

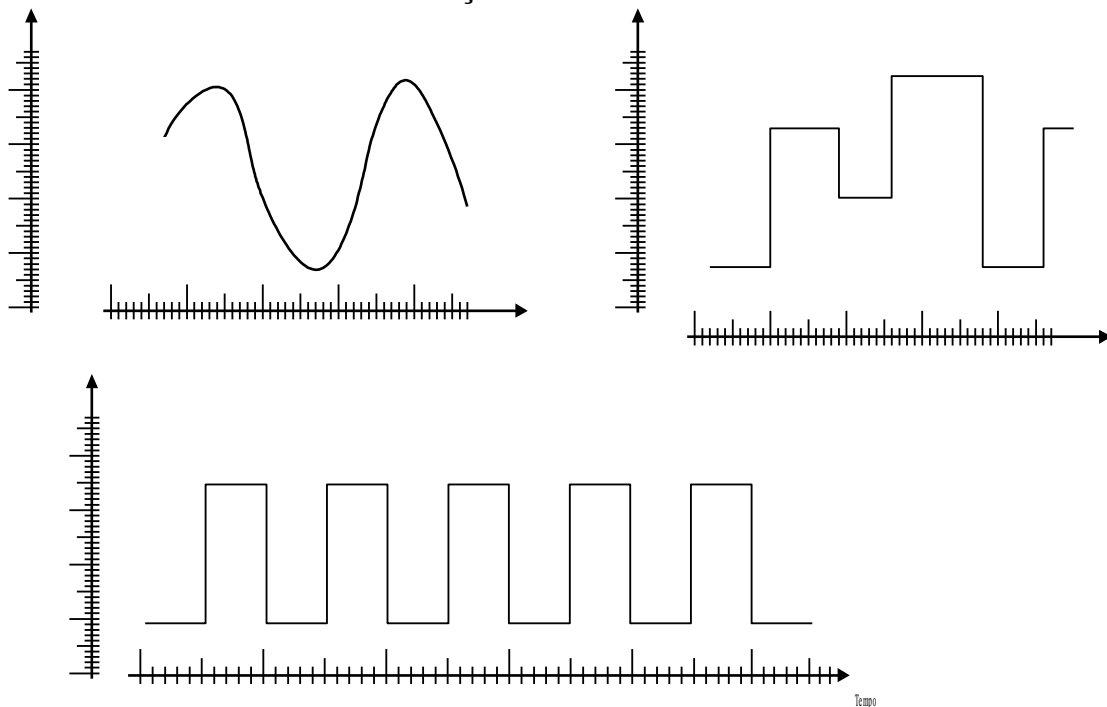
INTRODUÇÃO

Eletrônica digital é a parte da eletrônica que trabalha com sinais discretos. Em contrapartida existe a eletrônica analógica que trabalha com sinais analógicos ou também contínuos.

Ambas as eletrônicas, digital e analógica usam os mesmos componentes ou sejam resistores, diodos, transistores, fios condutores, etc. Na realidade, a eletrônica digital reúne casos particulares de circuitos simplificados da eletrônica analógica.

Sinais Analógicos: São sinais contínuos no tempo. No sinal analógico a passagem de uma condição para outra se dá de forma suave, sem descontinuidade. O mundo físico real é essencialmente analógico, onde os sinais, que representam informações, aparecem de modo contínuo.

Sinais Digitais: São sinais discretos no tempo, de tal forma que sempre existe uma descontinuidade entre uma condição e outra.



Na figura observe que o sinal contínuo pode assumir qualquer valor. O sinal discreto pode assumir somente alguns valores possíveis e um exemplo de sinal discreto digital assumindo somente dois valores, que é o sinal utilizado na maioria dos sistemas digitais.

Um bom exemplo de sinais contínuos e sinais discretos é o mostrador de um termômetro analógico, aquele que usa uma coluna de mercúrio e um termômetro de mostrador digital. Note bem que o mostrador pode ser analógico ou digital mais a grandeza física que eles representam sempre será analógica. No termômetro analógico, por exemplo, você pode fazer infinitas leituras entre as marcações 22 graus e 23 graus, mas um termômetro de marcador digital ou ele indica 22 graus ou 23 graus, ele não mostra valores intermediários.

