Capítulo 2

Sistemas Numéricos e a Representação Interna dos Dados no Computador

2.0 Índice

2.1 Sistemas Numéricos

Sistemas numéricos são sistemas de notação usados para representar quantidades abstratas denominadas números. Um sistema numérico é definido pela base que utiliza. A base é o número de símbolos diferentes, ou algarismos, necessários para representar um número qualquer, dos infinitos possíveis no sistema. Por exemplo, o **sistema decimal**, utilizado hoje de forma universal, utiliza dez símbolos diferentes ou dígitos para representar um número e é, portanto, um sistema numérico na base 10.

Valores posicionais

Em um sistema de número posicional, um número é representado por uma seqüência de dígitos onde cada posição de dígito tem um peso associado. Tomando como exemplo o sistema decimal, ou base 10, que é sistema numérico que utilizamos diariamente (0, 1, 2, ... 9), o valor D de um número decimal de 4 dígitos $d_3d_2d_1d_0$ é $D = d_3*10^3 + d_2*10^2 + d_1*10^1 + d_0*10^0$. Cada dígito d_i tem um peso de 10^i. Por exemplo, o número 3.098.323 (base 10) é a representação de $3*10^6 + 0*10^5 + 9*10^4 + 8*10^3 + 3*10^2 + 2*10^1 + 3*10^0$.

2.1.1 Sistema Binário

O sistema binário, ou base 2, apresenta unicamente dois dígitos: 0,1. Neste sistema a contagem é realizada como segue: 0, 1, 10, 11, 100, 101, 110, 111, 1000, ...

Conversão Binário para Decimal

Sendo binário um sistema de número posicional, o valor B de um número binário de 8 dígitos $b_7b_6b_5b_4b_3b_2b_1b_0$ é $B = b_7*2^7 + b_6*2^6 + b_5*2^5 + d_4*2^4 + d_3*2^3 + d_2*2^2 + d_1*2^1 + d_0*2^0$. Cada dígito b_i tem um peso de 2^i. Assim o valor binário 10101010_b é calculado como segue $10101010_b = 0*1+1*2+0*4+1*8+0*16+1*32+0*64+1*128 = 170_d$. Esta é a conversão de um número binário para decimal. Outro exemplo $10011001_b = 1+8+16+128=153_d$

Conversão Decimal para Binário

No sistema decimal, por exemplo, o número 654 corresponde a 4 unidades, 5 dezenas e 6 centenas. Para verificar isto, divide-se o número pela sua base (que é 10):

654/10 =	65		Resto 4 (*1)
	/10 =	6	Resto 5 (*10)
		/10	Resto 6 (*100)

Para a conversão de decimal para binário utilizamos o mesmo processo. Por exemplo, para obtermos o correspondente binário do número 200_d, dividimos primeiramente este valor por 2 e anotamos o resto de cada divisão. Em seguida, dividimos novamente o dividendo da operação anterior por 2 e anotamos novamente o resto da divisão. Isto é repetido até que o resto da divisão seja 0, conforme abaixo:

200/2=100	Resto 0
100/2= 50	Resto 0
50/2 = 25	Resto 0
25/2 = 12	Resto 1
12/2 = 6	Resto 0
6/2 = 3	Resto 0
3/2 = 1	Resto 1
1/2 = 0	Resto 1

O correspondente binário de 200_d é obtido unindo-se os restos da divisão por 2 na ordem inversa, assim $200_d=11001000_b$.

2.1.2 Sistema Octal

O sistema binário ou base 8 apresenta oito dígitos: 0, 1, 2, 3, 4, 5, 6, 7. Neste sistema, a contagem é realizada como segue: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20,...

Conversão Octal para Decimal

Sendo o sistema octal um sistema de número posicional, o valor O de um número octal de 4 dígitos $o_3o_2o_1o_0$ é O = $d_3*8^3 + d_2*8^2+ d_1*8^1 + d_0*8^0$. Cada dígito o_i tem um peso de 8^i. Assim o valor octal 175_8 é calculado como segue $175_8 = 5*1+7*8+1*64 = 125_{10}$. Esta é a conversão de um número octal para decimal.

