CHƯƠNG 1: CÁC HỆ THỐNG SỐ & MÃ

❁ **NGUYÊN LÝ CỦA VIỆC VIẾT SỐ**
❁ **CÁC HỆ THỐNG SỐ**
- Hệ cơ số 10 (thập phân)
- Hệ cơ số 2 (nhị phân)
- Hệ cơ số 8 (bát phân)
- Hệ cơ số 16 (thâp lục phân)

❁ **BIẾN ĐỔI QUA LẠI GIỮA CÁC HỆ THỐNG SỐ**
- Đổi từ hệ b sang hệ 10
- Đổi từ hệ 10 sang hệ b
- Đổi từ hệ b sang hệ b^k & ngược lại
- Đổi từ hệ b^k sang hệ b^p

❁ **CÁC PHÉP TOÁN Số NHị PHÂN**
- Phép cộng
- Phép trừ
- Phép nhân
- Phép chia

❁ **MÃ HÓA**
- Mã BCD
- Mã Gray

Nhu cầu về định lượng trong quan hệ giữa con người với nhau, nhất là trong những trao đổi thương mại, đã có từ khi xã hội hình thành. Đã có rất nhiều cố gắng trong việc tìm kiếm các vật dụng, các ký hiệu . . . dùng cho việc định lượng này như các que gỗ, vỏ sò, số La mã . . . Hiện nay số Ả rập tỏ ra có nhiều ưu điểm khi được sử dụng trong định lượng, tính toán. . . ..

Việc sử dụng hệ thống số hằng ngày trở nên quá quen thuộc khiến chúng ta có thể đã quên đi sự hình thành và các qui tắc để viết các con số.

Chương này nhắc lại một cách sơ lược nguyên lý của việc viết số và giới thiệu các hệ thống số khác ngoài hệ thống thập phân quen thuộc, phương pháp biến đổi qua lại của các số trong các hệ thống khác nhau. Chúng ta sẽ đặc biệt quan tâm đến hệ thống nhị phân là hệ thống được dùng trong lãnh vực điện tử-tin học như là một phương tiện để giải quyết các vấn đề mang tính logic.

Phần cuối của chương sẽ giới thiệu các loại mã thông dụng để chuẩn bị cho các chương kế tiếp.

1.1 Nguyên lý của việc viết số

Một số được viết bằng cách đặt kề nhau các **ký hiệu**, được chọn trong một tập hợp xác định. Mỗi ký hiệu trong một số được gọi là **số mã** (số hạng, digit).

Thí dụ, trong hệ thống thập phân (cơ số 10) tập hợp này gồm 10 ký hiệu rất quen thuộc, đó là các con số từ 0 đến 9:

$$S_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Khi một số gồm nhiều số mã được viết, giá trị của các số mã tùy thuộc vị trí của nó trong số đó. Giá trị này được gọi là **trọng số** của số mã.

Thí dụ số 1998 trong hệ thập phân có giá trị xác định bởi triển khai theo đa thức của 10:

$1998_{10} = 1x10^3 + 9x10^2 + 9x10^1 + 9x10^0 = 1000 + 900 + 90 + 8$

Trong triển khai, số mũ của đa thức chỉ vị trí của một ký hiệu trong một số với qui ước vị trí của hàng đơn vị là 0, các vị trí liên tiếp về phía trái là 1, 2, 3, ... Nếu có phần lẻ, vị trí đầu tiên sau dấu phẩy là -1, các vị trí liên tiếp về phía phải là -2, -3, ...

Ta thấy, số 9 đầu tiên (sau số 1) có trọng số là 900 trong khi số 9 thứ hai chỉ là 90.

Có thể nhận xét là với 2 ký hiệu giống nhau trong hệ 10, ký hiệu đứng trước có trọng số gấp 10 lần ký hiệu đứng ngay sau nó. Điều này hoàn toàn đúng cho các hệ khác, thí dụ, đối với hệ nhị phân (cơ số 2) thì tỉ lệ này là 2.

Tổng quát, một hệ thống số được gọi là **hệ b** sẽ gồm b ký hiệu trong một tập hợp:

$$S_b = \{S_0, S_1, S_2, \ldots, S_{b-1}\}$$

Một số N được viết:

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

Sẽ có giá trị:

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

$$= \sum_{i=-m}^{n} a_i b^i$$

$a_i b^i$ chính là trọng số của một ký hiệu trong S_b ở vị trí thứ i.

