Chương 1

HỆ THỐNG SỐ ĐẾM VÀ KHÁI NIỆM VỀ MÃ

1.1. HỆ THỐNG SỐ ĐẾM

1.1.1. Hệ đếm

1. Khái niệm

Hệ đếm là tập hợp các phương pháp gọi và biểu diễn các con số bằng các kí hiệu có giá trị số lượng xác định gọi là các chữ số.

2. Phân loại

Có thể chia các hệ đếm làm hai loại: hệ đếm theo vị trí và hệ đếm không theo vị trí.

a. Hệ đếm theo vị trí:

Hệ đếm theo vị trí là hệ đếm mà trong đó giá trị số lượng của chữ số còn phụ thuộc vào vị trí của nó đứng trong con số cụ thể.

Ví dụ: Hệ thập phân là một hệ đếm theo vị trí. Số **1991** trong hệ thập phân được biểu diễn bằng 2 chữ số "1" và "9", nhưng do vị trí đứng của các chữ số này trong con số là khác nhau nên sẽ mang các giá trị số lượng khác nhau, chẳng hạn chữ số "1" ở vị trí hàng đơn vị biểu diễn cho giá trị số lượng là 1 song chữ số "1" ở vị trí hàng nghìn lại biểu diễn cho giá trị số lượng là 1000, hay chữ số "9" khi ở hàng chục biểu diễn giá trị là 90 còn khi ở hàng trăm lại biểu diễn cho giá trị là 900.

b. Hệ đếm không theo vị trí:

Hệ đếm không theo vị trí là hệ đếm mà trong đó giá trị số lượng của chữ số không phụ thuộc vào vị trí của nó đứng trong con số.

Hệ đếm La Mã là một hệ đếm không theo vị trí. Hệ đếm này sử dụng các ký tự "I", "V", "X"... để biểu diễn các con số, trong đó "I" biểu diễn cho giá trị số lượng 1, "V" biểu diễn cho giá trị số lượng 5, "X" biểu diễn cho giá trị số lượng 10... mà không phụ thuộc vào vị trí các chữ số này đứng trong con số cụ thể.

Các hệ đếm không theo vị trí sẽ không được đề cập đến trong giáo trình này.

1.1.2. Cơ số của hệ đếm

Một số A bất kỳ có thể biểu diễn bằng dãy sau:

$$A = a_{m-1}a_{m-2}.....a_0a_{-1}......a_{-n}$$

Trong đó a_i là các chữ số, ($i = \overline{-n \div m-1}$); i là các hàng số, i nhỏ: hàng trẻ, i lớn: hàng già.

Giá trị số lượng của các chữ số a_i sẽ nhận một giá trị nào đó sao cho thỏa mãn bất đẳng thức sau:

$$0 \le a_i \le N-1 \qquad (a_i \text{ nguyên})$$

N được gọi là cơ số của hệ đếm. Cơ số của một hệ đếm là số lượng ký tự phân biệt được sử dụng trong một hệ đếm. Các hệ thống số đếm được phân biệt với nhau bằng một cơ số N của hệ đếm đó. Mỗi ký tự biểu diễn một chữ số.

Trong đời sống hằng ngày chúng ta quen sử dụng hệ đếm thập phân (*decimal*) với N=10. Trong hệ thống số còn sử dụng những hệ đếm khác là hệ đếm nhị phân (*binary*) với N=2, hệ đếm bát phân (*octal*) với N=8 và hệ đếm thập lục phân (*hexadecimal*) với N=16.

- Hệ nhị phân : N =2 $\Rightarrow$ a_i = 0, 1.
- Hệ thập phân : N =10 $\Rightarrow$ a_i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Hệ bát phân : N =8 $\Rightarrow$ a_i = 0, 1, 2, 3, 4, 5, 6, 7.
- Hệ thập lục phân : N =16 $\Rightarrow$ a_i = 0, 1, 2, …8, 9, A, B, C,D, E, F.