Conversão Decimal para Octal

Para a conversão de decimal para octal utilizamos o mesmo processo da conversão do sistema decimal para binário. Por exemplo, para obtermos o correspondente octal do número 200_d, dividimos primeiramente este valor por 8 e anotamos o resto de cada divisão. Em seguida, dividimos novamente o dividendo da operação anterior por 8 e anotamos novamente o resto da divisão. Isto é repetido até que o resto da divisão seja 0, conforme abaixo:

200/8= 25 Resto 0
25/8 = 3 Resto 1
3/8 = 0 Resto 3

O correspondente octal de 200_d é obtido unindo-se os restos da divisão por 8 na ordem inversa, assim $200_d=310_o$.

2.1.3 Sistema Hexadecimal

Na base hexadecimal tem-se 16 dígitos que vão de 0 à 9 e da letra A até F. Estas letras representam os números 10_d a 15_d. Assim nós contamos os dígitos hexadecimais da seguinte forma: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, ..., 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, ...

Conversão Binário para Hexadecimal

A conversão entre números binários e hexadecimais é simples. A primeira coisa a fazer é dividir o número binário em grupos de 4 bits, começando da direita para a esquerda, os lugares que faltam são complementados por zeros. Por exemplo, o número 101011_b ($1+2+8+32=43_d$), nós dividimos este em grupos de 4 bits e nós temos 10;1011. Nós completamos o último grupo com zeros: 0010;1011. Após nós tomamos cada grupo como um número independente e nós convertemos estes em dígitos decimais: 0010;1011=2;11. Mas desde que nós não podemos representar o número hexadecimal como 211 porque isto é um erro, nós temos que substituir todos os números decimais maiores que 9 pelas suas respectivas representações em hexadecimal, com o que nós obtemos: $2B_h$. A tabela abaixo pode auxiliar na conversão de números binário para hexadecimal.

Binário	Hexadecimal	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11

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

Afim de obter um número hexadecimal em binário é apenas necessário inverter os passos.

Conversão Hexadecimal em Decimal

Para converter um número hexadecimal em decimal, nós utilizamos a mesma fórmula utilizada na conversão binário para decimal, sendo que a base 2 é trocada por 16. Por exemplo, para converter B2A$_h$ em decimal:

$$
\begin{array}{lrcr}
B \rightarrow & 11*16^2 & = & 2816_d \\
2 \rightarrow & 2*16^1 & = & 32_d \\
A \rightarrow & 10*16^0 & = & \underline{10_d} \\
 & & & 2858_d
\end{array}
$$

Conversão Decimal para Hexadecimal

Para converter um número decimal em hexadecimal, nós utilizamos a mesma fórmula utilizada na conversão de um número decimal para binário, dividindo por 16 em vez de 2. Por exemplo, para converter 1069$_d$ em hexadecimal:

$$
\begin{array}{lcll}
1069/16 & = 66 & \text{Resto} & 13_d = D_h \\
66/16 & = 4 & \text{Resto} & 2_d = 2_h \\
\underline{4/16} & \underline{= 0} & \underline{\text{Resto}} & \underline{4_d = 4_h} \\
1069d & = & & 42D_h
\end{array}
$$

2.2 Operações Aritméticas

2.2.1 Aritmética Binária

Esta seção apresenta as quatro operações básicas no sistema binário: adição, subtração, divisão e multiplicação.

Adição

Para somar dois números binários, fazem-se as contas coluna a coluna, da direita para a esquerda, como de costume, fazendo o transporte de um **(<e vai um>)** quando for o caso. Para isto, observe as seguintes operações básicas:

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 1 = 10 (1 mais 1 é igual a 0 e vai 1)
- 1 + 1 + 1 = 11 (1 mais 1 mais 1 é igual a 1 e vai 1)

Exemplos:

+ 1 1		*1 11*	*111*
101	100100	11001	101110
+ 1101	+ 10010	+ 10011	+ 1110
10010	110110	101100	111100

Subtração

Existem duas formas para fazer a subtração binária:

- Como o conjunto de símbolos contém apenas 2 dígitos, ao se efetuar a subtração parcial entre 2 dígitos, um do diminuendo e outro do diminuidor, se o segundo (diminuidor) exceder o primeiro (diminuendo), subtrai-se uma unidade ao dígito imediatamente à esquerda no diminuendo (se existir e o seu valor for 1), convertendo-o a 0. Em seguida, substituímos o diminuendo por 2, que corresponde à equivalência 1*2, da unidade extraída. Se o dígito imediatamente à esquerda for 0, procura-se nos dígitos consecutivos.

