1 INTRODUÇÃO

Existem inúmeras bases de numeração. No dia a dia, faz-se uso do sistema de numeração decimal, isto é, de base 10, pela comodidade e pelo fato de que a contagem dos dedos da mão é igual a 10, facilitando a contagem. Além da base 10 existem várias outras, sendo as mais utilizadas as de base 2 (sistema de numeração binário), base 8 (sistema de numeração octal), base 16 (sistema de numeração hexadecimal e a base 60 (sistema de numeração sexagesimal). As bases 2, 8 e 16 são muito utilizadas na computação, enquanto a base 60 é utilizada na geometria para a medição de ângulos (medição em graus) e no cotidiano para medir o tempo (horas, minutos e segundos).

2 CONVERSÃO DE SISTEMAS DE NUMERAÇÃO

Consiste basicamente em escrever um determinado número n em uma base b em uma base b_k. Há três formas de fazer a conversão entre bases: divisão, posição (ou por polinômio) e por agrupamento de bits.

2.1 DIVISÃO INTEIRA

É útil para converter o sistema decimal (base 10) para qualquer outro sistema de numeração. Nele, basta dividir o número dado em decimal pela base que se queira converter. Todos os quocientes obtidos devem ser divididos até que seu valor seja menor que o valor da base. O resultado será o valor do último quociente justaposto aos restos, da direita para a esquerda.

Ex1: $1.537_{(10)}$ para base 2

O sistema de numeração binário faz uso dos seguintes elementos: {0, 1}.

Valor decimal/ Dividendo	Base/Divisor	Quociente	Resto
1537	2	768	1
768	2	384	0
384	2	192	0
192	2	96	0
96	2	48	0
48	2	24	0
24	2	12	0
12	2	6	0
6	2	3	0
3	2	1	1

$1.537_{(10)}$ = último quociente + sucessão de restos, de baixo para cima ou da direita para a esquerda. Logo:

$1.537_{(10)} = 11000000001_{(2)}$

No caso específico da conversão de decimal pata binário, pode-se também fazer uso da subtração. Nesse caso, lista-se todas as potências de 2 até que a potência seja a mais próxima do valor em decimal. Em seguida, subtrai-se o valor informado pela maior potência listada. Marca-se um para essa potência. Repete-se o processo para cada um dos restos obtidos até que seja zero. Veja:

Potências	2048	1024	512	256	128	64	32	16	8	4	21	1
Potenciações	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
Binário		1	1	0	0	0	0	0	0	0	0	1
Subtrações		1537	513									1
	-	1024	512									1
		513	1									0

Logo: $1.537_{(10)} = 11000000001_{(2)}$

Ex 2: $1.537_{(10)}$ para base 6 (hexal)

O sistema de numeração hexal faz uso dos seguintes elementos: {0, 1, 2, 3, 4, 5}.

Valor decimal/ Dividendo	Base/Divisor	Quociente	Resto
1537	6	256	1
256	6	42	4
42	6	7	0
7	6	1	1

$1.537_{(10)} = 11041_{(6)}$

Ex 3: $1.537_{(10)}$ para base 8 (octal)

O sistema de numeração hexal faz uso dos seguintes elementos: {0, 1, 2, 3, 4, 5, 6, 7}.

Valor decimal/ Dividendo	Base/Divisor	Quociente	Resto
1537	8	192	1
192	8	24	0
24	8	3	0

$1.537_{(10)} = 3001_{(8)}$

Ex 4: $1.537_{(10)}$ para base 16 (hexadecimal)

O sistema de numeração hexal faz uso dos seguintes elementos: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}. Ou seja, para 10, 11, 12, 13, 14 e 15, usa-se as letras A, B, C, D, E e F respectivamente.

Valor decimal/ Dividendo	Base/Divisor	Quociente	Resto
1537	16	96	1
96	16	6	0

$1.537_{(10)} = 601_{(16)}$

Ex 5: $243.895_{(10)}$ para base 16 (hexadecimal)

Valor decimal/ Dividendo	Base/Divisor	Quociente	Resto	Representação
243895	16	15243	7	7
15243	16	952	11	B
952	16	59	8	8
59	16	3	11	B

$243.895_{(10)} = 3B8B7_{(16)}$

2.2 POLINOMIAL

Esse sistema pode ser usado para a conversão de qualquer base. Para isso usa-se o seguinte polinômio como lei de formação:

$$NCD = a_n b^n + a_{n-1} b^{n-1} + \ldots + a_{n-k} b^{n-k} + \ldots + a_0 b^0$$

Onde NCD é número convertido para decimal, a_n é um algarismo, b_n é a base e n é a quantidade de algarismos subtraído em uma unidade (1).

