Mudança de Base: decimal, binária, octal, hexadecimal

Universidade Estadual de Mato Grosso do Sul – UEMS
Ciência da Computação
Linguagem de Montagem
Prf Dr Osvaldo Vargas Jaques
ojacques@comp.uems.br

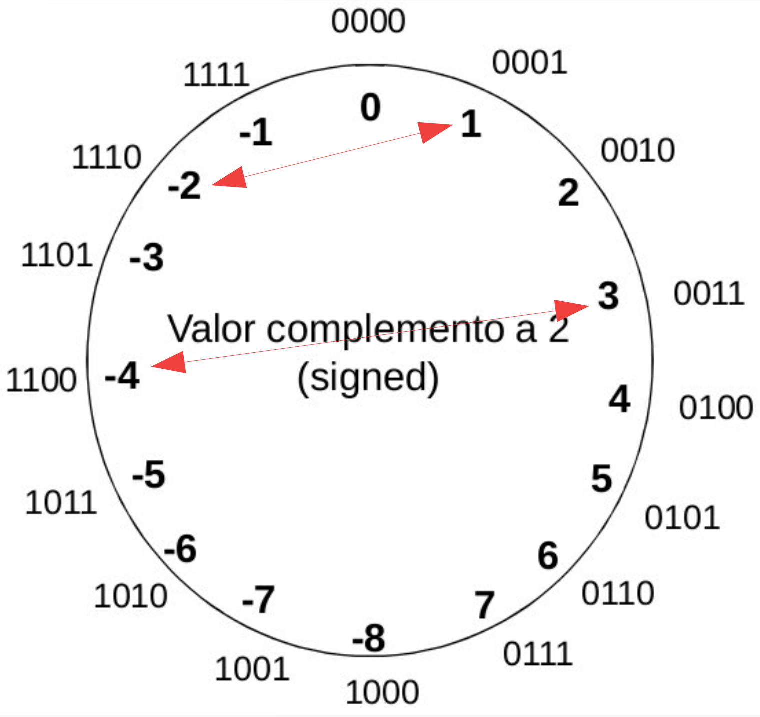
Número com sinal

- Para representar um número com n bits com sinal
  - O bit mais significativo ou primeiro bit será multiplicado por -2^{n-1}. No arquivo 2.1.Complemento2.pdf, ele chama o número de bits de w. E usa a seguinte fórmula:

$$B2T(X) = -x_{w-1} \cdot 2^{w-1} + \sum_{i=0}^{w-2} x_i \cdot 2^i$$

x	1	0	0	1	1	0	0	1
w-1	7	6	5	4	3	2	1	0

  - Se temos w=8, ou seja, 8 bits, isso significa que o valor da sequência acima (**1**001 1001) seria em decimal:
  - $\mathbf{-1}.2^7 + 0.2^6 + 0.2^5 + 1.2^4 + 1.2^3 + 0.2^2 + 0.2^1 + 1.2^0 = -128 + 16 + 8 + 1 = -103$



Número com sinal

Idéia central:

$(2^n + x) \bmod 2^n$

Se x>= 0 $rep_2(x) = x$
Se x<0 então $rep_2(x) = 2^n + x$

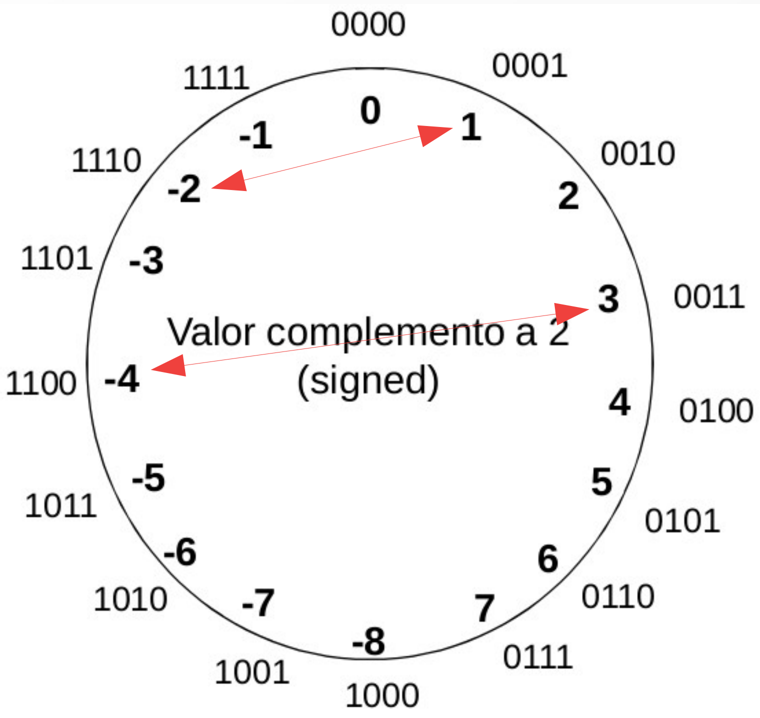
Exemplos:

$rep_2(-2) = 2^4 + (-2) = 14 = [1110]$
$rep_2(-8) = 2^4 + (-8) = 8 = [1000]$
$rep_2(-1) = 2^4 + (-1) = 15 = [1111]$

$[1111] = 15 = 2^4 + x \rightarrow \boxed{x = 15 - 16 = -1}$
$[1000] = 8 = 2^4 + x \rightarrow \boxed{x = 8 - 16 = -8}$

Encontrando x

binário	Compl-2	binário	Compl-2
0000	0	1111	-1
0001	1	1110	-2
0010	2	1101	-3
...	..	1001	-7
0111	7	1000	-8



Número com sinal

Idéia central:

$(2^n + x) \bmod 2^n$

Se $x >= 0$ $rep_2(x) = x$
Se $x<0$ então $rep_2(x) = 2^n + x$

Exemplos:

$rep_2(-2) = 2^4 + (-2) = 14 = [1110]$
$rep_2(-8) = 2^4 + (-8) = 8 = [1000]$
$rep_2(-1) = 2^4 + (-1) = 15 = [1111]$

$[1111] = 15 = 2^4 + x \rightarrow x = 15 - 16 = -1$
$[1000] = 8 = 2^4 + x \rightarrow x = 8 - 16 = -8$

Se x é número positivo, x terá valor máximo $2^{n-1}-1$.

Para $n=4$, *digamos que* $x=7=2^n-1$. *Então*

$$resto\left(\frac{2^4+7}{2^4}\right)=resto\left(\frac{23}{16}\right)=7$$

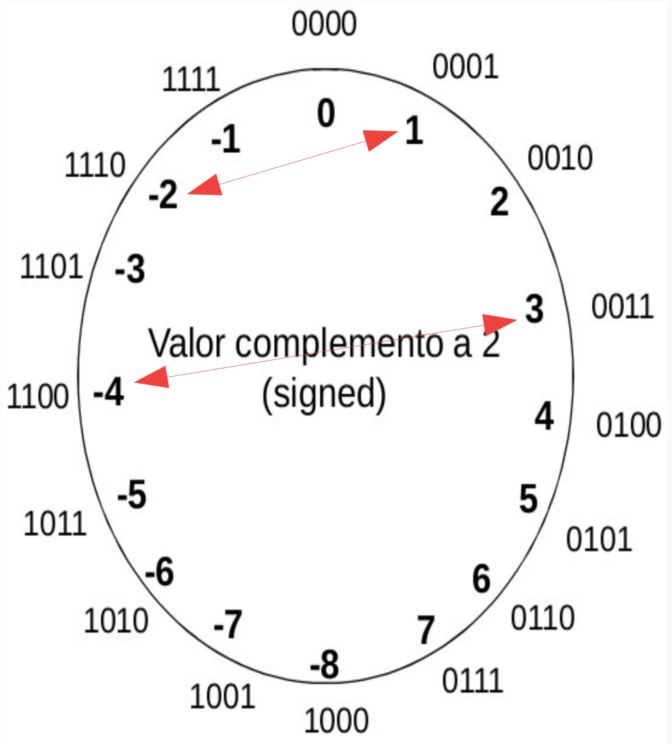
Ou seja, o resto é x.

Se x é número negativo, 2^n+x *será menor* 2^n.