Exemplos: 11101 - 111

		02
	02	*02*
11101	11~~1~~01	1~~11~~01
-111	-111	-111
0	10	10110

Exemplos: 11000 - 111

0112	*0112*
01~~2~~0	*01~~2~~0*
0~~2~~00	*0~~2~~00*
1~~1~~000	1~~1~~000
-111	-111
	10001

- A segunda forma de realizar a subtração, por exemplo de a-b, e realizar a soma de a por -b. Esta subtração é feita pelo chamado método do complemento de dois. O complemento de dois transforma um número positivo em negativo. Neste método, o diminuendo (a) é somado com o complemento de dois do diminuidor (-b). Note que o número de dígito dos operandos devem ser o mesmo: para isto complemente o operando com menor número de dígitos com zeros a esquerda (antes do complemento). Para realizar o complemento de dois, basta trocar os uns pelos zeros e vice-versa e adicionar um ao resultado. Por exemplo, a subtração de 1110-101 é feita da seguinte maneira:

1 . Completa-se o número de dígitos do diminuidor: 0101
2. Realiza-se o complemento de dois do diminuidor: 1010+1=1011.
3. Soma-se os dois operandos 1110+1011=11001
4. Despreza-se o transporte final: 1001

Multiplicação

A multiplicação na base 2 - ou em qualquer outra base - pode fazer-se por adições sucessivas; para calcular A*B basta somar A a si própria B vezes.

Exemplo: 101_b * 100_b = ? Lembrado que 100_b = 4_b, então

101 * 100 =

```
      1
    1 1
    1 0 1 ┐
  1 1 0 1 │ 100 vezes
  1 1 0 1 │
  + 1 0 1 ┘
1 0 1 0 0
```

Uma forma, e a ideal, é fazer a operação semelhante à multiplicação decimal, exceto pelo fato da soma final dos produtos se fazer em binário. Para tal, as seguintes igualdades devem ser respeitadas:

- 0*0=0; 0*1=0; 1*0=0; 1*1=1

Exemplos:

- Multiplicar os números 1011 e 1101.

```
        1 0 1 1 ———— 1 1
      * 1 1 0 1 ———— 1 3
        1 0 1 1
      0 0 0 0
    1 0 1 1
  + 1 0 1 1
1 0 0 0 1 1 1 1 ———— 1 4 3
```

- Multiplicar os números 1001 e 1101.

```
        1 0 0 1 ———————    9
      * 1 1 0 1 ——————— 1 3
      ---------
        1 0 0 1
      0 0 0 0
    1 0 0 1
+ 1 0 0 1
---------------
  1 1 1 0 1 0 1 ——————— 1 1 7
```

Divisão

Analogamente, a divisão pode ser feita por subtrações sucessivas, até obtermos uma diferença igual a zero (no caso de uma divisão exata), ou um número menor que o divisor.

Exemplo:

$$\frac{10000}{1000} = ?$$

```
  1 0 0 0 0
-   1 0 0 0
-----------
  0 1 0 0 0 ——— 1 vez
-   1 0 0 0
-----------
  0 0 0 0 0 ——— 2 vezes
```

O resultado é $2_{(10)}$, isto é, $10_{(2)}$

Mas esta divisão pode ser feita de maneira idêntica à divisão decimal, exceto pelo fato das multiplicações e subtrações internas ao processo serem feitas em binário.

Exemplo:

- Dividir 11011 e 101.

```
1 1 0 1 1 = 2 7
    1 0 1 = 5

    1 1 0 1 1   |1 0 1
  - 1 0 1        -----
  ---------      1 0 1
    0 0 1 1 1
    -   1 0 1                o quociente é   1 0 1
    ---------
        0 1 0                e o resto é     1 0
```

- Dividir 1010101 e 101.

```
1 0 1 0 1 0 1 = 8 5
        1 0 1 = 5

    1 0 1 0 1 0 1   |1 0 1
  - 1 0 1            ---------
  -----------        1 0 0 0 1
    0 0 0 0 1 0 1
        -   1 0 1
        ---------
            0 0 0
```

