Notação indu-arábica

- Notação posicional → permite representar números grandes e realizar operações sobre os números.

- Base 10 = posição do algarismo determina as potências de 10 que irão multiplicar o número denotado pelo algorismo.

Bases

- $496_{10} = 4 \text{ x} 10^2 + 9 \text{ x } 10^1 + 6 \text{ x } 10^0 = 496_{10}$
- $1001_2 = 1 \text{ x } 2^3 + 0 \text{ x } 2^2 + 0 \text{ x } 2^1 + 1 \text{ x } 2^0 = 9_{10}$
- $760_8 = 7 \text{ x } 8^2 + 6 \text{ x } 8^1 + 0 \text{ x } 8^0 = 496_{10}$
- $1F0_{16} = 1 \text{ x } 16^2 + 15 \text{ x } 16^1 + 0 \text{ x } 16^0 = 496_{10}$

- $111110000_2 = 496_{10}$
  (Exercício verificar)

Octal x Binário

$0_8 \leftrightarrow 000_2$

$1_8 \leftrightarrow 001_2$

$2_8 \leftrightarrow 010_2$

$3_8 \leftrightarrow 011_2$

$4_8 \leftrightarrow 100_2$

$5_8 \leftrightarrow 101_2$

$6_8 \leftrightarrow 110_2$

$7_8 \leftrightarrow 111_2$

$\underline{235}_8 \leftrightarrow \underline{010011101}_2$

$$
\begin{aligned}
235_8 &= 2 \text{ x } 8^2 + 3 \text{ x } 8^1 + 5 \text{ x } 8^0 \\
&= 2 \text{ x } 64 + 3 \text{ x } 8 + 5 \text{ x } 1 \\
&= 128 + 24 + 5 \\
&= 157_{10}
\end{aligned}
$$

$157_{10} = 10011101_2$

Hexadecimal x Binário

$0_{16} \leftrightarrow 0000_2$

$1_{16} \leftrightarrow 0001_2$

$2_{16} \leftrightarrow 0010_2$

$3_{16} \leftrightarrow 0011_2$

$4_{16} \leftrightarrow 0100_2$

$5_{16} \leftrightarrow 0101_2$

$6_{16} \leftrightarrow 0110_2$

$7_{16} \leftrightarrow 0111_2$

$8_{16} \leftrightarrow 1000_2$

$9_{16} \leftrightarrow 1001_2$

$A_{16} \leftrightarrow 1010_2$

$B_{16} \leftrightarrow 1011_2$

$C_{16} \leftrightarrow 1100_2$

$D_{16} \leftrightarrow 1101_2$

$E_{16} \leftrightarrow 1110_2$

$F_{16} \leftrightarrow 1111_2$

$\underline{9D}_{16} \leftrightarrow \underline{10011101}_2$

$$
\begin{aligned}
9D_{16} &= 9 \text{ x } 16^1 + 13 \text{ x } 16^0 \\
&= 9 \text{ x } 16 + 13 \text{ x } 1 \\
&= 144 + 13 \\
&= 157_{10}
\end{aligned}
$$

$157_{10} = 10011101_2$

Exercícios (1)

- Determine a representação, no sistema de numeração binário, octal e hexadecimal, de cada um dos seguintes números, escritos na base decimal:

(a) 19

(b) 458

(c) 67

(d) 321

Exercícios (2)

- Determine a representação, no sistema de numeração octal, decimal e hexadecimal, de cada um dos seguintes números, escritos na base binária:

(a) 1110_2

(b) 110110_2

(c) 10101010_2

(d) 11110000_2

Tabela do “Vai um” Adição de binários

vai um anterior	n	m	n+m	vai um próximo
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabela do “Vai um” Subtração de binários

vai um anterior	n	m	n-m	vai um próximo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

Exercícios (3)

- Realize as operações abaixo, sobre números representados no sistema de numeração binário:

(a) $1011_2 + 101_2$

(b) $10100_2 - 1101_2$

(c) $00000001_2 + 01111111_2$

(d) $10000000_2 - 00000001_2$

Números negativos em binário
Sinal-Magnitude

- sinal-magnitude (um bit indica o sinal)
  - Convenção: 1 negativo, 0 positivo

- $1\underline{101}_2 = 1 \text{ x } 2^2 + 0 \text{ x } 2^1 + 1 \text{ x } 2^0 = -5_{10}$
- $0\underline{101}_2 = 1 \text{ x } 2^2 + 0 \text{ x } 2^1 + 1 \text{ x } 2^0 = 5_{10}$

Problema: Duas representações para o valor “ZERO”

