



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN

NGUYỄN KIM GIAO

KỸ THUẬT ĐIỆN TỬ SỐ



NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

EBOOKBKMT.COM

NGUYỄN KIM GIAO

KỸ THUẬT ĐIỆN TỬ SỐ

NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

LỜI NÓI ĐẦU

Môn học kỹ thuật số (còn được gọi là kỹ thuật điện tử số) được chọn làm môn cơ sở trong chương trình đào tạo của khoa Điện tử Viễn thông - trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

Thời lượng môn học này chiếm 4 đơn vị học trình lý thuyết, hai đơn vị học trình thực tập (mỗi đơn vị học trình lý thuyết là 15 tiết học, mỗi đơn vị học trình thực tập là 30 tiết sinh viên thực tập ở phòng thí nghiệm).

Phần lý thuyết giáo trình cung cấp cho sinh viên những kiến thức cơ sở về kỹ thuật điện tử số, hiểu rõ nguyên lý hoạt động, chức năng của các vi mạch logic, biết được các phương pháp thiết kế logic để tạo được các môđun tổ hợp lớn. Môn học kỹ thuật điện tử số có thể được xem là môn học hỗ trợ cho môn Cấu trúc máy tính, đo lường điều khiển ghép nối vi tính. Nó cung cấp cho sinh viên những kiến thức cơ sở để sinh viên có thể dễ dàng hiểu sâu hơn các khối chức năng trong cấu trúc phần cứng của máy tính.

Đối với sinh viên thuộc chuyên ngành kỹ thuật điện tử và viễn thông cần phải hiểu rõ yêu cầu của môn học này là: hiểu rõ chức năng của các mạch logic, biết phương pháp thiết kế logic và sử dụng được các vi mạch logic có mức độ tổ hợp nhỏ và vừa (SSI, MSI) để thiết kế được các khối chức năng dùng trong kỹ thuật điện tử, đo lường số, xử lý số và điều khiển số.

Phần bài tập thực hành có trong cuối các chương mục, để sinh viên tự ôn luyện trau dồi kiến thức cơ sở tập làm quen sử dụng vi mạch để thiết kế các môđun chức năng dùng trong kỹ thuật số. Đây cũng là những bài tập bắt buộc nếu sinh viên trong quá trình học không làm được các bài tập này thì không thể nào qua được kỳ kiểm tra tiên quyết để đạt được điều kiện thi.

Phần thực tập ở phòng thí nghiệm, giúp sinh viên hiểu rõ chức năng của các vi mạch logic, làm quen với phương pháp đoán nhận kiểm tra chức năng logic của vi mạch có mức độ tổ hợp vừa (MSI) ghép nối thành các môđun chuyên dụng trong kỹ thuật điện tử số.

Là sinh viên khoa điện tử viễn thông sinh viên cần phải biết vận dụng các thành tựu mới của công nghệ thông tin, phải biết sử dụng các chương trình phần mềm chuyên dụng trong lĩnh vực điện tử để phục vụ cho việc nâng cao hiểu biết của mình trong lĩnh vực kỹ thuật điện tử tương tự và kỹ thuật điện tử số, biết thiết kế sơ đồ nguyên lý, thiết kế mạch in, nghiên cứu

thiết kế mô phỏng các mạch điện tử số trên máy vi tính. Trong thời gian thực tập kỹ thuật số ở phòng thí nghiệm, thực tập làm đồ án học kỳ, đồ án tốt nghiệp, sinh viên cần tranh thủ học các chương trình phần mềm Protel, CircuitMaker để thiết kế mạch điện tử, thiết kế mô phỏng các vi mạch điện tử số. Trong chương cuối của giáo trình có giới thiệu chương trình phần mềm CircuitMaker, hướng dẫn cho sinh viên biết cách dùng chương trình phần mềm CircuitMaker để thiết kế vẽ các sơ đồ nguyên lý các mạch điện tử, thiết kế các thí nghiệm mô phỏng các mạch điện tử số.

Để phát huy thế mạnh và đặc thù của trường Đại học Công nghệ - ĐHQGHN, chú trọng nghiên cứu phát triển các ứng dụng công nghệ thông tin trong đào tạo, khuyến khích sinh viên học tin học và ứng dụng tin học trong lĩnh vực điện tử viễn thông, sinh viên đã coi vi tính là công cụ học tập, công cụ hành nghề của mình khi ra trường, các bài tập và thí nghiệm thực hành nêu trong giáo trình này đã được thiết kế mô phỏng để có thể tiến hành các thí nghiệm thực hành ngay trên máy vi tính. Sinh viên có thể tìm hiểu chức năng của của các vi mạch số, hoạt động của chúng thông qua các thí nghiệm mô phỏng này.

Chúng tôi hi vọng cuốn sách này sẽ là người bạn đồng hành, giúp ích cho sinh viên trong quá trình học tập lý thuyết cũng như thực hành kỹ thuật điện tử số và vi mạch số. Mặc dù đã cố gắng biên soạn song chắc chắn giáo trình không tránh khỏi những thiếu sót. Chúng tôi mong muốn nhận được nhiều ý kiến nhận xét, phê bình của bạn đọc và các bạn đồng nghiệp để giáo trình được sửa chữa, bổ sung hoàn thiện hơn cho lần xuất bản sau.

Xin chân thành cảm ơn những ý kiến đóng góp của bạn đọc.

Tác giả

Chương I

CÁC HỆ THỐNG ĐẾM VÀ MÃ SỐ

1.1. CÁC HỆ THỐNG ĐẾM

Để biểu diễn các đại lượng, người ta dùng các con số và một hệ thống đếm nào đó. Hệ thống đếm là phương pháp biểu diễn các số đo bằng tập hợp các ký hiệu gọi là các chữ số (Digits). Số lượng các chữ số dùng trong một hệ thống đếm gọi là *cơ số của hệ đếm* đó.

Ví dụ : Trong hệ thống đếm thập phân người ta dùng 10 chữ số làm cơ sở để xây dựng nên hệ thống đếm này là : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**. Hệ đếm thập phân có cơ số là 10.

Hệ đếm nhị phân chỉ dùng 2 chữ số **0** và **1**. Cơ số của hệ đếm này là 2

Hệ thống đếm được phân ra làm hai loại : định vị và không định vị

Hệ thống đếm định vị: giá trị của các chữ số trong hệ thống đếm này được xác định theo vị trí của nó trong cách viết con số, ở các hệ thống này, giá trị về số lượng của mỗi chữ số không những phụ thuộc vào kí hiệu của nó mà còn phụ thuộc vào vị trí của nó nằm trong hàng nào của con số. Các hệ thống đếm vừa kể trên thuộc loại này. Ví dụ trong hệ đếm thập phân chữ số 1 ở con số 201 có giá trị là 1 đơn vị, còn số 1 ở 102 có giá trị là 100 đơn vị.

Hệ thống đếm không định vị: ở hệ thống đếm này, giá trị về số lượng của mỗi chữ số không phụ thuộc vào vị trí của nó nằm trong con số. Ví dụ như hệ đếm chữ số La Mã: XXIII là số 23, XXXIX là số 39. Chữ số X nằm ở vị trí nào cũng có giá trị là 10.

Trong giáo trình này ta chỉ quan tâm đến các hệ thống đếm định vị.

Hệ thống đếm thập phân

Hệ thống đếm thập phân được loài người quen dùng trong đời sống hàng ngày từ trước đến nay, thói quen này có nguồn gốc từ xa xưa con người đã dùng 10 ngón tay để đếm. Trong hệ thống đếm này dùng 10 chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Cơ số của hệ đếm là 10, ta gọi là hệ thập phân hoặc hệ

10 (Decimal). Hệ đếm thập phân có ưu điểm là nó được dùng lâu đời và phổ biến, nó có một cơ sở hệ thống toán học rất hoàn chỉnh được con người qua các thế hệ không ngừng bổ sung phát triển hoàn thiện. Con số được viết trong hệ thập phân mọi người đều có thể hiểu được

Hệ thống đếm nhị phân

Hệ đếm nhị phân chỉ dùng có hai chữ số: **0** và **1**, cơ số hệ đếm là 2, ta còn gọi là hệ đếm 2 (Binary). Hệ đếm này được dùng trong kỹ thuật số và máy tính. Có một mối quan hệ tương đồng giữa hệ đếm nhị phân, đại số logic, các trạng thái vật lí của các mạch điện:

Hệ thống đếm nhị phân	1	0
Lôgíc	Đúng (TRUE)	Sai (FALSE)
Điện áp	Cao (HIGH)	Thấp (LOW)
Công tắc	Đóng (ON)	Ngắt (OFF)

Hệ đếm nhị phân có những ưu điểm sau đây:

Hai số 1 và 0 trong hệ đếm nhị phân có thể được dùng để đặc trưng cho hai trạng thái vật lí của mạch điện: role, công tắc điện đóng hoặc ngắt, có hoặc không có dòng điện chạy trong mạch điện, điện áp cao hoặc điện áp thấp ... Các thông tin vật lí một khi đã được số hoá biểu diễn dưới dạng nhị phân có thể lưu trữ và xử lí nhờ máy tính điện tử.

Việc sử dụng hệ đếm nhị phân trong kỹ thuật số, kỹ thuật máy tính cho phép ta áp dụng được đại số logic (đại số Boole) - một công cụ toán học rất sắc bén để thiết kế các vi mạch số các khối chức năng dùng trong kỹ thuật đo lường số, kỹ thuật máy tính và điều khiển tự động.

Máy tính chỉ làm việc với các thông tin biểu diễn dưới dạng nhị phân. Để đưa số liệu vào máy tính ta phải chuyển đổi các số liệu từ thập phân ra nhị phân để máy tính có thể làm việc được. Kết quả thu được trong quá trình tính toán xử lí thông tin nhờ máy tính lại phải chuyển đổi từ nhị phân ra thập phân để con người dễ dàng nhận biết.

Các con số biểu diễn trong hệ đếm nhị phân bao giờ cũng dài hơn nhiều so với các con số biểu diễn trong hệ thập phân. Biểu diễn một số lớn trong hệ đếm nhị phân cần phải dùng tới nhiều hàng, các hàng lại chỉ toàn các con số 0 hoặc 1 nên rất dễ bị nhầm lẫn khi đọc hoặc viết các con số này. Để khắc phục nhược điểm này trong kỹ thuật số người ta còn dùng các hệ thống đếm có cơ số là 8 và 16, cơ số của các hệ đếm này đều là 2 lũy thừa: $8=2^3$, $16=2^4$

nên các con số nhị phân chuyển sang các hệ này sẽ được rút gọn, ít bị nhầm lẫn hơn.

Hệ thống đếm bát phân

- Hệ đếm bát phân người ta dùng 8 chữ số: **0, 1, 2, 3, 4, 5, 6, 7**. Cơ số của hệ thống đếm là $8=2^3$, còn gọi là hệ đếm 8 (Octal). Mỗi chữ số của hệ bát phân có thể biểu diễn bằng một nhóm ba bit nhị phân.

Hệ thống đếm thập lục phân

Hệ thập lục phân dùng 16 chữ số, hệ có cơ số đếm là $16=2^4$, còn gọi là hệ đếm 16 hoặc hệ 16 (Hexadecimal). Các chữ số dùng trong hệ đếm này là: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**. Mỗi chữ số trong hệ đếm này có thể được biểu diễn bằng một nhóm 4 bit nhị phân.

Cách biểu diễn con số trong các hệ thống đếm

Trong các hệ đếm, một số được viết ra thường có ý nghĩa về số lượng như sau :

$$(1997)_{10} = 1.10^3 + 9.10^2 + 9.10^1 + 7.10^0$$

$$(1101)_2 = 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0$$

$$(46)_8 = 4.8^1 + 6.8^0$$

$$(2AE)_{16} = 2.16^2 + 10.16^1 + 14.16^0$$

Từ đó ta rút ra phương pháp tổng quát để biểu diễn một số A bất kỳ trong hệ đếm cơ số B nào đó có dạng sau :

$$(A)_B = A_n B^n + A_{n-1} B^{n-1} + A_{n-2} B^{n-2} + \dots + A_1 B^1 + A_0 B^0 + A_{-1} B^{-1} + \dots \quad (1.1)$$

trong đó: A_n, A_{n-1}, \dots là chữ số của các hàng; B^n, B^{n-1}, \dots là trọng số của các hàng.

Trong cách biểu diễn con số ở dạng số nhị phân mỗi một hàng người ta còn gọi là một *bit*, hàng ngoài cùng bên trái là *bit lớn nhất*, hàng tận cùng bên phải là *bit nhỏ nhất*.

Bảng 1-1 cho ta 21 số đầu tiên của các hệ thống đếm thông dụng.

Bảng 1-1 : 21 số đầu tiên của các hệ đếm thông dụng

Thập phân	Nhị phân	Bát phân	Thập lục phân
0	0	0	0
1	1	1	1
2	10	2	2

3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	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
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

1.2. CHUYỂN ĐỔI MỘT SỐ TỪ HỆ ĐẾM NÀY SANG HỆ ĐẾM KHÁC

1.2.1. Chuyển từ các hệ đếm khác sang hệ đếm thập phân

Để chuyển một con số viết ở hệ đếm bất kỳ sang hệ đếm thập phân ta viết chúng dưới dạng tổng quát (1.1) với các chữ số và trọng số.

Ví dụ :

$$(1101)_2 = 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0 = (13)_{10}$$

$$(123)_8 = 1.8^2 + 2.8^1 + 3.8^0 = (83)_{10}$$

$$(3FF)_{16} = 3.16^2 + 15.16^1 + 15.16^0 = (1023)_{10}$$

1.2.2. Chuyển từ hệ thập phân sang các hệ khác

Muốn chuyển một số nguyên ở hệ 10 sang hệ khác ta hãy làm phép chia liên tiếp số ở hệ 10 đó cho cơ số của hệ đếm cần chuyển sang cho đến khi kết quả chia bằng 0 (lần chia cuối cùng thứ n). Số dư của các lần chia chính

là các chữ số của các hàng và chúng được viết theo chiều ngược lại, nghĩa là các số dư của các lần chia thứ 1, 2, 3, ... n chính là các chữ số của các hàng thứ 1, 2, 3, ... n của kết quả cần tìm. Chú ý rằng số dư của lần chia đầu tiên nằm ở hàng có trọng số nhỏ nhất, số dư của lần chia cuối cùng n nằm ở hàng có trọng số lớn nhất.

Ví dụ :

Chuyển số $(35)_{10}$ sang hệ 2 ta làm như sau :

$$35 \mid 2$$

$$\text{Bít nhỏ nhất} \rightarrow 1 \ 17 \mid 2$$

$$1 \ 8 \mid 2$$

$$0 \ 4 \mid 2$$

$$0 \ 2 \mid 2$$

$$0 \ 1 \mid 2$$

$$\text{Bít lớn nhất} \rightarrow 1 \ 0$$

Lấy các số dư của các phép chia trên ta được kết quả :

$$(35)_{10} = (100011)_2$$

- Chuyển số $(35)_{10}$ sang hệ 8 ta làm tương tự :

$$35 \mid \underline{8}$$

$$\text{Hàng có trọng số nhỏ nhất} \rightarrow 3 \ 4 \mid \underline{8}$$

$$\text{Hàng có trọng số lớn nhất} \rightarrow 4 \ 0$$

$$\text{Vậy: } (35)_{10} = (43)_8$$

- Chuyển số $(35)_{10}$ sang hệ 16 ta cũng làm như sau :

$$35 \mid \underline{16}$$

$$\text{Hàng có trọng số nhỏ nhất} \rightarrow 3 \ 2 \mid \underline{16}$$

$$\text{Hàng có trọng số lớn nhất} \rightarrow 2 \ 0$$

$$\text{Kết quả : } (35)_{10} = (23)_{16}$$

Đổi phân lẻ sau dấu phẩy:

Trên đây ta làm phép chuyển một số nguyên ở hệ 10 sang hệ khác. Trong thực tế nhiều khi số ở hệ 10 còn có phần lẻ sau dấu phẩy. Trong công thức tổng quát (1.1) phần lẻ sau dấu phẩy là các số hạng có trọng số mang dấu mũ âm.

Để chuyển phần lẻ của một con số từ hệ đếm thập phân sang hệ đếm khác ta dùng thuật toán *nhân liên tiếp phần lẻ với cơ số của hệ thống đếm*

cần chuyển sang. Ví dụ, muốn chuyển phần lẻ ở hệ đếm thập phân sang hệ đếm nhị phân ta chuyển theo quy tắc như sau :

Lấy phần lẻ của số ở hệ 10 nhân với 2, nếu kết quả nhỏ hơn 1 ta được một số 0 đặt sau dấu phẩy của số ở hệ 2, nếu kết quả lớn hơn 1 ta có số 1 sau dấu phẩy ở hệ đếm 2. Tiếp theo ta lại lấy phần lẻ của kết quả nhận được trong phép nhân lần thứ nhất nhân tiếp với 2. Nếu kết quả lớn hơn 1 ta ghi thêm số 1 vào kết quả ở hệ 2 và lại lấy phần lẻ nhận được sau lần nhân thứ hai này nhân tiếp với 2. Quá trình sẽ chấm dứt khi phần lẻ bằng 0 hoặc đã đạt được số số lẻ sau dấu phẩy theo yêu cầu. Ví dụ : Đổi $(0,375)_{10}$ sang hệ 2 ta làm như sau :

$$\begin{array}{l} 0,375 \times 2 = 0,75 \\ 0,75 \times 2 = 1,5 \\ 0,5 \times 2 = 1,0 \end{array}$$

$$(0,375)_{10} = (0, 0 \quad 1 \quad 1)_2$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$\frac{1}{2^1} \quad \frac{1}{2^2} \quad \frac{1}{2^3}$$

Các trọng số của phần lẻ:

Đổi $(0,625)_{10}$ sang hệ 2, ta làm như sau:

$$\begin{array}{l} 0,625 \times 2 = 1,25 \\ 0,250 \times 2 = 0,5 \\ 0,5 \times 2 = 1,0 \end{array}$$

$$(0,625)_{10} = (0, 1 \quad 0 \quad 1)_2$$

Đổi $(15,625)_{10}$ sang hệ 2 ta được:

$$(15,625)_{10} = (1111,101)_2$$

Chú ý :

Các số lớn ở hệ 10 khi chuyển sang hệ 2 phải chia cho 2 nhiều lần, phép tính dài và dễ nhầm lẫn. Có một cách làm nhanh gọn hơn là ta hãy chuyển số hệ 10 đó sang hệ 16 trước, sau đó chuyển từng chữ số của hệ 16 thành số nhị phân 4 bit và hạ tương ứng xuống hàng dưới, ta sẽ được số hệ 2 biểu diễn số hệ 10 ban đầu. Ví dụ :

$$(144)_{10} = (9\ 0)_{16}$$

↓ ↓

$$(144)_{10} = (1001\ 0000)_2$$

$$(4091)_{10} = (\text{FFB})_{16} = (1111\ 1111\ 1011)_2$$

1.3. CÁC PHÉP TÍNH SỐ HỌC TRONG HỆ ĐẾM NHỊ PHÂN

1.3.1. Phép cộng

Phép tính cộng trong hệ thống đếm nhị phân cũng được thực hiện theo các quy tắc giống như phép cộng trong hệ thống đếm thập phân, chỉ khác là số nhớ chuyển lên hàng trên khi tổng các bit có cùng trọng số bằng 2

Quy tắc chung :

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Ví dụ:	6	ở hệ 2:	110
	<u>+4</u>		<u>+100</u>
	10		1010

1.3.2. Phép trừ

Phép tính trừ hai số trong hệ đếm nhị phân thực hiện giống như trừ hai số ở hệ đếm thập phân.

Ví dụ :

13	ở hệ 2:	1101
<u>-3</u>		<u>-0011</u>
10		1010

Về nguyên tắc, chúng ta có thể thực hiện phép tính trừ trong hệ thống đếm nhị phân như phép trừ trong hệ thống đếm thập phân. Tuy nhiên các nhà chế tạo máy tính không hay làm như vậy. Để thực hiện phép tính trừ: $A-B$ người ta thực hiện bằng phép tính cộng: $A+(-B)$ và sử dụng ngay bộ tổng trong máy tính để làm phép tính trừ. Vấn đề còn lại là làm cách nào để biểu diễn một số âm dưới dạng nhị phân. Trong máy tính, người ta dùng số bù nhị

phân để biểu diễn số âm và do đó dễ dàng thực hiện phép tính trừ bằng phép cộng với số bù nhị phân của số trừ.

Để tìm được số bù nhị phân, người ta làm theo các bước sau:

- Viết số đó dưới dạng nhị phân.
- Đảo các bit của từ nhị phân.
- Cộng thêm 1 vào hàng có trọng số bé nhất.

Ví dụ:

$$12 - 7 = 5 \quad (7)_{10} = 0111$$

$$\text{đảo các bit:} \quad 1000$$

$$\text{cộng thêm 1: } (-7)_{10} = 1001$$

Để thực hiện phép tính trừ $12 - 7 = 5$, ta thực hiện $12 + (-7) = 5$, bằng số nhị phân ta có:

$$\begin{array}{r} 12 \qquad 1100 \\ +(-7) \qquad +1001 \\ \hline 5 \quad \text{bit nhớ} \rightarrow 10101 \end{array}$$

Bit nhớ nếu là 1 thì đó là phép trừ, bit nhớ bằng 0 đó là phép cộng.

1.3.3. Phép nhân

Phép nhân trong hệ đếm nhị phân hoàn toàn tương tự như phép nhân thực hiện trong hệ đếm thập phân.

Quy tắc chung :

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

Ví dụ :

$$\begin{array}{r} 11 \text{ ở hệ 2: } 1011 \\ \times 3 \\ \hline 33 \end{array} \qquad \begin{array}{r} 1011 \\ \times 0011 \\ \hline 1011 \\ 10001 \end{array} \qquad \begin{array}{r} 2,5 \text{ ở hệ 2: } 10,1 \\ \times 6 \\ \hline 15,0 \end{array} \qquad \begin{array}{r} 10,1 \\ \times 110 \\ \hline 1010 \\ 101 \\ \hline 1111,0 \end{array}$$

1.3.4. Phép chia

Phép tính chia trong hệ đếm nhị phân cũng giống như trong hệ thập phân, áp dụng quy tắc nhân và trừ nhị phân ta có thể thực hiện phép chia nhị phân.

Ví dụ

$$\begin{array}{r}
 33 \lfloor_{11} \text{ ở hệ 2:} \quad 100001 \lfloor_{1011} \\
 0 \quad 3 \quad \quad \quad \quad \underline{-1011} \quad 11 \\
 \quad \quad \quad \quad \quad 01011 \\
 \quad \quad \quad \quad \quad \quad \underline{-1011} \\
 \quad \quad \quad \quad \quad \quad \quad 0000
 \end{array}$$

Nếu khi chia còn dư ta sẽ hạ thêm số 0 xuống và nhớ đánh dấu nảy ở kết quả rồi tiếp tục chia. Nếu còn dư mãi ta chỉ lấy đến một số số lẻ cần thiết.

Ví dụ :

$$\begin{array}{r}
 9,5 \lfloor_4 \text{ ở hệ 2:} \quad 1001,1 \lfloor_{100} \\
 15 \quad 2,375 \quad \quad \quad 000110 \quad 10,011 \\
 30 \quad \quad \quad \quad \quad \quad 100 \\
 20 \quad \quad \quad \quad \quad \quad \quad 0100 \\
 0 \quad \quad \quad \quad \quad \quad \quad \quad 0
 \end{array}$$

Trường hợp số bị chia nhỏ hơn số chia.

Ví dụ:

$$\begin{array}{r}
 7 \lfloor_{10} \text{ ở hệ 2:} \quad 111 \lfloor_{1010} \\
 0 \quad 0,7 \quad \quad \quad \quad 1110 \quad 0,1010 \\
 \quad \quad \quad \quad \quad \quad \quad \underline{-1010} \\
 \quad \quad \quad \quad \quad \quad \quad \quad 010000 \\
 \quad \quad \quad \quad \quad \quad \quad \quad \quad \underline{-1010} \\
 \text{dư} \quad \quad \quad 00110
 \end{array}$$

Tóm lại : Các phép tính số học cộng, trừ, nhân, chia ở hệ 2 được thực hiện tương tự như ở hệ 10 như ta vẫn thường làm từ trước đến nay.

1.4. MÃ HOÁ SỐ CỦA HỆ THẬP PHÂN

Trong kỹ thuật số để chuyển đổi các con số giữa 2 hệ đếm cơ số 2 và cơ số 10 một cách tự động người ta dùng phương pháp biểu diễn nhị thập phân. Người ta dùng một nhóm 4 bit nhị phân để biểu diễn mười con số của hệ đếm thập phân. Phương pháp biểu diễn này được gọi là phương pháp mã hoá các con số trong hệ đếm 10 bằng các nhóm mã hệ nhị phân (Binary-coded decimal BCD). Thực ra đó là số thập phân được viết kiểu nhị phân. Các chữ số của hệ 10 từ 0, 1, ... 9 đều được biểu diễn bằng 1 số nhị phân có 4 chữ số. Số nhị phân có 4 chữ số có trọng số 8 - 4 - 2 - 1 được gọi là mã BCD 8421 (hoặc mã BCD trọng số tự nhiên).

Ví dụ :

<u>Số hệ thập phân</u>	:	<u>Biểu diễn bằng mã BCD 8421</u>
16	:	$(0001\ 0110)_{BCD} \neq (10000)_2$
32	:	$(0011\ 0010)_{BCD} \neq (100000)_2$
2578	:	$(0010\ 0101\ 0111\ 1000)_{BCD}$

Mã BCD 8421 được dùng để chuyển các con số từ hệ 10 sang hệ 2 và ngược lại. Nhìn một con số lớn viết ở hệ nhị phân ta khó hình dung độ lớn của nó ở hệ 10. Nhưng viết ở mã BCD ta dễ hình dung ra độ lớn của nó.

Bảng 1-2: Các loại mã BCD có trọng số khác nhau

Số hệ 10	Mã BCD			
	Trọng số 8 4 2 1	Trọng số 7 4 2 1	Trọng số 2 4 2 1	Trọng số 5 1 2 1
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	0 1 0 1	0 1 0 1	1 0 0 0
6	0 1 1 0	0 1 1 0	1 1 0 0	1 0 0 1
7	0 1 1 1	1 0 0 0	1 1 0 1	1 0 1 0
8	1 0 0 0	1 0 0 1	1 1 1 0	1 0 1 1
9	1 0 0 1	1 0 1 0	1 1 1 1	1 1 1 1

Trong thực tế, đôi khi mã BCD 8421 dùng không thuận lợi. lúc đó người ta còn dùng các mã BCD có các trọng số 2421, 5121, 7421. Các loại mã BCD có trọng số khác đó biểu diễn các số hệ 10 từ 0 đến 9 cho trong bảng 1.2.

Ngoài mã BCD nói trên là những mã có trọng số ra, ta còn gặp một số mã thông dụng nữa không có trọng số được nêu ra trong bảng 1-3.

Bảng 1-3: các loại mã không có trọng số

Số hệ 10	Số hệ 2 B ₃ B ₂ B ₁ B ₀	Mã dư 3 A ₃ A ₂ A ₁ A ₀	Mã Gray G ₃ G ₂ G ₁ G ₀	Mã 2 trên 5 D ₄ D ₃ D ₂ D ₁ D ₀	MãJohnson J ₁ J ₂ J ₃ J ₄ J ₅
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 1 1	0 0 0 0 0
1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 1 0 1	1 0 0 0 0
2	0 0 1 0	0 1 0 1	0 0 1 1	0 0 1 1 0	1 1 0 0 0
3	0 0 1 1	0 1 1 0	0 0 1 0	0 1 0 0 1	1 1 1 0 0
4	0 1 0 0	0 1 1 1	0 1 1 0	0 1 0 1 0	1 1 1 1 0
5	0 1 0 1	1 0 0 0	0 1 1 1	0 1 1 0 0	1 1 1 1 1
6	0 1 1 0	1 0 0 1	0 1 0 1	1 0 0 0 1	0 1 1 1 1
7	0 1 1 1	1 0 1 0	0 1 0 0	1 0 0 1 0	0 0 1 1 1
8	1 0 0 0	1 0 1 1	1 1 0 0	1 0 1 0 0	0 0 0 1 1
9	1 0 0 1	1 1 0 0	1 1 0 1	1 1 0 0 0	0 0 0 0 1

Mã dư 3: (thừa 3) được tạo nên bằng cách cộng thêm 3 vào mã BCD 8421. Mã này dùng trong các thiết bị tính toán số học và xử lý tín hiệu số.

Mã Gray: Đặc điểm của mã này là hai số kế tiếp nhau chỉ khác nhau có một bit. Vì vậy tốc độ đếm của mã Gray trong máy tính nhanh hơn so với mã nhị phân. Mã Gray có thể được suy từ mã BCD, hoặc mã nhị phân bằng cách mỗi chữ số đứng ở bên phải số 1 ở mã BCD hoặc mã nhị phân khi chuyển sang mã Gray phải đổi thành chữ số ngược với nó. Ví dụ:

BCD	0 0 1 0	hoặc	Nhị phân	1 0 1 1 1
↓	↓ ↓ ↓ ↓		↓	↓ ↓ ↓ ↓ ↓
Gray	0 0 1 1		Gray	1 1 1 0 0

Mã 2 trên 5: mã này sử dụng 5 chữ số hệ 2 để biểu diễn các chữ số hệ 10. Mỗi tổ hợp mã bao giờ cũng có hai chữ số "1" và ba chữ số "0".

Mã Johnson: cũng sử dụng 5 chữ số hệ 2 để biểu diễn các số hệ 10. Đặc điểm là khi chuyển sang số tiếp theo mã sẽ thay chữ số "0" bằng chữ số "1"

bắt đầu từ phải sang trái cho đến khi đạt đến 11111 ứng với số 5 hệ 10 thì lại thay thế dần chữ số "1" bằng chữ số "0" và cũng theo chiều từ phải sang trái.

Ngoài cách mã hoá dùng 1 tổ hợp có 4 hoặc 5 chữ số hệ 2 như nói trên, người ta còn dùng các loại mã có tổ hợp 8 hoặc 10 chữ số. Nhược điểm của mã này là từ mã dài chiếm nhiều thời gian trong kênh thông tin. Nhưng ưu điểm là mã này có thể sử dụng những bit thừa để phát hiện sai trong nhiều trường hợp có thể sửa được.

1.5. MÃ CÁC CHỮ CÁI, CHỮ SỐ

Trong máy tính, ngoài các tin tức số, cần ghi các thông tin không số như các chữ cái, dấu phép tính, các ký tự khác. Có khoảng 26 chữ cái, 10 chữ số 0 ... 9, các loại dấu khác nhau, ký tự đồ hoạ, ký tự điều khiển truyền tin... tổng số khoảng 87 mã. Như vậy cần ít nhất là 7 bit ghi theo mã nhị phân. Thực tế người ta dùng 1 byte = 8 bit để biểu diễn 1 ký tự.

Có 2 loại mã phổ biến nhất hiện nay là : ASCII (American Standart Code for Information Interchange): mã trao đổi thông tin tiêu chuẩn Mỹ : được dùng cho máy tính cá nhân và trong hệ thống truyền tin dùng trong việc dịch các chữ cái trên mặt phím máy chữ, phím đàn... thành ngôn ngữ máy.

Mã thứ hai là EBCDIC (Extended Binary Decimal Interchange Code): mã trao đổi nhị phân thập phân mở rộng. Mã này cũng quan trọng vì nó được phát triển bởi hãng IBM là hãng chế tạo máy tính lớn của Hoa Kỳ, có sản phẩm rất phổ cập trên thế giới. Đây là một mã 8 bit có nhiều biểu tượng và đặc trưng hơn mã ASCII và nó được dùng trong các hệ thống số có kích thước lớn. Trong bảng 1.4 giới thiệu hai loại mã này.

Bảng 1-4: Mã ASCII và mã EBCDIC

Dấu, chữ số	ASCII	EBCDIC	Kí tự	ASCII	EBCDIC
space	010 0000	0100 0000	A	100 0001	1100 0001
!	010 0001	0101 1010	B	100 0010	1100 0010
//	010 0010	0111 1111	C	100 0011	1100 0011
#	010 0011	0111 1011	D	100 0100	1100 0100
\$	010 0100	0101 1011	E	100 0101	1100 0101
%	010 0101	0110 1100	F	100 0110	1100 0110
&	010 0110	0101 0000	G	100 0111	1100 0111

Dấu, chữ số	ASCII	EBCDIC	Kí tự	ASCII	EBCDIC
/	010 0111	0111 1101	H	100 1000	1100 1000
(010 1000	0100 1101	I	100 1001	1100 1001
)	010 1001	0101 1101	J	100 1010	1101 0001
*	010 1010	0101 1100	K	100 1011	1101 0010
+	010 1011	0100 1110	L	100 1100	1101 0011
'	010 1100	0110 1011	M	100 1101	1101 0100
-	010 1101	0110 0000	N	100 1110	1101 0101
.	010 1110	0100 1011	O	100 1111	1101 0110
/	010 1111	0110 0001	P	101 0000	1101 0111
0	011 0000	1111 0000	Q	101 0001	1101 1000
1	011 0001	1111 0001	R	101 0010	1101 1001
2	011 0010	1111 0010	S	101 0011	1110 0010
3	011 0011	1111 0011	P	101 0100	1110 0011
4	011 0100	1111 0100	U	101 0101	1110 0100
5	011 0101	1111 0101	V	101 0110	1110 0101
6	011 0110	1111 0110	W	101 0111	1110 0110
7	011 0111	1111 0111	X	101 1000	1110 0111
8	011 1000	1111 1000	Y	101 1001	1110 1000
9	011 1001	1111 1001	Z	101 1010	1110 1001

1.6. KHÁI NIỆM VỀ SỐ BÙ

Trong hệ nhị phân người ta sử dụng số bù 1 và số bù 2. Ví dụ cho một số $(0011)_2$, ta hãy lấy phân bù 1 của từng chữ số được $(1100)_2$. Đây chính là số bù 1 của số đã cho. Gọi là bù 1 vì tổng 2 chữ số có trọng số tương ứng trong 2 số nói trên luôn là 1. Ta có thể tìm số bù 1 của số nhị phân bằng cách đảo ngược các bit của số nhị phân đó. Còn tìm số bù 2 thì ta lấy số bù 1 cộng thêm 1. Ở đây số bù 2 của $(0011)_2$ là $(1101)_2$. Trong kỹ thuật số, số bù 2 của một số dùng để biểu diễn số âm của số nhị phân đó. Trong ví dụ trên ta có:

$$(0011)_2 = (3)_{10} . \text{ Số bù 2 là :}$$

$$(1101)_2 = (-3)_{10} .$$

Cách biểu diễn số âm trong hệ nhị phân như vậy được dùng trong máy tính để biến các phép trừ thành phép cộng với số âm. Ví dụ muốn tính $A-B$, máy tính sẽ thực hiện $A+(-B)$.

Với các số của hệ 10 biểu diễn bằng mã BCD 2421 người ta cũng sử dụng số bù 9 và số bù 10. Ví dụ :

Số $(134)_{10} = (0001\ 0011\ 0100)_{\text{BCD } 2421}$. Số bù 9 của nó là

$(865)_{10} = (1110\ 1100\ 1011)_{\text{BCD } 2421}$.

Gọi là bù 9 vì ở đây ta cũng có tổng 2 chữ số có cùng trọng số luôn luôn bằng 9:

$1 + 8 = 9; 3 + 6 = 9; 4 + 5 = 9$. Còn số bù 10 ở đây cũng bằng số bù 9 cộng 1.

Vậy số bù 10 của số $(134)_{10}$ là $(1110\ 1100\ 1100)_{\text{BCD } 2421}$. Nhìn số bù 9 của mã BCD 2421 ta thấy chúng là số bù 1 ở dạng nhị phân. Đặc điểm này được sử dụng khi xây dựng các mạch số học.

Trong hệ 16 người ta cũng có số bù 15 và số bù 15 cộng 1 gọi là số bù 16.

Ví dụ cho một số : $(4A)_{16} = (0100\ 1010)_2$.

Số bù 15 của nó là: $(B5)_{16} = (1011\ 0101)_2$. Đây chính là số bù 1 của nhau ở dạng nhị phân. Gọi là bù 15 vì tổng hai chữ số có cùng trọng số luôn bằng 15, ở đây ta có:

$(4 + B)_{16} = 15$ và $(A + 5)_{16} = 15$.

Còn số bù 16 bằng số bù 15 cộng 1, đó là $(B6)_{16} = (1011\ 0110)_2$. Đây chính là số bù 2 của dạng nhị phân. Như vậy số bù 16 trong hệ 16 giúp ta viết số âm ở hệ nhị phân của một số có giá trị lớn một cách dễ dàng.

Ví dụ muốn chuyển số $(-4091)_{10}$ sang hệ nhị phân ta làm như sau :

$(4091)_{10} = (\text{FFB})_{16}$, bù 15 của nó là $(004)_{16}$, bù 16 là $(005)_{16}$. Viết số bù 16 này dưới dạng nhị phân ta sẽ được số nhị phân biểu diễn số

$(-4091)_{10} = (005)_{16} = (0000\ 0000\ 0101)_2$.

1.7. BIỂU DIỄN SỐ ÂM TRONG HỆ NHỊ PHÂN

Trong hệ thập phân ta dùng dấu + và - cho số dương và số âm. Trong hệ nhị phân người ta quy ước : tận cùng bên trái của 1 số nếu ghi số 1 là biểu thị **dấu âm**, nếu ghi số 0 là biểu thị dấu dương. Ví dụ : - 5 ở hệ nhị phân biểu diễn: **1 0101**.

+9 ở hệ nhị phân biểu diễn : 0 1001

Đây là kiểu biểu diễn "Dấu" và "Trị số thật" (số 1 và số 0 đầu tiên biểu diễn dấu âm và dấu dương, 4 bit còn lại biểu diễn trị số thật của số 5 và số 9). Cách biểu diễn số âm như vậy không thuận tiện khi làm các phép trừ. Ví dụ: $12 - 5$. Trong máy tính sẽ thực hiện theo cách $12 + (-5)$. Muốn vậy ta phải biểu diễn số (-5) dưới dạng số âm ở hệ nhị phân, đó là số bù 2 của số 5 ở dạng nhị phân. Vậy ta sẽ làm phép tính trên như sau :

$$(12)_{10} = (1100)_2$$

$(5)_{10} = (0101)_2$. Lấy bù 2 bằng cách đảo các bit rồi +1 ta được :

$(-5)_{10} = (1011)_2$. Thay $12 + (-5)$ ở dạng nhị phân vào ta được :

$$1100$$

$$+1011$$

$$10111$$

Ở kết quả xuất hiện bit thứ 5 là số 1, số 1 đó không có giá trị trong kết quả, nó chỉ chứng tỏ rằng ta vừa thực hiện phép trừ nhị phân.

Kết quả của phép tính chỉ là 4 bit còn lại $(0111)_2 = (7)_{10}$. Điều này là hợp lý vì khi 2 số nhị phân 4 bit trừ đi nhau không thể có kết quả là số nhị phân 5 bit. Với tất cả các phép trừ thực hiện theo cách trên ở kết quả đều xuất hiện thêm số 1 vào trước bit lớn nhất của kết quả.

Ví dụ : $61 - 12$ thực hiện ở dạng nhị phân là :

$$(61)_{10} = (3D)_{16} = (0011\ 1101)_2$$

$(12)_{10} = (0000\ 1100)_2$. Lấy bù 2 để được :

$(-12)_{10} = (1111\ 0100)_2$. Thay vào phép cộng :

$$\begin{array}{r} 61 \rightarrow \quad 0011\ 1101 \\ +(-12) \rightarrow \quad + 1111\ 0100 \\ \hline 49 \quad \quad \quad \mathbf{1\ 0011\ 0001} \end{array}$$

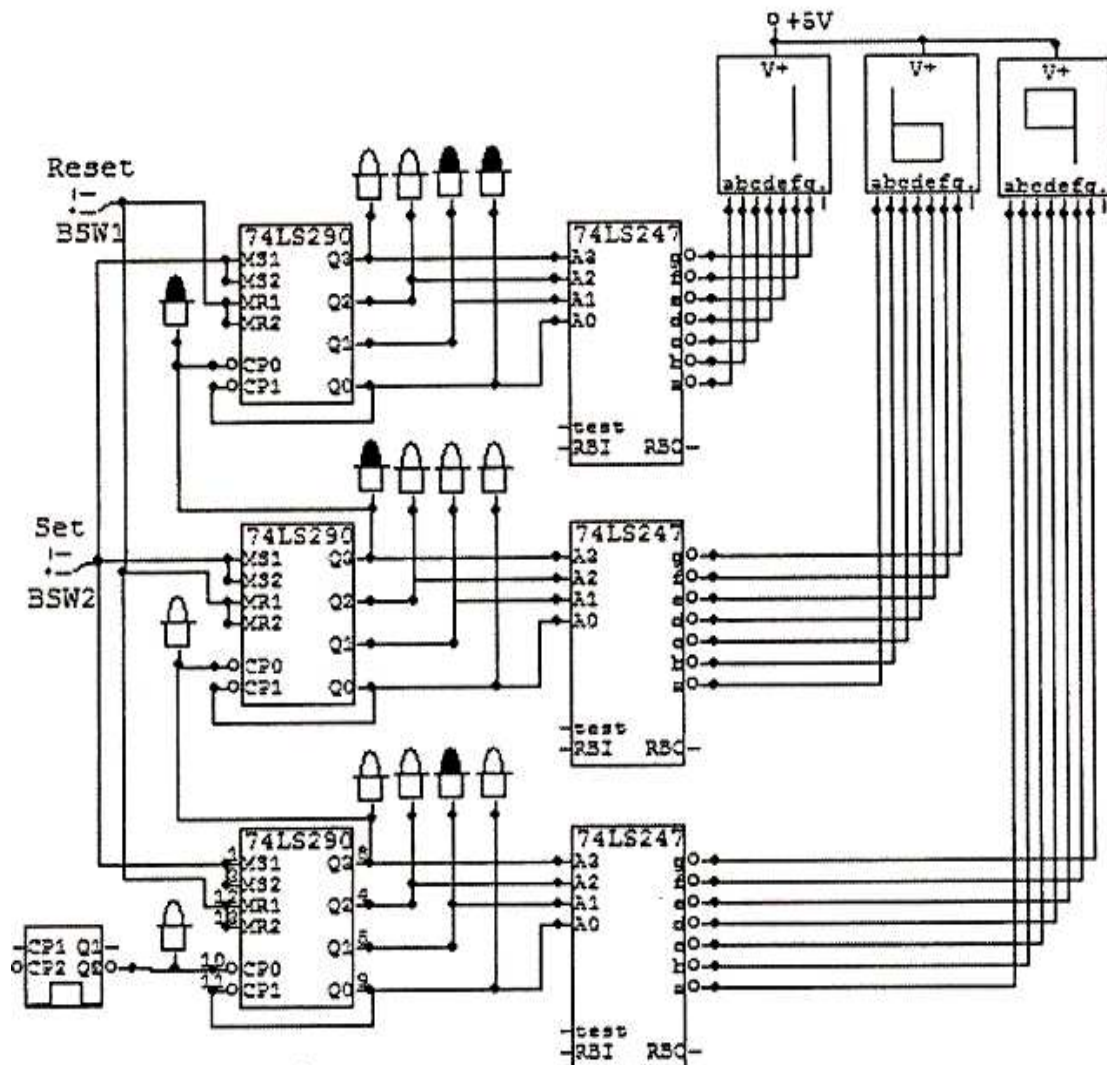
↓ ↓

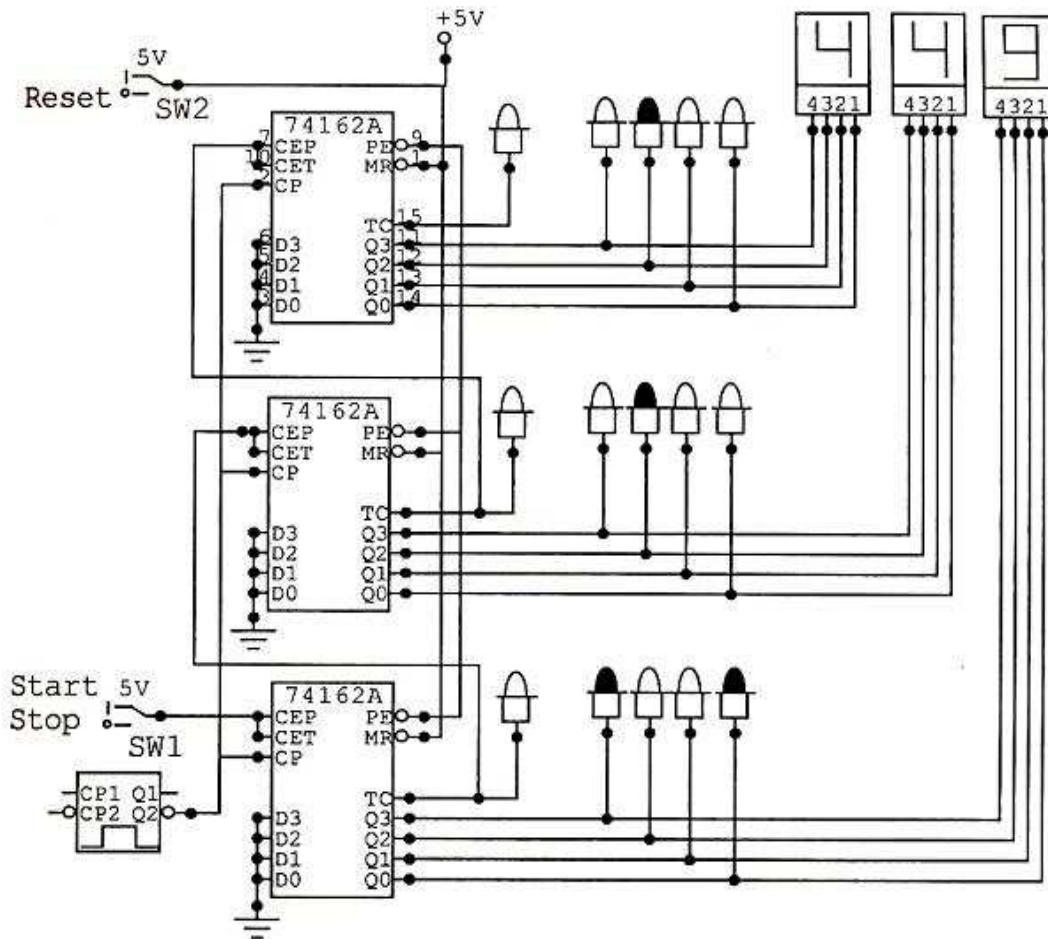
Thể hiện phép trừ Kết quả = $(0011\ 0001)_2 = (49)_{10}$

1.8. BÀI TẬP

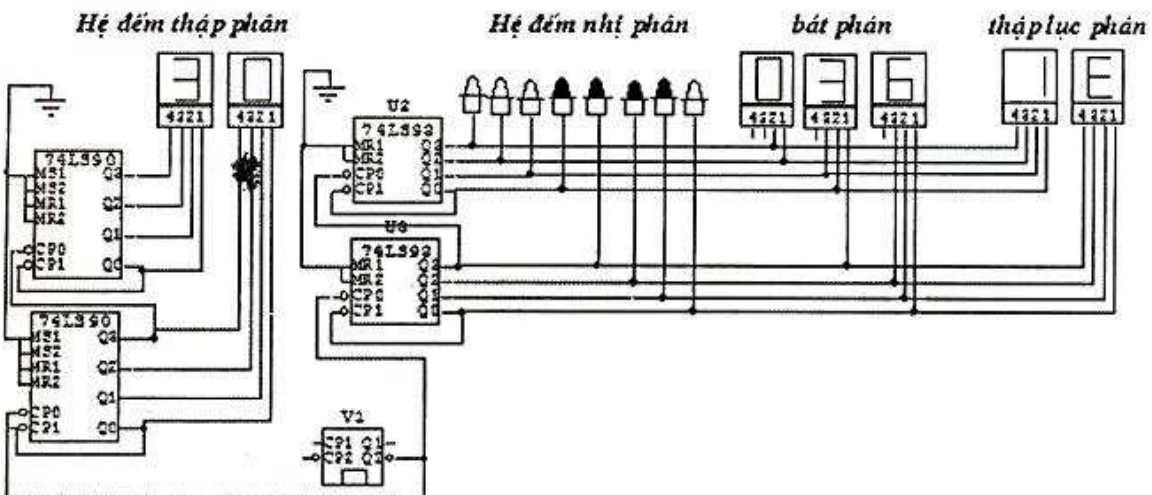
1. Chuyển đổi các con số thập phân sau đây sang hệ đếm nhị phân, bát phân và thập lục phân: 92_{10} , 144_{10} , 409_{10} , 254_{10} , 256_{10} , 64522_{10} , 2000_{10}

2. Chuyển đổi các số nhị phân sau đây thành số bù nhị phân 8 bit: -1_{10} , -5_{10} , -17_{10} , -64_{10} , -127_{10} , -128_{10} , $+25_{10}$.
3. Thực hiện phép tính số học: $61_{10} - 17_{10} = ?$ trong hệ đếm nhị phân.
4. Biểu diễn con số thập phân sau đây bằng mã nhị phân và mã BCD (8421): 16_{10} , 1999_{10} .
5. Thiết kế thí nghiệm mô phỏng các hệ thống đếm sau đây:
 - Hệ đếm 10 mã BCD. (Hình 1.1).
 - Hệ đếm nhị phân, thập lục phân. (Hình 1.2)
 - Các hệ thống đếm 10, nhị phân, bát phân, thập lục phân. (Hình 1.3).





Hình 1.2. Thiết kế mô phỏng hoạt động của bộ đếm nhị phân, thập lục phân.



Hình 1.3. Thí nghiệm mô phỏng hoạt động các hệ thống đếm, minh họa cách chuyển đổi các con số giữa các hệ thống đếm khác nhau, kiểm chứng bảng 1.1

Chương II

ĐẠI SỐ LOGIC

Đại số logic còn được gọi là Đại số Boole. Lí thuyết này do nhà toán học người Anh George Boole đưa ra năm 1847. Đại số logic là công cụ toán học được dùng cho hệ thống đếm nhị phân. Khác với hệ thống đếm thập phân chúng ta quen dùng hàng ngày phải dùng tới mười chữ số để biểu diễn các con số, hệ thống đếm nhị phân chỉ dùng có hai chữ số là 0 và 1 cũng đủ để biểu diễn các con số.

Một đặc điểm của đại số logic là các hàm và biến chỉ nhận một trong hai giá trị **0** hoặc **1**. Hai giá trị này biểu thị hai trạng thái logic khác nhau **đúng** hoặc **sai**, đối với mạch điện tử hai giá trị 1 hoặc 0 được dùng để biểu thị hai mức điện áp: **điện áp cao** (V_H) hoặc **thấp** (V_L), công tắc **đóng** hoặc **ngắt**, **có** hoặc **không** có dòng điện chạy trong mạch.

2.1. BA PHÉP TÍNH CƠ BẢN TRONG ĐẠI SỐ LÔGIC

Phép cộng logic: $y = x_1 + x_2$

Phép nhân logic: $y = x_1 \cdot x_2$

Phép phủ định: $y = \bar{x}$

2.2. CÁC ĐỊNH LUẬT CƠ BẢN CỦA ĐẠI SỐ BOOLE

2.2.1. Các mệnh đề cơ sở

$$x + \bar{x} = 1$$

$$x \cdot \bar{x} = 0$$

$$x + 1 = 1$$

$$x \cdot 1 = x$$

2.2.2. Định luật hấp thụ

$$x + x = x$$

$$x \cdot x = x$$

2.2.3. Định luật phủ định của phủ định

$$\overline{\overline{x}} = x$$

2.2.4. Định luật kết hợp

$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$$

$$(x_1 \cdot x_2) \cdot x_3 = x_1 (x_2 \cdot x_3)$$

2.2.5. Định luật giao hoán

$$x_1 + x_2 = x_2 + x_1$$

$$x_1 \cdot x_2 = x_2 \cdot x_1$$

2.2.6. Định luật phân phối

$$x_1(x_2 + x_3) = x_1x_2 + x_1x_3$$

$$\begin{aligned}(x_1 + x_2)(x_1 + x_3) &= x_1x_1 + x_1x_3 + x_2x_1 + x_2x_3 \\ &= x_1(1+x_2+x_3) + x_2x_3 \\ &= x_1 + x_2x_3\end{aligned}$$

2.2.7. Định lý DE MORGAN

$$\overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}$$

$$\overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}$$

Định lý này giúp ta chuyển phép cộng logic thành phép nhân logic và ngược lại. Vận dụng định lý De Morgan chúng ta có thể giải các bài toán thiết kế mạch logic tổ hợp theo các cửa logic cơ bản cho sẵn. Các phép tính cơ bản và các định lý nêu trên đều có thể mở rộng phạm vi áp dụng cho *hàm nhiều biến*. Các định luật giao hoán, phân phối, kết hợp tương tự như đại số thông thường. Riêng các mệnh đề cơ sở, định lý hấp thụ, định lý phủ định hai lần, định lý De Morgan mang tính đặc thù của đại số Boole. Các định luật nêu trên chúng ta có thể chứng minh bằng cách lập bảng chân lý (bảng sự thật) trên đó ghi các giá trị tổ hợp biến và giá trị tương ứng của hàm.

2.3. PHƯƠNG PHÁP BIỂU DIỄN HÀM LOGIC

2.3.1. Khái niệm về minterm (số hạng tối thiểu) và maxterm (số hạng tối đa)

Một hàm logic có n biến, mỗi biến có thể nhận một trong hai giá trị 0 hoặc 1, như vậy ta sẽ có 2^n tổ hợp biến. Mỗi tổ hợp biến ta có thể tạo thành một số hạng là tích tất cả các biến có trong cùng một tổ hợp biến. Các số hạng này được gọi là *minterm* (số hạng tối thiểu). Gọi là số hạng tối thiểu vì minterm là tích các biến có trong một tổ hợp biến, tích này chỉ bằng một chỉ khi tất cả các biến đều bằng 1. Như vậy, ứng với mỗi một minterm ta chỉ tìm được một tổ hợp giá trị biến tương ứng với nó để nó bằng 1 và chỉ có một tổ hợp biến mà thôi.

Mỗi tổ hợp biến ta cũng có thể tạo thành một số hạng là tổng tất cả các biến có trong cùng một tổ hợp biến các số hạng này được gọi là *maxterm* (số hạng tối đa). Maxterm là tổng tất cả các biến có trong tổ hợp biến nên chỉ cần một trong các biến bằng 1 thì maxterm bằng 1, maxterm bằng 0 chỉ trong một trường hợp duy nhất ứng với tất cả các biến trong tổ hợp biến đều bằng 0. Như vậy các trường hợp maxterm bằng 1 là tối đa, trường hợp minterm bằng 1 là tối thiểu. Một hàm có n biến ta có 2^n minterm và 2^n maxterm.

Bảng 2.1: Các minterm và maxterm của hàm logic 3 biến

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

Ví dụ một hàm $F(A, B, C)$ có 3 biến là A, B, C ta có 8 tổ hợp biến được sắp xếp một cách trình tự theo số nhị phân là: 000, 001, 010, 011, 100, 101,

110, 111. Tương ứng với 8 tổ hợp biến này ta có 8 số hạng tối thiểu minterm, kí hiệu là $m_0, m_1, m_2, \dots, m_7$, và 8 số hạng tối đa maxterm $M_0, M_1, M_2, \dots, M_7$. Trên bảng 2.1 trình bày các minterm và maxterm của hàm này.

Trong một minterm và maxterm có mặt tất cả các biến số có trong tổ hợp biến của hàm, các biến số này chỉ xuất hiện một lần dưới dạng đảo hoặc không đảo. Hàm logic có thể được biểu diễn dưới dạng là tổng các minterm hoặc tích các maxterm.

2.3.2. Các tính chất của maxterm và minterm

$$\overline{M_0} = m_0$$

$$\overline{M_1} = m_1$$

$$\overline{M_2} = m_2$$

$$\overline{M_3} = m_3$$

$$\overline{M_4} = m_4$$

$$\overline{M_5} = m_5$$

$$\overline{M_6} = m_6$$

$$\overline{M_7} = m_7$$

Hai maxterm và minterm của số hạng có cùng chỉ số (ví dụ M_0 và m_0 , M_1 và m_1 , M_2 và m_2 ...) là phủ định của nhau.

Tổng logic của tất cả các minterm = 1.

Tích logic của tất cả các maxterm = 0.

Tích hai minterm khác nhau bất kỳ = 0.

Tổng hai maxterm khác nhau bất kỳ = 1.

2.3.3. Phương pháp biểu diễn hàm logic

Có bốn phương pháp được dùng để biểu diễn hàm logic đó là: bảng hân lý, bảng Karnaugh, phương trình logic, ký hiệu logic. Trong kỹ thuật điện tử số người ta còn dùng các kí hiệu logic và sơ đồ logic để biểu diễn các hàm logic và phương trình logic. Trong các tiết sau chúng ta sẽ làm quen các kí hiệu logic này và cách xây dựng các sơ đồ logic tương ứng với từng phương trình logic. Chúng ta cần nắm vững từng phương pháp biểu diễn hàm, biết vận dụng những ưu việt của từng phương pháp, chuyển đổi từ phương pháp này sang phương pháp kia.

2.3.3.1. Bảng chân lý

Bảng chân lý là bảng miêu tả quan hệ giữa các giá trị của hàm số tương ứng với mọi giá trị có thể của biến số. Trong đó có các cột ghi các giá trị của các biến đầu vào và cột ghi các giá trị của hàm đầu ra tương ứng với từng tổ hợp biến. Bảng chân lý biểu thị hàm logic dưới dạng bảng số một cách rõ ràng tường minh. Sau khi xác định giá trị các tổ hợp biến đầu vào ta có thể dùng bảng chân lý để xác định giá trị tương ứng của hàm đầu ra.

Trong các sổ tay tra cứu các vi mạch số đều có bảng chân lý kèm theo kí hiệu logic hoặc sơ đồ logic để giới thiệu chức năng của vi mạch. Người ta thường dùng các chữ A, B, C,... hoặc X_1, X_2, X_3, \dots để kí hiệu các biến logic, và dùng các chữ F, Y, Z, W để kí hiệu các hàm logic. Trong trường hợp dùng các chữ để kí hiệu hàm và biến logic chúng ta cần phân biệt rõ đâu là các biến tương ứng với các lối vào của mạch logic và đâu là hàm tương ứng với lối ra của mạch logic. Mỗi biến đầu vào có thể nhận hai giá trị 0 và 1, nếu hàm có n biến đầu vào thì sẽ có 2^n tổ hợp các giá trị khác nhau của chúng.

Để nhận được bảng chân lý chúng ta cần phải liệt kê tất cả các giá trị của hàm ở đầu ra tương ứng với các tổ hợp biến lối vào. Để khỏi bỏ sót hoặc trùng lặp ta nên sắp xếp các tổ hợp biến lối vào tuần tự theo số đếm nhị phân.

Ví dụ ta có hàm logic: $F = \bar{A}B + A\bar{B}$.

Bảng chân lý của hàm này được trình bày trên bảng 2.2.

Bảng 2.2: Bảng chân lý của hàm $F = \bar{A}B + A\bar{B}$

Thứ tự sắp xếp các tổ hợp biến i	Lối vào		Lối ra
	A	B	F_i
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Dùng bảng chân lý để biểu diễn hàm tuy có ưu điểm là rõ ràng trực quan, nhưng có nhược điểm là cách biểu diễn này sẽ trở nên rối rắm khi hàm có nhiều biến.

2.3.3.2. Phương trình logic

Biểu diễn hàm logic bằng các phương trình logic cho thấy rõ mối quan hệ giữa hàm và biến thông qua các phép toán cộng logic, nhân logic và phủ định là phương pháp biểu diễn thích hợp trong mọi trường hợp, kể cả các quan hệ logic phức tạp, hàm có nhiều biến. Dùng phương trình logic biểu diễn hàm sẽ đơn giản, gọn ghẽ hơn là dùng bảng chân lý và rất tiện để thực hiện các phép toán logic và tối thiểu hoá các hàm bằng phương pháp đại số.

Phương trình logic có thể được xác lập theo các cách sau:

Cách 1: Lấy tổng của các tích tức là lấy tổng các minterm:

$$F = \sum f_i m_i$$

Như vậy, ta chỉ lấy tổng các minterm nào có giá trị tương ứng của hàm $f_i = 1$. Đây là phương pháp hay dùng trong giáo trình Kỹ thuật số này.

Cách 2: Lấy tích của các tổng tức là lấy tích của các maxterm:

Như vậy, chỉ lấy tích các maxterm tương ứng có $f_i = 0$.

Ví dụ : Một hàm 3 biến có bảng chân lý được trình bày trên bảng 2.3

Bảng 2.3: Bảng chân lý của 1 hàm 3 biến

I	A	B	C	f_i	m	M
0	0	0	0	1	m_0	M_7
1	0	0	1	0	m_1	M_6
2	0	1	0	0	m_2	M_5
3	0	1	1	1	m_3	M_4
4	1	0	0	1	m_4	M_3
5	1	0	1	0	m_5	M_2
6	1	1	0	0	m_6	M_1
7	1	1	1	0	m_7	M_0

Ta có thể xác định hàm logic theo hai cách nói trên:

Cách 1: Lấy tổng chuẩn các minterm ứng với $f_i = 1$ ta được:

$$F = m_0 + m_3 + m_4$$

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

Cách 2: Lấy tích chuẩn các maxterm ứng với $f_i = 0$ đó là:

$$F = M_6 \cdot M_5 \cdot M_2 \cdot M_1 \cdot M_0$$

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

Hàm logic F xác định theo 2 cách trên là như nhau.

2.3.3.3. Bảng Karnaugh

Khi một hàm logic có số lượng biến tương đối nhỏ ($k \leq 6$) người ta thường biểu diễn chúng dưới dạng một bảng gọi là bảng Karnaugh. Theo phương pháp này, một hàm n biến được biểu diễn trên một bảng gồm 2^n ô vuông. Mỗi ô vuông tương ứng với 1 minterm của hàm cần biểu diễn. Lưu ý rằng các tổ hợp biến ở đây được xếp theo thứ tự của mã Gray, hai ô liền kề các minterm chỉ khác nhau có một bit.

Trên bảng 2.4 là bảng Karnaugh của một số hàm logic có 2, 3, 4 biến, ở dưới mỗi bảng là phương trình logic tương ứng của các hàm này.

Bảng 2.4: Bảng Karnaugh của một số hàm có 2, 3, 4 biến

a/

	B	0	1
A			
0			1
1		1	

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

b/

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

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

c/

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

$$F = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$$

d/

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

$$F = \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + A\bar{B}C\bar{D}$$

Trong các ô của bảng Karnaugh, nếu ứng với tổ hợp biến nào mà hàm có giá trị là 1 thì ô vuông đó được ghi giá trị 1. Còn các tổ hợp biến hàm có giá trị 0 thì bỏ trống (hoặc ghi 0). Bảng Karnaugh 2.4a được xác lập từ bảng

chân lý 2.2. Bảng Karnaugh có thể được thiết lập từ bảng chân lý hoặc phương trình logic, ngược lại từ bảng Karnaugh ta cũng có thể tìm được phương trình logic đơn giản, rút gọn hàm logic. Dựa vào bảng Karnaugh ta xác định hàm F bằng tổng các minterm ứng với hàm $f_i = 1$ (ô vuông chứa 1).

2.3.4. Phương pháp tối giản hàm logic

Trong việc thiết kế các khối chức năng logic, tìm ra được một sơ đồ logic đơn giản đáp ứng đầy đủ các yêu cầu của khối chức năng cần thiết kế, thì yêu cầu hàng đầu của công tác thiết kế các mạch điện tử là tính kinh tế và mạch phải có tính ổn định, độ tin cậy cao. Để đảm bảo các yêu cầu này thì sơ đồ logic phải bao gồm số các phần tử logic cơ bản ít nhất, các sơ đồ càng đơn giản càng có độ tin cậy và ổn định cao. Để xây dựng được một sơ đồ như vậy chúng ta phải tìm ra được một phương trình logic tối giản mô tả đúng chức năng logic của mạch điện tử cần thiết kế. Các hàm logic mà ta thường gặp thường không phải là dạng tối giản, nếu ta xây dựng mạch dựa trên phương trình này thì sẽ tốn kém vì phải dùng nhiều phần tử linh kiện logic, sơ đồ càng phức tạp, độ ổn định, độ tin cậy càng kém, xác suất hư hỏng càng tăng. Vì thế, trước khi xây dựng mạch bao giờ cũng phải tìm cách rút gọn hàm, đưa phương trình biểu diễn hàm về dạng tối giản. Có hai phương pháp rút gọn hàm logic: Rút gọn theo phương pháp đại số thông thường và rút gọn theo phương pháp hình học dùng bảng Karnaugh. Ta sẽ đi qua những nét cơ bản của các phương pháp này

a/ Rút gọn hàm logic theo phương pháp đại số

Từ các phương trình logic biểu diễn hàm logic dưới dạng tổng các minterm, ta áp dụng các định luật của đại số logic để đơn giản hàm logic sao cho hàm cuối cùng là tối giản, thực hiện hàm cần ít phần tử logic cơ bản nhất. Vì trong thực tế các biểu thức logic rất đa dạng, từ một hàm logic cũng có thể biểu diễn bằng nhiều cách khác nhau nên khó có thể tìm ra một quy trình tối ưu để tìm ra được biểu thức logic tối giản một cách nhanh nhất. Con đường dẫn đến kết quả nhanh nhất, lời giải đẹp nhất tùy thuộc vào sự nắm chắc các định luật của đại số Boole và kinh nghiệm của mỗi người rút ra được trong quá trình chịu khó làm các bài tập.

Nhìn chung phương pháp tối giản hàm logic theo phương pháp đại số cũng tương tự như phương pháp rút gọn các biểu thức đại số thông thường, cũng khai triển biểu thức, nhóm thừa số chung, đơn giản rút gọn hàm số như

ta đã quen làm trong chương trình học phổ thông trước đây nhưng nếu biết khai thác các mệnh đề cơ sở, các định luật đặc thù của đại số logic như định luật phủ định hai lần, định luật De Morgan, nhiều khi chúng ta được các biểu thức logic đơn giản hết sức bất ngờ thú vị. Chúng ta hãy thử làm một bài tập ví dụ sau đây:

Đơn giản biểu thức logic:

$$F = \bar{A} \bar{B} C D + \bar{A} \bar{B} C \bar{D} + A \bar{B} \bar{C} \bar{D} + A \bar{B} \bar{C} D + A \bar{B} \bar{C} \bar{D} + A \bar{B} C \bar{D}$$

Trong biểu thức trên có số hạng $A \bar{B} \bar{C} \bar{D}$. Cộng thêm vào vế phải của biểu thức số hạng này rồi nhóm thừa số chung ta có:

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

$$F = \bar{A} \bar{B} C + A \bar{C} \bar{D} + A \bar{B} \bar{D}$$

Trong cách làm này ta đã vận dụng định lý hấp thụ $x + x = x$ và mệnh đề cơ sở của đại số Boole: $x + \bar{x} = 1$

b/ Rút gọn hàm logic theo phương pháp dùng bảng Karnaugh:

Ta hãy ghép các minterm ứng với $f_i = 1$ (các ô có số 1) ở các ô kề nhau theo hàng ngang hoặc hàng dọc và ghép các ô bằng 1 nằm đối diện nhau trong bảng như các đường khoanh vòng các số 1 với nhau trong bảng 2.4. Các minterm được ghép như vậy nhất định sẽ có thừa số chung và sẽ đơn giản được 1 biến bù nhau.

Ví dụ như trường hợp hàm logic tương ứng với bảng 2.4b, theo phương pháp này ta có thể đơn giản hàm logic như sau:

$$F = \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + A \bar{B} \bar{C} + ABC \text{ ghép 2 minterm kề nhau:}$$

$$F = \bar{A} \bar{C} (\bar{B} + B) + A \bar{B} \bar{C} + ABC \text{ vì } \bar{B} + B = 1 \text{ nên ta có:}$$

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

Để đơn giản 2 ô đối diện, ở đây ta viết thêm số hạng $\bar{A} \bar{B} \bar{C}$ vào hàm F vẫn không thay đổi (vì $\bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} \bar{C} = \bar{A} \bar{B} \bar{C}$). Ta có

$$F = \bar{A} \bar{C} + ABC + A \bar{B} \bar{C} + \bar{A} \bar{B} \bar{C} \text{ ghép 2 số hạng cuối:}$$

$$F = \bar{A} \bar{C} + ABC + \bar{B} \bar{C} (A + \bar{A}) \text{ cuối cùng:}$$

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

Làm tương tự với bảng 2.4c ta được:

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

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

Với bảng 2.4d : Dựa vào bảng Karnaugh ta có thể rút gọn hàm F bằng cách ghép các ô liền kề có giá trị bằng 1, nhóm thừa số chung để rút gọn:

$$F = \bar{A} \bar{B} CD + \bar{A} \bar{B} C \bar{D} + AB$$

$$\bar{C} \bar{D} + A \bar{B} \bar{C} \bar{D} + A \bar{B} \bar{C} D + A \bar{B} C \bar{D}$$

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

$$F = \bar{A} \bar{B} C + A \bar{C} \bar{D} + A \bar{B} \bar{D}$$

Các phương pháp tối giản hàm logic nói trên là rất quan trọng, giúp ta thiết kế các sơ đồ logic tổ hợp một cách tối ưu, tiết kiệm tối đa số cửa logic cơ bản cần dùng.

Qua một số ví dụ cụ thể rút gọn hàm Boole bằng bảng Karnaugh nêu trên, ta có thể tóm tắt phương pháp rút gọn hàm Boole bằng bảng Karnaugh cần tiến hành theo các bước sau:

Đưa hàm Boole về dạng tổng các minterm có giá trị bằng 1.

Xác lập chính xác bảng Karnaugh các minterm được sắp xếp theo trình tự của mã Gray hai ô liền kề theo hàng và cột chỉ khác nhau có một biến, sự khác nhau đó là ở hai giá trị nghịch đảo nhau của biến này. Chỉ điền 1 vào các ô ứng với minterm có mặt trong phương trình, các ô ứng với các minterm không có mặt trong phương trình thì để trống.

Vẽ các đường hình chữ nhật hoặc hình vuông bao quanh các ô liền kề có giá trị bằng 1. Khi vẽ cần tuân theo quy tắc sau:

Các đường bao quanh phải là hình chữ nhật hay hình vuông và trong đó chỉ chứa ô có giá trị bằng 1.

Số ô chứa trong đường bao phải là 2^n (với $n=1,2,3,\dots$) tức là bằng 2,4,8,...

Các hàng trên cùng và các hàng cuối cùng cũng được xem là các hàng kế cận. Các cột tận cùng bên trái và các cột tận cùng bên phải cũng được xem là các cột kế cận.

Mỗi đường bao khép kín phải chứa tối đa số ô bằng 1. Điều này cũng có nghĩa là tổng các đường bao quanh này càng nhỏ càng tốt. Số hạng tối giản bằng số đường bao khép kín các ô. Số đường này càng ít hàm Boole càng được rút gọn.

Sau khi đã thực hiện phép ghép dán các ô kế cận để tìm được các phần tử tối giản, dạng rút gọn của hàm Boole chính là tổng các phần tử tối giản ở trong đường bao và các phần tử ở ngoài các đường bao không thể ghép được với một ô nào khác nữa.

BÀI TẬP

1. Rút gọn hàm bằng phương pháp đại số và bằng phương pháp bảng Karnaugh các hàm ứng với bảng Karnaugh 2.4.b
2. Rút gọn hàm bằng phương pháp đại số và bằng phương pháp bảng Karnaugh các hàm ứng với bảng Karnaugh 2.4.d.
3. Viết phương trình logic và rút gọn hàm bằng phương pháp đại số và phương pháp dùng bảng Karnaugh các hàm tuân theo bảng chân lý sau :

A	B	C_1	S	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Trong đó A, B, C_1 là các biến lối vào. S, C_0 là các hàm ở lối ra.

2.4. CÁC HÀM LOGIC CƠ BẢN

2.4.1. Hàm Hoặc (OR) - Phép cộng logic

Hàm logic: $y = x_1 + x_2$

Bảng chân lý của hàm OR được trình bày trong bảng 2.5. Ký hiệu logic: cho trên Hình 2.1:

a) Theo tiêu chuẩn quân lực Mỹ;

- b) Theo Ủy ban Kỹ thuật quốc tế;
- c) Hình ảnh mạch điện minh họa.

Bảng 2.5: Bảng chân lý của hàm OR

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

Mở rộng cho trường hợp tổng quát có n biến:

$$y = x_1 + x_2 + \dots + x_n$$

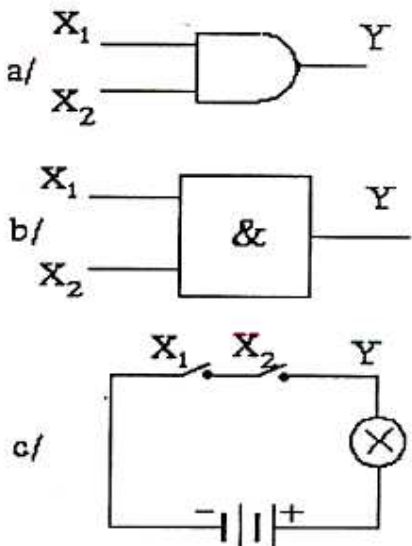
Ta có mạch hoặc n lối vào kí hiệu như Hình 2.2

2.4.2. Hàm Và (AND) - Phép nhân logic

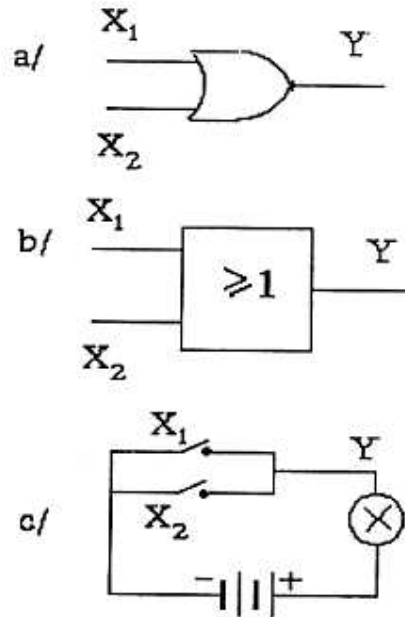
Hàm logic : $y = x_1 \cdot x_2$

Bảng chân lý cho trong bảng 2.6.

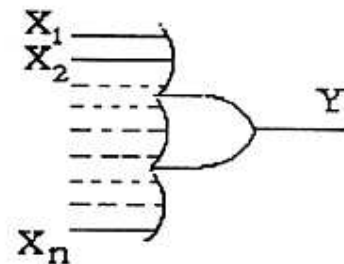
Kí hiệu và hình ảnh minh họa cho trên hình 2.3.



Hình 2.3: Các kí hiệu mạch AND



Hình 2.1: Các kí hiệu mạch OR



Hình 2.2: Mạch OR nhiều lối vào

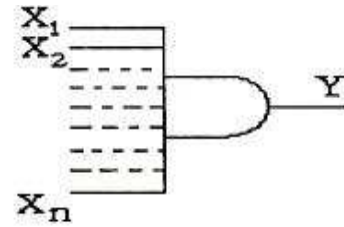
Bảng 2.6: Bảng chân lý của hàm AND

x_1	x_2	Y
0	0	0
1	0	0
0	1	0
1	1	1

Trường hợp tổng quát có n biến :

$$y = x_1 \cdot x_2 \cdot \dots \cdot x_n$$

Ta có mạch và n lối vào kí hiệu như hình 2.4.



Hình 2.4: Kí hiệu mạch AND có n lối vào

2.4.3. Hàm Đảo (Inverter) - Phép Phủ định (NOT)

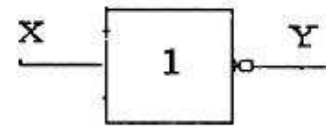
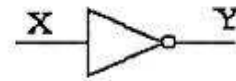
Hàm logic : $y = \bar{x}$

Trên hình 2.5 trình bày kí hiệu logic của mạch đảo

Bảng chân lý cho trong bảng 2.7.

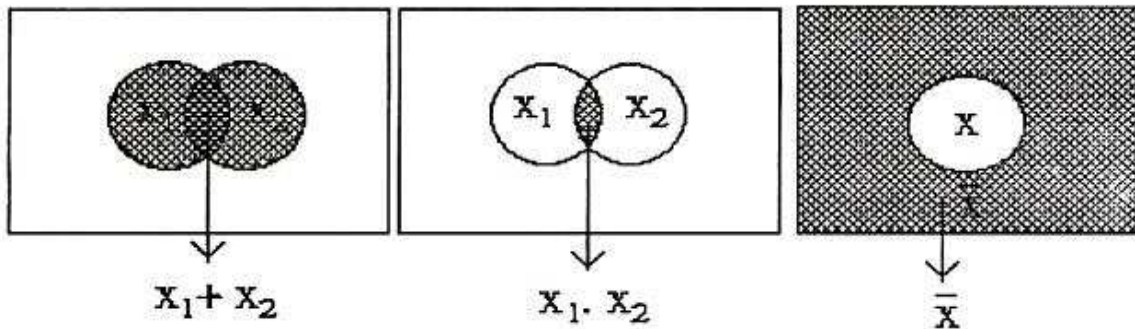
Bảng 2.7: Bảng chân lý hàm Đảo NOT

x	$y = \bar{x}$
0	1
1	0



Hình 2.5: ký hiệu logic mạch Đảo

Để biểu diễn trực quan các phép toán trong đại số logic người ta dùng giản đồ Venn trình bày trong hình 2.6.



Hình 2.6: Giản đồ Venn

2.4.4. Hàm Không Hoặc (NOR)

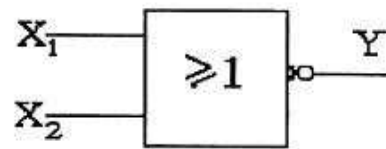
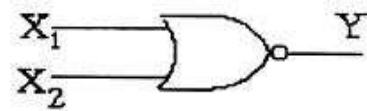
Hàm logic : $y = \overline{x_1 + x_2}$

Kí hiệu logic như hình 2.7.

Bảng chân lý cho trong bảng 2.8.

Bảng 2.8 : Bảng chân lý của hàm Không Hoặc

x_1	x_2	$y = \overline{x_1 + x_2}$
0	0	1
1	0	0
0	1	0
1	1	0



Hình 2.7: Kí hiệu logic của hàm NOR

Trong trường hợp tổng quát nếu có n biến ta cũng có:

$$y = \overline{x_1 + x_2 + \dots + x_n}$$

2.4.5. Hàm Không Và (NAND)

Hàm logic : $y = \overline{x_1 \cdot x_2}$

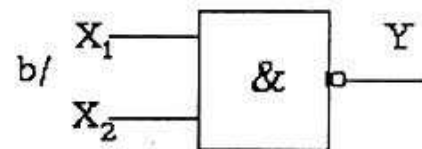
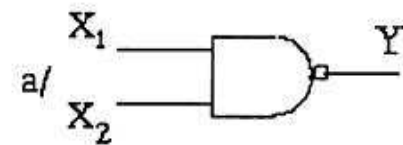
Kí hiệu như hình 2.8.

Bảng 2.9: Bảng chân lý hàm Không Và

x_1	x_2	$y = \overline{x_1 \cdot x_2}$
0	0	1
1	0	1
0	1	1
1	1	0

Tổng quát nếu có n biến ta cũng có:

$$y = \overline{x_1 \cdot x_2 \cdot \dots \cdot x_n}$$



Hình 2.8: Kí hiệu mạch NAND

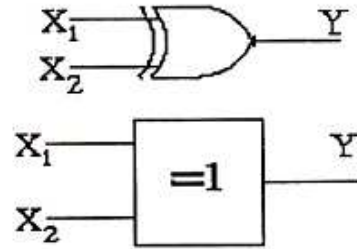
2.4.6. Hàm Hoặc Tuyệt Đối còn gọi là hàm Hoặc Loại Trừ XOR (Exclusive OR)

Kí hiệu hàm XOR được trình bày trên hình 2.9.

Bảng chân lý: bảng 2.10.

Bảng 2.10: Bảng chân lý hàm Hoặc tuyệt đối

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



Hình 2.9: Kí hiệu XOR

Hàm logic : $y = x_1 \bar{x}_2 + \bar{x}_1 x_2$

Được viết là $y = x_1 \oplus x_2$

Sơ đồ logic của mạch tương ứng với phương trình trên được trình bày trên hình 2.10.

Có nhiều cách xây dựng mạch XOR :

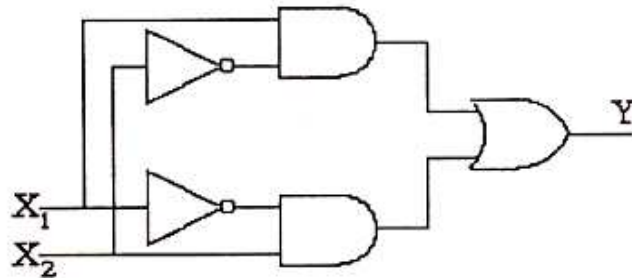
Ta có thể thiết kế sơ đồ mạch XOR bằng NOT và NOR.

Muốn vậy ta áp dụng định luật phủ định của phủ định, phủ định hai lần hai vế của phương trình trên rồi dùng định lí De Morgan ta có:

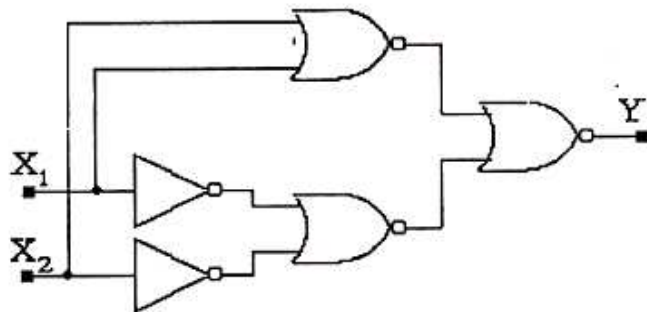
$$y = \overline{x_1 \bar{x}_2 + x_1 x_2} = \overline{x_1 \bar{x}_2} \cdot \overline{x_1 x_2} = (\overline{x_1} + x_2) \cdot (x_1 + \bar{x}_2) =$$

$$\overline{x_1 \bar{x}_2 + x_1 x_2} = \overline{x_1 \bar{x}_2} \cdot \overline{x_1 x_2} = \overline{x_1} + x_2 + x_1 + \bar{x}_2$$

Sơ đồ logic của mạch tương ứng với phương trình này được trình bày trên hình 2.11.



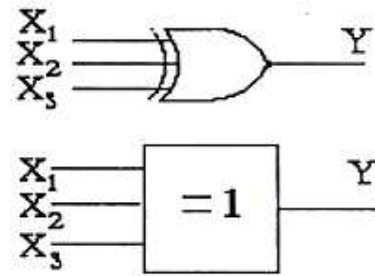
Hình 2.10: Sơ đồ logic của mạch XOR



Hình 2.11: Sơ đồ mạch XOR

Bảng 2.11: Bảng chân lý mạch XOR 3 lối vào

x_1	x_2	x_3	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



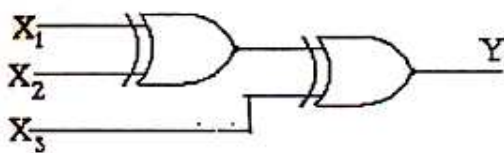
Hình 2.12: Ký hiệu logic mạch XOR 3 lối vào

+ Nhận xét bảng chân lý :

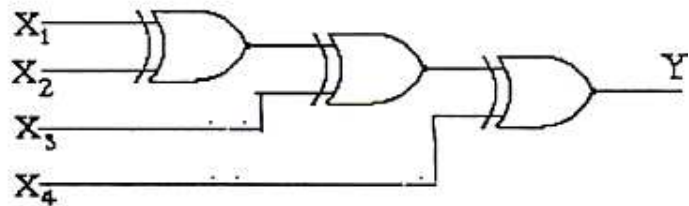
- Nếu số mức logic 1 ở lối vào là lẻ \rightarrow lối ra là mức logic 1
- Nếu số mức logic ở lối vào là chẵn \rightarrow lối ra là mức logic 0

+ Nguyên tắc này áp dụng cho các mạch XOR nhiều lối vào (n lối vào).

+ Ta có thể xây dựng cửa XOR 3 lối vào hoặc 4 lối vào từ các cửa XOR 2 lối vào như hình 2.13, hình 2.14.



Hình 2.13: Mạch XOR 3 lối vào



Hình 2.14: Mạch XOR 4 lối vào

2.4.7. Hàm Không Hoặc Tuyệt Đối (XNOR)

Cửa XNOR được tạo thành khi ta mắc nối tiếp cửa XOR với cửa NOT.

Ví dụ cửa XNOR 2 lối vào :

$$\text{Hàm logic : } y = \overline{x_1 \oplus x_2}$$

Kí hiệu logic của cửa XNOR được trình bày trên hình 2.15.

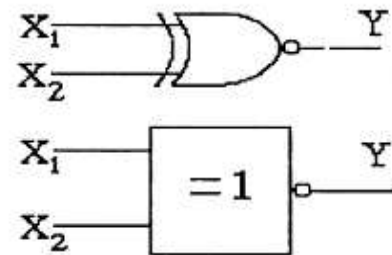
Bảng 2.12 là bảng chân lý của cửa XNOR 2 lối vào.

Từ bảng chân lý này ta có nhận xét: Lối ra của XNOR là đảo của XOR. Kết luận này đúng cho các cửa XNOR n lối vào.

Ta cũng có thể xây dựng được các cửa XNOR nhiều lối vào bằng cách tương tự như xây dựng XOR nhiều lối vào.

Bảng 2.12: Bảng chân lý của mạch XNOR

X_1	X_2	Y
0	0	1
0	1	0
1	0	0
1	1	1

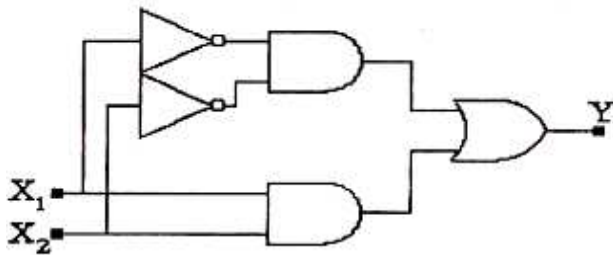


Hình 2.15: Kí hiệu mạch XNOR

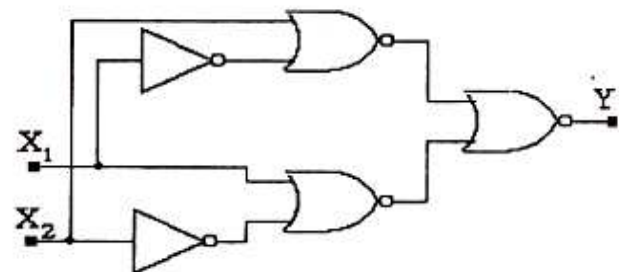
Từ các phần tử logic cơ bản AND, OR, NOT hoặc NAND và NOT hay NOR và NOT ta có thể tạo được các cửa XNOR.

Trên hình 2.16a giới thiệu sơ đồ logic mạch XNOR hai lối vào được xây dựng từ các phần tử logic cơ bản NOT, AND, OR.

Trên hình 2.16b là sơ đồ logic mạch XNOR hai lối vào được xây dựng từ các phần tử logic cơ bản NOT, NOR.

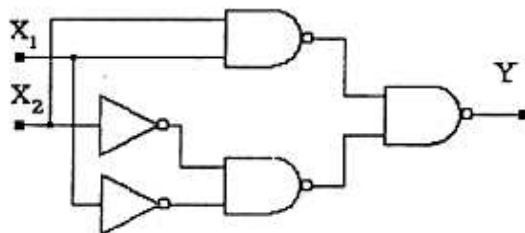


Hình 2.16a: Sơ đồ logic mạch XNOR dùng AND, OR và NOT

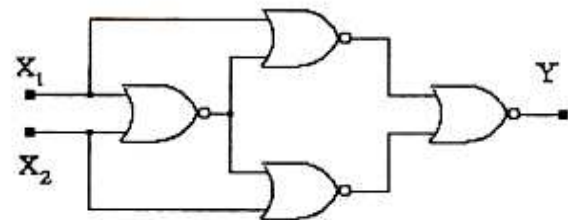


Hình 2.16b: Sơ đồ logic mạch XNOR dùng NOR và NOT

Ta cũng có thể thiết kế mạch XNOR bằng các cửa NOT và NAND, hoặc chỉ bằng cửa NOR. Trên hình 2.17a vẽ sơ đồ logic của mạch XNOR được tạo nên từ cửa NOT và NAND. Sơ đồ logic mạch XNOR chỉ dùng các cửa NOR được trình bày trên hình 2.17b.



Hình 2.17a



Hình 2.17b

Chương III

CÁC HỌ VI MẠCH LOGIC CƠ BẢN

3.1. ĐẶC ĐIỂM CHUNG CỦA CÁC VI MẠCH LOGIC

Khác với các vi mạch tương tự, các mạch logic có các đặc điểm sau:

- Lối vào và lối ra của các vi mạch logic chỉ có hai mức điện áp V_L và V_H tương ứng với mức logic 0 và 1 (có thể viết tắt là mức L và H).

- Các mạch logic phải được nuôi bằng nguồn nuôi có một điện áp chuẩn đã được quy định.

- Cùng một chức năng logic nhưng kỹ thuật điện tử có thể thực hiện theo những sơ đồ nguyên lý khác nhau.

- Những vi mạch được xây dựng trên cùng một kiểu sơ đồ nguyên lý được xếp vào một họ logic. Các vi mạch logic trong cùng một họ logic phải được nuôi bằng nguồn điện có điện áp bằng điện áp nuôi chuẩn cho họ logic đó. Các mức logic của các vi mạch này phải như nhau. Các vi mạch logic có mức logic phù hợp có thể ghép nối trực tiếp với nhau.

Các thông số cơ bản của vi mạch logic:

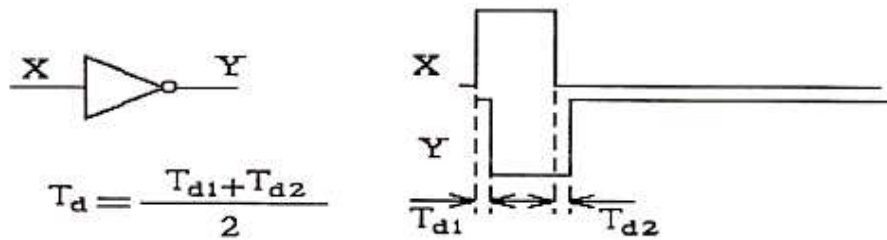
1. *Trở kháng ra (output impedance)*: thay đổi theo trạng thái đầu ra cao hay thấp. Nếu mạch ra dùng một transistor với trở tải ở collector thường có trở kháng ra $Z_{out} \approx 2000\Omega$ cao hơn mạch dùng 2 transistor ($Z_{out} \approx 70\Omega$).

2. *Hệ số mắc tải (Fan out)*: cho biết lối ra có thể điều khiển đồng thời được bao nhiêu lối vào song song của các mạch khác.

3. *Hệ số hợp lối vào (Fan in)*: cho biết có thể mắc song song bao nhiêu lối vào vẫn đảm bảo hợp thông số.

4. *Thời gian trễ (Propagation delay per gate)*: Thời gian trễ trên một cửa T_D là thời gian từ lúc lối vào nhận được tín hiệu đến lúc lối ra bắt đầu thay đổi trạng thái. Thời gian trễ T_d được xác định như ví dụ hình 3.0.

T_d càng nhỏ tốc độ làm việc càng cao.



Hình 3.0: Cửa đảo và thời gian trễ qua cửa đảo.

5. *Nguồn nuôi (Power Supply)*: Nguồn nuôi cho các mạch logic phải là nguồn ổn áp có điện áp ra đúng với điện áp nuôi quy định riêng cho từng họ mạch logic. Khi lối vào, lối ra thay đổi trạng thái làm cho cường độ dòng điện trong toàn mạch thay đổi đột ngột, sự thay đổi này có thể làm rối loạn hoạt động của các mạch khác. Để khắc phục hiện tượng này, giữa chân nguồn và đất của các vi mạch người ta thường mắc thêm tụ lọc để loại bỏ nhiễu, các tụ này phải dùng tụ gốm có điện dung cỡ chừng 0,1 đến 1 μ F.

6. *Công suất tiêu thụ đối với một cửa logic (Power dissipated per gate)*: Công suất càng lớn khi mạch có nhiều điện trở có giá trị nhỏ và transistor làm việc ở chế độ bão hoà. Trong cùng một họ logic các sê-ri khác nhau công suất tiêu thụ trên một cửa cũng khác nhau. Ví dụ họ logic TTL sê-ri 74 có công suất tiêu thụ trên một cửa là 10mW, 74L công suất tiêu thụ trên một cửa là 1mW; 74H công suất tiêu thụ trên một cửa là 22mW

Họ PMOS, NMOS và CMOS tiêu thụ công suất rất nhỏ so với các họ logic khác.

7. *Mức độ chống tạp âm (Noise immunity level)*: là biên độ tạp âm lớn nhất có thể vào mạch mà không làm thay đổi trạng thái lối ra.

8. *Tần số xung nhịp cực đại (Maximum clock rate)*: Khi các cửa logic dùng làm trigger thì loại cửa nào có T_d nhỏ sẽ làm việc được với những xung nhịp tần số cao, tốc độ chuyển mạch nhanh.

3.2. HỌ LOGIC RTL (RESISTOR - TRANSISTOR - LOGIC)

3.2.1. Mạch đảo (NOT)

Các IC họ RTL được nuôi bằng nguồn $V_{CC} = +3,6V$, điện áp vào và ra của mạch chỉ có hai mức. Với họ logic RTL giá trị điện áp của các mức logic như sau :

$$L = "0" = 0 \text{ v}$$

$$H = "1" > 1,5 \text{ v}$$

Trên hình 3.1 trình bày sơ đồ nguyên lý của mạch đảo họ RTL.

Điện áp ra của mạch phụ thuộc vào điện áp vào tuân theo hàm logic :

$$y = \bar{x} .$$

Sau đây ta sẽ khảo sát nguyên lí hoạt động của mạch.

Từ sơ đồ nguyên lí hình 3.1 chúng ta thấy: điện áp ra của mạch là điện thế trên collector của transistor (V_c).

Thế lối ra :

$$V_c = V_{cc} - I_c R_2$$

Khi $x = L = 0$: $I_B = 0 \rightarrow I_c = 0$

$$y = V_c = V_{cc} = H = "1"$$

Khi $x = H = 1$: $I_B \uparrow \rightarrow I_c \uparrow$

$$y = V_c = V_{cc} - I_c R_2 = L = "0"$$

Hoạt động của mạch đúng như phương trình logic của mạch đảo.

3.2.2. Mạch Không Hoặc (NOR)

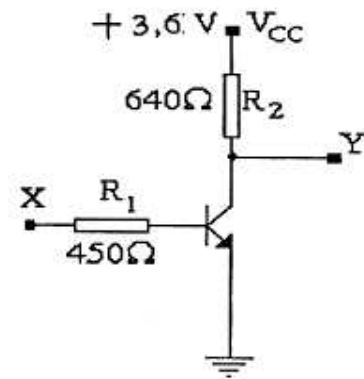
Hàm logic: $y = \overline{x_1 + x_2}$

Bảng chân lý cho trên Bảng 3.1.

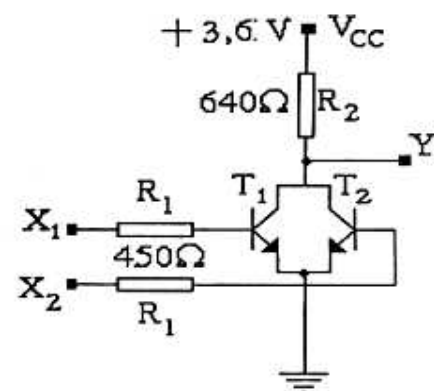
Sơ đồ mạch trên hình 3.2.

Bảng 3.1: Bảng chân lý của mạch NOR

x_1	x_2	$y = \overline{x_1 + x_2}$
L	L	H
H	L	L
L	H	L
H	H	L



Hình 3.1: Mạch đảo họ RTL



Hình 3.2: Mạch NOR họ RTL

Hai lối vào $x_1 = x_2 = L$ thì T_1, T_2 cấm, lối ra ở mức H.

Một trong 2 lối vào ở mức cao H \rightarrow lối ra ở mức L.

Hai lối vào ở mức cao H \rightarrow 2 transistor thông bão hoà \rightarrow lối ra ở mức L.

Mạch hoạt động đúng như bảng chân lý (3.1).

3.2.3. Mạch và AND

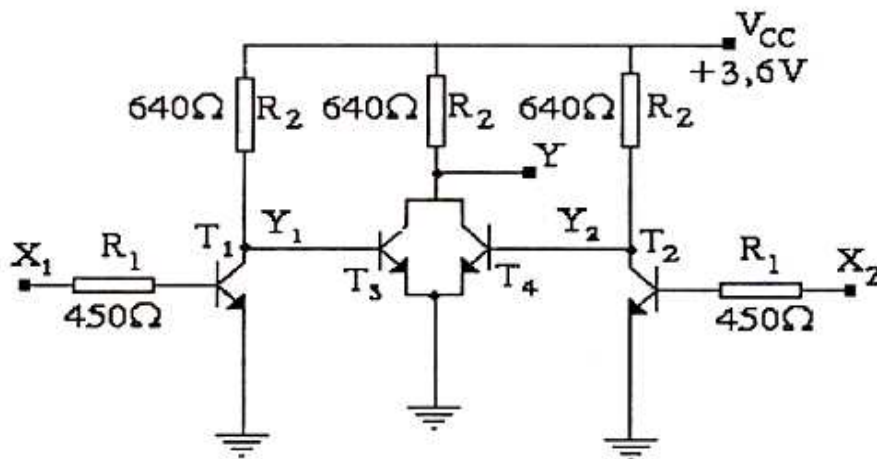
Hàm logic: $y = x_1 \cdot x_2$

Bảng chân lý cho trên Bảng 3.2.

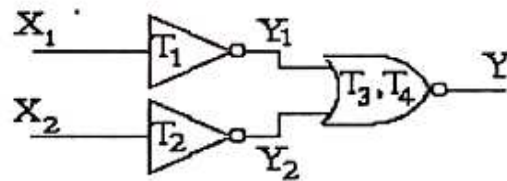
Bảng 3.2: Bảng chân lý của mạch AND

x_1	x_2	$y = x_1 \cdot x_2$
L	L	L
H	L	L
L	H	L
H	H	H

Sơ đồ mạch cho trên hình 3.3.



Hình 3.3: Sơ đồ nguyên lý của mạch AND họ RTL



Hình 3.4: Sơ đồ logic của mạch VÀ

Từ sơ đồ nguyên lí ta thấy T_1, T_2 là hai phần tử đảo; T_3, T_4 tạo thành mạch NOR, vì vậy mạch có sơ đồ logic tương ứng được vẽ trên hình 3.4.

Từ hình 3.4 ta thấy: $y = \overline{\overline{x_1 + x_2}}$

Áp dụng định lí De-Morgan ta có:

$$y = \overline{\overline{x_1 \cdot x_2}} = x_1 \cdot x_2$$

Vậy đây chính là mạch VÀ, làm việc đúng như bảng chân lý là bảng 3.2.

Họ logic RTL hiện nay không còn được sản xuất nữa, tuy nhiên nó vẫn được dùng trong các mạch điều khiển.

3.3. MẠCH LOGIC HỘ DTL (DIODE - TRANSISTOR - LOGIC)

3.3.1. Mạch đảo (NOT)

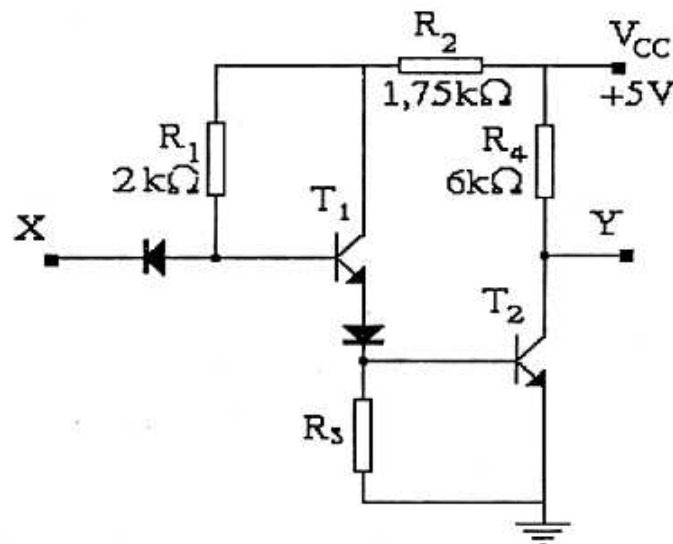
Hàm logic: $y = \overline{x}$

Sơ đồ mạch như trên hình 3.5.

Giải thích nguyên lý hoạt động của mạch:

- Bình thường nếu lối vào để hở mạch V_{B1} luôn ở mức cao (H) làm cho T_1 thông, T_2 thông và

$$y = L.$$



Hình 3.5: Mạch NOT hộ DTL

- Nếu $x = H (> 2,2V)$ điôt không dẫn và

$V_{B1} = H$, T_1, T_2 thông và $y = L$.

- Nếu $x = L (= 0 \text{ V}) \rightarrow V_{B1} = L$, T_1 cấm, T_2 cấm và $y = H$.

Như vậy y luôn luôn là đảo của x .

3.3.2. Mạch Không Và (NAND)

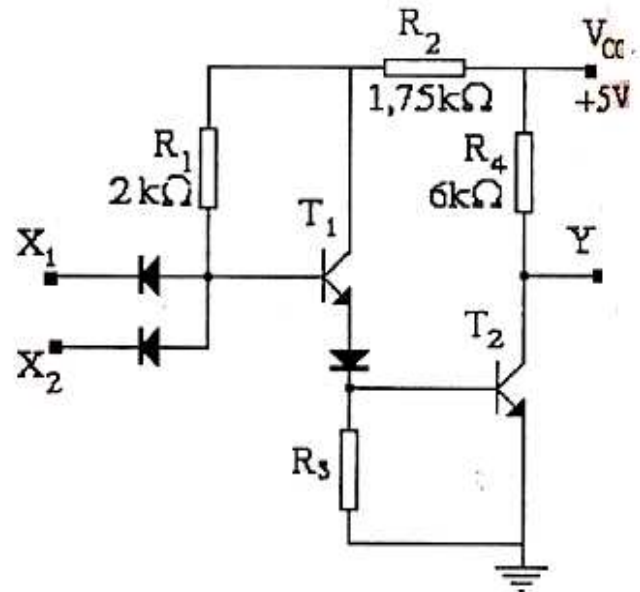
Hàm logic: $y = \overline{x_1 \cdot x_2}$

Bảng chân lý cho trên bảng 3.3

Sơ đồ mạch như trên hình 3.6.

Bảng 3.3: Bảng chân lý của mạch NAND

x_1	x_2	$y = \overline{x_1 \cdot x_2}$
H	H	L
L	H	H
H	L	H
L	L	H



Hình 3.6: Mạch NAND họ DTL

Sơ đồ mạch đảo trên hình 3.5 nếu ta mắc song song 2 điốt với 2 lối vào x_1, x_2 như hình 3.6 ta sẽ được mạch NAND hai lối vào. Phân tích tương tự như với mạch đảo ta thấy hoạt động của sơ đồ này theo đúng bảng chân lý mạch NAND cho trên bảng 3.3.

- Nếu ở lối vào ta mắc song song n điốt ta sẽ được mạch NAND n lối vào theo hàm logic:

$$y = \overline{x_1 \cdot x_2 \cdot \dots \cdot x_n}$$

- Đặc điểm của họ logic DTL là tốc độ làm việc chậm ($f < 1 \text{ MHz}$), được dùng trong điện tử công nghiệp, điện tử y tế. Ngày nay họ DTL không được sản xuất nữa. Trong các thiết bị điện tử có dùng loại vi mạch thuộc họ logic này nếu vi mạch bị hỏng cần phải thay thế ta có thể dùng các vi mạch logic họ TTL có chức năng tương tự.

3.4. HỌ LOGIC TTL (TRANSISTOR - TRANSISTOR - LOGIC)

$$V_{outL} = 0,2V \div 0,4V ;$$

$$V_{inL} < 0,8V, V_{inH} \geq 2V$$

3.4.1. Mạch Đảo (NOT)

Hàm logic: $y = \bar{x}$

Sơ đồ mạch trên hình 3.7.

Từ sơ đồ ta thấy:

- Nếu lối vào x ở mức H (hoặc hở mạch): điện áp base của T_1 , T_2 đều ở mức cao: V_{B1} , V_{B2} cao, T_2 thông bão hoà, V_{C2} thấp, T_3 cấm không có dòng chạy qua. Điện áp base T_4 được xác định theo dòng emitter của T_2 . Từ sơ đồ nguyên lý ta có:

$$V_{B4} = V_{E2} = I_{E2} \cdot R_3$$

Khi T_2 thông bão hoà V_{B4} ở mức cao T_4 thông mạch và y ở mức thấp (L).

Như vậy, khi: $x = H \rightarrow y = L$

Nếu lối vào x ở mức L: T_2 cấm, T_4 cấm, T_3 thông và y ở mức cao (H).

Ta có: $x = H \rightarrow y = L$

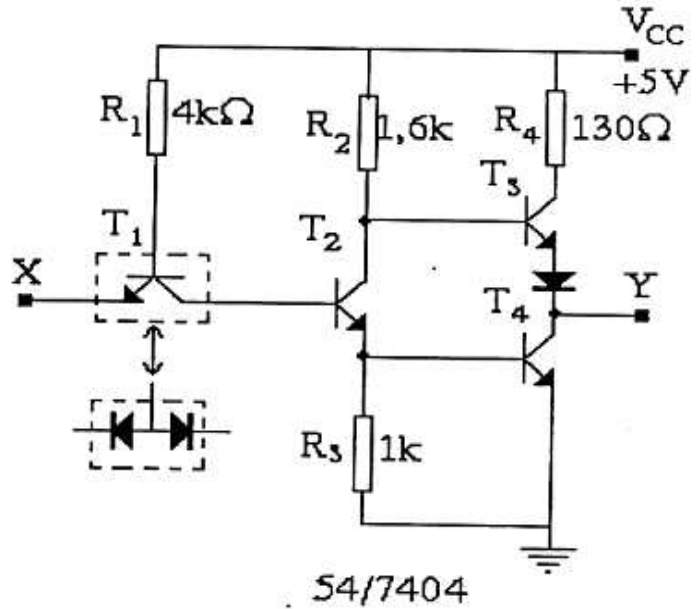
Hoạt động của mạch diễn ra đúng như hàm NOT.

3.4.2. Mạch Không Và (NAND)

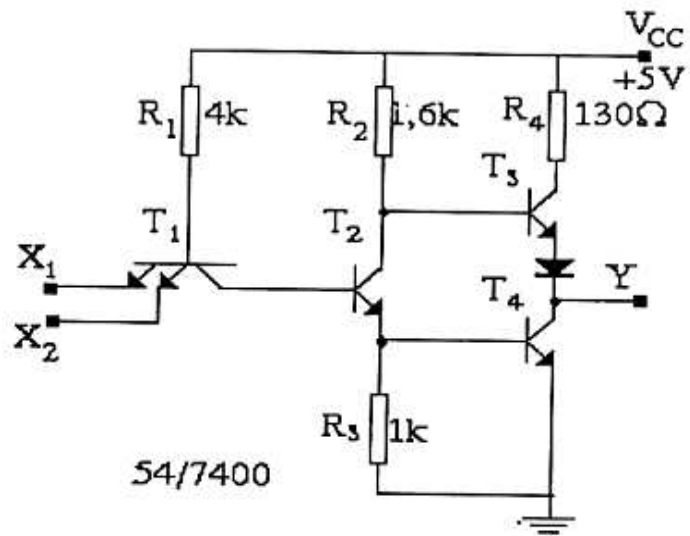
$$Hàm logic: y = \overline{x_1 \cdot x_2}$$

Bảng chân lý cho trên bảng 3.4.

Sơ đồ mạch được vẽ trên hình 3.8.



Hình 3.7: Sơ đồ nguyên lý của mạch NOT họ TTL



Hình 3.8: Mạch NAND 54/7400 họ TTL

Bảng 3.4: Bảng chân lý của mạch NAND

x_1	x_2	$y = \overline{x_1 \cdot x_2}$
H	H	L
H	L	H
L	H	H
L	L	H

Cũng phân tích nguyên lý hoạt động của mạch tương tự như với mạch đảo nói trên ta có:

a/ Nếu $x_1 = x_2 = H \rightarrow T_2, T_4$ thông, T_3 cấm $y = L$.

b/ Nếu x_1 hoặc x_2 hoặc cả hai lối vào đều ở mức L thì T_2, T_4 cấm, T_3 thông $y = H$.

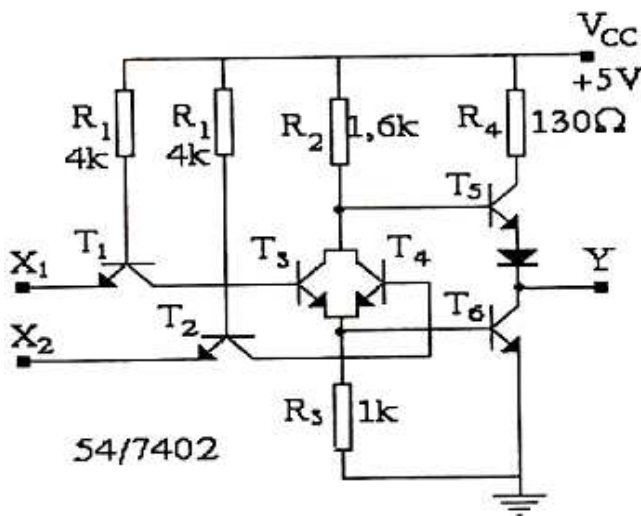
Ta thấy sơ đồ hoạt động theo đúng bảng chân lý bảng 3.4.

3.4.3. Mạch Không Hoặc (NOR)

Hàm logic: $y = \overline{x_1 + x_2}$

Bảng chân lý của mạch NOR cho trên bảng 3.5.

Sơ đồ mạch được trình bày trên hình 3.9.



Hình 3.9: Mạch NOR 54/7402 họ TTL

Bảng 3.5: Bảng chân lý của mạch NOR

x_1	x_2	$y = \overline{x_1 + x_2}$
H	H	L
H	L	L
L	H	L
L	L	H

Nhìn sơ đồ ta thấy mạch hai lối vào X_1 và X_2 có cấu trúc giống nhau (giống mạch NOT) được ghép song song với nhau

qua cặp transistor T_3 và T_4 , chỉ khi cả hai lối vào ở mức điện áp thấp ($L = 0V$) cả hai transistor T_3 và T_4 đều bị cấm, T_5 thông T_6 cấm, lối ra Y ở mức cao H ($H > 2,2V$).

- Phân tích hoạt động của sơ đồ trên, ta cũng thấy rằng hoạt động của mạch tuân theo bảng chân lý: bảng 3.5.

- Họ logic TTL có thể ghép nối với họ DTL được vì cùng giá trị nguồn vào V_{cc} và cùng mức logic H và L. Họ TTL hiện nay vẫn còn được sản xuất.

Ký hiệu mạch logic họ TTL: SN 7400, SN 5400, DM7400, DM 7402, Các chữ cái đầu chỉ kí hiệu riêng của từng hãng sản xuất.

SN: Hãng sản xuất: TEXAS INSTRUMENTS.

Hai số đầu chỉ dải nhiệt độ làm việc: ví dụ: SN 5402, SN 7402

74: $0^{\circ}\text{C} \div 70^{\circ}\text{C}$, 54: $- 55^{\circ}\text{C} \div 125^{\circ}\text{C}$

Mai hoặc ba số sau chỉ chức năng logic: 02: mạch NOR (có 4 NOR 2 lối vào $V_{cc} = +5\text{v}$).

SN 7400: 00: Mạch NAND 2 lối vào (4 NAND, $V_{cc} = + 5\text{v}$).

SN 7401: 01: Mạch NAND 2 lối vào có lối ra là transistor để hở mạch collector.

SN 7402: 02: Mạch NOR hai lối vào.

SN 74L00: Chữ L chen giữa chỉ công suất tiêu thụ thấp (low Power).

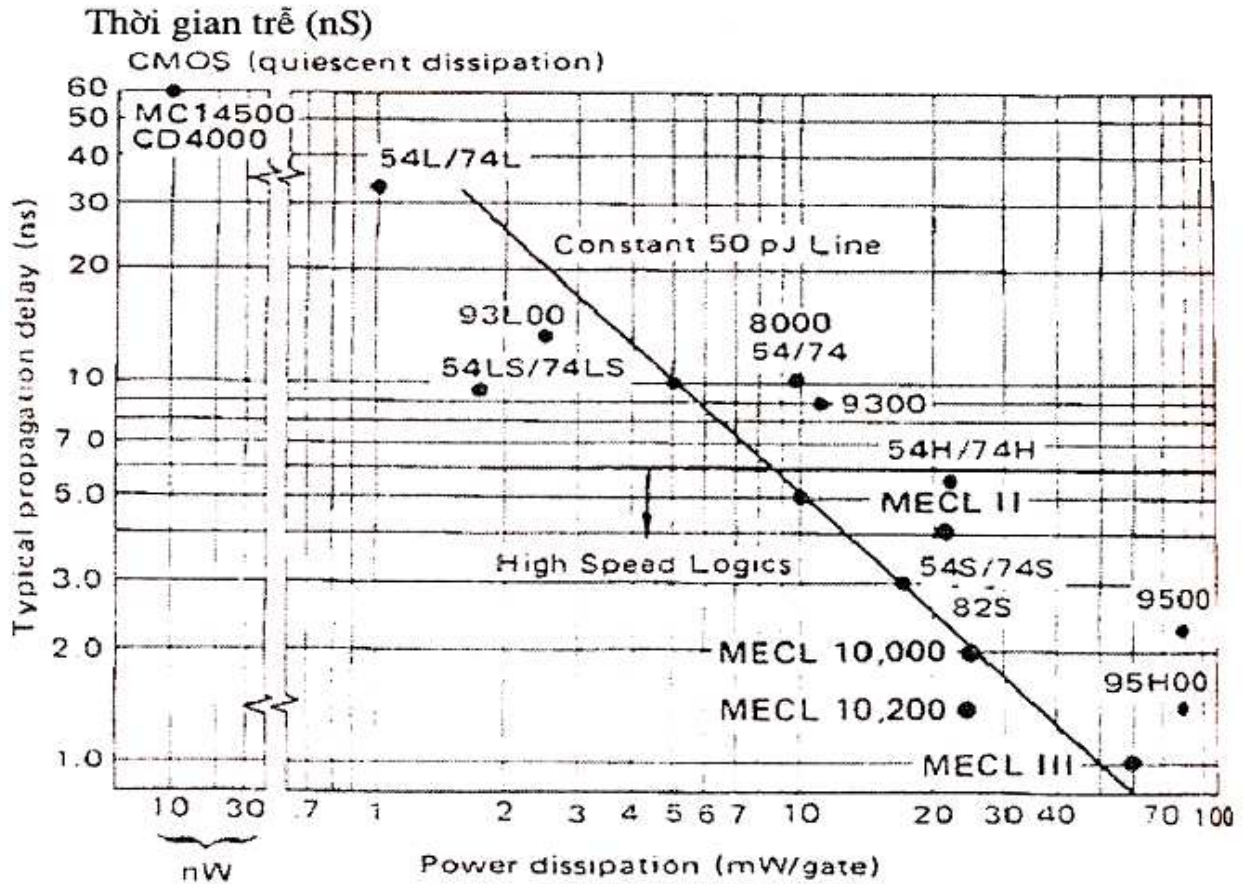
SN 74H00: Chữ H: High speed: tốc độ cao thời gian trễ nhỏ $T_D = 6 \text{ nS}$, tần số làm việc cao $f = 50 \text{ MHz}$. Còn SN 7400: $T_D = 10 \text{ nS}$ $f = 20 \text{ MHz}$.

SN 74S00: Chữ S chỉ rõ trong mạch có dùng diốt Schottky và transistor Schottky, chính vì vậy mà tốc độ chuyển mạch của loại này có tốc độ nhanh nhất và tần số làm việc của mạch cao như loại có ký hiệu chữ H.

SN 74LS00: Chữ LS: công suất tiêu thụ nhỏ có diốt Schottky và transistor Schottky.

Quy luật chung là muốn có tần số f cao thì công suất tiêu thụ phải lớn. Trong họ TTL thì loại vi mạch có thêm ký hiệu chữ H và chữ S tiêu thụ công suất lớn nhất nhưng cũng có tốc độ chuyển mạch nhanh nhất (thời gian trễ nhỏ nhất). Ví dụ như vi mạch NAND có ký hiệu SN 74LS00 công suất tiêu thụ nhỏ hơn SN 74H00 và SN 74S00 nhưng tốc độ chuyển mạch lại chậm hơn.

Đồ thị hình 3.10 ở dưới cho chúng ta thấy mối tương quan giữa thời gian trễ với công suất tiêu thụ năng lượng điện cho một cửa logic của các họ logic khác nhau. Từ đồ thị chúng ta thấy rằng: thời gian trễ tỷ lệ nghịch với công suất.



Hình 3.10. Công suất tiêu thụ năng lượng điện cho một cửa logic (mW/cửa logic)

3.4.4. Hai loại mạch có tốc độ nhanh 74H và 74S thuộc họ TTL

Từ đồ thị đặc trưng thời gian trễ – công suất của các họ mạch logic ở trên chúng ta thấy rằng: tiêu chuẩn để phân định vi mạch có tốc độ nhanh là thời gian trễ $T_d < 6 \text{ ns}$ (tần số làm việc $> 50 \text{ MHz}$). Trong họ logic TTL có hai loại 54H/74H và 54S/74S là hai loại vi mạch có tốc độ cao. Sau đây chúng ta sẽ khảo sát vi mạch SN 74H00 là mạch NAND Chữ "H": High speed. Đây là vi mạch có tốc độ làm việc cao. Tần số làm việc cỡ 50 MHz, có ưu điểm này là vì sơ đồ mạch được cải tiến hơn. Sơ đồ nguyên lí của mạch SN 74H00 được vẽ trên hình 3.11.

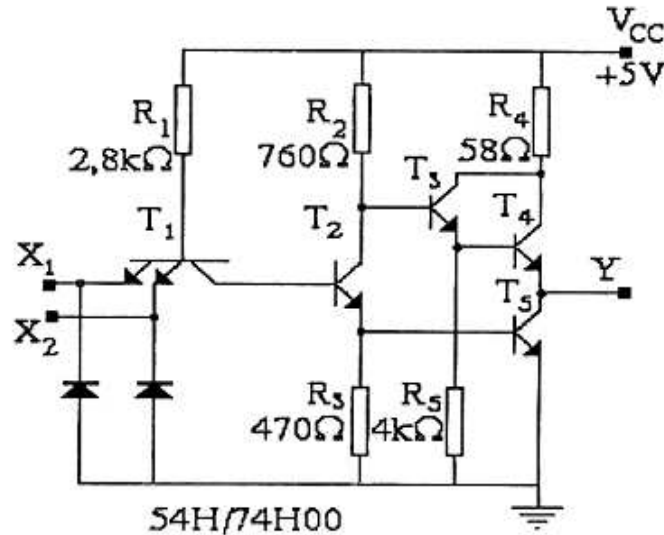
So sánh với sơ đồ mạch 7400 ta thấy ở đây các điện trở nhỏ đi làm cho công suất tiêu thụ tăng. Hơn nữa trong sơ đồ có thêm 2 diốt để bảo vệ lối vào: lối vào không được âm quá so với đất (chỉ được phép $\leq -0,6\text{V}$), khi mạch làm việc ở tần số cao do quá trình quá độ xuất hiện các dao động Rolắc, các dao động này được 2 diốt này cắt đi. Một cải tiến quan trọng nữa

ở lối ra của mạch có các transistor T_3 và T_4 được mắc theo kiểu tổ hợp **Darlington**, theo cách mắc này mạch cho dòng ra rất lớn. Cặp transistor T_3 và T_4 có thể coi như là một transistor có Emitơ chính là Emitơ của T_4 , dòng Emitơ của T_4 rất lớn. Thật vậy, nếu xét riêng cặp transistor T_3 và T_4 ta thấy:

$$I_{B3} = (1 + \beta_3) I_{B3} = I_{B4}$$

$$I_{E4} = (1 + \beta_4) (1 + \beta_3) I_{B3} \text{ và}$$

$$I_C = I_{C3} + I_{C4} = \beta_3 I_{B3} + (1 + \beta_3) \beta_4 I_{B3}$$



Hình 3.11: sơ đồ vi mạch NAND SN 74H00

Như vậy là hệ số khuếch đại dòng $\sim \beta_3\beta_4$ là rất lớn. Điện trở lối vào của sơ đồ Darlington cũng rất lớn hơn của transistor thường. Transistor thường có điện trở vào $r_v = r_B + (1 + \beta)r_E$.

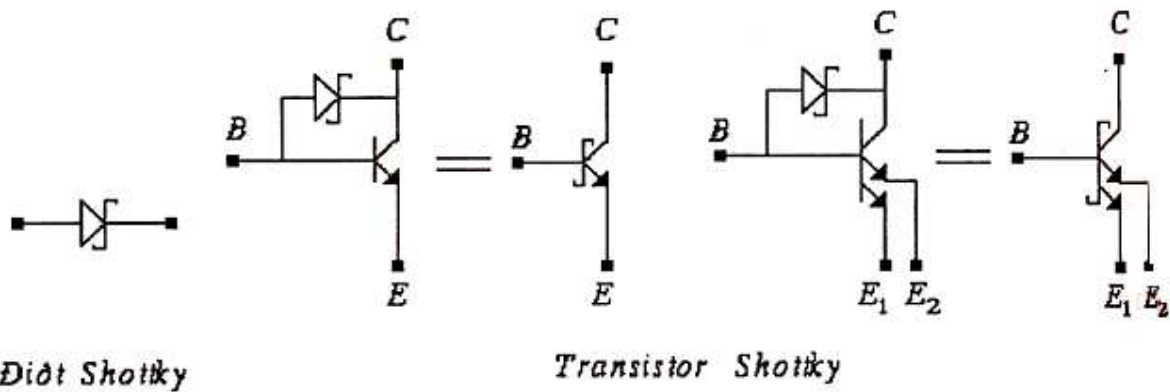
Sơ đồ Darlington trên có:

$$R_v = r_{B3} + (1 + \beta_3)(r_{E3} + r_{B4}) + (1 + \beta_3)(1 + \beta_4)r_{E4}: \text{rất lớn}$$

Như vậy ở SN 74H00 có tầng khuếch đại ở lối ra có hệ số khuếch đại rất cao có trở vào lại rất lớn. Đây là ưu điểm lớn nhất của vi mạch.

Vi mạch dùng điốt Shottky và transistor Shottky. Kí hiệu của điốt Shottky và transistor Shottky được trình bày trên hình 3.12.

Các vi mạch logic dùng điốt Shottky và transistor Shottky được kí hiệu bằng chữ S sau hai chữ số đầu chỉ nhiệt độ ví dụ như là: 74Sxx hoặc 54Sxx. Đặc điểm cơ bản của sơ đồ là dùng điốt Shottky và transistor Shottky.



Hình 3.12

Transistor Shottkylàm việc ở chế độ không bão hoà. Chuyển động của hạt tải qua chuyển tiếp không có hiện tượng tích tụ điện tích không gian nên dòng dịch chuyển có tác động rất nhanh, làm cho tần số làm việc cao hơn hẳn các vi mạch thuộc họ TTL không dùng điốt Shottky và transistor Shottky.

Điốt Shottky và transistor Shottky.

Trong điốt Shottky có lớp tiếp giáp giữa kim loại và bán dẫn loại n (lớp tiếp giáp Shottky). Giảm độ năng lượng của lớp tiếp giáp kim loại và bán dẫn loại n này được trình bày trên hình 3.13.

Kim loại có công thoát nhiệt điện tử lớn hơn bán dẫn $A_M > A_S$.



Hình 3.13

Điện tử trong bán dẫn n dễ dàng chuyển sang kim loại tại chỗ tiếp giáp kim loại bán dẫn, kim loại tích điện âm (-), bán dẫn tích điện dương (+). Điện thế tiếp giáp giữa kim loại và bán dẫn.

$$U_i = (A_M - A_S)/e$$

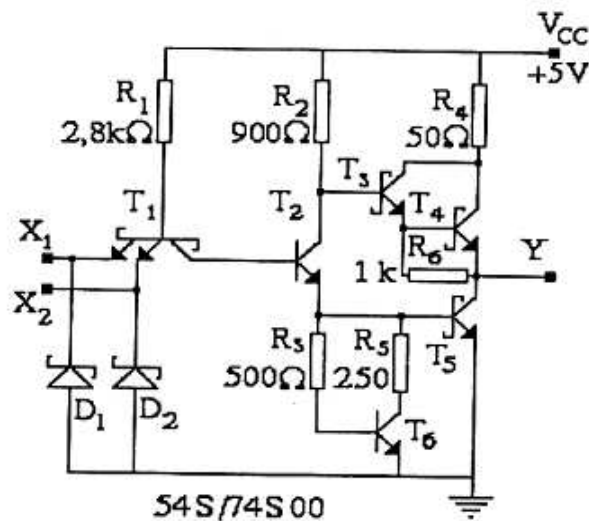
Điện thế này nhỏ nên điện áp mở của diốt Schottky cỡ $0,1 \div 0,3V$. Điện áp ngược cực đại của diốt Schottky cũng thấp so với diốt tách sóng thông thường

Khi phân cực thuận chiều của điện trường ngoài ngược với chiều điện trường của điện áp tiếp giáp U_i , điện tử từ bán dẫn dễ dàng chuyển sang kim loại. Dòng điện chạy một chiều từ kim loại sang bán dẫn loại n. Khi mắc diốt phân cực ngược điện trường ngoài cùng chiều với điện trường của điện áp tiếp giáp U_i , điện tử từ bán dẫn loại n không khuếch tán sang kim loại được nữa, dòng điện không qua được diốt. Tuy dòng điện qua diốt Schottky chỉ theo một chiều nhưng vì do cấu trúc của diốt Schottky với hàng rào thế năng Schottky trên hình 3.13 nên trong diốt Schottky không có hiện tượng tồn trữ điện tích, không đưa thêm thời gian trễ vào mạch, điện áp ngược đặt vào diốt Schottky chỉ cần vượt quá U_{ngmax} diốt Schottky đã thông mạch.

Transistor Schottky có cấu trúc gồm một transistor lưỡng cực thông thường có thêm một diốt Schottky nối giữa cực B và cực C của transistor (hình 3.12) Nhờ mắc thêm diốt Schottky song song với lớp tiếp giáp BC như vậy mà chênh lệch điện áp giữa cực Base và cực C của transistor không bao giờ vượt quá $0,5V$, chính vì thế Transistor Schottky luôn hoạt động ở chế độ không bị thông bão hòa như các transistor thông thường khác.

Dòng điện trong diốt Schottky là dòng các phân tử tải điện cơ bản là điện tử. Dùng diốt Schottky giảm hệ số ồn đáng kể.

Diốt Schottky được ứng dụng rất rộng rãi trong các sơ đồ tách sóng và trộn sóng ở dải tần số siêu cao tần, diốt Schottky được dùng trong các sơ đồ xung tốc độ chuyển mạch nhanh tần số làm việc có thể tới $500 MHz$. Trong các vi mạch số có tốc độ chuyển mạch nhanh đều phải dùng diốt Schottky.



Hình 3.14: Sơ đồ Vi mạch SN74S00

Sơ đồ vi mạch SN 74S00 thuộc họ TTL trong đó có dùng điốt Shottky và các Transistor Shottky cho trên hình 3.14.

Sơ đồ này về cơ bản giống SN 74H00 nhưng các điốt và các Transistor T_1, T_3, T_4, T_5 đã được thay bằng các điốt Shottky và Transistor Shottky. Các điện trở dùng trong 74S00 cũng có giá trị nhỏ như 74H00 nên công suất tiêu thụ của mạch này cũng lớn như mạch 74H00. Trong thực tế, khi sử dụng người ta thường chọn loại SN 74LS00 vì loại này vừa có công suất tiêu thụ nhỏ (L) vừa có tần số cao (S).

3.5. CÁC VI MẠCH LOGIC DÙNG TRANSISTOR TRƯỜNG

3.5.1. Khái quát về Transistor trường

Transistor trường là một loại dụng cụ bán dẫn mà hoạt động của nó dựa trên hiệu ứng trường. Dòng qua Transistor trường là dòng các phân tử tải điện cơ bản chạy qua kênh dẫn, kênh này được điều khiển bởi điện trường.

Người ta đã chế tạo 2 loại Transistor trường có cửa điều khiển hạt tải điện qua kênh dẫn cấu tạo khác nhau.

1. Transistor trường loại JFET (Junction Field Effect Transistor): là Transistor trường điều khiển hạt tải điện qua kênh dẫn bằng lớp tiếp giáp p-n hoặc bằng hàng rào Shottky.

2. MOSFET (Metal Oxide Semiconductor Field Effect Transistor) là loại Transistor trường điều khiển hạt tải điện qua kênh dẫn bằng cửa cách điện.

Transistor trường có những đặc điểm đặc biệt sau:

Transistor trường có điện trở lối vào rất lớn $r_v \approx \infty$ ($r_v \approx 10^{14} \Omega$), khác với Transistor lưỡng cực, dòng điện qua Transistor trường được điều khiển bằng điện áp chứ không phải bằng dòng điện như Transistor lưỡng cực, dòng vào của Transistor trường vô cùng nhỏ $I_v \approx 0$ ($I_v \approx 10\text{pA}$).

Transistor trường có tạp âm nội rất nhỏ. Tạp âm nội trong Transistor trường nhỏ hơn rất nhiều so với Transistor lưỡng cực.

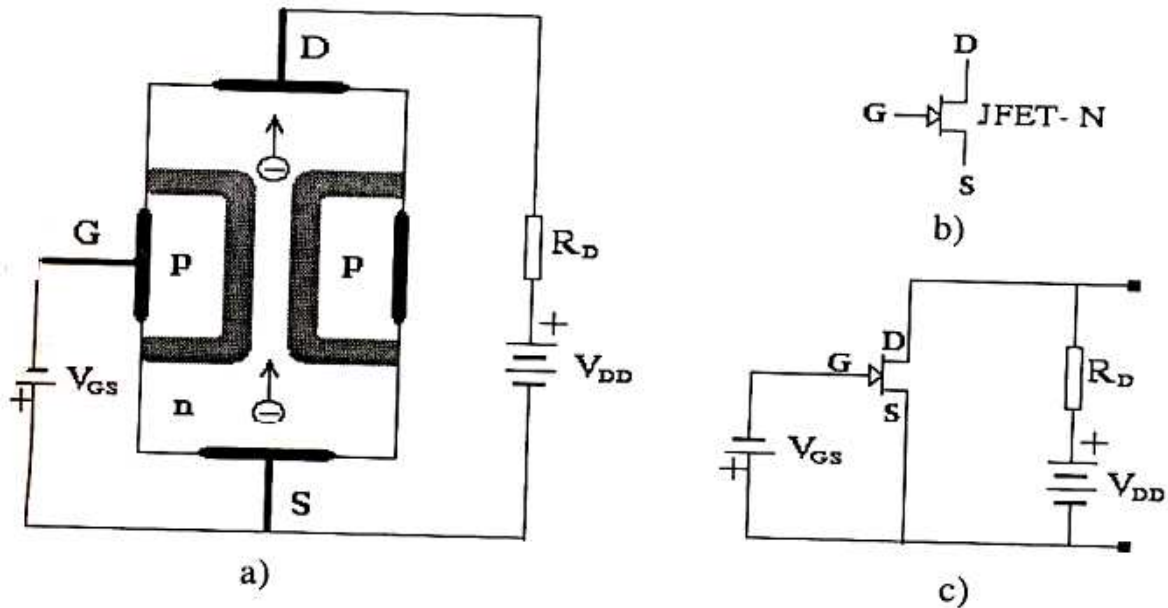
Công suất tiêu tán trên Transistor trường rất bé.