Khi đã xuất hiện cơ số N, ta có thể biểu diễn số A dưới dạng một đa thức theo cơ số N, được ký hiệu là $A_{(N)}$:

$$A_{(N)} = a_{m-1}.N^{m-1} + a_{m-2}.N^{m-2} + ... + a_0.N^0 + a_{-1}.N^{-1} + ... + a_{-n}.N^{-n}$$

Hay:

$$A_{(N)} = \sum_{i=-n}^{m-1} a_i N^i \qquad (1.1)$$

Với N=10 (hệ thập phân):

$$A_{(10)} = a_{m-1}.10^{m-1} + a_{m-2}.10^{m-2} + ... + a_0.10^0 + ... + a_{-n}.10^{-n}$$

$$1999,959_{(10)} = 1.10^3 + 9.10^2 + 9.10^1 + 9.10^0 + 9.10^{-1} + 5.10^{-2} + 9.10^{-3}$$

Với N=2 (hệ nhị phân):

$$A_{(2)} = a_{m-1}.2^{m-1} + a_{m-2}.2^{m-2} + ... + a_0.2^0 .... + a_{-n}2^{-n}$$

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

Với N=16 (hệ thập lục phân):

$$A_{(16)} = a_{m-1}.16^{m-1} + a_{m-2}.16^{m-2} + ... + a_0.16^0 + a_{-1}16^{-1} + ... + a_{-n}16^{-n}$$

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

Với N=8 (hệ bát phân):

$$A_{(8)} = a_{m-1}.8^{m-1} + a_{m-2}.8^{m-2} + ... + a_0.8^0 + a_{-1}.8^{-1} + ... + a_{-n}.8^{-n}$$

$$376_{(8)} = 3.8^2 + 7.8^1 + 6.8^0 = 254_{(10)}$$

Như vậy, biểu thức (1.1) cho phép đổi các số ở bất kỳ hệ nào sang hệ thập phân (hệ 10).

1.1.3. Đổi cơ số

1. Đổi từ cơ số d sang cơ số 10

Để chuyển đổi một số ở hệ đếm cơ số d sang hệ đếm cơ số 10 người ta khai triển con số trong cơ số d dưới dạng đa thức theo cơ số của nó (theo biểu thức 1.3).

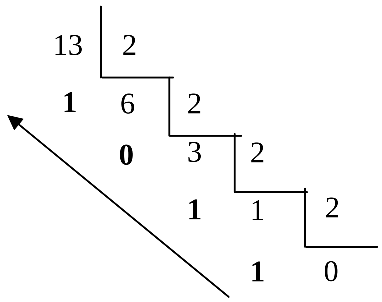
Ví dụ 1.1 Đổi số $1101_{(2)}$ ở hệ nhị phân sang hệ thập phân như sau:

$$1011_{(2)} = 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 11_{(10)}$$

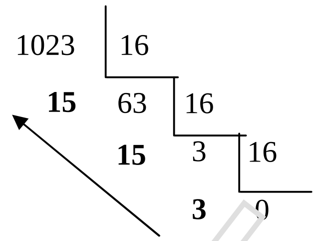
2. Đổi từ cơ số 10 sang cơ số d

Để chuyển đổi một số từ cơ số 10 sang cơ số d (d = 2, 8, 16) người ta lấy con số trong cơ số 10 chia liên tiếp cho d đến khi thương số bằng không thì dừng lại. Kết quả chuyển đổi có được trong hệ đếm cơ số d là tập hợp các số dư của phép chia được viết theo thứ tự ngược lại, nghĩa là số dư đầu tiên có trọng số nhỏ nhất. (xem ví dụ 1.2)

<u>Ví dụ 1.2</u>:



$A_{(10)}=13 \rightarrow A_{(2)}=1101$



$A_{(10)}=1023 \rightarrow A_{(16)}=3FFH$

Kết luận: Gọi d_1, d_2, ..,d_n lần lượt là dư số của phép chia số thập phân cho cơ số d ở lần thứ 1, 2, 3, 4, .., n thì kết quả chuyển đổi một số từ hệ đếm cơ số 10 (thập phân) sang hệ đếm cơ số d sẽ là:

$$\mathbf{d_n d_{n-1} d_{n-2} \ldots d_1,}$$

nghĩa là dư số sau cùng của phép chia là bít có trọng số cao nhất (MSB), còn dư số đầu tiên là bít có trọng số nhỏ nhất (LSB).

Trong các ví dụ trên, cơ số của hệ đếm được ghi ở dạng chỉ số bên dưới. Ngoài ra cũng có thể ký tự chữ để phân biệt như sau:

B - Hệ nhị phân (Binary) O - Hệ bát phân (Octal)
D - Hệ thập phân (Decmal) H - Hệ thập lục phân (Hexadecimal)

Ví dụ: 1010B có nghĩa là $1010_{(2)}$
37FH có nghĩa là $37F_{(16)}$

Quy tắc chuyển đổi giữa các hệ đếm cơ số 2, 8, 16 ?