Se quisermos um termômetro digital com melhor resolução nos números mostrados,

teremos que comprar um mais caro que outro que marque décimos de grau. Assim podemos ler: 22,0 22,1; 22,2... Mas não poderemos ler entre, por exemplo, 22,1 e 22,2. No termômetro com mostrador por coluna de mercúrio se eu colocar uma lupa no mostrador eu poderei ler valores entre décimos ou até milésimos.

Como já mencionado, o sinal discreto digital poderá assumir somente valores discretos que são expressos por afirmações declarativas (cada valor está associado a um significado) e são mutuamente exclusivos.

As vantagens dos sinais digitais são as seguintes:

- Mais Fáceis de Projetar – pois possuem variáveis controláveis;
- Facilidade de Armazenamento da Informação Digital – difícil e limitado realizar isso com sistemas analógicos;
- Menos Suscetível a Ruídos pois pequenas diferenças na amplitude não afetam seu significado;
- Maior Integração dos Circuitos Integrados (CIs);
- E muitos outros...

Como no mundo físico real os sinais são analógicos e na Eletrônica digital queremos lidar com sinais discretos, terá que existir uma transformação de sinais analógicos para sinais digitais. Estas transformações e o seu tratamento no que se refere a representação mais simples de números, letras e símbolos para dentro de sistemas computacionais serão vistas ao longo deste curso.

1.SISTEMAS DE NUMERAÇÃO

1.1Introdução

Raras são as pessoas que se interessam por História da Ciência, em geral, e História da Matemática, em particular. É uma pena, pois a história mostra quão difícil foi chegarmos a este estágio de nossa civilização. Mesmo o Teorema de Pitágoras que, segundo os gregos, data de cerca de 500 anos antes de Cristo, ou seja, aproximadamente 2500 anos atrás, já teve esta datação questionada. Segundo o livro de Gillings, A Matemática na Era dos Faraós, foi encontrado um pergaminho que, após ser decifrado, fez os historiadores da ciência acreditarem que este teorema já era conhecido há cerca de, pelo menos, 1000 anos, antes, isto é, há cerca de 3500 anos. O mesmo acontece com a idéia de **números**.

Acredita-se que a necessidade de criação de números veio com a necessidade de contar. Seja o número de animais, alimentos, ou coisas do tipo. Como a evolução nos legou algumas características, como os cinco dedos em cada mão (fingers) e cinco dedos em cada pé (toes), seria muito natural que os primeiros sistemas de numeração fizessem uso das bases 10 (decimal) e 20 (vigesimal).

O sistema de numeração normalmente utilizado, o sistema decimal, apresenta dez dígitos (algarismos), são eles: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. No sistema decimal, 10 é a base do sistema e seu dígito máximo é 9.

Descrição geral de um número em qualquer base:

$$\dots + X \cdot \text{Base}^Y + \dots$$

posição do dígito, em relação a vírgula

base do sistema de numeração

dígito do número em questão

Nesta generalização, Y vale 0 para o primeiro dígito a direita da vírgula, 1 para o segundo dígito e assim por diante. Para esquerda da vírgula Y passa a valer -1 para o primeiro dígito, -2 para o segundo e assim por diante.

Observe que para um sistema de base N, os dígitos vão de 0 à N-1. Quando se atinge a contagem N, um novo dígito é acrescentado à direita do número (parte inteira)

$$\begin{aligned} \text{Ex.: } 328451,52_{10} &= 3 \times 10^5 + 2 \times 10^4 + 8 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 1 \times 10^0 + 5 \times 10^{-1} \\ &\quad + 2 \times 10^{-2} \\ &= 300000 + 20000 + 8000 + 400 + 50 + 1 + 0,5 + 0,02 \\ &= 328451,52_{10} \end{aligned}$$

Para nosso estudo, sempre utilizaremos o número representado por sua base como neste exemplo. Em regra geral, quando não representamos a base subscrita logo a direita do número, significa que o mesmo é de base 10.

1.2 Sistema de Numeração Binário

Este sistema de numeração, como o próprio nome sugere, apresenta base 2. Os números 0 e 1 são os dígitos deste sistema.

O sistema binário é de grande importância, pois apresenta correspondência direta com os estados de um sistema digital. Por exemplo: para o dígito 0 pode-se atribuir o valor de tensão 0 V e para o dígito 1 pode-se atribuir o valor de tensão de 5 V.

$$\begin{aligned} \text{Ex.: } 1001101_2 &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 64 + 0 + 0 + 8 + 4 + 0 + 1 \\ &= 77_{10} \end{aligned}$$

1.2.1 Conversão de um número no sistema binário para o equivalente no sistema decimal.

Regra geral: multiplica-se cada dígito pelo valor da base elevada a uma dada potência, definida pela posição do dígito, e finalmente realiza-se a soma.