3.5.2. Cấu tạo và nguyên tắc hoạt động của JFET

3.5.2.1. Cấu tạo của JFET

Hình 3.15 a,b,c trình bày cấu trúc, ký hiệu và cách mắc nguồn điện cung cấp cho Transistor trường JFET kênh n. Transistor trường đơn giản

nhất có thể tạo từ một tấm bán dẫn loại n, có lớp tiếp giáp p-n. Tại phần trung tâm của tấm bán dẫn, người ta tạo một lớp tiếp giáp p-n tạo thành cửa điều khiển. Người ta đặt vào cửa điều khiển một điện áp phân cực ngược. Điện thế đặt vào cực cửa tăng sẽ làm thay đổi bề dày của vùng điện tích không gian lớp tiếp giáp.



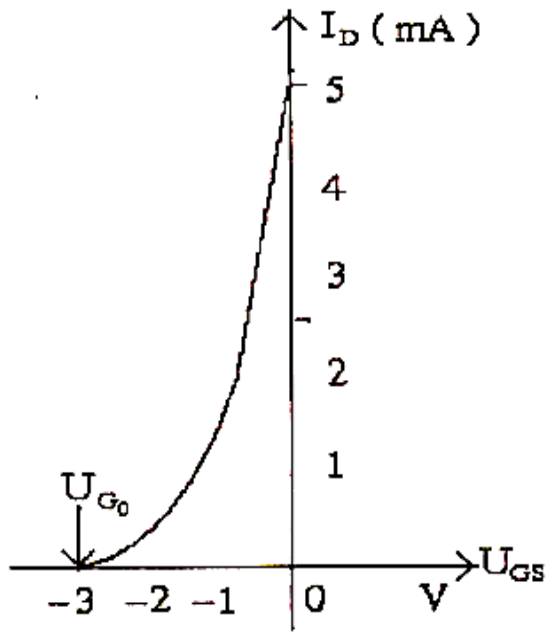
Hình 3.15. a) Cấu trúc của JFET kênh n;
b) Ký hiệu của JFET kênh n; c) Cách mắc nguồn điện

Tiết diện của kênh dẫn hẹp đi, điện trở của kênh dẫn tăng, dòng điện qua kênh dẫn giảm bởi vì phần tử tải điện cơ bản của bán dẫn loại n ở cực S đi vào miền điện tích không gian của lớp tiếp giáp p-n sẽ trung hoà với lỗ trống.

Thay đổi điện áp đặt vào G có thể thay đổi dòng qua kênh dẫn, điện áp đặt vào cực cửa đóng vai trò điện áp điều khiển dòng điện qua transistor trường. Dòng máng phụ thuộc vào điện áp giữa cực máng và cực nguồn U_{DS} và phụ thuộc vào điện áp giữa cực cửa điều khiển với cực nguồn: $I_D = f(U_{GS}, U_{DS})$

3.5.2.2. Các đặc trưng của JFET

Đặc trưng $I_D = f(U_{GS})$: Đường đặc trưng dòng máng phụ thuộc vào điện áp giữa cực cửa và cực nguồn khi $U_{DS} = \text{const}$ được trình bày trên hình 3.16. Đường đặc trưng trên được xác lập từ thực nghiệm.



$$I_D = f(U_{DS}, U_{GS})$$

$$I_D = I_{D0} \left(1 - \frac{U_{GS}}{U_{G0}} \right)^n$$

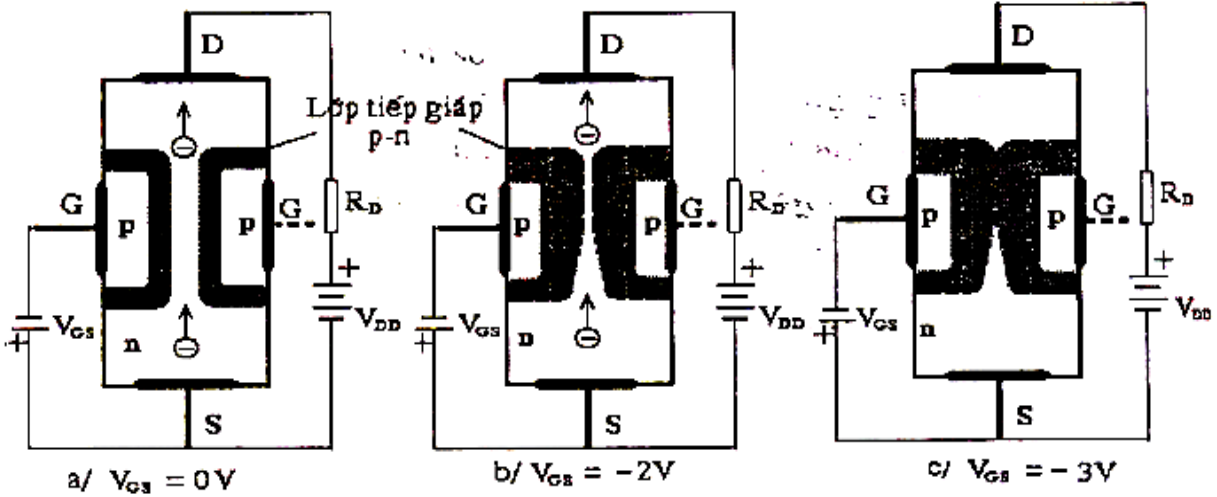
$$n = 1,5 \div 2,5$$

Độ dốc của đường đặc trưng

$$S \left(\frac{\text{mA}}{\text{V}} \right) = \frac{dI_D}{dU_{GS}} \Big|_{U_{DS} = \text{const}}$$

$$r_D = \frac{dU_{DS}}{dI_D} \Big|_{U_{DS} = \text{const}}$$

Hình 3.16

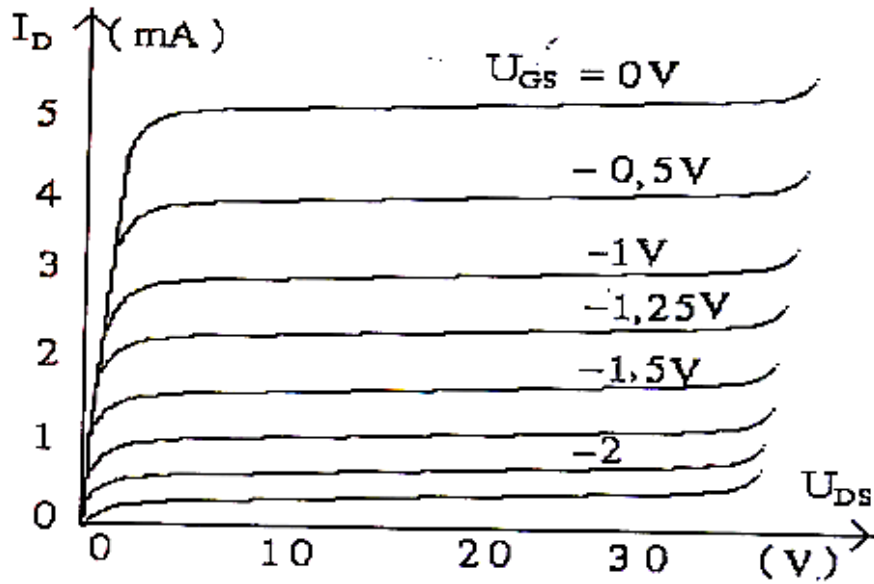


Hình 3.17

Trên hình 3.17 trình bày các hình ảnh minh họa sự thay đổi độ rộng kênh dẫn theo điện áp phân cực ngược đặt vào điốt: độ rộng kênh dẫn hẹp đi khi điện áp phân cực ngược đặt vào cực cửa tăng lên.

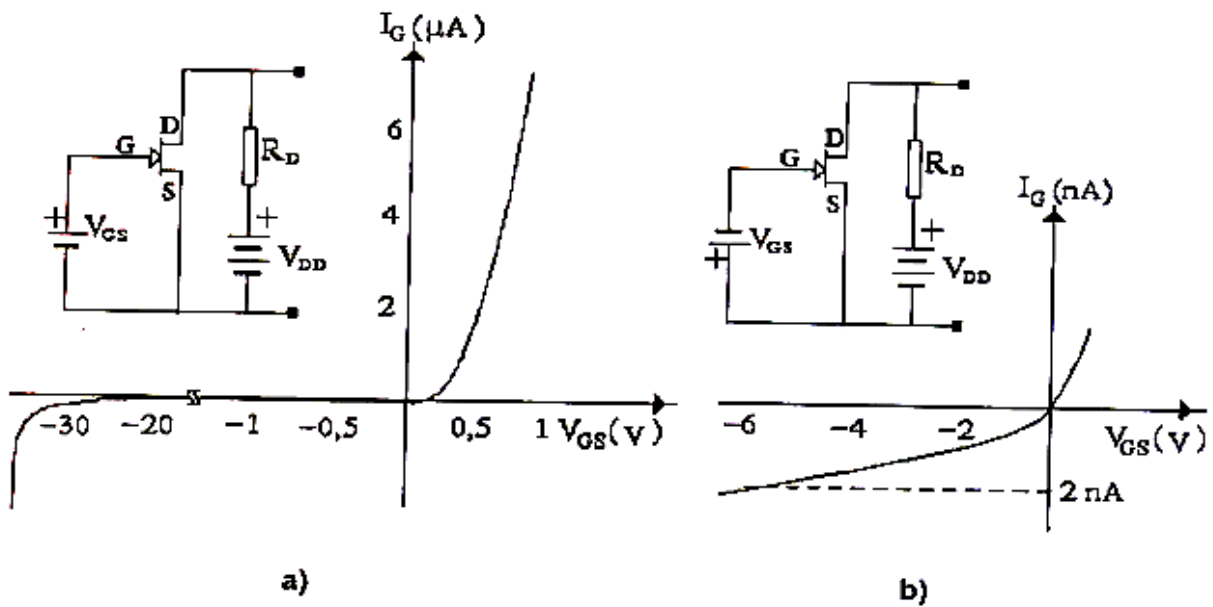
Đặc trưng $I_D = f(U_{DS})$

Họ các đường đặc trưng dòng máng phụ thuộc vào điện áp giữa cực cửa và cực nguồn ứng với các giá trị điện áp khác nhau của $U_{DS} = \text{const}$ được trình bày trên hình 3.18.



Hình 3.18

Đặc trưng $I_G = f(U_{GS})$



Hình 3.19. Đặc trưng ở lối vào của JFET kênh n
 a) Điện áp V_{GS} phân cực thuận; b) Điện áp V_{GS} phân cực ngược

Hình 3.19 biểu diễn đặc trưng dòng vào I_G của JFET. Khi điện áp V_{GS} mắc theo chiều phân cực thuận (hình 3.19a) đặc trưng có dạng giống như đặc trưng của diốt chỉnh lưu.

Từ hình 3.19b chúng ta thấy dòng I_G ứng với điện áp V_{GS} phân cực cực kỳ nhỏ, điều này cũng có nghĩa là điện trở lối vào của JFET kỳ lớn. Để JFET có thể khuếch đại được tín hiệu thì trong các sơ đồ khuếch đại dùng JFET phải đảm bảo cho cực cửa có điện áp phân cực ngược.

3.5.3. Cấu trúc và nguyên tắc hoạt động của Transistor loại MOSFET

3.5.3.1. Cấu trúc của MOSFET

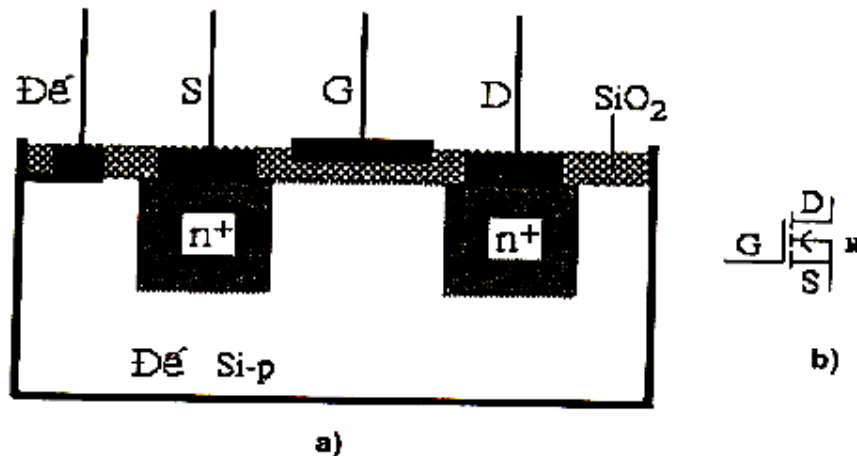
Điểm khác biệt về cấu trúc giữa Transistor trường loại JFET MOSFET là cực cửa G của MOSFET bằng kim loại không tiếp giáp trực tiếp với bán dẫn, nó được cách điện hoàn toàn với chất bán dẫn nhờ một lớp điện môi là lớp ôxit bán dẫn. Tùy theo cấu trúc của MOSFET chúng lại được chia thành hai loại:

- 1- loại có kênh cảm ứng
- 2- loại có kênh tạo sẵn

1. Transistor trường MOSFET loại có kênh cảm ứng

a) Cấu trúc của Transistor trường có kênh cảm ứng:

Cấu trúc và kí hiệu Transistor trường MOSFET kênh -n cảm ứng được trình bày trên hình 3.20.



Hình 3.20. a) Cấu trúc của MOSFET kênh n; b) Ký hiệu MOSFET - n

Trên một đế bán dẫn Si loại p có điện trở suất rất cao, người ta tạo một lớp điện môi SiO_2 cách điện bằng cách nung nóng ở nhiệt độ $1000^\circ C$ trong môi trường ôxy.

Bằng phương pháp quang khắc, người ta tạo 2 cửa sổ để rồi qua đó bằng phương pháp khuếch tán tạo ra hai vùng bán dẫn loại n pha tạp mạnh (n^+). Hai miền này có điện trở suất nhỏ tạo thành cực máng và cực nguồn.

Để tạo cực cửa, người ta phủ lớp kim loại lên lớp oxit nằm giữa 2 miền S và D.

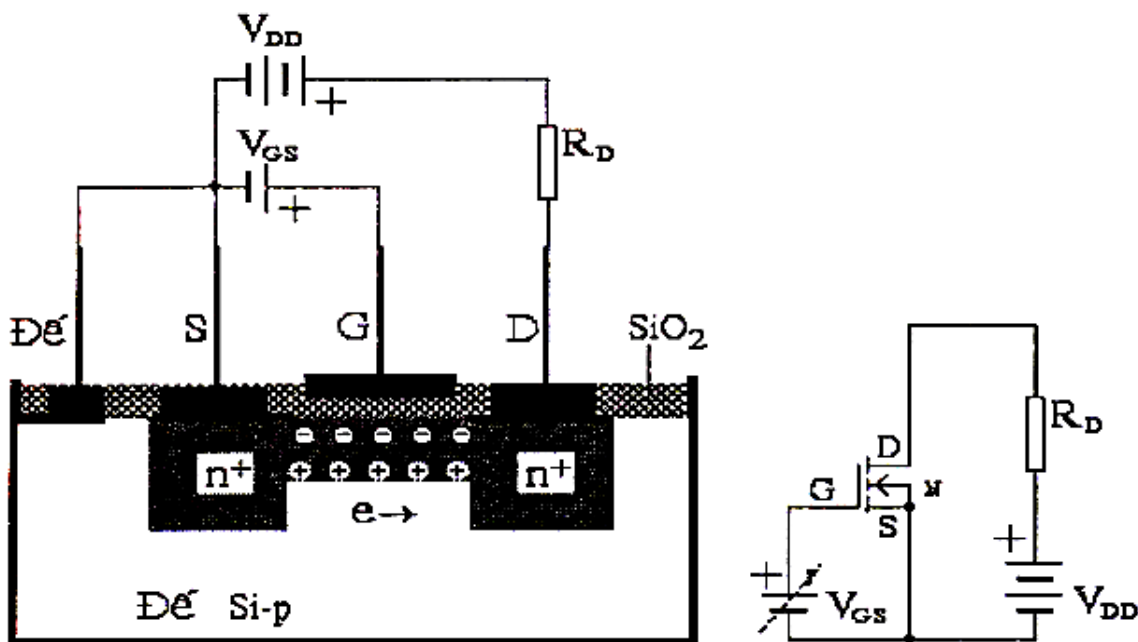
Lối ra lấy trên diện cực kim loại này gọi là cực cửa G.

Như vậy ta thấy giữa cực cửa và đế bán dẫn có một lớp điện môi rất mỏng là SiO_2 .

Hình 3.21 là hình ảnh minh họa sự tạo thành kênh cảm ứng khi có điện áp phân cực đặt vào cực cửa G.

b/ Nguyên tắc hoạt động của Transistor trường MOSFET kênh cảm ứng:

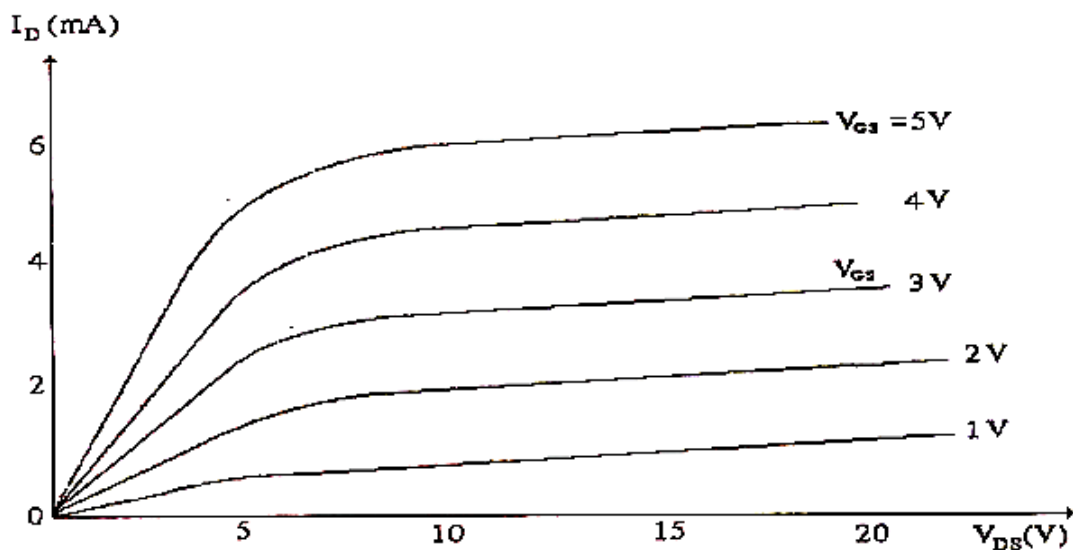
Sơ đồ mắc nguồn điện phân cực cho Transistor trường MOSFET kênh cảm ứng được trình bày trên hình 3.21. Khi chưa có điện áp đặt vào cực cửa, cực đế và cực nguồn thì kênh dẫn giữa S và D chưa được hình thành. Khi ta nối cực cửa với cực dương và nối đế và cực S với cực âm của nguồn điện V_{GS} . Do cách cấu tạo như trên mà giữa cực cửa và cực đế hình thành một tụ điện với khoảng cách giữa hai bản tụ là chiều dày lớp điện môi SiO_2 .



Hình 3.21. Sự hình thành kênh cảm ứng khi có điện áp phân cực

Do đó phần bên trong đế, ngay sát bề mặt cực cửa sẽ hình thành một lớp điện tích âm trái dấu. Lớp điện tích âm này sẽ tạo thành một kênh dẫn loại n nối liền hai cực nguồn S và cực máng D. Điện trường giữa hai cực cửa và đế càng lớn thì tiết diện của kênh cảm ứng càng lớn, do đó dòng điện giữa cực S và D càng lớn. Vì vậy, thay đổi điện áp điều khiển tác dụng lên cực cửa ta có thể điều khiển dòng điện qua Transistor.

Trên hình 3.22 trình bày đặc trưng Von – Ampe biểu diễn sự phụ thuộc của dòng máng vào điện thế giữa cực máng và cực nguồn của Transistor trường MOSFET kênh cảm ứng .



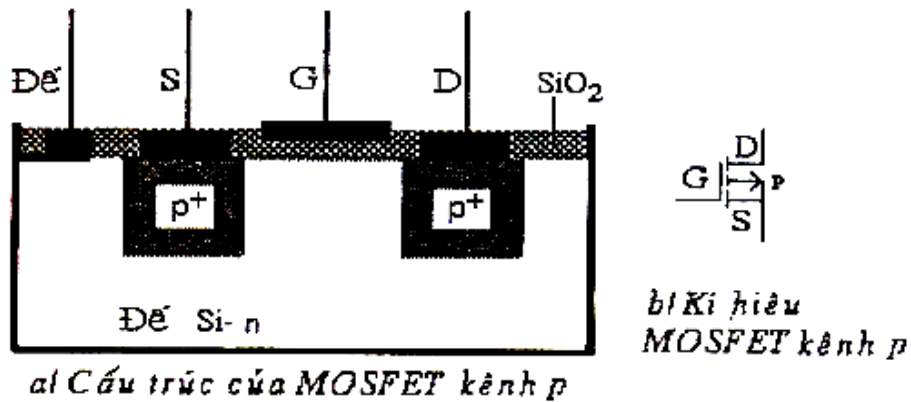
Hình 3.22

Transistor trường MOSFET kênh cảm ứng được dùng rất phổ biến trong các mạch vi điện tử logic. Phần sau chúng ta sẽ khảo sát các mạch vi điện tử số họ CMOS trong đó người ta đã sử dụng phối hợp hai loại transistor trường MOSFET kênh n và kênh p.

Bài tập:

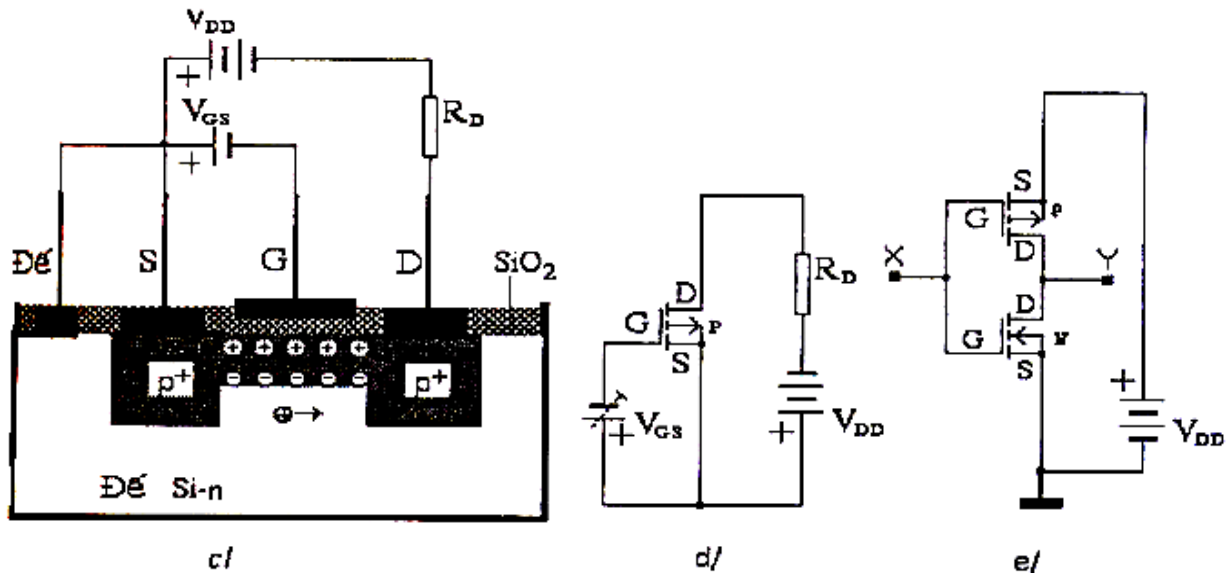
1/ Trên hình 3.23 a,b trình bày cấu trúc của Transistor trường MOSFET kênh -p cảm ứng và ký hiệu của Transistor trường MOSFET kênh -p cảm ứng.

Hãy giải thích cấu tạo và nguyên tắc hoạt động của transistor này dựa vào hình ảnh minh họa sự tạo thành kênh dẫn cho trên hình 3.23 c khi có điện áp phân cực mắc theo sơ đồ hình 3.23d.



Hình 3.23a,b

2/ Xác định điện áp ở lối ra Y của sơ đồ mạch hình 3.23e trong hai trường hợp: Điện áp lối vào X=0, X = V_{DD}.

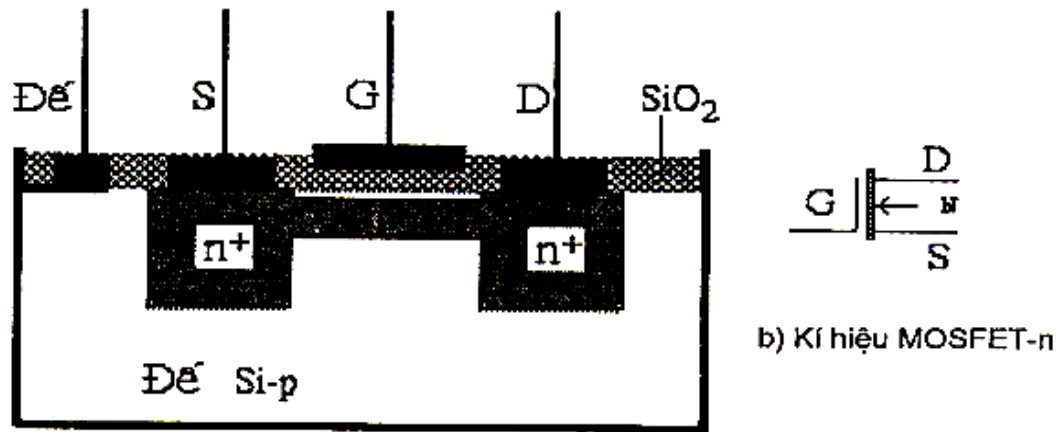


Hình 3.23c,d,e

2. Transistor trường MOSFET loại có kênh tạo sẵn

Quy trình chế tạo MOSFET có kênh được tạo sẵn cũng trải qua các bước tương tự như chế tạo MOSFET có kênh cảm ứng. Trên hình 3.24 trình bày cấu trúc của MOSFET kênh -n được tạo sẵn.

Kênh dẫn điện loại n nối liền giữa cực nguồn S và cực máng D cũng được tạo ra bằng phương pháp khuếch tán như cực nguồn và cực máng nhưng với nồng độ tạp chất ít hơn.



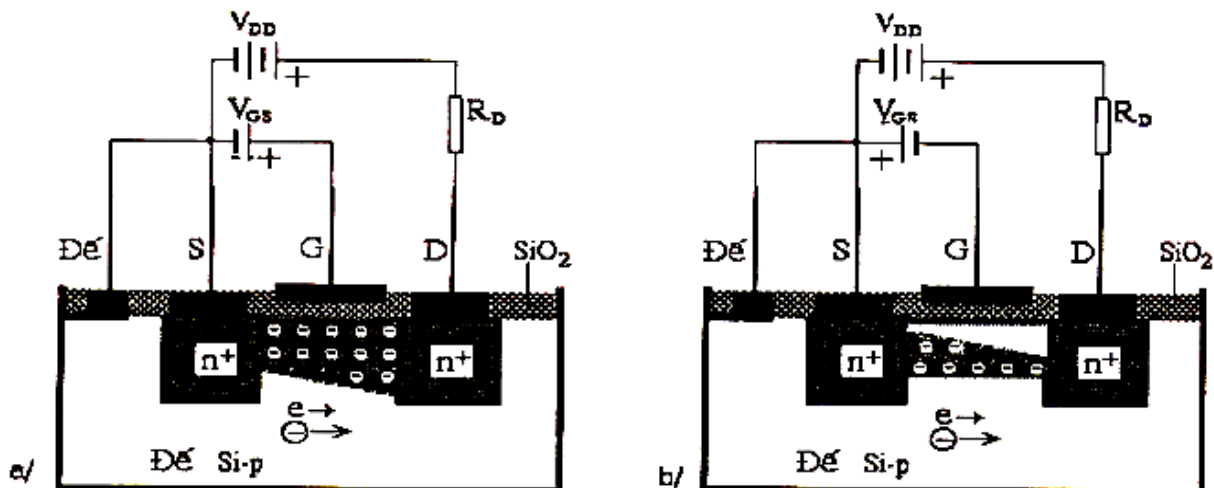
a) Cấu trúc của MOSFET kênh n được tạo sẵn

Hình 3.24

Sau khi hoàn thành công đoạn tạo được kênh dẫn n ở cực cửa người ta lại nung nóng ở nhiệt độ cao trong môi trường oxy để tạo được một lớp cách điện mỏng SiO_2 , và người ta dùng phương pháp phun kim loại trong chân không để phủ một lớp kim loại ở trên lớp ôxít nằm trên kênh dẫn n để tạo thành cực cửa G

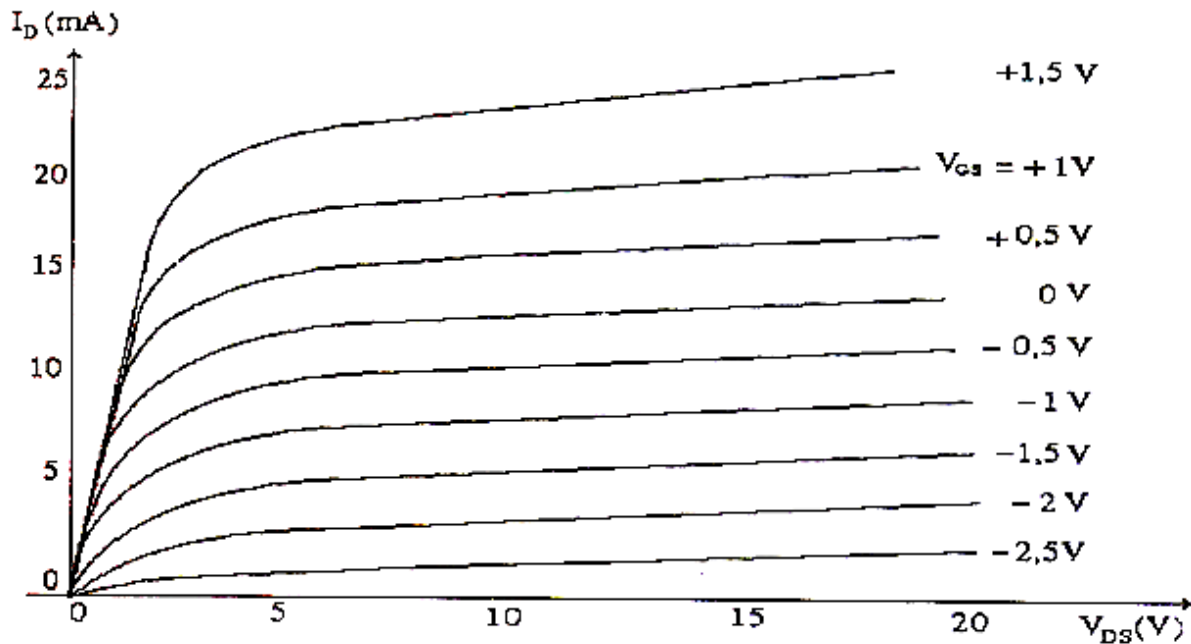
Nguyên tắc hoạt động của Transistor MOSFET có kênh tạo sẵn.

Hình 3.25 là sơ đồ mắc nguồn nuôi cho Transistor MOSFET có kênh tạo sẵn.



Hình 3.25

Khi nối cực cửa với nguồn điện thế âm (hình 3.25b), trong kênh n sẽ hình thành lớp nghèo phần tử tải điện. Do đó, nó làm giảm tiết diện của kênh, làm tăng điện trở của kênh dẫn và làm cho dòng I_{DS} giảm.

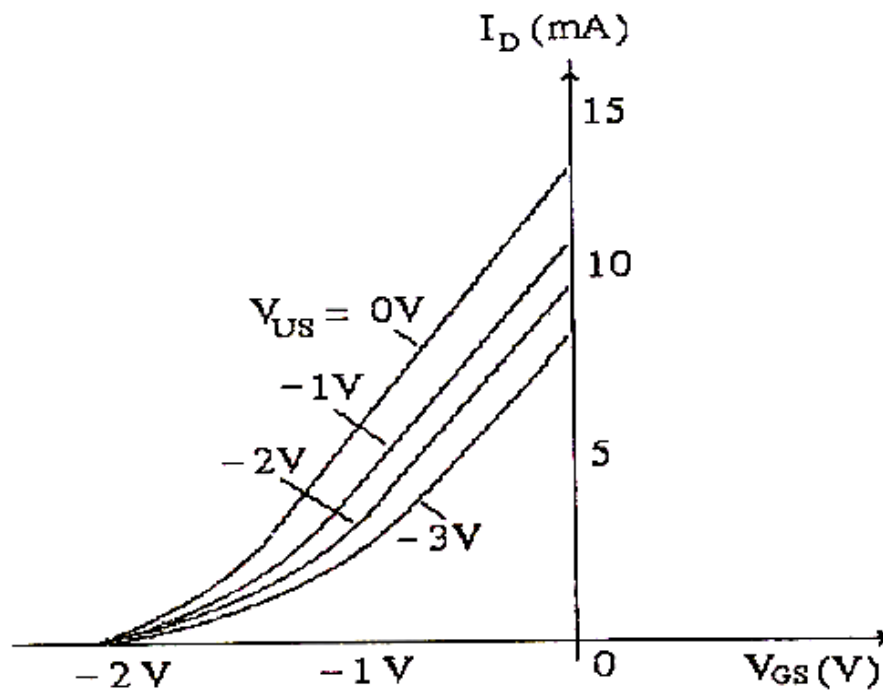


Hình 3.26

Đặt điện thế dương vào cực cửa (hình 3.25 a), tiết diện kênh dẫn tăng lên, điện trở kênh dẫn giảm làm cho dòng I_{DS} tăng. Như vậy, thay đổi thế phân cực đặt vào cực cửa ta sẽ thay đổi được tiết diện của kênh dẫn, và do đó thay đổi được dòng điện giữa D và S. Đặc trưng Von - Ampe $I_D = f(V_{DS})$ của MOSFET kênh tạo sẵn được trình bày trên hình 3.26.

Hình 3.27 là đặc trưng $I_D = f(V_{GS})$ ứng với các giá trị điện áp khác nhau giữa cực đế và cực nguồn V_{US} . Dựa vào các đường đặc trưng này ta có thể xác định điểm làm việc ban đầu cho các tầng khuếch đại dùng MOSFET kênh tạo sẵn.

Đặc trưng hình 3.27 cho chúng ta thấy khi đế và cực nguồn nối tắt với nhau ($V_{US} = 0$), đường đặc trưng có dạng tuyến tính và độ dốc tốt nhất nên trong thực tế người ta hay nối cực đế với cực nguồn S.

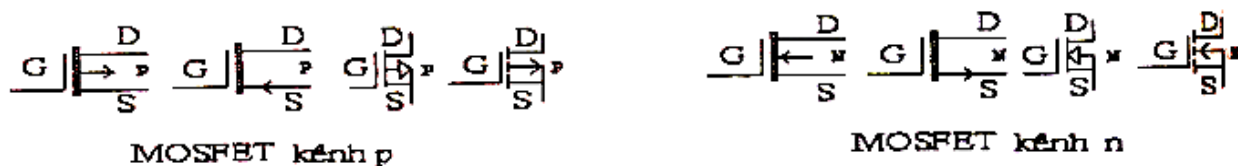


Hình 3.27

Vi mạch logic họ MOS

Transistor trường MOSFET được dùng để tạo các mạch logic. Các mạch logic thuộc họ MOS có công suất tiêu thụ nhỏ, ngày nay các mạch vi điện tử cỡ lớn LSI thường dùng MOSFET.

Các kí hiệu khác nhau của Transistor trường MOSFET thường gặp trong các tài liệu nước ngoài được nêu trên hình 3.28.



Hình 3.28: Kí hiệu Transistor trường MOSFET

Cửa NOT họ NMOS

Trên hình 3.29 giới thiệu sơ đồ các cổng NOT họ NMOS. Hình 3.29a là sơ đồ nguyên lí mạch NOT dùng Transistor trường MOSFET kênh n cảm ứng có điện trở tải R_D . Như ở trên ta đã khảo sát hoạt động của loại

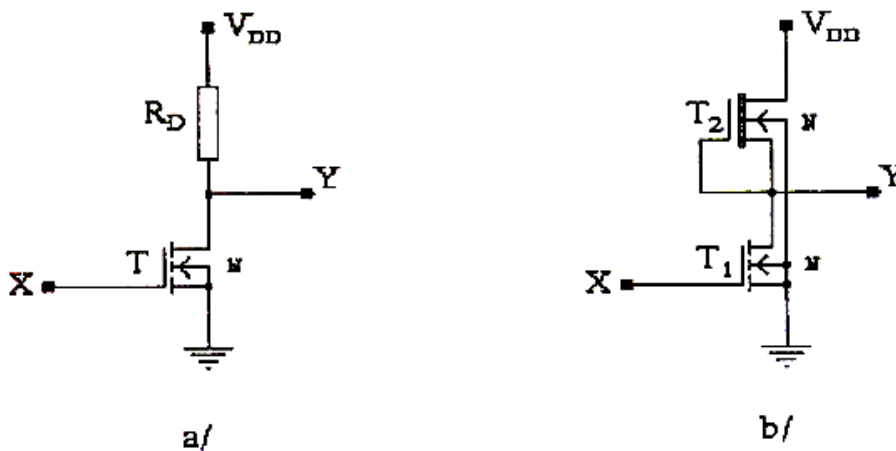
Transistor này, nếu điện áp đặt vào cực cửa $V_{GS} = 0\text{ V}$ thì kênh dẫn không được hình thành, Transistor trường MOSFET kênh n ở trạng thái cấm. Điều đó cũng có nghĩa là khi lối vào X của mạch ở mức thấp:

$X = L = 0\text{V}$ thì không có dòng qua Transistor, điện áp ra Y sẽ ở mức cao: $Y = H = V_{DD}$.

Khi $X = H = V_{DD}$ kênh dẫn được hình thành trong Transistor trường MOSFET kênh n cảm ứng, Transistor trường thông mạch cho dòng đi qua, điện áp lối ra: $Y = L = 0\text{V}$.

Như vậy hoạt động của mạch diễn ra đúng như mạch đảo NOT.

Sơ đồ nguyên lý mạch đảo hình 3.29b cũng tương tự như hình 3.29a, T_1 là MOSFET kênh n cảm ứng tương tự như T hình 3.29a, còn T_2 là Transistor trường MOSFET kênh n được tạo sẵn được dùng làm trở tải thay thế cho điện trở R_D . Hoạt động của mạch tương tự như mạch hình 3.29a.



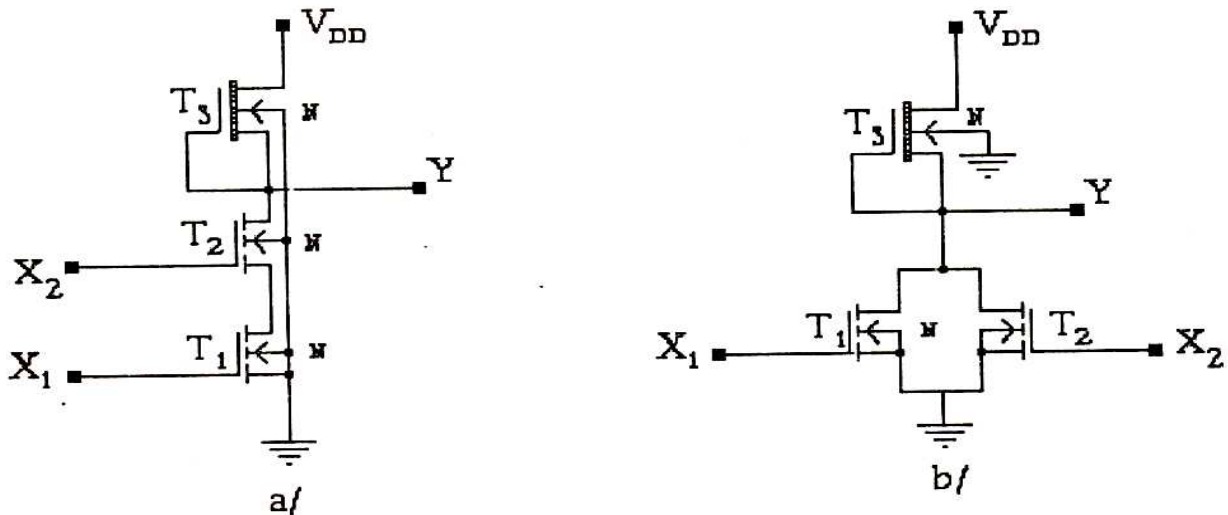
Hình 3.29: Sơ đồ nguyên lý các cửa NOT dùng NMOS

Hình 3.30a là cửa NAND họ NMOS. Mạch được tạo từ hai Transistor trường MOSFET kênh n cảm ứng T_1 và T_2 , T_3 là Transistor trường MOSFET kênh n được tạo sẵn. Hoạt động của mạch được giải thích như sau:

Khi $X_1 = X_2 = H$, cả hai lối vào X_1, X_2 đồng thời ở mức cao trong cả hai Transistor đều hình thành kênh dẫn T_1, T_2 , đều thông mạch lối ra Y ở mức thấp: $Y = L$.

Khi một trong hai lối vào ở thế thấp hoặc cả hai lối vào ở thế thấp, không có dòng qua T_1, T_2 , lối ra Y ở thế cao: $Y = H$. Như vậy trạng thái logic ở lối ra của mạch phụ thuộc vào các tổ hợp biến lối vào đúng như bản chân lý của hàm NAND.

Trên hình 3.30b trình bày sơ đồ nguyên lí mạch NOR họ NMOS, mạch được tạo từ hai Transistor trường MOSFET kênh n cảm ứng T_1 và T_2 được mắc song song với nhau, T_3 là Transistor trường MOSFET kênh n được tạo sẵn đóng vai trò trở tải của mạch. Khi cả hai lối vào X_1, X_2 của mạch đều ở mức logic thấp T_1, T_2 đều cấm lối ra $Y = H$. Lối ra $Y = L$ khi một trong hai lối vào ở mức thấp hoặc cả hai lối vào ở mức thấp hoạt động tuân theo hàm NOR.



Hình 3.30: Sơ đồ nguyên lí mạch NAND (a) và Mạch NOR (b) họ NMOS

3.6. HỌ CMOS (COMPLEMENTARY - METAL - OXYDE SEMICONDUCTOR).

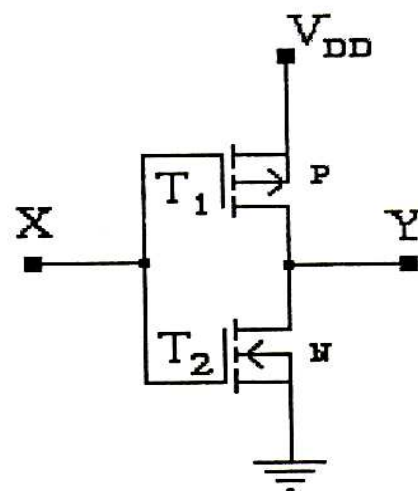
Họ vi mạch này sử dụng phối hợp 2 loại transistor trường MOSFET kênh n và MOSFET kênh p.

3.6.1. Mạch Đảo (Phủ định NOT) họ CMOS

+ Hàm logic: $y = \bar{x}$

+ Sơ đồ mạch như trên hình 3.31.

Mạch hoạt động như sau: Khi lối vào X ở mức điện áp cao (H), ở đây mức H bằng điện áp nguồn $V_{DD} = (3V \div 15V)$



Hình 3.31

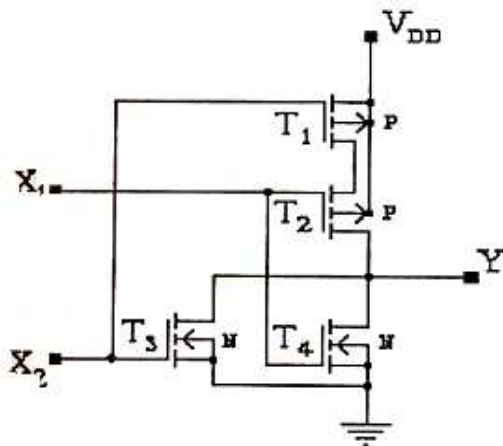
transistor trường kênh p T_1 cấm, transistor trường kênh n T_2 thông, lối ra Y ở mức L. Khi $X = L$ (mức $L = 0v$) T_2 cấm T_1 thông mạch $Y = H$.

Như vậy điện áp ra luôn là đảo của điện áp vào. hoạt động của mạch diễn ra đúng như phương trình logic của mạch NOT đã nêu ở trên

3.6.2. Mạch Không Hoặc (NOR)

Sơ đồ nguyên lý mạch NOR họ CMOS được trình bày trên hình 3.32.

Bảng chân lý là bảng 3.6.



Bảng 3.6: Bảng chân lý của mạch NOR

x_1	x_2	$y = \overline{x_1 + x_2}$
L	L	H
H	L	L
L	H	L
H	H	L

Hình 3.32: Mạch NOR họ CMOS

Giải thích hoạt động của mạch:

a/ Khi 2 lối vào $x_1 = x_2 = L$: T_1, T_2 thông và T_3, T_4 cấm dẫn đến $y = H$.

b/ Khi: $x_1 = H, x_2 = L$: Từ sơ đồ mạch ta thấy khi $x_1 = H \rightarrow T_2$ cấm, T_4 thông nên y nối đất $y = L$

$x_2 = L \rightarrow T_3$ cấm, T_1 thông nhưng T_2 cấm nên lối ra y vẫn được nối xuống đất qua T_4 làm cho $y = L$.

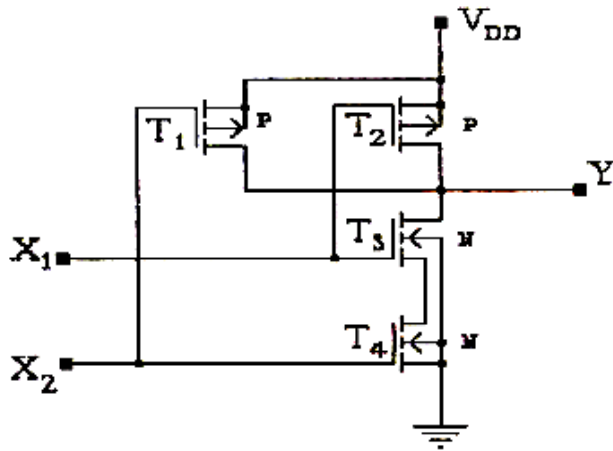
c/ Khi: $x_1 = L, x_2 = H \rightarrow T_4, T_1$ cấm; T_2, T_3 thông $\rightarrow y = L$.

d/ Khi $x_1 = x_2 = H$ cả T_1, T_2 cấm T_3, T_4 thông $y = L$.

Mạch hoạt động đúng như bảng chân lý 3.6.

3.6.3. Mạch Không Và (NAND)

Sơ đồ mạch NAND họ CMOS được trình bày trên hình 3.33. Bảng chân lý cho ở bảng 3.7.



Hình 3.33: Mạch NAND họ CMOS

b/ Khi $x_1 = H$; $x_2 = L$: T_2 cấm T_3 thông; T_1 thông T_4 cấm.

Lối ra y được nối với nguồn V_{DD} : $y = H$.

c/ Khi $x_2 = H$; $x_1 = L$: T_1 cấm, T_2 thông T_3 cấm, T_4 thông $\rightarrow y = H$.

d/ Khi $x_1 = x_2 = H$: T_1, T_2 cấm, T_3, T_4 thông $\rightarrow y = L$.

* Ưu điểm của họ CMOS: Có điện trở lối vào rất lớn.

- Vì mạch chỉ tiêu thụ dòng điện khi chuyển mạch (lúc chuyển mạch mới có dòng qua). Vì vậy công suất tiêu thụ cực kỳ nhỏ (cỡ nW).

- CMOS có hệ số mắc tải ở lối ra (FAN OUT) rất lớn. (Số cổng logic cùng loại có thể mắc vào lối ra của nó là 50, gấp 10 ÷ 20 lần họ TTL).

* Nhược điểm:

- Tốc độ chuyển mạch thấp nên tần số làm việc không cao và phụ thuộc vào nguồn nuôi. Khi $V_{DD} = 5\text{ V} \rightarrow f = 1\text{ MHz}$, $V_{DD} = 10\text{ V} \rightarrow f = 1,6\text{ MHz}$, $V_{DD} = 15\text{ V} \rightarrow f = 2\text{ MHz}$.

3.6. 4. Bảo vệ CMOS khỏi bị hỏng

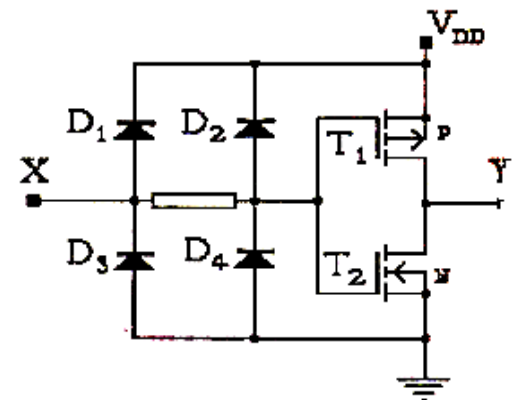
- Vì điều khiển bằng điện trường, lớp điện môi cực cửa rất mỏng nên lối vào dễ bị hỏng khi điện áp đặt vào lớn. Để bảo vệ lối vào IC họ CMOS người ta mắc thêm các mạch bảo vệ cực cửa khỏi bị quá áp như hình 3.34.

Bảng 3.7: Bảng chân lý của mạch NAND

X_1	X_2	$y = \overline{x_1 \cdot x_2}$
L	L	H
H	L	H
L	H	H
H	H	L

Giải thích hoạt động của mạch:

a/ Khi $x_1 = x_2 = L$: T_1, T_2 thông, T_3, T_4 cấm $\rightarrow y = H$.



Hình 3.34: Mạch bảo vệ lối vào CMOS

Mạch bảo vệ gồm 4 điôt.

- Nếu xung dương vào x lớn hơn V_{DD} thì D_1, D_2 dẫn thông mạch lối vào x được nối lên nguồn V_{DD} .

- Nếu xung vào x âm so với đất thì D_3, D_4 thông mạch, lối vào x được nối xuống đất. Nhờ các điôt bảo vệ mà điện áp đưa vào cực cửa các transistor T_1, T_2 không vượt quá mức logic V_H và V_L của họ logic CMOS. Như vậy, các transistor trường MOS đã được bảo vệ khi các xung lối vào quá lớn.

Mặc dù đã có mạch bảo vệ lối vào nhưng khi dùng hoặc bảo quản vi mạch ta vẫn phải có biện pháp phòng ngừa các điện áp tĩnh điện có điện áp cao hoặc nguồn cao áp xoay chiều cảm ứng thâm nhập vào phá hỏng cửa cách điện của vi mạch CMOS. Khi hàn các panel trên đó có vi mạch thuộc họ CMOS mở hàn, tay người hàn phải được nối đất thật tốt để tránh những hư hỏng đáng tiếc không ngờ tới xảy ra trong khi hàn mạch.

Khi dùng nguồn nuôi 5V ta có thể ghép CMOS với TTL và DTL. Họ CMOS được nuôi bằng nguồn $V_{DD} = +3V \div +15V$, có kí hiệu CD40 xx hoặc 40xxx các chữ ở đầu dùng để chỉ hãng sản xuất; hai hoặc 3 chữ số đứng sau số 40 để chỉ chức năng của mạch, ví dụ:

CD4001: CD là ký hiệu của hãng RCA của Hoa Kỳ; 40 chỉ rõ là vi mạch thuộc họ CMOS; hai số sau cùng 01 chỉ chức năng của mạch là mạch có 4 cửa NOR hai lối vào.

4011: 4 cửa NAND hai lối vào họ CMOS, 4049: 6 cửa NOT, 4050: 6 cửa đệm.

Ký hiệu vi mạch CMOS của hãng Motorola: MC 14001 là mạch 4 mạch NOR hai lối vào họ CMOS ta thấy 4 số sau trong ký hiệu của hãng Motorola đều giống ký hiệu của các hãng khác.

Các vi mạch CMOS có ký hiệu giống như ở họ TTL có chức năng và bố trí chân giống như vi mạch TTL có thể thay thế cho vi mạch TTL 74L xx, 74LS xx có ký hiệu đặc trưng bằng chữ C để chỉ họ CMOS, ví dụ:

74C00: NAND giống như 7400 nhưng là CMOS có thể dùng thay thế cho 74L00.

74C02: NOR - CMOS có thể thay thế cho 74L02.

74HC00: NAND họ CMOS có tốc độ cao có thể thay cho 74LS00.

Khi dùng vi mạch họ CMOS ghép nối với họ TTL thì CMOS phải nuôi bằng nguồn +5V cùng chung nguồn nuôi với điện áp nuôi vi mạch họ TTL.

3.7. HỘ ECL (EMITTER COUPLED LOGIC):

Hộ logic ECL là hộ IC logic có tốc độ nhanh nhất trong các hộ IC logic hiện có. Sở dĩ như vậy là do hộ ECL có các đặc điểm sau đây:

- Các Transistor làm việc ở chế độ không bão hoà (chế độ tuyến tính), do đó có thể giảm nhỏ thời gian trễ rất nhiều. Thời gian trễ liên quan đến quá trình giải phóng các điện tích dư thừa trong lớp tiếp giáp của Transistor khi hoạt động ở chế độ bão hoà (chế độ khoá).

- Do sơ đồ dùng khuếch đại vi sai có cấu trúc đối xứng nên khi sơ đồ lật trạng thái (chuyển mức logic từ 0 đến 1 hoặc ngược lại) thì không làm thay đổi dòng tiêu thụ của toàn mạch.

- Sơ đồ có mức tạp âm nhỏ cho phép giảm điện áp tương ứng giữa hai mức logic nên giảm thời gian quá độ gây ra bởi các điện dung ký sinh.

3.7.1. Cửa OR/NOR 2 lối vào hộ ECL

Trên hình 3.35 trình bày sơ đồ nguyên lý của cửa NOR, OR 2 lối vào.

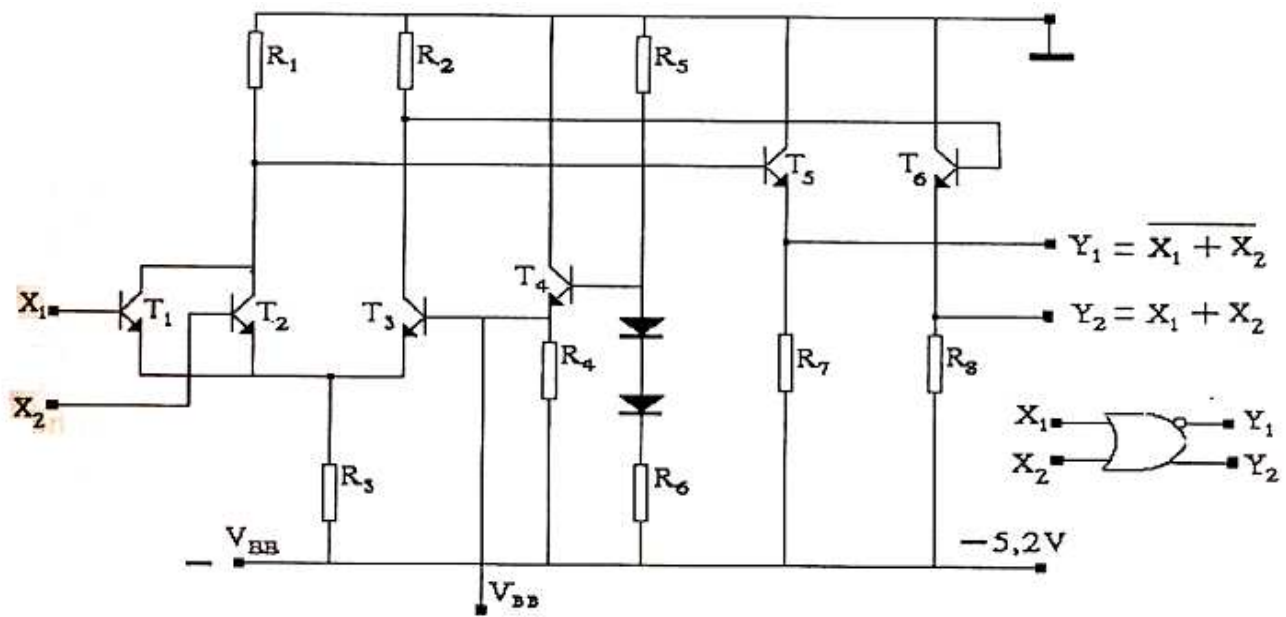
Từ sơ đồ nguyên lý của mạch chúng ta thấy mạch gồm 3 tầng:

Tầng 1: tầng khuếch đại vi sai ở lối vào có các transistor T_2, T_3 hai điện trở gánh là R_1, R_2 , điện trở R_3 mắc ở mạch emitter của các transistor này đóng vai trò nguồn dòng không đổi. Trong các giáo trình Kỹ thuật điện tử chúng ta đã khảo sát kỹ về khuếch đại vi sai, ở đây ta chúng ta chỉ nhắc lại một đặc điểm của khuếch đại vi sai là nó có khả năng triệt nhiễu cùng pha, chính vì thế chênh lệch mức điện áp giữa hai mức logic chỉ có 0,8V. Mức logic 1: $V_H = -0,75V$; mức logic 0: $V_L = -1,6V$

Tầng 2 gồm T_4 các Diode và các điện trở R_4, R_5, R_6 là tầng ổn áp, chức năng của nó là tạo điện áp phân cực Base V_{BB} cho các transistor của tầng khuếch đại vi sai ở lối vào.

Vì 2 mức logic chênh lệch ít nên thế Base của các transistor của tầng khuếch đại vi sai ở lối vào phải để ở giá trị trung bình nằm giữa hai mức điện áp V_L, V_H :

$$V_{BB} = \frac{-1,6v + (-0,75v)}{2} = -1,175v$$



Hình 3.35: Cửa OR/NOR 2 lối vào họ ECL

Tầng 3 là tầng khuếch đại công suất, ở lối ra nó được mắc theo sơ đồ coléctơ chung hay còn gọi là sơ đồ lặp Emitter, điện áp lấy ra ở Emitter giống như điện áp đưa vào Base, tuy không được khuếch đại điện áp nhưng tầng này có hệ số khuếch đại dòng rất lớn, nó có điện trở lối vào rất lớn, điện trở lối ra nhỏ, dòng vào nhỏ, dòng ra rất lớn nên hệ số mắc tải của nó lớn nhất trong các họ logic dùng transistor lưỡng cực (Fanout = 25). Phân tích sơ đồ hình 3.35 ta thấy: các transistor hoạt động ở chế độ tuyến tính, không có chế độ cấm thông bão hoà cho nên luôn luôn có dòng qua transistor, chỉ là lớn hoặc nhỏ tùy theo mức điện áp ở lối vào.

Khuếch đại vi sai có cấu tạo đối xứng: Emitter là nguồn dòng không đổi. Nếu 2 lối vào điện áp cùng tăng hoặc cùng giảm thì dòng Colectơ 2 bên không đổi, tức là nó không khuếch đại tín hiệu cùng pha. Nó chỉ khuếch đại tín hiệu vi sai ở 2 lối vào. Trong sơ đồ trên Base của T_3 được duy trì ở mức điện áp V_{BB} không đổi.

a/ Nếu cả hai lối vào ở mức điện áp thấp: $x_1 = x_2 = L$, dòng điện qua T_1 , T_2 giảm thế trên colectơ của chúng tăng; dòng qua T_3 tăng thế V_{C3} ở mức thấp, trạng thái lối ra của mạch: $y_1 = H$, $y_2 = L$.

b/ Nếu một trong hai lối vào có mức điện áp cao:

x_1 hoặc $x_2 = V_H > V_{BB}$, I_{C1} hoặc I_{C2} tăng điện áp trên collector của các transistor này giảm, I_{C3} giảm, V_{C3} ở mức cao $y_1 = L$, $y_2 = H$.

c/ Khi $x_1 = x_2 = H$: thì I_{C1} , I_{C2} tăng nên V_{C1} , V_{C2} giảm, $V_{C1} = V_{C2} = L$. Vì nguồn dòng qua điện trở R_3 mắc ở mạch emitter của các transistor T_1 , T_2 , T_3 là không đổi nên khi: I_{C1} , I_{C2} tăng thì I_{C3} giảm, V_{C3} tăng Lối ra $V_{C3} = H$.

Trạng thái của mạch ở lối ra: $y_1 = L$; $y_2 = H$.

Qua sự phân tích hoạt động của mạch nêu ở trên ta thấy mạch hoạt động đúng như bảng chân lý cho trên bảng 3.8. Kí hiệu logic của mạch như hình 3.35.

Bảng 3.8: Bảng chân lý của mạch OR/NOR

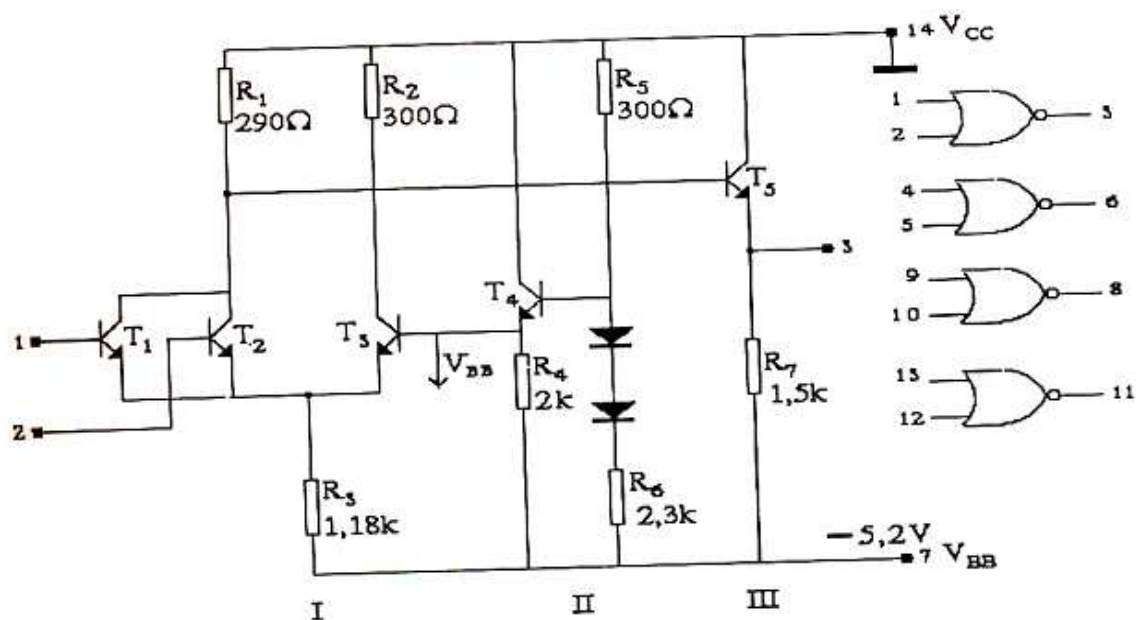
x_1	x_2	$y_1 = \overline{x_1 + x_2}$	$y_2 = x_1 + x_2$
L	L	H	L
H	L	L	H
L	H	L	H
H	H	L	H

3.7.2. Cửa NOR

Trên hình 3.36 trình bày sơ đồ nguyên lý vi mạch MC1010, MC1210. Trong vi mạch có 4 cửa NOR 2 lối vào, hình vẽ trên là sơ đồ nguyên lý vẽ cho một cửa, ba cửa còn lại có sơ đồ cũng tương tự. Mạch tạo điện áp phân cực V_{BB} cho tầng khuếch đại vi sai ở lối vào được dùng chung cho cả 4 cửa NOR.

Cấu trúc của mạch cũng gồm 3 tầng: khuếch đại vi sai lối vào, tầng lặp Emitter ở lối ra và tầng tạo điện áp phân cực V_{BB} tương tự như mạch trên hình 3.35.

Vi mạch MC1010, MC1210 thuộc seri MECL II của hãng Motorola mạch có thời gian trễ $T_D = 4,5$ nS, hệ số tải lối ra Fan out = 25.



Hình 3.36: Sơ đồ nguyên lý mạch MC 1010, MC 1020

Bài tập

1/ Dùng chương trình phần mềm CircuitMaker thiết kế mô phỏng các vi mạch logic NOT, NOR 2 lối vào, AND 2 lối vào thuộc họ RTL.

Tiến hành thí nghiệm mô phỏng đo kiểm tra điện áp ở lối vào và lối ra trên sơ đồ điện tử của các mạch thiết kế để nghiệm lại bảng chân lý của vi mạch.

2/ Dùng chương trình phần mềm CircuitMaker thiết kế mô phỏng các vi mạch logic NOT, NAND 2 lối vào thuộc họ DTL.

Tiến hành thí nghiệm mô phỏng đo kiểm tra điện áp ở lối vào và lối ra trên sơ đồ điện tử của các mạch thiết kế để nghiệm lại bảng chân lý của vi mạch.

3/ Dùng chương trình phần mềm CircuitMaker thiết kế mô phỏng các vi mạch logic NOT, NOR 2 lối vào, NAND 2 lối vào thuộc họ TTL.

Tiến hành thí nghiệm mô phỏng đo kiểm tra điện áp ở lối vào và lối ra trên sơ đồ điện tử của các mạch thiết kế để nghiệm lại bảng chân lý của vi mạch.

4/ Dùng chương trình phần mềm CircuitMaker thiết kế mô phỏng các vi mạch logic NOT, NOR 2 lối vào, NAND 2 lối vào thuộc họ CMOS.

Tiến hành thí nghiệm mô phỏng đo kiểm tra điện áp ở lối vào và lối ra trên sơ đồ điện tử của các mạch thiết kế để nghiệm lại bảng chân lý của vi mạch.

Chương IV

MẠCH LOGIC TỔ HỢP

Căn cứ vào đặc điểm và chức năng logic, ta phân loại các mạch logic tổ hợp thành hai loại chính:

1. Mạch tổ hợp (Combinational Circuits) được tạo thành từ các phần tử logic cơ bản. Trạng thái lối ra của mạch tại một thời điểm bất kỳ chỉ phụ thuộc vào tổ hợp các trạng thái logic ở lối vào tại thời điểm đó, nó không phụ thuộc vào trạng thái lối ra ở thời điểm trước đó (trạng thái quá khứ). Các loại mạch sau đây thuộc loại này: các bộ số học, hợp kênh, phân kênh, biến đổi mã, giải mã,...

2. Mạch dãy (Sequential Circuits) cũng được tạo thành từ các mạch logic cơ bản nhưng trạng thái lối ra của mạch được xác lập không chỉ phụ thuộc vào tổ hợp các trạng thái lối vào ở thời điểm hiện tại mà còn phụ thuộc vào trạng thái logic ở lối ra của nó tại thời điểm trước đó. Mạch dãy gồm có: các phần tử nhớ, các trigger các bộ đếm, ghi dịch..., các mạch này sẽ được trình bày trong chương sau.

4.1. PHƯƠNG PHÁP THIẾT KẾ CÁC MẠCH LOGIC TỔ HỢP

Với một mạch logic tổ hợp bất kỳ, nếu cho trước chức năng ta đều có thể thiết kế và thực hiện được. Các bước tiến hành như sau:

1. Từ yêu cầu chức năng ta lập bảng chân lý của hàm.
2. Từ bảng chân lý suy ra phương trình logic.
3. Tối giản hoá hàm logic.
4. Từ hàm logic thiết kế mạch thực hiện bằng các phần tử cơ bản.

Ví dụ: Xây dựng mạch so sánh 2 số A và B có chức năng như sau:

Nếu $A = B$ thì lối ra $C = 1$;

$A \neq B$ thì lối ra $C = 0$

+ Ta lập bảng chân lý như bảng 4.1.

Bảng 4.1: Bảng chân lý của mạch so sánh 2 số A và B

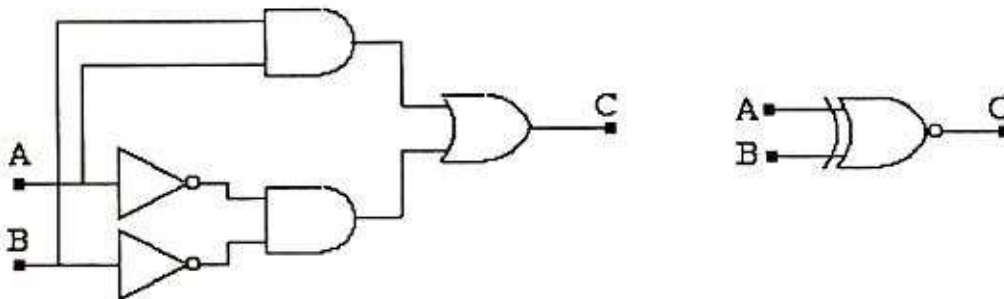
A	B	C
0	0	1
1	0	0
0	1	0
1	1	1

+ Tìm phương trình logic:

Ở đây ta dùng phương pháp giải tích: Lối ra C là tổng các tích các lối vào có giá trị bằng 1 (lấy tổng các minterm ứng với $C = 1$) ta được phương trình logic :

$$C = \bar{A}\bar{B} + AB \quad (4.1)$$

+ Phương trình (4.1) đã được viết ở dạng tối giản rồi ta không cần phải rút gọn nữa. Từ phương trình logic 4.1 có thể suy ra sơ đồ logic của bộ so sánh được trình bày trên hình 4.1.



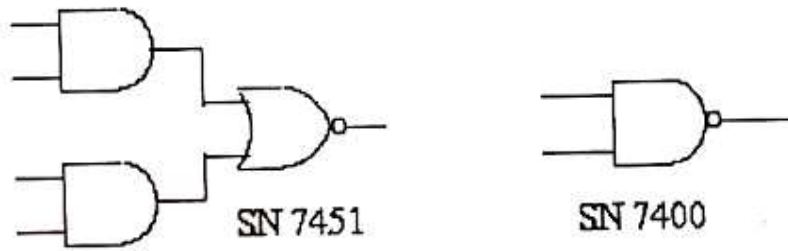
Hình 4.1: Mạch so sánh 2 số A và B

Đối chiếu bảng chân lý 4.1 với bảng chân lý 2.12 ta thấy hai bảng chân lý này hoàn toàn như nhau. Phương trình 4.1 cũng là phương trình mô tả hàm Không Hoặc tuyệt đối XNOR, áp dụng đại số Boole ta có thể chứng minh được:

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

Các bài tập thường cho dưới dạng: từ các mạch logic cơ bản đã cho, hãy xây dựng 1 sơ đồ chức năng nào đó. Như vậy, hàm logic cần tìm phải biến đổi sao cho sử dụng được phần tử đã cho.

Ví dụ: Cho mạch SN 7451 và mạch NAND 7400 hình 4.2.



Hình 4.2: Sơ đồ logic và kí hiệu logic của vi mạch SN 7451 và cửa NAND 7400

Hãy dùng 2 loại vi mạch này thiết kế bộ bán tổng thực hiện phép cộng 2 số A và B hoạt động theo bảng chân lý 4.2

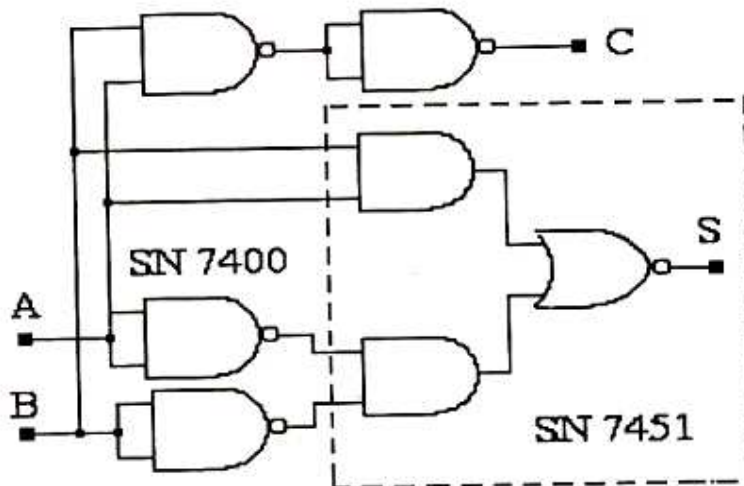
Bảng 4.2: Bảng chân lý bộ bán tổng

A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Từ bảng chân lý ta xác định được hàm logic của bộ bán tổng:

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

Nhớ $C = AB$. Từ đây vận dụng các định luật của đại số logic biến đổi S và C thành các phương trình tương ứng với sơ đồ hình 4.3.



Hình 4.3: Sơ đồ mạch cộng 2 số A và B

4.2. MẠCH TÍNH TOÁN SỐ HỌC

4.2.1. Bộ tổng bán phần (Half Adder: HA)

Mạch bán tổng có 2 lối vào A và B, có 2 lối ra là tổng S (Sum) và lối ra nhớ chuyển sang hàng sau C_0 (Carry - out). Hoạt động của bộ bán tổng được diễn tả bởi bảng chân lý 4.3.

Bảng 4.3: Bảng chân lý bộ bán tổng

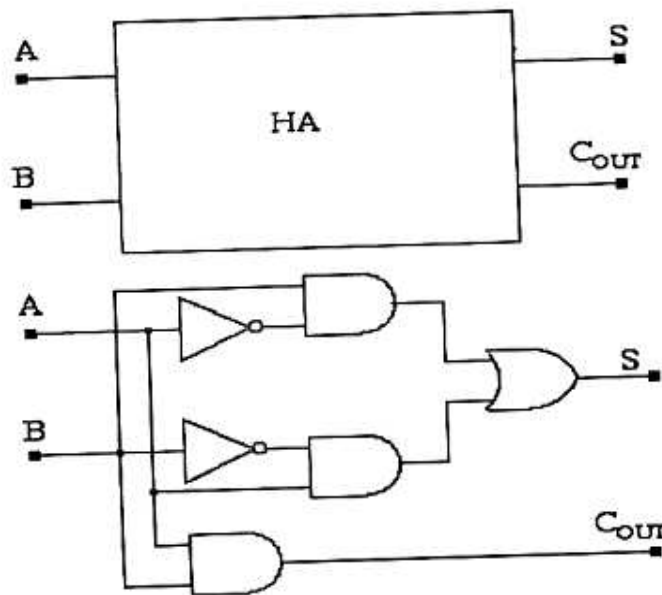
A	B	S	C_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Với số nhị phân 1 bit ta có:

- Hàm logic HA:

$$\begin{cases} S = A \oplus B \\ C_0 = A.B \end{cases}$$

Sơ đồ khối và sơ đồ mạch logic tổng bán phần được nêu trên hình 4.4.



Hình 4.4: Sơ đồ khối và sơ đồ logic HA

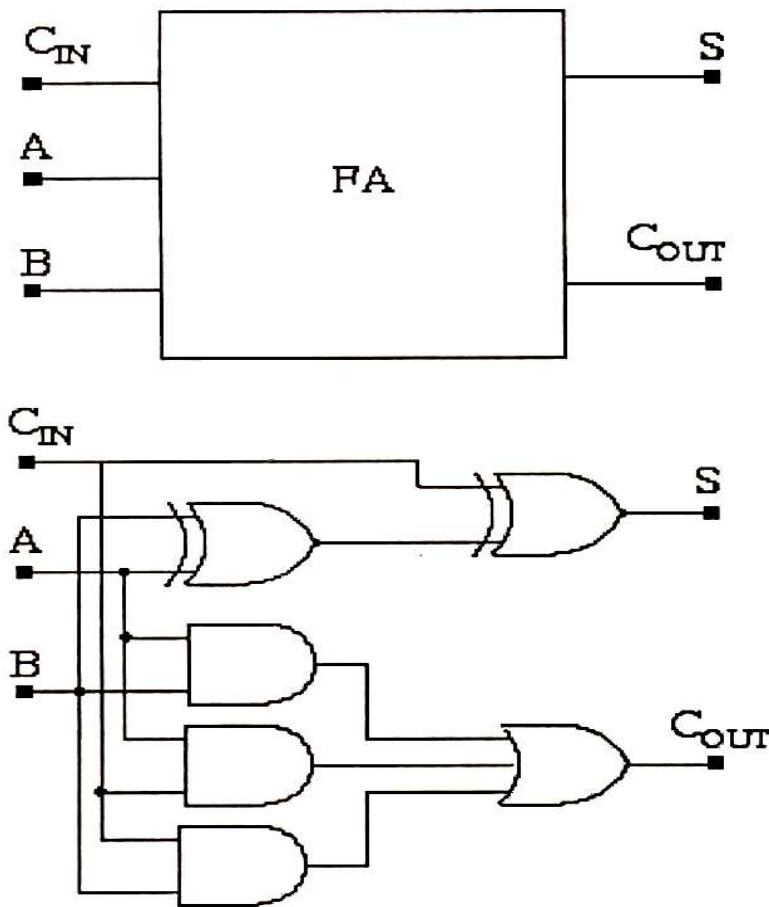
Ta gọi là bộ bán tổng vì riêng nó chưa thực hiện được phép cộng. Ta phải dùng 2 bộ bán tổng mới tạo ra một mạch tính cộng.

4.2.2. Bộ tổng toàn phần (Full Adder: FA)

Bộ tổng toàn phần có ba lối vào A, B, C_i (Carry - in), hai lối ra là tổng S và lối ra nhớ chuyển sang hàng sau C_o (Carry - out).

- Hàm logic của FA:

$$\begin{cases} S = A \oplus B \oplus C_i \\ C_o = AB + AC_i + BC_i \end{cases}$$



Hình 4.5: Sơ đồ mạch Tổng toàn phần

- Sơ đồ khối và mạch logic: cho trên hình 4.5.
- Bảng chân lý: trên bảng 4.4

Bảng 4.4: Bảng chân lý bộ FA

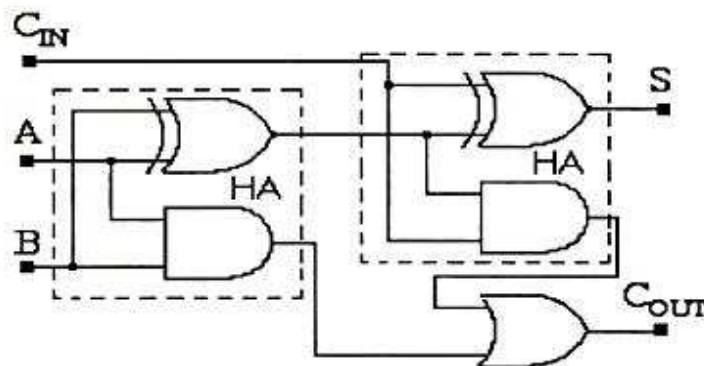
A B C _i	S C _o
0 0 0	0 0
0 0 1	1 0
0 1 0	1 0
0 1 1	0 1
1 0 0	1 0
1 0 1	0 1
1 1 0	0 1
1 1 1	1 1

Từ bảng chân lý ta rút ra phương trình logic:

$$\begin{aligned} C_o &= \overline{C_i}(A.B) + C_i(A.\overline{B} + \overline{A}.B + A.B) \\ &= (\overline{C_i} + C_i)(A.B) + C_i(A.\overline{B} + \overline{A}.B) \\ &= A.B + C_i(A \oplus B) \end{aligned}$$

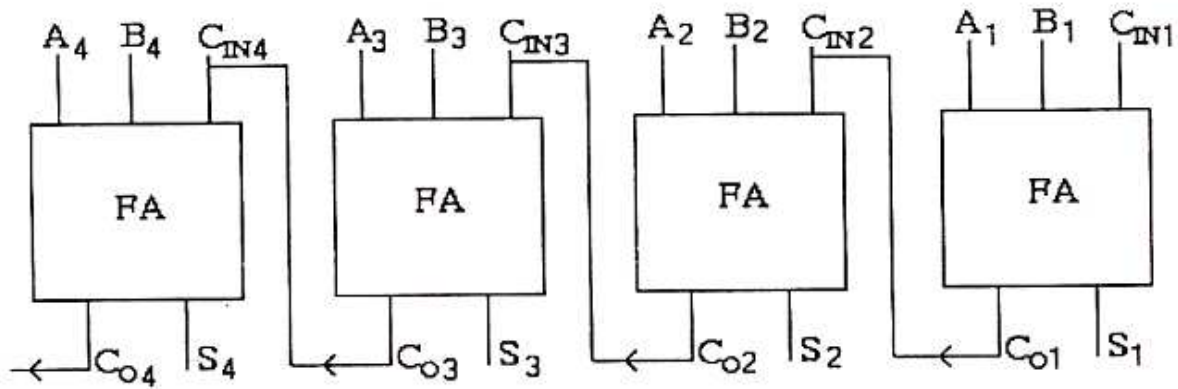
$$\begin{aligned} S &= \overline{C_i}(A.\overline{B} + \overline{A}.B + C_i(\overline{A}\overline{B} + AB)) \\ &= \overline{C_i}(A\overline{B} + \overline{A}B) + C_i(\overline{A}\overline{B} + AB) \\ &= A \oplus B \oplus C_i \end{aligned}$$

Ta có thể xây dựng mạch tổng toàn phần FA từ hai bộ bán tổng HA. Sơ đồ logic của mạch được vẽ trên hình 4.6.



Hình 4.6: Sơ đồ mạch FA

Ghép nối bộ tổng toàn phần thành mạch cộng nhị phân 4 bit.



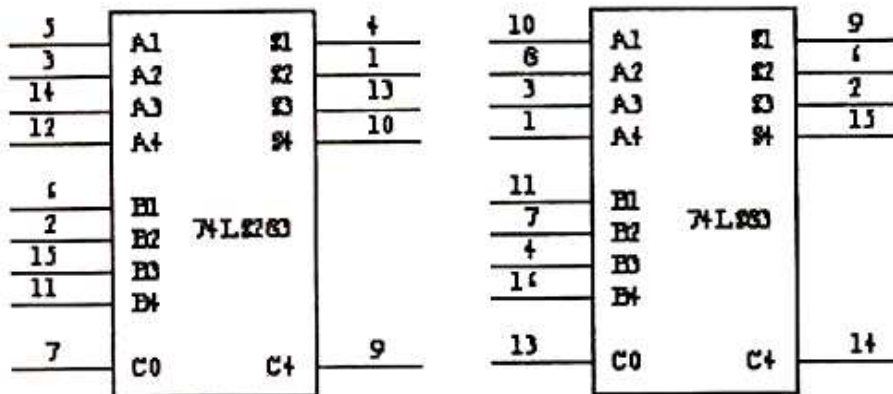
Hình 4.7: Mạch cộng hai số nhị phân 4 bit

Muốn xây dựng bộ cộng nhị phân 4 bit ta thực hiện mắc song song 4 bộ tổng toàn phần một bit, chỉ cần chú ý rằng lối ra nhớ của bộ tổng toàn phần của bit trước phải được nối vào lối vào nhớ của bộ tổng toàn phần ứng với bit tiếp theo. Sơ đồ mắc bộ tổng của hai số nhị phân 4 bit dùng 4 FA được nêu trên hình 4.7.

Sơ đồ trên thực hiện tổng 2 số nhị phân 4 bit:

$$\begin{array}{r} A_4 A_3 A_2 A_1 \\ + B_4 B_3 B_2 B_1 \\ \hline \end{array}$$

Kết quả: $C_{O4} S_4 S_3 S_2 S_1$



Hình 4.8: Kí hiệu logic của IC cộng nhị phân 4 bit 74LS283, 74LS83

Trong thực tế ta thường gặp các vi mạch 7483, 74LS283 là vi mạch 16 chân gồm 4 bộ FA được mắc thành mạch cộng nhị phân hai số nhị phân 4 bit. Hình 4.8 là kí hiệu logic của các mạch này.

Bài tập thực hành:

1/ Trên một panel thí nghiệm có 2 loại IC: 74LS283 và 7486. Vì mạch 74LS283 có 4 mạch hoặc tuyệt đối hai lối vào, hãy dùng mạch này tạo thành bộ đảo từ nhị phân 4 bit và kết hợp với mạch cộng nhị phân 4 bit 74LS283 tạo thành bộ tính số học với cả phép cộng và trừ nhị phân 4 bit. Kiểm tra hoạt động của mạch 74LS283 hình 4.8 trên panel thí nghiệm, hoặc dùng phần mềm thiết kế mô phỏng Circuit Maker.

2/ Dùng 7486 (4 mạch hoặc tuyệt đối 2 lối vào) và mạch 7400 (4 mạch NAND 2 lối vào) tạo bộ tổng toàn phần.

3/ Dùng hai mạch 7486 và hai mạch cộng nhị phân 4 bit 74LS283 mắc thành bộ cộng trừ 8 bit. Hãy trình bày sơ đồ mắc mạch. Thực hiện các phép tính cộng, tính trừ hai số nhị phân 8 bit (hai số nhị phân 8 bit A và B chọn trong phạm vi từ 1÷255).

4.2.3. Mạch hiệu bán phần (Half Subtractor: HS)

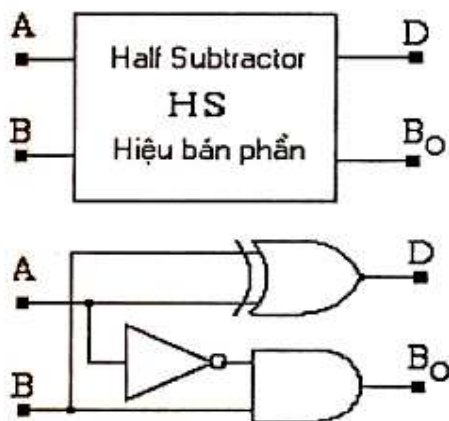
Mạch hiệu bán phần là một mạch logic thực hiện phép trừ 2 số nhị phân 1 bit. Đầu vào là A, B, lối ra là hiệu D (Difference) và lối ra mượn B_0 (Borrow out).

- Hàm logic HS:

$$\begin{cases} D = A \oplus B \\ B_0 = \bar{A}.B \end{cases}$$

- Sơ đồ khối và mạch logic: cho trên hình 4.9.

- Bảng chân lý: trên bảng 4.5.



Bảng 4.5: Bảng chân lý bộ HS

Đầu vào		Đầu ra	
A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Hình 4.9: Sơ đồ khối và mạch logic bộ HS

Số với mạch HA, mạch HS chỉ khác ở chỗ có thêm cửa đảo ở lối vào A của cửa V_a.

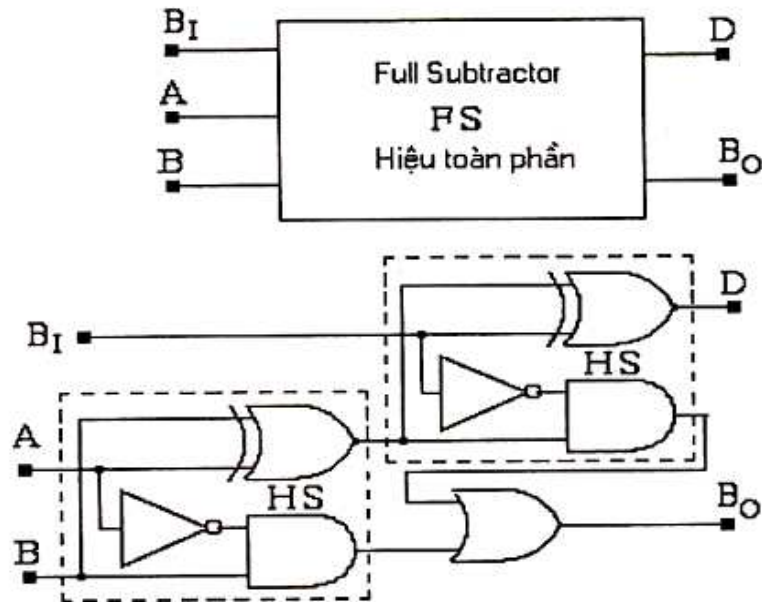
4.2.4. Mạch Hiệu toàn phần (Full Subtractor: FS)

Mạch Hiệu toàn phần có 3 lối vào A, B, B_i (Borrow in: lối vào mượn) Hai lối ra là hiệu D và lối ra mượn B_o.

- Bảng chân lý: bảng 4.6.

Bảng 4.6: Bảng chân lý mạch FS

Đầu vào			Đầu ra	
A	B	B _i	D	B _o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



Hình 4.9: Sơ đồ khối và mạch logic của FS

- Hàm logic FS:

$$\begin{cases} D = A \oplus B \oplus B_i \\ B_0 = \bar{A}B + \bar{A}B_i + BB_i \end{cases}$$

- Sơ đồ khối và sơ đồ logic của mạch cho trên hình 4.9.

Ta có thể xây dựng mạch FS từ 2 mạch HS và thêm 1 cửa hoặc ở lối ra B_0 như sơ đồ logic của mạch trên hình 4.9.

4.2.5. Mạch tổng hoặc hiệu hai số nhị phân n bit

4.2.5.1. Mạch tổng hai số nhị phân n bit

Giả sử có 2 số nhị phân 4 bit $A = A_4 A_3 A_2 A_1$
 $B = B_4 B_3 B_2 B_1$

kết nối với nhau thì cần dùng 4 mạch FA mắc như hình 4.7. Lưu ý rằng FA thứ nhất có $C_i = 0$ (mức thấp) làm nhiệm vụ như HA. Mạch tổng mắc theo kiểu này gọi là mạch tổng song song n bit. Các lối ra C_0 của FA được nối với C_i của FA tiếp sau. C_0 của FA cuối cùng là C_0 ở lối ra.

4.2.5.2. Mạch hiệu hai số nhị phân n bit

Ta có thể dùng các mạch FA thay cho các mạch FS. Hàm logic của FS:

$$\begin{cases} D = A \oplus B \oplus B_i & (a) \\ B_0 = \bar{A}B + \bar{A}B_i + BB_i & (b) \end{cases}$$

Biến đổi (b) :

$$\bar{B}_0 = \overline{\bar{A}B \cdot \bar{A}B_i \cdot BB_i} = (A + \bar{B})(A + \bar{B}_i)(\bar{B} + \bar{B}_i)$$

$$\bar{B}_0 = (A + \bar{B})[\bar{B}_i + (A\bar{B})] = A\bar{B}_i + \bar{B}\bar{B}_i + A\bar{B}$$

$$\bar{B}_0 = A\bar{B} + A\bar{B}_i + \bar{B}\bar{B}_i$$

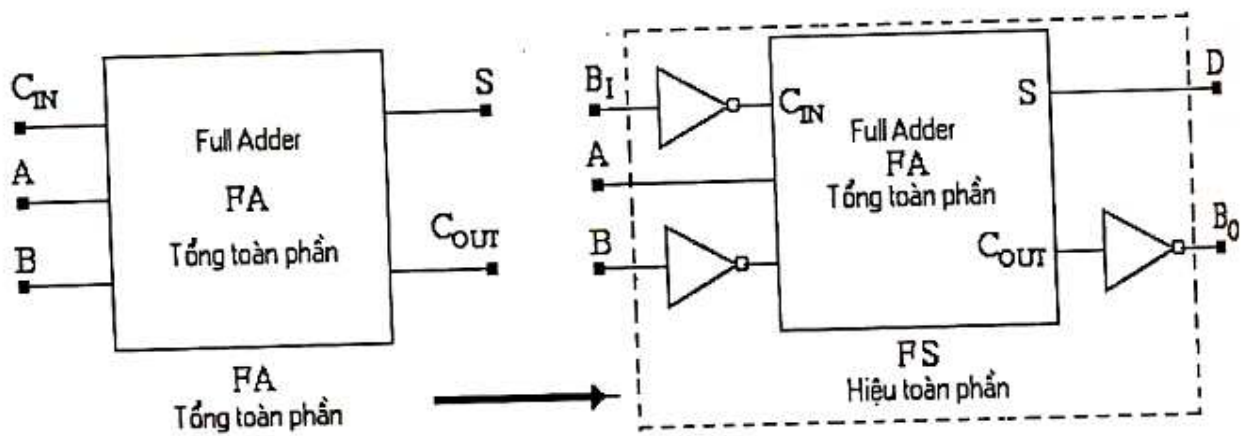
Còn (a) có giá trị tương tự :

$$D = A \oplus B \oplus B_i = A \oplus \bar{B} \oplus \bar{B}_i$$

Hàm logic viết cho FS sẽ là:

$$\begin{cases} D = A \oplus \bar{B} \oplus \bar{B}_i \\ \bar{B}_0 = A\bar{B} + A\bar{B}_i + \bar{B}\bar{B}_i \end{cases}$$

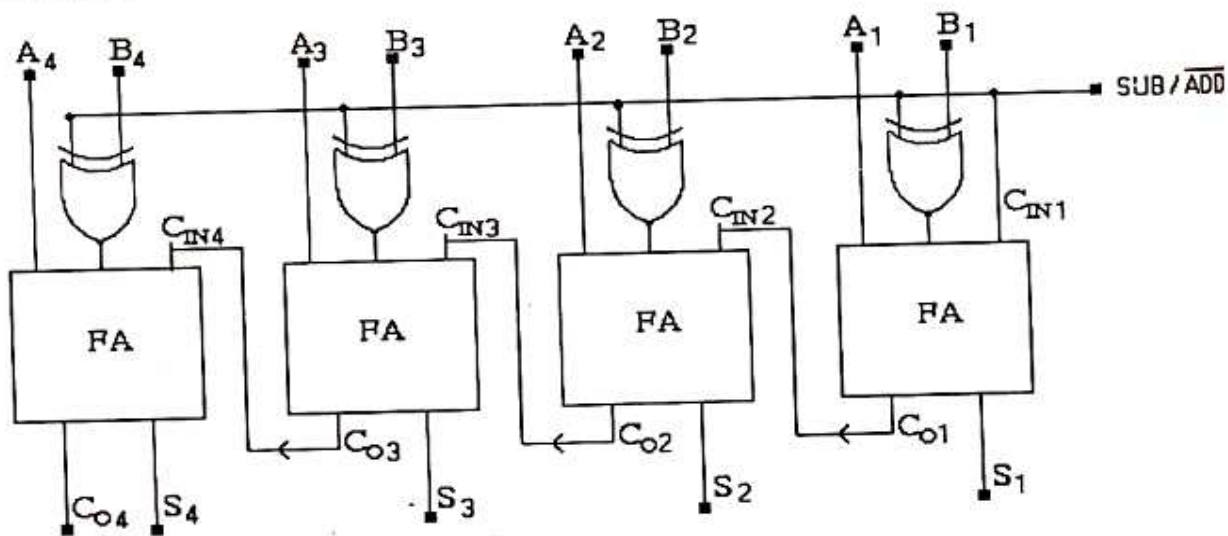
So sánh với hàm logic của FA ta thấy có thể xây dựng mạch FS từ mạch FA dễ dàng: Muốn chuyển FA thành FS ta chỉ cần mắc thêm các cửa đảo vào các đầu B, B_1 và B_0 như hình 4.11.



Hình 4.11: Cách chuyển mạch FA thành mạch FS

4.2.5.3. Mạch Tổng/Hiệu hai số nhị phân 4 bit

Sơ đồ khối của mạch tổng song song và hiệu song song hai số nhị phân n bit gần như nhau. Muốn dùng các mạch FA để thực hiện cả hai phép tính tổng hiệu ta cần thêm đầu điều khiển $\overline{\text{SUB/ADD}}$ như sơ đồ tổng hiệu 4 bit trên hình 4.12.



Hình 4.12: Sơ đồ mạch tổng/hiệu hai số nhị phân 4 bit

Khi $\overline{\text{SUB/ADD}} = 0$ (phép tổng): Các số liệu $B_1...B_4$ qua các cửa XOR không đổi và được đưa vào FA để làm phép cộng hai số A và B kết quả $S = C_0 S_4 S_3 S_2 S_1$.

Khi $\overline{\text{SUB/ADD}} = 1$ (phép hiệu): Các số liệu $B_1...B_4$ sẽ bị đảo khi đi qua các cửa XOR tức là $B_4 B_3 B_2 B_1 \rightarrow \overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1}$.

Đầu C_i được nối với $\overline{\text{SUB/ADD}} = 1$ tức là $C_i = 1$. Như vậy số bù một $\overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1}$ được cộng với $C_i = 1$ trở thành số bù hai, nghĩa là mạch thực hiện phép tính: $A + (-B)$. Vậy kết quả của phép trừ là: $D = B_0 D_4 D_3 D_2 D_1$.

Trong kết quả B_0 là lối ra mượn ta không cần quan tâm (giống như số "1" biểu diễn phép trừ). Kết quả phép trừ là $D = D_4 D_3 D_2 D_1$.

Trong thực tế ta có thể dùng vi mạch cộng nhị phân 4 bit 74LS283 hoặc 74LS83 ghép với vi mạch 74LS86 (có 4 cửa XOR) sẽ được 1 bộ tổng/hiệu 4 bit như sơ đồ hình 4.12.

4.2.6. Mạch so sánh

4.2.6.1. Mạch so sánh bằng nhau 1 bit

Có 2 số A và B là 2 số nhị phân 1 bit

$$\text{Nếu } \begin{cases} A = B \rightarrow \text{lối ra } C=1 \\ A \neq B \rightarrow \text{lối ra } C=0 \end{cases}$$

ta lập được bảng chân lý 4.7.

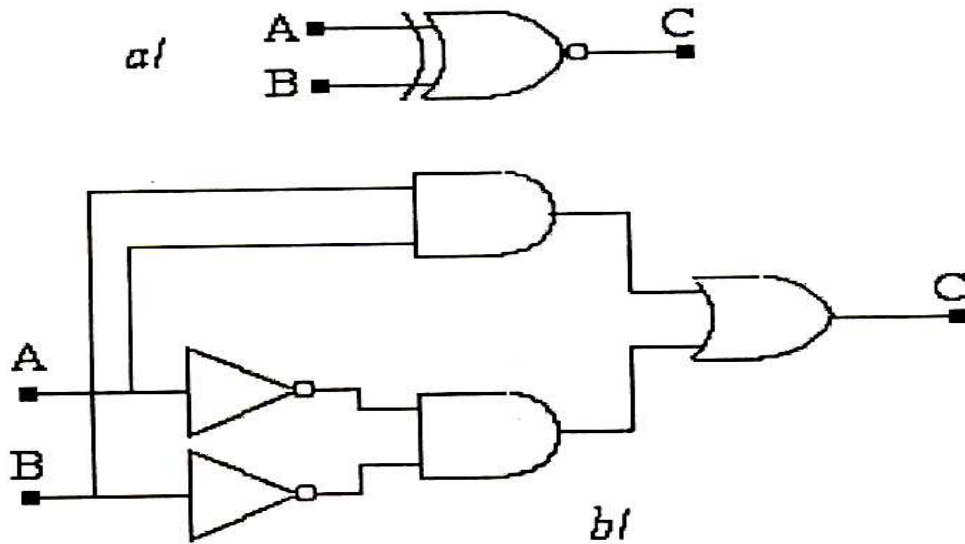
Bảng 4.7: Bảng chân lý của XNOR

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

Từ bảng chân lý ta suy ra hàm logic: $C = \overline{A} \overline{B} + AB$, theo định lí De Morgan ta biến đổi:

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

Đây chính là hàm Không Hoặc tuyệt đối (XNOR) kí hiệu như hình 4.13.



Hình 4.13: Mạch XNOR mạch so sánh 2 số A và B

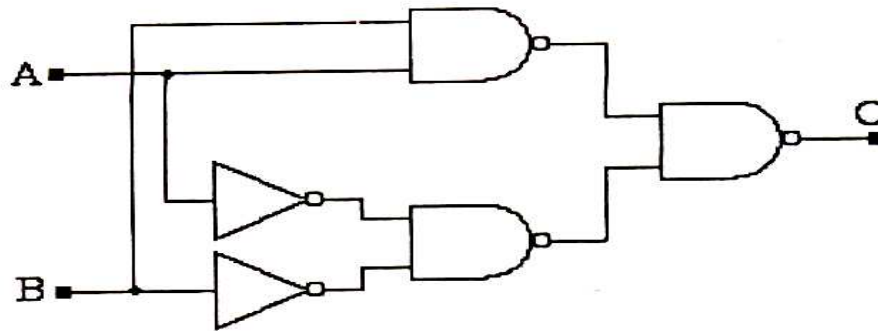
Có thể có nhiều cách thiết kế mạch này :

- Nếu muốn dùng cửa NOT và NAND:

Ta hãy biến đổi hàm logic $C = \bar{A}\bar{B} + AB$

$$C = \overline{\overline{\bar{A}\bar{B} + AB}} = \overline{\overline{\bar{A}\bar{B}} \cdot \overline{AB}}$$

Từ đây ta suy ra sơ đồ mạch như hình 4.14.



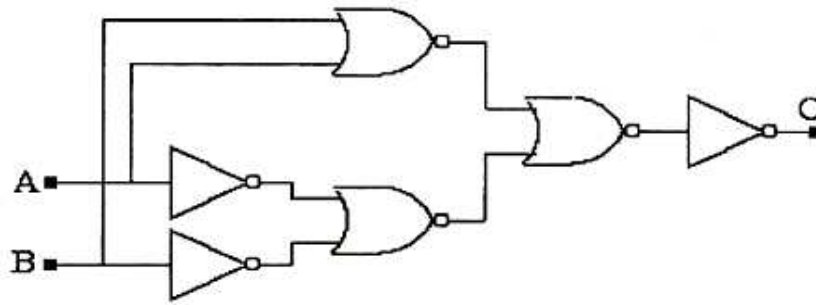
Hình 4.14: Mạch so sánh 2 số A và B dùng NOT và NAND

- Nếu dùng NOT và NOR: cũng từ $C = \bar{A}\bar{B} + AB$; Ta biến đổi:

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

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

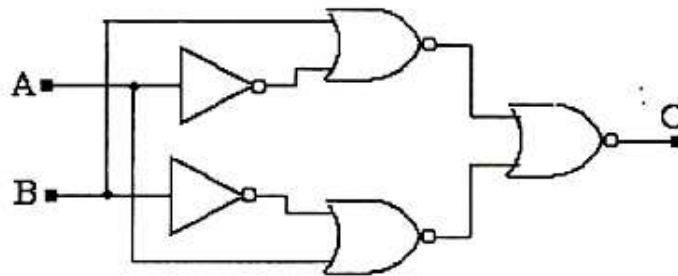
Từ đó suy ra sơ đồ mạch hình 4.15.



Hình 4.15: Mạch XNOR so sánh 2 số A và B dùng NOT và NOR

Bài tập

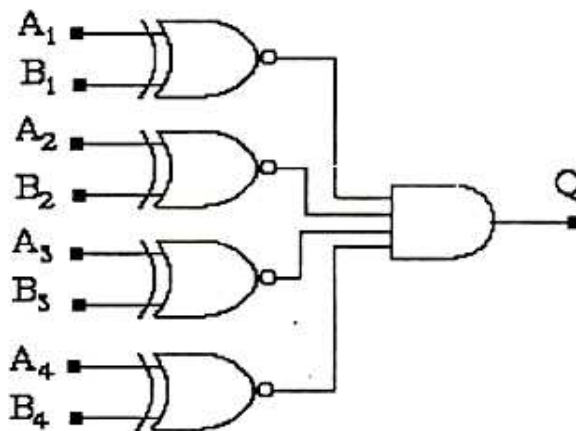
Sơ đồ hình 4.15 chưa tối ưu, vận dụng đại số logic chứng minh rằng ta có thể thiết kế mạch theo sơ đồ hình 4.16.



Hình 4.16

4.2.6.2. Mạch so sánh bằng nhau 2 số nhị phân 4 bit

Sơ đồ mạch cho trên hình 4.17.



Hình 4.17: Mạch so sánh 2 số nhị phân 4 bit

Ta có 2 số nhị phân 4 bit: $A_4 A_3 A_2 A_1$ và $B_4 B_3 B_2 B_1$

Khi 2 số bằng nhau lối ra $Q = 1$.

Khi 2 số khác nhau lối ra $Q = 0$.

(Khi $A = B$: lối ra của các XNOR đều bằng 1 cả, 4 lối vào mạch AND đều là 1, vậy lối ra $Q = 1$.)

Khi có 1 bit tương ứng của A và B khác nhau sẽ có mức 0 đưa vào mạch AND và lối ra $Q = 0$).

4.2.6.3. Bộ so sánh 1 bit

Bộ so sánh là mạch điện thực hiện chức năng logic xác định trong 2 số số nào lớn hơn số nào.

- Có 2 bit A và B nếu $A > B$ lối ra $Y_2 = 1, Y_1 = Y = 0$

$A < B$ lối ra $Y_1 = 1, Y_2 = Y = 0$

$A = B$ lối ra $Y = 1, Y_2 = Y_1 = 0$.

Từ đây ta lập được bảng chân lý như bảng 4.8.

Bảng 4.8: Bảng chân lý của bộ so sánh 1 bit

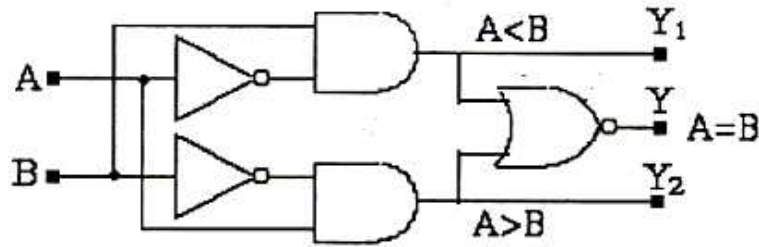
A	B	Y_2	Y_1	Y	kết luận
0	0	0	0	1	$A = B$
0	1	0	1	0	$A < B$
1	0	1	0	0	$A > B$
1	1	0	0	1	$A = B$

Từ bảng chân lý ta có hàm logic:

$$Y_2 = A\bar{B}, Y_1 = \bar{A}B$$

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

- Các hàm logic trên tương ứng với sơ đồ hình 4.18.



Hình 4.18: Sơ đồ logic mạch so sánh 1 bit

Bài tập áp dụng

Cho vi mạch 7402 là mạch NOR hai lối vào, hãy thiết kế mạch so sánh hai số nhị phân 1 bit có các lối vào lối ra như sơ đồ logic hình 4.18.

4.3. HỢP KÊNH MUX (MULTIPLEXER)

Chức năng của bộ hợp kênh (Multiplexer):

Bộ hợp kênh hay còn gọi là bộ dồn kênh là một mạch logic tổ hợp làm chức năng chọn lọc, truyền dữ liệu từ nhiều đường vào dồn về một đường ra. Các đường vào của bộ hợp kênh gồm hai loại: các đường vào dữ liệu và các đường vào điều khiển chọn kênh. Nếu số lối vào dữ liệu là n thì số lối vào điều khiển S tối thiểu $S = \log_2 n$.

Tùy theo tổ hợp các giá trị ở lối vào điều khiển chọn kênh mà lối ra được nối với một lối vào dữ liệu tương ứng. Để người dùng không bị nhầm lẫn trong việc xác định địa chỉ kênh, các nhà sản xuất vi mạch đã dùng các chỉ số kênh 0,1,2,... trùng với số nhị phân của tổ hợp lối vào điều khiển. Ví dụ như trường hợp bộ hợp kênh có 4 đường vào dữ liệu kí hiệu là D_0, D_1, D_2, D_3 ; hai đường vào điều khiển là A, B. Lối ra được kí hiệu là Y.

Sơ đồ khối mô tả chức năng hoạt động của bộ hợp kênh 4 lối vào dữ liệu, một đường ra dữ liệu được trình bày trên hình 18a.

Bảng chân lý của bộ hợp kênh 4 đường được trình bày trên bảng 4.11.

Từ bảng chân lý ta có thể tìm được phương trình logic của bộ hợp kênh này.

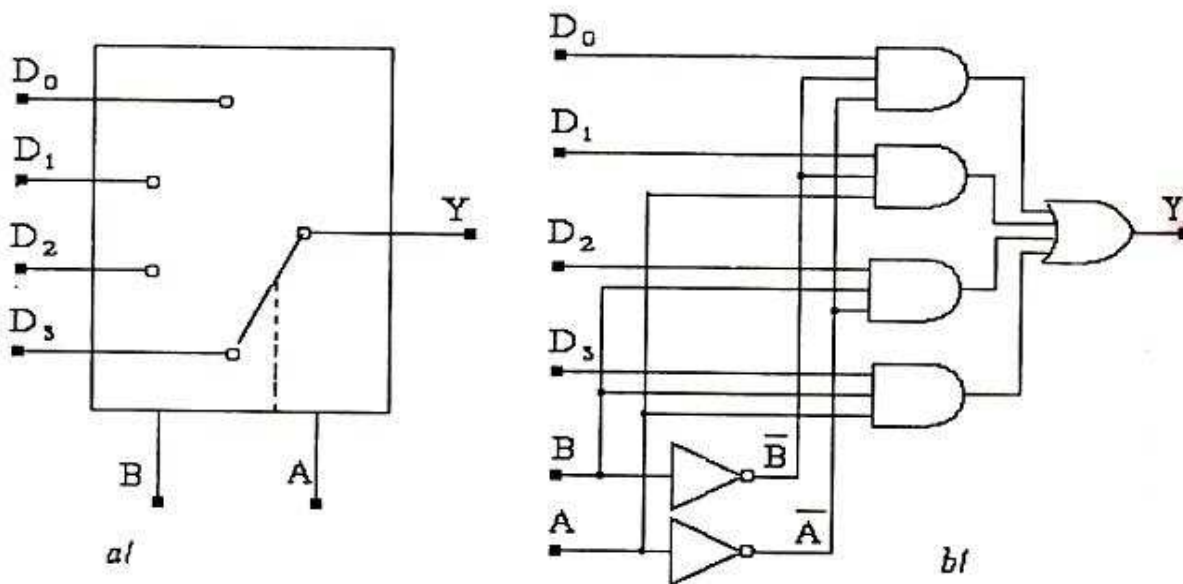
Bảng 4.11: Bảng chân lý của hợp kênh 4 lối vào và 1 lối ra dữ liệu

Lối vào dữ liệu	B	A	Y
D ₀	0	0	D ₀
D ₁	0	1	D ₁
D ₂	1	0	D ₂
D ₃	1	1	D ₃

Phương trình logic của bộ hợp kênh:

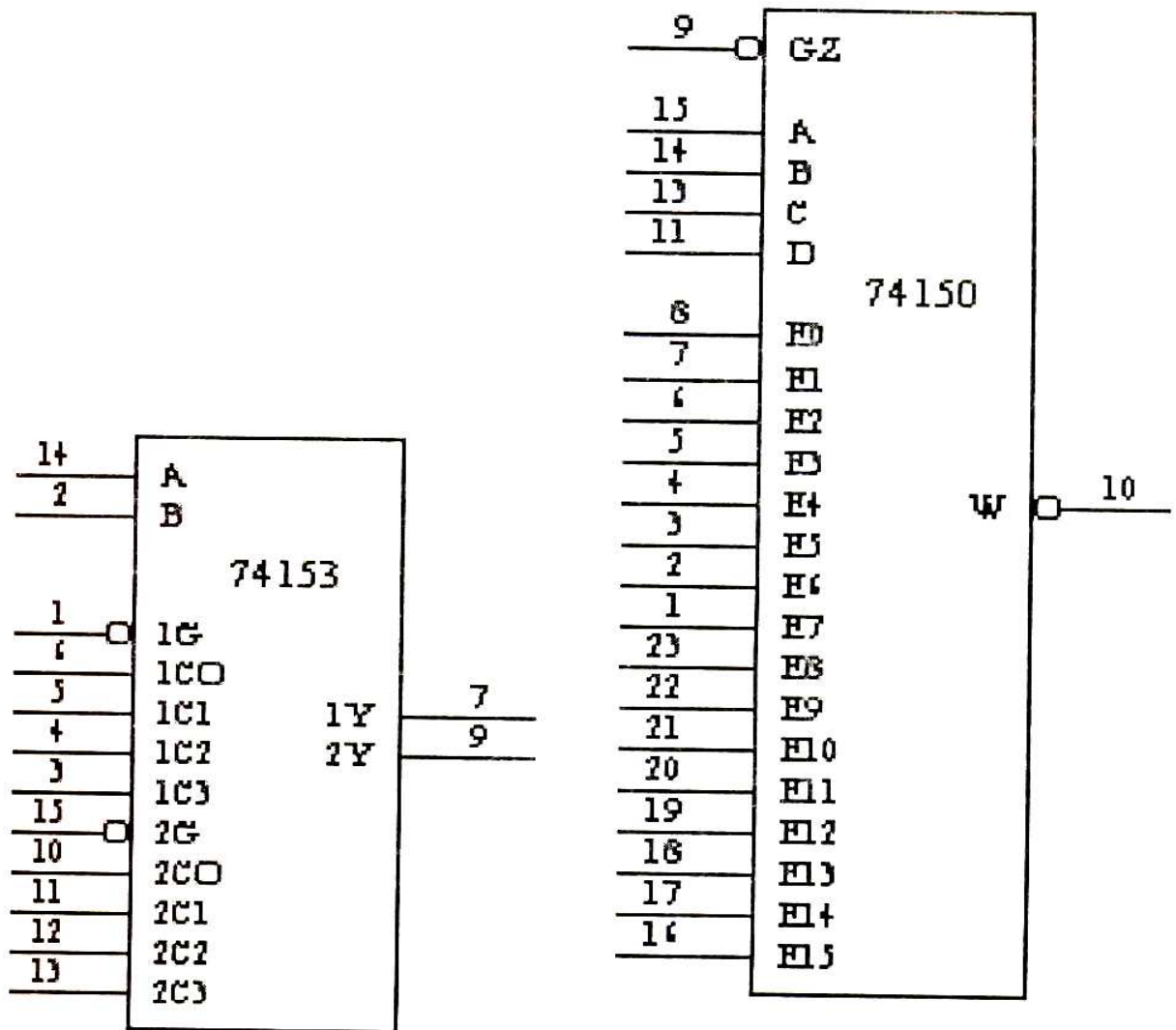
$$Y = \bar{A}.\bar{B}.D_0 + A.\bar{B}.D_1 + \bar{A}.B.D_2 + A.B.D_3$$

Sơ đồ logic bộ hợp kênh trình bày trên hình 4.19b



Hình 4.19: a/ Sơ đồ khối; b/ Sơ đồ logic của bộ hợp kênh 4 đường vào dữ liệu

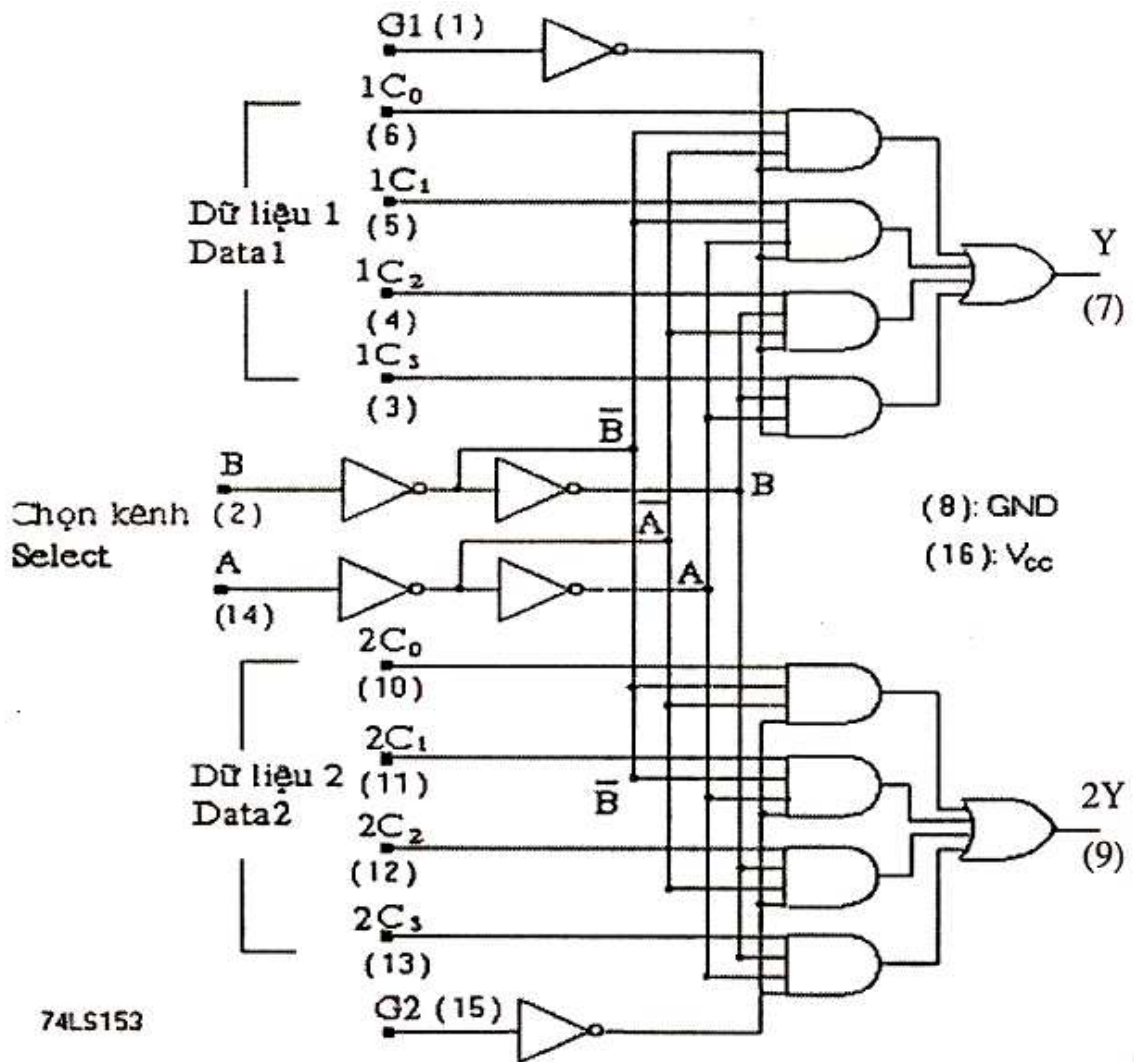
Trong thực tế, người ta đã chế tạo các bộ hợp kênh có 4, 8 hoặc 16 đường vào dữ liệu. Trên hình 4.20 giới thiệu kí hiệu của IC74153 và IC74150. Sơ đồ logic của vi mạch 74LS153 được vẽ trên hình 4.21. Trong vi mạch này có hai bộ hợp kênh có 4 đường vào dữ liệu kí hiệu là C₀, C₁, C₂, C₃ và một đường ra Y, cả hai bộ hợp kênh đều có chung hai lối vào điều khiển chọn kênh A, B. Cấu trúc của bộ hợp kênh này giống như sơ đồ logic của bộ hợp kênh 4 đường ta đã khảo sát trên hình 4.19b.



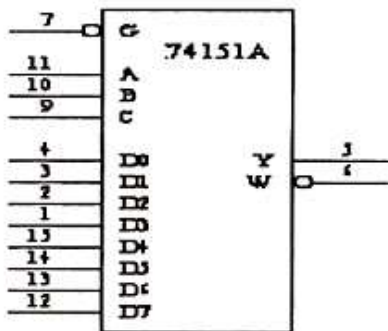
Hình 4.20: Kí hiệu của IC 74LS153 và IC 74150

Một điểm khác biệt là sơ đồ logic của mạch 74LS153 có thêm một lối vào điều khiển cho phép ra kí hiệu là G.

Các chữ số viết trong ngoặc đơn là kí hiệu chân của vi mạch. Mạch thuộc họ logic TTL chân 16 là nguồn nuôi mạch V_{CC} : +5V, chân 8 là đất (GND): 0V.



Hình 4.21: Sơ đồ logic mạch hợp kênh 4 đường 74LS153



Hình 4.22: Kí hiệu IC 74151

được trình bày trên hình 4.22 .

Vi mạch 74150 có kí hiệu logic cho trên hình 4.20 là bộ hợp kênh có 16 đường vào dữ liệu ($E_0, \dots, E_{14}, E_{15}$), 4 đường vào điều khiển chọn kênh A, B, C, D, một lối vào điều khiển cho phép ra GZ. Lối ra dữ liệu của mạch được kí hiệu là W.

Mạch IC 7451 là hợp kênh 8 đường có hai lối ra Y và W là đảo của Y chân 7 (G) điều khiển ra. Mạch có 8 đường vào dữ liệu là D_0, \dots, D_7 , 3 lối vào điều khiển A, B, C. Sơ đồ khối của hợp kênh 8 lối vào 1 lối ra 7451

Bảng chân lý của bộ hợp kênh này cho trên bảng 4.12, ở đây kí hiệu X là các thông tin logic ở lối vào D_i ứng với các giá trị bất kỳ có thể là L hoặc H.

Bảng 4.12: Bảng chân lý của hợp kênh 7451

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

Để tìm phương trình logic ta chú ý đến cột các lối vào điều khiển và cột lối ra.

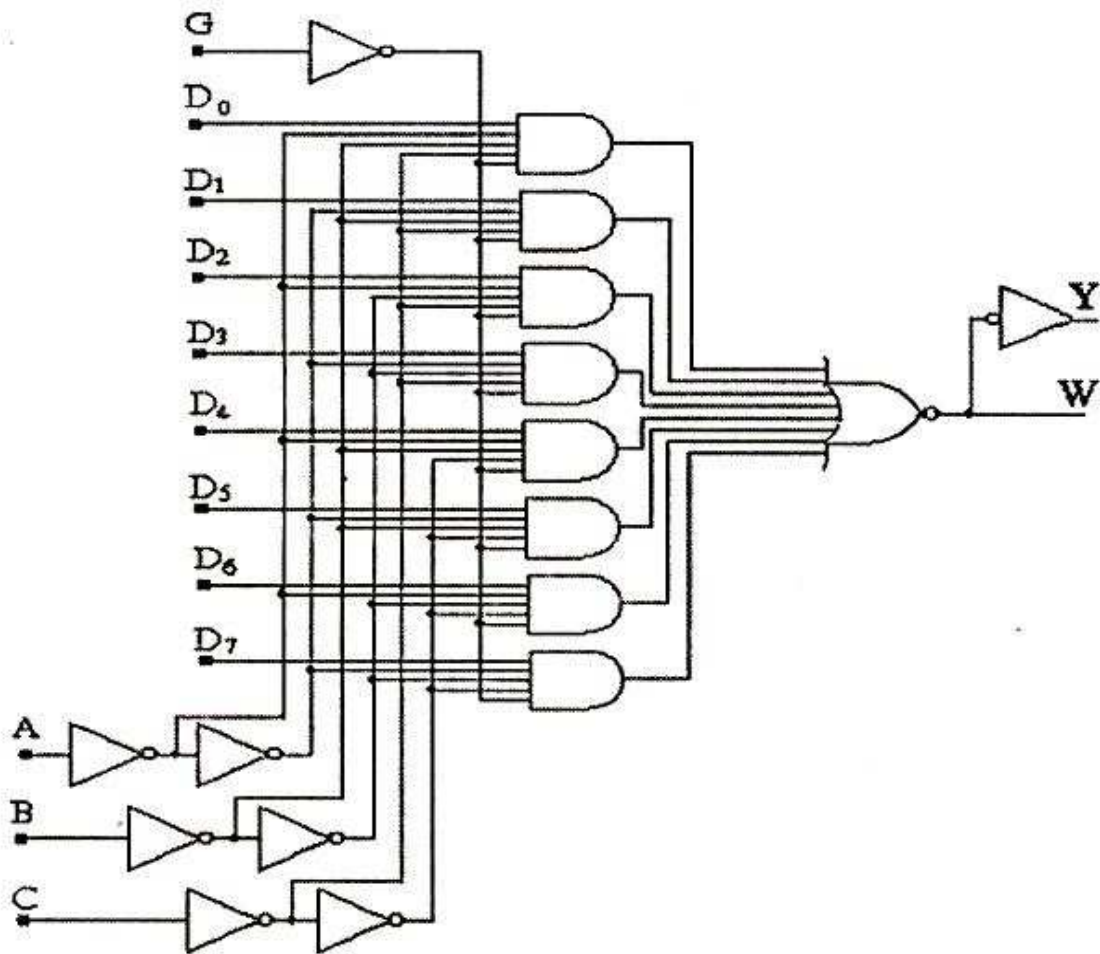
Từ bảng chân lý ta viết được hàm logic:

$$Y = \bar{A}\bar{B}\bar{C}\bar{G}D + \bar{A}\bar{B}\bar{C}GD + \bar{A}\bar{B}C\bar{G}D + \bar{A}\bar{B}CGD + \bar{A}BC\bar{G}D + \bar{A}BCGD + A\bar{B}\bar{C}\bar{G}D + A\bar{B}\bar{C}GD + ABC\bar{G}D + ABCGD$$

$$W = \bar{Y}$$

Từ phương trình logic ta thấy để thiết kế bộ hợp kênh này cần dùng: 8 mạch AND 5 lối vào, 1 mạch OR 8 lối vào và 4 cửa đảo. Sơ đồ logic thực tế của mạch 7451 vẽ ở hình 4.23.

Bộ hợp kênh làm nhiệm vụ chọn lọc số liệu D_i nào được đưa ra lối ra, vì vậy nó còn được gọi là bộ chọn (Selector).



Hình 4.23: Sơ đồ logic của vi mạch hợp kênh 8 đường 7451

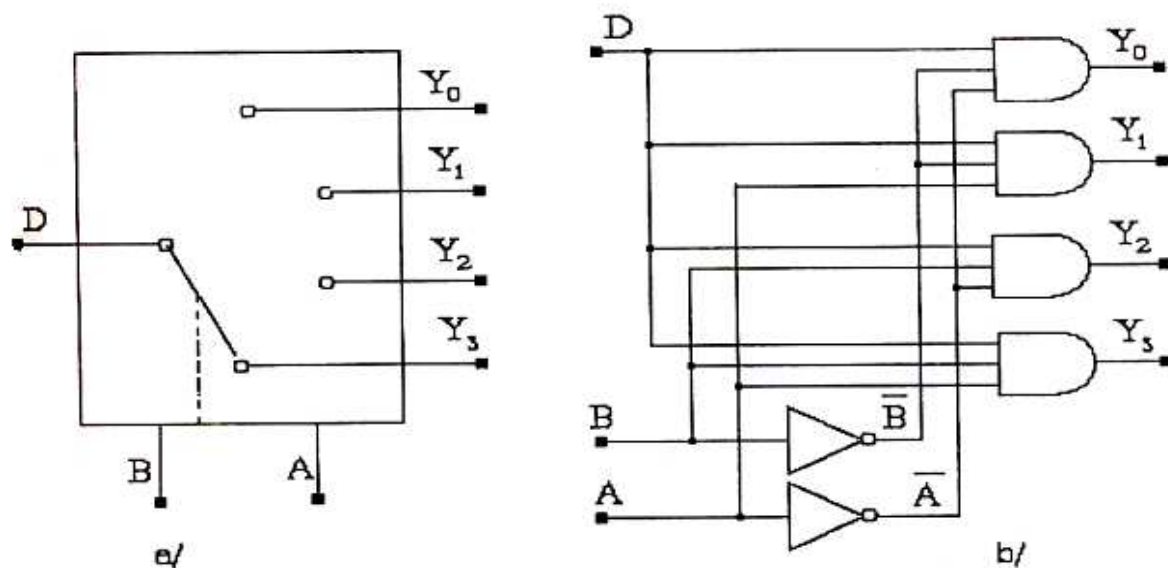
4.4. BỘ PHÂN KÊNH DEMUX (DEMULTIPLEXER)

4.4.1. Bộ phân kênh logic

Chức năng của bộ phân kênh (Demultiplexer):

Bộ phân kênh là một mạch logic tổ hợp có một đường vào và nhiều đường ra dữ liệu. Khác với bộ hợp kênh, bộ phân kênh làm chức năng chọn lọc, truyền dữ liệu từ một đường vào dữ liệu và đưa đến các đường ra riêng biệt. Cũng giống bộ hợp kênh, bộ phân kênh cũng có các đường vào điều khiển để chọn lối ra. Lối vào được nối với lối ra nào là tùy theo tổ hợp các giá trị của lối vào điều khiển.

Sơ đồ khối mô tả chức năng hoạt động của bộ phân kênh 4 lối ra được trình bày trên hình 4.24a.



Hình 4.24: Sơ đồ khối (a) và sơ đồ logic (b) của bộ phân kênh 1 lối vào, 4 lối ra.

Từ sơ đồ khối trên ta thấy bộ phân kênh này có 2 đầu vào điều khiển là A, B, một lối vào dữ liệu D và 4 lối ra dữ liệu được kí hiệu là Y_0, Y_1, Y_2, Y_3 . Ta có thể lập bảng chân lý biểu diễn sự hoạt động của bộ phân kênh được nêu trên bảng 4.13.