1.2 Các hệ thống số

1.2.1 Hệ cơ số 10 (thập phân, Decimal system)

Hệ thập phân là hệ thống số rất quen thuộc, gồm 10 số mã như nói trên.

Dưới đây là vài ví dụ số thập phân:

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

$N = 3{,}14_{10} = 3x10^0 + 1x10^{-1} + 4x10^{-2} = 3x1 + 1x1/10 + 4x1/100$

1.2.2 Hệ cơ số 2 (nhị phân, Binary system)

Hệ nhị phân gồm hai số mã trong tập hợp

$$S_2 = \{0, 1\}$$

Mỗi số mã trong một số nhị phân được gọi là một **bit** (viết tắt của binary digit).

Số N trong hệ nhị phân:

$N = (a_n a_{n-1} a_{n-2} \ldots a_i \ldots a_0 , a_{-1} a_{-2} \ldots a_{-m})_2$ (với $a_i \in S_2$)

Có giá trị là:

$N = a_n 2^n + a_{n-1}2^{n-1} + \ldots + a_i 2^i + \ldots + a_0 2^0 + a_{-1} 2^{-1} + a_{-2} 2^{-2} + \ldots + a_{-m}2^{-m}$

a_n là bit có **trọng số lớn nhất**, được gọi là bit **MSB** (Most significant bit) và a_{-m} là bit có **trọng số nhỏ nhất**, gọi là bit **LSB** (Least significant bit).

Thí dụ: $N = 1010{,}1_2 = 1x2^3 + 0x2^2 + 1x2^1 + 0x2^0 + 1x2^{-1} = 10{,}5_{10}$

1.2.3 Hệ cơ số 8 (bát phân ,Octal system)

Hệ bát phân gồm tám số trong tập hợp

$$S_8 = \{0,1, 2, 3, 4, 5, 6, 7\}.$$

Số N trong hệ bát phân:

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

Có giá trị là:

$N = a_n 8^n + a_{n-1}8^{n-1} + a_{n-2}8^{n-2} +. . + a_i8^i . . .+a_08^0 + a_{-1} 8^{-1} + a_{-2} 8^{-2} +. . .+ a_{-m}8^{-m}$

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

1.2.4 Hệ cơ số 16 (thập lục phân, Hexadecimal system)

Hệ thập lục phân được dùng rất thuận tiện để con người giao tiếp với máy tính, hệ này gồm mười sáu số trong tập hợp

$S_{16} =\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F \}$

(A tương đương với 10_{10} , B =11_{10} , ... , F=15_{10}) .

Số N trong hệ thập lục phân:

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

Có giá trị là:

$N = a_n 16^n + a_{n-1}16^{n-1} + a_{n-2}16^{n-2} +. . + a_i16^i . . .+a_016^0+ a_{-1} 16^{-1} + a_{-2} 16^{-2} +. . .+ a_{-m}16^{-m}$

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

Thí dụ: $N = 20EA,8H = 20EA,8_{16} = 2x16^3 + 0x16^2 + 14x16^1 + 10x16^0 + 8x16^{-1}$
$= 4330,5_{10}$

1.3 Biến đổi qua lại giữa các hệ thống số

Khi đã có nhiều hệ thông số, việc xác định giá trị tương đương của một số trong hệ này so với hệ kia là cần thiết. Phần sau đây cho phép ta biến đổi qua lại giữa các số trong bất cứ hệ nào sang bất cứ hệ khác trong các hệ đã được giới thiệu.

1.3.1 Đổi một số từ hệ b sang hệ 10

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

Một số N trong hệ b:

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

Có giá trị tương đương trong hệ 10 là:

$N = a_n b^n + a_{n-1}b^{n-1} +. . .+ a_ib^i +. . . + a_0b^0+ a_{-1} b^{-1} + a_{-2} b^{-2} +. . .+ a_{-m}b^{-m}$.

