2. Sistemas de Numeração, Operações e Códigos

Objetivos

- Revisar o sistema de numeração decimal
- Contar no sistema de numeração binário
- Converter de decimal para binário e vice-versa
- Aplicar operações aritméticas em números binários
- Determinar os complementos de 1 e de 2 de um número binário
- Expressar números binários sinalizados nos formatos sinal-magnitude, complemento de 1, complemento de 2 e ponto flutuante.
- Realizar conversões entre os sistemas de numeração binário e hexadecimal
- Somar números na forma hexadecimal
- Realizar conversões entre os sistemas de numeração binário e octal
- Expressar números decimais na forma de decimal codificado em binário (BCD)
- Somar números BCD
- Explicar como detectar e corrigir erros de código

Introdução

- O sistema de numeração binário e os códigos digitais são fundamentais para os computadores e para a eletrônica digital em geral
- Estudaremos o sistema de numeração binário e as suas relações com outros sistemas de numeração como decimal, hexadecimal e octal
- O método da paridade para detecção de erros em códigos é introduzido e um método de correção de erro é descrito

2. Sistemas de Numeração, Operações e Códigos

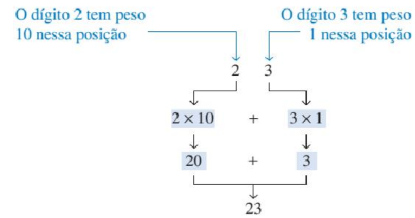
1. Números Decimais

Introdução

- Temos familiaridade com o sistema de numeração decimal porque deve usar números decimais todos os dias
- No sistema de numeração **decimal**, cada um dos dígitos, de 0 a 9, representa uma certa quantidade
- Como sabemos, os dez símbolos (dígitos) não nos limita a expressar apenas dez quantidades diferentes porque usamos vários dígitos posicionados adequadamente formando um número para indicar a magnitude (módulo) da quantidade

Introdução

- Se, por exemplo, queremos expressar a quantidade vinte e três, usamos (pela suas respectivas posições no número) o dígito 2 para representar a quantidade vinte e o dígito 3 para representar a quantidade três, conforme ilustrado a seguir



Introdução

- A posição de cada dígito em um número decimal indica a magnitude da quantidade representada e pode ser associada a um **peso**
- Os pesos para os números inteiros são potências de dez positivas que aumentam da direita para a esquerda, começando com $10^0 = 1$
  - ... 10^5 10^4 10^3 10^2 10^1 10^0
- Para números fracionários, os pesos são potências de dez negativas que diminuem da esquerda para a direita começando com 10^{-1}
  - 10^2 10^1 10^0,10^{-1} 10^{-2} 10^{-3}...

    vírgula decimal

Introdução

- Expresse o número decimal 568,23 como uma soma dos valores de cada dígito

$$
\begin{aligned}
568{,}23 &= (5 \times 10^2) + (6 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (3 \times 10^{-2}) \\
&= (5 \times 100) + (6 \times 10) + (8 \times 1) + (2 \times 0{,}1) + (3 \times 0{,}01) \\
&= \mathbf{500} + \mathbf{60} + \mathbf{8} + \mathbf{0{,}2} + \mathbf{0{,}03}
\end{aligned}
$$

Revisão

- 1. Qual é o peso que o dígito 7 tem em cada um dos seguintes números?
  (a) 1370 **(b)** 6725 **(c)** 7051 **(d)** 58,72
- 2.Expresse cada um dos seguintes números decimais como uma soma dos produtos obtidos pela multiplicação de cada dígito pelo peso apropriado:
  (a) 51 **(b)** 137 **(c)** 1492 **(d)** 106,58

Respostas

- 1. **(a)** 10 **(b)** 100 **(c)** 1000 **(d)** 0,1
- 2.
  - **(a)** 51 = (5 x 10) + (1 x 1)
  - **(b)** 137 = (1 x 100) + (3 x 10) + (7 ×1)
  - **(c)** 1492 = (1 x 1000) + (4 x 100) + (9 x 10) + (2 x 1)
  - **(d)** 106,58 =
    (1 x 100) + (0 x 10) + (6 x 1) + (5 x 0,1) + (8 x 0,01)

2. Sistemas de Numeração, Operações e Códigos

2. Números Binários

Introdução

- O sistema de numeração binário é uma outra forma de representar quantidades
- Ele é menos complicado que o sistema decimal porque usa apenas dois dígitos
- O sistema decimal com os seus dez dígitos é um sistema de base dez
  - O sistema binário com seus dois dígitos é um sistema de base dois
  - Os dois dígitos binários (bits) são 1 e 0
- Os pesos em um número binário são baseados em potência de dois

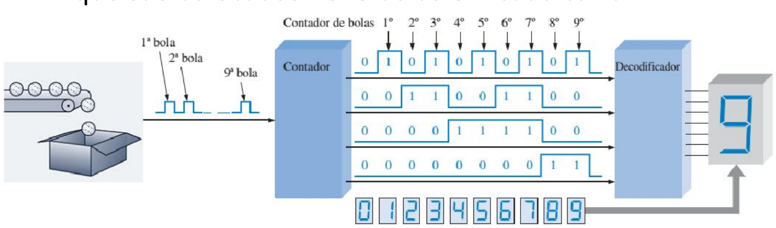
Contagem em Binário

- Começando a contagem: 0, 1
- Nesse momento, usamos os dois dígitos, assim incluímos uma nova posição de dígito e continuamos: 10, 11
- Esgotamos todas as combinações de dois dígitos, de forma que é necessário uma terceira posição
- Com posições para três dígitos podemos continuar a contagem: 100, 101, 110 e 111
- E assim por diante

NÚMERO DECIMAL	NÚMERO BINÁRIO			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Uma Aplicação

- Contagem de bola de tênis colocadas em uma caixa a partir de uma correia transportadora. Considere que são colocadas nove bolas em cada caixa



A Estrutura de Pesos dos Números Binários

- Um número binário é um número em que os dígitos apresentam pesos
- O bit mais à direita é o bit menos significativo (LSB - *least significant bit*) em um número inteiro binário e tem um peso de $2^0 = 1$
- Os pesos aumentam da direita para a esquerda em potências de dois para cada bit
- O bit mais à esquerda é o mais significativo (MSB - *most significant bit*); seu peso depende do tamanho do número binário

A Estrutura de Pesos dos Números Binários

- Números fracionários também podem ser representados em binário colocando os bits à direita da vírgula binária
- O bit mais à esquerda é o MSB em um número binário fracionário e tem um peso de 2^{-1}=0,5
- Os pesos da parte fracionária diminuem da esquerda para a direita por uma potência negativa de dois para cada bit

$$2^{n-1} \ldots 2^3\ 2^2\ 2^1\ 2^0,\ 2^{-1}\ 2^{-2} \ldots 2^{-n}$$

vírgula binária

A Estrutura de Pesos dos Números Binários

- Podemos estender facilmente a tabela dobrando o peso da potência de dois positiva mais significativa e reduzindo pela metade o peso da potência de dois negativa menos significativa, por exemplo, 2^9=512 e 2^{-7}=0,00787125

POTÊNCIAS DE DOIS POSITIVAS NÚMEROS INTEIROS									POTÊNCIAS DE DOIS NEGATIVAS NÚMEROS FRACIONÁRIOS					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0,5	0,25	0,125	0,0625	0,03125	0,015625

Conversão de Binário para Decimal

- O valor decimal de um número binário pode ser determinado somando-se os pesos de todos os bits que são 1 e descartando todos os pesos dos bits que são 0
- Converta o número binário inteiro 1101101 para decimal

$$\text{Peso: } 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$$
$$\text{Número binário: } 1\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1$$
$$\begin{aligned} 1101101 &= 2^6 + 2^5 + 2^3 + 2^2 + 2^0 \\ &= 64 + 32 + 8 + 4 + 1 = \mathbf{109} \end{aligned}$$

Conversão de Binário para Decimal

- Converta o número binário fracionário 0,1011 para decimal

$$\text{Peso: } 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}$$
$$\text{Número binário: } 0,1\ \ 0\ \ 1\ \ 1$$
$$\begin{aligned} 0{,}1011 &= 2^{-1} + 2^{-3} + 2^{-4} \\ &= 0{,}5 + 0{,}125 + 0{,}0625 = \mathbf{0{,}6875} \end{aligned}$$

Revisão

- 1. Qual é o maior número decimal que pode ser representado em binário por 8 bits?
- 2. Determine o peso do bit 1 no número binário 10000.
- 3. Converta o número binário 10111101,011 para decimal.

