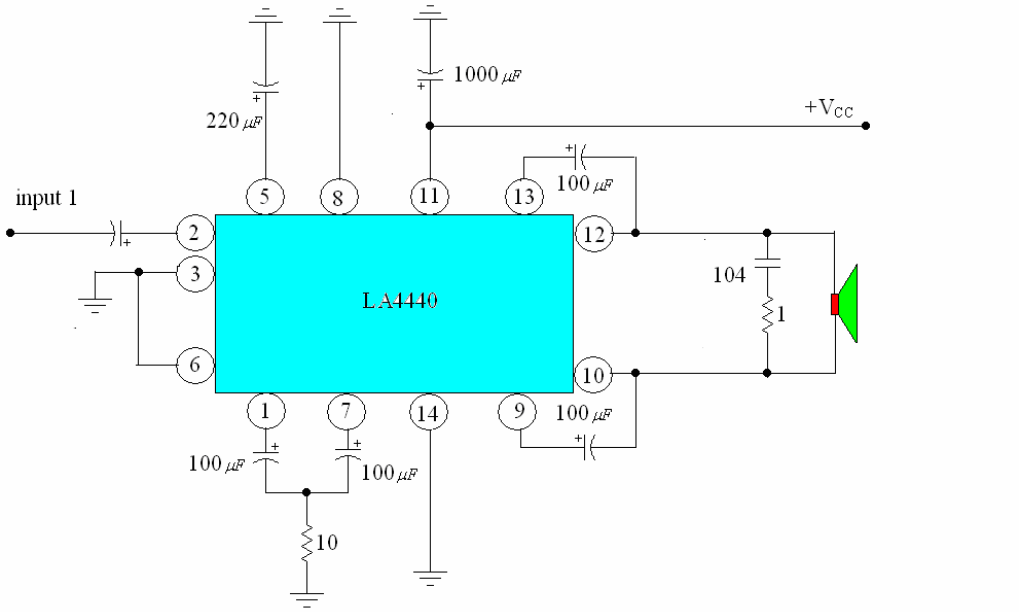
Chân 11: Nhận điện áp cung cấp + V_{CC}

Sơ đồ nguyên lý mạch khuếch đại công suất hai kênh (Hình 1.22)



Hình 1.22: *Khuếch đại công suất âm tần dùng vi mạch LA4440 mắc stereo*

Khi muốn LA4440 cho ta một công suất lớn ở loa, ta mắc theo sơ đồ nguyên lý mạch khuếch đại công suất một kênh (Hình 1.23)



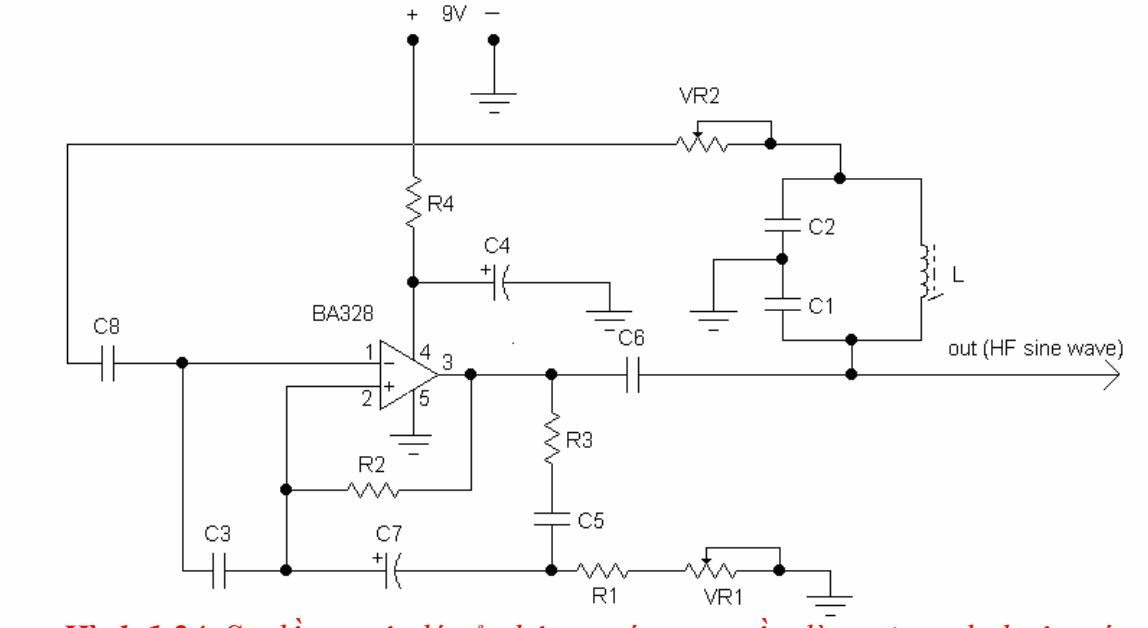
Hình 1.23: *Khuếch đại công suất âm tần dùng vi mạch LA4440 mắc mono*

1.3.7. Bộ tạo sóng điện hình sin dùng vi mạch tuyến tính:

a/Tạo sóng hình sin cao tần

Vi mạch thuật toán có thể làm bộ khuếch đại trong khâu khuếch đại của bộ tạo sóng hình sin cao tần hoặc âm tần.

Hình 1.24 là mạch tạo sóng cao tần dùng vi mạch thuật toán BA328, BA328 là một loại vi mạch rất thông dụng, rất dễ kiếm trên thị trường, hoạt động rất tin cậy.



Hình 1.24: *Sơ đồ nguyên lý của bộ tạo sóng cao tần dùng vi mạch thuật toán*

Tác dụng của từng linh kiện:

R_1, R_2,VR_1 quyết định hệ số khuếch đại của vi mạch

R_3, C_5 tạo thành mạch hồi tiếp âm chọn lọc để thực hiện việc hồi tiếp âm ở tần số cao. C_2,C_1 L tạo thành khung dao động Colpitts.

C_3 dẫn điện áp từ đầu ra (chân 3) của vi mạch, qua khung dao động Colpitts về đầu vào (chân 1) để thực hiện hồi tiếp. Hai điện áp này cùng pha nên hồi tiếp là hồi tiếp dương.

R_4 và C_4 tạo thành bộ lọc điện cho nguồn cung cấp

BA 328: Vi mạch thuật toán, dùng để khuếch đại tín hiệu.

Điện áp cung cấp 9VDC

Hoạt động của sơ đồ:

Bộ tạo sóng cao tần trên thuộc loại 3 điểm điện dung. Khi đóng mạch điện, khung dao động hoạt động và tạo ra dao động hình sin có tần số xác định bằng công thức:

$$f_0 = \frac{1}{2\Pi\sqrt{L\dfrac{C_1C_2}{C_1+C_2}}}$$

Dao động này là dao động tắt dần, nếu chúng ta không có biện pháp bù đắp năng lượng bị tiêu hao dưới tác dụng nhiệt của điện trở làm nên cuộn dây L.

Trước khi tắt một phần năng lượng được đưa về đầu vào thuận của vi mạch BA328, vì điện áp hồi tiếp và điện áp tín hiệu cùng pha (mắc thuận) nên là hồi tiếp dương, đảm bảo điều kiện duy trì dao động. Hệ số hồi tiếp quyết định bởi tỷ số C_1 và C_2

$$\beta = \frac{C_1}{C_2}$$

Điện áp hồi tiếp được điều chỉnh bởi VR_2 nhằm bảo đảm điều kiện $K\beta = 1$. Khi điều kiện này được đáp ứng, sóng điện tạo ra có dạng hình sin rất đều đặn và đẹp. Mạch hoạt động ổn định.

Để mạch khuếch đại chạy ổn định và khuếch đại trung thực, một mạch hồi tiếp âm chọn lọc gồm R_3C_5 được mắc vào hai chân 2 và 3 của vi mạch. Hồi tiếp âm càng sâu Vi mạch BA328 càng khuếch đại trung thực, tuy nhiên ta chỉ có thể thực hiện âm hồi ở một mức nào đó vì còn phải đảm bảo khâu dương hồi nhằm duy trì dao động.

Dao động hình sin cao tần được lấy ra ở chân 3 là đầu ra của BA328.

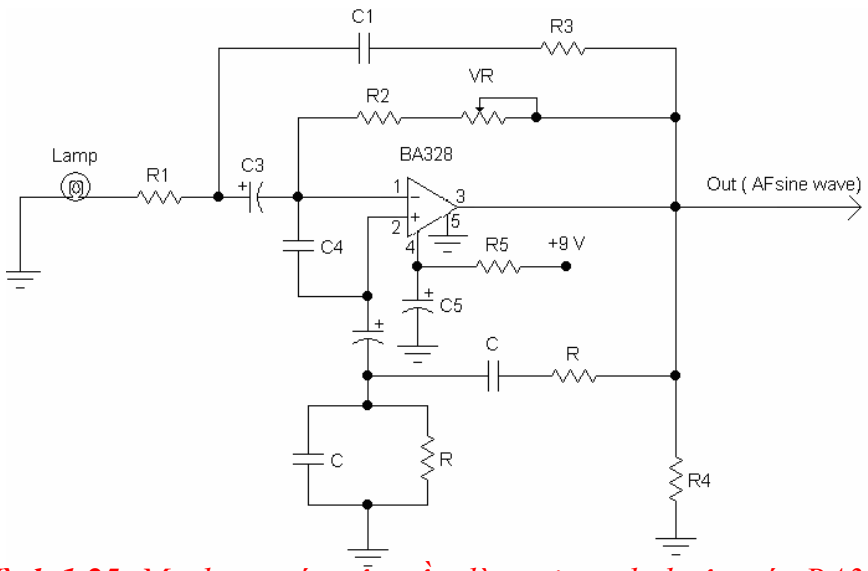
b/Tạo sóng hình sin âm tần:

Hình 1.25 là sơ đồ nguyên lý bộ tạo sóng hình sin âm tần dùng vi mạch tuyến tính BA 328 trong khâu khuếch đại. Trong đó:

BA 328: Vi mạch thuật toán, dùng để khuếch đại tín hiệu.

R_1,R_2, VR, đèn báo: quyết định hệ số khuếch đại của vi mạch.

R_3, C_1 tạo thành mạch hồi tiếp âm chọn lọc để thực hiện hồi tiếp âm ở tần số cao, R_3 càng



Hình 1.25: *Mạch tạo sóng âm tần dùng vi mạch thuật toán BA328*

bé thì hồi tiếp âm càng sâu. Tuy nhiên, ta không thể tuỳ tiện hạ trị số của R_3 vì phải bảo đảm khâu dương hồi để duy trì dao động.

Hai cặp RC tạo thành cầu Wien.

C_2 dẫn điện áp hồi tiếp từ đầu ra của vi mạch thuật toán về đầu vào để thực hiện hồi tiếp dương.

R_5 và C_5 cùng với tụ lọc nguồn tạo thành bộ lọc hình π để lọc điện.

Đèn báo tham gia khâu quyết định hệ số khuếch đại, thực hiện việc thay đổi hệ số khuếch đại của vi mạch BA328 một cách tự động nhằm ổn định biên độ sóng điện tạo ra.

Mạch hoạt động như sau:

Khi đóng mạch điện, cầu Wien sẽ tạo dao động âm tần có tần số quyết định bằng công thức: $f_0 = \frac{1}{2\pi RC}$ Dao động sẽ tắt dần, tuy nhiên trước khi tắt nó được đưa vào đầu vào của BA328 qua tụ C_2, dao động được BA328 khuếch đại lên sau đó đưa về cầu Wien để duy trì dao động.

Sóng điện hình sin âm tần được lấy ra ở đầu ra của BA328. Khi dao động có biên độ lớn hơn thiết kế (xác định từ trước bởi hệ số khuếch đại của BA328), cường độ đi qua đèn báo lớn hơn, nhiệt độ tăng lên, điện trở của dây tóc đèn báo tăng theo làm giảm hệ số khuếch đại, biên độ sóng điện tạo ra giảm xuống. Ngược lại, khi dao động có biên độ nhỏ hơn thiết kế, cường độ đi qua đèn báo giảm xuống, nhiệt độ giảm, điện trở của nó giảm theo làm tăng hệ số khuếch đại, biên độ sóng điện ra tăng lên.

1.4.VI MẠCH LOGIC

1.4.1. Tổng quan:

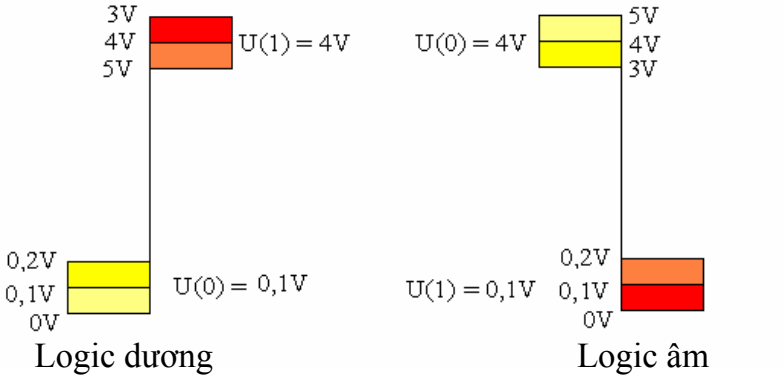
Vi mạch logic còn gọi là vi mạch số, làm việc theo cơ chế nhị phân. Nghĩa là chúng chỉ

tồn tại dưới hai trạng thái dẫn hoàn toàn và tắt hẳn, được ký hiệu (1) và (0).

Trong hệ thống logic, một trong hai mức điện áp nói trên được gọi là một bit.

Nếu chúng ta quy ước một trong hai mức trên mức nào dương hơn là 1 và mức kia là 0, ta có hệ thống logic dương ngược lại là logic âm (Hình 1.26) .

Vi mạch logic được phân ra thành nhiều họ: RTL, DTL, TTL, ECL, CMOS. Trong đó họ TTL được dùng khá phổ biến.



Hình 1.26: *Các mức logic*

Khi sử dụng vi mạch logic ta chú ý đến những đặc trưng sau :

a/*Hệ số chịu tải:* Đó là số cổng cực đại có thể mắc đồng thời ở lối ra của cổng đã cho. Hệ số này càng lớn thì khả năng logic của mạch càng cao.

b/H*ệ số ghép mạch lối vào*: Lối vào cực đại của cổng đã cho.

c/*Thời gian trễ* là thời gian cần thiết để truyền xung qua mạch

d/ *Thời gian đóng mạch* là thời gian kể từ lúc lối vào đạt một giá trị nào đó trên mức 0 đến lúc lối ra đạt một giá trị nào đó dưới mức 1.

e/*Thời gian ngắt mạch* là khoảng thời gian kể từ lúc lối vào đạt một giá trị nào đó dưới mức 1, đến lúc lối ra đạt giá trị nào đó trên mức 0

1.4.2. Một số lưu ý khi sử dụng Vi mạch số

Một thiết bị sẽ sử dụng nhiều loại, nhiều họ vi mạch số khác nhau, các tham số của các vi mạch này cũng rất khác nhau. Để thiết bị được sử dụng lâu bền, hoạt động ổn định ta cần phải lưu ý khi cần phối ghép các vi mạch số với nhau.

Trong thực tế sử dụng, một số cổng logic trong vi mạch số không được sử dụng đến, việc dư thừa này lại hay xảy ra, Thí dụ: một mạch số cần 3 cổng NAND hai lối vào. Không có vi mạch số loại 3 cổng NAND, ta phải dùng loại 4 cổng NAND hai lối vào. Như vậy, dư một cổng. Cổng dư thừa này do không sử dụng lại gây trở ngại cho hoạt động của toàn hệ thống. Ta phải xử lý việc dư thừa này. Theo các cách sau:

a/ Nối đầu vào thừa đến $+V_{CC}$, V_{DD}, hoặc $-V_{CC}$, V_{SS} sao cho chức năng logic ban đầu của cổng vẫn không thay đổi.

b/ Nối các đầu vào của các cổng thừa đến $+V_{CC}$, V_{DD}, hoặc $-V_{CC}$, V_{SS} sao cho đầu ra của

nó luôn ở trở thành logic cao H, nghĩa là làm cho nó tiêu thụ công suất ít đi.

Làm như trên, ngoài việc giảm công suất tiêu thụ của vi mạch số, ta còn thực hiện được việc chống nhiễu cho toàn hệ thống.

1.4.3 Một số IC số thông dụng

Sau đây là một số vi mạch số thông dụng, thường gặp trong các mạch số, khi thực hiện các bài tập, chúng tôi cũng thường sử dụng các vi mạch này.

Bảng 1.2: *Một số vi mạch số thông dụng*

TTL			MOS			Cổng
Ký hiệu	Số cổng/ IC	Số lối vào/cổng	Ký hiệu	Số cổng/ IC	Số lối vào /cổng	
74LS04	6	1	4069B	8		NOT
74LS05	6	1				NOT hở cực C
74LS15	3	3	4073B	3	3	AND (C hở)
74LS21	2	4	4081B	4	2	AND
74LS32	4	2	4071B	4	2	OR
74LS00	4	2	4012B	4	2	NAND
74LS01	4	2	4023B	2	2	NAND (C hở)
74LS03	4	2	4068B	3	3	NAND (C hở)
74LS10	3	3		1	8	NAND
74LS11	3	3				NAND
74LS12	3	3				NAND C hở
74LS20	2	4				NAND
74LS22	2	4				NAND hở cực C
74LS30	1	8				NAND
74LS37	4	2				Bộ đệm NAND
74LS38	4	2				Bộ đệm NAND C hở
74LS40	4	2				Bộ đệm NAND
74LS133	1	13				NAND
74LS27	3	3				NOR
74LS28	4	2				Bộ đệmNOR
74LS33	4	2				Bộ đệm NOR C hở
74LS86	4	2				XOR
74LS386	4	2	4070B	4	2	XOR

74LS136	4	2				XOR C hở
74LS13	2	4				NAND + Schmitt
74LS14	6	1				NAND + Schmitt
74LS132	4	2				NAND + Schmitt
74LS125A	4		4093B	4	2	Đệm 3 tt với $\overline{E}$
74LS126A	4		4053B	6		Đệm 3 tt với E

CHƯƠNG 2 CƠ SỞ TOÁN HỌC CỦA ĐIỆN TỬ SỐ

2.1 KHÁI NIỆM VỀ THÔNG TIN VÀ MÃ

2.1.1 Thông tin :

Khái niệm về thông tin rất rộng, ở đây ta quan tâm chủ yếu đến khái niệm thông tin hẹp hơn theo lý thuyết thông tin thống kê (hay lý thuyết thông tin truyền tín hiệu).

Để đánh giá số lượng thông tin người ta đưa ra đơn vị thông tin là bit (***bi***nary digi***t***).

Bit chính là số lượng thông tin cần thiết để nhận biết một trong hai trạng thái có xác suất xuất hiện như nhau ($p_1 = p_2 = 1/2$) của một đối tượng hai trạng thái nào đó (có khi còn gọi là đối tượng nhị phân).

Trên cơ sở bit thông tin, người ta còn đưa ra các đơn vị khác như byte:

1 byte = 8 bit, 1 kB (kilobyte) = 1024 byte

1 Megabyte = 1.048. 576 byte.

Byte là nhóm tám bit kề liền nhau, tạo thành đơn vị dữ liệu cơ sở của máy tính cá nhân. Do được lưu trữ tương đương một ký tự, nên byte cũng là đơn vị cơ sở để đo sức chứa của máy tính. Cấu trúc máy tính (đối với hầu hết các bộ phận).

Dựa trên cơ sở số nhị phân, nên các byte được tính theo luỹ thừa của 2. Thuật ngữ kilô (trong kilô byte) và mega (trong megabyte) thường được dùng làm bội số trong việc đếm byte, nhưng không được chính xác lắm vì chúng có nguồn gốc thập phân (cơ số 10). Nhiều nhà khoa về máy tính đã chỉ trích các thuật ngữ này, tuy nhiên những thuật ngữ này vẫn được dùng vì chúng cho những khái niệm quen thuộc theo cách đếm thập phân và thuận lợi trong việc đo lường dung lượng bộ nhớ.

Để đo tốc độ truyền tin người ta dùng đơn vị baud :

1 baud = 1 bit / giây .