1.2. HỆ ĐẾM NHỊ PHÂN VÀ KHÁI NIỆM VỀ MÃ

1.2.1. Hệ đếm nhị phân

1. Khái niệm

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

Trong hệ đếm nhị phân người ta quy ước như sau:

- Một nhóm 4 bít gọi là 1 nibble.
- Một nhóm 8 bít gọi là 1 byte.
- Nhóm **nhiều bytes gọi là từ (word),** có thể có từ 2 bytes (16 bít), từ 4 bytes (32 bít), ...

Để hiểu rõ hơn một số khái niệm, ta xét số nhị phân 4 bít: $a_3a_2a_1a_0$. Biểu diễn dưới dạng đa thức theo cơ số của nó là:

$$a_3a_2a_1a_{0\,(2)} = a_3.2^3 + a_2.2^2 + a_1.2^1 + a_0.2^0$$

Trong đó:

- $2^3, 2^2, 2^1, 2^0$ (hay 8, 4, 2, 1) được gọi là các trọng số.
- a_0 được gọi là bit có trọng số nhỏ nhất, hay còn gọi bit có ý nghĩa nhỏ nhất (**LSB** - **L**east **S**ignificant **B**it), còn gọi là bít trẻ nhất.

- a_3 được gọi là bit có trọng số lớn nhất, hay còn gọi là bít có ý nghĩa lớn nhất (**MSB** - **M**ost **S**ignificant **B**it), còn gọi là bít già nhất.

Như vậy, với số nhị phân 4 bit $a_3a_2a_1a_0$ trong đó mỗi chữ số a_i (i từ 0 đến 3) chỉ nhận được hai giá trị {0,1} ta có $\mathbf{2^4 = 16}$ tổ hợp nhị phân phân biệt.

Bảng sau đây liệt kê các tổ hợp mã nhị phân 4 bít cùng các giá trị số thập phân, số bát phân và số thập lục phân tương ứng.

Từ bảng này hãy cho biết mối quan hệ giữa các số trong hệ nhị phân với các số trong hệ bát phân (N=8) và hệ thập lục phân (N=16)? Từ đó suy ra phương pháp chuyển đổi nhanh giữa các hệ này?

Số thập phân	$a_3a_2a_1a_0$	Số bát phân	Số thập lục phân
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	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

Bảng 1.1. Các tổ hợp mã nhị phân 4 bít

Sự chuyển đổi giữa các hệ thống số đếm khác nhau giữ vai trò quan trọng trong máy tính số. Chúng ta biết rằng $2^3 = 8$ và $2^4 = 16$, từ bảng mã trên có thể nhận thấy mỗi chữ số trong hệ bát phân tương đương với một nhóm ba chữ số (3 bít) trong hệ nhị phân, mỗi chữ số trong hệ thập lục phân tương đương với một nhóm bốn chữ số (4 bít) trong hệ nhị phân. Do đó, khi biểu diễn số nhị phân nhiều bit trên máy tính để tránh sai sót người ta thường biểu diễn thông qua số thập phân hoặc thập lục phân hoặc bát phân.

<u>Ví dụ 1.3</u>: Xét việc biểu diễn số nhị phân $1011111011111110_{(2)}$.

1 3 7 3 7 6

1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0

B E F E

Vậy, có thể biểu diễn : $137376_{(8)}$ theo hệ bát phân

hoặc : $BEFE_{(H)}$ theo hệ thập lục phân.

*Với số nhị phân **n bít** có bao nhiêu tổ hợp nhị phân khác nhau? Xét trường hợp số nhị phân **8 bít** (n=8)* $a_7a_6a_5a_4a_3a_2a_1a_0$ *có bao nhiêu tổ hợp nhị phân (từ mã nhị phân) khác nhau?*

2. Các phép tính trên số nhị phân

a. Phép cộng

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

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

Ví dụ 1.4:

+ 3	→	+ 0011
2	→	0010
5	→	$0101 = 1.2^2 + 1.2^0 = 5_{(10)}$

b. Phép trừ

0 - 0 = 0 mượn 0
0 - 1 = 1 mượn 1
1 - 0 = 1 mượn 0
1 - 1 = 0 mượn 0

Ví dụ 1.5:

- 7	→	- 0111
5	→	0101
2	→	$0010 = 0.2^3 + 0.2^2 + 1.2^1 + 0.2^0 = 2_{(10)}$

c. Phép nhân

0 . 0 = 0
0 . 1 = 0
1 . 0 = 0
1 . 1 = 1

Ví dụ 1.6:

```
x 7     →   x   0111
  5     →       0101
 35             0111
               0000
              0111
             0000
            0100011
```

$0100011 = 1.2^5 + 1.2^1 + 1.2^0 = 35_{(10)}$

d. Phép chia

0 : 1 = 0
1 : 1 = 1

Lưu ý: Khi chia số chia phải khác 0

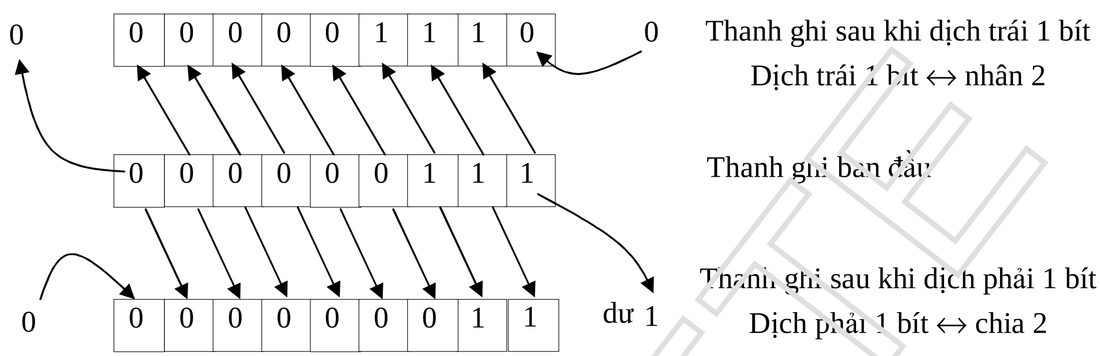
Ví dụ 1.7: 10 | 5 → 1010 | 101

2 101 $10_{(2)} = 2_{(10)}$

00

0

Ứng dụng thanh ghi dịch thực hiện phép toán nhân hai, chia hai:



Hình 1.1. Ứng dụng thanh ghi dịch thực hiện phép toán nhân và chia 2

1.2.2. Khái niệm về mã

1. Đại cương

Trong đời sống hàng ngày, con người giao tiếp với nhau thông qua một hệ thống ngôn ngữ qui ước, nhưng trong máy tính và các hệ thống số chỉ xử lý các dữ liệu nhị phân. Do đó, một vấn đề đặt ra là làm thế nào tạo ra một giao diện dễ dàng giữa người và máy tính, nghĩa là máy tính thực hiện được những bài toán do con người đặt ra.

Vì các máy tính số hiện nay chỉ hiểu các số 0 và số 1, nên bất kỳ thông tin nào dưới dạng các chữ số, chữ cái hoặc các ký tự phải được biến đổi thành dạng số nhị phân trước khi nó có thể được xử lý bằng các mạch số.

Để thực hiện điều đó, người ta đặt ra vấn đề về mã hóa dữ liệu. Như vậy, mã hóa là quá trình biến đổi những ký hiệu quen thuộc của con người sang những ký hiệu quen thuộc với máy tính. Những số liệu đã mã hóa này được nhập vào máy tính, máy tính tính toán xử lý và sau đó máy tính thực hiện quá trình ngược lại là giải mã để chuyển đổi các bít thông tin nhị phân thành các ký hiệu quen thuộc với con người mà con người có thể hiểu được.

Các lĩnh vực mã hóa bao gồm:

- Mã hóa số thập phân
- Mã hóa ký tự
- Mã hóa tập lệnh
- Mã hóa tiếng nói
- Mã hóa hình ảnh ..v..v..

Phần tiếp theo chúng ta khảo sát lĩnh vực mã hóa đơn giản nhất là mã hóa số thập phân bằng cách sử dụng các từ mã nhị phân. Việc mã hóa ký tự, tập lệnh, tiếng nói, hình ảnh... đều dựa trên cơ sở mã hóa số thập phân.