Respostas

- 1. $2^8 - 1 = 255$
- 2. O peso é 16.
- 3. 10111101,011 = 189,375

2. Sistemas de Numeração, Operações e Códigos

3. Conversão de Decimal para Binário

Introdução

- Ao final do estudo desta seção você deverá ser capaz de
  - Converter um número decimal para binário usando o método da soma dos pesos
  - Converter um número inteiro decimal para binário usando o método da divisão sucessiva por dois
  - Converter um número fracionário decimal para binário usando o método da multiplicação sucessiva por dois

Método da Soma dos Pesos

- Para obter um número binário a partir de um número decimal dado, determine os pesos que somados resultam no número decimal
- Um jeito fácil de lembrar dos pesos binários é saber que o menor dos pesos é 1, que corresponde a 2^0, e que dobrando esse peso obtemos o próximo peso de maior ordem; assim, uma lista de sete pesos em binário consta os pesos 64, 32, 16, 8, 4, 2, 1

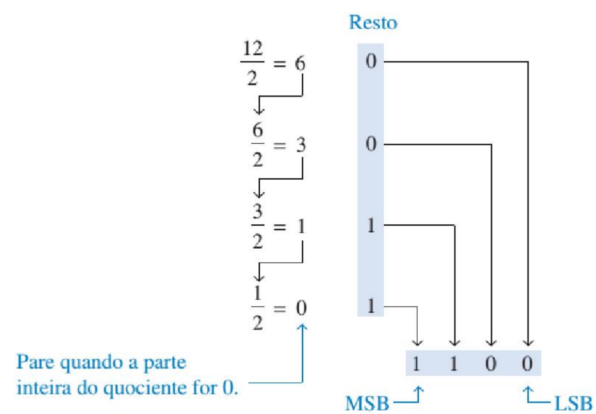
Método da Soma dos Pesos

- Converta os seguintes números decimais para binário
  - **(a)**12 **(b)** 25 **(c)** 58 **(d)** 82
- Solução
  - (a) $12 = 8 + 4 = 2^3 + 2^2 = 1100_2$
  - (b) $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0 = 11001_2$
  - (c) $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1 = 111010_2$
  - (d) $82 = 64 + 16 + 2 = 2^6 + 2^4 + 2^1 = 1010010_2$

Método da Divisão Sucessiva por 2

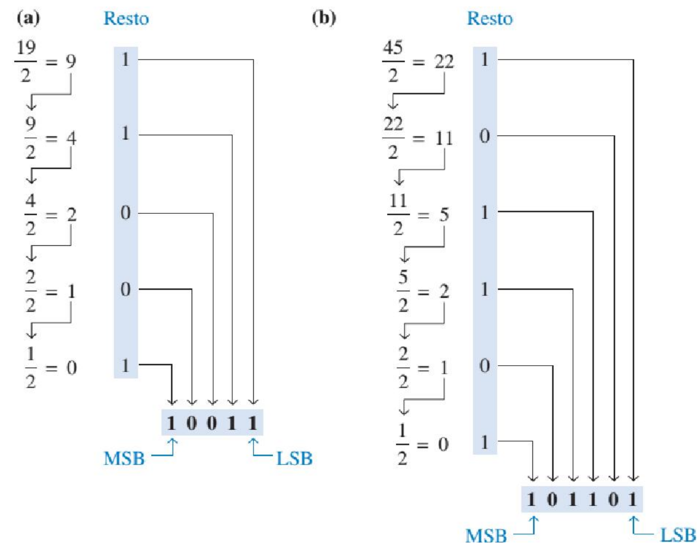
- Para obter o número binário que corresponde a um dado número decimal, divida o número decimal por 2 até que o quociente seja 0 (zero)
- Os restos formam o número binário
- O primeiro resto gerado é o LSB (bit menos significativo) no número binário e o último resto gerado é o MSB (bit mais significativo)

Método da Divisão Sucessiva por 2



Método da Divisão Sucessiva por 2

- Converta os seguintes números decimais em binário
  - **(a)** 19 **(b)** 45



Conversão de Decimal Fracionário em Binário

- Uma forma fácil de lembrar dos pesos da parte fracionária de um número binário é lembrar que o peso do bit mais significativo é 0,5, que equivale a 2^{-1}, e que dividindo qualquer peso por dois obtemos o próximo peso menos significativo; portanto, uma lista de quatro pesos binários fracionários seria 0,5; 0,25; 0,125; 0,0625

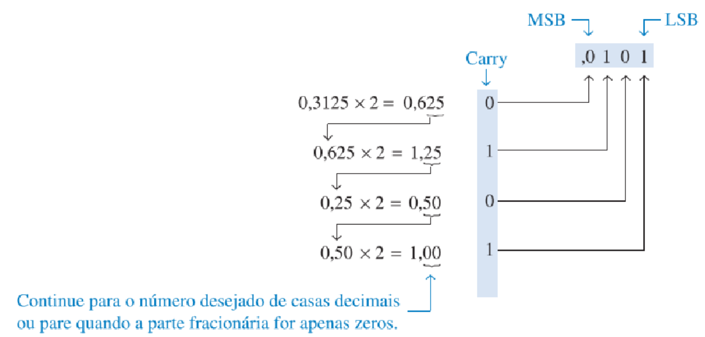
Conversão de Decimal Fracionário em Binário

- ***Soma dos Pesos***: O método da soma dos pesos pode ser aplicado a números decimais fracionários, conforme mostra o exemplo a seguir:
  - $0{,}625 = 0{,}5 + 0{,}125 = 2^{-1} + 2^{-3} = 0{,}101_2$

Conversão de Decimal Fracionário em Binário

- ***Multiplicações Sucessivas por 2***: Decimais fracionários podem ser convertidos para binário por meio de multiplicações sucessivas por 2
  - Para converter o fracionário decimal 0,3125 para binário, comece multiplicando 0,3125 por 2 e então multiplicar por 2 cada parte fracionária resultante do produto até que o produto seja 0 ou até que o número desejado de casas decimais seja alcançado
  - Os dígitos de *carry*, ou ***carries***, gerados pela multiplicação formam o número binário
  - O primeiro *carry* gerado é o MSB e o último é o LSB

Conversão de Decimal Fracionário em Binário



Revisão

- 1.Converta cada número decimal a seguir em binário usando o método da soma dos pesos.
  - **(a)** 23 **(b)** 57 **(c)** 45,5
- 2.Converta cada número decimal a seguir em binário usando o método das divisões sucessivas por 2 (multiplicações sucessivas por 2 no caso da parte fracionária):
  - **(a)** 14 **(b)** 21 **(c)** 0,375

Respostas

- 1.
  - **(a)** 23 = 10111
  - **(b)** 57 = 111001
  - **(c)** 45,5 = 101101,1
- 2.
  - **(a)** 14 = 1110
  - **(b)** 21 = 10101
  - **(c)** 0,375 = 0,011

2. Sistemas de Numeração, Operações e Códigos

4. Aritmética Binária

Introdução

- A aritmética binária é essencial em todos os computadores digitais e em muitos outros tipos de sistemas digitais
- Para entender os sistemas digitais, temos que saber os fundamentos das operações de soma, subtração, multiplicação e divisão em binário

Adição Binária

- As quatro regras básicas para a adição de dígitos binários (bits) são

$0 + 0 = 0$	Resultado: 0; carry: 0
$0 + 1 = 1$	Resultado: 1; carry: 0
$1 + 0 = 1$	Resultado: 1; carry: 0
$1 + 1 = 10$	Resultado: 0; carry: 1

Adição Binária

- Efetue as seguintes adições de números binários:
  - **(a)** 11 + 11 **(b)** 100 + 10 **(c)** 111 + 11 **(d)** 110 + 100

(a)
$$\begin{array}{r} 11 \\ \underline{+11} \\ \mathbf{110} \end{array} \qquad \begin{array}{r} 3 \\ \underline{+3} \\ 6 \end{array}$$

(b)
$$\begin{array}{r} 100 \\ \underline{+10} \\ \mathbf{110} \end{array} \qquad \begin{array}{r} 4 \\ \underline{+2} \\ 6 \end{array}$$

(c)
$$\begin{array}{r} 111 \\ \underline{+\ 11} \\ \mathbf{1010} \end{array} \qquad \begin{array}{r} 7 \\ \underline{+3} \\ 10 \end{array}$$

(d)
$$\begin{array}{r} 110 \\ \underline{+100} \\ \mathbf{1010} \end{array} \qquad \begin{array}{r} 6 \\ \underline{+4} \\ 10 \end{array}$$

Subtração Binária

- As quatro regras básicas para a subtração de bits são

$$0 - 0 = 0$$
$$1 - 1 = 0$$
$$1 - 0 = 1$$
$$10 - 1 = 1 \text{ sendo o empréstimo igual a } 1$$

- Quando subtraímos números, às vezes temos que fazer um empréstimo (*borrow*) da próxima coluna à esquerda
- Em binário um *borrow* é necessário apenas quando tentamos subtrair 1 de 0

Subtração Binária

- Efetue as seguintes subtrações binárias:
  - **(a)** 11 - 01 **(b)** 11 - 10

$$\textbf{(a)}\quad \begin{array}{r} 11 \\ -\,01 \\ \hline \mathbf{10} \end{array} \qquad \begin{array}{r} 3 \\ -\,1 \\ \hline 2 \end{array} \qquad \textbf{(b)}\quad \begin{array}{r} 11 \\ -\,10 \\ \hline \mathbf{01} \end{array} \qquad \begin{array}{r} 3 \\ -\,2 \\ \hline 1 \end{array}$$

- Efetue a subtração de 011 a partir de 101

$$\begin{array}{r} 101 \\ -011 \\ \hline \mathbf{010} \end{array} \qquad \begin{array}{r} 5 \\ -3 \\ \hline 2 \end{array}$$

Multiplicação Binária

- As quatro regras básicas para a multiplicação de bits são

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

Multiplicação Binária

- Realize as seguintes multiplicações binárias:
  - **(a)** 11 x 11 **(b)** 101 x 111

$$\textbf{(a)}\quad \begin{array}{r} 11 \\ \times\,11 \\ \hline \text{Produtos parciais} \left\{ \begin{array}{r} 11 \\ +11 \end{array} \right. \\ \hline \mathbf{1001} \end{array} \qquad \begin{array}{r} 3 \\ \times\,3 \\ \hline 9 \end{array} \qquad \textbf{(b)}\quad \begin{array}{r} 111 \\ \times\,101 \\ \hline \text{Produtos parciais} \left\{ \begin{array}{r} 111 \\ 000 \\ +111 \end{array} \right. \\ \hline \mathbf{100011} \end{array} \qquad \begin{array}{r} 7 \\ \times\,5 \\ \hline 35 \end{array}$$

Divisão Binária

- A divisão binária segue os mesmos procedimentos que a divisão decimal
- Realize as seguintes divisões binárias:
  - (a) 110 ÷ 11 (b) 110 ÷ 10

$$\textbf{(a)}\quad \begin{array}{r} \mathbf{10} \\ 11\overline{)110} \\ \underline{11} \\ 000 \end{array} \qquad \begin{array}{r} 2 \\ 3\overline{)6} \\ \underline{6} \\ 0 \end{array} \qquad\qquad \textbf{(b)}\quad \begin{array}{r} \mathbf{11} \\ 10\overline{)110} \\ \underline{10\ } \\ 10 \\ \underline{10} \\ 00 \end{array} \qquad \begin{array}{r} 3 \\ 2\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