Từ bảng chân lý 4.13 ta có thể thiết lập các phương trình logic của bộ phân kênh. Ta có phương trình logic của bộ phân kênh:

Bảng 4.13: Bảng chân lý của bộ phân kênh 1 đường vào 4 đường ra dữ liệu

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

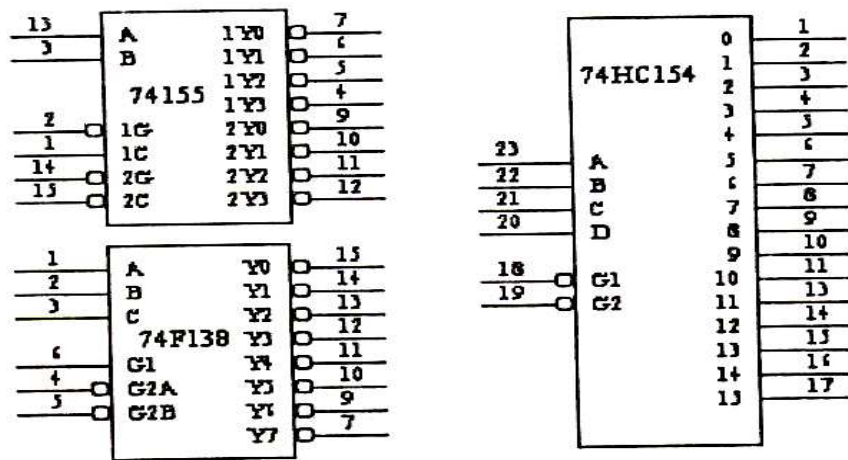
$$Y_0 = \bar{A} \cdot \bar{B} \cdot D$$

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

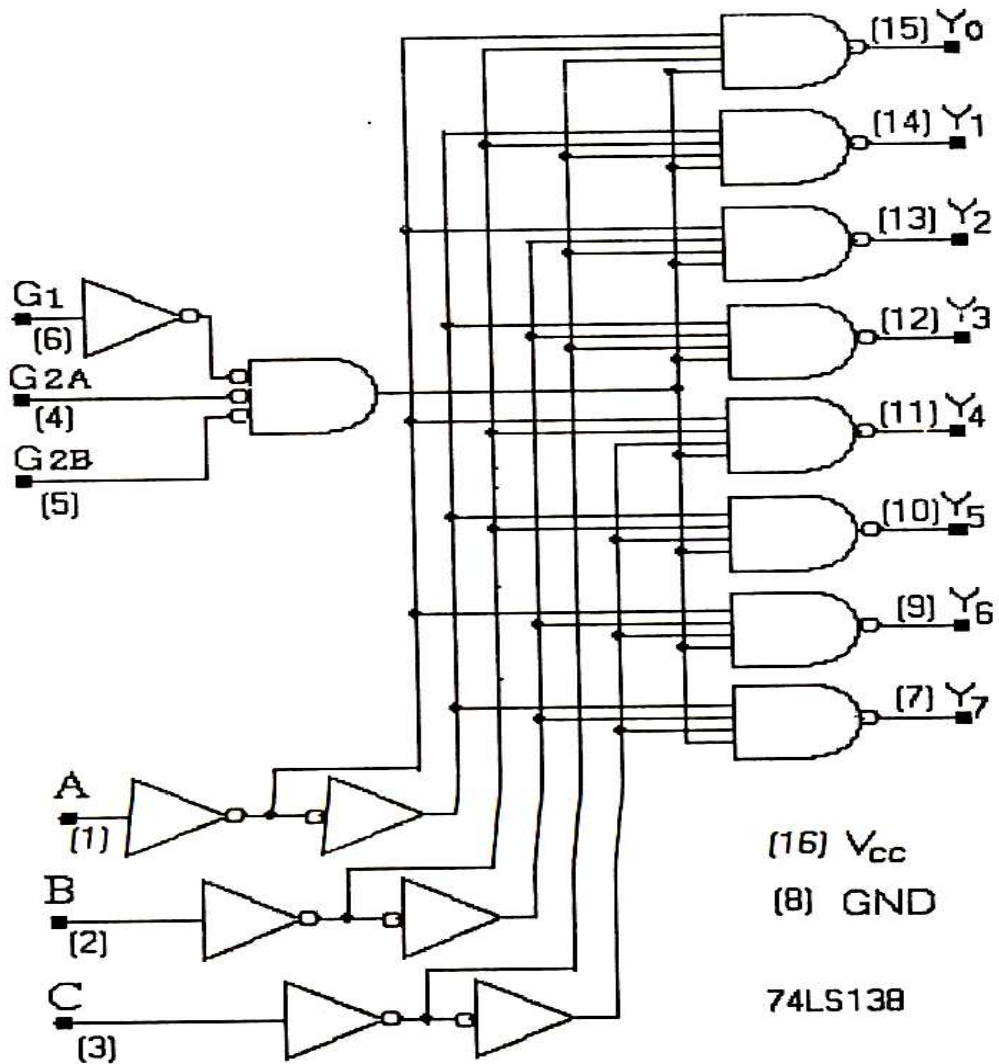
$$Y_2 = \bar{A} \cdot B \cdot D$$

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

Từ các phương trình logic trên ta xây dựng được sơ đồ logic của bộ phân kênh, nó được trình trên hình 4.24b. Hình 4.25 là các kí hiệu của các vi mạch phân kênh thường gặp trong thực tế, IC74155 có 2 bộ phân kênh 1 đường vào 4 đường ra dữ liệu. Hai lối vào điều khiển chọn kênh A, B được dùng chung cho cả hai bộ phân kênh. Mạch 74138 là bộ phân kênh 8 đường; 74154 là mạch phân kênh từ 1 ra 16 đường. Các mạch phân kênh còn được dùng làm mạch giải mã, trong trường hợp này dữ liệu được đưa vào lối vào điều khiển, ví dụ như mạch 74154 nếu dữ liệu đưa vào các lối vào A, B, C, D là mã nhị phân nó sẽ trở thành mạch giải mã 4 ra 16.



Hình 4.25: Kí hiệu các IC phân kênh



Hình 4.26: Sơ đồ logic của mạch giải mã, phân kênh 74LS138

Trên hình 4.26 trình sơ đồ logic của vi mạch 74LS138.

Mạch có thể dùng như bộ phân kênh một đường vào đưa ra 8 đường $Y_0, Y_1, Y_2, \dots, Y_7$. Các chỉ số 0,1,2,...,6,7 của các kênh ra tương ứng với tổ hợp các bit nhị phân ở lối vào điều khiển chọn kênh A,B,C.

Khi dùng IC 74LS138 làm bộ phân kênh thì các lối vào cho phép G_1, G_{2A}, G_{2B} được dùng làm lối vào dữ liệu (data).

Nếu dùng mạch 74LS138 làm mạch giải mã, chỉ cần mắc thêm một cửa đảo vào lối vào cho phép là ta có thể có được bộ giải mã đến 32.

4.4.2 Hợp kênh và phân kênh tương tự họ CMOS (Analog Multiplexer / demultiplexer)

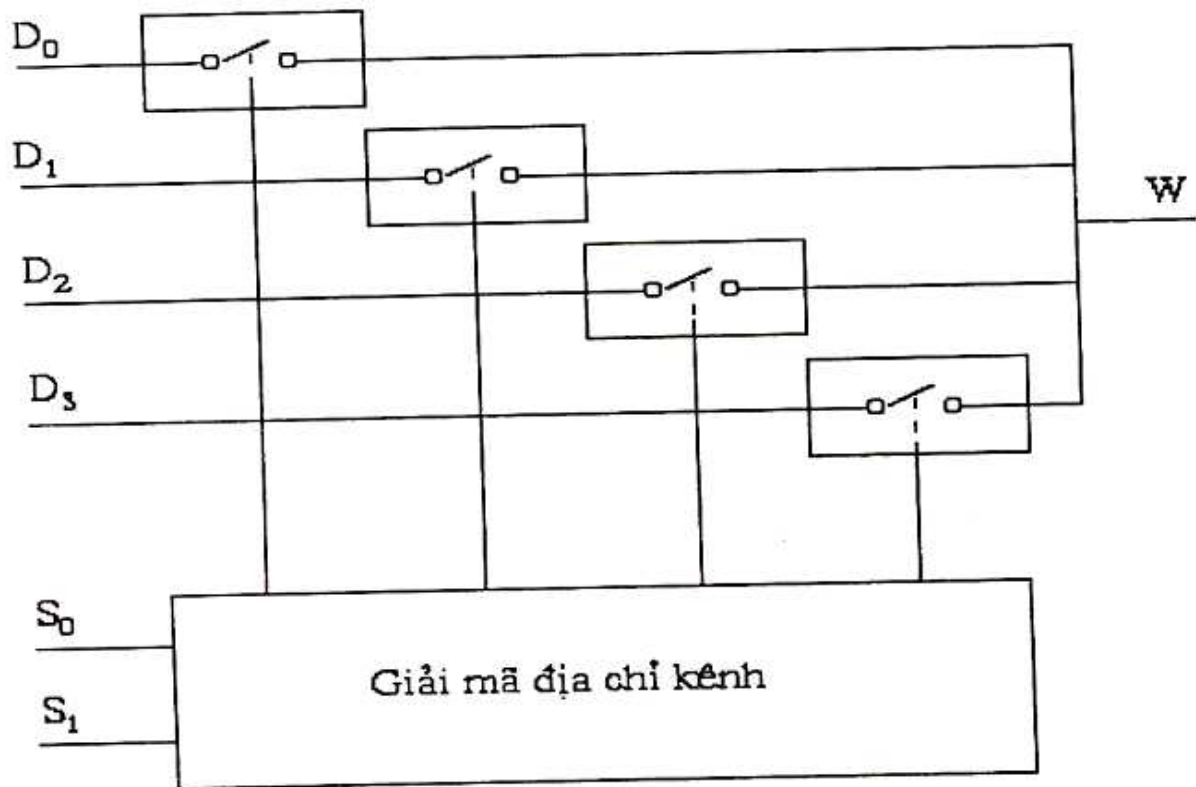
Bộ hợp kênh và phân kênh tương tự có chức năng tương tự như bộ hợp kênh và phân kênh số ta đã khảo sát ở trên. Tuy nhiên giữa chúng cũng có những điểm khác biệt như sau:

Điện áp vào và ra của hợp kênh tương tự là tín hiệu tương tự, nó có thể biến thiên liên tục theo thời gian tín hiệu ra giống hệt như tín hiệu vào. Bộ hợp kênh số và phân kênh số có tín hiệu vào và ra đều tương ứng với số nhị phân, nó chỉ có hai mức giá trị điện áp.

Trên cùng một mạch hợp kênh tương tự ta có thể làm mạch phân kênh tương tự bằng cách hoán đổi đầu vào thành đầu ra và đầu ra thành đầu vào.

Đối với hợp kênh số ta không thể làm như vậy. Hợp kênh số và phân kênh số có cấu trúc riêng biệt. Trên hình 4.27 trình cấu trúc sơ đồ khối của bộ hợp kênh tương tự có 4 đường vào dữ liệu. Nó gồm các chuyển mạch tương tự được ghép nối với nhau có các đầu vào riêng biệt còn đầu ra cùng hợp nhất về một đường.

Việc điều khiển đóng ngắt các chuyển mạch tương tự nhờ các tín hiệu lấy từ bộ giải mã địa chỉ kênh. Để hiểu rõ hoạt động của bộ hợp kênh tương tự, trước tiên ta cần khảo sát cấu trúc và nguyên lí hoạt động của bộ chuyển mạch tương tự hay còn gọi là bộ khoá tương tự.



Hình 4.27 cấu trúc của bộ hợp kênh tương tự

4.4.3. Bộ khoá tương tự điều khiển bằng logic

Trên hình 4.28a trình cấu trúc, sơ đồ nguyên lí của một chuyển mạch tương tự thuộc họ logic CMOS. Chuyển mạch tương tự là một khoá điện tử có hai lối vào A, C và lối ra B. C là lối vào điều khiển. Khi điện áp ở lối vào điều khiển:

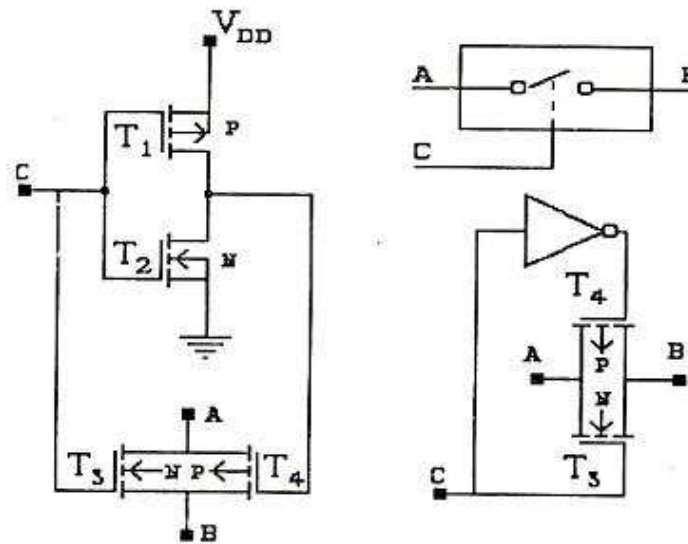
$V_C = H$ thì T_3, T_4 thông mạch: A được nối với B.

$V_C = L$ thì T_3, T_4 bị cấm: A được ngắt khỏi B.

Bộ khoá tương tự ở đây khác với bộ khoá logic. Ở bộ khoá logic, tín hiệu vào và ra đều là tín hiệu nhị phân chỉ có hai mức điện áp cố định. Trong bộ khoá tương tự thì tín hiệu lối vào và ra đều là tín hiệu tương tự và tín hiệu ra phải bảo đảm hoàn toàn giống hệt như tín hiệu ở lối vào cả về biên độ, tần số, pha và dạng của tín hiệu.

Nói cách khác, bộ khoá tương tự phải có tính chất như bộ khoá cơ học ~~trở lại~~ và phải có tốc độ đóng ngắt nhanh mà các bộ khoá cơ học do quán ~~lực~~ lớn không thể làm việc được với tần số đóng ngắt cao.

Chức năng của khoá tương tự giống như công tắc điện cơ khí, ta có thể ~~hoán vị~~ hoán vị hai đầu vào ra dữ liệu A,B như là công tắc cơ khí.



Hình 4.28a: Khoá điện tử

Các thông số cơ bản của bộ khoá tương tự

Bộ khoá tương tự có các thông số cơ bản là:

- Tốc độ chuyển mạch hay tần số làm việc. Tần số truyền cao có thể lên đến 10 MHz.

- Điện trở thông mạch R_{ON} nhỏ, điện trở ngắt mạch R_{OFF} lớn.

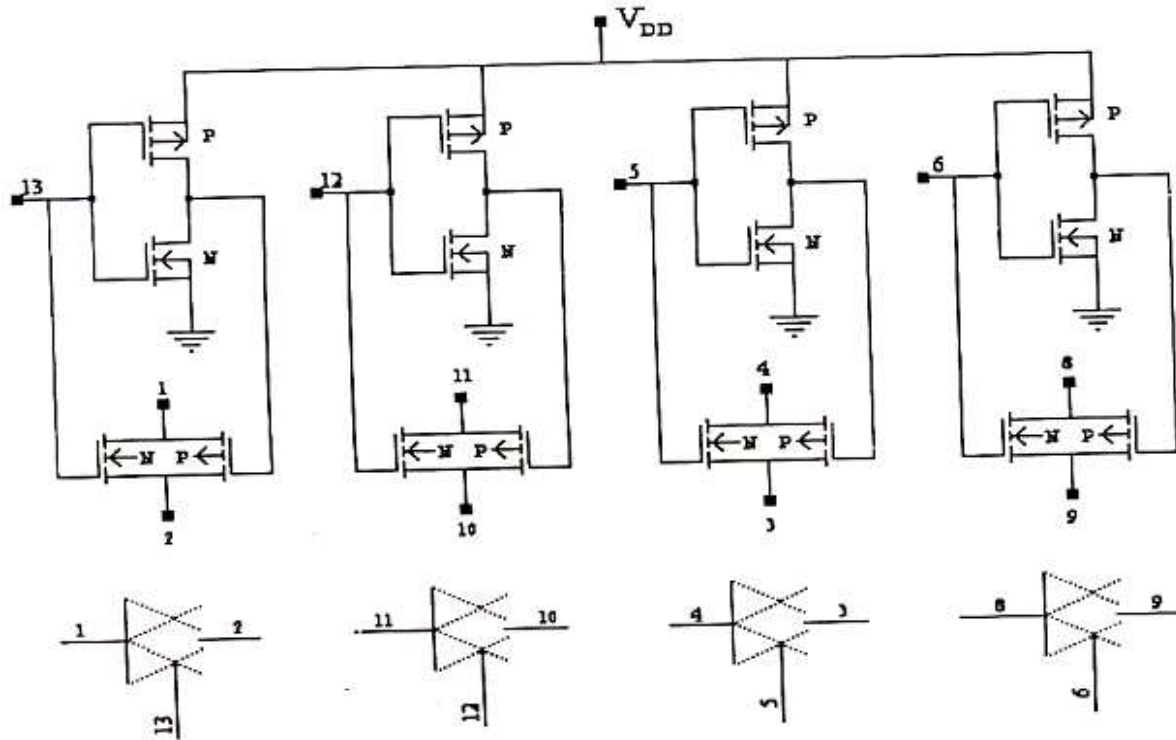
Để đảm bảo yêu cầu trên, người ta thường thiết kế bộ khoá tương tự dựa trên cơ sở Transistor trường JFET hoặc Transistor MOSFET.

Sơ đồ nguyên lý và ký hiệu của chuyển mạch tương tự họ CMOS

Trên hình 4.28b trình sơ đồ nguyên lý và ký hiệu của vi mạch chuyển mạch tương tự CD4016 thuộc họ CMOS. Vi mạch này có 14 chân có 4 bộ khoá tương tự, chân 7 là đất (GND), chân 14 là chân nguồn nuôi V_{DD} . Vì mạch thuộc họ CMOS nên có thể nuôi mạch từ 3 đến 12V. Chú ý: Để mạch làm việc được bình thường thì điện áp đưa vào các chân không được lớn hơn điện áp nguồn. Mạch có các thông số như sau: điện trở lối vào cực của vô

cùng lớn: $10^{12}\Omega$, tín hiệu điều khiển hoàn toàn cách li với tín hiệu truyền
 R_{ON} : 300Ω , R_{OFF} vô cùng lớn, dòng qua công tắc khi ngắt mạch: $10pA$.

Mạch có thể truyền các tín hiệu tương tự và tín hiệu số có biên độ đỉnh cực đại V_{Smax} lớn: $V_{Smax} = \pm 7,5V$ khi nguồn nuôi mạch $V_{DD} - V_{SS} = 15V$.



Hình 4.28b: Sơ đồ nguyên lý và ký hiệu chuyển mạch tương tự CD4016

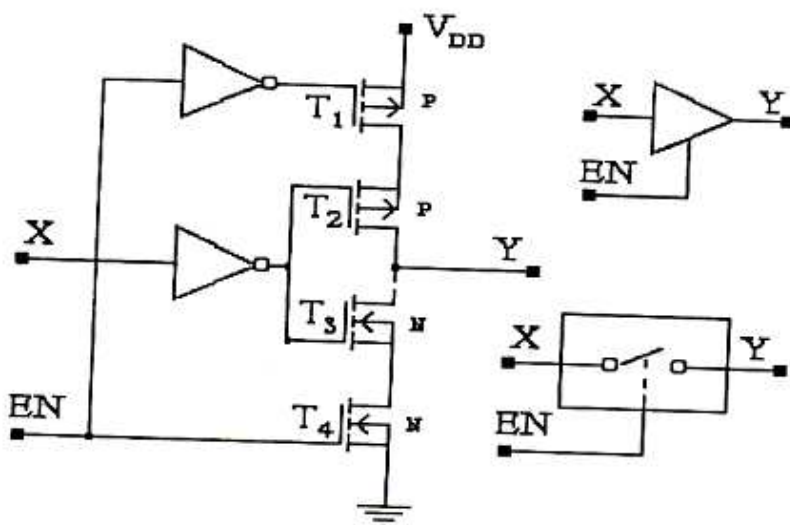
4.4.4. Cửa đệm ba trạng thái họ CMOS

Để có thể so sánh sự khác biệt giữa chuyển mạch tương tự với chuyển mạch logic, trên hình 4.29 trình sơ đồ nguyên lý của cửa đệm ba trạng thái có chức năng như một khoá chuyển mạch logic (khóa điện tử dùng cho mạch logic). Từ sơ đồ hình 4.29 ta thấy khi đặt mức điện áp cao vào chân cho phép:

$$EN \text{ (Enable)} = H$$

Hai transistor T_1 T_4 đều thông mạch, cực S của T_2 được nối lên nguồn còn cực S của T_3 được nối xuống đất, hai transistor T_2 và T_3 tạo thành cửa đảo. Mạch gồm hai cửa đảo mắc nối tiếp tương đương như cửa đệm ta có: $Y = X$ điều này tương đương như chuyển mạch điện tử ở trạng thái đóng mạch. Khi chân EN đặt ở mức điện áp thấp: $EN = L$. Cả hai T_1 và

T_4 cấm, T_2 , T_3 ngắt khỏi nguồn và đất, lối ra có trở kháng cao, điều này tương đương như chuyển mạch điện tử ở trạng thái ngắt mạch.



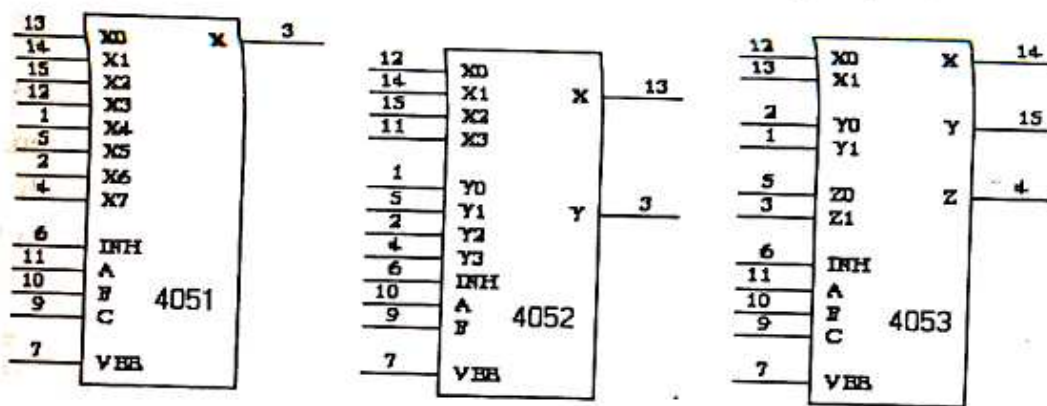
Hình 4.29: Cửa đệm 3 trạng thái họ logic CMOS

Những ứng dụng của chuyển mạch tương tự:

Chuyển mạch tương tự ngoài việc dùng làm các bộ hợp kênh, phân kênh tín hiệu số và tín hiệu tương tự, nó còn được dùng trong các mạch điều chế, giải điều chế, biến đổi A/D, D/A.

Chuyển mạch tương tự còn được dùng trong các mạch điều khiển số: điều khiển tần số, pha, điện trở và hệ số khuếch đại của tín hiệu tương tự.

4.4.5. Các vi mạch hợp kênh tương tự có trong thực tế



Hình 4.30: Kí hiệu các IC hợp kênh tương tự CD4051, CD4052, CD4053

Hình 4.30 giới thiệu các kí hiệu một số mạch hợp kênh, phân kênh tương tự thường gặp trong thực tế.

Các vi mạch này đều thuộc họ CMOS, vừa là hợp kênh vừa là phân kênh. Các đường vào ra đều có thể thực hiện theo hai chiều.

IC CD4051 là bộ hợp kênh, có 8 đường vào dữ liệu là: X_0, X_1, \dots, X_7 , một đường ra chung là X. Ba đường điều khiển chọn kênh là: A, B, C. Đường điều khiển cấm INH (inhibit) ngăn chặn không cho dữ liệu ra. Mạch CD4051 có thể làm bộ phân kênh khi đó đầu ra chung X trở thành đầu vào chung còn các đầu vào X_0, \dots, X_7 trở thành các đầu ra của bộ phân kênh.

IC CD4052 có hai bộ hợp kênh 4 đường. Bộ thứ nhất có các đường vào dữ liệu là X_0, \dots, X_3 , đường ra chung là X. Bộ thứ hai có các lối vào là Y_0, \dots, Y_3 , lối ra chung là Y. Cả hai bộ hợp kênh này đều có chung 2 đường điều khiển chọn kênh A, B và đường điều khiển INH.

IC CD4053 có 3 bộ hợp kênh hai đường vào một đường ra và một lối vào điều khiển chọn kênh riêng cho từng bộ. Bộ 1 có lối vào điều khiển chọn kênh là A hai lối vào dữ liệu là X_0, X_1 , lối ra chung là X. Bộ 2 có lối vào điều khiển chọn kênh là B hai lối vào dữ liệu là Y_0, Y_1 , lối ra chung là Y. Bộ 3 có lối vào điều khiển chọn kênh là C hai lối vào dữ liệu là Z_0, Z_1 , lối ra chung là Z.

Cả ba bộ hợp kênh chỉ có chung nhau lối vào INH.

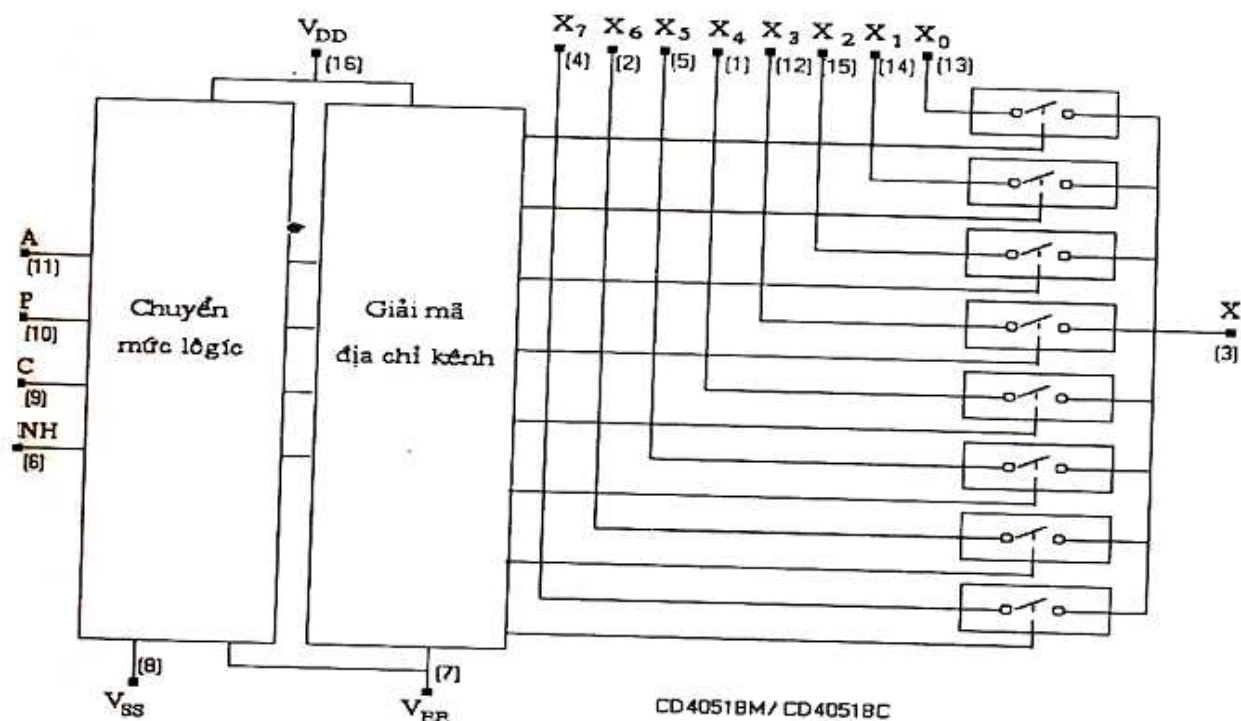
Cấu trúc của các bộ hợp kênh tương tự:

Cấu trúc của các bộ hợp kênh nêu trên trên hình 4.30 đều giống nhau nó có các chuyển mạch tương tự ghép song song đầu ra chung được nối hợp nhất lại với nhau thành một đường. Việc điều khiển chọn kênh được tạo thành từ hai khối: chuyển đổi mức logic và giải mã địa chỉ kênh tạo ra các xung điều khiển đóng ngắt chuyển mạch tương tự.

Trên hình 4.31 trình sơ đồ khối mô tả cấu trúc của vi mạch 4051. Bộ hợp kênh tương tự 8 đường vào một đường ra đã được giới thiệu ở trên. Nhờ khối chuyển mức logic mà tín hiệu ở các lối vào điều khiển chọn kênh và lối vào INH có thể là các tín hiệu tương tự bất kỳ có biên độ V_{in} nằm trong phạm vi điện áp nguồn V_{EE}, V_{DD} : $-V_{EE} < V_{in} < V_{DD}$.

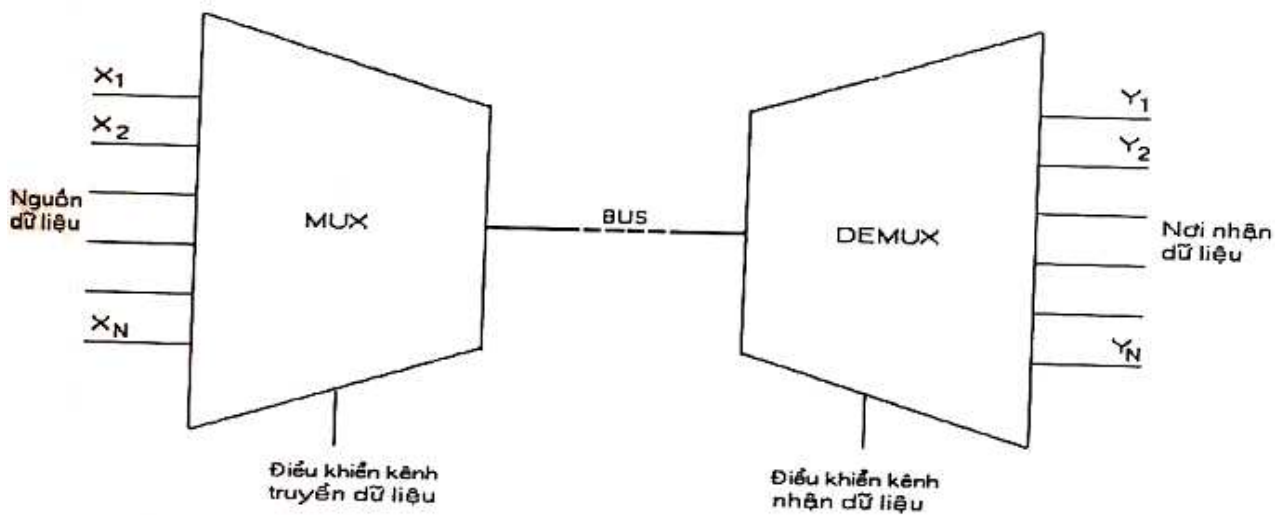
Ví dụ: nếu ta nuôi mạch bằng nguồn $V_{DD} = +5V$, $V_{SS} = 0V$, $V_{EE} = -5V$ thì các tín hiệu tương tự có giá trị từ $-5V$ đến $+5V$ đều có thể điều khiển các lối vào số. Nguồn nuôi $V_{DD} - V_{EE}$ hoàn toàn độc lập với nguồn nuôi mạch logic CMOS V_{DD}, V_{SS} .

Nếu ta dùng nguồn $V_{DD} = +5V$; $V_{SS} = V_{EE} = 0V$ thì các tín hiệu tương đưa vào các lối vào điều khiển có giá trị nằm trong phạm vi $0V < V_{in} < +5V$ đều có thể điều khiển chọn kênh.



Hình 4.31: Sơ đồ khối của bộ hợp kênh tương tự 4051

4.4.6. Những ứng dụng của hợp kênh và phân kênh



Hình 4.32: Sơ đồ mô hình chọn và truyền số liệu

Các bộ phân kênh còn được dùng để giải mã, ta sẽ gặp ở phần sau. Ở đây ta nêu ra một ứng dụng cụ thể của bộ hợp kênh và phân kênh trong sơ đồ chọn và truyền số liệu theo mô hình nêu trong hình 4.32.

Sơ đồ hình 4.32 cho chúng ta thấy từ nguồn số liệu $X_1, X_2 \dots X_n$, bộ hợp kênh sẽ chọn 1 trong các số liệu đưa lên BUS để truyền đi. Ở đầu kia của BUS, nơi nhận số liệu bộ phân kênh sẽ điều khiển số liệu đến 1 nơi nhận xác định Y_N nào đó.

Bài tập thực hành:

1/ IC 7450 là mạch hợp kênh 16 đường. Hãy thiết lập bảng chân lý của bộ hợp kênh này.

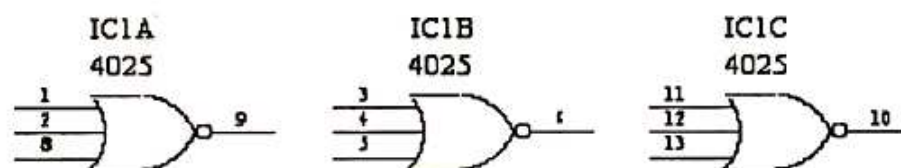
2/ Khảo sát hoạt động lập bảng chân lý của vi mạch hợp kênh tương tự 8 đường họ CMOS 4051.

3/ Dùng chuyển mạch tương tự 4016 (sơ đồ logic hình 4.28b) và 4025 là mạch không hoặc 3 lối vào có kí hiệu logic được trình ở hình 4.33 tạo thành bộ hợp kênh tương tự 4 đường vào dữ liệu.

4/ Dùng vi mạch 4025 tạo thành:

- hợp kênh 4 đường vào dữ liệu;
- phân kênh 4 đường ra dữ liệu.

5/ Nêu những ứng dụng của hợp kênh, phân kênh trong kỹ thuật điện tử và viễn thông.



Hình 4.33: Kí hiệu logic IC4025

4.5. CÁC MẠCH MÃ HOÁ VÀ GIẢI MÃ

4.5.1. Mạch mã hoá (Encoder)

Máy tính điện tử chỉ tiếp nhận thông tin dưới dạng nhị phân (ứng với một trong hai trạng thái 0, 1), vì vậy các số liệu, các lệnh cho máy tính thực

hiện đều phải được mã hoá dưới dạng nhị phân, mỗi chữ số thập phân đều được biểu diễn bằng một từ mã nhị phân 4 bit.

Một điều rõ ràng là đối với con người thì dùng hệ đếm thập phân vẫn là thuận lợi nhất. Chính vì thế mà trong kỹ thuật số và máy tính, người ta thường hay sử dụng các bộ biến đổi mã để trao đổi và xử lý thông tin giữa con người và máy tính, giữa các khối chức năng trong máy tính. Chẳng hạn, để đưa số liệu vào máy tính, ta cần biến đổi mã thập phân sang mã nhị phân. Kết quả tính toán từ máy tính đưa ra phải được giải mã từ nhị phân sang thập phân.

Trong máy tính và kỹ thuật số, người ta đã dùng rất nhiều các loại mã khác nhau. Trong giáo trình này ta chỉ khảo sát một số các mạch chuyển đổi mã cơ bản nhất như mạch chuyển đổi mã từ thập phân sang mã BCD, từ mã nhị phân sang mã bù nhị phân, mã Gray v.v...

1. Sơ đồ biến đổi mã nhị phân sang mã bù nhị phân.

Trong máy tính điện tử người ta dùng số bù nhị phân để biểu diễn số thập phân, nhờ vậy mà người ta có thể dễ dàng thực hiện phép tính trừ bằng cách cộng số bị trừ với số bù của số trừ.

Theo định nghĩa, số bù nhị phân B của số nhị phân A nào đó là một số mà khi cộng với số nhị phân A thì cho ta một số mà tất cả các bit bằng 0 và một số nhớ bằng 1 ở bit cao nhất. Quy tắc tìm số bù nhị phân như sau:

- Viết số đó dưới dạng nhị phân.
- Đảo các bit của từ nhị phân.
- Cộng thêm 1 vào hàng có trọng số bé nhất.

Dựa vào quy tắc trên ta có thể lập được bảng chân lý của bộ biến đổi mã có các đầu vào là A_0, A_1, A_2, A_3 là mã nhị phân 4 bit, bốn đầu ra là 4 bit mã bù nhị phân B_0, B_1, B_2, B_3 .

Ta có bảng chân lý trên bảng 4.14. Từ bảng chân lý ta tìm được các phương trình logic biểu diễn phép biến đổi mã nhị phân 4 bit sang mã bù nhị phân 4 bit. Hình 4.34 là sơ đồ logic bộ biến đổi mã nhị phân 4 bit sang mã bù nhị phân 4 bit.

Bảng 4.14: Bảng chân lý bộ biến đổi mã nhị phân sang mã bù nhị phân

Số thập phân	Mã nhị phân				Mã bù nhị phân			
	A ₃	A ₂	A ₁	A ₀	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1
2	0	0	1	0	1	1	1	0
3	0	0	1	1	1	1	0	1
4	0	1	0	0	1	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	0
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	0
9	1	0	0	1	0	1	1	1
10	1	0	1	0	0	1	1	0
11	1	0	1	1	0	1	0	1
12	1	1	0	0	0	1	0	0
13	1	1	0	1	0	0	1	1
14	1	1	1	0	0	0	1	0
15	1	1	1	1	0	0	0	1

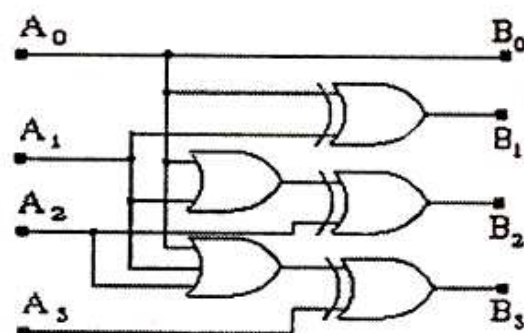
Phương trình logic bộ biến đổi mã :

$$B_0 = A_0$$

$$B_1 = A_0 \oplus A_1$$

$$B_2 = (A_0 + A_1) \oplus A_2$$

$$B_3 = (A_0 + A_1 + A_2) \oplus A_3$$



Hình 4.34: sơ đồ logic bộ biến đổi mã nhị phân sang mã bù nhị phân

Từ bảng chân lý vận dụng đại số Boole ta có thể tìm được các phương trình logic biểu diễn phép biến đổi mã nhị phân sang mã bù nhị phân.

Từ hệ phương trình này ta xây dựng được sơ đồ logic của hàm biến đổi dùng các phần tử hoặc tuyệt đối và hoặc.

2. Sơ đồ biến đổi mã nhị phân sang mã Gray

Bảng 4.15: Bảng chân lý biến đổi mã nhị phân sang mã Gray

TP	A ₃	A ₂	A ₁	A ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Từ bảng chân lý ta có phương trình logic biến đổi từ mã nhị phân sang mã Gray:

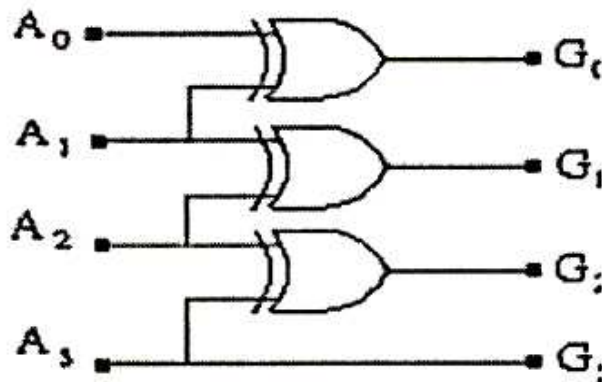
$$G_0 = A_0 \oplus A_1$$

$$G_1 = A_1 \oplus A_2$$

$$G_2 = A_2 \oplus A_3$$

$$G_3 = A_3$$

Từ phương trình logic ta có sơ đồ logic biến đổi từ mã nhị phân sang mã Gray:



Hình 4.35: Sơ đồ logic bộ biến đổi mã nhị phân sang mã Gray

Mã nhị phân rất thuận tiện khi tiến hành tính toán bằng máy tính nhưng nó có nhược điểm lớn khi sử dụng cho các thiết bị đo đạc, trong đó có sử dụng các quá trình quá độ giữa hai giá trị đo kế tiếp nhau của đầu cảm biến. Ví dụ như trong hệ điều khiển tự động góc quay của bánh răng, hay hệ thống xác định vị trí của một chuyển động nào đó. Khi thông tin về hai vị trí kế tiếp nhau được ghi nhận bởi đầu cảm biến được mã hoá dưới dạng mã nhị phân thì có thể gặp trường hợp có các hàng bit khác nhau rất nhiều, chẳng hạn như hai vị trí 7 và 8 biểu diễn theo mã nhị phân tương ứng sẽ là 0111 và 1000. Trong trường hợp này ta thấy có sự khác nhau về trạng thái của cả bốn bit mặc dù hai vị trí cần xác định là kế cận nhau. Các hệ điều khiển trong thời gian thực thường có thời gian trễ khi chuyển trạng thái từ vị trí này sang vị trí khác trong khoảng thời gian quá độ ngắn ngủi này trạng thái của hệ không có mã số xác định. Hệ thống có thể cho ra các mã số 1101, 1010, 0110, ..., các giá trị trung gian này khác xa giá trị ban đầu 0111 cũng như giá trị cuối cùng 1000. Điều này có thể gây nên một sai số trong phép đo. Ta có thể khắc phục nhược điểm này bằng cách sử dụng mã Gray.

Ưu điểm của mã Gray so với mã nhị phân là hai số kế tiếp nhau, nếu biểu diễn bằng mã Gray thì chỉ khác nhau 1 bit còn các bit khác vẫn giữ nguyên. Vì vậy, trong thời gian quá độ tín hiệu ra của hệ thống chỉ có thể là số đo của vị trí đầu hoặc vị trí tiếp theo mà thôi, nghĩa là sự bất định đã giảm đi rất nhiều so với dùng mã nhị phân.

3. Sơ đồ biến đổi mã thập phân sang mã BCD

Bộ chuyển đổi mã thập phân sang mã BCD có 10 lối vào L_0 đến L_9 ứng với các số từ 0 đến 9 của hệ 10 và lối ra A, B, C, D ứng với 1 từ nhị phân 4 bit mã BCD.

Bảng chân lý của phép biến đổi thập phân sang BCD được cho trên bảng 4.16:

Bảng 4.16.

N	L ₉	L ₈	L ₇	L ₆	L ₅	L ₄	L ₃	L ₂	L ₁	L ₀	D	C	B	A
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	0	1	0	1
6	0	0	0	1	0	0	0	0	0	0	0	1	1	0
7	0	0	1	0	0	0	0	0	0	0	0	1	1	1
8	0	1	0	0	0	0	0	0	0	0	1	0	0	0
9	1	0	0	0	0	0	0	0	0	0	1	0	0	1

Từ bảng chân lý ta có phương trình:

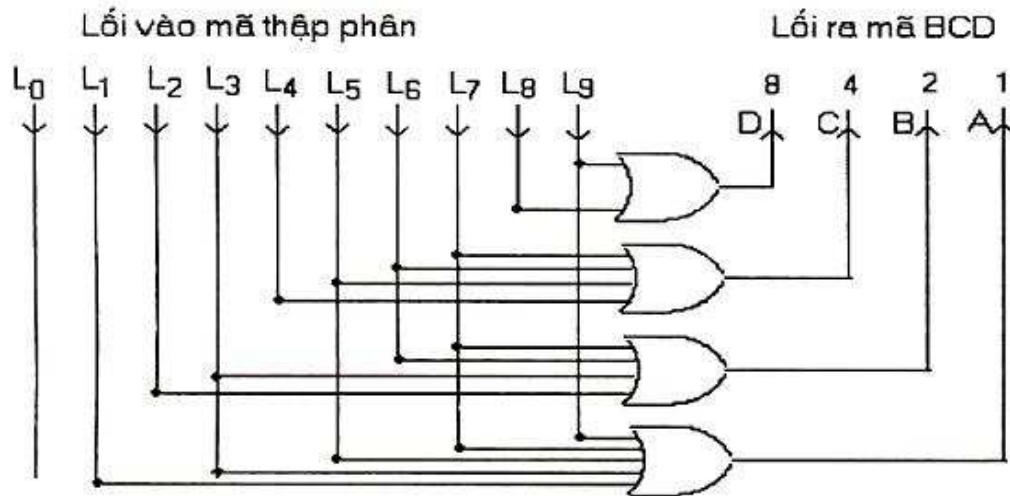
$$A = L_1 + L_3 + L_5 + L_7 + L_9$$

$$B = L_2 + L_3 + L_6 + L_7$$

$$C = L_4 + L_5 + L_6 + L_7$$

$$D = L_8 + L_9$$

Từ hệ phương trình trên, ta dễ dàng thiết kế bộ biến đổi mã thập phân sang mã BCD bằng các phần tử hoặc có nhiều lối vào hình 4.36. Tuy nhiên, trong thực tế, ta thường gặp các phần tử logic cơ bản có hai lối vào, do đó ở đây ta sẽ thiết kế bộ biến đổi mã từ các phần tử logic 2 lối vào.

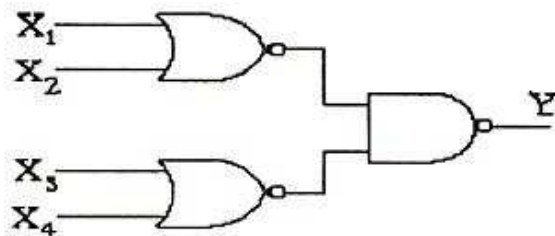


Hình 4.36: Sơ đồ logic bộ biến đổi mã thập phân ra mã BCD

Để tạo mạch hoặc 4 lối vào, dựa vào định luật De Morgan ta có:

$$X_1 + X_2 + X_3 + X_4 = \overline{\overline{(X_1 + X_2)} + \overline{(X_3 + X_4)}} = \overline{(X_1 + X_2)(X_3 + X_4)}$$

Từ đó ta có sơ đồ logic của mạch hoặc 4 lối vào:



4.5.2. Mạch giải mã (Decoder)

Quá trình ngược với mã hoá được gọi là giải mã. Nghĩa là từ một tổ hợp giá trị của nhóm mã n chữ số hệ 2 ta tìm lại được 1 trong N ký hiệu hoặc số tương ứng với tổ hợp đó. Về thực chất, các bộ giải mã cũng là các bộ biến đổi mã, chúng biến đổi từ các mã nhị phân, BCD sang mã thập phân hay mã 7 đoạn. Để xây dựng các bộ giải mã, chúng ta có thể áp dụng phương pháp thiết kế logic chúng ta đã làm quen ở các tiết trước để tạo thành các bộ giải mã từ các phần tử logic cơ bản. Thực tế hiện nay người ta không làm như vậy, mà thường dùng các vi mạch giải mã có sẵn trên thị trường. Trong mục này chúng tôi sẽ giới thiệu các vi mạch này

để bạn đọc có điều kiện nắm vững được nguyên tắc hoạt động của các mạch này, dễ dàng dùng được chúng.

4.5.2.1. Giải mã BCD sang thập phân

Bộ giải mã BCD sang thập phân có các lối vào A, B, C, D; các lối ra là $L_0, L_1 \dots L_9$ biểu diễn các số thập phân từ 0, 1 ... 9. Bảng chân lý cho trên bảng 4.17. Mỗi $L_i = 1$ một lần ở đường chéo bảng, còn lại là bằng 0 cả.

Bảng 4.17: Bảng chân lý Giải mã BCD sang mã 10

D	C	B	A	L_0	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Từ bảng chân lý ta có thể suy ra phương trình logic của bộ giải mã: Từ hàm logic, ta có thể xây dựng sơ đồ giải mã này khi dùng 4 mạch NOT và 10 mạch AND 4 lối vào.

$$L_0 = \bar{A}\bar{B}\bar{C}\bar{D}$$

$$L_5 = A\bar{B}\bar{C}\bar{D}$$

$$L_1 = A\bar{B}\bar{C}D$$

$$L_6 = \bar{A}B\bar{C}\bar{D}$$

$$L_2 = \bar{A}B\bar{C}D$$

$$L_7 = A\bar{B}C\bar{D}$$

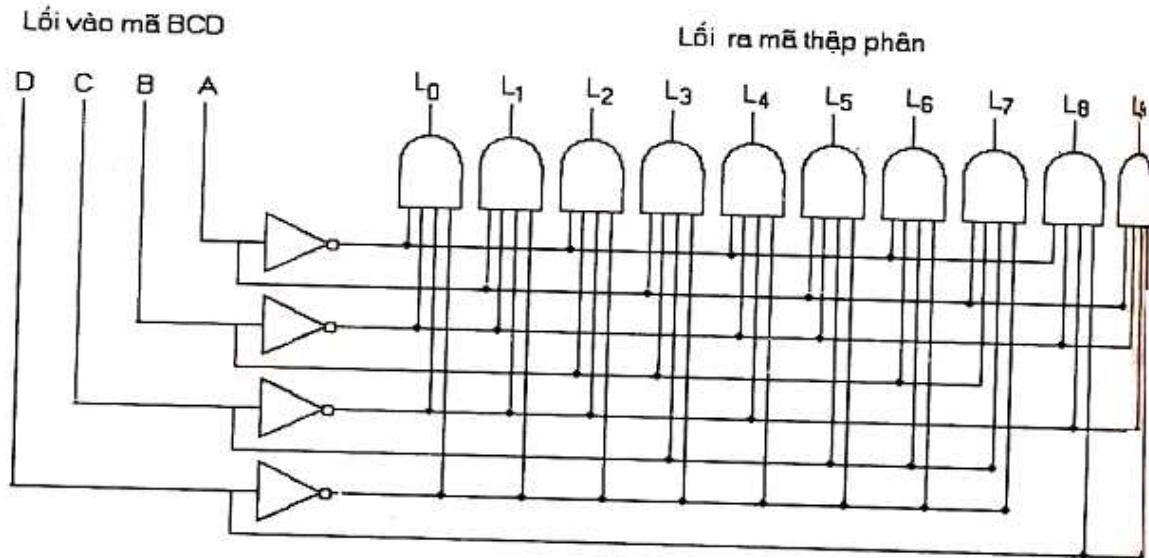
$$L_3 = A\bar{B}C\bar{D}$$

$$L_8 = \bar{A}B\bar{C}D$$

$$L_4 = \bar{A}B\bar{C}D$$

$$L_9 = A\bar{B}C\bar{D}$$

Sơ đồ bố trí mạch như trên hình 4.37.

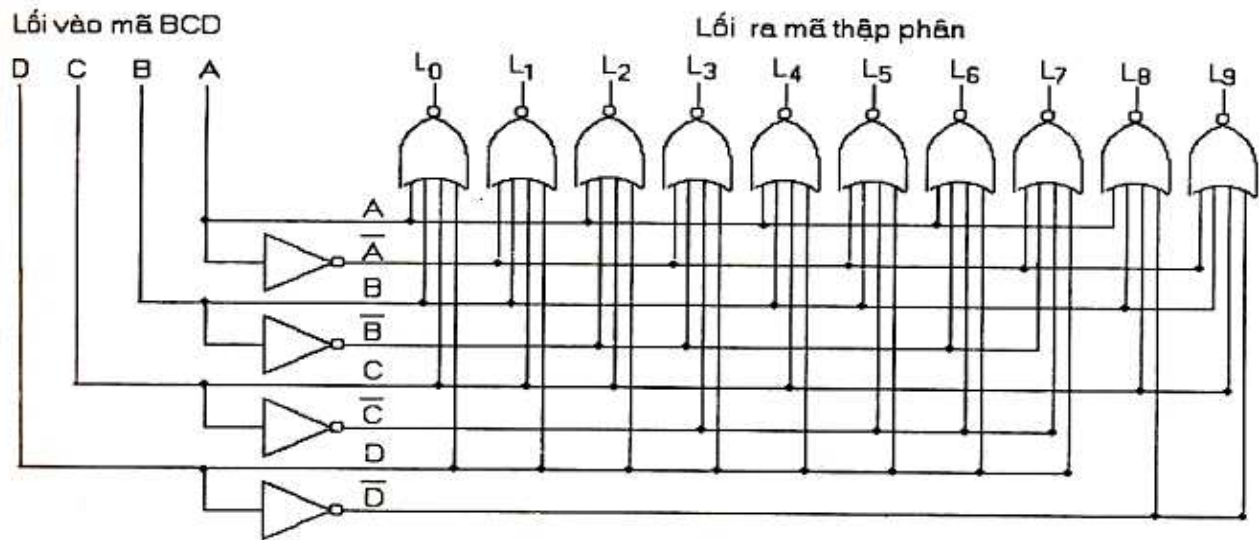


Hình 4.37: Sơ đồ mạch giải mã BCD sang mã 10

Mạch giải mã này có thể xây dựng từ các mạch NOR. Muốn vậy, ta làm phép đảo 2 lần các hàm logic L_i rồi dùng định lý De Morgan chuyển từ phép nhân logic thành phép cộng logic ta được:

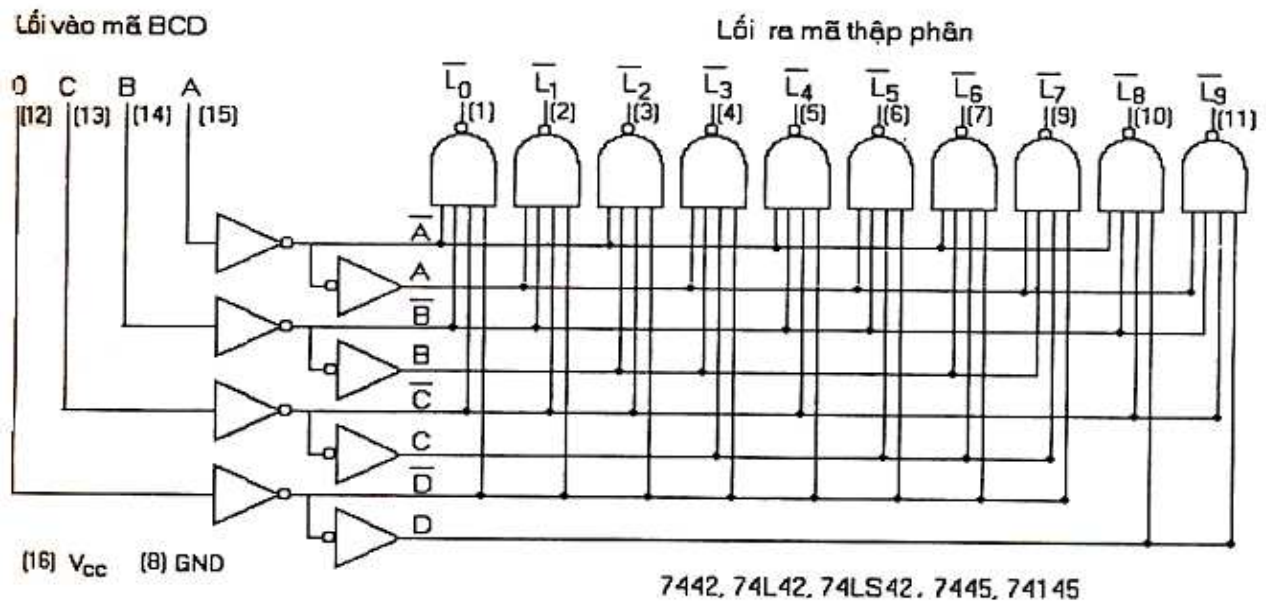
$$\begin{aligned}
 L_0 &= \overline{A\overline{B}\overline{C}\overline{D}} = \overline{A + B + C + D} \\
 L_1 &= \overline{A\overline{B}C\overline{D}} = \overline{\overline{A} + B + C + D} \\
 L_2 &= \overline{A\overline{B}C\overline{D}} = \overline{A + \overline{B} + C + D} \\
 L_3 &= \overline{A\overline{B}C\overline{D}} = \overline{\overline{A} + \overline{B} + C + D} \\
 L_4 &= \overline{A\overline{B}C\overline{D}} = \overline{A + B + \overline{C} + D} \\
 L_5 &= \overline{A\overline{B}C\overline{D}} = \overline{\overline{A} + B + \overline{C} + D} \\
 L_6 &= \overline{A\overline{B}C\overline{D}} = \overline{A + \overline{B} + \overline{C} + D} \\
 L_7 &= \overline{A\overline{B}C\overline{D}} = \overline{\overline{A} + \overline{B} + \overline{C} + D} \\
 L_8 &= \overline{A\overline{B}C\overline{D}} = \overline{A + B + C + \overline{D}} \\
 L_9 &= \overline{A\overline{B}C\overline{D}} = \overline{\overline{A} + B + C + \overline{D}}
 \end{aligned}$$

Từ hàm logic ta thấy để lắp mạch giải mã này cần dùng 4 mạch NOT và 10 mạch NOR 4 lối vào và sơ đồ logic của bộ giải mã được trình trên hình 4.38.



Hình 4.38: Sơ đồ logic của bộ giải mã từ mã BCD ra thập phân dùng các mạch logic cơ bản NOR và NOT

Các vi mạch giải mã có trong thực tế như 7442, 74L42, 74LS42, 7445, 74145 các vi mạch này có 16 chân có 4 lối vào A, B, C, D và 10 chân lối ra tác động thấp 0, 1, 2... 9. Sơ đồ nguyên lí và kí hiệu chân của các IC này được trình trên hình 4.39. Các IC này tuy có kí hiệu khác nhau nhưng đều có cùng một sơ đồ lôgic và kí hiệu các chân giống nhau, chữ số ghi trong ngoặc là kí hiệu các chân của IC.

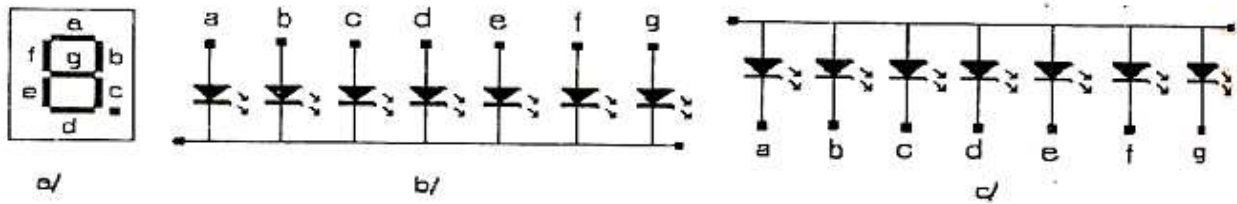


Hình 4.39: Sơ đồ logic của các IC 7442, 74L42, 74LS42, 7445, 74145

4.5.2.2. Giải mã BCD sang mã 7 đoạn

Mã nhị phân BCD được chuyển sang thập phân và hiển thị các số thập phân bằng 7 đoạn sáng. 7 đoạn sáng này có thể là LED (hoặc tinh thể lỏng) ứng với mỗi tổ hợp xác định các thanh sáng sẽ hiển thị cho ta một chữ số ở hệ 10.

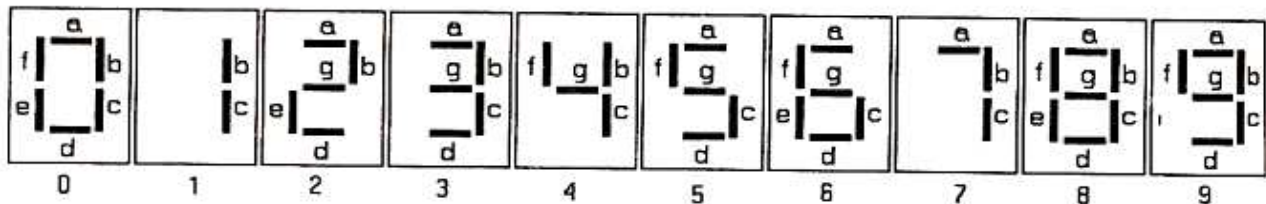
Hình 4.40 mô tả cấu trúc của đèn hiển thị số 7 đoạn dùng điốt phát quang, cách kí hiệu các đoạn bằng các chữ cái a, b, c, d, e, f, g. Bảy đoạn là 7 điốt phát quang



Hình 4.40: Cấu trúc của chỉ thị số 7 đoạn dùng đèn LED

Các LED màu đỏ sẽ phát sáng khi điện áp giữa anốt và catốt là 1,6V tương ứng với dòng qua điốt là 5mA. Các đèn chỉ thị số dùng LED phát ánh sáng màu xanh cần điện áp giữa anốt và catốt cao hơn cỡ chừng 2,5V đến 3,5V. Ngoài 7 LED tạo ra 7 đoạn sáng dùng để biểu thị mười chữ số, các đèn chỉ thị số còn có thêm một điốt phát quang nữa để tạo dấu chấm thập phân.

Khi chế tạo đèn chỉ thị số người ta đã nối chung các anốt của các điốt ở trong vỏ và dẫn ra bằng 2 lối ra chung hình 4.40c. Bộ giải mã 7 đoạn dùng với đèn chỉ thị này có các lối ra hoạt động ở mức thấp lối ra của bộ giải mã được nối với catốt của các điốt qua điện trở hạn chế dòng. Cũng có loại đèn chỉ thị 7 đoạn khi chế tạo người ta đã nối catốt chung với nhau để dùng cho các vi mạch giải mã có các lối ra hoạt động ở mức cao. Sự hiển thị mười chữ số thập phân bằng đèn LED 7 đoạn chỉ thị số được minh họa trên hình 4.41.



Hình 4.41: Sự hiển thị các chữ số thập phân ở trên đèn hiển thị số 7 đoạn

Từ hình 4.41 chúng ta có thể thiết lập được bảng chân lý của bộ giải mã từ BCD ra mã 7 đoạn, bảng chân lý này được nêu trên bảng 4.18, trong đó

các giá trị tổ hợp biến lối vào A, B, C, D được viết theo mã BCD và các giá trị của tổ hợp các hàm 7 thanh sáng a, b, c, d, e, f, g ở lối ra được thiết lập theo sự hiển thị 10 chữ số thập phân nêu trên hình 4.41.

Bảng 4.18: Bảng chân lý của bộ chuyển mã BCD sang mã 7 đoạn

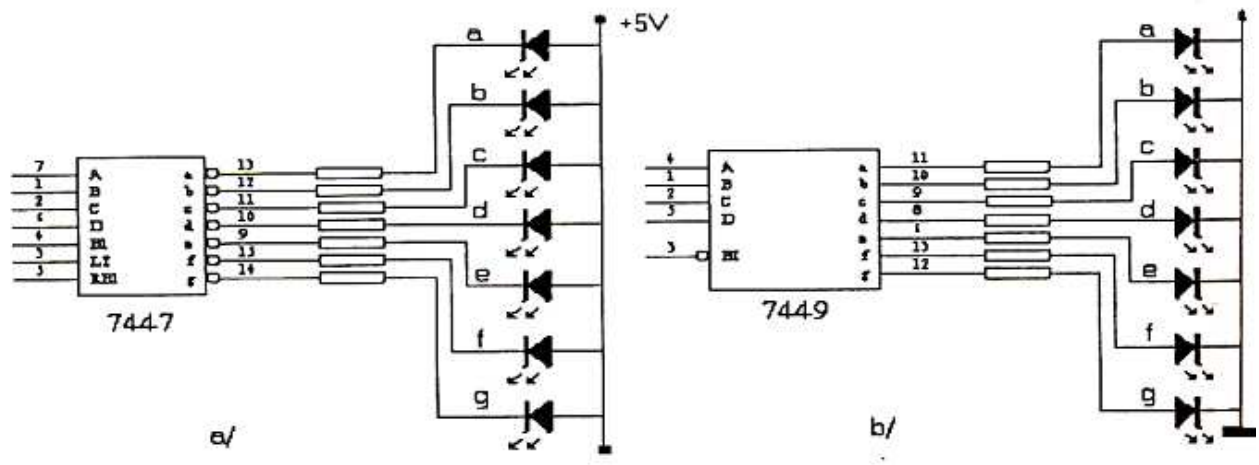
D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1

Nhìn bảng chân lý 4.18 ta thấy rằng giá trị của 7 hàm a, b,.... g có giá trị là mức 1 thì các thanh LED phát sáng, bảng chân lý này tương ứng với các vi mạch giải mã có lối ra tác động ở mức điện áp cao. Từ bảng chân lý ta có thể viết được các hàm logic và xây dựng được sơ đồ logic cho mạch giải mã này từ các cửa logic cơ bản. Đó là trên nguyên tắc, còn trong thực tế người ta đã chế tạo sẵn các vi mạch để giải mã nhị phân ra 7 đoạn. Tốt hơn hết là chúng ta hãy tìm hiểu và sử dụng chúng.

Các vi mạch giải mã 7 đoạn

Các vi mạch 7448, 74LS48, 7449, 74LS49 là các IC giải mã 7 đoạn có lối ra tác động ở mức cao, ta có thể dùng chúng để giải mã từ mã BCD ra thập phân. Quy luật hiển thị các chữ số thập phân của các vi mạch này về cơ bản giống như hình 4.41, chỉ khác đôi chút là số 6 không dùng thanh a và số 9 không dùng thanh d.

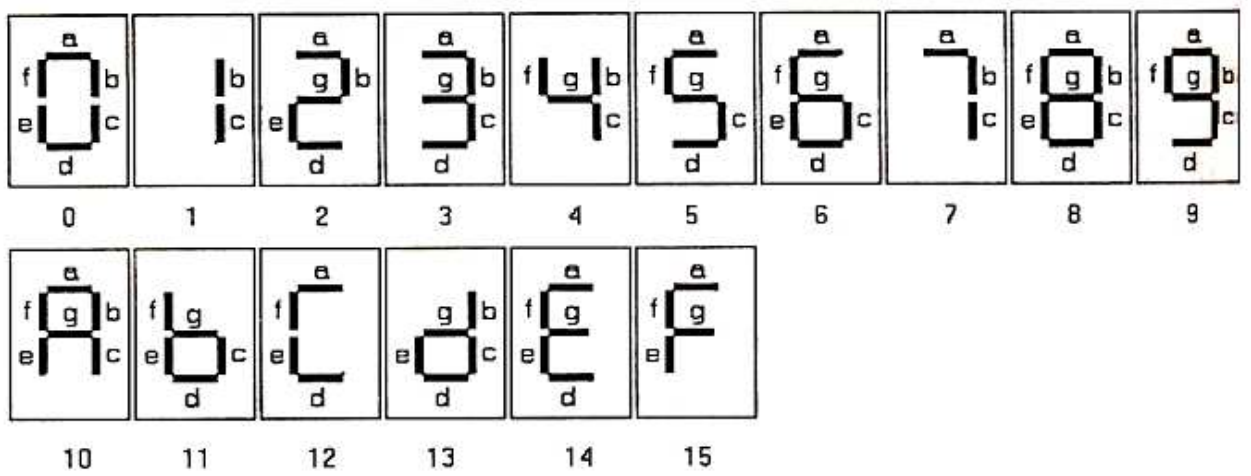
Các mạch giải mã 7 đoạn 7447A, 74L47, 74S47 là các vi mạch 16 chân, số 6 và số 9 chỉ có 5 thanh sáng giống như 7448, 7449. Vi mạch có lối ra tác động thấp (mức 0) nên đèn chỉ thị 7 đoạn có anốt chung. Trên hình 4.42 giới thiệu cách mắc các vi mạch giải mã với đèn LED 7 đoạn



Hình 4.42: Sơ đồ mắc IC giải mã 7 đoạn với đèn LED chỉ thị

Mạch giải mã MC 14495 cũng là giải mã nhị phân ra 7 đoạn. Vì mạch 16 chân, có lối ra tác động cao (mức 1) nên đèn chỉ thị 7 đoạn có catốt chung. Số 6 và số 9 có 6 thanh sáng, các số thập phân: 10,11,12,13,14,15 được hiển thị giống như các chữ số trong hệ thập lục phân .

Trên hình 4.43 minh họa sự hiển thị của các đèn chỉ thị số theo mã 7 đoạn khi nó được dùng với mạch giải mã MC14495



Hình 4.43: Sự hiển thị các chữ số của IC MC14495

4.5.2.3. Giải mã Johnson sang thập phân

Bộ giải mã Johnson có các lối vào là một tổ hợp 5 bit mã Johnson, lối ra là $L_0, L_1, L_2, \dots, L_8, L_9$ tương ứng mười chữ số từ 0 đến 9 của hệ thập phân.

Bảng chân lý của bộ giải mã này được nêu trên bảng 4.19.

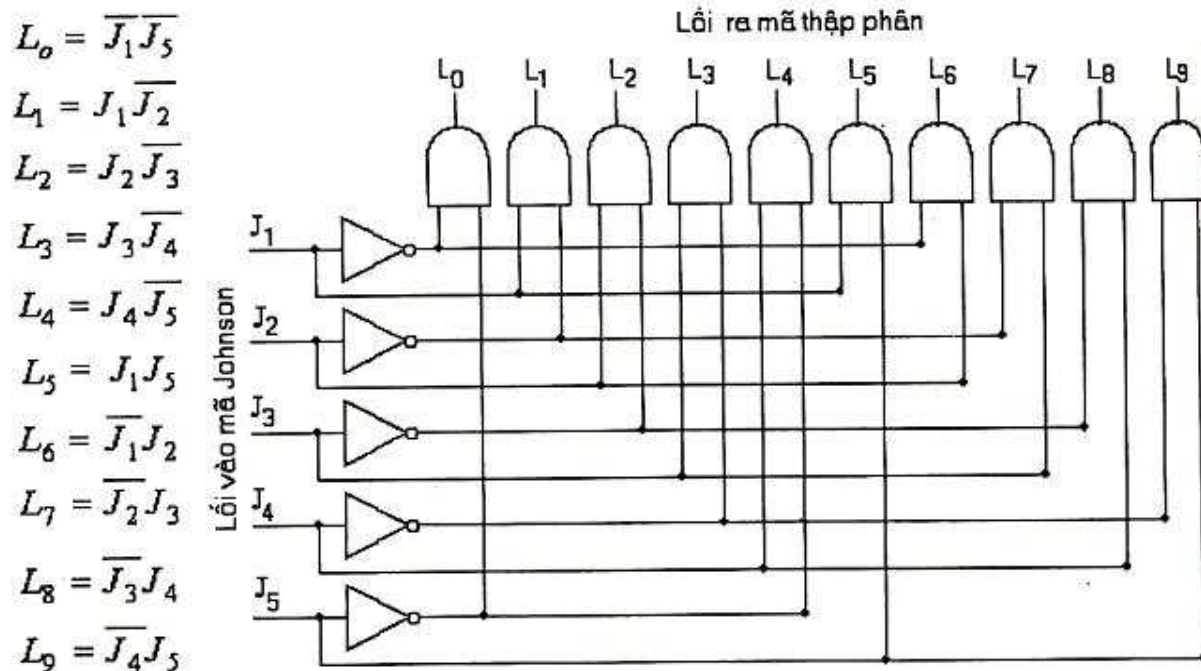
Bảng 4.19: Bảng chân lý của bộ giải mã từ mã Johnson ra thập phân

Số thập phân	J_1	J_2	J_3	J_4	J_5	L_0	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9
0	<u>0</u>	0	0	0	<u>0</u>	1	0	0	0	0	0	0	0	0	0
1	<u>1</u>	<u>0</u>	0	0	0	0	1	0	0	0	0	0	0	0	0
2	1	<u>1</u>	<u>0</u>	0	0	0	0	1	0	0	0	0	0	0	0
3	1	1	<u>1</u>	<u>0</u>	0	0	0	0	1	0	0	0	0	0	0
4	1	1	1	<u>1</u>	<u>0</u>	0	0	0	0	1	0	0	0	0	0
5	<u>1</u>	1	1	1	<u>1</u>	0	0	0	0	0	1	0	0	0	0
6	<u>0</u>	<u>1</u>	1	1	1	0	0	0	0	0	0	1	0	0	0
7	0	<u>0</u>	<u>1</u>	1	1	0	0	0	0	0	0	0	1	0	0
8	0	0	<u>0</u>	<u>1</u>	1	0	0	0	0	0	0	0	0	1	0
9	0	0	0	<u>0</u>	<u>1</u>	0	0	0	0	0	0	0	0	0	1

Từ bảng chân lý 4.19 vận dụng đại số Boole ta có thể viết được các phương trình logic lối ra của bộ giải mã. Cũng theo phương pháp đã nêu ở mục 4.1 ta viết phương trình lối ra là tổng các tích của các biến lối vào có giá trị bằng 1, nhưng ở đây chúng ta cần lưu ý một điểm đặc biệt đối với mã Johnson, đáng lẽ ta phải thực hiện phép nhân đối với tổ hợp 5 biến ta chỉ cần thực hiện phép nhân đối với tổ hợp 2 biến đặc trưng cho mỗi số thập phân. Trên bảng 4.16 các con số được gạch chân là tổ hợp biến đặc trưng cho từng con số.

Ví dụ: $J_1 = 0, J_5 = 0$ là tổ hợp biến đặc trưng cho số "0", nó không xuất hiện ở các con số khác. $J_1 = 1, J_5 = 1$ là tổ hợp biến đặc trưng cho số "1", chỉ có số "1" mới có tổ hợp biến này. Như vậy, nếu dùng các tổ hợp biến đặc trưng ta có thể viết phương trình lối ra đơn giản hơn nhiều. Ta tìm được các phương trình lối ra của bộ giải mã.

Từ các phương trình lối ra ta có thể xây dựng sơ đồ logic của bộ giải mã hình 4.44.

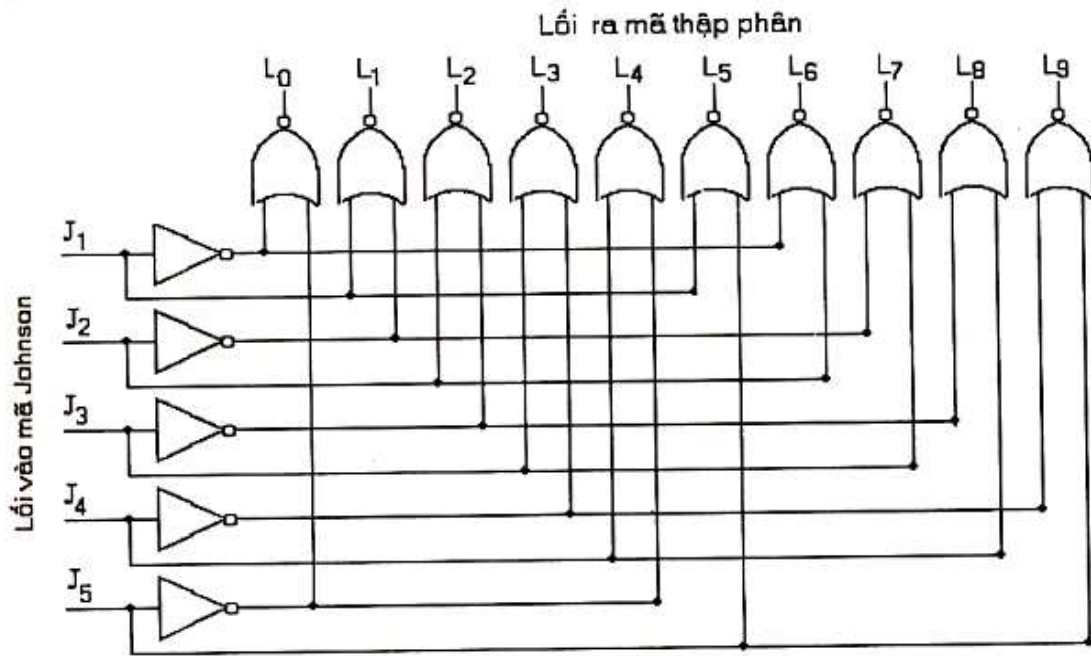


Hình 4.44: Sơ đồ logic bộ giải mã Johnson ra thập phân

Ta có thể tạo được bộ giải mã từ các mạch NOR hai lối vào và NOT. Muốn vậy ta viết hàm logic cho các L_i lối ra, làm phép đảo 2 lần rồi dùng định lý De Morgan chuyển các phép nhân logic thành các phép cộng logic. Ta được:

$L_0 = \overline{J_1 J_5} = \overline{J_1 + J_5}$	$L_5 = J_1 J_5 = \overline{\overline{J_1 + J_5}}$
$L_1 = J_1 \overline{J_2} = \overline{\overline{J_1 + J_2}}$	$L_6 = \overline{J_1 J_2} = \overline{J_1 + J_2}$
$L_2 = J_2 \overline{J_3} = \overline{\overline{J_2 + J_3}}$	$L_7 = \overline{J_2 J_3} = \overline{J_2 + J_3}$
$L_3 = J_3 \overline{J_4} = \overline{\overline{J_3 + J_4}}$	$L_8 = \overline{J_3 J_4} = \overline{J_3 + J_4}$
$L_4 = J_4 \overline{J_5} = \overline{\overline{J_4 + J_5}}$	$L_9 = \overline{J_4 J_5} = \overline{J_4 + J_5}$

Từ hàm logic này ta có thể xây dựng bộ giải mã Johnson ra thập phân dùng 10 mạch NOR 2 lối vào và 5 mạch NOT. Sơ đồ logic của bộ giải mã này được trình trên hình 4.45.



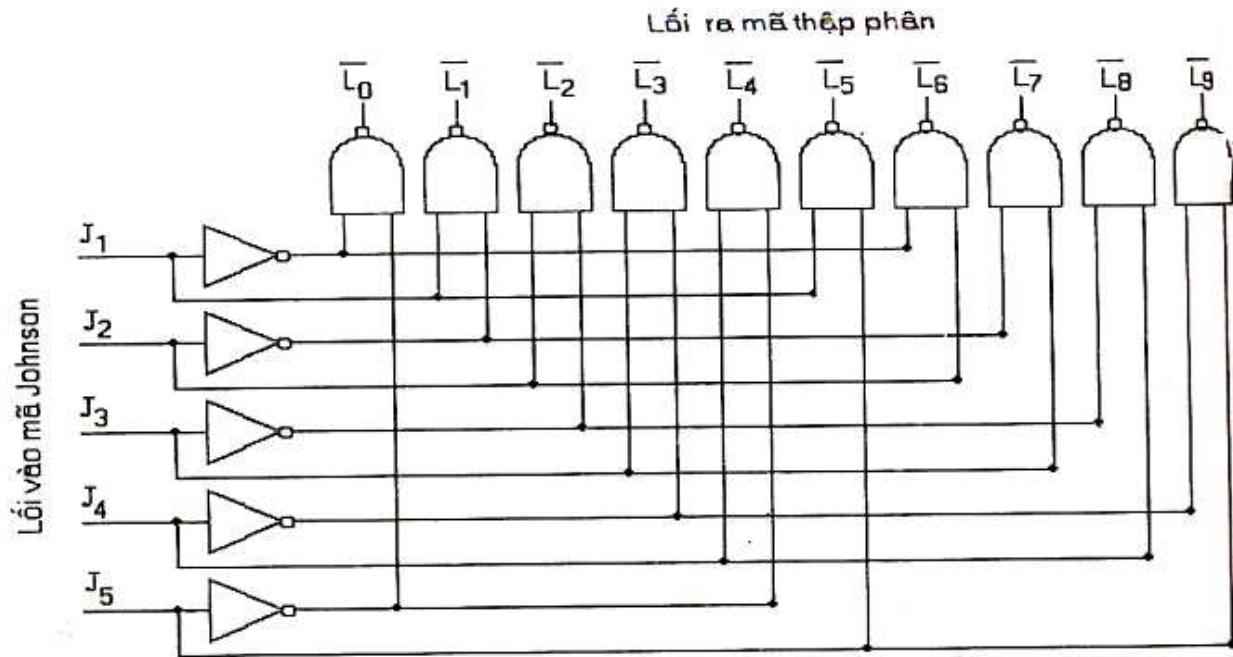
Hình 4.45: Sơ đồ logic của bộ giải mã Johnson ra thập phân dùng NOR và NOT

Trong trường hợp dùng mạch NAND và NOT làm bộ giải mã. Ta có bảng chân lý của bộ giải mã được nêu trên bảng 4.20. Bộ giải mã này có lối ra hoạt động ở mức logic thấp.

Sơ đồ logic của bộ giải mã được trình trên hình 4.46 .

Số thập phân	J_1	J_2	J_3	J_4	J_5	$\overline{L_0}$	$\overline{L_1}$	$\overline{L_2}$	$\overline{L_3}$	$\overline{L_4}$	$\overline{L_5}$	$\overline{L_6}$	$\overline{L_7}$	$\overline{L_8}$	$\overline{L_9}$
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1
2	1	1	0	0	0	1	1	0	1	1	1	1	1	1	1
3	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1
4	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
6	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1
7	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1
8	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1
9	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0

Bảng 4.20: Bảng chuyển mã Johnson sang mã 10 dùng NAND



Hình 4.46: Sơ đồ mạch giải mã Johnson sang mã 10

4.6. CÁC BÀI TẬP THỰC HÀNH

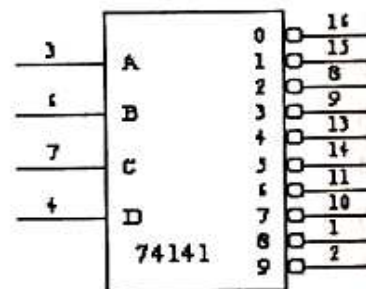
1/ Nghiên cứu hoạt động của bộ biến đổi mã nhị phân 4 bit sang mã Gray 4 bit. Kiểm tra bảng chân lý của bộ chuyển đổi mã. Sơ đồ mắc mạch của bộ chuyển đổi mã dùng vi mạch XOR hai lối vào 74LS86.

2/ Nghiên cứu hoạt động của bộ biến đổi mã nhị phân sang mã bù nhị phân. Dùng mạch NAND 74LS00 tạo thành bộ biến đổi mã nhị phân 4 bit sang mã bù nhị phân 4 bit.

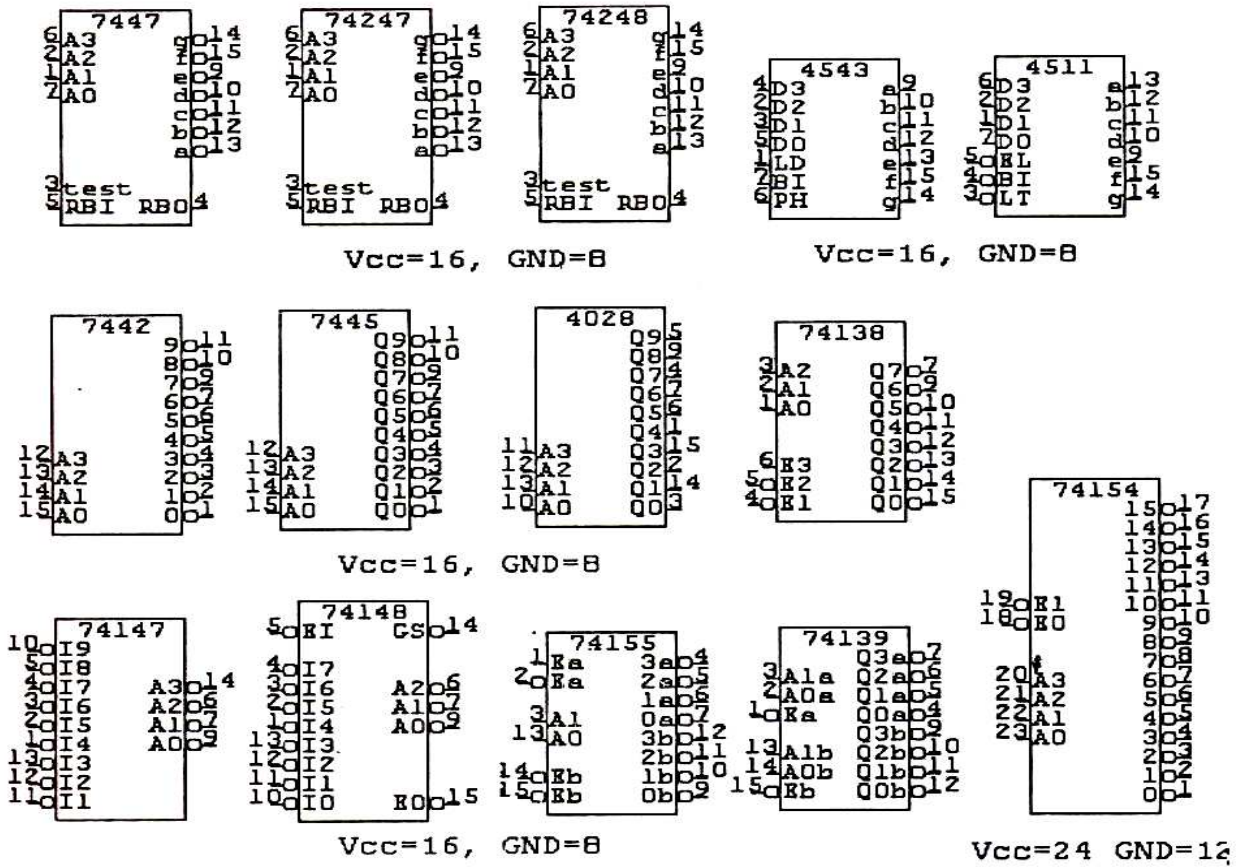
3/ Cho hai loại vi mạch 74LS00 (NAND 2 lối vào) và 74LS02 (NOR 2 lối vào). Trình sơ đồ mắc mạch chuyển đổi mã thập phân sang mã BCD dùng hai loại vi mạch trên. Sơ đồ tối ưu dùng hết 2 mạch 7402 và 1 mạch 7400

4/ Xác lập bảng sự thật của vi mạch 74141 có ký hiệu như hình 4.47 ở bên.

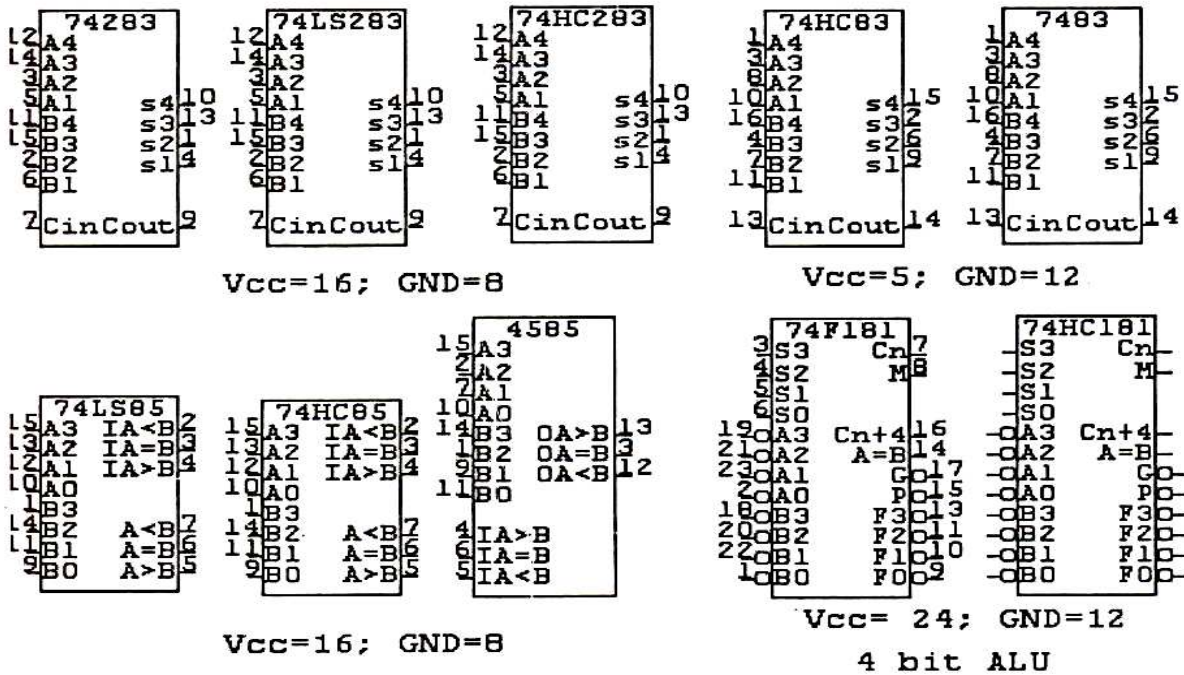
Trên hình 48a giới thiệu kí hiệu của các mạch logic tổ hợp dùng để biến đổi mã, giải mã. Các IC dùng để tính toán, so sánh số học, ALU được trình bày trên hình 4.48b. Căn cứ vào các kí hiệu chân của các vi mạch này ta có thể xác lập bảng chân lý, chức năng của các vi mạch này.



Hình 4.47: Kí hiệu của IC74141



Hình 4.48a: Các IC dùng để biến đổi mã và giải mã

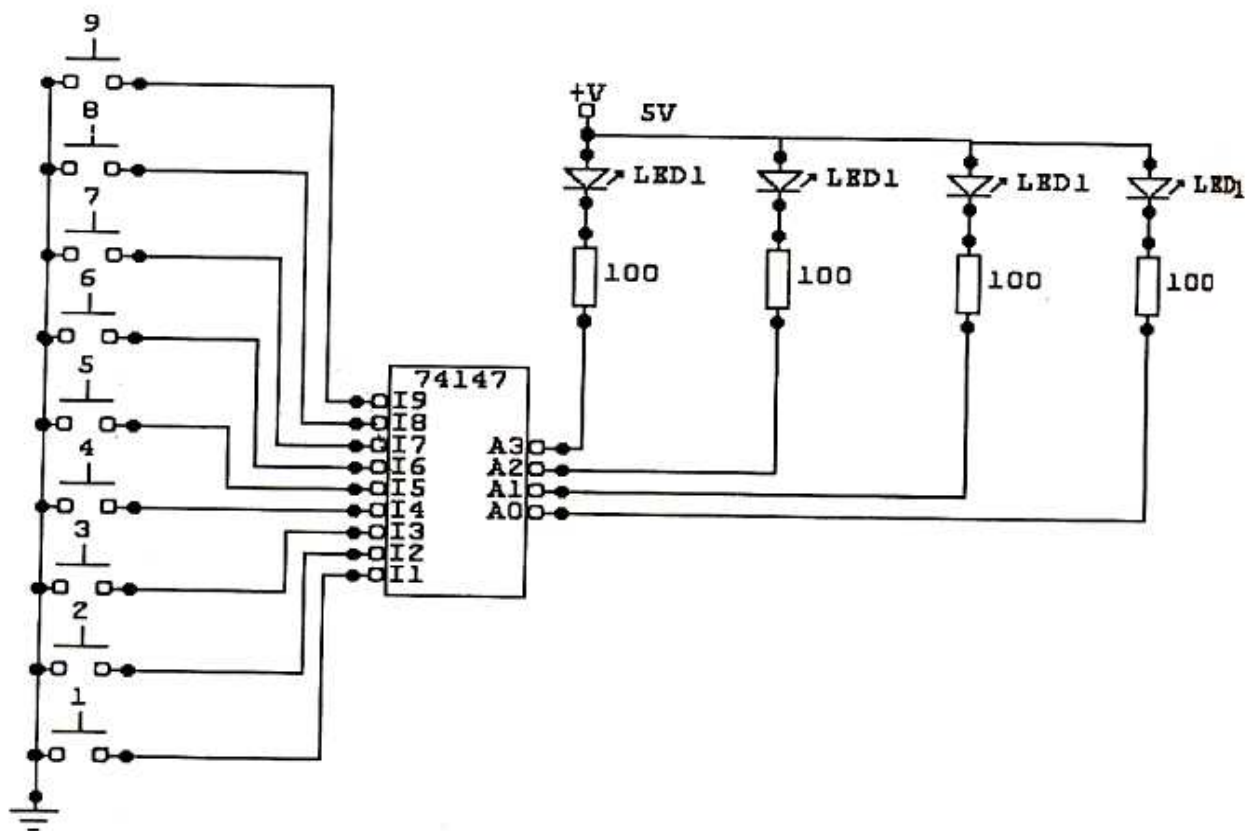


Hình 4.48b: Các IC dùng cho tính toán, so sánh số học

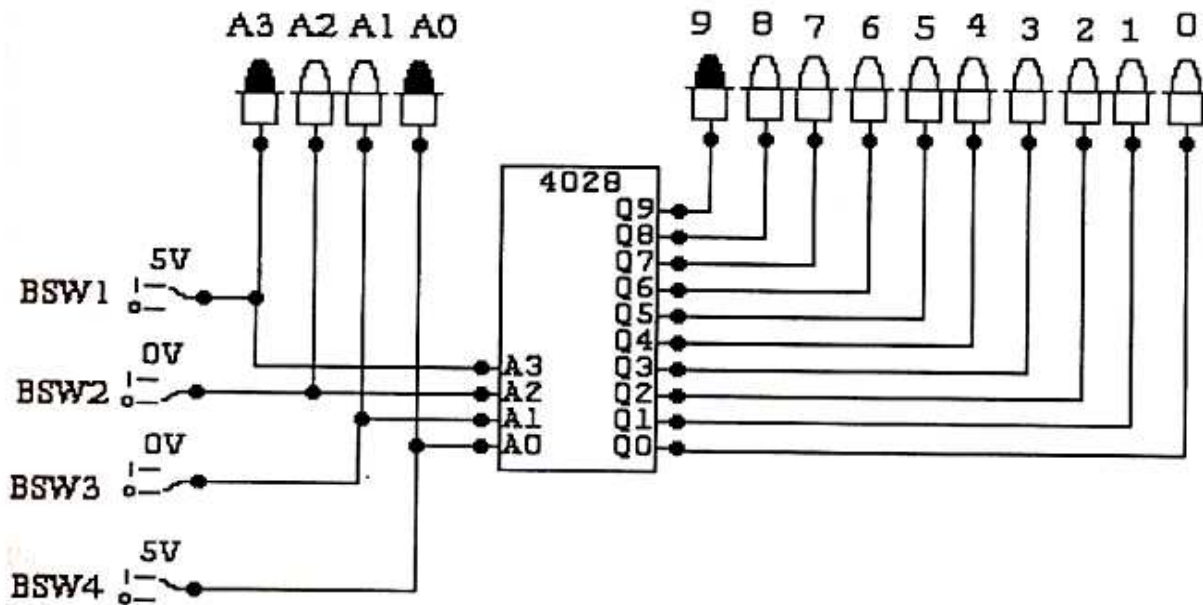
Ta cũng có thể xác lập bảng chân lý và kiểm tra chức năng các vi mạch này bằng thực nghiệm. Sau đây xin giới thiệu một số panen thí nghiệm đơn giản bạn đọc có thể tự làm để tìm hiểu chức năng của các vi mạch này. Các panen thí nghiệm này được thiết kế để có thể làm thí nghiệm mô phỏng trên máy vi tính với sự trợ giúp của chương trình phần mềm Circuit Maker.

5/ Trên hình 4.49 là một panen thí nghiệm dùng để tìm hiểu chức năng của vi mạch 74147. Trong sơ đồ này đèn LED1 chỉ phát sáng khi lối ra của vi mạch ở mức thấp. Các công tắc nhấn được nối với các lối vào của vi mạch, khi không ấn công tắc thì các lối vào để hở chúng đang ở mức logic 1. Khi ấn công tắc thì lối vào được nối đất, lối vào ở mức logic 0. Hãy dùng panen thí nghiệm này để xác định bảng chân lý của vi mạch và cho biết chức năng của vi mạch này.

6/ Trên hình 4.50 là một panen thí nghiệm dùng để tìm hiểu chức năng của vi mạch CD4028, BSW1, BSW2, BSW3, BSW4 là các công tắc bit, chúng được dùng để đặt mức logic 1 hoặc logic 0 ở các lối vào A3, A2, A1, A0 của vi mạch.



Hình 4.49

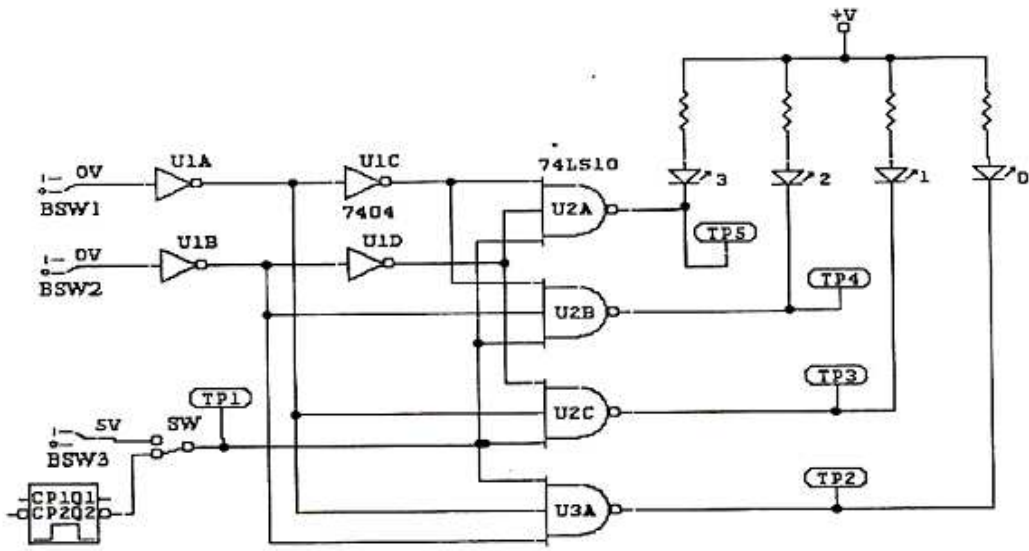


Hình 4.50

Các đèn LED được dùng để chỉ thị mức logic ở lối vào hoặc lối ra của các mạch logic. Đèn LED sáng ứng với mức logic 1, trong trường hợp này khác với sơ đồ thí nghiệm hình 4.49 các đèn LED chỉ sáng khi lối ra các vi mạch ở mức logic thấp.

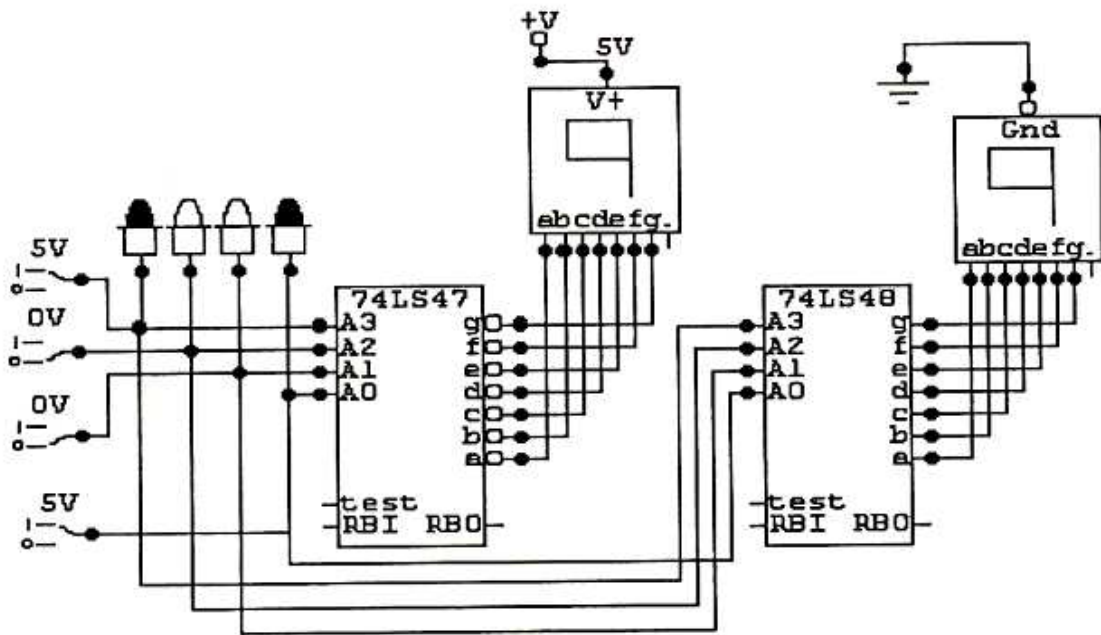
Dùng panen thí nghiệm hình 4.50 xác lập bảng chân lý của vi mạch 4028 và cho biết chức năng của vi mạch này.

7/ Trên hình 4.51 là một panen thí nghiệm dùng để tìm hiểu chức năng của vi mạch, trong đó khối có kí hiệu CP1Q1, CP2Q2 là một môđun tạo xung nhịp. Khi chuyển mạch SW chuyển về vị trí nối với môđun này thì ở TP1 ta sẽ quan sát thấy các xung vuông liên tục được đưa vào nối chung của các cửa NAND. Hãy dùng panen thí nghiệm hình 4.51 khảo sát hoạt động của sơ đồ mạch logic hình 4.51. Hãy cho biết muốn dùng mạch làm bộ phân kênh từ một đường vào ra 4 đường thì phải làm như thế nào? Muốn mạch trở thành bộ giải mã từ nhị phân ra thập phân thì làm như thế nào?



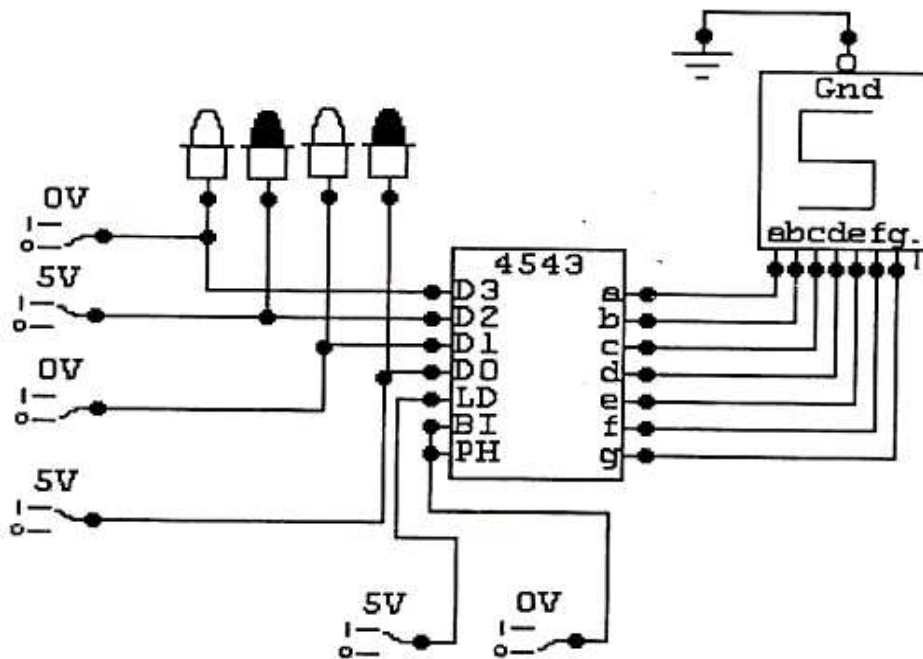
Hình 4.51

8/ Trên hình 4.52 là một panen thí nghiệm dùng để tìm hiểu chức năng của vi mạch giải mã 7 đoạn 74LS47 và LS7448, hãy kiểm tra hoạt động của 2 vi mạch này bằng thực nghiệm.



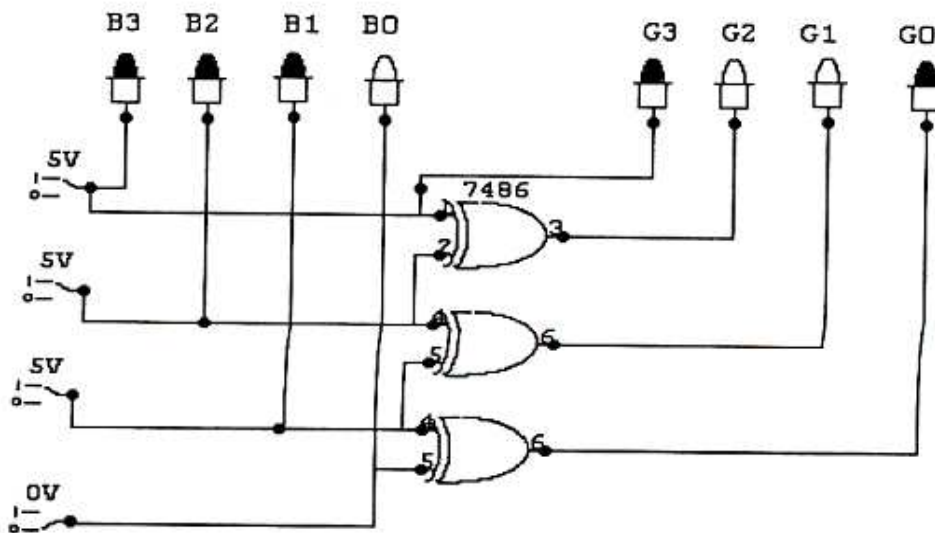
Hình 4.52

9/ Trên hình 4.53 là một panen thí nghiệm dùng để tìm hiểu chức năng của vi mạch giải mã 7 đoạn 4543 thuộc họ logic CMOS. Hãy kiểm tra sự hiển thị của bộ giải mã này bằng thực nghiệm.



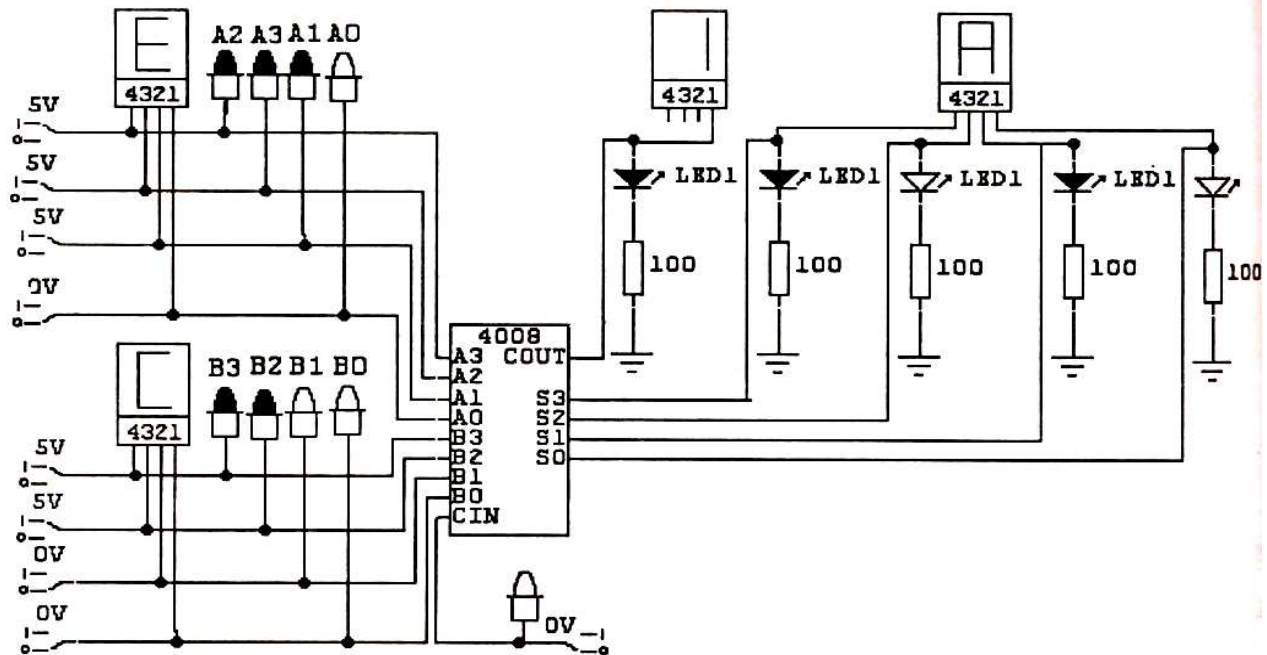
Hình 4.53

10/ Trên hình 4.54 là một panen thí nghiệm dùng để tìm hiểu chức năng của vi mạch 7486. Mạch được dùng làm thành bộ biến đổi mã từ mã nhị phân 4 bit ra mã Gray 4 bit. Hãy kiểm tra hoạt động của bộ giải mã này.



Hình 4.54

11/ Trên hình 4.55 là một panen thí nghiệm dùng để tìm hiểu chức năng của vi mạch cộng nhị phân 4 bit. Hãy kiểm tra hoạt động của bộ cộng này.



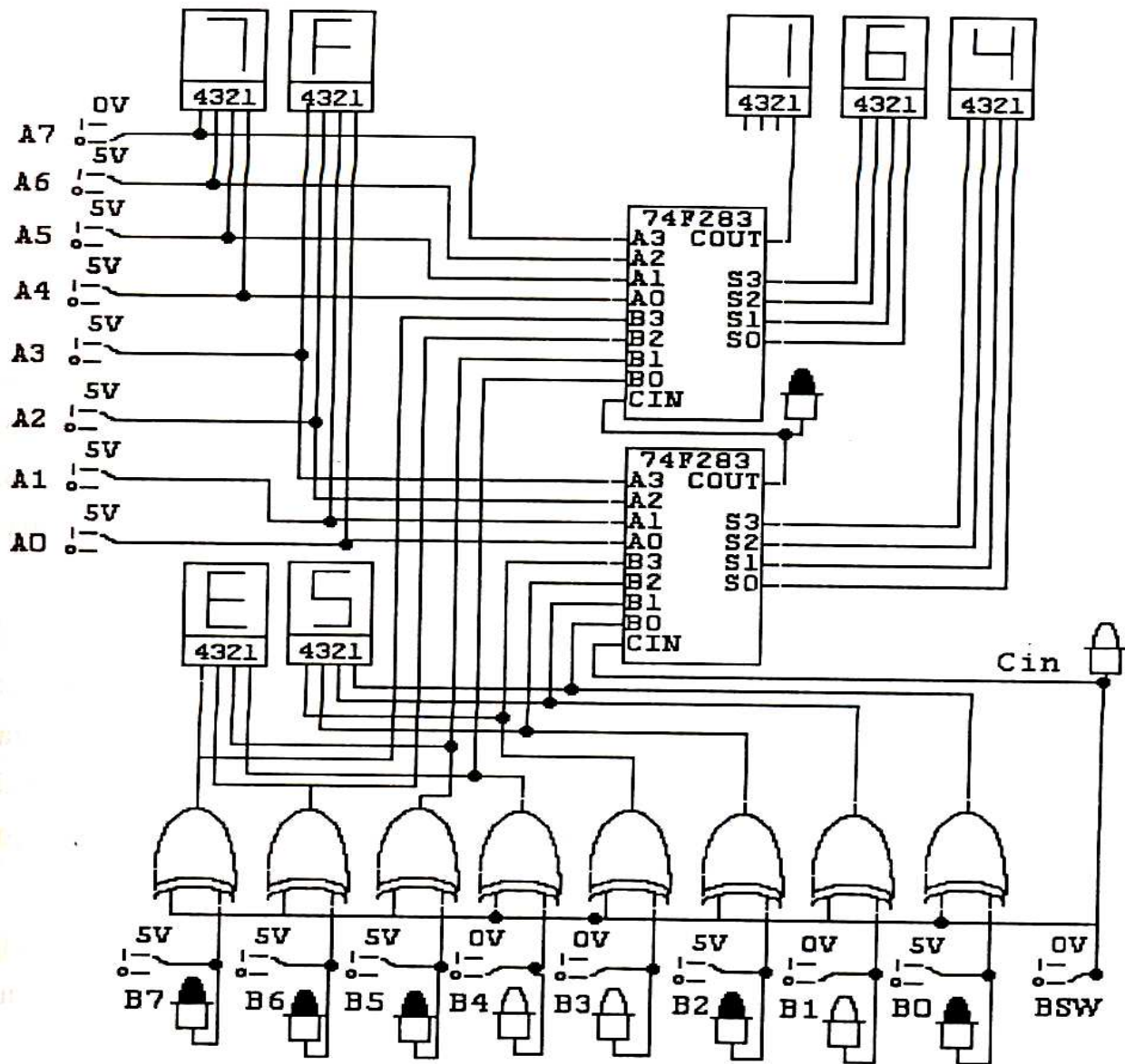
Hình 4.55

11/ Trên hình 4.56 là một panen thí nghiệm dùng để thực hiện các phép tính cộng, trừ nhị phân 8 bit dùng vi mạch 74F283 và vi mạch 7486.

Trong sơ đồ này các công tắc bit: A0, A1, A2, A3,....., A6, A7 được dùng để đặt dữ liệu cho các bit của số nhị phân A. Các công tắc bit: B1, B2, B3,....., B6, B7 được dùng để đặt dữ liệu cho các bit nhị phân B.

Chuyển mạch BSW được dùng để điều khiển chọn phép tính, khi chuyển mạch này đặt ở mức logic 1 thì mạch thực hiện phép tính trừ: $A+(-B)$, khi chuyển mạch đặt ở mức logic 0 thì mạch thực hiện phép tính cộng: $A+B$.

Hãy kiểm tra và giải thích hoạt động của bộ cộng này.



Hinh 4.56

Chương V

TRIGƠ

Trigơ trong tiếng Anh gọi là Flip - Flop viết tắt là FF. Nó là một phần tử nhớ có 2 trạng thái cân bằng ổn định tương ứng với 2 mức logic 0 và 1. Dưới tác dụng của các tín hiệu điều khiển ở lối vào, trigơ có thể chuyển về một trong hai trạng thái cân bằng, và giữ nguyên trạng thái đó chừng nào chưa có tín hiệu điều khiển làm thay đổi trạng thái của nó.

Trạng thái tiếp theo của trigơ phụ thuộc không những vào tín hiệu ở lối vào mà còn phụ thuộc vào cả trạng thái đang hiện hành của nó. Đang chạy, nếu ngừng các tín hiệu điều khiển ở lối vào nó vẫn có khả năng giữ trạng thái hiện hành của mình trong một thời gian dài, chừng nào mà nguồn điện nuôi mạch trigơ không bị ngắt thì thông tin dưới dạng nhị phân lưu giữ trong trigơ vẫn được duy trì. Như vậy nó được sử dụng như một phần tử nhớ.

Trigơ được tạo thành từ các phần tử logic cơ bản. Có nhiều loại trigơ, trong mục này chúng ta chỉ nghiên cứu một số loại trigơ điển hình được dùng nhiều trong kỹ thuật số, đó là các trigơ RS, trigơ JK và trigơ D.

5.1. TRIGƠ RS

Trigơ RS là một phần tử nhớ cơ bản có hai trạng thái cân bằng bền vững, nó có 2 lối vào R, S và 2 lối ra Q và \bar{Q} . Hai lối ra này bao giờ cũng ở trong trạng thái ngược nhau.

Ký hiệu logic của nó được trình bày trên hình 5.1. Hoạt động của trigơ tuân theo bảng chân lý cho trên bảng 5.1. Trong đó lối vào S (Set) là lối vào đặt, R (Reset) là lối vào xoá.

Chỉ số n chỉ trạng thái khởi đầu, n+1 chỉ trạng thái lúc kết thúc.

Bảng 5.1: Bảng chân lý của trigơ RS

S_n	R_n	Q_{n+1}	Một hoạt động
0	0	Q_n	Nhớ
1	0	1	Xác lập
0	1	0	Xoá
1	1	Không cho phép	Cấm dùng

Hình 5.1: Ký hiệu trigơ RS

Từ bảng chân lý ta có phương trình logic:

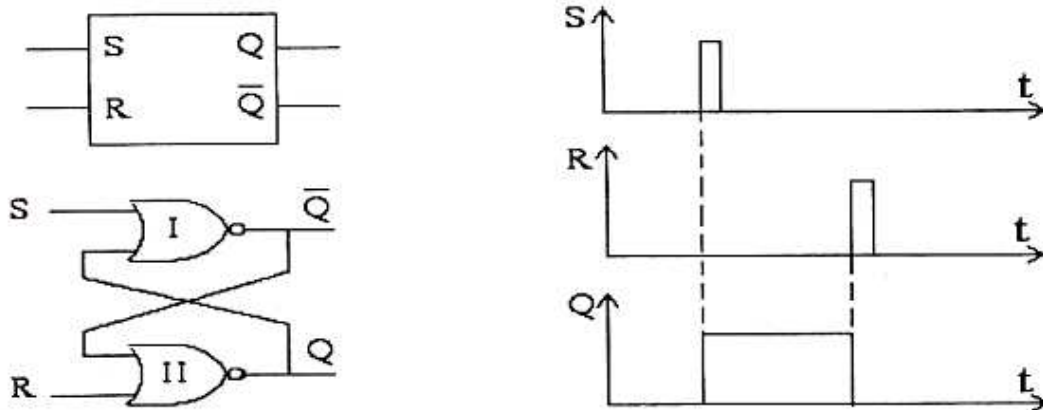
$$Q_{n+1} = \bar{R}_n Q_n + S_n$$

Phương trình trên cho ta thấy: lối ra không những là hàm số của lối vào mà còn phụ thuộc vào trạng thái trước đó của lối ra.

Từ hàm logic trên, ta xây dựng sơ đồ logic của trigơ RS từ mạch NOR, lối vào tác động ở mức cao. Để tránh trường hợp $S=R=1$, xung đưa vào để đặt và xoá là các xung kim dương.

Sơ đồ logic và giản đồ xung biểu diễn trạng thái của trigơ theo các xung đặt S và xung xoá R được trình bày trên hình 5.2.

Chúng ta sẽ khảo sát hoạt động của trigơ RS ứng với các tổ hợp biến khác nhau nêu trong bảng 5.1.



Hình 5.2: Sơ đồ logic và giản đồ xung của trigơ RS

Từ sơ đồ logic hình 5.2 chúng ta thấy:

a/ Khi $S = 0$, $R = 0$: Lối ra vẫn giữ trạng thái cũ Q_n : Trạng thái nhớ.

Nếu trạng thái ban đầu $Q_n = 0$, cả hai lối vào của cửa logic NOR I đều bằng 0 nên lối ra của nó là $\overline{Q} = 1$ và lối ra của NOR II là $Q = 0$ trùng với trạng thái ban đầu trước đó. Tức là nó đã nhớ trạng thái $Q = 0$. Nếu trạng thái ban đầu $Q_n = 1$ thì lối ra $\overline{Q} = 0$ và lối ra của NOR II là $Q = 1$ trùng với trạng thái ban đầu trước đó. Tức là nó đã nhớ trạng thái $Q = 1$.

b/ Khi $S = 1, R = 0$: Lối ra $Q_{n+1} = 1$: Trạng thái xác lập.

Lúc khởi đầu $S_n = 1, R_n = 0$ lối ra NOR I là $\overline{Q} = 0$, cả hai lối vào của NOR II đều bằng 0 nên lối ra Q của nó ở trạng thái kết thúc $Q_{n+1} = 1$: trạng thái xác lập.

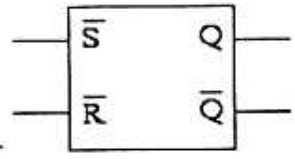
c/ Khi $S = 0, R = 1$: Lối ra $Q_{n+1} = 0$: Trạng thái xoá.

Nếu lúc khởi đầu $S_n = 0, R_n = 1$, mạch logic NOR II có một lối vào $R_n = 1$ nên lối ra của nó $Q = 0$, cả hai lối vào của NOR I đều bằng 0 nên lối ra $\overline{Q} = 1$, vậy trạng thái kết thúc của trigơ $Q_{n+1} = 0$ ứng với trạng thái xoá của trigơ.

d/ Trạng thái cấm không được dùng.

Ta hãy khảo sát trường hợp cấm không được dùng đối với trigơ RS ứng với tổ hợp biến lối vào S và lối vào R đều bằng 1. Nếu lúc khởi đầu $S_n = 1, R_n = 1$, cả hai lối ra của trigơ Q và \overline{Q} đều bằng 0, một tình huống như vậy không được phép xảy ra vì trái với luật logic là Q và \overline{Q} phải có giá trị ngược nhau, chính vì vậy đối với trigơ RS tuyệt đối không được để xảy ra trường hợp cả lối vào S , và R đồng thời ở mức 1, chúng ta phải tạo hai xung đặt S và xung xoá R là xung kim dương có độ kéo dài nhỏ để chánh khỏi trường hợp xung S và xung R cùng ở mức cao.

Bảng 5.2: Bảng chân lý của trigơ RS lối vào tác động thấp



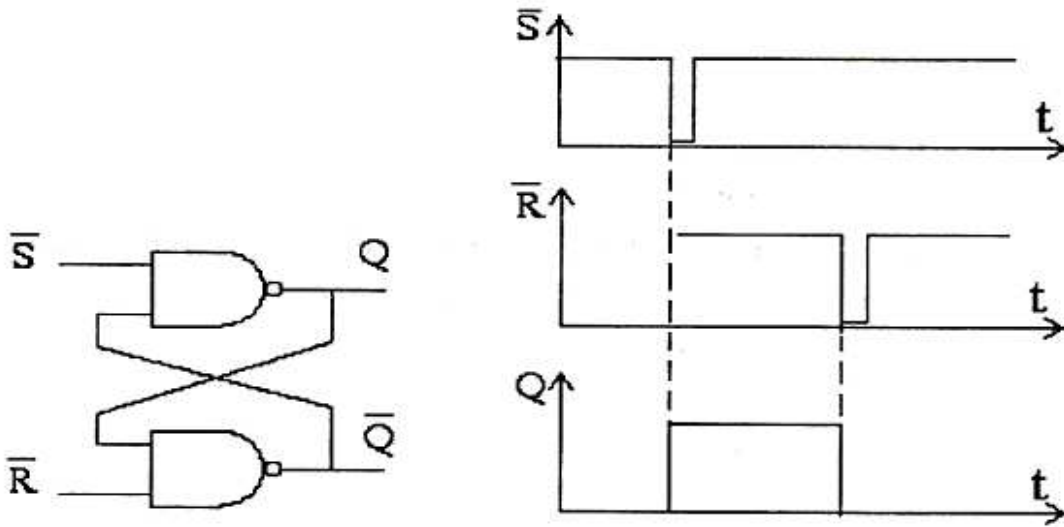
\overline{S}_n	\overline{R}_n	Q_{n+1}	Một hoạt động
1	1	Q_n	Nhớ
1	0	0	Xác lập
0	1	1	Xoá
0	0	Không cho phép	Cấm dùng

Hình 5.3: Ký hiệu logic

Ta cũng có thể xây dựng trigơ RS không đồng bộ tác động bằng mức logic thấp từ hai phần tử logic NAND, ký hiệu logic được nêu trên hình 5.3 Bảng chân lý 5.2 diễn tả hoạt động của trigơ này.

Chỉ số n trong bảng chân lý 5.2 chỉ trạng thái khởi đầu, $n+1$ chỉ trạng thái lúc kết thúc.

Sơ đồ logic và giản đồ xung minh họa hoạt động của trigơ được cho trên hình 5.4.



Hình 5.4: Sơ đồ logic và giản đồ xung

Hoạt động của sơ đồ hình 5.4 cũng tương tự như sơ đồ hình 5.2. Điểm khác biệt ở đây là sơ đồ hình 5.4 hoạt động với xung tác động lối vào ở mức thấp.

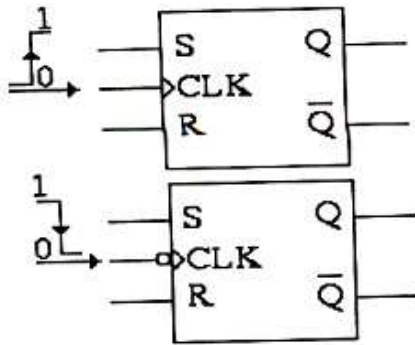
5.2. TRIGƠ ĐỒNG BỘ RST

Trigơ RS nhạy cảm với tác động của đầu vào R và S, trạng thái của trigơ sẽ không ổn định khi lối vào chịu ảnh hưởng của nhiễu. Để khắc phục nhược điểm trên người ta dùng trigơ RST có thêm một đầu vào xung nhịp CLOCK điều khiển chung cho cả hai lối vào. Chỉ khi nào có tác động của xung nhịp này thì trigơ mới chuyển trạng thái theo tác động của R hay S. Kí hiệu của trigơ RST cho trên hình 5.5. Bảng 5.3 là bảng chân lý của trigơ RST.

Trên bảng chân lý của trigơ RST: t_n chỉ thời điểm trước khi có xung nhịp tác dụng, t_{n+1} là thời điểm sau khi có một xung nhịp tác dụng. Chỉ số n chỉ trạng thái trước khi có xung nhịp tác dụng, $n+1$ chỉ trạng thái sau khi có xung nhịp tác dụng.

Bảng 5.3: Bảng chân lý của trigơ RST

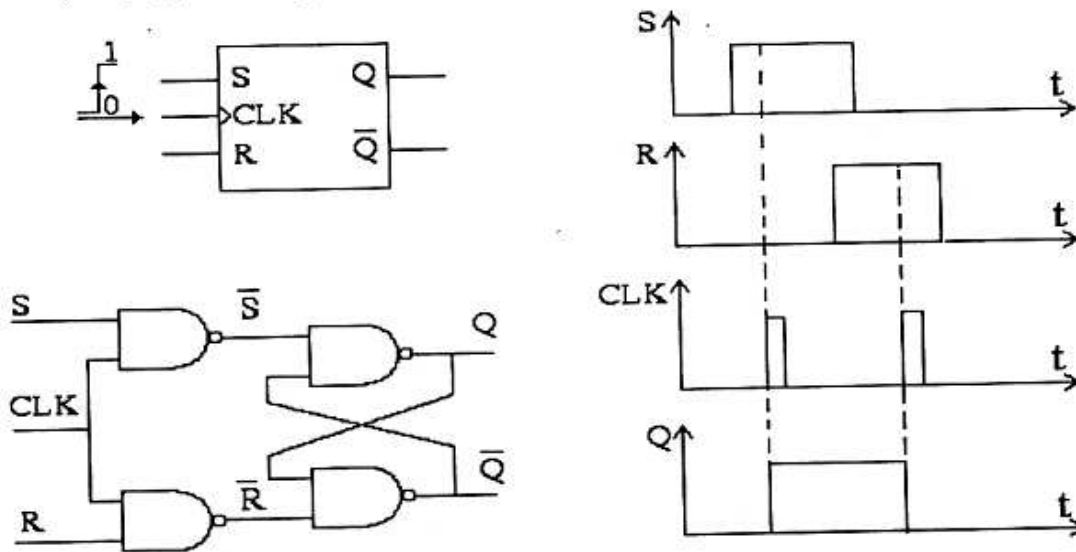
t_n		t_{n+1}
S_n	R_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	Không cho phép



Hình 5.5: Ký hiệu logic của trigơ RST

Lối vào xung nhịp (Clock Pulse được ký hiệu là CLK) điều khiển chung cho cả 2 lối vào S và R. Các giá trị lối vào R và S chỉ làm cho trigơ hoạt động khi có xung nhịp CLK tác dụng.

Sự chuyển trạng thái của trigơ RST và tất cả các loại trigơ đồng bộ khác xảy ra có thể vào thời điểm sau khi xung nhịp đã chuyển từ mức logic 0 lên mức logic 1 hoặc sau khi xung nhịp đã chuyển từ mức logic 1 về mức logic 0 (hình 5.5). Tùy theo cấu trúc cụ thể của từng loại trigơ, khi dùng ta cần chú ý đến ký hiệu của trigơ, nếu trên đó có ghi vòng tròn ở lối vào chân $\overline{\text{CLK}}$ hoặc trên chữ CLK trên có gạch ngang dấu hiệu của hàm phủ định ($\overline{\text{CLK}}$) thì trạng thái lối ra của trigơ được xác lập khi xung nhịp chuyển từ mức logic 1 về mức logic 0. Sơ đồ logic của trigơ RST và giản đồ xung diễn tả trạng thái hoạt động của trigơ được trình bày trên hình 5.6.



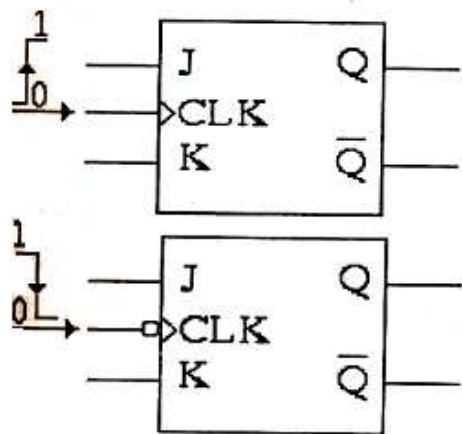
Hình 5.6: Sơ đồ logic của trigơ RST và giản đồ xung diễn tả trạng thái hoạt động của trigơ được xác lập sau khi xung nhịp chuyển từ mức logic thấp lên mức logic cao

5.3. TRIGƠ JK

Trigơ RST có một tổ hợp biến cấm dùng là $S = R = 1$, trạng thái của trigơ nay sẽ không được xác định nếu gặp phải tổ hợp này. Ta có thể khắc phục tình trạng này bằng cách dùng hai mạch phản hồi từ Q về R và \bar{Q} về S sẽ tạo được trigơ JK. Ký hiệu của trigơ JK cho trên hình 5.7. Bảng chân lý của nó cho trên bảng 5.4.

Trigơ JK giống trigơ RS : J tương ứng với S, K tương ứng với R. Nhưng khác với trigơ RS, trigơ JK không có trạng thái cấm mà khi $J = K = 1$ lối ra đổi trạng thái ngược lại với trạng thái cũ trước đó (TOGGLE), t_n thời điểm trước khi có xung nhịp; t_{n+1} thời điểm sau khi có một xung nhịp tác dụng vào trigơ. Chỉ số n chỉ trạng thái của trigơ trước khi có xung nhịp tác dụng, n+1 trạng thái sau khi có một xung nhịp tác dụng vào trigơ.

Bảng 5.4: Bảng chân lý của trigơ JK



t_n		t_{n+1}
J_n	K_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	\bar{Q}_n

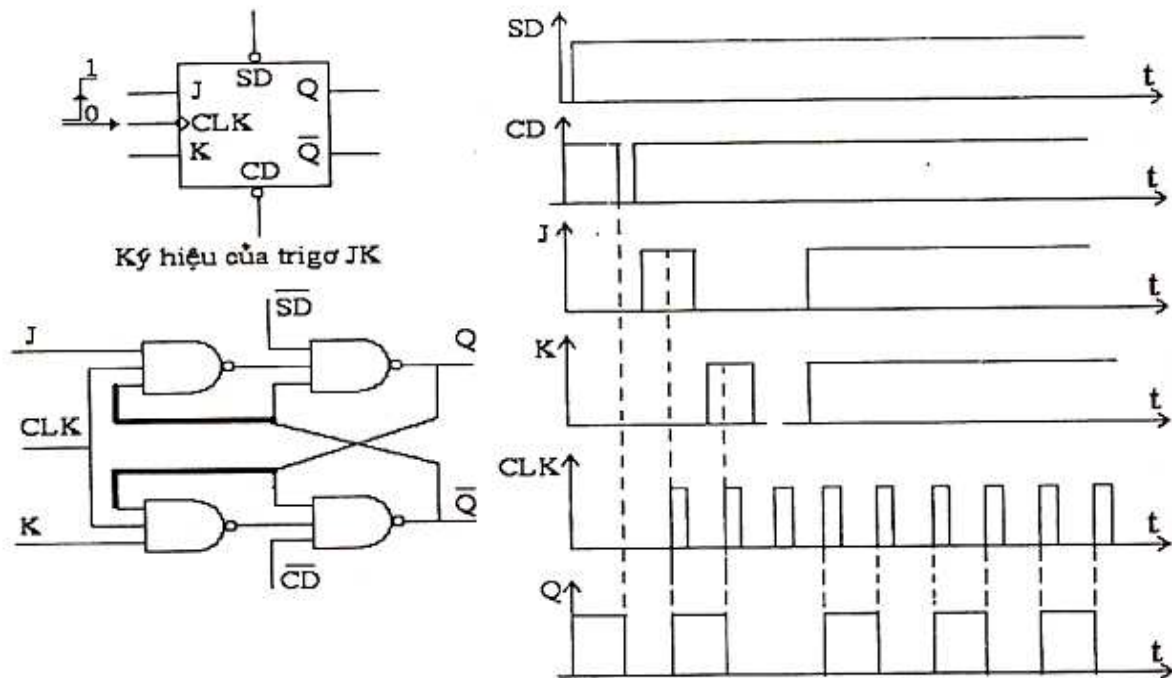
Hình 5.7: Ký hiệu của trigơ JK

Phương trình logic của trigơ JK:

$$Q_{n+1} = J_n \bar{Q}_n + \bar{K}_n Q_n$$

Sơ đồ logic của trigơ JK cho trên hình 5.7. Hình 5.8 là giản đồ xung mô tả các trạng thái hoạt động của trigơ này

Như ta đã thấy trên sơ đồ logic cấu trúc của trigơ JK so với trigơ RST chỉ khác nhau có hai đường phản hồi được tô đậm, nếu bỏ hai đường này đi trigơ JK sẽ trở thành trigơ RST lối vào, J,K sẽ trở thành S,R. Trong công nghệ vi điện tử người ta đã chế tạo các mạch trigơ RST có hai lối vào S và hai lối vào R. Người dùng muốn dùng như một trigơ JK thì chỉ việc đấu thêm hai đường phản hồi ở mạch ngoài nối từ Q về R và \bar{Q} về S.



