Resumo para a prova 1 ICC

Aluna: Anna Gabriele Marques de Oliveira

Contents

1 Lista 1

1.1 Base N para base 10

$123,4_7 = 1*7^2 + 2*7^1 + 3*7^0 + 4*7^{-1} = 66,571428571_{10}$

1.2 Base 2 para base 10

101101_2 :

1 0 1 1 0 1
128 64 32 16 8 4 2 1 pesos

$32 + 8 + 4 + 1 = 45_{10}$

lembrete: Sistema Hexadecimal: A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.

2 Lista 2

2.1 Base 10 para base N

$86_{10} = x_7$=

86/7, Q = 12, r= 2
12/7, Q = 1, r = 5
1/7, Q = 0, r = 1

$= 152_7$

2.1.1 Com vírgula

$23,7_{10} = x_2$:

23/2, Q = 11, R = 1
11/2, Q = 5, R = 1
5/2, Q= 2, R=1 De baixo pra cima 10111
2/2, Q = 1, R= 0
1/2, Q = 0, R=1

0,7 *2 = 1,4
0,4*2 = 0,8
0,8*2 = 1,6 De cima para baixo 10110
0,6*2 = 1,2
0,2*2 = 0,4

= 10111,10110.

2.2 Base 16 para Base 2

por pesos 8421:

37C:
3 = 0011
7 = 0111
C = 1100

2.3 Base 2 para base 8/ Base 2 para Base 16

Base 2 para base 8: (De base 2 p base 16 divide de 4 em 4)

101,01101:

101, = 5,
011 = 3
010 = 2 $= 5,32_8$

2.4 Base 16 para Base 8

Base 16 para base 8:

$79,328_{16}$ =

7 = 0111
9, = 1001,
3 = 0011
2 = 0010
8 = 0100

= 01111001,001100100100.

$\Downarrow$

= 001 111 001, 001 100 100 100
= 1 7 1, 1 4 4 4

=$171,1444_8$

3 Lista 3

3.1 Soma, subtração e multiplicação de binários sem sinal

3.1.1 Soma:

$153_{16} + 297_3 =$

153_{16} por pesos 8421:

$$\begin{array}{ccc} 1 & 5 & 3 \\ \downarrow & \downarrow & \downarrow \\ 0001 & 0101 & 0011 \end{array} \quad = 000101010011$$

$297_3 = 2 * 3^2 + 9 * 3^1 + 7 * 3^0 = 52_{10}$

$52_{10} = 110100$ por pesos 8421.

$$\begin{array}{rr} & \overset{}{0\,0\,0\,1\,0\,1\,\overset{1}{0}\,1\,0\,0\,1\,1} \\ + & \\ & 1\,1\,0\,1\,0\,0 \\ \hline & 1\,0\,1\,0\,1\,0\,1\,1\,1 \end{array} \qquad = 101010111_2 = 1 + 2 + 4 + 16 + 64 + 256 = 343_{10}$$

3.1.2 Subtração:

$111001_2 - 10011_2 =$

$$\begin{array}{rcccccc} & 1 & 1 & \overset{0}{\cancel{1}} & \overset{\overset{1}{\cancel{2}}}{\cancel{0}} & \overset{2}{\cancel{0}} & 1 \\ - & & & & & & \\ & & 1 & 0 & 0 & 1 & 1 \\ \hline & 1 & 0 & 0 & 1 & 1 & 0 \end{array} \qquad = 100110_2 = 2 + 4 + 32 = 38_{10}$$

3.1.3 Multiplicação:

$1101_2 \times 1011_2 =$

$$\begin{array}{rr} & 1\,1\,0\,1 \\ \times & \\ & 1\,0\,1\,1 \\ \hline & 1\,1\,0\,1 \\ & 1\,1\,0\,1 \\ + & 0\,0\,0\,0 \\ & 1\,1\,0\,1 \\ \hline & 1\,0\,0\,0\,1\,1\,1\,1 \end{array}$$

4 Lista 4

4.1 representação de binários com sinal

4.1.1 Sinal magnitude

Positivo: É o número em binário na quantidade de casas desejadas.

Negativo: Possui o primeiro dígito de paridade 1.

Ex: -39_{10} em 8 bits

Em binário 8 bits = 00100111

Em Sinal Magnitude = 10100111

Sinal magnitude compreende o intervalo $-127 \leq x \leq +127$ na representação em 8 bits.

Assim, a representação do -128, por exemplo, compreende mais de 8 bits.

A faixa de representação em uma quantidade determinada de bits sera 2^n. Assim, com 8 bits podemos formar 256 combinações de números.

O intervalo pode ser calculado pela fórmula:

$$-[2^{N-1} - 1] \leq x \leq [2^{N-1} - 1]$$

Tal que N é o número de bits requerido.

4.1.2 Complemento de 1

Positivo: É o número em binário na quantidade de casas desejadas.

Negativo: É a representação de SM com todos os bits trocados. 1 vira 0 e 0 vira 1, a não ser pelo dígito de paridade.

Ex: -39_{10} em 8 bits

Em Sinal Magnitude = 10100111

Em Complemento de 1 = 11011000

Complemento de 1 compreende o intervalo $-127 \leq x \leq +127$ na representação em 8 bits.