Thí dụ:

* Đổi số $10110,11_2$ sang hệ 10

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

* Đổi số 4BE,ADH sang hệ 10

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

1.3.2 Đổi một số từ hệ 10 sang hệ b

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

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

$N = (a_na_{n-1} . . .a_0 , a_{-1}a_{-2} . . .a_{-m})_b = (a_na_{n-1} . . .a_0)_b + (0,a_{-1}a_{-2} . . .a_{-m})_b$

Trong đó

$(a_na_{n-1} . . .a_0)_b$ = PE(N) là phần nguyên của N

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

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

◆ **Phần nguyên**:

Giá trị của phần nguyên xác định nhờ triển khai:

$$PE(N) = a_nb^n + a_{n-1}b^{n-1} + \ldots + a_1b^1 + a_0b^0$$

Hay có thể viết lại

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

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

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

Lặp lại bài toán chia PE'(N) cho b:

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

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

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

◆ **Phần lẻ**:

Giá trị của phần lẻ xác định bởi:

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

Hay viết lại

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

Nhân PF(N) với b, ta được : $bPF(N) = a_{-1} + (a_{-2}b^{-1} + \ldots + a_{-m}b^{-m+1}) = a_{-1} + PF'(N)$.

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

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

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

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

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

Thí dụ:

* Đổi $25,3_{10}$ sang hệ nhị phân

Phần nguyên: 25 : 2 = 12 dư 1 ⇒ $a_0 = 1$

12 : 2 = 6 dư 0 ⇒ $a_1 = 0$

6 : 2 = 3 dư 0 ⇒ $a_2 = 0$

3 : 2 = 1 dư 1 ⇒ $a_3 = 1$

thương số cuối cùng là 1 cũng chính là bit a_4:

⇒ $a_4 = 1$

Vậy PE(N) = 11001

Phần lẻ: 0,3 * 2 = 0,6 ⇒ $a_{-1} = 0$

0,6 * 2 = 1,2 ⇒ $a_{-2} = 1$

0,2 * 2 = 0,4 ⇒ $a_{-3} = 0$

0,4 * 2 = 0,8 ⇒ $a_{-4} = 0$

0,8 * 2 = 1,6 ⇒ $a_{-5} = 1$ . . .

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

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

PF(N) = 0,01001.

Kết quả cuối cùng là:

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

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

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

86 : 16 = 5 số dư = 6 $\Rightarrow a_1 = 6$ & $\Rightarrow a_2 = 5$

1376_{10} = 560H

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

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

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

Nếu chỉ cần lấy 3 số lẻ: $0,85_{10}$= 0,D99H

Và kết quả cuối cùng:

$1376,85_{10}$ = 560,D99H

1.3.3 Đổi một số từ hệ b sang hệ b^k và ngược lại

Từ cách triển khai đa thức của số N trong hệ b, ta có thể nhóm thành từng k số hạng từ dấu phẩy về hai phía và đặt thành thừa số chung

$$N = a_nb^n + \ldots + a_5b^5 + a_4b^4 + a_3b^3 + a_2b^2 + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + a_{-2}b^{-2} + a_{-3}b^{-3} \ldots + a_{-m}b^{-m}$$

Để dễ hiểu, chúng ta lấy thí dụ k = 3, N được viết lại bằng cách nhóm từng 3 số hạng, kể từ dấu phẩy về 2 phía

$$N = \ldots + (a_5b^2 + a_4b^1 + a_3b^0)b^3 + (a_2b^2 + a_1b^1 + a_0b^0)b^0 + (a_{-1}b^2 + a_{-2}b^1 + a_{-3}b^0)b^{-3} + \ldots$$

Phần chứa trong mỗi dấu ngoặc luôn luôn nhỏ hơn b^3 , vậy số này tạo nên một số trong hệ b^3 và lúc đó được biểu diễn bởi ký hiệu tương ứng trong hệ này.

Thật vậy, số N có dạng:

$N = \ldots + A_2B^2 + A_1B^1 + A_0B^0 + A_{-1}B^{-1} + \ldots$

Trong đó:

$B=b^3$ ($B^0=b^0$; $B^1=b^3$; $B^2=b^6$, $B^{-1}=b^{-3}$ ....)

$A_2= a_8b^2 + a_7b^1 + a_6b^0 = b^3(a_8b^{-1} + a_7b^{-2} + a_6b^{-3}) < B=b^3$

$A_1= a_5b^2 + a_4b^1 + a_3b^0 = b^3(a_5b^{-1} + a_4b^{-2} + a_3b^{-3}) < B=b^3$

$A_0= a_2b^2 + a_1b^1 + a_0b^0 = b^3(a_2b^{-1} + a_1b^{-2} + a_0b^{-3}) < B=b^3$

Các số A_i luôn luôn nhỏ hơn $B=b^3$ như vậy nó chính là một phần tử của tập hợp số tạo nên hệ $B=b^3$

Ta có kết quả biến đổi tương tự cho các hệ số k khác.