Revisão

- 1.Realize as seguintes adições binárias:
  - **(a)** 1101 + 1010 **(b)** 10111 + 01101
- 2.Realize as seguintes subtrações binárias:
  - **(a)** 1101 - 0100 **(b)** 1001 - 0111
- 3.Realize as operações binárias indicadas:
  - **(a)** 110 ×111 **(b)** 1100 ÷ 011

Respostas

- 1.
  - **(a)** 1101 + 1010 = 10111
  - **(b)** 10111 + 01101 = 100100
- 2.
  - **(a)** 1101 - 0100 = 1001
  - **(b)** 1001 - 0111 = 0010
- 3.
  - **(a)** 110 × 111 = 101010
  - **(b)** 1100 ÷ 011 = 100

2. Sistemas de Numeração, Operações e Códigos

5. Complementos de 1 e de 2 de Números Binários

Introdução

- O complemento de 1 e o complemento de 2 de um número binário são importantes porque eles permitem a representação de números negativos
- O método da aritmética do complemento de 2 é geralmente usado em computadores na operação com números negativos

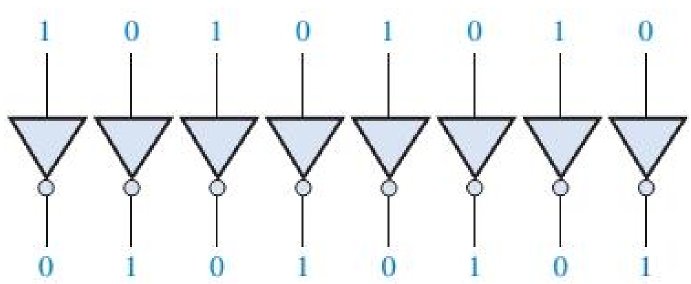
Determinação do Complemento de 1

- O **complemento** de 1 de um número binário é determinado trocando-se todos os 1s por 0s e todos os 0s por 1s

1	0	1	1	0	0	1	0	Número binário
↓	↓	↓	↓	↓	↓	↓	↓	
0	1	0	0	1	1	0	1	Complemento de 1

- A forma mais simples de obter o complemento de 1 de um número binário com um circuito digital é usar inversores em paralelo (circuitos NOT)

Determinação do Complemento de 1



Determinação do Complemento de 2

- O complemento de 2 de um número binário é determinado somando 1 ao LSB do complemento de 1
  - complemento de 2 = (complemento de 1) + 1
- Determine o complemento de 2 de 10110010.

10110010	Número binário
01001101	Complemento de 1
+ 1	Soma-se 1
01001110	Complemento de 2

Determinação do Complemento de 2

- Um método alternativo para determinar o complemento de 2 de um número binário é
  - 1. Comece à direita com o LSB e escreva os bits como eles aparecem até o primeiro 1 (inclusive)
  - 2. Tome o complemento de 1 dos bits restantes
- Determine o complemento de 2 de 10111000 usando o método alternativo

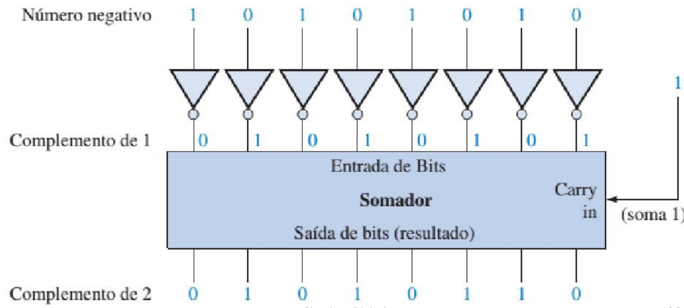
10111000 Número binário

Complemento de 1 dos bits originais → **01001000** Complemento de 2

Estes bits permanecem os mesmos

Determinação do Complemento de 2

- O complemento de 2 de um número binário negativo pode ser obtido usando inversores e um somador



Determinação do Complemento de 2

- Para converter a partir do complemento de 1 ou de 2 de volta para a forma binária verdadeira (não complementada), usamos os mesmos dois procedimentos descritos anteriormente
  - Para passar do complemento de 1 de volta para o binário verdadeiro, inverta todos os bits
  - Para passar do complemento de 2 de volta para a forma binária verdadeira, tome o complemento de 1 do número na forma do complemento de 2 e some 1 ao bit menos significativo

Revisão

- 1.Determine o complemento de 1 e cada número binário a seguir:
  - **(a)** 00011010
  - **(b)** 11110111
  - **(c)** 10001101
- 2.Determine o complemento de 2 de cada número binário a seguir:
  - **(a)** 00010110
  - **(b)** 11111100
  - **(c)** 10010001

Respostas

- 1.
  - **(a)** Compl. de 1 de 00011010 = 11100101
  - **(b)** Compl. de 1 de 11110111 = 00001000
  - **(c)** Compl. de 1 de 10001101 = 01110010
- 2.
  - **(a)** Compl. de 2 de 00010110 = 11101010
  - **(b)** Compl. de 2 de 11111100 = 00000100
  - **(c)** Compl. de 2 de 10010001 = 01101111

2. Sistemas de Numeração, Operações e Códigos

6. Números Sinalizados

Introdução

- Os sistemas digitais, como o computador, têm que ser capazes de operar com números positivos e negativos
- Um número binário sinalizado é constituído de duas informações: sinal e magnitude
  - O sinal indica se um número é positivo ou negativo e a magnitude é o valor do número
- Existem três formas por meio das quais os números inteiros podem ser representados em binário: sinal-magnitude, complemento de 1 e complemento de 2. Dentre esses, a forma do complemento de 2 é a mais importante e a forma sinal-magnitude é a menos usada. Os números fracionários (não-inteiros) e muito grandes ou muito pequenos podem ser expressos na forma de ponto flutuante

O Bit de Sinal

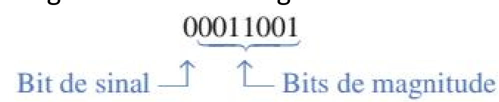
- O bit mais à esquerda em um número binário sinalizado é o bit de sinal, o qual nos diz se o número é positivo ou negativo
  - Um bit de sinal 0 indica um número positivo e um bit de sinal 1 indica um número negativo

Forma Sinal-Magnitude

- Quando um número binário sinalizado é representado na forma sinal-magnitude, o bit mais à esquerda é o bit de sinal e os bits restantes são os bits de magnitude
- Os bits de magnitude estão na forma de binário verdadeiro (não-complementado) tanto para números positivos quanto para negativos

Forma Sinal-Magnitude

- o número decimal +25 é expresso como um número binário sinalizado de 8 bits usando a forma sinal-magnitude como a seguir



- O número decimal −25 é expresso como
  - 10011001
  - Na forma sinal-magnitude, um número negativo tem os mesmos bits de magnitude como o número positivo correspondente mas o bit de sinal é 1 em vez de zero

Exemplo

- Expresse o número decimal –39 como um número de 8 bits nas formas sinal-magnitude, complemento de 1 e complemento de 2
- Primeiro escreva o número de 8 bits para +39
  - 00100111
- Na *forma sinal-magnitude*, –39 é gerado alterando o bit de sinal para 1 e deixando os bits de magnitude como estavam
  - 10100111
- Na *forma do complemento de 1*, 11011000
- Na *forma do complemento de 2*, 11011001

O Valor Decimal de Números Sinalizados

- **Sinal-magnitude**
  - Os valores decimais de números positivos e negativos na forma sinal-magnitude são determinados somando os pesos de todos os bits de magnitude que são 1s e ignorando aqueles que são zeros
  - O sinal é determinado pela análise do bit de sinal

O Valor Decimal de Números Sinalizados

- Determine o valor decimal do número binário que vem a seguir expresso na forma sinal-magnitude: 10010101
- Os sete bits de magnitude e os pesos em potências de dois são

2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	0	1	0	1

- Somando os pesos dos bits que são 1s temos
  - 16 +4 +1 =21
- O bit de sinal é 1; portanto, o número decimal é –21

O Valor Decimal de Números Sinalizados

- **Complemento de 1**
  - Valores decimais de números positivos na forma do complemento de 1 são determinados somando os pesos de todos os bits 1s e ignorando os pesos relativos aos zeros
  - Os valores decimais de números negativos são determinados atribuindo um valor negativo ao peso do bit de sinal, somando os pesos relativos aos bits 1s e somando 1 ao resultado

O Valor Decimal de Números Sinalizados

- Determine os valores decimais dos números binários sinalizados expressos em complemento de 1
  - **(a)** 00010111
  - **(b)** 11101000
- **(a)** Os bits e os respectivos pesos em potências de dois são

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	0	1	1	1

- Somando os pesos correspondentes aos bits 1, temos
  - 16 + 4 + 2 + 1 = +23

O Valor Decimal de Números Sinalizados

- **(b)** Os bits e os respectivos pesos em potências de dois para o número negativo são mostrados a seguir. Observe que o bit de sinal negativo tem um peso de -2^7 ou –128

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	0	0	0

- Somando os pesos em que os bits são 1s, temos:
  - –128 + 64 + 32 + 8 = –24
- Somando 1 ao resultado, o número decimal final é
  - –24 +1 =–23

O Valor Decimal de Números Sinalizados

- Complemento de 2
  - Valores decimais de números positivos e negativos na forma do complemento de 2 são determinados somando os pesos das posições de todos os bits 1s e ignorando as posições em que os bits são zeros
  - O peso do bit de sinal em números negativos é dado com um valor negativo
- Determine os valores decimais dos números binários sinalizados a seguir expressos na forma do complemento de 2
  - **(a)** 01010110
  - **(b)** 10101010

O Valor Decimal de Números Sinalizados

- **(a)** Os bits e seus respectivos pesos em potências de dois para números positivos são