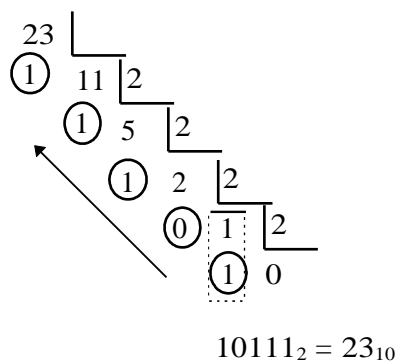
$$\begin{aligned} \text{Ex.: } 11001101_2 &= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 128 + 64 + 0 + 0 + 8 + 4 + 0 + 1 \\ &= 205_{10} \end{aligned}$$

1.2.2 Conversão de decimal para binário.

Ex.: Conversão do número 23_{10} para binário.

$$\begin{array}{l}
 23 \begin{array}{|l} 2 \\ \hline 1 \quad 11 \end{array} \longrightarrow 23 = 2 \times 11 + 1 \\
 11 \begin{array}{|l} 2 \\ \hline 1 \quad 5 \end{array} \longrightarrow 23 = 2 \times (2 \times 5 + 1) + 1 = 5 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 5 \begin{array}{|l} 2 \\ \hline 1 \quad 2 \end{array} \longrightarrow 23 = (2 \times 2 + 1) \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 2 \begin{array}{|l} 2 \\ \hline 0 \quad 1 \end{array} \longrightarrow 23 = (1 \times 2) \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 \qquad \qquad \qquad = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 \qquad \qquad \qquad = 23_{10}
 \end{array}$$

Regra prática:



1.2.3 Conversão de números fracionários

Regra de formação:

$$\text{Decimal: } 197,526_{10} = 1 \times 10^2 + 9 \times 10^1 + 7 \times 10^0 + 5 \times 10^{-1} + 2 \times 10^{-2} + 6 \times 10^{-3}$$

$$\text{Binário: } 101101,101 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

1.2.3.1 Conversão de binário para decimal

$$\begin{aligned}
 1101,111_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\
 &= 8 + 4 + 0 + 1 + 0,5 + 0,25 + 0,125 \\
 &= 13 + 0,875 \\
 &= 13,875_{10}
 \end{aligned}$$

1.2.3.2 Conversão de decimal para binário

$$\begin{aligned}
 35,625_{10} &= ?_2 \\
 35,625_{10} &= \underbrace{35}_{\text{parte inteira}}_{10} + \underbrace{0,625}_{\text{parte fracionária}}_{10}
 \end{aligned}$$

A conversão da parte inteira segue o procedimento já descrito:

$$35_{10} = 100011_2$$

A conversão da parte fracionária segue a seguinte regra prática:

Multiplica-se a parte fracionária pelo valor da base.

O número resultante a esquerda da vírgula é o dígito (0 ou 1) procurado.

Se o dígito à esquerda for 0 (zero) continuar a multiplicação pela base.

Se o dígito à esquerda for 1 este é retirado e prossegue-se a multiplicação.

O processo continua até obter-se 0 (zero) como resultado ou atingir-se a resolução estabelecida, no caso de dízima.

A leitura dos dígitos, ao contrário do caso da parte inteira, é feita de cima para baixo.

$$\begin{array}{r} 0,625_{10} \\ \times 2 \quad \leftarrow \text{base do sistema} \\ \hline \textcircled{1},250 \\ 0,25_{10} \\ \times 2 \\ \hline \textcircled{0},50 \\ \times 2 \\ \hline \textcircled{1},00 \end{array}$$

$$0,625_{10} = 0,101_2$$

$$35,625_{10} = 100011,101_2$$

Exercícios:

1. Converter para decimal:

- a) 110101010101_2
- b) 11011010101_2
- c) 11000001_2
- d) $1,01010101_2$
- e) $1001,010101_2$
- f) 1101_2
- g) $11,010101_2$
- h) $11010101,010101_2$
- i) $11010,1101_2$

2. Converter para binário com 5 casas após a vírgula:

- a) $234,435_{10}$
- b) $945,3445_{10}$
- c) 5235_{10}
- d) $12,234_{10}$
- e) $23,3457_{10}$
- f) $94,345_{10}$
- g) $43,956_{10}$
- h) $9,4567_{10}$

1.3 Sistema Octal de Numeração

A base de um sistema numérico é igual o número de dígitos que ela usa. Portanto, o sistema octal, que apresenta base 8, tem 8 dígitos a saber: 0, 1, 2, 3, 4, 5, 6, 7 (base $N=8$ dígitos $0 \quad N-1 = 7$).