Tóm lại, để đổi một số từ hệ b sang hệ b^k, từ dấu phẩy đi về hai phía, ta nhóm từng k số hạng, giá trị của mỗi k số hạng này (tính theo hệ b) chính là số trong hệ b^k .

Thí dụ:

* Đổi số N = 10111110101 , 01101_2 sang hệ 8 = 2^3

Từ dấu phẩy, nhóm từng 3 số hạng về hai phía (nếu cần, thêm số 0 vào ở nhóm đầu và cuối để đủ 3 số hạng mà không làm thay đổi giá trị của số N):

N = 010 111 110 101 , 011 010_2

Ghi giá trị tương ứng của các số 3 bit, ta được số N trong hệ 8

N = 2 7 6 5 , 3 2_8

* Đổi số N trên sang hệ 16 = 2^4

Cũng như trên nhưng nhóm từng 4 số hạng

$N = 0101\ \ 1111\ \ 0101\ ,\ 0110\ \ 1000_2$

$N = \ \ 5\ \ \ \ F\ \ \ \ 5\ \ ,\ \ 6\ \ \ \ 8_{16}$

Từ kết quả của phép đổi số từ hệ b sang hệ b^k, ta có thể suy ra cách biến đổi ngược một cách dễ dàng: Thay mỗi số hạng của số trong hệ b^k bằng một số gồm k số hạng trong hệ b.

Thí dụ để đổi số $N = 5\ F5,\ 68_{16}$ (hệ 2^4) sang hệ nhị phân (2) ta dùng 4 bit để viết cho mỗi số hạng của số này:

$N = 0101\ \ 1111\ \ 0101\ ,\ 0110\ \ 1000_2$

1.3.4 Đổi một số từ hệ b^k sang hệ b^p

Qua trung gian của hệ b, ta có thể đổi từ hệ b^k sang hệ b^p. Muốn đổi số N từ hệ b^k sang hệ b^p, trước nhất đổi số N sang hệ b rồi từ hệ b tiếp tục đổi sang hệ b^p.

Thí dụ:

- Đổi số $1234,67_8$ sang hệ 16

$1234,67_8 = 001\ 010\ 011\ 100,110\ 111_2 = 0010\ 1001\ 1100,1101\ 1100_2 = 29C,DCH$

- Đổi số ABCD,EFH sang hệ 8

$ABCD,EFH = 1010\ 1011\ 1100\ 1101,1110\ 1111_2 = 1\ 010\ 101\ 111\ 001\ 101,111\ 011\ 110_2 = 125715,736_8$

Dưới đây là bảng kê các số đầu tiên trong các hệ khác nhau:

Thập phân	Nhị phân	Bát phân	Thập lục phân	Thập phân	Nhị phân	Bát phân	Thập lục phân
0	0	0	0	13	1101	15	D
1	1	1	1	14	1110	16	E
2	10	2	2	15	1111	17	F
3	11	3	3	16	10000	20	10
4	100	4	4	17	10001	21	11
5	101	5	5	18	10010	22	12
6	110	6	6	19	10011	23	12
7	111	7	7	20	10100	24	14
8	1000	10	8	21	10101	25	15
9	1001	11	9	22	10110	26	16
10	1010	12	A	23	10111	17	17
11	1011	13	B	24	11000	30	18
12	1100	14	C	25	11001	31	19

Bảng 1.1

1.4 Các phép tính trong hệ nhị phân

Các phép tính trong hệ nhị phân được thực hiện tương tự như trong hệ thập phân, tuy nhiên cũng có một số điểm cần lưu ý

1.4.1 Phép cộng

Là phép tính làm cơ sở cho các phép tính khác.
Khi thực hiện phép cộng cần lưu ý:
0 + 0 = 0 ;
0 + 1 = 1 ;
1 + 1 = 0 nhớ 1 (đem qua bít cao hơn).

