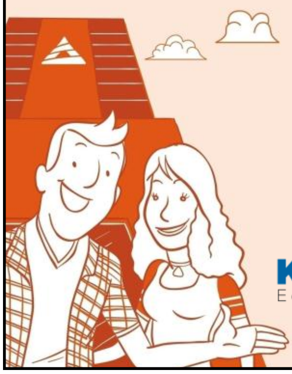

Sistemas Microprogramados

Aula 01
Unidade Jundiaí

Kroton Educacional
Anhanguera
Aqui o seu esforço ganha força.

Plano de Ensino

- **Sistemas de Numeração**
- Arquitetura de Computadores
- Linguagem de Máquina
- Microcontroladores

Livro-Texto

- Livro-Texto:
  - » PEREIRA, Fabio. Microcontroladores PIC - Técnicas avançadas. 4ª ed. São Paulo: Erica, 2006.
- Bibliografia Complementar:
  - » GIMENEZ, S.P.. Microcontroladores 8051. 2ª ed. São Paulo: Pearson Education, 2005.

1. Sistemas de Numeração

- Como simplificação, uma base numérica é um conjunto de símbolos (ou algarismos) com os quais podemos representar uma quantidade ou um número.
- A base decimal (base 10) é a mais difundida e é composta por 10 números: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Para expressarmos números maiores que 9, devemos somar um dígito ao número original, levando-se então a: 9 + 1 = 10, 99 + 1 = 100, 999 + 1 = 1000 e assim por diante.

1. Sistemas de Numeração

- Tal seqüência repete-se indefinidamente, seguindo o padrão, podemos representar os números como uma seqüência de base^n que, neste caso equivale a 10^n.

Posição do Dígito	4	3	2	1	0
Peso	10^4=10000	10^3=1000	10^2=100	10^1=10	10^0=1

- Por exemplo:
  - » O número **1735** seria representado por: $1x10^3 + 7x10^2 + 3x10^1 + 5x10^0 = 1000 + 700 + 30 + 5 = 1735$

1. Sistemas de Numeração

- Com base nesta premissa, podemos representar uma seqüência numérica de base decimal como: $\sum_{i=0}^{n-1} x_i B^i$
- Outras bases possuem representação de seus símbolos similar à base decimal, ou seja, com a sequenciação de símbolos, obedecendo a seus limites de dígitos.

1. Sistemas de Numeração

- Base = 10 (decimal) ➔ 0_{10}, 1_{10}, 2_{10}, 3_{10}, 4_{10}, 5_{10}, 6_{10}, 7_{10}, 8_{10}, 9_{10}, 10_{10}, 11_{10}, 12_{10}, 13_{10}, 14_{10}, 15_{10}, 16_{10}, 17_{10}, …
- Base = 8 (octal) ➔ 0_8, 1_8, 2_8, 3_8, 4_8, 5_8, 6_8, 7_8, 10_8, 11_8, 12_8, 13_8, 14_8, 15_8, 16_8, 17_8, 20_8, 21_8, 22_8, 23_8, ...
- Base = 16 (hexadecimal) ➔ 0_{16}, 1_{16}, 2_{16}, 3_{16}, 4_{16}, 5_{16}, 6_{16}, 7_{16}, 8_{16}, 9_{16}, A_{16}, B_{16}, C_{16}, D_{16}, E_{16}, F_{16}, 10_{16}, 11_{16}, 12_{16}, 13_{16}, 14_{16}, 15_{16}, 16_{16}, 17_{16}, 18_{16}, 19_{16}, $1A_{16}$, $1B_{16}$, $1C_{16}$, $1D_{16}$, ...
- Base = 2 (binária) ➔ 0_2, 1_2, 10_2, 11_2, 100_2, 101_2, 110_2, 111_2, 1000_2, ...

1. Sistemas de Numeração – Conversão

- Para encontrar a correspondência entre um número de base cinco e um número na base decimal, basta computarmos o somatório dos pesos de cada um dos algarismos em relação à base 10.

$$N = A_nB^n + ... + A_3B^3 + A_2B^2 + A_1B^1 + A_0B^0$$

  - Por exemplo, convertendo o número **34_5** para a base decimal:

n^1	n^0	Conversão
3	4	**$=3x5^1 + 4x5^0 = 15 + 4 = 19$**

1. Sistemas de Numeração – Conversão

- Para conversão de um número em qualquer base para a base decimal pode ser feita utilizando uma operação de divisão inteira com resto:

$$N/B = A_nB^{n-1} + ... + A_3B^2 + A_2B^1 + A_1B^0 + A_0B^{-1}$$

  - Por exemplo, convertendo o número **53** para a base binária:

Número	Resultado	Resto	Binário
53 / 2	26	1	1
26 / 2	13	0	01
13 / 2	6	1	101
6 / 2	3	0	0101
3 / 2	1	1	10101
1 / 2	0	1	110101

  - Desta forma o número decimal 53 equivale a **110101_2**.