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	0	1	1	0

- Somando-se os pesos relativos aos bits 1s, temos:
  - 64 + 16 + 4 + 2 = +86

O Valor Decimal de Números Sinalizados

- **(b)** Os bits e seus respectivos pesos em potências de dois para números positivos são os seguintes. Observe que o bit de sinal negativo tem um peso de -2^7 ou –128

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	0	1	0

- Somando-se os pesos relativos aos bits 1s, temos:
  - –128 + 32 + 8 + 2 = –86

Números em Ponto Flutuante

- Para representar números **inteiros** muito grandes, são necessários muitos bits
- Existe também um problema quando números que têm parte inteira e fracionária, como 23,5618, precisam ser representados
- O sistema de numeração de **ponto flutuante**, baseado em notação científica, é capaz de representar números muito grandes e muito pequenos sem o aumento do número de bits e também representa números que têm parte inteira e fracionária

Números em Ponto Flutuante

- Um **número em ponto flutuante** (também conhecido como número real) consiste em duas partes mais um sinal
  - A **mantissa** é a parte do número em ponto flutuante que representa a magnitude do número
  - O **expoente** é a parte do número em ponto flutuante que representa o número de casas decimais que a vírgula decimal (ou vírgula binária) é movida

Números em Ponto Flutuante

- Números Binários de Ponto Flutuante de Precisão Simples
  - o formato padrão para um número binário de precisão simples, o bit de sinal (S) é o bit mais à esquerda, o expoente (E) corresponde aos próximos 8 bits e a mantissa ou parte fracionária (F) inclui os 23 bits restantes

32 bits

S	Expoente (E)	Mantissa (fracionária, F)
1 bit	8 bits	23 bits

Números em Ponto Flutuante

- Converta o número decimal $3{,}248 \times 10^4$ para um número binário no formato de ponto flutuante de precisão simples
- Converta o número decimal em binário
  - $3{,}248 \times 10^4 = 32480 = 111111011100000_2 = 1{,}11111011100000 \times 2^{14}$
- O MSB não ocupa a posição de um bit porque ele é sempre um 1
  - Portanto, a mantissa é o número binário fracionário de 23 bits 11111011100000000000000 e o expoente polarizado é
  - $14 + 127 = 141 = 10001101_2$
- O número completo em ponto flutuante é

0	10001101	11111011100000000000000

Revisão

- 1. Expresse o número decimal +9 como um número binário de 8 bits no sistema sinal-magnitude.
- 2. Expresse o número decimal –33 como um número binário de 8 bits no sistema de complemento de 1.
- 3. Expresse o número decimal –46 como um número binário de 8 bits no sistema de complemento de 2.
- 4. Faça uma lista especificando as três partes de um número sinalizado no formato de ponto flutuante.

Respostas

- 1. Sinal-magnitude: +9 = 00001001
- 2. Complemento de 1: –33 =11011110
- 3. Complemento de 2: –46 =11010010
- 4. Bit de sinal, expoente e mantissa.

2. Sistemas de Numeração, Operações e Códigos

7. Números Hexadecimais

Introdução

- O sistema de numeração hexadecimal tem dezesseis caracteres; ele é usado principalmente como uma forma compacta de apresentar ou escrever números binários, e é muito fácil realizar conversões entre binário e hexadecimal
- Números binários longos são difíceis de serem lidos e escritos porque é fácil omitir ou trocar um bit
- Como os computadores entendem apenas 1s e 0s, é necessário usar esses dígitos quando se programa em "linguagem de máquina"
- Imagine escrever uma instrução de dezesseis bits para um sistema microprocessado em 1s e 0s
- É muito mais eficiente usar hexadecimal ou octal
- O sistema hexadecimal é bastante usado em aplicações de computador e microprocessador

Introdução

- O sistema de numeração **hexadecimal** tem uma base de dezesseis; ou seja, ele é composto de 16 **caracteres numéricos** e alfabéticos
- A maioria dos sistemas digitais processa dados binários em grupos que são múltiplos de quatro bits, tornando o número hexadecimal muito conveniente porque cada dígito hexadecimal representa um número binário de 4 bits

Introdução

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

Contagem em Hexadecimal

- Como contar em hexadecimal uma vez atingida a contagem F? Simplesmente inicie uma nova coluna e continue como mostrado a seguir:
  - 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31...
- Com dois dígitos hexadecimais, podemos contar até FF_{16}, que corresponde ao decimal 255
- Para contar além desse valor, são necessários três dígitos hexadecimais. Por exemplo, 100_{16} equivale ao decimal 256, 101_{16} equivale ao decimal 257 e assim por diante
- O maior número hexadecimal de três dígitos é FFF_{16}, que equivale ao decimal 4095. O maior número hexadecimal de quatro dígitos é $FFFF_{16}$, que equivale ao decimal 65.535

Conversão de Binário para Hexadecimal

- A conversão de um número binário para hexadecimal é um procedimento direto
- Simplesmente separe o número binário em grupos de 4 bits começando do bit mais à direita e substituindo cada grupo de 4 bits pelo símbolo hexadecimal equivalente

Conversão de Binário para Hexadecimal

- Converta os seguintes números binários para hexadecimal:
  - **(a)** 1100101001010111
  - **(b)** 111111000101101001

(a) $\underbrace{1100}_{\downarrow\, C}\underbrace{1010}_{\downarrow\, A}\underbrace{0101}_{\downarrow\, 5}\underbrace{0111}_{\downarrow\, 7} = \mathbf{CA57_{16}}$

(b) $\underbrace{0011}_{\downarrow\, 3}\underbrace{1111}_{\downarrow\, F}\underbrace{0001}_{\downarrow\, 1}\underbrace{0110}_{\downarrow\, 6}\underbrace{1001}_{\downarrow\, 9} = \mathbf{3F169_{16}}$

Conversão de Hexadecimal para Binário

- Para converter um número de hexadecimal para binário, o processo é inverso, sendo que substituímos cada símbolo hexadecimal pelos quatro bits correspondentes
- Determine os números binários correspondentes aos seguintes números hexadecimais:
  - (a) $10A4_{16}$
  - (b) $CF8E_{16}$
  - (b) 9742_{16}

Conversão de Hexadecimal para Binário

(a) $\overbrace{1}^{1}\overbrace{0000}^{0}\overbrace{1010}^{A}\overbrace{0100}^{4}$ = **1000010100100**

(b) $\overbrace{1100}^{C}\overbrace{1111}^{F}\overbrace{1000}^{8}\overbrace{1110}^{E}$ = **1100111110001110**

(c) $\overbrace{1001}^{9}\overbrace{0111}^{7}\overbrace{0100}^{4}\overbrace{0010}^{2}$ = **1001011101000010**

Conversão de Hexadecimal para Decimal

- Uma forma de determinar o equivalente decimal de um número hexadecimal é primeiro converter o número hexadecimal em binário e em seguida converter de binário para decimal
- Converta o seguinte número hexadecimal em decimal:
  - **(a)** $1C_{16}$
  - **(b)** $A85_{16}$

Conversão de Hexadecimal para Decimal

(a) $\overbrace{0001}^{1}\overbrace{1100}^{C} = 2^4 + 2^3 + 2^2 = 16 + 8 + 4 = \mathbf{28_{10}}$

(b) $\overbrace{1010}^{A}\overbrace{1000}^{8}\overbrace{0101}^{5} = 2^{11} + 2^9 + 2^7 + 2^2 + 2^0 =$
$2048 + 512 + 128 + 4 + 1 = \mathbf{2693_{10}}$

Conversão de Hexadecimal para Decimal

- Outra forma de converter um número hexadecimal no seu equivalente decimal é multiplicar o valor decimal de cada dígito hexadecimal pelo seu peso e então realizar a soma desses produtos
- Os pesos de um número hexadecimal são potências de 16 crescentes (da direita para a esquerda)
- Para um número hexadecimal de 4 dígitos, os pesos são

16^3	16^2	16^1	16^0
4096	256	16	1

Conversão de Hexadecimal para Decimal

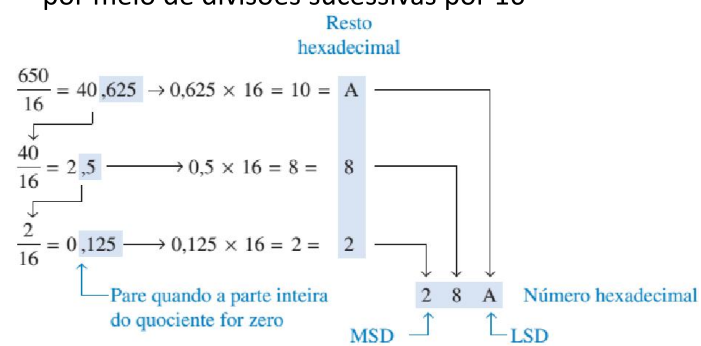
- Converta os seguintes números hexadecimais em números decimais:
  - **(a)** $E5_{16}$
  - **(b)** $B2F8_{16}$
- Resposta
  - **(a)** $E5_{16}$ = (E × 16) + (5 × 1) = (14 ×16) (5 ×1) = 224 + 5 = 229_{10}
  - **(b)** $B2F8_{16}$ = (B*4096) + (2*256) + (F*16) + (8*1)
    = (11*4096) + (2*256) + (15*16) + (8*1)
    = 45.056 + 512 + 240 + 8 = 45.816_{10}

Conversão de Decimal para Hexadecimal

- Divisões sucessivas de um número decimal por 16 produzem o número hexadecimal equivalente, formado pelos restos das divisões
- O primeiro resto produzido é o digito menos significativo (LSD – *least significant digit*)
- Cada divisão sucessiva por 16 resulta num resto que se torna num dígito no número hexadecimal equivalente
- Observe que quando o quociente tem uma parte fracionária, essa parte é multiplicada pelo divisor para se obter o resto

Conversão de Decimal para Hexadecimal

- Converta o número decimal 650 em hexadecimal por meio de divisões sucessivas por 16