Dữ liệu là những ký hiệu Vật lý dùng để biểu diễn thông tin nhằm mục đích lưu trữ, trao đổi và xử lý.

2.1.2.Phân loại thông tin

Có nhiều phương pháp phân loại thông tin, ở đây ta chỉ đề cập đến phương pháp dựa vào đặc tính thời gian của các dữ liệu biểu diễn thông tin truyền đi. Đó là đặc tính rời rạc và đặc tính liên tục của tín hiệu. Tín hiệu được hiểu là sự thực hiện Vật lý các dữ liệu. Do đó ta có thông tin liên tục và thông tin rời rạc.

a/Thông tin liên tục: Thông tin liên tục được biểu diễn nhờ 1 hệ thống các đại lượng mà giá trị của chúng có đặc tính liên tục theo thời gian.

Dạng thông tin này thường gặp trong các máy tính tương tự hoặc các thiết bị xử lý thông tin liên tục như xử lý hình ảnh, âm thanh thông thường (không qua bộ biến đổi tương tự - số). Dạng thông tin tương tự thường gặp trong thực tế cuộc sống.

bThông tin rời rạc: Thông tin rời rạc được biểu diễn bằng một hệ thống các đại lượng mà giá trị của chúng có đặc tính rời rạc.

Dạng thông tin này thường gặp trong máy tính số cũng như thiết bị điều khiển số, đo lường số, âm thanh số ...

2.1.3. Mã (code)

Các thiết bị điện tử số là những thiết bị xử lý thông tin. Các ký hiệu Vật lý dùng để biểu diễn thông tin nhằm mục đich lưu trữ, trao đổi và xử lý ta gọi là dữ liệu.

Thông tin có thể diễn đạt ở dạng thông thường thông qua các ký hiệu thông dụng (chữ số, ký tự ...) mà mọi người đều hiểu. Đồng thời thông tin cũng có thể được biểu diễn dưới dạng đặc biệt bất kỳ khác dạng thông thường, ta gọi đó là mã. Mục đích để bảo mật, dễ tính toán và sửa chửa sai sót .

Có nhiều loại mã khác nhau tùy theo nhu cầu sử dụng. Trong điện tử số thường dùng mã số. Đó là sự tương ứng giữa một ký hiệu của bộ chữ cái với tập hợp các con số của một hệ đếm nào đó. Việc chuyển từ các ký hiệu thông dụng sang các mã gọi là lập mã (còn gọi là mã hóa).

Trong thiết bị số ta dùng hai ký hiệu 0 và 1 gọi là mã nhị phân. Ngoài ra còn có các mã khác như thập phân, bát phân, thập lục phân. Chúng ta thường gặp một số mã sau:

a/ Mã BCD: Đây là một mã nhị phân tự nhiên, trong đó các chữ số thập phân từ 0 đến 9 được biến điệu dưới dạng 4 bit nhị phân tương đương của nó. Đây là một mã rất thuận tiện và hữu ích cho các phép vào ra trong mạch số.

b/ Mã Gray: Rất thuận lợi vì để biểu diễn hai số liên tiếp ta chỉ cần thay đổi giá trị của một bit.

c/Mã ASSCII: Mã trao đổi thông tin theo tiêu chuẩn Mỹ.

2.2. CÁC HỆ THỐNG ĐẾM SỐ :

2.2.1. Định nghĩa :

Hệ thống đếm là tổ hợp các quy tắc gọi và biểu diễn các con số có giá trị xác định. Nhờ các hệ đếm, ta có thể biểu diễn một con số bất kỳ theo các hệ thống số khác nhau. Hiện nay tồn tại nhiều hệ thống đếm nhưng chỉ có hai loại :

a/Hệ đếm không theo vị trí :

Hệ thống đếm mà giá trị về mặt số lượng của chữ số không phụ thuộc vào vị trí của nó nằm trong con số.

Thí dụ: Hệ thống đếm theo chữ số La mã

X = 10 đơn vị

XII = 12 đơn vị

XXIX = 29 đơn vị

Giá trị của các chữ số X ở các con số trên không phụ thuộc vào vị trí của nó nằm trong con số. Khi biểu diễn những con số lớn hệ đếm này trở nên kồng kềnh, do đó hiện nay người ta

ít dùng. Ngoài ra hệ đếm Lamã chưa có số 0 nên việc xử lý số học gặp nhiều khó khăn.

b/Hệ đếm theo vị trí:

Hệ thống đếm mà trong đó giá trị mỗi chữ số không những phụ thuộc vào bản thân chữ số mà còn phụ thuộc vào vị trí của nó trong con số .

Hệ đếm theo vị trí thông dụng nhất hiện nay là hệ đếm thập phân .

Thí dụ : 1234 : Chữ số 4 biểu diễn giá trị 4 đơn vị

5422 : Chữ số 4 biểu điễn giá trị 4 x 10^2 đơn vị

4321 : Chử số 4 biểu diễn giá trị 4 x 10^3đơn vị .

Mỗi hệ đếm theo vị trí có tập hợp các chữ số khác nhau cần thiết để biểu diễn một con số bất kỳ. Số lượng các chữ số khác nhau đó gọi là cơ số của hệ đếm :

Hệ đếm thập phân (Decimal) :{0,1,2,3,4,5,6,7,8,9}

Hệ đếm bát phân (Octal) : {0,1,2,3,4,5,6,7}

Hê đếm thập lục phân (Hexan) :{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Hệ đếm nhị phân (Binary) : {0,1}

Hệ thập phân được người Hindu thiết lập vào khoảng thế kỷ thứ 5 và được người Arab bổ sung thêm ký hiệu 0. Đây là hệ đếm quen thuộc nhất của con người.

Trong các hệ đếm trên, hệ nhị phân có nhiều ưu điểm vượt trội khi cần tính toán bằng các thiết bị điện tử, vì hệ đếm này chỉ có hai chữ số là 1 và 0 ứng với hai trạng thái ổn định của một Trigger hoặc ứng với hai trạng thái thông hoặc không thông của mạch điện.

Việc chuyển trạng thái từ 0 lên 1 và ngược lại được thực hiện rất nhanh, nên tốc độ tính toán trong hệ nhi phân nhanh hơn các hệ khác. Độ tin cậy cao.

Các mạch điện tử trong máy tính sẽ phát hiện sự khác nhau giữa hai trạng thái (dòng điện mức cao, dòng điện mức thấp) và biểu diễn các trạng thái đó dưới dạng một trong hai số nhị phân 1 hoặc 0. Các đơn vị cơ bản cao/thấp, đúng/sai, có/không….được gọi là bit Việc chế tạo một mạch điện tử tin cậy có thể phân biệt được sự khác nhau giữa 1 và 0 là tương đối dễ dàng và rẻ tiền, cho nên máy tính có khả năng xử lý nội bộ các thông tin nhị phân một cách rất chính xác, theo tiêu chuẩn, nó mắc ít hơn một lỗi nội bộ trong 100 tỉ thao tác xử lý.

Nhược điểm của hệ là biểu diễn dài, mất nhiều thời gian viết và đọc.

Các máy tính điện tử đều dùng hệ đếm cơ số hai để lưu trữ và biểu diễn thông tin.

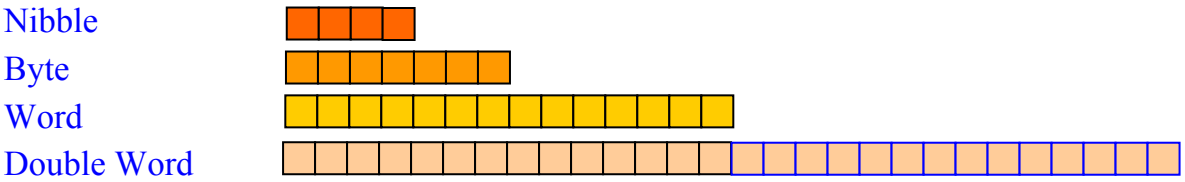
Một nhóm gồm 4 bit các số nhị phân được gọi là một nibble *Thí dụ:* 1011

Một nhóm gồm 8 bit được gọi là một byte *Thí dụ:* 10111101

Một nhóm gồm 16 bit được gọi là một từ (word) *Thí dụ:* 1011111000111100

Một nhóm gồm 32 bit được gọi là một từ kép (double word)

Thí dụ: 10111110001111001011111000111100



Hình 2.1. *Các đơn vị đo độ dài của hệ nhị phân dẫn xuất từ bit*

Chữ số đầu tiên bên trái con số của hệ nhị phân được gọi là bit có nghĩa lớn nhất (MSB *Most significant bit*), bit tận cùng bên phải con số của hệ nhị phân được gọi là bit có nghĩa bé nhất (LSB *Least significant bit*)

2.2.2. Nguyên lý chung của các hệ đếm :

Ví dụ một số A có 4 chữ số a_i được biểu diễn trong hệ đếm có cơ số là B:

$A(B) = a_3a_2a_1a_0$

Mỗi chữ số a_i có thể có giá trị giữa 0 và (B-1). Ngoài ra chữ số a_i có trọng lượng B_i. Giá trị của A là tổng các tích của mỗi chữ số a_i với trọng lượng của nó.

Một cách tổng quát ta có thể viết:

$$A = a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + \ldots + a_oB^o + a_{-1}B^{-1} + a_{-2}B^{-2} + \ldots$$

Thí dụ:

Số 7550,51 trong hệ đếm 10 :

$7550,51 = 7.10^3 + 5.10^2 + 5.10^1 + 0.\ 10^0 + 5.10^{-1} + 1.10^{-2}$

Số 101101,01 trong hệ đếm nhị phân:

$101101,01 = 1.2^5 + 0 + 1.2^3 + 1.2^2 + 0 + 1.2^0 + 0 + 1.2^{-2} = 45,25$

Số 7550 trong hệ 8:

$7550 = 7.8^3 + 5.8^2 + 5.8^1 + 0 = 3944$

CAFÉ trong hệ đếm 16 :

$12.16^3 + 10.16^2 + 15.16 + 14 = 51966$

Nhờ các hệ thống đếm ta có thể biểu diễn một con số bất kỳ theo các hệ thống các chữ số khác nhau.

Ta có bảng tương đương giữa các hệ đếm (Bảng 2.1):

Bảng 2.1: *Tương đương giữa các hệ đếm*

Hệ đếm 10	Hệ đếm 2	Hệ đếm 8	Hệ đếm16
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5

6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

2.2.3. Phương pháp chuyển đổi giữa các hệ đếm :

Ta có thể chuyển đổi tương đương giữa các hệ đếm:

a/Chuyển từ hệ đếm 8 sang hệ đếm 2

Muốn chuyển một số từ hệ đếm 8 sang hệ đếm 2 ta chỉ việc thay đổi mỗi chữ số theo hệ đếm 8 bằng một số tương đương theo hệ đếm 2 với 3 cột số:

$(650,3)_8 = (110\ \ 101\ \ 000\ ,\ 011)_2$

b/Chuyển từ hệ đếm 16 sang hệ đếm 2

Muốn chuyển một số từ hệ đếm 16 sang hệ đếm 2 ta chỉ việc thay mỗi chữ số theo hệ đếm 16 bằng một số tương đương theo hệ đếm 2 với 4 cột số.

$(B7A,D5)_{16} = (1011\ \ 0111\ 1010\ ,\ 1101\ \ 0101)_2$

c/Chuyển từ hệ đếm 2 sang hệ đếm 8 hoặc 16

Muốn chuyển một số từ hệ đếm 2 sang hệ đếm 8 hoặc 16 ta chia số nhị phân thành từng nhóm 3 cột số (hoăc 4 cột số), lầy từ dấu phẩy sang trái và sang phải. Các nhóm cuối cùng bên trái và bên phải nếu chưa đủ 3 (hoặc 4) cột số thì bổ sung thêm bằng các con số 0. Thay mỗi nhóm 3 cột số (hay 4 cột số) bằng một chữ số theo hệ 8 (hay 16).

$(1011101110,1011)_2 = 010\ \ 111\ \ 001\ \ 101,\ \ 101\ \ 100 = (2715,54)_8$

$(1101111011,101011)_2 = 0110\ \ 1111\ \ 0111,\ 1010\ \ 1100 = (6F7,AC)_{16}$

d/Chuyển từ hệ đếm 10 sang hệ đếm 2

Muốn chuyển một số từ hệ đếm 10 sang hệ đếm 2, ta chia liên tiếp số đó và thương số nhận được cho 2, cho đến khi thương số nhận được bé hơn 2 (1 hoặc 0). Thương số cuối cùng là chữ số bậc cao nhất của hệ đếm 2, các chữ số còn lại là số dư của các lần chia trước nhưng theo chiều ngược lại..

97 : 2 = 48 dư 1

48 : 2 = 24 dư 0
24 : 2 = 12 dư 0
12 : 2 = 6 dư 0
6 : 2 = 3 dư 0
3 : 2 = 1 dư 1
Vậy : $(97)_{10} = (1100001)_2$

e/Chuyển từ hệ đếm 2 sang hệ đếm 10

$(1101,11)_2 = 1.2^3 + 1.2^2 + 0 + 1.2^0 + 1.2^{-1} + 1.2^{-2} = (13,75)_{10}$

f/Chuyển từ hệ đếm 16 sang hệ đếm 10

$(B26E)_{16} = 11.16^3 + 2.16^2 + 6.16^1 + 14 = (45678)_{10}$

g/Chuyển đổi giữa hai hệ đếm bất kỳ

Giả sử ta cần chuyển chữ số x từ hệ đếm cơ số B sang hệ đếm cơ số C, trên cơ sở thực hiện các phép tính cần thiết theo hệ đếm cơ số C:

Trước hết, ta biểu diễn x dưới dạng:

$$x = a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + \ldots + a_oB^o + a_{-1}B^{-1} + a_{-2}B^{-2} + \ldots$$

Sau đó biểu diễn B và các a_i theo hệ đếm cơ số C đồng thời thực hiện các phép tính cần thiết theo hệ đếm cơ số C để tính tổng .

Thí dụ:

Chuyển sang hệ đếm 10 các số sau :

$x = (3A7,C)_{16} = 3.16^2 + 10.16^1 + 7.16^0 + 12.16^{-1} = (935,75)_{10}$

$x = (1101)_2 = 1.2^3 + 1.2^2 + 0 + 1 = (13)_{10}$

Nếu muốn chuyển một số x từ hệ đếm B sang hê đếm cơ số C, trên cơ sở thực hiện các phép tính cần thiết theo hệ đếm cơ số B thì ta phải phân biệt hai trường hợp: x là số nguyên và x là số phân:

1/ Khi x là số nguyên ta chia liên tiếp số x và thương số nhận được cho cơ số C đến khi thương số nhận được bé hơn C. Thương số cuối cùng chính là chữ số bậc cao nhất của x biểu diễn theo hệ đếm C, còn các chữ số tiếp theo sẽ là các số dư của phép chia trước đó nhưng theo chiều ngược lại.

Thí dụ 1: Chuyển số 57 hệ 10 sang hệ nhị phân

57 : 2 = 28 dư 1
28 : 2 = 14 dư 0
14 : 2 = 7 dư 0
7 : 2 = 3 dư 1

3 : 2 = 1 dư 1

Vậy : $(57)_{10} = (111001)_2$

Thí dụ 2: Chuyển số 3287 hệ 10 sang hệ 8:

3287: 8 = 410 dư 7

410: 8 = 51 dư 2

51: 8 = 6 dư 3

Vậy : $(3287)_{10} = (6327)_8$

Thí dụ 3: Chuyển số 51966 hệ 10 sang hệ 16:

51966: 16 = 3247 dư 14

3247: 16 = 202 dư 15

202: 16 = 12 dư 10

Vậy : $(51966)_{10} = (CAFE)_{16}$

2/ Khi x là số phân ta nhân liên tiếp x và tích số nhận được sau mỗi phép nhân với cơ số C. Các phần nguyên nhận được sau mỗi phép nhân cho ta tuần tự số biểu diễn số phân x theo hệ đếm C. Nếu tích số lớn hơn 1 thì lấy tích số trừ đi 1trước khi nhân. Nếu tích số nhỏ hơn 1 ta nhân bình thường. Quá trình sẽ chấm dứt khi hiệu số bằng 0 hoặc đã đạt được số lẻ theo yêu cầu .

Thí dụ 1: Chuyển số 0,03125 thập phân sang nhị phân:

0,03125 x 2 = 0,06250

0,06250 x 2 = 0,125

0,125 x 2 = 0,25

0,25 x 2 = 0,50

0,50 x 2 = 1

Vậy : $(0,03125)_{10} = (0,00001)_2$

Thí dụ 2: Chuyển số 0,8125 thập phân sang nhị phân:

0.8125 x 2 = 1,625

0,625 x 2 = 1,25

0,25 x 2 = 0,50

0,5 x 2 = 1

Vậy: $(0,8125)_{10} = (0,1101)_2$

Thí dụ 3: Chuyển số 0,625 thập phân sang thập lục phân:

0,625 x 16 = 10

Vậy: $(0,625)_{10} = (A)_{16}$

2.3. CÁC MÃ NHỊ PHÂN ĐẶC BIỆT

Trong hệ thống số, ngoài các số nhị phân tự nhiên, ta còn sử dụng mã nhị phân đặc biệt, mã này cũng sử dụng hai con số 1 và 0.

2.3.1. Mã BCD (Binary Coded Decimal)

Mã nhị phân BCD không phải là một hệ đếm mà là một sự mã hóa các số thập phân bằng số nhị phân. Mã BCD làm cho việc đổi ra thập phân dễ dàng và đơn giản hơn.

Mỗi chữ số thập phân từ 0 đến 9 được biểu diễn bằng 4 bit nhị phân. Ta gọi là mã hoá số thập phân bằng nhị phân.

Trong mã này mỗi chữ số của một số thập phân đươc diễn tả bằng mã nhị phân tương đương, còn hàng của chữ số đó (hàng đơn vị, hàng chục, hàng trăm v.v...) vẫn đứng ở vị trí theo hệ thập phân. Như vậy, thực chất nó vẫn là hệ đếm thập phân nhưng các ký hiệu để đếm là các từ nhị phân 4 bit. Nghĩa là nó vừa có ưu điểm rõ ràng của hệ thập phân vừa có mật độ cao của hệ nhị phân, nhờ vậy ta có thể xử lý bằng máy tính số.

Bit có nghĩa lớn nhất có trọng số là 8, bit có nghĩa bé nhất có trọng số là 1 (BCD 8421) ngoài ra còn có các mã BCD 4221, BCD5421, BCD7421 ít thông dụng hơn. (Bảng 2.2)

Bảng 2.2: *Bảng mã BCD*

Số hệ 10	BCD8421	BCD7421	BCD4221	BCD5421
0	0000	0000	0000	0000
1	0001	0001	0001	0001
2	0010	0010	0010	0010
3	0011	0011	0011	0011
4	0100	0100	1000	0100
5	0101	0101	0111	1000
6	0110	0110	1100	1001
7	1111	1000	1101	1010
8	1000	1001	1110	1011
9	1001	1010	1111	1100

Ta dùng BCD khi các thông tin cần trực tiếp đưa vào đầu vào và ở đầu ra cho ta kết quả trực tiếp bằng số thập phân. Việc xử lý thông tin được thực hiện bằng nhị phân.Với 4 bit ta có 16 tổ hợp xuất hiện. Do đó mã BCD còn thừa 6 tổ hợp không sử dụng: 1010, 1011, 1100, 1101, 1110, 1111). Khi 6 tố hợp này xuất hiện ở kết quả ta phải cộng thêm 6 (0110) vào kết quả để có kết quả chính xác.

Phương pháp chuyển một số thập phân thành mã BCD

Ta viết các số thập phân thành nhị phân tương ứng, giữ nguyên vị trí các hàng (đơn vị, chục, trăm, ngàn ...)

Thí dụ: Số thập phân 3 7 9 2 3 0

Mã BCD 0011 0111 1001 0010 0011 0000

a/ Chuyển mã BCD thành số nhị phân

Bước 1: Chuyển mã BCD ra số thập phân:

Mã BCD: 0001 0000 0011, 0101

Số thập phân tương ứng: 1 0 3 , 5

Bước 2: Chuyển đổi số thập phân ra số nhị phân:

$(103,5)_{10} = (1100111,1)_2$

b/Chuyển số nhị phân thành mã BCD

Ta đổi số nhị phân ra số thập phân tương ứng, sau đó mỗi chữ số thập phân được thay bằng từ nhị phân tường ứng 4 bit.