- $1\underline{000}_2 = 0 \text{ x } 2^2 + 0 \text{ x } 2^1 + 0 \text{ x } 2^0 = 0_{10}$
- $0\underline{000}_2 = 0 \text{ x } 2^2 + 0 \text{ x } 2^1 + 0 \text{ x } 2^0 = 0_{10}$

Números negativos em binário Complemento de dois

- Seja p um número , $0 < p <= 2^{n-1}$(n bits)
- Complemento de 2 de p (com n bits) é $2^n - p$
  $C^4(5) = 2^4 - 5 = 11$

- Número inteiro positivo
  - representados na base binária, da maneira usual
- Número inteiro negativo
  - representados pelo complemento de 2 do valor absoluto do número

Números negativos em binário Complemento de dois

- Número 5 (Maneira usual)
  $5_{10} = 0101_2$

- Número -5 => Valor absoluto 5

  $C^4(5) = 2^4 - 5 = 11$
  $-5_{10} = 1011_2$

Complemento de dois (4 bits)

$0_{10} \rightarrow 0000_2$	$C^4(-8) \rightarrow 8$	$-8_{10} \rightarrow 1000_2$
$1_{10} \rightarrow 0001_2$	$C^4(-7) \rightarrow 9$	$-7_{10} \rightarrow 1001_2$
$2_{10} \rightarrow 0010_2$	$C^4(-6) \rightarrow 10$	$-6_{10} \rightarrow 1010_2$
$3_{10} \rightarrow 0011_2$	$C^4(-5) \rightarrow 11$	$-5_{10} \rightarrow 1011_2$
$4_{10} \rightarrow 0100_2$	$C^4(-4) \rightarrow 12$	$-4_{10} \rightarrow 1100_2$
$5_{10} \rightarrow 0101_2$	$C^4(-3) \rightarrow 13$	$-3_{10} \rightarrow 1101_2$
$6_{10} \rightarrow 0110_2$	$C^4(-2) \rightarrow 14$	$-2_{10} \rightarrow 1110_2$
$7_{10} \rightarrow 0111_2$	$C^4(-1) \rightarrow 15$	$-1_{10} \rightarrow 1111_2$

Complemento de dois

- O bit mais à esquerda da representação de um número inteiro na notação de complemento de 2 é igual a **1**, se o número for negativo, e igual a **0**, caso contrário.
  - Esse bit pode ser, portanto, interpretado como **o sinal do número.**
  - Podem ser representados com n bits ($-2^{n-1} .. 2^{n-1} - 1$)
  - n = 8 → -128 .. 127 (em Java: tipo byte)
  - n = 16 → -32768 .. 32767 (Java: tipo small)
  - n = 32 → -2147483648 .. 2147483647 (Java: tipo int)
  - n = 64 → $-2^{63} .. 2^{63} - 1$ (Java: tipo long)
    - -2^{63} ↔ -9223372036854775808 (19 casas decimais)
    - $2^{63} - 1$ ↔ 9223372036854775807

Complemento de dois (adição)

- Usando essa representação, a adição de dois números inteiros *n* e *m* pode ser feita da maneira usual, sendo descartado o bit “vai um” obtido mais à esquerda.

0001_2 (1_{10})

$+$

1100_2 (-4_{10})

1101_2 (-3_{10})

1110_2 (-2_{10})

$+$

1101_2 (-3_{10})

1011_2 (-5_{10})

Complemento de dois (subtração)

- Usando essa representação, a subtração de dois números inteiros *n* e *m* pode ser feita da maneira usual, sendo descartado o bit “vai um” obtido mais à esquerda.

0001_2 (1_{10})	1110_2 (-2_{10})
-	-
1100_2 (-4_{10})	1101_2 (-3_{10})
--------	----------
0101_2 (3_{10})	0001_2 (1_{10})

Complemento de dois (overflow/underflow)

- A adição/subtração de números binários (representados em n bits) pode ter como resultado um número que não pode ser representado com apenas n bits.
  - *"Overflow/Underflow" - Resultado inconsistente*
  - "8" e "-9" não podem ser representados com apenas 4 bits

0011_2 (3_{10})	1001_2 (-7_{10})
+	-
0101_2 (5_{10})	0010_2 (2_{10})
--------	---------
1000_2 (-1_{10})	0111_2 (7_{10})

Exercícios (4)

- Determine a representação na notação de complemento de 2, com 8 bits, de cada um dos seguintes números:

(a) 23

(b) 108

(c) -25

(d) -123

Exercícios (5)

- Indique como seria feito o cálculo das seguintes operações, em um computador que utiliza notação de complemento de 2 para representação de números (de tamanho 8 bits):

(a) 57 + (-118)

(b) (-15) + (-46)

(c) 37 - 23

(d) -23 - (-34)