1. Sistemas de Numeração – Conversão

- Números com fração devemos multiplicar a fração pela base até obtermos a precisão deseja.
  - » Por exemplo, convertendo o número **0,828125** para a base binária:

Número	Resultado	Inteiro	Binário
0,828125 x 2	1,65625	1	0,1
0,65625 x 2	1,3125	1	0,11
0,3125 x 2	0,625	0	0,110
0,625 x 2	1,25	1	0,1101
0,25 x 2	0,5	0	0,11010
0,5 x 2	1	1	0,110101

  - » Desta forma o número decimal 0,828125 equivale a $\mathbf{0,110101_2}$.

1. Sistemas de Numeração – Conversão

- O projeto de sistemas digitais envolve tipicamente o uso de circuitos integrados formados por componentes semicondutores, os quais são polarizados de forma a se comportar como “chaves digitais”.
- Estas chaves podem assumir tipicamente dois valores:
  - » “aberto”, permitindo a passagem de corrente elétrica, e
  - » “fechado”, impedindo a sua passagem.
- Fazendo a correspondência entre estes estados e os algarimos 0 e 1 (0 para “chave aberta” e 1 para “chave fechada”), pode-se exprimir o estado de um conjunto de chaves digitais na forma de um número de base 2 (número binário).

1. Sistemas de Numeração – Conversão

- Para um número binário qualquer de N dígitos, o menor valor representável é 0 e o maior valor representável é igual a $2^N – 1$. Assim, um número de um byte (ou oito bits) pode assumir qualquer valor entre 0 e $2^8 – 1$, ou seja, entre 0 e 255.
- Para bases equivalentes a 2^N ($2^1 = 2$ binária, $2^3 = 8$ octal, $2^4 = 16$ hexadecimal), pode-se fazer a equivalência de forma direta.
  - » Assim, o número $\mathbf{2BD_{16}}$ é equivalente a $\mathbf{1010111101_2}$.

2	B	D
0010	1011	1101

1. Sistemas de Numeração – Sinal

- No sistema decimal, o símbolo '–' é usado para indicar números negativos e '+' (ou simplesmente um espaço vazio) para números positivos.
- Para permitir números com sinal na base, utiliza-se um dígito (normalmente o mais significativo) para representá-lo.
- Com isto ganha-se a possibilidade de representar inteiros negativos, mas a faixa de representação é reduzida porque tem-se agora somente (n–1) dígitos para representar a magnitude.
  - » Por exemplo, para decimal com dois dígitos: tem-se a faixa de 0 até 99; com a representação em sinal magnitude obtém-se a faixa de –9 até +9 e duas representações para o zero: –0 e +0.

1. Sistemas de Numeração – Sinal

- De um modo geral, para uma base qualquer, das B^n combinações possíveis usam-se somente $2*B^{n-1}-1$ (descontando-se o duplo zero).
  - » No exemplo acima, das 100 combinações são utilizadas somente 19, ou seja, $2*10^{2-1}-1 = 20-1 = 19$.

1. Sistemas de Numeração – Valor Numérico

- Um número em sinal magnitude, independente de qual a base utilizada, é formado por duas parcelas, escritas lado a lado. A parcela à esquerda (S(a)) representa o sinal e a parcela à direita (M(a)) a magnitude:

a = S(a)M(a)

Número	Binário
+3	011
+2	010
+1	001
+0	000
-0	100
-1	101
-2	110
-3	111

1. Sistemas de Numeração – Valor Numérico

- Para base 3 com três dígitos, tem-se 17 representações. A metade é 8.5 (8 arredondando-se); assim os números de 0 a 7 são positivos, e os de 9 a 16 são negativos:
  - » $2*B^{n-1}-1 = 2*3^{3-1}-1 = 2*3^{2}-1 = 18-1 = 17$ representações

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
+22	+21	+20	+12	+11	+10	+02	+01	+00	-01	-02	-10	-11	-12	-20	-21	-22

- Para base 4 com três dígitos, tem-se 31 combinações. A metade é 15.5 (15 arredondando-se); então os números de 0 a 14 são positivos, e os de 16 a 30 são negativos:
  - » $2*B^{n-1}-1 = 2*4^{3-1}-1 = 2*4^{2}-1 = 32-1 = 31$ representações

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
+33	+32	+31	+30	+23	+22	+21	+20	+13	+12	+11	+10	+03	+02	+01	+00

...	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
...	-01	-02	-03	-10	-11	-12	-13	-20	-21	-22	-23	-30	-31	-32	-33

1. Sistemas de Numeração – Valor Numérico

- Para base 2 com quatro dígitos, tem-se 15 combinações. A metade é 7.5 (7 arredondando-se); assim os números de 0 a 6 são positivos, e os de 7 a 14 são negativos:
  - » $2*B^{n-1}-1 = 2*2^{4-1}-1 = 2*3^{3}-1 = 16-1 = 15$ representações