Adição Hexadecimal

- A adição pode ser feita diretamente com números hexadecimais lembrando que os dígitos hexadecimais de 0 a 9 são equivalentes aos dígitos decimais de 0 a 9 e que os dígitos hexadecimais de A a F são equivalentes aos números decimais de 10 a 15
- Quando somar dois números hexadecimais, use as regras a seguir
  - 1. Para qualquer coluna de um problema de adição, pense nos dois dígitos hexadecimais em termos dos seus valores decimais. Por exemplo, $5_{16} = 5_{10}$ e $C_{16}=12_{10}$
  - 2. Se a soma dos dois dígitos for 15_{10} ou menos, registre o dígito hexadecimal correspondente
  - 3. Se a soma dos dois dígitos for maior que 15_{10}, registre o valor da soma que excede a 16_{10} e gere um *carry* de 1 para a próxima coluna

Adição Hexadecimal

- Efetue a soma dos seguintes números hexadecimais:
  - **(a)** $23_{16} + 16_{16}$
  - **(b)** $58_{16} + 22_{16}$
  - **(c)** $2B_{16} + 84_{16}$
  - **(d)** $DF_{16} + AC_{16}$

(a) $23_{16} + 16_{16} = \mathbf{39}_{16}$

coluna da direita: $3_{16} + 6_{16} = 3_{10} + 6_{10} = 9_{10} = 9_{16}$

coluna da esquerda: $2_{16} + 1_{16} = 2_{10} + 1_{10} = 3_{10} = 3_{16}$

(b) $58_{16} + 22_{16} = \mathbf{7A}_{16}$

coluna da direita: $8_{16} + 2_{16} = 8_{10} + 2_{10} = 10_{10} = A_{16}$

coluna da esquerda: $5_{16} + 2_{16} = 5_{10} + 2_{10} = 7_{10} = 7_{16}$

Adição Hexadecimal

(c) $2B_{16} + 84_{16} = \mathbf{AF}_{16}$

coluna da direita: $B_{16} + 4_{16} = 11_{10} + 4_{10} = 15_{10} = F_{16}$

coluna da esquerda: $2_{16} + 8_{16} = 2_{10} + 8_{10} = 10_{10} = A_{16}$

(d) $DF_{16} + AC_{16} = \mathbf{18B}_{16}$

coluna da direita: $F_{16} + C_{16} = 15_{10} + 12_{10} = 27_{10}$

$27_{10} - 16_{10} = 11_{10} = B_{16}$ com um carry de 1

coluna da esquerda: $D_{16} + A_{16} + 1_{16} = 13_{10} + 10_{10} + 1_{10} = 24_{10}$

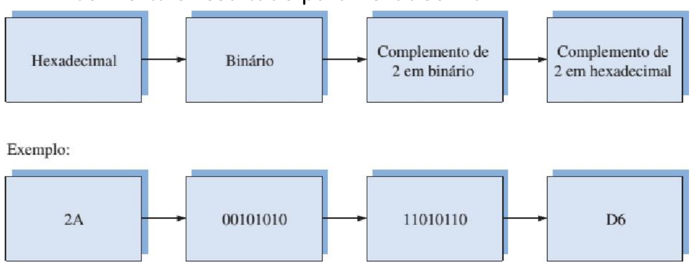
$24_{10} - 16_{10} = 8_{10} = 8_{16}$ com um carry de 1

Subtração Hexadecimal

- O complemento de 2 nos permite subtrair números binários por meio da adição
- Como um número hexadecimal pode ser usado para representar um número binário, ele também pode ser usado para representar o complemento de 2 de um número binário
- Existem três formas de obter o complemento de 2 de um número hexadecimal
  - O método 1 é o mais comum e fácil de ser usado
  - Os métodos 2 e 3 são alternativos

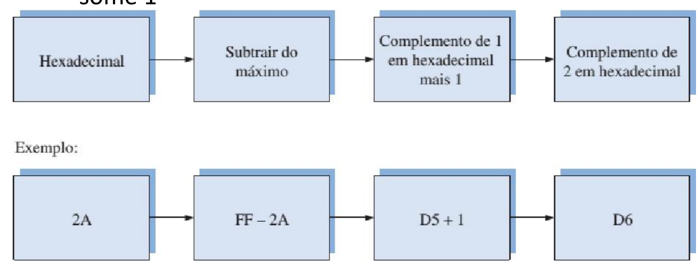
Subtração Hexadecimal

- Método 1
  - Converta o número hexadecimal para binário
  - Obtenha o complemento de 2 do número binário
  - Converta o resultado para hexadecimal



Subtração Hexadecimal

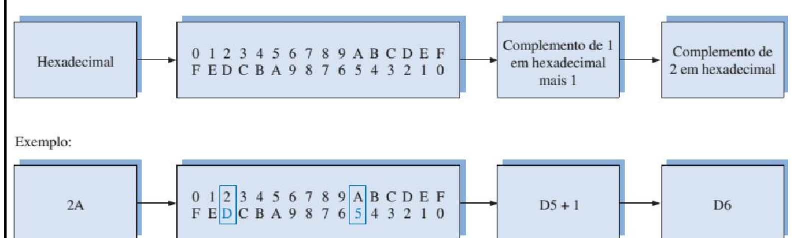
- Método 2
  - Subtraia o número hexadecimal do maior número hexadecimal (com a mesma quantidade de dígitos) e some 1



Subtração Hexadecimal

- Método 3
  - Escreva a sequência dos números hexadecimais de um dígito
  - Escreva a sequência inversa abaixo da sequência direta
  - O complemento de 1 de cada dígito hexa é o dígito diretamente abaixo dele
  - Some 1 ao número resultante para obter o complemento de 2

Subtração Hexadecimal



- Efetue a subtração dos seguintes números hexadecimais:
  - **(a)** $84_{16} + 2A_{16}$
  - **(b)** $C3_{16} + 0B_{16}$

(a) $2A_{16} = 00101010$

O complemento de 2 de $2A_{16} = 11010110 = D6_{16}$ (usando o Método 1)

$$\begin{array}{r} 84_{16} \\ +\ D6_{16} \\ \hline \cancel{1}5A_{16} \end{array}$$

Soma

Desconsiderar o carry, como na adição do complemento de 2

A diferença é $\mathbf{5A_{16}}$.

(b) $0B_{16} = 00001011$

O complemento de 2 de $0B_{16} = 11110101 = F5_{16}$ (usando o Método 1)

$$\begin{array}{r} C3_{16} \\ +\ F5_{16} \\ \hline \cancel{1}B8_{16} \end{array}$$

Soma

Desconsiderar o carry

A diferença é $\mathbf{B8_{16}}$.

Revisão

- 1. Converta os seguintes números binários em hexadecimais.
  - **(a)** 10110011 **(b)** 110011101000
- 2. Converta os seguintes números hexadecimais em binários.
  - **(a)** 57_{16} **(b)** $3A5_{16}$ **(c)** $F80B_{16}$
- 3. Converta $9B30_{16}$ em decimal.
- 4. Converta o número decimal 573 em hexadecimal.
- 5. Some os seguintes números hexadecimais diretamente:
  - **(a)** $18_{16} + 34_{16}$ **(b)** $3F_{16} + 2A_{16}$
- 6. Efetue as seguintes subtrações de números hexadecimais.
  - **(a)** $75_{16} - 21_{16}$ **(b)** $94_{16} - 5C_{16}$

Respostas

- 1. **(a)** $10110011 = B3_{16}$ **(b)** $110011101000 = CE8_{16}$
- 2. **(a)** $57_{16} = 01010111$ **(b)** $3A5_{16} = 001110100101$
- **(c)** $F8OB_{16} = 1111100000001011$
- 3. $9B30_{16} = 39.728_{10}$
- 4. $573_{10} = 23D_{16}$
- 5. **(a)** $18_{16} + 34_{16} = 4C_{16}$ (b) $3F_{16} + 2A_{16} = 69_{16}$
- 6. **(a)** $75_{16} - 21_{16} = 54_{16}$ **(b)** $94_{16} - 5C_{16} = 38_{16}$

2. Sistemas de Numeração, Operações e Códigos

8. Números Octais

Introdução

- Assim como o sistema de numeração hexadecimal, o sistema de numeração octal proporciona uma forma conveniente de expressar números binários e códigos
- Entretanto, ele é usado menos frequentemente que o sistema hexadecimal em conjunção com computadores e microprocessadores para expressar quantidades binárias para fins de entrada e saída

Introdução

- O sistema de numeração **octal** é composto de oito dígitos, os quais são
  - 0, 1, 2, 3, 4, 5, 6, 7
- Para contar acima de 7, inicie uma nova coluna e continue
  - 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, ...
- A contagem em octal é similar à contagem em decimal, exceto que os dígitos 8 e 9 não são usados

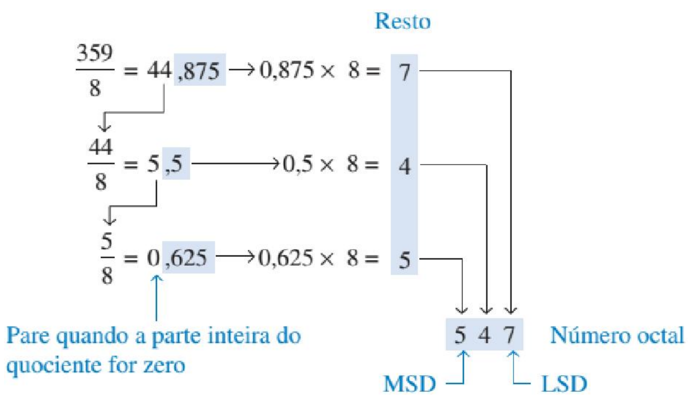
Conversão de Octal para Decimal

- Como o sistema de numeração octal tem uma base de oito, cada posição sucessiva de um dígito é uma potência crescente de oito, começando pela coluna mais à direita com 8^0
- O cálculo de um número octal em termos do seu equivalente decimal é realizado multiplicando-se cada dígito pelo seu peso e somando os produtos