Sua utilidade nos sistemas digitais vem do fato de que, associando-se os algarismos de um número binário (bits) em grupos de três, obtém-se uma correspondência direta com os dígitos do sistema octal. Observaremos nitidamente este mais adiante.

1.3.1 Conversão de Octal em Decimal

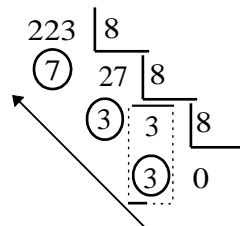
$$1247,235_8 = ?_{10}$$

$$1 \times 8^3 + 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} + 3 \times 8^{-2} + 5 \times 8^{-3}$$
$$512 + 128 + 32 + 7 + 1/8 + 3/64 + 5/512$$

$$1247,235_8 = 679,1816406_{10}$$

1.3.2 Conversão de Decimal para Octal

Converter o número 223 da base decimal para a octal.



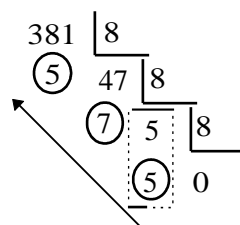
Parte fracionária:

$$223_{10} = 337_8$$

Converter o número fracionário 381,796 da base decimal para octal (4 casas decimais após a vírgula).

$$381,796_{10} = 381_{10} + 0,796_{10}$$

Parte inteira:



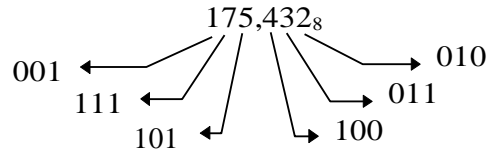
$$\begin{array}{r}
 381_{10} = 575_8 \\
 0,796_{10} \\
 \times 8 \\
 \hline
 \textcircled{6},368 \\
 0,368_{10} \\
 \times 8 \\
 \hline
 \textcircled{2},944 \\
 \times 8 \\
 \hline
 \textcircled{7},952 \\
 \times 8 \\
 \hline
 \textcircled{4},416 \\
 0,796_{10} \quad 0,6274_8 \text{ (aproximado)}
 \end{array}$$

1.3.3 Conversão de Octal em Binário

Para converter um número expresso em uma determinada base é normal convertermos o primeiro para um número na base 10 e, em seguida, fazer a conversão para a base desejada. Entretanto, como já foi dito, no caso do octal para o binário (e vice-versa) podemos fazer a conversão diretamente, sem passar pelo sistema decimal, já que, 8 é terceira potência de 2 e, portanto, são múltiplos e tem correspondência direta um com o outro.

Regra: Cada dígito octal, a partir da vírgula, é representado pelo equivalente a três dígitos binários. A tabela de equivalência é mostrada a seguir.

<i>Octa</i> <i>l</i>	<i>Binário</i>
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



$$175,432_{10} = 001\ 111\ 101,100\ 011\ 010_2$$

1.3.4 Conversão de Binário em Octal

Agrega-se os dígitos binários, a partir da vírgula, em grupos de três e converte-se para o equivalente em octal. Caso os dígitos extremos, da direita ou esquerda, não formarem um grupo completo de três, adiciona-se zeros até que isto ocorra.

Converter os seguintes números de binário para octal.

$$101110,011101_2 = \underline{101}\ \underline{110} , \underline{011}\ \underline{101}_2$$

5 6 , 3 5 8

$$1011,11101_2 = \underline{001}\ \underline{011} , \underline{111}\ \underline{010}_2$$

1 3 , 7 2 8

Converter o número 677_{10} para binário.

1ª alternativa: dividir 677_{10} sucessivamente por 2. Solução bastante extensa.

2ª alternativa: converter 677_{10} para octal e, em seguida, converter para binário. Solução menos trabalhosa).

$$677_{10} = 1245_8 = 1010100101_2$$

1.4 Sistema de Numeração Hexadecimal

Este sistema apresenta base igual a 16. Portanto 16 dígitos distintos. São usados os dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Como no sistema de numeração octal, o hexadecimal apresenta equivalência direta entre seus dígitos e grupos de quatro dígitos binários. A tabela a seguir mostra esta equivalência.

