

TRƯỜNG CAO ĐẲNG NGHỀ ĐỒNG THÁP
KHOA ĐIỆN – ĐIỆN TỬ



GIÁO TRÌNH

MÔN HỌC: THỰC HÀNH KỸ THUẬT SỐ
NGÀNH, NGHỀ: CNKT ĐIỆN, ĐIỆN TỬ
TRÌNH ĐỘ: CAO ĐẲNG/TRUNG CẤP

*(Ban hành kèm theo Quyết định Số: /QĐ-CĐNĐT ngày... tháng...năm 2018
của Hiệu trưởng Trường Cao đẳng nghề Đồng Tháp)*

Đồng Tháp, năm 2018

TUYÊN BỐ BẢN QUYỀN

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

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

TaiLieu.vn

LỜI GIỚI THIỆU

Kỹ thuật số là một trong những mô đun cơ sở của nghề Điện tử công nghiệp được biên soạn dựa theo chương trình khung đã xây dựng và ban hành năm 2017 của trường Cao đẳng nghề Đồng Tháp trước đây dành cho nghề Điện tử công nghiệp hệ Cao đẳng và Trung cấp.

Giáo trình được biên soạn làm tài liệu học tập, giảng dạy nên giáo trình đã được xây dựng ở mức độ đơn giản và dễ hiểu, trong mỗi bài học đều có thí dụ và bài tập tương ứng để áp dụng và làm sáng tỏ phần lý thuyết.

Khi biên soạn, nhóm biên soạn đã dựa trên kinh nghiệm thực tế giảng dạy, tham khảo đồng nghiệp, tham khảo các giáo trình hiện có và cập nhật những kiến thức mới có liên quan để phù hợp với nội dung chương trình đào tạo và phù hợp với mục tiêu đào tạo, nội dung được biên soạn gắn với nhu cầu thực tế.

Nội dung giáo trình được biên soạn với dung lượng thời gian đào tạo 90 giờ gồm có:

Bài 1 MĐ14-01: Đại cương về kỹ thuật số.

Bài 2 MĐ14-02: FLIP – FLOP.

Bài 3 MĐ14-03: Mạch đếm và thanh ghi.

Bài 4 MĐ14-04: Mạch logic MSI.

Bài 5 MĐ14-05: Họ vi mạch TTL – CMOS.

Bài 6 MĐ14-06: Bộ nhớ.

Bài 7 MĐ14-07: Kỹ thuật ADC – DAC.

Giáo trình cũng là tài liệu giảng dạy và tham khảo tốt cho các nghề Công nghệ kỹ thuật Điện – Điện tử, Điện tử dân dụng, Điện công nghiệp.

Mặc dù đã cố gắng tổ chức biên soạn để đáp ứng được mục tiêu đào tạo nhưng không tránh được những thiếu sót. Rất mong nhận được sự đóng góp ý kiến của các thầy, cô, bạn đọc để nhóm biên soạn sẽ điều chỉnh hoàn thiện hơn.

Đồng Tháp, ngày tháng năm 2018

Tham gia biên soạn

1. Chủ biên: Lâm Văn Vũ

2. Trần Thế Thông

MỤC LỤC

	Trang
TUYÊN BỐ BẢN QUYỀN	1
LỜI GIỚI THIỆU	3
MỤC LỤC	4
BÀI 01: ĐẠI CƯƠNG VỀ KỸ THUẬT SỐ	6
1. Tổng quan về mạch tương tự và mạch số	6
2. Hệ thống số và mã số	7
3. Các cổng Logic cơ bản	12
4. Biểu thức Logic và mạch điện	17
5. Đại số Boole và định lý Demorgan	22
6. Đơn giản biểu thức logic	22
7. Giới thiệu một số IC số cơ bản:	27
BÀI 02: FLIP – FLOP	32
1. Flip - Flop R-S:	32
2. FF R-S tác động theo xung lệnh	33
3. Flip - Flop J-K	35
4. Flip - Flop T	35
5. Flip - Flop D	36
6. Flip - Flop M-S (Master – Slaver):	36
7. Flip - Flop với ngõ vào Preset và Clear	37
8. Tính toán, lắp ráp một số mạch ứng dụng cơ bản	38
BÀI 03: MẠCH ĐẾM VÀ THANH GHI	41
1. Mạch đếm	41
2. Thanh ghi	49
3. Giới thiệu một số IC đếm và thanh ghi thông dụng	50
4. Tính toán, lắp ráp một số mạch ứng dụng cơ bản	52
BÀI 04: MẠCH LOGIC MSI	55
1. Mạch mã hóa (Encoder)	55
2. Mạch giải mã (Decoder)	59
3. Mạch ghép kênh	65
4. Mạch tách kênh	67
5. Giới thiệu một số IC mã hóa và giải mã thông dụng.	69
6. Tính toán, lắp ráp một số mạch ứng dụng cơ bản	72
BÀI 05: HỘ VI MẠCH TTL – CMOS	75
1. Cấu trúc và thông số cơ bản của TTL	75
2. Cấu trúc và thông số cơ bản của CMOS	84
3. Giao tiếp TTL và CMOS	92
4. Giao tiếp giữa mạch logic và tải công suất	94
5. Tính toán, lắp ráp một số mạch ứng dụng cơ bản	96
BÀI 06: BỘ NHỚ	98
1. ROM	99
2. RAM	106
3. Mở rộng dung lượng bộ nhớ	108
4. Giới thiệu IC	110

BÀI 07: KỸ THUẬT ADC – DAC	112
1. Mạch chuyển đổi số sang tương tự (DAC)	112
2. Mạch chuyển đổi tương tự sang số (ADC)	118
3. Giới thiệu IC	126
TÀI LIỆU THAM KHẢO.....	133

TaiLieu.vn

BÀI 01: ĐẠI CƯƠNG VỀ KỸ THUẬT SỐ

Giới thiệu:

Trong khoa học, công nghệ hay cuộc sống đời thường, ta thường xuyên phải tiếp xúc với số lượng. Số lượng có thể đo, quản lý, ghi chép, tính toán nhằm giúp cho các xử lý, ước đoán trở nên ít phức tạp hơn.

Mục tiêu:

- Trình bày được các khái niệm cơ bản về mạch tương tự và mạch số.
- Trình bày được cấu trúc của hệ thống số và mã số.
- Trình bày được cấu tạo, nguyên lý hoạt động của các cổng logic cơ bản
- Trình bày được các định luật cơ bản về kỹ thuật số, các biểu thức toán học của số

- Chủ động, sáng tạo và đảm bảo trong quá trình học tập

Nội dung

1. Tổng quan về mạch tương tự và mạch số

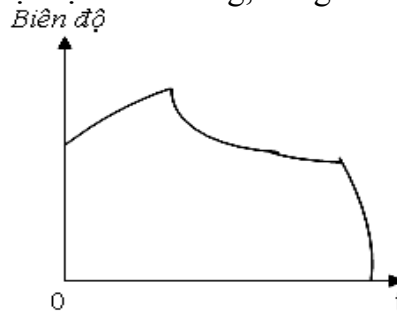
1.1 Định nghĩa

- Tín hiệu

Tín hiệu là biến thiên của biên độ, thường là điện áp hay dòng điện theo thời gian. Đường biểu diễn của tín hiệu là dạng sóng

- Tín hiệu tương tự (hình 4.1)

Tín hiệu tương tự là tín hiệu có biên độ liên tục theo thời gian. Trong thực tế các đại lượng vật lý như vận tốc, nhiệt độ môi trường, tiếng nói...đều là tín hiệu tương tự.