Hình 5.8: Giải đồ thời gian mô tả các trạng thái hoạt động của trigơ JK

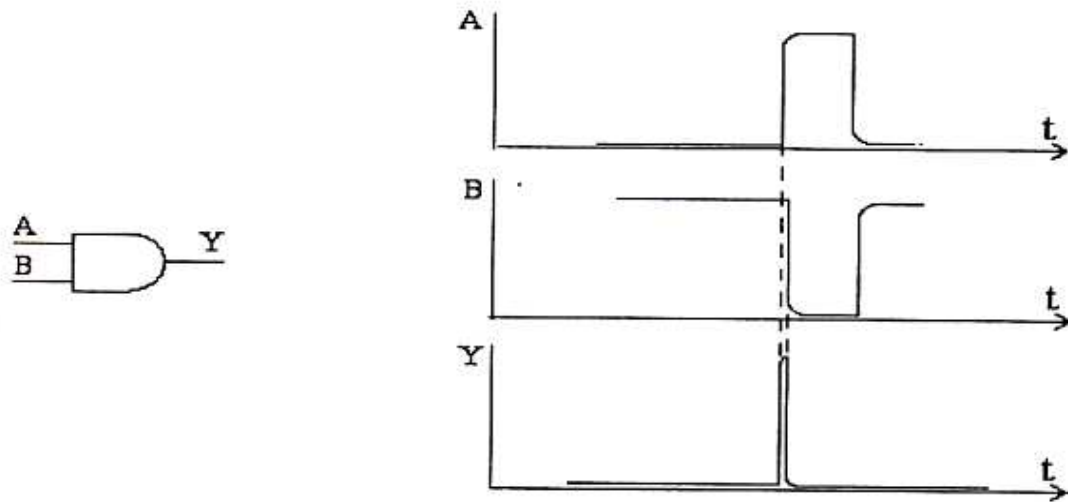
Các trigơ JK trong thực tế ngoài các lối vào J,K hoạt động đồng bộ với lối vào xung nhịp CLK, trigơ còn có lối vào không đồng bộ là lối vào xoá CD (Clear Data) và lối vào đặt SD (Set Data). Hai lối vào này hoạt động độc lập không phụ thuộc vào xung nhịp và các lối vào J,K. Trạng thái ra của trigơ JK phụ thuộc vào các mức điện áp ở lối vào không đồng bộ này cũng giống như là trigơ RS.

Để trigơ hoạt động được ở chế độ đồng bộ, hai lối vào không đồng bộ đặt, xoá phải để đúng mức điện áp, nếu trên ký hiệu của trigơ các lối vào không đồng bộ có vòng tròn nhỏ hoặc trên chữ xoá, đặt có gạch ngang ở trên ($\overline{SD}, \overline{RD}$) thì các chân này phải để ở mức cao.

Khi J,K để ở mức cao, cứ mỗi lần có xung nhịp tác dụng trigơ lại chuyển trạng thái một lần, sau hai nhịp tác dụng trigơ lại chuyển về trạng thái cũ (hình 5.8), người ta dùng trường hợp này để tạo thành bộ đếm nhị phân từ các trigơ JK.

Trong các mạch logic tổ hợp có hiện tượng chạy đua vòng quanh (Race around) là sự xuất hiện tín hiệu giả (xung nhiễu) do quá trình quá độ khi 2 lối vào chuyển trạng thái theo 2 hướng ngược nhau nhưng sự chuyển mạch diễn ra ở hai chân không cùng một lúc ở lối ra cửa logic xuất hiện xung kim.

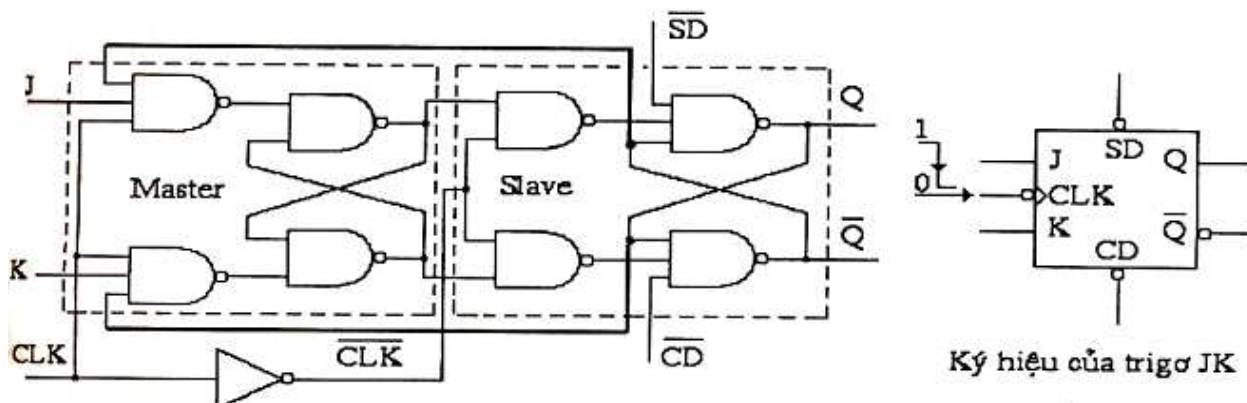
Sự tạo thành xung nhiễu ở các cửa logic trong hiện tượng chạy đua được minh họa trên hình 5.9.



Hình 5.9: Sự hình thành xung nhiễu trong hiện tượng chạy đua

Do xung tác động vào B bị trễ so với xung tác động vào A nên có thời điểm cả A và B đều ở mức cao cho nên lối ra Y có xung kim nhiễu. Nếu tải của lối ra là các phần tử nhạy như trigơ, xung nhiễu này có thể làm cho nó chuyển trạng thái không theo điều khiển. Vì vậy, cần phải loại trừ hiện tượng chạy đua này. Hiện tượng này còn sinh ra do quá trình quá độ của một mạch tổ hợp nối tiếp nhiều phần tử logic làm cho sự trễ ở lối ra so với lối vào tăng dần lên. Để khắc phục hiện tượng chạy đua người ta dùng trigơ JK chủ - tớ.

Trigơ JK chủ - tớ (Master - Slave)



Hình 5.10: Sơ đồ logic và ký hiệu của trigơ JK chủ - tớ

Sơ đồ logic và ký hiệu của trigơ JK Master-Slave cho trên hình 5.10. Nó được cấu tạo từ 2 trigơ RST mắc nối tiếp nhau với hai mạch phản hồi từ lối ra Q và \overline{Q} trở về các lối vào xoá và đặt (đường được tô đậm trên sơ đồ hình 5.10).

Trigơ chủ (Master) điều khiển trigơ tớ (Slave) chỉ những thay đổi trạng thái của trigơ chủ mới là nguyên nhân thay đổi trạng thái lối ra của trigơ Slave.

Khi xung nhịp chuyển từ mức logic 0 lên 1, thông tin ở lối vào JK được nạp vào trigơ chủ, trạng thái của trigơ chủ được xác lập theo tín hiệu điều khiển ở lối vào JK trigơ chủ chỉ thay đổi trạng thái một lần duy nhất trong khoảng thời gian kéo dài của xung nhịp.

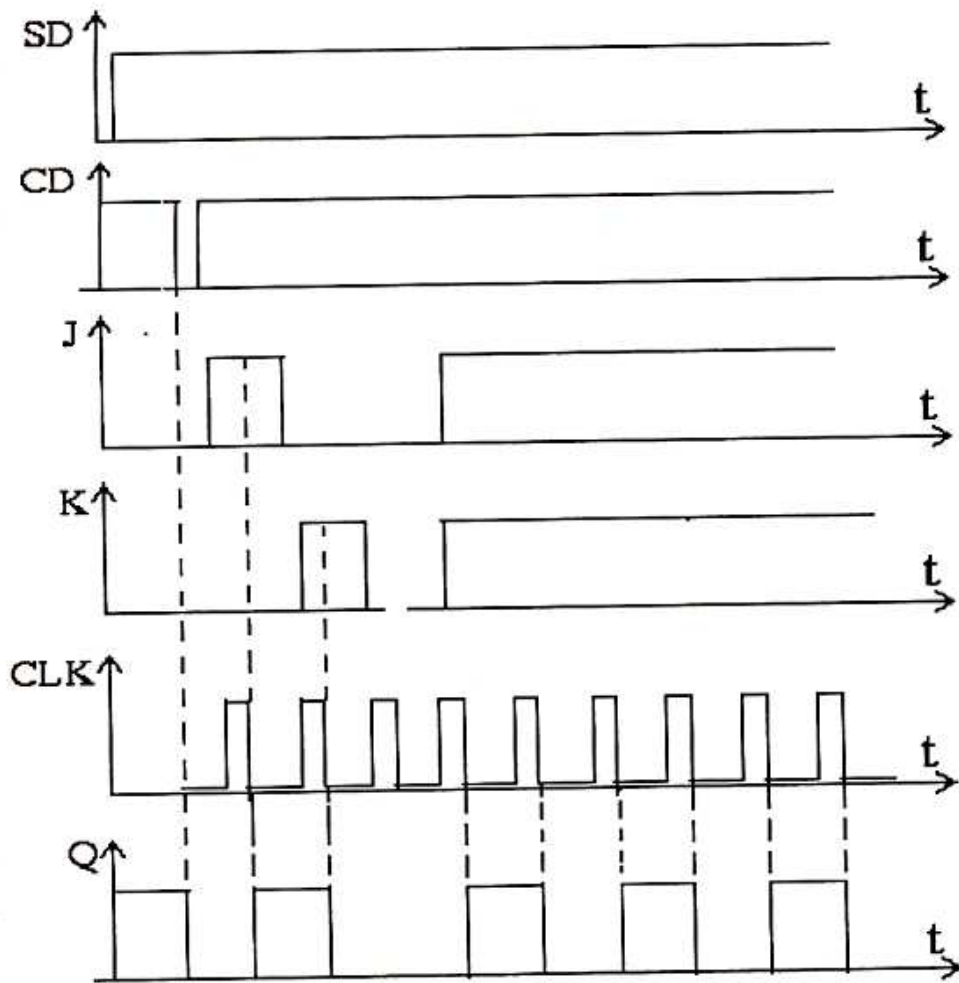
Khi xung nhịp CLK chuyển từ mức logic 1 về 0 (\overline{CLK} chuyển từ 0 lên 1) trigơ chủ ở trong trạng thái nhớ, trigơ tớ (Slave) sao chép lại trạng thái của trigơ chủ. Bởi vì trong thời gian trigơ Slave xác lập trạng thái thì trigơ chủ ở trong trạng thái nhớ, nên các thay đổi ở bên ngoài không hề ảnh hưởng đến quá trình xác lập trạng thái ở lối ra của trigơ, chính vì thế hoạt động của trigơ JK Master-Slave mang tính dứt khoát và ổn định cao hơn là trigơ khác.

Giản đồ thời gian mô tả các trạng thái hoạt động của trigơ JK Master-Slave được trình bày trên hình 5.11.

Chức năng của trigơ JK Master - Slave và bảng chân lý của nó giống hệt như trigơ JK, nhưng hoạt động của nó dứt khoát hơn. Giản đồ thời gian hình 5.11 tương tự như hình 5.8, điểm khác nhau duy nhất là trạng thái của trigơ trong trường hợp này được xác lập sau khi xung nhịp đã chuyển từ 1 về 0.

Đầu ra Q sẽ chuyển trạng thái ứng với sườn âm của xung nhịp tác dụng.

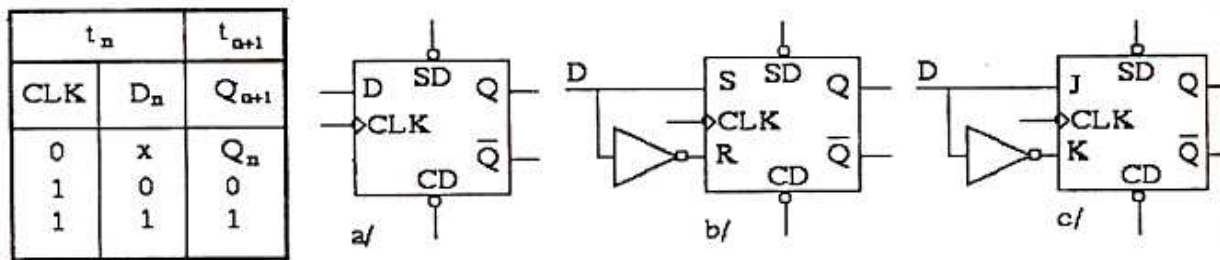
Cũng giống như trigơ JK, đối với trigơ JK Master-Slave khi J và K đồng thời ở mức cao, qua giản đồ xung hình 5.11 ta thấy cứ 2 chu kỳ CLK tương ứng với 1 chu kỳ ở lối ra Q. Vì vậy, ta có thể dùng trigơ làm các bộ đếm nhị phân chia tần với hệ số chia là 2^n (n là số trigơ).



Hình 5.11: Giản đồ xung của trigơ JK chủ - tớ

5.4. TRIGƠ D

Trigơ D là loại trigơ được dùng nhiều trong các bộ ghi lưu trữ các bit thông tin nhị phân. Ký hiệu logic của trigơ D được cho trên hình 5.12. Nó có một lối vào dữ liệu được ký hiệu bằng chữ D hoạt động đồng bộ với lối vào xung nhịp CLK, trigơ D hoạt động theo nguyên tắc sau: số liệu ở lối vào D sẽ được chuyển đến lối ra Q của trigơ sau một xung nhịp, tức là số liệu được chuyển đến lối ra chậm mất một khoảng thời gian bằng độ rộng của xung nhịp. Chính vì vậy mà nó có tên là trigơ D, lấy theo chữ đầu của thuật ngữ tiếng Anh Delay có nghĩa là trễ. Ký hiệu và bảng chân lý cho trên hình 5.12.

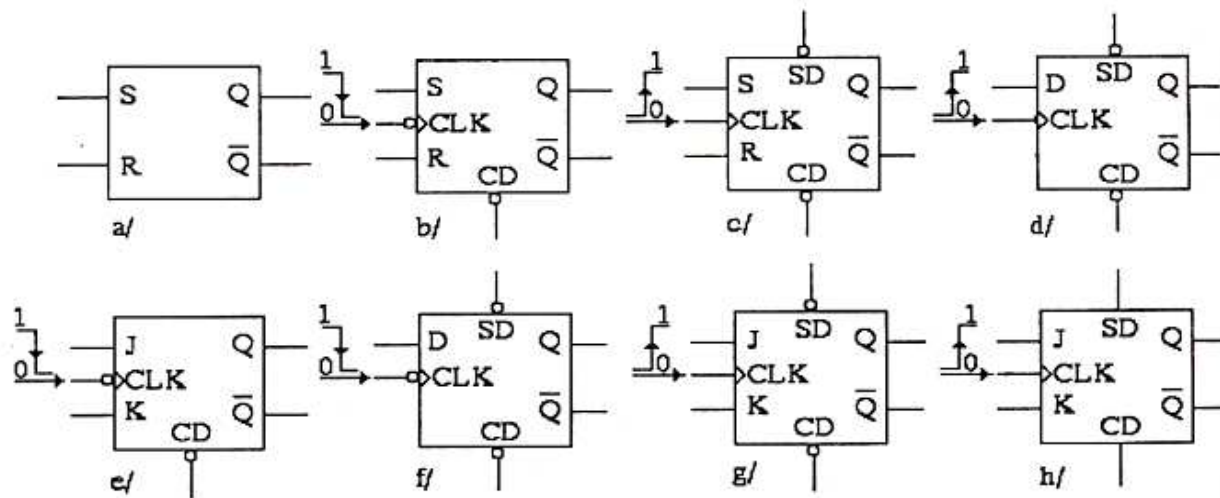


Hình 5.12: Ký hiệu trigơ D và bảng chân lý

Trigơ D có thể xây dựng từ trigơ RS hoặc trigơ JK khi ta mắc như hình 5.12b và 5.12c.

4. Bài tập thực hành

- 1- Khảo sát hoạt động của trigơ JK 74LS76 và trigơ D 74LS74.
- 2- Dùng trigơ JK 7476 hoặc trigơ D 7474 mắc thành bộ đếm nhị phân 4 bit.
- 3- Tạo trigơ RS, RST, JK, D có các ký hiệu như hình 5.13 từ mạch NAND 2 lối vào 7400, và NAND 3 lối vào 7410.



Hình 5.13

Chương VI

CÁC BỘ ĐẾM

6.1. CÁC SƠ ĐỒ ĐẾM NHỊ PHÂN

Đếm nhị phân được xây dựng từ các trigơ đồng bộ có lối vào xung nhịp như trigơ D và trigơ JK, muốn dùng trigơ RST để làm bộ đếm nhị phân thì phải mắc thêm phần tử đảo để chuyển nó thành trigơ D (hình 5.12 b). Các bộ đếm nhị phân là thành phần cơ bản của các hệ thống số, nó dùng để đếm xung, chia tần tạo xung thời gian làm các xung đồng hồ dùng trong máy tính và thiết bị thông tin.

Bộ đếm nhị phân có nhiều loại:

- Dựa theo cách điều khiển của xung nhịp đưa vào đếm người ta phân thành: Đếm nhị phân đồng bộ (Synchronous), đếm nhị phân không đồng bộ (Asynchronous).

- Dựa theo hướng đếm người ta đã tạo các bộ đếm tiến (đếm lên), hoặc đếm lùi.

- Trong kỹ thuật máy tính người ta còn dùng các bộ đếm có khả năng lập trình để thay đổi cấu trúc hệ đếm nhằm thay đổi phương thức đếm, cơ số của bộ đếm.

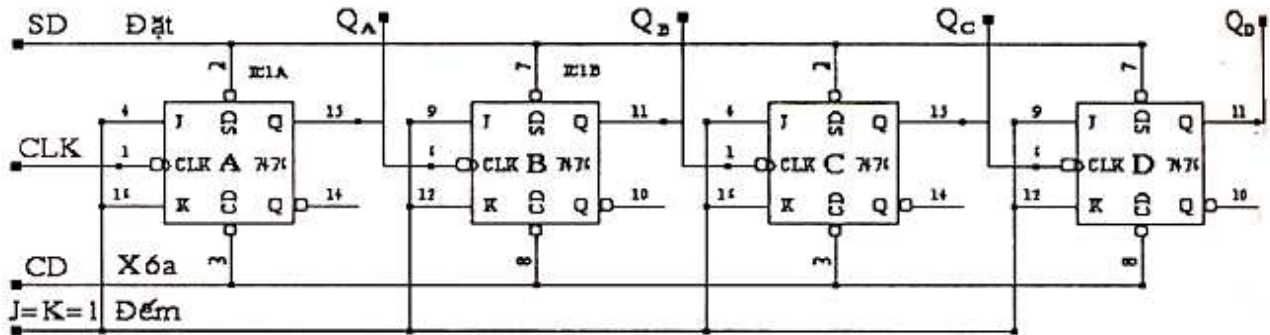
6.1.1. Đếm nhị phân không đồng bộ

Đếm nhị phân không đồng bộ còn được gọi là đếm *nối tiếp* : các trigơ mắc nối tiếp nhau, lối ra của trigơ trước được nối vào lối vào xung nhịp của trigơ sau.

- Đặc điểm của bộ đếm này là xung nhịp đếm không được đưa đồng thời vào các trigơ mà chỉ được đưa vào lối vào xung nhịp CLK của trigơ đầu tiên làm chuyển trạng thái trigơ đầu tiên, xung ở lối ra của trigơ này lại là xung nhịp làm chuyển trạng thái của trigơ tiếp theo...

6.1.1.1. Đếm tiến

Sơ đồ đếm nhị phân không đồng bộ 4 bit đếm tiến dùng trigger JK 7476 được trình bày trên hình 6.1, giản đồ xung của bộ đếm được trình bày trên hình 6.2.



Hình 6.1: Sơ đồ mạch đếm nhị phân không đồng bộ đếm tiến dùng trigger JK

Như ký hiệu trigger ghi trên sơ đồ, vì mạch 7476 là trigger JK Master-Slave có hai lối vào không đồng bộ xóa và đặt SD, CD đều tác động ở mức thấp.

Muốn xóa ta để SD = 1, CD = 0, muốn đặt ta để CD = 1, SD = 0. Dựa vào nguyên lý hoạt động của trigger JK chúng ta giải thích hoạt động của bộ đếm này.

Để bộ đếm làm việc ở chế độ đếm ta để lối vào điều khiển xóa, đặt CD = 1, SD = 1.

Đầu tiên xóa mạch đếm bằng xung xóa CD = 0. Lúc đó trạng thái lối ra của cả bốn trigger đều chuyển về 0:

$$Q_D Q_C Q_B Q_A = 0000$$

Sau đó để lối vào xóa và đặt ở mức cao: CD = SD = 1.

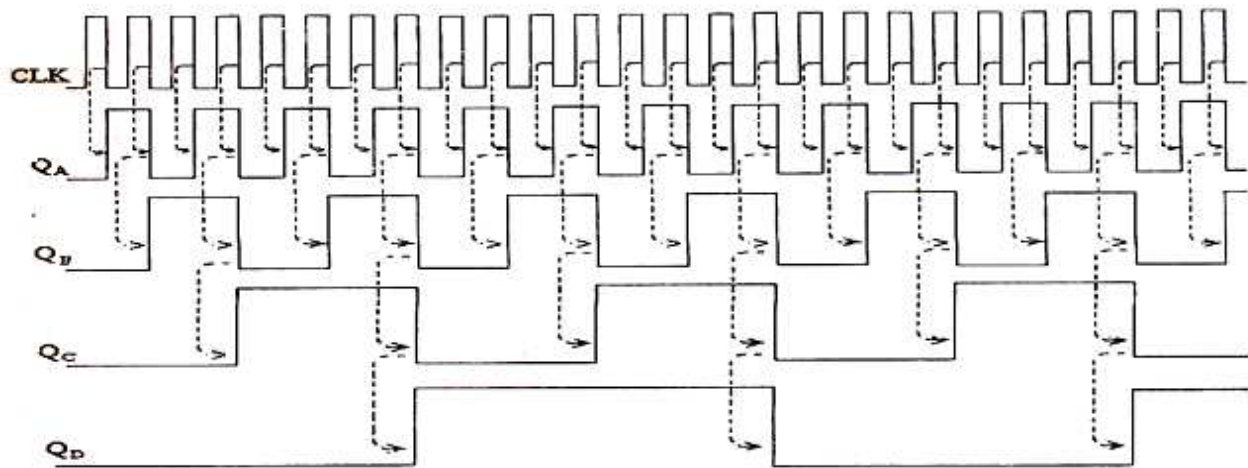
Đặt lối vào đếm J = K = 1: Mạch đếm bắt đầu hoạt động theo trạng thái của các lối vào đồng bộ J, K, và xung nhịp như giản đồ xung hình 6.6. Tất cả bốn trigger đều có J = K = 1 nên khi có xung nhịp tác dụng, các trigger đều chuyển trạng thái.

Trigger A chuyển trạng thái với mọi xung nhịp tác dụng chuyển từ 1 về 0.

Trigger B chuyển trạng thái khi Q_A chuyển từ 1 về 0.

Trigger C chuyển trạng thái khi Q_B chuyển từ 1 về 0.

Trigger D chuyển trạng thái khi Q_C chuyển từ 1 về 0.



Hình 6.2: Giản đồ xung của bộ đếm nhị phân

Trạng thái ra của bộ đếm theo số xung nhịp đưa vào bộ đếm được trình bày trên bảng 6.1.

Nhìn giản đồ xung ta thấy mỗi trigơ chia tần số xung nhịp làm 2. Có 4 trigơ sẽ chia tần số xung nhịp $2^4 = 16$ lần. Nếu có n trigơ sẽ có bộ chia 2^n lần. Như vậy bộ đếm cũng là bộ chia tần.

Bảng 6.1: Quá trình đếm của bộ đếm nối tiếp

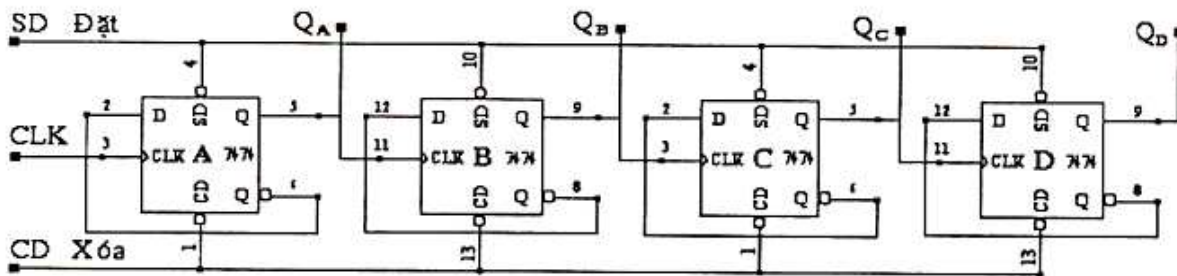
Xung nhịp	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

Ta cũng có thể dùng trigơ D mắc thành bộ đếm nhị phân. Muốn vậy ta phải mắc lối ra \bar{Q} của trigơ D với lối vào D của nó. Khi đó trạng thái lối ra của trigơ sẽ được xác định theo phương trình sau:

$$Q_{n+1} = D_n = \bar{Q}_n$$

Trường hợp này tương tự như đối với trigơ JK khi các lối vào $J=K=1$, nghĩa là cứ sau mỗi lần có xung nhịp tác dụng trigơ lại chuyển trạng thái một lần.

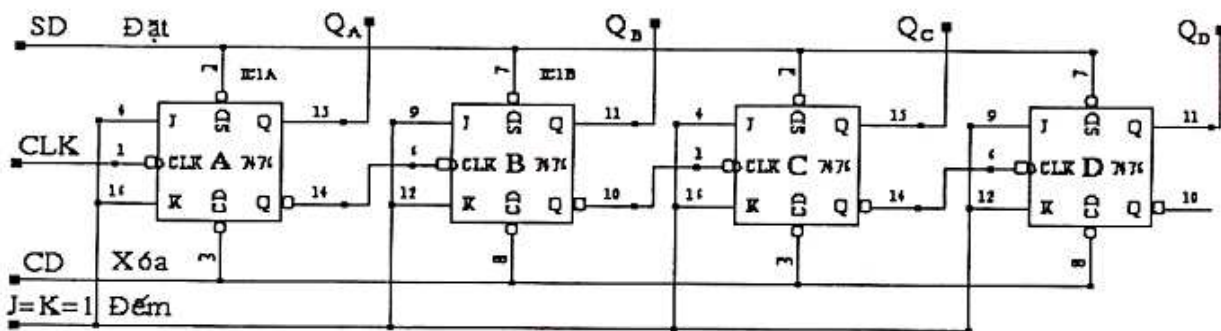
Sơ đồ đếm nhị phân 4 bit dùng trigơ D 7474 cho trên hình 6.3.



Hình 6.3: Sơ đồ bộ đếm nhị phân 4 bit không đồng bộ dùng trigơ D

Sơ đồ đếm nối tiếp modul 16 dùng 4 trigơ JK nêu trên hình 6.1 là sơ đồ đếm tiến.

6.1.1.2. Đếm lùi



Hình 6.4: Sơ đồ mạch đếm lùi nhị phân không đồng bộ dùng trigơ JK

Ta có thể xây dựng mạch đếm lùi nhị phân 4 bit dùng trigơ JK 7476, mắc mạch theo sơ đồ cho trên hình 6.4

Trạng thái ra của các trigơ trong bộ đếm thay đổi theo xung nhịp đượ trình bày trên bảng 6.2 dưới đây:

Bảng 6.2

Xung nhịp	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	0	1	1	1
10	0	1	1	0
11	0	1	0	1
12	0	1	0	0
13	0	0	1	1
14	0	0	1	0
15	0	0	0	1
16	0	0	0	0

Ở mạch đếm lùi ta thấy lối ra \bar{Q} của trigơ trước đượ nối vào CLK của trigơ sau nên trigơ sau sẽ chuyển trạng thái khi trigơ đứng trước nó chuyển từ 0 lên 1.

Quá trình đếm như sau: Đầu tiên cho xung xoá $CD = 0$ thì lối ra:

$$Q_D Q_C Q_B Q_A = 0000.$$

Sau xung nhịp thứ nhất tác dụng trigơ A chuyển từ 0 lên 1, các trigơ B,C,D lần lượt theo nhau chuyển từ 0 lên 1 nên trạng thái ở lối ra của bộ đếm là 1111.

Sau xung nhịp thứ hai tác dụng các lối ra của bộ đếm là: 1110

.....

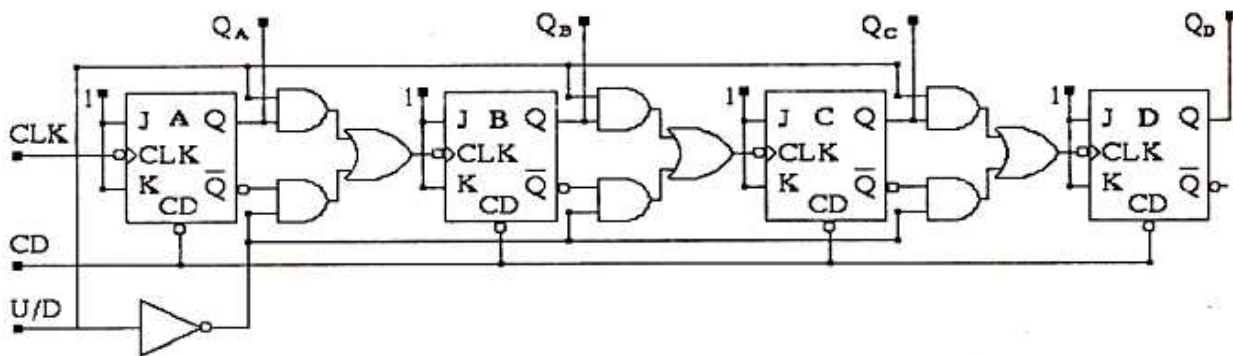
Sau xung nhịp thứ 15 tác dụng các lối ra của bộ đếm là: 0001

Sau xung nhịp thứ 16 tác dụng các lối ra của bộ đếm là : 0000

như lúc đầu.

6.1.1.3. Đếm tiến lùi

Để có 1 bộ đếm vừa đếm tiến vừa đếm lùi được ta thêm 1 đầu vào điều khiển tiến lùi UP/DOWN sơ đồ mạch đếm tiến lùi như hình 6.5

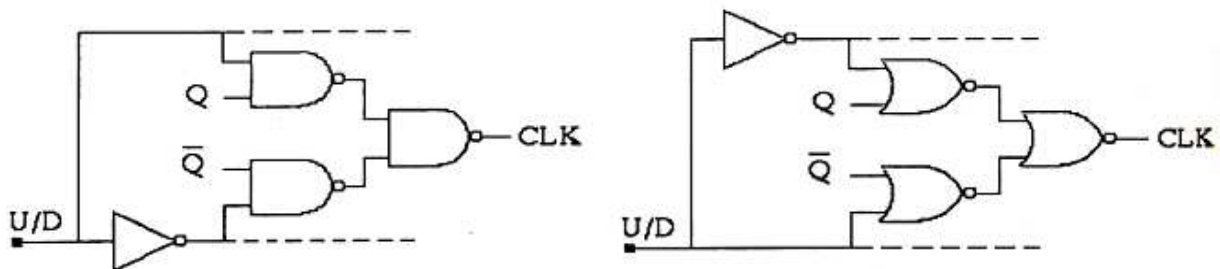


Hình 6.5: Sơ đồ bộ đếm tiến hoặc lùi 4 bit mã nhị phân

Đếm tiến: Khi cho lối vào điều khiển tiến lùi U/D (UP/DOWN) = "1" lối ra Q của trigơ trước nối với CLK của trigơ tiếp theo, sơ đồ tương đương như hình 6.5, ta có mạch đếm tiến.

Đếm lùi : Khi cho lối vào điều khiển tiến lùi U/D (UP/DOWN) = "0" lối ra \bar{Q} của trigơ trước nối với CLK của trigơ tiếp theo, sơ đồ tương đương như hình 6.8, ta có mạch đếm lùi .

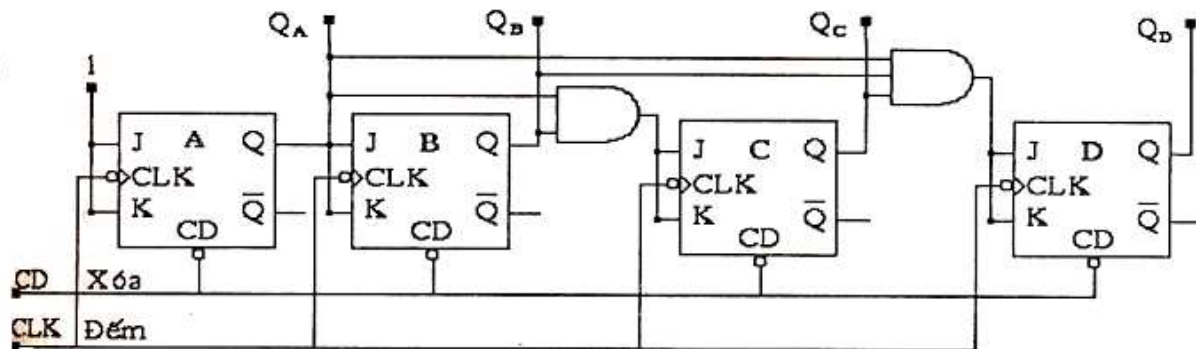
Ba cụm cửa logic của mạch UP/DOWN chen giữa các trigơ có thể thay bằng mạch NAND 7400 hoặc mạch NOR 7402, sơ đồ mắc như hình 6.6.



Hình 6.6: Cụm cửa logic dùng cho mạch UP/DOWN

6.1.2. Đếm nhị phân đồng bộ

Đếm nhị phân đồng bộ còn gọi là đếm song song. Đếm không đồng bộ có nhược điểm là tốc độ chậm vì có quá trình trễ khi đi qua các trigơ. Để khắc phục nhược điểm đó người ta dùng mạch đếm song song, nghĩa là các xung nhịp đồng thời tác dụng vào tất cả các trigơ. Sơ đồ đếm nhị phân đồng bộ 4 bit cho trên hình 6.7.



Hình 6.7: Sơ đồ đếm nhị phân đồng bộ 4 bit

Từ sơ đồ trên ta thấy: tuy xung nhịp tác động đồng thời vào các trigơ nhưng chỉ trigơ nào có $J = K = 1$ thì nó mới chuyển trạng thái. Từ sơ đồ hình 6.7 ta có được các điều kiện chuyển trạng thái của các trigơ trong bộ đếm như sau:

Trigơ A chuyển trạng thái với mọi xung nhịp CLK.

Trigơ B chuyển khi $Q_A = 1$

Trigơ C chuyển khi $Q_A = Q_B = 1$

Trigơ D chuyển khi $Q_A = Q_B = Q_C = 1$

Như vậy các trigơ sau chỉ chuyển trạng thái khi tất cả lối ra Q của các trigơ ở trước nó đồng thời bằng 1. Quá trình đếm của sơ đồ có thể mô tả như sau:

Sau khi tác dụng xung xoá CD thì $Q_D Q_C Q_B Q_A = 0000$.

Khi có xung nhịp đầu tiên tác dụng chỉ trigơ A chuyển trạng thái từ 0 lên 1, các trigơ B, C, D không chuyển trạng thái vì J, K của chúng bằng 0. trạng thái lối ra của bộ đếm sau khi kết thúc xung nhịp thứ nhất là: 0001.

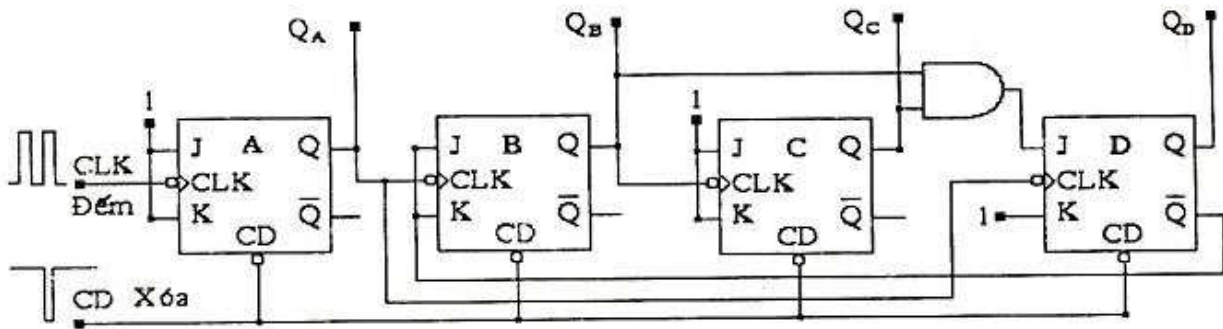
Khi có xung nhịp thứ hai tác dụng: J, K của trigơ B là 1 nên B và A đều chuyển trạng thái, Q_A từ 1 về 0, Q_B từ 0 lên 1; trigơ C, D vẫn chưa chuyển trạng thái, trạng thái ở lối ra của bộ đếm sau khi kết thúc xung nhịp thứ hai là 0010.

Quá trình hoạt động của bộ đếm nhị phân đồng bộ cũng diễn ra tiếp tục như bộ đếm nhị phân không đồng bộ, nó có bảng trạng thái và giản đồ xung như bộ đếm nhị phân không đồng bộ đã được nêu trên hình 6.2 và bảng 6.1.

6.2. ĐẾM 10 MÃ BCD (DECADE)

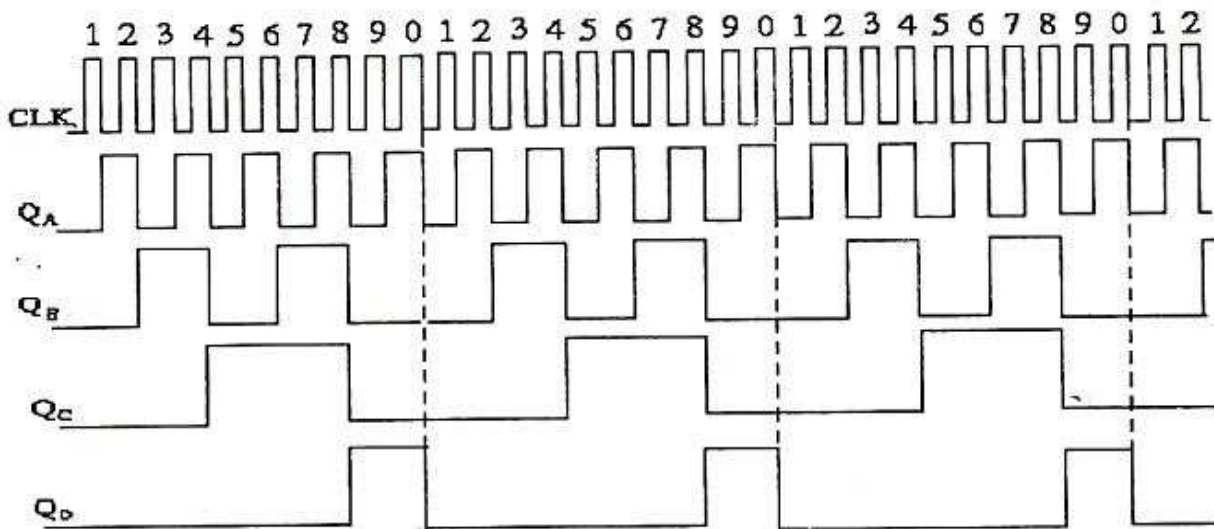
6.2.1. Bộ đếm 10 không đồng bộ mã BCD

Sơ đồ bộ đếm 10 không đồng bộ mã BCD được cho trên hình 6.8. Giản đồ xung minh họa quá trình hoạt động của bộ đếm được trình bày trên hình 6.9.



Hình 6.8: Mạch đếm 10 mã BCD

Bảng chân lý của bộ đếm được trình bày trên bảng 6.3.



Hình.6.9: Giản đồ xung bộ đếm 10 mã BCD

Sơ đồ đếm 10 mã BCD không đồng bộ hình 6.8 cho chúng ta thấy: Trùng A chuyển trạng thái với mọi xung nhịp tác dụng.

Điều kiện để trigơ B chuyển trạng thái là:

$Q_D = 0$ (tức là $\bar{Q}_D = 1$), Q_A chuyển từ 1 về 0.

Điều kiện để trigơ C chuyển trạng thái là: Q_A chuyển từ 1 về 0.

Điều kiện để trigơ D chuyển trạng thái là: $Q_B = Q_C = 1$, Q_A chuyển từ 1 về 0. Khi Q_B hoặc Q_C bằng 0, Q_A chuyển từ 1 về 0 thì $Q_D = 0$

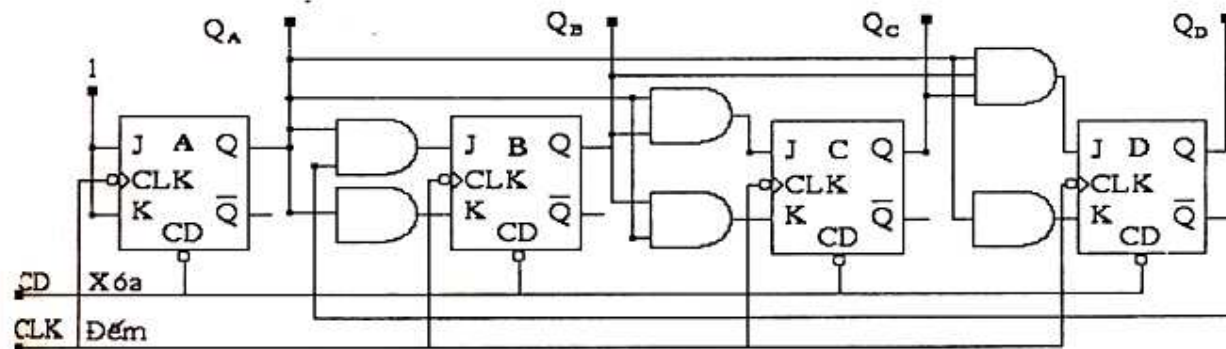
Từ sự phân tích các điều kiện chuyển trạng thái của các trigơ trong bộ đếm chúng ta thấy rằng quá trình đếm diễn ra theo trình tự nêu trong bảng 6.3.

Kết thúc xung thứ tám, trạng thái ở lối ra của bộ đếm là 1000, $\bar{Q}_D = 0$ trigơ B và C không còn chuyển trạng thái được nữa.

Kết thúc xung thứ 9, trạng thái của bộ đếm là: 1001, lúc này $J_D = 0$, $K_D = 1$ nên khi có xung thứ 10 tác dụng $Q_D = 0$, Q_A cũng chuyển từ 1 về 0, lúc này trạng thái ở các lối ra của bộ đếm là: 0000. Quá trình đếm lại tiếp tục theo một chu trình lặp lại như trước.

6.2.2. Đếm 10 mã BCD đồng bộ

Sơ đồ bộ đếm 10 mã BCD đồng bộ cho trên hình 6.14. Bảng chân lý và giản đồ xung của bộ đếm này hoàn toàn như là bộ đếm không đồng bộ



Hình 6.10: Bộ đếm 10 đồng bộ mã BCD

Sơ đồ đếm 10 mã BCD đồng bộ được xây dựng từ sơ đồ đếm nhị phân đồng bộ 4 bit. Để bộ đếm chỉ đếm đến 10 tương ứng với 10 tổ hợp trạng thái ở lối ra (là 0000, 0001, 0010, 0011, 0100, 0101, 0111, 1000, 1001) người ta đã bổ sung đường phản hồi nối từ \bar{Q}_D về lối vào J K của trigơ B. Chúng ta hãy phân tích hoạt động của bộ đếm này. Trước tiên chúng ta cần nêu các điều kiện chuyển trạng thái của các trigơ. Từ sơ đồ mạch hình 6.10 ta thấy:

Trigơ A chuyển mạch với mọi xung nhịp tác dụng

Trigơ B chuyển mạch khi $Q_A = 1$ và $Q_D = 0$ (tức là $\bar{Q}_D = 1$)

Trigơ C chuyển mạch khi $Q_A = Q_B = 1$

Trigơ D chuyển mạch khi $Q_A = Q_B = Q_C = 1$

Với sơ đồ trên, quá trình đếm diễn ra như bảng 6.3.

Bảng 6.3: Bảng chân lý của bộ đếm 10 mã BCD

C_K	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

Sau khi kết thúc xung nhịp thứ 9, $J_D = 0$, $K_D = 1$, khi có xung thứ 10 tác dụng Q_D chuyển từ 1 về 0. Q_A cũng chuyển từ 1 về 0 trạng thái lối ra của bộ đếm là: 0000. Bộ đếm lại bắt đầu một chu trình mới giống như trước.

Các vi mạch đếm nhị phân và đếm 10 mã BCD

Các tiết trước chúng ta đã khảo sát cấu trúc hoạt động của các bộ đếm nhị phân và đếm 10 mã BCD. Các bộ đếm này được tạo thành từ các trigơ rời rạc, công nghệ chế tạo các vi mạch logic hiện nay đã phát triển ở trình độ rất cao. Đã có rất nhiều vi mạch mức độ tổ hợp cỡ trung bình và cỡ lớn (MSI, LSI) giá thành các vi mạch này rất rẻ. Trên thị trường hiện nay có rất nhiều loại vi mạch cỡ trung bình MSI là các bộ đếm hoàn chỉnh. Sau đây giới thiệu 1 số mạch đếm họ TTL và CMOS.

Các vi mạch đếm họ TTL:

Các vi mạch đếm nhị phân 4 bit: 7493, 74LS293, 74161, 74163, 74193, 74191.

Các vi mạch đếm 10 mã BCD: 7490, 7492, 74160, 74190.

7490 : mạch đếm 10 không đồng bộ (phần đầu chia 2, phần sau chia 5)

7492 : mạch đếm 10 đồng bộ mã BCD có các mode điều khiển

7493 : mạch đếm 16 không đồng bộ mã nhị phân

74160 : mạch đếm 10 đồng bộ đặt trước, theo mã BCD

74161 : mạch đếm 16 đồng bộ mã nhị phân

74190 : mạch đếm 10 đồng bộ, theo mã BCD

74191 : mạch đếm 16 đồng bộ có các mode điều khiển.

Các vi mạch đếm thuộc họ CMOS:

CD4029 : mạch đếm tiến lùi mã nhị phân và mã BCD

74C193 : mạch đếm nhị phân đồng bộ 4bit tiến /lùi

74C192 : mạch đếm 10 đồng bộ tiến /lùi

74C90 : mạch đếm 10 không đồng bộ mã BCD.

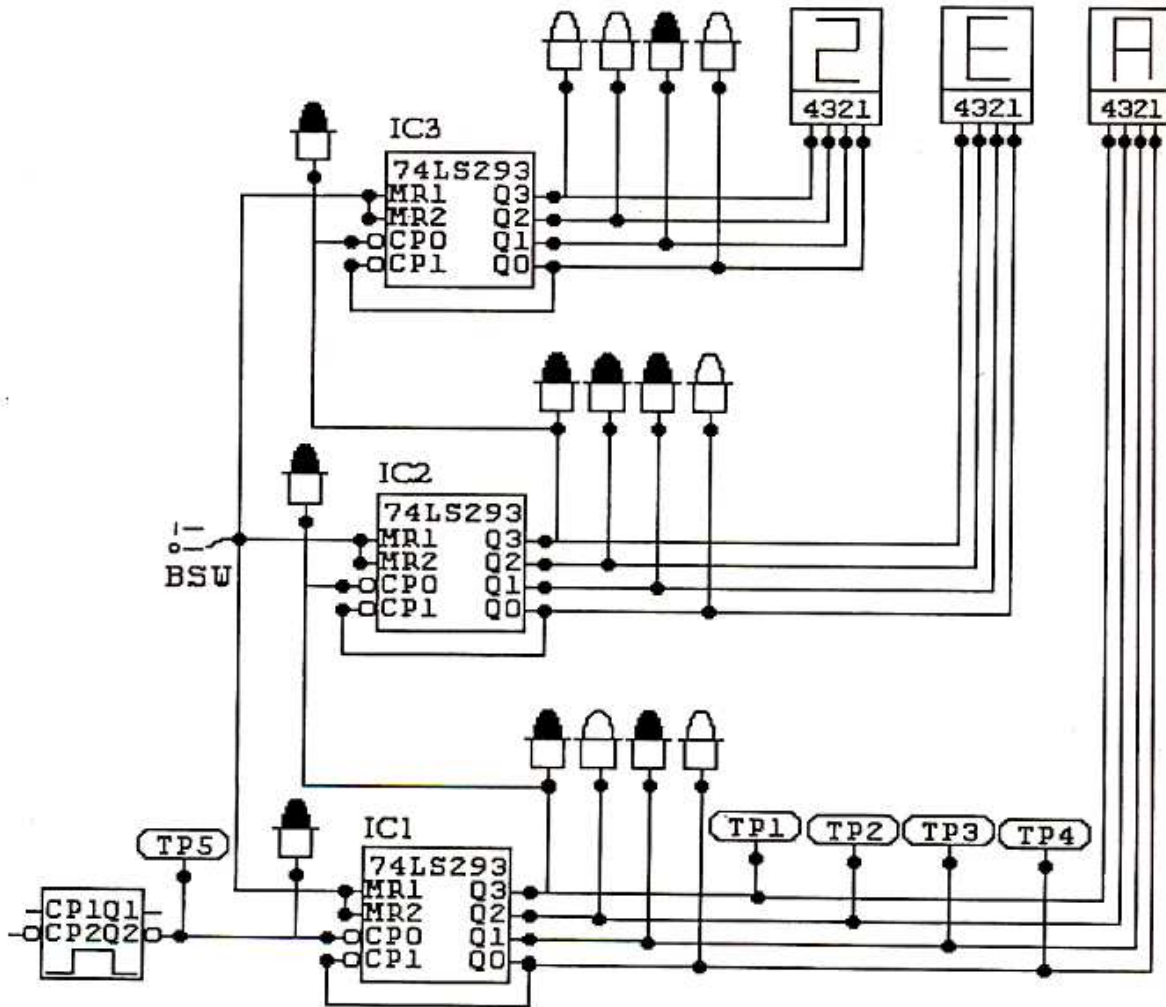
Sau đây sẽ giới thiệu một số các sơ đồ đếm nhị phân và đếm 10 mã BCD dùng các vi mạch đếm này.

Trên hình 6.11 là sơ đồ bộ đếm nhị phân không đồng bộ 12 bit dùng 3 vi mạch 74LS293 mỗi vi mạch là một bộ đếm nhị phân không đồng bộ 4 bit được tách làm hai : mạch đếm 2 có lối vào xung nhịp kí hiệu là CP0, lối ra Q0 và bộ đếm 8 có lối vào xung nhịp là CP1, ba lối ra của mạch đếm 8 là Q1, Q2, Q3. Để mạch đếm được 16 cần phải ghép nối tiếp bộ đếm 2 với bộ đếm 8 bằng cách nối Q0 với CP1.

Công tắc bit BSW được nối các chân MR1, MR2 của vi mạch dùng để xoá bộ đếm, khi các chân này ở mức logic 1 bộ đếm được xoá các lối ra Q0, Q1, Q2, Q3 của bộ đếm đều ở mức logic 0. Để bộ đếm ở trạng thái đếm các chân này phải đặt ở mức logic 0.

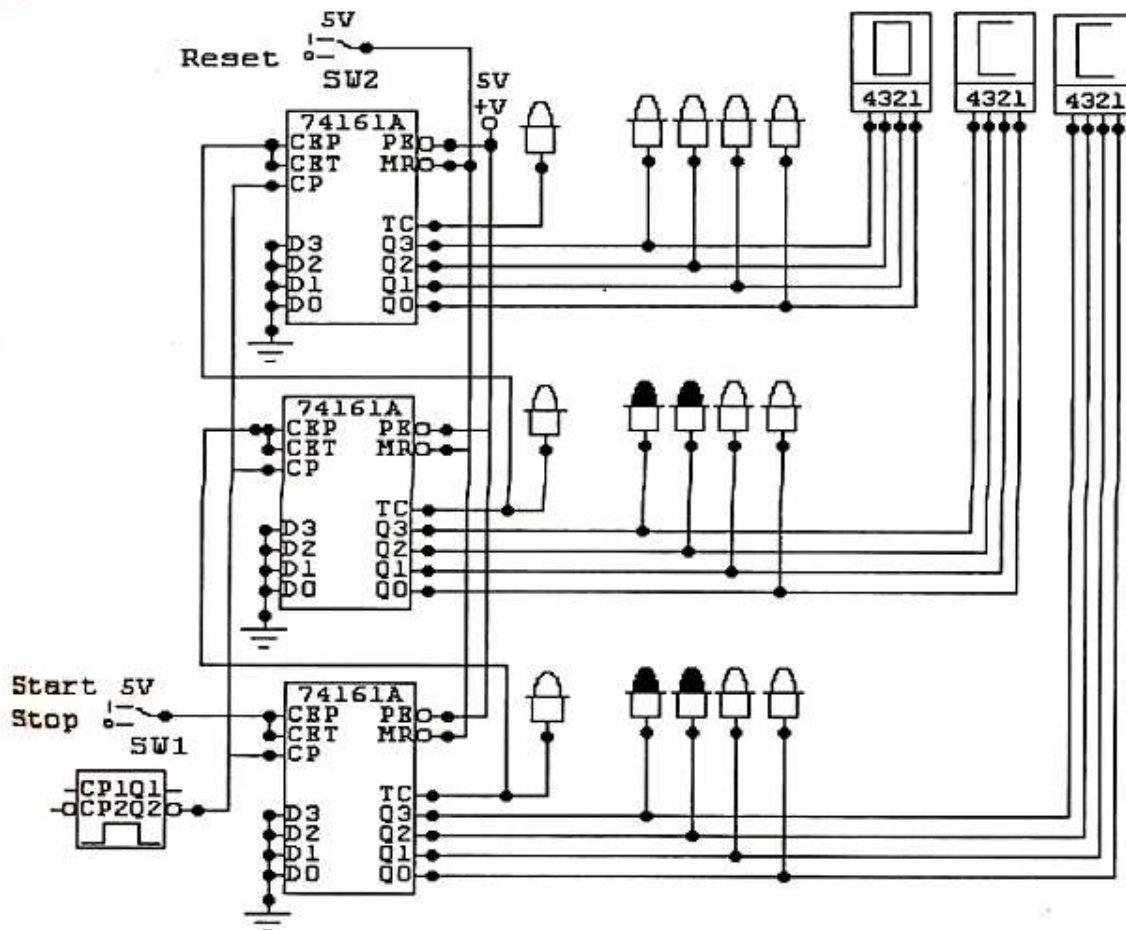
Trên sơ đồ hình 6.11: CP1Q1, CP2Q2 là khối tạo xung đồng hồ. Các bộ hiển thị chữ số thập lục phân được tạo từ bộ giải mã từ mã nhị phân 4 bit ra

mã 7 đoạn hiển thị chữ số trong hệ thập lục phân kết hợp đèn LED 7 đoạn. Để chỉ thị các bit theo mã nhị phân có thể dùng đèn LED nối tiếp với điện trở 100Ω nối vào cực đất của vi mạch. Cách mắc các đèn LED và chỉ thị 7 đoạn chúng ta đã đề cập ở chương 4. Sơ đồ trên có thể chạy thử mô phỏng trên vi tính dùng phần mềm Circuit Maker.



Hình 6.11: Sơ đồ đếm nhị phân không đồng bộ 12 bit dùng IC 74LS293

Trên hình 6.12 trình bày sơ đồ bộ đếm nhị phân đồng bộ 12 bit dùng 3 vi mạch 74161A, mỗi vi mạch là một bộ đếm nhị phân đồng bộ 4 bit có các lối vào đặt trước D0, D1, D2, D3. Muốn nạp dữ liệu đặt trước vào bộ đếm thì chân điều khiển cho phép nạp dữ liệu PE của các vi mạch phải được đặt ở mức logic 0.



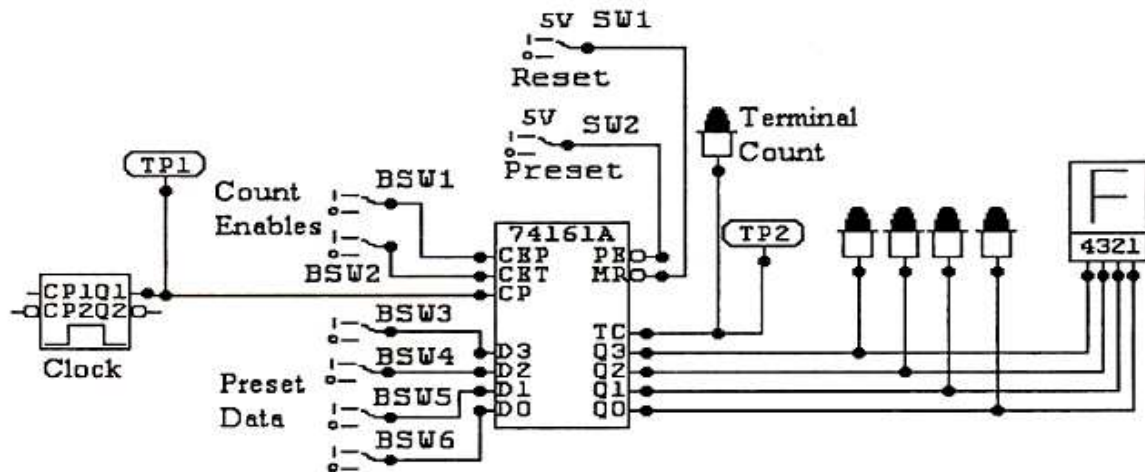
Hình 6.12: Sơ đồ bộ đếm nhị phân đồng bộ 12 bit dùng IC74161

Để các vi mạch ở chế độ đếm bình thường thì các chân PE, MR (điều khiển xoá bộ đếm) và các chân CEP, CET (điều khiển cho phép đếm) tất cả đều phải đặt ở mức logic 1. Thực hiện việc ghép nối 3 bộ đếm 4 bit để tạo thành bộ đếm nhị phân đồng bộ 12 bit trong trường hợp này có khác với bộ đếm nhị phân không đồng bộ 12 bit hình 6.11, ở đây các lối vào xung nhịp của cả 3 bộ đếm đều được đấu với nhau, xung nhịp được đưa vào đồng thời vào cả 3 bộ đếm.

Trên sơ đồ hình 6.12 ta thấy chuyển mạch SW1 nối với các chân CEP, CET của IC 74161 đầu tiên (tương ứng với 4 bit đầu của bộ đếm 12 bit) được dùng để điều khiển đếm. Khi chuyển mạch ở vị trí ứng với mức logic 1 - bộ đếm bắt đầu đếm (Start), khi chuyển mạch ở vị trí ứng với mức 0 - bộ đếm ngừng đếm (Stop).

Chuyển mạch SW2 nối với các chân MR của các IC 74161 được dùng để điều khiển xoá bộ đếm. Muốn xoá phải để chuyển mạch ở vị trí sao cho MR ở mức logic 0. Để bộ đếm tiếp tục làm việc thì chân này phải để ở mức 1.

Hình 6.13 trình bày một sơ đồ bố trí thí nghiệm để khảo sát hoạt động của các chân điều khiển của IC74161. Ta có thể nạp trước dữ liệu vào bộ đếm, xoá dữ liệu, khởi phát bộ đếm và dừng đếm. Chuyển mạch SW1 dùng để điều khiển xoá (Reset), chuyển mạch SW2 dùng để nạp trước dữ liệu vào bộ đếm. Các công tắc bit BSW1, BSW2 cho phép bộ đếm hoạt động. Các công tắc bit BSW3, BSW3,..., BSW6 dùng để đặt 4 bit dữ liệu.

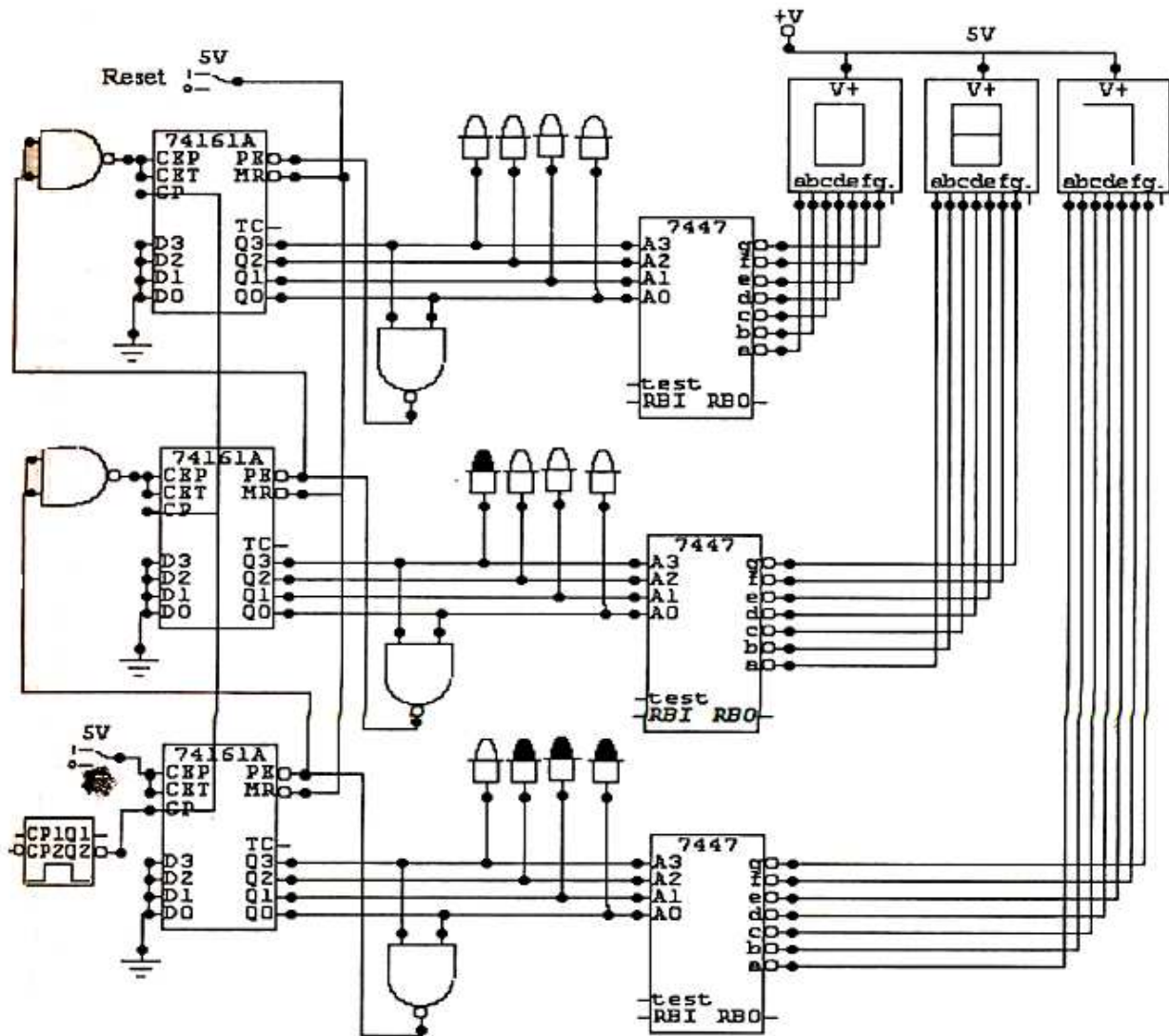


Hình 6.13: Sơ đồ thí nghiệm tìm hiểu chức năng các chân điều khiển của IC74161

Vì mạch 74161 là bộ đếm nhị phân đồng bộ, nhưng nếu mắc thêm các phần tử logic thực hiện mạch phản hồi điều khiển chân PE ta có thể dùng nó làm bộ đếm 10 mã BCD. Trên hình 8.14 là sơ đồ đếm 10 ba chữ số (digit) mã BCD dùng 3 IC 74161.

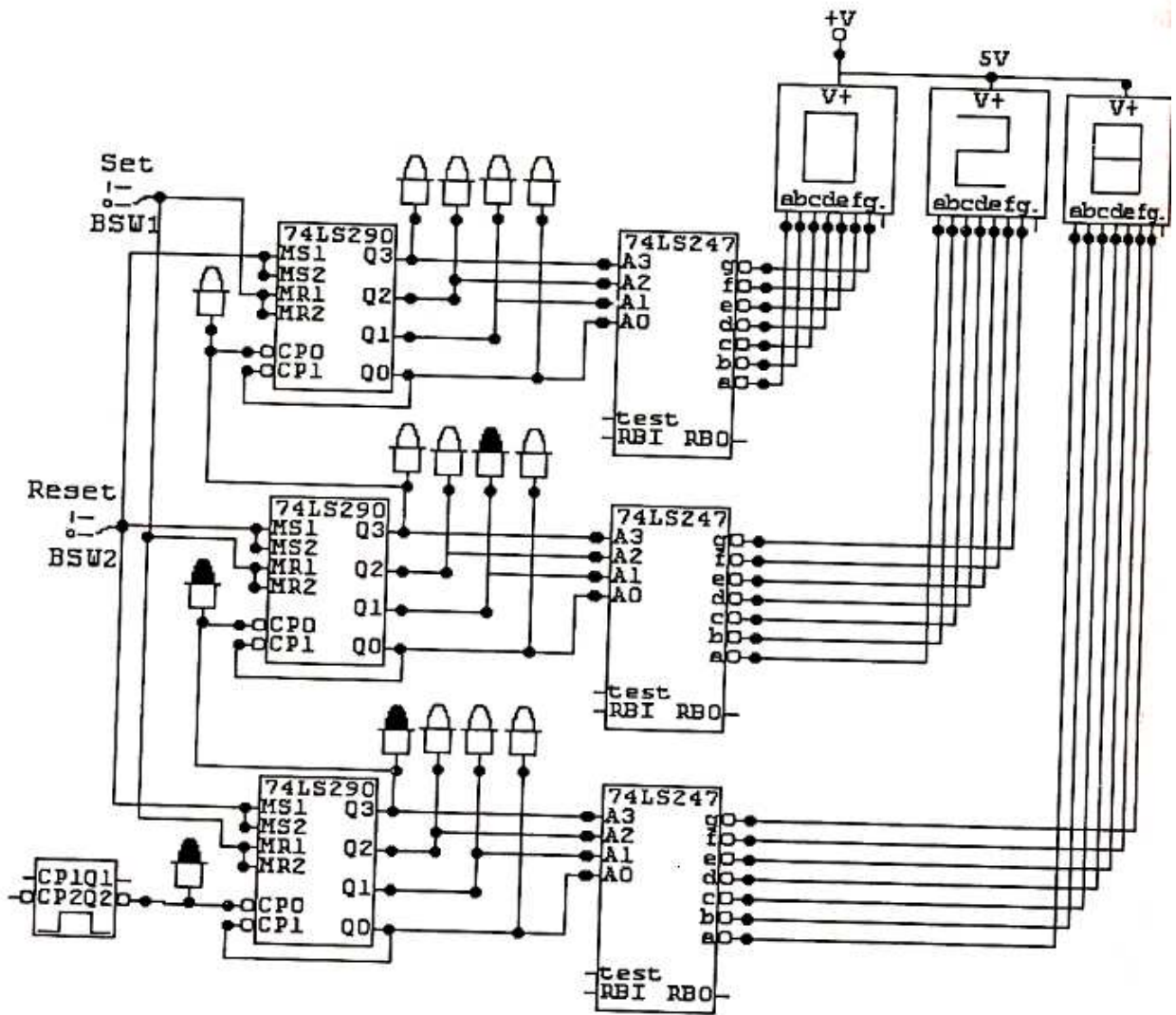
Từ sơ đồ hình 6.14 chúng ta thấy các phần tử NAND có hai lối vào nối vào chân Q3, Q0 của các bộ đếm, khi các bộ đếm đếm đến 9 (1001) lối ra của các phần tử NAND này sẽ chuyển về 0 tạo thành một xung âm điều khiển chân PE cho phép bộ đếm nạp dữ liệu. Vì các lối vào dữ liệu đều ở mức 0 nên cả 4 bit trong bộ đếm đều chuyển về 0, mạch lại bắt đầu đếm từ 0 đến 9. Xung âm điều khiển đưa các bộ đếm trước về không qua phần tử đảo tạo thành xung dương điều khiển cho phép bộ đếm sau bắt đầu đếm khi xung này kết thúc bộ đếm sau cũng ngừng đếm. Chính vì thế tuy xung nhịp tác dụng đồng thời vào cả 3 bộ đếm nhưng các bộ đếm sau chỉ đếm trong

khoảng thời gian khi trạng thái lối ra của bộ đếm đứng trước nó chuyển từ trạng thái 1001 về 0000.



Hình 6.14: Bộ đếm 10 đồng bộ mã BCD dùng các vi mạch đếm 74161 kết hợp các mạch giải mã 7 đoạn 7447

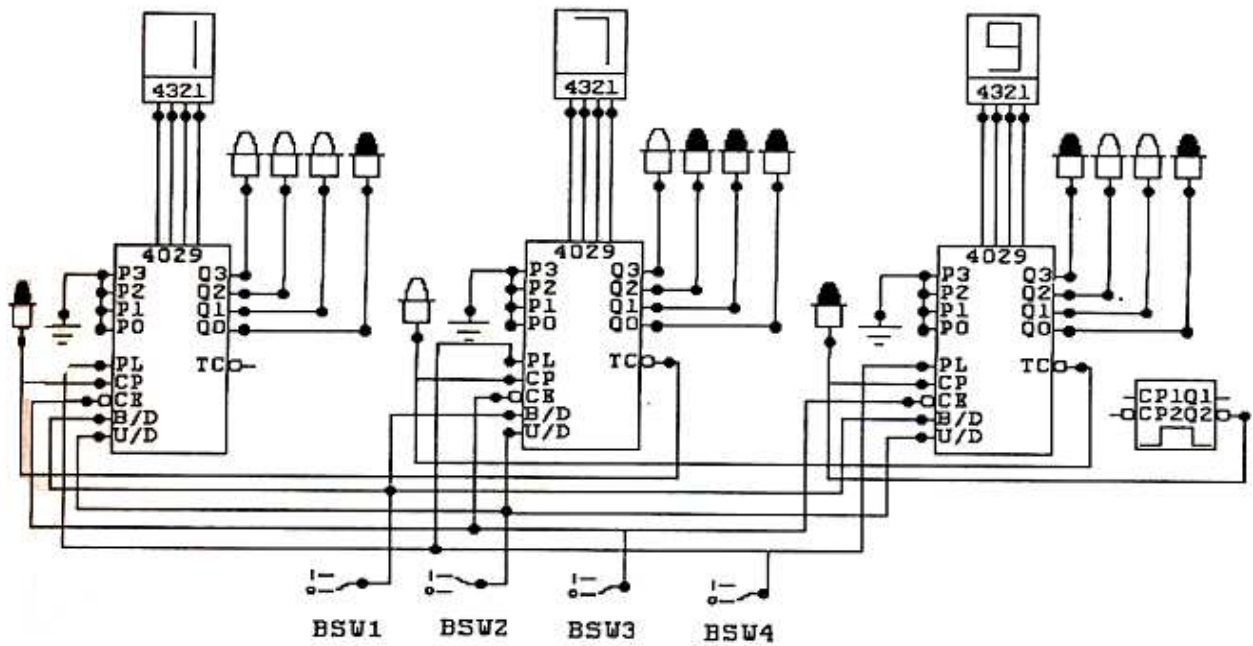
Trên hình 6.15 trình bày sơ đồ đếm 10 không đồng bộ mã BC dùng vi mạch 74LS290. IC 74LS290 là bộ đếm 10 không đồng bộ mã BCD, mỗi vi mạch tương ứng một con số thập phân. Mạch có lối vào xoá (MR1,MR2) Khi xoá hai chân này đặt ở mức 1. Khi ở trong trạng thái đếm hai chân này phải được đặt ở trong trạng thái 0. Hai chân có kí hiệu MS1, MS2 được dùng để đặt. Khi hai lối vào này ở mức logic 1 thì trạng thái lối ra của bộ đếm là: 1001 (tương ứng với số 9 thập phân). Khi đếm thì hai chân MS1, MS2 phải để ở mức logic 0.



Hình 6.15: Sơ đồ mạch đếm 10 ba digit mã BCD dùng IC74LS290

Mỗi vi mạch 74LS290 là một bộ đếm 10 mã BCD không đồng bộ 4 bit được tách làm hai : mạch đếm 2 có lối vào xung nhịp kí hiệu là CP0, lối ra Q0 và bộ đếm 5 có lối vào xung nhịp là CP1 ba lối ra của mạch đếm 5 là Q1, Q2, Q3. Để mạch đếm được 10 cần phải ghép nối tiếp bộ đếm 2 với bộ đếm bằng cách nối Q0 với CP1. Các IC74LS247 là bộ giải mã từ mã BCD ra mã 7đoạn có thể được thay thế bằng IC7447.

Trên hình 6.16 là sơ đồ mạch đếm 10 không đồng bộ theo mã BCD và đếm 16 không đồng bộ theo mã nhị phân. Trong sơ đồ này ta dùng vi mạch đếm 4029 thuộc họ logic CMOS.



Hình 6.16: Sơ đồ bộ đếm tiến lùi không đồng bộ 12 bit theo mã BCD và mã nhị phân

Công tắc bit BSW1 được đấu vào lối vào có kí hiệu B/D dùng để điều khiển bộ đếm 16 theo mã nhị phân hoặc đếm 10 theo mã BCD. Khi chuyển mạch đặt chân B/D ở mức logic 1, bộ đếm đếm theo mã nhị phân 4 bit. Khi chuyển mạch BSW1 đặt chân B/D ở mức logic 0, thì bộ đếm trở thành bộ đếm 10 theo mã BCD.

Công tắc bit BSW2 được đấu vào lối vào có kí hiệu U/D dùng để điều khiển bộ đếm theo chiều tiến, hoặc lùi. Khi chuyển mạch đặt chân U/D ở mức logic 1, bộ đếm đếm theo chiều tiến. Khi chuyển mạch đặt chân U/D ở mức logic 0, bộ đếm trở thành bộ đếm lùi.

Công tắc bit BSW3 được đấu vào lối vào có kí hiệu CE chân điều khiển cho phép đếm dùng để điều khiển bộ đếm. Khi chuyển mạch đặt chân CE ở mức logic 1 bộ đếm sẽ ngừng hoạt động. Để bộ đếm có thể đếm được chân cho phép đếm phải được đặt ở mức logic 0.

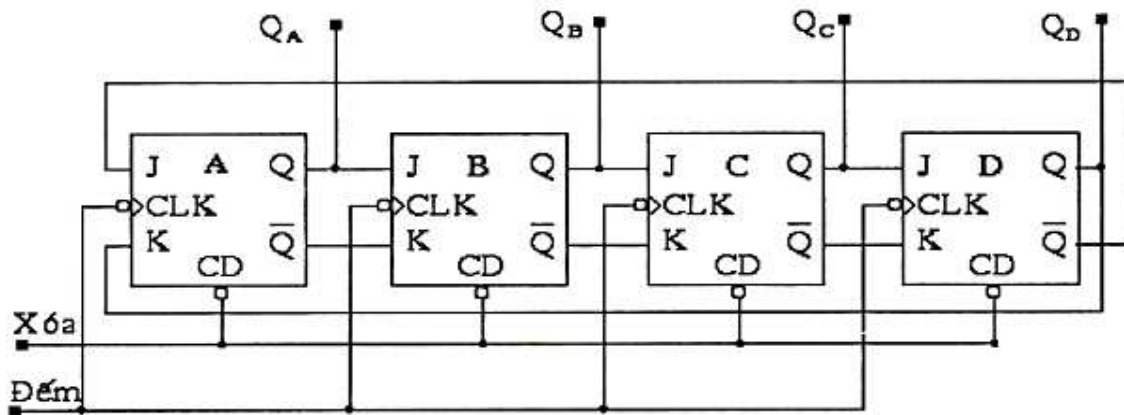
Công tắc bit BSW4 được đấu vào lối vào có kí hiệu PL chân điều khiển cho phép nạp dữ liệu vào bộ đếm. IC 4029 có 4 đường vào dữ liệu có kí hiệu là P0, P1, P2, P3. Khi chân PL để ở mức logic 1 thì dữ liệu từ 4 lối vào dữ liệu này được nạp vào bộ đếm.

Trong sơ đồ hình 6.16, các chân P0, P1, P2, P3 đều ở mức 0 nên khi chân PL ở mức 1 thì bộ đếm được xoá tất cả các bit trong bộ đếm đều bằng 0. Để bộ đếm hoạt động thì chân này phải đặt ở mức 0.

Các vị trí chuyển mạch trong sơ đồ hình 6.16 tương ứng với trường hợp bộ đếm đang đếm 10 theo mã BCD, đếm theo chiều tiến. Xung được đưa vào lối vào xung nhịp (có kí hiệu là CP) của bộ đếm đầu tiên được lấy từ khối tạo xung đồng hồ có kí hiệu CP1Q1, CP1Q2, bộ đếm đầu tiên tương ứng với số có trọng số nhỏ nhất nằm ở tận cùng bên phải. Xung lối ra ở chân TC của bộ đếm đầu tiên chính là xung nhịp cho bộ đếm tiếp theo. Xung ở lối ra TC được tạo thành khi bộ đếm đã tràn, nghĩa là với trường hợp đếm 10 thì xung bắt đầu được tạo thành giữa hai trạng thái lối ra của bộ đếm từ 1001 chuyển sang 0000 (9 chuyển lên 10), hoặc đếm theo mã nhị phân chuyển từ tổ hợp 1111 về 0000 (15 chuyển lên 16).

6.3. CÁC SƠ ĐỒ ĐẾM VÒNG THEO KIỂU GHI DỊCH MÃ JOHNSON

Trên hình 6.17 giới thiệu sơ đồ đếm 8 theo mã Johnson dùng 4 trigơ JK.



Hình 6.17: Sơ đồ đếm 8 mã Johnson

Đây là một sơ đồ đếm đồng bộ. Xung đếm được đưa đồng thời vào các lối vào xung nhịp của các trigơ, dưới sự điều khiển của xung nhịp trạng thái lối ra của các trigơ đứng sau lại được xác lập theo giá trị lối ra của trigơ đứng trước nó. Hoạt động của bộ đếm này được giải thích dựa trên nguyên tắc hoạt động của trigơ JK đã được nêu trên bảng chân lý của nó (bảng 5.4):

Nếu $J = 1, K = 0$ khi có xung nhịp tác dụng lối ra của trigơ nhận giá trị $Q = 1$.

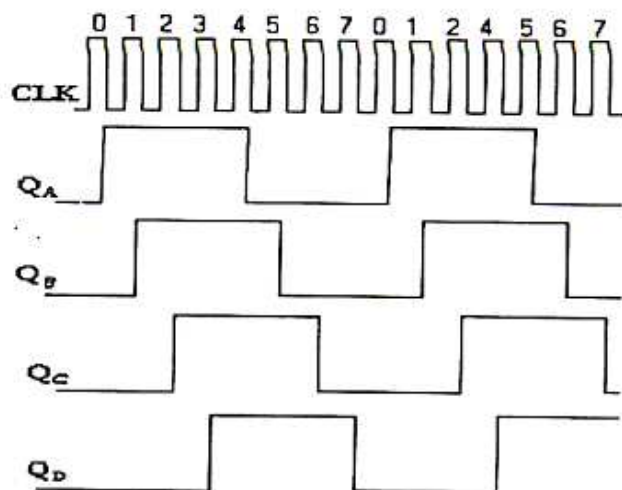
Nếu $J = 0, K = 1$ khi có xung nhịp tác dụng trigơ nhận giá trị $Q = 0$.

Khi ta cho xung xoá tác động vào lối vào xoá của bộ đếm, lối ra của cả 4 trigơ đều bằng 0, lối vào J của trigơ A: $J_A = 1, K_A = 0, J_B = J_C = J_D = 0, K_B = K_C = K_D = 1$, nên khi có xung nhịp đầu tiên được đưa vào bộ đếm: Q_A chuyển lên 1, Q_B, Q_C, Q_D vẫn ở trạng thái 0.

Kết thúc xung nhịp 1 ta có: $J_A = J_B = 1$; $K_A = K_B = 0$; $J_C = J_D = 0$, $K_C = K_D = 1$, nên khi có xung nhịp thứ 2 đưa vào bộ đếm $Q_A = Q_B = 1$, $Q_C = Q_D = 0$. Như vậy, cứ sau mỗi lần có một xung nhịp đưa vào bộ đếm, các trigơ lần lượt theo nhau chuyển lên 1, sau khi có 4 xung vào bộ đếm cả 4 trigơ đều chuyển lên 1. Trạng thái các lối vào của các trigơ sau khi có 4 xung đã vào bộ đếm:

$J_A = 0$, $K_A = 1$, $J_B = J_C = J_D = 1$, $K_B = K_C = K_D = 0$. Khi có xung thứ 5 vào bộ đếm ta có:

$Q_A = 0$, $Q_B = Q_C = Q_D = 1$, khi có các xung tiếp theo các trigơ lại lần lượt theo nhau chuyển về 0. Ta có bảng chân lý của bộ đếm được trình bày trên bảng 6.4. Giả đồ xung minh hoạ cho hoạt động của bộ đếm mã Johnson được trình bày trên hình 6.18.



Hình 6.18: Giả đồ xung của bộ đếm 8 theo mã Johnson

Bảng 6.4: Bảng chân lý của bộ đếm 8 theo mã Johnson

Xung	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

Bộ đếm này đếm đến 8, sau 8 xung vào bộ đếm các trigơ lại trở về trạng thái 0 ban đầu.

Đặc điểm của bộ đếm Johnson này là:

- Khi đếm trigơ sau được ghi lại kết quả của trigơ trước giống như ghi dịch.
- Đếm theo mã Johnson cần nhiều trigơ hơn đếm theo mã nhị phân. Nếu ta gọi M là môđun của bộ đếm, số xung lớn nhất bộ đếm có thể đếm được và n là số trigơ dùng trong bộ đếm ta có hệ thức biểu diễn mối tương quan giữa M và n sau đây đối với hệ đếm nhị phân:

$$M = 2^n$$

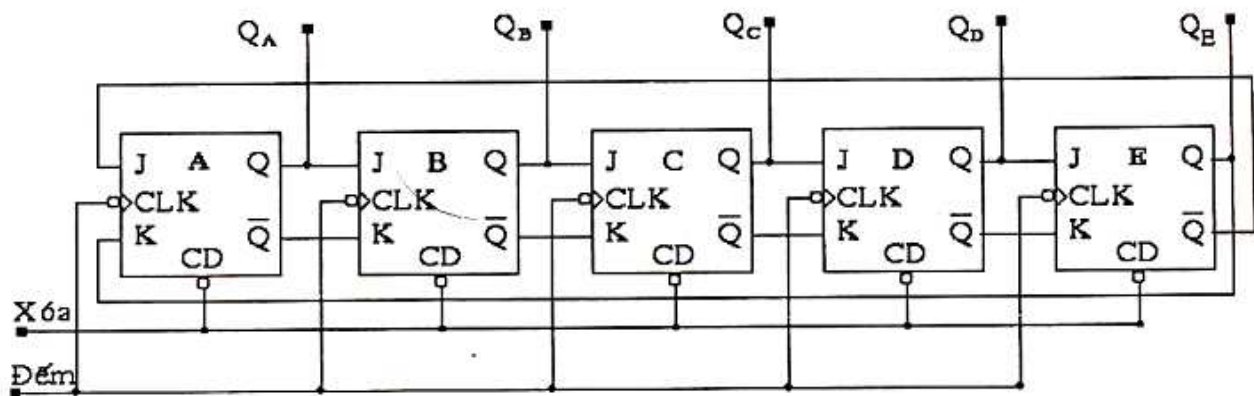
Đối với bộ đếm Johnson ta có:

$$M = 2n.$$

Ở bộ đếm có 4 trigơ, đếm được đến 8. Cũng với số trigơ là 4 bộ đếm nhị phân có thể đếm được 16.

Bộ đếm Johnson tuy phải dùng nhiều trigơ nhưng tốc độ đếm nhanh và ổn định, đặc biệt là giải mã của nó ra thập phân đơn giản hơn nhiều so với việc giải mã từ bộ đếm nhị phân ra thập phân.

Tương tự như trên ta có thể xây dựng bộ đếm 10 mã Johnson từ trigơ JK. Sơ đồ cho trên hình 6.19.



Hình 6.19: Sơ đồ đếm 10 mã Johnson dùng trigơ JK

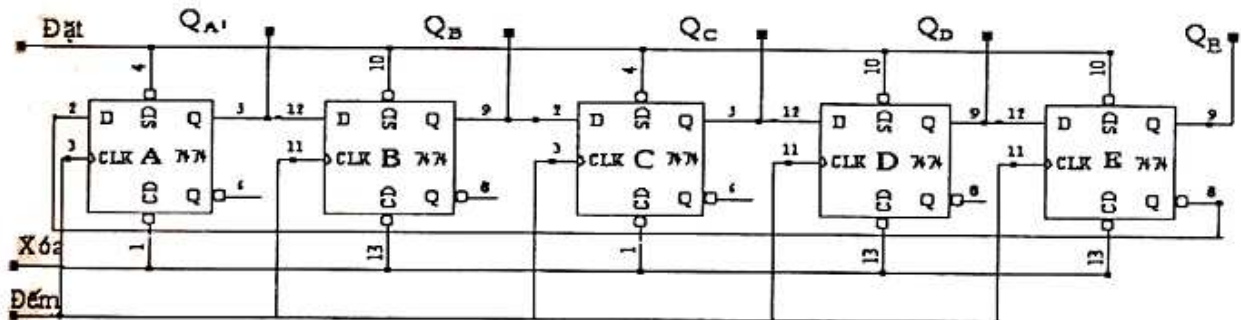
Quá trình đếm xảy ra như trong bảng chân lý của bộ đếm bảng 6.5

Bảng 6.5: Bảng chân lý của bộ đếm 10 mã Johnson

Xung	Q_A	Q_B	Q_C	Q_D	Q_E
0	<u>0</u>	0	0	0	<u>0</u>
1	<u>1</u>	<u>0</u>	0	0	0
2	1	<u>1</u>	<u>0</u>	0	0
3	1	1	<u>1</u>	<u>0</u>	0
4	1	1	1	<u>1</u>	<u>0</u>
5	<u>1</u>	1	1	1	<u>1</u>
6	<u>0</u>	<u>1</u>	1	1	1
7	0	<u>0</u>	<u>1</u>	1	1
8	0	0	<u>0</u>	<u>1</u>	1
9	0	0	0	<u>0</u>	<u>1</u>
10	0	0	0	0	0

Ta cũng có thể xây dựng bộ đếm mã Johnson có $M = 2n$ (số chẵn) dùng trigơ D và trigơ RST. Sơ đồ mắc các bộ đếm mã Johnson dùng trigơ RS tương tự như là dùng trigơ JK vì có mối tương đồng giữa J với S và K với R.

Để lắp bộ đếm 10 mã Johnson dùng trigơ D ta có thể mắc mạch theo sơ đồ như hình 6.20.



Hình 6.20: Bộ đếm 10 mã Johnson dùng trigơ D 7474

Sơ đồ đếm 12 mã Johnson dùng trigơ RST, cách đấu giống sơ đồ dùng trigơ JK. Bảng mã đếm 12 lập giống bộ đếm 10, chỉ cần thêm 1 cột Q_F cho trigơ thứ 6.

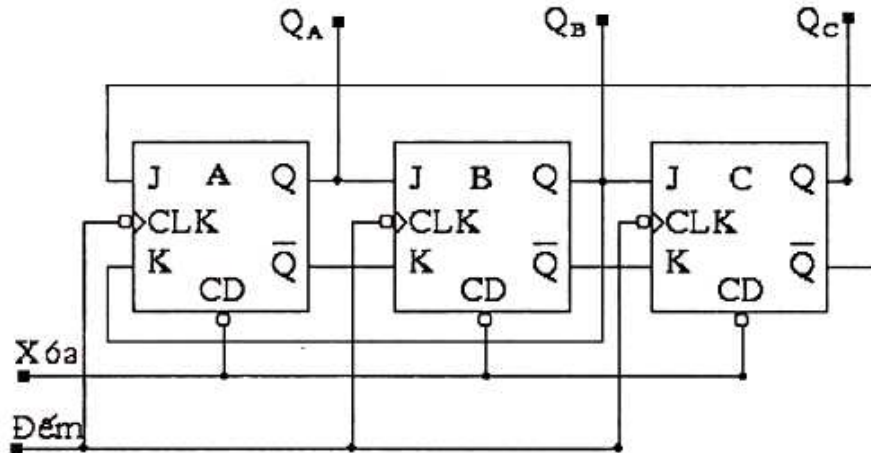
Bên cạnh các sơ đồ đếm chẵn ta cũng có thể xây dựng được các sơ đồ đếm lẻ theo mã Johnson.

Ví dụ: Bộ đếm 5 mã Johnson dùng trigơ JK có sơ đồ cho trên hình 6.21. Giải đồ xung của bộ đếm này được trình bày trên hình 6.22.

Chú ý: Đếm 5 thì mắc K_A với Q_B (cũng là J_C).

Hoạt động của bộ đếm diễn ra như bảng chân lý bảng 6.6.

Ta có thể giải thích quá trình đếm diễn ra theo bảng 6.6 như sau:



Hình 6.21: Sơ đồ mắc bộ đếm 5 mã Johnson dùng trigơ JK

Đầu tiên ta cho xung xoá vào lối vào xoá của bộ đếm:

$$Q_A = 0, Q_B = 0, Q_C = K_A = 0, \\ \bar{Q}_C = J_A = 1.$$

Như vậy, khi có xung nhịp thứ nhất đưa vào bộ đếm CLK_1 thì $Q_A = 1$ còn Q_B và Q_C vẫn bằng 0. Trạng thái lối vào của các trigơ trong bộ đếm sau khi có xung nhịp thứ nhất đưa vào bộ đếm:

$$J_A = J_B = 1, K_A = K_B = 0, \\ J_C = 0, K_C = 1$$

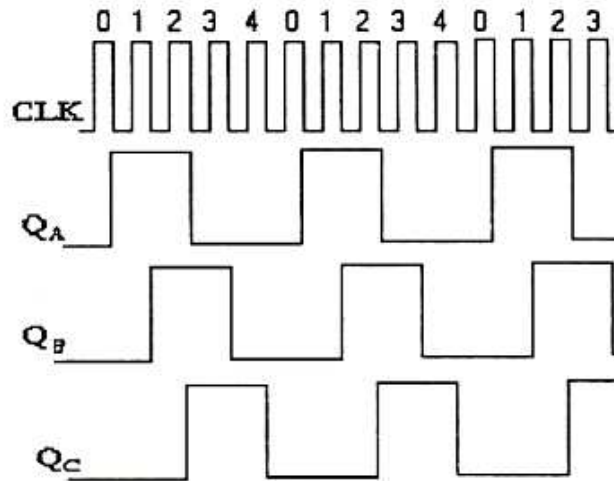
Khi có xung thứ hai CLK_2 vào bộ đếm thì $Q_A = 1$ và

$$Q_B = 1 \text{ còn } Q_C = 0.$$

Sau khi có xung thứ hai vào bộ đếm ta có: $J_A = 1, J_B = 1$ và $J_C = K_A = 1, K_B = K_C = 0$.

Vì $J_A = K_A = 1$ nên khi có xung CLK_3 vào bộ đếm Q_A chuyển trạng thái từ 1 về 0, $Q_B = 1, Q_C = 1$. Trạng thái lối vào của các trigơ sau khi có xung thứ 3 vào bộ đếm:

$$J_A = J_B = 0, J_C = 1, K_A = K_B = 1, K_C = 0.$$



Hình 6.22: Giản đồ xung của bộ đếm 5 mã Johnson

Khi có xung CLK_4 vào bộ đếm thì $Q_A = 0$, $Q_B = 0$, $Q_C = 1$.

Trạng thái các lối vào của các trigơ sau khi có xung thứ tư vào bộ đếm:

$$J_A = J_B = J_C = 0, K_A = 0, K_B = 1, K_C = 1.$$

Sau khi có xung thứ năm CLK_5 vào bộ đếm thì Q_C cũng chuyển về 0 nốt. Như vậy, sau khi có 5 xung vào bộ đếm cả ba trigơ đều chuyển trở về trạng thái ban đầu $Q_A = Q_B = 0$, $Q_C = 0$.

Bảng 6.6: Bảng chân lý của bộ đếm 5 mã Johnson

Xung	Q_A	Q_B	Q_C	$J_A = \bar{Q}_C$	$K_A = Q_B$	$J_B = Q_A$	$J_C = Q_B$
0	0	0	0	1	0	0	0
1	1	0	0	1	0	1	0
2	1	1	0	1	1	1	1
3	0	1	1	0	1	0	1
4	0	0	1	0	0	0	0
5	0	0	0	1	0	0	0

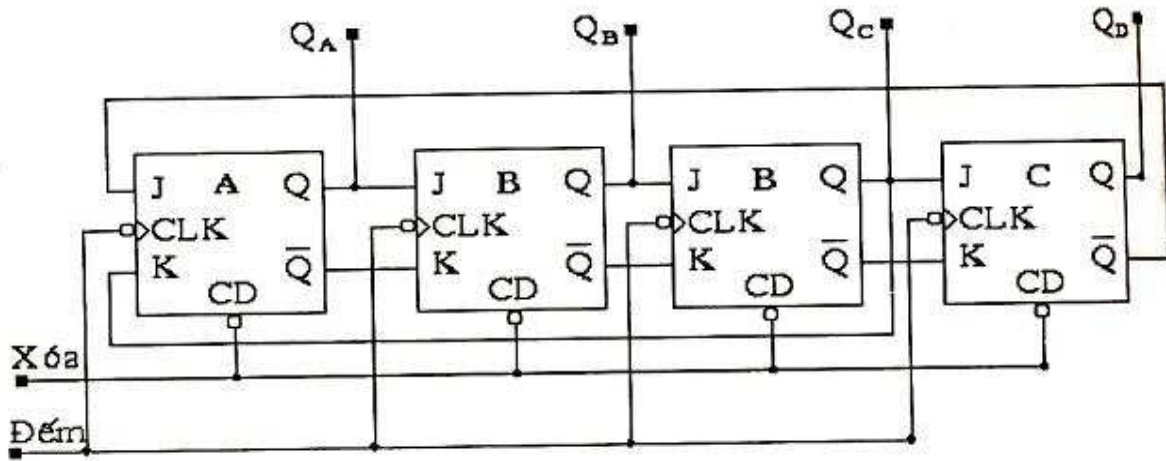
Với các bộ đếm Johnson có M là lẻ dùng trigơ JK ta có mối tương quan giữa M và n :

$$M = 2n - 1$$

Bằng cách mắc có cấu trúc tương tự như sơ đồ hình 6.21 chúng ta có thể mắc các bộ đếm với M là số lẻ tùy chọn.

Ví dụ: với $n = 4$ ta có bộ đếm $M = 7$. Bộ đếm 7 mã Johnson dùng 4 trigơ JK.

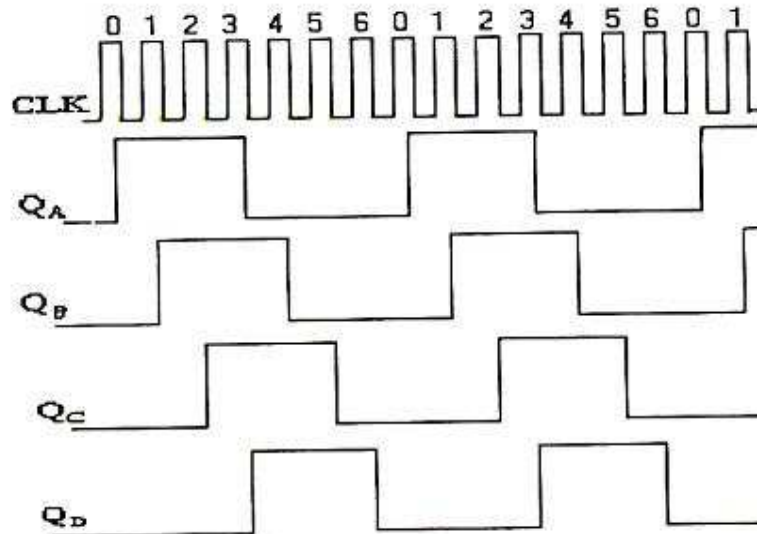
Sơ đồ bộ đếm này cho trên hình 6.23.



Hình 6.23: Sơ đồ đếm 7 mã Johnson

Muốn đếm 7 phải nối K_A với Q_C (cũng là J_D).

Chú ý: Các bộ đếm lẻ này không dùng được trigơ D và cũng không dùng được trigơ RS vì $R = S = 1$ cấm dùng.



Hình 6.24: Giải đồ xung bộ đếm 7 mã Johnson

Bảng chân lý của bộ đếm 7 mã Johnson cho trên bảng 6.7 và giải đồ xung của bộ đếm cho trên hình 6.24.

Bảng 6.7: Bảng chân lý của bộ đếm 7 mã Johnson

Xung	Q _A	Q _B	Q _C	Q _D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	0	1	1	1
5	0	0	1	1
6	0	0	0	1
7	0	0	0	0

6.4. CÁC BỘ GHI DỊCH (SHIFT REGISTER)

Bộ ghi dịch (còn gọi là thanh ghi dịch) là phần tử không thể thiếu được trong CPU, trong các hệ vi xử lý... Nó có khả năng ghi giữ và dịch thông tin (sang phải hoặc trái).

Bộ ghi dịch cấu tạo từ một dãy phần tử nhớ đơn bit (trigơ) được mắc liền tiếp với nhau và 1 số cửa logic cơ bản hỗ trợ.

Muốn ghi và truyền 1 từ nhị phân n bit ta cần n phần tử nhớ (n trigơ). Trong các bộ ghi dịch thường dùng các trigơ đồng bộ như trigơ RST, trigơ JK, trigơ D mà chúng ta đã khảo sát chúng ở chương V.

Thông thường người ta hay dùng các trigơ D hoặc các trigơ khác nhưng mắc theo kiểu trigơ D để tạo thành các bộ ghi.

Có 2 cách ghi:

+ Ghi song song: Các bit của từ nhị phân được ghi đồng thời cùng 1 lúc vào bộ ghi.

+ Ghi nối tiếp: Các bit của từ nhị phân được đưa vào bộ ghi một cách tuần tự theo thứ tự của từ nhị phân.

6.4.1. Bộ ghi song song

Sơ đồ bộ ghi song song 4 bit cho trên hình 6.25.

Trong sơ đồ trên người ta thêm 1 mạch điều khiển ra dùng 4 cửa VÀ 2 lối vào.

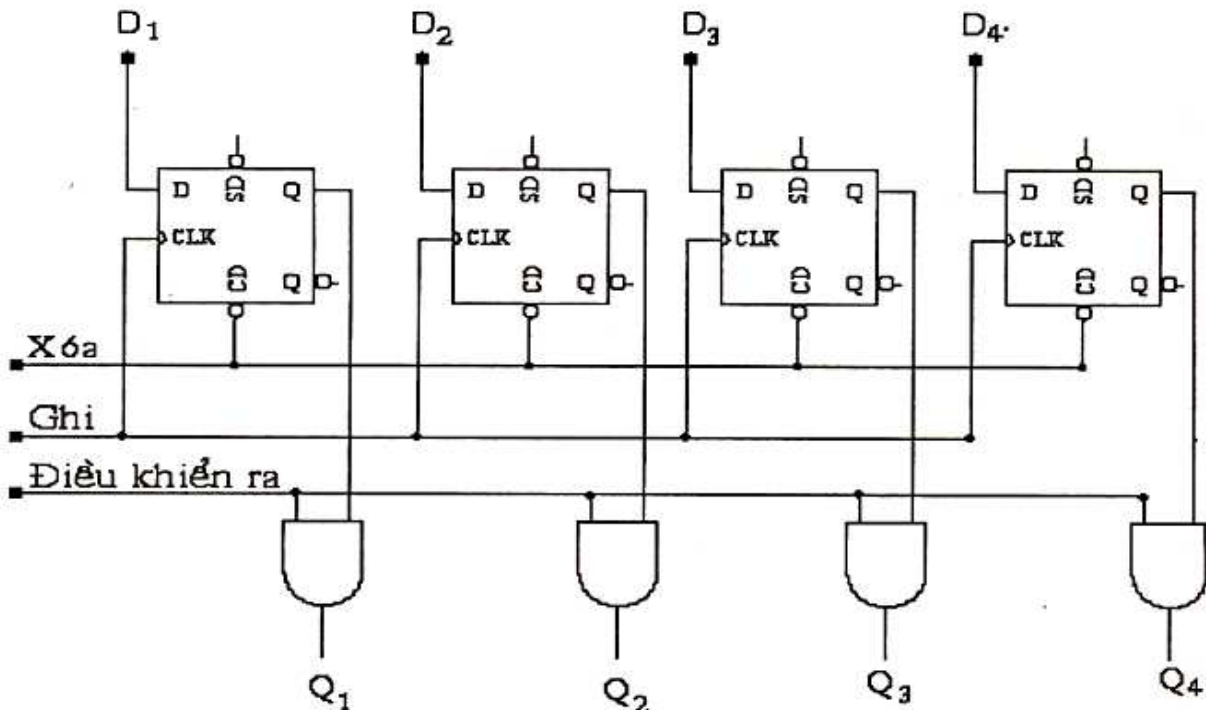
Hoạt động của sơ đồ như sau:

Trước tiên dùng xung xoá $CD = 0$ để xoá, lối ra $Q_1Q_2Q_3Q_4 = 0000$.

Các số liệu cần ghi đưa vào lối vào $D_1D_2D_3D_4$.

Khi có xung điều khiển ghi đưa vào lối vào CLK, dữ liệu được nạp vào bộ nhớ song song và cho lối ra song song $Q_1Q_2Q_3Q_4 = D_1D_2D_3D_4$.

Mỗi lối ra Q được đưa vào 1 lối vào của các cửa VÀ. Muốn cho dữ liệu ra bằng lối ra thì lối vào "điều khiển ra" phải bằng 1. Nếu chưa muốn cho dữ liệu ra lối ra thì để "điều khiển ra" bằng 0.

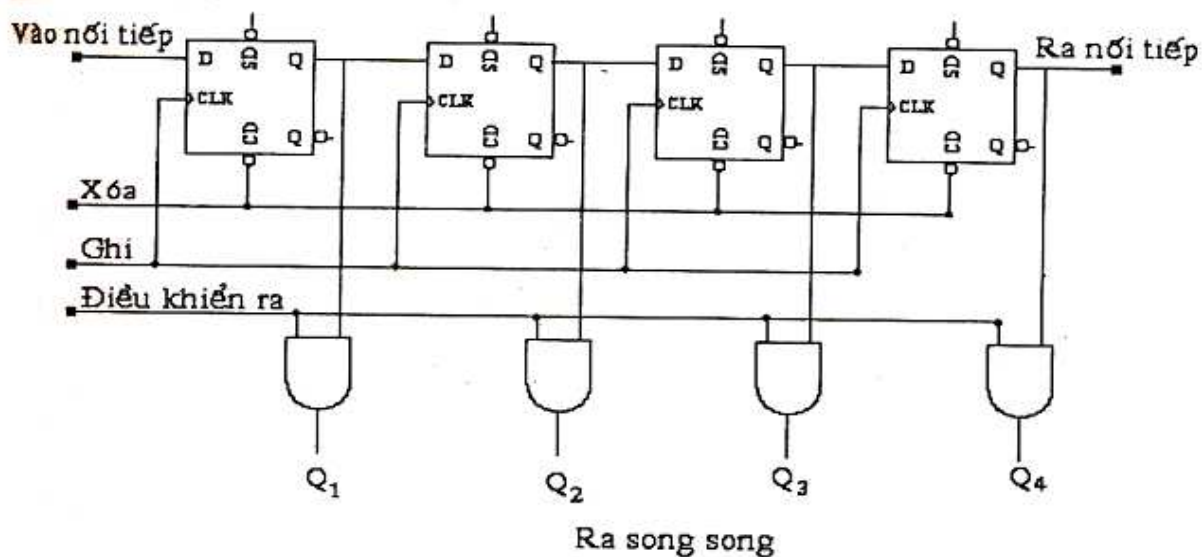


Hình 6.25: Sơ đồ bộ ghi song song 4 bit

6.4.2. Bộ ghi dịch nối tiếp

6.4.2.1. Bộ ghi nối tiếp dịch phải có các lối ra song song và ra nối tiếp

Bộ ghi nối tiếp có thể có dịch phải, dịch trái và cho ra song song hoặc ra nối tiếp. Trên hình 6.26 giới thiệu sơ đồ bộ ghi nối tiếp dịch phải có các lối ra song song và ra nối tiếp.



Hình 6.26: Bộ ghi nối tiếp dịch phải

Đây là sơ đồ chỉ có lối vào nối tiếp, còn lối ra có cả ra song song và ra nối tiếp.

Khi cho một xung kim âm tác động vào lối vào xóa, các lối ra Q của cả bốn trigơ trong bộ ghi đều ở trong trạng thái 0.

Muốn ghi ta phải đưa các bit thông tin nối tiếp về thời gian truyền lần lượt vào lối vào nối tiếp theo sự điều khiển đồng bộ của các xung nhịp. Cứ sau mỗi xung nhịp, trạng thái của trigơ lại được xác lập theo thông tin ở lối vào D của nó.

Trong sơ đồ hình 6.26, lối ra của trigơ trước lại được nối vào lối vào D của trigơ sau nên sau mỗi lần có xung nhịp tác động trigơ sau lại nhận giá trị của trigơ đứng trước nó.

Trên bảng 6.8 trình bày trạng thái lối ra của các trigơ trong bộ ghi theo số xung nhịp điều khiển ghi. Ta thấy qua mỗi lần xung nhịp điều khiển ghi các bit dữ liệu lại dịch sang phải một hàng nên người ta gọi bộ ghi hình 6.26 là bộ ghi nối tiếp dịch phải.

Bảng 6.8

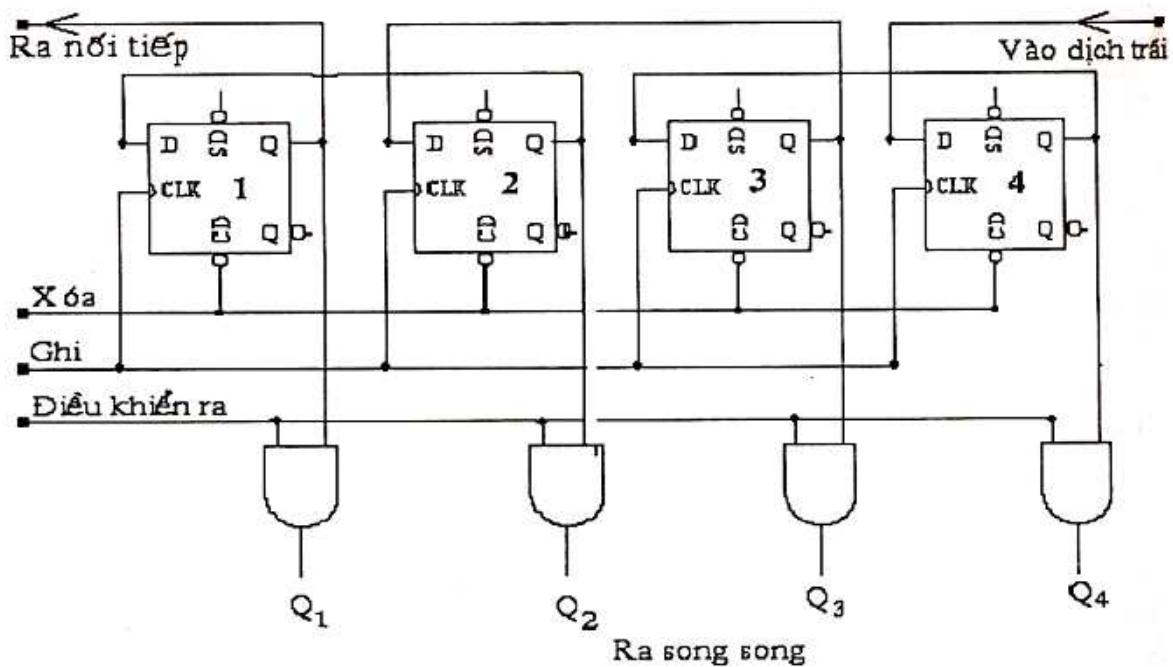
C_K	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	D_4	0	0	0
2	D_3	D_4	0	0
3	D_2	D_3	D_4	0
4	D_1	D_2	D_3	D_4

Sau khi kết thúc quá trình ghi, muốn đưa dữ liệu ra ở các lối ra song song ta đặt mức 1 ở lối điều khiển ra, lối ra của các cửa AND ở lối ra song song sẽ được xác lập theo trạng thái Q1,Q2,Q3,Q4 của các trigơ trong bộ ghi. Trong cách điều khiển dữ liệu ra song song này thông tin trong bộ ghi vẫn được duy trì.

Để điều khiển dữ liệu ra nối tiếp, ta phải tác động một nhóm 4 xung nhịp cũng ở lối vào điều khiển ghi. Sau 4 xung nhịp tác động 4 bit dữ liệu lần lượt được đưa ra khỏi bộ ghi.

Như vậy, quá trình điều khiển ghi nối tiếp 4 bit dữ liệu mới cũng là quá trình đưa 4 bit dữ liệu cũ ra khỏi bộ ghi qua lối ra nối tiếp.

6.4.2.2. Bộ ghi nối tiếp dịch trái có các lối ra song song và lối ra nối tiếp



Hình 6.27: Bộ ghi nối tiếp dịch trái, ra song song và nối tiếp

Bộ ghi nối tiếp dịch trái có các lối ra song song và lối ra nối tiếp được trình bày trên hình 6.27.

Cấu trúc của bộ ghi này cũng tương tự như bộ ghi dịch phải hình 6.26, nó chỉ khác trật tự sắp xếp các trigơ trong bộ ghi.

Trigơ 4 lại là trigơ đầu, trigơ 1 là trigơ cuối.

Quá trình điều khiển xoá, điều khiển ghi vào và đưa dữ liệu ra hoàn toàn tương tự như bộ ghi dịch phải hình 6.26.

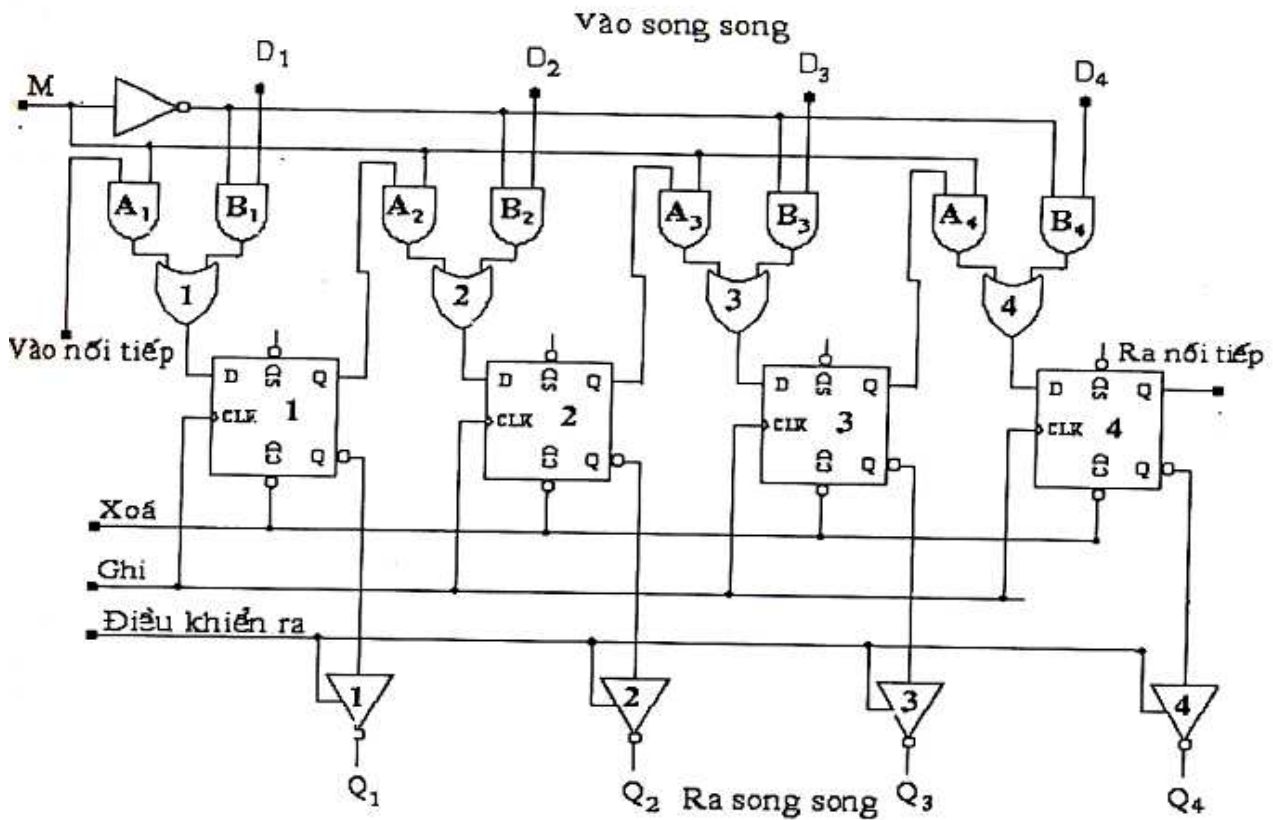
Ví dụ: ta có một chuỗi dữ liệu $D_1D_2D_3D_4$ được truyền đến lối vào của bộ ghi theo trình tự bit D_1 đến trước nhất. Quá trình ghi dịch bốn bit dữ liệu đối với bộ đếm này diễn ra như bảng 6.9.

Bảng 6.9: Quá trình ghi dịch trái

C_K	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	0	0	0	D_1
2	0	0	D_1	D_2
3	0	D_1	D_2	D_3
4	D_1	D_2	D_3	D_4

6.4.3. Mạch vừa ghi nối tiếp dịch phải, vừa ghi song song

Sơ đồ bộ ghi vừa ghi nối tiếp dịch phải, vừa ghi song song cho trên hình 6.28.



Hình 6.28: Bộ ghi nối tiếp dịch phải và ghi song song

Sơ đồ có thêm lối vào M: một điều khiển ghi song song hoặc ghi dịch nối tiếp.

Khi $M = 1$: các lối ra của các cửa AND B_1, B_2, B_3, B_4 đều ở mức 0, lối ra của trigơ trước được nối vào lối vào D của trigơ sau, lối vào nối tiếp được nối vào lối vào D của trigơ 1. Thông tin từ các lối vào song song không được truyền vào trong mạch. Như vậy, khi $M = 1$ mạch lúc này trở thành bộ ghi nối tiếp dịch phải. Muốn ghi đủ 4 bit vào thanh ghi ta phải có một nhóm 4 xung đưa vào C_K điều khiển ghi. Khi quá trình ghi kết thúc, muốn đưa ra ở các lối ra song song ta chỉ cần đặt mức logic 1 ở chân điều khiển cho phép ra của cửa 3 trạng thái. Muốn lấy ra ở lối ra nối tiếp ta cần phải tác động vào lối vào xung nhịp một nhóm gồm 4 xung thì toàn bộ 4 bit được đẩy ra khỏi bộ ghi dịch 4 bit. Như vậy, trong sơ đồ mạch hình 6.28 lối vào xung nhịp được dùng để điều khiển ghi nối tiếp đồng thời cũng là điều khiển ra. Cứ mỗi lần ghi 4 bit mới vào bộ ghi thì đồng thời có 4 bit cũ được đưa ra khỏi thanh ghi.

Khi $M = 0$, các lối ra của các cửa AND A_1, A_2, A_3, A_4 ở mức 0. Thông tin không được truyền vào lối vào D của trigơ 1, thông tin từ lối ra Q của trigơ trước không được truyền vào lối vào D của trigơ sau. Thông tin từ các lối vào song song được truyền trực tiếp vào các lối vào D của các trigơ. Mạch trở thành bộ ghi song song ra song song. Muốn ghi chỉ cần 1 xung nhịp điều khiển ghi là dữ liệu từ các lối vào song song được nạp vào bộ ghi. Muốn đưa số liệu ra lối ra song song đặt chân điều khiển cho phép ra của cửa 3 trạng thái bằng 1.

6.4.5. Bộ ghi nối tiếp vừa dịch phải vừa dịch trái

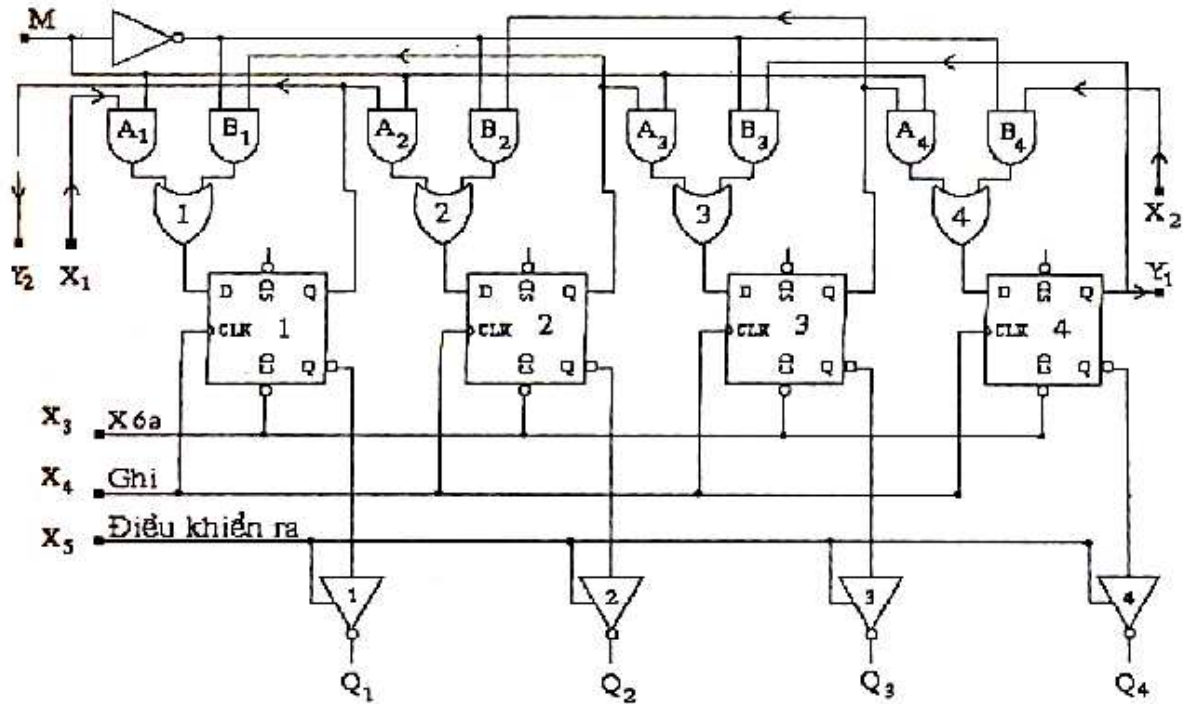
Sơ đồ mạch cho trên hình 6.29

Khi $M = 1$: các lối ra của các cửa AND B_1, B_2, B_3, B_4 đều ở mức 0, đường truyền thông tin từ lối vào X_2 ra Y_2 bị ngắt. Đường truyền thông tin theo chiều từ X_1 ra Y_1 được thiết lập. Mạch trở thành bộ ghi dịch phải nối tiếp, quá trình ghi, điều khiển dữ liệu ra nối tiếp hoặc ra song song diễn ra giống như là quá trình ghi dịch phải trong các bộ đếm hình 6.26, 6.28.

$M = 0$: Mạch trở thành bộ ghi dịch trái nối tiếp, có đường vào X_2 ra nối tiếp Y_2 hoặc ra song song.

Sau đây xin giới thiệu một số vi bộ ghi dịch thường gặp :

54/ 74 195, 54/ 74 S195, 54/ 74 LS195A: 4 bit, vào song song hoặc nối tiếp, ra song song, có đầu xoá.



Hình 6.29: Bộ ghi nối tiếp dịch phải, dịch trái

54/74 91A, 54/74 L91, 54/74 LS91: bộ ghi dịch phải 8 bit vào ra nối tiếp.

54/74 96, 54/74 LS96: bộ ghi dịch 5 bit vào song song hoặc nối tiếp, ra song song, có dấu xoá.

54/74 164, 54/74 LS164: bộ ghi dịch 8 bit vào nối tiếp ra song song, có dấu xoá.

54/74 199: 8 bit, vào song song hoặc nối tiếp, ra song song, có dấu xoá.

6.5. CÁC BỘ CHIA TẦN

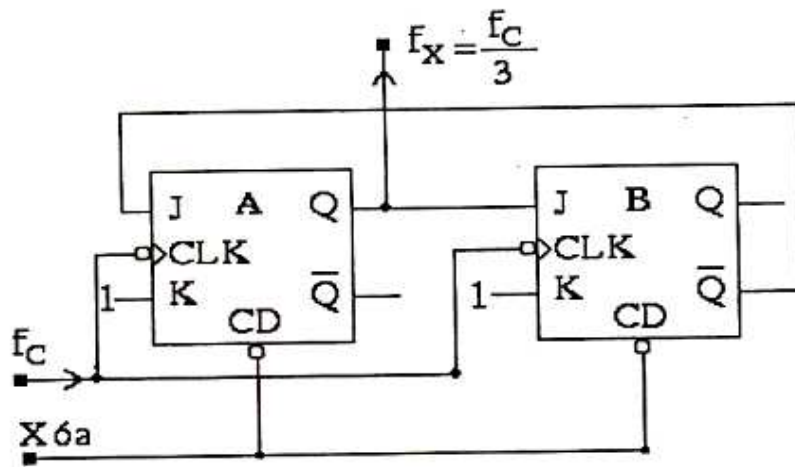
Các bộ chia tần được sử dụng rộng rãi trong kỹ thuật số, chẳng hạn như để tạo được các xung đồng hồ có độ ổn định tần số cao người ta thường thực hiện bằng cách: từ mạch tạo dao động dùng thạch anh tạo ra các xung cao tần có độ ổn định tần số cao, rồi dùng các bộ chia tần với hệ số chia N xác định để tạo ra các xung đồng hồ có tần số thấp có độ ổn định tần số cao dùng để đồng bộ các khối chức năng khác nhau của thiết bị.

Các bộ đếm ta đã khảo sát ở các mục trước đây thực chất cũng là các bộ chia tần. Gọi hệ số chia là M . Trong trường hợp hệ số chia $M = 10^n$ ta có thể

dùng bộ đếm 10 mã BCD, số trigơ là n , nếu dùng bộ đếm nhị phân ta có quan hệ $N = 2^n$. Hệ số chia dùng các mạch đếm nhị phân luôn là một số chẵn. Như vậy việc chia tần chẵn thực hiện rất đơn giản. Sau đây xin giới thiệu một số bộ chia có hệ số chia là 1 số lẻ (3, 5, 7...).

6.5.1. Mạch chia 3

Sơ đồ mạch cho trên hình 6.30. Sơ đồ dùng 2 trigơ JK. Tín hiệu cần chia tần f_c đưa vào các C_K . Các chân K để ở mức 1. Lối ra có tần số $f_x = \frac{f_c}{3}$ lấy ra ở Q_A . Phân tích hoạt động của sơ đồ ta có bảng chân lý 6.10.

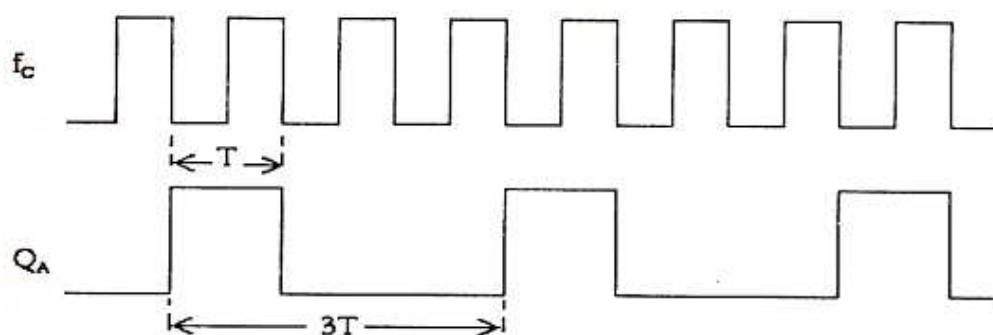


Hình 6.30: Sơ đồ mắc mạch bộ chia 3

Bảng 6.10: Bảng chân lý của bộ chia 3

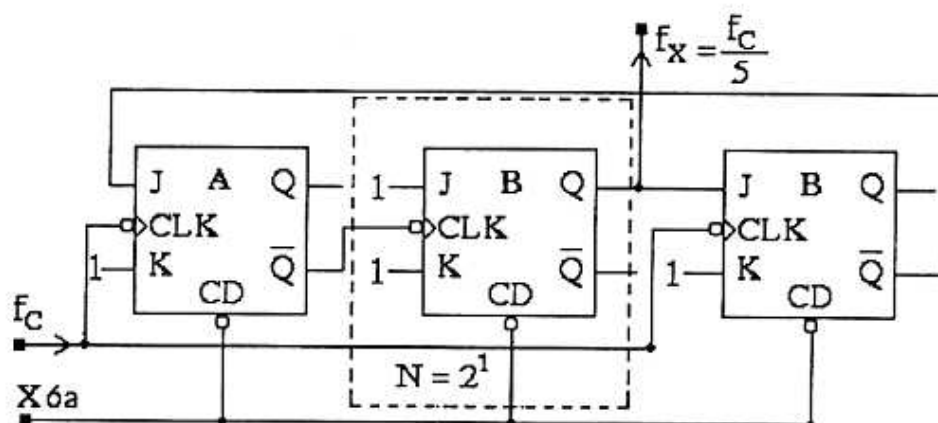
xung f_c	Q_A	Q_B
0	0	0
1	1	0
2	0	1
3	0	0

Từ bảng chân lý 6.10 ta vẽ được giản đồ xung như trong hình 6.31. Ở đây ta thấy cứ 3 xung lối vào (tín hiệu f_c) xung có chu kỳ T , ở lối ra Q_A ta được 1 xung có chu kỳ $3T$.



Hình 6.31: Giảm đồ xung bộ chia 3

Như vậy là ở lối ra Q_A ta nhận được tín hiệu có tần số $f_x = \frac{f_c}{3}$.



Hình 6.32: Sơ đồ mắc mạch bộ chia 5

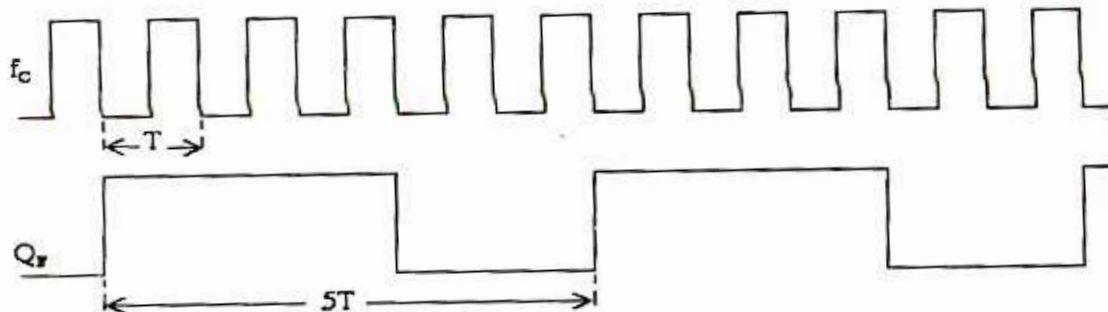
Mạch chia 5 : Mắc tương tự như trên nhưng cần 3 trigơ. Sơ đồ mạch cho trên hình 6.32. Bảng 6.11 là bảng chân lý của mạch chia 5.

Bảng 6.11: Bảng chân lý của mạch chia 5

xung f_c	Q_A	Q_B	Q_C
0	0	0	0
1	1	1	0
2	0	1	1
3	0	1	0
4	1	0	1
5	0	0	0

Các lối vào J, K bỏ trống đều phải để ở mức 1. Lối ra $f_x = \frac{f_c}{5}$ lấy ở Q_B

Phân tích hoạt động của mạch theo xung của tín hiệu f_c ta được bảng 6.11 và vẽ được giản đồ xung như trong hình 6.33.



Hình 6.33: Giản đồ xung bộ chia 5

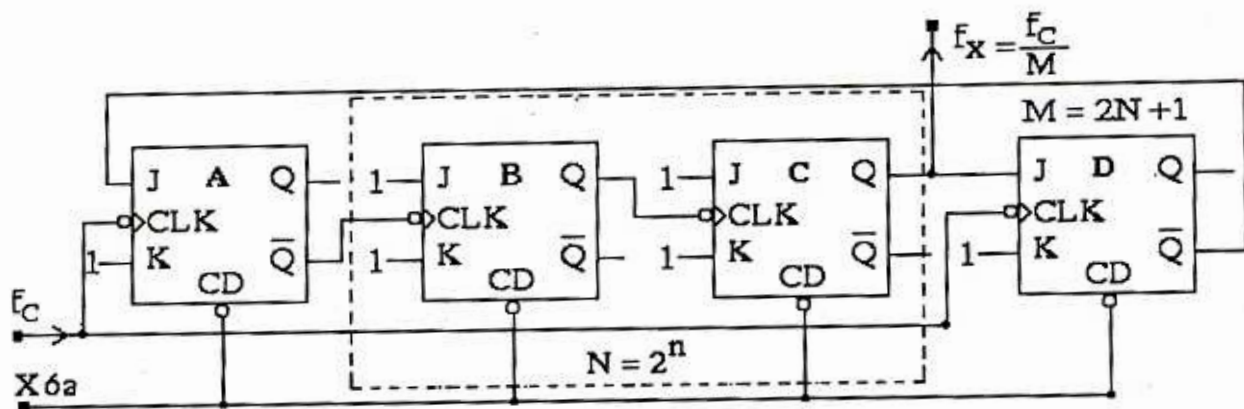
Nhìn giản đồ xung ta thấy cứ 5 xung của tín hiệu f_c vào thì có 1 xung ở lối ra Q_B . Như vậy tần số tín hiệu ở lối ra Q_B là :

$$f_x = \frac{f_c}{5}$$

Một cách tổng quát ta có thể dùng sơ đồ mắc theo kiểu tương tự như bộ chia 5 để tạo các bộ chia có hệ số chia là $M = 2N + 1$. Bằng cách thay đổi số trigger trong bộ đếm nhị phân N (nằm trong khung chữ nhật được vẽ bằng các đường không liền nét). Lối ra của bộ chia M được lấy ở lối vào J của trigger cuối cùng. Ví dụ như bộ chia M vẽ trên hình 6.28 trong bộ chia này bộ đếm nhị phân N có 2 trigger: $n = 2$ hệ số chia M của bộ đếm này là:

$$M = 2.2^2 + 1 = 9$$

Sơ đồ cho trên hình 6.34 chính là bộ chia 9.



Hình 6.34: Sơ đồ bộ chia 9

Chương VII

BỘ NHỚ BÁN DẪN

1.1. KHÁI NIỆM CƠ BẢN

Chức năng của bộ nhớ

Trong mục 6.3 khi khảo sát các bộ ghi chúng ta đã nhắc đến khả năng lưu trữ của các bộ ghi. Các bộ ghi thường được dùng để lưu trữ thông tin về trạng thái của máy hoặc số liệu trung gian trong một thời gian ngắn. Để lưu trữ được các chương trình điều khiển tính toán và các kết quả tính toán trong thời gian dài cần phải có các bộ nhớ có dung lượng lớn. Đối với các thiết bị số và máy tính điện tử, khả năng nhớ được dữ liệu là một yêu cầu quan trọng. Ví dụ trong máy tính, các chương trình tính toán các con số cần thiết cho phép toán phải được lưu trữ ngay trong máy. Còn trong các thiết bị điều khiển số thì các lệnh điều khiển cũng phải được lưu trữ để điều khiển quá trình vận hành theo một trình tự đã được định trước do người lập trình vạch ra. Vì vậy các bộ nhớ đảm nhiệm chức năng lưu trữ thông tin không thể thiếu được trong các thiết bị số.

Các thông số cơ bản của bộ nhớ

Thông tin dữ liệu được tạo thành từ một đơn vị cơ bản gọi là từ (word). Tùy theo từng loại máy, 1 từ có thể là 8 bit, 16 bit, 32 bit ... Các thiết bị chỉ truyền đi hay nhận vào nguyên 1 từ hay nhiều từ chứ không phải chỉ vài bit của từ. Tuy nhiên đơn vị nhớ cơ bản là bit.