A prova é:

```
      1 0 0 0 1    quociente
        * 1 0 1    divisor
    -----------
      1 0 0 0 1
    0 0 0 0 0
+ 1 0 0 0 1
---------------
  1 0 1 0 1 0 1    dividendo
```

2.2.2 Aritmética Hexadecimal

Adição

Como exemplo, suponha a adição de 8_h+5_h, se somada em decimal o valor seria 13. Em hexadecimal, o valor 13 é representado por D_h. Deve-se reparar que, tal como nos habituamos a fazer na Escola Primária, sempre que o resultado iguala ou ultrapassa a base, subtraímos esta ao resultado, e fazemos um transporte para a coluna seguinte («e vai um», neste caso). Suponha agora a adição de 19 por 9:

- Em decimal, o resultado seria 28;

- Em hexadecimal, inicialmente somamos os dígitos menos significativos: 9_h+9_h = 18; como o resultado é maior que a base (16), então 18-16 = 2 e vai um para o dígito mais significativo. Portanto, 19_h+9h=22_h;

  Não é preciso converter os números $F8_h$ e $A34_h$ para decimal, somá-los e reconverter o resultado para a base 16. Podemos fazer a conta coluna a coluna. Então $F8_h$ + $A34_h$ é calculado da seguinte forma:

$$\begin{array}{ccccc} & & F & 8 & H \\ + & A & 3 & 4 & H \\ \hline & & & & \end{array} \longrightarrow \begin{array}{ccccc} & & F & 8 & H \\ + & A & 3 & 4 & H \\ \hline & & & C & H \end{array}$$

$$\Big\downarrow$$

$$\begin{array}{ccccc} & & F & 8 & H \\ + & A & 3 & 4 & H \\ \hline & & 2 & C & H \end{array} \quad \text{e vai um}$$

$$\Big\downarrow$$

$$\begin{array}{ccccc} & & F & 8 & H \\ + & A & 3 & 4 & H \\ \hline & B & 2 & C & H \end{array} \quad \text{porque A + 1 = B}$$

Subtração

Vamos ver a subtração a partir de um exemplo: 27H-1EH. Efetuamos a operação de subtração coluna a coluna. Na primeira coluna, o diminuidor (E) é superior ao diminuendo (7). Então, adicionamos a base ao diminuendo, executamos a subtração, e há transporte de uma unidade que somamos ao diminuidor da coluna seguinte.

$$\begin{array}{cccc} & 2 & 7 & H \\ - & 1 & E & H \\ \hline & & 9 & H \end{array} \quad \text{«e vai um»}$$

retirando o número transportado do diminuendo da coluna da esquerda, 2-1, obtemos 1, e subtraindo 1 do diminuidor, obtemos 0:

$$\begin{array}{cccc} & 2 & 7 & H \\ - & 1 & E & H \\ \hline & 0 & 9 & H \end{array}$$

Multiplicação

Esta operação pode fazer-se facilmente por meio da tabela de dupla entrada apresentada:

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
2	02	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
3	03	06	09	0C	0F	12	15	18	1B	1E	21	24	27	2A	2D
4	04	08	0C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	05	0A	0F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	06	0C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	07	0E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	08	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	09	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0E	1C	2A	38	4C	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Como se vê, temos todos os algarismos hexadecimais (exceto o zero) nas entradas verticais e horizontais da tabela. Se quiséssemos calcular 5_h*9_h, por exemplo, encontraríamos o resultado na intercessão da coluna 5 com a linha 9. Então, 5_h*9_h = $2D_h$. Uma vez que a multiplicação é comutativa, então, o mesmo resultado se verifica na intercessão da coluna 9 com a linha 5.

- 1º. Exemplo:

  A_h* 2_h = _______ (complete)

  2_h*7_h = _______ (complete)

- 2º. Exemplo:

```
  6 7 A H
*   3 2 H
---------
```

Procedendo como de costume, vamos começar pelo produto do multiplicando pelo algarismo mais à direita do multiplicador:

```
  6 7 A H
*   3 2 H
---------
```

```
  6 7 A H        Na tabela, vemos que A * 2 = 14; então,
*   3 2 H        escrevemos «4» e há transporte de 1 unidade
---------        (e vai um)
      4 H
```

```
  6 7 A H        2H * 7H = 0EH; 0EH + 1 (de transporte)
*   3 2 H        = 0FH; escreve-se «F», e não há transporte.
---------
    F 4 H
```

```
  6 7 A H        2H * 6H = 0CH; escreve-se «C», e não há
*   3 2 H        transporte.
---------
  C F 4 H
```

