Universidade Tecnológica Federal do Paraná – UTFPR
Bacharelado em Ciência da Computação

BCC32B – Elementos de Lógica Digital

Prof. Rodrigo Hübner

Aula 01 – Sistemas de Numeração

Sistemas de Numeração

Motivação:

- **Dispositivos que operam com diferentes sistemas de numeração. Ex: *displays* BCD, simuladores, calculadoras**

- **Circuitos Digitais usam 2 estados para representar uma informação. Ex: Circuito Base ⇒ Transistor**

- **Números binários podem ser muito extensos ⇒ Difíceis de representar ⇒ Usa base com menos algarismos**

Formato Geral

- Generalizando: dado uma base **b**, qualquer quantidade **N** pode ser representada com um número tal como segue:

$$N_b = a_n \cdot b^n + a_n - 1 \cdot b^n - 1 + ... + a_2 \cdot b^2 + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + ... + a_{-n} \cdot b^{-n}$$

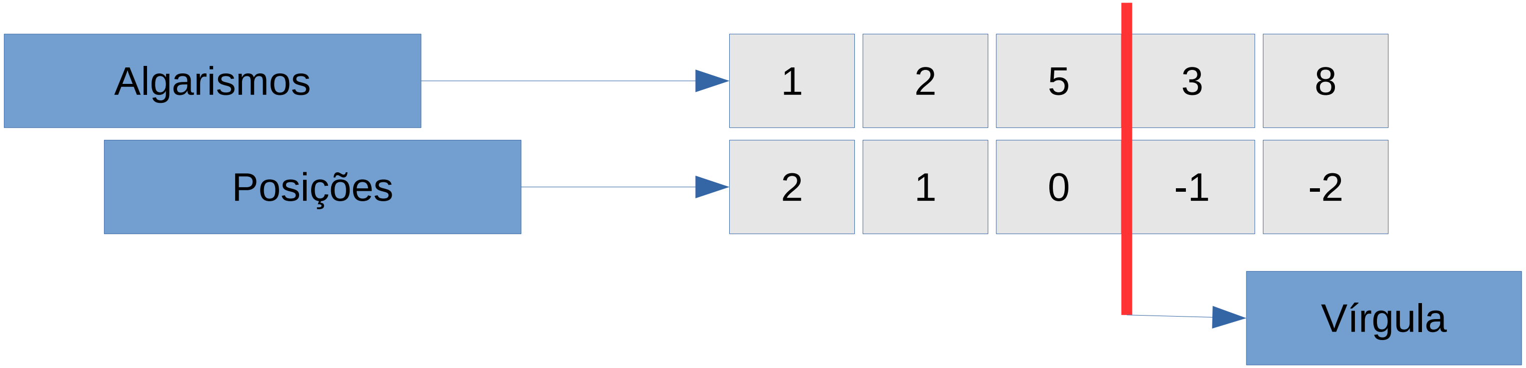
$$N_b = \sum_{i=n}^{-n} a_i \cdot b^i$$

Onde *a* é um algarismo válido para a base *b* e *n* é a posição que o algarismo ocupa na formação do número.

- **Parte inteira:** $a_n \cdot b^n + a_n - 1 \cdot b^n - 1 + ... + a_2 \cdot b^2 + a_1 \cdot b^1 + a_0 \cdot b^0$
- **Parte fracionária:** $a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + ... + a_{-n} \cdot b^{-n}$

Formato Geral

- Exemplo: 125,38 (base 10)



$$1\times10^{2}+2\times10^{1}+5\times10^{0}+3\times10^{-1}+8\times10^{-2}$$
$$1\times100+2\times10+5\times1+3\times0{,}1+8\times0{,}01$$

$$100+20+5+0{,}3+0{,}08$$

$$125+0{,}38$$

$$125{,}38$$

Sistemas de Numeração

<u>Base:</u>

- **É a quantidade de algarismos ou símbolos disponíveis para representar todos os números no sistema de numeração. 0 a (base -1)**
- **Exemplos:**
  - **Base 10 ⇒ 10 dígitos: 0, 1, 2, …, 9**
  - **Base 2 ⇒ 2 dígitos: 0 e 1**
  - **Base 8 ⇒ 8 dígitos: 0, 1, 2, …, 7**
  - **Base 16 ⇒ 16 dígitos: 0, 1, 2, …, 9, A, B, C, D, E, F**