Khi nói đến bộ nhớ, người ta quan tâm ngay đến 2 thông số cơ bản :

1. Dung lượng nhớ (Capacity): là lượng thông tin hay dữ liệu lưu trữ được trong bộ nhớ .

Các đơn vị nhớ là: bit, byte, KB, MB, GB.

$$1 \text{ byte} = 8 \text{ bit}$$

$$1 \text{ KB} = 2^{10} \text{ byte} = 1024 \text{ byte} = 2^{13} \text{ bit}$$

$$1 \text{ MB} = 2^{10} \text{ KB} = 1024 \text{ KB}$$

$$1 \text{ GB} = 2^{10} \text{ MB} = 1024 \text{ MB}$$

2. Thời gian truy nhập (Access time - at): Thời gian này gồm 2 phần là thời gian xác định vị trí từ và thời gian đọc nội dung từ lưu trữ trong bộ nhớ. at càng nhỏ, tốc độ làm việc của máy càng cao.

Phân loại bộ nhớ

Việc phân loại bộ nhớ có nhiều cách. Xét về cấu trúc của máy tính ta thấy có 2 loại bộ nhớ:

* Bộ nhớ trong : gồm có ROM, RAM, đĩa cứng

* Bộ nhớ ngoài : gồm các đĩa mềm, các đĩa CD ROM...

Các thiết bị nhớ đĩa cứng, đĩa mềm và CDROM bạn đọc có thể tham khảo trong các sách nói về cấu trúc máy vi tính. Trong phạm vi giáo trình kỹ thuật điện tử này chúng ta chỉ đề cập đến các bộ nhớ bán dẫn, các bộ nhớ này có trong các máy tính điện tử và các thiết bị điện tử điều khiển số.

Các bộ nhớ bán dẫn được chia làm hai loại:

– Bộ nhớ truy cập ngẫu nhiên RAM (Random Acces Memory), RAM là bộ nhớ bán dẫn tác động nhanh, có thể ghi số liệu vào và đọc số liệu ra từ RAM ở thời điểm nào cũng được.

– Bộ nhớ chỉ đọc ra ROM (Read Only Memory) trong đó các dữ liệu đã được các nhà sản xuất ghi vào và nó chỉ được đọc ra khi dùng.

7.2. BỘ NHỚ CHỈ ĐỌC - ROM

ROM là bộ nhớ vĩnh viễn, khác với RAM thông tin chứa trong ROM không bị mất đi khi không còn nguồn điện. Các dữ liệu đã được nạp vào ROM do nhà chế tạo thực hiện khi sản xuất có thể là các hằng số vật lí, toán học như số π , số e, các công thức toán học, các hàm số lượng giác sin, cos,... các bộ biến đổi mã, giải mã các kí tự v.v... Dữ liệu cũng có thể là các lệnh điều khiển khởi động máy tính, các chương trình con điều khiển sự hoạt động của máy tính hay các thiết bị điều khiển tự động. Nó chỉ dùng để đọc ra trong quá trình vận hành của thiết bị. Trong ROM còn có các loại:

PROM (Programmable ROM) là bộ nhớ được chương trình hoá tại nhà máy sản xuất.

Các vi mạch loại PROM: 54/74S188 (32×8bit), 54/74S287 (256×4bit), 54/74S472 (512×8bit).

EPROM (Erasable Programmable ROM) là bộ nhớ có thể lập trình hoá có thể xoá và nạp lại được. EPROM có hai loại:

UV-EPROM (Ultra - Violet). Bộ nhớ lập trình hoá có thể xoá bằng tia cực tím, ta có thể dễ dàng nhận ra loại này từ hình dạng bên ngoài của nó loại này được đóng trong vỏ nhựa có cửa sổ tròn trong suốt, qua cửa sổ ta có thể nhìn chíp bán dẫn ở bên trong. Nhờ có cửa sổ này ta có thể xoá nội dung thông tin ghi trong EPROM bằng cách dùng đèn tử ngoại chiếu tia cực tím vào chíp. Bộ nhớ sau khi xoá có thể lập trình lại. Khi dùng loại EPROM này cần che cửa sổ để tránh ánh sáng mặt trời rọi vào làm mất thông tin ghi trong mạch.

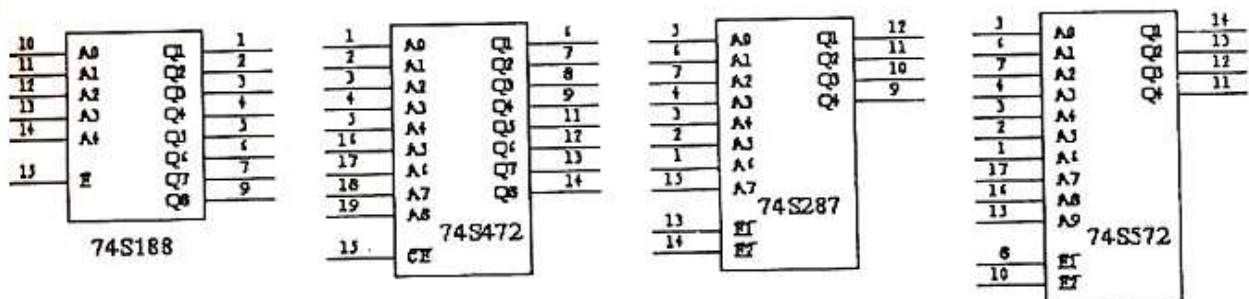
Các vi mạch UV Erasable PROM của hãng National Semiconductor: M2708 (1028×8bit), MM2704 (512×8bit).

E - EPROM (Electically Programmable and Erasable ROM). Bộ nhớ lập trình hoá xoá bằng xung điện 20 V. Các vi mạch thuộc loại này của hãng National Semiconductor có các kí hiệu sau:

MM4203/MM5203 2048bit (256×8 hoặc 512×4bit);

MM4204/MM5204 4096bit (512×8bit).

Trên hình 7.1 nêu kí hiệu của một số vi mạch nhớ chỉ đọc PROM.



Hình 7.1: Kí hiệu các vi mạch PROM

7.3. BỘ NHỚ TRUY CẬP NGẪU NHIÊN - RAM

RAM là bộ nhớ vừa đọc ra vừa ghi được dữ liệu vào. Thông tin chứa trong RAM sẽ bị mất khi không còn nguồn điện nuôi nó, nó được chia ra các loại:

- RAM lưu trữ (Nonvolatile): Dùng CMOS nuôi bằng pin lithium liên tục được 10 năm.

- RAM không lưu trữ (Volatile): Khi ngắt nguồn nuôi các dữ liệu đã nhớ bị xoá. Loại RAM này lại có hai loại:

+ RAM tĩnh (Static): Cấu tạo từ các trigơ lưỡng cực, MOS hoặc CMOS. Ta gọi là RAM tĩnh vì các dữ liệu sẽ còn nguyên vẹn khi nguồn nuôi còn duy trì.

+ RAM động (Dynamic): Cấu tạo từ các transistor MOS hoặc CMOS có thêm các tụ điện để lưu trữ số liệu. Vì qua tụ có dòng dò nên theo thời gian điện thế trên tụ giảm dần nên số liệu không duy trì được nguyên vẹn mặc dù chưa ngắt nguồn. Để duy trì được dữ liệu chứa trong RAM động ta phải thường xuyên đưa các xung điện kích nạp điện cho tụ, thao tác này được gọi là "làm tươi" dữ liệu chứa trong RAM.

Các bộ nhớ nói trên được xây dựng từ các ô nhớ (Memory Cell). Các ô nhớ này có cấu tạo khác nhau ở các bộ nhớ khác nhau.

Ví dụ:

* RAM tĩnh: Đơn vị nhớ là 1 trigơ. Trigơ có thể cấu tạo từ transistor lưỡng cực, có thể từ MOSFET.

* RAM động: ô nhớ là MOSFET có sử dụng điện dung của cực cửa (có như có tụ C) chứ không mắc theo kiểu trigơ.

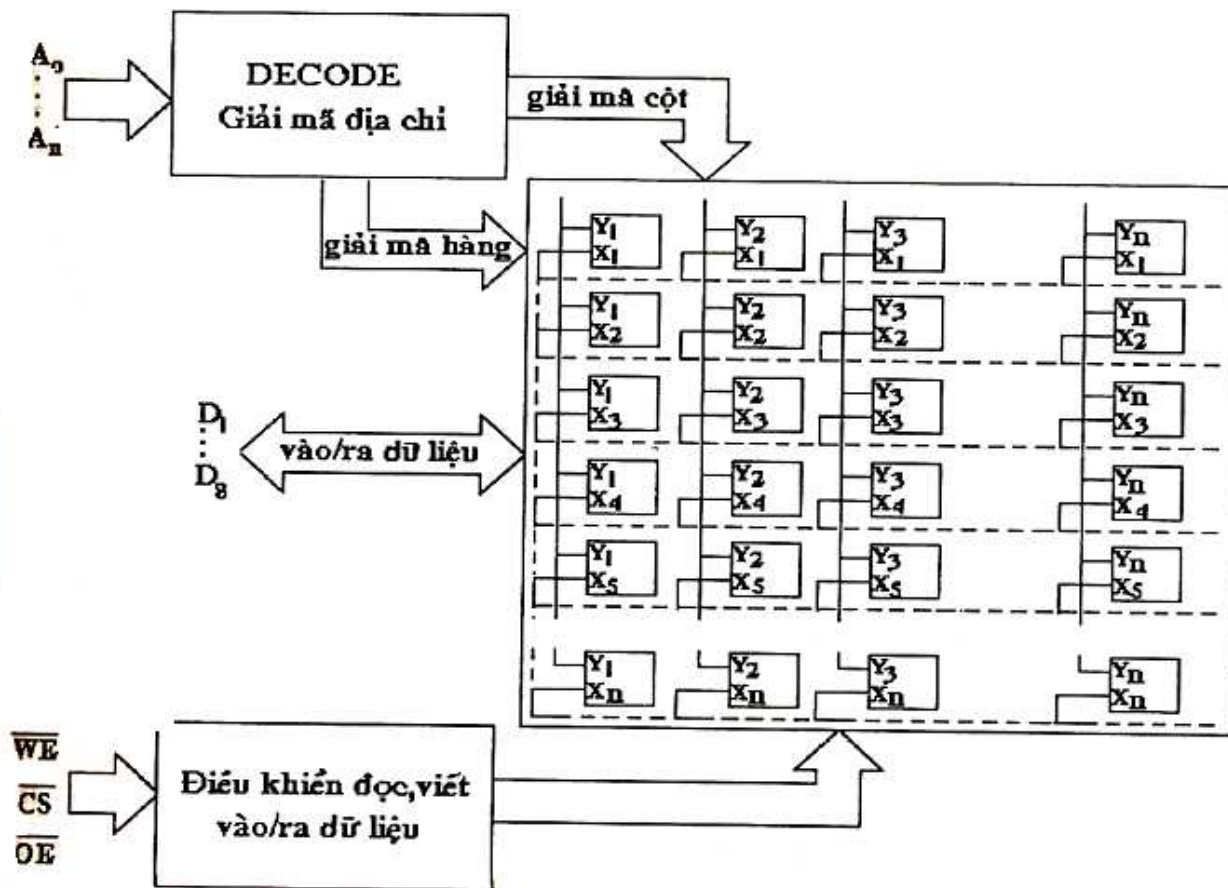
* ROM: đơn vị nhớ cấu tạo từ transistor lưỡng cực và MOSFET (có thể là diode, tụ điện).

7.3.1. Bộ nhớ truy cập ngẫu nhiên - RAM tĩnh

Cấu trúc tổng quát của RAM tĩnh: RAM được tổ chức dưới dạng một ma trận các ô nhớ. Mỗi ô nhớ là một trigơ như RST hoặc trigơ D. Để có thể ghi lưu trữ và đọc thông tin được dễ dàng, các ô nhớ được sắp xếp và định vị theo hàng và cột. Việc truy cập vào từng ô nhớ để đọc hoặc ghi thông tin được giải quyết nhờ khối giải mã địa chỉ hàng và giải mã địa chỉ cột. Theo nguyên tắc tổ chức nêu trên, người ta chế tạo các RAM tĩnh có $(2^n \times b)$ bit.

RAM có các lối vào địa chỉ (Address), các lối vào dữ liệu (Data), các lối vào điều khiển (Control) và các lối ra dữ liệu. Các RAM tĩnh dùng trong máy tính các đường dữ liệu vào ra thường kết hợp làm một nhờ dùng các cửa 3 trạng thái.

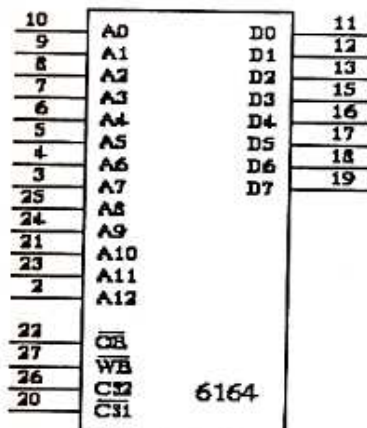
Trên hình 7.2 giới thiệu sơ đồ khối mô tả cấu trúc của RAM tĩnh.



Hình 7.2: Cấu trúc của RAM

Như ta đã thấy trên hình 7.2, cấu trúc của RAM gồm: ma trận các ô nhớ, khối giải mã địa chỉ và điều khiển đọc, viết, vào ra dữ liệu.

Vi mạch RAM 6164, 6264 có cấu trúc như hình 7.3.



Hình 7.3. Ký hiệu của RAM 6164, 6264

RAM 6164 có 13 đường địa chỉ là: A_0, A_1, \dots, A_{12} , 4 đường điều khiển và 8 đường dữ liệu vào và ra chung nhau là D_0, D_1, \dots, D_7 . Xem ký hiệu và các chân cụ thể của nó trên ký hiệu logic của RAM hình 7.3.

Đầu vào điều khiển:

CS : Chip Selection : chọn chip

OE : Out put Enable : Đầu điều khiển cho phép ra

WE : Write Enable : Đầu điều khiển cho phép ghi

Hoạt động của nó tuân theo bảng chân lý 7.1.

Bảng 7.1: Bảng chân lý của RAM tĩnh 6164, 6264

\overline{CS} \overline{WE} \overline{OE}	MỐT VẬN HÀNH	ĐẦU VÀO	ĐẦU RA
0 0 1	Ghi số liệu	Được nối	Thả nổi
0 1 0	Đọc số liệu	Thả nổi	Được nối
0 1 1	Không làm gì	Thả nổi	Thả nổi
1 x x	Ngừng	Thả nổi	Thả nổi

x : bất kỳ hoặc không quan tâm

Các thủ tục ghi dữ liệu vào RAM và đọc kết quả ra phải căn cứ theo bản chân lý nêu trên nhưng cần lưu ý các điểm sau đây:

1/ Ghi dữ liệu vào bộ nhớ RAM:

Trong khi tiến hành các thao tác ghi lối vào điều khiển chọn chip phải đặt ở mức 1: $\overline{CS} = 1$. Các đường vào ra dữ liệu để ở trạng thái thả nổi $\overline{OE} = 1$, $\overline{WE} = 1$, sau khi đã đặt xong các địa chỉ cần truy nhập và dữ liệu mới điều khiển ghi để các lối vào $\overline{OE} = 1$, $\overline{WE} = 0$, $\overline{CS} = 0$ để ghi. Ghi xong lại để lối vào điều khiển chọn chip $\overline{CS} = 1$.

2/ Đọc kết quả từ bộ nhớ ra :

Đặt $\overline{WE} = 1$, $\overline{CS} = 1$, đặt xong các địa chỉ cần truy cập mới điều khiển đọc đặt $\overline{OE} = 0$, $\overline{CS} = 0$.

Trên hình 7.4 trình bày sơ đồ khối mô tả cấu trúc của một ô nhớ trong RAM có chung đường vào ra. Khối giải mã địa chỉ xác định vị trí ô nhớ cần truy cập theo các thông tin đặt vào các lối vào địa chỉ, nó tạo ra tín hiệu cho phép truy cập vào ô nhớ. Giả sử ta muốn ghi dữ liệu vào ô nhớ, trình tự tiến hành như sau: $\overline{CS} = 1$, $\overline{OE} = 1$, $\overline{WE} = 1$, lối ra hai cửa AND 1 và AND 2, ở trạng thái 0, cửa ba trạng thái 1 và 2 thả nổi ô nhớ cách li hoàn toàn với đường vào ra. Đưa thông tin vào lối vào địa chỉ, đặt địa chỉ ô nhớ cần ghi. Đưa dữ liệu vào lối vào dữ liệu, chuyển các lối vào chọn chip về không và vẫn để lối vào $\overline{OE} = 1$, $\overline{WE} = 0$, $\overline{CS} = 0$, lối ra cửa AND 1 chuyển lên 1, lối ra cửa AND 2 chuyển về 0, cửa 3 trạng thái 1 thông mạch, cửa 3 trạng

thái 2 thả nổi làm hở mạch lối ra, đường vào ra trở thành lối vào dữ liệu qua cửa 3 trạng thái 1 được nạp vào ô nhớ.

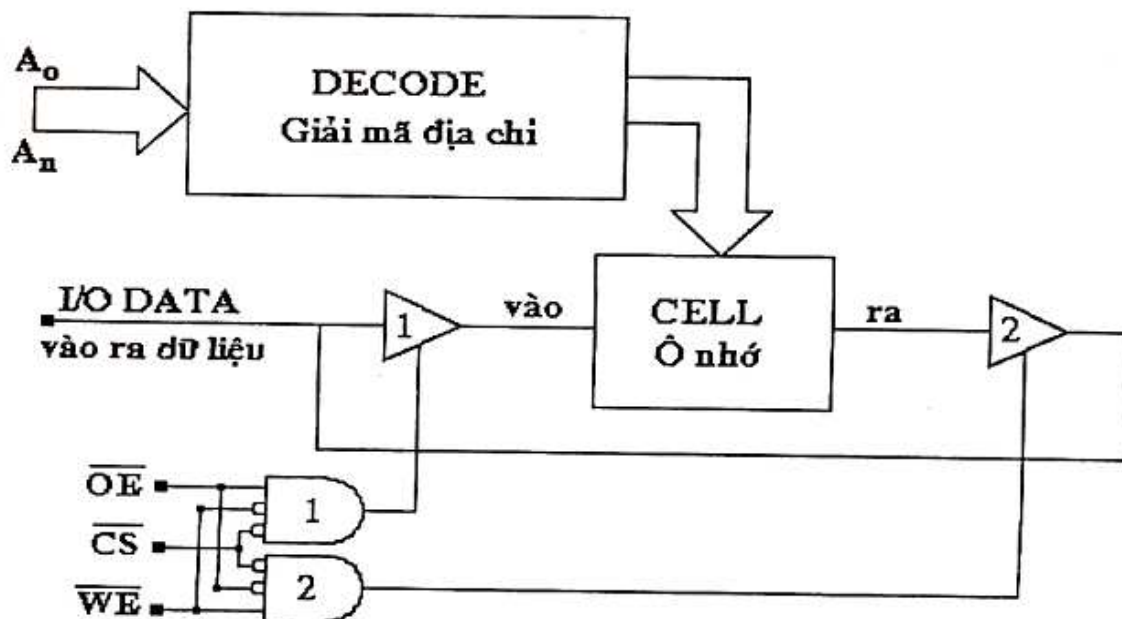
Muốn đọc dữ liệu từ ô nhớ ta cũng làm theo trình tự tương tự như thao tác ghi nhưng bước cuối ta đặt:

$$\overline{WE} = 1,$$

$$\overline{OE} = 0,$$

$$\overline{CS} = 0$$

lối ra cửa AND 1 chuyển về 0, lối ra cửa AND 2 chuyển lên 1 cửa 3 trạng thái 1 thả nổi làm hở mạch lối vào, cửa 3 trạng thái 2 thông mạch dữ liệu được đưa ra qua cửa 3 trạng thái 2.



Hình 7.4: Cấu trúc 1 ô nhớ của RAM tĩnh

7.3.2. Các bộ nhớ của RAM động

MOSFET có trở kháng vào rất lớn, người ta thường dùng nó để chế tạo bộ nhớ RAM động.

Cấu tạo một ô nhớ của RAM động dùng MOSFET cho trên hình 7.5.

Mạch có 3 MOSFET nhưng không đấu kiểu trigơ mà:

T_1, T_3 : Hoạt động như các khoá điện tử

T_2 : Như 1 mạch đảo.

Dữ liệu lưu trữ dưới dạng tích điện ở tụ điện C chính là điện dung cửa của T_2 . Đơn vị nhớ có 4 đường:

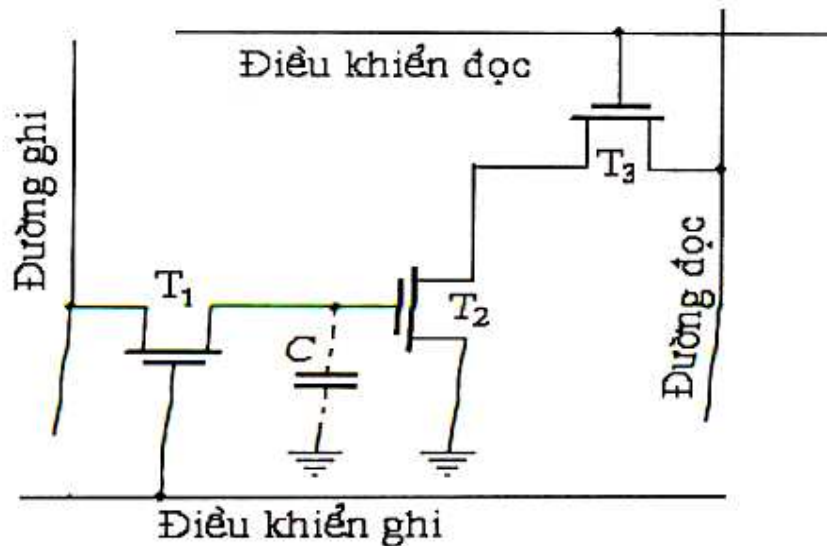
- Đường điều khiển đọc và điều khiển viết thuộc về 1 từ (WORD)
- Đường "Viết" để ghi dữ liệu vào đường "Đọc" để lấy dữ liệu ra.

* Thao tác viết : (Giả sử MOS kênh P)

- Cho xung âm vào "điều khiển viết", T_1 dẫn.
- Nếu đường "viết" ở 0V tụ C không tích điện : ghi bit "0"
- Nếu đường "viết" ở (- U) tụ C có điện tích : ghi bit "1"

* Thao tác đọc :

- Cho xung âm vào đường "điều khiển đọc" và vào đường "đọc"
- Nếu tụ C không có điện tích (bit "0") thì T_2 cấm làm T_3 cũng cấm. đường đọc không có dòng điện ra : đọc "0"
- Nếu tụ C có điện tích (bit "1") thì T_2 dẫn làm T_3 cũng dẫn đường đọc có xung dòng điện : đọc "1"



Hình 7.5: Cấu tạo 1 ô nhớ của RAM động

Mạch có thể đơn giản hơn nếu dùng 1 đường chung cho điều khiển viết và đọc - lúc đó cần chọn xung điều khiển viết nhỏ hơn xung điều khiển đọc để khi đọc T_1 không được dẫn vì T_1 dẫn sẽ làm thay đổi dữ liệu ở C.

Vì mạch luôn mất dần điện tích trên tụ điện C (do hiện tượng dò) nên dữ liệu không lưu trữ được vĩnh viễn. Do đó mạch cần được "viết" lại liên tục gọi là làm tươi (Refresh) bộ nhớ (chu kỳ viết lại cỡ 2 m sec).

* Đặc điểm của RAM động:

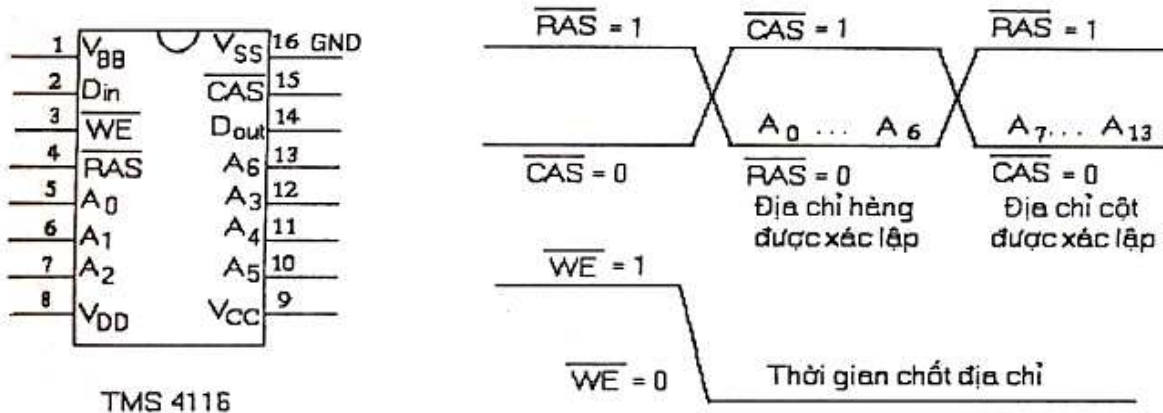
- Khi không "đọc" không "viết" hoặc không viết lại, mạch hầu như không tiêu thụ công suất do đó công suất tiêu thụ cực nhỏ.

- Vì chỉ sử dụng ít transistor cho 1 ô nhớ nên dễ chế tạo mạch có dung lượng lớn trên diện tích nhỏ.

- Thời gian thâm nhập at nhỏ hơn RAM tĩnh.

- Nhược điểm: Luôn phải Refresh, không phù hợp khi ghép với họ TTL.

Ví dụ : RAM động TMS 4116 có kí hiệu và giản đồ thời gian xác lập địa chỉ bộ nhớ được trình bày trên hình 7.6.



Hình 7.6: Ký hiệu RAM động TM4116 và giản đồ thời gian chọn địa chỉ khi đọc, viết

TMS 4116 thuộc loại MOSFET có cấu trúc $16K \times 1$ bit.

16K tương ứng 16.384 ô nhớ, vậy cần 14 đường địa chỉ (vì $2^{14} = 16.384$)

Để giảm số đường địa chỉ tức là số chân IC người ta dùng địa chỉ kiểu hợp kênh (multiplex) tức là cần 7 đường khi đọc hay viết:

đầu tiên dùng 7 địa chỉ chứa thông tin hàng,

sau đó dùng 7 địa chỉ chứa thông tin cột

Vì vậy 2 tín hiệu **RAS** và **CAS** có biểu đồ thời gian ngược pha nhau như hình vẽ trên giản đồ xung hình 7.6.

Khi **RAS** thấp: Mạch chốt địa chỉ hàng thông ($A_0 \dots A_6$)

Khi **CAS** thấp: Mạch chốt địa chỉ cột thông ($A_7 \dots A_{13}$)

WE : để đưa tín hiệu cho phép "viết" vào mạch.

Khi viết : **WE** : thấp, thông tin qua Din được "viết" vào địa chỉ đã chọn

Khi đọc : **WE** : cao, từ địa chỉ đã chọn thông tin ra Dout.

Chương VIII

BỘ CHUYỂN ĐỔI SỐ - TƯƠNG TỰ VÀ TƯƠNG TỰ SỐ (DIGITAL TO ANALOG & ANALOG TO DIGITAL CONVERTER)

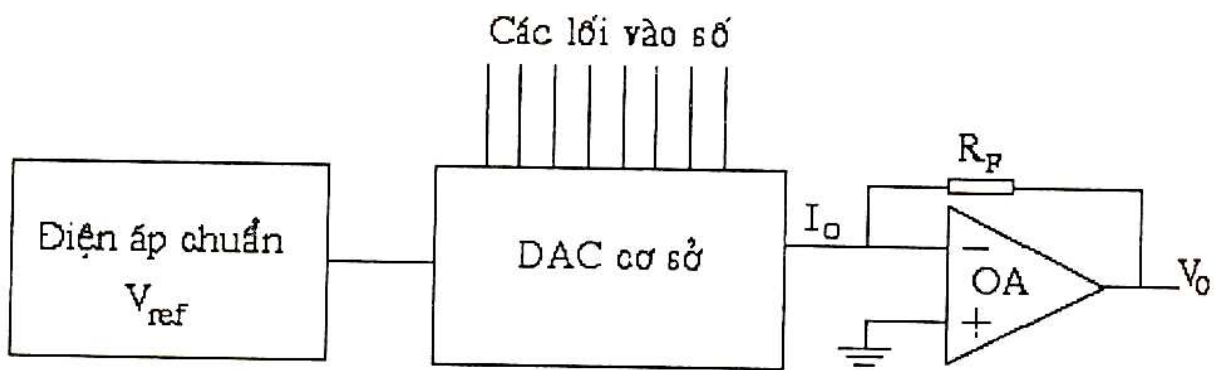
8.1. KHÁI QUÁT VỀ DAC

1. Chức năng

Bộ chuyển đổi số - tương tự DAC tiếp nhận một mã số n bit song song ở lối vào và biến đổi nó ra dòng điện hoặc điện áp tương tự ở lối ra. Dòng điện hoặc điện áp ở lối ra DAC là hàm số của mã số lối vào và phải biến thiên phù hợp với sự biến thiên của mã số này.

2. Cấu trúc

Sơ đồ khối của bộ biến đổi DAC hoàn chỉnh bao gồm 3 phần tử cơ bản: một điện áp chuẩn (referent voltage) ổn định bên ngoài, một DAC cơ sở và một mạch khuếch đại thuật toán (hình 8.1).



Hình 8.1: Sơ đồ khối bộ biến đổi DAC

Với một mã số nhị phân tự nhiên ở lối vào của DAC, ta có điện áp V_0 ở lối ra là:

$$V_0 = V_{REF} (B_1 \cdot 2^{-1} + B_2 \cdot 2^{-2} + \dots + B_n \cdot 2^{-n}) \quad (1)$$

trong đó:

B_n : là bit thấp nhất

B_1 : là bit cao nhất

V_{REF} : thế chuẩn

DAC cơ sở được tạo thành từ những chuyển mạch tương tự được điều khiển bởi mã số ở lối vào và một loạt điện trở chính xác. Nhờ có các chuyển mạch tương tự điều khiển dòng hoặc điện áp trích từ điện áp chuẩn tạo nên dòng điện hoặc điện áp lối ra tương tự với mã đó.

Mạch khuếch đại thuật toán dùng để chuyển đổi dòng điện thành điện áp và đồng thời có chức năng của tầng đệm. Nhưng do có một khoảng thời gian trễ trong quá trình thiết lập trạng thái ở lối ra theo sự biến đổi dòng điện ở lối vào của bộ khuếch đại thuật toán cho nên trong các ứng dụng cần tốc độ cao, có thể dùng một điện trở thay thế cho khuếch đại thuật toán để thực hiện việc chuyển đổi này. Tuy nhiên cũng vì thế mà phạm vi biến đổi số - tương tự bị giới hạn.

Độ chính xác

Ta thấy rằng đại lượng ra tương tự không liên tục mà nhận một giá trị rời rạc tương ứng với 1 trong 2^n tổ hợp của các mã nhị phân n bit tại lối vào số của một bộ DAC n bit.

Từ công thức (1) ta suy ra rằng, giá trị cực đại của lối ra không đạt tới giá trị lớn nhất của thang trị số FS (Full Scale) mà chỉ có thể đạt tới giá trị $FS \cdot (1 - 2^{-n})$.

Độ lớn của mỗi đơn vị điện áp ra ứng với bit có trọng số nhỏ nhất LSB (Least Significant Bit), nó có giá trị là $FS / 2^n$.

Một DAC có số bit càng lớn thì số mức điện áp ra càng lớn và giá trị của mỗi nấc càng nhỏ, và như vậy tập hợp các giá trị của đại lượng tương tự càng liên tục và bộ DAC có độ chính xác càng cao.

$$\text{Độ chính xác của DAC} = 1/2^n$$

với n là số bit.

Ứng dụng

DAC được sử dụng nhiều trong việc điều khiển nguồn nuôi cho các thiết bị kiểm tra tự động, các bộ tạo sóng và các bộ phận điều khiển quá trình.

8.2. CÁC LOẠI MÃ SỐ DÙNG CHO DAC

Các loại mã số dùng cho DAC gồm có mã nhị phân tự nhiên, mã bù nhị phân, mã BCD. Mã BCD mã hoá mỗi chữ số thập phân 4 bit. Như vậy, ở các bộ DAC có cùng số bit, bộ DAC sử dụng mã BCD sẽ có độ chính xác kém hơn khi dùng mã nhị phân tự nhiên. Độ chính xác của bộ DAC dùng mã BCD được tính bằng $1 / 10^D$ trong đó D là tổng các con số BCD.

Ví dụ: Với bộ DAC có 12 bit

Dùng mã BCD : Độ chính xác là: $1/10^3 = 0,001$

Dùng mã nhị phân tự nhiên : Độ chính xác là: $1/2^{12} = 0,00024$

Dùng mã nhị phân tự nhiên ta có được độ chính xác lớn gấp 4 lần độ chính xác dùng mã BCD.

Do đó những DAC dạng vi mạch thường sử dụng mã nhị phân hoặc mã bù nhị phân.

8.3. CÁC LOẠI DAC

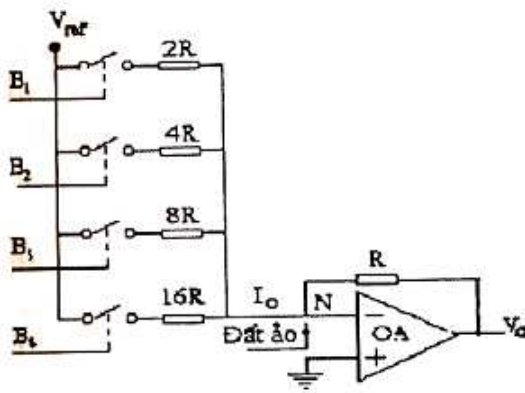
8.3.1. DAC dùng mạng điện trở trọng số (DAC with weighted resistor network)

Trên hình 8.2 trình bày sơ đồ nguyên lí của bộ biến đổi số tương tự 4 bit dùng mạng điện trở trọng số. Trong phương pháp này để thực hiện biến đổi số - tương tự, người ta tạo ra một dòng điện I_0 là tổng các dòng thành phần tương ứng I_k được chọn tương thích với mã số ở lối vào nhờ sự điều khiển bởi trạng thái các bit của mã số.

Dòng I_0 tỷ lệ với mã số lối vào được chuyển thành điện áp ra tỷ lệ với mã số nhị phân ở lối vào nhờ mạch khuếch đại thuật toán mắc theo kiểu cộng đảo pha. Điện áp ở lối ra của bộ DAC chính là điện áp ở lối ra của khuếch đại thuật toán.

Các dòng thành phần I_k được xác định theo các giá trị các điện trở trọng số $2R, 4R, R, 16R$ và các bit nhị phân B_k theo các hệ thức sau đây:

$$I_1 = B_1 \cdot V_{ref} / 2R; \quad I_2 = B_2 \cdot V_{ref} / 4R; \quad I_3 = B_3 \cdot V_{ref} / 8R; \quad I_4 = B_4 \cdot V_{ref} / 16R;$$



Hình 8.2

này là: số bit càng tăng thì số điện trở khác nhau về giá trị cũng tăng, việc chọn các điện trở chính xác càng khó khăn hơn. Để khắc phục nhược điểm này người ta đã đưa ra loại DAC dùng mạng điện trở R-2R loại này chỉ dùng có hai loại điện trở.

8.3.2. DAC dùng mạng điện trở R - 2R

Khác với DAC dùng điện trở trọng số, mạch DAC sử dụng mạng điện trở R-2R chỉ cần dùng 2 loại giá trị của trở. Nhưng so với các DAC dùng trở trọng số có cùng số bit thì số lượng điện trở đòi hỏi phải nhiều hơn. Sơ đồ nguyên lý bộ DAC dùng mạng điện trở R-2R được vẽ trên hình 8.3.

Đối với DAC loại R - 2R, các chuyển mạch điện tử dù ở vị trí nào 1 hay 0 đều được nối đất: hoặc được nối đất thực sự (vị trí 0), hoặc được nối qua điểm đất ảo ở lối vào N ($V_N = 0$) của mạch khuếch đại thuật toán (ở vị trí 1). Như vậy dòng qua các điện trở nối với các chuyển mạch có giá trị cố định cho mỗi trở, không phụ thuộc trạng thái của chuyển mạch.

Các điện trở mắc theo chuỗi Thevenin với phân bố dòng trên các trở qua mỗi nút của mạng điện trở dòng điện lại giảm đi một nửa đúng với quy luật của mã nhị phân như trong hình vẽ. Chỉ bằng mạng điện trở R - 2R, cộng thêm một bộ khuếch đại thuật toán, ta có thể xây dựng được một bộ chuyển đổi DAC. Điện trở phản hồi âm của mạch khuếch đại thuật toán nếu chọn đúng bằng R thì vùng biến thiên của điện áp ra sẽ phù hợp với kết quả tính toán cho DAC dùng điện trở trọng số. Điện áp ra trong trường hợp này cũng được xác định theo hệ thức (1). Tuy nhiên tùy theo yêu cầu cụ thể ta có thể chọn nó xung quanh giá trị R.

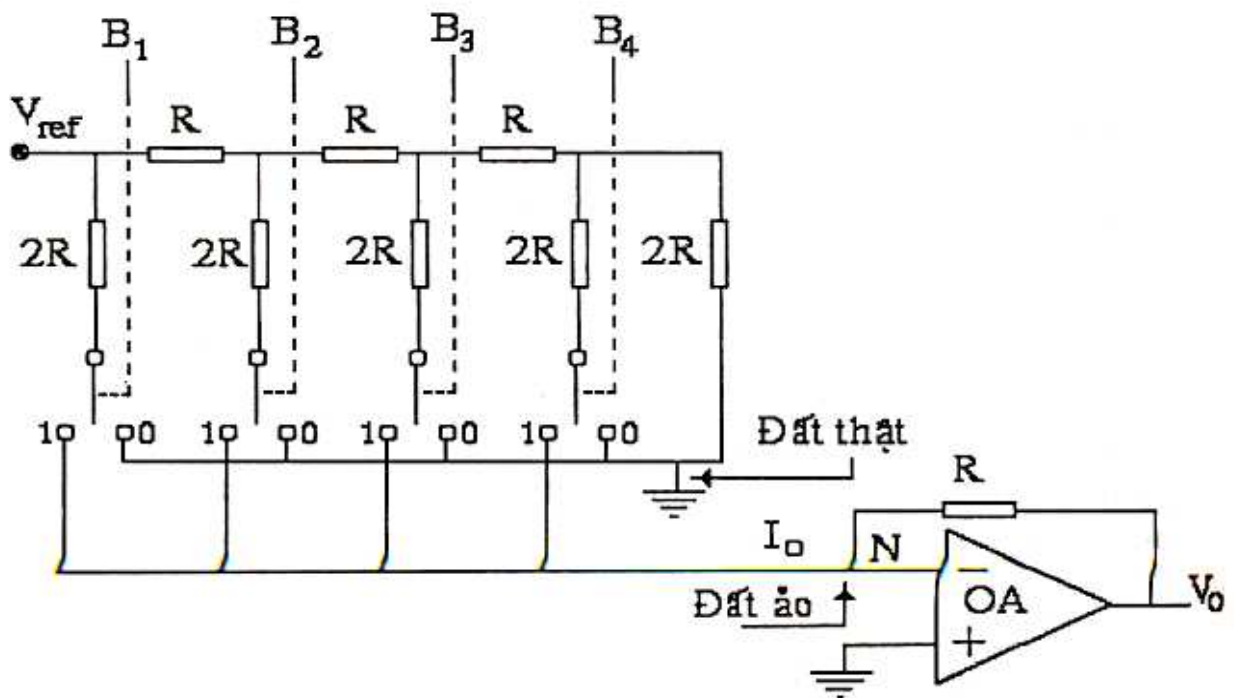
Từ sơ đồ ta có:

$$V_o = -I_o \cdot R \quad (2)$$

$$I_o = \frac{V_{ref}}{R} \left(\frac{B_1}{2} + \frac{B_2}{4} + \frac{B_3}{8} + \frac{B_4}{16} \right) \quad (3)$$

$$V_o = V_{ref} \cdot \left(\frac{B_1}{2} + \frac{B_2}{4} - \frac{B_3}{8} + \frac{B_4}{16} \right) \quad (4)$$

Như vậy điện áp ra tỷ lệ thuận với mã số lối vào theo hệ số tỷ lệ là điện áp chuẩn. Nhược điểm của mạch



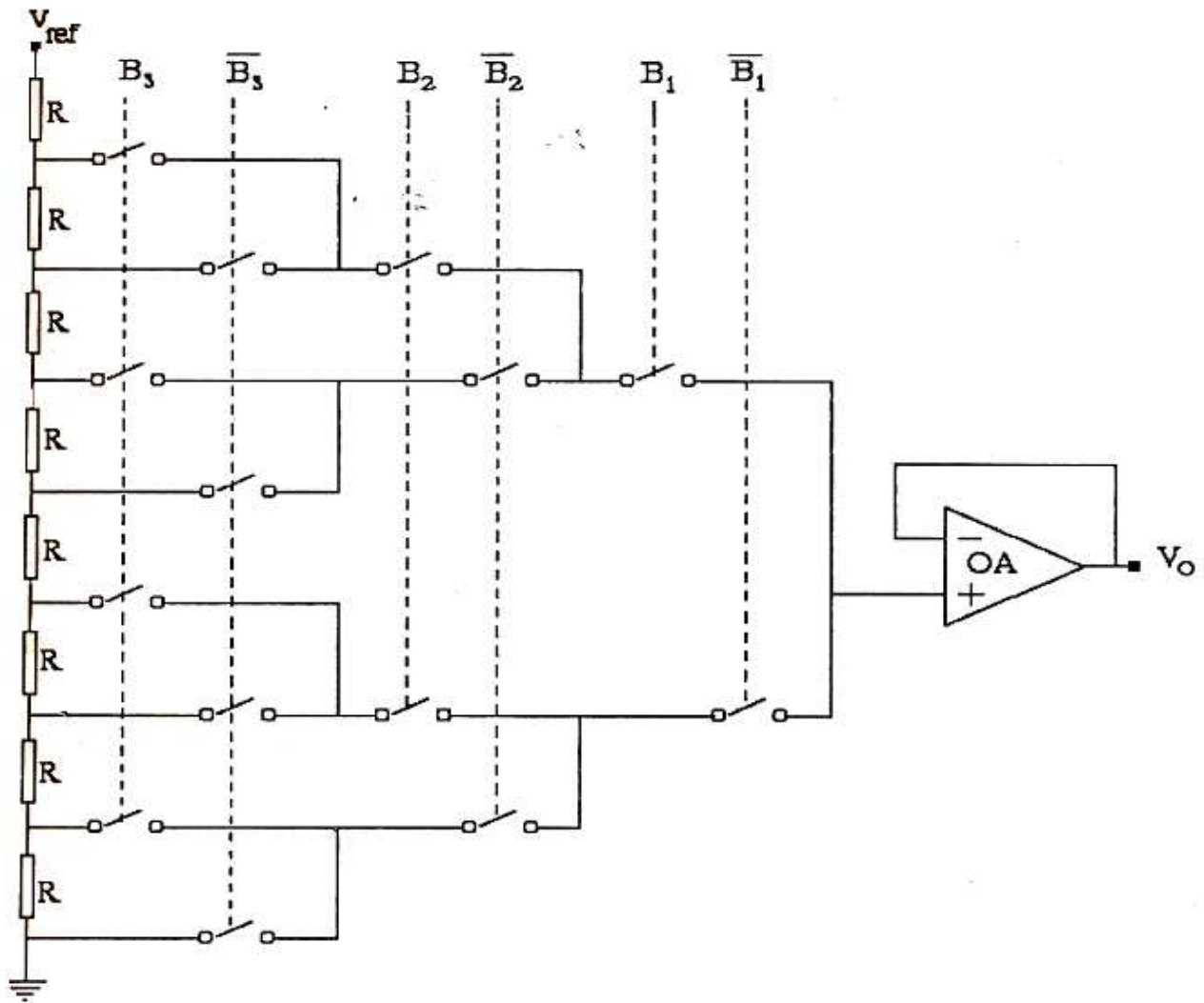
Hình 8.3

8.3.3. DAC dùng 2^n điện trở bằng nhau

DAC loại này có thể được xây dựng bằng cách sử dụng 2^n điện trở có cùng một trị số R . Trong đó n là số bit của bộ DAC.

Ngoài số điện trở trên, DAC loại này cần đòi hỏi có $(2^n - 1)$ chuyển mạch điện tử. Trên hình 8.4 trình bày sơ đồ nguyên lý bộ biến đổi DAC 4 bit thuộc loại này.

Do phương pháp này đòi hỏi phải sử dụng một số lượng điện trở và chuyển mạch khá lớn nên DAC loại này được sản xuất theo công nghệ MOS-LSI (vi mạch cỡ lớn MOS). Các điện trở được ghép nối tiếp với nhau để tạo thành mạch chia điện áp. Mạch này tách điện áp chuẩn V_{REF} thành 2^n mức. Các chuyển mạch được ghép nối theo hình cây và được điều khiển bằng mã số ở lối vào. Mạch khuếch đại thuật toán ở đây được mắc theo sơ đồ lặp lại điện áp.



Hình 8.4

8.3.4. DAC hai biến số

Cấu trúc DAC hai biến số được trình bày trên hình 10.5.

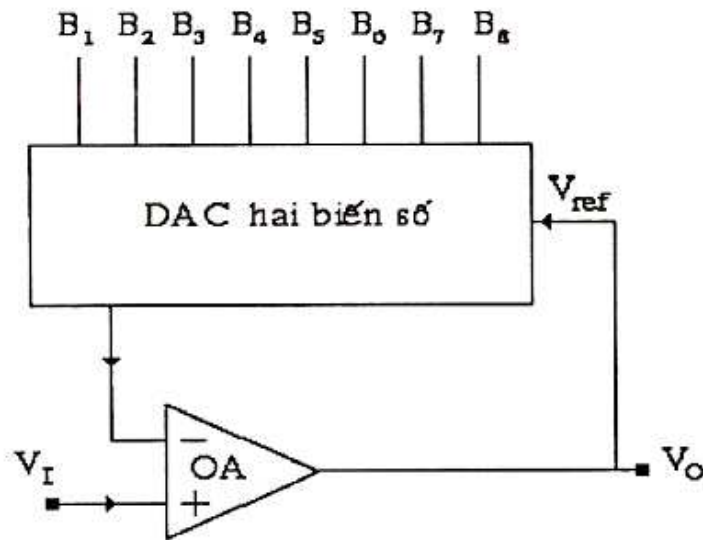
Từ công thức 1 và 2 ta có :

$$V_o = -I_o \cdot R$$

$$V_o = -V_{REF} (B_1 \cdot 2^{-1} + B_2 \cdot 2^{-2} + \dots + B_n \cdot 2^{-n}) \quad (1)$$

Khi giá trị của điện áp chuẩn V_{REF} thay đổi, giá trị của điện áp ra V_o cũng thay đổi. Nếu ở các lối vào của bộ DAC giữ ở mã cố định thì điện áp ra thay đổi tuyến tính với sự thay đổi của điện áp chuẩn V_{REF} . DAC làm việc với cả hai đại lượng vào là mã số và điện áp chuẩn được gọi là DAC hai biến số.

Trong mặt phẳng hai biến số : mã số và điện áp chuẩn. DAC hai biến số sẽ có bốn vùng làm việc, tùy theo mỗi biến số là đơn cực hay lưỡng cực.



Hình 8.5: Cấu trúc DAC hai biến số

DAC hai biến số dùng trong những hệ vi xử lý để điều khiển độ nhạy của bộ khuếch đại bằng phương pháp số.

Ở đây DAC hai biến số đóng vai trò điều chỉnh mạch phản hồi âm mắc cho mạch khuếch đại thuật toán.

Gọi F là hệ số phản hồi âm được xác định theo mã số vào. Ta có:

$$V_1 = F \cdot V_0$$

$$V_0 = V_1 / F \quad (5)$$

8.4. ĐẶC TÍNH CỦA DAC

8.4.1. Đặc tính chuyển đổi số tương tự của DAC

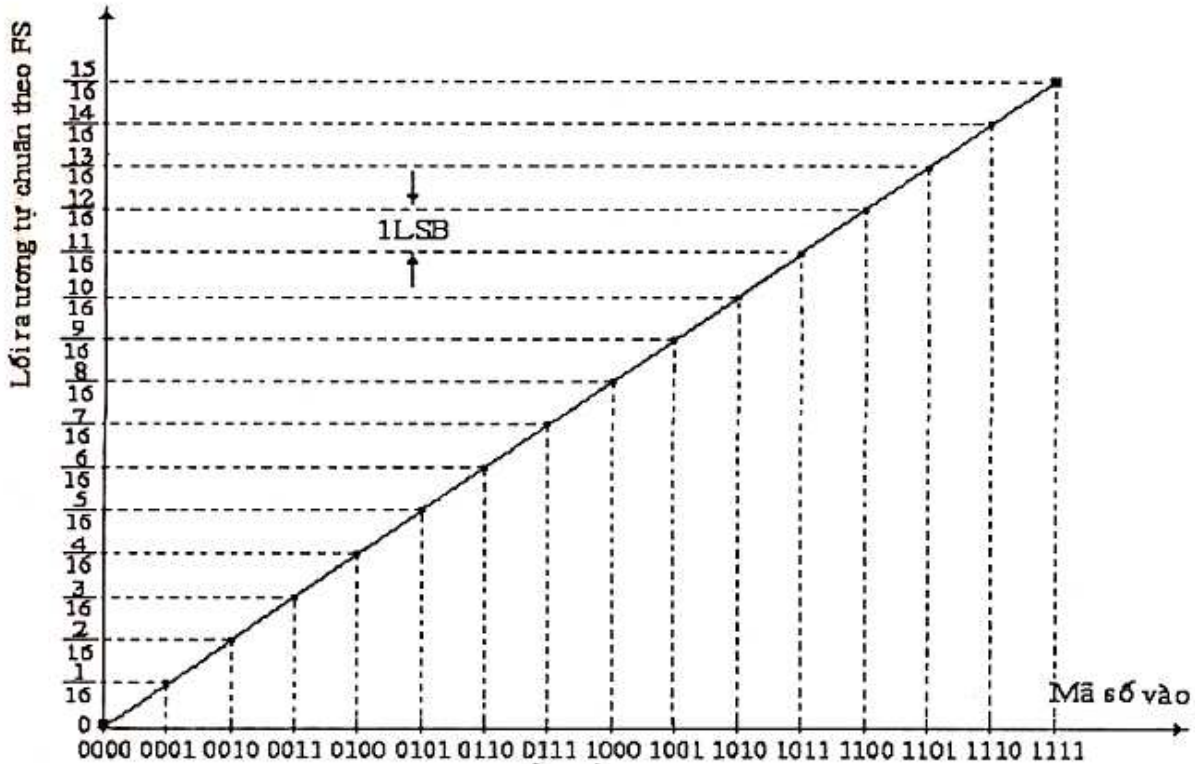
Đặc trưng chuyển đổi số tương tự của DAC là một đồ thị biểu diễn sự phụ thuộc điện áp lối ra của DAC vào các mã số lối vào. Trong trường hợp lý tưởng nó là đường thẳng qua điểm 0 gốc tọa độ (tương ứng với mã số lối vào DAC và điện áp ra DAC đều bằng 0) và điểm ứng với điện áp ra cực đại và các bit của lối vào DAC đều bằng 1. Hình 8.6 là đặc trưng chuyển đổi số tương tự của DAC 4 bit.

Xác lập được đặc trưng chuyển đổi số tương tự cho phép chúng ta đánh giá được phẩm chất của DAC. Chất lượng của một bộ DAC có thể được đánh giá thông qua sai số của nó. Đại lượng biểu diễn sai số của DAC là độ lệch tối đa giữa điện áp (dòng điện) ra và đường thẳng nối điểm 0 với điểm (FS - 1LSB) trên đường chuyển đổi số tương tự trong hình 8.6. Sai số của một DAC

cũng có thể định nghĩa là độ lệch giữa điện áp (hoặc dòng điện) ra trên thực tế và tính toán lý thuyết.

8.4.2. Những sai số của DAC

Sai số độ lệch (offset error), sai số khuếch đại (gain error), sai số tuyến tính (linearity), sai số phi tuyến tính vi sai (differential nonlinearity). Các sai số này được biểu diễn theo đơn vị %FS hoặc %LSB.



Hình 8.6: Đặc trưng chuyển đổi số tương tự lý tưởng

Sai số độ lệch là giá trị điện áp ra của bộ DAC khi lối vào là mã ứng với 0. Sai số này được tính bằng mV hoặc %LSB hoặc %FS. Để hiệu chỉnh giảm nhỏ sai số độ lệch, ta đưa mã ứng với 0 vào lối vào của bộ DAC và điều chỉnh chiết áp độ lệch sao cho điện áp lối ra là 0V. Đây là quá trình chỉnh điểm 0 của DAC.

Sai số khuếch đại là độ lệch giữa điện áp ra tính theo lý thuyết và giá trị thực tế khi lối vào ở một mã số nào đó, thường là mã số ứng với FS - 1LSB. Sai số khuếch đại được hiệu chỉnh sau khi đã hiệu chỉnh sai số độ lệch bằng cách đưa mã ứng với FS-1LSB vào lối vào rồi điều chỉnh chiết áp độ nhạy sao cho điện áp ra có giá trị là (FS - 1LSB).

Các sai số độ lệch và độ khuếch đại phụ thuộc vào sự thay đổi nhiệt độ của môi trường. Khi nhiệt độ môi trường thay đổi, các sai số này lại xuất hiện. Giá trị này thường vào cỡ vài phần nghìn cho 1°C . Đó là sự nhạy cảm về nhiệt độ của DAC.

Sai số tuyến tính là loại sai số không thể hiệu chỉnh được và là do thiết kế của bộ DAC. Mức tuyến tính của DAC cho biết độ lệch điện áp ra so với một đường thẳng đi qua những điểm nút của đặc tuyến chuyển đổi. Đường cong đặc tuyến là đơn điệu nếu sự thay đổi về độ lệch trên không đổi dấu, hay độ lệch này phải lớn hơn 0 cho mỗi nấc thang.

Do vậy mức độ tuyến tính của DAC phải nhỏ hơn hoặc bằng $1/2$ LSB.

Sai số phi tuyến vi sai là đại lượng cho biết độ lệch giữa giá trị thực tế và đặc trưng lý tưởng của một mạch DAC ứng với mọi thay đổi của mã số vào. Đại lượng này cho biết về độ nhẵn của đường cong đặc tuyến chuyển đổi của DAC.

8.4.3. Thời gian thiết lập

Thời gian thiết lập đối với DAC là thời gian cần thiết để điện áp ra đạt tới giá trị ổn định cuối cùng khi có sự thay đổi mã số lối vào. Thời gian thiết lập trước hết phụ thuộc vào kiểu chuyển mạch, kiểu điện trở và kiểu khuếch đại dùng để xây dựng DAC. Thời gian thiết lập không bao hàm thời gian trễ như đã được định nghĩa trong phần trước.

Ở DAC cũng như mọi thiết bị điện tử luôn có mặt nhiễu kim. Trong trường hợp của DAC là nhiễu số. Đó là nhiễu cấy trên điện áp ra của một bộ DAC gây nên bởi trạng thái trung gian của mã số vào mỗi khi chúng thay đổi. Khi có nhiễu này xuất hiện, điện áp bị kích lên một giá trị khác hẳn với giá trị thực của nó, do tốc độ thay đổi trạng thái của các chuyển mạch điện tử là không đồng đều.

8.5. ĐIỆN ÁP CHUẨN CỦA DAC

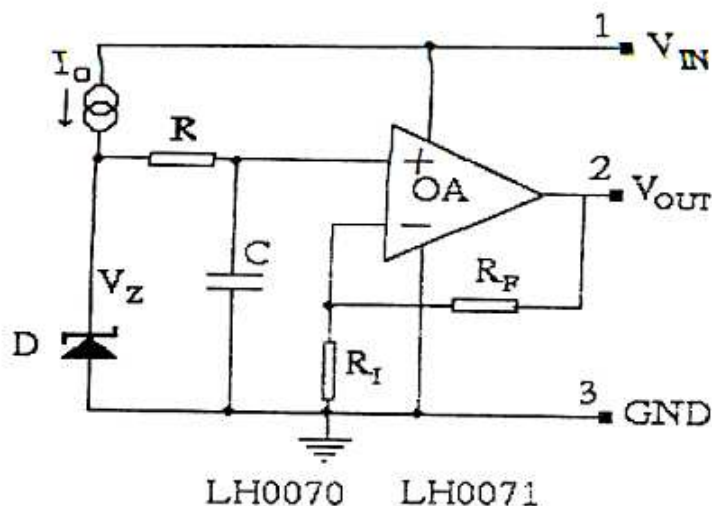
Nguồn điện áp chuẩn là thành phần hết sức quan trọng cho hoạt động của DAC. Có loại DAC được cấy điện áp chuẩn bên trong được trích từ điện áp nuôi và trong mạch đã cấy diode ổn áp có ổn nhiệt. Nhưng các điện áp chuẩn bên ngoài vẫn được sử dụng để đạt độ chính xác cao.

Những mạch tạo nguồn điện áp chuẩn ngày nay đã được chế tạo dưới dạng vi mạch. Trong mạch dùng diode zener ổn nhiệt được kích thích bằng

nguồn dòng không đổi và tăng cường công suất bằng khuếch đại điện áp dùng thuật toán. Độ chính xác của điện áp ra rất cao, đạt tới 0,01 %.

Ví dụ: mạch LH0070 và LH0071 của hãng National có điện áp chuẩn tải loại là 10V dùng cho DAC mã BCD và 10,24V dùng cho DAC mã nhị phân.

Sơ đồ của mạch được vẽ trên hình 8.7.



Hình 8.7

8.6. BỘ BIẾN ĐỔI TƯƠNG TỰ SỐ ADC (ANALOG TO DIGITAL CONVERTER)

Bộ biến đổi tương tự - số (ADC) là mạch biến đổi tín hiệu tương tự thành tín hiệu số có mã số tỷ lệ với giá trị biên độ tín hiệu tương tự ở lối vào. Giá trị của mã nhị phân ở lối ra của DAC biểu diễn độ lớn của tín hiệu tương tự tại thời điểm thực hiện phép biến đổi.

8.6.1. Mạch lấy mẫu và duy trì mẫu (sample and hold)

Một vài loại ADC đòi hỏi tín hiệu giữ nguyên không đổi trong thời gian thực hiện quá trình biến đổi, còn đối với một số khác điều kiện trên không cần thiết. Đối với loại DAC đòi hỏi tín hiệu không đổi trong quá trình biến đổi thì có thể dùng mạch lấy mẫu và duy trì mẫu (sample and hold circuit) đặt vào giữa lối vào của bộ DAC.

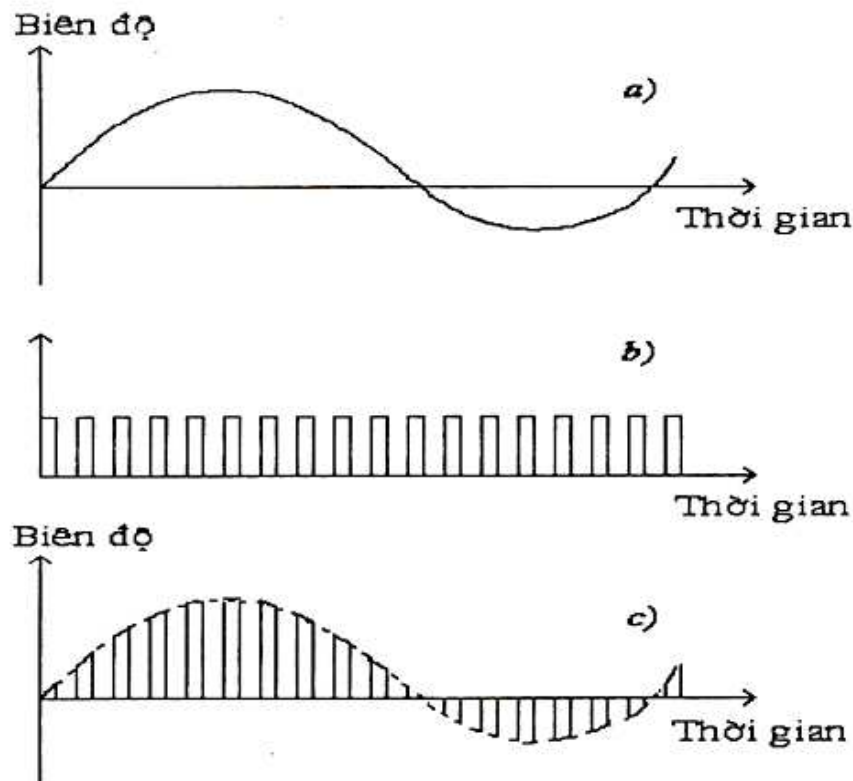
Mạch lấy mẫu và duy trì mẫu có chức năng sau: lấy mẫu ở những thời điểm xác định và duy trì giá trị đó cho đến thời điểm lấy mẫu tiếp theo.

Hình 8.8 trình bày dạng tín hiệu lối vào và ra của mạch "sample and hold".

Một mạch "sample and hold" được sử dụng phổ biến có sơ đồ như hình 8.9

Trong sơ đồ (a), quá trình tích mẫu và duy trì được thực hiện bởi hai transistor loại JFET hoặc MOSFET: T_1 và T_2 .

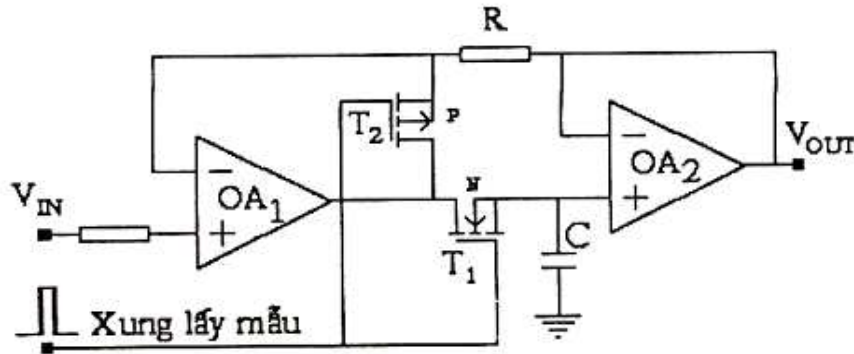
Khi lấy mẫu (sample) thì T_1 thông, T_2 ngắt. Sơ đồ tương đương khi lấy mẫu được trình bày trong hình (b), trong đó R_{DS} là điện trở thông mạch của T_1 (giá trị cỡ 100Ω). Chú ý rằng các bộ khuếch đại OA_1 và OA_2 mắc theo sơ đồ trên đều có hệ số khuếch đại bằng 1 nên mặc dù $R_{DS} \neq 0$, tụ điện C vẫn được nạp đến điện áp $E_{vào}$ và E_{ra} cũng bằng chính giá trị đó. Hằng số thời gian $R_{DS}C$ có thể dễ dàng chọn đủ nhỏ. Hệ số khuếch đại của tầng hai bằng 1 do đó E_{ra} sẽ đạt mức $E_{vào}$ trước khi tụ điện C nạp đầy.



- a) Tín hiệu tương tự
- b) Dãy xung điều khiển hoạt động bộ lấy mẫu
- c) Tín hiệu ra của bộ lấy mẫu

Hình 8.8

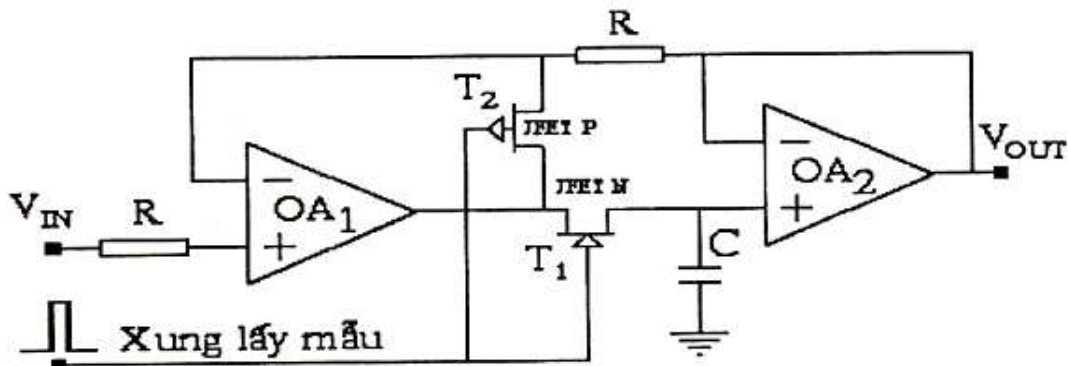
Trạng thái duy trì được thực hiện khi T_1 ngắt và T_2 thông. Mạch tải cho tụ điện C khi đó là điện trở transistor T_1 khi ngắt mạch và trở kháng lối vào của A_2 . Các giá trị này vào khoảng hàng chục, hàng trăm $M\Omega$ nên điện tích của C được duy trì giữa hai lần trích lấy mẫu.



Hình 8.9a: Mạch lấy mẫu giữ mẫu dùng MOSFET

Nói chung độ giảm điện áp duy trì có thể được tính theo công thức sau:

$$\text{độ giảm điện áp duy trì} = \frac{5 \cdot R_{ds}}{R_{ht}} \cdot \frac{I_h}{I_s}$$



Hình 8.9b: Mạch lấy mẫu và giữ mẫu dùng JFET

Trên hình 8.9b trình bày mạch lấy mẫu và giữ mẫu dùng JFET.

8.6.2. Các loại biến đổi ADC

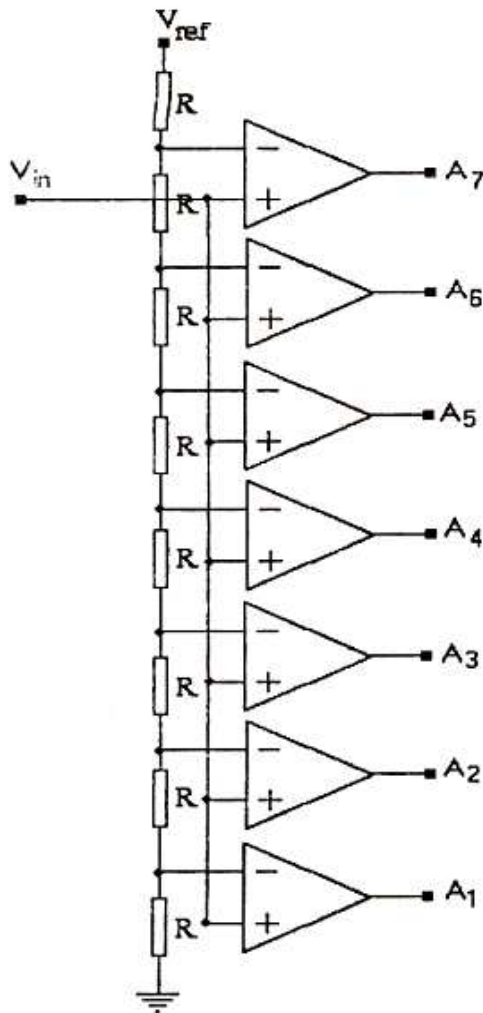
Có nhiều loại ADC tùy theo kết cấu của nó: ADC tức thời (ADC kiểu flash), ADC kiểu tích phân, ADC kiểu servo và ADC kiểu flash.

Mạch ADC kiểu tích phân gồm các loại: biến đổi thế - tần số, biến đổi tích phân hai sườn xung. Loại ADC này được dùng làm von kế số. ADC kiểu flash sử dụng 2^n bộ so sánh (comparator) với n bit lối vào và bộ lập mã ưu

tiên để số hoá tín hiệu tương tự. ADC kiểu servo là mạch số trong đó dùng DAC để phát ra một thế analog rồi đem so sánh với độ lớn của tín hiệu cần được số hoá. Mã lỗi vào được biến đổi theo hướng sao cho hai thế analog để bằng nhau. Lúc đó mã lỗi vào của DAC sẽ ứng với độ lớn tín hiệu cần số hoá. Ở đây ta chỉ xét ADC kiểu flash và servo.

8.6.2.1. ADC kiểu tức thời (kiểu flash)

Sơ đồ bộ biến đổi tương tự số theo kiểu Flash được trình bày trên hình 8.10.



Hình 8.10

phản ứng của các bộ so sánh tương tự mà thôi.

Nhược điểm của loại này là kích thước rất công kênh, nếu lối ra cần n bit thì lối vào cần $2^n - 1$ bộ so sánh tương tự. Nếu $n=8$ thì ta cần có 255 bộ so sánh tương tự. Và bộ giải mã ưu tiên với 255 lối vào cũng rất phức tạp.

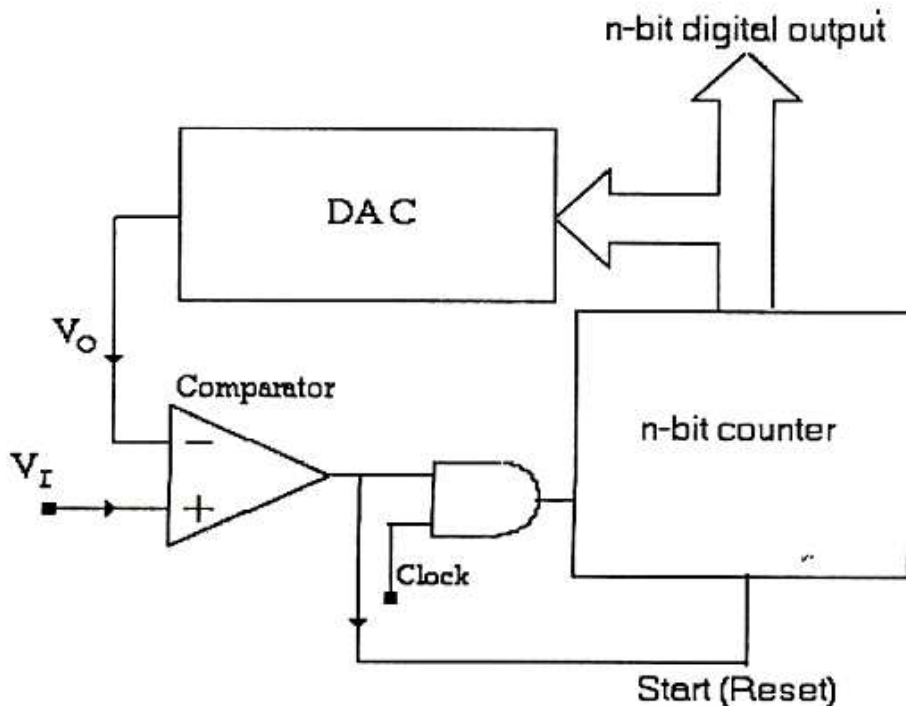
Đây là loại biến đổi A/D đơn giản nhất và có tốc độ nhanh nhất. Sơ đồ của loại này được trình bày trong hình bên. Nếu bộ biến đổi có n bit ở lối ra thì ta dùng n mạch so sánh tương tự ở lối vào để so sánh tín hiệu V_A với các mức điện áp chuẩn: $\Delta V_A, 2\Delta V_A, \dots, n \Delta V_A = V_{Amax}$. Với V_{Amax} là giới hạn thang đo của bộ biến đổi A/D. Như vậy, ở đây thang đo V_{Amax} được chia làm n mức và khoảng cách giữa các mức $\Delta V_A = V_{Amax} / n$. Biên độ của tín hiệu tương tự V_A được xác định bằng giá trị nằm giữa hai mức nào đó. Giả sử nếu V_A nằm giữa hai giá trị $10\Delta V_A$ và $11\Delta V_A$ thì 10 bộ so sánh tương tự sẽ cho giá trị ở mức thấp còn các bộ khác sẽ cho giá trị ở mức cao. Bộ giải mã ưu tiên sẽ cho ta kết quả là 10 và tạo ra mã nhị phân của số 10 ở lối ra.

Tốc độ của bộ biến đổi A/D tức thời rất nhanh, nó chỉ phụ thuộc vào tốc độ

8.6.2.2. ADC kiểu bậc thang (kiểu servo)

Trên hình 8.11 giới thiệu sơ đồ khối của bộ ADC kiểu bậc thang. Nguyên tắc hoạt động của nó như sau:

Chu trình biến đổi bắt đầu khi xung start xoá bộ đếm nhị phân n bit (n bit counter). Vì rằng V_O nhỏ hơn V_I nên lối ra bộ so sánh ở mức 1, cổng AND mở cho các xung Clock vào bộ đếm. Số đếm tăng dần cho tới khi V_O bắt đầu vượt quá V_I , lối ra của COMP. sẽ trở về 0 và khoá cổng AND lại. Mã số lối ra bộ đếm lúc này tương ứng với độ lớn thế analog cần biến đổi. Nếu đo dạng sóng V_O trong một chu kỳ biến đổi, ta sẽ thấy một sóng hình bậc thang.



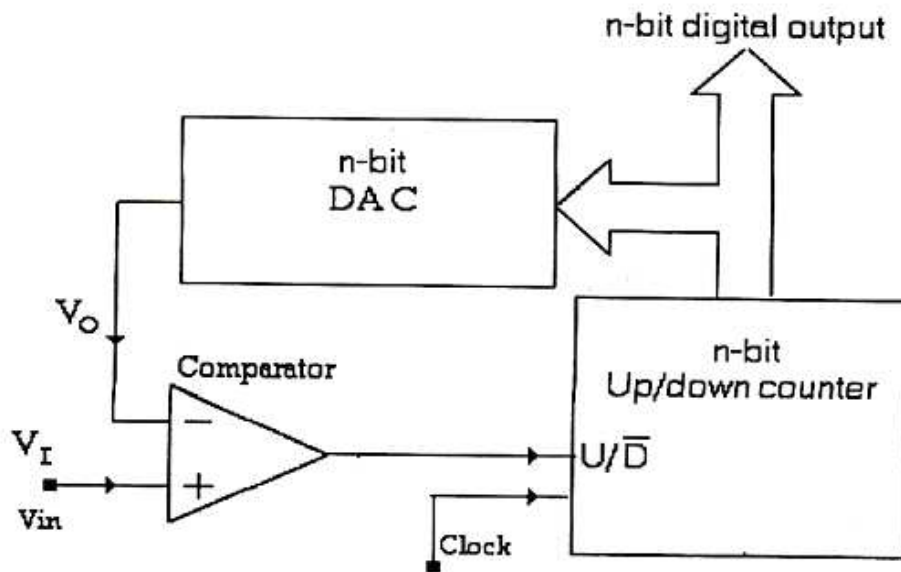
Hình 8.11: Sơ đồ khối bộ biến đổi ADC kiểu bậc thang

ADC loại này có kết cấu đơn giản nhưng có nhược điểm là thời gian biến đổi phụ thuộc vào độ lớn thế cần biến đổi.

8.6.2.3. ADC bám sát (tracking)

Trên hình 8.12 vẽ sơ đồ khối của ADC bám sát. Nếu giá trị V_I chỉ biến đổi quanh một giá trị nào đó thì loại ADC này tỏ ra tiện lợi hơn. Nguyên tắc của nó là dùng bộ đếm lên - xuống (up-down counter). Mạch được thiết kế sao cho nếu $V_O < V_I$ (thế lối ra của COMP. bằng 1), bộ đếm sẽ ở trạng thái

đếm lên, nếu $V_o > V_i$ (thế ra của COMP bằng 0) thì bộ đếm sẽ ở trạng thái đếm xuống. Như vậy, thế lối ra của DAC luôn có xu hướng "bám sát" thế vào cần biến đổi.



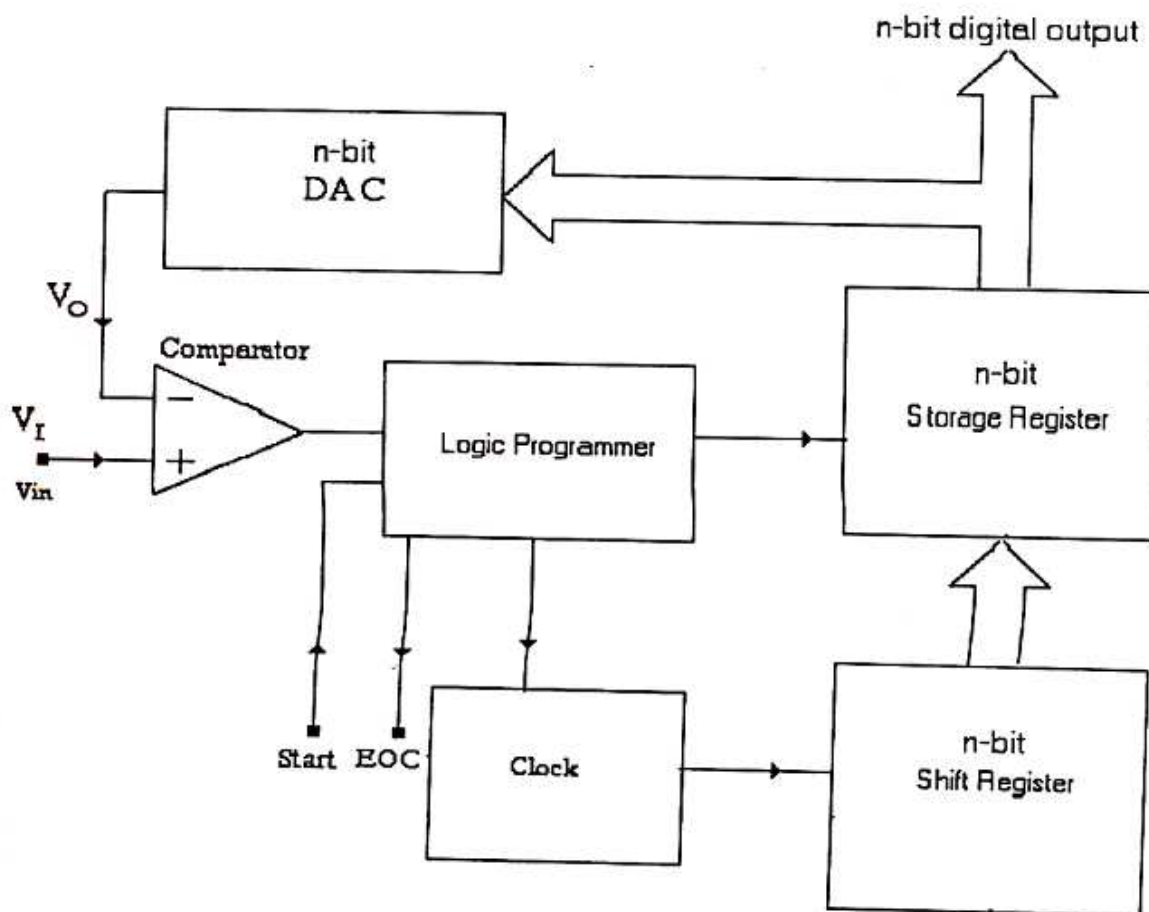
Hình 8.12: Sơ đồ khối của ADC bám sát

8.6.2.4. ADC xấp xỉ liên tiếp (successive approximation ADC)

Khác với loại bậc thang, loại này chỉ biến đổi và so sánh n nhị lối vào ADC. Do vậy, thời gian biến đổi là cố định, không phụ thuộc vào độ lớn của tín hiệu cần biến đổi. Sơ đồ khối mô tả cấu trúc ADC xấp xỉ liên tiếp được trình bày trên hình 8.13

Chu trình biến đổi bắt đầu với xung Start qua bộ điều khiển logic (logic programmer) đặt hai thanh ghi dịch và thanh ghi lưu trữ (shift register và store register) có giá trị tới nửa thang (100..00) ở nhị lối một. Tức là nó đặt cho bit nhiều ý nghĩa nhất MSB của các thanh ghi bằng 1, các bit còn lại bằng 0, làm cho lối ra của DAC $V_o = 1/2FS$. Tùy thuộc vào trạng thái ra của bộ so sánh COMP, tức là tùy thuộc vào tương quan giữa hai thế V_o và V_i , bộ điều khiển logic sẽ điều khiển như sau: nếu $V_o < V_i$, trong nhị lối thứ hai, giá trị 1 đó được giữ lại và nếu $V_o > V_i$ giá trị đó được xoá về 0, đồng thời thanh ghi dịch sẽ dịch số 1 tới bit tiếp theo bit MSB. Mã số được DAC biến đổi tương ứng thành thế V_o và lại được so sánh trong nhị lối thứ hai này. Quá trình cứ tiếp diễn với từng bit như vậy cho tới bit ít ý nghĩa nhất LSB được đưa lên 1.

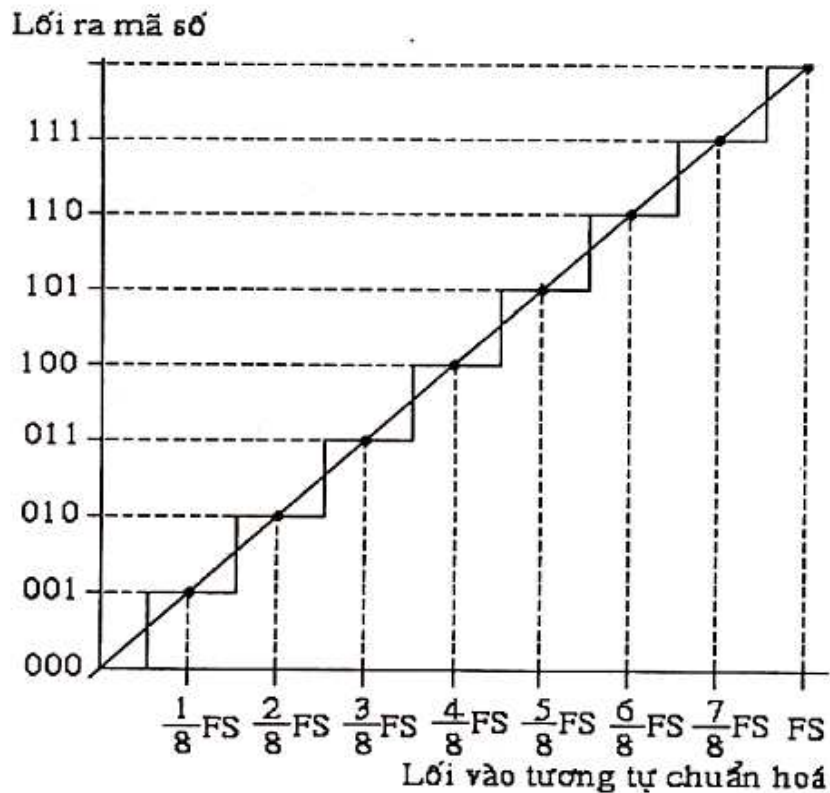
Sau n xung nhịp, một xung EOC (end of convert) được phát ra báo rằng một quá trình biến đổi đã xong. Lúc này mã số ở lối ra thanh ghi lưu trữ (hay ở lối vào của DAC) chính là tương ứng với thế cần biến đổi.



Hình 8.13: Sơ đồ khối của ADC xấp xỉ liên tiếp

8.6.3. Các đặc trưng kỹ thuật của ADC

Hình 8.14 cho ta thấy một thí dụ về đặc tính chuyển đổi của một ADC 3 bit. Ta thấy độ lớn của mỗi đơn vị lấy mẫu do phép lượng tử hoá quy định $Q = FS / 2^3$. Điểm giữa mỗi mẫu là giá trị điện áp tương tự được biểu diễn bằng một mã tương ứng với mẫu đó. Như vậy, phép lượng tử hoá tự nó đã bao hàm sai số là $\pm Q/2$. Ở trường hợp lý tưởng, một giá trị M ở lối ra biểu thị giá trị $M \pm Q/2$. Do đó, để giảm đi sai số này, cách duy nhất là tăng số bit ở lối vào hay nói cách khác: độ phân giải của ADC được nâng cao khi số bit chuyển đổi càng lớn.



Hình 8.14: Đặc trưng chuyển đổi của ADC 3bit.

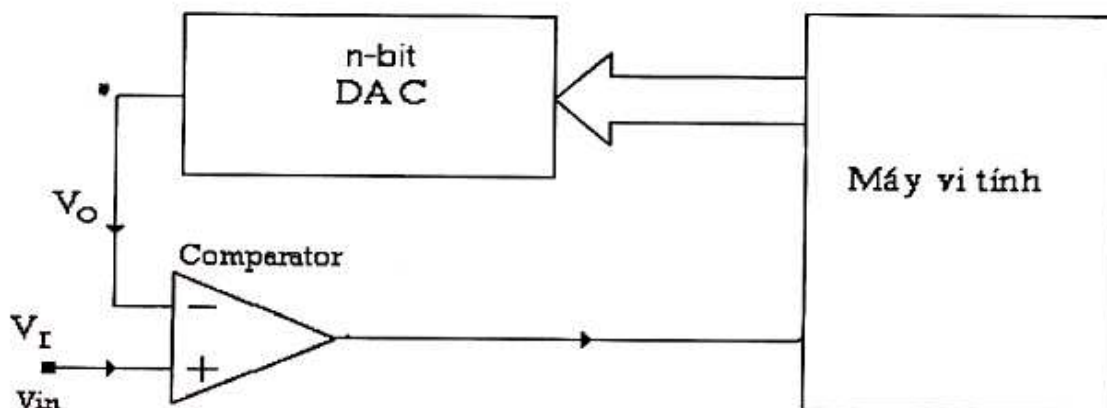
8.7. XÂY DỰNG HỆ GHÉP NỐI ADC, DAC VỚI MÁY TÍNH

Việc thiết kế các mạch DAC, ADC sẽ đơn giản đi nhiều về mạch điện tử nếu ta sử dụng kỹ thuật ghép nối máy vi tính, vì lúc đó phần mềm với các thanh ghi trong máy tính sẽ làm nhiệm vụ thay thế cho các thanh ghi, bộ đếm và bộ điều khiển phức tạp. Trong trường hợp đó, ta chỉ cần lắp một ADC, DAC với phần cứng tối thiểu. Hình 8.15 ở dưới là một sơ đồ khối điều hành của một DAC, ADC ghép nối với máy tính.

Ở đây, các thanh ghi, bộ điều khiển logic và bộ đếm ngoài không còn cần thiết nữa. Chính các thanh ghi đệm lỗi ra (IN/OUT buffer register) của các cổng vào/ra sẽ làm thay nhiệm vụ chúng. Cũng vậy, các thanh ghi vào của máy tính cũng sẽ làm nhiệm vụ kiểm tra trạng thái lỗi ra của bộ so sánh COMP. để trên cơ sở đó máy tính sẽ quyết định đặt các trị số thích hợp cho các thanh ghi trong các nhịp so sánh.

Ta biết rằng, bộ vi xử lý (CPU) của máy tính liên lạc với các thiết bị bên ngoài thông qua một số địa chỉ gọi là các cổng vào/ra (IN/OUT port). Để tiện cho việc ghép nối, bảo đảm an toàn cho máy vi tính khi phần cứng, hoặc

phần mềm điều khiển card có lỗi, trong các thí nghiệm cho sinh viên thực tập, chúng ta nên sử dụng cổng máy in để liên lạc giữa máy tính và phần cứng tối thiểu được ghép nối ở bên ngoài



Hình 8.15

Cổng in song song LPT1 có một địa chỉ thanh ghi đệm số liệu 8 bit là \$378, một thanh ghi đọc trạng thái là \$379, một thanh ghi điều khiển là \$37A. Việc đưa số liệu ra hoặc đọc số liệu vào được thực hiện dễ dàng bằng ngôn ngữ Pascal như sau:

Đọc giá trị của cổng vào biến x $x := \text{PORT}[\text{địa chỉ cổng}];$
 Viết ra cổng $\text{PORT}[\text{địa chỉ cổng}] := \text{giá trị};$

Việc đọc vào hoặc viết ra giá trị tùy ý của một vài bit trên các thanh ghi mà không ảnh hưởng tới giá trị của các bit còn lại (thuộc phạm vi khái niệm gọi là xử lý bit : manipulation of bit) có thể thực hiện nhờ các toán tử logic AND, OR, SHL, SHR với lưu ý rằng trong PASCAL, việc thực hiện AND hoặc OR giữa hai số nhị phân phải được tiến hành với từng bit một tương ứng.

Do đó, ta đặt bit thứ n lên một bằng các OR giá trị của thanh ghi với 2^n , đưa bit thứ n xuống 0 bằng cách AND giá trị của thanh ghi với $255 - 2^n$. Việc kiểm tra giá trị của thanh ghi thứ n cũng được thực hiện bằng phép AND hoặc OR.

Thí dụ: $x := \text{PORT}[\$378] \text{ OR } 32;$ {đưa bit 5 lên 1}
 $x := \text{PORT}[\$378] \text{ AND } (255 - 32)$ {đưa bit 5 xuống 0}
 if (PORT[\$379] AND 32 = 32) then writeln('bit 5 bang 1');
 if (PORT[\$379] AND 32 = 0) then writeln('bit 5 bang 0');

Về mặt vật lý, cổng LPT chuẩn có đầu ra chuẩn 25 chân ở sau máy tính với sự bố trí các chân như sau:

- Chân 1: STROBE
- Chân 2 đến 9: 8 bit số liệu từ D0-D7
- Chân 10: -ACK
- Chân 11: -BUSY
- Chân 12: -PE
- Chân 13: -SLCT
- Chân 14: -AUTOFXD
- Chân 15: -ERROR
- Chân 16: -INIT
- Chân 17: -SLCTIN
- Chân 18 đến 25: đất GND

Thanh ghi \$379 là thanh ghi đọc trạng thái vào, với các bit như sau:

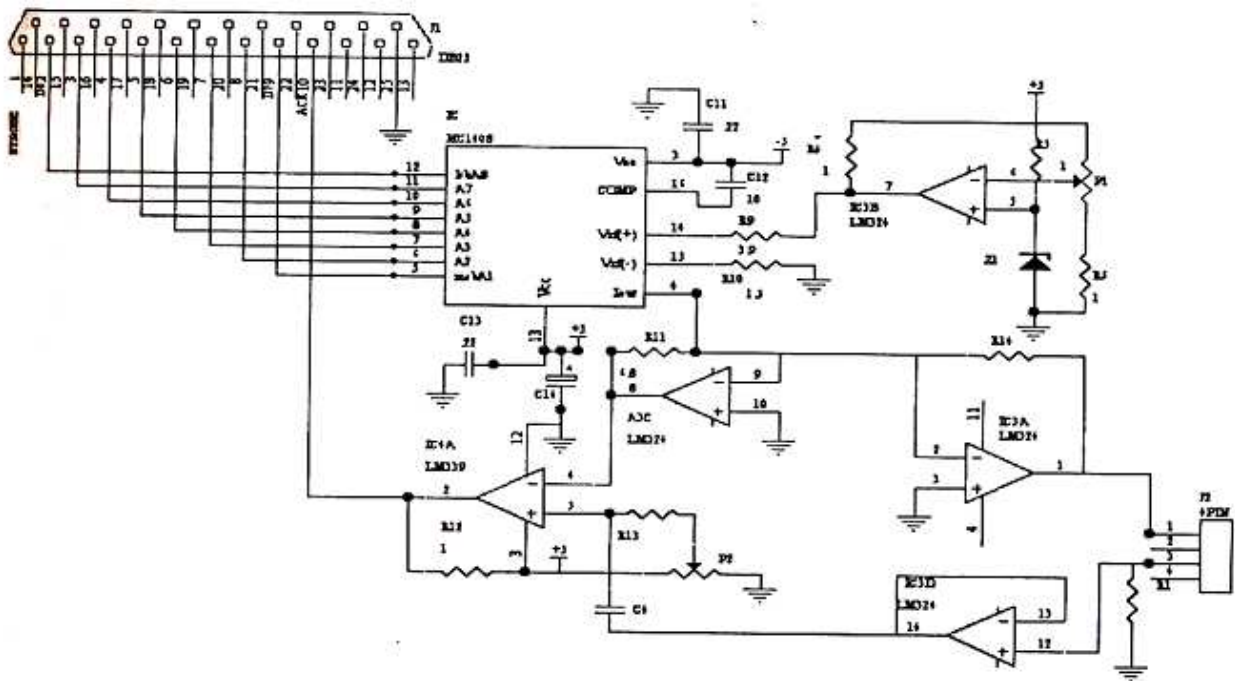
D7	D6	D5	D4	D3	D2	D1	D0
-BUSY	-ACK	-PE	-SLCT	ERROR	-	-	-

Trong thực nghiệm, lối ra của bộ so sánh Comparator được đưa vào chân 10 của bộ cắm 25 chân. Do đó việc kiểm tra trạng thái của bộ so sánh sẽ được thực hiện ở bit D6 (ACK) của thanh ghi này.

Thanh ghi \$37A là thanh ghi điều khiển lối ra, ở đây ta không dùng đến nhưng vẫn liệt kê ra để tham khảo khi ghép nối máy tính với máy phát tín hiệu điều khiển số:

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	IRQ	SLCTIN	INIT	AUTOFXD	STROBE

Trên hình 8.16 giới thiệu sơ đồ nguyên lý một Card DAC/ADC 8 bit ghép nối với máy vi tính qua cổng máy in LPT.



Hình 8.16

Điều khiển thiết bị ghép nối bằng máy tính và chương trình điều khiển

1. Nguyên tắc chung của điều khiển bằng máy tính

Mỗi thiết bị khi ghép nối với máy tính đều có một địa chỉ xác định, thuật ngữ máy tính gọi là "địa chỉ cổng vào ra" (Port address I/O). Thông qua địa chỉ này, máy tính (bộ vi xử lý của CPU) có thể truy nhập hay truy xuất để đọc hay ghi dữ liệu lên thiết bị ngoại vi, từ đó CPU có thể nhận hay điều khiển thiết bị ngoại vi.

Việc đọc ghi với cổng vào ra được thực hiện bằng lệnh của Pascal:

Đọc: `x:=PORT[Địa chỉ cổng];`

Ghi: `PORT[Địa chỉ cổng]:=Giá trị;`

2. Chương trình điều khiển DAC và ADC

a) Chương trình điều khiển DAC

Ta có thể trực tiếp đưa ra giá trị số qua BUS dữ liệu của máy tính vào các lối vào dữ liệu của DAC để biến đổi thành thế tương tự bằng lệnh: `PORT[DAC-Addr]:= giá trị số` (*DAC-Addr* là địa chỉ cổng ghép nối của DAC).

Giá trị số này có thể nhận được từ phần mềm điều khiển hệ ghép nối, qua hệ thống này ở lối ra của DAC ta sẽ được dạng tín hiệu Analog.

Nếu tạo được phần mềm tương thích ta có thể tạo một máy phát chức năng (FUNCTION GENERATOR) đơn giản, có thể phát được các dạng sóng sin, vuông, răng cưa, tam giác; biên độ và tần số của các dạng sóng này có thể điều khiển thay đổi được nhờ chương trình phần mềm, một thiết bị có chức năng như vậy không thể nào thiếu được trong các thí nghiệm điện tử điều chế thông tin.

Dưới đây giới thiệu một số thủ tục tạo các dạng sóng của máy phát chức năng nêu trên:

```
PROCEDURE RANGCUA (BD:BYTE; TS:INTEGER) ;
VAR I:INTEGER;
BEGIN
clrscr;
writeln('TAO XUNG RANG CUA');
REPEAT
FOR I:=0 TO BD DO
BEGIN
PORT[ $378] :=I;
DELAY (TS) ;
END;
UNTIL KEYPRESSED;
END;

PROCEDURE TAMGIAC (BD:BYTE; TS:INTEGER) ;
```



```

VAR I:INTEGER;
BEGIN
CLRSCR;
WRITELN('TAO XUNG TAM GIAC');
REPEAT
    FOR I:=0 TO BD DO
        BEGIN
            PORT[ $378] :=I;
            DELAY(TS);
        END;
    FOR I:=BD-1 DOWNT0 1 DO
        BEGIN
            PORT[ $378] :=I;
            DELAY(TS);
        END;
UNTIL KEYPRESSED;
END;

PROCEDURE VUONG(BD:BYTE;TS:INTEGER);
BEGIN
CLRSCR;

```

```
WRITELN('TAO XUNG VUONG');
```

```
REPEAT
```

```
    PORT[ $378] :=BD;
```

```
    DELAY(TS);
```

```
    PORT[ $378] :=0;
```

```
    DELAY(TS);
```

```
UNTIL KEYPRESSED;
```

```
END;
```

```
PROCEDURE SONGSIN(BD:BYTE;TS:INTEGER);
```

```
VAR A:ARRAY[0..359] OF BYTE;
```

```
I:INTEGER;
```

```
BEGIN
```

```
CLRSCR;
```

```
WRITELN('TAO SONG SIN');
```

```
    FOR I:=0 TO 359 DO
```

```
        A[ I] :=127+TRUNC(BD*SIN(TS*I*PI/180));
```

```
    REPEAT
```

```
        FOR I:=1 TO 359 DO
```

```
            PORT[ $378] :=A[ I];
```

```
    UNTIL KEYPRESSED;
```

```
END.
```

Dựa trên các thủ tục nêu trên ta có thể biến đổi biên độ hoặc tần số của tín hiệu phát thông qua các biến truyền giá trị BD, TS.

b/ Điều khiển ADC

Với cấu hình phần cứng nêu trên hình 8.15 ta cũng có thể điều khiển Carde ghép nối làm chức năng như một ADC có thể ghi nhận các tín hiệu tương tự từ lối vào của ADC biến đổi thành số đưa vào lưu trữ trong máy vi tính. Để làm được điều này ta cần phải tạo được phần mềm tương thích điều khiển thiết bị. Như trên đã giới thiệu ba phương pháp biến đổi ADC theo kiểu servo, ta cũng có ba thuật toán điều khiển biến đổi tương ứng ba phương pháp nêu trên:

```
Function ADC_BACTHANG:byte;
Var i:Byte;
Begin
  I:=0;
  Repeat
Port[ $378] :=I;

  I:=I+1;
  Until (Port[ $379] AND 64 = 0);
  ADC_BACTHANG:=I-1;
End;
Function ADC_BAM:byte;
Var i:Byte;
Begin
  Port[ $378] :=I;
  IF (Port[ $379] AND 64) = 1 THEN
REPEAT
```

```

    Port[ $378] :=I;
    I:=I+1;
UNTIL Port[ $379] AND 64 = 0
    ELSE
REPEAT
    Port[ $378] :=I;
    I:=I-1;
UNTIL Port[ $379] AND 64 = 1;
    ADC_BAM:=I;
End;

Function ADC_XAPXI:byte;
VAR I,J,K:INTEGER;
BEGIN
REPEAT
    I:=128;K:=64;
    FOR J:=1 TO 8 DO
    BEGIN
        PORT[ $378] :=I;
        IF (PORT[ $379] AND 64<>0) THEN
            I:=I+K

```

```
ELSE I:=I-K;
K:= K DIV 2;
END;
ADC_XAPXI:=I;
UNTIL KEYPRESSED;
END;
```

Bài tập thực hành:

1/ Khảo sát hoạt động của DAC 8bit MC1408. Lập trình điều khiển DAC tạo các tín hiệu hình sin, xung răng cưa, xung tam giác và xung vuông góc.

2/ Khảo sát hoạt động của ADC xây dựng trên cơ sở DAC 8 bit MC1408 kết hợp mạch so sánh vi sai LM 311 ghép nối với vi tính qua cổng LPT1.

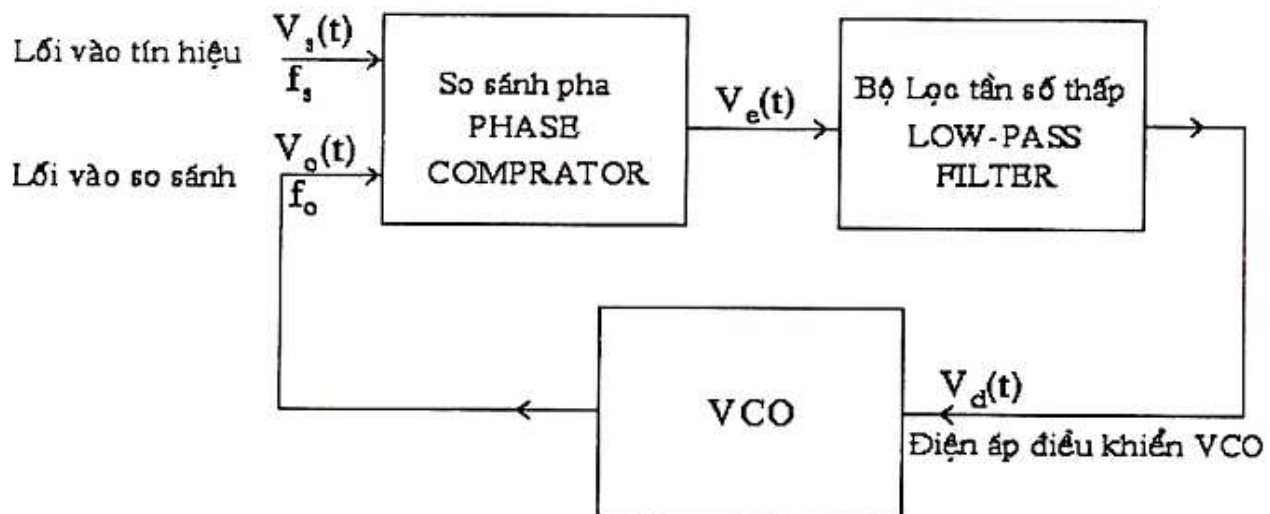
Viết chương trình phần mềm cho ADC theo 3 phương pháp: ADC bậc thang, ADC bám sát, ADC xấp xỉ liên tiếp.

Chương IX

VÒNG BẮM PHA

9.1. KHÁI QUÁT VỀ VÒNG BẮM PHA

Hệ thống tự động điều chỉnh tần số theo pha hay còn gọi là vòng bám pha PLL (Phase Locked Loop) được sử dụng rộng rãi trong việc xử lý tín hiệu tương tự và các hệ thống số. Một số ứng dụng quan trọng của vòng bám pha là: điều chế và giải điều chế FM, giải điều chế FSK (FSK Demodulation), giải mã âm, nhân tần, đồng bộ hoá xung đồng hồ, tổ hợp tần số, máy phát điều tần. Trong chương này giới thiệu những điểm rất cơ bản về vòng bám pha vi mạch họ CMOS CD4046 và một số ứng dụng của nó. Công suất tiêu thụ của vi mạch này rất nhỏ ($600 \mu W$ ở tần số 10 kHz). Do tiêu thụ rất ít năng lượng điện nên vi mạch được dùng trong các thiết bị viễn thông xách tay dùng pin. Hệ thống vòng bám pha cơ bản được trình bày trong sơ đồ khối (hình 9.1). Nó gồm ba phần: bộ so sánh pha, bộ lọc tần số thấp, máy phát xung tần số được điều khiển bằng điện áp VCO.



Hình 9.1: Sơ đồ khối của vòng bám pha

Ba khối này hợp thành một hệ thống phản hồi về tần số khép kín (close loop frequency feedback).

Khi không có tín hiệu vào PLL, sự chênh lệch điện áp $V_e(t)$ ở lối ra của bộ so sánh pha bằng không. Điện áp $V_d(t)$ ở lối ra của bộ lọc tần thấp cũng bằng không. Máy phát xung tần số điều khiển bằng điện áp VCO hoạt động ở tần số định sẵn f_0 gọi là tần số dao động trung tâm (center Frequency). Khi có tín hiệu đưa vào hệ thống PLL, bộ so pha sẽ so pha và tần số của tín hiệu lối vào với pha và tần số của VCO, tạo ra một điện áp sai $V_e(t)$ tỷ lệ với sự lệch pha và chênh lệch tần số của tín hiệu lối vào và VCO, tức là phản ánh sự khác nhau về pha và tần số của hai tín hiệu. Điện áp sai này được lọc rồi đưa vào lối vào điều khiển của VCO. Điện thế điều khiển $V_d(t)$ thúc đẩy tần số của VCO thay đổi theo hướng giảm bớt sự khác nhau về tần số giữa f_0 và tín hiệu lối vào. Khi tần số của tín hiệu vào f_s tiến dần đến f_0 , do tính chất hồi tiếp của PLL sẽ thúc đẩy VCO đồng bộ hoặc bắt chập với tín hiệu lối vào. Sau khi chập, tần số của VCO sẽ bằng tần số của tín hiệu lối vào, tuy nhiên sẽ vẫn có một độ lệch pha nào đó. Sự khác biệt về pha này là cần thiết vì nó sẽ tạo ra điện áp sai $V_e(t)$ để chuyển tần số dao động tự do của VCO thành tần số của tín hiệu vào f_s , như vậy sẽ giữ cho PLL ở trạng thái giữ chập tần số. Đương nhiên, không phải tín hiệu nào VCO cũng bắt chập được tần số mà chỉ có tần số ở trong một dải hữu hạn nào đó gần với f_0 thì VCO mới có thể bắt chập được. Dải tần số trên đó hệ duy trì tình trạng chập với tín hiệu lối vào được gọi là **dải giữ chập** hay **dải bám (lock range)** của hệ thống PLL.

Dải tần số trên đó hệ thống PLL có thể bắt chập một tín hiệu vào gọi là **dải bắt chập (capture range)**. Dải bắt chập bao giờ cũng nhỏ hơn dải giữ chập.

Chúng ta có thể dùng một cách khác để miêu tả hoạt động của PLL: bộ so sánh pha thực chất là mạch nhân, nó trộn tín hiệu vào với tín hiệu VCO. Sự trộn này tạo tần số tổng và tần số hiệu $f_s \pm f_0$. Khi mạch ở trạng thái chập thì hiệu tần số $f_s - f_0 = 0$, do đó nó tạo ra thành phần một chiều. Bộ lọc tần số thấp loại bỏ thành phần tần số tổng $f_s + f_0$ nhưng tiếp nhận thành phần điện áp một chiều, tức là chỉ cho thành phần một chiều đi qua. Thành phần một chiều này điều khiển VCO hoạt động ở trạng thái giữ chập với tín hiệu vào. Cần chú ý rằng, dải giữ chập độc lập với dải tần số của bộ lọc tần số thấp, vì rằng khi mạch ở trạng thái giữ chập thành phần hiệu tần số bao giờ cũng là dòng một chiều.

Sau đây chúng ta sẽ nghiên cứu các hiện tượng quan trọng trong hệ thống PLL là bắt chập và giữ chập. Khi mạch chưa ở trạng thái chập, bộ so pha trộn tín hiệu vào với tín hiệu VCO để tạo ra thành phần tổng và hiệu hai

tần số. Nếu thành phần hiệu nằm ngoài biên dải tần số của bộ lọc tần thấp thì nó sẽ bị loại bỏ cùng thành phần tổng tần số, do đó trong mạch sẽ không có thông tin nào được truyền qua mạch lọc và VCO sẽ tiếp tục hoạt động ở tần số trung tâm ban đầu. Khi tần số tín hiệu vào tiến dần đến gần tần số phát của VCO thì thành phần hiệu giảm xuống và tiến gần đến biên dải tần số của bộ lọc tần thấp. Lúc đó, một phần của thành phần hiệu đi qua được bộ lọc tần thấp và thúc đẩy VCO chuyển đến tần số của tín hiệu vào theo hướng sao cho thành phần hiệu tần số giảm và cho phép nhiều thông tin nữa đi qua bộ lọc tần thấp đến VCO. Đây là cơ chế hồi tiếp dương thúc đẩy VCO chập với tín hiệu vào.

Dải bắt chập là dải tần số lân cận tần số dao động tự do ban đầu của VCO mà trên đó hệ có thể bắt chập với tín hiệu vào. Dải bắt chập thể hiện cho thấy tần số của tín hiệu vào phải tiến lại gần tần số của VCO như thế nào để tần số phát của VCO chuyển thành có cùng tần số với tín hiệu vào. Dải bắt chập phụ thuộc vào dải tần của bộ lọc tần thấp và hệ số khuếch đại chung của hệ thống.

Dải giữ chập là dải tần số ở lân cận tần số dao động tự do ban đầu của VCO mà trong đó mạch phản hồi có thể theo dõi tín hiệu vào sau khi đã chập tần số. Khi mạch đã ở trạng thái chập thành phần, hiệu tần số của tín hiệu ra của bộ so pha (điện áp sai $V_e(t)$) là dòng một chiều và đi qua bộ lọc tần số thấp. Như vậy, dải giữ chập được giới hạn bằng khoảng biến thiên điện áp V_d có thể được tạo ra độ lệch tần tương ứng của VCO. Dải giữ chập chủ yếu là thông số dòng một chiều và không chịu ảnh hưởng dải tần của bộ lọc tần thấp.

Chúng ta cần phân biệt giữa dải bắt chập và dải giữ chập. Dải bắt chập có thể có bất cứ giá trị nào trong phạm vi khoảng giữ chập. Dải bắt chập giảm khi hằng số thời gian của bộ lọc tần thấp giảm, trong khi dải giữ chập không bị chi phối bởi bộ lọc mà chỉ do hệ số khuếch đại của hệ quyết định.

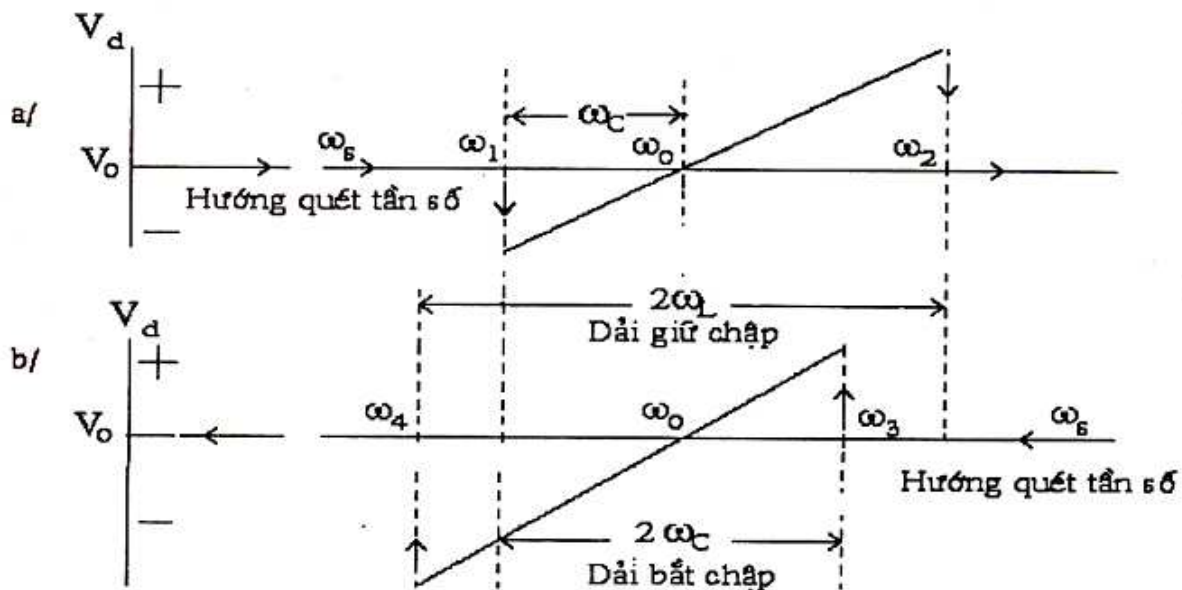
Hình 9.2 cho chúng ta thấy đặc trưng chuyển tần số sang điện áp điển hình của PLL. Cho tín hiệu vào PLL, tần số của nó được quét từ từ trên một dải rộng. Trục thẳng đứng là điện thế tương ứng V_d của mạch. Trên hình 9.2a là trường hợp tần số tín hiệu tăng dần, mạch không có phản ứng gì với tín hiệu cho đến khi tần số tín hiệu đạt đến tần số ω_1 tương ứng với biên dưới của vùng bắt chập. Lúc đó hệ bắt chập với tín hiệu vào và tạo ra bước nhảy của điện thế V_d với dấu âm. Sau đó, VCO thay đổi tần số với hệ số góc bằng nghịch đảo của hệ số khuếch đại lối vào VCO ($1/K_0$) và đi qua giá trị V_0 khi $\omega_1 = \omega_0$, mạch bám sát tín hiệu vào cho đến khi tần số tín hiệu vào đạt đến ω_2

tương ứng với biên trên của dải giữ chặt. Khi đó hệ mất bám, điện thế V_d tụt xuống V_0 và tạo ra tần số dao động tự do của VCO (hình 9.2a).

Nếu ta lại cho tần số tín hiệu vào quét theo chiều hướng giảm dần thì quá trình lặp lại nhưng đảo ngược so với trước (hình 9.2b) mạch bắt chặt lại với tín hiệu ở ω_3 tương ứng với biên trên của dải bắt chặt và bám sát theo tín hiệu vào cho đến khi tần số của tín hiệu vào bằng ω_4 tương ứng với biên dưới của dải giữ chặt.

Như vậy, dải giữ chặt của hệ là (ω_4, ω_2) và dải bắt chặt là (ω_1, ω_3) .

Do đặc trưng chuyển tần số - điện áp như trên nên PLL có tính chọn lọc với tần số trung tâm của VCO, nó chỉ có phản ứng đối với những tần số tín hiệu vào sai lệch so với ω_0 là ω_C hoặc ω_L ($\omega_C = (\omega_3 - \omega_1)/2$ và $\omega_L = (\omega_2 - \omega_4)/2$), tùy theo mạch bắt đầu có hay không có điều kiện giữ pha ban đầu.



Hình 9.2: Đặc trưng chuyển tần số-điện áp của PLL

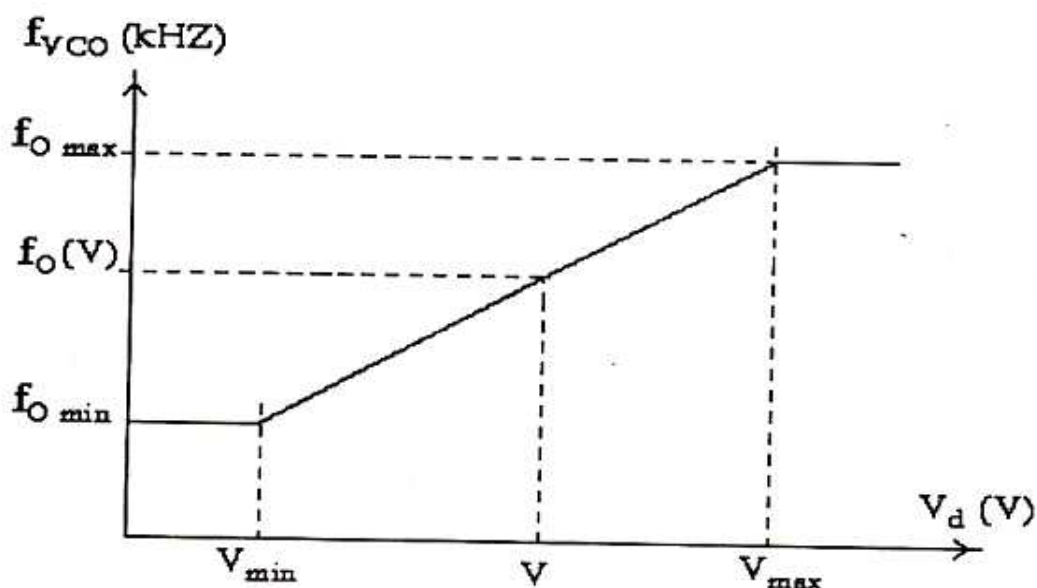
Sự tuyến tính của đặc trưng chuyển tần số sang điện áp của PLL chỉ do hệ số chuyển đổi của VCO quyết định, do đó ta thường đòi hỏi VCO có đặc trưng chuyển điện áp sang tần số ở mức độ tuyến tính cao.

Trên hình 9.3 trình bày đường đặc trưng của sự phụ thuộc tần số phát của VCO vào điện áp điều khiển V_d ở đây f_{max} và f_{min} là tần số của máy phát VCO tương ứng với tần số góc ω_2 và ω_4 .

Khi đó dải giữ chặt của hệ là:

$$\Delta f_L = f_{\max} - f_{\min}$$

Nếu gọi f_s là tần số của tín hiệu lỗi vào thì dải bắt chập của hệ PLL là :



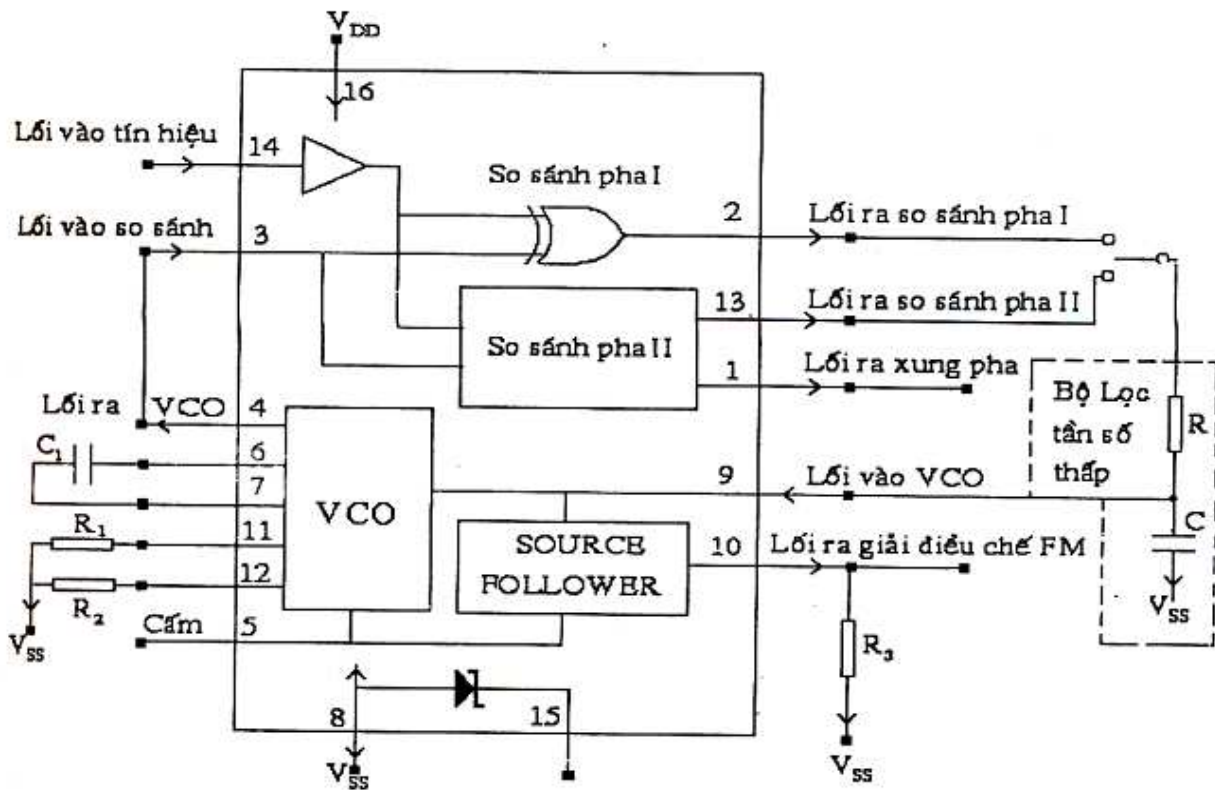
Hình 9.3: Sự phụ thuộc của tần số VCO vào điện áp $\Delta f_c = f_{s\max} - f_{s\min}$

Trong đó $f_{s\max}$ và $f_{s\min}$ là tần số của tín hiệu tương ứng với tần số góc ω_3 và ω_1 .

9.2. CẤU TẠO CỦA VÒNG BẮM PHA CMOS PLL CD4046

Sơ đồ khối của CMOS PLL CD4046 được trình bày trong hình 9.4 dưới đây:

CMOS PLL CD4046 là một vi mạch đơn khối được đóng trong một vỏ có 16 chân. Vòng bám pha PLL bao gồm máy phát điều khiển bằng điện áp VCO công suất thấp, tuyến tính, và hai bộ so sánh pha có cùng bộ khuếch đại tín hiệu vào và cùng một lối vào so sánh chung. Diode Zener có điện áp $V_Z = 5,2$ V để tạo ra điện áp một chiều ổn định dùng để điều chỉnh nếu cần thiết. VCO được nối trực tiếp hoặc qua bộ chia tần tới bộ so sánh pha. Bộ lọc tần số thấp được nối ở mạch ngoài để có thể thay đổi cấu trúc của hệ trong từng ứng dụng cụ thể. Sau đây chúng ta sẽ lần lượt xét từng khối chức năng có trong vi mạch này.



Hình 9.4: Sơ đồ khối của vòng bám pha CMOS PLL CD4046

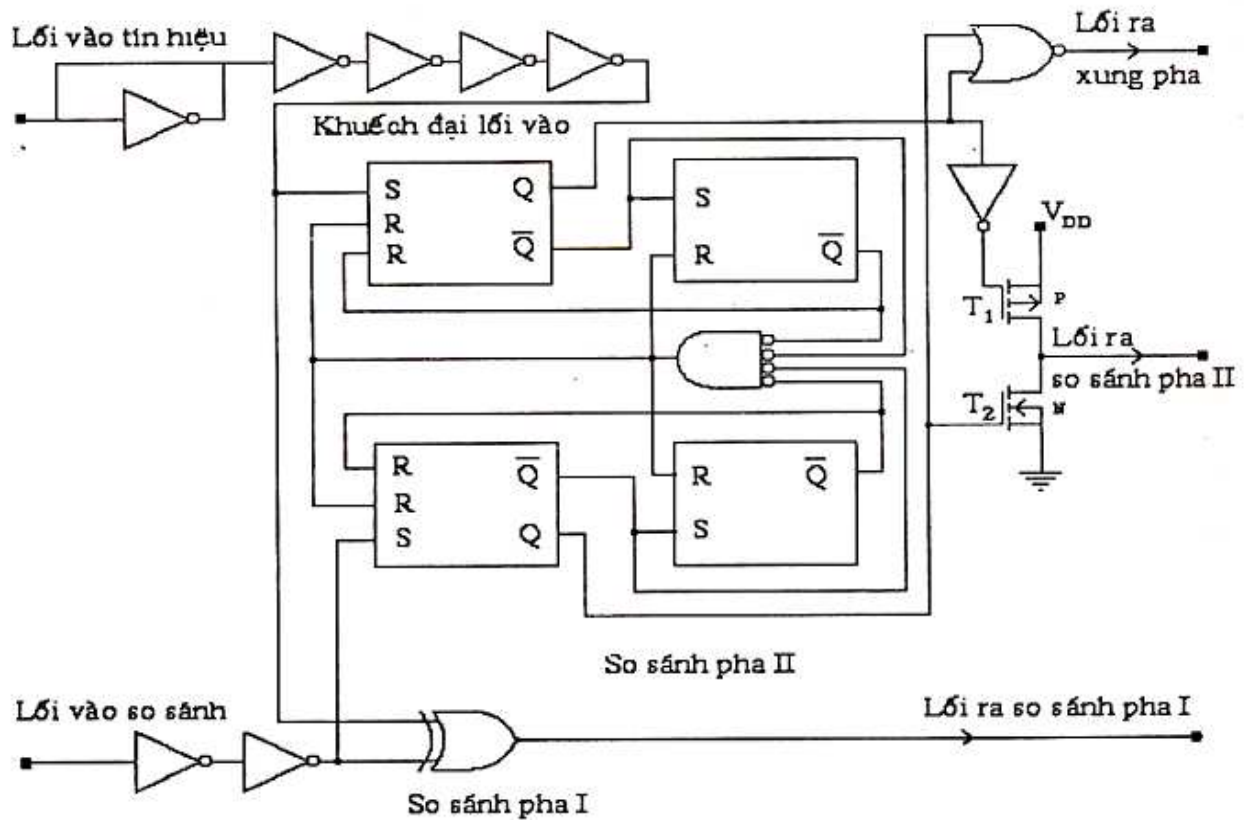
9.2.1. Bộ so sánh pha

Hầu hết các vòng bám pha tuyến tính dùng bộ trộn cân bằng (balanced mixer) được tạo từ các mạch khuếch đại tín hiệu tương tự có điều khiển để làm bộ so sánh pha. Công nghệ chế tạo CMOS khó thực hiện việc khuếch đại tín hiệu tương tự, do đó thiết kế PLL trình bày trong hình 9.5 dùng bộ so sánh pha kiểu số. Cả hai bộ so sánh pha có chung bộ khuếch đại lối vào, nó được thiết lập bởi tầng tạo điện thế phân cực và bốn tầng khuếch đại đảo. Biên độ dao động ứng với các mức logic của CMOS ($\text{logic } 0 < 30\% (V_{DD} - V_{SS})$, $\text{logic } 1 > 70\% (V_{DD} - V_{SS})$). Đối với các tín hiệu nhỏ hơn nữa thì nó phải được nối qua điện dung để tạo ra một tín hiệu số vượt ngưỡng kích thích vào bộ so sánh pha.

9.2.1.1. Bộ so sánh pha I

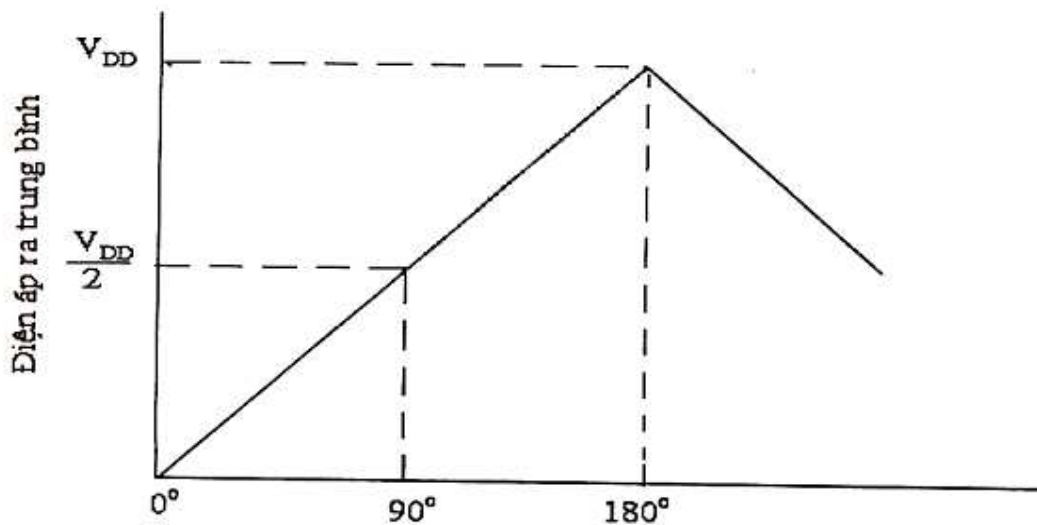
Bộ so pha I là một mạch hoặc tuyệt đối (XOR). Mạch này hoạt động tương ứng với tín hiệu ngưỡng của bộ trộn cân bằng. Để đạt được dải giữ

chập lớn nhất, các xung ở lối vào tín hiệu và lối vào so sánh phải là các xung vuông cân có độ rộng xung bằng 50% chu kỳ. Khi không có tín hiệu hoặc nhiễu ở lối vào, ở lối ra của bộ so sánh pha I có điện áp trung bình bằng $V_{DD}/2$. Bộ lọc tần thấp nối với lối ra của bộ so sánh pha I cung cấp điện áp trung bình cho lối vào VCO, điện áp này điều khiển VCO phát ra các xung vuông có tần số dao động là tần số dao động trung tâm f_0 . Với bộ so sánh pha I, dải tần số trong đó PLL có thể thiết lập trạng thái bám (dải bắt chập) phụ thuộc vào đặc trưng của bộ lọc tần thấp và có thể làm cho dải bắt chập lớn bằng dải giữ chập. Bộ so sánh pha I giữ cho PLL ở trạng thái giữ chập mặc dù nhiễu ở lối vào có thể rất lớn.



Hình 9.5: Sơ đồ bộ so sánh pha của CMOS PLL

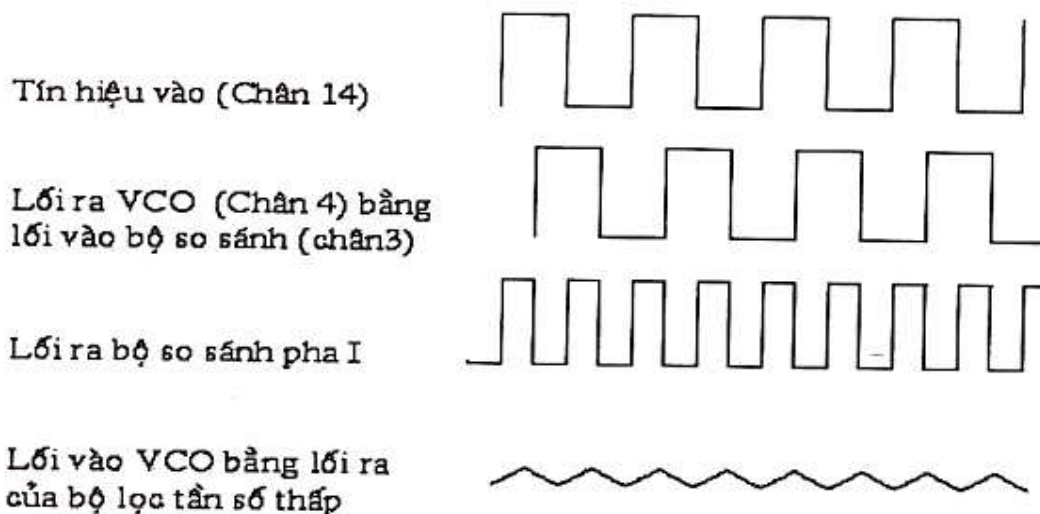
Một đặc trưng quan trọng của so pha kiểu này là nó có thể bám vào tần số rất gần với hoạ ba của tần số trung tâm VCO. Đặc trưng thứ hai là góc lệch pha giữa tín hiệu và xung phát ra từ VCO ở lối vào so sánh nằm trong khoảng $0^\circ \div 180^\circ$ và bằng 90° tại tần số trung tâm.



Hình 9.6: Đặc trưng biểu diễn sự phụ thuộc của điện áp ra trung bình ở lối ra của bộ lọc thông thấp vào độ lệch pha của tín hiệu vào và VCO

Hình 9.6 biểu diễn tam giác đặc trưng của sự lệch pha của các tín hiệu ở hai lối vào bộ so sánh pha. Dạng sóng đặc trưng của vòng bám pha PLL dùng bộ so pha I trong trạng thái giữ chấp được trình bày trên hình 9.7.

9.2.1.2. Bộ so sánh pha II



Hình 9.7: Dạng sóng của PLL dùng so sánh pha I ở trạng thái bám sát của f_0

Bộ so sánh pha II là một mạng nhớ số được điều khiển bằng sườn xung. Bộ so pha II bao gồm 4 tầng RS có chung cửa điều khiển và mạch 3 trạng thái ở lối ra (hình 9.5).