Ngoài ra nếu cộng nhiều số nhị phân cùng một lúc ta nên nhớ :
- Nếu số bit 1 chẵn, kết quả là 0;
- Nếu số bit 1 lẻ kết quả là 1
- Và cứ 1 cặp số 1 cho 1 số nhớ (bỏ qua số 1 dư, thí dụ với 5 số 1 ta kể là 2 cặp)

Thí dụ: Tính   011 + 101 + 011 + 011

```
     1 1   ← số nhớ
   1 1 1   ← số nhớ
     0 1 1
+    1 0 1
     0 1 1
     0 1 1
   --------
   1 1 1 0
```

1.4.2 Phép trừ

Cần lưu ý:
0 - 0 = 0 ;
1 - 1 = 0 ;
1 - 0 = 1 ;
0 - 1 = 1 nhớ 1 cho bit cao hơn
Thí dụ: Tính 1011 - 0101

```
   1       ← số nhớ
   1 0 1 1
 - 0 1 0 1
   ---------
   0 1 1 0
```

1.4.3 Phép nhân

Cần lưu ý:
0 x 0 = 0 ;
0 x 1 = 0 ;
1 x 1 = 1
Thí dụ: Tính 1101 x 101

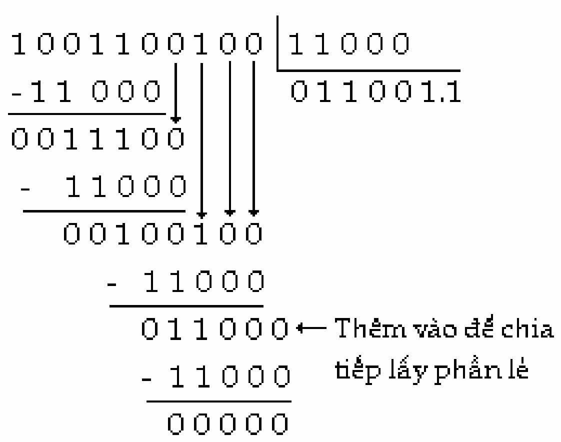
```
      1 1  0 1
 x       1 0 1
      ---------
       1 1 0 1
    0 0 0 0
  1 1 0 1
---------------
```

1 0 0 0 0 0 1

1.4.4 Phép chia

Thí dụ: 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, sau đó ta lấy 6 bit của số bị chia để chia tiếp (tương ứng với việc dịch phải số chia 1 bit trước khi thực hiện phép trừ)



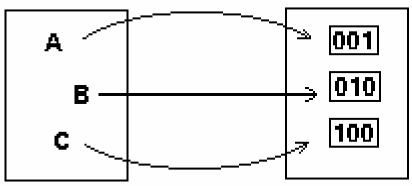
Kết quả : $(11001.1)_2 = (25.5)_{10}$

1.5 Mã hóa

1.5.1 Tổng quát

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

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



(H 1.1)

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

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

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

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

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

Trong kỹ thuật số ta thường dùng các mã sau đây:

1.5.2 Mã BCD (Binary Coded Decimal)

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

Thí dụ:

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

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

1.5.3 Mã Gray

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

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

$$0111 \rightarrow 0110 \rightarrow 0100 \rightarrow 0000 \rightarrow 1000$$

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

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

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

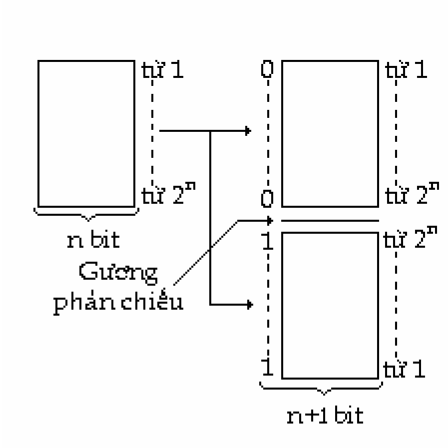
Người ta có thể thiết lập mã Gray bằng cách dựa vào tính đối xứng này:

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

- Viết ra 2^n từ mã theo thứ tự từ nhỏ đến lớn

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

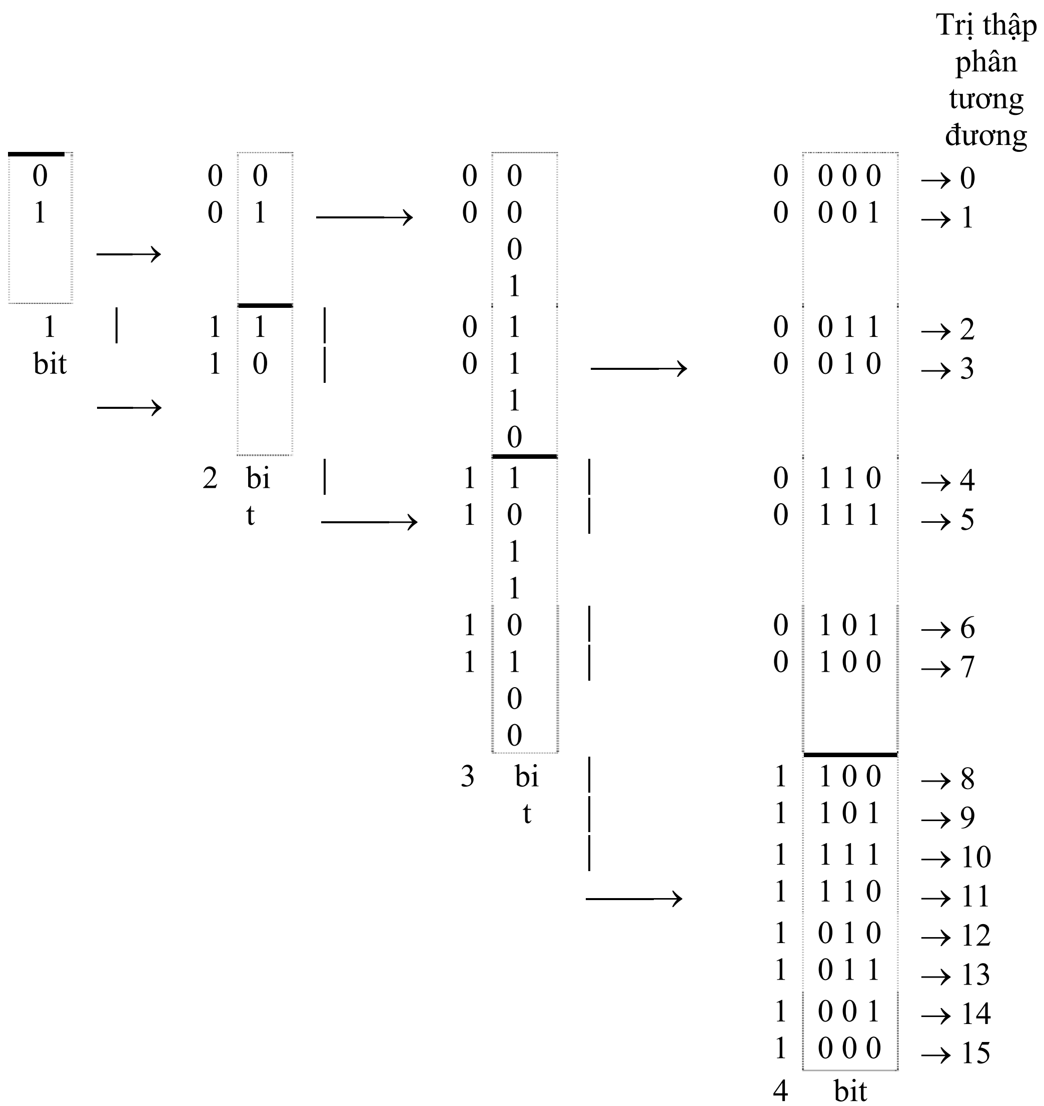
- Phần thứ hai của tập hợp gồm các từ mã giống như phần thứ nhất nhưng trình bày theo thứ tự ngược lại (giống như phản chiếu qua gương) và phía trước thêm vào số 1 thay vì số 0 (H 1.2).



(H 1.2)

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

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



(H 1.3)

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

Bài Tập

1. Đổi các số thập phân dưới đây sang hệ nhị phân và hệ thập lục phân :
a/ 12 b/ 24 c/ 192 d/ 2079 e/ 15492
f/ 0,25 g/ 0,375 h/ 0,376 i/ 17,150 j/ 192,1875

2. Đổi sang hệ thập phân và mã BCD các số nhị phân sau đây:
a/ 1011 b/ 10110 c/ 101,1 d/ 0,1101
e/ 0,001 f/ 110,01 g/ 1011011 h/ 10101101011

3. Đổi các số thập lục phân dưới đây sang hệ 10 và hệ 8:
a/ FF b/ 1A c/ 789 d/ 0,13 e/ ABCD,EF

4. Đổi các số nhị phân dưới đây sang hệ 8 và hệ 16:
a/ 111001001,001110001 b/ 10101110001,00011010101
c/ 1010101011001100,1010110010101 d/ 1111011100001,01010111001

5. Mã hóa số thập phân dưới đây dùng mã BCD :
a/ 12 b/ 192 c/ 2079 d/15436 e/ 0,375 f/ 17,250