<u>Convenção:</u> Bases maiores que 10 usam letras para representar algarismos maiores que 9

Sistemas de Numeração

Sistema Decimal

Base 10:

Base 10 ⇒ 10 dígitos: 0,1,2,...9

- **Exemplo:**

$$\mathbf{1303}_{10}$$

$$1\text{x}10^3 + 3\text{x}10^2 + 0\text{x}10^1 + 3\text{x}10^0$$

$$1000 + 300 + 0 + 3 = 1303$$

Notação Posicional

Sistemas de Numeração

<u>Sistema Binário</u>

<u>Base 2:</u>

Base 2 ⇒ 2 dígitos: 0 e 1 cada dígito é chamado de bit (<u>bi</u>nary digi<u>t</u>)

- **Convenção:**
  - 1 dígito: bit
  - 4 dígitos: nibble
  - 8 dígitos: byte
- **Exemplo: 00101111**

Sistemas de Numeração

Sistema Binário

Conversões de Bases:

Binário para Decimal

- **Exemplo:**

$$101111_2$$

$$1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$32 + 0 + 8 + 4 + 2 + 1 = 47$$

Sistemas de Numeração

<u>Sistema Binário</u>

<u>Conversões de Bases:</u>

Decimal para Binário

- Parte Inteira:

- 2 Métodos: soma de potências e divisões sucessivas
- Exemplo de <u>Soma de Potências</u>:

$$47 = 32 + 0 + 8 + 4 + 2 + 1$$

$$1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0$$

$$101111_2$$

Sistemas de Numeração

Sistema Binário

Conversões de Bases:

Decimal para Binário

- **Exemplo de Divisões Sucessivas:**

47	2				
1	23	2			
	1	11	2		
		1	5	2	
			1	2	2
				0	1

$\boxed{47_{10} = 101111_2}$

Parte Fracionária

- Se o número for fracionário, a conversão se fará em duas etapas distintas:
  - Conversão da parte inteira: como já vimos.
  - Conversão da parte fracionária.

Parte Fracionária

- **O algoritmo para a parte fracionária:**
  - Consiste em uma série de multiplicações sucessivas;
  - A parte inteira do resultado da primeira multiplicação será o valor da primeira casa fracionária e a parte fracionária será de novo multiplicada pela base
  - Assim por diante até:
    - O resultado dar zero ou
    - Encontrarmos o número de casas decimais desejado.

Parte Fracionária

- **Por exemplo, $15{,}65_{10}$ para a base 2, com 5 dígitos e com 10 dígitos de precisão:**
- **Parte inteira se processa normalmente: $15_{10} = 1111_2$**
- **Parte fracionária:**

Com 5 dígitos	**Ampliando para 10 dígitos**
0,65 x 2 = 1,3	0,8 x 2 = 1,6
0,30 x 2 = 0,6	0,6 x 2 = 1,2
0,60 x 2 = 1,2	0,2 x 2 = 0,4
0,20 x 2 = 0,4	0,4 x 2 = 0,8
0,40 x 2 = 0,8	0,8 x 2 = 1,6

Precisão 5

$15{,}65_{10} = 1111{,}10100_2$

Precisão 10

$15{,}65_{10} = 1111{,}1010011001_2$

Com 5 dígitos:
0,65 = 0,10100

Com 10 dígitos:
0,65 = 0,1010011001

Parte Fracionária

- **A conversão foi interrompida quando encontramos o número de algarismos da precisão solicitada no enunciado.**
  - Como não encontramos resultado 0 em nenhuma das multiplicações, poderíamos continuar efetuando multiplicações indefinidamente até encontrar (se encontrarmos) resultado zero.
  - O resultado encontrado, no caso de interrompermos por chegarmos no número de dígitos especificado, **é aproximado**.
  - Fazendo a conversão inversa é possível notarmos o efeito da aproximação.