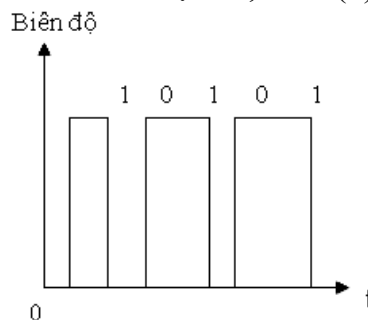
a- tín hiệu tương tự

Hình 4.1

Trong kỹ thuật điện tử mạch tương tự là mạch xử lý các tín hiệu tương tự có dạng như hình vẽ có nghĩa là trong cùng một khoảng thời gian xác định mạch phải xử lý n mức tín hiệu khác nhau

- Tín hiệu số (hình 4.2)

Tín hiệu số là tín hiệu có biên độ gián đoạn theo thời gian. Biên độ chỉ có hai mức như hình vẽ, mức (1) đại diện cho biên độ cao, mức (0) đại diện cho biên độ thấp.



b- tín hiệu số

Hình 4.2

Mạch số chỉ xử lý hai mức tín hiệu 0 hoặc 1 trong cùng một khoảng thời gian mà thôi.

1.2 Ưu và nhược điểm của kỹ thuật số so với kỹ thuật tương tự

Kỹ thuật số có nhiều ưu điểm so với kỹ thuật mạch tương tự khiến cho kỹ thuật số ngày càng phổ biến ở gần như hầu hết các lĩnh vực như: đo lường, điều khiển tính toán, thông tin... Tuy nhiên kỹ thuật mạch tương tự cũng có những đặc tính riêng mà mạch số không thể thay thế.

❖ Ưu điểm:

Nhìn chung thiết bị số dễ thiết kế hơn: Đó là do mạch được sử dụng các vi mạch chuyên dùng đã được thiết kế với chức năng định trước. Khả năng chống nhiễu và sự méo dạng cao: Do đặc thù của hệ thống là chỉ xử lý hai mức tín hiệu 1 và 0 và thời gian chuyển tiếp giữa chúng là rất nhanh nên khả năng chống nhiễu rất cao, hơn nữa biên độ của tín hiệu nhiễu không đủ khả năng làm thay đổi giữa hai mức tín hiệu từ 0 sang 1 và ngược lại từ 1 sang 0. Trong khi đó ở thiết bị tương tự độ chính xác bị giới hạn vì mạch phải xử lý các tín hiệu liên tục theo thời gian, hơn nữa các linh kiện sử dụng không thực sự tuyến tính.

Do đó biên độ của tín hiệu nhiễu dễ dàng xâm nhập vào hệ thống và làm mất tính ổn định của hệ thống.

Lưu trữ và truy cập dễ dàng, nhanh chóng: Do tín hiệu số chỉ có hai mức nên việc lưu trữ ở những môi trường khác nhau (bộ nhớ bán dẫn, băng từ...) và truy cập rất thuận tiện.

Độ chính xác và độ phân giải cao: Trong việc đo đạc thời gian, tần số, điện thế v.v... kỹ thuật số cho độ chính xác và độ phân giải cao hơn kỹ thuật tương tự.

Có thể lập trình hoạt động của hệ thống kỹ thuật số: Hoạt động của hệ thống kỹ thuật có thể điều khiển theo một qui luật định trước bằng một tập lệnh gọi là chương trình. Cùng với việc ra đời của các vi xử lý và vi điều khiển làm cho việc tự động điều khiển hệ thống trở nên dễ dàng hơn.

❖ Nhược điểm

Hầu hết các đại lượng vật lý đều mang bản chất của tín hiệu tương tự. Chính những đại lượng này thường là đầu vào và đầu ra của các hệ thống điều khiển. Ví dụ như các đại lượng nhiệt độ, áp suất, lưu lượng, vận tốc... Phân tích các đại lượng này theo thời gian đó chính là các đại lượng tương tự.

Trong kỹ thuật người ta thường phải thực hiện biến đổi từ tín hiệu tương tự sang tín hiệu số và ngược lại. Điều này làm cho thiết bị thêm phức tạp và giá thành cao hơn. Tuy nhiên những bất lợi này bị lấn lướt bởi ưu điểm của kỹ thuật số nên việc chuyển đổi qua lại giữa kỹ thuật số và kỹ thuật tương tự là việc cần thiết và trở nên phổ biến trong công nghệ ngày nay.

Để tận dụng được những ưu điểm của kỹ thuật số và kỹ thuật tương tự người ta sử dụng cả hai loại vào trong một hệ thống. Ở những hệ thống này khâu thiết kế cần quyết định khâu nào dùng kỹ thuật tương tự và khâu nào dùng kỹ thuật số.

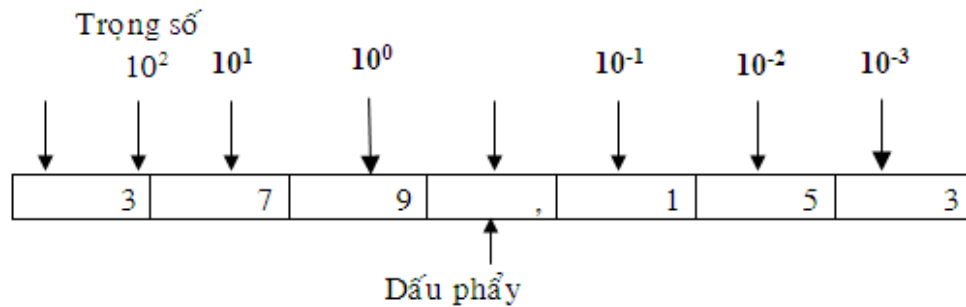
2. Hệ thống số và mã số

2.1 Hệ thống số thập phân (Decimal system)

Trong hệ thập phân người ta sử dụng 10 ký tự từ 0 đến 9 kết hợp với các dấu chấm, dấu phẩy để chỉ về lượng:

Trong dãy số thập phân: $d_{n-1}...d_2d_1d_0$ theo qui ước từ phải qua trái vị trí của chúng thể hiện hàng đơn vị, hàng chục, hàng trăm, hàng nghìn... với phần nguyên và ngược lại từ trái qua phải là phần chục, phần trăm, phần nghìn... đối với phần lẻ sau dấu phẩy.

Nói tóm lại bất kì số nào cũng là tổng các tích giữa giá trị của mỗi chữ số với giá trị (gọi là trọng số) của nó.