$(1100111,1)_2 = (103,5)_{10} = (0001\ 0000\ 0011, 0101)_{BCD}$

Các số 0 đứng trước mỗi nhóm 4 bit vẫn có ý nghĩa và ta không được bỏ đi.

2.3.2. Mã Gray

Đây cũng là mã nhị phân. Trong bảng mã Gray mỗi đại lượng tiếp sau chỉ làm thay đổi trạng thái của 1 bit (Bảng 1.3):

Bảng 1.3: *Bảng mã Gray*

Số thập phân	Số nhị phân	Mã Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

Nhờ ưu điểm trên mã Gray được dùng nhiều trong thực tế như truyền động điện, điều chỉnh tốc độ quay của động cơ, trong các hệ thống đo lường. Nó làm cho hệ thống có độ tin cậy cao.

Phương pháp chuyển một số nhị phân sang mã Gray

Bước 1: Viết số nhị phân cần chuyển.

Bước 2: Lấy bit có nghĩa lớn nhất của số nhị phân chuyển xuống vi trí tận cùng bên trái của mã Gray. Như vậy bit bên trái của mã Gray luôn luôn là bit bên trái của số nhị phân.

Bước 3: Dịch lùi bit số cột cao xuống thấp dần. Ghi bit này trên số nhị phân tương ứng.

Bước 4: Thực hiện phép cộng nhị phân (bỏ qua nhớ nếu có). Kết qủa phép tính mang

xuống dòng dưới làm các bit của mã Gray.

Thí dụ: chuyển số nhị phân 1001 thành mã Gray:

100
+ 1001
1101 Gray

Phương pháp chuyển từ mã Gray sang số nhị phân

Bước 1: Viết mã Gray cần chuyển.

Bước 2: Dùng bit bên trái của mã Gray làm bit bên trái của số nhị phân.

Bước 3: Đưa bit này lên trên bit tiếp theo.

Bước 4: Thực hiện phép cộng (bỏ qua nhớ nếu có), kết quả của phép cộng là bit tiếp theo của số nhị phân.

Bước 5: Tiếp tục cho đến bit cuối cùng bên phải.

Thí dụ: Chuyển mã Gray 1000 sang số nhị phân:

1 1 1
+ 1 0 0 0
1 1 1 1

2.3.3. Mã đếm vòng

Mã đếm vòng là loại mã mà mỗi tổ hợp mã chỉ chứa một bit 1 các bit còn lại là bit 0. Do đó, việc ghi mã được tiến hành bằng cách dịch liên tiếp bit 1 từ phải qua trái (bảng 2.4)

Bảng 2.4 *Mã đếm vòng*

Thập phân	Mã đếm vòng
0	0000000001
1	0000000010
2	0000000100
3	0000001000
4	0000010000
5	0000100000
6	0001000000
7	0010000000
8	0100000000
9	1000000000

2.3.4. Mã ký tự

Mã ký tự dùng để ghi các thông tin không số như chữ cái, dấu các phép tính, các ký tự

đặc biệt *, @, &.... Mã ký tự dùng 1 byte = 8 bit để biểu diễn một ký tự. Có hai mã ký tự phổ biến nhất hiện nay:

Mã ASCII *(American Standard Code for Information Interchanger)*

Mã trao đổi thông tin theo tiêu chuẩn Mỹ. Sử dụng chủ yếu trong máy tính cá nhân (PC) và trong hệ thống truyền tin.

Mã ASCII là một mã nhị phân 7 bit được dùng để mã hoá các ký tự trong xử lý văn bản (các chữ cái, chữ số, các ký tự đặc biệt ...). Số ký tự tối đa có thể lên đến 2^7= 128. Bảng 1.4 là bảng mã hoá các ký tự của mã ASCII. Trong bảng chỉ có 95 ký tự, ứng với các từ mã $b_7b_6b_5b_4b_3b_2b_1$ từ 0100001đến 1111110 là có thể in và hiển thị trên màn hinh máy tính. Các từ mã còn lại dùng dùng mã hoá các ký tự điều khiển quá trình truyền thông, do đó nó không hiển thị được lên màn hình. Bit thứ 8 thường là bit kiểm tra chẵn lẻ khi trao đổi thông tin.

Thí dụ

Từ mã	Ký tự điều khiển	Ý nghĩa lệnh điều khiển
0000111	BEL	Chuông
0001001	HT	Lập bảng hàng ngang
0001011	VT	Lập bảng hàng dọc
0001010	LF	Xuống dòng
0100000	SP	Giản cách
1111111	DEL	Xóa

Bảng 2.5: *Mã ASCII*

				$b_7b_6b_5$							
				000	001	010	011	100	101	110	111
0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	“	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	/	7	G	W	g	ư
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	l
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~

1	1	1	1	SI	US	/	?	O	_	o	DEL

Thí dụ: Chữ A: 1001000, chữ a: 1101000

Mã EBCDIC *(Extended Binary Code Decimal Interchanger Code)*

Sử dụng trong máy tính lớn, được phát triển bởi hảng IBM. Mã sử dụng 8 bit để biểu thị thông tin. Nó có khả năng biểu thị ký tự nhiều hơn mã ASCII nhưng không có kiểm tra chẵn lẻ.

2.4. CÁC PHÉP TÍNH SỐ HỌC TRONG HỆ NHỊ PHÂN

Như ta đã biết các hệ thống số chỉ làm việc trong mã hệ nhị phân. Các nguyên tắc thực hiện các phép tính trong hệ nhị phân cũng tương tự như trong hệ thập phân.

2.4.1. Phép cộng nhị phân

Đây là phép tính làm cơ sở cho các phép tính khác. Hai số nhị phân cộng với nhau cũng giống như phép cộng thập phân.

Muốn cộng hai số nhị phân, ta viết chúng chồng lên nhau, các bit cùng trọng lượng nằm trên cùng một cột. Các số nhớ được xem như một bit mới để cộng với bit của cột có trọng lượng cao hơn kế bên (từ phải sang trái). Ta có quy tắc cộng:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 nhớ 1 cho vị trí trên
1 +1 + 1 = 1 nhớ 1 cho vị trí trên

Trường hợp 5 ứng với trường hợp khi 2 bit đã là 1, lại còn nhớ thêm 1 ở vị trí dưới đưa lên.

Thí dụ:

```
   10101          111001,01
 + 10111        + 110110,11
 ------         ----------
 101100         1110000,00
```

2.4.2. Phép trừ nhị phân

Ta có quy tắc sau:

0 - 0 = 0
1 - 0 = 1
1 - 1 = 0
0 - 1 = 1 nhớ 1 cho vị trí phía trên

Đối với máy tính thì phép trừ hai số nhị phân A- B được thực hiện dưới dạng cộng A+(-B). Như vậy, số nhị phân B phải mang dấu âm. Để ghi dấu - hoặc + của một số nhị phân, máy tính dùng thêm một bit gọi là bit dấu, đặt ở vị trí cực trái của số nhị phân với quy tắc :

1 : - 0 : +

Thí dụ: Một số nhị phân 8 bit thì bit cực trái thể hiện dấu, 7 bit còn lại thể hiện trị tuyệt

đối của số đó.

[1] 001100 = -52

[0] 110100 = +52

Để diễn tả số âm máy tính thường dùng phần bù cấp 2. Với cách này ta không dùng trị tuyệt đối của nó mà dùng phần bù cấp 2 của nó.

Để lấy phần bù cấp hai một số nhị phân, ta tiến hành như sau:

Bước 1: Lấy phần bù cấp 1 bằng cách đổi các bit 0 thành 1 và 1 thành 0

Bước 2: Cộng 1 vào phần bù cấp 1 để có phần bù cấp 2.

Thí dụ : Chuyển số 0110100 sang phần bù cấp 2:

Phần bù cấp 1 : 1001011

Phần bù cấp 2 : 1001011 + 1 = 1001100

Ta xét các trường hợp cụ thế sau:

Cộng hai số dương:

```
  [0] 1001  (+9)
+ [0] 0100  (+4)
  [0] 1101  (+13)
```

Cộng một số dương và một số âm có trị tuyệt đối bé hơn :

```
  [0] 1001  (+9)
+ [1] 1100  (-4)
  [0] 0101  (+5)
```

Chú ý: Ta viết -4 dưới dạng phần bù cấp 2 của 4

Cộng một số dương và một số âm có trị tuyệt đối lớn hơn.

```
  [1] 0111  (-9)
+ [0] 0100  (+4)
  [1] 1011  (-5}
```

Vì bit dấu là 1 nên kết quả có dạng phần bù cấp 2. Ta phải chuyển sang trị tuyệt đối.

Cộng hai số âm :

```
  [1] 0111  (-9)
+ [1] 1100  (-4)
  [1] 0011 (-13)
```

Như vậy, khi kết quả có bit dấu là 1 nó sẽ là phần bù cấp hai của trị tuyệt đối, ta phải chuyển sang trị tuyệt đối.

Thí dụ: [1] 0011 (-13) ta tiến hành như sau:

Đưa phần bù cấp hai về phần bù cấp1: 0011 – 1 = 0010

Chuyển về trị tuyệt đối: 0010 → 1101

Trường hợp đặc biệt:

a/ Khi số dương ít bit hơn số âm:

Khi số dương ít bit hơn số âm ta thêm số 0 vào các bit thiếu, thiếu bao nhiêu bit ta thêm

bấy nhiêu số 0.

Thí dụ: +45

+ -70

-25

$(+45)_{10}$ = (0) 101101

$(70)_{10} = (1000110)_2$

Chuyển sang phần bù cấp hai: 0111010

Ta có: $(-70)_{10}$ = (1) 0111010

Thực hiện phép cộng ta được:

(0) 0101101

+ (1) 0111010

(1) 1100111

Bit dấu là 1, do đó kết quả không phải là trị tuyệt đối mà là phần bù cấp hai. Ta chuyển sang trị tuyệt đối:

1100111

\- 1

1100110 trị tuyệt đối là: 0011001

b/ Khi số âm ít bit hơn số dương:

Khi số âm ít bit hơn số dương ta thêm số 1 vào các bit thiếu, thiếu bao nhiêu bit ta thêm bấy nhiêu số 1.

Thí dụ: -45

+ +70

+25

$(70)_{10}$= $(1000110)_2$

$(+70)_{10}$= (0) 1000110

$(+45)_{10}$ = (0) 101101 Chuyển sang phần bù cấp hai: 010011

$(-45)_{10}$ = (1) 010011

Thực hiện phép cộng ta được:

(0) 1000110

+ (1) 1010011

(0) 0011001

2.4.3. Phép nhân nhị phân

Cũng thực hiện như phép nhân thập phân, nghĩa là nhân từng bit của số nhân với các bit của số bị nhân (bắt đầu từ bit cực phải có trọng lượng nhỏ nhất đến bit cực trái có trọng lượng lớn nhất). Mỗi tích viết một hàng và cũng dịch trái một vị trí khi viết tích thứ hai. Sau đó cộng từng cột có cùng một một vị trí.

Ta có quy tắc sau :

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x1 = 1

Thí dụ:

```
  1 0 0 1
x 1 0 1 1
  1 0 0 1
1 0 0 1
1 0 0 1
1 1 0 0 0 1 1
```

Vậy phép nhân nhị phân có thể thực hiện bằng cách dịch và cộng

2.4.4. Phép chia nhị phân

Cũng giống như phép chia thập phân

```
1100011 | 10010
10010   | 101,1
0011011
  10010
------
 010010
  10010
------
  00000
```

Vậy phép chia nhị phân có thể thực hiện bằng cách trừ liên tiếp

2.5. PHÉP CỘNG BCD

Phép công BCD phức tạp hơn phép cộng nhị phân. Do dùng 4 bit để mã hóa nên sẽ xuất hiện 6 tổ hợp thừa. Dùng 4 bit nhị phân vậy sẽ có $2^4 = 16$ tổ hợp xuất hiện, Để mã hóa ta chỉ cần dùng 10 tổ hợp, vì vậy sẽ thừa 6 tổ hợp: 1010 (số 10), 1011 (số 11), 1100 (số 12), 1101 (số 13), 1110 (số 14) và 1111 (số 15). Sự xuất hiện các tổ hợp thừa này trong kết quả sẽ gây ra lỗi dư, làm kết quả sai.

2.5.1: Cộng hai mã BCD có kết quả nhỏ hơn 10:

Trường hợp này không bao giờ có các tổ hợp thừa ở kêt quả, vì vậy ta thực hiện như phép cộng nhị phân.

```
Thí dụ:   25    0010  0101
        +52     0101  0010
          77    0111  0111
```

2.5.2. Cộng hai mã BCD có kết quả lớn hơn 9

Trường hợp này sẽ có các tổ hợp thừa ở kết quả, vì vậy ta phải thực hiện việc sửa sai. Việc sửa sai được thực hiện bằng cách cộng thêm 6 (0110) vào tổ hợp thừa. Nơi nào có tổ hợp thừa xuất hiện ta cộng ngay vào để sửa sai.

Thí dụ: 45 0100 0101

+77 +0111 0111

122 1011 110 0

Kết quả sai, ta sửa sai bằng cách cộng 6 (0110) vào tổ hợp thừa

0100 0101

+0111 0111

1011 1100

+0110 0110

0001 0010 0010

Kết quả đúng.

2.5.3. Phép cộng BCD có dấu

Mã BCD cũng dùng bit 1 làm dấu âm và bit 0 làm dấu dương. Bit dấu cũng đặt ở tận cùng bên trái mã BCD.

Số âm không được biểu diễn bằng trị tuyệt đối mà bằng phần bù cấp 10. Muốn tìm phần bù cấp 10 ta tìm phần bù cấp 9 sau đó cộng 1 vào .

Thí dụ: $(57)_{10} = (0101\ \ 0111)_{BCD}$

$(+57)_{10} = (0)\ \ 0101\ \ 0111$

Muốn tìm -57, ta đổi 57 thành phần bù cấp 10.

Bước 1: Chuyển 57 sang phần bù cấp 9: 42 (phần bù cấp 9 của một số là tổng của nó và số đó bằng 9).

Bước 2: Cộng thêm 1 vào phần bù cấp 9 để có phần bù cấp 10: 42 + 1 = 43

Vậy $(-57)_{10} = (1)\ 43 = (1)\ 0100\ \ 0011$

Thí dụ: +75

+ -57

+18

57 chuyển sang phần bù cấp 10 là 43

Bài toán như sau:

(0) 75 (0) 0111 0101

+(1) 43 +(1) 0100 0011

(1) 1011 1000

Sửa sai: + 0110 0000

(0) 0001 1000

Trường hợp đặc biệt:

a/ Khi số dương ít cột hơn số âm:

Khi số dương ít cột hơn số âm ta thêm số 0 vào các cột thiếu, thiếu bao nhiêu cột ta thêm bấy nhiêu số 0.

Thí dụ: + 45
+ -270
-225

Bài toán trở thành:

(0) 045 (0) 0000 0100 0101
+(1) 730 +(1) 0111 0011 0000
(1) 0111 0111 0101

Kết quả là số âm nên không phải là trị tuyệt đối mà là phần bù cấp 10. Ta chuyển sang trị tuyệt đối.

(1) 0111 0111 0101 = -775 chuyển sang trị tuyệt đối ta được:
-775 có trị tuyệt đối: 225

b/ Khi số âm ít cột hơn số dương:

Khi số âm ít cột hơn số dương ta thêm số 9 vào các cột thiếu, thiếu bao nhiêu cột ta thêm bấy nhiêu số 9.

Thí dụ: - 45
+ +270
+225

Bài toán trở thành:

(1) 955 (1) 1001 0101 0101
+(0) 270 +(0) 0010 0111 0000
(1) 1011 1100 0101
+ 0110 0110 0000
(0)0010 0010 0101

2.6. CÁC HÀM ĐẠI SỐ LOGIC

2.6.1. Định nghĩa về đại số Logic :

Theo lý thuyết tập hợp, ta có thể coi đại số logic là tập hợp S các đối tượng A,B,C… trong đó xác định hai phép toán cộng và nhân logic.

Nó có các tính chất sau:

Đối với mọi phần tử A,B,C ở trong S

1/ S chứa A + B và A x B (Tính đóng kín)

2/ A + B = B + A
A x B = B x A (luật giao hoán)

3/ A + (B + C) = (A + B) + C

A x (B x C) = (A x B) x C (luật kết hợp)

4/ A x (B + C) = A x B + A x C

5/ S chứa các phần tử 1 và 0 sao cho với mọi A thuộc S :

A + 0 = A ; A x 0 = 0 ; A x 1 = A ; A + 1 = 1

6/ Với mọi phần tử A, S chứa một phần tử $\overline{A}$ (gọi là phần bù của A cũng có thể viết là $\overline{A}$ hoặc 1-A) sao cho :

$A + \overline{A} = 1$; $A \times \overline{A} = 0$

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

$\overline{A.B} = \overline{A} + \overline{B}$ (luật De Morgan)

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

Bảng 2.5: *Các định lý của đại số Boole*

Số TT	Định lý	Số TT	Định lý
1	$A + 0 = A$	12	$A(A + B) = A$
2	$A \cdot 1 = A$	13	$A + \overline{A}B = A + B$
3	$A + 1 = 1$	14	$A(\overline{A} + B) = AB$
4	$A \cdot 0 = 0$	15	$AB + A\overline{B} = A$
5	$A + A = A$	16	$(A + B)(A + \overline{B}) = A$
6	$A \cdot A = A$	17	$AB + \overline{A}C = (A + C)(\overline{A} + B)$
7	$A + \overline{A} = 1$	18	$(A + B)(\overline{A} + C) = AC + \overline{A}B$
8	$A \cdot \overline{A} = 0$	19	$AB + \overline{A}C + BC = AB + \overline{A}C$
9	$A(B + C) = AB + AC$	20	$(A + B)(\overline{A} + C)(B + C) = (A + B)(\overline{A} + C)$
10	$A + BC = (A + B)(A + C)$	21	$\overline{A.B.C..} = \overline{A} + \overline{B} + \overline{C} + ...$
11	$A + AB = A$	22	$\overline{A + B + C + ...} = \overline{A}.\overline{B}.\overline{C}..$

2.6.2. Các toán tử Logic

Các biến số Bool có thể xử lý bằng các toán tử logic: KHÔNG (NO), VÀ (AND), HOẶC (OR), KHÔNG HOẶC (NOR), KHÔNG VÀ (NAND). Ba toán tử NO, AND, OR làm thành một hệ thống logic hoàn chỉnh còn mỗi toán tử NOR, NAND là các hàm phụ thuộc.

a/Toán tử KHÔNG NO (phủ định)

Nếu A = 1 thì không A = $\overline{A}$ = 0

Nếu A = 0 thì không A = $\overline{A}$ = 1

A và $\overline{A}$ có thể lấy giá trị 1 hoặc 0. Ta quy định 0 là phần bù của 1 và ngược lại.

b/Toán tử VÀ

Còn gọi là toán tử giao: A VÀ B VÀ C = ABC = 1 Nếu và chỉ nếu A =B = C =1 còn các trường hợp khác đều bằng 0. Toán tử VÀ có thể viết theo ký hiệu sau: AB, A.B hoặc A ∩ B

c/Toán tử HOẶC OR

Còn gọi là toán tử hội: A HOẶC B HOẶC C = A+B+C = A ∪ B ∪ C = 0 nếu và chỉ nếu A = B = C = 0, các trường hợp khác đều bằng 1.

c/Toán tử KHÔNG HOẶC NOR, KHÔNG VÀ NAND, là phủ định của 2 toán tử trên.

Các toán tử được ký hiệu như sau :

NO AND OR NAND NOR

2.6.3. Giãn đồ Venn

Để biểu diễn một cách trực quan các phép toán trong Đại số logic ta dùng giãn đồ Venn theo nguyên tắc sau:

Tập hợp mọi trạng thái của một mệnh đề X có thể biểu diễn bằng hình học.