Decimal	Binário	Hexadecimal
---------	---------	-------------

0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1.4.1 Conversão de Hexadecimal para Decimal

A regra é a mesma da conversão de qualquer sistema de numeração para o decimal.

$$AFC0,7D_{16} = ?_{10}$$

$$A \times 16^3 + F \times 16^2 + C \times 16^1 + 0 \times 16^0 + 7 \times 16^{-1} + D \times 16^{-2}$$

$$10 \times 16^3 + 15 \times 16^2 + 12 \times 16^1 + 0 \times 16^0 + 7 \times 16^{-1} + 13 \times 16^{-2}$$

$$44992,48828_{10}$$

1.4.2 Conversão de Decimal para Hexadecimal

A regra é a mesma da conversão do decimal para qualquer sistema de numeração.

$$637,33_{10} = ?_{16}$$

$$637,33_{10} = 637_{10} + 0,33_{10}$$

Parte inteira

$$\begin{array}{r}
 637 \overline{) 16} \\
 \underline{13} \\
 39 \overline{) 16} \\
 \underline{7} \\
 2 \overline{) 16} \\
 \underline{2} \\
 0
 \end{array}$$

$637_{10} = 27D_{16}$

Parte Fracionária

$$\begin{array}{r}
 0,33_{10} \\
 \times 16 \\
 \hline
 \textcircled{5},28 \\
 \\
 0,28_{10} \\
 \times 16 \\
 \hline
 \textcircled{4},48 \\
 \\
 \times 16 \\
 \hline
 \textcircled{7},68 \\
 \\
 \times 16 \\
 \hline
 \textcircled{10},88
 \end{array}$$

$0,33_{10} \quad 0,547A_{16} \text{ (aproximado)}$

1.4.3 Conversão de Hexadecimal em Binário

Da mesma forma que no sistema octal, não é necessário converter o número para o sistema decimal e depois para binário. Basta representar cada dígito hexadecimal, a partir da vírgula, em grupos de quatro dígitos binários equivalentes. A base 16 é a quarta potência da base 2. A tabela de equivalência é a que foi apresentada acima.

$$FACA, CACA_{16} = ?_2$$

$$\begin{array}{cccccccc}
 F & A & C & A & , & C & A & C & A &_{16} \\
 1111 & 1010 & 1100 & 1010 & , & 1100 & 1010 & 1100 & 1010 &_2
 \end{array}$$

$$FACA, CACA_{16} = 1111101011001010,1100101011001010_2$$

1.4.4 Conversão de Binário para Hexadecimal

Como no caso da conversão de binário para octal, agrega-se os dígitos binários, a partir da vírgula, em grupos de quatro e converte-se para o equivalente em hexadecimal. Caso os dígitos extremos, da direita ou esquerda, não formarem um grupo completo de quatro, adiciona-se zeros até que isto ocorra.

$$100101010,00111_2 = ?_{16}$$

$$\begin{array}{ccccccc} 0001 & 0010 & 1010 & , & 0011 & 1000 & _2 \\ 1 & 2 & A & , & 3 & 8 & _{16} \end{array}$$

$$100101010,00111_2 = 12A,38_{16}$$

1.4.5 Tabela resumo de conversão de sistemas de numeração.

A tabela a seguir mostra um resumo das conversões entre os sistemas de numeração estudados. Vale lembrar que o nosso maior objetivo é a base 2. As bases 8 e 16 foram abordadas pois elas são as bases múltiplas da base 2 mais empregadas nos sistemas computacionais para facilitar a escrita da nossa base natural 10 para os sistemas digitais.

de/para	2	8	10	16
2				
8				
10				
16				

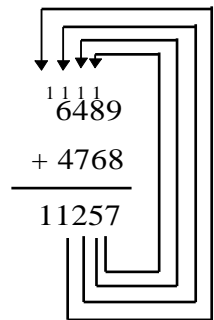
1.5 Operações Aritméticas em Sistemas de Numeração

Na Base 10, já temos bem fixada a idéia do “vai um” e “empresta um” pois foi assim que aprendemos desde o ensino fundamental. Mas isso foi um conceito que envolve o sistema decimal e temos a liberdade de utilizar quantos dígitos nosso cálculo desejar, antes e depois da vírgula. Para as outras bases 2, 8 e 16 alguns detalhes devem ser levados em consideração antes de fazer estas operações pois estamos tratando de bases que são utilizadas especialmente para sistemas computacionais que

tem a quantidade de representação de números limitada, seja inteiro positivo, negativo ou fracionário.