Para $n=4$, *digamos que* $x=-2$. *Então*

$$resto\left(\frac{2^4+(-2)}{2^4}\right)=resto\left(\frac{14}{16}\right)=14$$

Ou seja, nesse caso o resto é 2^n+x



Número com sinal

Por que funciona (para x < 0)?

$$2^n + x = \boxed{(2^n - 1) - (-x)} + 1$$

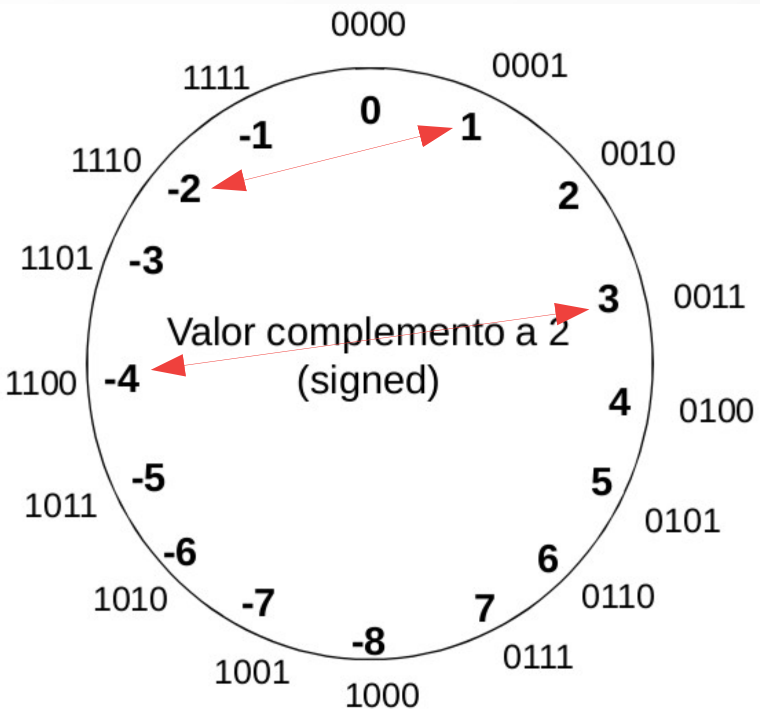
Complemento bit-a-bit de (-x)

```
  1111 1111
- 0000 0101
-----------
  1111 1010
```

5

$1 - 1 = 0$

$1 - 0 = 1$



Complemento de 0000 0101

Mudança de base

- Para que possamos fazer mudança de base, levemos em conta que computacionalmente trabalhamos com bases binárias, octais e hexadecimais. Todas essas bases são múltiplas de 2. Se desejar fazer as conversões de decimais para cada base, esteja à vontade, embora não seja recomendável.
- Basta transformar uma base decimal para uma das bases, geralmente a binária e as demais seguem desta.
- Observe que para um dígito octal são necessários 3 dígitos binários, pois temos 2^3 dígitos octais.
- Para um dígito hexadecimal, são necessários 4 dígitos binários, pois temos 2^4 dígitos hexadecimais.

- Passando 1010 1110 para octal, basta separar de 3 em 3 a sequência, fazendo 10.101.110 e substituímos cada sequência de 3 ou menos dígitos pelo equivalente em octal.

  Logo, temos que $1010\ 1110_b = 256_O$.
- Para hexadecimal basta ver a sequência de 4 em 4 dígitos, ou seja, 1010.1110.

- Logo, temos que $1010\ 1110_b = AE_h$
- Dica: Se estamos tratando de números com sinal, veja que todos os hexadecimais negativos começam com os dígitos de A a E.

d	b	o	h
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Exercícios

1) Converta para o sistema decimal os números binários abaixo. Veja que estão em diferentes quantidades de bits, todos com sinal. Lembre que quando o bit mais significativo é 1 o valor será negativo.

a) 100110_2

b) 011110_2

c) 111011_2

d) 1010000_2

e) 11000101_2

f) 11010110_2

g) 0110011001101012_2

2) Converta os valores decimais apurados para 16 bits binários

3) Converta para octal e hexadecimal o que foi resolvido em 2)

Exercícios

Os números abaixo devem ser transformados em binários (b), octal(o), decimal (d) e hexadecimal (x). Todos para 16 bits com sinal

a) 123_d

b) 277_o

c) 1011111_b

d) $AFAD_x$