Mạch ba trạng thái gồm hai transistor trường MOS FET kênh p và kênh n. Khi transistor p-MOS hoặc transistor n-MOS ở trạng thái thông mạch (trạng thái ON) chúng đặt lối ra tới V_{DD} hoặc V_{SS} tương ứng. Bộ so pha kiểu này chỉ hoạt động ở sườn dương chuyển từ mức logic 0 lên mức logic 1 của các xung đưa vào lối vào tín hiệu và lối vào so sánh. Nếu tần số xung ở lối vào tín hiệu cao hơn tần số lối vào so sánh thì transistor p-MOS liên tiếp ở trạng thái thông mạch nối lối ra với V_{DD} . Nếu tần số lối vào tín hiệu thấp hơn tần số lối vào so sánh thì transistor n-MOS liên tiếp thông mạch. Nếu hai tín hiệu này cùng tần số nhưng tín hiệu vào chậm pha hơn tín hiệu lối vào của bộ so sánh thì transistor n-MOS sẽ giữ ON trong khoảng thời gian tương ứng với sự lệch pha. Nếu tín hiệu vào sớm pha hơn tín hiệu ở lối vào so sánh thì transistor p-MOS sẽ thông mạch lối ra của bộ so pha được nối lên nguồn V_{DD} trong khoảng thời gian tương ứng với sự lệch pha. Điện thế trên tụ điện ở bộ lọc thông thấp tăng lên. Điện thế trên tụ điện chính là điện áp điều khiển VCO nên tần số phát của VCO tăng dần lên theo tương ứng với sự tăng của điện áp trên tụ. Rồi sau đó, điện thế trên tụ điện ở bộ lọc tần số thấp được điều chỉnh cho tới khi cả hai tín hiệu này bằng nhau về tần số và pha. Tại điểm hoạt động ổn định này cả hai transistor p-MOS và n-MOS ở trạng thái ngắt mạch (trạng thái OFF) lối ra so pha II không được nối với nguồn V_{DD} và V_{SS} , nó trở thành mạch hở và giữ điện thế của tụ tại bộ lọc tần số thấp không đổi. Hơn nữa, tín hiệu tại lối ra của xung pha (phase pulse) ở mức cao có thể dùng để chỉ thị trạng thái giữ chặt. Chính vì vậy, không có sự lệch pha nào tồn tại giữa hai tín hiệu lối vào và tín hiệu so sánh trong toàn bộ dải tần của VCO. Hơn nữa, công suất tổn hao trên bộ lọc tần thấp giảm bớt khi dùng so pha loại này vì cả hai transistor p-MOS và transistor n-MOS ở trạng thái OFF trong hầu hết chu kỳ tín hiệu vào. Chú ý rằng *dải giữ chặt của so pha theo kiểu này bằng dải bắt chặt và không phụ thuộc vào dải thông của bộ lọc tần thấp*. Khi không có tín hiệu ở lối vào, VCO được điều chỉnh phát ở tần số thấp nhất của bộ so pha II. Dạng sóng của PLL dùng so sánh pha II ở trạng thái giữ chặt được mô tả ở hình 9.8 dưới đây:

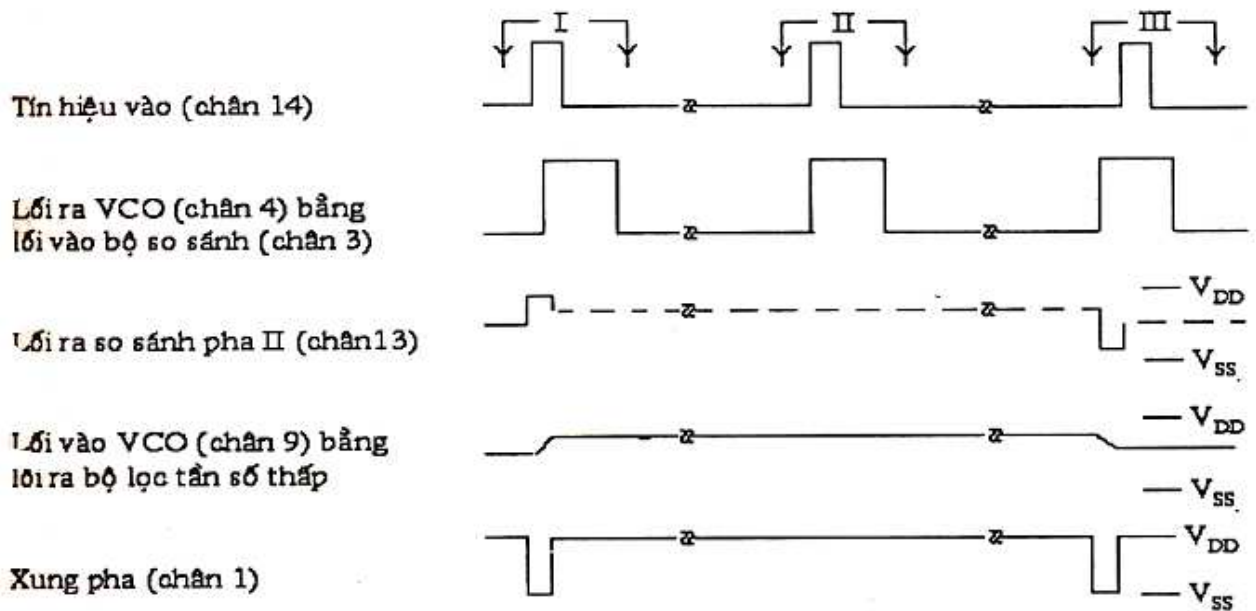
Dạng sóng của PLL dùng so sánh pha II ở trạng thái giữ chặt f_0 (hình 98) được tách làm 3 phần:

- Phần I: Trạng thái tín hiệu lối vào sớm pha so với lối vào bộ so sánh pha.
- Phần II: Miêu tả trạng thái pha của tín hiệu lối vào trùng với pha tín hiệu so sánh.
- Phần III: Trạng thái tín hiệu ở lối vào tín hiệu chậm pha so với tín hiệu ở lối vào so sánh.

Hình 9.9 mô tả giản đồ trạng thái của bộ so sánh pha II; mỗi vòng tròn biểu diễn một trạng thái của bộ so sánh. Số trên cùng trong mỗi vòng tròn biểu diễn trạng thái của bộ so sánh, mức logic 0 hoặc 1 của lối vào tín hiệu và lối vào so sánh được cho bởi chữ số bên trái và phải tương ứng nằm ở hàng dưới cùng trong mỗi một vòng tròn.

Sự chuyển mức logic ở lối vào tín hiệu và lối vào so sánh được biểu thị bởi chữ I và chữ C với các mũi tên lên hoặc xuống. Mũi tên lên ứng với trường hợp mức logic chuyển từ 0 lên 1, mũi tên xuống ứng với trường hợp chuyển mức logic từ 1 về 0.

Khi các lối vào bộ so sánh pha II ở trạng thái 3, 5, 9, 11 thì transistor p-MOS ở lối ra của bộ so sánh pha II thông mạch. Transistor n-MOS thông mạch khi các lối vào bộ so sánh pha II ở một trong các trạng thái 2, 4, 10, 12. Trạng thái 1, 6, 7, 8 tương ứng với trường hợp lối ra so sánh II có trở kháng cao nghĩa là khi đó cả hai transistor n-MOS và p-MOS đều bị ngắt mạch. Lối ra xung pha (chân 1) trong trường hợp này ở mức cao còn các trường hợp khác thì đều ở mức thấp.

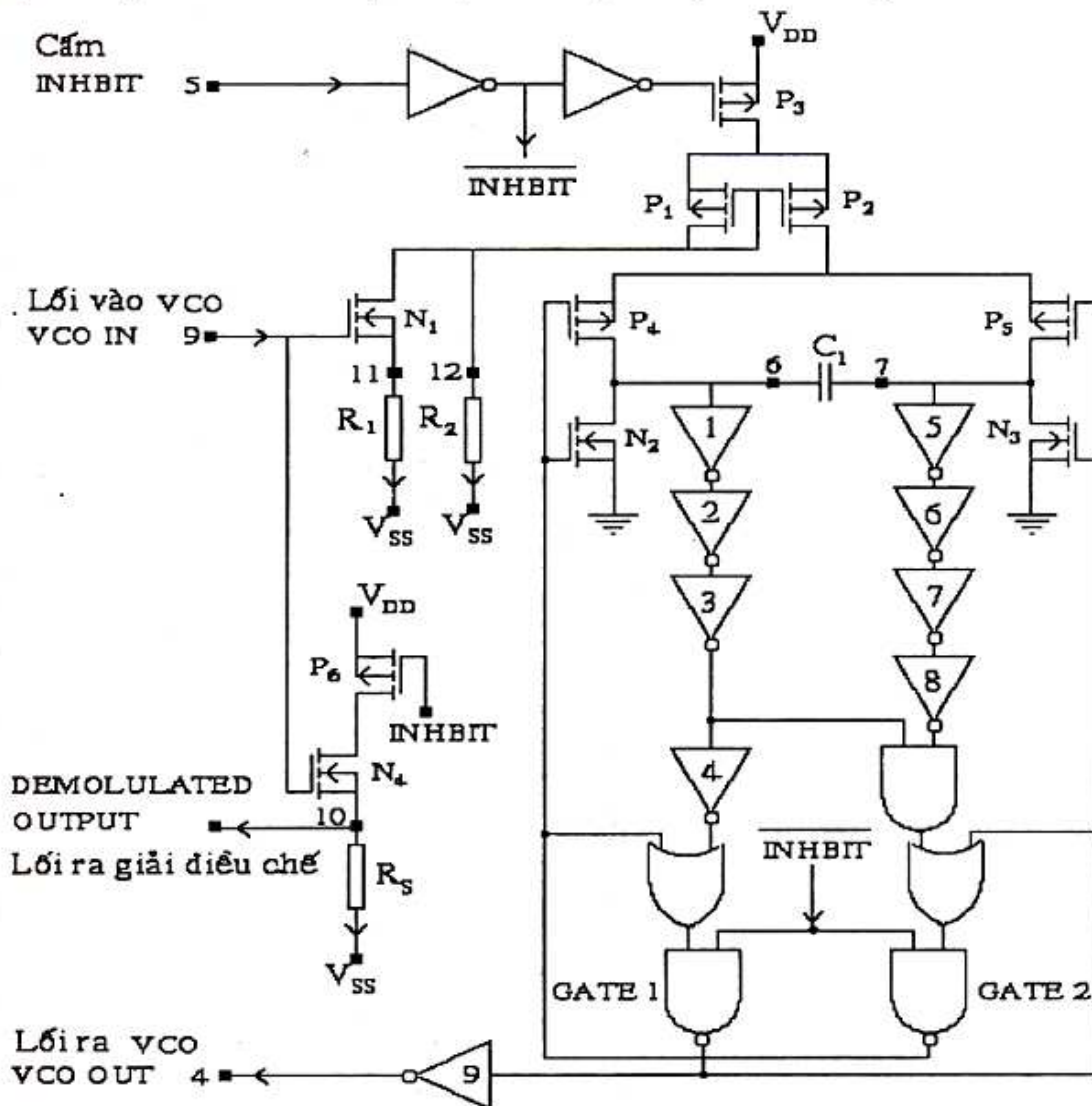


Hình 9.8: Dạng sóng của PLL dùng so sánh pha II ở trạng thái giữ chấp f_0

Chúng ta có thể dùng giản đồ trạng thái để khảo sát hoạt động của mạch so sánh pha II ứng với ba tình huống được nêu trong hình 9.8 miêu tả trạng thái giữ chấp của vòng bám pha, nghĩa là khi đó cả hai tín hiệu ở lối vào tín hiệu và lối vào so sánh có cùng tần số nhưng lệch pha nhau chút ít.

9.2.2. Máy phát điều khiển bằng điện áp VCO (Voltage Controlled Oscillator).

Trên hình 9.10 trình bày sơ đồ của máy phát được điều khiển bằng điện áp (VCO). Để đảm bảo công suất tiêu tán của toàn hệ nhỏ, mạch lọc tần số thấp mắc ở mạch ngoài cũng phải có công suất tiêu thụ nhỏ. Ví dụ, trong mạch lọc cụ thể trên hình 9.10 gồm điện trở RC, giá trị điện trở R phải thật lớn điện dung tụ điện C phải nhỏ. Thiết kế VCO trong trường hợp này có lối vào là transistor trường kênh n được mắc theo sơ đồ lặp lại nguồn nên điện trở lối vào rất lớn. Lối vào của VCO có thể làm thay đổi đặc trưng tần số của mạch lọc, vì lối vào của VCO trong trường hợp này có điện trở vô cùng lớn nên đặc trưng tần số của mạch lọc không hề bị ảnh hưởng của VCO.



Hình 9.10: Sơ đồ của máy phát được điều khiển bằng điện áp (VCO)

Từ nguyên lý của vòng bám pha, chúng ta thấy rằng sự khác nhau về tần số giữa máy phát VCO và tín hiệu lối vào của mạch tách sóng pha tạo thành điện áp sai, qua mạch lọc tần số thấp tạo thành điện áp điều khiển tác động vào máy phát VCO, điều chỉnh dịch tần số của máy phát cho trùng khớp với tần số của tín hiệu vào. Với lập luận trên thì một điện áp sai, điện áp này đóng vai trò điện áp điều khiển tác động vào máy phát VCO làm thay đổi tần số phát f_0 . Một yêu cầu rất quan trọng đặt ra đối với VCO là sự phụ thuộc của tần số vào điện áp điều khiển phải tuyến tính trên khoảng tuyến tính của VCO.

Mạch VCO (hình 9.10) hoạt động như sau: Khi lối vào cấm (inhibit input) có mức logic thấp, P_3 ở trạng thái thông sẽ nối cực nguồn (Source) của P_1 và P_2 với V_{DD} ; cửa logic 1 và cửa logic 2 có chức năng như trigơ RS được tạo thành từ mạch NOR. Transistor trường kênh n N_1 được nối với điện trở ngoài R_1 tạo thành tầng lặp lại nguồn SF (Source - follower).

Tương tự như tầng lặp lại emitor dùng transistor lưỡng cực, đối với tầng lặp lại nguồn nếu thực hiện được điều kiện điện trở R_1 lớn hơn điện trở kênh dẫn của transistor trường kênh n N_1 khi thông mạch, điện trở R_1 phải có giá trị lớn hơn $10\text{ k}\Omega$, khi đó dòng qua R_1 là phụ thuộc tuyến tính vào điện thế lối vào VCO, dòng này đi qua P_1 , hai transistor trường kênh p: P_1 và P_2 tạo thành một mạch dòng gương. Điện trở mắc thêm vào ở mạch ngoài R_2 bổ sung thêm dòng cho nguồn dòng không đổi qua P_1 .

Điện thế điều khiển đặt vào lối vào VCO điều khiển dòng điện qua N_1 , P_1 cũng có nghĩa là điều khiển dòng P_2 nạp điện cho tụ điện C_1 . Trạng thái đặt xoá của trigơ có tác dụng làm đóng ngắt các transistor P_4 và N_3 hoặc P_5 và N_2 . Một bản cực của tụ C_1 được nối với đất (V_{SS}), còn bản cực kia được nguồn dòng không đổi P_2 tích điện. Khi tụ C_1 được nạp đủ điện đến ngưỡng chuyển của các phân tử đảo 1 hoặc 5 trigơ thay đổi trạng thái. Lúc đó, một bản cực của tụ C_1 được nối đất, còn bản mạch kia tích điện âm và tụ phóng điện qua cực máng (drain) của transistor trường n-MOS và một nửa chu kỳ mới bắt đầu. Dòng gương P_2 bằng dòng qua P_1 . Điện áp điều khiển cực của N_1 lối vào của VCO làm thay đổi dòng qua P_1 cũng là dòng P_2 làm thay đổi chu kỳ phóng nạp của tụ C_1 cũng là thay đổi tần số phát của VCO f_0 . Nếu ta không lấy tín hiệu qua bộ lọc tần số thấp, mà lấy ra ở cực nguồn S lối ra của tầng lặp lại nguồn (Source - follower) lối ra chân 10 được dùng làm lối ra giải điều chế. Khi đó, một điện trở tải $R_s = 10\text{ k}\Omega$ hay lớn hơn được nối từ chân này tới đất. Nếu không dùng thì để hở. Mức logic 0 ở lối vào *cấm* (inhibit - chân 5) cho phép VCO và tầng SF ở trạng thái hoạt động. Lối vào *cấm* ở mức logic 1 đặt VCO và Source - follower ở trạng thái chờ tức là trạng thái mà năng lượng tiêu thụ là cực tiểu.

9.2.3. Bộ lọc tần số thấp

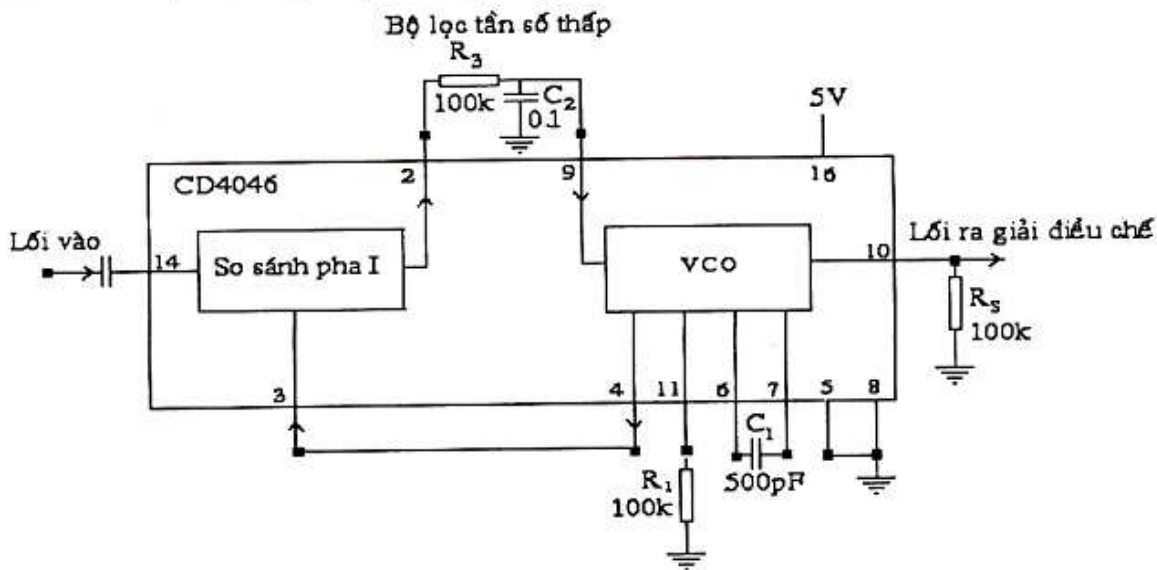
Sự khác nhau về tần số giữa VCO và tín hiệu lối vào qua bộ tách sóng pha và bộ lọc tần số thấp tạo thành điện áp sai $V_e(t)$. Điện áp này đóng vai trò điện áp điều khiển $V_d(t)$ nó tần số phát của VCO.

Nếu tần số tín hiệu lối vào f_s và tần số phát của VCO f_o bằng nhau một cách chính xác thì tín hiệu lối ra của bộ lọc tần số thấp sẽ là một dòng không đổi (một chiều) mà biên độ của nó phụ thuộc vào hiệu pha của hai tín hiệu f_s và f_o .

9.3. NHỮNG ỨNG DỤNG CỦA VÒNG BẮM PHA COS/MOS PLL

9.3.1. Giải điều chế FM

Sơ đồ giải điều chế FM được trình bày trên hình 9.11. Vòng bám pha I được dùng làm mạch tách sóng tín hiệu điều tần. Điện trở R_3 , C_2 tạo thành mạch lọc tần số thấp. Khi vòng bám pha bắt chập với tín hiệu FM, máy phát tần số điều khiển bằng điện áp VCO bám một cách tức thời vào tần số của tín hiệu vào. Bộ tách sóng pha (phase detector) lọc điện áp sai của điện áp lối vào của VCO, điện áp này tương ứng với lối ra điều chế.



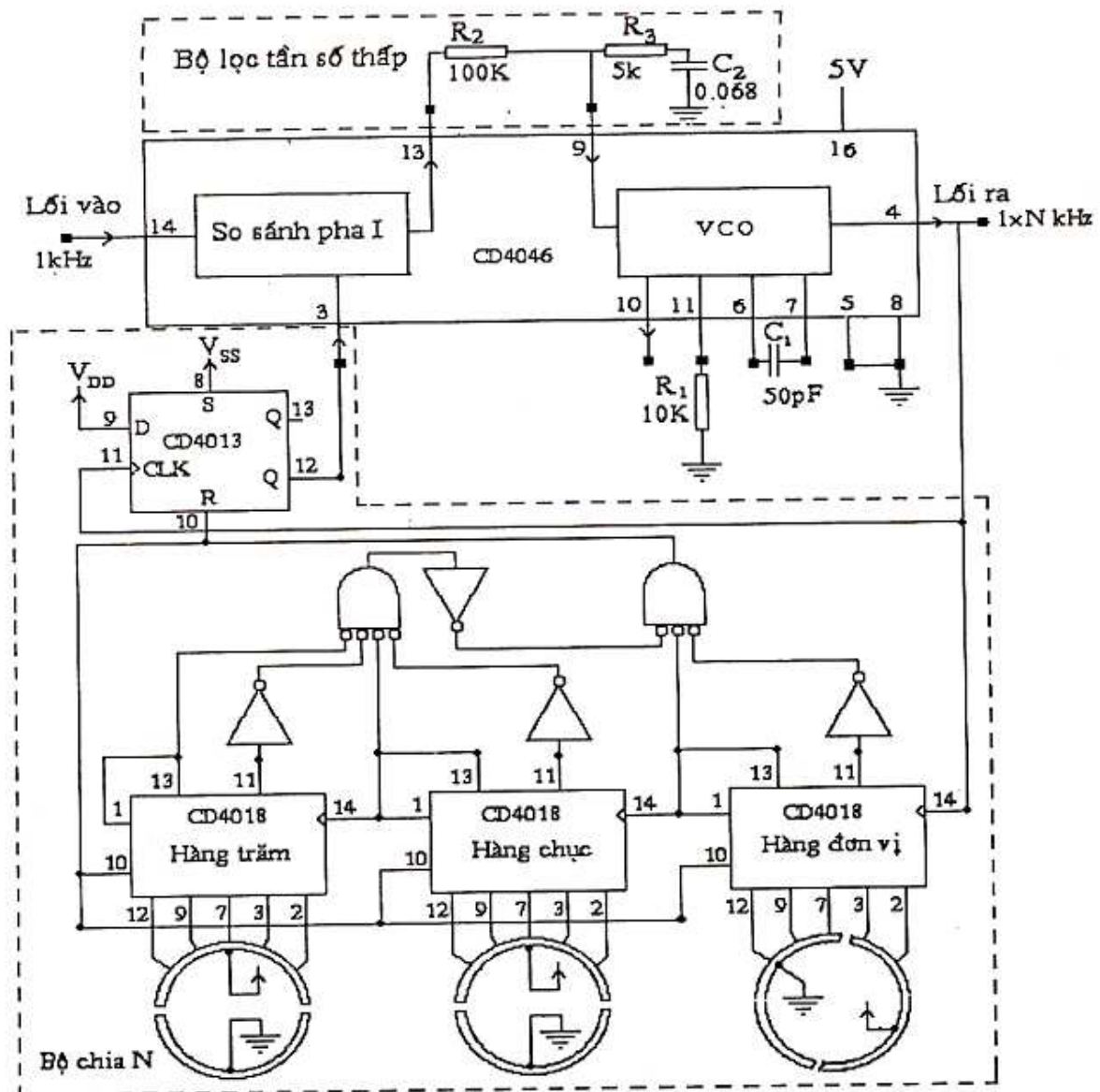
Hình 9.11: Bộ giải điều chế FM

Ví dụ một tín hiệu FM có tần số sóng mang 10 kHz được điều chế bởi một tín hiệu âm tần 400 Hz. Biên độ tổng cộng của tín hiệu FM là 500 mV. Bộ so sánh pha I được dùng vì hệ thống PLL có tần số trung tâm bằng tần số sóng mang FM và đặc tính giảm tạp âm của nó.

Tần số trung tâm của VCO được thiết kế sao cho bằng với tần số sóng mang và bằng 10 kHz. Nếu chọn các linh kiện $C_1 = 500 \text{ pF}$, $R_1 = 100 \text{ k}\Omega$, điện áp cung cấp $V_{DD} = 5 \text{ V}$ ta sẽ được tần số trung tâm là $f_0 = 10 \text{ kHz}$ và dải bắt chập là $f_c \approx \pm 0,4 \text{ kHz}$. Các giá trị của bộ lọc thấp ($R_3 = 100 \text{ k}\Omega$, $C_2 = 0,1 \text{ }\mu\text{F}$) quyết định dải bắt chập trên.

Nếu dùng điện áp nuôi 5 V cho giải điều chế FM, với tỷ số tín hiệu trên tạp là 4 dB thì dòng máng tổng cộng là $132 \text{ }\mu\text{A}$ và với tỷ số tín hiệu trên tạp là 10 dB thì dòng máng tổng cộng là $90 \text{ }\mu\text{A}$.

9.3.2. Bộ tổ hợp tần số



Hình 9.12: Bộ tổ hợp tần số thấp dùng 3 bộ chia thập phân lập trình hoá

Bộ tổng hợp tần số tạo nên từ vòng bảm pha bằng cách đặt một bộ chia vào mạch phản hồi giữa lối ra của VCO và lối vào của bộ so sánh. Hình 9.12 mô tả bộ tổ hợp tần số thấp dùng bộ chia lập trình hoá được gồm 3 bộ chia thập phân. Bộ chia module N có thể thay đổi từng bước từ 1 đến 999 với bước nhảy là 1 đơn vị. Khi hệ thống PLL ở trạng thái giữ chặt, tín hiệu vào và lối vào bộ so sánh cùng tần số và $f = N.1 \text{ kHz}$.

Vì vậy, độ rộng dải tần số của bộ tổ hợp này là từ 3 đến 999 kHz với bước tăng là 1 kHz.

9.4. KỸ THUẬT TỔ HỢP TẦN SỐ DÙNG VÒNG BẢM PHA GHÉP NỐI VỚI MÁY VI TÍNH

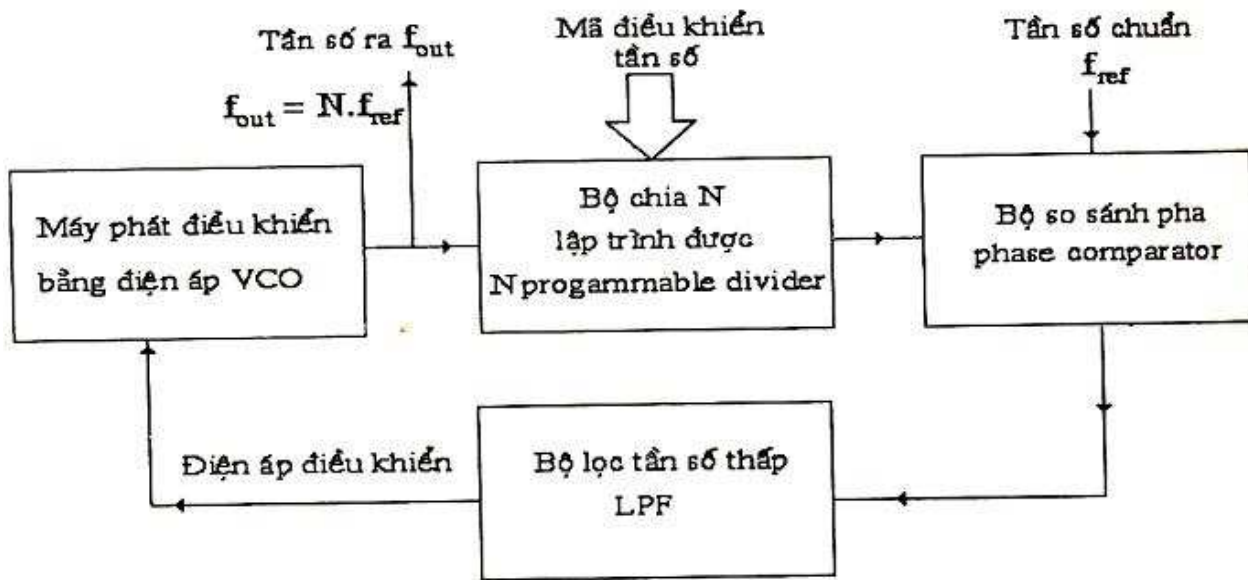
Trên hình 9.13 trình bày sơ đồ khối bộ tổ hợp tần số dùng vòng bảm pha ghép nối với máy vi tính.

Trong sơ đồ này có dùng bộ chia N lập trình được (có thể dùng vi mạch 8253), bộ chia này được ghép nối với vi tính qua cổng máy in LPT1, nhờ vậy ta có thể thay đổi mã số điều khiển thay đổi hệ số chia N một cách linh hoạt.

Tần số lối ra của bộ chia N - đếm/định thời gian lập trình hoá - bảm pha với tần số chuẩn được lấy từ lối ra của một dao động thạch anh. Hệ số chia N có thể thay đổi được nhờ mã điều khiển tần số (Frequency Control Code). Bộ so sánh pha (Phase Comparator) sẽ so pha giữa tần số lối ra của VCO (Voltage Controlled Oscillator) qua bộ chia N với tần số chuẩn tạo ra từ dao động thạch anh f_{ref} , điện áp sai qua bộ lọc tần thấp (LPF - Low Pass Filter) chuyển thành điện áp một chiều biến đổi chậm V_d được đưa vào điều khiển VCO làm cho tần số lối ra của bộ chia N bảm pha với tần số chuẩn. Như vậy, ta sẽ có tần số lối ra của VCO là $f_{out} = N.f_{ref}$, do đó chỉ cần thay đổi hệ số chia N thì ta sẽ tạo được các tần số khác nhau ở lối ra của VCO.

Đối với vòng bảm pha dùng trong các hệ thông tin, tần số lối ra của VCO luôn ở dải tần VHF và UHF, nghĩa là từ 30 đến 400 MHz, do đó việc thiết kế bộ chia N hoạt động trong dải tần số này là rất khó khăn. Thậm chí chung ta cũng không thể dùng mạch công suất lớn ECL cho bộ chia N trên 200 MHz để sử dụng cho các thiết bị thông tin xách tay công suất thấp. Chính vì vậy cần phải hạ thấp tần số, có 2 cách để giảm tần số. Đó là :

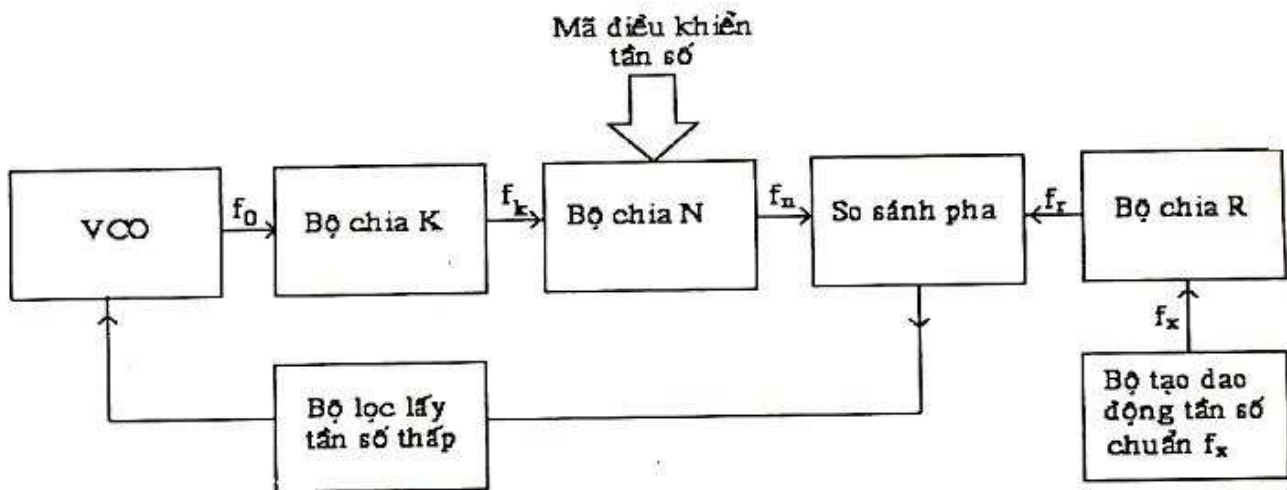
- Kỹ thuật chia thang tỉ lệ trước (Prescaling).
- Kỹ thuật biến đổi hạ thấp tần số (Heterodyne down conversion) mà chúng ta xét đến sau này.



Hình 9.13: Sơ đồ khối bộ tổ hợp tần số dùng vòng bám pha ghép nối với máy vi tính

9.4.1. Kỹ thuật chọn trước thang tỉ lệ (Prescaling)

Trong hình 9.14 trình bày sơ đồ bộ tổ hợp tần số dùng kỹ thuật chọn trước thang tỷ lệ, tần số lối ra của VCO được giảm bởi bộ chia K tới giá trị tần số lớn nhất f_{max} được điều khiển bởi vi mạch chia N. Tần số chuẩn f_r bằng tần số dẫn cách kênh f_c (Channel spacing frequency).



Hình 9.14: Sơ đồ bộ tổ hợp tần số dùng kỹ thuật Prescaling

Tần số lối ra f_{out} VCO được chia với hệ số chia cố định K và hệ số chia thay đổi được N rồi đưa vào bộ so pha, tần số này là:

$$\frac{f_{out}}{K.N}$$

Tần số chuẩn phát ra từ dao động thạch anh f_x sau khi qua bộ chia R cố định cũng được đưa vào bộ so pha và ta sẽ có:

$$f_r = \frac{f_x}{R}$$

Tại thời điểm bám pha, ta có :

$$\frac{f_{out}}{k.N} = \frac{f_x}{R}$$

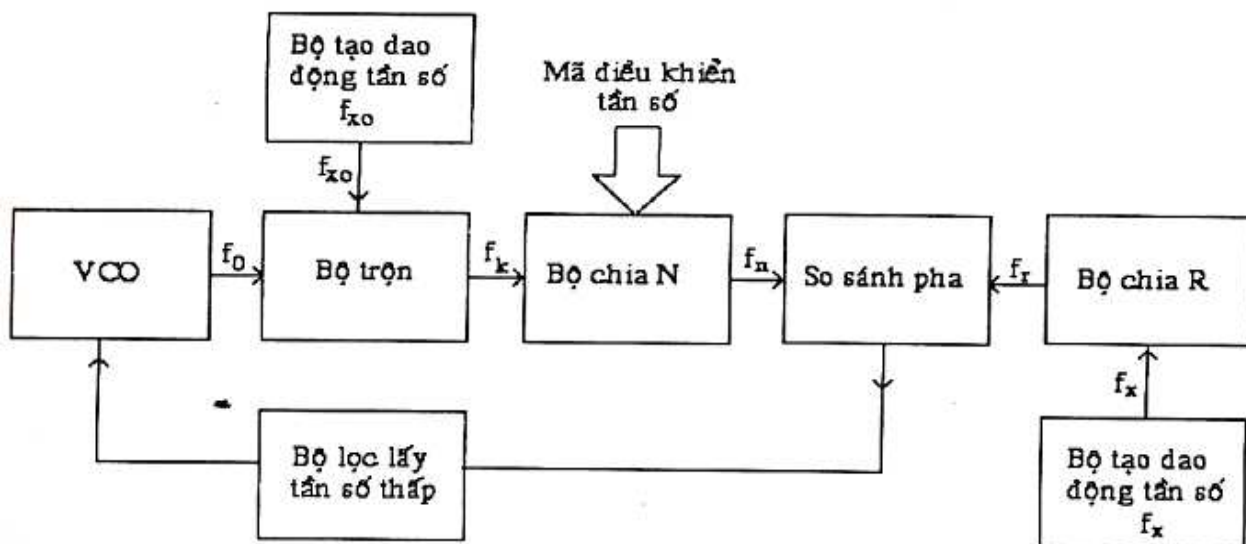
hay:

$$f_{out} = \frac{f_x . K . N}{R}$$

Khi dùng máy tính thay đổi hệ số chia N ta được tần số lối ra f_{out} là bội của tần số $\frac{f_x . K}{R}$.

Kỹ thuật Prescaling chỉ phù hợp với tần số phát ra của VCO không cao. Muốn phát ra ở tần số cao hơn người ta dùng kỹ thuật Heterodyne down conversion. Kỹ thuật này dùng bộ trộn tần để hạ thấp tần số.

9.4.2. Kỹ thuật Heterodyne down conversion



Hình 9.15: Sơ đồ bộ tổ hợp tần số dùng kỹ thuật Heterodyne down conversion

Chúng ta có sơ đồ của bộ tổ hợp tần số dùng kỹ thuật Heterodyne down conversion như hình 9.15. Máy phát dao động thạch anh offset được dùng để trộn xuống (Mix down). Tần số lối ra của bộ trộn là f_k :

$$f_k = f_{out} - f_{xo}$$

Đây cũng chính là tần số lối vào của bộ chia N. Khi hệ chấp với tín hiệu ta có:

$$f_{out} - f_{xo} = \frac{Nf_x}{R}$$

hay:
$$f_{out} = \frac{N \cdot f_x}{R} + f_{xo}$$

Khoảng chia N được xác định bởi hai biểu thức sau:

$$N_{max} = \frac{f_{k\ max}}{f_c}$$

$$N_{min} = \frac{f_{k\ min}}{f_c} \quad (f_c \text{ là tần số độ rộng kênh})$$

Trong các hệ tổ hợp tần số dùng kỹ thuật Heterodyne down conversion có thể có hơn một giá trị của tần số offset f_{xo} được đặt. Ví dụ, nếu dải tần số lối ra VCO được chia làm hai khoảng bằng nhau và nếu giá trị của f_{xo} được đặt cho hai khoảng tương ứng thì hệ số chia N sẽ bằng đúng một nửa hệ số chia dùng trong hệ thống Presscaling.

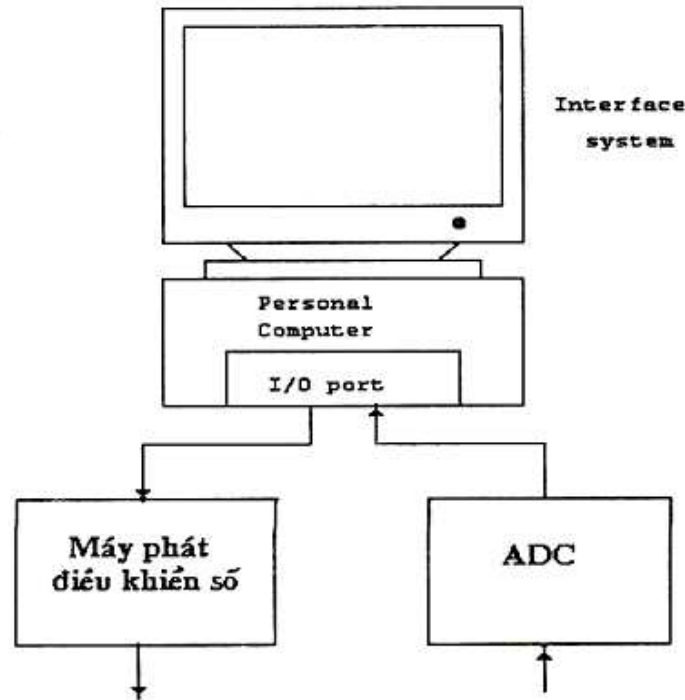
Ưu điểm của kỹ thuật Heterodyne down conversion là:

- Tần số chuẩn f_r bằng tần số độ rộng kênh f_c nên dải thông của hệ rộng hơn.
- Khi dùng kỹ thuật này thì mất mát năng lượng là ít nhất, do đó tần số dao động offset và bộ trộn (Mixer) hoạt động với hiệu suất rất cao.

9.5. MÁY PHÁT ĐIỀU KHIỂN SỐ GHÉP NỐI VỚI MÁY VI TÍNH

Như chúng ta đã biết, máy phát VCO (Voltage Controlled Oscillator) có thể điều khiển được tần số phát bằng điện áp. Ghép nối DAC với máy vi tính, từ máy tính qua bộ DAC ta có thể đưa vào lối vào của VCO một điện áp. Điện áp này có thể dễ dàng điều khiển được nhờ một chương trình phần mềm.

9.5.1. Cấu trúc của máy phát điều khiển số



Hình 9.16a

Nội dung mục này nhằm giới thiệu một mô hình điều khiển số, điều khiển thay đổi băng tần và tần số phát của một máy phát xung sơ đồ khối mô tả hệ ghép nối được trình bày trên hình 9.16a.

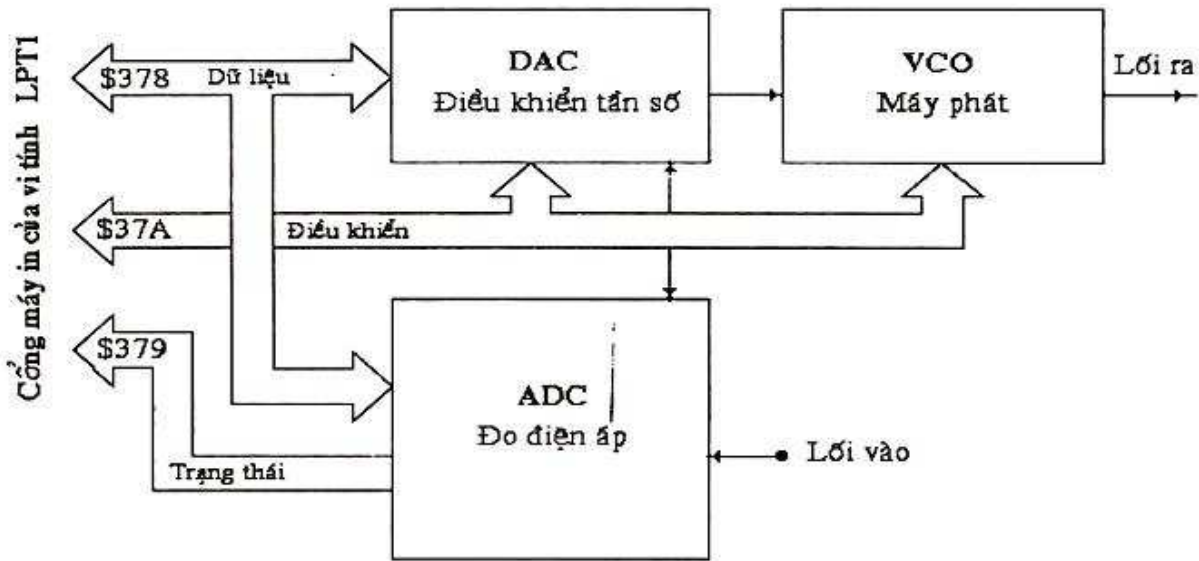
Cấu trúc của máy phát điều khiển số được trình bày trên hình 9.16b. Hình 9.17 là sơ đồ nguyên lý của máy.

Máy phát điều khiển số được tạo thành từ hai khối cơ bản là: DAC MC1408 và vòng bảm CMOS CD 4046 chúng ta đã khảo sát chúng ở trên. Trong bài tập thực hành này chúng ta dùng chúng cùng với chuyển mạch tương tự 4016 để tạo thành một máy phát điều khiển số. Máy được ghép nối với vi tính thông qua cổng máy in LPT1.

Như chúng ta đã biết, trong mạch vòng bảm pha có bộ dao động được điều khiển bằng điện áp VCO, tần số dao động của VCO có thể thay đổi nhờ thay đổi điện áp đưa vào lối vào điều khiển VCO.

Máy vi tính ở đây đóng vai trò của thiết bị điều khiển thay đổi tần số phát của VCO thông qua việc điều khiển dữ liệu số đưa vào DAC nhằm làm thay đổi điện áp ra của nó và điện áp này được dùng để điều khiển VCO.

Hình 9.16b cho chúng ta thấy tần số phát được điều khiển qua hai đường:



Hình 9.16b

a/ Đường thứ nhất qua BUS điều khiển của cổng máy in LPT1 có địa chỉ là \$37A, điều khiển chuyển mạch tương tự 4016 được nối với hệ các tụ điện của mạch VCO làm thay đổi điện dung tương đương của hệ thống tụ này. Như vậy, đường điều khiển thứ nhất chỉ điều khiển thay đổi dải tần số của VCO. Tùy theo lệnh điều khiển của máy tính máy có thể phát ở 3 băng tần số khác nhau.

b/ Đường thứ hai điều khiển qua BUS dữ liệu của cổng LPT1 có địa chỉ là: \$378. Đường điều khiển này điều khiển mã số lối vào của DAC làm thay đổi từng nấc điện áp ra của DAC. Điện áp này điều khiển làm cho tần số phát của VCO cũng thay đổi từng nấc tương ứng với mã số lối vào DAC. Mạch MC1408 là bộ DAC 8 bit nên điện áp ra của nó có 255 mức điện áp, tuy nhiên đặc trưng điện áp - tần số của VCO có đoạn thay đổi điện áp vào mà tần số phát của nó không thay đổi nên khi làm thí nghiệm gặp phải trường hợp này chúng ta cũng đừng vội sốt ruột.

Ngoài khối phát điều khiển số chúng ta đã đề cập đến ở trên, trong thiết bị còn khối ADC cũng được ghép nối với vi tính được dùng để đo các điện áp tương tự biến đổi các tín hiệu tương tự thành dạng số lưu giữ trong máy tính chúng ta sẽ khảo sát chức năng của nó trong bài thí nghiệm thực hành

sau. Trên hình 9.17 là sơ đồ nguyên lý của máy phát điều khiển số tương ứng với sơ đồ khối hình 9.16b. Trong sơ đồ này IC1, IC2, IC3A, IC3B tạo thành bộ biến đổi DAC 8 bit. Điện áp ra của DAC này được dùng để điều khiển làm thay đổi tần số phát của VCO do IC5 và IC6 tạo nên. Giắc cắm J2 được nối với BUS điều khiển \$37A của cổng LPT1. Qua các đường điều khiển này ta có thể lập trình điều khiển đóng ngắt các chuyển mạch tương tự có trong IC5 CD4016 làm thay đổi các giá trị của tụ tương đương ghép ở giữa hai chân CA, CB của mạch PLL 4046. Nhờ cách điều khiển này ta có thể điều khiển thay đổi băng tần phát của VCO. Trên hình 9.17 ta còn thấy một mạch DAC nữa là IC7. Vi mạch này kết hợp với các mạch khuếch đại thuật toán IC3C, IC3D và mạch so sánh vi sai tạo thành bộ biến đổi tương tự số ADC ghép nối với vi tính cũng qua cổng máy in.

9.5.2. Nội dung bài tập thực hành

1. Dụng cụ thí nghiệm

- Máy phát điều khiển số
- Máy tính
- Dao động ký
- Von kế hiện số
- Panel thí nghiệm trên đó có các vi mạch: CD4046 (mạch vòng bảm pha CMOS), MC1408 (DAC 8bit) LM311 (mạch so sánh vi sai) LM324 (khuếch đại thuật toán). Sơ đồ nguyên lý của panel thí nghiệm này tương tự như sơ đồ nguyên lý của máy phát điều khiển số (xem hình 9.17).

2. Nội dung thực hành

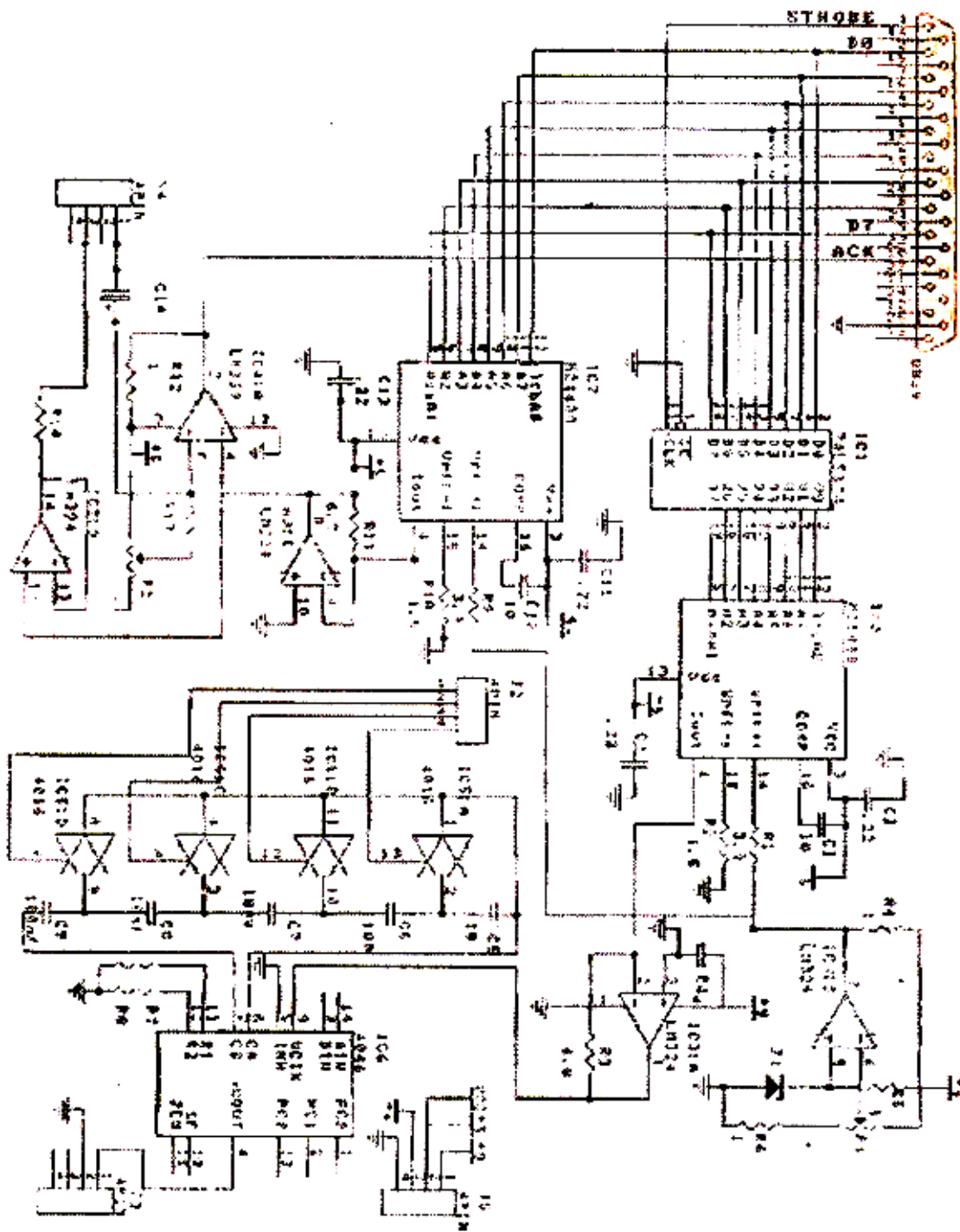
1/ Khảo sát nguyên tắc hoạt động của PLL

a- Khảo sát hoạt động của VCO. Đo đặc trưng biểu diễn sự phụ thuộc tần số phát của VCO vào điện áp điều khiển V_d .

b- Khảo sát hoạt động của bộ so sánh pha. Đo dải bắt chấp và dải giữ chấp của vòng bảm pha.

2/ Khảo sát hoạt động của máy phát điều khiển số

a- Cho chạy chương trình kiểm tra máy phát có tên tệp là: TESTOSC. Dùng dao động ký xác định tần số phát ứng với 3 băng tần.



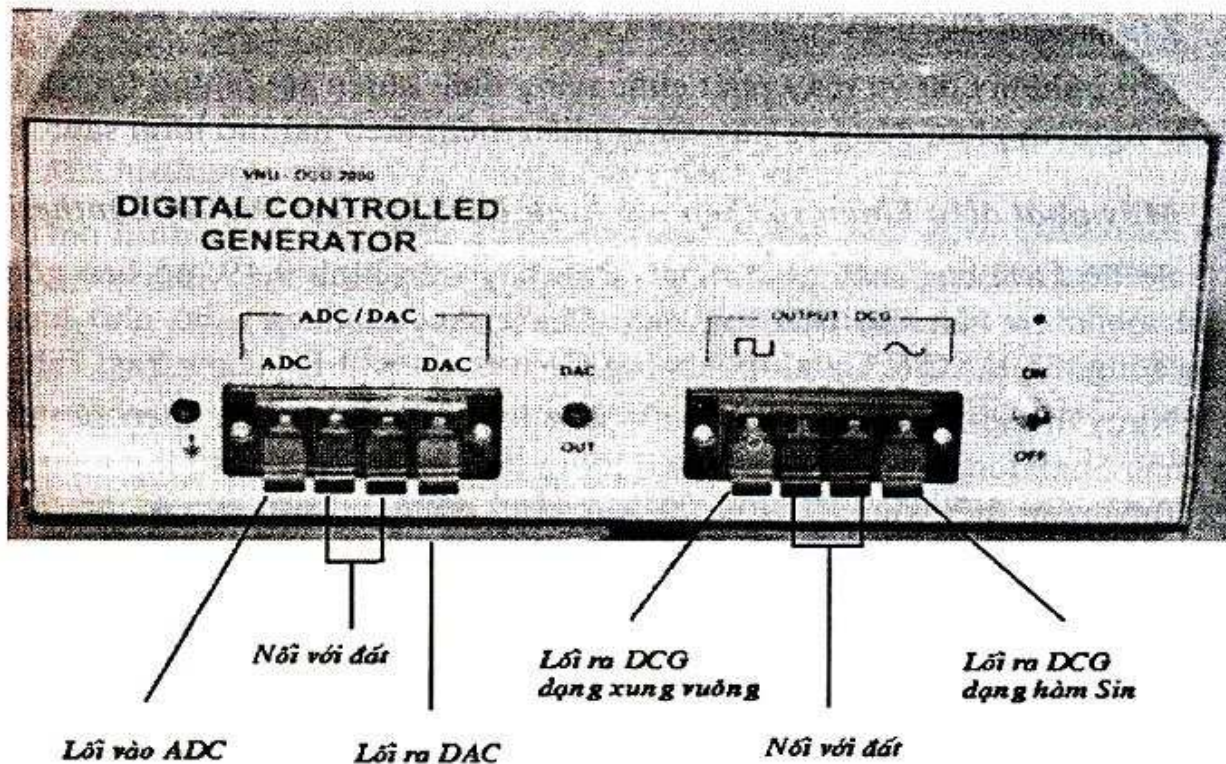
Hình 9.17: Sơ đồ nguyên lý máy phát điều khiển số

Đánh các giá trị kết quả xác định tần số vào máy tính sau khi đo xong 255 giá trị cho một băng tần, nhớ lưu kết quả với tên tệp BAND (số thứ tự của băng tần).DAT.

Lưu ý: chương trình bắt buộc phải đo xong 255 giá trị mới được kết thúc và phải đánh đúng số thứ tự của băng tần vừa đo.

b- Cho chạy chương trình điều khiển phát DCG với 3 tệp dữ liệu vừa đo được: BAND1.DAT; BAND2.DAT; BAND3.DAT. Đối chiếu tần số phát được hiển thị trên màn hình máy tính và tần số phát hiển thị trên dao động ký. Đánh giá kết quả độ chính xác của tần số máy phát.

Hình 9.18 là ảnh chụp mặt trước máy phát điều khiển số.



Hình 9.18: Mặt trước của máy phát điều khiển số

Mặt sau máy có giắc cắm cổng LPT1 dùng để ghép nối với cổng máy in của máy vi tính. Sơ đồ nguyên lý bên trong máy tương tự sơ đồ hình 9.17.

Máy phát điều khiển số dùng PLL, tích hợp cả ADC và DAC. Dùng máy này sinh viên có thể lập trình điều khiển ứng dụng các ngôn ngữ lập trình điều khiển khác nhau, điều khiển DAC, ADC, PLL. Dưới đây giới thiệu máy phát điều khiển số dùng ngôn ngữ lập trình Delphi.

9.6. MÁY PHÁT ĐIỀU KHIỂN SỐ VÀ LẬP TRÌNH MÔ PHÒNG

Để tường minh cho những ứng dụng của kỹ thuật số trong lĩnh vực kỹ thuật điện tử, đo lường điều khiển đoạn cuối chương này chúng tôi trình bày

kết quả nghiên cứu thiết kế máy phát điều khiển số và lập trình mô phỏng ứng dụng trong công tác giảng dạy. Máy phát có cấu hình phần cứng tối thiểu gồm: biến đổi số tương tự DAC, vòng khoá pha PLL, biến đổi tương tự số ADC. Kết quả nghiên cứu đã công bố trên tạp chí “Khoa học và công nghệ” của Trung tâm Khoa học và Công nghệ Quốc gia [11], bài báo cũng giới thiệu phần mềm điều khiển mô phỏng. Chương trình điều khiển mô phỏng được xây dựng từ công cụ lập trình DELPHI.

9.6.1. Giới thiệu máy phát điều khiển số

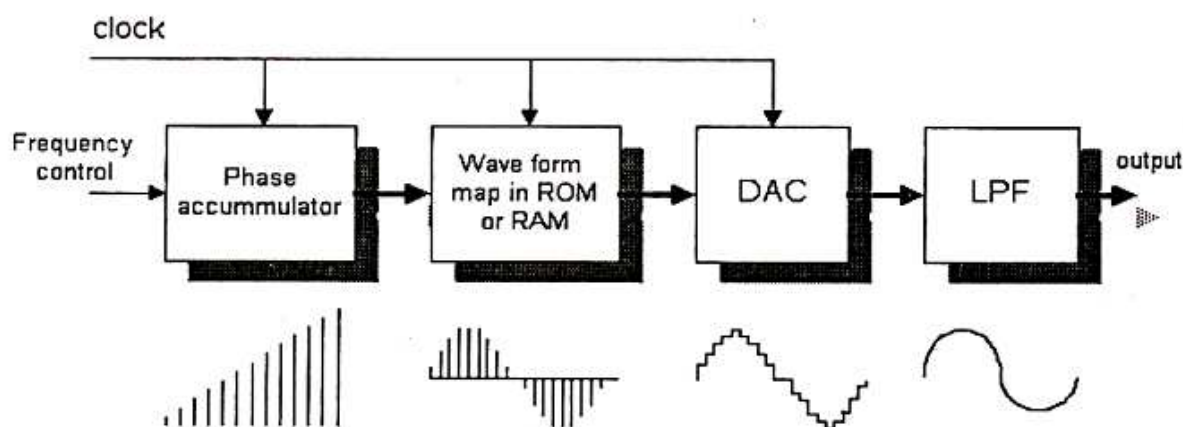
Khi nghiên cứu về máy phát chức năng điều khiển số, chúng tôi đã thiết kế thử nghiệm phần cứng và xây dựng phần mềm theo hai mô hình sau:

Máy phát điều khiển số theo mô hình DDS (Direct Digital Synthesis)

Sơ đồ khối của máy phát được trình bày trên hình 9.19, nó bao gồm 4 thành phần cơ bản: bộ tích lũy pha (Phase Accumulator), bộ nhớ lưu trữ dạng sóng, bộ DAC (Digital to Analog Converter) và LPF (Low Pass Filter).

Nguyên tắc hoạt động của máy phát loại DDS như sau: theo lý thuyết lấy mẫu Shannon, để đảm bảo tần số lối ra là $F_0 = 2\pi/\omega_0$, tần số lấy mẫu F_{clock} phải thoả mãn điều kiện:

$$F_{\text{clock}} \geq 2F_0$$



Hình 9.19

Giả sử dạng sóng sin phát ra được biểu diễn bởi phương trình $A \sin(\omega_0 t + \varphi_0)$

Ta có $\varphi(t) = \omega_0 t + \varphi_0$ là một hàm tuyến tính theo thời gian nên:

$$\frac{d\varphi(t)}{dt} = \omega_0.$$

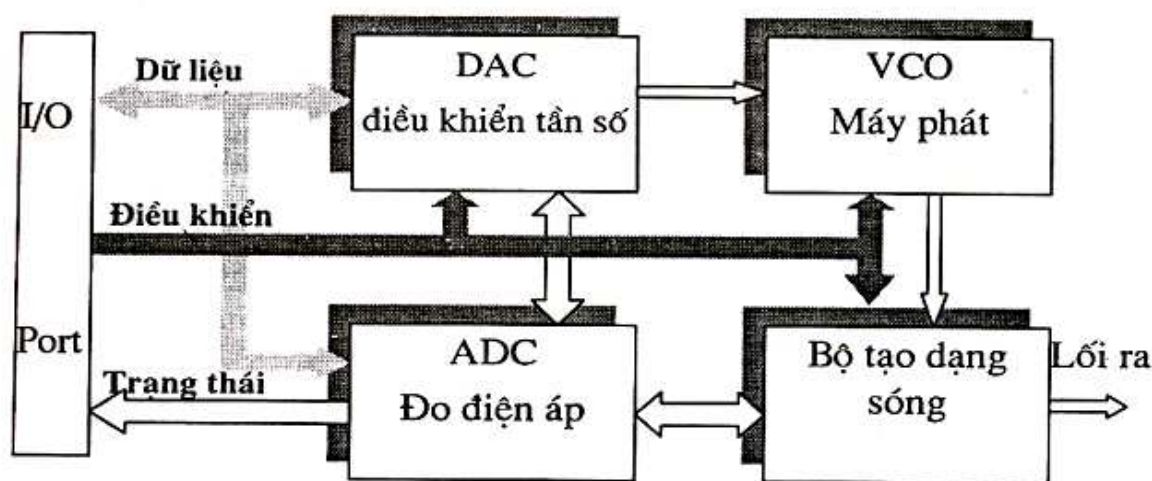
Do vậy bộ tích lũy sẽ phải tạo ra các mẫu $S(n)$ rời rạc liên tiếp cách nhau khoảng thời gian $T=1/F_{\text{clock}}$, được biểu diễn bởi phương trình:

$$S(n) = S(n - 1) + \omega_0$$

Để thu được tín hiệu lối ra hình sin ta phải biến đổi $S(n)$ thành dạng sin $[S(n)]$, sau đó đưa vào bộ nhớ lưu trữ dạng sóng. Lối ra của bộ nhớ lưu trữ dạng sóng được đưa tới bộ biến đổi DAC. Bộ lọc thông thấp LPF có tác dụng làm trơn tru dạng sóng sin thu được ở lối ra, muốn tìm hiểu sâu về phương pháp này xin xem tài liệu tham khảo [9]. Xây dựng máy phát điều khiển số theo mô hình này giúp ta dễ dàng tạo ra nhiều dạng sóng khác nhau bằng các thuật toán phần mềm.

Mô hình máy phát điều khiển bằng điện áp

Mô hình máy phát điều khiển số (hình 9.20), bao gồm DAC một máy phát được điều khiển bằng điện áp VCO (Voltage Controlled Oscillator) có trong mạch vòng bám pha PLL. Điện áp điều khiển được lấy từ điện áp lối ra của DAC. Bộ tạo dạng sóng tạo ra hai tín hiệu có dạng sóng khác nhau là xung vuông và dạng sóng hình sin ở lối ra của thiết bị. Bộ ADC dùng để giám sát biên độ tín hiệu phát, nó còn được dùng để cấu thành một hệ đo đặc trưng tần số của mạch truyền hoặc các bộ khuếch đại dùng trong kỹ thuật điện tử. Các khối chức năng này được ghép nối với cổng vào ra của máy tính theo các đường bus điều khiển, dữ liệu và trạng thái.



Hình 9.20

Chúng tôi đã thiết kế máy phát điều khiển số tương ứng với sơ đồ khối hình 9.20, để tạo thành DAC chúng tôi đã dùng các vi mạch 74LS374, MC1408, LM324. Khối VCO được tạo thành từ các vi mạch chuyển mạch

tương tự CD4016, mạch vòng tám pha CD4046. Khối tạo dạng sóng được thực hiện bằng phần cứng. Lối ra của máy phát cho ra hai dạng sóng:

- Dạng xung vuông tần số phát F1. Dải tần từ 20 Hz đến 500 kHz được chia làm 3 băng tần. Độ ổn định tần số của máy phát đạt 10^{-3} . Tần số phát được đo lấy mẫu trực tiếp bằng tần kế số có độ chính xác 10^{-7} . Kết quả đo được lưu giữ thành 3 file dữ liệu tương ứng với 3 băng tần của máy phát với tên file là: band1.dat, band2.dat, band3.dat. Tần số phát được hiển thị trực tiếp trên màn hình điều khiển của máy tính. Biên độ xung và sườn xung tương hợp với mức logic họ vi điện tử, họ TTL và họ CMOS dùng nguồn nuôi vi mạch $V_{DD} - V_{SS} = +5$ V.
- Dạng điều hoà hình sin có tần số phát F2. Dải tần phát: 2 Hz ÷ 50 kHz được chia làm 3 băng tần. Tần số F2 = F1/10. Biên độ tín hiệu ra cực đại $V_{PP\ MAX} = 5$ V, trở kháng ra 10 Ω . Tín hiệu ra hình sin đồng bộ với xung vuông rất tiện lợi khi dùng thiết bị này để tiến hành các thí nghiệm về điều chế và giải điều chế số dùng trong truyền thông số.

Sơ đồ nguyên lý của thiết bị tương tự sơ đồ 9.17, thiết bị được ghép nối với PC qua cổng máy in LPT1. Máy vi tính ở đây đóng vai trò thiết bị điều khiển thay đổi tần số phát của VCO thông qua việc điều khiển dữ liệu số đưa vào DAC nhằm làm thay đổi điện áp ra của nó và điện áp này được dùng để điều khiển VCO. Chúng ta có thể xây dựng các chương trình phần mềm để máy tính thực hiện các chức năng điều khiển, giám sát và mô phỏng các hoạt động của máy phát điều khiển số một cách linh hoạt.

9.6.2. Phần mềm mô phỏng thiết bị

Với mô hình phần cứng được thiết kế tương ứng theo các mô hình trên, chúng tôi đã xây dựng phần mềm điều khiển và mô phỏng thiết bị. Theo mô hình DDS để tạo ra các dạng sóng khác nhau có biên độ và tần số thay đổi được bằng phần mềm, ta có thể sử dụng các thủ tục đơn giản sau:

```
const sm=10000;  
var ap : array[0..sm] of byte;  
{ Tạo bộ tích lũy }  
Procedure MangPhatRangCua (ts:word, bd:byte);  
var k,m :word;  
begin
```



```

    for i:=0 to ts-1 do
        for m:=0 to (sm div ts) do
            ap[m+k*(sm div ts)]:= trunc(bd*m/(sm div ts));
        end;
    end;
Procedure MangPhatVuong (ts:word, bd:byte);
var k,m :word;
begin
    for k:=0 to ts-1 do
        for m:=0 to (sm div ts) do
            if m<=(sm div ts) div 2 then ap[m+k*(sm div ts)]:=bd
                else ap[m+k*(sm div ts)]:=0;
        end;
    end;
Procedure MangPhatSin (ts:word, bd:byte);
var k,m :word;
begin
    for m:=0 to sm do
        ap[m]:=bd + trunc ((bd/2)*sin((m/sm)*ts*2*Pi));
    end;
    {đưa dữ liệu điều khiển ra cổng LPT1}
Procedure PhatMang;
var p :word;
begin
    repeat
        for p:=0 to sm do port[$378]:=ap[p];
    until keypressed;
end;

```

Theo mô hình máy phát điều khiển số hình 9.20 ta có thể lập trình điều khiển thay đổi băng tần phát, thay đổi tần số phát bằng con trỏ chuột. Dưới đây là một số thủ tục trong phần mềm điều khiển máy phát điều khiển số tương ứng với sơ đồ khối hình 9.20.

Các thao tác hiển thị dạng sóng trên màn hình máy tính được thực hiện trong thủ tục CapNhatHienThi tương ứng với các thay đổi trong điều khiển phần cứng. Khi đó ta dùng các thủ tục sau để điều khiển thay đổi băng tần và phát tần số theo chế độ tự động tăng giảm (mã nguồn này được viết trong Delphi).

```
procedure TForm1.Band1BtnClick(Sender: TObject);
begin
    outport$37A($3); {Điều khiển thay đổi băng tần và
chuyển mạch tương tự lối ra}
    CapNhatHienThi;
end;

procedure TForm1.Band2BtnClick(Sender: TObject);
begin
    outport$37A($15); {Điều khiển thay đổi băng tần và
chuyển mạch tương tự lối ra}
    CapNhatHienThi;
end;

procedure TForm1.Band3BtnClick(Sender: TObject);
begin
    outport$37A($9); {Điều khiển thay đổi băng tần và
chuyển mạch tương tự lối ra}
    CapNhatHienThi;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
{Thủ tục tự động quét tần số theo thời gian}
begin
```

```

outport$378(i);
CapNhatHienThi;
i:=i+1;
if i:=256 then i:=0;
end;

procedure TForm1.ScrollBarTanSoChange(Sender:
TObject);
{Thủ tục thay đổi tần số bằng thanh cuộn}
begin
outport$378(ScrollBarTanSo.Position);
(đưa dữ liệu ra DAC điều khiển VCO)
CapNhatHienThi;
end;

```

Phần mềm điều khiển và mô phỏng được xây dựng bằng công cụ lập trình Delphi của hãng Borland. Đây là công cụ thiết kế phát triển ứng dụng nhanh RAD (Rapid Application Development) giúp xây dựng các ứng dụng Window32 có giao diện người dùng đồ họa GUI (Graphic User Interface) đẹp và thân thiện.

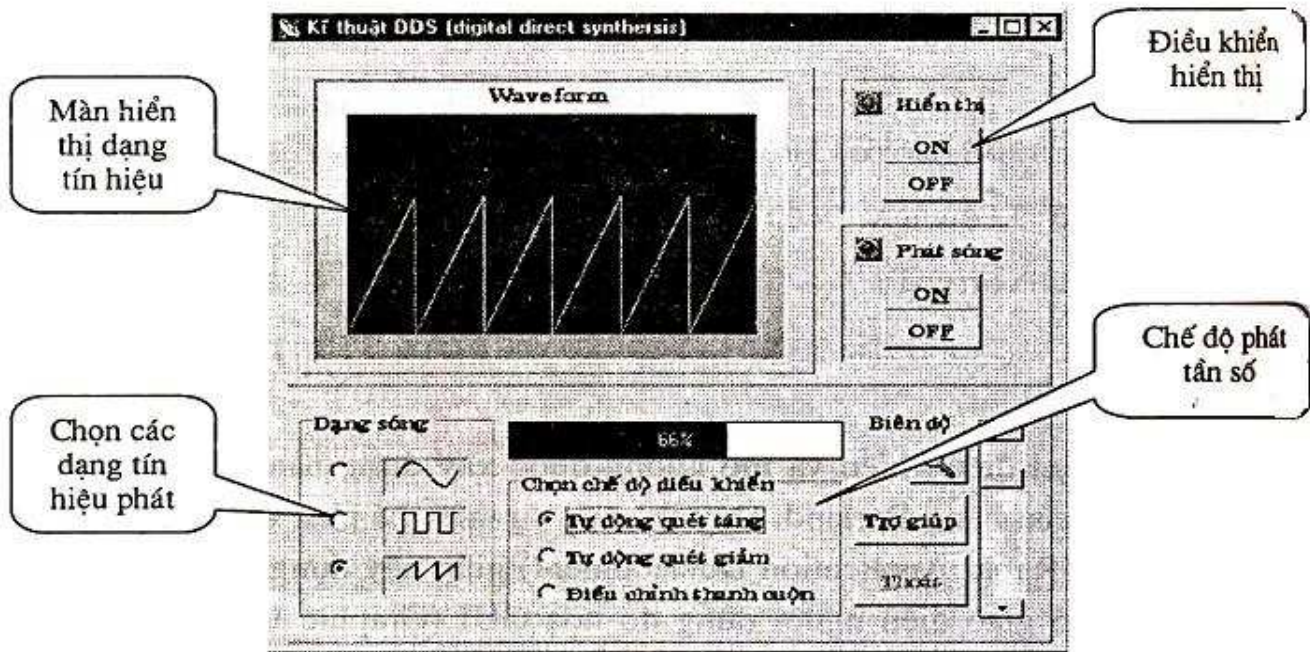
Cấu trúc hướng đối tượng và hướng sự kiện của Delphi giúp người lập trình có khả năng thiết kế mô phỏng nhanh và hiệu quả [10].

Khả năng tương tác của chương trình với phần cứng được thực hiện thông qua cổng LPT1 với ba thanh ghi có địa chỉ 378h, 379h và 37Ah. Chương trình gồm hai môđun chính PLL và DAC được lập trình theo hai mô hình máy phát điều khiển số đã được đề cập ở trên.

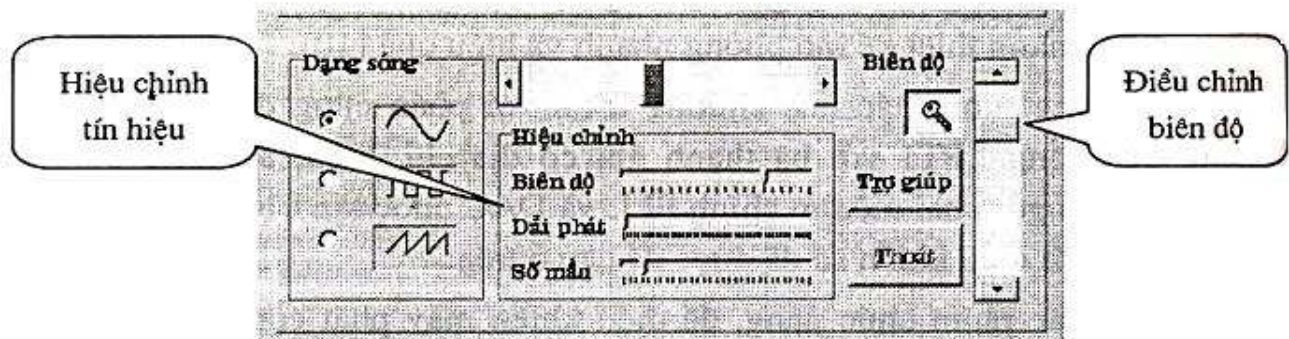
Ngoài các phím chức năng, để điều khiển máy phát còn có màn hình hiển thị mô phỏng dạng sóng giống như tín hiệu quan sát trên dao động ký ở lối ra của máy phát điều khiển số. Bên cạnh đó, chương trình cũng có khả năng cảnh báo các lỗi của thiết bị phần cứng.

Hình 9.21 và 9.22 là màn hình điều khiển và hiển thị mô phỏng dạng sóng của máy phát điều khiển số có cấu hình tương ứng với sơ đồ khối hình 9.20.

Thiết bị điều khiển số và đặc biệt là phần mềm mô phỏng thiết bị trên máy tính mang hiệu quả sử dụng cao do tính đơn giản của thiết bị phần cứng cùng với tính năng mở dễ dàng thay đổi cập nhật của phần mềm. Như đã thấy trên hình 9.23 và hình 9.24, với phần cứng tối thiểu rẻ tiền, dễ thực hiện trong điều kiện thực tế Việt Nam nhờ kết nối với vi tính đã phát huy được khả năng điều khiển mềm dẻo và xử lý thông tin đa dạng của máy vi tính.

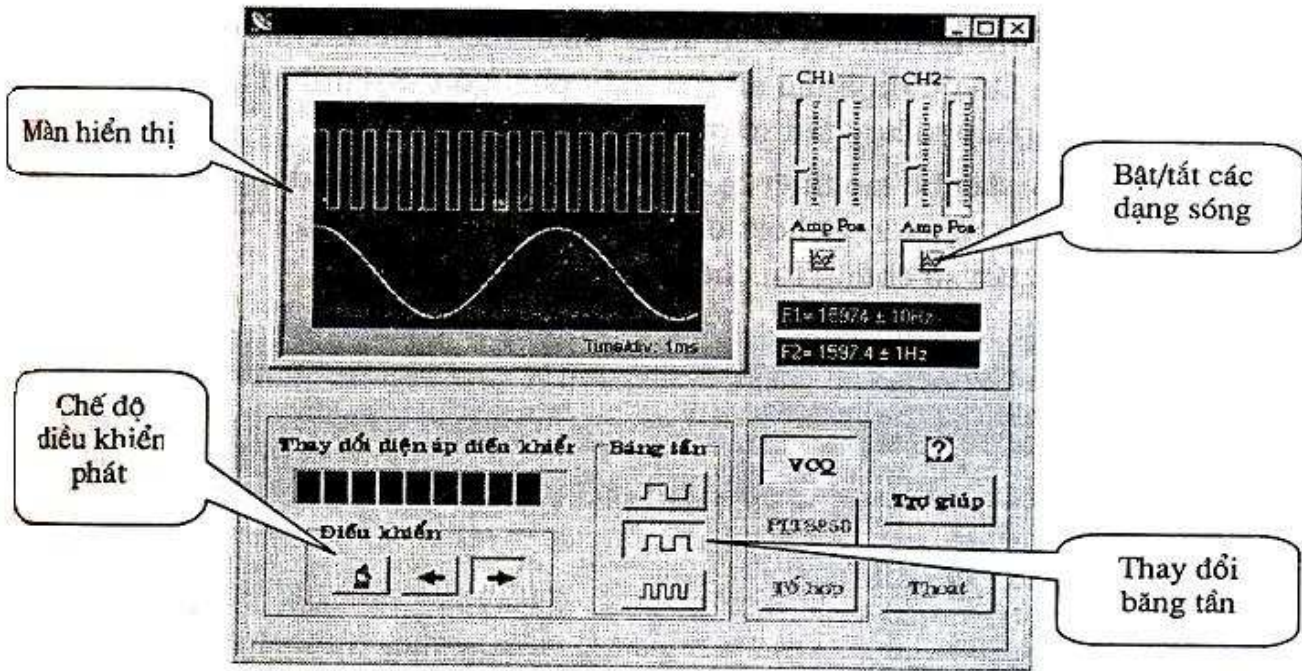


Hình 9.21

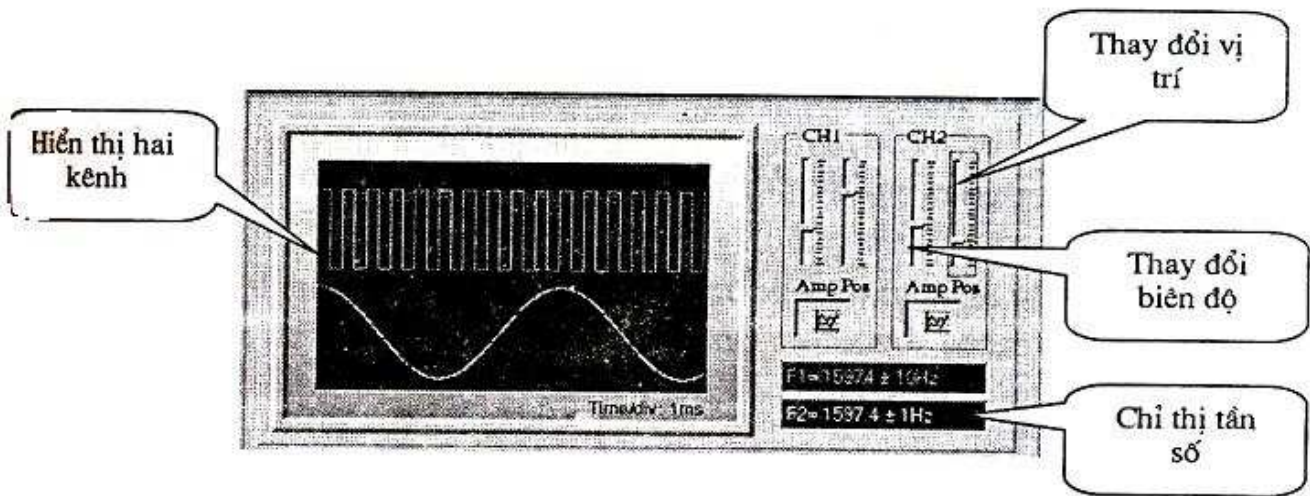


Hình 9.22

Hình 9.23 và 9.24 là màn hình điều khiển và hiển thị mô phỏng dạng sóng của máy phát điều khiển số có cấu hình tương ứng với sơ đồ khối hình 9.20.



Hình 9.23

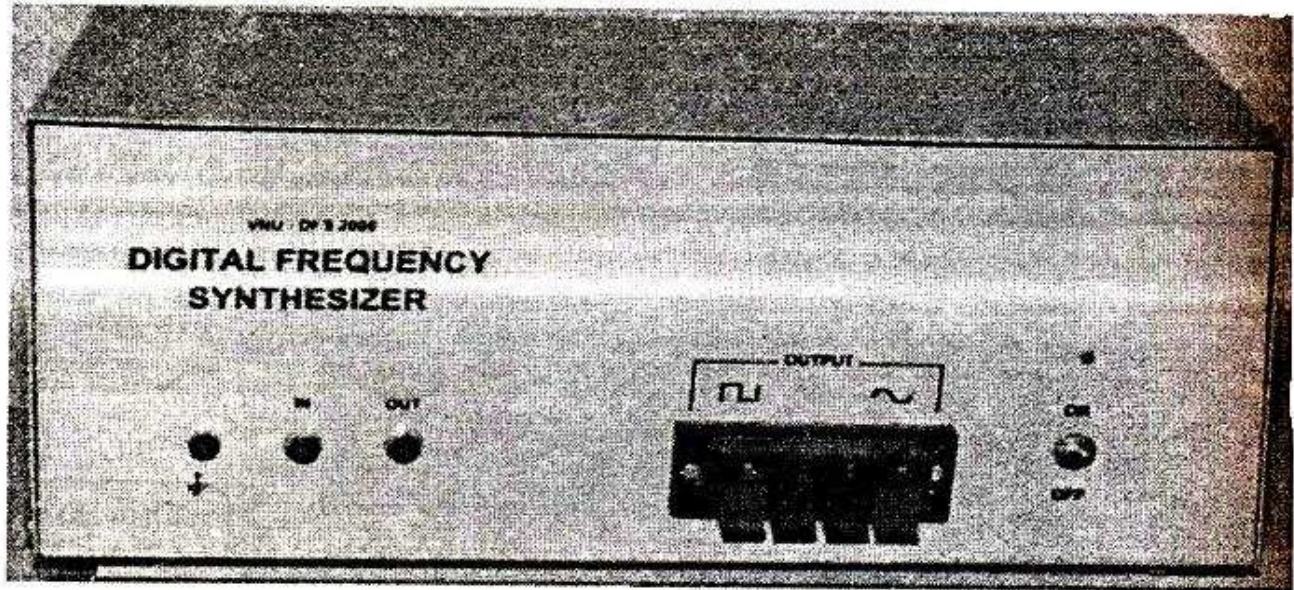


Hình 9.24

Thiết bị đã có tính năng tương tự như một máy phát chức năng dùng trong kỹ thuật vô tuyến điện tử, có khả năng điều khiển thay đổi biên độ, tần số, chọn dạng tín hiệu, đặc biệt còn có thể chọn chế độ tự động quét tần số, chức năng này rất cần trong các hệ tự động đo đặc trưng tần số của các môđun chức năng trong các hệ điện tử. Với phần mềm mô phỏng, ta đã tích hợp được cả chức năng hiển thị tín hiệu phát như một máy hiện sóng. Thiết bị này được dùng cho sinh viên, học viên cao học thực hành kỹ thuật điện tử, thực hành lập trình điều khiển các thiết bị điện tử, triển khai ứng dụng các

thành tựu mới trong lĩnh vực công nghệ phần mềm thuộc chuyên ngành điện tử viễn thông. Độ ổn định tần số của máy phát có thể đạt được mức 10^{-6} nếu dùng phương pháp tổ hợp tần số dùng vòng khoá pha.

Chúng tôi đã chế tạo thiết bị điều khiển tổ hợp tần số dùng PLL, thiết bị có cấu trúc theo nguyên lý hoạt động hình 9.14. Trên hình 9.25 là ảnh chụp thiết bị tổ hợp tần số chúng tôi chế tạo .



Hình 9.25. Thiết bị tổ hợp tần số

Chương X

THIẾT KẾ MÔ PHỎNG CÁC MẠCH ĐIỆN TỬ SỐ

CircuitMaker (CM) là một phần mềm ứng dụng máy vi tính phục vụ cho công tác thiết kế vẽ các mạch điện tử tương tự và điện tử số. Chúng ta có thể tiến hành các thí nghiệm mô phỏng nhằm khảo sát nguyên lý hoạt động của các mạch điện, điện tử ở ngay trên sơ đồ nguyên lý được vẽ từ chương trình. Chương trình CircuitMaker là một sở hữu của công ty Microcode Engineering, tính đến thời điểm này CM đã có phiên bản CM.4, CM.6, CM.Professional và CM 2000 với những tính năng mạnh gấp bội. Phần mềm nay có một thư viện dồi dào hơn 6000 linh kiện, thiết bị dùng cho thí nghiệm mô phỏng. Nó như một phòng thí nghiệm kỹ thuật điện tử ảo hiện đại với các máy đo hiện đại như: máy hiện sóng nhiều kênh Scope; máy đo vạn năng Multimeter có thể đo điện áp, dòng điện một chiều DC và điện áp xoay chiều AC; máy phân tích phổ tín hiệu, đo nhiễu ồn. Chúng tôi chọn phiên bản CircuitMaker.Pro để giới thiệu trong giáo trình này vì nó có các tính năng vượt trội thích hợp cho công tác thiết kế chuyên nghiệp, các thanh công cụ tường minh dễ sử dụng, giao diện của nó giống với phần mềm công ty Microcode Engineering cung cấp miễn phí cho sinh viên (CircuitMaker for Student).

Sinh viên có thể tải phần mềm CircuitMaker for Student này ở trên mạng địa chỉ website: www.microcode.com

Yêu cầu về phần cứng và phần mềm máy tính để cài đặt chương trình:

- Máy vi tính IBM cấu hình 486 hoặc PC cấu hình cao hơn với một ổ cứng có dung lượng còn tối thiểu 15MB còn trống (đề nghị là 20MB), chuột và màn hình VGA hoặc SVGA.

- Đối với các máy tính mới có tốc độ cao, việc mô phỏng diễn ra nhanh hơn.

Đối tượng sử dụng:

Học sinh các trường trung học dạy nghề, học viên các trung tâm dạy nghề, sinh viên các trường đại học kỹ thuật và những người chơi vô tuyến điện tử nghiệp dư.

Một vài điểm cần lưu ý:

CircuitMaker chỉ là một chương trình mô phỏng gần như thật chứ không phải là kết quả cuối cùng. Bởi ở đây, sản phẩm mà chúng ta thiết kế hoàn toàn trên máy tính, chúng chỉ hoạt động tốt trong việc xử lý tín hiệu trong môi trường hoàn toàn số hoá. Còn thực tế, khi thiết kế mô phỏng các mạch điện tử tương tự chỉ gần đúng, mặc dầu công ty Microcode Engineering đã cung cấp một sản phẩm có nhiều ứng dụng đa dạng, nhưng không thể đưa ra được hết những kết quả thoả mãn hoàn toàn với tất cả mọi phân ứng dụng. Chúng ta cần phải kiểm chứng các kết quả thiết kế trong thực tế.

Trong chương này chúng tôi giới thiệu sơ lược về CM, hướng dẫn sử dụng CM, thiết kế mạch điện tử, thí nghiệm mô phỏng mạch điện tử. Phần cuối chương nêu một số bài thí nghiệm mô phỏng các mạch điện tử dùng phần mềm CM.Pro.

10.1. GIỚI THIỆU SƠ LƯỢC VỀ CHƯƠNG TRÌNH CIRCUITMAKER

10.1.1. Các tính năng vẽ sơ đồ mạch điện tử

a/ Tính năng chung

Trình duyệt linh kiện theo hình đồ hoạ.

Tính năng tìm kiếm linh kiện theo từ khoá.

Khả năng thiết lập các phím nóng để đặt vị trí linh kiện.

Bộ soạn thảo biểu tượng để thiết kế các biểu tượng theo ý muốn.

Tính năng Macro cho các mạch phân cấp thiết kế tổ hợp các mô đun.

Các khung viền đơn và nhiều trang.

Khối tiêu đề thay đổi được.

Lưới vị trí và tính năng dịch chuyển ngăn linh kiện.

Đặt tên chân linh kiện theo số thứ tự, các chân thẳng hàng dọc.

Viết tên chân theo chiều dọc và chiều ngang.

Đặt văn bản ở mọi nơi trên bản vẽ.

b/ Tính năng nối dây

Khả năng định tuyến nối dây tự động/nhân công kết hợp trong chương trình.
Tính năng nối dây một cách thông minh SmartWire đảm bảo kết nối dây chính xác.

Hỗ trợ dây kiến Bus.

Khả năng dây nối tự co giãn khi di chuyển các phần của dây.

Đặt lại vị trí dây bằng cách nhấn chuột, giữ và di sang chỗ mới.

Có thể cắt hoặc kéo dài dây.

10.1.2. Mô phỏng và phân tích

a/ Tính năng chung

Bộ mô phỏng 32 bit trên nền SPICE 3f5/XSPICE.

Mô phỏng hỗn hợp tương tự/số.

Mode mô phỏng logic số hoàn toàn tương hỗ.

Thư viện mô hình linh kiện toàn diện.

Khả năng nhập cảng các mô hình SPICE để tạo ra các linh kiện mới.

Hỗ trợ các mô hình BSIM 1, 2, 3, MOS 6.

Ngôn ngữ phát triển SimCode để tạo ra các mô hình digital tùy biến.

b/ Phân tích

Phân tích một chiều DC.

Phân tích xoay chiều AC.

Điểm hoạt động một chiều.

Quá độ.

Quét tham số (tới 2 biến).

Phân tích Fourier.

Đồ thị trở kháng.

Quét nhiệt độ.

Mô phỏng Monte Carlo.

Trường hợp điểm đỉnh.

Hàm truyền.

Tạp âm.

10.1.3. Các tùy chọn nhập, xuất và in ấn

Nhập mạch in dạng SPICE để mô phỏng.

Kết xuất mạch điện và dạng sóng như các hình đồ họa.

Danh sách tài nguyên dữ liệu về linh kiện tùy biến được.

Tạo ra mạch in SPICE3.

In ra mọi máy in/máy vẽ lựa chọn được trong Windows.

Thang tỉ lệ in điều chỉnh được 10% đến 1000%.

Tự động điều chỉnh kích thước bản vẽ phù hợp với màn hình.

Đường chia trang dịch chuyển được cho phép điều chỉnh tỉ lệ in dễ dàng.

10.1.4. Kết xuất mạch in PCB

Khuôn dạng TraxMaker.

Khuôn dạng Protel.

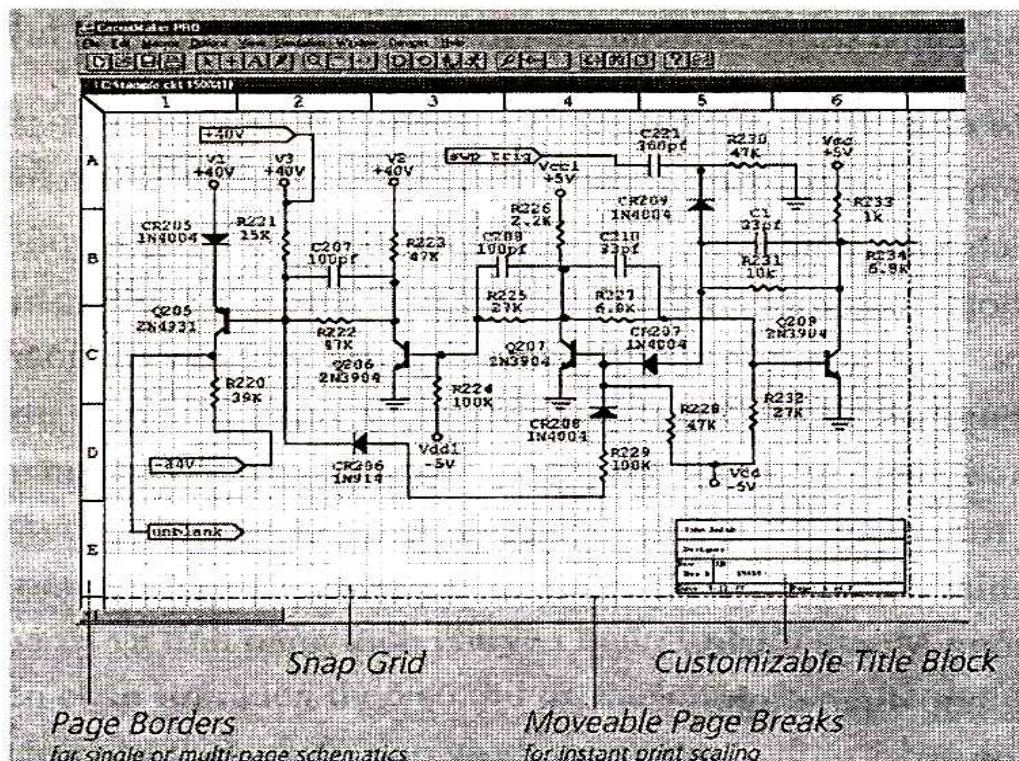
Khuôn dạng Tango.

Khuôn dạng OrCAD PCBII.

Khuôn dạng PADS PCB.

Khuôn dạng Cadnetix .

Khuôn dạng Calay & Calay 90.



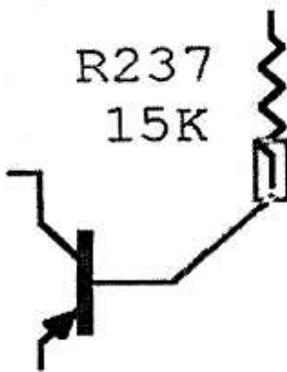
Hình 10.1: Bản vẽ sơ đồ mạch điện tử dùng phần mềm CircuitMaker

10.1.5. Chọn linh kiện để thiết kế mạch

Trên hình 10.1 nêu sơ đồ mạch điện tử dùng phần mềm CM. Tạo ra các bản vẽ chất lượng cao trong CircuitMaker chỉ đơn giản là chọn các linh kiện và đặt chúng lên vùng làm việc. Duyệt qua hàng nghìn linh kiện và mô hình của CircuitMaker bằng **trình duyệt linh kiện theo đồ họa**, hoặc sử dụng tính năng **tìm kiếm linh kiện** để tìm kiếm theo các từ khoá hoặc số.

Đối với các linh kiện hay dùng nhất, bạn có thể đặt các **phím nóng** để đặt chúng vào vị trí chỉ với một lần bấm phím.

Nối chúng lại với nhau. Chỉ cần nhấp chuột vào một chân linh kiện hoặc dây nào đó với công cụ nối dây, giữ và kéo chuột đến một chân linh kiện



hoặc dây khác và CircuitMaker sẽ ngay lập tức nối dây cho bạn. Tính năng **SmartWire** (nối dây thông minh) hiện sáng điểm sẽ được nối trước khi bạn nhả chuột, đảm bảo đặt dây chính xác tuyệt đối. Bạn cũng có thể đặt dây thủ công tại mọi nơi trên vùng làm việc. Không giống các chương trình vẽ mạch khác, CircuitMaker cho phép bạn toàn quyền điều khiển vị trí đặt dây.

Điểm kết nối được tự động hiện sáng để nối dây dễ dàng.

Để tiện lợi hơn nữa, các dây sẽ tự động kéo giãn khi bạn di chuyển linh kiện, và các dây có thể bị cắt, kéo dài hoặc dịch chuyển tùy thích.

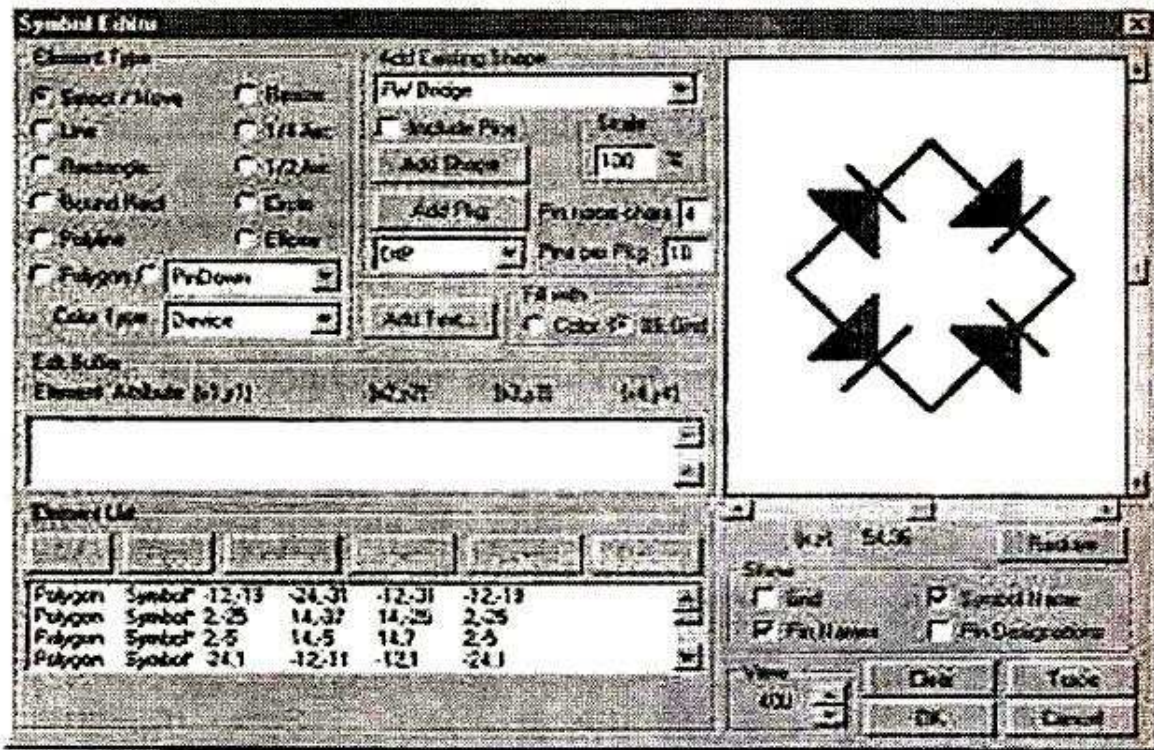
CircuitMaker cũng hỗ trợ các **dây dạng bus** và các dây "đứt nét". Các đầu bus chứa nhiều dây con riêng biệt, và mỗi dây bus được đánh số cho cả bus và từng dây con trong bus. Sử dụng dây bus giúp làm bản vẽ gọn gàng hơn. Dây "đứt nét" là lý tưởng để làm nổi bật các khối logic hoặc phần của mạch điện để làm tài liệu.

Điều khiển

CircuitMaker cho bạn toàn quyền điều khiển hình ảnh bản vẽ của mình. Dùng các tùy chọn khung viền, khối tiêu đề tùy biến được và tính năng đánh dấu chân ưu việt để tạo ra các bản vẽ chất lượng để in ấn, duyệt thiết kế, trình bày hoặc lưu trữ.

Mở rộng thư viện

Bộ soạn thảo biểu tượng của CircuitMaker cho bạn tính linh hoạt để tạo ra các biểu tượng theo ý muốn và thêm chúng vào thư viện. Không có giới hạn số lượng chân nên bạn có thể tạo ra biểu tượng cho các bộ vi xử lý và các thiết bị phức tạp khác. Thiết kế biểu tượng mới của bạn dựa trên một biểu tượng sẵn có, hoặc thiết kế từ nhiều mẫu nhỏ ban đầu.

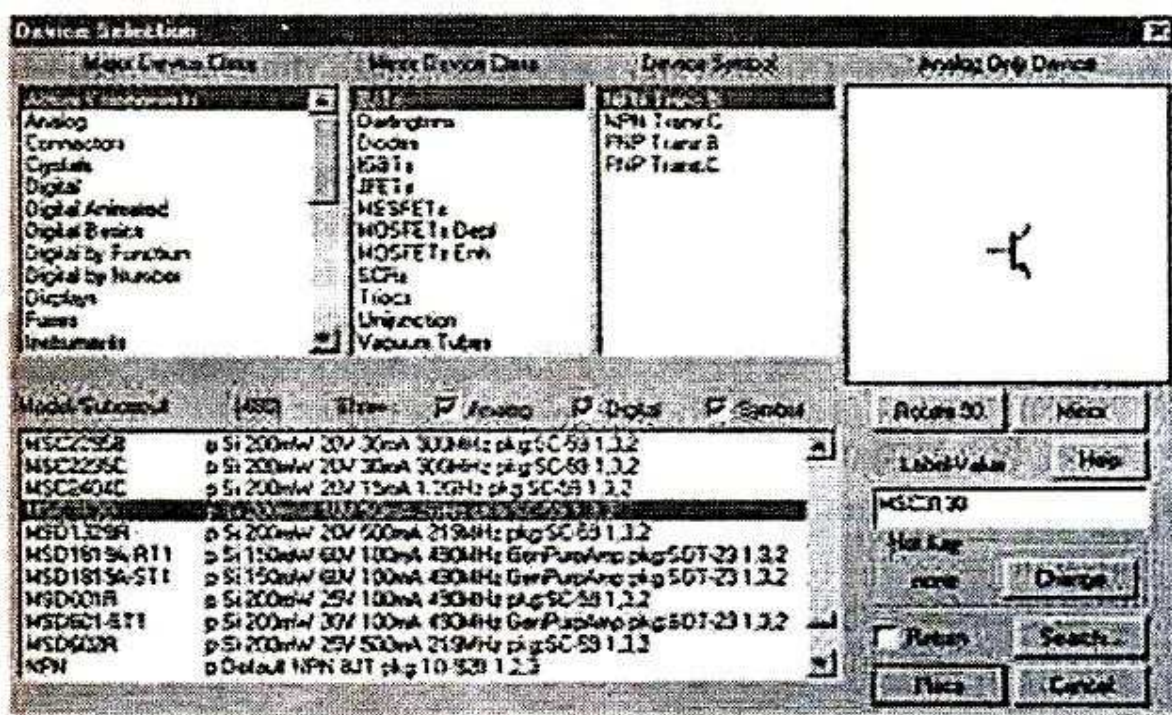


Hình 10.2: Giao diện màn hình soạn thảo biểu tượng

Dùng bộ soạn thảo biểu tượng của CircuitMaker để tạo ra các biểu tượng cho các linh kiện mới.

Dùng một biểu tượng đã có hoặc biểu tượng mới, bạn có thể tạo ra các linh kiện theo ý muốn *có thể mô phỏng được*. Bạn có thể đổi các tham số của một linh kiện đang tồn tại và lưu nó như linh kiện mới, hoặc dùng tính năng Macro để tạo ra một mạch con mới của một linh kiện và thêm nó vào thư viện.

Trình duyệt linh kiện theo đồ họa cho phép bạn duyệt nhanh qua hàng nghìn linh kiện có trong thư viện của CircuitMaker.



Hình 10.3: Giao diện màn hình chọn linh kiện.

CircuitMaker cũng trực tiếp nhập các mô hình SPICE2 và SPICE3 (các mô hình này thường được các nhà sản xuất cung cấp) để tạo ra các linh kiện hoạt động mới.

Nhanh chóng soạn thảo các linh kiện

Trong CircuitMaker, bạn chỉ cần nhấp đúp vào một linh kiện nào đó để sửa chữa rất nhiều tham số. Bạn có thể chọn các mô hình, hoặc sửa chữa các giá trị của linh kiện, nhãn và tên gọi, dữ liệu SPICE và rất nhiều khía cạnh khác của linh kiện.

In và kết xuất thiết kế của bạn

Vì CircuitMaker sử dụng đồ họa dựa trên vector, các bản vẽ mạch điện sẽ được in ra rất sạch và sắc nét với mọi kích thước. Bạn cũng có thể Copy và dán (Paste) các bản vẽ và dạng sóng vào các chương trình khác để chế bản tài liệu.

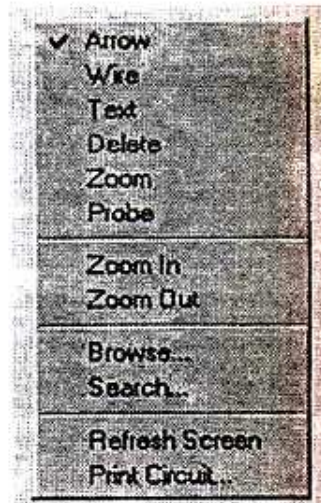
CircuitMaker cũng sẽ tạo ra một danh sách tài nguyên giúp bạn thiết kế.

PCB Layout

CircuitMaker được liên kết chặt chẽ với TraxMaker, nên bạn có thể chuyển tự động mạch thiết kế của mình thành dạng mạch bản mạch in PCB chỉ cần nhấn một nút. CircuitMaker cũng làm việc được với hầu hết các chương trình có PCB layout, và đưa ra mạch in PCB dưới khuôn dạng TraxMaker, Protel và Tango. CircuitMaker PRO cũng đưa ra mạch in PCB dưới khuôn dạng OrCAD PCB II, PADS PCB, Cadnetix, Calay và Calay 90.

Các tính năng vẽ sơ đồ mạch khác

- Lưới gắn linh kiện và dịch ngắn bằng phím mũi tên
- Quay linh kiện mỗi lần 90 độ
- Quay, tạo đối xứng gương và lặp lại chọn linh kiện
- Undo, Redo, tạo bản sao và ghép mạch.
- Phóng to nhỏ 10% - 1000%
- Màu do người dùng tạo ra.
- Văn bản nhiều dòng chấp nhận nhiều mẫu chữ.
- Đánh số chân; chân thẳng hàng dọc; viết tên thẳng đứng hoặc nằm ngang
- Tính năng tự động co giãn khớp mạch vào cửa sổ.
- Kích và rê chuột để đặt lại vị trí dây.
- Vùng kết nối dây do người dùng tự đặt.
- Nhiều dây nối vào một chân linh kiện.
- Cắt và kéo dài dây.
- Trợ giúp trực tuyến theo tình huống.
- Nhấn chuột phải bật trình đơn để nhanh chóng truy cập các tính năng và công cụ soạn thảo.



10.1.6. Mô phỏng các mạch điện tử tương tự và điện tử số

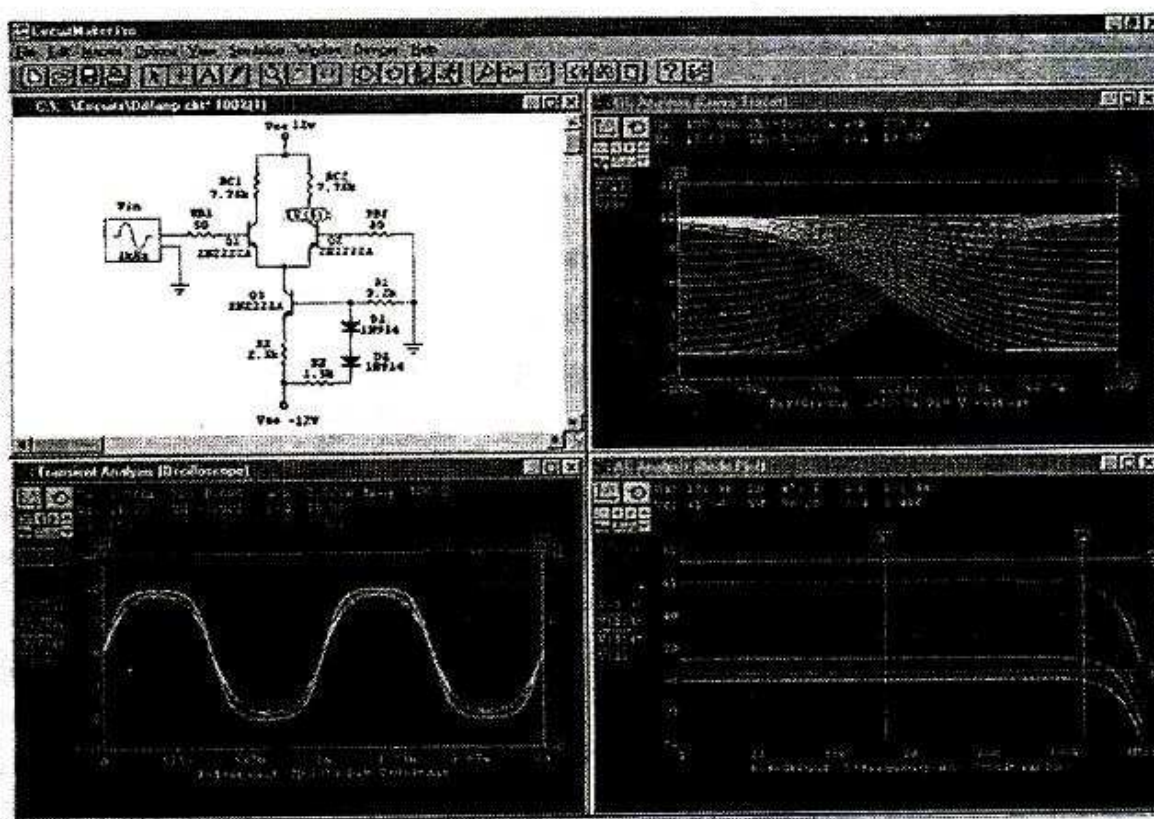
a/ Mô phỏng các mạch điện tử tương tự

Mô phỏng SPICE chính xác

Các mạch của Circuit Maker có khả năng mô phỏng rất nhanh chóng và phân tích các thiết kế của bạn trong cùng một vùng làm việc. Và bạn không

cần phải khởi tạo netlists, hay xác định trước các điểm kiểm tra để chạy mô phỏng. Chỉ cần nhấn phím **run**

Circuit Maker sử dụng SPICE chính xác và được cải tiến (bộ mô phỏng cơ sở đưa ra các kết quả thực tế mà bạn có thể tin được). Trễ đường truyền, thời gian giữ và thiết lập, và nguồn cung cấp - phụ thuộc vào các mức lối ra của mỗi phần tử số là được tạo mô hình một cách chính xác để có thể mô phỏng được các cấu hình thực tế. Circuit Maker kết hợp với SPICE 3f5 và XSPICE mới nhất để đưa ra sự kết hợp chính xác trong mô phỏng tương tự/số. Có nghĩa là bạn có thể mô phỏng bất cứ sự kết nối nào giữa các thiết bị tương tự và số mà không cần phải chèn vào các bộ biến đổi D/A hay A/D.

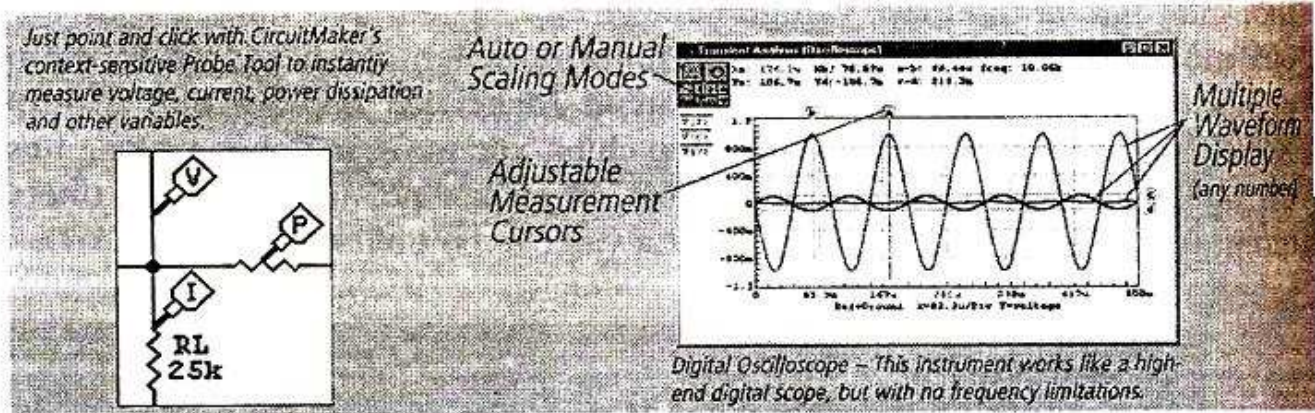


Hình 10.4: Giao diện các màn hình mô phỏng mạch khuếch đại vi sai

Phân tích nhanh hơn, dễ dàng hơn

Ưu điểm lớn nhất mà làm cho Circuit Maker vượt xa các sản phẩm cạnh tranh khác là nó dễ dàng tiến hành việc đo. Đơn giản là chỉ đến và kích chuột với công cụ đầu đo nhạy cảm context - sensitive Probe hiển thị các kết quả tương ứng với các đại lượng vật lý khi mô phỏng. Các kết quả ngay lập tức được hiển thị trên thiết bị được lựa chọn. Nhấn chuột vào các điểm trên

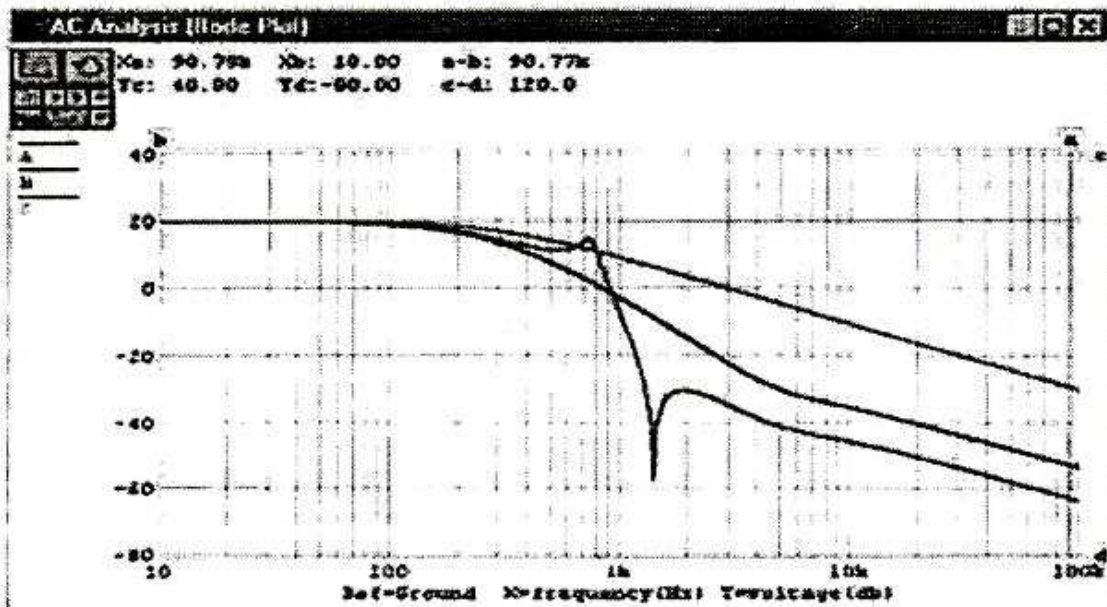
mạch để hiển thị các dạng sóng bất kỳ nào. Hơn thế nữa, bạn không phải thực hiện bằng tay khi vẽ đường hay chỉ ra các điểm cần kiểm tra của bạn. Đây là chương trình hiệu quả nhất cho việc phân tích mạch của bạn một cách nhanh chóng.



Hình 10.5: Giao diện màn hình hiển sóng tại các điểm đo

Một loạt các công cụ trực quan

Circuit Maker đưa ra hàng loạt các công cụ trực quan, cho phép bạn nhanh chóng vẽ các dạng dữ liệu tại bất cứ điểm nào trong mạch. Các công cụ này cung cấp các khả năng tiến bộ như hiển thị bảm và khả năng lưu trữ và gọi ra các dạng sóng.

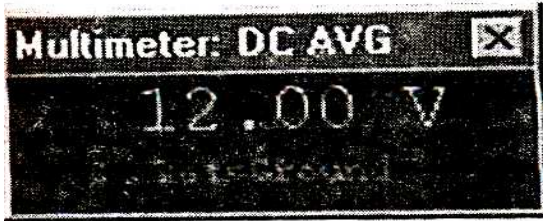


Hình 10.6: Giao diện màn hình đồ thị Bode

Vẽ đồ thị Bode - công cụ này có thể được dùng để phân tích đáp ứng tần số.

Có thể điều khiển bao nhiêu cũng được khi bạn muốn

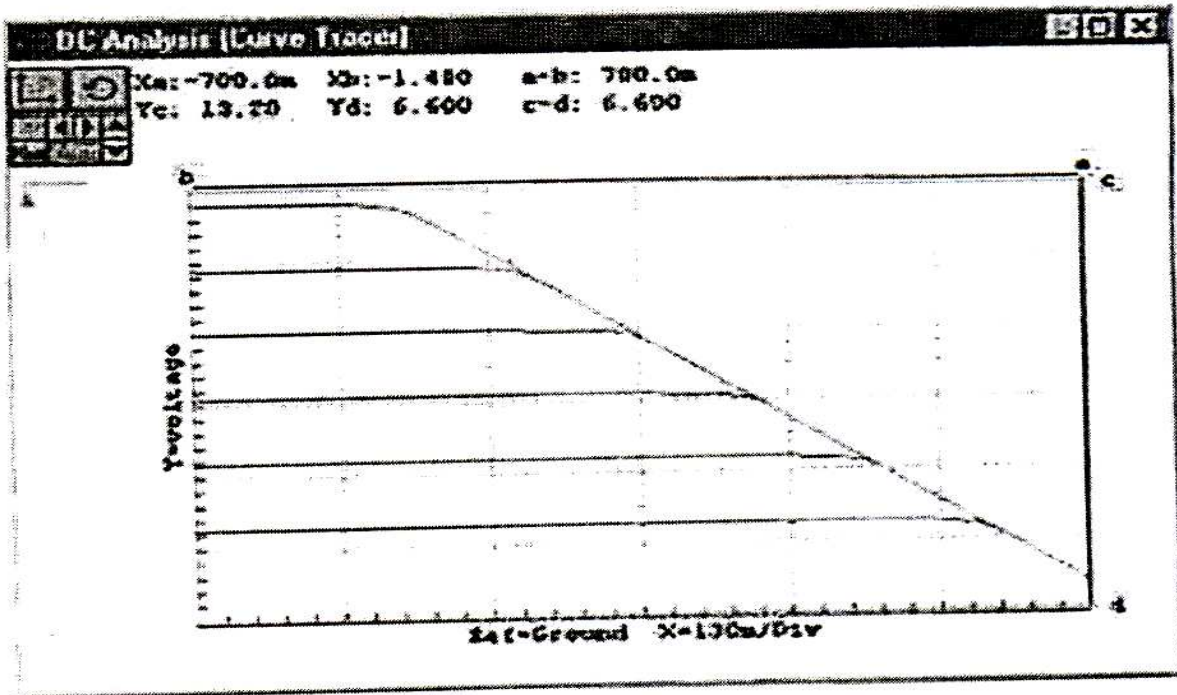
Circuit Maker cho phép bạn điều khiển rất nhiều tham số mô phỏng khi bạn muốn. Bạn có thể điều chỉnh các lựa chọn SPICE, bao gồm: khoảng thời gian, giới hạn lặp và phương thức tích hợp. Bạn cũng có thể quyết định nơi mà dữ liệu được lựa chọn hay các nút được chỉ định.



Hình 10.7: Giao diện màn hình đồng hồ đa năng

Đồng hồ đo số đa năng: nó có thể được sử dụng để đo nhanh chóng điện áp xoay chiều AC, điện áp một chiều DC, đo dòng điện, đo công suất tiêu hao.

Nhưng với những người không muốn sử dụng các kỹ thuật này, Circuit Maker xử lý mọi thứ và sẽ không bắt buộc bạn vì các lệnh SPICE hay các cú pháp phức tạp.



Hình 10.8: Giao diện màn hình Bộ bám đường Trace, công cụ trực quan này có thể dùng để quét toàn bộ các giá trị của nguồn dữ liệu đo

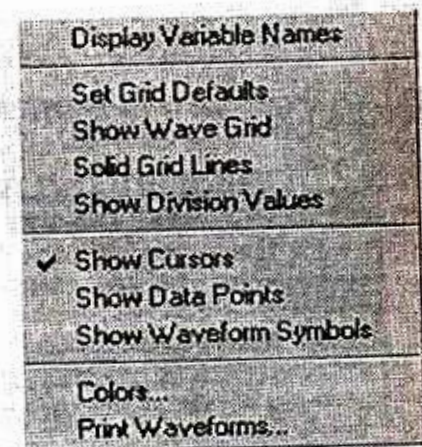
b/ Mô phỏng Logic số

Circuit Maker là một trong các chương trình mô phỏng đưa ra 2 chế độ mô phỏng khác biệt. Chế độ số của Circuit Maker có sẵn khi bạn thuận tụy chỉ muốn mô phỏng logic. Tương tác hoàn toàn, mô phỏng Logic số đáp lại bất cứ thay đổi nào. Mở hay đóng khoá và xem các kết quả khi chúng xảy ra. Các LED và 7 thanh ghi hiển thị ngay lập tức các mức logic thay đổi. Cấu hình bấm đặc biệt hiển thị sáng trạng thái của các đường khi chạy mô phỏng, với sự dàn xếp các tranh chấp nhanh chóng.

Chế độ mô phỏng số của phần mềm Circuit Maker vẫn quan tâm tới trẻ đường truyền, nhưng không yêu cầu nguồn cung cấp và trạng thái logic được hiển thị thay vì các mức điện thế tương tự thực tế.

Các tính năng mô phỏng

- 32 bit SPICE 3f5/XSPICE - bộ mô phỏng cơ sở
- Mô phỏng hỗn hợp tương tự/số.
- Chế độ mô phỏng logic số tương tác hoàn toàn.
- Nhập và mô phỏng netlists SPICE đang tồn tại.
- Các menu Right-click pop up cho việc truy nhập nhanh chóng tới các cửa sổ phân tích (Analysis Window).



Các công cụ trực quan

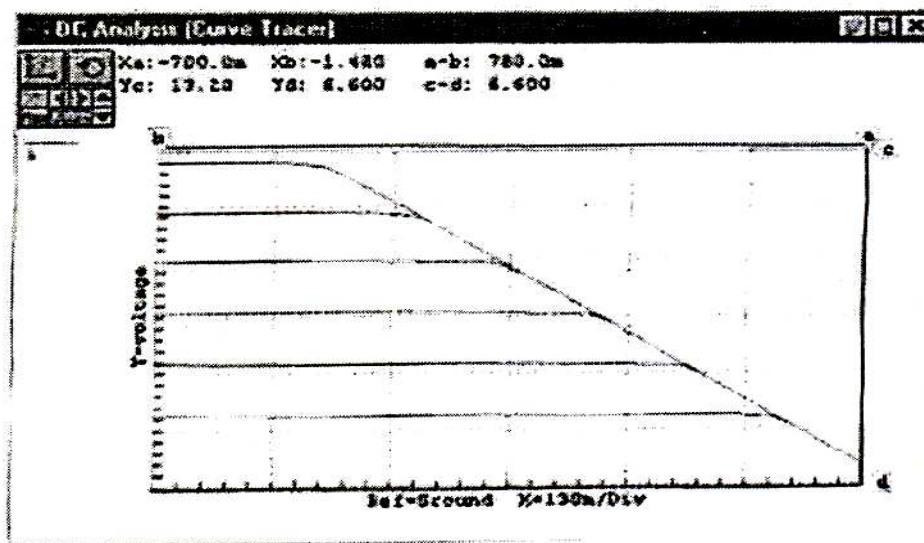
- Bộ Oscilloscope số.
- Bộ bấm đường.
- Bộ vẽ đồ thị Bode.
- Đồng hồ vạn năng số.
- Chuỗi dữ liệu.
- Bộ tạo tín hiệu.
- Bộ phân tích Logic.
- Bộ phát xung/ kiểm tra Logic.

10.1.7. Electronic Lab™ trực quan

Khả năng phân tích cao của Circuit Maker cho phép bạn phân tích các mạch y hệt như việc kiểm tra thực sự. Không những thế, công cụ trực quan Electronic Lab này còn giúp bạn xử lý linh hoạt và tốc độ hơn, cho phép thử các kiểu "what-if", cái gì sẽ xảy ra nếu ta thay đổi các điều kiện thí nghiệm khác nhau? trong một thời gian ngắn ta đã có ngay kết quả giải đáp.

Phân tích một chiều DC (DC sweep)

Bộ phân tích một chiều DC khởi tạo đầu ra giống như một bộ bám đường. Nó thực hiện phân tích điểm hoạt động tại các thời điểm để xác định đường cong đặc trưng một chiều DC. Nguồn sơ cấp và thứ cấp có thể được sử dụng, và bạn có thể xác định các giá trị bắt đầu, kết thúc, và các thời điểm của mỗi nguồn.

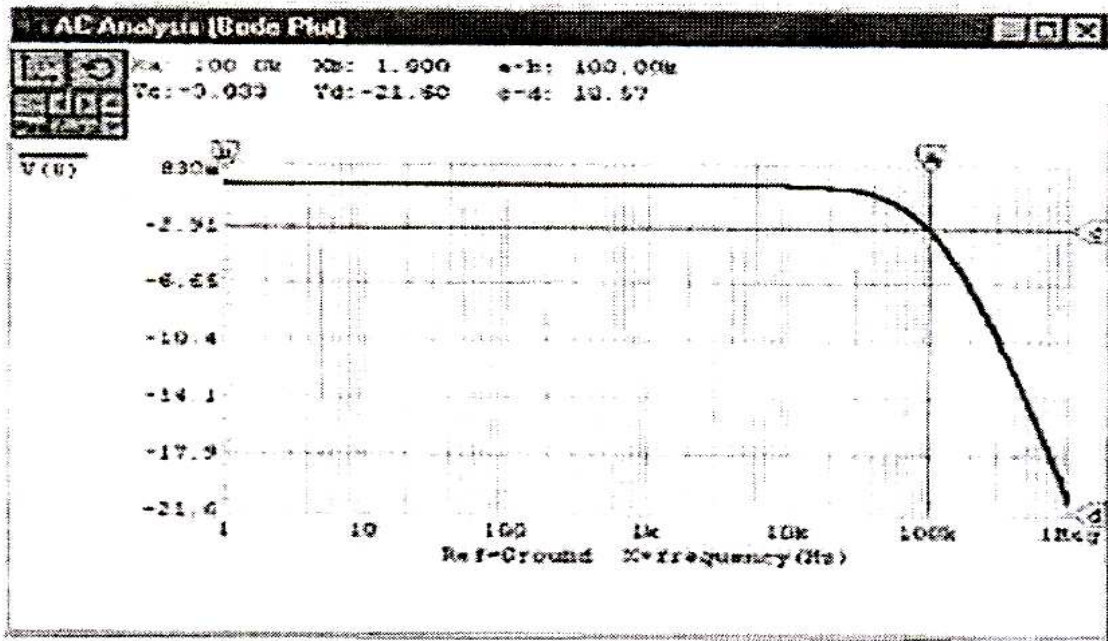


Hình 10.9: Giao diện màn hình Phân tích DC (quét DC)

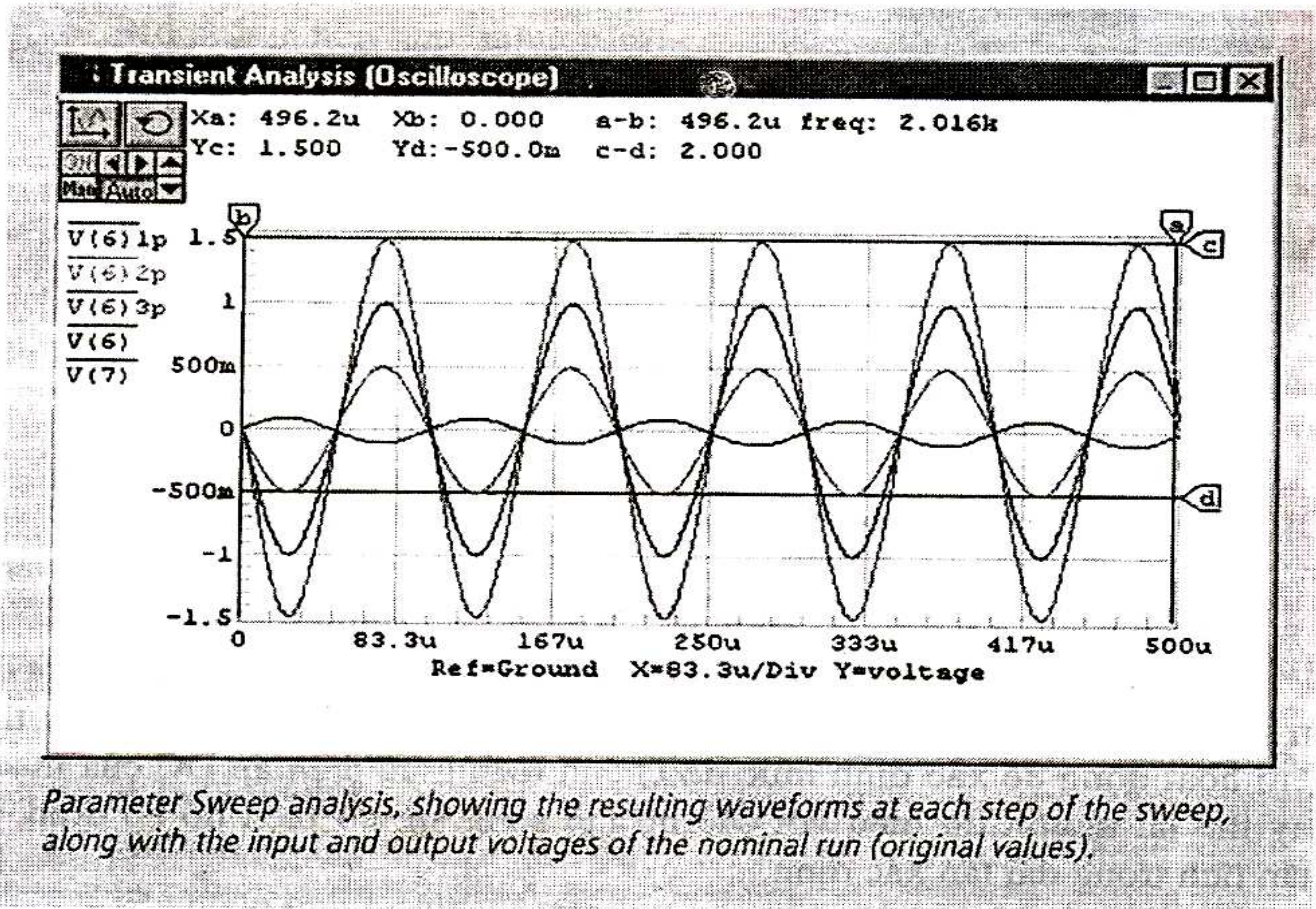
Phân tích xoay chiều AC (AC sweep)

Phân tích AC cho phép bạn đánh giá đáp ứng AC của mạch trên một dải tần số xác định. Lối ra giống như bộ vẽ đồ thị Bode. Nó thực hiện phân tích điểm hoạt động để xác định mức độ chênh lệch mức điện áp DC của mạch, thay thế lại nguồn tín hiệu với các bộ tạo sóng sin có biên độ cố định, và phân tích trong dải tần xác định.

Đường quét được có thể là tuyến tính, decade hay octave, tần số bắt đầu, tần số kết thúc và số lượng các điểm kiểm tra cũng được xác định.

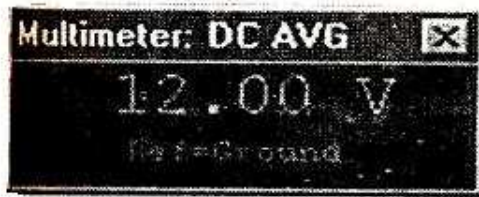


Hình 10.10: Giao diện màn hình Phân tích AC (AC sweep)



Hình 10.11: Giao diện màn hình hiện sóng

Điểm hoạt động DC



Điểm hoạt động DC khởi tạo đầu ra giống như việc đọc đồng hồ DC. Nó xác định mức độ chênh lệch mức điện áp một chiều DC trong toàn mạch với các điện cảm và điện dung, và sử dụng việc tuyến tính hoá, các mẫu tín hiệu nhỏ cho tất cả các thiết bị phi tuyến.

Phân tích quá độ

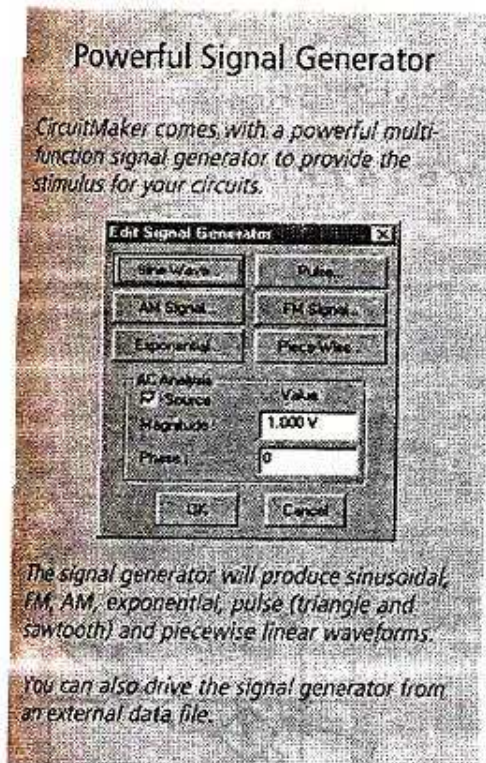
Phân tích quá độ đưa ra biểu đồ giống như của 1 oscilloscope, tính những giá trị quá độ đầu ra (điện áp, dòng, công suất..) trong những khoảng thời gian xác định.