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0111	**0**110	**0**101	**0**100	**0**011	**0**010	**0**001	0000	**1**001	**1**010	**1**011	**1**100	**1**101	**1**110	**1**111

S(a)	S(c)	S(d)	M(d)	Exemplo
+	+	+	M(a) + M(c)	5 + 7 = 12
–	–	–	M(a) + M(c)	-5 + -7 = -12
+	–	se M(a)≥M(c),+	M(a) – M(c)	7 + -5 = 2
		se M(a)>M(c),–	M(c) – M(a)	5 + -7 = -2
–	+	se M(a)<M(c),–	M(a) – M(c)	-7 + 5 = -2
		se M(a)≤M(c),+	M(c) – M(a)	-5 + 7 = 2

1. Sistemas de Numeração – Complemento B-1

- Para permitir que a operação de soma seja realizada de forma única, sem preocupação com os sinais dos operandos, é utilizada a representação em complemento.
- Números positivos são representados na forma normal, e números negativos são representados em complemento.

1. Sistemas de Numeração – Complemento B-1

- B = base;
- Para números binários B-1=1 (complemento de 1);
- Complemento de 1: A'=not(A)
  - » Ex.: A = 0001 (1 decimal)
  - » A' (complemento de 1 de A) = 1110 (inverte-se os bits).

Sem complemento

+7	+6	+5	+4	+3	+2	+1	0	-1	-2	-3	-4	-5	-6	-7
0111	0110	0101	0100	0011	0010	0001	0000	1001	1010	1011	1100	1101	1110	1111

Com complemento B-1

+7	+6	+5	+4	+3	+2	+1	0	-1	-2	-3	-4	-5	-6	-7
0111	0110	0101	0100	0011	0010	0001	0000	1110	1101	1100	1011	1010	1001	1000

1. Sistemas de Numeração – Complemento B-1

- Exemplo 1: 3 – 2 = 3 + (-2)
  - » 3 = 0011 e –2 = 1010
  - » Complemento de 1 para -2 = 101

011	Como 3 > -2 então sinal = 0 (+)
+101	0001

1000 ➔ 000+1=001	**Volta ➔ 0001**

- Exemplo 2: 5 – 7 = 5 + (-7)
  - » 5 = 0101 e –7 = 1111
  - » Complemento de 1 para -7 = 000

101	Como 5 > -7 então sinal = 1 (-)
+000	1101
----------	
101 ➔ 101	**Volta ➔ 1010**

1. Sistemas de Numeração – Complemento B-1

- Exemplo 3: 21 – 9 = 21 + (-9)
  - » 21 = 010101 e –9 = 11001 ou 101001 (mesmo nº de casas)
  - » Complemento de 1 para -9 = 10110

10101	Como 21 > -9 então sinal = 0 (+)
+10110	001100

101011➔ 01011+1=01100	**Volta ➔ 001100**

1. Sistemas de Numeração – Subtração

- A operação de subtração, seja qual for o método de representação utilizado, pode ser facilmente realizada transformando-a em uma soma:

$$d = a - c = a + (-c)$$

- Assim, para realizar subtrações, pode-se simplesmente trocar o sinal do subtraendo e somá-lo ao minuendo. A troca de sinal e a soma seriam então realizadas de acordo com o sistema de representação utilizado.

1. Sistemas de Numeração – Estouro

Sinal de A	Sinal de B	Sinal de D (obtido)	Sinal de D (real)	Estouro?
+	+	+	+	Não
+	+	–	+	Sim
–	–	–	–	Não
–	–	+	–	Sim
+	–	+/–	+/–	Nunca
–	+	–/+	–/+	Nunca

- Se dois operandos tiverem sinais diferentes nunca ocorre estouro.
- Se os dois operandos tiverem mesmo sinal e o resultado da soma tiver sinal diferente dos mesmos, ocorreu estouro e o resultado é incorreto.

1. Sistemas de Numeração – Estouro

- Ex.: Números de 4 bits:
  - » 1111+0001=1110 ➔ -7+1=-6
    (correto: sem estouro e sem vai um)
  - » 1111+1001=0001 ➔ -8+(-1)=1
    (errado, deveria ser –9; estouro: estouro com vai um)
  - » 0111+1001 = 0110 ➔ 7+(-1) = 6
    (correto – sem estouro com vai 1)
  - » 0111+0011= 1010 ➔ 7+3=-2
    (errado, deveria ser 10; sem estouro mais com vai um; bit mais significativo é número negativo)

Anhanguera

Aqui o seu esforço ganha força.

Sistemas Microprogramados – Aula 01

Ciência da Computação

clayton.valdo@anhanguera.com