Hình 4.3

Đối với một dãy số thập phân có n số hạng thì có 10^n giá trị và giữa hai giá trị liền kề nhau chênh lệch nhau 10 lần

2.2 Hệ thống số nhị phân (Binary system)

- Ký tự số : 0,1
- Cơ số: 2

Để biểu diễn số nhị phân người ta dùng hai kí số (digit) 0 và 1 để diễn tả về lượng của một đại lượng nào đó. Một dãy số nhị phân chỉ tính phần nguyên được biểu diễn như sau:

$$b_{n-1}b_{n-2} \dots b_2b_1b_0$$

Qui ước mỗi số hạng là một bit. Bit tận cùng bên trái gọi là MSB (tức là bit có giá trị lớn nhất) và bit ở tận cùng bên phải gọi là LSB (tức là bit có giá trị nhỏ nhất).

Như vậy số nhị phân có n bit thì sẽ có 2^n giá trị khác nhau. Giá trị nhỏ nhất là 0. . .000 và giá trị lớn nhất là 1. . .111. Trọng số các bit từ thấp đến cao lần lượt là 1, 2, 4, 8 và giữa hai bit kề nhau chênh lệch nhau 2 lần.

❖ Chuyển đổi từ số nhị phân sang thập phân.

Quy tắc chuyển như sau:

$$b_{n-1}b_{n-2} \dots b_2b_1b_0 = b_{n-1} \cdot 2^{n-1} + b_{n-2} \cdot 2^{n-2} \dots b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

❖ Chuyển đổi từ số thập phân sang nhị phân.

Quy tắc chuyển như sau:

Sử dụng qui tắc chia 2 liên tiếp số A_{10} và lấy phần dư

- Phần dư đầu tiên của phép chia là bit LSB
- Phần dư cuối cùng của phép chia là bit MSB

2.3 Hệ thống số bát phân (Octal system)

- Ký tự số : 0,1,2,3,4,5,6,7
- cơ số: 8

Trong hệ thống số bát phân người ta dùng các số từ 0 đến 7 để mô tả về lượng của một đại lượng và cũng theo luật vị trí trọng số của 8^m ($m = \dots -2, -1, 0, 1, 2, \dots$). Một dãy số octal được biểu diễn như sau: $0_{n-1}0_{n-2} \dots 0_20_10_0$

Trong đó một dãy số bát phân có n số hạng thì sẽ có 8^n giá trị khác nhau, giá trị thấp nhất là 0. . .000 và giá trị lớn nhất là 7. . .777. Trọng số các bit từ thấp đến cao lần lượt là 1, 8, 64. . . và giữa hai số liền kề nhau chênh lệch nhau 8 lần

❖ Chuyển đổi từ bát phân sang thập phân

Quy tắc chuyển như sau:

$$0_{n-1}0_{n-2} \dots 0_20_10_0$$

❖ Chuyển đổi số thập phân sang biểu diễn số bát phân

Quy tắc chuyển như sau:

Để thực hiện chuyển từ A_{10} sang A_8 ta thực hiện phép chia của A_{10} cho A_8 rồi lấy phần dư

❖ **Chuyển đổi một số bát phân sang số nhị phân**

Để thực hiện chuyển đổi ta thay thế một ký tự số bằng một số nhị phân 3 bit tương ứng theo bảng sau:

Bảng hình 4.1

ký tự số bát phân	0	1	2	3	4	5	6	7
Nhóm nhị phân	000	001	010	011	100	101	110	111

2.4 Hệ thống số thập lục phân (Hexadecimal system)

- Ký tự số : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- Cơ số: 16

Hệ HEX sử dụng 16 ký tự bao gồm 10 số tự nhiên từ 0 đến 9 và các chữ cái in hoa gồm A, B, C, D, E, F để diễn tả 16 số thập phân từ 0 đến 15.

Lý do dùng hệ thập lục phân là vì một số nhị phân 4 bit có thể diễn tả được $2^4 = 16$ giá trị khác nhau nên rất thuận lợi cho hệ thống số nào đó chỉ dùng một ký tự mà có thể tương ứng với số nhị phân 4 bit, đó là hệ thập lục phân.

Một dãy Hex được biểu diễn như sau: $h_{n-1}h_{n-2} \dots h_2h_1h_0$

Như vậy trong dãy số Hex có n số hạng thì sẽ có 16^n giá trị khác nhau, giá trị nhỏ nhất là 0. . .000 và giá trị lớn nhất là F. . .FFF. Trọng số các bit lần lượt là 1, 16, 256. . . và trọng số của hai số hạng kề nhau chênh lệch nhau 16 lần.

❖ **Chuyển đổi số thập lục phân sang số thập phân**

Ví dụ: $2E_{16} = 2.16^1 + 14.16^0 = 46_{10}$

$$012C, D_{16} = 0.16^3 + 1.16^2 + 2.16^1 + 12.16^0 + 13.16^{-1} \\ = 0 + 256 + 32 + 12 + 0,0625 = 300,065_{10}$$

Ghi chú: nếu số thập lục phân bắt đầu bằng chữ thì khi viết phải thêm số 0 vào trước, ví dụ: EF $\rightarrow$ 0EF.

❖ **Chuyển đổi số thập phân sang số thập lục phân**

- Thực hiện theo quy tắc lấy A_{10} chia cho A_{16} rồi lấy phần dư

❖ **Chuyển đổi thập lục phân sang biểu diễn số nhị phân**

- Thực hiện theo quy tắc biểu diễn một ký số thập lục phân bằng một nhóm tổ hợp 4 bit nhị phân

Ví dụ: Với $A_{16} = 4EFB$ suy ra $A_2 = 0100\ 1110\ 1111\ 1011$

Với $A_{16} = BCD2$ suy ra $A_2 = 1011\ 1100\ 1101\ 0010$

Bảng hình 4.2 mô tả quan hệ giữa hệ thập phân, thập lục phân và nhị phân 4 bit

Thập phân	Thập lục phân	Nhị phân
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010

11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

2.5 Mã BCD (Binary code decimal)

Thông tin được xử lý trên mạch số điều là các số nhị phân nên mọi thông tin dữ liệu dù là số lượng, các chữ, các dấu, các mệnh lệnh sau cùng phải ở dạng nhị phân thì mạch số mới hiểu và xử lý được. Do đó phải qui định cách thức mà các số thập phân dùng để biểu diễn các dữ liệu khác nhau từ đó xuất hiện các mã số. Trước tiên mã thập phân thông dụng nhất là mã BCD (Binary code decimal: mã của số thập phân được mã hóa theo số nhị phân).

Vì ký số thập phân lớn nhất là 9 nên ta cần 4 bit để mã hóa mỗi ký số thập phân

Mỗi số thập phân được đổi sang số nhị phân tương đương và luôn luôn dùng 4 bit cho từng số thập phân

Mã BCD biểu diễn mỗi số thập phân bằng một số nhị phân 4 bit và ta nhận thấy rằng chỉ có các số từ 0000 đến 1001 được sử dụng, ngoài các nhóm số nhị phân 4 bit này không được dùng làm mã BCD.

➤ **Ưu điểm:** Chính của mã BCD là dễ dàng chuyển đổi từ mã thập phân sang nhị phân và ngược lại bằng cách chỉ cần nhớ các nhóm mã 4 bit ứng với các ký số thập phân từ 0 đến 9.