1.5.1 Adição e Subtração no Sistema de Numeração Decimal

Adição:




Vai um - Transporte - "Carry"

$$\begin{array}{r} \overset{1}{\overset{1}{\overset{1}{\overset{1}{\downarrow}}}} \\ 6489 \\ + 4768 \\ \hline 11257 \end{array}$$

Obs.: Toda vez que a soma de dígitos for maior ou igual ao valor da base (10) ocorre um vai um.

Subtração:



Em pre sta um - Ve m um - "Bor row"

$$\begin{array}{r} \overset{1}{\overset{1}{\overset{1}{\downarrow}}} \\ 8123 \\ - 678 \\ \hline 7445 \end{array}$$

1.5.2 Adição e Subtração no Sistema de Numeração Binário

Já sabemos que os circuitos digitais funcionam utilizando como base os números binários. Em microprocessadores e microcomputadores todas as operações aritméticas são realizadas através de somas e subtrações destes números binários. Ainda assim, para facilitar e economizar hardware, veremos que mesmo a subtração de dois números $A-B$ em base 2 é feita pela soma do minuendo (A) com seu subtraendo (B) representado como um número negativo.

Cabe então perguntar: Como serão os procedimentos que realizarão as operações de soma e subtração em circuitos digitais?

Este subitem responde esta questão. Primeiro veremos como representar números positivos e negativos através de seqüências de bits em um tamanho definido pelo hardware do sistema computacional, ou seja, pelo número de registradores de bits – em geral são 8 bits, mas podemos ter sistemas de 16, 32, 64 ou até mesmo menores que 8 bits. Em seguida veremos como realizar a soma ou subtração destes números levando em consideração um tamanho conhecido desta seqüência de bits. Nessa abordagem exploraremos somente números inteiros para facilitar o

entendimento. Para números fracionários o número de bits da sequência deve ser considerada para a parte fracionária da mesma forma que para a parte inteira. O procedimento é exatamente o mesmo.

1.5.2.1 Representação de Números Binários Sinalizados

A subtração (A-B) entre dois números binários A e B, é calculada como a soma entre o número A e o negativo do número B (-B). Para tanto, deve-se calcular o negativo do segundo número.

A representação binária de números, estudada até agora, referia-se a números positivos. Para representar números negativos serão utilizadas 3 representações; (1) sinal-magnitude, (2) complemento de um e (3) complemento de dois.

Sinal-Magnitude, neste caso o bit mais à esquerda é utilizado para o sinal (0 quando positivo e 1 quando negativo). Os bits restantes contêm o valor (magnitude) absoluto do valor. O número negativo é formado simplesmente trocando o bit de sinal do número positivo de 0 para 1. Por exemplo, os números $+9_{10}$ e -9_{10} em um formato de 8 bits serão:

$$\begin{aligned} +9_{10} &= 00001001_2 \\ -9_{10} &= 10001001_2 \end{aligned}$$

Sendo o formato de 8 bits, é possível representar $2^8=256$ números válidos. No entanto, existem apenas 255 números diferentes pois $+0$ (00000000_2) e -0 (10000000_2) representam o mesmo número. Assim, os números se estendem no intervalo de -127 até $+127$.

Complemento de um, o complemento de um de um número binário é obtido trocando todos os zeros por uns e os uns por zeros. utilizado para o sinal (0 quando positivo e 1 quando negativo). Por exemplo, os números $+9_{10}$ e -9_{10} em um formato de 8 bits serão:

$$\begin{aligned} +9_{10} &= 00001001_2 \\ -9_{10} &= 11110110_2 \end{aligned}$$

O bit mais à esquerda do número é 1 quando o número é negativo, e 0 quando o número é positivo. Novamente, em um formato de 8 bits existem $+0$ (00000000_2)

e -0 (11111111_2) representam o mesmo número e os números se estendem no intervalo de -127 até $+127$.

