Instituto Federal do Piauí
Tecnologia em Análise e Desenvolvimento de Sistemas
Introdução a Computação
Prof. Ricardo Ramos

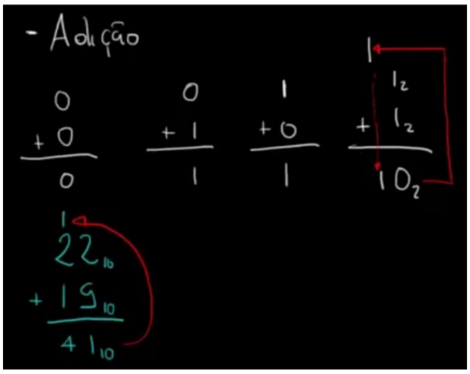
1.0 Tabela de conversão de decimal para binário (1 byte)

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Decimal
128	64	32	16	8	4	2	1	
0	0	0	0	1	0	1	0	10_{10}
0	1	1	0	0	1	0	0	100_{10}
1	1	1	1	1	1	1	1	255_{10}
0	1	1	1	1	0	0	0	120_{10}

Exercícios:
1. Converter 87_{10} para binário
2. Converter 270_{10} para binário
3. Converter 10101110_2 para decimal
4. Converter 110000010_2 para decimal

2.0 Aritmética em Binário (numero inteiro e positivo)

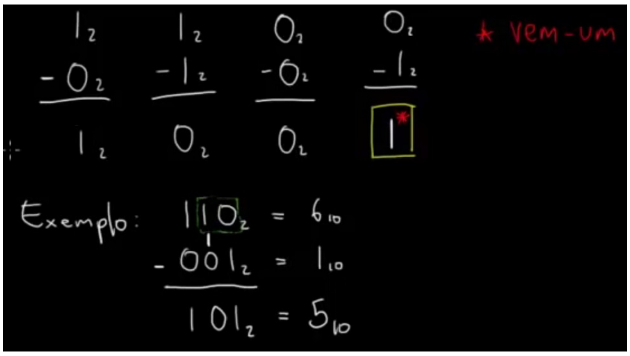
2.1 Adição



Ex 1: 2 + 3 (nibble)
1 Vai um
0 0 1 0
0 0 1 1
----
0 1 0 1 (+5)

Ex 2: 7 + 3 (nibble)
1 1 1 Vai um
0 1 1 1
0 0 1 1
----
1 0 1 0 (+10)

2.2 Subtração

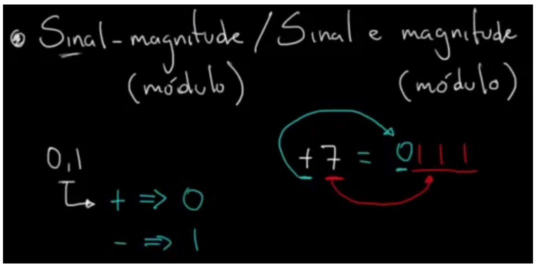


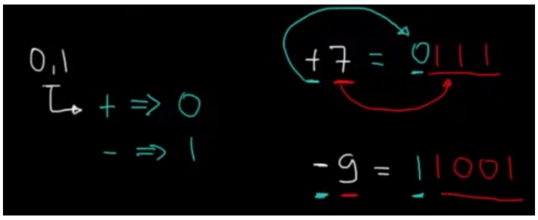
Ex 2: 6 - 3 (nibble)

```
0 1 1 0
  1 1      Vem um
0 0 1 1
---------
0 0 1 1 (+3)
```

3.0 Representação de números inteiros negativos

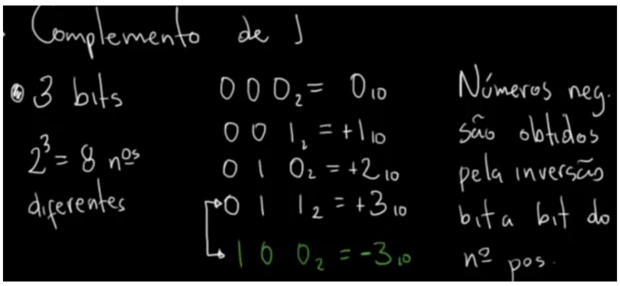
3.1 Sinal e magnitude





Observação: duas representações para o número 0 (zero). Ver tabela abaixo.