Conversão de Decimal para Octal

- Um método de conversão de um número decimal para octal é o da divisão sucessiva por 8, similar ao método usado na conversão de números decimais para binário ou para hexadecimal
- Para mostrar como se faz, vamos converter o número decimal 359 para octal
- Cada divisão sucessiva por 8 resulta num resto que se torna um dígito do número octal equivalente
- O primeiro resto gerado é o dígito menos significativo (LSD)

Conversão de Decimal para Octal



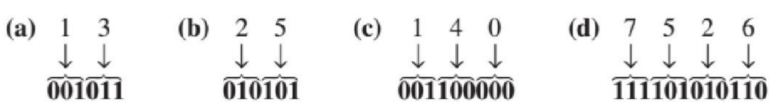
Conversão de Octal para Binário

- Como o dígito octal pode ser representado por 3 bits, é muito fácil converter de octal para binário
- Cada dígito octal é representado por três bits

DÍGITO OCTAL	0	1	2	3	4	5	6	7
BINÁRIO	000	001	010	011	100	101	110	111

Conversão de Octal para Binário

- Converta cada um dos seguintes números octais para binário
  - **(a)** 13_8
  - **(b)** 25_8
  - **(c)** 140_8
  - **(d)** 7526_8



Conversão de Binário para Octal

- A conversão de binário para octal é a operação inversa da conversão de octal para binário
- O procedimento é o seguinte: comece pelo grupo de três bits mais à direita e, percorrendo os grupos de bits da direita para a esquerda, converta cada grupo no seu dígito octal correspondente
- Caso o grupo mais à esquerda não tiver três bits, acrescente um ou dois zeros para completar o grupo
- Esses zeros à esquerda não afetam o valor do número binário

Conversão de Binário para Octal

- Converta cada número binário a seguir no seu equivalente em octal:
  - **(a)** 110101 **(b)** 101111001
  - **(c)** 100110011010 **(d)** 11010000100

(a) $\underbrace{110}_{6}\underbrace{101}_{5} = \mathbf{65_8}$

(b) $\underbrace{101}_{5}\underbrace{111}_{7}\underbrace{001}_{1} = \mathbf{571_8}$

(c) $\underbrace{100}_{4}\underbrace{110}_{6}\underbrace{011}_{3}\underbrace{010}_{2} = \mathbf{4632_8}$

(d) $\underbrace{011}_{3}\underbrace{010}_{2}\underbrace{000}_{0}\underbrace{100}_{4} = \mathbf{3204_8}$

Revisão

- 1. Converta os seguintes números octais em decimais:
  - **(a)** 73_8 **(b)** 125_8
- 2.Converta os seguintes números decimais em octais:
  - **(a)** 98_{10} **(b)** 163_{10}
- 3.Converta os seguintes números octais em binários:
  - **(a)** 46_8 **(b)** 723_8 **(c)** 5624_8
- 4.Converta os seguintes números binários em octais:
  - **(a)** 110101111 **(b)** 1001100010 **(c)** 10111111001

Respostas

- 1.
  - **(a)** $73_8 = 59_{10}$ **(b)** $125_8 = 85_{10}$
- 2.
  - **(a)** $98_{10} = 142_8$ **(b)** $163_{10} = 243_8$
- 3.
  - **(a)** 46_8 = 100110 **(b)** 723_8 = 111010011
  - **(c)** 5624_8 = 101110010100
- 4.
  - (a) 110101111 = 657_8 (b) 1001100010 = 1142_8
  - (c) 10111111001 = 2771_8

2. Sistemas de Numeração, Operações e Códigos

9. Decimal Codificado em Binário (BCD, *Binay Coded Decimal*)

Introdução

- Decimal codificado em binário (BCD – *binary coded decimal*) é uma forma de expressar cada dígito decimal com um código binário
- Existem apenas dez grupos de códigos no sistema BCD, de forma que é muito fácil converter decimal em BCD
- Como preferimos ler e escrever em decimal, o código BCD provê uma excelente interface com o sistema binário
- Exemplos de tais interfaces são as entradas do teclado e leituras digitais

O Código 8421

- O código 8421 é um tipo de código BCD (decimal codificado em binário)
- Decimal codificado em binário significa que cada dígito decimal, de 0 a 9, é representado por um código binário de quatro bits
- A designação 8421 indica os pesos binários dos quatro bits (2^3, 2^2, 2^1, 2^0)
- A facilidade de conversão entre números em código 8421 e números decimais é a principal vantagem desse código
- Tudo o que precisamos fazer é lembrar as dez combinações binárias que representam os dez dígitos
- O código 8421 é o código BCD predominante, e quando nos referirmos a BCD, queremos dizer que o código é o 8421, a menos que seja relatado o contrário

O Código 8421

DÍGITO DECIMAL	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

- Códigos inválidos
  - Percebemos que, com quatro bits, podemos representar dezesseis números (de 0000 a 1111), porém, no código 8421, apenas dez deles são usados
  - As seis combinações do código que não são usadas (1010, 1011, 1100, 1101, 1110 e 1111) são inválidas no código BCD 8421

O Código 8421

- Converta em BCD cada um dos seguintes números decimais
  - **(a)** 35 **(b)** 98 **(c)** 170 **(d)** 2469

(a) $\underbrace{0011}_{3}\underbrace{0101}_{5}$

(b) $\underbrace{1001}_{9}\underbrace{1000}_{8}$

(c) $\underbrace{0001}_{1}\underbrace{0111}_{7}\underbrace{0000}_{0}$

(d) $\underbrace{0010}_{2}\underbrace{0100}_{4}\underbrace{0110}_{6}\underbrace{1001}_{9}$

O Código 8421

- Determinar um número decimal a partir de um número BCD é igualmente fácil
- Comece pelo bit mais à direita separando o código em grupos de 4 bits
- Em seguida, escreva o dígito representado por cada grupo de quatro bits

O Código 8421

- Converta cada um dos seguintes códigos BCD em decimal:
  - **(a)** 10000110 **(b)** 001101010001
  - **(c)** 1001010001110000

(a) $\underbrace{1000}_{8}\underbrace{0110}_{6}$ **(b)** $\underbrace{0011}_{3}\underbrace{0101}_{5}\underbrace{0001}_{1}$ **(c)** $\underbrace{1001}_{9}\underbrace{0100}_{4}\underbrace{0111}_{7}\underbrace{0000}_{0}$

Adição em BCD

- BCD é um código numérico e pode ser usado em operações aritméticas
- A adição é a operação mais importante porque as outras três operações (subtração, multiplicação e divisão) podem ser realizadas através da adição

Adição em BCD

- Eis como dois números BCD são somados:
  - Passo 1 Some os dois números BCD, usando as regras de adição binária
  - Passo 2 Se um resultado de 4 bits for igual ou menor que 9, ele é um número BCD válido
  - Passo 3 Se um resultado de 4 bits for maior que 9, ou se um *carry* de saída de um grupo de 4 bits for gerado, ele será um resultado inválido
  - Some 6 (0110) ao resultado de 4 bits para “pular” os seis estados inválidos e retornar ao código 8421
  - Se ocorrer um *carry* quando 6 for somado, simplesmente acrescente o *carry* ao próximo grupo de 4 bits

Adição em BCD

- Some os seguintes números BCD:
  - **(a)** 0011 + 0100
  - **(b)** 00100011 + 00010101
  - **(c)** 10000110 + 00010011
  - **(d)** 010001010000 + 010000010111

(a)

0011		3
+ 0100		+ 4
0111		7

(b)

0010	0011	23
+ 0001	0101	+ 15
0011	**1000**	38

Adição em BCD

(c)

1000	0110	86
+ 0001	0011	+ 13
1001	**1001**	99

(d)

0100	0101	0000	450
+ 0100	0001	0111	+ 417
1000	**0110**	**0111**	867

Adição em BCD

- Some os seguintes números BCD:
  - **(a)** 1001 + 0100
  - **(b)** 1001 + 1001
  - **(c)** 00010110 + 00010101
  - **(d)** 01100111 + 01010011

(a)

	1001		9
	+ 0100		+4
	1101	Número BCD inválido (>9)	13
	+ 0110	Somar 6	
0001	**0011**	Número BCD válido	
↓	↓		
1	3		

Adição em BCD

(b)

	1001		9
	+ 1001		+ 9
1	0010	Inválido por causa do carry	18
	+ 0110	Somar 6	
0001	**1000**	Número BCD válido	
↓	↓		
1	8		

Adição em BCD

(c)	0001	0110		16
	+ 0001	0101		+ 15
	0010	1011	O grupo da direita é inválido (>9) e o grupo da esquerda é válido.	31
		+ 0110	Some 6 ao código inválido. Some o carry, 0001, ao próximo grupo.	
	0011	**0001**	Número BCD válido	
	↓	↓		
	3	1		

Adição em BCD

(d)		0110	0111		67
		+ 0101	+ 0011		+ 53
		1011	1010	Ambos os grupos são inválidos (>9)	120
		+ 0110	0110	Some 6 aos dois grupos	
	0001	**0010**	**0000**	Número BCD válido	
	↓	↓	↓		
	1	2	0		

Revisão

- 1.Qual é o peso binário de cada bit 1 nos números BCD a seguir?
  - **(a)** 0010 **(b)** 1000 **(c)** 0001 **(d)** 0100
- 2.Converta os seguintes números decimais em números BCD:
  - **(a)** 6 **(b)** 15 **(c)** 273 **(d)** 849
- 3.Quais números decimais são representados por cada código BCD?
  - **(a)** 10001001 **(b)** 001001111000 **(c)** 000101010111
- 4.Na adição BCD, quando um resultado de 4 bits é inválido?