Complemento de dois, o complemento de dois de um número binário é obtido calculando primeiro o complemento de 1 do número e depois somando 1. Por exemplo, para os números $+9_{10}$ e -9_{10} em um formato de 8 bits, soma-se 1 ao número obtido no exemplo anterior (11110110_2) :

$$\begin{aligned} +9_{10} &= 00001001_2 \\ -9_{10} &= 11110111_2 \text{ (complemento de um +1)} \end{aligned}$$

O bit mais á esquerda do número também é 1 quando o número é negativo, e 0 quando o número é positivo. No formato de 8 bits, é possível representar $2^8=256$ números válidos, pois $+0$ (00000000_2) e -0 (00000000_2) são representados pela mesma seqüência binária. Os números, neste caso, se estendem no intervalo de -128 até 127 . Esta é a representação mais freqüentemente utilizada e a que adotaremos em nosso curso.

A tabela 1 mostra as três representações de números em sistema com seqüência de três bits.

Decimal	Sem Sinal	Sinal Magnitude	Complemento de 1	Complemento de 2
+7	111	-	-	-
+6	110	-	-	-
+5	101	-	-	-
+4	100	-	-	-
+3	011	011	011	011
+2	010	010	010	010
+1	001	001	001	001
+0	000	000	000	000
-0	-	100	111	000
-1	-	101	110	111
-2	-	110	101	110
-3	-	111	100	101
-4	-	-	-	100

Tabela 1: Representação de números de três bits.

1.5.2.2 Subtração Binária

Para realizar a subtração entre dois números, é necessário calcular o complemento de dois do subtraendo e somar com o minuendo. Isto resulta em economia de hardware e redução de sua complexidade. O bit mais significativo vai indicar se o resultado desta subtração ainda é um número negativo. Sendo, você pode saber sua magnitude se fizer o processo inverso ou seja, debitar de 1 o resultado e complementar todos os

bits, ou melhor ainda, realizar o complemento 2 sobre este número. Ou seja complementar 2 vezes é retornar ao número original em magnitude com sinal positivo. Se o resultado da soma (seja ela entre números positivos ou negativos) extrapolar o número da seqüência de bits definida pelo sistema digital, basta ignorar o bit excedente à esquerda do mais significativo da seqüência.

Exemplos:

$$5_{10} - 2_{10}$$

$$\begin{array}{r} +5 \\ -2 \\ \hline +3 \end{array} \quad + \begin{array}{r} 0101 \\ 1110 \\ \hline 10011 \end{array} = 0011$$

↑
Ignora-se

$$-5_{10} - 2_{10}$$

$$\begin{array}{r} -5 \\ -2 \\ \hline -7 \end{array} \quad + \begin{array}{r} 1011 \\ 1110 \\ \hline 11001 \end{array} = 1001$$

↑
Ignora-se

$$5_{10} - (-2_{10})$$

$$\begin{array}{r} +5 \\ -(-2) \\ \hline +3 \end{array} \quad + \begin{array}{r} 0101 \\ 0010 \\ \hline 0111 \end{array} = 0111$$

$$-5_{10} - (-2_{10})$$

$$\begin{array}{r} -5 \\ -(-2) \\ \hline -3 \end{array} \quad + \begin{array}{r} 1011 \\ 0010 \\ \hline 1101 \end{array} = 1101$$

1.5.2.3 Over Flow (estouro da capacidade da seqüência)

Over Flow é a mudança no sinal do resultado devido a realização de operações com números que levam ao estouro da capacidade do registrador (seqüência de bits). Esta situação ocorre quando se realiza operações equivalentes de soma de dois números positivos ou de dois números negativos.

Exemplos:

Utilizando um registrador de 4 bits, considerando representação em complemento de dois. O quarto bit à esquerda (o mais significativo) representa o sinal e os 3 menos significativos a magnitude:

a) 3+2

$$\begin{array}{r}
 0011 \\
 + 0010 \\
 \hline
 0101_2 \quad (= +5. \text{ Resultado correto!})
 \end{array}$$

b) 5+4

$$\begin{array}{r}
 0101 \\
 + 0100 \\
 \hline
 1001_2 \quad (= -7. \text{ Resultado errado! Houve over-flow. Número um no bit mais significativo indica número negativo, portanto pela representação de complemento de dois o resultado obtido foi } -7. \text{ Ou seja } 1001_2 - 1 = 1000_2, \text{ complementando-se os 3 bits menos significativos (magnitude) } = 111_2 (=7).
 \end{array}$$

c) -3 -2

$$\begin{array}{r}
 1101 \\
 + 1110 \\
 \hline
 1011_2 \quad (= -5. \text{ Resultado correto! Ou seja } 1011 - 1 = 1010, \text{ complementando-se os 3 bits menos significativos (magnitude) } = 101_2 (=5).
 \end{array}$$

d) -5 -4

$$\begin{array}{r}
 1011 \\
 + 1100 \\
 \hline
 0111_2 \quad (= +7. \text{ Resultado errado! Houve over-flow. Número zero no bit mais significativo indica número positivo portanto o resultado obtido foi } +7)
 \end{array}$$

1.5.3 Adição e Subtração no Sistema de Numeração Octal e Hexadecimal

A forma mais rápida e prática de efetuar uma operação aritmética em um número octal ou hexadecimal é transformá-lo em binário, efetuar a operação e depois reconvertê-lo para octal ou hexadecimal.