Parte Fracionária

- **$15{,}65_{10}$ = Parte inteira: $1111_2 = 15_{10}$**
- **Parte fracionária:**
  - **Com precisão 5:**

$$0{,}10100_2 = 1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 0x2^{-4} + 0x2^{-5} = 0{,}5 + 0{,}125 = 0{,}625_{10}$$

- **Com precisão 10:**

$$0{,}1010011001_2 = 1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 0x2^{-4} + 0x2^{-5} + 1x2^{-6} + 1x2^{-7} + 0x2^{-8} + 0x2^{-9} + 1x2^{-10}$$
$$0{,}1010011001_2 = 1/2 + 1/8 + 1/64 + 1/128 + 1/1024$$
$$0{,}1010011001_2 = 0{,}5 + 0{,}125 + 0{,}015625 + 0{,}0078125 + 0{,}0009765625 = 0{,}6494140625_{10}$$

- **Quanto maior o número de algarismos forem considerados, melhor será a aproximação.**

Parte Fracionária

- **Por exemplo, $15,96875_{10}$ para a base 2, com 5 dígitos e encontrando zero**
- **Parte inteira se processa normalmente: 15_{10}= 1111_2**
- **Parte fracionária:**

Com 5 dígitos
0,96875 x 2 = 1,9375
0,9375 x 2 = 1,875
0,875 x 2 = 1,75
0,75 x 2 = 1,50
0,50 x 2 = 1,0

Precisão 5
$15,96875_{10} = 1111,11111_2$

Precisão 10
$15,96875_{10} = 1111,1111100000_2$

Com 5 dígitos:
0,96875 = 0,11111

Exercícios

Conversões de Bases

- Converter 1001_2 para decimal
- Converter 400_{10} para binário

Solução dos Exercícios

<u>Conversões de Bases</u>

- **Converter 1001_2 para decimal**

$$1001_2$$

$$1\text{x}2^3 + 0\text{x}2^2 + 0\text{x}2^1 + 1\text{x}2^0$$

$$8 + 0 + 0 + 1 = 9_{10}$$

Solução dos Exercícios

Conversões de Bases

- **Converter 400_{10} para binário**
  - **Método de Divisões Sucessivas:**

```
400 | 2
  0   200 | 2
        0   100 | 2
              0   50 | 2
                   0   25 | 2
                        1   12 | 2
                             0   6 | 2
                                 0   3 | 2
                                     1   1
```

$$\boxed{400_{10} = 110010000_2}$$

Sistemas de Numeração

Sistema Octal

Base 8:

Base 8 ⇒ 8 dígitos: 0,1,2,3,4,5,6,7

Decimal	Octal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20

Sistemas de Numeração

Sistema Octal

Conversões de Bases:

Octal para Decimal

- **Exemplo:**

$$144_8$$

$$1 \times 8^2 + 4 \times 8^1 + 4 \times 8^0$$

$$64 + 32 + 4 = 100_{10}$$

Sistemas de Numeração

Sistema Octal

Conversões de Bases:

Decimal para Octal

- **Exemplo de Divisões Sucessivas:**

92	8		
4	11	8	
	3	1	

$92_{10} = 134_8$

Exercícios

Conversões de Bases

- **Converter 77_8 para decimal**
- **Converter 74_{10} para octal**

Soluções dos Exercícios

Conversões de Bases

- **Converter 77_8 para decimal**

$$77_8$$

$$7 \times 8^1 + 7 \times 8^0$$

$$56 + 7 = 63_{10}$$

Soluções dos Exercícios

Conversões de Bases

- Converter 74_{10} para octal
  - Método de Divisões Sucessivas:

$$
\begin{array}{l|l}
74 & 8 \\ \hline
2 & 9
\end{array}
\qquad
\begin{array}{l|l}
9 & 8 \\ \hline
1 & 1
\end{array}
$$

$$\boxed{74_{10} = 112_8}$$

Sistemas de Numeração

<u>Sistema Octal</u>

<u>Conversões de Bases:</u>

Octal para Binário: Transforma cada algarismo octal no correspondente binário (para cada octal são necessários 3 bits ⇒ $2^3 = 8$ → Base octal)