Ex 1: $1111101_{(2)}$ para base 10 (decimal)

$$NCD = 1\times2^6 + 1\times2^5 + 1\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$$

NCD = 64 + 32 + 16 + 8 + 4 + 0 + 1
NCD = 125

$1111101_{(2)} = 125_{(10)}$

Ex 2: $237_{(8)}$ para a base 5

$$NCD = 2\times8^2 + 3\times8^1 + 7\times8^0$$

NCD = 128 + 24 + 7
NCD = 159

Para converter para a base 5, usa-se o processo das divisões:

Valor decimal/ Dividendo	Base/Divisor	Quociente	Resto
159	5	31	4
31	5	6	1
6	5	1	1

$237_{(8)} = 1114_{(5)}$

2.3 AGRUPAMENTO DE DÍGITOS ou REAGRUPAMENTO

Esse sistema é útil para converter diretamente números em diferentes bases de numeração, se as bases potências entre si. Com efeito, é muito utilizada para conversão entre binários, octais e hexadecimais, pois:

$8 = 2^3$
$16 = 2^4$

Com base nisso, monta-se uma tabela com os valores convertido até o maior dígito da maior base que, no caso é 15 da base 16:

Base	Dígitos															
Decimal (referência)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binário	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Se as bases forem potências uma das outras, a conversão pode ser feita diretamente. Nesse caso em particular, se pode converter diretamente de binário para octal e de binário para hexadecimal e vice-versa. Contudo, não se pode converter diretamente de octal para hexadecimal ou, de hexadecimal para octal pois as bases não são potências entre si. Com efeito, nessa situação, deve-se fazer uma conversão intermediária para uma base que permita a conversão direta. Como a base 2 permite a conversão direta para octal e para hexadecimal, deve-se necessariamente converter para binário, para só então realizar a conversão final.

No processo de conversão por reagrupamento, o número dado deve ser agrupado em blocos, formados da direita para a esquerda. A quantidade de dígitos do bloco se relaciona com a quantidade de dígitos da menor base para representar o último algarismo da maior base. Por exemplo, em uma conversão de binário para octal, os blocos terão 3 dígitos, já que para representar 7, são necessários três dígitos binários, como pode ser visto na tabela. Na conversão de binário para hexadecimal, são necessários 4 dígitos, já que 15 é representado por 4 dígitos em binário.

Ex1: $10101110100_{(2)}$ para octal

Como para representar um octal são necessários 3 algarismos binários. Então, cada bloco será formado por 3 algarismos:

10.101.110.100 = 2.5.6.4

Reagrupando: 2564. Portanto, $10101110100_{(2)} = 2564_{(8)}$

Ex2: $1643_{(8)}$ para hexadecimal

Como as bases não potências entre si, então deve-se primeiro converter para binário:

1.6.4.3 = 001.110.100.011

Reagrupando: $1110100011_{(2)} = 1643_{(8)}$

Por fim, converte-se o binário para hexadecimal:

11.1010.0011 = 3. A .3
Logo: $1643_{(8)}$ = 3A3.

3 OPERAÇÕES MATEMÁTICAS NA BASE 2

As operações aritméticas básicas são: adição, subtração, multiplicação e divisão. Em termos gerais, as operações aritméticas na base 2 são conduzidas da mesma forma que no sistema decimal, guardadas as devidas peculiaridades.

3.1 ADIÇÃO

Os resultados são os mesmos esperados na adição na base 10. No entanto, enquanto a base 10 transporta o valor que exceder a 9, no sistema binário, o transporte ocorre ao exceder 1. Logo:

0 + 0 = 0
0 + 1 = 1
1 + 1 = 0 com transporte de 1 à esquerda

Ex1: $1110100011_{(2)} + 111_{(2)}$

							1	1	1	
	1	1	1	0	1	0	0	0	1	1
+								1	1	1
	1	1	1	0	1	0	1	0	1	0

Note que na segunda coluna (da direita para a esquerda) 1 + 1 = 0 e transporta 1. Mas como recebeu um transporte da primeira coluna: 0 + 1 = 1. De fato, 931 + 7 = 938.

3.1.1 COMPLEMENTO DE UM

O complemento de 1 de um número binário é o valor que um número binário deve ter que, ao ser somado com outro, resulta em terceiro número binário em que todos os algarismos são uns (1).

Por exemplo, o complemento de $111_{(2)}$ é $000_{(2)}$, pois $111_{(2)} + 000_{(2)} = 111_{(2)}$. O complemento de $1110100011_{(2)}$ é $0001011100_{(2)}$, pois $1110100011_{(2)} + 0001011100_{(2)} = 1111111111_{(2)}$.

É importante que se complete o número menor (subtraendo) a fim de que ele tenha o mesmo número de dígitos do minuendo. Isso é importante, porque o complemento é definido também em número de dígitos. Logo, por exemplo, $000_{(2)}$ é complemento de $111_{(2)}$ com três dígitos.