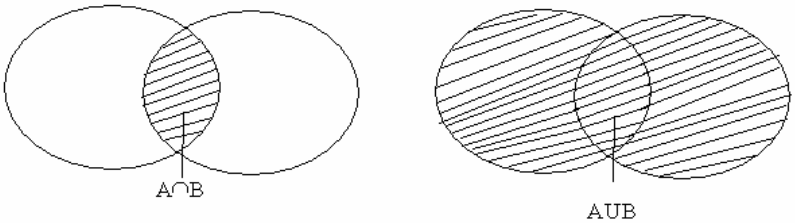
Ta quy định khi biểu diễn trên mặt phẳng, những tập hợp mà mệnh đề X là đúng (ứng với trạng thái 1) là diện tích nằm trong một vòng tròn. Còn ngoài vòng tròn diễn tả trạng thái X=0, nghĩa là mệnh đề X là sai.

Nếu chúng ta xét 2 mệnh đề X và Y trong đó miền nghiệm đúng có phần chung, thì 2 vòng tròn diễn tả các tập hợp con X và Y chia mặt phẳng làm 4 vùng hoặc 4 tập hợp con tương ứng với nghiệm đúng của 4 mệnh đề. Trong đó vùng (1) là vùng chung của " X đúng" và "Y đúng".

Từ đó ta thấy:

-Toán tử và là toán tử giao, bởi vì trong vùng (1) "X và Y đều đúng"

-Toán tử hoặc là toán tử hợp, bởi vì trong vùng (1), (2), (3), (4) là "vùng X hoặc Y đúng"



2.6.4. Phương pháp biểu diễn hàm logic

a/Khái niệm về Maxterm và minterm

Ta có hàm logic 3 biến như sau:

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

Ta thấy mỗi số hạng của tích số ở vế phải đều có chứa các biến của hàm ở dưới dạng trực tiếp (thí dụ A) hoặc dạng bù (thí dụ $\overline{A}$) được gọi là tích của các tổng. Mỗi tổng trong ngoặc được gọi là một maxterm.Trong một maxterm mỗi biến chỉ xuất hiện một lần (dưới dạng bù hoặc dạng chuẩn).

Ta ký hiệu maxterm bằng chữ M kèm theo một con số.

Một hàm logic k biến có 2^k maxterm

Thí dụ: Hàm hai biến F(A,B) có 4 maxterm :

$M_0 = \overline{A} + \overline{B}$ $M_1 = \overline{A} + B$ $M_2 = A + \overline{B}$ $M_3 = A + B$

Tương tự như trên, ta có khái niệm về minterm trong đó chỉ thay thế phép nhân logic vào vị trí các phép cộng logic đối với maxterm.

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

Một hàm 2 biến có 4 miterm như sau:

$m_0 = \overline{A}\,\overline{B}$ $m_1 = \overline{A}B$ $m_2 = A\overline{B}$ $m_3 = A\,B$

Các minterm được ký hiệu bằng chữ m với một chỉ số .

b/Biểu diễn hàm logic bằng hình học - bảng Karnaugh

Khi một hàm logic có số lượng biến tương đối bé (k < 6) người ta thường biểu diễn chúng dưới dạng một bảng gọi là bảng Karnaugh (còn gọi là ma trận Karnaugh). Thực chất bảng Karnaugh là cách viết cải tiến của bảng trạng thái.

Theo phương pháp này, hàm logic có k biến được biểu diễn trên một bảng gồm có 2^k ô vuông. Mỗi ô vuông ứng với một minterm của hàm cần biểu diễn.

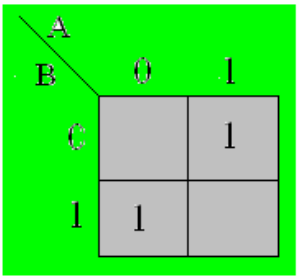
Muốn biểu diễn một hàm bằng bảng Karnaugh ta tiến hành như sau:

Bước 1: Lập bảng Karnaugh tương ứng với số biến của hàm nhưng trong các ô vuông chưa ghi giá trị.

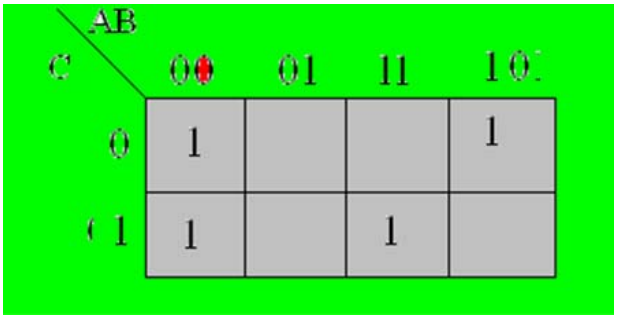
Bước 2: Trong các ô vuông cúa bảng Karnaugh, nếu ứng với một bộ giá trị của biến mà hàm có giá trị là 1 thì ô vuông biểu diễn minterm tươug ứng được ghi là 1, ngược lại ta để trống .

Chú ý: thứ tự hàng và cột theo mã Gray

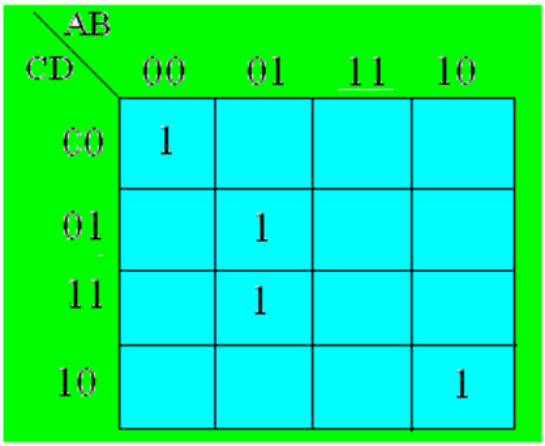
Thí dụ 1: Hàm 2 biến $F = \overline{A}B + A\overline{B}$



Thí dụ 2: Hàm 3 biến $F = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{C} + ABC$



Thí dụ 3: Hàm 3 biến $F = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,B\,\overline{C}\,D + \overline{A}\,B\,C\,D + A\,\overline{B}\,C\,\overline{D}$



Trong bảng Karnaugh không những các ô kế cận khác nhau chỉ một biến mà các ô đầu dòng và cuối dòng, đầu cột và cuối cột cũng chỉ khác nhau một biến. Các ô đó được gọi là ô kế cận. Tính chất này được gọi là tính chất tuần hoàn của bảng Karnaugh. Áp dụng nhiều trong việc tối thiểu hóa hàm trạng thái.

2.6.5. Tối thiểu hóa hàm trạng thái

Trong thực tế ta cần phải làm thế nào đó, để có thể thực hiện dễ dàng các sơ đồ mạch điện dùng các phần tử logic. Một mạch điện dùng càng ít phần tử càng dễ thực hiện. Vì vậy, ta cần tối thiểu hóa các phần tử tạo thành mạch bằng cách tối thiểu hóa hàm trạng thái.

Ta xét vài phương pháp thông dụng và tương đối đơn giản:

a/Dùng tính chất của đại số logic để biến đổi trực tiếp dạng giải tích của hàm

Khi số biến ít người ta thường dùng phương pháp biến đổi trực tiếp dạng giải tích của hàm, bằng cách dùng các tính chất giải tích của đại số Bool:

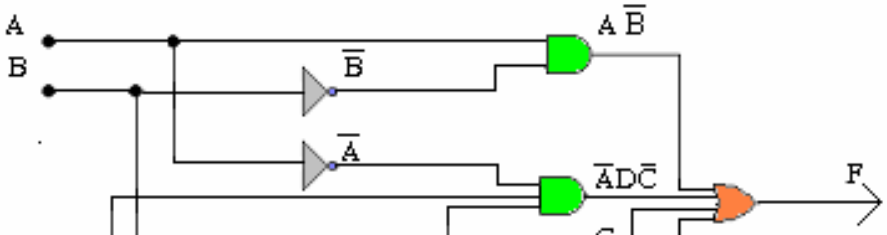
Thí dụ: Cho hàm logic 4 biến : $F(A,B,C,D) = A\overline{B} + C + A\overline{C}D + B\overline{C}D$

Hãy: 1/Vẽ mạch logic thực hiện hàm trên

2/Tối thiểu hóa hàm trạng thái trên

3/Vẽ lại mạch logic sau khi đã tối thiểu hóa hàm trạng thái.

Để vẽ mạch logic trên, ta dùng các toán tử logic : OR , AND , NO (Hình 2.1)



Hình 2.1: *Mạch logic chưa tối thiểu*

2/Tối thiểu hóa hàm trạng thái

Dùng phương pháp biến đổi trực tiếp ta tối thiểu hóa hàm F như sau:

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

Ta có : $\overline{A}\,\overline{C}\,D + B\,\overline{C}\,D = \overline{C}(\overline{A}\,D + B\,D)$

Hàm F(A,B,C,D) có thể viết lại:

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

Mà: $A + \overline{A}\,B = A + B$

Nên: $C + \overline{C}\,(\overline{A}\,D + B\,D) = C + (\overline{A}\,D + B\,D)$

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

Theo định luật De Morgan : $\overline{A} + \overline{B} = \overline{A\,B}$

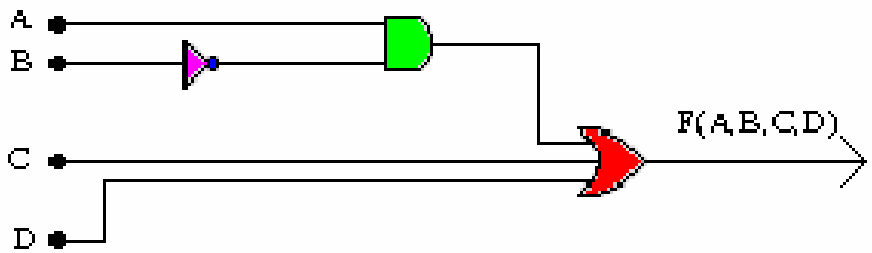
$F(A,B,C,D) = A\overline{B} + C + D(\overline{A\overline{B}}) = C + (A\overline{B}) + (\overline{A\overline{B}})D$

Aïp dụng tính chất: $A + \overline{A}\,B = A + B$ ta có :

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

3/Vẽ lại mạch logic sau khi đã tối thiểu hóa

Từ hàm đã rút gọn trên ta vẽ lại sơ đồ : (Hình 2.2)



Hình 2.2: *Mạch logic sau khi đã tối thiểu*

b/ Phương pháp ma trận Karnaugh:

Ta có thể tối thiểu hóa hàm logic biểu diễn bằng bảng Karnaugh theo các bước sau:

Bước 1: Khoanh tròn các ô có giá trị 1 không thể kết hợp với bất kỳ ô nào khác. Ở kết quả

ta ghi minterm này vào.

Bước 2: Xác định các ô chỉ kết hợp với 1 ô khác môt cách duy nhất. Khoanh tròn các tổ hợp hai ô này lại với nhau.

Bước 3: Xác định các ô có thể kết hợp với ba ô khác một cách duy nhất. Nếu tất cả 4 ô kết hợp như vậy không bao trùm hết các nhóm 2 ô thì ta khoanh tròn nhóm 4 ô này lại. Các ô có thể gộp lại để tạo thành nhóm 4 ô theo nhiều cách tạm thời bỏ qua.

Bước 4: Thực hiện như trên với nhóm 8 ô

Bước 5: Các ô chưa được khoanh có thể kết hợp với nhau hoặc kết hợp với các ô đã được khoanh một cách tùy ý. Ta kết hợp thế nào đó cho số nhóm là ít nhất.

Ta chú ý:

- Cùng một số 1 có thể tham gia nhiều nhóm (ô kết hợp rồi có thể kết hợp lại)
- Số ô ở trong mỗi nhóm càng lớn, kết quả thu được càng tối giản.

Thí dụ: Cho hàm 4 biến: $F(A,B,C,D) = \Sigma m\ (0,1,3,5,6,9,11,12,13,15)$

Hãy cực tiểu hóa bằng bảng Karnaugh.

Tiến hành các bước như trên :

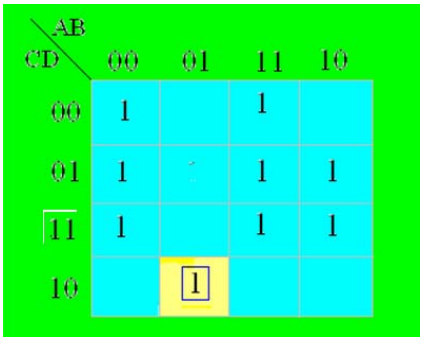
* Ô m_6 không thể kết hợp với bất cứ ô nào khác, ta khoanh vùng ô này lại. Đây là ô tối giản sẽ có ở kết quả. (Hình 2.3a)

* Các ô m_0 và m_{12}chỉ có thể kết hợp với chỉ một ô khác, ta khoanh các tổ hợp này lại tạo thành nhóm 2 ô (Hình 2.3b). Các ô khác có thể kết hợp để trở thành nhóm 2 ô theo nhiều cách khác nhau tạm thời ta chưa đề cập đến.

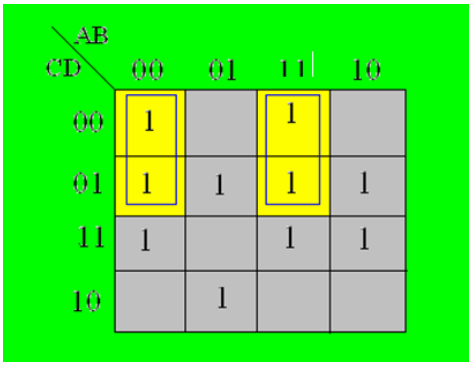
*Các ô m_3,m_5,và m_{15} có thể tham gia vào các nhóm 4 ô theo một cách duy nhất, ngoài ra ta cũng thấy rằng các nhóm 4 ô kết hợp như vậy không bao trùm hết các nhóm 2 ô nên ta khoanh vòng các nhóm 4 ô này lại (Hình 2.3c)

* Sau cùng các ô đã được khoanh vùng (Hình 2.3d) . Ta có được kết quả :

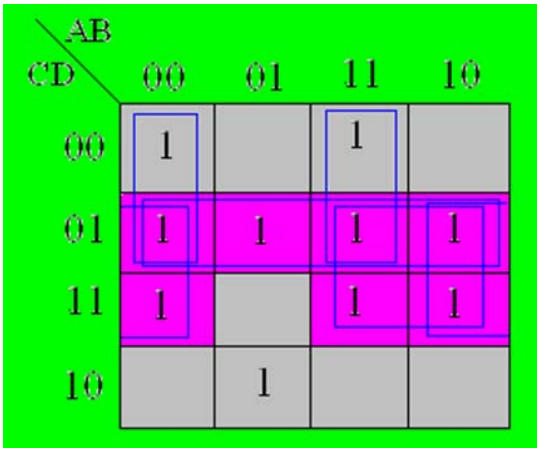
$$F(A,B,C,D) = \overline{A}\,B\,\overline{C}\,D + \overline{A}\,\overline{B}\,\overline{C} + A\,B\,\overline{C} + \overline{C}\,D + \overline{B}\,D + A\,D$$



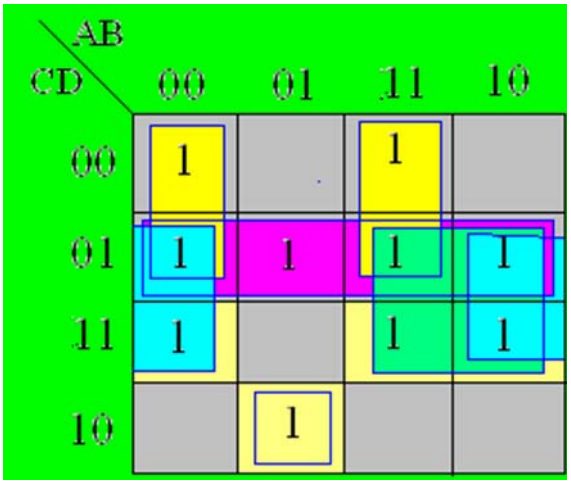
Hình 2.3a: *Ô không thể kết hợp với bất cứ ô nào*



Hình 2.3b: *Một ô kết hợp với một ô khác thành bộ 2 ô*



Hình 2.3c: *Một ô kết hợp với 3 ô khác thành bộ 4 ô*



Hình 2.3d: *Các ô đã được kết hợp*

CHƯƠNG 3 CÁC CỔNG LOGIC

3.1.KHÁI NIỆM CHUNG

Các đại lượng nhị phân trong thực tế là những đại lượng Vật lý khác nhau (dòng điện, điện áp,áp suất...). Các đại lượng đó có thể thể hiện bằng hai trạng thái có '1' hoặc không '0'.

Các cổng logic là các phần tử đóng vai trò chủ yếu để thực hiện các chức năng logic đơn giản nhất trong các sơ đồ logic nhằm thực hiện một hàm logic nào đó. Quan hệ logic cơ bản nhất có ba loại: AND, OR, NOT.

Cổng logic gồm các phần tử có nhiều đầu vào và chỉ có một đầu ra. Đầu ra là tổ hợp của các đầu vào.

Từ các cổng logic ta có thể kết hợp lại để tạo ra nhiều mạch logic thực hiện các hàm logic phức tạp hơn.

3.2 CÁC CỔNG LOGIC CƠ SỞ:

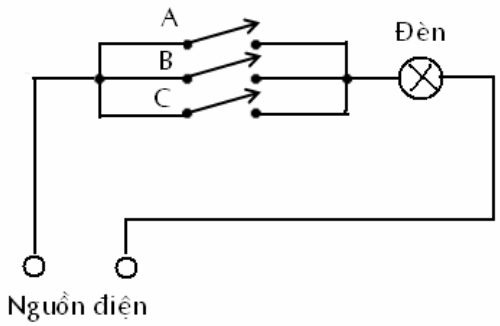
3.2.1. Cổng HOẶC (OR)

Cổng HOẶC có 2 hoặc nhiều lối vào và chỉ có một lối ra. Lối ra ở mức 1 nếu có ít nhất một lối vào ở mức 1 (Lối ra có tín hiệu khi một lối vào có tín hiệu).Ta có bảng chân lý sau:

Vào		Ra
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Ta viết Y = A + B và nói cổng HOẶC thực hiện phép cộng logic

Ta có thể xem cổng HOẶC như một mạch điện mắc song song (Hình 3.1)



Hình 3.1: *Cổng hoặc dùng các chuyển mạch cơ khí*

Trong mạch điện ở hình 3.1, ta thấy chỉ cần một chuyển mạch A, B hoặc C đóng, đèn sẽ sáng ngay.

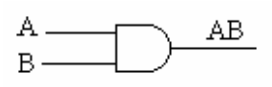
Cổng logic OR thực hiện quan hệ: một sự kiện sẽ xảy ra khi chỉ cần một điều kiện quyết định sự kiện đó được đáp ứng.

3.2.2.Cổng VÀ (AND)

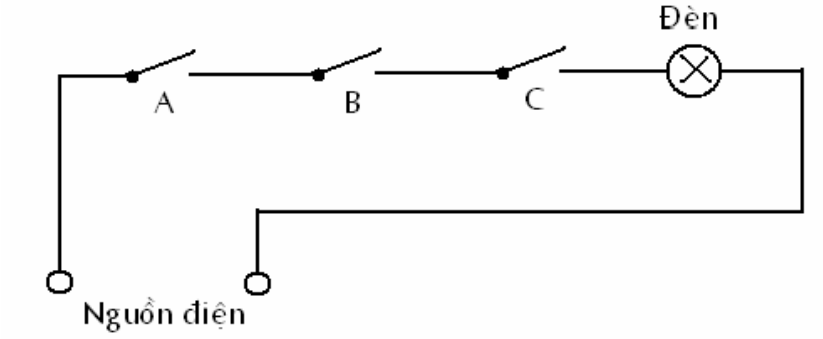
Cổng VÀ có 2 hoặc nhiều lối vào và chỉ có một lối ra. Lối ra chỉ ở mức 1 nếu tất cả lối vào đều ở mức 1 (Lối ra có tín hiệu khi tất cả lối vào đều có tín hiệu).