Calculamos em seguida o produto do multiplicando pelo 2º. algarismo (contando a partir da direita) do multiplicador.

```
      6 7 A H
    *   3 2 H
    ---------
      C F 4
1 3 6 E
-------------
```

Agora é só somar os produtos parciais. Fica:

```
      6 7 A H
    *   3 2 H
    ---------
      C F 4
1 3 6 E
-------------
1 4 3 D 4 H
```

- Fazer o produto dos seguintes números hexadecimais: B12H e 3FCH

```
        B 1 2 H
    *   3 F C H
      ---------
      8 4 D 8
    A 6 0 E
  2 1 3 6
---------------
  2 C 1 B B 8 H
```

Divisão

Esta é a operação mais difícil de fazer sem recorrermos à tabela anterior. Veja alguns exemplos:

- 1º. Exemplo: dividir os números hexadecimais 2F por 12.

```
2 F   |1 2
  B    2        2 * 2 = 4; F - 4 = B

2 F   |1 2
0 B    2        2 * 1 = 2; 2 - 2 = 0
```

Para verificarmos se é verdade, nada melhor que fazer a prova:

```
  1 2 H     divisor
*   2 H     quociente
-------
  2 4 H
+   B H     resto
-------
  2 F H     dividendo
```

- 2º. Exemplo: Dividir os números hexadecimais 3F4 por A1

3 F 4	1 A	
B	2	2 * A = 4 (e vai um); F - 4 = B
3 F 4	1 2	
0 B	2	2 * 1 = 2 + 1 do transporte = 3; 3 - 3 = 0
3 F 4	1 A	
0 B 4	2 6	6 * A = C (e vão três); 4 - C = 8 (e vão 4)
1 8		6 * 1 = 6 + 4 do transporte = 10; B - A =1

2.3 Operações Lógicas

Existem quatro tipos de operações lógicas que se podem operar sobre números binários: AND, OR, XOR (ou exclusivo), e NOT.

2.3.1 Operações lógicas com bits

AND

A operação lógica AND é uma operação que aceita dois operandos. Estes operando são binários simples (base 2). A operação AND é

0 and 0 = 0
0 and 1 = 0
1 and 0 = 0
1 and 1 = 1
Uma maneira compacta de representar a operação lógica AND é com a tabela verdade, apresentada abaixo. As duas colunas a esquerda representam os dois operandos da operação *AND Op1 OP2*.

Op1	Op2	AND Op1 Op2
0	0	0
0	1	0
1	1	1

Em português, a operação lógica AND é: "se o primeiro operando é 1 e o segundo operando é 1, o resultado é 1, senão o resultado é 0".

OR

A operação lógica OR também é uma operação com dois operandos. Ela é definida como:

0 or 0 = 0
0 or 1 = 1
1 or 0 = 1
1 or 1 = 1

A tabela verdade da operação OR tem a seguinte forma:

Op1	Op2	OR Op1 Op2
0	0	0
0	1	1
1	0	1
1	1	1

A operação lógica OR significaria: "Se o primeiro operando ou o segundo operando (ou os dois) forem 1, o resultado é 1, senão o resultado é 0. Esta operação também é conhecida como ou inclusivo *(inclusive-OR)*.

XOR

A operação lógica XOR (ou exclusivo) também é uma operação com dois operandos. Ela é definida como:

0 xor 0 = 0
0 xor 1 = 1
1 xor 0 = 1
1 xor 1 = 0

A tabela verdade da operação XOR tem a seguinte forma:

Op1	Op2	XOR Op1 Op2
0	0	0
0	1	1
1	0	1
1	1	0

Em português a operação lógica XOR significaria: “Se o primeiro operando ou o segundo operando, mas não os dois, for 1, o resultado é 1, senão o resultado é 0.

NOT

A operação lógica XOR (ou exclusivo) também é uma operação com um operando. Ela é definida como:

not 0 = 1
not 1 = 1

A tabela verdade da operação NOT tem a seguinte forma:

Op1	NOT Op1
0	1
0	0

Em português a operação lógica NOT significaria: “Se o operando for 1, o resultado é 0, senão o resultado é 1”.