- So sánh mã BCD và mã nhị phân

Ta cần phải hiểu rằng mã BCD không phải là một hệ thống số như hệ thống số thập phân, nhị phân, bát phân và thập lục phân. Mà thật ra, BCD là hệ thập phân với từng ký số được mã hóa thành giá trị nhị phân tương ứng và cũng phải hiểu rằng mã BCD không phải là một mã nhị phân quy ước.

Mã nhị phân quy ước biểu diễn số thập phân hoàn chỉnh ở dạng nhị phân, còn mã BCD chỉ chuyển đổi từng ký số thập phân sang số nhị phân tương ứng

2.6 Mã ASCII

Ngoài dữ liệu dạng số máy tính còn có khả năng thao tác thông tin khác số như mã biểu thị mẫu tự abc, dấu chấm câu, những ký tự đặc biệt cũng như ký tự số. Những mã này được gọi chung là mã chữ số. Bộ mã chữ số hoàn chỉnh bao gồm 26 chữ thường, 26 chữ hoa, 10 ký tự số, 7 dấu chấm câu và chừng độ 20 đến 40 ký tự khác. Ta có thể nói rằng mã chữ số biểu diễn mọi ký tự và chữ số có trên bàn phím máy tính.

Mã chữ số được sử dụng rộng rãi hiện nay là mã ASCII(American Standard Code Information Interchange).

Mã ASCII là bộ mã có 7 bit nên có $2^7 = 128$ nhóm mã đủ để biểu thị tất cả các ký tự trên bàn phím máy tính.

Ngoài dữ liệu dạng số máy tính còn có khả năng thao tác thông tin khác số như mã biểu thị mẫu tự abc, dấu chấm câu, những ký tự đặc biệt cũng như ký tự số. Những mã này được gọi chung là mã chữ số. Bộ mã chữ số hoàn chỉnh bao gồm 26 chữ thường, 26 chữ hoa, 10 ký tự số, 7 dấu chấm câu và chừng độ 20 đến 40 ký tự khác. Ta có thể nói rằng mã chữ số biểu diễn mọi ký tự và chữ số có trên bàn phím máy tính.

Các phép tính trên hệ thống số

❖ Cộng và trừ hai số nhị phân

- Cộng hai số nhị phân

Như ta đã biết cộng hai số thập phân là hàng đơn vị cộng trước và nếu tổng nhỏ hơn 10 thì viết tổng, nếu tổng lớn hơn 10 thì phải viết hàng đơn vị và nhớ 1 cho lần cộng kế trên.

Trong phép cộng nhị phân cũng tạo ra số nhớ. Đầu tiên cộng hai bit nhị phân có nghĩa ít nhất (LSB) nếu kết quả cộng hai bit ≤ 1 thì viết kết quả và nếu kết quả cộng hai bit > 1 thì phải có nhớ vào kết quả của phép cộng ở bit kế tiếp.

- Quy tắc cộng hai số nhị phân một bit như sau:

$$0 + 0 = 0; \quad 0 + 1 = 1; \quad 1 + 1 = 0 \text{ nhớ } 1$$

• Trừ hai số nhị phân:

Trong phép trừ nếu số bị trừ nhỏ hơn số trừ, cụ thể là 0 trừ đi 1, thì phải mượn 1 ở hàng cao kế mà là 2 ở hàng đang trừ và số mượn này phải trả cho hàng cao kế tương tự như phép trừ hai số thập phân.

- Quy tắc trừ hai số nhị phân một bit

$$0 - 0 = 0; \quad 1 - 1 = 0; \quad 1 - 0 = 1; \quad 0 - 1 = 1$$

Đề ý rằng $0 - 1$ không phải là bằng 11 mà là 1 với 1 là số mượn. Khi trừ hai số nhiều bit thì mượn ở hàng nào thì phải cộng vào với số trừ của hàng đó trước khi thực hiện việc trừ.

❖ Nhân và chia hai số nhị phân

- Quy tắc nhân hai số nhị phân một bit

Cần lưu ý:

$$\begin{aligned} 0 \times 0 &= 0 \\ 0 \times 1 &= 0 \\ 1 \times 1 &= 1 \end{aligned}$$

- Quy tắc nhân hai số nhị phân một bit

Ví dụ: Thực hiện phép chia 1001100100 cho 11000

Lần chia đầu tiên, 5 bit của số bị chia nhỏ hơn số chia nên ta được kết quả là 0, sai đó ta lấy 6 bit của số bị chia tiếp (tương ứng với việc dịch phải số chia 1 bit trước khi thực hiện phép trừ)

$$\begin{array}{r} 1001100100 \quad | \quad 11000 \\ - 11000 \quad | \quad \downarrow \downarrow \downarrow \downarrow \downarrow \\ \hline 0011100 \quad | \quad \downarrow \downarrow \downarrow \downarrow \downarrow \\ - 11000 \quad | \quad \downarrow \downarrow \downarrow \downarrow \downarrow \\ \hline 00100100 \quad | \quad \downarrow \downarrow \downarrow \downarrow \downarrow \\ - 11000 \quad | \quad \downarrow \downarrow \downarrow \downarrow \downarrow \\ \hline 011000 \quad | \quad \leftarrow \text{Thêm vào để chia} \\ - 11000 \quad | \quad \text{tiếp lấy phần lẻ} \\ \hline 00000 \end{array}$$

Kết quả ta được: $11001.1_2 = 25.5_{10}$

❖ Cộng và trừ hai số thập lục phân

• Cộng hai số thập lục phân

Khi cộng hai số thập phân nếu tổng lớn hơn 9 thì ta viết con số đơn vị và nhớ số hàng chục lên hàng cao kế. Tương tự như vậy đối với số thập lục phân nếu tổng lớn hơn F (15 trong hệ 10) thì ta viết con số đơn vị và nhớ con số hàng thập lục lên hàng cao kế.

Cộng hai số thập lục phân chỉ có một số

$$\begin{array}{cccc} 8 & 8 & 8 & 8 \\ + 7 & + 8 & + A & + F \\ \hline F & 10 & 12 & 17 \end{array}$$

Ta thấy:

- Trường hợp $8 + 7 = 15$ tương ứng với F
- Trường hợp $8 + 8 = 16$, ta viết $16 - 16 = 0$ và nhớ 1 và kết quả là 10
- Trường hợp $8 + A = 18$, ta viết $18 - 16 = 2$ và nhớ 1 và kết quả là 12
- Trường hợp $8 + F = 23$, ta viết $23 - 16 = 7$ và nhớ 1 và kết quả là 17
- Cùng quy luật trên áp dụng khi cộng hai số Hex có nhiều con số và dĩ nhiên số nhớ cho hàng nào thì phải cộng thêm cho hàng đó.

Trừ hai số thập lục phân

Khi trừ hai số Hex nếu số trừ lớn hơn số bị trừ ta mượn 16 để thêm vào số bị trừ và trả 1 cho số trừ ở hàng cao kế.