2. Mã hóa số thập phân

a. Khái niệm

Trong thực tế để mã hóa số thập phân người ta sử dụng các số nhị phân 4 bit ($a_3a_2a_1a_0$) theo quy tắc sau:

0 $\rightarrow$ 0000 ; 5 $\rightarrow$ 0101
1 $\rightarrow$ 0001 ; 6 $\rightarrow$ 0110
2 $\rightarrow$ 0010 ; 7 $\rightarrow$ 0101
3 $\rightarrow$ 0011 ; 8 $\rightarrow$ 1000
4 $\rightarrow$ 0100 ; 9 $\rightarrow$ 1001

Các số nhị phân dùng để mã hóa các số thập phân được gọi là các số BCD (Binary Coded Decimal: Số thập phân được mã hóa bằng số nhị phân).

b. Phân loại

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

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

Mặc dù tồn tại nhiều loại mã BCD khác nhau, nhưng có thể chia làm hai loại chính: *Mã BCD có trọng số* và *mã BCD không có trọng số*.

b1. Mã BCD có trọng số là loại mã cho phép phân tích thành đa thức theo trọng số của nó. Mã BCD có trọng số được chia làm 2 loại là: mã BCD tự nhiên và mã BCD số học.

Mã BCD tự nhiên là loại mã mà trong đó các trọng số thường được sắp xếp theo thứ tự tăng dần. Ví dụ: Mã BCD 8421, BCD 5421.

Mã BCD số học là loại mã mà trong đó có tổng các trọng số luôn luôn bằng 9.Ví dụ: BCD 2421, BCD 5121, BCD8 4-2-1

Đặc trưng của mã BCD số học là có tính chất đối xứng qua một đường trung gian. Do vậy, để tìm từ mã BCD của một số thập phân nào đó ta lấy bù (đảo) từ mã BCD của số bù 9 tương ứng.

Ví dụ xét mã BCD 2421. Đây là mã BCD số học (tổng các trọng số bằng 9), trong đó số 3 (thập phân) có từ mã la 0011, số 6 (thập phân) là bù 9 của 3. Do vậy, có thể suy ra từ mã của 6 bằng cách lấy bù từ mã của 3, nghĩa là lấy bù 0011, ta sẽ có từ mã của 6 là 1100.

b2. Mã BCD không có trọng số là loại mã không cho phép phân tích thành đa thức theo trọng số của nó. Các mã BCD không có trọng số là: Mã Gray, Mã Gray thừa 3.

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

Ví dụ: Mã Gray: 2 $\rightarrow$ 0011
3 $\rightarrow$ 0010
4 $\rightarrow$ 0110

Còn với mã BCD 8421:
3 $\rightarrow$ 0011
4 $\rightarrow$ 0100

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

Bảng 1.2: *Các mã BCD tự nhiên.*

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

Bảng 1.3: *Các mã BCD số học*

BCD 2421				BCD 5121				BCD 84-2-1				Số thập phân
a_3	a_2	a_1	a_0	b_3	b_2	b_1	b_0	c_3	c_2	c_1	c_0	
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	1	0	2
0	0	1	1	0	0	1	1	0	1	0	1	3
0	1	0	0	0	1	1	1	0	1	0	0	4
1	0	1	1	1	0	0	0	1	0	1	1	5

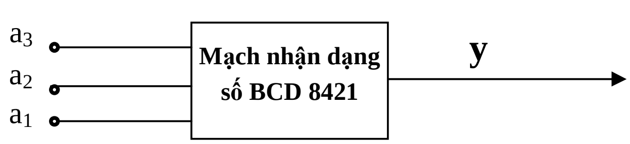
1	1	0	0	1	1	0	0	1	0	1	0	6
1	1	0	1	1	1	0	1	1	0	0	1	7
1	1	1	0	1	1	1	0	1	0	0	0	8
1	1	1	1	1	1	1	1	1	1	1	1	9

Bảng 1.4: *BCD tự nhiên và mã Gray.*

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

Chú ý: Mã Gray được suy ra từ mã BCD 8421 bằng cách: các bit 0,1 đứng sau bit 0 (ở mã BCD 8421) khi chuyển sang mã Gray được giữ nguyên, còn các bit 0,1 đứng sau bit 1 (ở mã BCD 8421) khi chuyển sang mã Gray thì đảo bít, nghĩa là từ bit 1 thành bit 0 và bit 0 thành bit 1.