3.2 Complemento de 1



Observação: duas representações para o número 0 (zero). Ver tabela abaixo.

3.3 Complemento de 2

Faz-se o complemento de 1 e soma-se 1 ao algarismo mais à direita.

Ex: -5 (C2)

+5 = 0101
C1 = 1010
          1

C2   1011

Conteúdo da memória (nibble)	Sem sinal (decimal)	Sinal e magnitude	Complemento de 1	Complemento de 2
0000	0	**+0**	**+0**	+0
0001	1	+1	+1	+1
0010	2	+2	+2	+2
0011	3	+3	+3	+3
0100	4	+4	+4	+4
0101	5	+5	+5	+5
0110	6	+6	+6	+6
0111	7	+7	+7	+7
1000	8	**-0**	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3
1110	14	-6	-1	-2
1111	15	-7	**-0**	-1

Sinal e magnitude: menor valor = -7; maior valor = +7 menor valor = $-(2^{k-1}-1)$ k é o número de bits
Complemento de 1: menor valor = -7; maior valor = +7 menor valor = $-(2^{k-1}-1)$ k é o número de bits
Complemento de 2: menor valor = -8; maior valor = +7 menor valor = $-(2^{k-1})$ k é o número de bits
maior valor = $(2^{k-1}-1)$ onde k é o número de bits para todas as representações

Ex: 1 + 7 = ? Utilizando um nibble e a representação sinal e magnitude. **Overflow!**

Observação 1: a representação sinal e magnitude não é utilizada para armazenar números inteiros, mas é utilizada para armazenar partes de números reais. Aplicada quando quantizamos um sinal análogo, como áudio.

Observação 2: Atualmente, a representação complemento de 2 é o padrão para armazenar números inteiros em computadores.

4.0 Aritmética computacional: Sistema binário com números negativos

4.1. Sinal e Magnitude (04 bits)

Soma:
4.1.1 Mesmo sinal: somam-se as magnitudes e conserva-se o sinal das parcelas
Ex 1: 2 + 3

```
  1      Vai um
0 0 1 0
0 0 1 1
---------
0 1 0 1 (+5)
```

Ex 2: (-4) + (-3)

```
         Vai um
1 1 0 0
1 0 1 1
---------
1 1 1 1 (-7)
```

4.1.2 Sinais diferentes: subtrai-se a magnitude menor da maior e conserva-se o sinal da que possuir maior magnitude. Obs: serve também para a subtração, ver Ex 3.

Ex 1: 2 + (- 5)

```
1 1 0 1
  1        Vem um
0 0 1 0

---------
1 0 1 1 (-3)
```

Ex 2: (-7) + 3

```
1 1 1 1
         Vem um
0 0 1 1
---------
1 1 0 0 (-4)
```

Ex 3: 6 - 3 = 6 + (-3)

```
0 1 1 0
  1 1      Vem um
1 0 1 1
---------
0 0 1 1 (+3)
```

4.2. Complemento de 1 (4 bits).

Inverte-se o número de cada algarismo, ou seja, quem for 0 passa a ser 1 e quem for 1 passa para 0.
Ex: -4 = ?
+4 = 0100 e +6 = 0110
-4 = 1011 e -6=1001

Ex: 6 - 4 = 6 + (-4)