❖ **Cộng và trừ hai số BCD**

• Cộng hai số BCD

Cộng hai số BCD khác với cộng hai số nhị phân bình thường. Khi tổng ở mỗi số hạng của số BCD bằng 9 ($= 1001$) hay nhỏ hơn 9 thì đó là kết quả cuối cùng.

Khi tổng hai số nhị phân lớn hơn 9 tức là từ 1010 trở lên thì tổng phải được cộng phải được cộng thêm 6 ($= 0110$) để có tổng là 9 hoặc nhỏ hơn và số nhớ 1 lên hàng BCD có nghĩa cao hơn.

Lý do cộng thêm 6 vì mã BCD không dùng 6 mã cao nhất của số nhị phân 4 bit đó là các mã từ 1010 đến 1111.

• Trừ hai số BCD

Trừ hai số BCD cũng giống như trừ hai số nhị phân nhiều bit. Nếu số bị trừ nhỏ hơn số trừ thì phải mượn 1 ở hàng có nghĩa trên mà là 10 ở hàng đang trừ. Để tiện sắp xếp ta chuyển 1 ở hàng có nghĩa trên thành 10 ở hàng đang trừ rồi cộng vào số bị trừ trước khi thực hiện phép trừ.

✓ **Bài tập:**

1. Biến đổi các số nhị phân sau sang thập phân:
 - a) 10110_2
 - b) 10001101_2
2. Biến đổi các số thập phân sau sang số nhị phân:
 - a) 37
 - b) 14
 - c) 189
3. Biến đổi các số thập lục phân sau sang nhị phân:
 - a) 47_8
 - b) 23_8
 - c) 170_8
4. Biến đổi các số thập phân sau sang bát phân:
 - a) 111
 - b) 97
 - c) 234
5. Biến đổi các số thập phân sau sang thập lục phân:
 - a) 22
 - b) 321
 - c) 2007
6. Hãy chuyển đổi các mã số sau:
 - a. Từ mã Binary sang Hexadecimal: 111001011_2
 - b. Từ mã Hexadecimal sang Octal: ED_H
7. Mã hóa số thập phân dưới đây dùng mã BCD :
a/ 12 b/ 192 c/ 2079 d/ 15436 e/ 0,375 f/ 17,250

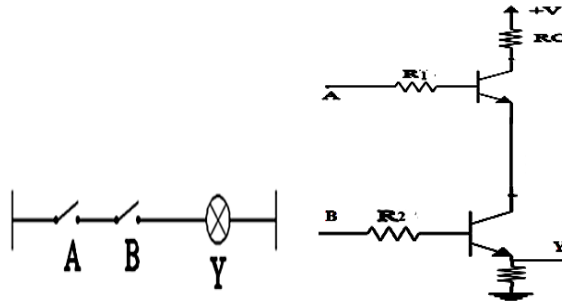
3. Các cổng Logic cơ bản

Trong kỹ thuật điện tử người ta dùng những linh kiện điện tử cần thiết kết nối với nhau theo các quy luật nhất định tạo nên các phần tử cơ bản và từ đó hình thành các mạch chức năng phức tạp hơn. Những phần tử cơ bản này gọi là các cổng logic căn bản.

Một cổng logic căn bản bao gồm một hay nhiều ngõ vào nhưng có duy nhất một ngõ ra và giữa các ngõ vào và ngõ ra biểu thị mối quan hệ với nhau được biểu diễn qua các số nhị phân 0 và 1.

Xét về mức điện áp thì 0 đặc trưng cho điện áp thấp và 1 đặc trưng cho điện áp cao và các cổng logic cơ bản bao gồm các cổng sau.

3.1 Cổng AND



Hình 4.4a

Ký hiệu:

Bảng trạng thái



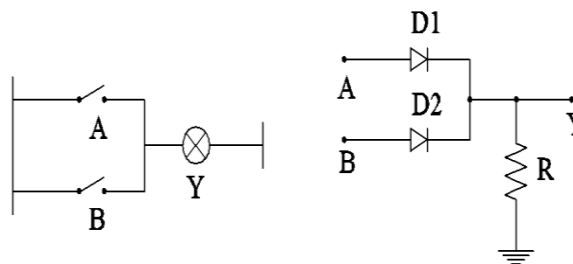
A	B	$Y = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

Hình 4.4b: ký hiệu và bảng trạng thái

Nhận xét:

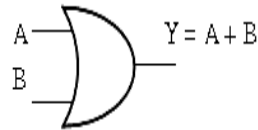
- Cổng AND thực hiện toán nhân thông thường giữa 0 và 1
- Ngõ ra cổng AND bằng 0 khi có ít nhất một ngõ vào bằng 0
- Ngõ ra cổng AND bằng 1 khi tất cả các ngõ vào đều bằng 1

3.2 Cổng OR



Hình 4.5a

Ký hiệu:



Bảng trạng thái

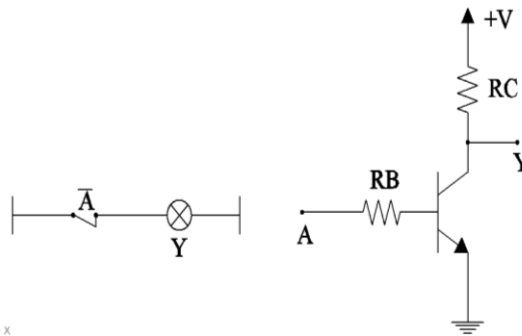
A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Hình 4.5b: ký hiệu và bảng trạng thái

Nhận xét:

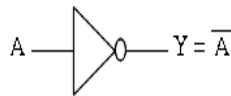
- Ngõ ra cổng OR bằng 0 khi tất cả các ngõ vào bằng 0
- Ngõ ra cổng OR bằng 1 khi có ít nhất một ngõ vào bằng 1

3.3 Cổng NOT



Hình 4.6a

Ký hiệu:



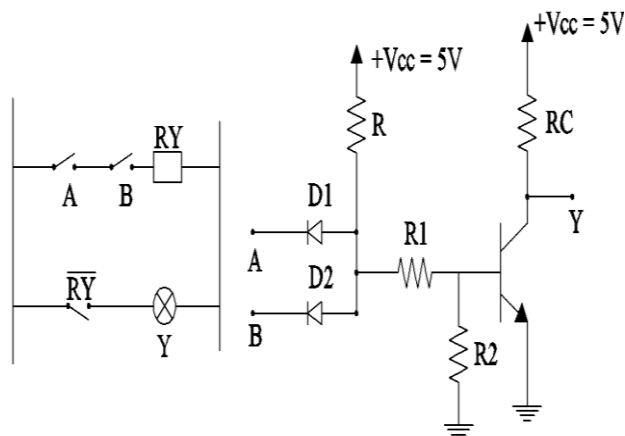
Bảng trạng thái

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

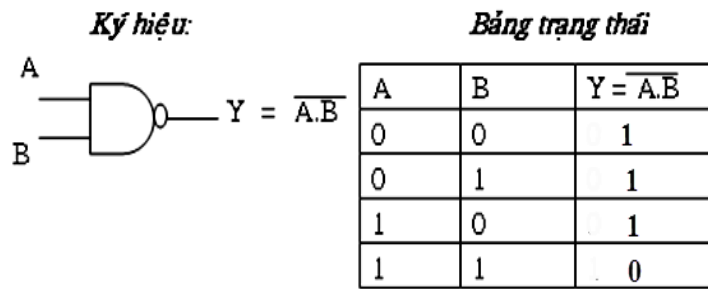
Hình 4.6b: ký hiệu và bảng trạng thái

Nhận xét: Trạng thái ngõ vào và ngõ ra của cổng NOT luôn đối nhau

3.4 Cổng NAND



Hình 4.7a

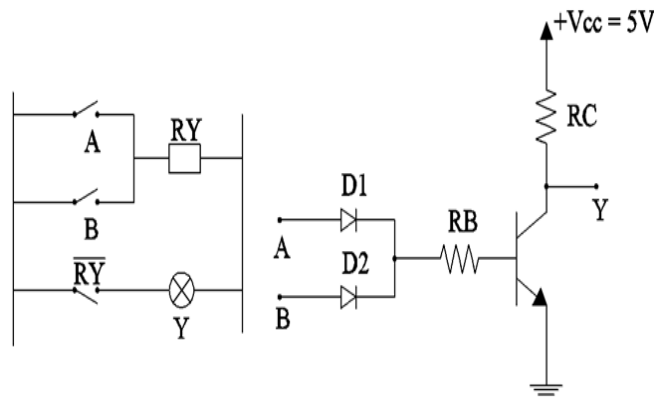


Hình 4.7b: ký hiệu và bảng trạng thái

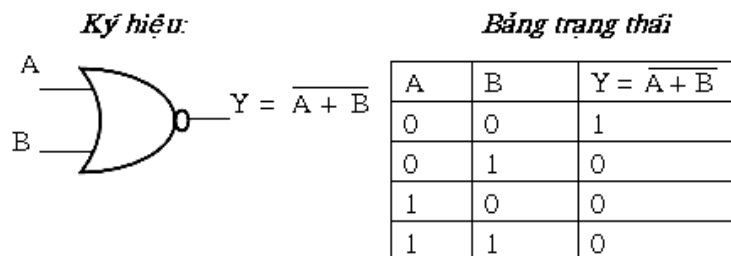
Nhận xét:

- Cổng NAND là đảo trạng thái ngõ ra của cổng AND
- Ngõ ra cổng NAND bằng 0 khi có tất cả các ngõ vào bằng 1
- Ngõ ra cổng AND bằng 1 khi có ít nhất một ngõ vào bằng 0

3.5 Cổng NOR



Hình 4.8a

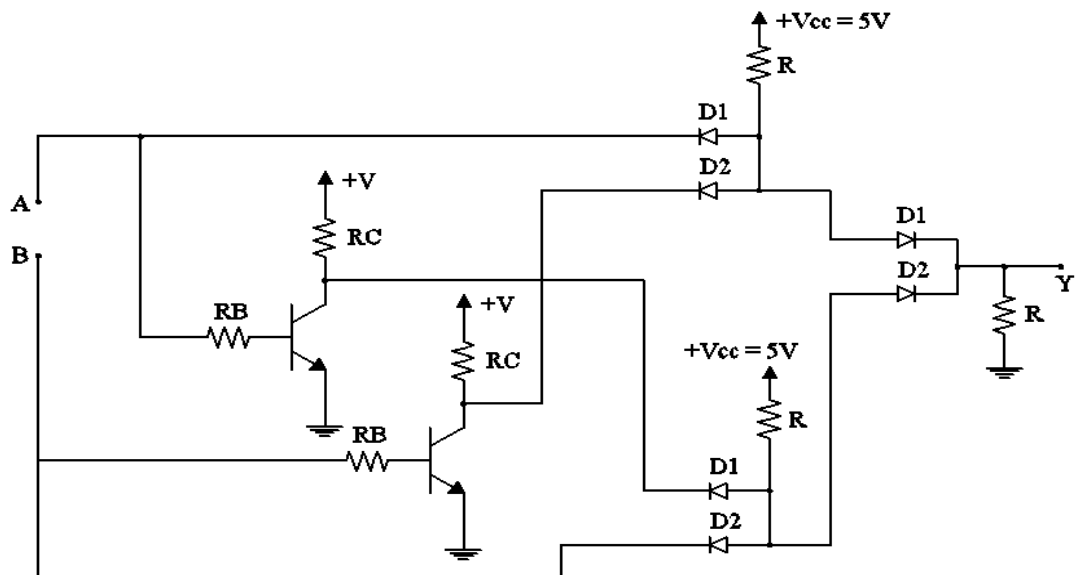


Hình 4.8b: ký hiệu và bảng trạng thái

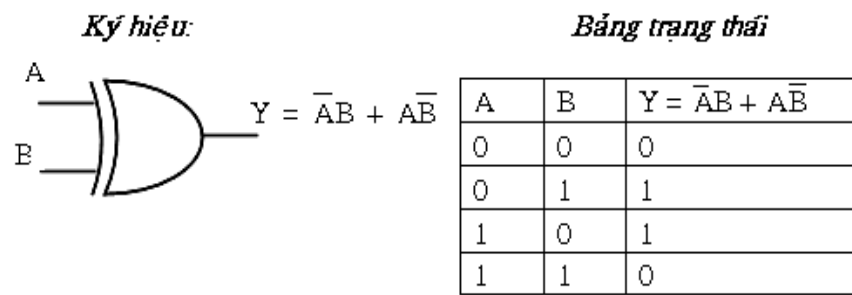
Nhận xét:

- Cổng NOR là đảo của cổng OR
- Ngõ ra cổng NOR bằng 0 khi có ít nhất một ngõ vào bằng 1
- Ngõ ra cổng NOR bằng 1 khi tất cả các ngõ vào bằng 0