Quét các tham số

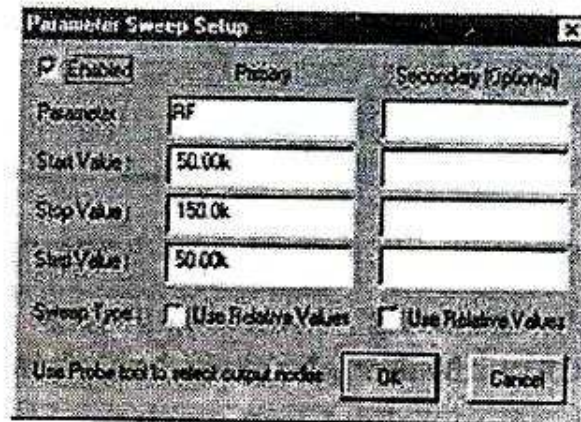
Chức năng quét tham số cho phép bạn quét đưa ra các giá trị linh kiện (như giá trị của điện trở hay các tham số kiểu linh kiện (như hệ số khuếch đại của 1 transistor) trong 1 dải có thể định nghĩa được.

Dùng chức năng quét sơ và thứ cấp cho phép bạn nhanh chóng ước lượng được sự ảnh hưởng lẫn nhau của 2 linh kiện trong 1 dải quét xác định.

Lựa chọn "use relative value" để thêm các giá trị bắt đầu, kết thúc và các bước nhảy cho giá trị chuẩn của các tham số.



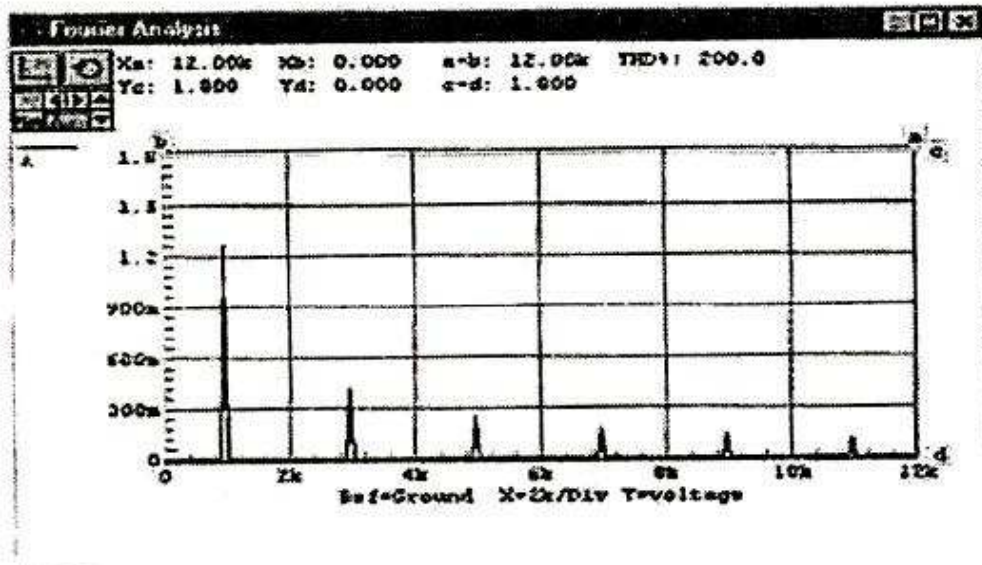
Hình 10.13: Giao diện màn hình đặt các tham số cho máy phát tín hiệu.



Hình 10.12: Giao diện màn hình Hộp hội thoại cài đặt quét tham số

Phân tích Fourier

Phân tích Fourier này dùng cho phổ tần số của bất kỳ dạng sóng nào. Bằng cách nhấn vào các nút với công cụ thăm dò để vẽ bản đồ phổ tần số của điểm đó. Bạn có thể định nghĩa tần số gốc cũng như số hàm điều hòa trong phân tích này.



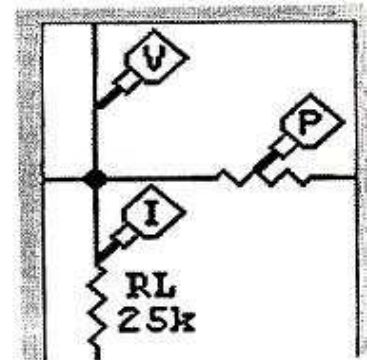
Hình 10.14: Giao diện màn hình Phân tích Fourier

Những phân tích chuẩn trong CircuitMaker 6 Và PRO

- Phân tích DC.
- Phân tích AC.
- Điểm hoạt động DC.
- Nhanh.
- Quét tham số.
- Fourier.

Tìm kiếm "không có phiền nhiễu" nhanh

Không giống các chương trình mô phỏng khác, CircuitMaker cho phép bạn nhanh chóng đo và xem xét mọi điểm trên mạch mà không bắt bạn phải định nghĩa các điểm test hay các dụng cụ đường dây trong mạch.



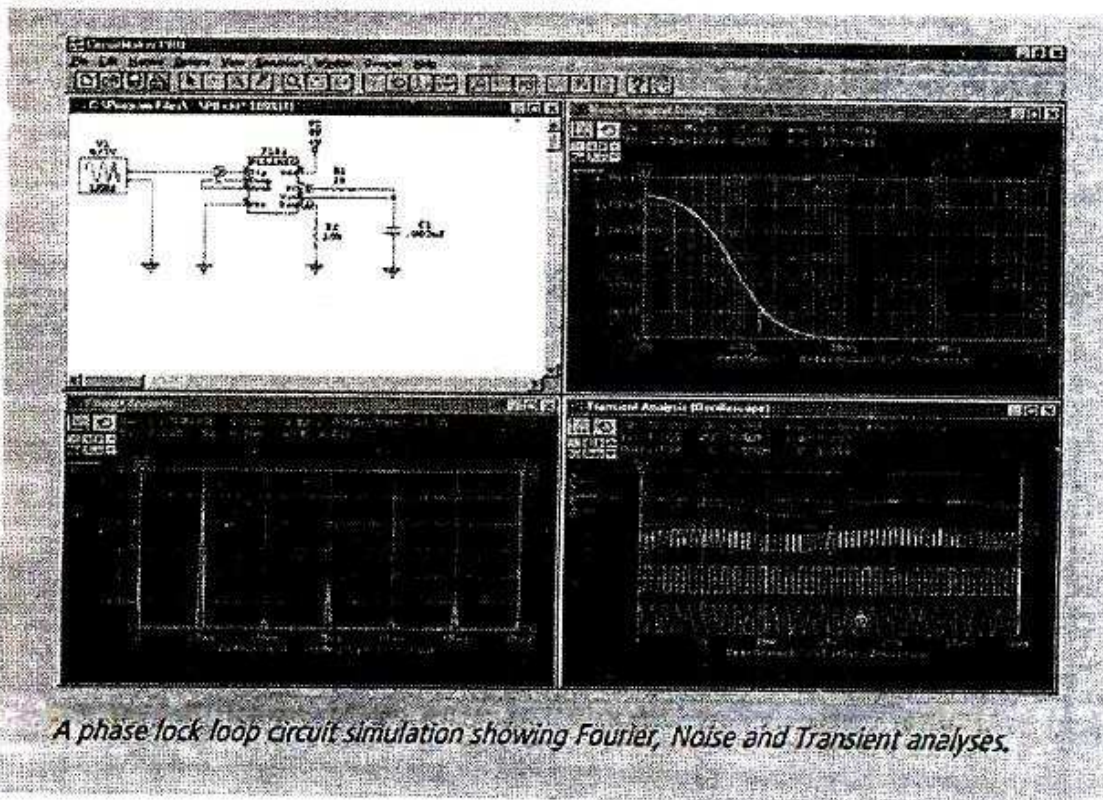
Trong quá trình mô phỏng chỉ cần nhấn vào mọi nơi trên mạch với dụng cụ thăm dò, kết quả sẽ xuất hiện lập tức trên công cụ lựa chọn hoặc nơi phân tích.

Giữ phím Shift khi nhấn và bạn sẽ vẽ được biểu đồ mô phỏng dạng sóng.

CircuitMaker PRO

Cùng với các phân tích chuẩn, CircuitMaker PRO cũng cung cấp các phân tích khác cho các người sử dụng cao cấp.

Dải phân tích rộng của CircuitMaker cho phép phân tích hoạt động của mạch như bạn đang làm thí nghiệm thật. Nhưng "Phòng thí nghiệm ảo" này cho các tốc độ và sự mềm dẻo cao hơn cho phép bạn thử rất nhiều sự kiện tương lai.

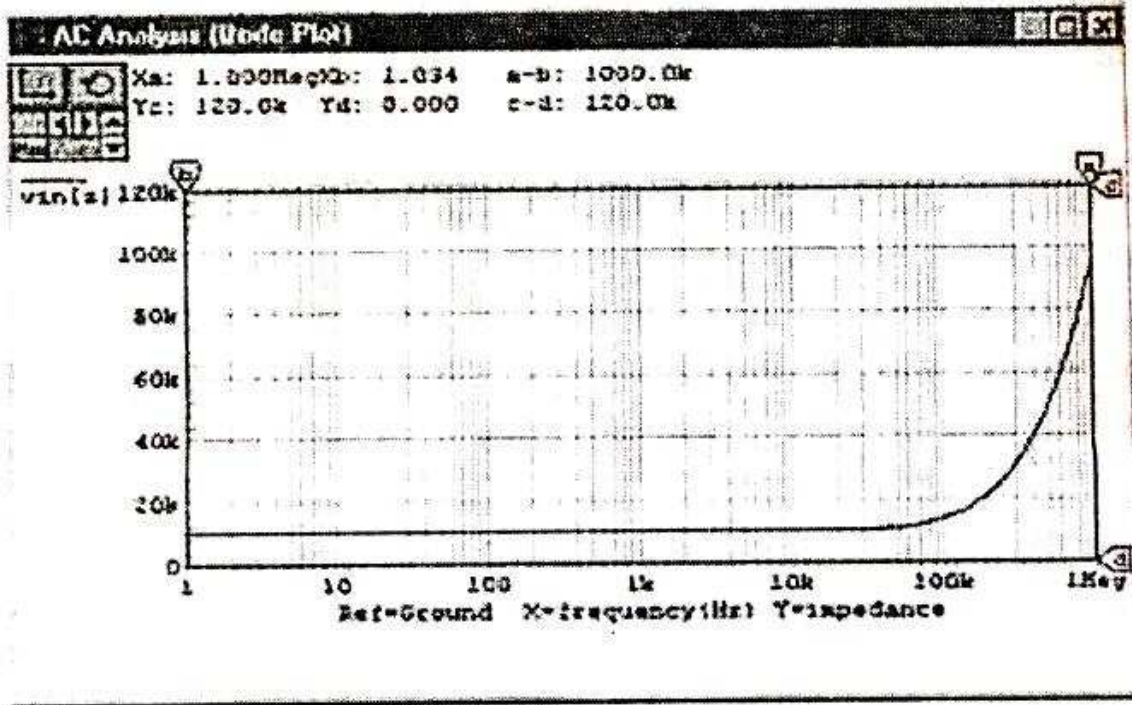


Hình 10.14b: Giao diện màn hình phân tích hoạt động của PLL

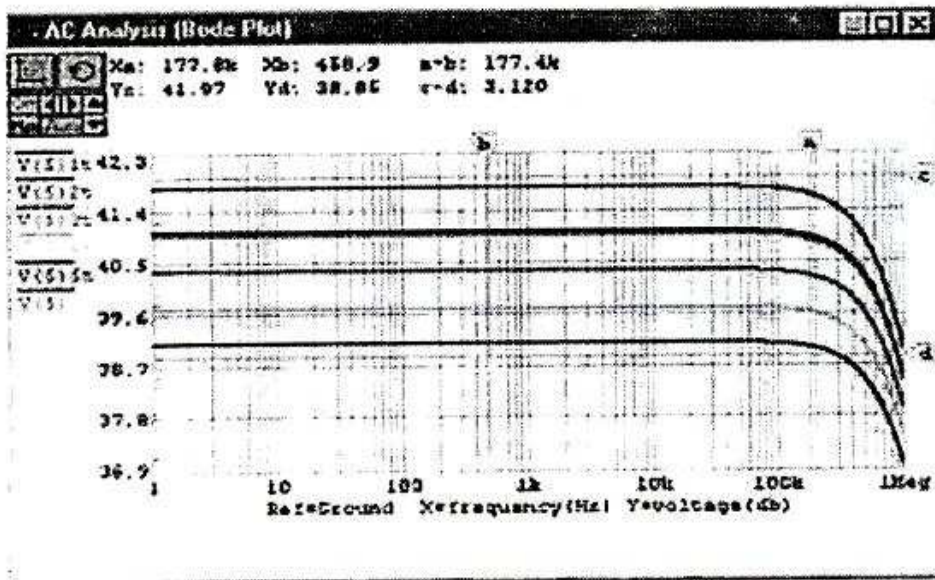
"Cái gì xảy ra – Nếu ta thay đổi các dữ kiện thí nghiệm" trong khoảng thời gian ngắn.

Đồ thị trở kháng

Phân tích phần trở kháng của CircuitMaker cho giá trị trở kháng chịu bởi bất kỳ 1 nguồn 2 cực nào.



Hình 10.15: Giao diện màn hình Đồ thị trở kháng



Hình 10.16: Giao diện màn hình Phân tích quét nhiệt độ

Trở kháng được tính bằng cách chia điện áp ở cực dương được cung cấp với dòng đi ra cực này.

Quét nhiệt độ

Phân tích quét nhiệt độ cho phép vẽ khảo sát nhiệt độ của mạch ở dải mà người dùng được định nghĩa.

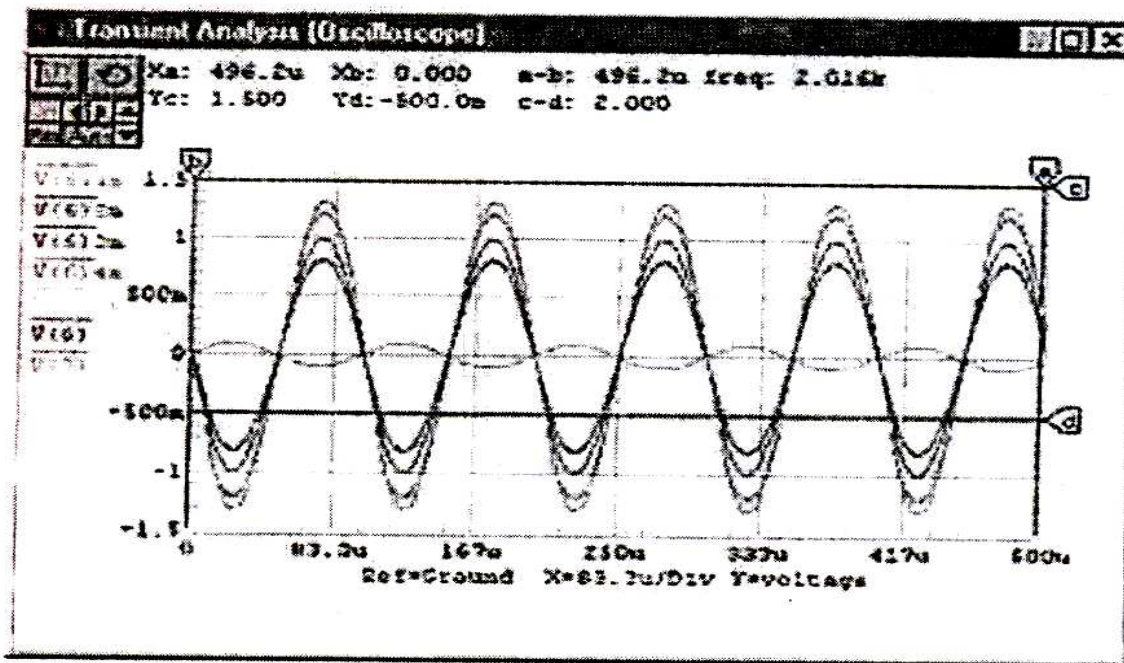
Những dạng sóng cho mỗi bước này được vẽ lần lượt trên các công cụ được lựa chọn.

Monte Carlo

Phân tích Monte Carlo này chạy nhiều mô phỏng với những giá trị của thiết bị ngẫu nhiên dựa trên các dung sai xác định.

Bạn có thể chọn số của mô phỏng cần chạy, kiểu phân bố (đều, Gaussier, trường hợp điểm đỉnh) và các dung sai chuẩn cho 6.

Loại chung của thiết bị. Bạn cũng có thể thay đổi các dung sai chuẩn này bằng cách gán bất kỳ số của dung sai riêng cho các thiết bị đặc biệt trong mạch.



Hình 10.17: Giao diện màn hình Phân tích Monte Carlo

Trường hợp điểm đỉnh

Phân tích Trường hợp điểm đỉnh nằm trong phân tích Monte Carlo. Lựa chọn này sẽ phân bố 1 cách ngẫu nhiên các giá trị tại các trường hợp điểm đỉnh của chúng (cực tiểu hoặc cực đại) của dải dựa trên các dung sai xác định.

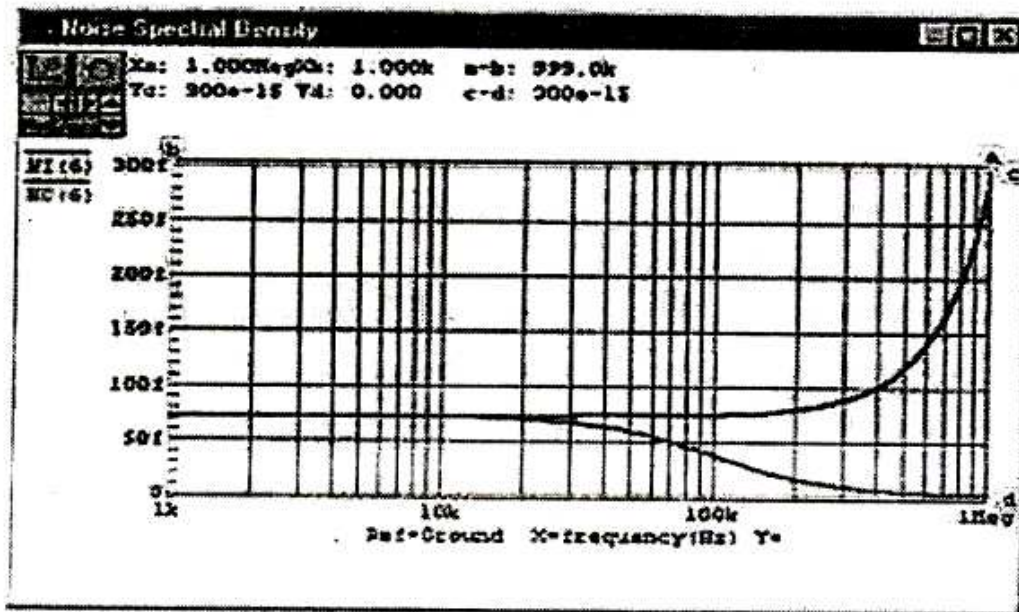
Hàm truyền

Phân tích hàm truyền tính toán trở kháng DC đầu vào, trở kháng DC ra, khuếch đại DC của 1 mạch. Bạn có thể nhấn vào bất kỳ 1 điểm nào trên mạch và ngay lập tức bạn sẽ thấy hàm truyền giữa đầu vào và điểm đó.

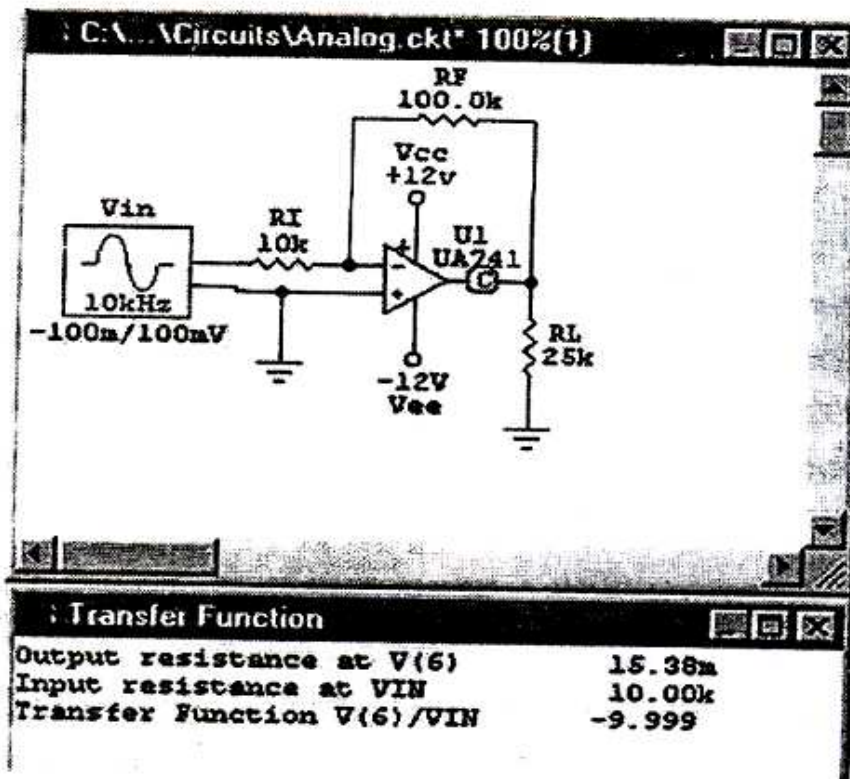
Không giống các mô phỏng khác, bạn không phải chạy 1 mô phỏng riêng cho mỗi nút. CM cho phép bạn ước lượng chức năng chuyển tải tại nhiều điểm trong một chương trình chạy mô phỏng duy nhất.

Tạp âm

Phân tích tạp âm của CM cho phép bạn đo tạp âm của mạch trên các linh kiện bán dẫn và các điện trở. Các kết quả sẽ được vẽ thành đồ thị như mật độ phổ tạp âm mà đơn vị đo của tạp là V^2/Hz . Các tạp âm vào, ra và tạp âm của linh kiện là có thể đo được.



Hình 10.18: Giao diện màn hình Phân tích nhiễu



Hình 10.19: Giao diện màn hình khảo sát hàm truyền của mạch khuếch đại thuật toán

CircuitMaker PRO: những phân tích và đặc điểm

- Phân trở kháng.
- Quét thăm dò nhiệt độ.
- Monte Carlo.
- Trường hợp điểm đỉnh.
- Tạp âm.
- Hỗ trợ cho kiểu BSIM 1, 2, 3 và MOS 6.
- Ngôn ngữ chương trình Mã Sim số cho các kiểu XSPICE số của khách hàng.

Thư viện mẫu toàn diện có thể mở rộng.

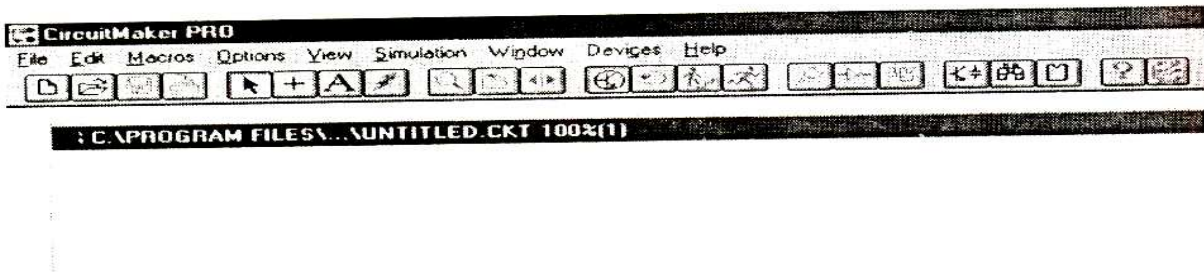
Một trong những thế mạnh lớn nhất của CircuitMaker là có 1 thư viện rất đầy đủ và có thể được cập nhật thường xuyên hàng nghìn loại thiết bị với các thông tin chuẩn xác cao. CircuitMaker PRO chứa hơn 6000 loại linh kiện thiết bị cơ bản và cao cấp. Gồm:

- Các bộ chuyển đổi A/D và D/A.
- PLL, VCO, các bộ chuyển đổi điện áp sang tần số và ngược lại.
- Linh kiện bán dẫn.
- IC số.
- IC tuyến tính.
- Các nguồn cung cấp điện áp.
- Đường truyền v.v...

10.2. HƯỚNG DẪN SỬ DỤNG CIRCUITMAKER

10.2.1. Giao diện sử dụng

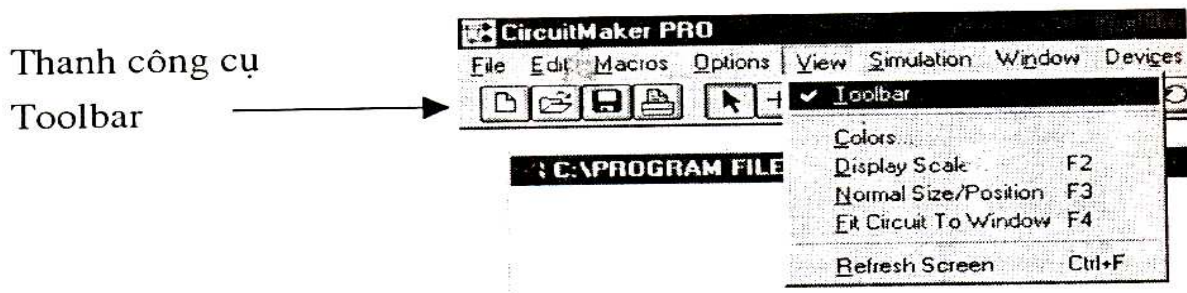
10.2.1.1. Giao diện chính



Hình 10.20: Giao diện màn hình chính

10.2.1.2. Hiện thị - ẩn đi các công cụ làm việc

- Bước 1: Mở trình đơn View\Toolbar. Ẩn đi nếu thanh Toolbar đã hiển thị.
- Bước 2: Mở trình đơn View\Display Toolbar. Hiện thị nếu thanh Toolbar đã ẩn đi.



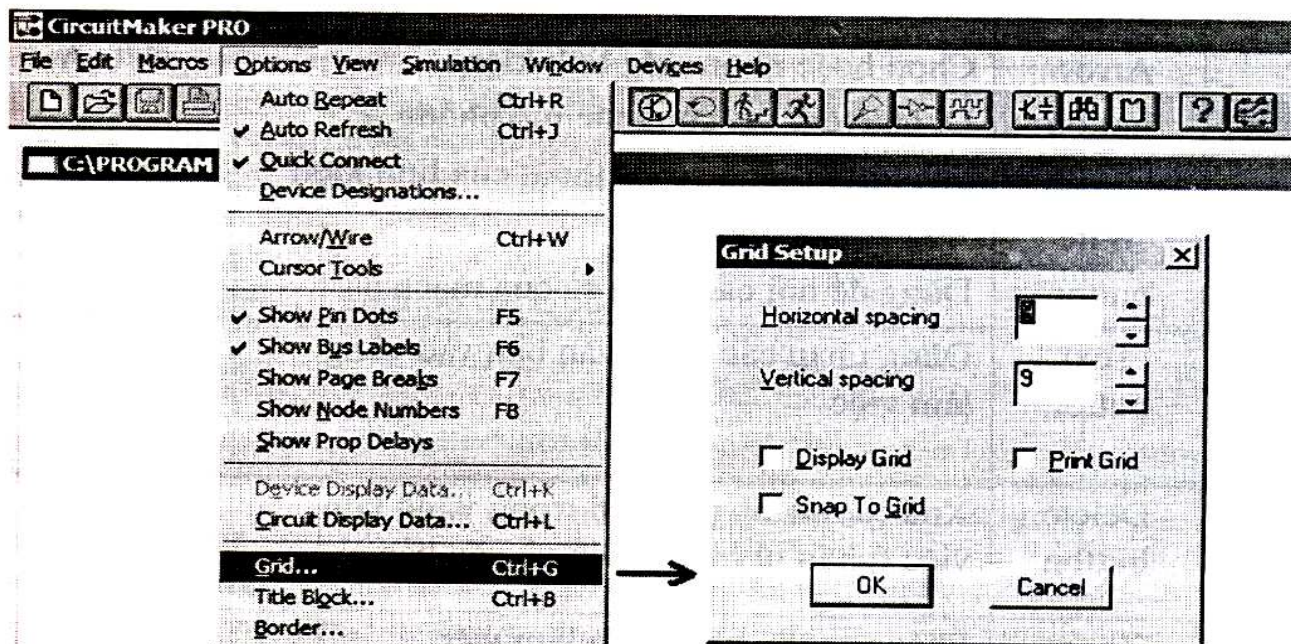
Hình 10.21: Giao diện trình đơn View

10.2.1.3. Lưới vẽ và tọa độ

Làm ẩn hiện một lưới vẽ.

Bước 1: Mở trình đơn Option\Grid

Hoặc gọi phím tắt Ctrl+G. Một hộp thoại xuất hiện như hình 10.22 sau:



Hình 10.22: Giao diện hộp thoại Option

- *Horizontal spacing*

Vùng (field) này cho phép ta nhập trị số chiều ngang của ô lưới.

- *Vertical Spacing*

Vùng (field) này cho phép ta nhập trị số chiều dọc của ô lưới.

- *Display Grid*

Nếu đánh dấu vào chex-box này thì các ô lưới sẽ hiển thị trên vùng làm việc như hình.

- *Print Grid*: Nếu đánh dấu vào chex-box này thì các ô lưới sẽ được in ra trên giấy khi gọi lệnh in sơ đồ.

- *Snap To Grid* : Nếu đánh dấu vào chex-box này thì khi dịch chuyển linh kiện trỏ chuột sẽ dịch theo đơn vị ô lưới do máy ấn định trên vùng làm việc.






Bước 2 :









-Nhập giá trị chiều ngang dọc của các ô lưới vào các vùng thích hợp.

- Đánh dấu vào chex-box Display Grid.

Bước 3 : Nhấn OK.

10.2.1.4. Tìm hiểu công cụ dùng để vẽ mạch điện

Nút	Tên gọi	Chức năng	Phím tắt
	Arrow button	Chọn hoặc nhóm các linh kiện Di chuyển, bật công tắc mô phỏng Thay đổi thông số kỹ thuật của linh kiện	Alt+A
	Wire button	Dùng để nối các phần tử của mạch.	Alt+W
	Text button	Dùng nhập các đoạn văn bản vào vùng làm việc	Alt+T
	Delete button	Xoá các thành phần có trong vùng làm việc (phần tử mạch điện và văn bản).	Alt+D
	Zoom button	Phóng to thu nhỏ vùng làm việc	F3/F4 PGD/PGU
	Part button	Gọi linh kiện từ thư viện chương trình	X
	Rotate 90°	Xoay linh kiện được chọn một góc 90°	Alt+R
	Mirror button	Lật linh kiện theo chiều ngang	Alt+M
	Macro button	Tạo hoặc chỉnh sửa một linh kiện	Ctrl+H

	Digital button	Cho phép thi hành mô phỏng số.	
	Analog button	Cho phép thi hành mô phỏng tương tự.	
	Track button	Xem trạng thái của mỗi dây dẫn khi mạch hoạt động.	F11
	Probe button	Đầu dò tín hiệu.	Alt+P
	Run button	Chạy một mô phỏng.	F10
	Step button	Sử dụng cho mô phỏng số thực hiện từng bước.	
	Stop button	Ngừng chạy mô phỏng.	
	Wave form	Mở một cửa sổ để xem dạng sóng.	

10.2.1.5. Thư viện linh kiện của CircuitMaker

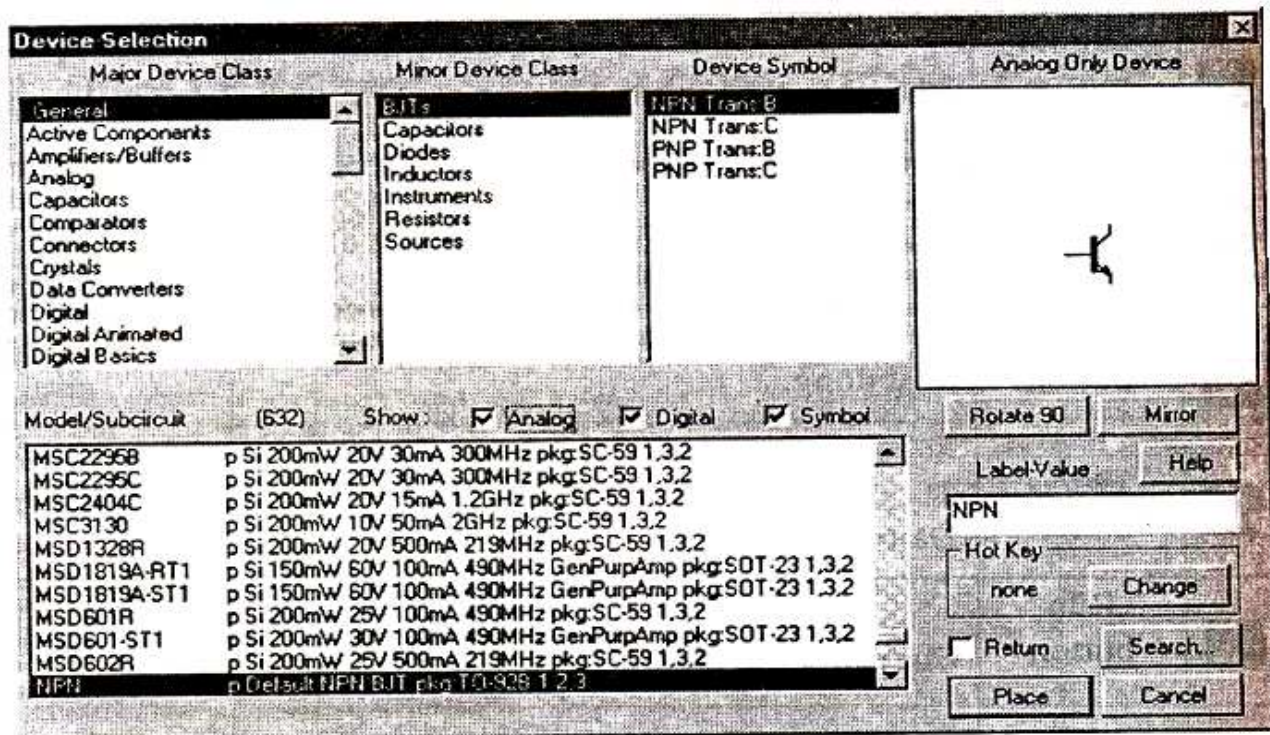
Các linh kiện trình bày trong sơ đồ được biểu diễn bởi những ký hiệu đặc biệt mà chúng ta phải lấy từ một thư viện có trong chương trình. CircuitMaker trong phiên bản PRO này cung cấp cho chúng ta hơn 6000 linh kiện điện tử, các linh kiện bán dẫn vi điện tử các loại .

Gọi một thư viện và lấy một linh kiện đặt trên vùng làm việc

Bước 1: Nhấn vào nút lệnh



Hộp thoại Device Selection hình 10.23 cho phép ta chọn lựa linh kiện mà ta cần dùng.



Hình 10.23: Hộp thoại Device Selection

Ý nghĩa của từng cửa sổ trong hộp thoại Device Selection như sau:

Major Device Class

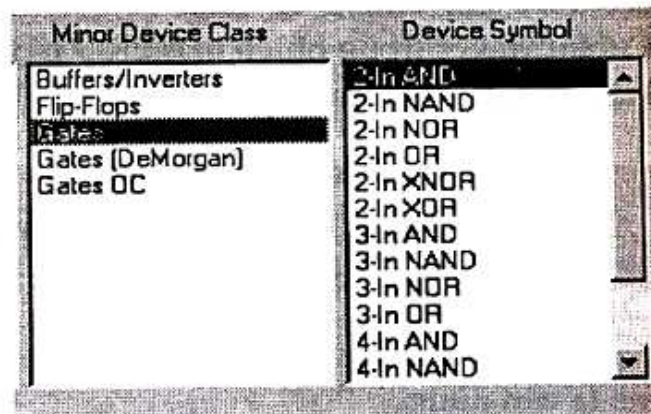
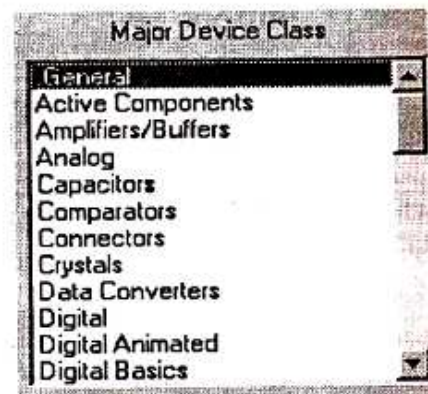
Cho phép ta lựa chọn thành phần chủng loại chính của linh kiện mà ta dùng.

Ví dụ: Ta chọn Digital tức là ta đã chọn các linh kiện thuộc chủng loại là Digital, các chủng loại khác không được đề cập đến.

Minor Device Class

Cho phép ta lựa chọn loại linh kiện mà ta cần dùng.

Ví dụ: Ta chọn Gates tức là chọn các cổng Logic.



Device Symbol

Chọn ra ký hiệu linh kiện.

Ví dụ : 2-In AND

Analog Only Device

Hiện thị cho ta thấy các ký hiệu các linh kiện bán dẫn chỉ dùng trong các mạch điện tử tương tự và tham số đặc trưng của linh kiện mà ta chọn lựa để dùng.



Hình 10.24: Giao diện cửa sổ Model/ Subcircuit

Model/Subcircuit

Vùng này cho ta chọn lựa lấy linh kiện theo mã số ký hiệu linh kiện của nhà sản xuất.

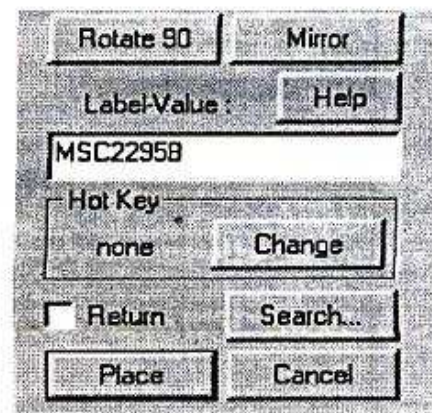
Ví dụ: Như transistor MSC 2295B p Si 200 mW 20V 30 mA 300 MHz pkg SC-59 1,3,2. Có ý nghĩa như sau: Đây là loại transistor NPN có số hiệu là MSC 2295B cấu tạo chất bán dẫn là Si, có công suất cực đại $P_{max} = 200mW$, chịu được điện áp lớn nhất $V_{max} = 20V$ và dòng tải cho phép lớn nhất $I_{max} = 30mA$, hoạt động ở tần số 300 MHz, có hình dạng đóng vỏ mã số: pkg SC-59 1,3,2.

Rotate

Xoay linh kiện một góc 90° .

Mirror

Lật linh kiện theo chiều ngang.



Label-Value: Thiết đặt nhãn gán cho linh kiện.

Hot Key: Thay đổi hoặc định nghĩa phím tắt gọi linh kiện.

Place : Nhấn vào nút này để đặt một linh kiện vào vùng làm việc mà ta vừa chọn trong các mục kể trên.

Return : Thiết lập đặt tự động quay trở lại hộp thoại trên khi đã lấy một linh kiện ra màn hình làm việc.

Các check-box như Show : Analog, Digital, Symbol có ý nghĩa giúp ta duyệt nhanh các linh kiện có hiệu quả.

Ví dụ : Nếu ta chỉ đặt Check-box Analog thì các linh kiện có mặt trên vùng chọn lựa là các linh kiện dùng cho mô phỏng Analog.

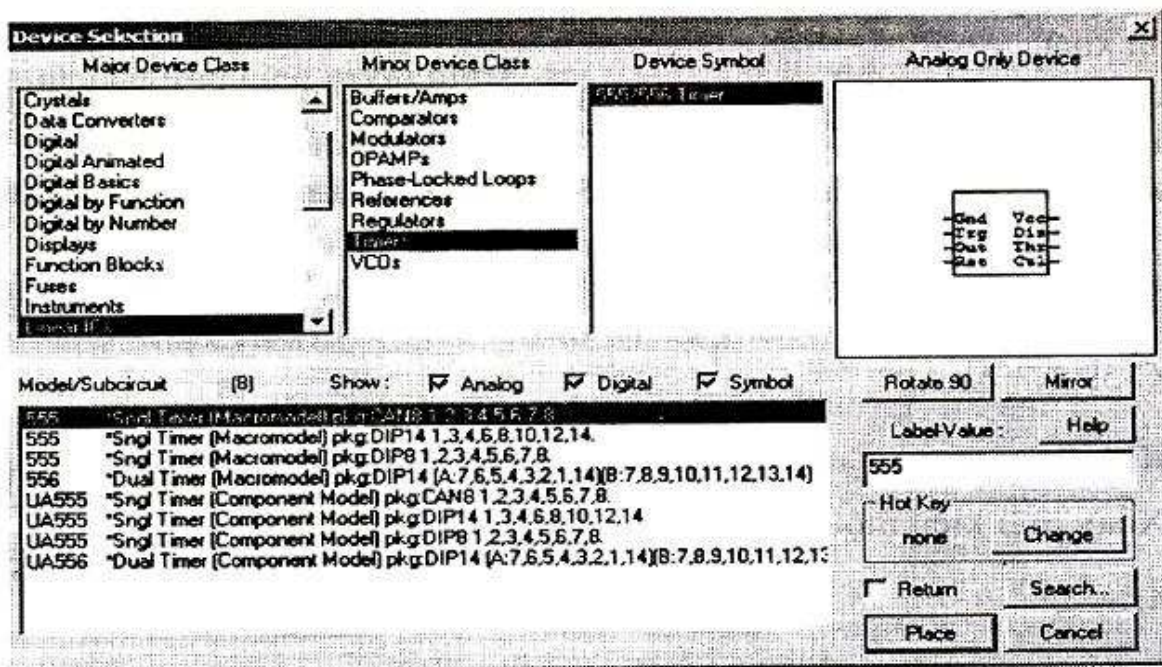
Và tất cả các check-box đều được đánh dấu thì mọi linh kiện từ Analog, Digital đến Symbol sẽ hiển thị trên vùng chọn lựa làm cho ta vất vả trong việc tìm kiếm.

Bước 2: Đặt linh kiện ra vùng làm việc.

Ví dụ: Cần vẽ mạch tạo xung dùng vi mạch vi mạch MC14555.

- Cửa sổ Major Device Class chọn là Linear Ics.
- Cửa sổ Minor Device Class chọn là Times.
- Cửa sổ Device Symbol chọn là 555/556 Time.

Cửa sổ Mode/Subcircuit dùng tùy chọn đầu tiên và nhấn Place.



Kế đến đặt linh kiện lên vùng làm việc và ta click phải chuột để xoay linh kiện theo chiều ta mong muốn trong sơ đồ mạch. Khi đúng chiều như hình vẽ click trái chuột để cố định vị trí của linh kiện.

Lần lượt theo trình tự trên ta có thể lấy ra bất kỳ linh kiện, linh kiện nào mà trong thư viện của CircuitMaker có.

Ví dụ như: Điện trở, tụ điện, các cổng Logic,... nguồn và Mass.

Cách khác: Để lấy các linh kiện như trình bày ở trên chúng ta còn được phép sử dụng phím tắt để gọi nhanh linh kiện ra từ thư viện mà không phải thực hiện nhiều bước như trên. Đương nhiên CircuitMaker chỉ cung cấp cho chúng ta một chừng mực nào đó về số lượng linh kiện ứng với số phím tắt mà chúng ta gõ trên bàn phím. Ngoài ra CircuitMaker còn cung cấp cho chúng ta cửa sổ Device Search tìm nhanh một linh kiện mà ta biết tên hoặc cùng dạng có trong thư viện.

Chúng ta sẽ tham khảo menu Device.

Menu này bao gồm:

Browse... x: Để hiển thị hộp thoại

Device Selection.

Search... X: Để hiển thị hộp thoại

Device Search.

Hotkeys1: Mở ra một list phím tắt

Hotkeys2: Mở ra một list phím tắt khác.

Dựa vào những ký hiệu viết tắt trên các phím nóng Hotkeys1 và Hotkeys2 (hình 10.25) ta có thể lấy được các linh kiện cần thiết một cách nhanh chóng và dễ dàng.



Hotkeys1		
Hotkeys2	Ground	0
	+V 5V	1
	Inverter 74LS04	2
	2-In AND 74LS08	3
	2-In OR 74LS32	4
	PNP Trans:C PNP	5
	2-In NOR 74LS02	6
	ADC0800 ADC0800	7
	2-In XNOR 4077	8
	Logic Display	9
	Ascii Display	a
	Ascii Key	A
	Battery 10V	b
	Capacitor 1uF	c
	Polar Cap 1uF	C
	Diode DIODE	d
	Zener Diode ZENER	D
	Fuse 1A	f
	Signal Gen -1/1V	g
	Data Seq DIGSRC	G
	Hex Display	h
	Hex Key	H
	I Source 100mA	i
	.IC 0V	I

Hotkey 1

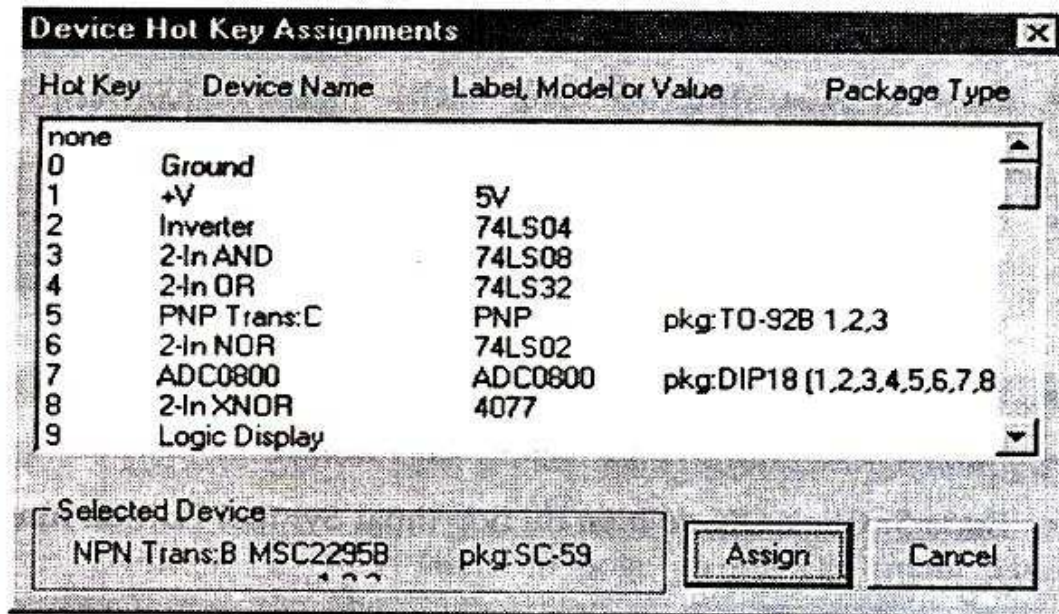
Hotkeys1		
Hotkeys2	NJFET:A NJFET	j
	PJFET:A PJFET	J
	Relay CoilA	k
	Contacts:A	K
	Inductor 1uH	l
	Var Inductor 1uH	L
	N-EMOS 3T:A NMOS	m
	P-EMOS 3T:A PMOS	M
	.NODESET 0V	N
	Op-Amp5 OPAMP5	o
	Pulser	p
	NPN Trans:C NPN	q
	PNP Trans:C PNP	Q
	Resistor 1k	r
	PNP Trans:C PNP	R
	Logic Switch 0V	s
	SPDT Switch	S
	Terminal	t
	SCOPE	T
	V Source 10V	v
	N-MESFET:A NMESFET	z
	P-MESFET:A PMESFET	Z

Hottkey 2

Hình 10.25: Màn hình hiển thị các phím nóng 1 và phím nóng 2

Đối với các IC hoặc các linh kiện khác không được gán phím tắt trên menu Hotkeys ta phải chọn linh kiện trong Part Browse và tiến hành các thao tác lựa chọn và sắp xếp linh kiện vào vùng làm việc của mình.

Ta có thể định nghĩa các phím tắt bằng cách nhấn vào nút **Change** trên **Device Selection** của **Part Browse**. Lúc này trên màn hình sẽ hiện lên một bảng liệt kê danh sách các linh kiện được định nghĩa (hình 10.26) cho phép ta có thể lựa chọn hoặc định nghĩa bằng cách click vào ký tự ta muốn định nghĩa, sau đó nhấn vào nút **Assign** và phần mềm sẽ giúp ta định nghĩa phím tắt đã lựa chọn cho linh kiện. Sau khi định nghĩa xong ta chỉ cần gõ vào ký tự đã định nghĩa là đã có thể lấy ngay được linh kiện.



Hình 10.26: Cửa sổ gán phím nóng cho các linh kiện

Mở rộng thư viện

Đôi khi thư viện chương trình không cung cấp đủ hoặc không có những linh kiện mà ta cần sử dụng. Vậy thì làm như thế nào để có thể mô phỏng hoặc thể hiện được trên sơ đồ. Để có thể tạo mới các linh kiện để mô phỏng trong CircuitMaker, có hai loại chính:

- Một là: Tạo ra linh kiện không chức năng chỉ để ta diễn tả sơ đồ đơn giản hoặc xuất tới một PCB netlist.
- Hai là: Tạo ra linh kiện có đầy đủ chức năng sẽ chạy mô phỏng được. Nó thực hiện bằng cách gán toán tử mô phỏng vào mạch nội bộ hoặc mô hình SPICE một mạch phụ cho một ký hiệu linh kiện.

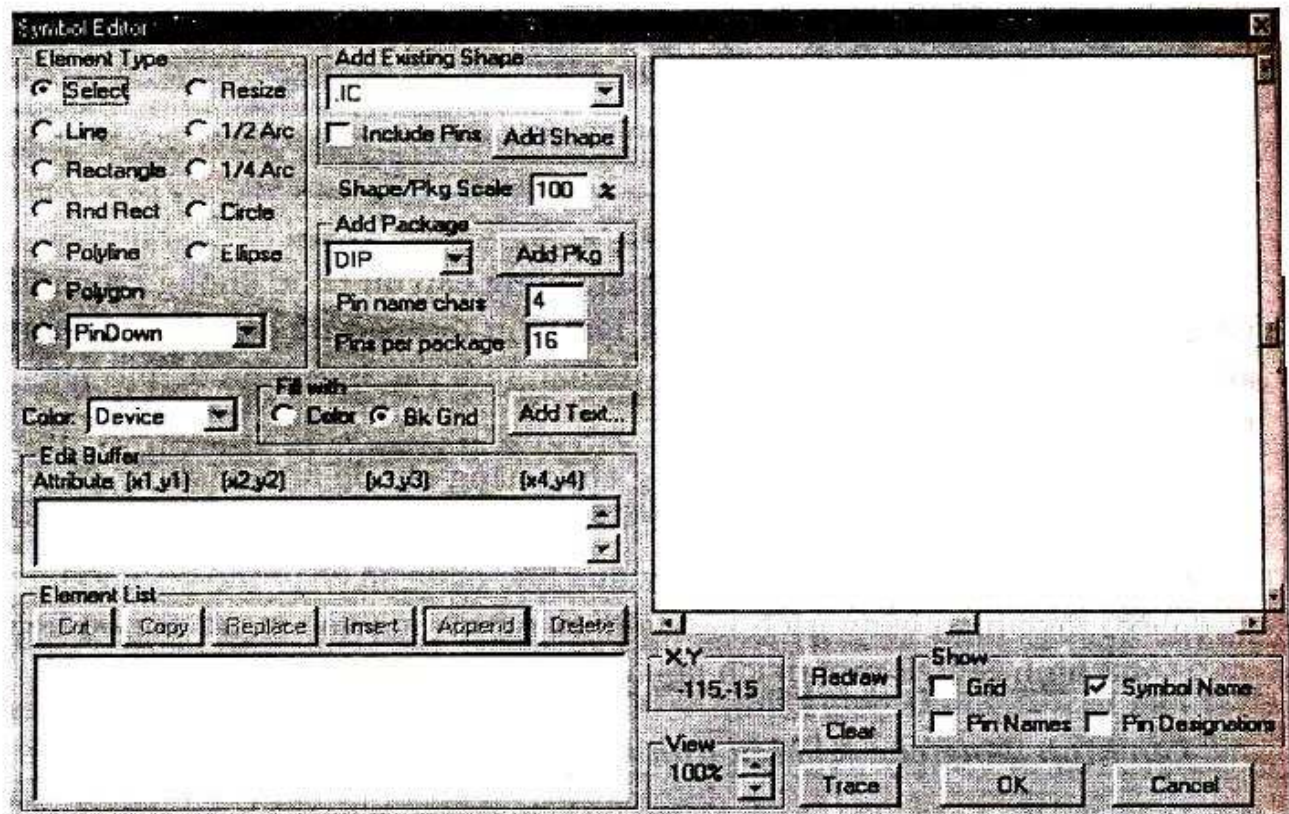
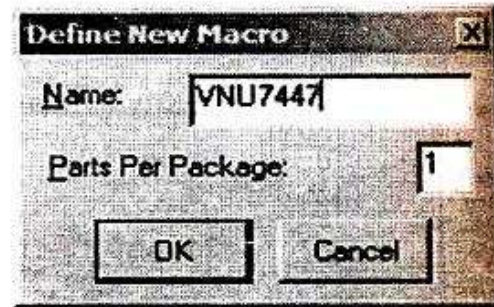
Tạo hình dáng cho ký hiệu linh kiện

Ta có thể tạo hoặc hiệu chỉnh những ký hiệu trên một hoặc nhiều cách sau:

- Vẽ một ký hiệu bằng chuột.
- Nhập vào một mô tả trên Element List.
- Thêm vào hình thể đang hiện có.
- Du nhập một tập tin trong Metafile.
- Thêm vào DIP, LCC và vỏ bao QFP.

Để tạo một ký hiệu mới

- Bước 1: Xóa vùng vẽ bằng cách click vào nút New trên Toolbar.
- Bước 2: Click vào nút Macro trên Toolbar. Hoặc vào menu Macro\New Macro.
- Bước 3: Nhập tên vào vùng Name trong hộp thoại Macro.
- Bước 4: Chỉ định số Part Per Package trên IC.
- Bước 5: Click OK để hiển thị hộp thoại Symbol Edition hình 10.26.



Hình 10.26: Hộp thoại Symbol Edition

Có thể xem Symbol Edition như là một trình vẽ với các tính năng đủ cung cấp cho chúng ta có thể tạo một hình thể mô tả linh kiện.

Ý nghĩa sử dụng các nút lệnh trong Symbol Editor

Dạng điều khiển	Nếu bấm vào...	Sẽ có tác dụng là...
CHECK BOX	Grid	Hiển thị hoặc ẩn đi lưới trên cửa sổ hiện hành View.
	Symbol Name	Hiển thị hoặc ẩn đi tên ký hiệu của linh kiện
	Pin Names	Hiển thị hoặc ẩn tên các chân linh kiện.
	Include Pins	Kèm theo chân khi hình thể xuất hiện trong cửa sổ vẽ.
	Pin Designation	Hiển thị hoặc ẩn tên các mệnh danh linh kiện
BUTTON	Redraw	Vẽ lại một ảnh
	Cut	Huỷ chọn những thành phần từ Element List và đặt chúng vào trong Edit Buffer.
	Copy	Đặt một bản sao những thành phần của chọn thể Edit Buffer.
	Replace	Thay thế thành phần đã chọn với những với nội dung của Edit Buffer.
	Insert	Chèn nội dung của Edit Buffer trước khi chọn thành phần Element List.
	Append	Gán nội dung của Edit Buffer đến cuối của Element List.
	Delete	Huỷ chọn những thành phần từ Element List.
BUTTON	Trace Step	Thông qua danh sách các phần tử Element List, bắt đầu chiếu sáng mỗi yếu tố tại một thời điểm mà nó đang được chọn hiện hành. Chọn yếu tố đầu tiên trên danh sách, click vào nút Delete để xoá cửa sổ đang vẽ, và tiếp theo click vào nút the Trace nhiều lần. Sự trợ giúp này giúp nhận dạng những yếu tố được ẩn sau các yếu tố khác.

BUTTON	Add Shape	Thêm vào một hình thể.
	Add Pkg	Thêm vào kiểu vỏ bọc.
	Add Text	Thêm một dòng văn bản vào hình thể.
	Line	Vẽ đường thẳng.
	1/4 Arc	Vẽ 1/4 cung
	1/2 Arc	Vẽ 1/2 cung
	Circle	Vẽ vòng tròn.
	Ellipse	Vẽ hình Ellipse
	Polygon	Vẽ hình đa giác.
	Polyline	Vẽ nhiều đường thẳng nối liền nhau.
	Rectangle	Vẽ hình chữ nhật.
	Round Rect	Vẽ hình chữ nhật bó góc.
	Select/Move	Chọn/Di chuyển một hình thể.
SCORLL	View	Quan sát cửa sổ vẽ từ 25% đến 800%

Các chân (pin) có thể hướng lên, hướng xuống, hướng qua trái hoặc qua phải và có thể hiển thị các chấm nhỏ diễn tả các đầu âm logic. Các chấm nhỏ này được ký hiệu là một dấu ngã (~).

Chú ý: Symbol Editor không sử dụng tính năng Snap To Grid.

Vẽ bằng lệnh

CircuitMaker cung cấp cho chúng ta dạng thức lệnh để vẽ các yếu tố như: đường thẳng, cung tròn,... bằng các lệnh được mô tả trong ElementList và Edit Buffer.

Chúng ta sẽ tham khảo một số lệnh sau:

Dạng tổng quát.

[Elementtype][thuộc tính] x1,y1x2,y2x3,y3x4,x4,y4 [số chân]

Đường thẳng

Thuộc tính

Màu bút

x1,y1

điểm khởi đầu của đường thẳng

x2,y2	điểm kết thúc của đường thẳng
x3,y3	không dùng
x4,y4	không dùng

Ví dụ : Line Device -30,-7 20,-7

Đường thẳng khép kín

Thuộc tính	màu bút
x1,y1	điểm đầu tiên của các đường thẳng khép kín
x2,y2	điểm thứ hai của các đường thẳng khép kín
x3,y3	điểm thứ ba của các đường thẳng khép kín
x4,y4	điểm thứ tư của các đường thẳng khép kín

Chú ý : Phía trước có dấu cộng(+Polyline) chỉ thị rằng đây là một sự mở rộng thành phần các đường thẳng khép kín.

Ví dụ : Polyline Device -24,-46 42,-22 24,35

Polyline Device -51,-54 46,34 34,42 -53,32

+ Polyline Device -53,-19

Polygon

Thuộc tính	màu bút
x1,y1	điểm đầu tiên của hình đa giác.
x2,y2	điểm thứ hai của hình đa giác (tùy ý nếu+Polygon)
x3,y3	điểm thứ ba của hình đa giác (tùy ý nếu+Polygon)
x4,y4	điểm thứ tư của hình đa giác (tùy chọn).

Chú ý : Phía trước có dấu cộng (+Polygon) để chỉ ra rằng đây là phần mở rộng thành phần hình đa giác.

Ví dụ: PolygonDevice -39,12 0,55 -61,75

PolygonDevice -30,-44 12,-44 29,-21 15,6

+ PolygonDevice -25,6 -42,16

Hình chữ nhật

Thuộc tính	màu bút
x1,y1	đỉnh góc trái trên cùng của hình chữ nhật
x2,y2	đỉnh góc dưới cùng bên phải của hình chữ nhật

x3,y3 không dùng

x4,y4 không dùng

Ví dụ: Rect Device -25,-30 15,28

RRect (hình chữ nhật bo tròn góc)

Thuộc tính màu bút

x1,y1 đỉnh góc trái của hình chữ nhật đầy đủ không bo tròn góc

x2,y2 đỉnh dưới cùng bên phải của hình chữ nhật đầy đủ không bo tròn góc

x3,y3 chiều rộng và chiều cao của ellipse hạn chế góc

x4,y4 không dùng

Ví dụ: RRect Device -56,-22 11,8 10,10

Ellipse

Thuộc tính màu bút

x1,y1 đỉnh góc trái trên cùng của hình chữ nhật định rõ giới hạn êlip

x2,y2 đỉnh góc dưới bên phải của hình chữ nhật định rõ giới hạn êlip

x3,y3 không dùng

x4,y4 không dùng

Ví dụ: Ellipse Device -49,-44, 9,33

Arc

Thuộc tính màu bút

x1,y1 đỉnh góc trái của hình chữ nhật ngoại tiếp ellipse

x2,y2 dưới đáy bên phải của hình chữ nhật ngoại tiếp ellips

x3,y3 điểm kết thúc đường thứ 1 tại điểm khởi đầu đến giữa của ellipse

x4,y4 điểm kết thúc của đường thứ 2 tại điểm khởi đầu đến giữa của ellipse

Chú ý: Cung bao ngoài kể sau của ellipse thì được vẽ ngược chiều kim đồng hồ từ đường thứ 1 tới đường thứ 2.

Các PinUp, PinDown, PinLeft, PinRight và PinUp~, PinDown~, PinLeft~, PinRight~.

Thuộc tính Tên Pin (tối đa 16 ký tự).

x1,y1 điểm mà chân được gắn vào vỏ bọc

x2,y2 không dùng

x3,y3 không dùng

x4,y4 không dùng

Ví dụ về mô tả một chân trên vỏ bọc:

Pinleft P1 -26,-27 [1]

PinRight P9 26,36 [9]

Ví dụ về mô tả hai chân trên vỏ bọc:

Pinleft P1 -26,-27 [1,2]

PinRight P9 26,36 [9,10]

Ví dụ của chân đảo ngược:

PinLeft~ P7 -26,27 [7]

10.2.2. Các mô hình SPICE và các mạch phụ

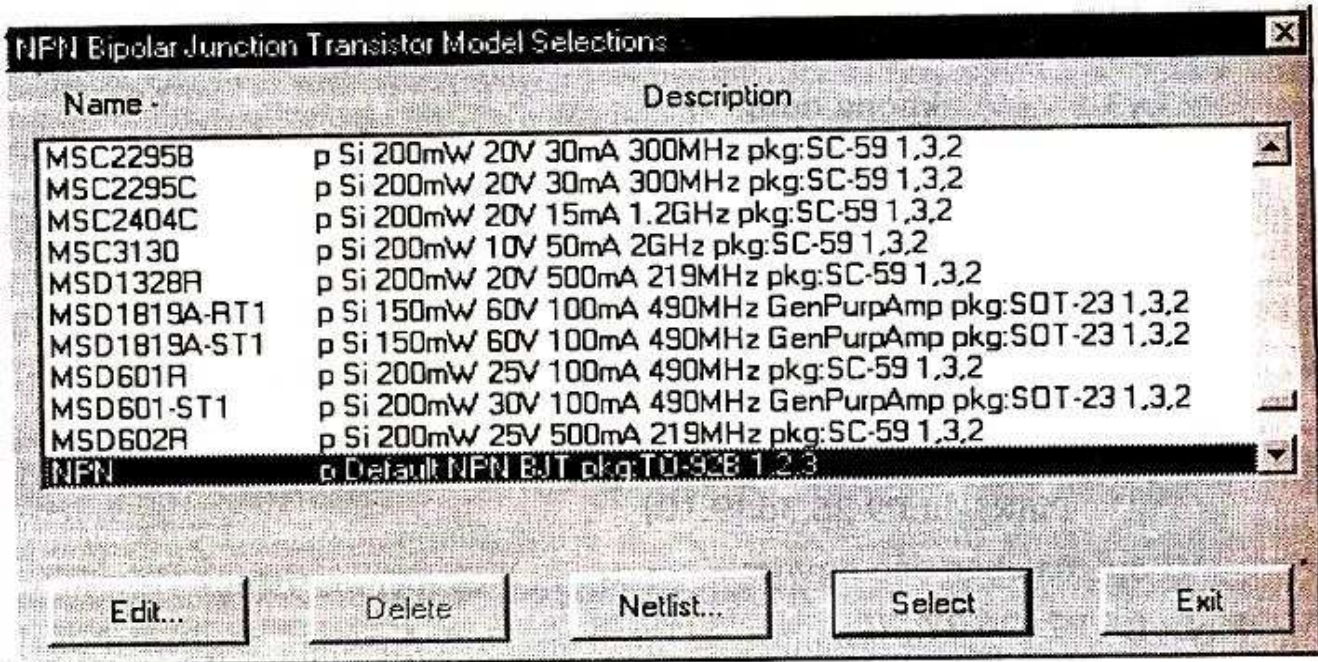
Chú ý: Nếu chúng ta có ý định tạo ra linh kiện ký hiệu mới để xuất ra một tập tin dạng PCB netlist hoặc đơn giản để vẽ sơ đồ mạch điện, ta không cần kèm theo thông tin SPICE. Tuy nhiên, để có thể chạy mô phỏng mạch điện được một cách dễ dàng và thuận tiện thì nên thêm vào các mô hình SPICE từ các nguồn khác. Có 3 kiểu cơ bản của các thành phần trong SPICE:

- Phần tử như là : điện trở, tụ điện, nguồn công suất, . . .
- Các mô hình xác định riêng biệt linh kiện như là: BJTs, J-FETs, MOSFETs,.
- Mạch phụ phối hợp nhiều mục chọn (như là các phần tử, thành phần, các mô hình và các mạch phụ khác) để tạo ra nhiều linh kiện phối hợp. Để tạo các linh kiện này thực hiện trong chế độ mô phỏng tương tự (Analog), phải có dữ liệu SPICE có sẵn cho mỗi linh kiện. Dữ liệu mô hình và mạch phụ được lưu trữ trong các file văn bản ASCII được nhận ra bằng phần đuôi mở rộng .MOD hoặc .SUB.

CircuitMaker cho phép thêm mô hình SPICE riêng và các thư viện chương trình mạch phụ có sẵn từ nhiều nguồn.

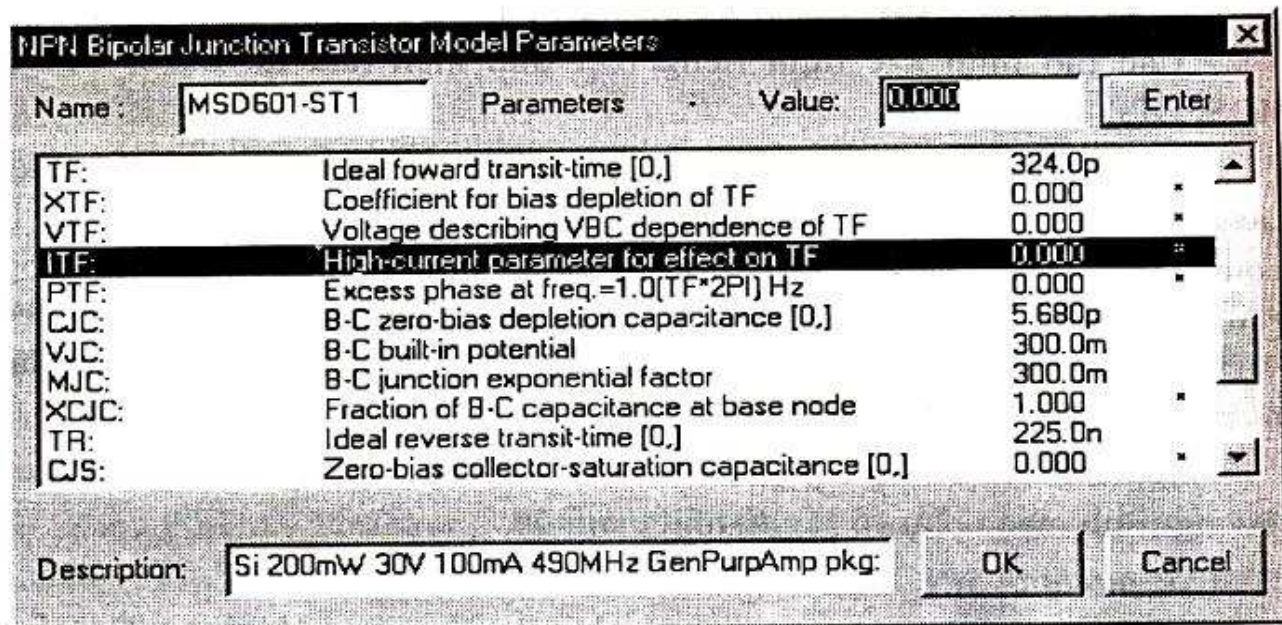
Các mô hình SPICE.

Khi ta nhấp đúp vào một linh kiện có các mô hình SPICE kết hợp với nó, hộp thoại sẽ xuất hiện như hình 10.27.



Hình 10.27

Mô hình được chọn hiện hành sẽ chiếu sáng trong danh sách, để chọn một mô hình khác nhấp nó bằng chuột sau đó chọn **Select**. Nếu bất kỳ các mạch phụ được tìm thấy trong file .MOD chúng được chỉ ra bằng một dấu (*) xem như ký tự đầu tiên trong phần mô tả. Để sửa đổi mô hình đang tồn tại nhấp lên nó bằng chuột sau đó nhấp nút **Edit** hộp thoại như hình 10.28 sẽ xuất hiện nhiều mạch phụ không thể sửa đổi theo phương pháp này.



Hình 10.28

Các tham số mô hình được trình bày chính là những tham số mô hình SPICE và không bị lẫn lộn với những tham số trong sách tra cứu dữ liệu, nhưng thường máy tính yêu cầu người thiết kế không nên thay đổi các mô hình SPICE đang tồn tại.

Nhiều giá trị được liệt kê đối với mỗi tham số biểu thị các giá trị được định rõ cho loại linh kiện đặc trưng đó. Nếu các giá trị được theo sau bằng dấu (*) đó chính là giá trị mặc định đại diện của mô hình DEFAULT không cần thiết giá trị đúng cho linh kiện đặc trưng này.

Để thay đổi giá trị một tham số SPICE ta chỉ việc chọn tham số và gõ giá trị mới vào vùng biên tập **Value** và nhấn **Enter**. Nếu muốn xác lập một tham số cụ thể bằng với tham số mô hình DEFAULT gõ một dấu (*) vào vùng biên tập **Value** và nhấn **Enter**. Để lưu mô hình dưới dạng một tên mới ta gõ tên mới vào vùng **Name** và nhấn nút **OK**.

10.3. VẼ VÀ SOẠN THẢO CÁC SƠ ĐỒ MẠCH ĐIỆN TỬ

10.3.1. Khái niệm cơ bản về sơ đồ nguyên lý mạch điện tử

Để thiết kế chế tạo được một thiết bị điện tử, người ta phải thực hiện các công đoạn cơ bản sau đây:

- Thiết kế sơ đồ nguyên lý.
- Thiết kế bản mạch in tương ứng với sơ đồ nguyên lý đã được thiết kế.
- Lắp ráp bản mạch, lắp ráp tổng thể thiết bị.
- Cho chạy thử, đo đạc kiểm tra hoạt động của thiết bị xem các thông số vật lý của thiết bị có phù hợp với yêu cầu thiết kế.

Thiết kế xây dựng sơ đồ nguyên lý của mạch điện tử là khâu đầu tiên rất quan trọng, nó quyết định sự thành công trong việc chế tạo ra thiết bị điện tử. Để thiết kế được sơ đồ nguyên lý của mạch điện tử, người thiết kế phải có kiến thức tích hợp hệ thống, hiểu biết sâu về kỹ thuật điện tử, am hiểu tường tận tính năng của từng linh kiện điện tử, biết sử dụng chúng trong từng trường hợp cụ thể.

Trên sơ đồ nguyên lý có các thành phần mạch là các linh kiện điện tử và các đường dây dẫn điện nối giữa chúng. Các sơ đồ mạch sử dụng các đường thẳng để biểu thị các đường nối (dây) và các ký hiệu tiêu chuẩn để biểu thị các thành phần mạch như các điện trở, transistor và các mạch tích hợp.

Người ta thường biểu diễn sơ đồ nguyên lý một mạch điện tử theo các khối chức năng, với mục đích sao cho người không trực tiếp thiết kế mạch điện tử đó, khi đọc dễ dàng nắm bắt nguyên tắc vận hành của mạch, giúp cho công tác bảo trì kiểm tra sửa chữa mạch nhanh chóng chính xác.

Sơ đồ nguyên lý trong CircuitMaker bao gồm các ký hiệu linh kiện được dùng trong thư viện linh kiện, các dây nối và khung bản vẽ. Các ký hiệu linh kiện có ý nghĩa rất quan trọng, qua khâu thiết kế bản mạch in, nó được chuyển thành bản mạch in để lắp ráp. Các linh kiện trong sơ đồ có kích thước và chân cắm tương ứng của một linh kiện trong thực tế lắp ráp trên mạch in. Trong chương trình CircuitMaker, mỗi linh kiện điện tử còn chứa những dữ liệu để chạy mô phỏng khi ta chạy chương trình mô phỏng mạch điện.

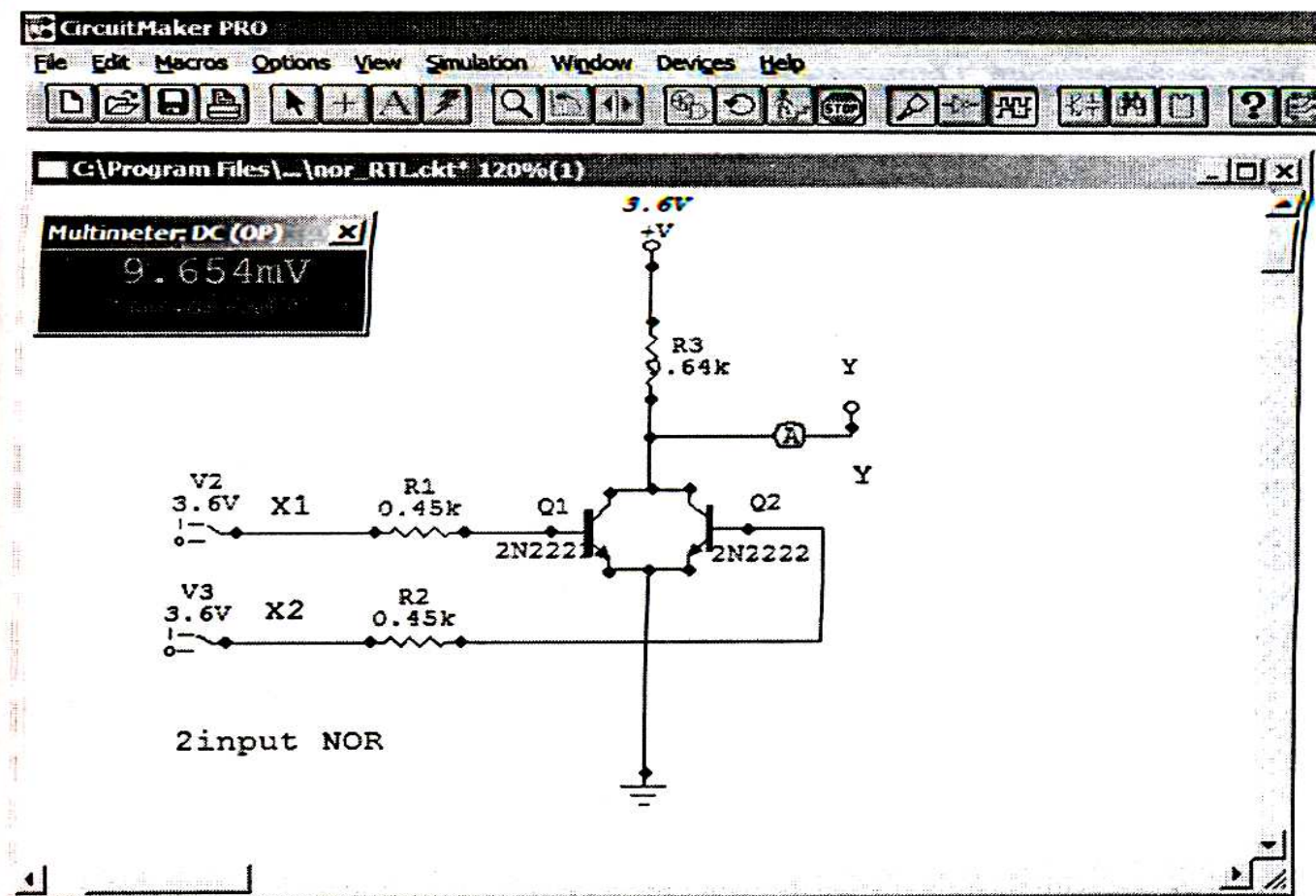
Sơ đồ nguyên lý trong CircuitMaker là cơ sở cho việc tạo một bản mạch in tương ứng với sơ đồ trong TraxMaker - một chương trình vẽ mạch in kèm theo phần mềm CircuitMaker của hãng Microcode Engineering .

10.3.2. Quy trình vẽ sơ đồ nguyên lý

Để vẽ sơ đồ trong CircuitMaker thật đơn giản, chúng ta chỉ cần bày các linh kiện có trong mạch rồi dùng các công cụ vẽ sẵn có để thực hiện vẽ mạch theo ý muốn.

Để thấy được khả năng ứng dụng vẽ mạch điện của CircuitMaker ta thực hiện một bài tập về thiết kế mô phỏng vi mạch NOR 2 lối vào thuộc họ logic RTL đã được đề cập mục 3.2, hình 3.2 (chương 3).

Dùng phần mềm CircuitMaker ta có thể thiết kế mô phỏng khảo sát hoạt động của vi mạch này, sơ đồ nguyên lý mạch NOR 2 lối vào họ RTL được trình bày trên hình 10.29.




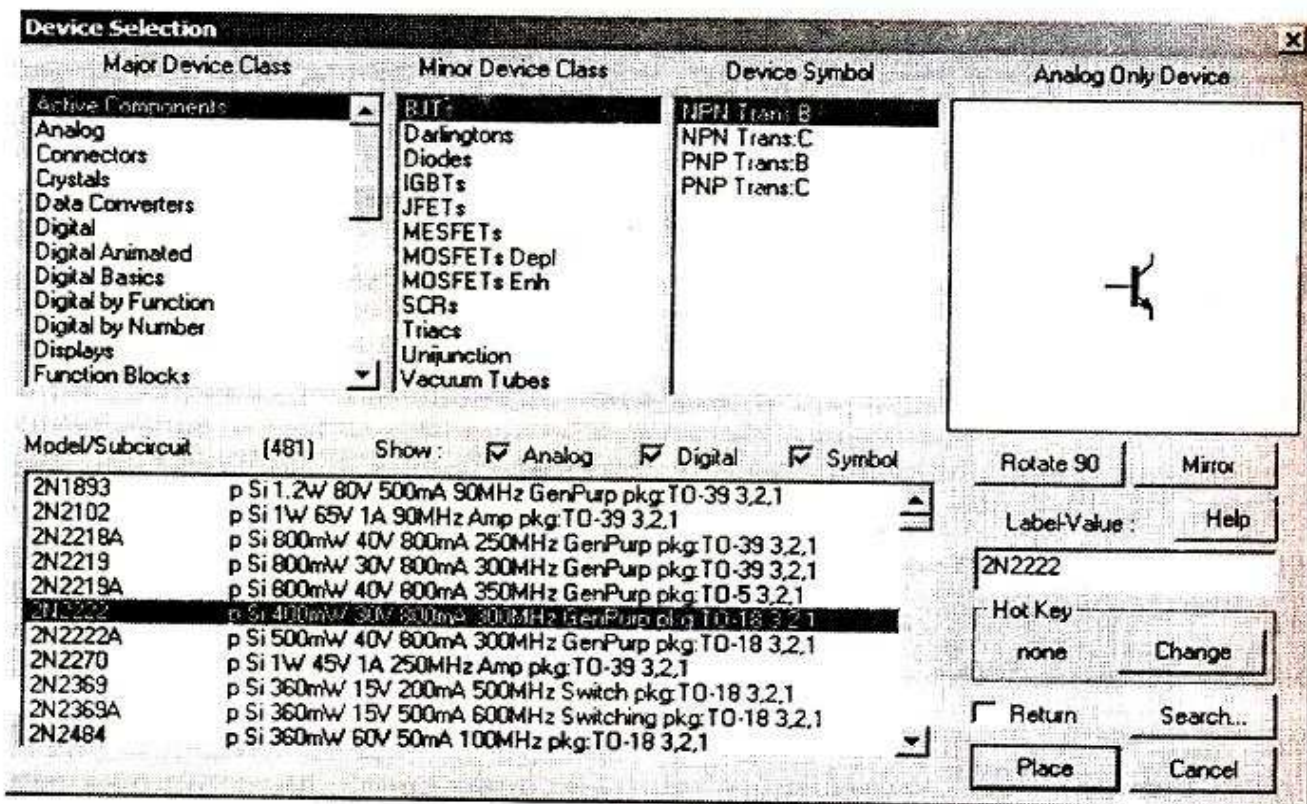
Hình 10.29: Thí nghiệm mô phỏng mạch NOR 2 lối vào họ logic RTL

Để vẽ được mạch điện như Hình 10.29 ta thực hiện theo trình tự sau:

10.3.2.1. Lấy linh kiện, thiết bị từ thư viện ra vùng làm việc


Đầu tiên ta mở trình đơn File\New, hoặc gọi lệnh bằng phím tắt: Ctrl+N, cách khác nhấn nút lệnh .

Các thiết bị linh kiện có thể tìm thấy hết trong hộp thoại Device Selection bằng cách nhấn chuột vào nút **Part button**  trên thanh công cụ, hoặc vào **Devices -> Browse**, hoặc nhấn phím x trên bàn phím để mở hộp thoại chọn linh kiện (hình10.30). Chọn linh kiện cần lấy, nhấn **Place** để đặt linh kiện vào vị trí trên mạch.

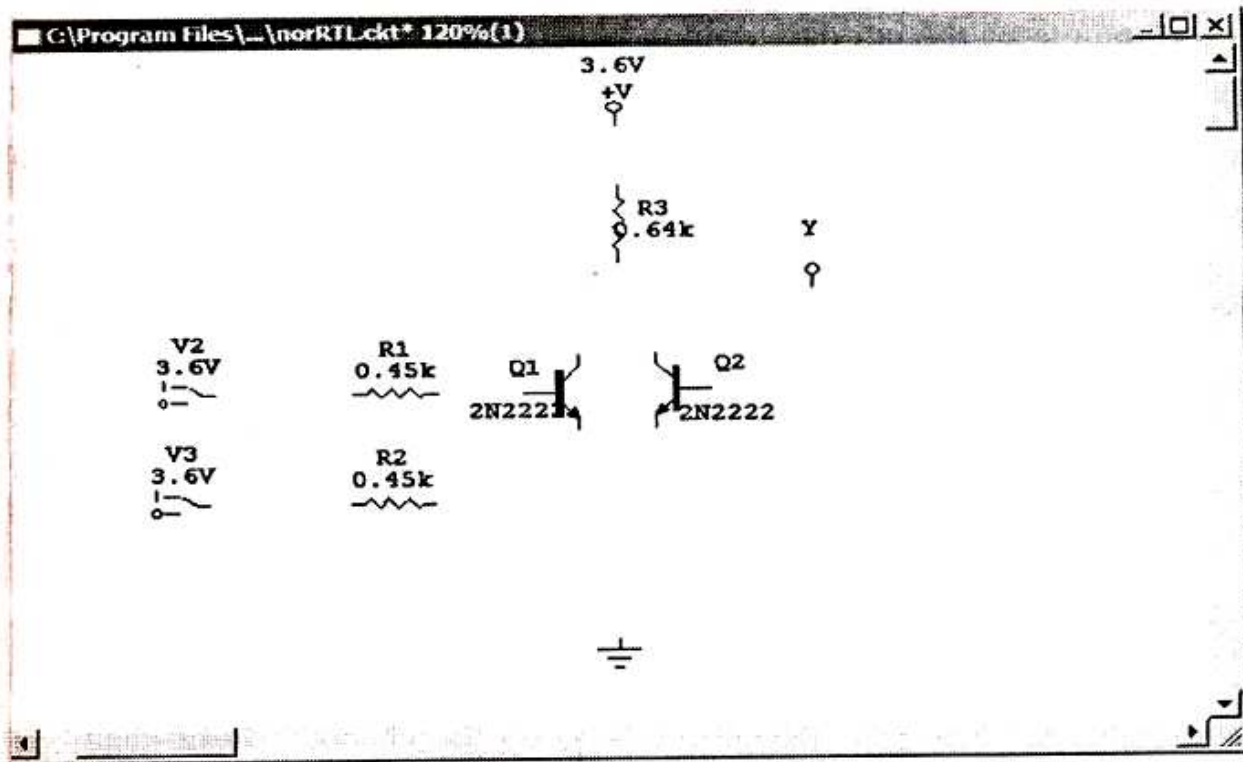


Hình 10.30: Cửa sổ chọn linh kiện

Các linh kiện có thể chọn trong hộp thoại Devices Selection, hoặc có thể sử dụng các phím tắt **Hotkey1&2** để lấy thiết bị linh kiện ra một cách dễ dàng và nhanh chóng.

Sau khi đã tìm kiếm lấy linh kiện sử dụng công cụ mũi tên  di chuyển linh kiện đặt linh kiện vào vị trí trong vùng làm việc theo các hướng dẫn sử dụng CM mục 10.2 ta được hình 10.31


Dùng công cụ nối dây nối các linh kiện, dùng công cụ soạn chữ **Text Tool** tạo các chữ chú giải trên sơ đồ, được sơ đồ hoàn thiện như hình 10.29.



Hình 10.31

10.3.2.2. Nối dây kết nối các linh kiện với nhau

Sau khi chúng ta đã có các linh kiện cần thiết để vẽ mạch điện như hình 10.30 trên vùng vẽ (chúng ta không nên quan tâm đến giá trị của các linh kiện, vì khi vẽ các đường nối xong thì việc sửa đổi các giá trị của chúng sẽ rất dễ dàng). Bây giờ đến công đoạn vẽ đường dây nối mạch.

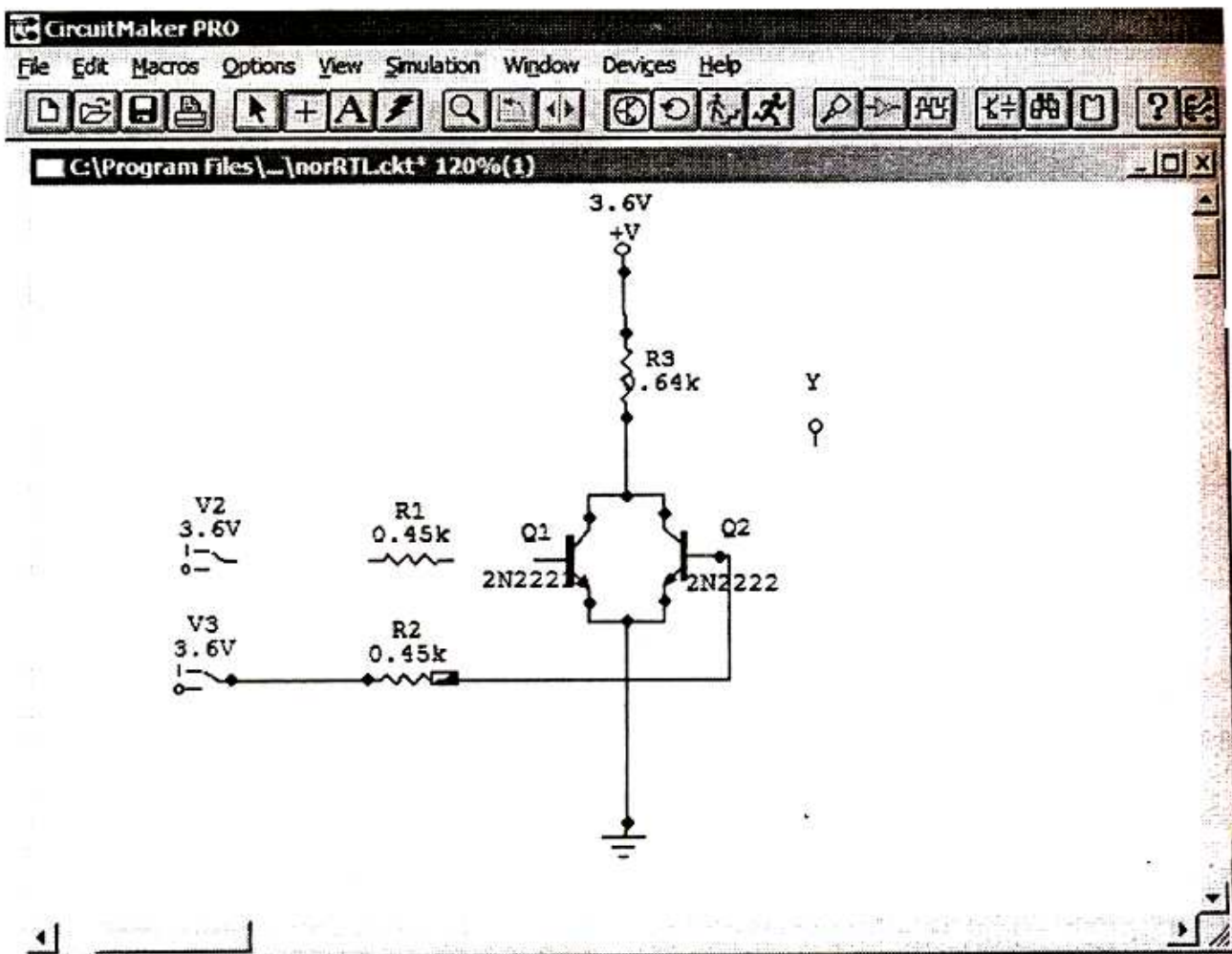
Bước 1: Bấm vào nút lệnh nối dây  để tạo dây nối.

Bước 2: Một dấu thập phân màu đen xuất hiện và dấu thập này sẽ biến thành hộp chữ nhật nhỏ màu đỏ khi ta di chuyển trỏ chuột đến vị trí chân linh kiện.

Bước 3: Đưa trỏ chuột về chân linh kiện và click trái chuột. Đường sáng màu xanh xuất hiện theo sự di chuyển của trỏ chuột. Ta di chuyển chuột về phía chân linh kiện cần nối đến rồi click trái chuột để ngắt lệnh.

Khi gặp một đường gấp khúc ta chỉ cần click trái chuột rồi tiếp tục rê chuột vẽ tiếp.

Chúng ta có thể nối dây tự động bằng cách di chuột đến điểm cần xuất phát, dây nối ở đó sẽ xuất hiện một ô vuông nhỏ màu đỏ, ta nhấp chuột và kéo rê đến điểm cần nối tiếp theo, ở đây cũng xuất hiện ô vuông màu đỏ thì ta nhả chuột, dây sẽ được nối tự động (hình 10.32).

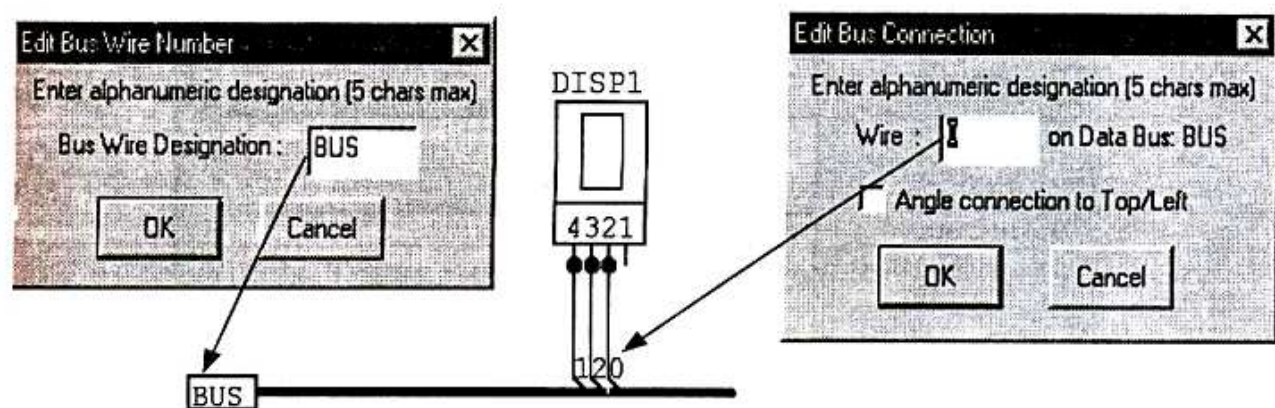


Hình 10.32: Thực hiện nối dây tự động

10.3.2.3. Sử dụng các đường BUS

Trong các sơ đồ mạch điện tử số, nhiều khi mạch có nhiều dây nối vào một thiết bị, người ta thực hiện việc nối dây theo kiểu BUS, thực hiện kiểu này các đường nối dây vào cùng một thiết bị được bó gọn lại thành một bó, sơ đồ trong thoáng hơn. BUS là một bó dây bao gồm rất nhiều dây đơn riêng biệt.

Để vẽ đường dây BUS ta giữ phím **Shift** và vẽ một dây bình thường qua việc dùng phương pháp vẽ bằng tay. Các dây BUS cũng có thể kéo dài, cắt hay đặt lại vị trí như dây đơn bình thường. Mỗi một đường BUS được nhận ra bằng một tên dài không quá 5 ký tự chữ hoặc số. Và mỗi dây đơn khi nối vào đường BUS thì sẽ phải ấn một con số địa chỉ (gồm 5 ký tự chữ hoặc số). Trên hình 10.33 minh hoạ kiểu nối dây theo kiểu BUS.



Hình 10.33: Nối dây theo kiểu BUS

Sau khi chúng ta đã hoàn tất việc nối dây, bây giờ là lúc để ta hiệu chỉnh lại một số các giá trị của linh kiện.

10.3.2.4. Thay đổi các dữ liệu, các tham số của linh kiện, thiết bị.

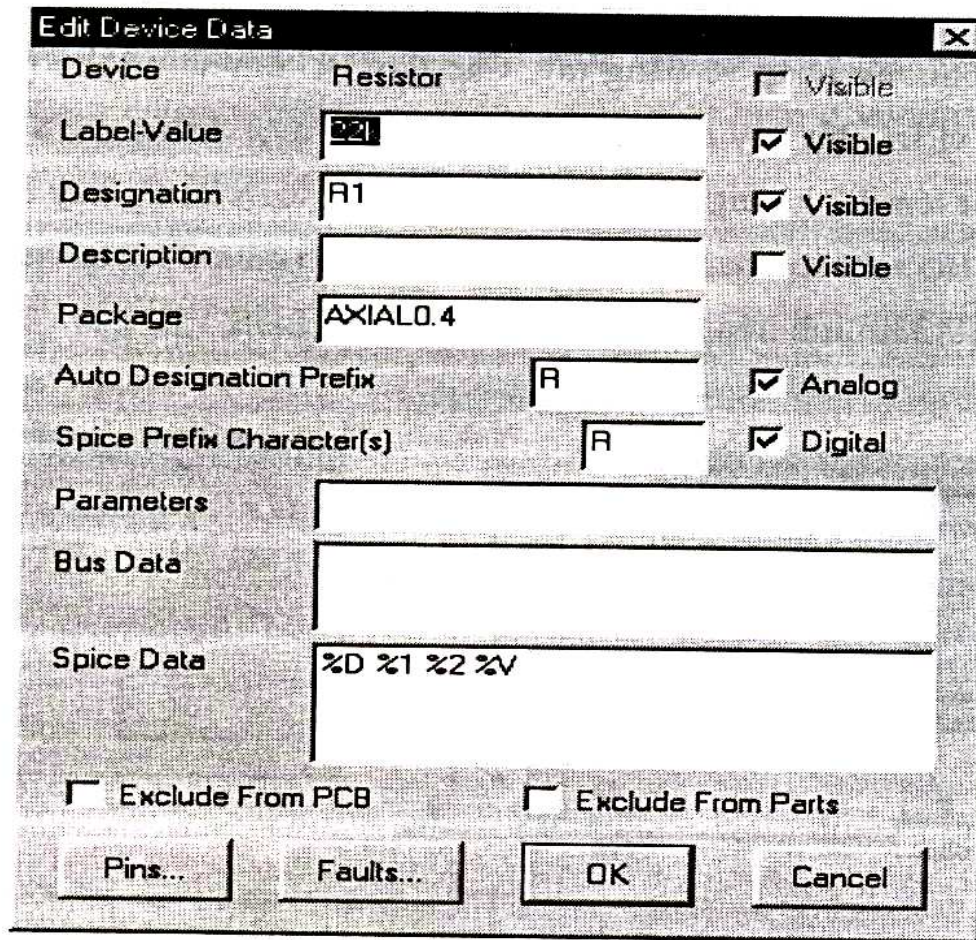
Bấm vào nút lệnh mũi tên  để chọn linh kiện cần sửa đổi

+ Cách 1: Click đúp trái chuột vào thiết bị, linh kiện cần thay đổi dữ liệu.

+ Cách 2: Mở hộp thoại **Edit Device Data** bằng cách click phải chuột vào thiết bị, linh kiện rồi chọn **Edit Device Data**, hoặc mở trình đơn:

Edit\Edit Item\Edit Device Data.

Ví dụ trong sơ đồ hình 10.30 ta di chuyển chuột đến vị trí điện trở R1, nhấp đúp phím trái chuột, ta có hộp thoại **Edit Device Data** như trên hình 10.34.



Hình 10.34: Hộp thoại Edite Divice Data

Bây giờ chúng ta sẽ nhập các giá trị phù hợp theo ý mình vào vùng. Bên cạnh các vùng ta thấy có các check-box Visible, ta sẽ đánh dấu vào nếu ta muốn hiển thị dòng đó trên vùng làm việc.

+ Device : Đây là vùng không thể sửa đổi vì nó là tên loại linh kiện, thiết bị mà ta sử dụng.

+ Label-Value : Vùng này dùng để nhập thông tin về thiết bị như nhãn linh kiện (2N3055, 2N2222,...) hoặc giá trị của linh kiện (1kΩ, 10μF, 1mH,...).

+ Designation : Vùng này dùng để nhập thông tin nhận dạng thiết bị trong mạch như : U1, R2, C, IC1,...

+ Package : Vùng này dùng để nhập thông tin về vỏ bao linh kiện, thiết bị.

+ Auto Designation Prefix : Vùng này dùng để nhập thông tin một cách tự động về các tiền tố được sử dụng trong ký hiệu của linh kiện, thiết bị. Do sự thiết đặt sẵn có trong "Set Auto Designation..." chọn từ menu Edit.

+ Spice Prefix Character(s): Vùng này dùng để nhập thông tin về các tiền tố được sử dụng trong ký hiệu của linh kiện, thiết bị. Do sự thiết đặt theo ý muốn, lưu ý các tiền tố mà ta thiết lập chỉ hợp lệ khi tối đa là 04 ký tự và nó phải phù hợp với các dữ liệu Spice.

Các tiền tố SPICE hợp lệ:

BV	Nonlinear Dependent Voltage Sources (Nguồn điện áp phụ thuộc phi tuyến).
BI	Nonlinear Dependent Current Sources (Nguồn dòng phụ thuộc phi tuyến).
C	Capacitor (Tụ điện).
D	Junction Diode (Đi ốt)
DZ	Zener Diode
E	Linear Voltage – Controlled Voltage Source
F	Linear Current – Controlled Current Source
G	Linear Voltage – Controlled Current Source
H	Linear Current – Controlled Voltage Source
JN	JFETs (N-Chanel) JFET kênh N
JP	JFETs (P-Chanel) JFET kênh P
K	Coupled (Mutual) Inductors (Hỗ cảm).
L	Inductors (cuộn kháng).
MN	MOSFETs (N-Chanel) MOSFET kênh N
MP	MOSFETs (P-Chanel) MOSFET kênh P
O	Lossy Transmission Lines (đường dây tải tổn thất).
T	Lossless Transmission Lines (đường dây tải không tổn thất).
QN	Bipolar Junction Transistors (NPN)
QP	Bipolar Junction Transistors (PNP)
R	Resistor (Điện trở).
S	Voltage Controlled Switches (Nguồn điện áp điều khiển bởi chuyển mạch).
W	Current Controlled Switches (Nguồn dòng điều khiển bởi chuyển mạch).

U	Uniform Distributed RC lines (Lossy). (Phân bố trên đường dây RC có tổn thất).
V	Independent Voltage Source (Nguồn điện áp độc lập)
I	Independent Current Source (Nguồn dòng độc lập).
X	Subcircuit (Mạch phụ).
ZN	MESFETs (N-Chanel) (GaAs FETs)- MESFETs kênh N
ZP	MESFETs (P-Chanel) (GaAs FETs)- MESFETs kênh P

+ Check-box Analog : Nếu được đánh dấu nó sẽ chỉ ra linh kiện, thiết bị Analog khai báo cho trình mô phỏng tương tự, xác nhận có dữ liệu Spice phù hợp cho linh kiện, thiết bị.

+ Check-box Digital : Nếu được đánh dấu nó sẽ chỉ ra linh kiện, thiết bị Digital khai báo cho trình mô phỏng số, xác nhận có mã mô phỏng phù hợp cho linh kiện, thiết bị.

+ Bus Data : Vùng này dùng để nhập thông tin định chân trên linh kiện, thiết bị được nối với nguồn điện hoặc các đường Bus nối đất, bởi vì các chân này không chỉ rõ trên vỏ linh kiện, thiết bị.

+ Spice Data : Vùng này dùng để nhập thông tin xác định dữ liệu mô phỏng Spice cho thiết bị. Các linh kiện, thiết bị Analog trong CircuitMaker đã cung cấp sẵn dữ liệu mặc định cần dùng. + Faults... : Nhấn nút lệnh này sẽ mở ra một hộp thoại Device Fault để bổ sung các dữ liệu hỏng cho thiết bị (PAN).

+ Check-box Exclude From Parts : Nếu được đánh dấu nó sẽ loại trừ các linh kiện, thiết bị từ danh sách liệt kê.

+ Check-box Exclude From PCB : Nếu được đánh dấu nó sẽ loại trừ một linh kiện, thiết bị từ danh sách PCB (PCB netlist) liệt kê.

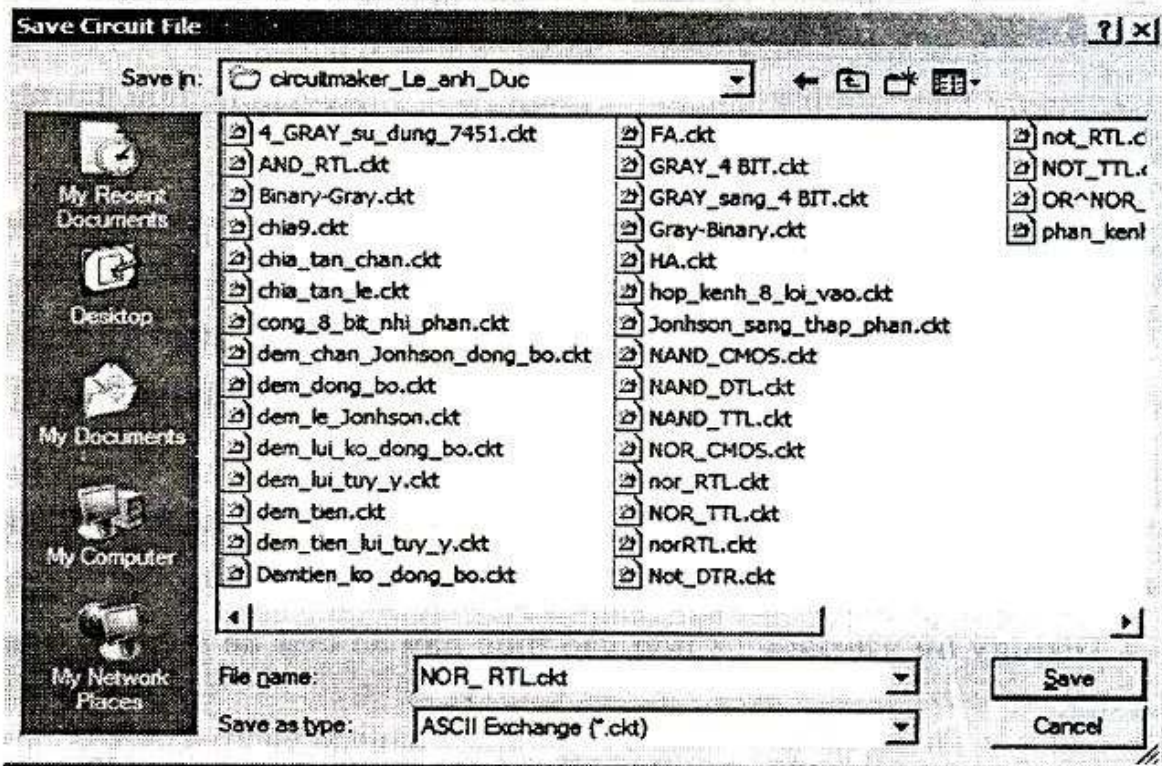
+ Pin... : Nhấn nút lệnh này sẽ xuất hiện hộp thoại Edit Device Pin Data, hộp thoại này sẽ cho ta sửa đổi con số chân của vỏ linh kiện, thiết bị.

Lần lượt thực hiện như thế cho đến khi nào các giá trị trong mạch hoàn chỉnh theo các thông số mà ta đã thiết kế. Khi ấy mạch điện của chúng ta sẽ hoàn chỉnh như hình 10.29 ở trên.

10.3.2.5. Lưu trữ sơ đồ thiết kế

Bước 1: Mở trình đơn **File\Save** (cho lần đầu tiên), **Save as** (cho các lần lưu sau), hoặc gọi lệnh bằng phím tắt: **Ctrl+S**.

Cách khác nhấn vào nút lệnh . Lần đầu tiên lưu tài liệu một hộp hội thoại **Save Circuit File** xuất hiện như hình 10.35. dưới đây:



Hình 10.35: Hộp thoại lưu dữ file

Bước 2:

+ **Save in** : Cho ta chọn thư mục hay Folder muốn lưu trữ.

File name : Vùng này dùng để đặt tên cho tập tin sơ đồ mạch điện.

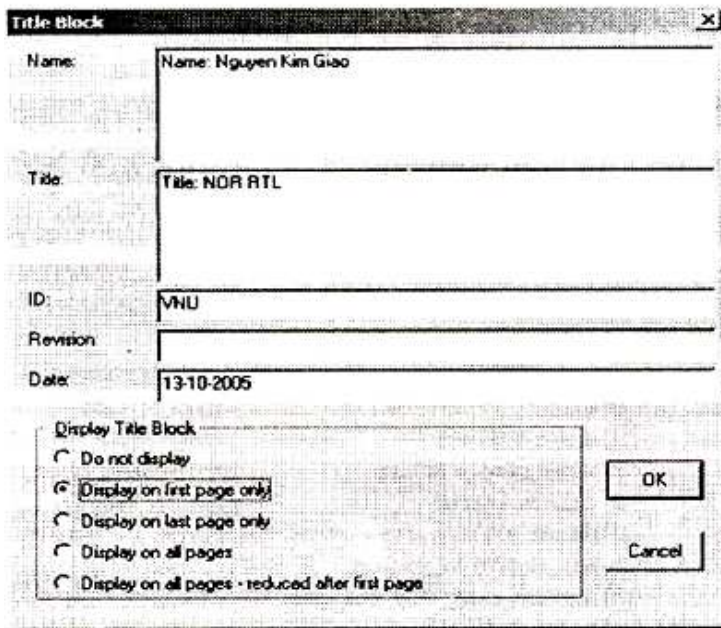
+ **Save as type** : Đây là vùng để ta chọn phần mở rộng của tập tin. Thông thường ta nên để giá trị mặc định.

Bước 3: Nhấn Save

10.3.2.1. Thiết kế tiêu đề bản vẽ

Bước 1: Mở trình đơn **Option\Title Box**, hoặc gọi lệnh bằng phím tắt: **Ctrl+B**.

Hộp thoại Title Box sẽ hiện ra như hình 10.36.



Hình 10.36: Hộp thoại Title Block

+ **Name:** Vùng này dùng để nhập tên.

+ **Title:** Vùng này dùng để nhập tiêu đề.

+ **ID:** Vùng này nhập danh xưng.

+ **Revision:** Ghi lại những sửa chữa.

+ **Date:** Ghi ngày tháng năm thiết kế.

Display Title Block bao gồm các radio button sau:

+ **Do not display** : Chọn vào mục này để không hiển thị hộp tiêu đề.

+ **Display on first page only**: Chọn vào mục này để tiêu đề chỉ hiển thị ở trang đầu.

+ **Display on all page** : Chọn vào mục này để tiêu đề hiển thị ở tất cả các trang.

+ **Display on all page - reduce after first page** : Chọn vào mục này thì hộp tiêu đề hiển thị ở các trang hoặc chèn vào trang trước.

Bước 2: Điền đầy đủ vào các mục như trình bày ở bước 1 và nhấn **OK**, ta sẽ có khung tiêu đề như hình 10.37 dưới đây.

Name: Nguyen Kim Giao	
Title: NOR RTL	
Rev	ID VNU
Date: 13-10-2005	Page: 1 of 1

Hình 10.37: Tiêu đề bản vẽ sơ đồ nguyên lý mạch

10.4. MÔ PHÒNG MẠCH ĐIỆN TỬ

Một trong những tính năng mạnh nhất của CircuitMaker là mô phỏng mạch điện, cho phép bạn thử các dạng thiết kế khác nhau và xử lý sự cố trên các thiết kế đó trước khi đầu tư thời gian và tiền bạc vào chế tạo các phần cứng. CircuitMaker mang đến cho chúng ta thế giới của một phòng thí nghiệm ảo, việc cho chạy một sơ đồ nguyên lý giống như ta đã thực hiện mạch trên board vạn năng trong phòng thí nghiệm thực.

10.4.1. Các mode mô phỏng của CircuitMaker

CircuitMaker là một trong số ít chương trình mô phỏng cung cấp 2 mode mô phỏng riêng biệt : mode tương tự (Analog Mode) và mode số (Digital Mode), mỗi mode có các ưu điểm phụ thuộc vào loại mô phỏng mà bạn cần, ngoài ra còn chế độ mô phỏng hỗn hợp tương tự với số (Analog/Mixer). Sự tách biệt 2 kiểu mô phỏng cho phép bạn thực hiện mô phỏng linh hoạt hơn và điều khiển dễ dàng các tình huống mô phỏng mạch điện tử tương tự và điện tử số.

10.4.1.1. Chế độ mô phỏng Analog

Mode tương tự là mode mô phỏng chính xác, “giống thực tế” mà bạn có thể sử dụng cho các mạch tương tự, số và mạch hỗn hợp. Mode này cho bạn các kết quả giống như thu được từ bản mạch thử. Trong mode tương tự, các linh kiện hoạt động giống như trong thực tế. Ví dụ, các IC số có thời gian trễ truyền dẫn, thời gian thiết lập trạng thái và thời gian trễ v.v... chính xác như thực. Lỗi ra của các linh kiện này chịu ảnh hưởng của tải và gần như mọi thông số trong thực tế đều được đưa vào để tính toán trong quá trình mô phỏng hoạt động của mạch.


Trong chế độ Analog, việc phân tích trạng thái rất rộng, có thể dùng chế độ Analog để kiểm tra và phân tích nhiều phương diện của mạch điện.

Chạy mô phỏng Analog

Để chạy mô phỏng Analog ta thực hiện các bước sau:

Bước 1 : Vẽ một mạch điện.

Bước 2 : Kích hoạt chế độ mô phỏng Analog bấm vào biểu tượng 

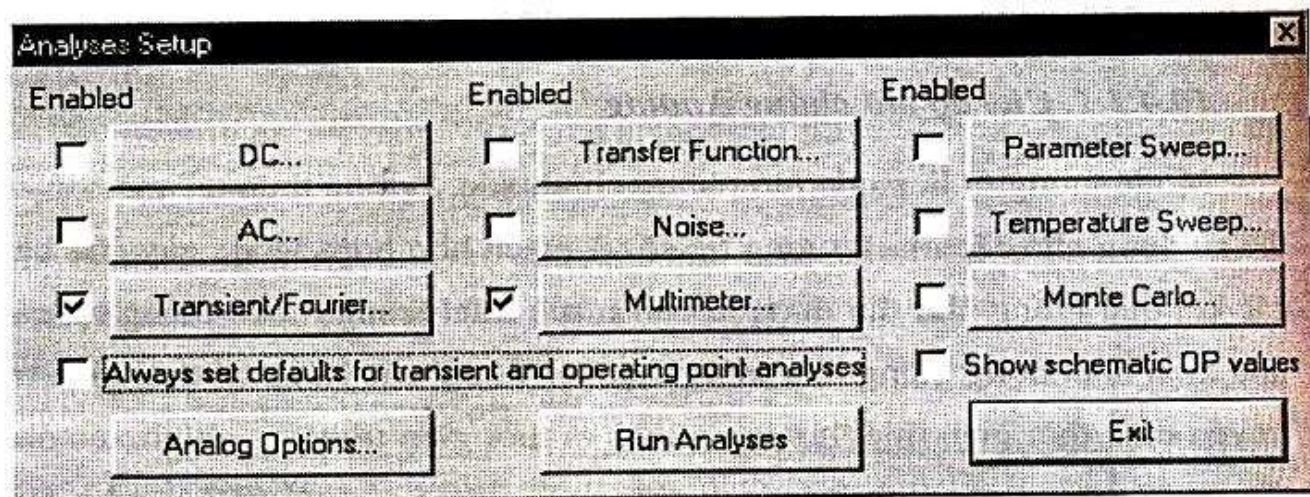
Bước 3 : Nhấn vào nút 

Cách khác : Vào menu Simulation\Run hoặc nhấn F10.

Chú ý : Muốn thực hiện việc phân tích tương tự thì tùy chọn Always Set Defaults phải được kích hoạt.

Dùng hộp thoại Analyses Setup (hình 10.38) trong menu Simulation\Analyses Setup hoặc nhấn tổ hợp phím Ctrl+F8, để thiết đặt phép phân tích tương tự.

Nếu như check-box này không được đánh dấu mà ta chạy mô phỏng thì một cảnh báo xuất hiện và thiết bị linh kiện ấy được xem như không thể mô phỏng được.



Hình 10.38

Thí nghiệm mô phỏng mạch điện tử tương tự

Để minh họa cho thấy rõ tiện ích của các công cụ dùng cho thí nghiệm mô phỏng của phần mềm CircuitMaker trong nghiên cứu thiết kế các mạch kỹ thuật điện tử tương tự, chúng tôi *xin nêu ra một ví dụ các mạch điện tử rất đơn giản để làm thí nghiệm mô phỏng.*

Trên hình 10.39a, hình 10.39b, hình 10.39c trình bày các thí nghiệm mô phỏng khảo sát hoạt động của các bộ khuếch đại dùng tranzito lưỡng cực. Hình 10.39a là bộ khuếch đại được mắc theo sơ đồ emitor chung EC, hình 10.39b là bộ khuếch đại mắc theo sơ đồ bazơ chung BC, hình 10.39b là bộ khuếch đại được mắc theo sơ đồ colectơ chung CC.

Dụng cụ đo lường dùng cho các bài thí nghiệm mô phỏng này là máy hiện sóng và đồng hồ đa năng, hai loại thiết bị này được dùng một cách phổ biến trong các phòng thực tập điện tử. Dùng máy hiện sóng ta có thể quan sát các dạng tín hiệu ở tất cả các điểm trong sơ đồ nguyên lý của bài thí nghiệm như kiểm tra trên mạch trong thực tế.

Đồng hồ đa năng cho phép đo điện áp một chiều, điện áp xoay chiều, đo dòng điện chạy trong mạch, nó được dùng để thực hiện các nội dung thực tập nêu trong bài như nội dung thực tập thực tế. Đồng hồ đa năng khi chuyển sang chế độ đo điện áp xoay chiều ta có thể dùng nó để đo điện áp ở lối vào và lối ra của bộ khuếch đại, và xác định được hệ số khuếch đại điện áp của bộ khuếch đại một cách nhanh chóng.

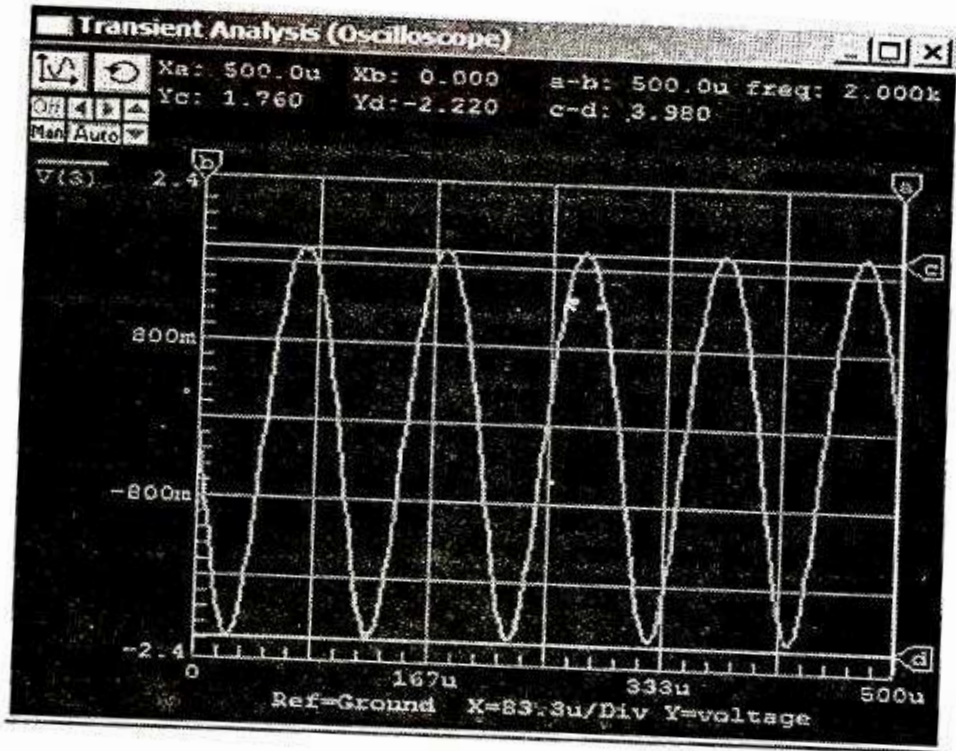
CircuitMaker PRO
 File Edit Macros Options View Simulation Window Devices Help

C:\Program Files\CircuitMaker\TAEC CST 100% (1)

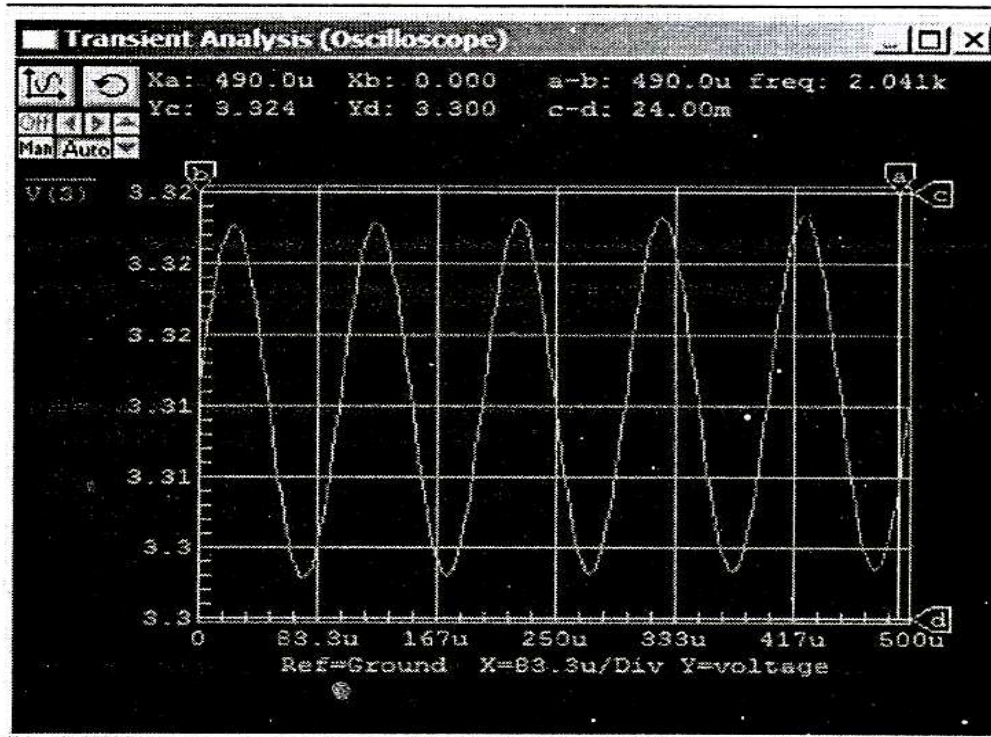
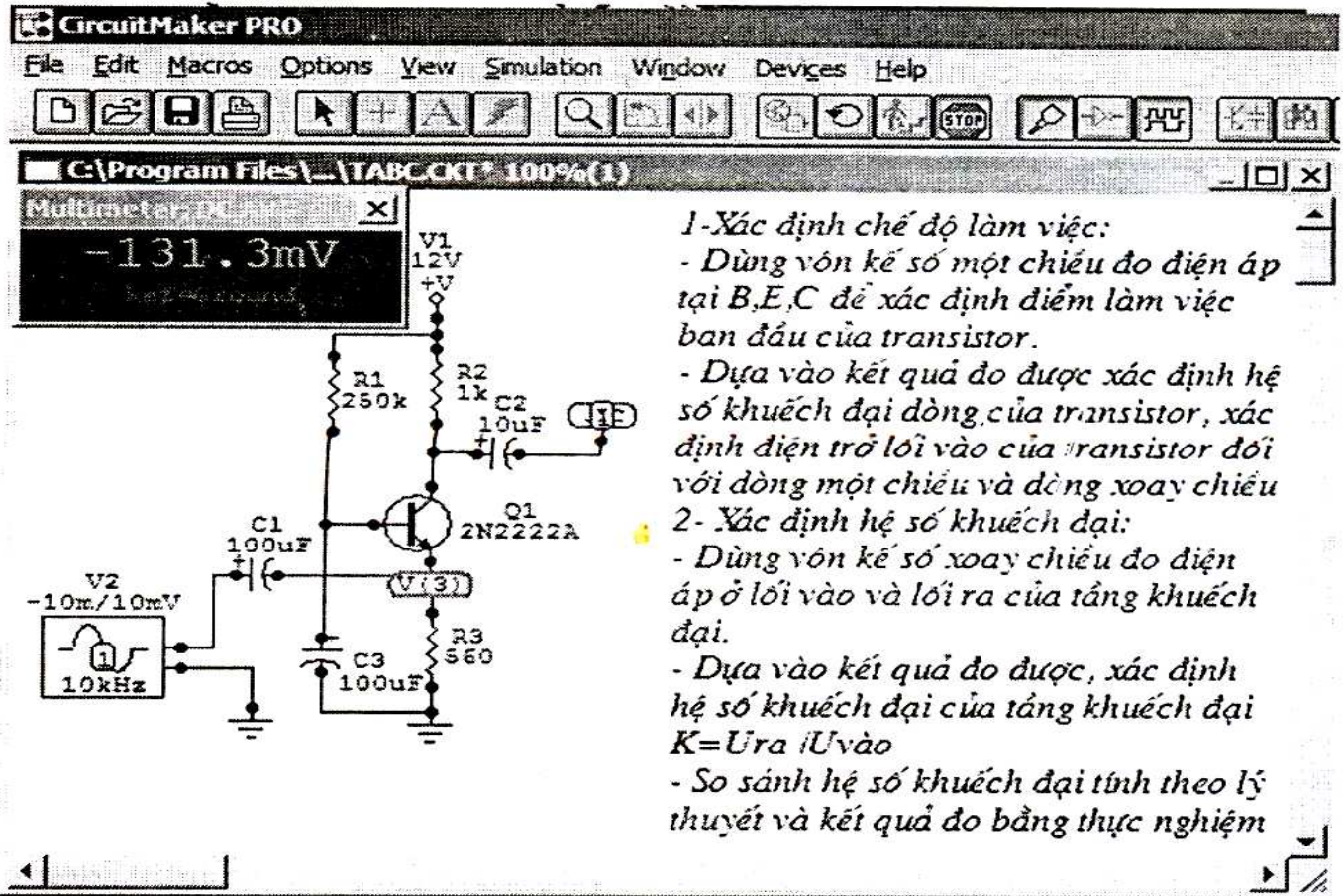
-48.72mV

1- Xác định chế độ làm việc:
 - Dùng vôn kế số một chiều đo điện áp tại B, E, C để xác định điểm làm việc ban đầu của transistor.
 - Dựa vào kết quả đo được xác định hệ số khuếch đại dòng của transistor, xác định điện trở lõi vào của transistor đối với dòng một chiều và dòng xoay chiều

2- Xác định hệ số khuếch đại:
 - Dùng vôn kế số xoay chiều đo điện áp ở lõi vào và lõi ra của tầng khuếch đại.
 - Dựa vào kết quả đo được, xác định hệ số khuếch đại của tầng khuếch đại $K = U_{ra} / U_{vào}$
 - So sánh hệ số khuếch đại tính theo lý thuyết và kết quả đo bằng thực nghiệm



Hình 10.39a: Thí nghiệm mô phỏng khảo sát hoạt động của bộ khuếch đại dùng transistor lưỡng cực, mắc theo đồ Emitơ chung EC



Hình 10.39b: Thí nghiệm mô phỏng khảo sát hoạt động của bộ khuếch đại dùng transistor lưỡng cực mắc theo sơ đồ Baza chung BC

Arrow Tool: Use to select, move and edit devices, wires and text

File Edit Macros Options View Simulation Window Devices Help

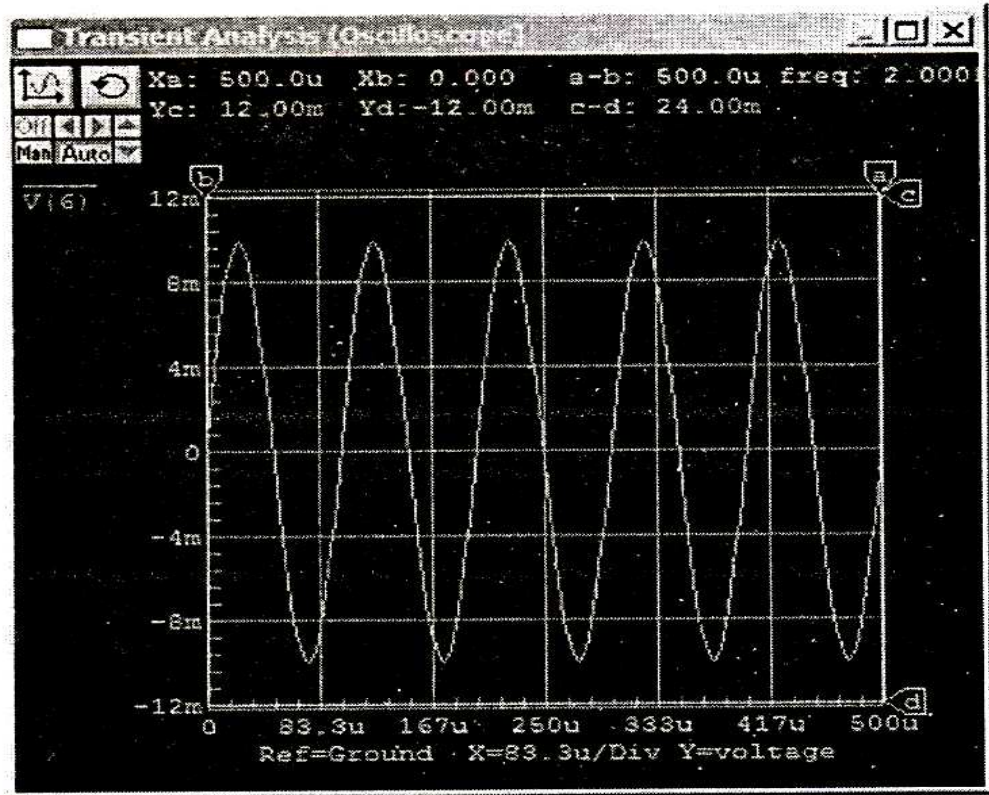
C:\CM61\OKTCM6\TACCCKT 100%(1)

1- Xác định chế độ làm việc:

- Dùng vôn kế số một chiều đo điện áp tại B, E, C để xác định điểm làm việc ban đầu của transistor.
- Dựa vào kết quả đo được xác định hệ số khuếch đại dòng của transistor, xác định điện trở lối vào của transistor đối với dòng một chiều và dòng xoay chiều

2- Xác định hệ số khuếch đại:

- Dùng vôn kế số xoay chiều đo điện áp ở lối vào và lối ra của tầng khuếch đại.
- Dựa vào kết quả đo được, xác định hệ số khuếch đại của tầng khuếch đại $K = U_{ra} / U_{vào}$
- So sánh hệ số khuếch đại tính theo lý thuyết và kết quả đo bằng thực nghiệm



Hình 10.39c: Thí nghiệm mô phỏng khảo sát hoạt động của bộ khuếch đại dùng transistor lưỡng cực mắc theo sơ đồ Colectơ chung

10.4.1.2. Chế độ mô phỏng số

Digital Mode được thiết kế cho mô phỏng số hoàn toàn. Mode này chỉ được dùng cho các mạch số, và chỉ phụ thuộc vào các trạng thái logic của các linh kiện trong mạch. Mô phỏng mode số cũng vẫn tính đến trễ truyền dẫn, nhưng đó là các thời gian trễ đơn vị thay vì trễ truyền dẫn thực. Mode này không cần dây nguồn, và các mức lối ra số của linh kiện được giữ cố định. Điện tử số là thế giới của máy vi tính. Các số cơ số 2 là 1 và 0 của máy vi tính thực ra là các mức điện áp cao và thấp của các linh kiện điện tử nhỏ bé được gọi là các mạch tích hợp. Do đó, mô phỏng logic số trở thành một công việc khá đơn giản vì chỉ có một số hữu hạn các trạng thái số cần được hiển thị. Bộ mô phỏng logic số của CircuitMaker rất nhanh và hoàn toàn tương tác tức thời, nghĩa là bạn có thể đổi trạng thái các chuyển mạch và thay đổi mạch trong khi đang mô phỏng và theo dõi đáp ứng ngay lập tức.

Các linh kiện và mô phỏng

CircuitMaker cung cấp 4 loại linh kiện có thể được sử dụng với các mode mô phỏng khác nhau.

Loại linh kiện

Linh kiện số

Linh kiện tương tự

Linh kiện số/tương tự

Biểu tượng sơ đồ

Sẽ hoạt động trong

Mode mô phỏng số

Mode mô phỏng tương tự

Mode số hoặc tương tự

Không có chức năng

Vì vậy mọi linh kiện số và số/tương tự sẽ hoạt động tốt trong mode số. Để tìm mode mô phỏng dành cho linh kiện, đọc dòng chữ trên biểu tượng trong hộp hội thoại "Lựa chọn linh kiện" hoặc xem trong sách "Thư viện Linh kiện".

Nếu bạn cố mô phỏng một linh kiện bằng mode mô phỏng không dành cho nó, CircuitMaker thể hiện một thông báo cảnh báo và linh kiện đó bị bỏ qua, coi như hở mạch.