Respostas

- 1. **(a)** 0010:2 **(b)** 1000:8 **(c)** 0001:1 **(d)** 0100:4
- 2. **(a)** 6_{10} = 0110 **(b)** 15_{10} = 00010101
  (c) 273_{10} = 001001110011
  (d) 849_{10} = 100001001001
- 3. **(a)** 10001001 = 89_{10}
  (b) 001001111000 = 278_{10}
  (c) 000101010111 = 157_{10}
- 4. Um resultado de 4 bits é inválido quando ele for maior que 9_{10}

2. Sistemas de Numeração, Operações e Códigos

10. Códigos de Detecção e Correção de Erro

Método da Paridade para Detecção de Erro

- Muitos sistemas usam um bit de paridade como um meio de **detecção de erro** de bit
- Qualquer grupo de bits possui um número de 1s par ou ímpar
- Um bit de paridade é acrescentado a um grupo de bits para tornar o número de 1s no grupo sempre par ou sempre ímpar
- Um bit de paridade par torna o número de 1s par e um bit de paridade ímpar torna ímpar o total de bits

Método da Paridade para Detecção de Erro

- Um dado sistema pode operar com **paridade** par ou ímpar, porém não ambas
- Por exemplo, se um sistema opera com paridade par, é feita uma verificação em cada grupo de bits recebido para certificar-se de que o número total de 1s no grupo seja par
- Caso exista um número ímpar de 1s, ocorreu um erro
- Detecção de um Erro
  - Um bit de paridade provê a detecção de erro num único bit (ou qualquer número ímpar de erros, que é bem pouco provável) mas não pode verificar dois erros num grupo

Método da Paridade para Detecção de Erro

PARIDADE PAR		PARIDADE ÍMPAR	
P	BCD	*P*	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001

Método da Paridade para Detecção de Erro

- Associe o bit de paridade par apropriado para os seguintes grupos de códigos
  - **(a)** 1010 **(b)** 111000 **(c)** 101101
  - **(d)** 1000111001001 **(e)** 101101011111
- Faça o bit de paridade 0 ou 1 conforme necessário para tornar o número total de 1s par
  - O bit de paridade será o bit mais à esquerda (colorido)

(a) 01010 (b) 1111000 (c) 0101101

(d) 01000111001001 (e) 1101101011111

Método da Paridade para Detecção de Erro

- Um sistema de paridade ímpar recebe os seguintes grupos de código: 10110, 11010, 110011, 110101110100 e 1100010101010
  - Determine quais grupos, se houver algum, estão com erro
- Como é informado que a paridade é ímpar, qualquer grupo com um número par de 1s está incorreto
  - Os seguintes grupos estão com erro: **110011** e **1100010101010**

Revisão

- 1.Qual código de paridade ímpar está errado?
  - **(a)** 1011 **(b)** 1110 **(c)** 0101 **(d)** 1000
- 2.Qual código de paridade par está errado?
  - **(a)** 11000110 **(b)** 00101000
  - **(c)** 10101010 **(d)** 11111011
- 3.Acrescente um bit de paridade par no final de cada um dos seguintes códigos:
  - **(a)** 1010100 **(b)** 0100000
  - **(c)** 1110111 **(d)** 10001100

Respostas

- 1. (c) 0101 tem um erro.
- 2. (d) 11111011 tem um erro.
- 3.
  - **(a)** 10101001 **(b)** 01000001
  - **(c)** 11101110 **(d)** 100011001

2. Sistemas de Numeração, Operações e Códigos

Resumo

Resumo

- O número binário é um número posicional em que o peso de cada dígito de um número inteiro é uma potência positiva de dois e o peso de cada dígito da parte fracionária é uma potência de dois negativa. Os pesos num número inteiro aumentam da direita para a esquerda (do dígito menos significativo para o mais significativo).
- Um número binário pode ser convertido para um número decimal somando os valores decimais dos pesos de todos os 1s no número binário.

Resumo

- Um número inteiro decimal pode ser convertido em binário usando a soma dos pesos ou o método da divisão sucessiva por 2.
- Um número decimal fracionário pode ser convertido para binário usando a soma dos pesos ou o método da multiplicação sucessiva por 2.
- As regras básicas para a adição binária são:
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 10

Resumo

- As regras básicas para a subtração binária são:
  - 0 − 0 = 0
  - 1 − 1 = 1
  - 1 − 0 = 1
  - 10 − 1 = 1
- O complemento de 1 de um número binário é obtido trocando 1s por 0s e 0s por 1s.
- O complemento de 2 de um número binário é obtido somando 1 ao complemento de 1.
- A subtração binária pode ser realizada por meio de adição usando o método do complemento de 1 ou de 2.

Resumo

- Um número binário positivo é representado por um bit de sinal 0.
- Um número binário negativo é representado por um bit de sinal 1.
- Para operações aritméticas, os números binários negativos são representados na forma do complemento de 2 ou complemento de 1.
- O sistema de numeração hexadecimal consiste de 16 dígitos e caracteres, de 0 a 9 seguidos de A até F.

Resumo

- Um dígito hexadecimal representa um número de 4 bits sendo a sua principal finalidade a simplificação de padrões de bits tornando-os de fácil leitura.
- Um número decimal pode ser convertido para hexadecimal usando o método da divisão sucessiva por 16.
- O sistema de numeração octal consiste de oito dígitos, de 0 a 7.
- Um número decimal pode ser convertido para octal usando o método da divisão sucessiva por 8.

Resumo

- A conversão de octal para binário é realizada simplesmente substituindo cada dígito octal pelo seu equivalente binário de 3 bits. O processo é invertido na conversão de binário para octal.
- Um número decimal é convertido para BCD substituindo cada dígito decimal pelo código binário de 4 bits apropriado.
- Um bit de paridade é usado para detectar um erro num código.

2. Sistemas de Numeração, Operações e Códigos

Exercícios de Fixação

Exercícios de Fixação

- 1. $2 \times 10^1 + 8 \times 10^0$ é igual a
  - (a) 10 (b) 280 (c) 2,8 (d) 28
- 2. O número binário 1101 é igual ao no número decimal
  - (a) 13 (b) 49 (c) 11 (d) 3
- 3. O número binário 11011101 é igual ao número decimal
  - (a) 121 (b) 221 (c) 441 (d) 256
- 4. O número decimal 17 é igual ao número binário
  - (a) 10010 (b) 11000 (c) 10001 (d) 01001

Exercícios de Fixação

- 5. O número decimal 175 é igual ao número binário
  - (a) 11001111 (b) 10101110 (c) 10101111 (d) 11101111
- 6. O resultado da soma de 11010 +01111 é igual a
  - (a) 101001 (b) 101010 (c) 110101 (d) 101000
- 7. A diferença de 110 – 010 é igual a
  - (a) 001 (b) 010 (c) 101 (d) 100
- 8. O complemento de 1 de 10111001 é
  - (a) 01000111 (b) 01000110 (c) 11000110 (d) 10101010
- 9. O complemento de 2 de 11001000 é
  - (a) 00110111 (b) 00110001 (c) 01001000 (d) 00111000

Exercícios de Fixação

- 10. O número decimal +122 é expresso na forma do complemento de 2 como
  - (a) 01111010 (b) 11111010 (c) 01000101 (d) 10000101
- 11. O número decimal –34 é expresso na forma do complemento de 2 como
  - (a) 01011110 (b) 10100010 (c) 11011110 (d) 01011101
- 12. Um número binário de ponto flutuante de precisão simples tem um total de
  - (a) 8 bits (b) 16 bits (c) 24 bits (d) 32 bits

Exercícios de Fixação

- 13. Na forma do complemento de 2, o número binário 10010011 é igual ao número decimal
  - (a) -19 (b) +109 (c) +91 (d) -109
- 14. O número binário 101100111001010100001 pode ser escrito em octal como
  - (a) 5471230_8 (b) 5471241_8 (c) 2634521_8 (d) 23162501_8
- 15. O número binário 10001101010001101111 pode ser escrito em hexadecimal como
  - (a) $AD467_{16}$ (b) $8C46F_{16}$ (c) $8D46F_{16}$ (d) $AE46F_{16}$

Exercícios de Fixação

- 16. O número binário equivalente a $F7A9_{16}$ é
  - (a) 1111011110101001 (b) 1110111110101001
  - (c) 1111111010110001 (d) 1111011010101001
- 17. O número BCD para o decimal 473 é
  - (a) 111011010 (b) 110001110011
  - (c) 010001110011 (d) 010011110011
- 18. O código que tem erro de paridade par é
  - (a) 1010011 (b) 1101000 (c) 1001000 (d) 1110111

Gabarito

- 1. (d)
- 2. (a)
- 3. (b)
- 4. (c)
- 5. (c)
- 6. (a)
- 7. (d)
- 8. (b)
- 9. (d)
- 10. (a)
- 11. (c)
- 12. (d)
- 13. (d)
- 14. (b)
- 15. (c)
- 16. (a)
- 17. (c)
- 18. (b)

2. Sistemas de Numeração, Operações e Códigos

7. Exercícios para Entregar na Próxima Aula

(Manuscrito, Individual)

Exercícios para Entregar na Próxima Aula

- 1. Qual é o peso do dígito 6 em cada um dos seguintes números decimais?
  - (a) 1386 (b) 54.692 (c) 671.920
- 2. Expresse cada um dos seguintes números decimais como uma potência de dez:
  - (a) 10 (b) 100 (c) 10.000 (d) 1.000.000
- 3. Determine o valor de cada dígito nos números decimais a seguir:
  - (a) 471 (b) 9356 (c) 125.000
- 4. Até que valor é possível contar com números decimais de 4 dígitos?