3.6 Cổng EX-OR



Hình 4.9a

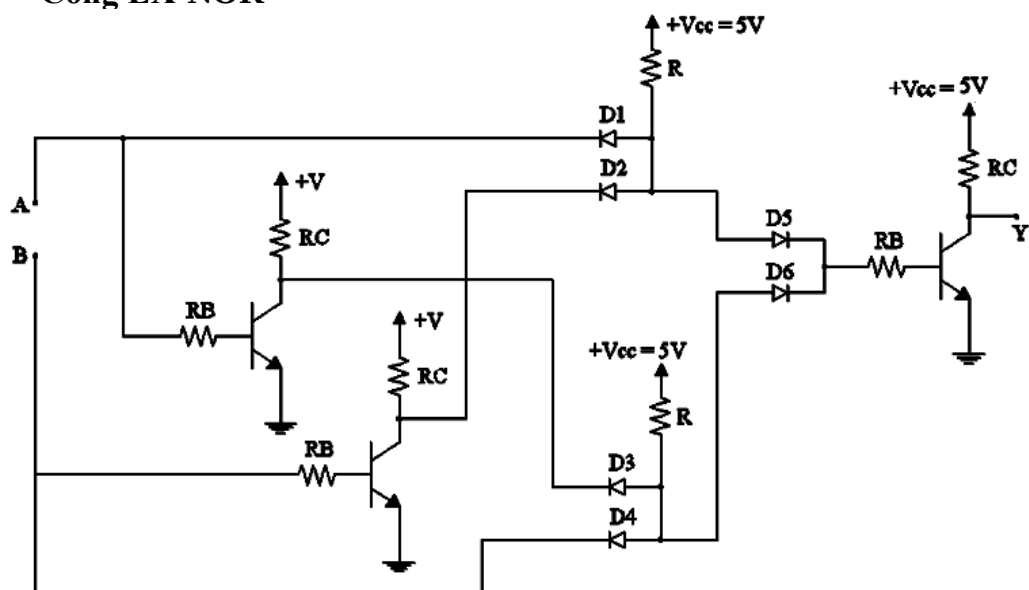


Hình 4.9b: ký hiệu và bảng trạng thái

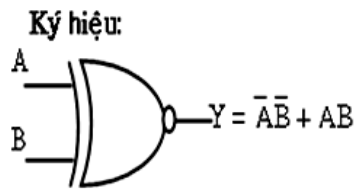
Nhận xét:

- Ngõ ra cổng EX-OR bằng 0 khi tất cả các ngõ vào cùng trạng thái
- Ngõ ra cổng EX-OR bằng 1 khi các ngõ vào khác trạng thái

Cổng EX-NOR



Hình 4.10a



Bảng trạng thái

A	B	$Y = \bar{A}\bar{B} + AB$
0	0	1
0	1	0
1	0	0
1	1	1

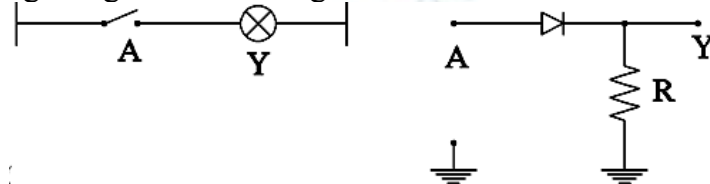
Hình 4.10b: ký hiệu và bảng trạng thái

Nhận xét:

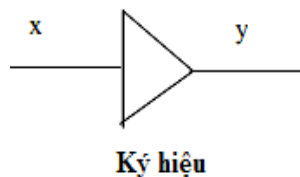
- Ngõ ra cổng EX-NOR chính là đảo của cổng EX-OR
- Ngõ ra cổng EX-NOR bằng 1 khi tất cả các ngõ vào cùng trạng thái
- Ngõ ra cổng EX-NOR bằng 0 khi các ngõ vào khác trạng thái

3.8 Cổng đệm (Buffer)

Cổng đệm (Buffer) hay còn gọi là cổng không đảo là cổng có một ngõ vào và một ký hiệu và bảng trạng thái hoạt động như hình sau



Hình 4.11a



Bảng trạng thái

x	y
0	0
1	1

Hình 4.11b: ký hiệu và bảng trạng thái của cổng đệm

Nhận xét:

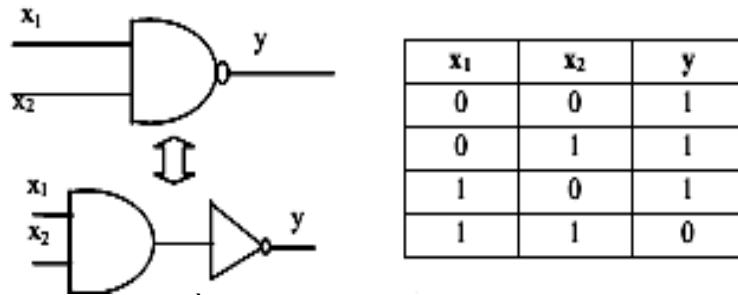
- X là ngõ vào, có trở kháng vào (Z_{in}) vô cùng lớn. Vì vậy dòng vào của cổng đệm rất nhỏ.
- Y là ngõ ra, có trở kháng ra (Z_{out}) rất nhỏ. Vì vậy cổng đệm có khả năng cung cấp dòng ngõ ra lớn.
- Dùng để phối hợp trở kháng vào.
- Dùng để cách ly và nâng dòng cho tải

4. Biểu thức Logic và mạch điện

4.1 Mạch điện biểu diễn biểu thức Logic

❖ Mạch tạo thành các cổng logic từ cổng NAND

Cổng NAND thực hiện phép toán nhân đảo, về sơ đồ Logic cổng NAND gồm 1 cổng AND mắc nối tầng với cổng NOT, ký hiệu và bảng trạng thái cổng NAND được cho như hình 4.16

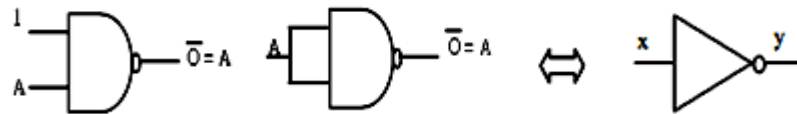


Hình 4.16: ký hiệu, sơ đồ logic tương đương và bảng trạng thái

- Sử dụng cổng NAND để tạo cổng NOT

Ta có thể sử dụng cổng NAND như một cổng NOT bằng cách nối n-1 đầu vào của cổng NAND lên mức 1, ngõ vào còn lại làm ngõ vào của mạch NOT.

Ví dụ: Tạo cổng NOT từ cổng NAND hai ngõ vào như hình 4.12

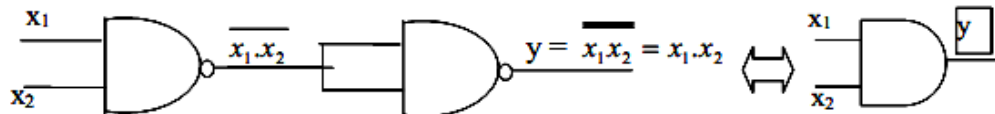


$$y = \overline{x_1 x_2} = \overline{x_1} + \overline{x_2} = \overline{x}$$

Hình 4.12 : Dùng cổng NAND để tạo cổng NOT

- Sử dụng cổng NAND để tạo thành cổng AND

Hàm NAND là đảo của hàm AND, do vậy hàm AND được xây dựng từ hàm NAND bằng cách mắc như hình 4.13

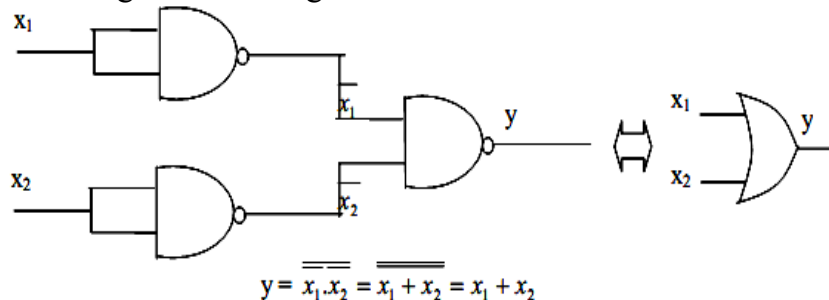


Hình 4.13: Sử dụng cổng NAND để tạo thành cổng AND

- Sử dụng cổng NAND để tạo thành cổng OR

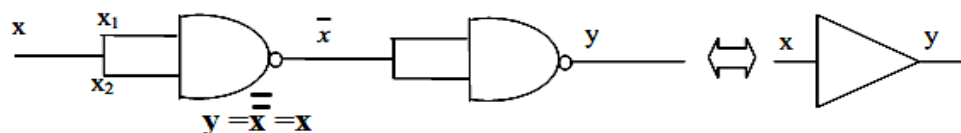
Hàm OR có thể được xây dựng từ các mạch NAND