Ta viết Y = AB và nói cổng VÀ thực hiện phép nhân logic

Vào		Ra
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



Ta có thể xem cổng AND như một mạch điện mắc nối tiếp (Hình 3.2)



Hình 3.2: *Cổng AND dùng các chuyển mạch cơ khí*

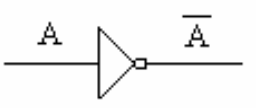
Trong mạch điện ở hình 3.2 ta thấy khi tất cả các chuyển mạch A, B, C đều đóng, đèn mới sáng được.

Cổng logic AND thực hiện quan hệ: một sự kiện sẽ xảy ra khi tất cả mọi điều kiện quyết định sự kiện đó được đáp ứng.

3.2.3.Cổng KHÔNG (NO)

Còn gọi là cổng đảo. Cổng chỉ có một lối vào và một lối ra. Cổng KHÔNG thực hiện phép phủ định logic. Cổng KHÔNG còn gọi là cổng chặn.

Vào	Ra
A	Y
0	1
1	0

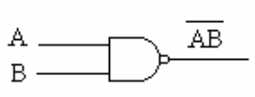


3.3. CÁC CỔNG LOGIC GHÉP

3.3.1.Cổng KHÔNG VÀ (NAND)

Cổng KHÔNG VÀ là cổng VÀ bị phủ định :

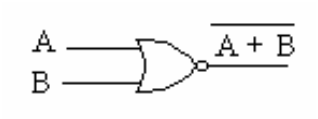
Vào		Ra
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



3.3.2.Cổng KHÔNG HOẶC (NOR)

Cổng KHÔNG HOẶC là cổng HOẶC bị phủ định

Vào		Ra
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

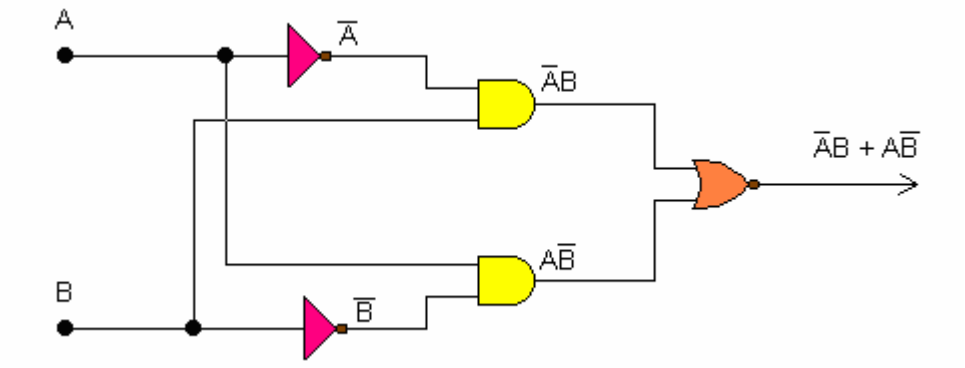


3.4. CỔNG KHÁC DẤU

3.4.1 Cổng HOẶC loại trừ (Exclusive OR)

Cổng hoặc loại trừ còn gọi là cổng cộng modul 2 hoặc là cộng không nhớ, gọi tắt là EXOR. Có biểu thức logic $Y = \overline{A}B + A\overline{B}$ hoặc $A \oplus B$

Ta có sơ đồ mạch nhưhình 3.3:

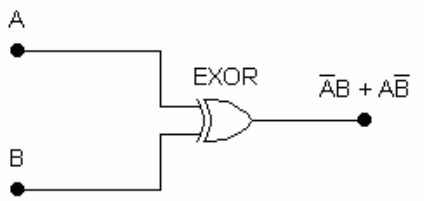


Hình 3.3: *Sơ đồ logic* $Y = \overline{A}B + A\overline{B}$

Bảng trạng thái của $Y = \overline{A}B + A\overline{B}$

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

So sánh với cổng logic OR, ta thấy 3 trạng thái đầu là của cổng logic OR chỉ khác trạng thái thứ tư, ta gọi là cổng logic KHÔNG đồng trị hay là HOẶC loại trừ (Exclusive OR), có ký hiệu như hình 3.4.



Hình 3.4: *Ký hiệu cổng XOR*

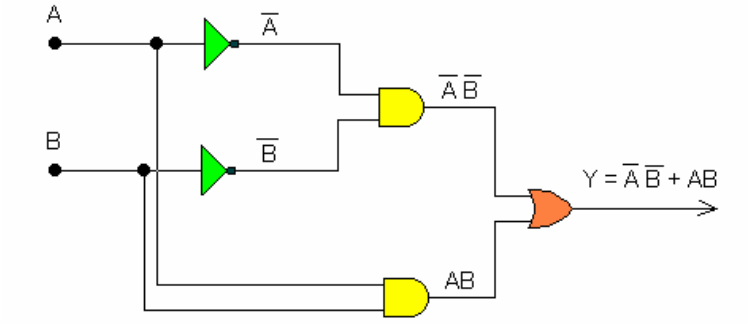
Đầu ra của cổng XOR bằng 1 khi hai đầu vào khác trạng thái và bằng 0 khi cùng trạng thái. Nếu nhiều đầu vào thì đầu ra sẽ bằng 1 khi số bit 1 ở đầu vào là số lẻ và bằng 0 khi số bit 1 ở đầu vào là số chẵn

3.4.2 Cổng KHÔNG HOẶC loại trừ (Exclusive NOR)

Một cổng logic khác cũng thường được sử dụng đó là cổng Exclusive NOR (XNOR) còn gọi là cổng đồng dấu.

Có biểu thức logic $\overline{A}\,\overline{B}+AB$ hoặc $\overline{A \oplus B}$

Mạch logic để thực hiện hàm logic trên (Hình 3.5)

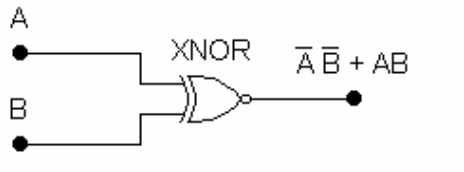


Hình 3.5: *Sơ đồ logic* $\overline{A}\,\overline{B}+AB$

Bảng trạng thái của mạch trên :

A	B	$\overline{A}$	$\overline{B}$	AB	$\overline{A}\,\overline{B}$	$\overline{A}\,\overline{B}+A\,B$
0	0	1	1	0	1	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	1	0	1

Ta thấy bảng trạng thái trên là đảo của bảng trạng thái XOR, nên gọi là XOR đảo hoặc là XNOR. Cổng XNOR có ký hiệu như hình 3.6



Hình 3.6: *Ký hiệu cổng XNOR*

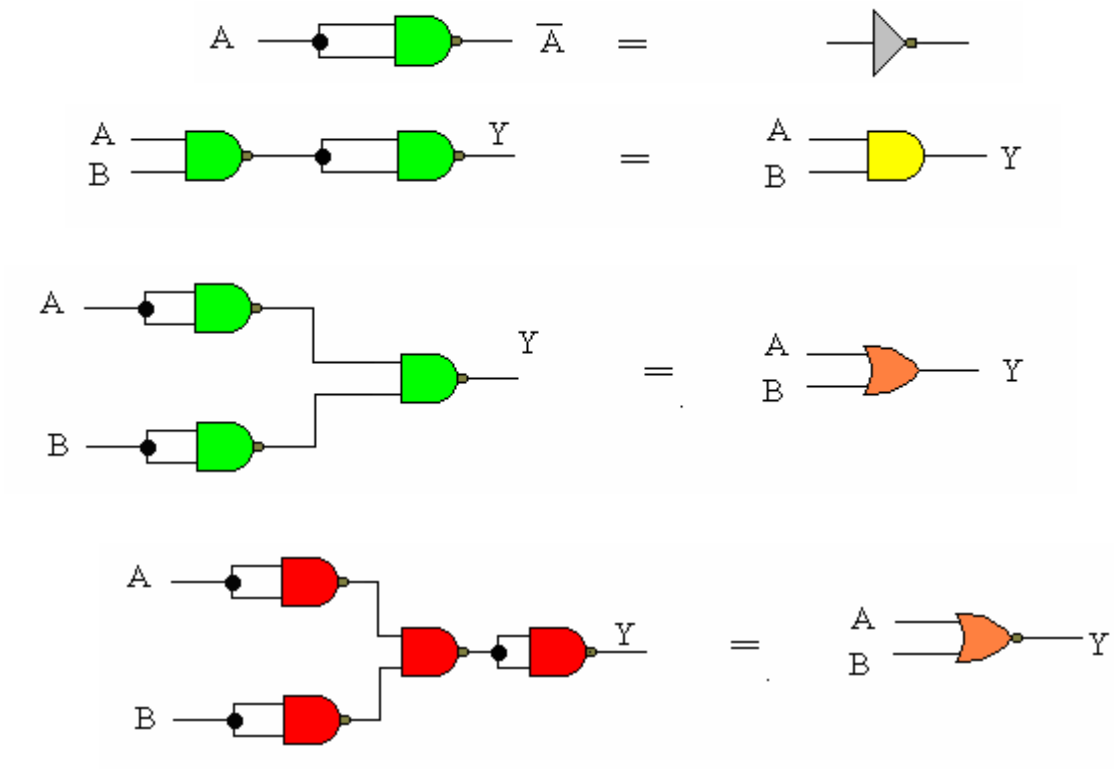
Đầu ra của cổng XNOR bằng 1 khi hai đầu vào cùng trạng thái và bằng 0 khi khác trạng thái. Nếu nhiều đầu vào thì đầu ra sẽ bằng 1 khi số bit 0 ở đầu vào là số lẻ và bằng 0 khi số bit 0 ở đầu vào là số chẵn. Thí dụ: bảng trạng thái của một cổng XNOR 3 đầu vào:

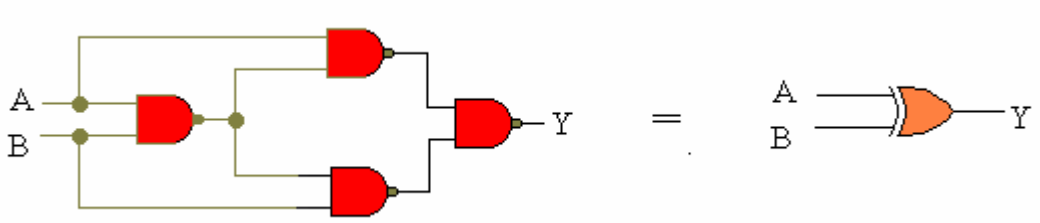
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Ta thường dùng các cổng XOR và XNOR trong các bộ so sánh, bộ cộng…

Trong các cổng trên, hai cổng NAND và NOR được dùng rất linh hoạt. Từ hai cổng này, ta có thể tạo ra các cổng logic cơ bản NO, AND, OR.

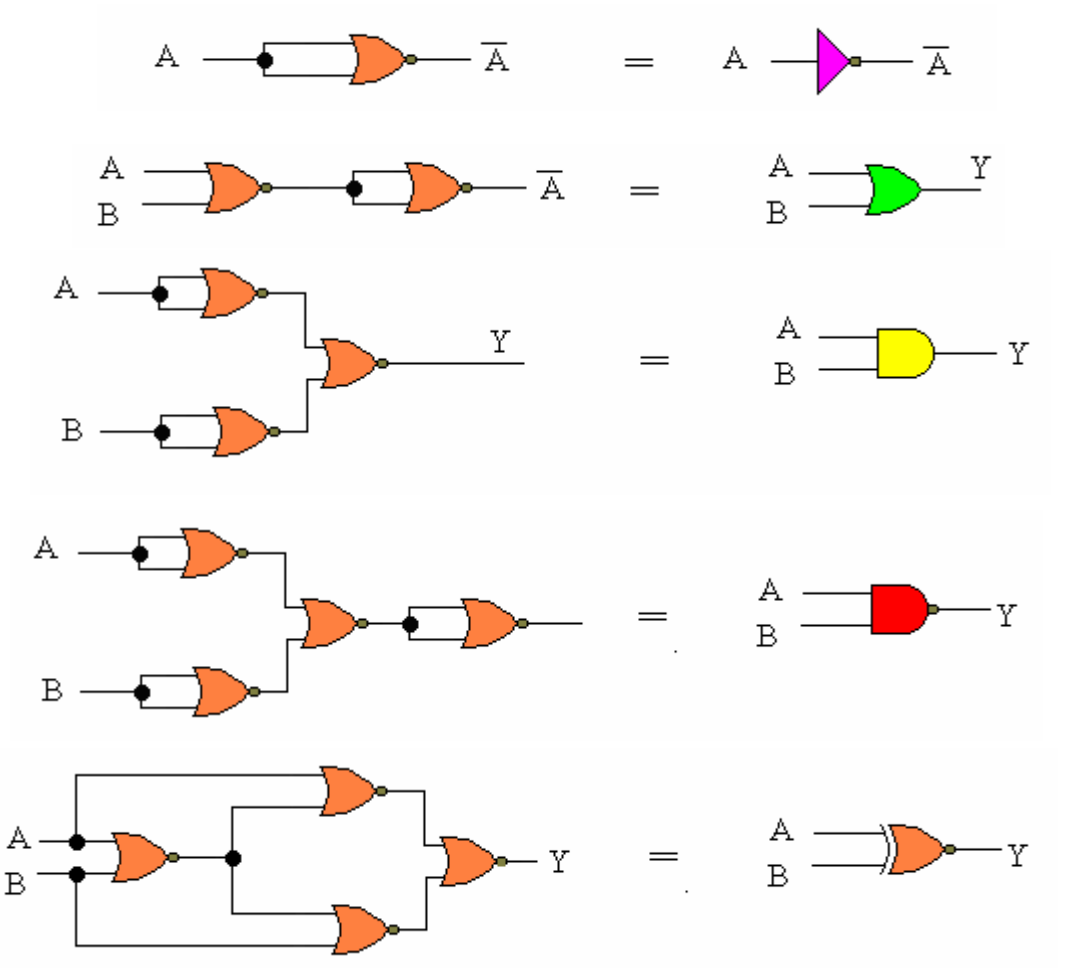
a/ Dùng các cổng NAND (Hình 3.7)





Hình 3.7: *Tạo các cổng logic từ cổng NAND*

b/ Dùng các cổng NOR (Hình 3.8)



Hình 3.8: *Tạo các cổng logic từ cổng NOR*

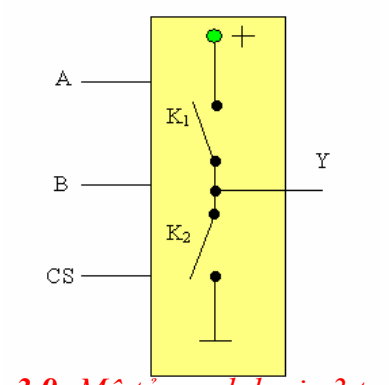
3.5. CỔNG LOGIC 3 TRẠNG THÁI TS (THREE STATE)

Cổng logic ba trạng thái là cổng logic mà đầu ra có thêm trạng thái thứ ba gọi là trạng thái treo ngoài hai trạng thái 1 và 0. Đầu ra Y có thể nằm ở một trong ba trạng thái sau:

Trạng thái mức cao và mức thấp 1 hoặc 0. Trạng thái thứ ba là trạng thái treo hay còn gọi

là trạng thái tổng trở cao. Lúc đó đầu ra Y tách ra khỏi hệ thống.

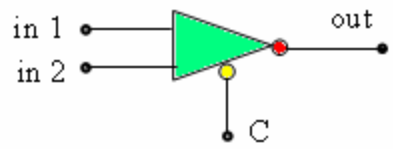
Hình 3.9 mô tả mạch logic 3 trạng thái:



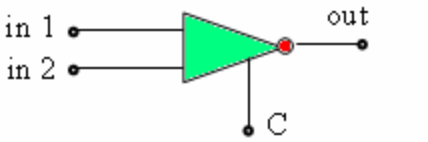
Hình 3.9: *Mô tả mạch logic 3 trạng thái.*

Khi K_1 đóng đầu ra có trạng thái 0, Khi K_1, K_2 đóng, đầu ra có trạng thái 1. Khi K_1, K_2 cùng tắt, mạch ở trạng thái thứ 3 tổng trở cao. Đầu ra Y tách khỏi mạch (dù thực tế nó vẫn nôi với mạch. CS (Chip Select) dùng để chọn chip. CS sẽ điều khiển mạch ở trạng thái thứ ba. Khi CS = 1 (hoặc 0 thì hai khóa đều mở, độc lập với tín hiệu vào A, B.

Cổng logic 3 trạng thái được sử dụng khi ta cần ghép kênh các tín hiệu cần truyền luân lưu trên một dây dẫn AB (AB còn gọi là bus).

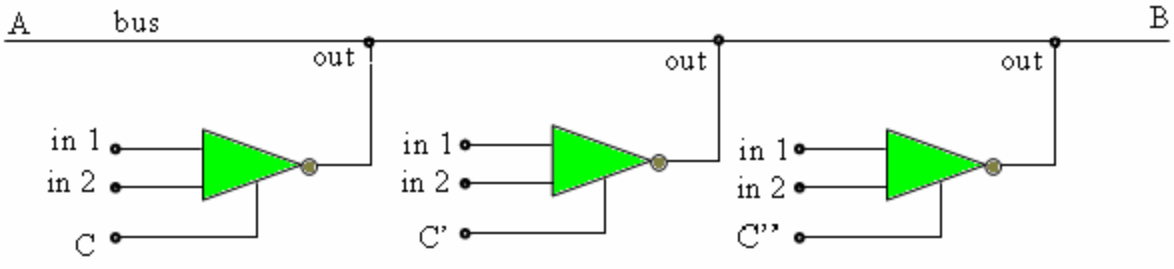


Hình 3.10a: *Trạng thái treo ở mức thấp*



Hình 3.10b: *Trạng thái treo ở mức cao*

Ưu điểm nổi bật của các vi mạch logic ba trạng thái là ta có thể nối đầu ra của vi mạch lên cùng một kênh truyền chung. Điều này làm đơn giản rất nhiều cho việc tạo lập kênh truyền số liệu trong một hệ thống logic. Hình 3.11 cho ta một ví dụ về việc nối vi mạch logic trên một kênh truyền.



Hình 3.11: *Nối các vi mạch logic với một kênh truyền chung*

Nếu tín hiệu điều khiển C, C', C'' có thứ tự thời gian ở mức cao, thì các tín hiệu dữ liệu ở ba nhóm đầu vào sau khi đã thực hiện quan hệ logic sẽ đưa ra bus luân lưu theo thứ tự thời gian tương ứng. Để các cổng TS hoạt động bình thường thì ở một thời điểm bất kỳ chỉ cho phép một

cổng duy nhất ở trạng thái công tác. Nếu không sẽ xảy ra trường hợp một lúc có đến hai đầu ra của cổng cùng thông với bus, nếu hai cổng này có đầu ra khác trạng thái một ở muác cao, một ở mức thấp sẽ đưa đến hỏng cổng.

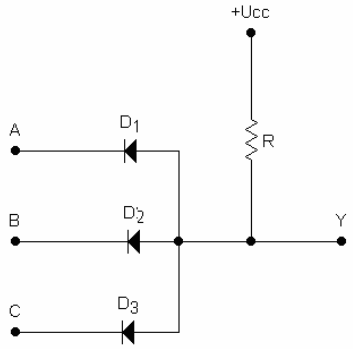
3.6 CÁC MẠCH LOGIC CƠ BẢN

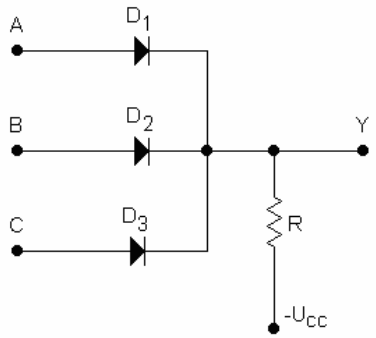
Trong kỹ thuật, các cổng logic được thực hiện bằng vật chất. Trước khi có kỹ thuật điện tử các cổng logic đươc thực hiện bằng mạch điện với các relay điện từ, các phương tiện cơ khí.Tiếp theo là các mạch bằng các linh kiện điện tử. Nhiều chức năng logic phức tạp đã được thực hiện theo nhiều dạng khác nhau, mỗi dạng gọi là một họ logic. Mỗi họ có một tính chất, vì vậy, tuy cùng một chức năng logic nhưng tham số các cổng hoàn toàn khác nhau. Ta có các họ logic cơ bản sau :

3.6.1. Họ DDL (Diode Diode Logic)

Cổng Logic họ này được hình thành từ các diode

Ta có sơ đồ nguyên lý cổng AND (Hình 3.12) và cổng OR (Hình 3.13) dùng diode sau:





Hình 3.12 : *Sơ đồ nguyên lý cổng AND* ***Hình 3.13:*** *Sơ đồ nguyên lý cổng OR*

Khi A, B và C đều ở mức thấp, diode phân cực thuận nên thông, có dòng điện chạy qua điện trở tải R, dòng điện này gây sụt áp trên điện trở R Điện áp tại đầu ra Y= 0.