Sử dụng bộ mô phỏng logic số

Mô phỏng số là hoàn toàn tương tác tức thời, nghĩa là mạch đáp ứng ngay với các thay đổi kích thích lối vào, và hoạt động của mạch được thể hiện trong thời gian thực như đang hiển thị trên màn hình. Bạn có thể theo dõi hoạt động của mạch theo những cách sau:

- Bật tính năng "Theo dõi" độc đáo của CircuitMaker để đồng thời thể hiện trạng thái của tất cả các nút trên mạch khi đang chạy mô phỏng. Trong mode này, các dây có mức logic 1 được tô màu đỏ, các dây có mức logic 0 được tô màu xanh nước biển, và các dây không rõ trạng thái hoặc có 3 trạng thái khác nhau được tô màu xanh lá cây (các màu này có thể thay đổi được bằng cách chọn View -> Colors).

- Kết nối số lượng không hạn chế các đầu dò của SCOPE với bất kỳ nút nào trên mạch, và sơ đồ thời gian của các nút này được thể hiện riêng biệt trên cửa sổ Waveforms (dạng sóng) số. Các thông tin thời gian này được cập nhật liên tục để thể hiện các thay đổi giống như trong thời gian thực

- Sử dụng Công cụ Đầu dò để dò dây bất kỳ trong mạch cả khi đang chạy mô phỏng lẫn khi đã dừng lại. Các trạng thái logic thể hiện bởi Công cụ Đầu dò cũng có thể được chuyển thành hình vẽ trong cửa sổ Dạng sóng

10.4.2. Các công cụ mô phỏng trong kỹ thuật điện tử số

Một vài nút trong thanh công cụ Toolbar được chỉ định mô phỏng. Đoạn này trình bày các công cụ này.

Chú ý : Những chức năng của các nút này hơi có phần khác với trong chế độ Analog của CircuitMaker.

Nút Digital/Analog. 

Nhấn nút Digital/Analog để chọn mode mô phỏng bạn muốn dùng. Khi biểu tượng cổng AND được hiển thị, bạn đang ở mode số; khi biểu tượng transistor được hiển thị, bạn đang ở mode tương tự.

Nút Reset 

Click vào nút Reset để khởi động lại mô phỏng, ta có thể thiết đặt lại bằng cách chọn Simulation\Reset hoặc nhấn tổ hợp phím Ctrl+Q.

Nút Step 

Click vào nút Step để chạy mô phỏng từng bước một hoặc mô phỏng từng "tick". Cũng có thể chọn : Vào menu Simulation\Step hoặc nhấn F9. Sử dụng tính năng Simulation\Digital Setup để điều chỉnh kích cỡ cho từng bước. Khi ta click vào nút này, thì chạy mô phỏng cho bước một và dừng lại. Lệnh này thuận tiện cho việc gỡ rối một mạch điện, nhất là khi sử dụng phối hợp với nút Trace.

Nút Run/Stop



Biểu tượng Stop Sign sẽ làm quá trình mô phỏng dừng lại.

Click vào biểu tượng Stop Sign để dừng chạy mô phỏng ta có thể chọn:

Biểu tượng Stop thay thế biểu tượng Run khi mà ta nhấn nút Run.

Dữ liệu trước đây thu thập được nối tiếp đến khi kết thúc và cửa sổ phân tích được hiển thị, nhấn vào nút Stop sẽ dừng chạy mô phỏng, đóng cửa sổ phân tích và trở về chế độ soạn thảo sơ đồ.

Công cụ Probe Tool



Sử dụng công cụ Probe Tool để kiểm soát trạng thái của bất cứ điểm (nút) nào trong mạch điện hoặc xem một trạng thái trong điểm (nút) đó. Ta có thể kích hoạt công cụ Probe Tool bằng cách nhấn tổ hợp phím Alt+P hoặc click phải chuột và chọn Probe từ menu thả xuống.

Để xem trạng thái của một điểm (nút), trong khi mô phỏng đang chạy hoặc sau khi nó dừng hẳn ta chạm đầu mũi công cụ Probe Tool trên một dây nối hoặc trên chân linh kiện. Công cụ này hiển thị một trong 3 chữ: H(High), L(Low), P(Power), hoặc không có chữ.

Chọn menu Simulation\Scope Probe sẵn trước để có đồ thị dạng sóng ta xem với công cụ này trong cửa sổ dạng sóng Waveforms.

Để quan sát một trạng thái trong một điểm (nút) ta thực hiện theo từng bước sau :

+ Bước 1: Chạm đầu mũi công cụ Probe trên một dây nối hoặc chân thiết bị.


+ Bước 2: Click trái chuột.

Nút Trace



Nhấn vào nút Trace (hay ấn F11) để bật hay tắt tính năng theo dõi Trace. Dùng Trace để gỡ rối mạch khi gặp trục trặc trong thiết kế mạch. Nó cung cấp một phương tiện giám sát hoạt động của mạch. Trace hiển thị trạng thái của toàn bộ các nút trong mạch khi điện khi đang hoạt động, bằng cách hiển thị các đường dây dẫn ở các màu khác nhau, để chỉ ra trạng thái logic của chúng. Ở trạng thái cao dây có màu đỏ, ở trạng thái thấp dây có màu xanh lam, khi cửa 3 trạng thái ở trạng thái thả nổi hoặc hở mạch dây có màu xanh lá cây.

Lưu ý : Vì các đường dây trong mạch sẽ được vẽ lại mỗi lần nó thay đổi trạng thái, chuyển lựa chọn này lên **on** có thể giảm tốc độ mô phỏng.

* **Nút dạng sóng** : 

Cửa sổ dạng sóng Digital Waveforms.

Bằng cách gán SCOPes đến những điểm cần chú ý trong mạch điện, ta có thể vẽ đồ thị trạng thái của những điểm nút trên khắp thời gian chạy mô phỏng.

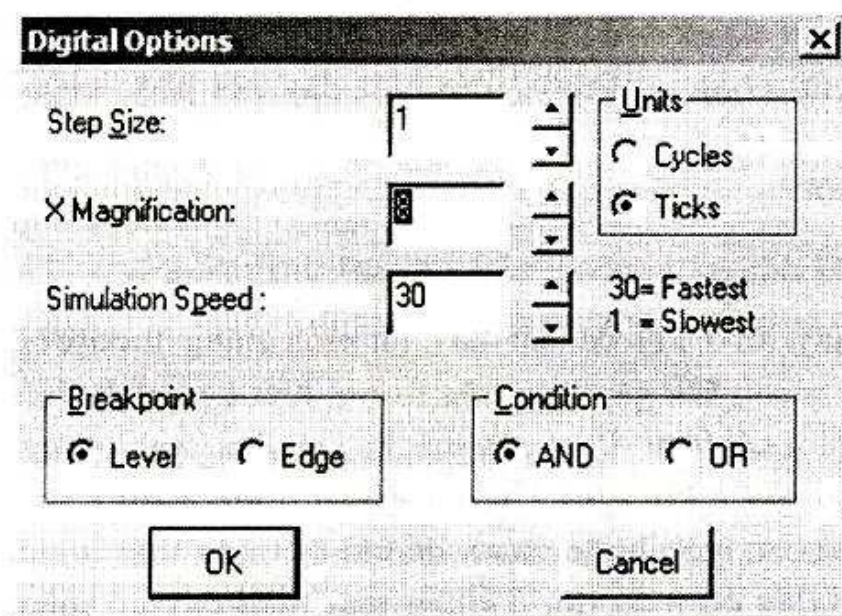
Vào menu Simulation\Display Waveforms hoặc click vào biểu tượng Waveforms.

Trước khi ta có thể nhìn điều chỉnh dạng sóng cho bất cứ điểm nút nào trong mạch điện ta phải nối một SCOPE đến mỗi một điểm nút mà ta muốn kiểm soát.

Hoặc: Vào menu Simulation\Scope Probe để kiểm soát trạng thái của công cụ Probe Tool trong cửa sổ Waveforms

Tùy chọn Digital

Sử dụng hộp thoại Digital Options để điều khiển kích cỡ của một bước khi chạy mô phỏng trong chế độ bước đơn, ngoài ra tùy chọn này để thiết đặt điều kiện cho điểm ngắt và thiết đặt tốc độ mô phỏng.



Hình 10.40: Hộp thoại Digital Option

Dùng hộp đối thoại Digital Option để điều khiển thời gian của một bước khi chạy mô phỏng trong chế độ bước đơn, để đặt các điều kiện cho các điểm dừng và để đặt tốc độ cho mô phỏng.

Chọn Simulation > Digital Option để hiển thị hộp thoại Digital Option hình 10.40.

Bạn có thể định nghĩa thời gian dừng (Stop Size) theo với đơn vị nhịp (Ticks) hoặc hoặc chu kỳ (Cycles). Một chu kỳ thường gồm 10 nhịp. 1 nhịp là đơn vị nhỏ nhất cho độ trễ của mô phỏng số. Thông thường cần 1 nhịp để thực hiện 1 bước đơn trong mô phỏng của mọi thiết bị.

Điều chỉnh độ phóng đại trực X (Adjust X Magnification) để xem phân dạng sóng lớn hơn hay nhỏ hơn ở cửa sổ dạng sóng số Waveforms. Giá trị chuẩn phóng to được đặt là 8. Giá trị nhỏ là Zoomout, giá trị lớn là Zoomin.

Dùng Simulation Speed để điều khiển tốc độ chạy mô phỏng. Điều này là rất cần thiết bởi vì nếu chạy mô phỏng quá nhanh nên không theo dõi được trạng thái hiển thị thành 7 đoạn. Thay đổi giá trị ở trường này bạn có thể thấy được sự thay đổi thành 7 đoạn. Một phương pháp khác để làm chậm quá trình mô phỏng là chạy ở chế độ từng bước đơn hay đặt các điểm ngắt quãng.

Dùng lựa chọn Breakpoint và Conditions cùng lúc ở cửa sổ dạng sóng để đặt điểm ngắt quãng. Bảng dưới đây chỉ ra kết quả của việc đặt cùng lúc 1 số giá trị đó.

Sự kết hợp	Kết quả
Level - And	Mọi điều kiện ngắt phải có trước khi dừng mô phỏng.
Level - On	Bất kỳ 1 trong các điều kiện ngắt xảy ra sẽ dừng mô phỏng.
Edge - And	Mô phỏng dừng khi vạch thăm dò xuất hiện trên toàn bộ dạng sóng chỉ định.
Edge - On	Mô phỏng dừng nếu 1 sự chuyển tiếp đến bất kỳ một điều kiện xác định rõ nào xuất hiện.

Dùng hộp kiểm tra điểm dừng check-box breakpoints trên cửa sổ dạng sóng số Digital Waveforms để đặt điểm dừng trên 1 mạch.

Để đặt một điểm dừng :

1. Nhấn 1 lần lên hộp kiểm tra điểm dừng ở phía trái của nhãn Scope trên cửa sổ dạng sóng để điền lên đáy phân hình chữ nhật (CN) chỉ điều kiện dừng 0 (hình10.41).



Hình 10.41

2. Nhấn 2 lần để điền lên đỉnh phân chia của hình CN để chỉ điều kiện dừng 1 (hình 10.42)



Hình 10.42

3. Nhấn 3 lần để chuyển diện tích sang trạng thái rỗng chỉ ra là không có điều kiện dừng.

10.4.3. Các điểm kiểm tra hoạt động của mạch điện tử Test Points

Test points là gì?

Trong CircuitMaker, Test Points là những điểm được gán trong mạch điện. Những điểm này có nhiệm vụ tập trung dữ liệu mô phỏng và báo cho CircuitMaker biết. Ta không phải thiết đặt nó trước khi chạy mô phỏng (CircuitMaker thực hiện một cách tự động hoá). Test Points xác định các dữ liệu và ghi lại thời điểm làm việc vào một file có phần mở rộng là RAW, chúng xác định các biến số mà biến này được hiển thị trong cửa sổ phân tích khi chạy mô phỏng. Mặt khác, Test Points xác định các biến đổi và hiển thị sự biến đổi này trong cửa sổ phân tích khi chạy mô phỏng lần nữa.

Chú ý : Quá nhiều Test Points trong mạch sẽ làm cho kết quả mô phỏng sẽ chậm lại.

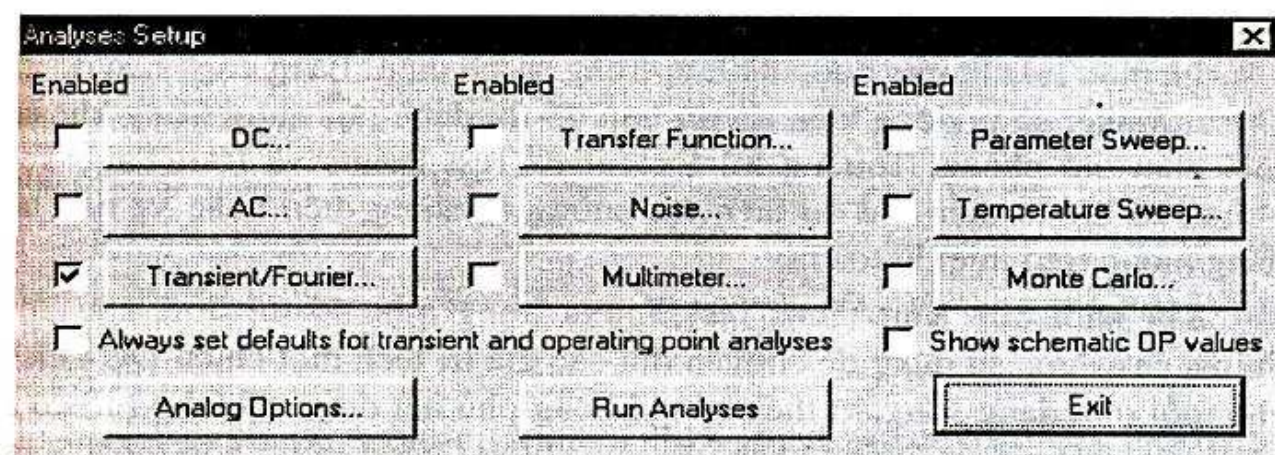
Các kiểu Test Points.

Có 3 kiểu Test Points được dùng trong CircuitMaker.

+ **Default Test Points** : Đây là chế độ ngầm định. CircuitMaker sẽ mặc định xếp đặt những Test Points này trên mạch điện, chế độ này tuân thủ theo các thiết đặt ở hộp thoại **Analog Option**. Ta không nhìn thấy trên mạch điện.

CircuitMaker sẽ ngầm định xếp đặt những Test Points này trên mạch điện. Để sử dụng Test Points này ta dùng Probe Tool click trái chuột trên một số dây nối hoặc các thiết bị đo lường như : điện áp, dòng điện hoặc nguồn điện.

Default Test Point được thiết đặt trong **Analog Option** (hình 10.43).



Hình10.43: Cửa sổ thiết lập các thiết bị ảo dùng trong mô phỏng

+ **Executive Test Points** : Đây là chế độ do người sử dụng chọn lựa mà đặt các Test Points vào trong mạch điện của mình từ các thiết bị đo áp, đo dòng, hoặc đo công suất. Ở chế độ Exclutive Test Points này các Default Test Points bị mất tác dụng. Dữ liệu thu thập trong Exclutive Test Points này chỉ tập trung duy nhất cho mỗi Exclutive Test Points. Ta có thể thiết đặt các Exclutive Test Point trên đường nối từ các điểm nút (điểm nối các đường dây, điểm có hình dấu chấm hạt đậu) của thiết bị, linh kiện hoặc chính sự tiêu tán công suất trên thiết bị, linh kiện. Tuy nhiên, một số thiết bị có mạch phụ thì dữ liệu không tập hợp được kết quả.

Để thiết đặt các Test Points này ta thực hiện từng bước sau :

Bước 1: Dừng chạy chế độ mô phỏng.

Bước 2: Chọn công cụ Probe từ Toolbar.

Bước 3: Nhấn, giữ Ctrl và click chuột vào những vị trí phù hợp trong mạch.

Để thiết đặt nhiều **Executive Test Points** trong mạch ta thực hiện như sau:

Bước 1: Dừng chạy chế độ mô phỏng.

Bước 2: Giữ phím Shift và phím Ctrl trong khi đó click trái chuột.

Bước 3: Huỷ các Exclutive Test Points ta thực hiện như sau:

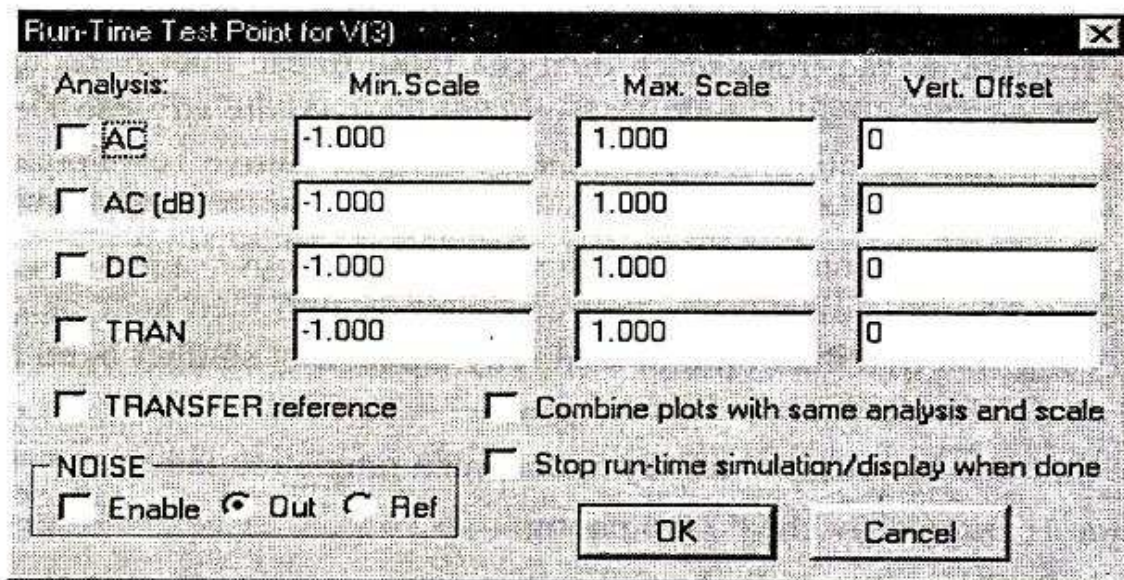
Dùng Probe Tool : Nhấn và giữ phím Ctrl trong khi đó click chuột lên vùng trắng bất kỳ của cửa sổ làm việc.

Chú ý: Khi ta huỷ bỏ thiết đặt chế độ Exclutive Test Points thì Default Test Points làm việc trở lại.

+ *Run-Time Test Points* :

Là một Test Points được định vị trong mạch điện. Bằng cách nào đó mà CircuitMaker sẽ tự động hiển thị dữ liệu đồ thị được tập trung trong khoảng thời gian mô phỏng. Thường thì Executive Test Points và Run-Time Test Points có thể là điểm để đo điện áp, cường độ dòng điện hoặc sự tiêu tán năng lượng trên thiết bị toả ra.

Khi phối hợp Run-Time Test Points trong một đồ thị đơn, các Test Points này được sử dụng để chuyển đổi các giá trị lệch một chiều DC Offset cho mỗi một dạng sóng, vì thế chúng không phủ hết được. Chính vì lẽ đó ta nên nhập một giá trị Offset khác vào vùng Vert.Offset trong hộp thoại Run-Time Test Points. Vùng Offset dành này cho mỗi một Run-Time Test Points.



Hình 10.44

Để thiết đặt các Test Points này ta thực hiện từng bước như sau:

Bước 1: Dừng chạy chế độ mô phỏng.

Bước 2: Chọn công cụ Probe từ Toolbar.

Bước 3: Click trái chuột vào những vị trí phù hợp trong mạch.

Hộp thoại **Run-Time Test Points** xuất hiện như hình 10.44.

Bước 4: Ta hãy thiết lập các giá trị vào các vùng

+ Min.Scale : Tỷ lệ nhỏ nhất có thể cho phép

+ Max.Scale : Tỷ lệ lớn nhất có thể cho phép

+ Vert.Offset : Độ lệch dọc của đại lượng điện (độ trôi).

Bước 5: Đánh một dấu vào check-box "*Combine plots with same analysis and scale*" để phối hợp các Run-Time Test Points lại. Kết quả là ta có được các dạng sóng sẽ hiển thị trong cùng một đồ thị trong suốt thời gian mô phỏng.

Bước 6: Nhấn OK.


Để thiết đặt nhiều Run-Time Test Points trong mạch ta thực hiện như sau:

- + Nhấn giữ phím Shift trong khi click chuột vào những điểm phù hợp.
- + Huỷ các Run-Time Test Points ta thực hiện như sau:

Dùng Probe Tool : Click trái chuột lên vùng trắng bất kỳ của cửa sổ làm việc.

10.4.4. Multimeter

Được thêm vào cửa sổ phân tích một công cụ đo Multimeter/Operating Point. Cũng như Electric Work Bench, CircuitMaker cung cấp kèm theo cho chúng ta một bộ công cụ đa năng dành cho việc đo lường trở kháng, DC, DC AVG (dòng điện hoặc điện thế trung bình một chiều) hoặc AC RMS (dòng điện hoặc điện thế xoay chiều hiệu dụng). Ta có thể đặt nhiều đồng hồ chức năng vào mạch. Khi ta chạy mô phỏng, giá trị đo lường sẽ hiển thị trên máy đo. Để lấy đồng hồ MULTIMETER này ta thực hiện tuần tự từng bước như sau:

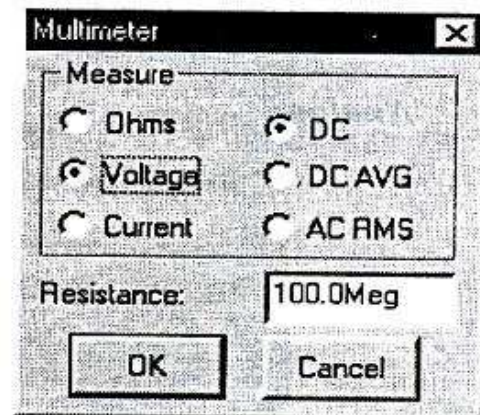
Bước 1 : Nhấn nút lệnh 

Bước 2: Mục Major Device Class chọn Instruments\Minor Device

Class chọn Analog\Device Symbol chọn Multimeter và Place.

Bước 3: Định vị thiết bị trên vùng làm việc.

Ý nghĩa các nút Radio button (hình 10.45):



Hình10.45

- + Ohms : Đo trở kháng.
- + Voltage : Đo điện áp.
- + Current : Đo dòng điện.
- + DC : Đo giá trị điện một chiều
- + DC AVG : Đo các đại lượng điện trung bình một chiều.
- + AC RMS : Đo các đại lượng điện hiệu dụng xoay chiều.
- + Vùng Resistance : Dùng để nhập giá trị trở kháng.

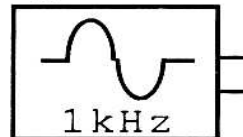
Chú ý : Để đo lường các giá trị DC AVG (dòng điện hoặc điện thế trung bình một chiều) hoặc AC RMS (dòng điện hoặc điện thế hiệu dụng xoay chiều) chức năng Transient Analysis phải được kích hoạt và phải mô phỏng đủ chu kỳ ngắn của dữ liệu thì công việc đo lường mới có nghĩa. Cũng như vậy, chức năng Operating Point Analysis (Multimeter) phải được kích hoạt để có các giá trị trở kháng và các giá trị DC.

Để các thiết bị đo không làm sai lệch các giá trị của dòng điện chạy trong mạch điện, trở kháng của vôn kế phải rất lớn và trở kháng của một ampe kế phải rất bé, như vậy thiết bị đo sẽ tác động ít trên mạch điện khi đo lường. Khi đo điện áp trên mạch điện có trở kháng cao, ta có thể gia tăng trở kháng của vôn kế theo nhu cầu . Khi đo dòng điện qua một mạch trở kháng thấp có thể giảm trở kháng của ampe kế. Khi đo trở kháng mới nối, yêu cầu bắt buộc gia tăng dòng điện của ôm kế.

10.4.5. Signal generator & Pulse

Analog:

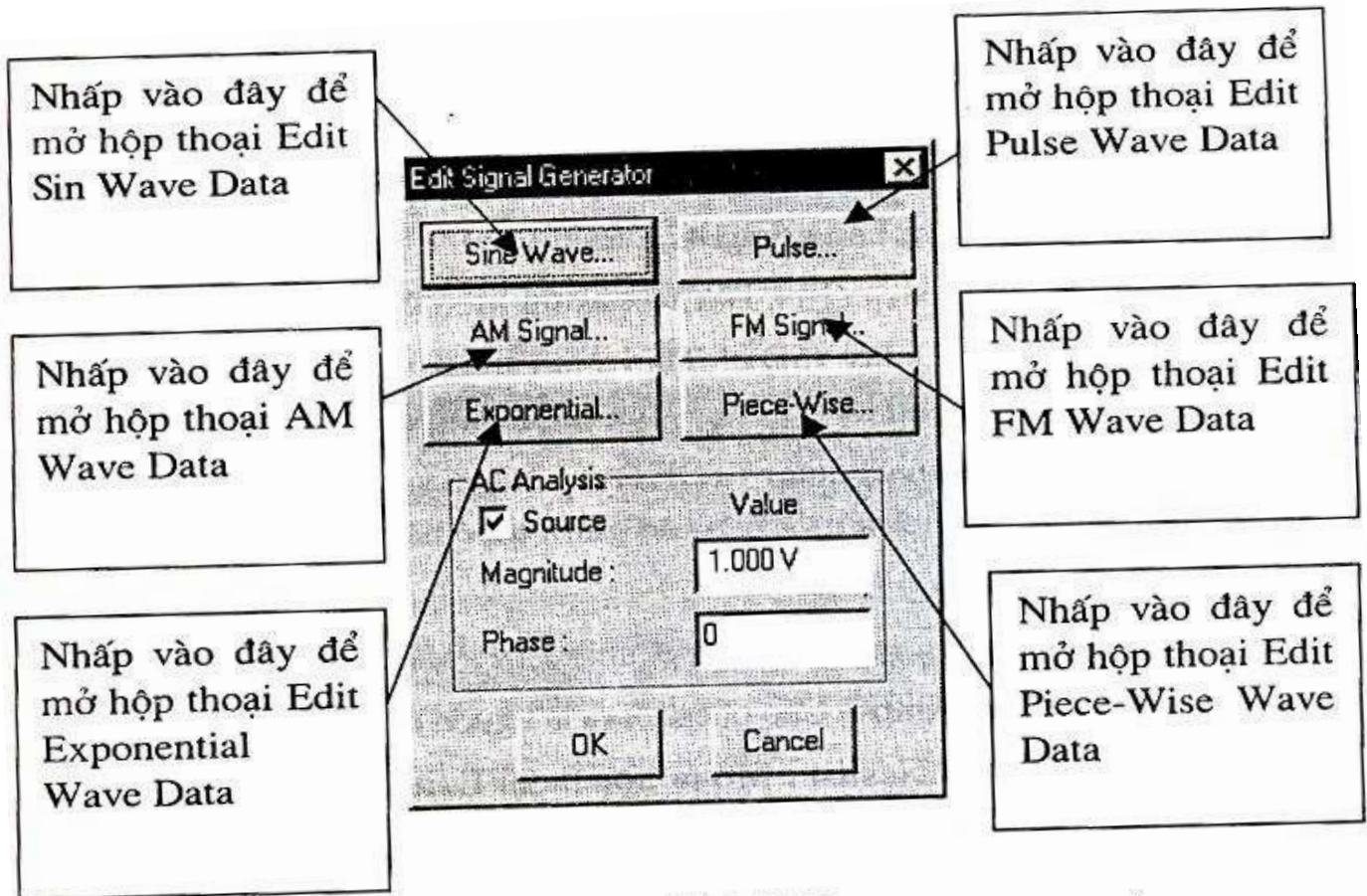
Signal Gen



Hình 10.46: Máy phát tín hiệu

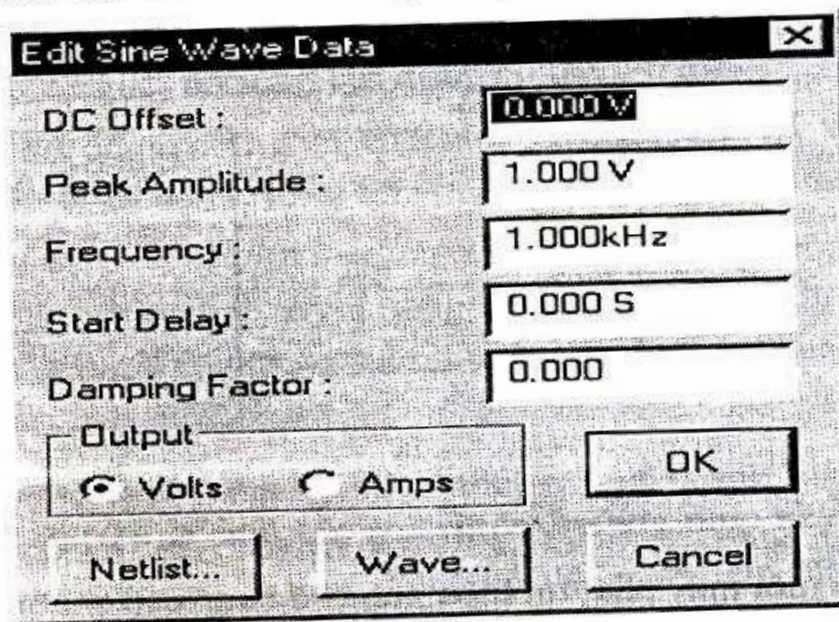
Để có thể lựa chọn dạng tín hiệu đầu ra máy phát ta thực hiện như sau:

Bước 1: Nhấp đúp trái chuột vào thiết bị \ nhấn Wave... để mở hộp thoại Edit Signal Generator hình 10.47.



Hình 10.47

Bước 2 : Nhấn Sin Wave.. ta có hộp thoại Edit Sin Wave Data như sau :



Hình 10.48: Hộp thoại Edit Sine Wave Data

Ý nghĩa các vùng nhập số liệu và các nút radio button như sau:

+ DC Offset (VO) : Điều chỉnh thiên áp DC

+ Peak Amplitude (VA) :

Biên độ đỉnh ở lối ra bao gồm DC Offset (đơn vị vôn hoặc ampe).

+ Frequency (F) : Tần số của tín hiệu ở lối ra (Hz).

+ Start Delay (SD) : Thời gian trễ truyền tại thời điểm khởi phát chu kỳ đầu khi xuất hiện sóng Sin.

+ Damping Factor (THETA) : Nhập vào giá trị dương thì kết quả cho ra một biên độ giảm theo hàm mũ.

Hệ thức: $V(T_0, T_{SD}) = VO$

$$V(T_{SD}, T_{STOP}) = VO + VA \sin(2\pi f(t-SD))e^{-(t-SD)THETA}$$

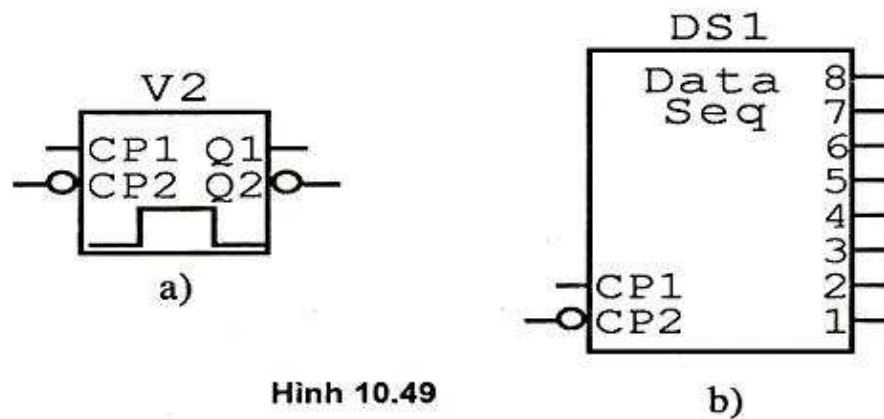
+ Volts/Amps : Cho phép lối ra là nguồn thế hay nguồn dòng.

+ Button Neslist : Mở ra hộp thoại Edit Device Data.

+ Wave... : Mở ra hộp thoại Edit Signal Generator.

Các thiết bị số Digital Instruments

Trong phần này giới thiệu hai dụng cụ kỹ thuật số là Pulser (hình 10.49a) và Data sequencer (hình 10.49b).



Hình 10.49

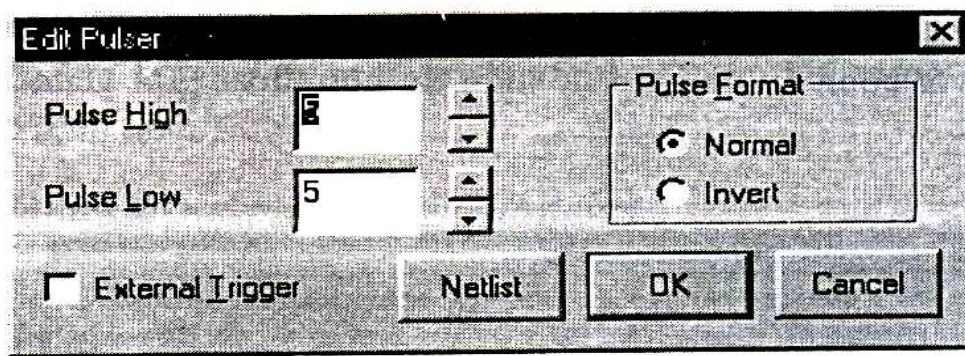
Pulser

Pulser là một máy phát tín hiệu xung số cung cấp chuỗi xung có biên độ có 2 mức điện áp cao - thấp liên tục. Dạng xung, time high, time low, và chế độ trigger riêng biệt có khả năng điều khiển được bằng các chỉ lệnh để có

thể thay đổi phù hợp với các yêu cầu của người dùng cho mỗi xung trong mạch điện.

Để hiệu chỉnh thiết lập các tham số của xung của máy phát xung Pulser chúng ta thực hiện theo từng bước sau:

Bước 1 : Double-click vào Pulser với công cụ mũi tên Arrow Tool để hiển thị hộp thoại *Edit Pulser* (hình 10.50).



Hình 10.50

Bước 2 : Thay đổi số ticks mô phỏng cho xung mà nó sẽ lưu lại mức cao và mức thấp, dạng thức của xung (thường hoặc đảo), và có xung hay xung là xung được chạy tự do hoặc chế độ kích đồng bộ từ xung bên ngoài (External Trigger).

Bước 3 : Đánh dấu vào check-box External Trigger để sử dụng Pulser như là một mạch đơn hài độ rộng của xung có thể lập trình.

Trong chế độ đồng bộ ngoài External Trigger, các lối vào xung nhịp CP1 và CP2 được hoạt động đồng bộ với sườn lên và sườn xuống của xung kích ở lối vào.

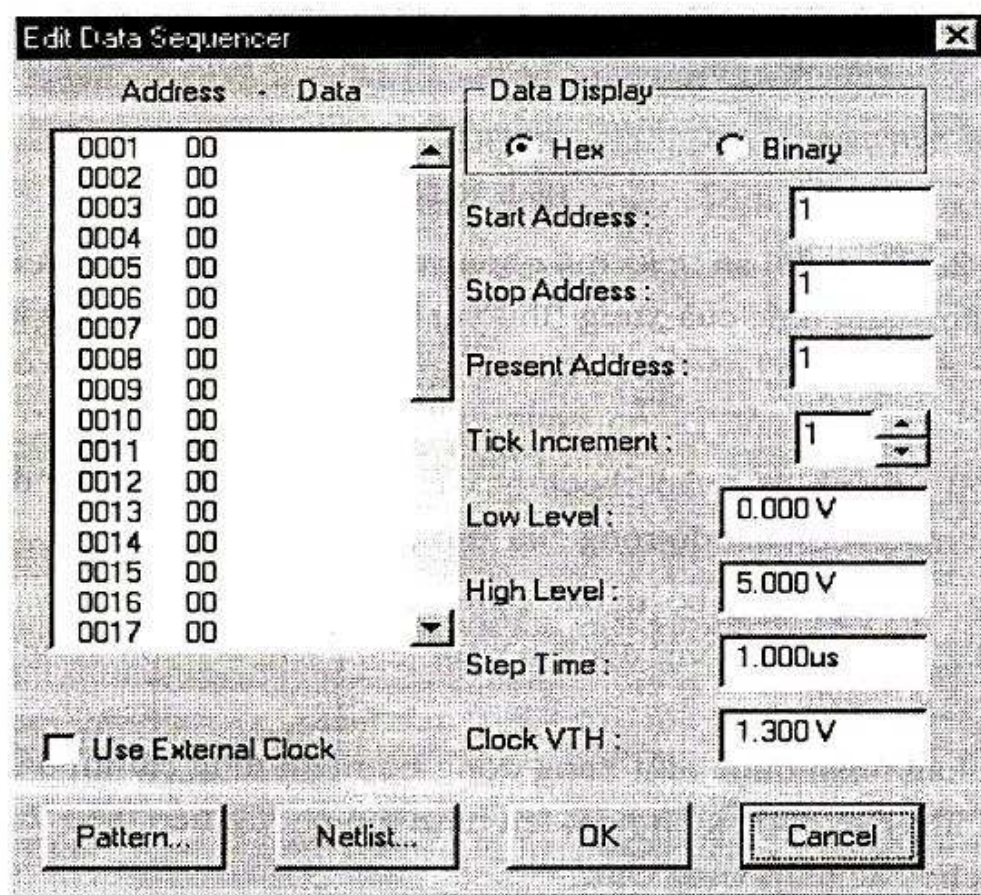
Nếu chân nhận được một xung đồng bộ riêng từng cái thì đầu ra của bộ tạo xung Pulser sẽ được kích hoạt lên nhịp mô phỏng tiếp theo và vẫn giữ sự kích hoạt cho các nhịp xung cao.

Nếu có cả hai xung đồng bộ ngoài đưa vào CP1, CP2 của bộ tạo xung, xung lối ra của Pulser sẽ chịu tác động trên 2 xung nhịp này, xung phát ra trong điều kiện mô phỏng này sẽ đồng bộ với sườn lên của CP1 và sườn xuống của xung đưa vào CP2. Ta có thể kích lại Pulser trong chế độ này bằng cách thêm vào các xung kích hoạt mới.

Data sequencer

Ta có thể sử dụng Data Sequencer thiết bị này cả trong chế độ mô phỏng kỹ thuật số và mô phỏng tương tự. Ngoài ra nó cũng còn được gọi là Data hoặc Word Generator, nó cho phép ta chỉ định tối đa tới 32767 từ 8 bit mà các từ này có thể xuất trong một trình tự định trước.

Bởi do không hạn chế về số Data Sequencers mà ta có thể sử dụng nó trong một mạch điện, ta có thể đặt song song vài cái Data Sequencers để tạo một dòng dữ liệu cho bất cứ chiều rộng nào. Double-click vào dụng cụ Data Sequencers với công cụ mũi tên để hiển thị hộp thoại Edit Data Sequencers (hình 10.51).



Hình 10.51

+ Start Address : là địa chỉ của dữ liệu lối ra trước tiên khi mô phỏng bắt đầu.

+ Stop Address : là địa chỉ của dữ liệu lối ra cuối cùng trước khi lặp lại chuỗi dữ liệu.

+ Use External Clock : Dùng check-box tạo xung đồng hồ vào CP1 và CP2 có sườn lên và sườn xuống riêng biệt. Nếu các xung đồng hồ đưa vào chân này bị khoá thì Data Squencer tiến đến địa chỉ kế tiếp.

Chỉ dành cho mô phỏng chế độ Digital

+ Lựa chọn Present Address : Chỉ ra địa chỉ để xuất dữ liệu ra tiếp theo nếu mạch điện không xác lập lại hoặc sửa đổi.

+ Sửa đổi Tick Increment : Cho biết có bao nhiêu nhịp mô phỏng (ticks) xảy ra trước khi lối ra tiến hành tới địa chỉ kế tiếp, khi xung đồng hồ bên ngoài bị vô hiệu hoá.

Chỉ dành cho mô phỏng chế độ Analog

+ Low Level và High Level : Chỉ ra các mức điện áp ở lối ra.

+ Step Time : là khoảng thời gian mà lối ra giữ nguyên tại mỗi địa chỉ, khi xung đồng hồ bên ngoài bị vô hiệu hoá.

+ Clock VTH : là mức điện áp ngưỡng tại các chân có xung đồng hồ bên ngoài tác động làm cho lối ra tiến đến địa chỉ kế cận.

Biên soạn mẫu (Pattern Editor)

Ta có thể đưa dữ liệu trực tiếp vào Data list box trong hộp thoại Data Sequencer. Tuy nhiên, khi tạo một mẫu lớn thì phương pháp này tốn nhiều thời gian. Chọn phím **Pattern** trong hộp thoại Data Sequencer. Patter Editor giúp ta tạo các mẫu lớn và phức tạp một cách nhanh chóng.

Trường **Increment** cho biết có bao nhiêu hàng sẽ chứa cùng một dữ liệu trước khi có sự thay đổi tiếp theo trong mẫu. Ví dụ, với Increment là 3, mẫu **Shift 1 Left** sẽ dịch mọi hàng mẫu thứ 3.

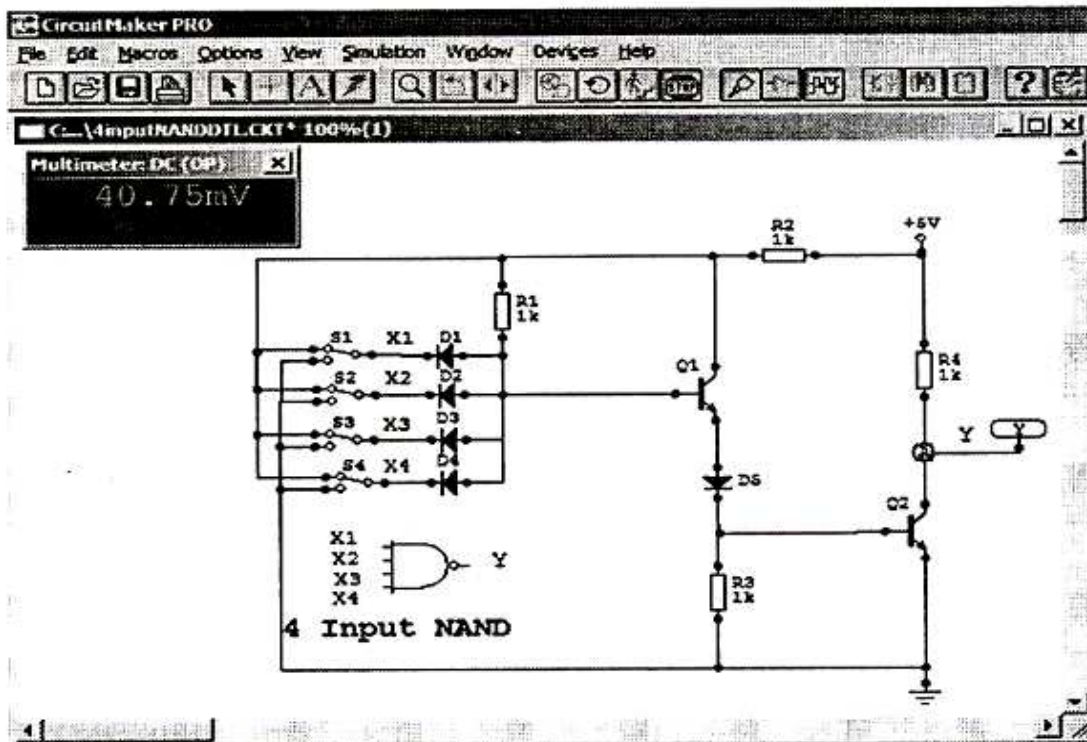
Ta có thể đặt kích cỡ mẫu tối đa cho mỗi Data Sequencer. Điều này giúp cho ta tạo được các mẫu nhỏ hơn cho một số Data Sequencer mà không cần một số lượng lớn bộ nhớ hay các file lớn. **Max.number of pattern lines**, theo mặc định được đặt là 32, nhưng khi cần có thể tăng lên tận 32767. Nó không bao giờ nhỏ hơn Stop Address. Khi ta tăng số hàng mẫu cực đại, các hàng khác sẽ bị điền đầy các số 0. Nếu ta giảm số hàng mẫu cực đại, bất kỳ dữ liệu nào được chứa trong các dữ liệu ở trên sẽ bị mất vĩnh viễn.

10.5. THIẾT KẾ MÔ PHỎNG CÁC VI MẠCH LÔGIC, CÁC MẠCH ĐIỆN TỬ SỐ

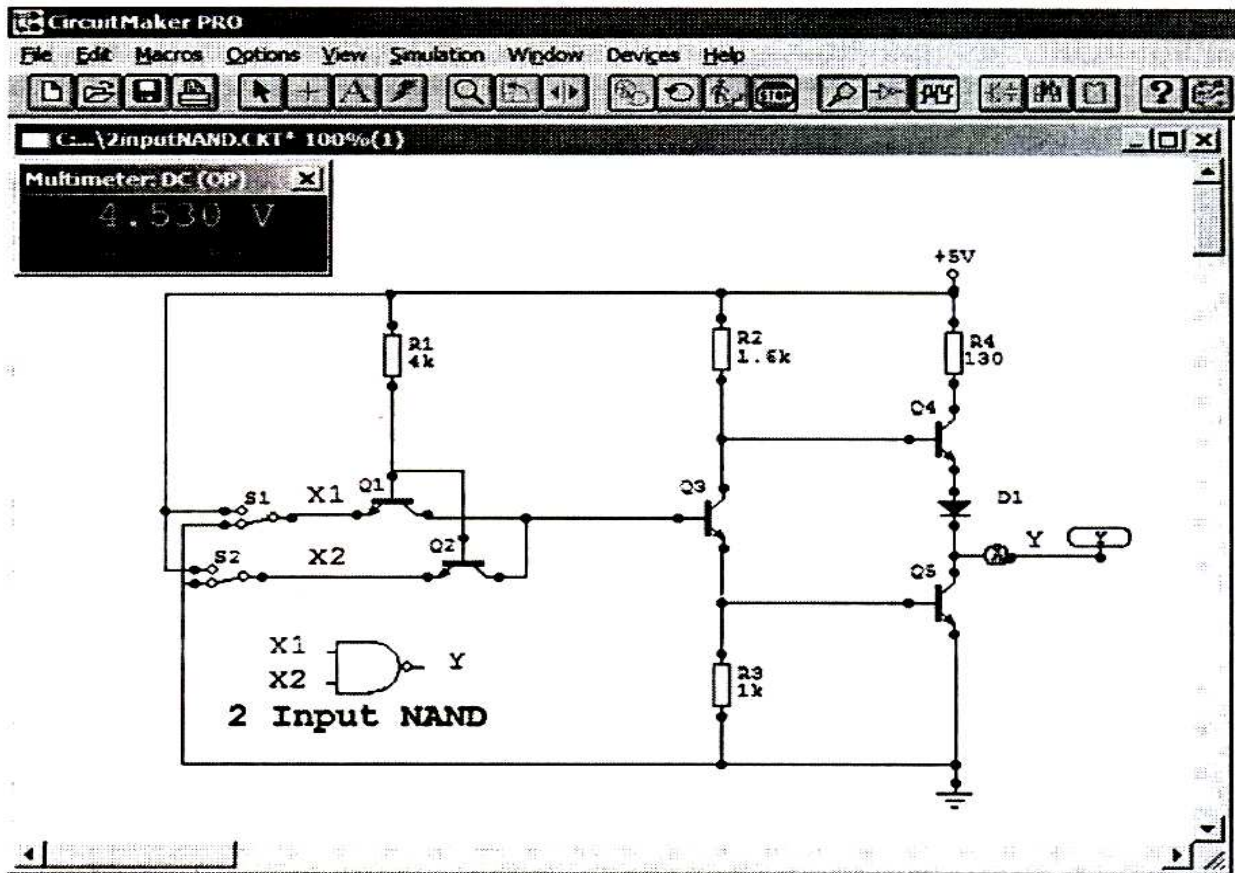
Dùng phần mềm CircuitMaker ta có thể thực hiện thiết kế mô phỏng các vi mạch logic. Đo kiểm tra điện áp các điểm trong sơ đồ điện tử, đo mức điện áp ra của các mạch logic. Các hình 10.52, hình 10.53, hình 10.54, hình 10.55, hình 10.56, là các giao diện màn hình thí nghiệm mô phỏng các vi mạch thuộc họ logic DTL, TTL, CMOS. Các thí nghiệm mô phỏng này có thể giúp sinh viên nắm bắt nhanh nội dung kiến thức trình bày trong chương 3.

Ta cũng có thể dùng phần mềm này nghiên cứu thiết kế mô phỏng các mạch lô gíc tổ hợp, các mạch tính toán số học, các bộ hợp kênh, phân kênh, các bộ biến đổi mã, giải mã. Sơ đồ các thí nghiệm mô phỏng này được trình bày trên các hình 10.57, hình 10.58, hình 10.59, ..., hình 10.69. Các thí nghiệm mô phỏng này có thể giúp sinh viên nắm bắt nhanh nội dung kiến thức trình bày trong chương 4.

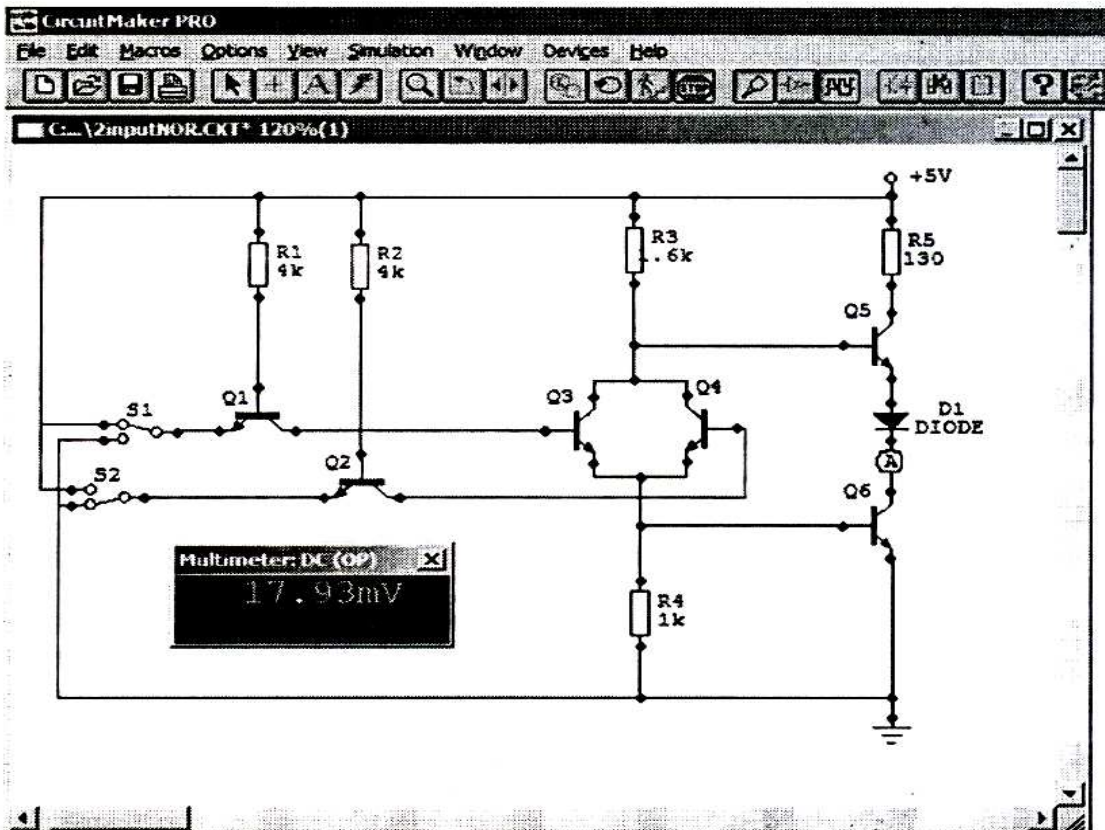
Các bộ đếm nhị phân, đếm 10 mã BCD, đếm vòng theo kiểu ghi dịch mã Johnson. Sơ đồ các thí nghiệm mô phỏng này được trình bày trên các hình 10.69, hình 10.70, ..., hình 10.76. Các bài thí nghiệm mô phỏng này minh họa cho nội dung chương 5 và 6.



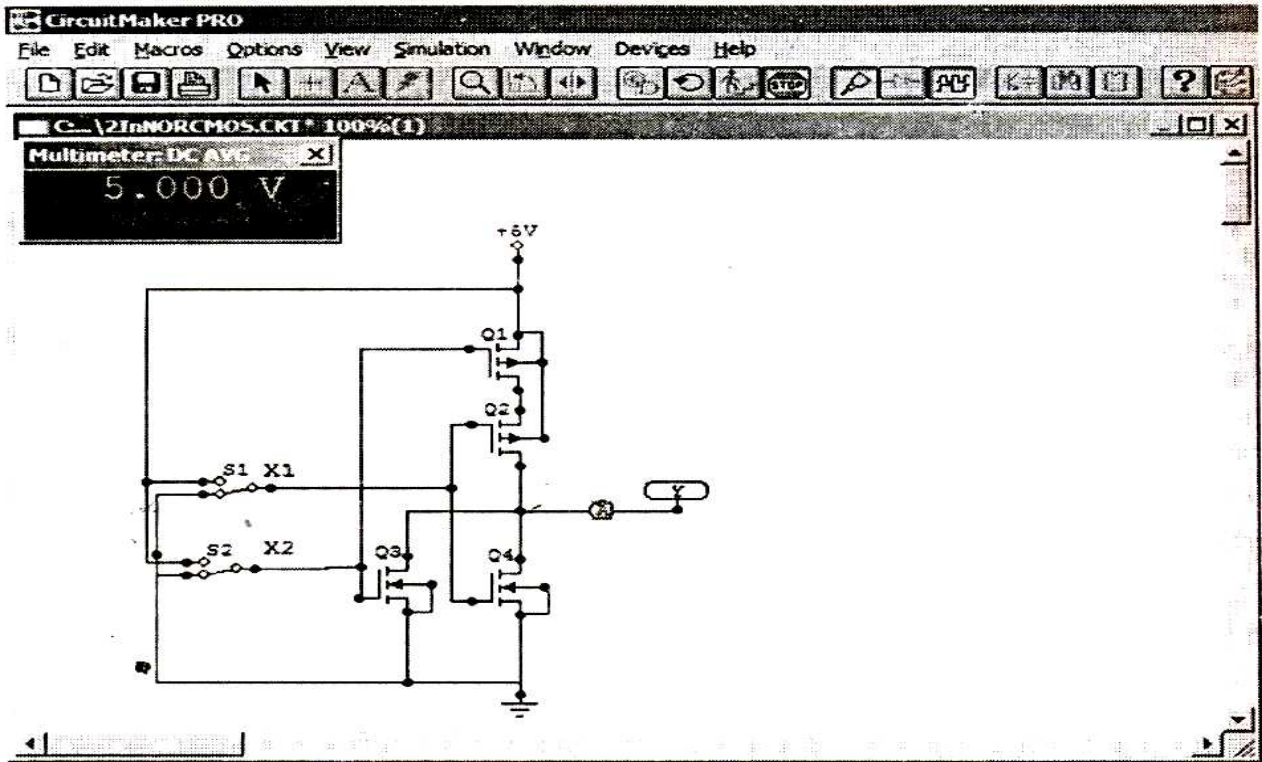
Hình 10.52: Thiết kế mô phỏng vi mạch NAND 4 lối vào thuộc họ logic DTL



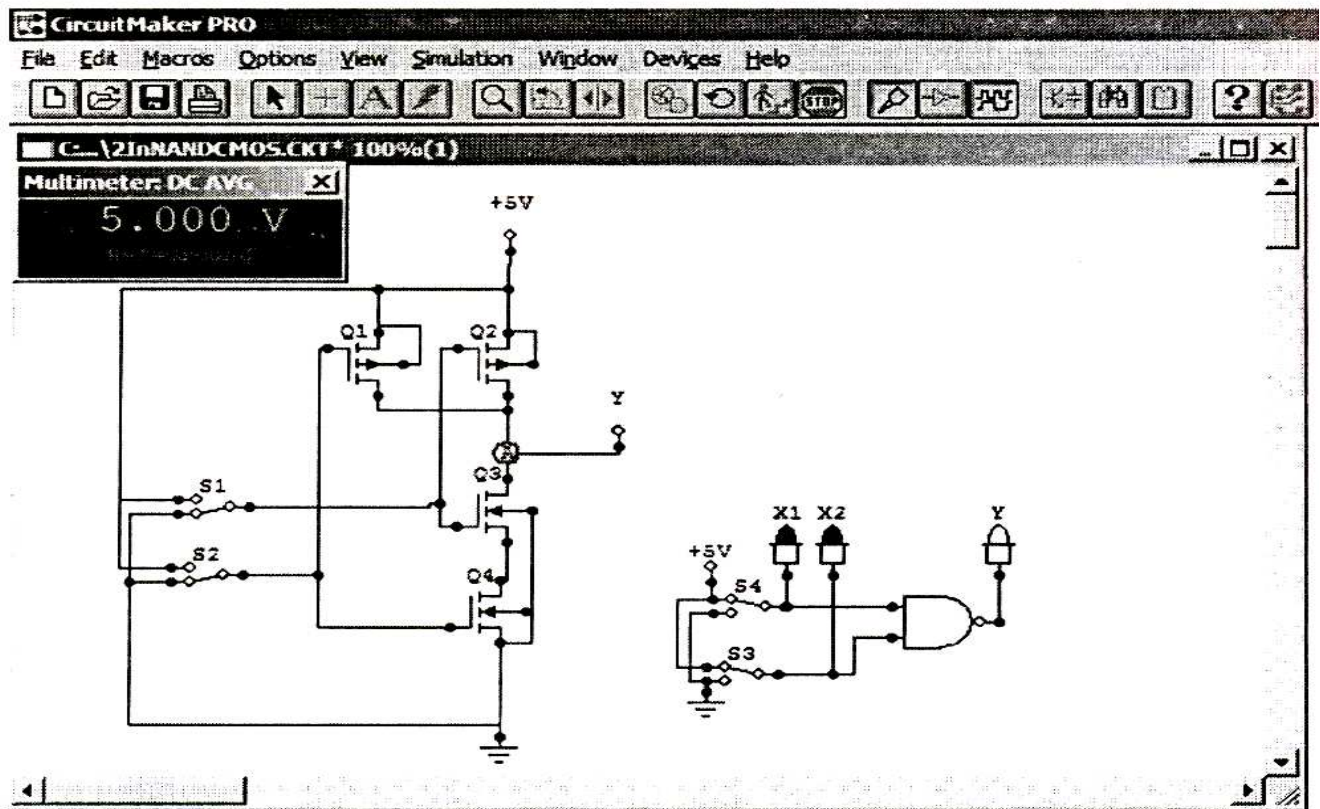
Hình 10.53: Thiết kế mô phỏng vi mạch NAND 2 lối vào họ logic TTL



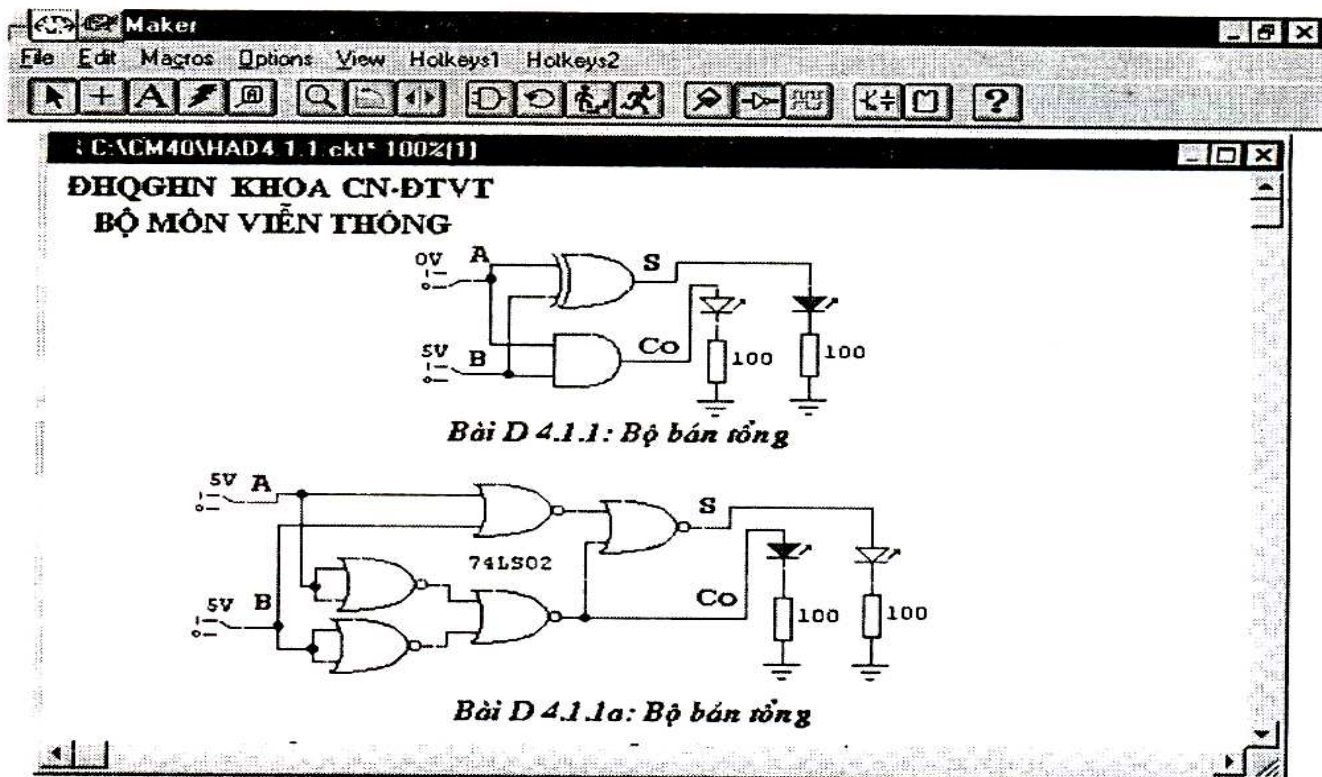
Hình 10.54: Thiết kế mô phỏng vi mạch logic NOR 2 lối vào thuộc họ logic TTL



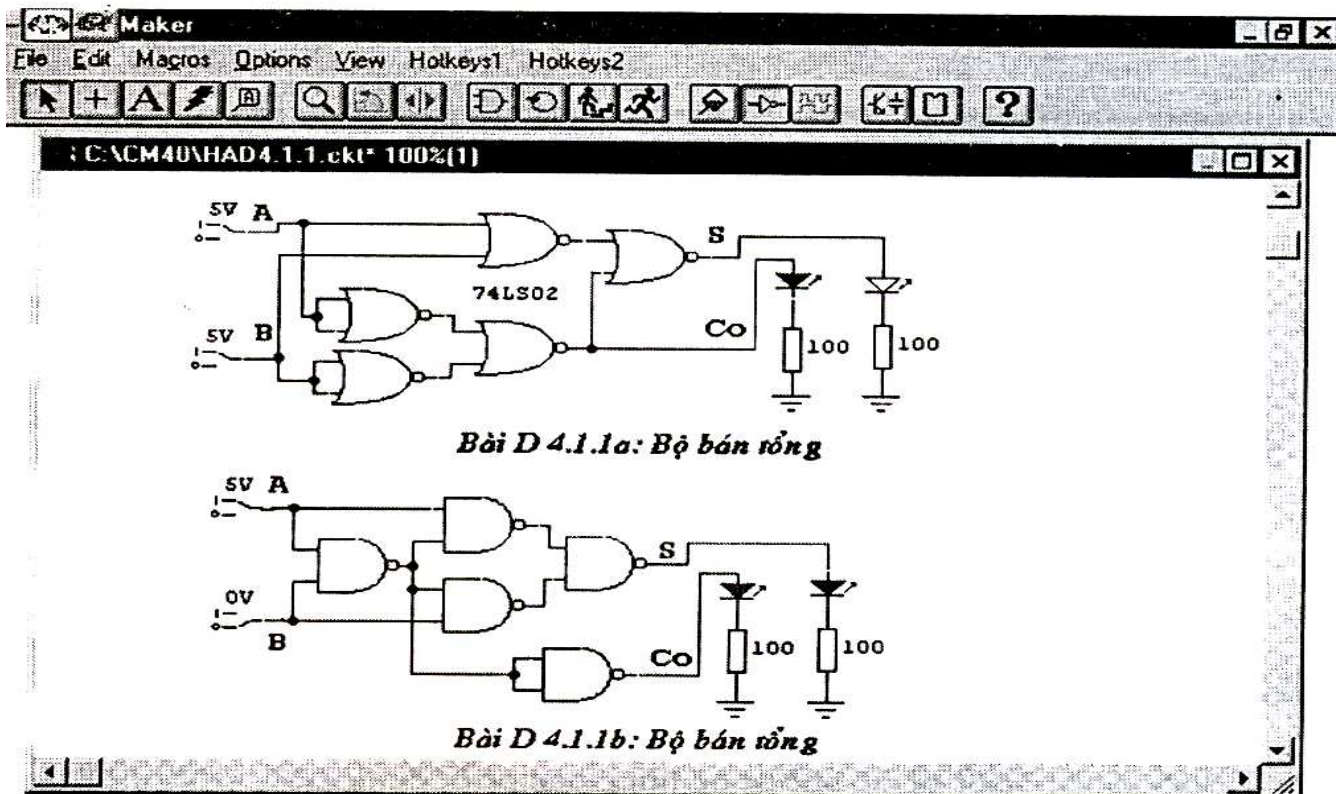
Hình 10.55: Thiết kế mô phỏng vi mạch NOR 2 lối vào họ logic CMOS



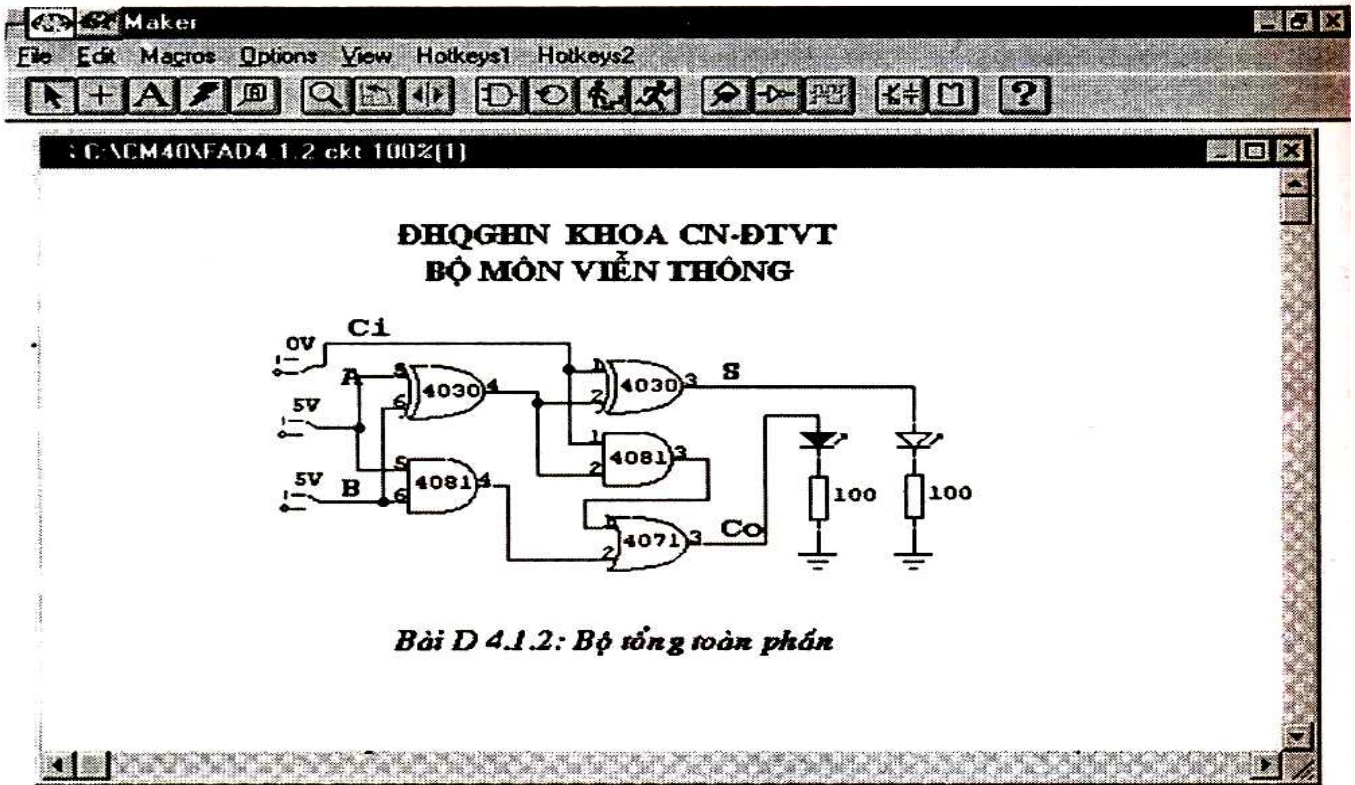
Hình 10.56: Thiết kế mô phỏng vi mạch NAND 2 lối vào họ logic CMOS



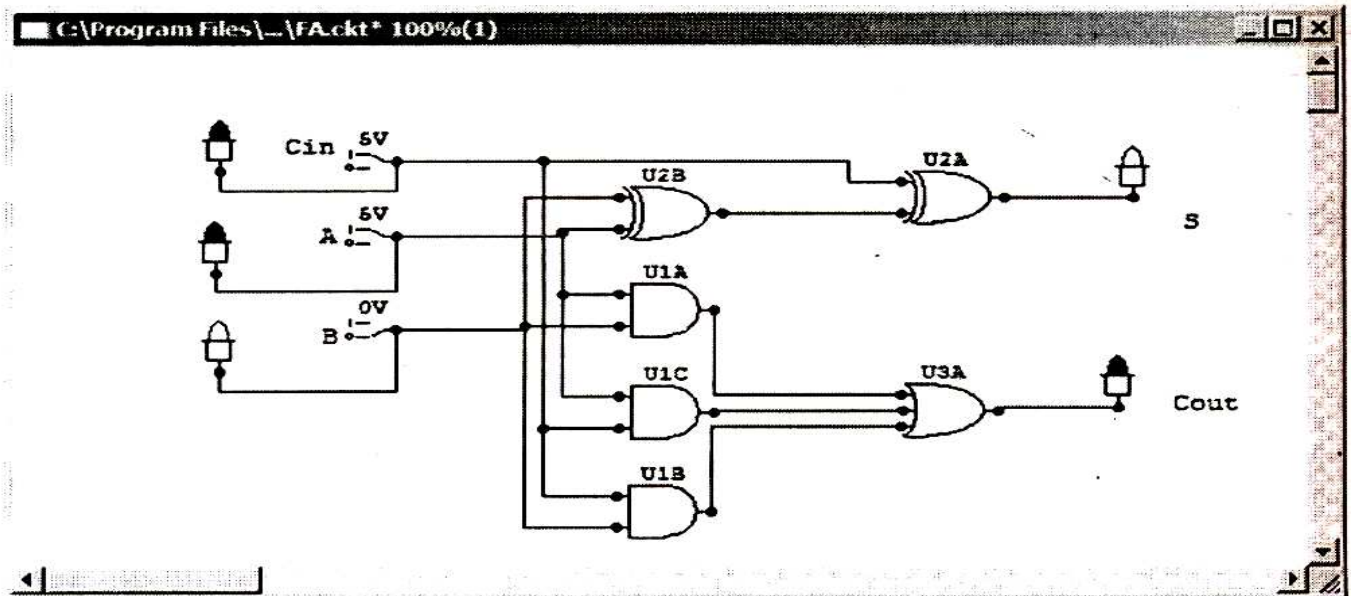
Hình 10.57: Thiết kế mô phỏng bộ bán tổng dùng các mạch XOR, AND, NOR



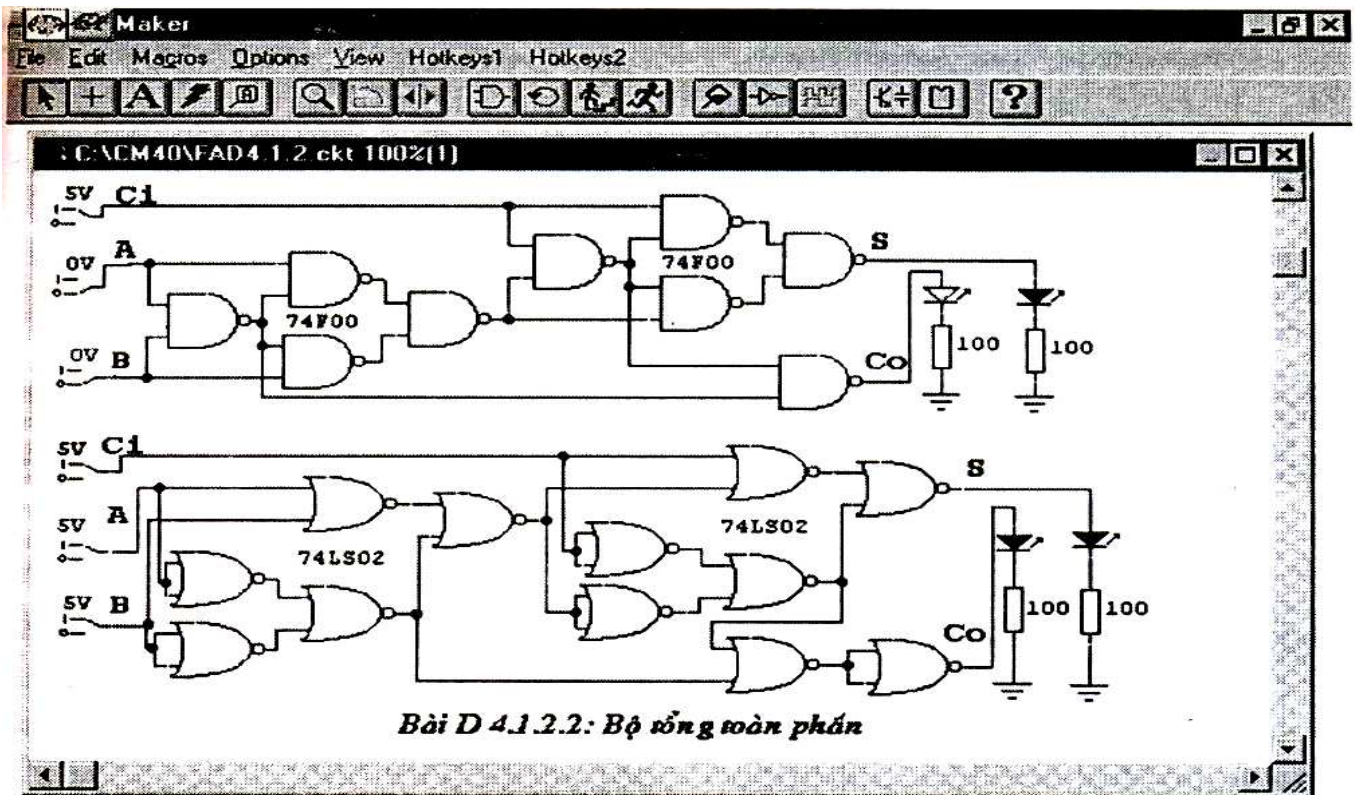
Hình 10.58: Thiết kế mô phỏng bộ bán tổng dùng các mạch NAND



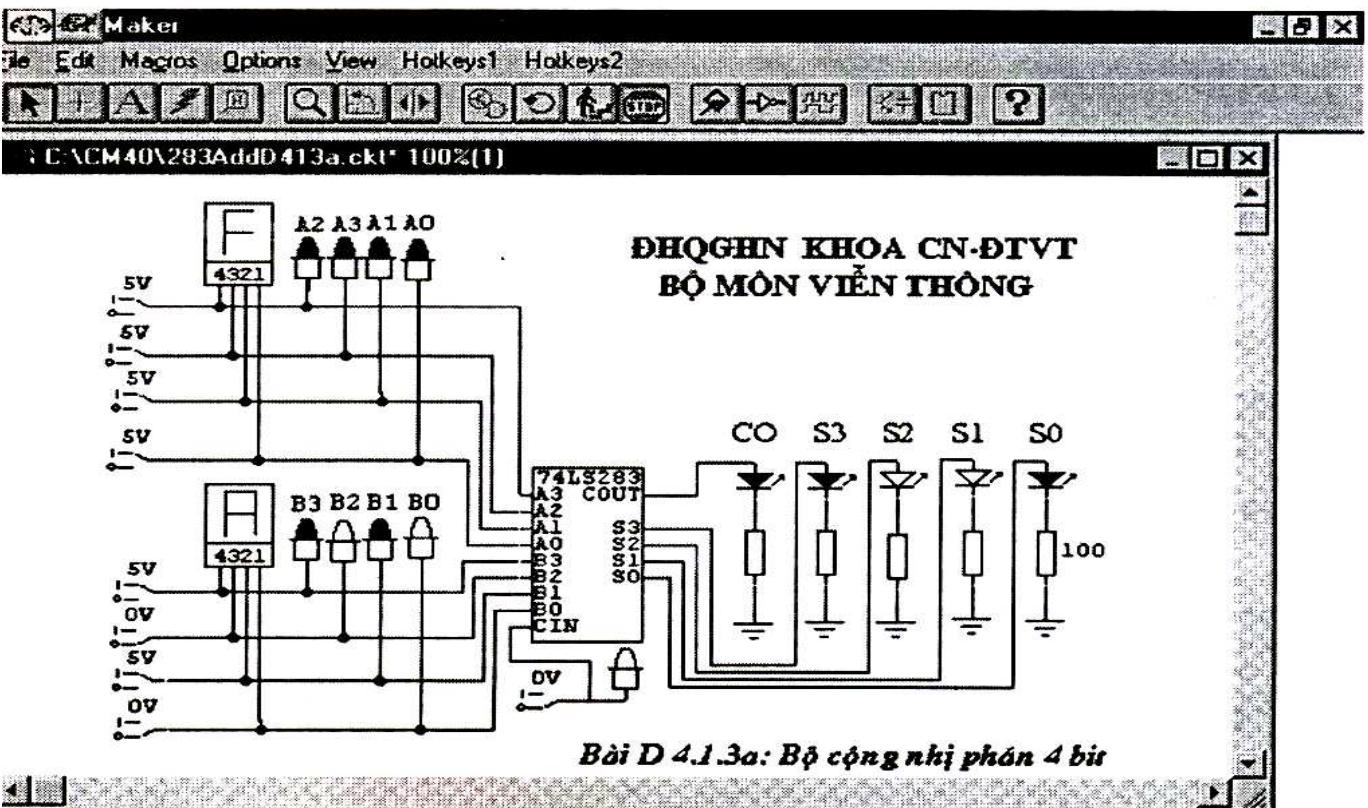
Hình 10.59: Thiết kế mô phỏng bộ tổng toàn phần Full Adder dùng XOR, AND, OR



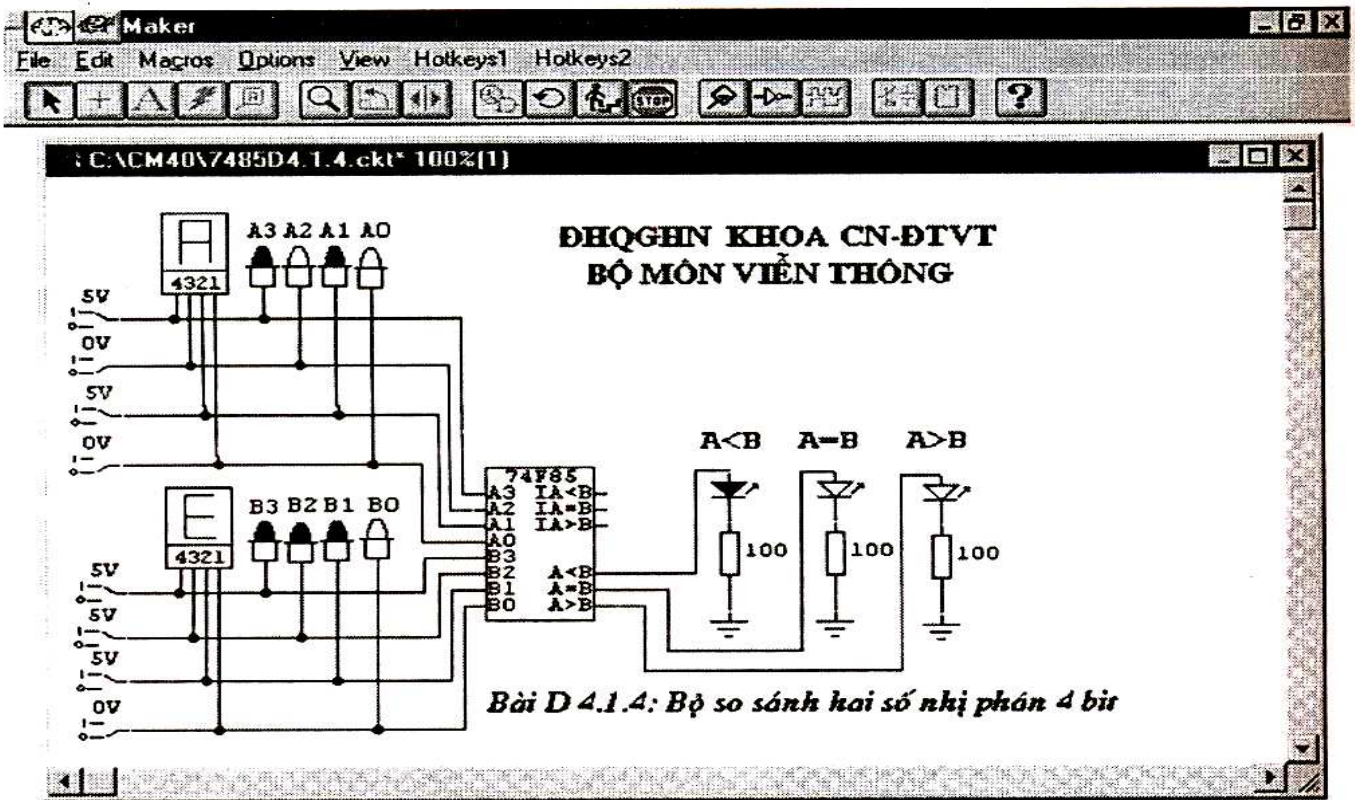
Hình 10.60: Thiết kế mô phỏng bộ tổng toàn phần dùng XOR, AND, OR



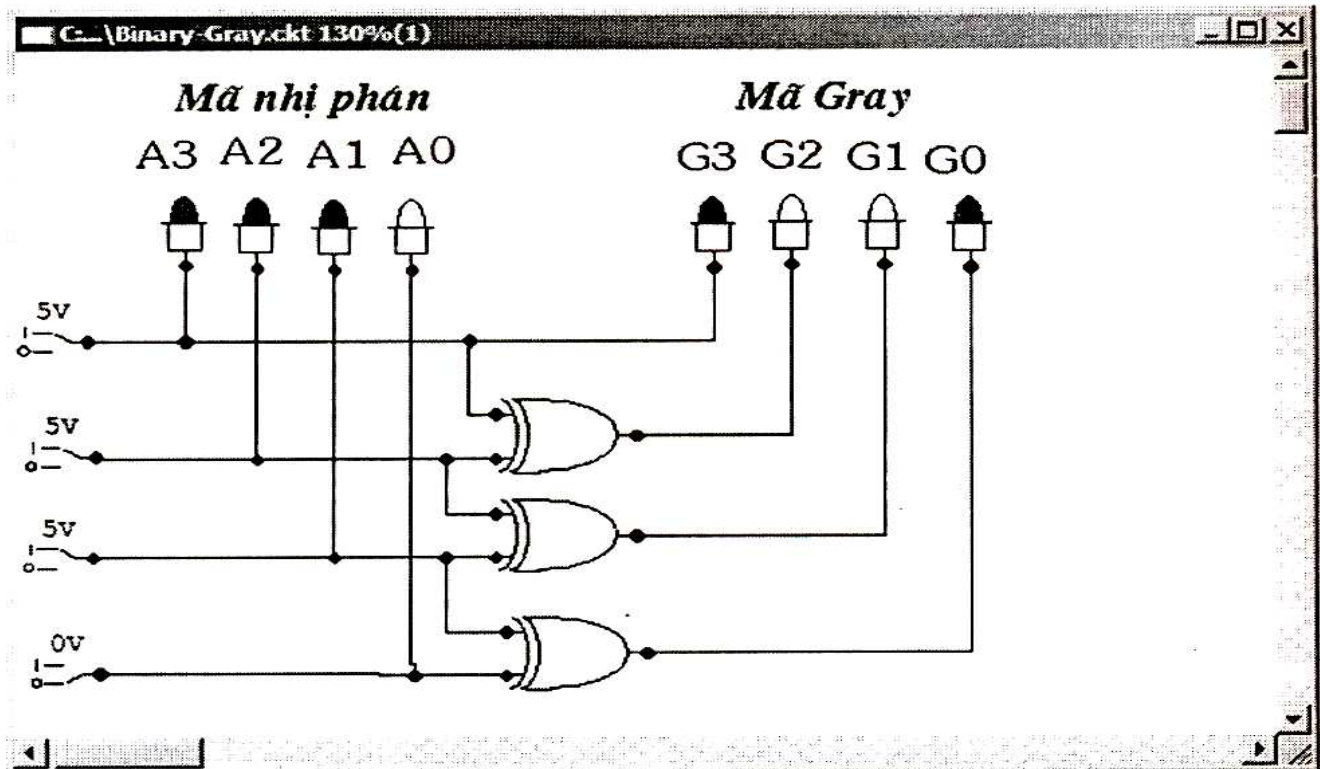
Hình 10.61: Thiết kế mô phỏng bộ tổng toàn phần dùng NAND, NOR



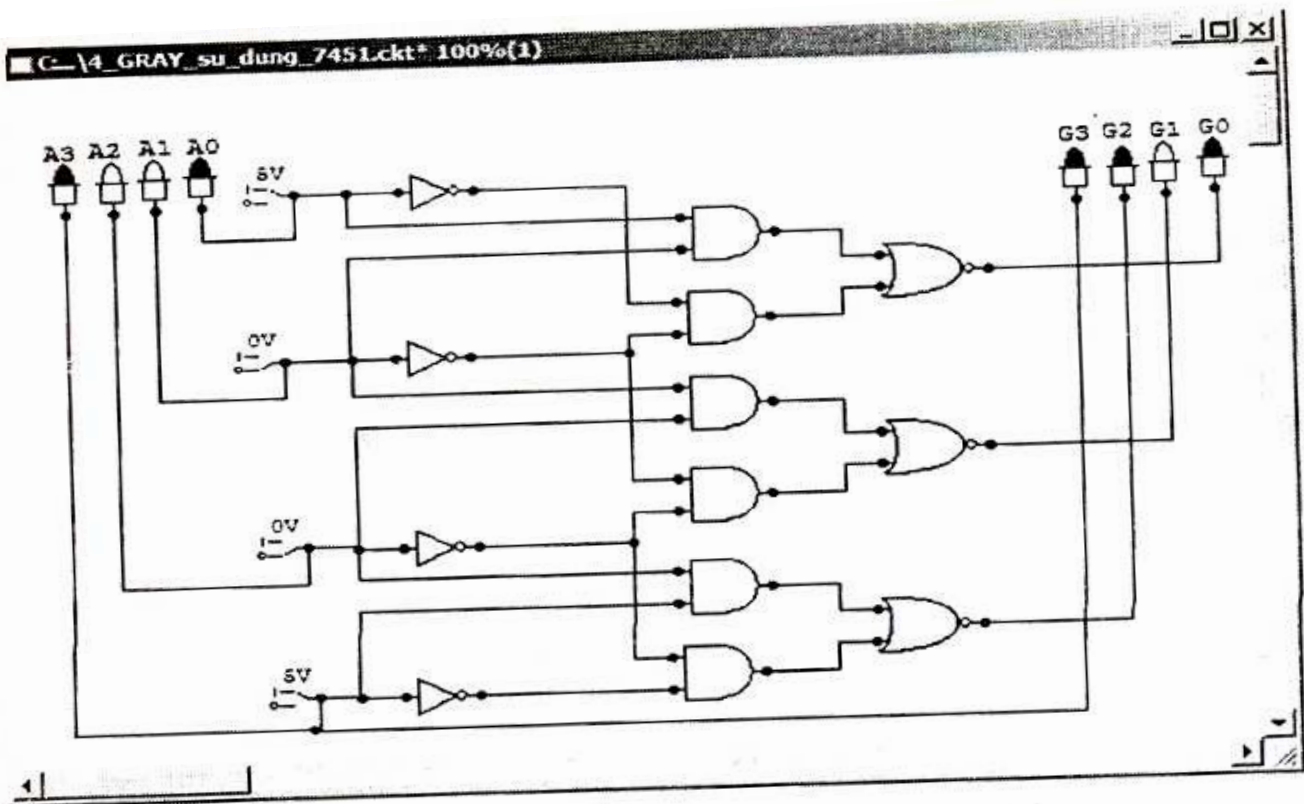
Hình 10.62: Thiết kế thí nghiệm mô phỏng bộ cộng nhị phân 4 bit



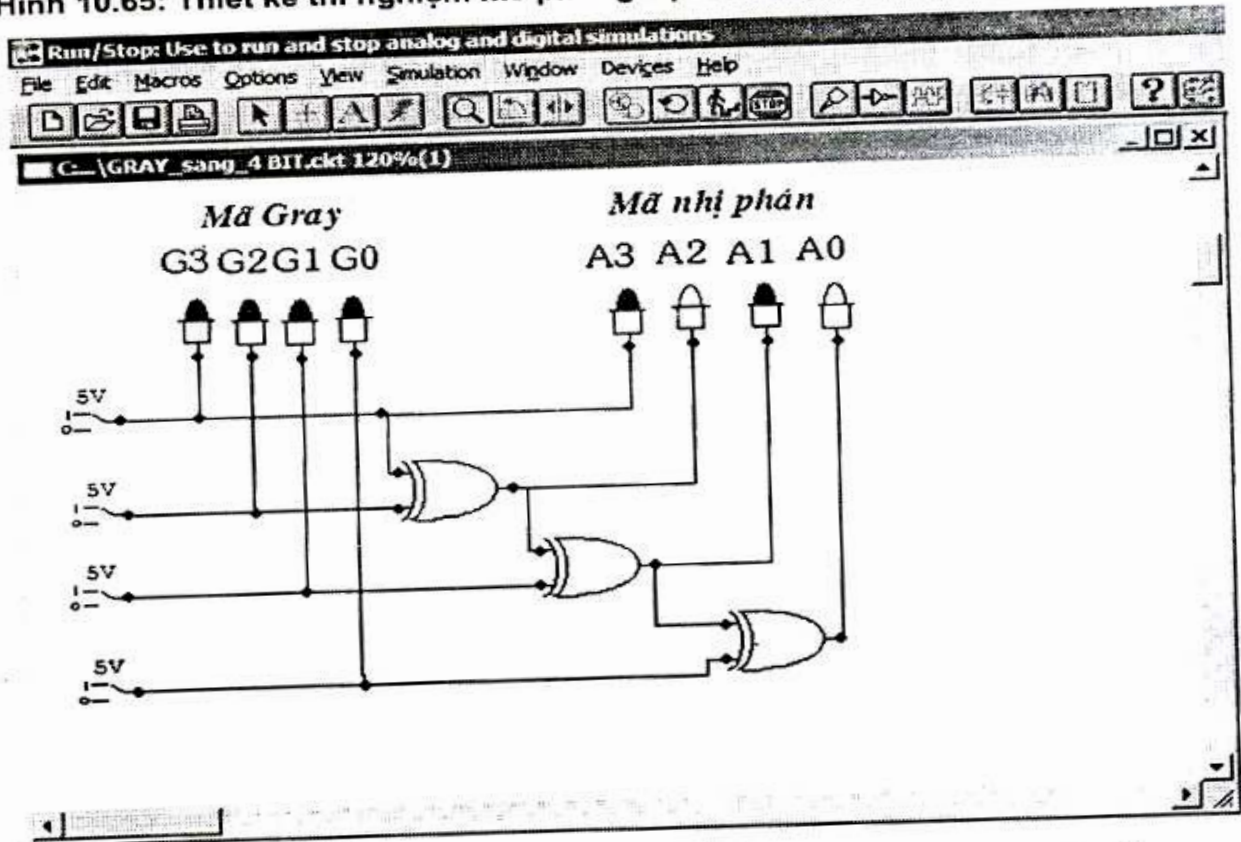
Hình 10.63: Thiết kế thí nghiệm mô phỏng bộ so sánh hai số nhị phân 4 bit



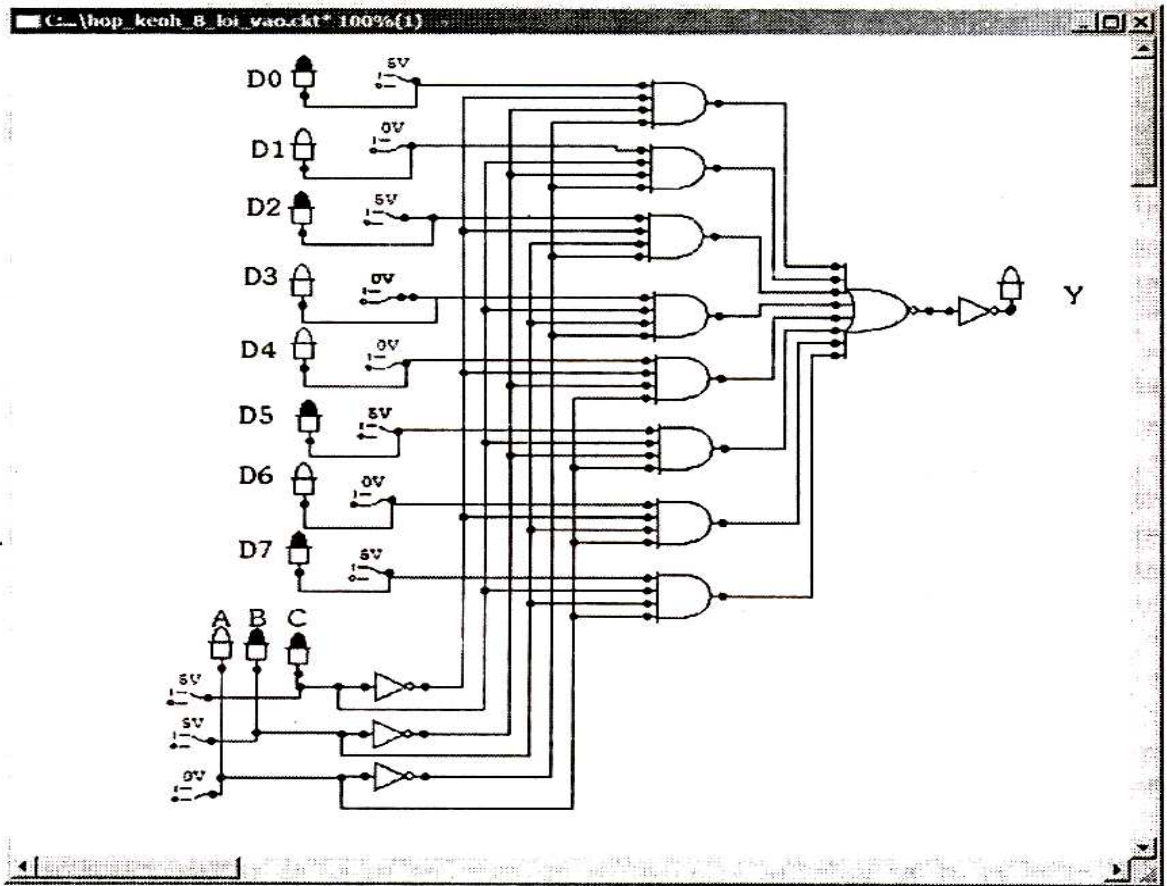
Hình 10.64: Thiết kế thí nghiệm mô phỏng bộ biến đổi mã nhị phân sang mã Gray



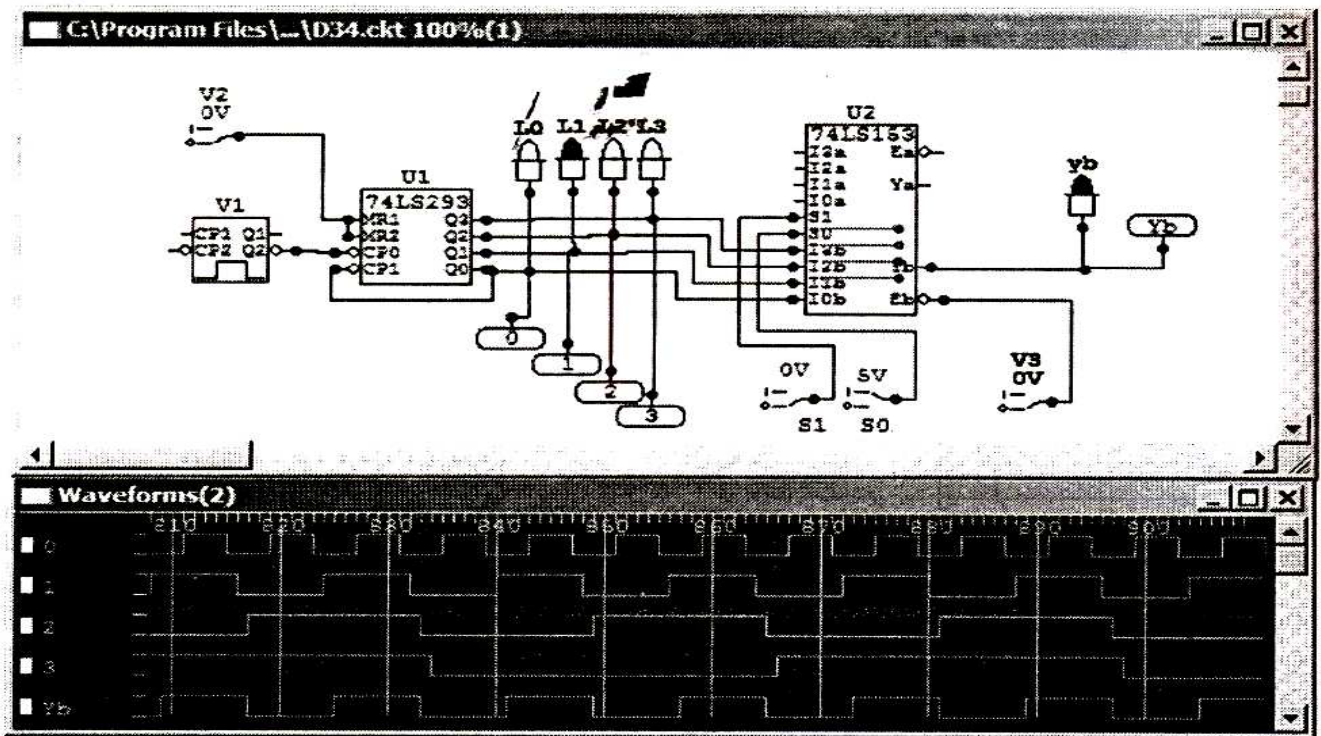
Hình 10.65: Thiết kế thí nghiệm mô phỏng bộ biến đổi mã nhị phân sang mã Gray



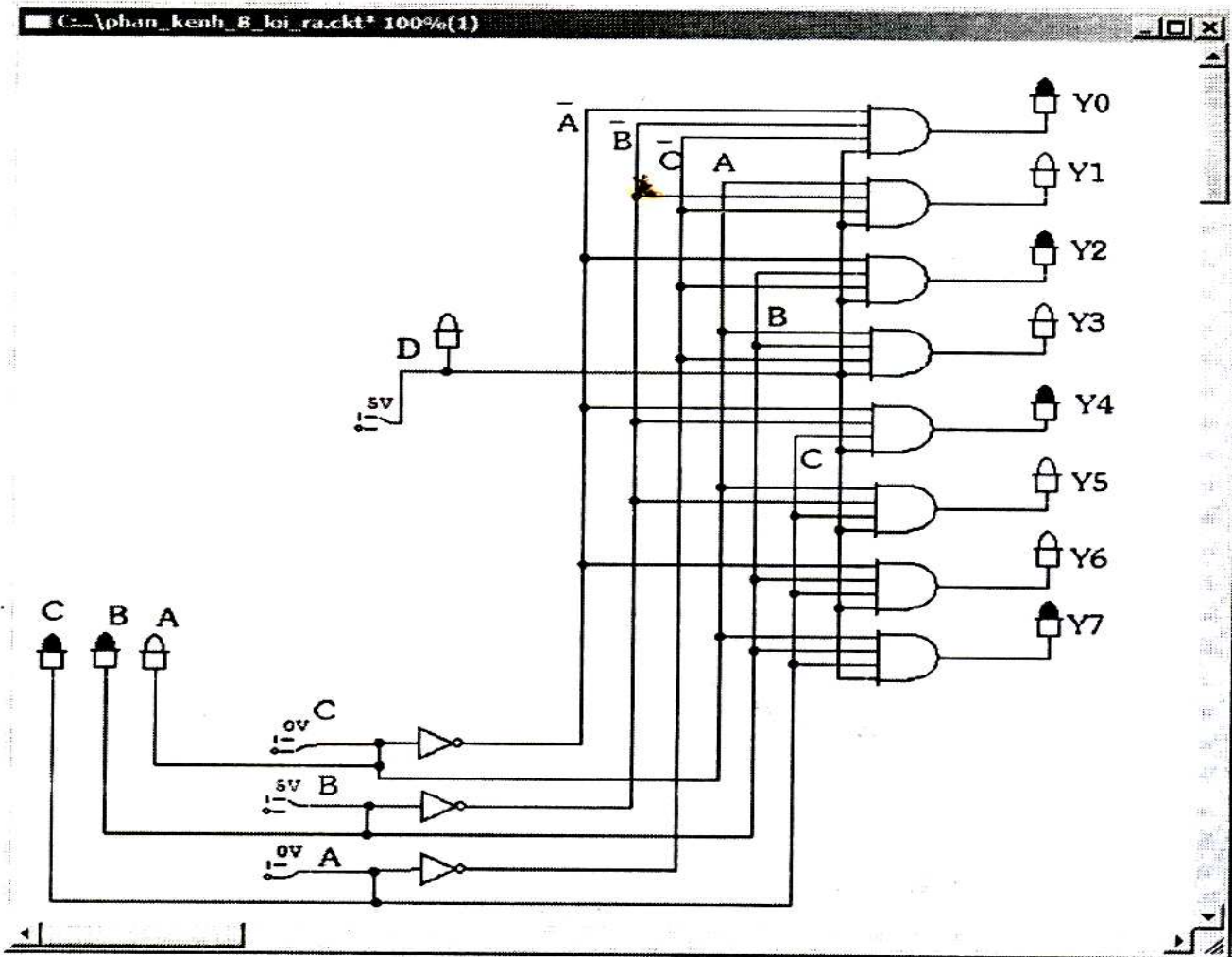
Hình 10.66: Thiết kế mô phỏng bộ biến đổi mã Gray sang mã nhị phân



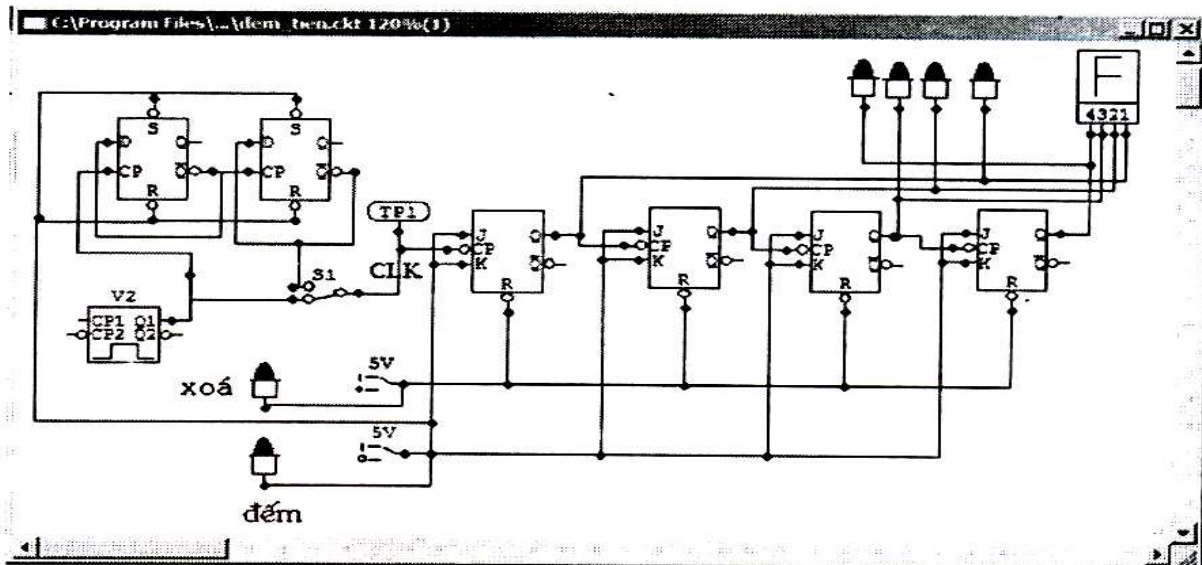
Hình 10.67: Thiết kế mô phỏng hoạt động của bộ hợp kênh 8 đường



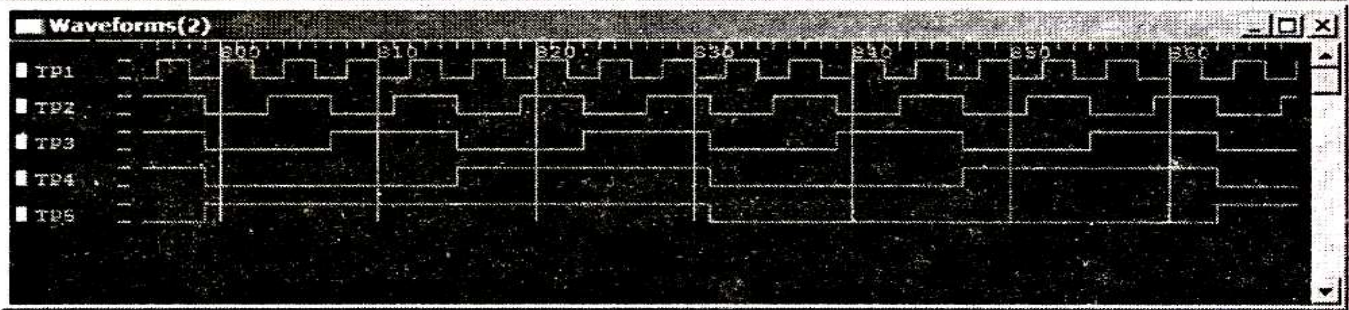
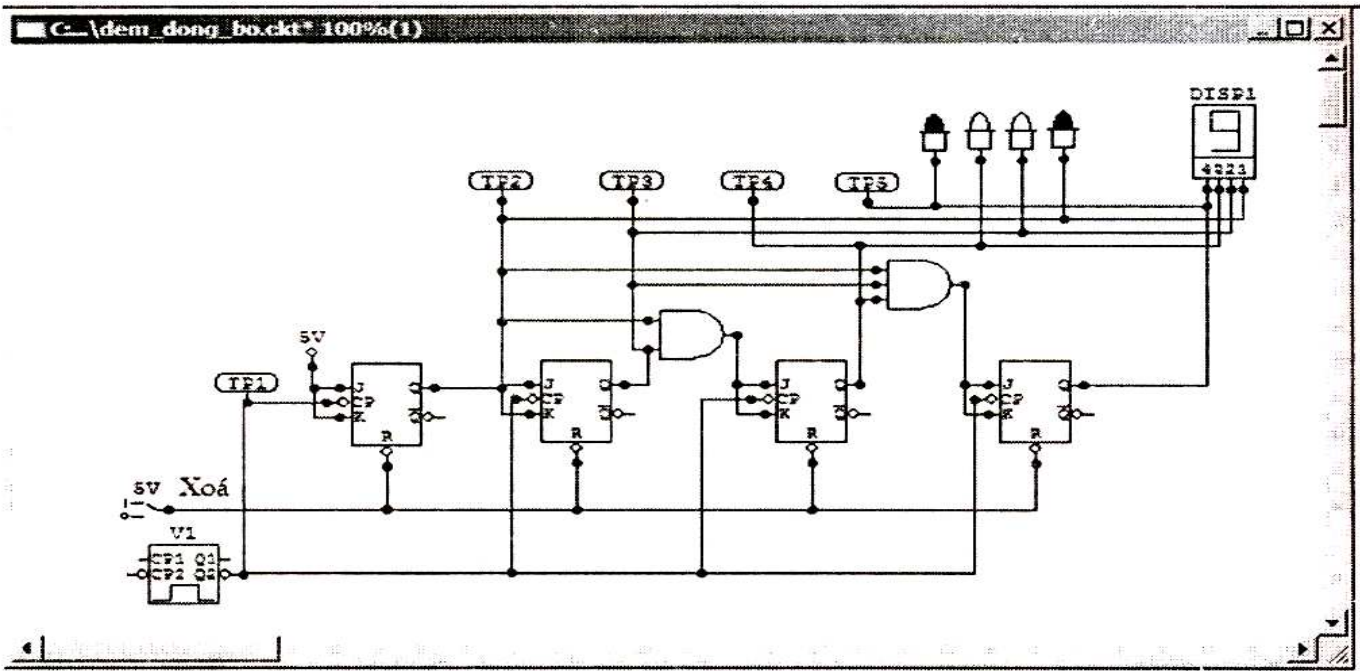
Hình 10.68: Thiết kế mô phỏng hoạt động của bộ hợp kênh 4 đường



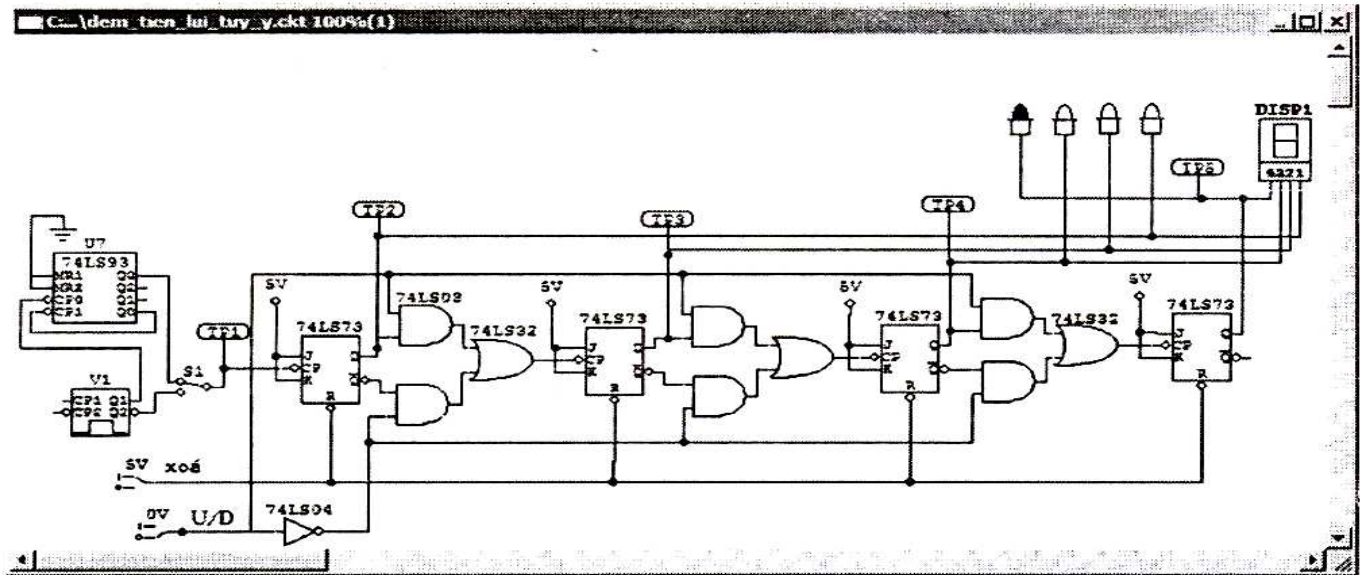
Hình 10.69: Thiết kế mô phỏng hoạt động của bộ phân kênh từ 1 vào ra 8 đường



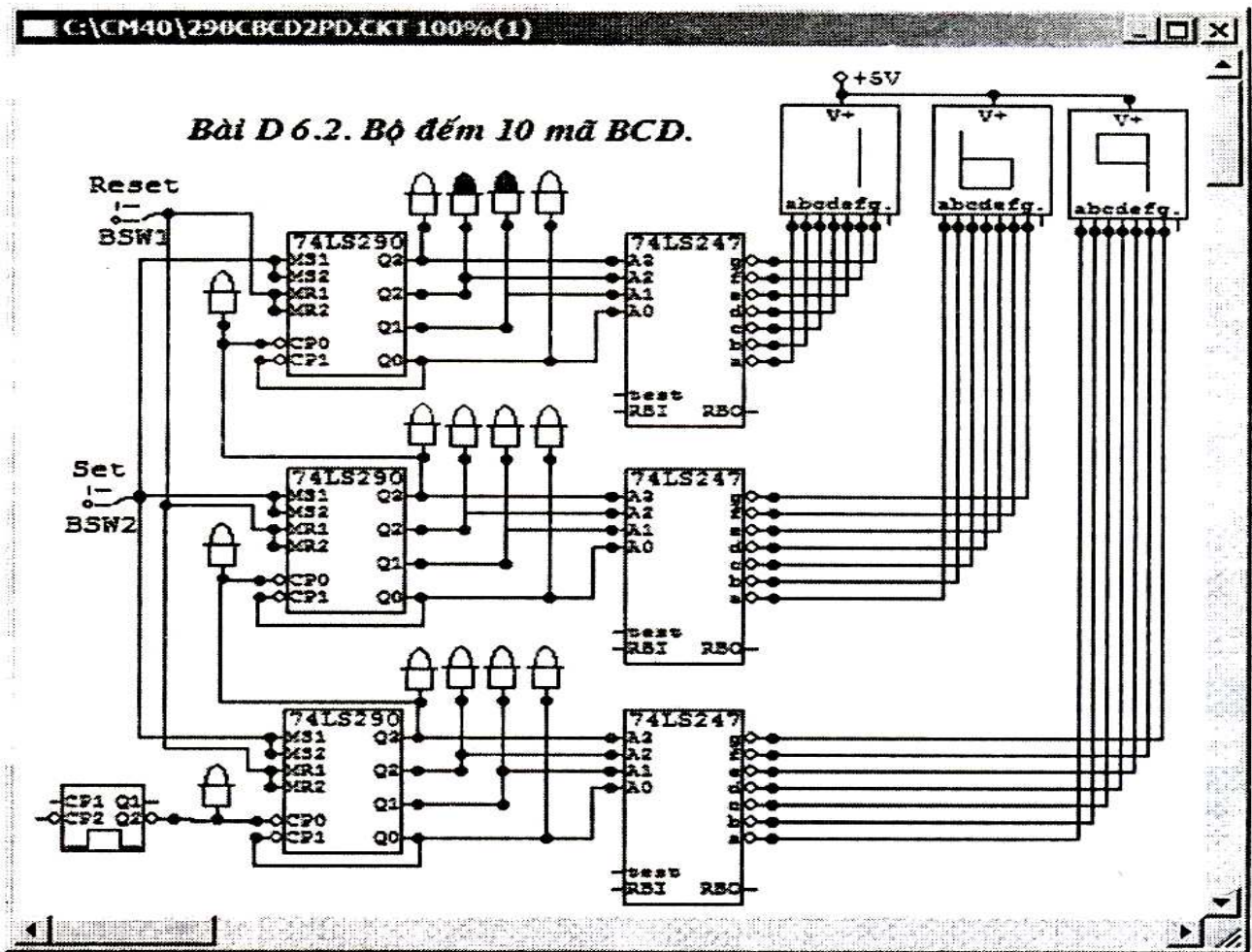
Hình 10.70: Thiết kế mô phỏng hoạt động của bộ đếm nhị phân không đồng bộ



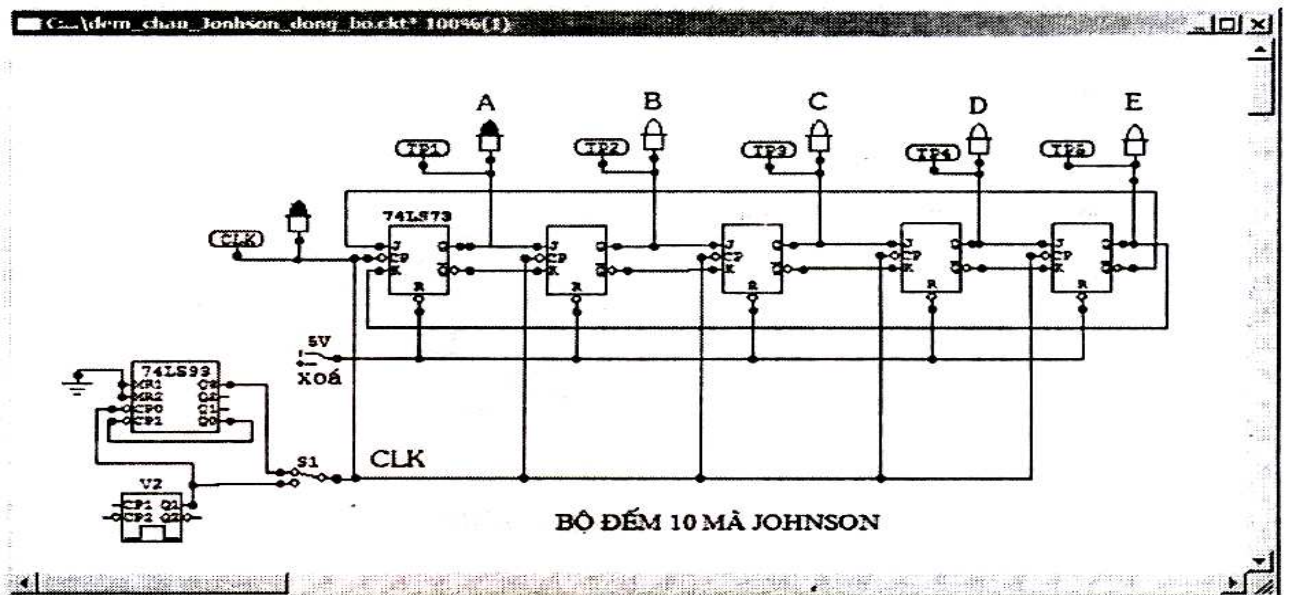
Hình 10.71: Thiết kế mô phỏng hoạt động của bộ đếm nhị phân đồng bộ



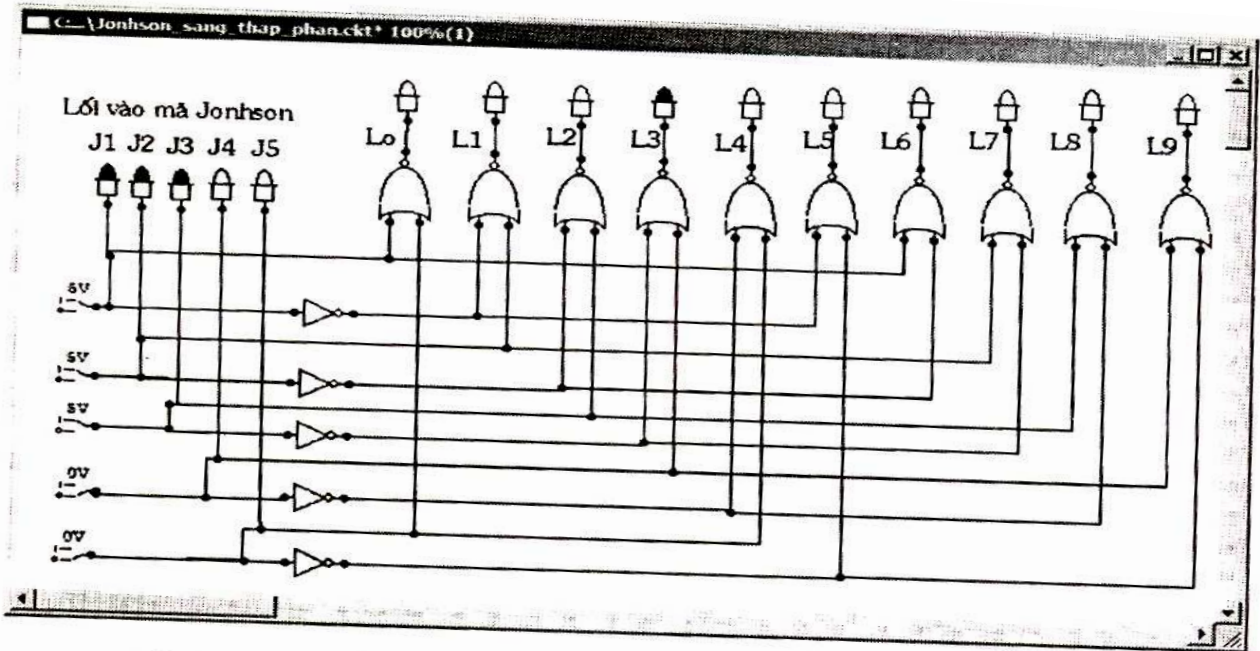
Hình 10.72: Thiết kế mô phỏng hoạt động của bộ đếm nhị phân tiến lùi



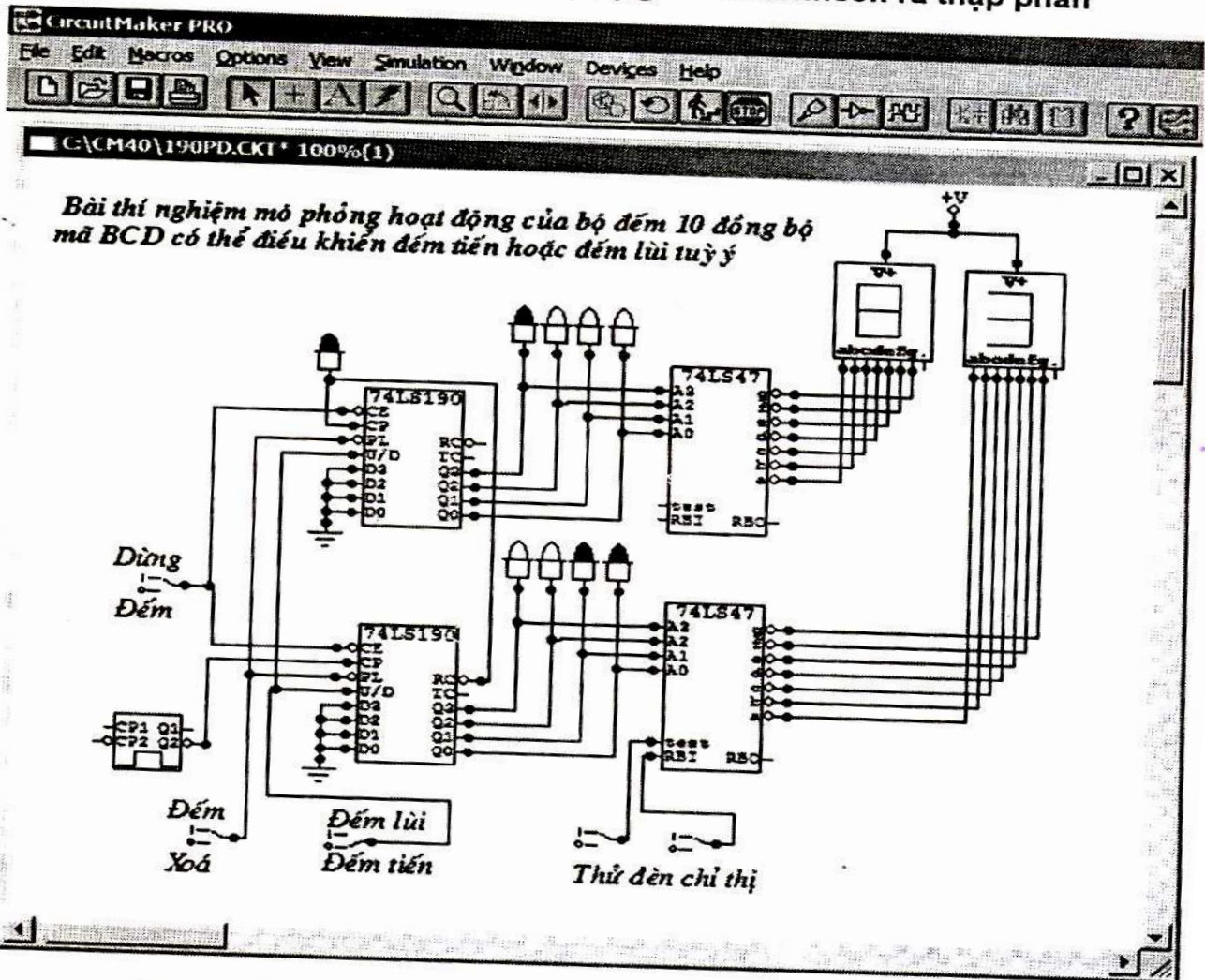
Hình 10.73: Thiết kế mô phỏng hoạt động của bộ đếm 10 mã BCD



Hình 10.74: Thiết kế mô phỏng bộ đếm 10 mã Johnson



Hình 10.75: Thiết kế mô phỏng bộ giải mã Johnson ra thập phân



Hình 10.76: Thiết kế mô phỏng hoạt động của bộ đếm 10 đồng bộ mã BCD

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Kim Giao. Lê Xuân Thê. *Kỹ thuật điện tử I*. NXB Giáo dục, 2000.
- [2]. Bộ môn Điện tử Đại học Thanh hoa Bắc Kinh. *Cơ sở kỹ thuật điện tử số*. Vũ Đức Thọ dịch. NXB Giáo dục, 1996.
- [3]. Huỳnh Đắc Thắng. *Kỹ thuật số thực hành*. NXB Khoa học Kỹ thuật, 1997.
- [4]. Richard J. Higgins. *Electronics with Digital and Analog Integrated Circuit*. Prentice-Hall, INC., Englewood Cliffs N.J.07632, 1983.
- [5]. Thomas C. Hayes Paul Horowitz. *The ART of electronics*, Havard University Cambridge University Press 1989.
- [6]. *RCA Solid State 74 Databook, COS/MOS Digital Integrated Circuits*.
- [7]. *Memory Databook*, National Semiconductor, 1977.
- [8]. *MECL integrated Circuit Databook*. Motorola Semiconductor Products Inc, 1972.
- [9]. Bar Giora Goldberd. *Digital techniques in frequency synthesis*, Mc Grow Hill, New York, 1996.
- [10]. Edward C.Weber, J. Neal Ford and Christopher R. Weber. *Developing with Delphi object- oriented techniques*. Prentice Hall 1996.
- [11]. Nguyễn Kim Giao, Phạm Phi Hùng. *Máy phát điều khiển số và lập trình mô phỏng*. Tạp chí Khoa học và Công nghệ, 18-24, XXXIX, số 5, 2001.

MỤC LỤC

<i>Lời nói đầu</i>	3
Chương 1. Các hệ thống đếm và mã số	5
1.1. Các hệ thống đếm	5
1.2. Chuyển đổi một số từ hệ đếm này sang hệ đếm khác.....	8
1.3. Các phép tính số học trong hệ đếm nhị phân.....	11
1.4. Mã hoá số của hệ thập phân	14
1.5. Mã hoá chữ cái, chữ số, và các kí tự khác	16
1.6. Khái niệm về số bù.....	17
1.7. Biểu diễn số âm trong hệ đếm nhị phân	18
Chương 2. Đại số lôgic	22
2.1. Ba phép tính cơ bản của đại số lôgic	22
2.2. Các định luật cơ bản của đại số Boole	22
2.3. Phương pháp biểu diễn hàm lôgic.....	24
2.4. Các hàm lôgic cơ bản	32
Chương 3. Các họ vi mạch lôgic cơ bản	39
3.1. Đặc điểm chung của các vi mạch lôgic	39
3.2. Họ lôgic RTL.....	40
3.3. Họ lôgic DTL	43
3.4. Họ lôgic TTL	44
3.5. Các vi mạch lôgic dùng transistor trường	52
3.6. Họ lôgic CMOS.....	64
3.7. Họ lôgic ECL.....	68
Chương 4. Mạch lôgic tổ hợp	72
4.1. Phương pháp thiết kế các mạch lôgic tổ hợp	72

4.2. Mạch tính toán số học:	75
4.3. Bộ hợp kênh	87
4.4. Bộ phân kênh	92
4.5. Các mạch mã hoá và giải mã	102
Chương 5. Trigơ	126
5.1. Trigơ RS	126
5.2. Trigơ đồng bộ RST	129
5.3. Trigơ JK	131
5.4. Trigơ D	135
Chương 6. Các bộ đếm	137
6.1. Các sơ đồ đếm nhị phân	137
6.2. Đếm 10 mã BCD	144
6.3. Các sơ đồ đếm vòng theo kiểu mã Jonhson	154
6.4. Các bộ ghi dịch	161
6.5 Các bộ chia tần	167
Chương 7. Bộ nhớ bán dẫn	171
7.1. Khái niệm cơ bản	171
7.2. Bộ nhớ chỉ đọc ROM	172
7.3. Bộ nhớ truy cập ngẫu nhiên RAM	173
Chương 8. Bộ Chuyển đổi số - tương tự và tương tự - số	180
8.1. Khái quát về DAC	180
8.2. Các mã số dùng cho DAC	182
8.3. Các loại DAC	182
8.4. Đặc tính của DAC	186
8.5. Điện áp chuẩn của DAC	188
8.6. Bộ biến đổi tương tự số ADC	189
8.7. Xây dựng hệ ghép nối ADC, DAC với máy tính	196

Chương 9. Vòng băm pha	206
9.1. Khái quát về vòng băm pha	206
9.2. Cấu tạo của vòng băm pha CMOS PLL CD4046.....	210
9.3. Những ứng dụng của vòng băm pha CMOS PLL	219
9.4. Kỹ thuật tổ hợp tần số dùng vòng băm pha ghép nối máy vi tính	221
9.5. Máy phát điều khiển số ghép nối với máy vi tính	224
9.6. Máy phát điều khiển số và lập trình mô phỏng	229
Chương 10. Thiết kế mô phỏng các mạch điện tử số	239
10.1. Giới thiệu sơ lược về chương trình CircuitMaker	240
10.2. Hướng dẫn sử dụng CircuitMaker	260
10.3. Vẽ và soạn thảo các sơ đồ mạch điện tử.....	277
10.4. Mô phỏng mạch điện tử.....	289
10.5. Thiết kế mô phỏng các vi mạch logic, các mạch điện tử số	310
<i>Tài liệu tham khảo</i>	323

NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

16 Hàng Chuối - Hai Bà Trưng - Hà Nội

Điện thoại: (04) 9718312; (04) 9724770. Fax: (04) 9714899

E-mail: nxb@vnu.edu.vn

Chịu trách nhiệm xuất bản:

Giám đốc: PHÙNG QUỐC BẢO

Tổng biên tập: NGUYỄN BÁ THÀNH

Chịu trách nhiệm nội dung:

Hội đồng nghiệm thu giáo trình
Trường Đại học Công nghệ - ĐHQGHN

TS NGUYỄN VĂN THÁI

TS TRẦN QUANG VINH

Biên tập: KIM HÀ - QUỐC THẮNG

Chế bản: THU HƯƠNG

Trình bày bìa: NGỌC ANH

KỸ THUẬT ĐIỆN TỬ SỐ

Mã số: 1K-44 ĐH2006

In 1000 cuốn, khổ 16 x 24 cm tại Nhà in Khoa học và Công nghệ

Số xuất bản: 406 - 2006/CXB/11 - 64/ĐHQGHN, ngày 22/05/2006

Quyết định xuất bản số: 363 KH/XB

In xong và nộp lưu chiểu quý IV năm 2006.