3. Mạch nhận dạng số BCD 8421:



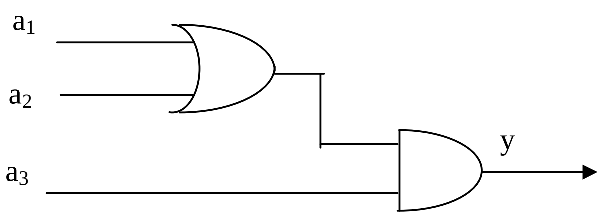
Mạch nhận dạng số BCD 8421 nhận tín hiệu vào là các bít a_3, a_2, a_1 của số nhị phân 4 bít $a_3a_2a_1a_0$, đầu ra y được quy định như sau:

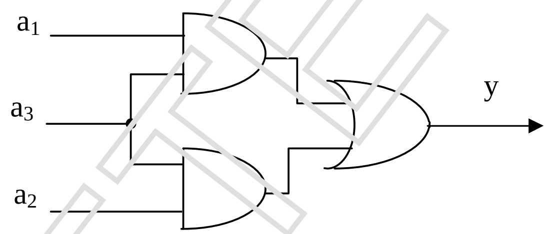
- Nếu y = 1 thì $a_3a_2a_1a_0$ không phải số BCD 8421
- Nếu y = 0 thì $a_3a_2a_1a_0$ là số BCD 8421

Như vậy, nếu một số nhị phân 4 bit không phải là một số BCD 8421 thì ngõ ra y = 1. Từ bảng 1.1 ta thấy một số nhị phân 4 bít không phải là số BCD 8421 khi ***bít a_3 luôn luôn bằng 1 và (bit a_1 bằng 1 hoặc bít a_2 bằng 1)***.

Suy ra phương trình logic của ngõ ra y: $y = a_3(a_1 + a_2) = a_3a_1 + a_3a_2$

Sơ đồ logic:





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

Để nhập số BCD thập phân hai chữ số thì máy tính chia số thập phân thành các đềcác và mỗi đềcác được biểu diễn bằng số BCD tương ứng. Chẳng hạn: $\mathbf{11_{(10)}}$ có thể được nhập vào máy tính theo 2 cách:

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

4. Các phép tính trên số BCD

a. Phép cộng

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

Ví dụ 1.8 Cộng 2 số BCD một đềcác:

```
+  5  → +  0101          +  7  →      +  0111
   3  →    0011             5  →         0101
   8       1000            12          + 1100
                                          0110  ←——  Số hiệu chỉnh
                                      0001 0010
```

Có hai trường hợp phải hiệu chỉnh kết quả của phép cộng 2 số BCD 8421:

- *Khi kết quả của phép cộng là một số không phải là số BCD 8421*
- *Khi kết quả của phép cộng là một số BCD 8421 nhưng lại xuất hiện số nhớ bằng 1.*

Việc hiệu chỉnh được thực hiện bằng cách cộng kết quả với số hiệu chỉnh là 6 (0110_2).

Ở *ví dụ 1.8* đã xem xét trường hợp hiệu chỉnh khi kết quả không phải là một số BCD 8421. Trường hợp hiệu chỉnh khi kết quả là một số BCD 8421 nhưng phép cộng lại xuất hiện số nhớ bằng 1 được xem xét trong ví dụ sau đây:

Ví dụ 1.9 Hiệu chỉnh kết quả cộng 2 số BCD một đềcác khi xuất hiện số nhớ bằng 1:

	8	$\rightarrow$	1000	
+	9	$\rightarrow$	+ 1001	
	17		1 0001	Kết quả là số BCD 8421 nhưng lại xuất hiện số nhớ bằng 1
			0110	Số hiệu chỉnh (6)
			0001 0111	Kết quả sau khi hiệu chỉnh là **17**

b. Phép trừ

Phép toán trừ 2 số BCD được thực hiện theo quy tắc sau đây:

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

Trong đó $\overline{B}$ là số bù 2 của B.

Ví dụ 1.10 Thực hiện trừ 2 số BCD một đềcác:

- 7	$\rightarrow$	- 0111	0111	
5	$\rightarrow$	0101	+ 1010	Bù 1 của 5
2		0010	1 0001	Bỏ đi số nhớ
			+ 1	Cộng 1 LSB để có bù 2 của 5
			0010	Kết quả cuối cùng

Lưu ý:

- *Bù 1 của một số nhị phân là lấy đảo tất cả các bít của số đó (bit 0 thành 1, bit 1 thành 0).*
- *Bù 2 của một số nhị phân bằng số bù 1 cộng thêm 1 vào bít LSB.*

Xét các trường hợp mở rộng sau đây:

1. Thực hiện trừ 2 số BCD 1 đềcác mà số bị trừ nhỏ hơn số trừ ?

2. Mở rộng cho cộng và trừ 2 số BCD nhiều đềcác ?