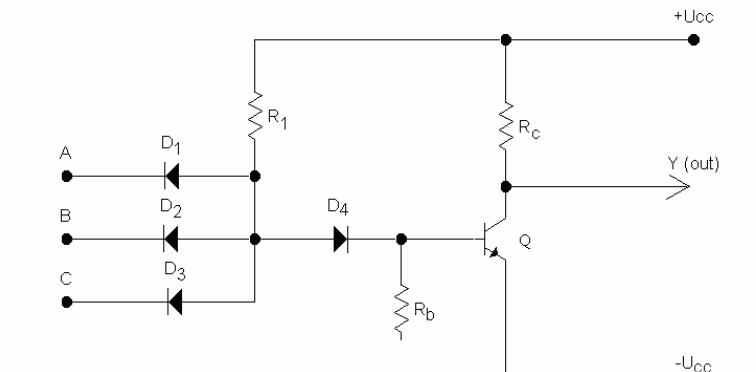
Khi có một đầu vào ở trạng thái cao, diode tương ứng bị phân cực ngược nên tắt nhưng do diode kia vẫn ở trạng thái thông nên ngõ ra vẫn ở mức 0.

Khi cả ba đầu vào đều ở mức cao, cả ba diode đều bị phân cực ngược nên tắt, không có dòng điện chạy qua diode do đó không gây sụt áp trên điện trở R. Điện áp ngõ ra ở mức cao Y = 1. Phân tích tương tự cho mạch OR.

Họ DDL có ưu điểm là đơn giản, dễ chế tạo các cổng logic nhiều đầu vào, tần số công tác cao, nhược điểm là hệ số ghép tải nhỏ, chống nhiễu kém.

3.6.2 Họ Logic DTL (Diode Transistor Logic)

Hình 3.14 là sơ đồ nguyên lý mạch NAND họ DTL .



Hình 3.14: *Sơ đồ nguyên lý mạch logic NAND họ DTL*

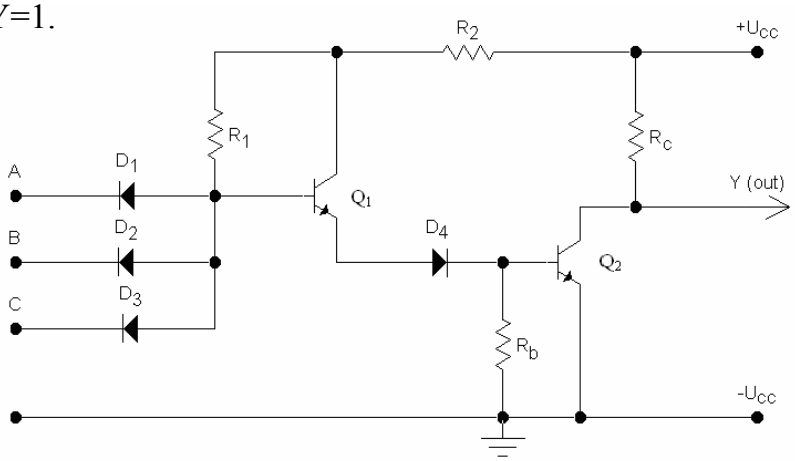
Khi cả 3 lối vào đều ở mức 0. Cả 3 diode đều dẫn vì cả 3 đều phân cực thuận. Điện thế tại điểm P sẽ thấp. D_4 tắt do phân cực nghịch. Transistor Q không được định thiên nên tắt. Dòng cực thu I_c=0. Điện áp lối ra ở mức cao. Đầu ra ở trạng thái 1.

Khi một lối vào nào đó ở mức 0. Diode tương ứng sẽ dẫn. Phân tích tương tự điện áp lối ra ở mức 1. Khi tất cả lối vào đều ở mức 1. Cả 3 diode đều tắt vì cả 3 đều phân cực nghịch, Điện thế tại điểm P xấp xỉ điện áp nguồn D_4 được phân cực thuận nên thông. Transistor Q được định thiên, dòng I_b đủ lớn sẽ làm cho Q bão hòa $I_c = I_{cmax}$ U_c= 0. Điện áp lối ra ở mức thấp, nghĩa là ở ứng với trạng thái 0.

Để tăng khả năng chịu tải cho lối ra, giảm được thời gian cắt dòng (tăng tốc độ tác động), người ta thay D_4 bằng transistor (Hình 3.15)

R_1 là điện trở định thiên cho Transistor Q_1 mắc theo kiểu C chung (mạch lập lại cực phát)

Khi một hoặc tất cả các đầu vào đều ở trạng thái thấp (mức 0) thì diode ở đầu vào sẽ thông. Điện áp tại cực B của Q_1 thấp, Q_1 không được định thiên nên dòng cực thu = 0.đầu ra Y ở trạng thái cao Y=1.



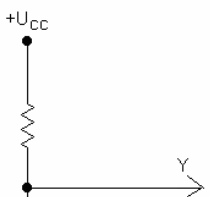
Hình 3.15: *Sơ đồ nguyên lý mạch logic NAND họ DTL có tăng khả năng chịu tải*

Khi tất cả các lối vào đều ở mức cao (mức 1), cả 3 diode đều phân cực nghịch. Dòng điện chạy từ nguồn cung cấp qua R_1 định thiên cho Q_1 làm cho Q_2 thông bão hòa và đầu ra ở mức 0.

3.6.3. Họ logic RTL (Resistor Transistor Logic)

a/ Cổng logic NO họ RTL:

Để thực hiện mạch logic không NO ta có sơ đồ sau (Hình 3.16)



Hình 3.16 *Sơ đồ nguyên lý cổng NO họ RTL*

Khi ở đầu vào (cực B) là 0 Tranistor không hoạt động, dòng $I_c = 0$.

Từ công thức $U_c = U_{cc} - R_cI_c$ ta có : $U_c = U_{cc}$ Đầu ra ở mức cao (1)

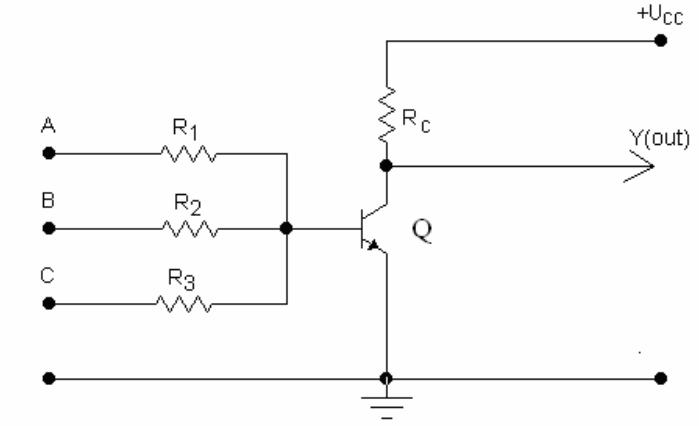
Khi ở đầu vào là 1 Tranistor hoạt động, dòng $I_c = I_{cmax}$. $U_C = 0$. Đầu ra ở mức thấp (0)

b/ Cổng logic NOR họ RTL:

Cổng NOR họ RTL có sơ đồ nguyên lý như hình 3.17.

Nếu một hoặc tất cả các lối vào ở mức 1. Transistor sẽ được định thiên. Dòng I_b đủ lớn làm cho Q bão hòa $I_c=I_{cmax}$, $U_c= 0$. Điện áp lối ra ở mức thấp. Lối ra Y ở trạng thái 0.

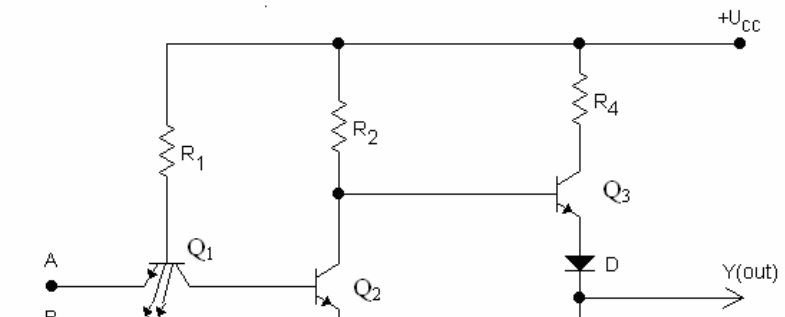
Nếu tất cả các lối vào đều ở mức 0. T_1 không được định thiên. Dòng $I_b= 0$, T không hoạt động $I_c= 0$ điện áp $U_c= U_{cc}$. Lối ra Y ở trạng thái 1 .



Hình 3.17: *Sơ đồ nguyên lý cổng logic NOR họ RTL*

3.6.4. Cổng logic họTTL (Transistor Transistor Logic)

Hình 3.18 là sơ đồ nguyên lý họ TTL thông dụng thực hiện cổng NAND có tốc độ tác động tương đối chậm do hảng Texas Intrument chế tạo.



Hình 3.18: *Sơ đồ nguyên lý mạch logic NAND họ RTL*

Q_1 là loại Transistor có nhiều cực phát (có bao nhiêu lối vào có bấy nhiêu cực phát), cực B của Q_1 được định thiên bởi R_1, lối ra là cực C được nối trực tiếp vào cực B của Q_2.Q_2 có tải vừa ở cực thu vừa ở cực phát, cực C của Q_2 được nối trực tiếp vào cực B của Q_3, cực E của Q_2 được nối trực tiếp vào cực B của Q_4. Như vậy, Q_2 định thiên cho Q_3 và Q_4 hoạt động. Q_4 là transistor ngõ ra. Tín hiệu ra lấy ở cực thu của Q_4. Transistor Q_3 là transistor đệm, có mục đích thay thế điện trở tải R_c của Q_4.

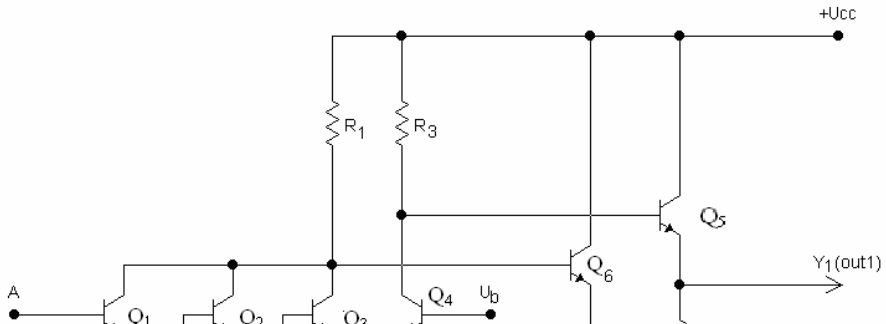
Khi một hoặc tất cả lối vào đều ở trạng thái 0, Q_1 thông. Điện áp ở cực C của Q_1 thấp,Q_2 không được định thiên, dòng cực thu I_c của nó bằng 0. $U_{c2} = U_{cc}$, Q_3 được định thiên nên thông. Điện áp ở cực E của Q_2 thấp nên Q_4 không được định thiên Q_4 tắt $I_{c4} = 0$. Điện áp lối ra bây giờ bằng điện áp cung cấp trừ đi sụt áp trên R_3và trên điện trở R_{ce} của tiếp giáp EC của Q_3. R_{ce} này rất bé do Q_3 đang hoạt động ở chế độ bão hòa.Vì vậy, ngõ ra Y ở trạng thái 1.

Khi tất cả các lối vào đều ở trạng thái 1.Tiếp giáp EB của Q_1 phân cực ngược còn tiếp giáp BC phân cực thuận. Q_1 làm việc ở chế độ nghịch đảo. Điện áp nguồn theo R_{bc} cung cấp thiên áp cho Q_2. Dòng I_b đủ lớn làm cho Q_2 bão hòa làm thông Q_4. Dòng I_{c4} cực đại do đó U_{c4}= 0. Lối ra Y ở trạng thái 0. Lúc này Q_3 vẫn tắt vì Q_3 không được cung cấp thiên áp.

3.6.5 Cổng logic họ ECL (Emitter Coupled Logic)

Tất cả các phương pháp chế tạo trên (RTL , DTL , TTL ...) có một nhược điểm chung là tốc độ tác động không nhanh. Nhược điểm này xuất phát từ chổ các Transistor được điều khiển đến chế độ bão hòa nên làm tăng thời gian chậm trễ. Để khắc phục nhược điểm này, người ta dùng một phương pháp chế tạo khác là công nghệ ECL.

Hình 3.19 là sơ đồ nguyên lý một mạch logic lập lại cực phát ECL thực hiện cổng logic OR và NOR



Hình 3.19: *Sơ đồ nguyên lý mạch logic NOR và OR họ ECL*

Khi tất cả lối vào đều ở trạng thái 0, tất cả các Transistor Q_1, Q_2, Q_3 đều tắt, điện áp ở cực thu của nó xấp xỉ V_{cc}. Các transistor này có cực C được nối với lối ra qua tầng lập lại cực phát (Q_6) nên ở lối ra Y_1 cũng ở trạng thái 1. Các cực E của Q_1, Q_2, Q_3 được nối chung với cực E của Q_4, khi các lối vào ở trạng thái 0 thì điện áp ở cực E của Q_4 cũng ở mức thấp, Q_4 thông. Điện áp ở cực thu của Q_4 xấp xỉ không. Cực C của Q_4 qua tầng lập lại cực phát (Q_5) nên ở lối ra Y_2 cũng ở trạng thái 0.

Nếu một hoặc tất cả lối vào ở trạng thái, các transistor tương ứng sẽ thông. Điện áp ở cực thu của chúng xấp xỉ 0. Qua tầng lập lại cực phát (Q_6) nên lối ra Y_1 cũng ở trạng thái 0. Tương tự, lối ra Y_2 ở trạng thái 1.

Trong các họ logic trên, họ logic TTL được sử dụng nhiều nhất do nó có nhiều ưu điểm: tốc độ đóng mở cao, điện áp ra đủ lớn, khả năng chống nhiễu lớn.

Họ ECL có ưu điểm như của TTL nhưng tiêu thụ công suất lớn, mức điện áp ra thay đổi theo nhiệt độ. ECL được dùng nhiều trong SSI và MSI có tốc độ cao và siêu cao.

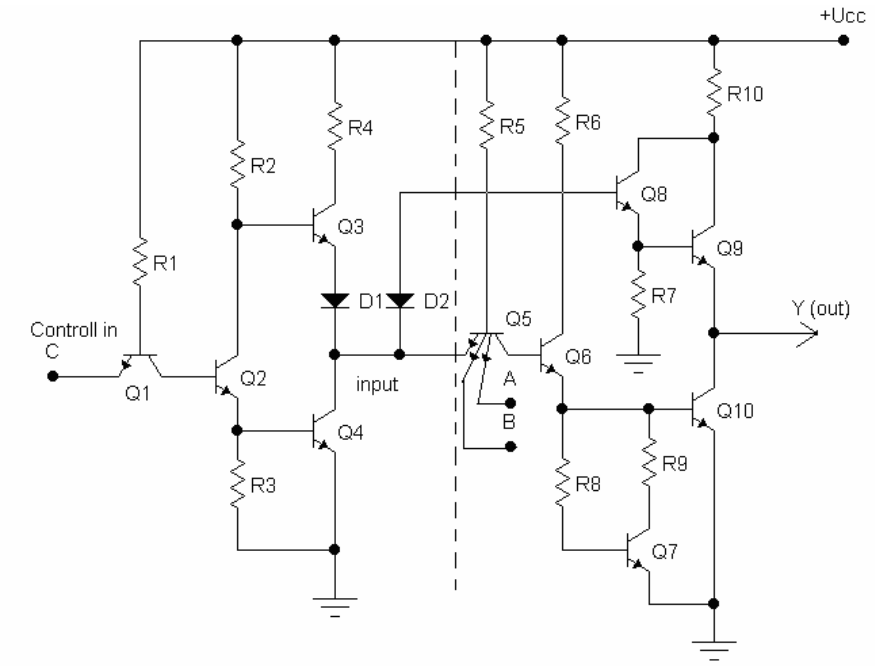
3.7. CỔNG LOGIC CÓ ĐẦU RA 3 TRẠNG THÁI TS (THREE STATE):

Hình 3.20 là sơ đồ nguyên lý cổng NAND ba trạng thái

Đầu ra Y của cổng NAND ba trạng thái có 3 trạng thái: cao, thấp, treo (trạng thái có trở kháng cao).

Phần bên phải đường đứt nét của sơ đồ chính là cổng NAND có hai ngõ vào A và B. Phần bên trái cũng là một cổng NAND có một ngõ vào, đây chính là ngõ vào điều khiển C (ngõ vào điều khiển 3 mức logic ở ngõ ra)

Khi đầu vào điều khiển C ở mức thấp (0), T_4 đưa ra tín hiệu ở mức cao (1) cho Q_5. Mạch NAND bên phải với hai đầu vào A và B thực hiện quan hệ logic NAND bình thường.



Hình 3.20: *Sơ đồ nguyên lý cổng NAND 3 trạng thái*

Khi đầu vào điều khiển C ở mức thấp (0), Q_4 đưa ra tín hiệu ở mức cao (1) cho Q_5. Mạch NAND bên phải với hai đầu vào A và B thực hiện quan hệ logic NAND bình thường.

Khi đầu vào điều khiển C ở mức cao (1), Q_4 đưa tín hiệu ở mức thấp (0) cho NAND bên phải, làm cho Q_6 Q_7 Q_{10} đều ngắt, làm cho đầu ra Y có trở kháng cao (trạng thái treo)

3.8.CÁC KHỐI CỔNG LOGIC THÔNG DỤNG

Các mạch logic OR, AND, NO, NAND, NOR được gọi chung là các cổng logic.Trong hệ thống số, các cổng đó được sử dụng lập đi lập lại nhiều lần, do đó người ta sản xuất những mạch tích hợp logic chứa nhiều cổng, mà theo thói quen ta hay gọi là vi mạch logic. Đó là các vi mạch chứa một số cổng logic cơ bản trong một vỏ.

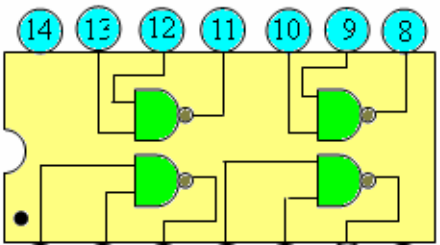
Thông dụng nhất là loại 14 chân chia làm 2 hàng. Các vi mạch họ TTL được giới thiệu là họ 7400 và74LS00, loại CMOS thuộc họ 4000. Dùng phổ biến là loại SN7410 do Mỹ sản xuất gồm 3 cổng NAND, mỗi cổng có 3 lối vào, một lối ra tổng cộng có 12 chân, hai chân còn lại dùng cho nguồn cung cấp. Muốn biết thêm về chi tiết các vi mạch Logic, ta tham khảo sách số liệu (data book) hoặc sổ tay hướng dẫn (handbook).