- **Exemplo:**

$$27_8$$

$$\overbrace{010}\overbrace{111}_2$$

Octal	Binário
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Sistemas de Numeração

Sistema Octal

Conversões de Bases:

Binário para Octal: Processo inverso – agrupa-se 3 bits a partir da direita

- **Exemplo:**

$$\underbrace{110}_{6}\,\underbrace{010}_{2}{}_{2}$$

$$6 \quad 2 = 62_8$$

Exercícios

Conversões de Bases

- Converter 34_8 para binário
- Converter 1010_2 para octal

Soluções dos Exercícios

<u>Conversões de Bases</u>

- Converter 34_8 para binário

$$34_8$$

$$\overbrace{011}\overbrace{100}{}_2$$

Soluções dos Exercícios

<u>Conversões de Bases</u>

- **Converter 1010_2 para octal**

$$\underbrace{1}\underbrace{010}_2$$

$$1 \quad 2 = 12_8$$

Insere 0s

Sistemas de Numeração

Sistema Hexadecimal

Base 16:

Base 16 ⇒ 16 dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Decimal	Hexadecimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10

Sistemas de Numeração

Sistema Hexadecimal

Conversões de Bases:

Hexadecimal para Decimal

- **Exemplo:**

$$3F_{16}$$

$$3\times16^1 + 15\times16^0$$

$$48 + 15 = 63_{10}$$

Sistemas de Numeração

Sistema Hexadecimal

Conversões de Bases:

Decimal para Hexadecimal

- **Exemplo de Divisões Sucessivas:**

$$
\begin{array}{r|l}
1000 & 16 \\ \hline
8 \quad 62 & 16 \\ \hline
14 & 3
\end{array}
$$

$$\boxed{1000_{10} = 3\ \ 14\ \ 8 = 3E8_{16}}$$

Exercícios

Conversões de Bases

- Converter $1C3_{16}$ para decimal
- Converter 134_{10} para hexadecimal

Soluções dos Exercícios

Conversões de Bases

- **Converter $1C3_{16}$ para decimal**

$$1C3_{16}$$

$$1 \times 16^2 + 12 \times 16^1 + 3 \times 16^0$$

$$256 + 192 + 3 = 451_{10}$$

Soluções dos Exercícios

Conversões de Bases

- Converter 134_{10} para hexadecimal

$$\begin{array}{r|l} 134 & 16 \\ \hline 6 & 8 \end{array}$$

$$\boxed{134_{10} = 86_{16}}$$

Sistemas de Numeração

Sistema Hexadecimal

Conversões de Bases:

Hexadecimal para Binário: Transforma cada algarismo hexa no correspondente binário (para cada hexa são necessários 4 bits ⇒ $2^4 = 16$ – Base hexa)

- **Exemplo:**

$C13_{16}$

110000010011_2

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

Sistemas de Numeração

Sistema Hexadecimal

Conversões de Bases:

- **Binário para Hexadecimal: Processo inverso – agrupa-se 4 bits a partir da direita**
- **Exemplo:**

$$\underbrace{1001}\underbrace{1000}_2$$

$$9 \quad 8 = 98_{16}$$

Exercícios

Conversões de Bases

- Converter $1ED_{16}$ para binário
- Converter 1100011_2 para hexadecimal

Soluções dos Exercícios

<u>Conversões de Bases</u>

- Converter $1ED_{16}$ para binário

$$1ED_{16}$$

$$000111101101_2$$

Soluções dos Exercícios

<u>Conversões de Bases</u>

- **Converter 1100011_2 para hexadecimal**

$$\underbrace{110}_{6}\,\underbrace{0011}_{3}{}_2$$

$6 \quad 3 = 63_{16}$

Insere 0

Próxima Aula

- **Representação de dados**