Ví dụ: Tạo cổng OR có 2 ngõ vào từ cổng NAND, hình 4.14



Hình 4.14: Sử dụng cổng NAND để tạo thành cổng OR

- Sử dụng cổng NAND để tạo thành cổng Buffer (cổng đệm), hình 4.15

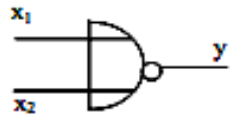


Hình 4.15

❖ **Mạch tạo thành các cổng logic từ cổng NOR**

- Cổng NOR còn gọi là cổng Hoặc – Không, là cổng thực hiện chức năng của phép toán cộng đảo Logic, cổng có hai ngõ vào và một ngõ ra có ký hiệu như hình 4.16

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



Ký hiệu theo Châu Âu



Ký hiệu theo Mỹ, Nhật

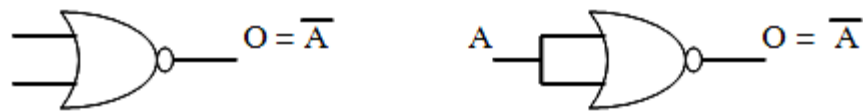
Hình 4.16: ký hiệu cổng NOR

- Bảng trạng thái mô tả hoạt động của cổng NOR, hình 4.17

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

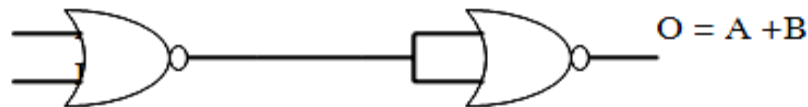
Hình 4.17: bảng trạng thái cổng NOR

- Dùng mạch NOR để tạo hàm NOT, hình 4.18



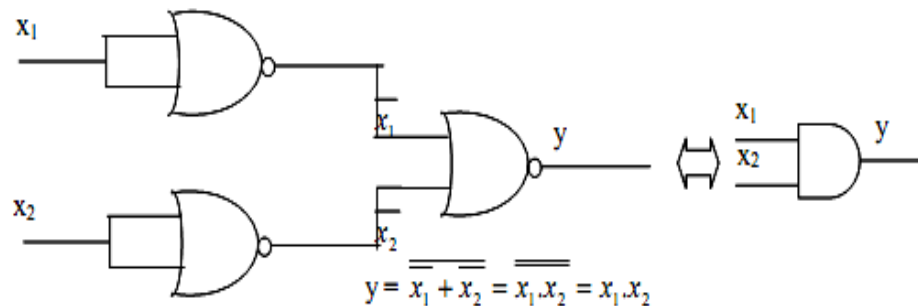
Hình 4.18

- Dùng mạch NOR để tạo hàm OR, hình 4.19



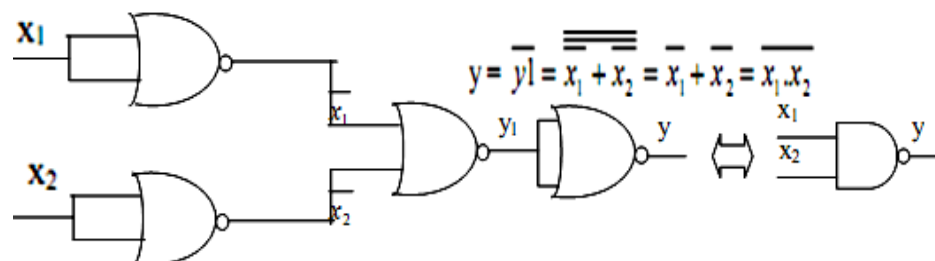
Hình 4.19

- Dùng mạch NOR để tạo hàm AND, hình 4.20



Hình 4.20

- Dùng mạch NOR để tạo hàm AND, hình 4.21



Hình 4.21: Sử dụng cổng NOR làm cổng NAND

❖ Cổng XOR (EX-OR):

Đây là cổng logic thực hiện chức năng của mạch cộng không nhớ, là cổng có hai ngõ vào và một ngõ ra có ký hiệu và bảng trạng thái như hình 4.22



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

Hình 4.22: Cổng XOR (EX-OR):

$$Y_{XOR} = \overline{x_1}x_2 + x_1\overline{x_2} = x_1 \oplus x_2$$

Cổng XOR được dùng để so sánh hai tín hiệu vào:

- Nếu hai tín hiệu là bằng nhau thì tín hiệu ngõ ra bằng 0
- Nếu hai tín hiệu vào là khác nhau thì tín hiệu ngõ ra bằng 1

Các tính chất của phép toán XOR:

1. $X_1 \oplus X_2 = X_2 \oplus X_1$
2. $X_1 \oplus X_2 \oplus X_3 = (X_1 \oplus X_2) \oplus X_3 = X_1 \oplus (X_2 \oplus X_3)$
3. $X_1 (X_2 \oplus X_3) = (X_1 \cdot X_2) \oplus (X_1 \cdot X_3)$

❖ Cổng XNOR (EX-NOR):

Đây là cổng logic thực hiện chức năng của mạch cộng đảo không nhớ, là cổng có hai ngõ vào và một ngõ ra có ký hiệu và bảng trạng thái như hình 4.23



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

Hình 4.23

$$y = \overline{x_1 \overline{x_2} + \overline{x_1} x_2} = \overline{x_1 \oplus x_2}$$

Tính chất của cổng XOR:

1. $\overline{(x_1 \oplus x_2)(x_3 \oplus x_4)} = \overline{(x_1 \oplus x_2)} + \overline{(x_3 \oplus x_4)}$
2. $\overline{(x_1 \oplus x_2)(x_3 \oplus x_4)} = \overline{(x_1 \oplus x_2)} \overline{(x_3 \oplus x_4)}$
3. $\overline{(x_1 \oplus x_2)} = \overline{x_1} \oplus x_2 = x_1 \oplus \overline{x_2}$
4. $\overline{x_1 \oplus x_2} = \overline{x_1} + x_2$
5. $\overline{x_1 \oplus x_2} = x_3 \Leftrightarrow \overline{x_1 \oplus x_3} = x_2$

4.2 Xây dựng biểu thức Logic theo mạch điện cho trước

Ví dụ : Dùng cổng NAND 2 ngõ vào thiết kế mạch tạo hàm $Y = f(A,B,C) = 1$ khi thỏa các điều kiện sau:

- a. $A=0$, $B=1$ và $C=1$
- b. $A=1$, $B=1$ bất chấp C