3.8.1: Vi mạch logic loại TTL/LS:

Các hình sau cho ta cấu trúc bên trong một vài vi mạch logic họ TTL/LS

a/ 7400/74LS00 (Hình 3.21)

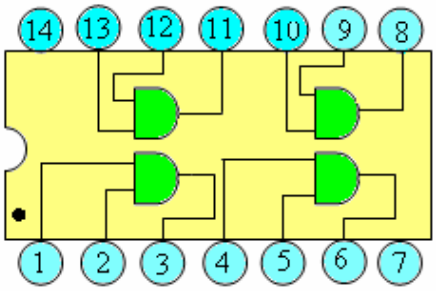
Chứa bốn cổng NAND. Đây là một trong ngững khối cổng cơ sở để thiết kế các mạch số và rất dễ sử dụng. Nguồn cung cấp U_{cc} = +5V



Hình 3.21: *Cấu trúc bên trong của vi mạch logic 7400/ 74LS00*

b/ 7408/74LS08 (Hình 3.22)

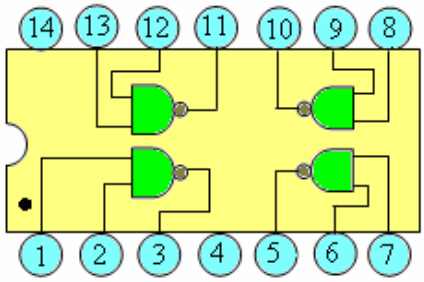
Chứa bốn cổng AND. Không thông dụng. Nguồn cung cấp Ucc = +5V



Hình 3.22: *Cấu trúc bên trong của vi mạch logic 7400/ 74LS00*

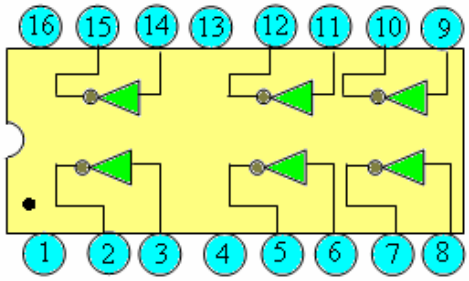
3.8.2: Vi mạch logic loại CMOS:

a/ 4011: Chứa 4 cổng NAND. Được sử dụng rất rộng rãi trong thiết kế mạch số. Chức năng tương tự 7400/74SL00. Nguồn cung cấp Ucc = +3V ÷ 15V. Các chân không sử dụng cần phải nối vào chân 7 (ground) hoặc 14 (U_{cc}). (Hình 3.23)



Hình 3.23: *Cấu trúc bên trong của vi mạch logic 4011*

b/ 4049: Chứa 6 cổng NO. Ngoài các ứng dụng đảo tín hiệu logic và phối ghép CMOS - TTL, nó thường được dùng trong các bộ dao động và phát xung. Nguồn cung cấp Ucc = +3V ÷ 15V. (Hình 3.23)



Hình 3.23: *Cấu trúc bên trong của vi mạch logic 4049*

CHƯƠNG IV TRIGGER

4.1 KHÁI NIỆM CHUNG:

4.1.1.Mô tả Trigger và hoạt động

Quá trình gia công và xử lý số liệu trong hệ thống số đòi hỏi ta phải nhớ tạm thời và nhớ lâu dài các thông tin. Trigger hay còn gọi là mạch lật (Flip-Flop) là một phần tử nhớ thông tin cơ bản nhất.

Trigger là một phần tử logic có 2 trạng thái ổn định và có thể xem như là một ôtômat cơ bản trong lý thuyết ôtômat vì trên cơ sở trigger ta có thể tổng hợp nhiều loại ôtômat khác nhau.

Trigger là một phần tử có nhiều đầu vào và hai đầu ra. Hai đầu ra có tính liên hợp nghĩa là đầu này là đảo của đầu kia và ngược lại. Ta thường ký hiệu Q và $\overline{Q}$, Q và $\overline{Q}$ chỉ có thể có hai trạng thái là 0 và 1 hay thấp (B) và cao (H).

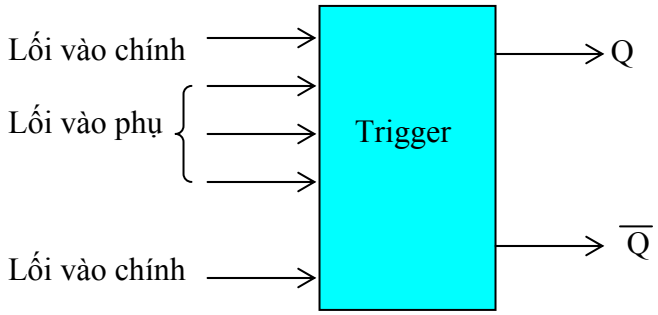
Các đầu vào điều khiển trạng thái logic của Q và hai đầu ra và được gọi tên tùy theo loại Trigger. Trạng thái của các đầu ra không những phụ thuộc ở các đầu vào mà còn phụ thuộc vào trạng thái quá khứ của nó. Nghĩa là trong một điều kiện logic như nhau của các đầu vào, đầu ra có thể chuyển trạng thái hoặc không tùy theo trước khi có kích thích nó đang ở trạng thái nào.

Về thông tin, Trigger chỉ làm nhiệm vụ nhớ thông tin chứ không làm biến đổi thông tin. Mỗi trigger chỉ nhớ một bit thông tin. Trạng thái của trigger xác định ở đầu ra của nó và thường chú ý ở đầu ra Q.

Trigger còn gọi là mạch lật Flip Flop

4.1.2.Hoạt động của Trigger

Trigger có ký hiệu tổng quát như hình 4.1



Hình 4.1: *Ký hiệu của một Trigger*

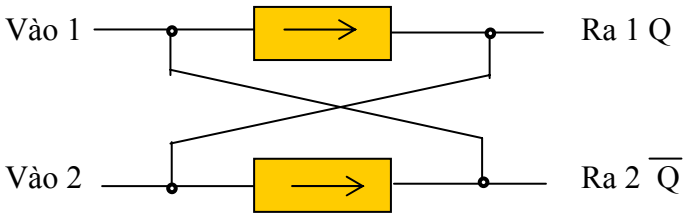
Về cấu tạo chi tiết các loại Trigger có thể khác nhau, nhưng mỗi Trigger vẫn có thể được coi như gồm hai phần chính:

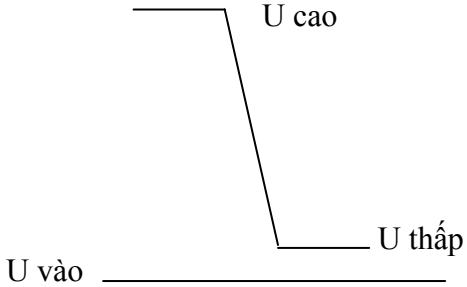
* Phần cơ bản của Trigger.

* Phần điều khiển.

a/Phần cơ bản của Trigger

Phần cơ bản của một Trigger gồm hai mạch điện tử giống nhau. Mỗi mạch có một hoặc nhiều đầu vào và một đầu ra (Hình 4.2) với sự quan hệ về mức độ điện thế giữa đầu vào và đầu ra như hình 4.3





Hình 4.2: *Phần cơ bản của một Trigger* Hình 4.3: *Quan hệ điện áp đầu vào và đầu ra*

Mỗi mạch như trên có đặt tính của hàm NOT (hay hiệu ứng của hàm NOT chứ không nhất thiết phải là mạch NOT). Nó được nối với nhau theo kiểu: đầu ra 1 được đấu vào đầu vào 2 và ngược lại, việc nối như vậy tạo thành vòng hồi tiếp.

Giả sử đầu ra của mạch 1 ở trạng thái cao (Q =1), như vậy đầu vào của mạch 2 cũng ở trạng thái cao, đầu ra của mạch 2 phải ở trạng thái thấp ($\overline{Q} = 0$). Trạng thái này thỏa mãn một cách chính xác trạng thái đầu vào của mạch 1 là trạng thái thấp. Cả hai đầu vào đều thỏa mãn, mạch ở trạng thái ổn định (hay còn gọi là trạng thái bền vững). Tương tự ta thấy trạng thái Q = 0 (đầu ra của mạch 1 ở trạng thái thấp) cũng là một trạng thái ổn định với $\overline{Q} = 1$.

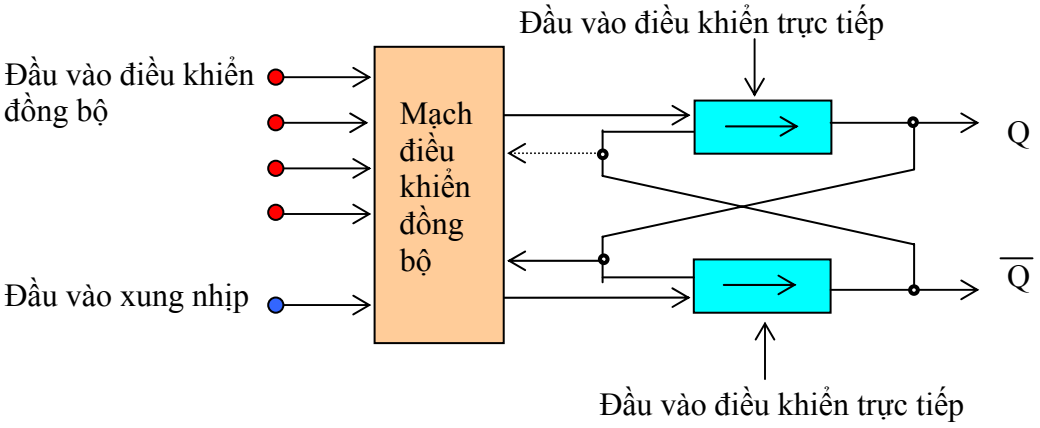
Tóm lại, một Trigger cơ bản có hai trạng thái ổn định. Nếu ta không có gì thay đổi ở mạch thì nó có thể ở một trạng thái ổn định và sẽ giữ mãi như thế. Thực tế việc này không có lợi vì ta không biết trước được khi cung cấp nguồn thì mạch đang ở trạng thái nào (Q = 0 hay Q = 1).

Để chủ động, ta cần kiểm soát được trạng thái của mạch và làm mạch thay đổi theo ý muốn, muốn được như vậy cần phải có thêm phần điều khiển.

b/ Phần điều khiển

Phần điều khiển Trigger có hai loại chính: Điều khiển trực tiếp và điều khiển đồng bộ (Hình 4.4). Các đầu vào điều khiển trực tiếp thường được đưa vào trực tiếp hai mạch thành phần của Trigger, chúng dùng để xác định trực tiếp trạng thái của Q hoặc buộc Q phải ở một trong hai trạng thái 1 hoặc 0. Khi một hoặc các đầu vào điều khiển trực tiếp đang hoạt động thì Q không tuân theo trạng thái của các đầu vào đồng bộ. Các đầu vào này thường dùng để xác định trước trạng thái của Q.

Các đầu vào đồng bộ điều khiển Trigger cơ bản qua trung gian của một mạch điều khiển đồng bộ, dưới sự kiểm soát của một xung nhịp được đưa vào theo một đầu vào riêng. Các đầu ra Q và $\overline{Q}$ chịu sự điều khiển của trạng thái logic của các đầu vào này khi có xung nhịp (vì vậy mà có tên đồng bộ).



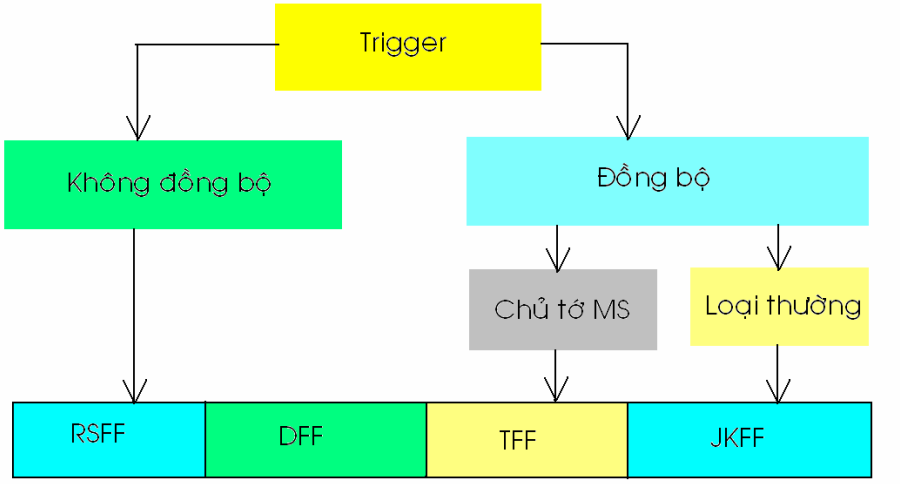
Hình 4.4: Sơ đồ khối của một Trigger

Khi không có xung nhịp, các đầu vào có thể thay đổi trạng thái mà không ảnh hưởng đến Q và $\overline{Q}$. Các đầu vào điều khiển đồng bộ thường có tên khác nhau và đây cũng chính là tên của Trigger. Mạch điều khiển đồng bộ có thể nhận các tín hiệu từ Q và $\overline{Q}$ đưa trở về.

4.1.3.Phân loại trigger

Trigger thường được phân loại theo đặc tính của các đầu vào. Những đặc tính này được ghi trong bảng chân lý, hay bảng mức độ điện thế (thấp - cao), cho thấy các trạng thái khác nhau của Q theo các trạng thái đầu vào.

Ta có các loại Trigger như hình 4.5.



***Hình 4.5:** Phân loại Triger*

Dựa vào phương thức hoạt động: Trigger được chia làm hai loại là đồng bộ và không đồng bộ.

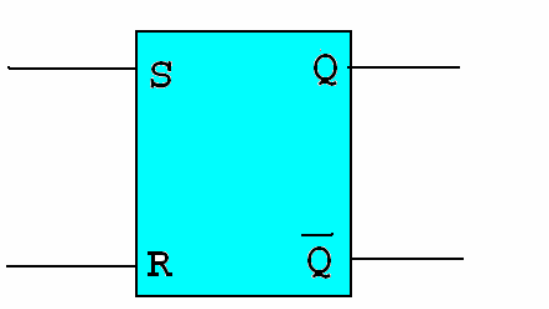
Dựa vào chức năng: Trigger được phân thành RSFF, JKFF, TFF, DFF. Các loại này khi hoạt động đồng bộ lại được chia thành loại chủ và tớ (Master - Slave).

Có nhiều loại trigger khác nhau, trong giáo trình này ta đề cập đến một số trigger thông dụng.

4.3. TRIGGER RS (RSFF SET - RESET FLIP FLOP)

4.3.1. RSFF điều khiển trực tiếp:

a/ Ký hiệu

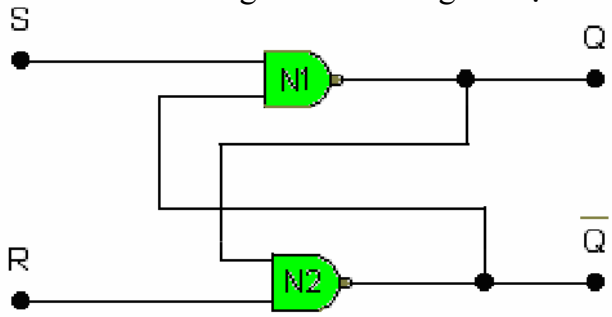


Hình 4.6: *Ký hiệu RSFF điều khiển trực tiếp*

Trigger RS là loại ôtômat có hai trạng thái, Có hai đầu vào mở (set) và đóng (reset), hai đâù ra bù trừ cho nhau Q và $\overline{Q}$. Ngoài ra SRFF cũng có một đầu vào đồng bộ C (đầu vào chuẩn hóa theo thời gian) để đồng bộ các hoat động giữa các đầu vào và đầu ra cũng như đồng bộ toàn hệ thống. RSFF có ký hiệu như hình 4.6

b/ Sơ đồ logic:

RSFF được cấu tạo gồm hai cổng NAND (hình 4.7), hoặc NOR, trong đó lối ra của cổng thứ nhất được nối vào lối vào của cổng thứ hai và ngược lại. Hai lối vào còn lại của hai cổng được để trống



Hình 4.7: *Sơ đồ logic của RSFF*

Trong họ TTL nếu để lơ lửng một lối vào nào đó thì lối vào đó đương nhiên ở mức 1. Muốn cho lối vào nào đó ở mức 0 ta nối nó với đất.

Giả sử ta nối S với đất, ta có:

* Cổng N_1 có một lối vào ở mức 0 (đâù vào S), lối ra sẽ là 1. Ta có Q = 1.

* Cổng N_2 có cả hai lối vào đều ở mức 1 vậy lối ra ở mức 0. Ta có $\overline{Q} = 0$

Ta có : Q = 1 , $\overline{Q} = 0$ (4.1)

Ngoài ra lối ra Q lại đưa ngược trở lại đầu vào của N_1. Cổng N_1 có một lối vào bằng 0 nên lối ra luôn luôn bằng 1. Q = 1 ngay cả khi S không còn nối đất nữa.

Tương tư, nếu R nối đất thì Q = 0 , $\overline{Q} = 1$ (4.2)

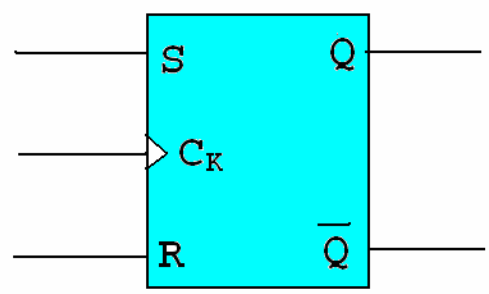
Các trạng thái (4.1) và (4.2) vẫn giữ nguyên dù ta không còn nối đất S và R (nghĩa là khi một xung tạo nên một trạng thái thì trạng thái đó vẫn duy trì ổn định sau khi xung đã hết và đó là đặc tính của nhớ).

4.3.2. RSFF điều khiển đồng bộ:

RSFF có khuyết điểm là điều khiển trực tiếp. Khi muốn cho các trạng thái của trigger thay đổi đồng bộ với các xung nhịp, ta dùng loại RSFF điều khiển đồng bộ.

a/ Ký hiệu

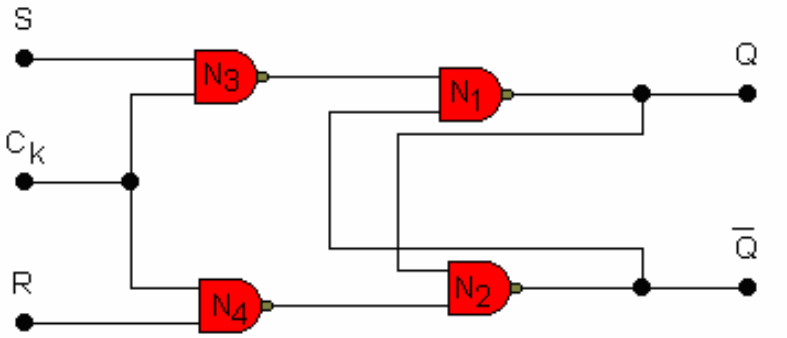
RSFF điều khiển đồng bộ có ký hiệu như hình 4.8. C_K là nơi nhận xung đồng bộ, đây là một chuỗi xung vuông, rất đều nên còn gọi là xung đồng hồ (clock pulse), dùng để đồng bộ hoạt động của các Trigger.



Hình 4.8: *Ký hiệu RSFF đồng bộ.*

b/ Sơ đồ logic:

Để có thể điều khiển đồng bộ, ta đưa thêm hai cổng N_3 và N_4. Xung nhịp C_K được đưa đồng thời vào N_3và N_4 (Hình 4.9)



Hình 4.9: *Sơ đồ logic của RSFF được điều khiển bởi xung nhịp C_k*

Từ sơ đồ nguyên lý trên và từ tính chất của cổng NAND ta có:

Khi xung nhịp C_k = 0 (không có xung nhịp đến):

Lối ra của N_3 và N_4 luôn luôn ở mức 1, bất kỳ R,S có giá trị nào.

Nếu Q = 1 thì nó giữ nguyên là 1

Nếu Q = 0 thì nó giữ nguyên là 0

Nghĩa là trigger không thay đổi trạng thái khi không có xung nhịp đến.

Khi xung nhịp $C_k = 1$ (có xung nhịp đến):

S = 0, R = 0: Các lối ra của N_3 và N_4 đều là 1. Lý luận tương tự như trên ta thấy các trạng thái của các triger vẫn không đổi. Nghĩa là:

$Q_n = Q_{n+1}$ Khi S = R = 0

S = 0, R = 1: Lối ra của N_3 là 1 lối ra của N_4 là 0 vì vậy lối ra của N_1 là 0 (Q = 0) còn lối ra của N_2 là 1 ($\overline{Q} = 1$). RSFF chuyển đến trạng thái tắt.