3.1.2 COMPLEMENTO DE DOIS

O complemento de 2 de um número binário é o complemento de um desse número acrescido de uma unidade. Por exemplo:

$C_1(111) = 000_{(2)}$

$C_2(111) = 001_{(2)}$

O complemento de dois é utilizado para representar números negativos.

3.2 SUBTRAÇÃO

A subtração de binários tem uma diferença em relação à subtração de decimais. A operação 0 – 1 em binário tem como diferença o valor 1 e ainda tem um transporte, pelo que na subtração de decimais, no conjunto dos números naturais, essa operação não existe. Por outro lado, no conjunto dos números inteiros, traria como resultado -1. Ressalva feita a esse fato, todas as outras operações seguem a mesma ideia da subtração em decimal:

0 – 0 = 0

0 – 1 = 1 com transporte de 1 à esquerda

1 – 0 = 1

1 – 1 = 0

Ex 1: $10110_{(2)} - 1011_{(2)}$

	1	0	1	1	0
-	1		1	1	
		1	0	1	1
	0	1	0	1	1

Outra particularidade em relação a operação de subtração em relação ao sistema decimal é que os transportes não são somados ao dígito que recebeu o transporte, mas sim subtraídos. Logo, há duas subtrações. A primeira entre o minuendo (valor do qual se subtrai) e o transporte; a segunda subtração ocorre entre o resultado da primeira subtração com o subtraendo (valor que subtrai). Repare que o transporte ocorre para o próximo dígito à esquerda e no subtraendo.

Ex 2: $1110000001_{(2)}$ - $1000000110_{(2)}$

	1	1	1	0	0	0	0	0	0	1
-			1	1	1	1	1	1		
	1	0	0	0	0	0	0	1	1	0
	0	1	0	1	1	1	1	0	1	1

De fato, 897 – 518 = 379.

Como o complemento de 2 representa números negativos, podemos usá-lo para simplificar as subtrações, ou seja, transformá-las em adições. A adição de um número com o complemento de 2 de outro sempre gera um dígito excedente, pelo que esse dígito deve ser excluído.

Ex 1: $10110_{(2)}$ – $1011_{(2)}$

1) completa-se o subtraendo: 01011
2) calcula-se o complemento de 1 do subtraendo: 10100
3) calcula-se o complemento de 2 do subtraendo: 10100 + 1 = 10101
4) realiza-se a subtração:

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

5) exclui-se o dígito excedente (o dígito mais à esquerda): 01011

De fato, 22 – 11 = 11. É justamente essa operação que o computador faz as subtrações.

3.3 MULTIPLICAÇÃO

A multiplicação em binário é exatamente igual à multiplicação em binário.

$0 \times 0 = 0$

$0 \times 1 = 0$

$1 \times 1 = 1$

$1 \times 0 = 0$

Ex 1: 11001×11

		1	1	0	0	1
	x				1	1
	1	1	1	0	0	1
+	1	1	0	0	1	0
1	0	0	1	0	1	1

De fato, $25 \times 3 = 75$

3.4 DIVISÃO

A divisão de números binários tem a mesma mecânica da divisão de números decimais, ou seja, escolhe-se uma parte (ou a totalidade) do dividendo que seja maior que o divisor. Acha-se o maior produto próximo (ou mesmo o produto exato) ao valor do dividendo. Subtrai-se a parte do quociente (ou todo ele caso o produto seja exato). Abaixa-se os próximos bits e assim sucessivamente. A parte que merece atenção é que deve-se descobrir o valor decimal das partes do dividendo para garantir que ele seja maior que o divisor.

Ex1: $1001011_{(2)} \div 11_{(2)}$

	1	0	0	1	0	1	1	1	1			
–	1	1						1	1	0	0	1
		1	1									
	0	0	1	1								
		–	1	1								
			0	0	0	1	1					
					–	1	1					
						0	0					

Escolheu-se $100_{(2)}$ que em decimal vale 4 e é menor que $11_{(2)}$ que em decimal vale 3. $1_{(2)}$ x $11_{(2)} = 11_{(2)}$ que resulta em 1 (4 – 3). Abaixa-se o próximo bit formando $11_{(2)}$ que também dá por 1. Abaixa-se $0_{(2)}$, mas como é menor que o divisor, indica-se no quociente que essa casa é nula. Abaixa-se o próximo bit que é $1_{(2)}$. Mesmo assim o valor formado é menor que o divisor, então marca-se no quociente que também essa casa é nula. Por fim abaixa-se o próximo (e último) bit. Forma-se então $11_{(2)}$ que produto exato do divisor por $1_{(2)}$. Logo:

$$1001011_{(2)} \div 11_{(2)} = 11001_{(2)}$$

De fato, 75/3 = 25.