Exercícios para Entregar na Próxima Aula

- 5. Converta para decimal os números binários a seguir:
  - (a) 11 (b) 100 (c) 111 (d) 1000
  - (e) 1001 (f) 1100 (g) 1011 (h) 1111
- 6. Converta os seguintes números binários para decimal:
  - (a) 1110 (b) 1010 (c) 11100 (d) 10000
  - (e) 10101 (f) 11101 (g) 10111 (h) 11111

Exercícios para Entregar na Próxima Aula

- 7. Converta cada número binário a seguir para decimal:
  - (a) 110011,11 (b) 101010,01 (c) 1000001,111
  - (d) 1111000,101 (e) 1011100,10101 (f) 1110001,0001
  - (g) 1011010,1010 (h) 1111111,11111
- 8. Qual o maior número decimal que pode ser representado pelas seguintes quantidades de dígitos binários (bits)?
  - (a) dois (b) três (c) quatro (d) cinco (e) seis
  - (f) sete (g) oito (h) nove (i) dez (j) onze

Exercícios para Entregar na Próxima Aula

- 9. Quantos bits são necessários para representar os seguintes números decimais?
  - (a) 17 (b) 35 (c) 49 (d) 68
  - (e) 81 (f) 114 (g) 132 (h) 205
- 10. Determine a sequência binária para cada sequência decimal a seguir:
  - (a) 0 a 7 (b) 8 a 15 (c) 16 a 31
  - (d) 32 a 63 (e) 64 a 75

Exercícios para Entregar na Próxima Aula

- 11. Converta cada número decimal a seguir para binário usando o método da soma dos pesos:
  - (a) 10 (b) 17 (c) 24 (d) 48
  - (e) 61 (f) 93 (g) 125 (h) 186
- 12. Converta cada fração decimal para binário usando o método da soma dos pesos:
  - (a) 0,32 (b) 0,246 (c) 0,0981
- 13. Converta cada número decimal para binário usando o método da divisão sucessiva por 2.
  - (a) 15 (b) 21 (c) 28 (d) 34
  - (e) 40 (f) 59 (g) 65 (h) 73

Exercícios para Entregar na Próxima Aula

- 14. Converta cada fração decimal para binário usando o método da multiplicação sucessiva por 2:
  - (a) 0,98 (b) 0,347 (c) 0,9028
- 15. Some os seguintes números binários:
  - (a) 11 + 01 (b) 10 + 10 (c) 101 + 11
  - (d) 111 + 110 (e) 1001 + 101 (f) 1101 + 1011
- 16. Use a subtração direta para os seguintes números binários:
  - (a) 11 - 1 (b) 101 - 100 (c) 110 - 101
  - (d) 1110 - 11 (e) 1100 - 1001 (f) 11010 - 10111

Exercícios para Entregar na Próxima Aula

- 17. Realize as seguintes multiplicações binárias:
  - (a) 11 × 11 (b) 100 × 10 (c) 111 × 101
  - (d) 1001 × 110 (e) 1101 × 1101 (f) 1110 × 1101
- 18. Faça a operação de divisão binária conforme indicado:
  - (a) 100 ÷ 10 (b) 1001 ÷ 11 (c) 1100 ÷ 100
- 19. Determine o complemento de 1 de cada número binário:
  - (a) 101 (b) 110 (c) 1010
  - (d) 11010111 (e) 1110101 (f) 00001

Exercícios para Entregar na Próxima Aula

- 20. Determine o complemento de 2 de cada número binário a seguir usando qualquer método:
  - (a) 10 (b) 111 (c) 1001 (d) 1101
  - (e) 11100 (f) 10011 (g) 10110000 (h) 00111101
- 21. Expresse cada número decimal a seguir em um número binário do tipo sinal-magnitude de 8 bits:
  - (a) +29 (b) -85 (c) +100 (d) -123
- 22. Expresse cada número decimal a seguir como um número de 8 bits na forma do complemento de 1:
  - (a) -34 (b) +57 (c) -99 (d) +115

Exercícios para Entregar na Próxima Aula

- 23. Expresse cada número decimal a seguir como um número de 8 bits na forma do complemento de 2:
  - (a) +12 (b) -68 (c) +101 (d) -125
- 24. Determine o valor decimal de cada número binário sinalizado a seguir na forma sinal-magnitude:
  - (a) 10011001 (b) 01110100 (c) 10111111
- 25. Determine o valor decimal de cada número binário sinalizado a seguir na forma do complemento de 1:
  - (a)10011001 (b) 01110100 (c) 10111111

Exercícios para Entregar na Próxima Aula

- 26. Determine o valor decimal de cada número binário sinalizado a seguir na forma do complemento de 2:
  - (a)10011001 (b) 01110100 (c) 10111111
- 27. Expresse cada um dos seguintes números binários no formato de ponto flutuante de precisão simples:
  - (a)0111110000101011 (b) 100110000011000
- 28. Determine os valores dos números em ponto flutuante de precisão simples a seguir:
  - (a) 1 10000001 01001001110001000000000
  - (b) 0 11001100 10000111110100100000000

Exercícios para Entregar na Próxima Aula

- 29. Converta para binário cada número hexadecimal a seguir:
  - (a) 38_{16} (b) 59_{16} (c) $A14_{16}$ (d) $5C8_{16}$
  - (e) 4100_{16} (f) $FB17_{16}$ (g) $8A9D_{16}$
- 30. Converta para hexadecimal cada número binário a seguir:
  - (a) 1110 (b) 10 (c) 10111
  - (d) 10100110 (e) 1111110000 (f) 100110000010
- 31. Converta para decimal cada número hexadecimal a seguir:
  - (a) 23_{16} (b) 92_{16} (c) $1A_{16}$ (d) $8D_{16}$
  - (e) $F3_{16}$ (f) EB_{16} (g) $5C2_{16}$ (h) 700_{16}

Exercícios para Entregar na Próxima Aula

- 32. Converta para hexadecimal cada número decimal a seguir:
  - (a) 8 (b) 14 (c) 33 (d) 52
  - (e) 284 (f) 2890 (g) 4019 (h) 6500
- 33. Realize as seguintes adições:
  - (a) $37_{16} + 29_{16}$ (b) $A0_{16} + 6B_{16}$ (c) $FF_{16} + BB_{16}$
- 34. Realize as seguintes subtrações:
  - (a) $51_{16} - 40_{16}$ (b) $C8_{16} - 3A_{16}$ (c) $FD_{16} - 88_{16}$

Exercícios para Entregar na Próxima Aula

- 35. Converta para decimal cada número octal a seguir:
  - (a) 12_8 (b) 27_8 (c) 56_8 (d) 64_8 (e) 103_8
  - (f) 557_8 (g) 163_8 (h) 1024_8 (i) 7765_8
- 36. Converta para octal cada número decimal a seguir fazendo divisões sucessivas por 8:
  - (a) 15 (b) 27 (c) 46 (d) 70
  - (e) 100 (f) 142 (g) 219 (h) 435

Exercícios para Entregar na Próxima Aula

- 37. Converta para binário cada número octal a seguir:
  - (a) 13_8 (b) 57_8 (c) 101_8 (d) 321_8 (e) 540_8
  - (f) 4653_8 (g) 13271_8 (h) 45600_8 (i) 100213_8
- 38. Converta para octal cada número binário a seguir:
  - (a) 111 (b) 10 (c) 110111 (d) 101010
  - (e) 1100 (f) 1011110 (g) 101100011001
  - (h) 10110000011 (i) 111111101111000

Exercícios para Entregar na Próxima Aula

- 39. Converta para BCD 8421 cada um dos seguintes números decimais:
  - (a) 10 (b) 13 (c) 18 (d) 21 (e) 25 (f) 36
  - (g) 44 (h) 57 (i) 69 (j) 98 (k) 125 (l) 156
- 40. Converta para binário direto cada um dos números do Problema 39 e compare o número de bits necessários nesses dois problemas.
- 41. Converta para BCD os seguintes números decimais:
  - (a) 104 (b) 128 (c) 132 (d) 150 (e) 186
  - (f) 210 (g) 359 (h) 547 (i) 1051

Exercícios para Entregar na Próxima Aula

- 42. Converta para decimal os números BCD a seguir:
  - (a) 0001 (b) 0110 (c) 1001
  - (d) 00011000 (e) 00011001 (f) 00110010
  - (g) 01000101 (h) 10011000 (i) 100001110000
- 43. Converta para decimal cada um dos números BCD a seguir:
  - (a) 10000000 (b) 001000110111 (c) 001101000110
  - (d) 010000100001 (e) 011101010100 (f) 100000000000
  - (g) 100101111000 (h) 0001011010000011
  - (i) 1001000000011000 (j) 0110011001100111

Exercícios para Entregar na Próxima Aula

- 44. Some os seguintes números BCD:
  - (a) 0010 + 0001 (b) 0101 + 0011 (c) 0111 + 0010
  - (d) 1000 + 0001 (e) 00011000 + 00010001
  - (f) 01100100 + 00110011 (g) 01000000 + 01000111
  - (h) 10000101 + 00010011
- 45. Some os seguintes números BCD:
  - (a) 1000 + 0110 (b) 0111 + 0101 (c) 1001 + 1000
  - (d) 1001 + 0111 (e) 00100101 + 00100111
  - (f) 01010001 + 01011000 (g) 10011000 + 10010111
  - (h) 010101100001 + 011100001000

Exercícios para Entregar na Próxima Aula

- 46. Converta para BCD cada par de números decimais e faça a soma conforme indicado:
  - (a) 4 + 3 (b) 5 + 2 (c) 6 + 4 (d) 17 + 12
  - (e) 28 + 23 (f) 65 + 58 (g) 113 + 101 (h) 295 + 157
- 47. Determine qual dos seguintes códigos com paridade par apresenta erro:
  - (a) 100110010 (b) 011101010 (c) 10111111010001010
- 48. Determine qual dos seguintes códigos com paridade ímpar apresenta erro:
  - (a) 11110110 (b) 00110001 (c) 01010101010101010

Exercícios para Entregar na Próxima Aula

- 49. Acrescente um bit de paridade par aos seguintes bytes de dados:
  - (a) 10100100 (b) 00001001 (c) 11111110
- 50. Acrescente um bit de paridade ímpar aos seguintes bytes de dados:
  - (a) 10100100 (b) 00001001 (c) 11111110