



INSTITUTO FEDERAL
Rio Grande do Sul

Técnico em Informática **Fundamentos da Computação**

Sistemas de Numeração **Conversão entre bases numéricas**

Prof. Ms. Frederico Schardong
Alterações: Prof. Ms. Gabriel Marchesan
Contato: gabriel.marchesan@rolante.ifrs.edu.br

Sistemas de numeração

- Um sistema de numeração é um conjunto finito de símbolos que através de uma lei/regra de formação permite representar qualquer quantidade
- Existem dois tipos:
 - Sistemas de numeração posicionais
 - Sistemas de numeração não-posicionais



Sistemas de numeração não posicional

- Símbolos, independente da posição, representam um valor único
- **Exemplo:** sistema romano de numeração
 - Composto por 7 símbolos (I, V, X, L, C, D, M) capazes de representar uma grande variedade de números, porém não todos (como o zero, por exemplo)
 - Exemplo 1: IV representa 4 e VI representa 6, entretanto I e V sozinhos representam outros números (1 e 5)
 - Exemplo 2: XX representa 20 em decimal. Entretanto, X isoladamente representa 10. Ou seja, XX é $10 + 10$. Entretanto, IX é obtido através de $10 - 1$



Sistemas de numeração posicional

- O valor do dígito em um número depende da sua posição
- Cada dígito tem um “peso”, dependendo do lugar que está:
 - $2011 = 2000 + 10 + 1$
 - $1991 = 1*10^3 + 9*10^2 + 9*10^1 + 1*10^0$
- Os pesos crescem da direita para a esquerda
 - $10,1 = 1*10^1 + 0*10^0 + 1*10^{-1}$
- Estes exemplos são do sistema **decimal**, como fica o **binário**?



Sistemas de numeração computacional

- Apesar de estarmos familiarizados com o sistema decimal, um computador que utilize esse sistema deve representar os 10 possíveis estados de um dígito com 10 tensões diferentes (0V, 1V, 2V, 3V ..., 9V).
- Entretanto, quanto maior o número de interpretações de um dígito, maior a probabilidade de erro, pois requer uma sensibilidade muito alta dos componentes eletrônicos.
- Consequentemente, foi adotado o sistema binário pois cada dígito é representado por apenas 2 valores.
- OBS: é **impossível** produzir um sistema de numeração com apenas 1 valor por dígito.



Sistemas de numeração computacional

- Outras vantagens de utilizar o sistema binário
 - Os computadores podem ser construídos com **transistores**, que naturalmente representam números binários (ligado ou desligado)
 - Álgebra tradicional (soma, subtração, multiplicação, divisão) pode ser implementada com transistores, utilizando **álgebra booleana**



Sistemas de numeração computacional

- Exemplo de binário (base 2), hexadecimal (base 16) e decimal (base 10)

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



Conversão de bases: decimal -> binário

- Crie uma tabela com as potências de 2, começando com 2^0

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

- Escolha o maior número que “cabe” no que está sendo convertido (exemplo: 156)

$$156 - 128 = 28$$

- Adicione o número 1 abaixo do 128 na tabela:

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	1							

Conversão de bases: decimal -> binário

- Repita o processo

$$28 - 16 = 12$$

- Adicione o número 1 abaixo do 16 na tabela:

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	1			1				

$$12 - 8 = 4$$

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	1			1	1			



Conversão de bases: decimal -> binário

$$4 - 4 = 0$$

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	1			1	1	1		

- Adicione o número 0 as posições que sobraram

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	1	1	1	0	0

- O número 156 em decimal é igual a 10011100 em binário
- Zeros a esquerda não são relevantes



Conversão de bases: decimal -> binário

- Exercício: converter os números 64, 129, 1032 e 31
- Exercício extra classe: converter os números 511, 8321, 89, 673

8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0



Conversão de bases: decimal -> binário

- Exercício: converter os números 64, 129, 1032 e 31
- Exercício extra classe: converter os números 511, 8321, 89, 673

8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

- 64 -> 100 0000
- 129 -> 1000 0001
- 1032 -> 100 0000 1000
- 31 -> 1 1111
- 511 -> 1 1111 1111
- 8321 -> 10 0000 1000 0001
- 89 -> 101 1001
- 673 -> 10 1010 0001



Conversão de bases: binário -> decimal

- Ex: converta o número binário **10011100** para decimal.
- Escreva as potências de 2 e abaixo o número em binário

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

- Escreva as potências de 2 abaixo dos bits 1 e some esses valores

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

- O resultado da soma é o número em decimal: $128 + 16 + 8 + 4 = 156$



Conversão de bases: binário -> decimal

- Exercício: converter os números para decimal:

1 1110

1 0111 0001

0100 0011

1010 1010

11 0011

256	128	64	32	16	8	4	2	1

Conversão de bases: binário -> decimal

- Exercício: converter os números para decimal:

1 1110 -> 30

1 0111 0001 -> 369

0100 0011 -> 67

1010 1010 -> 170

11 0011 -> 51

256	128	64	32	16	8	4	2	1