Assim, a representação do -128, por exemplo, compreende mais de 8 bits.

O intervalo utiliza da mesma fórmula que Sinal Magnitude:

$$-[2^{N-1} - 1] \leq x \leq [2^{N-1} - 1]$$

Tal que N é o número de bits requerido.

4.1.3 Complemento de 2

Positivo: É o número em binário na quantidade de casas desejadas.

Negativo: É a representação de C1 adicionando 1 .

Ex: -39_{10} em 8 bits

Em Complemento de 1 = 11011000

Em Complemento de 2 = 11011001

Complemento de 2 compreende o intervalo $-128 \leq x \leq +127$ na representação em 8 bits.

O intervalo utiliza da fórmula:

$$-[2^{N-1}] \leq x \leq [2^{N-1} - 1]$$

Tal que N é o número de bits requerido.

4.2 aritimética com sinal

Ex: $(-79_{10}) + 15_{10}$

$79_{10} =$
01001111 em binário 8 bits.
= 01001111 em SM
= 10110000 em C1
= 10110001 em C2

$15_{10} =$
00001111 em binário 8 bits.

Como a representação em SM, C1 e C2 para os dígitos positivos é a mesma, 00001111 já está em C2.

$$\begin{array}{r} 1\,0\,1\,1\,0\,0\,0\,1 \\ + \\ 0\,0\,0\,0\,1\,1\,1\,1 \\ \hline 1\,1\,0\,0\,0\,0\,0\,0 \end{array}$$

= 1100000.

5 Lista 5

5.1 Codificação binária

5.1.1 BCD

Utiliza de até 16 algarismos.

É calculado por pesos 8421.

Ex: $537_{10} =$

$5 = 0101$

$3 = 0011$

$7 = 0111 \qquad\qquad = 0101\ 0011\ 0111_{BCD}.$

5.1.2 Excesso de 3

Utiliza o mesmo cálculo de BCD, mas subtrai 3.

De Excesso de 3 à decimal:

$0111\ 0101_{ecx3} =$

$$0111 \swarrow \qquad\qquad 0101 \searrow$$

$$(0 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1) - 3 + (0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1) - 3 =$$

$$= (4 + 2 + 1) - 3 + (4 + 1) - 3 = 42_{10}$$

De decimal à excesso de 3:

$617_{10} =$

$6^{+3} = 9 = 1001$ (por pesos 8421)

$1^{+3} = 4 = 0100$ (por pesos 8421)

$7^{+3} = 10 = 1010$ (por pesos 8421) $\qquad = 1001\ 0100\ 1010_{exc3}.$

5.1.3 Código Gray

Usado para medidas analógicas (temperatura, pressão).

Aumenta de 1 bit em 1 bit.

Para converter para Gray, o número deve estar em binário anteriormente. Conversão $Decimal$ →Binário→ $Gray(4bits)$:

$473_{10} =$

256	128	64	32	16	8	4	2	1
1	1	1	0	1	1	0	0	1

Por pesos: 111011001_2

O bit de mais alta ordem é repetido, e os outros são somados em módulo de 2 com o bit da direita. Todo 'vai um' será desconsiderado.

$$1 \xrightarrow{+} 1 \xrightarrow{+} 1 \xrightarrow{+} 0 \xrightarrow{+} 1 \xrightarrow{+} 1 \xrightarrow{+} 0 \xrightarrow{+} 0 \xrightarrow{+} 1$$
$$\searrow \quad \searrow \quad \searrow \quad \searrow \quad \searrow \quad \searrow \quad \searrow \quad \searrow$$
$$1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1$$

$= 100110101_{gray}$

5.1.4 Código de Hamming

É um código de detecção e correção de erros.

Possui seus bits de paridade nas posições 2^n. As posições são contadas da esquerda para a direita.

Por exemplo, em 7 bits temos as posições:
2^0 = Posição 1,
2^1 = Posição 2,
2^2 = Posição 4.

O bit de posição 1, denominado como bit A, fará as verificações dos bits: 1, 3, 5 e 7.
O bit de posição 2, denominado como bit B, fará as verificações dos bits: 2, 3, 6 e 7.
O bit de posição 4, denominado como bit C, fará as verificações dos bits: 4, 5, 6 e 7.

Os bits verificados serão somados. Se o resultado for par, o resultado retorna 0; se for ímpar, retorna 1.

Se a combinação CBA der 000 o código recebido está correto.

Do contrário, a combinação CBA transformada em base 10 irá indicar a posição do bit a ser trocado.

Ex: 1101101:

A	B	3	C	5	6	7
1	1	0	1	1	0	1

P(1) $= 1 + 0 + 1 + 1 \rightarrow A = 1$
P(2) $= 1 + 0 + 0 + 1 \rightarrow B = 0$
P(4) $= 1 + 1 + 0 + 1 \rightarrow C = 1$
CBA $= 101 = 5_{10}$

O bit 5 está incorreto. Novo código: 1101001.

A	B	3	C	5	6	7
1	1	0	1	0	0	1

P(1) $= 1 + 0 + 0 + 1 \rightarrow A = 0$
P(2) $= 1 + 0 + 0 + 1 \rightarrow B = 0$
P(4) $= 1 + 0 + 0 + 1 \rightarrow C = 0$
CBA = 000 Código correto.