Exemplos:

$$\begin{array}{r}
 147_8 \\
 + 654_8 \\
 \hline
 1023_8
 \end{array}
 \qquad
 \begin{array}{r}
 \text{001100 111}_2 \\
 + \text{110 101 100}_2 \\
 \hline
 \underbrace{1}_{1} \underbrace{000}_{0} \underbrace{010}_{2} \underbrace{011}_{3_8}
 \end{array}$$

Exemplos: Transformar os números octais para binário e verificar se o resultado da operação está correto:

$$\begin{array}{r}
 147 \\
 - 121 \\
 \hline
 26
 \end{array}
 \qquad
 \begin{array}{r}
 100 \\
 + 37 \\
 \hline
 41
 \end{array}
 \qquad
 \begin{array}{r}
 6234 \\
 - 2351 \\
 \hline
 3663
 \end{array}$$

Exemplos: Transformar os números hexadecimais para binário e verificar se o resultado da operação está correto:

$$\begin{array}{r}
 \text{F0FC} \\
 + \text{A73} \\
 \hline
 \text{FB6F}
 \end{array}
 \qquad
 \begin{array}{r}
 900 \\
 + \text{CA1} \\
 \hline
 15\text{A1}
 \end{array}
 \qquad
 \begin{array}{r}
 \text{F731} \\
 - 11 \\
 \hline
 \text{F720}
 \end{array}
 \qquad
 \begin{array}{r}
 \text{BEBE} \\
 + \text{62DEB} \\
 \hline
 \text{6EC A9}
 \end{array}$$

1.5.4 Outras Operações Aritméticas em Sistemas de Numeração Binário, Octal e Hexadecimal

Não fica difícil perceber que operações aritméticas como multiplicação e divisão em sistemas digitais são feitos tais como a mesma idéia adotada na subtração que virou soma! Observe que uma multiplicação com sistema binário $1 \times 1 = 1$ e 0×0 , 0×1 ou 1×0 tem com o resultado 0. Lembrando de como você faz uma divisão ou multiplicação no sistema decimal somado a outras técnicas para tudo virar soma, acabamos entendendo que trabalhar com sistema binário é muito mais fácil do que parece. Estas e outras operações não fazem parte dos nossos objetivos deste curso. Se você quiser saber mais sobre isso, consulte uma literatura mais abrangente.

1.6 Exercícios Propostos:

1) Calcule:

- a) $11001 + 1011 + 1110$
- b) $10111 + 10001 + 10001 + 111110 + 11$
- c) $1001,11 + 11,111 + 11111,0101 + 1,1$
- d) $1111010 + 111111 + 101101$

2) Represente os seguintes números, utilizando a notação sinal-módulo (ou sinal-magnitude), a notação complemento de 1 e a notação complemento de 2:

- a) -42_{10}
- b) 50_8
- c) -251_{10}

3) Determine o complemento de 2 dos seguintes números:

- a) 1001001_2
- b) 1111001_2
- c) $1CFF_{16}$
- d) $2EE_{16}$

4) Utilizando o complemento de dois com um sistema de 8 bits, faça a subtração de:

- a) 11001_2 menos 10010_2
- b) 1111_2 menos 11010_2
- c) 11101_2 menos 10101_2

5) Um microprocessador de 32 bits usa a representação de complemento de 2. Como aparecem os seguintes números decimais (Será melhor expressar suas respostas em hexadecimal, para representar todos os bits):

- a) -190
- b) +137
- c) -225
- d) -147

6) Dados os seguintes números, converter para seu equivalente decimal

- a) $4CF_{16}$
- b) 110010001_2
- c) 011110010_2
- d) $1FA3_{16}$

6.1) Sendo que eles se encontram em sinal magnitude.

6.2) Sendo que eles se encontram em complemento de um.

6.3) Sendo que eles se encontram em complemento de dois.