1 1 1 Vai um
0 1 1 0
+ 1 0 1 1
----
0 0 0 1
1
----
0 0 1 0 C1

Ex: 4 - 6 = 4 + (-6)
Vai um
0 1 0 0
+1 0 0 1
----
1 1 0 1 C1

Observação: se sobrar 1 (houve o vai um) soma-se com o resultado
Observação: se sobrar 0 (não houve o vai um) a resposta foi encontrada

4.3. Complemento de 2 (4 bits).

Inverte-se o número de cada algarismo, ou seja, quem for 0 passa a ser 1 e quem for 1 passa para 0. Soma-se 1 ao algarismo mais à direita.
Ex:
+4 = 0100 e +6 = 0100
-4 = 1100 e -6 = 1010

Ex: 6 - 4 = 6 + (-4)
1 1 Vai um
0 1 1 0
1 1 0 0
----
0 0 1 0 C2

Ex: 4 - 6 = 4 + (-6)
0 Vai um
0 1 0 0
1 0 1 0
----
1 1 1 0 C2 (-2) Fazendo o complemento de 2 é 0 0 0 1 + 1 = 0 0 1 0 (+2)

Observação 1: se sobrar 1 (vai um) significa que o número é positivo e a resposta foi encontrada
Observação 2: se sobrar 0 (não houve o vai um) significa que este número é negativo e a resposta foi encontrada (faz-se o complemento de 2 para verificar se a resposta está correta).

5.0 Aritmética computacional: Sistema octal

51. Soma

Ex 1: 2247 + 3566
1 1 1 Vai um
2 2 4 7
3 5 6 6
----
6 0 3 5

1 Passo: 7 + 6 = 13 (não existe no sistema octal) - 8 = 5 (vai 1)
2 Passo: 1 + 4 + 6 = 11 (não existe no sistema octal) - 8 = 3 (vai 1)
3 Passo: 1 + 2 + 5 = 8 (não existe no sistema octal) - 8 = 0 (vai 1)
4 Passo: 1 + 2 + 3 = 6 (existe no sistema octal)

5.2. Subtração

Ex 1: 7654 – 2367
7 6 5 4
1 1 Vem um
2 3 6 7
----
5 2 6 5

1 Passo: 4 - 7 não é possível, então faz-se 8 + 4 = 12 - 7 = 5 (vem 1)
2 Passo: 5 - 1 = 4, 4 - 6 não é possível, então faz-se 8 + 4 = 12 - 6 = 6 (vem 1)
3 Passo: 6 - 1 = 5, 5 - 3 = 2
4 Passo: 7 - 2 = 5

6.0 Aritmética computacional: Sistema hexadecimal

6.1. Soma

Ex 1: ABCD + BBCC
Observação: A = 10, B = 11, C = 12, D = 13, E = 14 e F = 15
1 1 1 1 Vai um
A B C D
B B C C
----
1 6 7 9 9
1 Passo: 13 + 12 = 25 (não existe no sistema hexadecimal) - 16 = 9 (vai 1)
2 Passo: 1 + 12 + 12 = 25 (não existe no sistema hexadecimal) - 16 = 9 (vai 1)
3 Passo: 1 + 11 + 11 = 23 (não existe no sistema hexadecimal) - 16 = 7 (vai 1)
4 Passo: 1 + 10 + 11 = 22 (existe no sistema hexadecimal) -16 = 6 (vai 1)

6.2. Subtração

Ex 2: 765A – 236B Observação: A = 10, B = 11, C = 12, D = 13, E = 14 e F = 15

7 6 5 A
1 1 Vem um
2 3 6 B
----
5 2 E F

1 Passo: 10 - 11 não é possível, então faz-se 16 + 10 = 26 - 11 = 15 (F) (vem 1)
2 Passo: 5 - 1 = 4, 4 - 6 não é possível, então faz-se 16 + 4 = 20 - 6 = 14 (E) (vem 1)
3 Passo: 6 - 1 = 5, 5 - 3 = 2
4 Passo: 7 - 2 = 5