Nghĩa là: Q = 0 và $\overline{Q} = 1$ Khi S = 0 , R =1

S = 1, R = 0: Lối ra của N_3 là 0, lối ra của N_4 là 1 vì vậy lối ra của N_1 là 1 (Q =1) còn lối ra của N_2 là 0 ($\overline{Q} = 0$). RSFF chuyển đến trạng thái mở.

Nghĩa là: Q = 1 và $\overline{Q} = 0$ Khi S =1 , R = 0

S =1, R =1: Lối ra của N_3 là 0, lối ra của N_4 cũng là 0 vì vậy lối ra của N_1 là 1 (Q =1) và lối ra của N_2 cũng là 1 ($\overline{Q}$=1). Điều này không phù hợp về mặt logic vì hai lối ra Q và $\overline{Q}$ phải ngược trạng thái nhau. Tùy theo lối vào nào tăng từ 0 lên 1 nhanh hơn và tùy theo tính chất không đối xứng của mạch mà ta có một trong hai trạng thái sau:

Hoặc là Q = 1 , $\overline{Q} = 0$

Hoặc là Q = 0 , $\overline{Q} = 1$

Nghĩa là khi S = R = 1 thì RSFF dẫn đến trạng thái không rõ ràng, cần phải tránh. Cặp SR = (11) bị cấm, không sử dụng.

Ta có bảng trạng thái của RSFF như bảng 4.1 :

Bảng 4.1: *Bảng trạng thái của Trigger RS*

S	R	Q_{n+1}	Trạng thái tiếp theo
0	0	Q_n	RSFF giữ nguyên trạng thái cũ
1	0	1	RSFF chuyển đến trạng thái mở
0	1	0	RSFF chuyển đến trạng thái tắt
1	1	?	RSFF lập lờ, không xác định

4.3.3: Phương trình đặc trưng của RSFF:

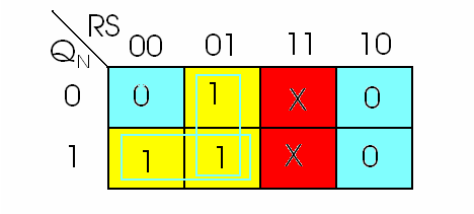
Bảng trạng thái đầy đủ của RSFF được trình bày ở bảng 4.2:

Bảng 4.2: *Bảng trạng thái đầy đủ của RSFF*

Q_n	R	S	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	X

1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	X

Ta tối thiểu hóa hàm trạng thái bằng bảng Karnaugh của RSFF



Từ bảng Karnaugh ta được:

$$Q_{n+1} = \overline{R}\,\overline{S}Q_n + R\overline{S}Q_n + \overline{R}S\overline{Q}_n + \overline{R}SQ_n$$

$$Q_{n+1} = \overline{R}Q_n + \overline{R}S$$

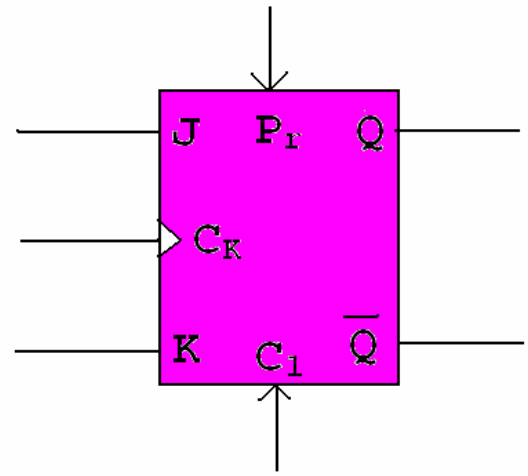
$$Q_{n+1} = \overline{R}Q_n + S$$

Đây chính là phương trình đặc trưng của RSFF

4.4. TRIGGER JK (JUMP KEEP FLIP FLOP)

Để khắc phục trường hợp S = R = 1 của Trigger RS người ta chế tạo loại JKFF, JKFFlà loại Trigger vạn năng có nhiều ứng dụng nhất trong điện tử số. JKFF có ký hiệu như hình 4.10

4.4.1 Ký hiệu:



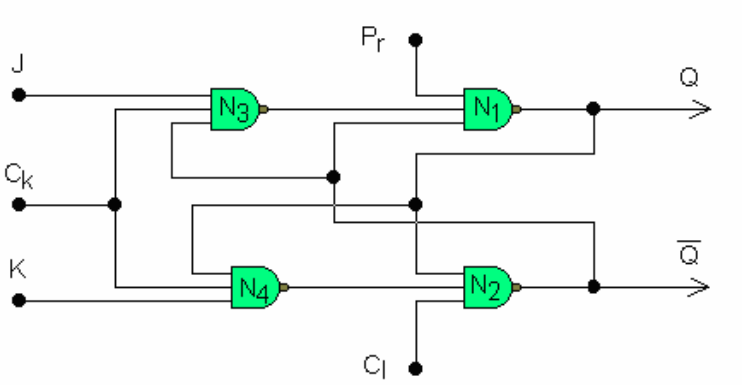
Hình 4.10: *Ký hiệu của JKFF*

Ngoài các đầu vào như RSFF, JKFF còn có thêm hai đầu vào đặc biệt

- Pr (Preset) : Đặt trước trạng thái 1 cho JKFF
- Cl (Clear) : Xóa cho đầu ra Q của JKFF về 0

4.4.2 Sơ đồ logic:

JKFF có sơ đồ logic như hình 4.11



Hình 4.11: *Sơ đồ logic của JKFF được điều khiển bởi xung nhịp C_k*

Sơ đồ nguyên lý của JKFF phức tạp hơn RSFF, nhưng chúng có ưu điểm là khi cả hai đầu vào đều có tín hiệu (J = K = 1) thì trạng thái tiếp theo là nghịch đảo của trạng thái cũ.

Bảng 4.3: *Bảng trạng thái của Trigger JK*

J	K	Q_{n+1}	Trạng thái tiếp theo
0	0	Q_n	JKFF giữ nguyên trạng thái cũ
1	0	1	JKFF chuyển đến trạng thái mở
0	1	0	JKFF chuyển đến trạng thái tắt
1	1	$\overline{Q}_n$	Nghịch đảo trạng thái cũ

Như thế 3 trạng thái đầu của JKFFgiống như SRFF nhưng trạng thái thứ tư (trạng thái không rõ ràng của RSFF) sẽ là:

Khi J=1 , K=1 thì $Q_{n+1} = \overline{Q}_n$. Nghĩa là với một tuần tự xung nhịp vào thì đầu ra sẽ chuyển lần lược (1,0,1,0...).Khái niệm luân phiên nhau các trạng thái đối lập được gọi là sự bập bềnh (basculement) và có khi còn gọi là đóng mở (on-off).

Bằng cách chưng minh tương tự như RSFF, ta có phương trình đặc trưng của JKFF:

$$Q_{n+1} = J\overline{Q} + \overline{J}Q$$

4.4.3.Tác dụng của các đầu vào đặc biệt:

a/Đầu vào chỉnh trước (preset): Chỉnh trước một Trigger nghĩa là làm cho trạng thái của đầu ra Q = 1 khi không có xung nhịp đến.

Khi $P_r = 0$ (nối Pr xuống đất); $C_l = 1$; $C_k = 0$ thì Triger được đặt trước trạng thái 1 (Q = 1)

. ***b/Đầu vào xóa (clear):*** Xóa một Triger nghĩa là làm cho trạng thái của đầu ra Q = 0 khi không có xung nhịp đến.

Khi $C_l = 0$; $P_r = 1$; $C_k = 0$ thì Triger có trạng thái 0.

Chú ý rằng các lối vào C_l và P_r là những lối vào không đồing bộ, nghĩa là nó không cần đồng bộ với các xung nhịp, các xung nhịp có thể được đưa vào bất kỳ lúc nào sau khi trạng thái của Triger đã được đặt trước một cách không đồng bộ (nhờ các lối vào P_r và C_l). Các lối vào

không đồng bộ này phải được đưa về các trạng thái $P_r = 1$, $C_l = 1$ trước khi các xung nhịp đến để cho Triger có thể làm việc với các xung nhịp này.

Bảng 4.3 cho ta các điều kiện cần thiết cho phép Triger làm việc đồng bộ với các xung nhịp (dòng 1) và để xóa (dòng 2) hoặc đặt trước (dòng 3).

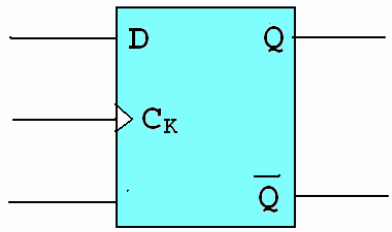
Bảng 4.3: *Điều kiện hoạt động của Cl và Pr*

	C	C_l	P_r	Q
Cho phép	1	1	1	
Xóa	0	0	1	
Đặt trước	0	1	0	1

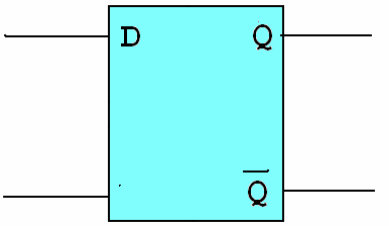
4.5.TRIGGER D (DELAY FLIP FLOP)

4.5.1. Ký hiệu:

Trigger D là loại FF chỉ có một đầu vào điều khiển. được dùng nhiều trong việc lưu trữ trong các mạch số. Có ký hiệu như hình 4.12a,b



Hình 4.12a: DFF đồng bộ



Hình 4.12b: DFF không đồng bộ

Nó có phương trình đặc trưng là: $Q_{n+1} = D$ (4.1)

Bảng trạng thái:

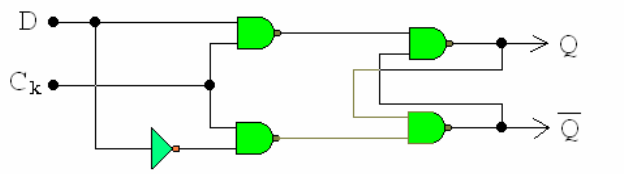
D	Q_{n+1}
0	0
1	1

D = 0 thì $Q_{n+1} = 0$ D = 1 thì $Q_{n+1} = 1$

Như vậy, với Trigger D thì trạng thái ở lối ra sau khi có xung nhịp đến giống trạng thái ở lối vào trước khi có xung nhịp đến. Nghĩa là tín hiệu đầu ra bị trễ so với tín hiệu đầu vào một khoảng thời gian nào đó. Trigger D được sử dụng làm đơn vị trễ, hoặc gọi là mạch chốt.

4.5.2. Sơ đồ logic:

DFF có sơ đồ nguyên lý như hình 4.13



Hình 4.13: *Sơ đồ logic của DFF được điều khiển bởi xung nhịp C_k*

4.5.3.Tạo DFF từ JKFF:

Ta có thể dùng JKFF để tạo ra một DFF:

Phương trình đặc trưng của JKFF:

$$Q_{n+1} = J\overline{Q}_n + \overline{J}Q_n \quad (4.2)$$

Chuyển 4.1 sạng dạng 4.2:

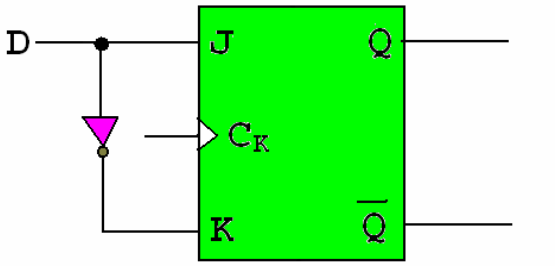
$$Q_{n+1} = D = D\,(\overline{Q}_n + Q_n) = D\,\overline{Q}_n + DQ_n \quad (4.3)$$

So sánh 4.3 và 4.1 ta có:

$$D = J$$

$$K = \overline{D}$$

Từ đó ta có sơ đồ của DFF được thiết kế từ JKFF như hình 4.14



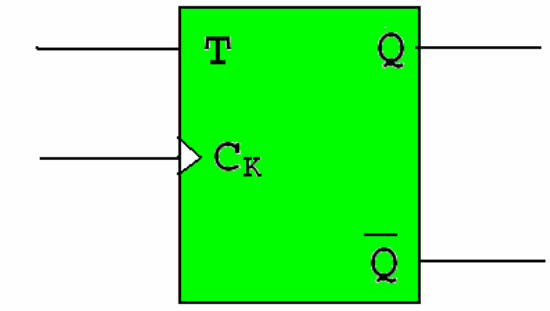
Hình 4.14: *Tạo DFF từ JKFF*

Nếu ta thêm vào JKFF một mạch đảo như hình vẽ. Sao cho K là nghịch đảo của J thì ta có Triger D :

4.6.TRIGGER T (TOGGLE)

4.6.1. Ký hiệu:

Trigger T có 2 đầu ra Q và $\overline{Q}_n$, một đầu vào T. TFF có chức năng duy trì và chuyển đổi trạng thái tuỳ thuộc tín hiệu đầu vào T. Có ký hiệu như hình 4.15



Hình 4.15: *Ký hiệu TFF*

Nó có phương trình đặc trưng của TFF là:

$$Q_{n+1} = T\overline{Q}_n + \overline{T}Q_n \quad (4.4)$$

Khi T = 0 thì $Q_{n+1} = Q_n$ (giữ nguyên trạng thái cũ)

Khi T = 1 thì $Q_{n+1} = \overline{Q}_n$ (nghịch đảo trạng thái cũ)

Bảng trạng thái:

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Nghĩa là:

T_n	Q_{n+1}
1	$\overline{Q}_n$
0	Q_n

Từ bảng trạng thái ta thấy Trigger T thay đổi trạng thái mỗi khi có xung nhịp đến. Như vậy với kích thích liên tục ở đầu vào thì đầu ra cũng thay đổi trạng thái liên tục.

4.6.2. Tạo TFF từ JKFF:

Phương trình đặc trưng của JKFF:

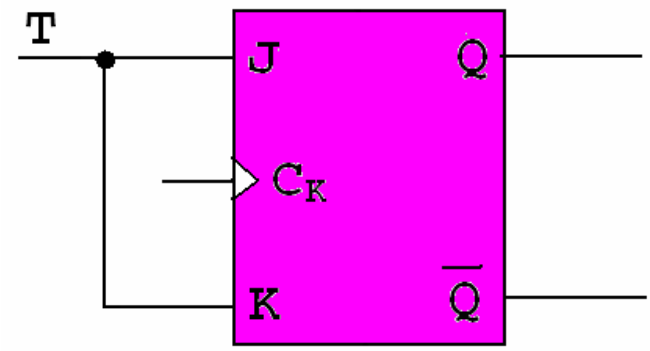
$$Q_{n+1} = J\overline{Q}_n + \overline{J}Q_n \quad (4.2)$$

So sánh 4.2 và 4.4 ta có:

J = T

K = T

Từ đó ta có sơ đồ của TFF được thiết kế từ JKFF như hình 4.16



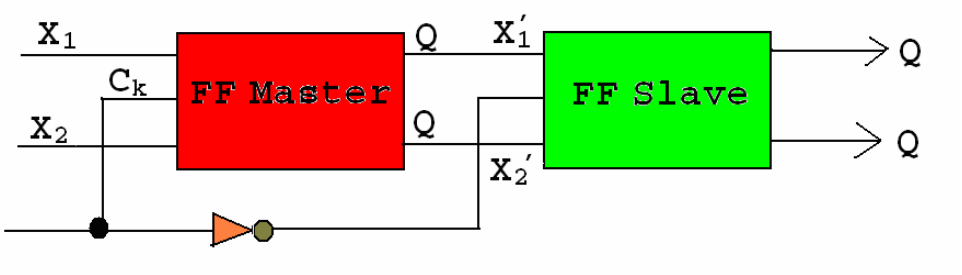
Hình 4.16: *Tạo TFF từ JKFF*

Từ hình 4.16 ta thấy JKFF có thể biến đổi thành TFF bằng cách cho J = K = 1. Nghĩa là ta nối 2 đầu J và K lại làm một và gọi là đầu vào T.

4.7: TRIGGER CHỦ - TỚ (MASTER – SLAVE)

4.7.1: Khái niệm:

Trigger chủ - tớ là Trigger có hai phần là hai Trigger, có hai khối điều khiển riêng nhưng lại có quan hệ với nhau. Một Trigger được gọi là chủ (master), một Trigger là tớ (slaver) (Hình 4.17)



Hình 4.17: *Sơ đồ khối một Trigger MS*

Trigger chủ thực hiện chức năng logic cơ bản của hệ

Trigger tớ nhớ trạng thái của hệ sau khi thực hiện xong việc ghi thông tin. Đầu ra của hệ là đầu ra của Trigger tớ. Mỗi trigger được điều khiển bằng một xung nhịp khác trạng thái, nghĩa là nếu C_k chủ = 0 thì C_k tớ bằng 1.

Dưới sự điều khiển của xung nhịp, việc ghi thông tin vào trigger MS được thực hiện qua 4 bước sau:

Bước 1: Cách ly chủ tớ.

Bước 2: Ghi thông tin vào chủ

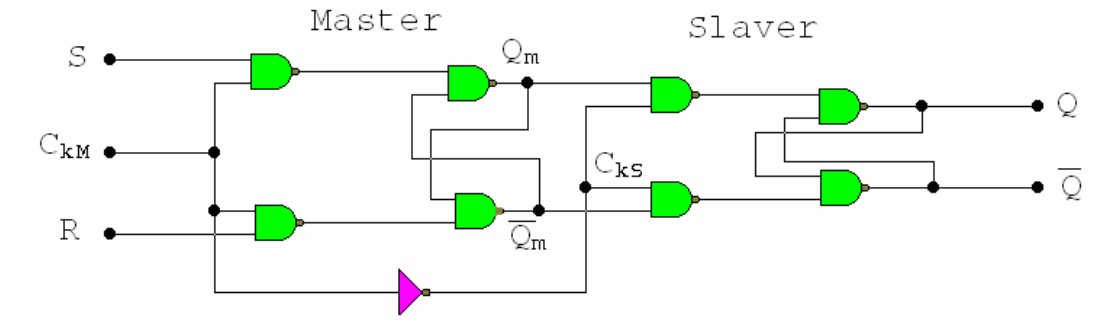
Bước 3: Cách ly giữa đầu vào và chủ

Bước 4: Chuyển thông tin từ trigger chủ sang trigger tớ.

Xung nhịp được đưa vào trigger chủ C_{kM} = 1 nên thông tin được đưa vào trigger chủ. Đầu vào đồng bộ của trigger tớ do tác động của cổng đảo nên không nhận xung đồng bộ C_{kS} = 0, trigger chủ và trigger tớ bị cách ly. Sau khi kết thúc xung đồng bộ C_{kM} = 0 làm cách ly giữa đầu vào và trigger chủ. Do tác động của cổng đảo C_{kS} = 1, trigger tớ mở chuyển thông tin từ trigger chủ sang trigger tớ. Để tránh loạn nhịp, quá trình ghi thông tin cần phải duy trì xung nhịp một cách chính xác.

4.7.2. RSFF chủ tớ:

RSFF có sơ đồ logic như hình 4.18



Hình 4.18: *Sơ đồ logic của Trigger RSFF chủ tớ*

RSFF chủ tớ gồm hai trigger mắc nối tiếp sử dụng chung một xung nhịp, nhưng trước khi đưa vào trigger tớ xung nhịp phải qua cổng đảo, nhờ vậy xung nhịp cung cấp cho hai trigger luôn luôn ngược dấu nhau.

Khi chưa có xung nhịp C_{kM} = 0: Trigger chủ ngắt, trigger tớ có C_{kS} = 1 nên hoạt động, nhận thông tin từ trigger chủ, cách ly tớ với đầu vào.

Khi có xung nhịp C_{kM} = 1: Trigger chủ nhận thông tin vào, C_{kS} = 0 trigger tớ bị ngắt, đầu ra Q và $\vec{Q}$ duy trì trạng thái cũ.

$$Q_m^{n+1} = S + \overline{R}Q_m^n$$

$$RS = 0$$

Khi C_{kM} đột biến xuống 0, trigger chủ ngắt, C_{kS} lên 1, trigger tớ tiếp nhận thông tin đã được trigger chủ ghi nhớ trước đó, trigger tớ chuyển trạng thái.

$$Q^{n+1} = S + \overline{R}Q$$

$$RS = 0$$