ICC 6. ARITMÉTICA BINÁRIA SUBTRAÇÃO E COMPLEMENTO

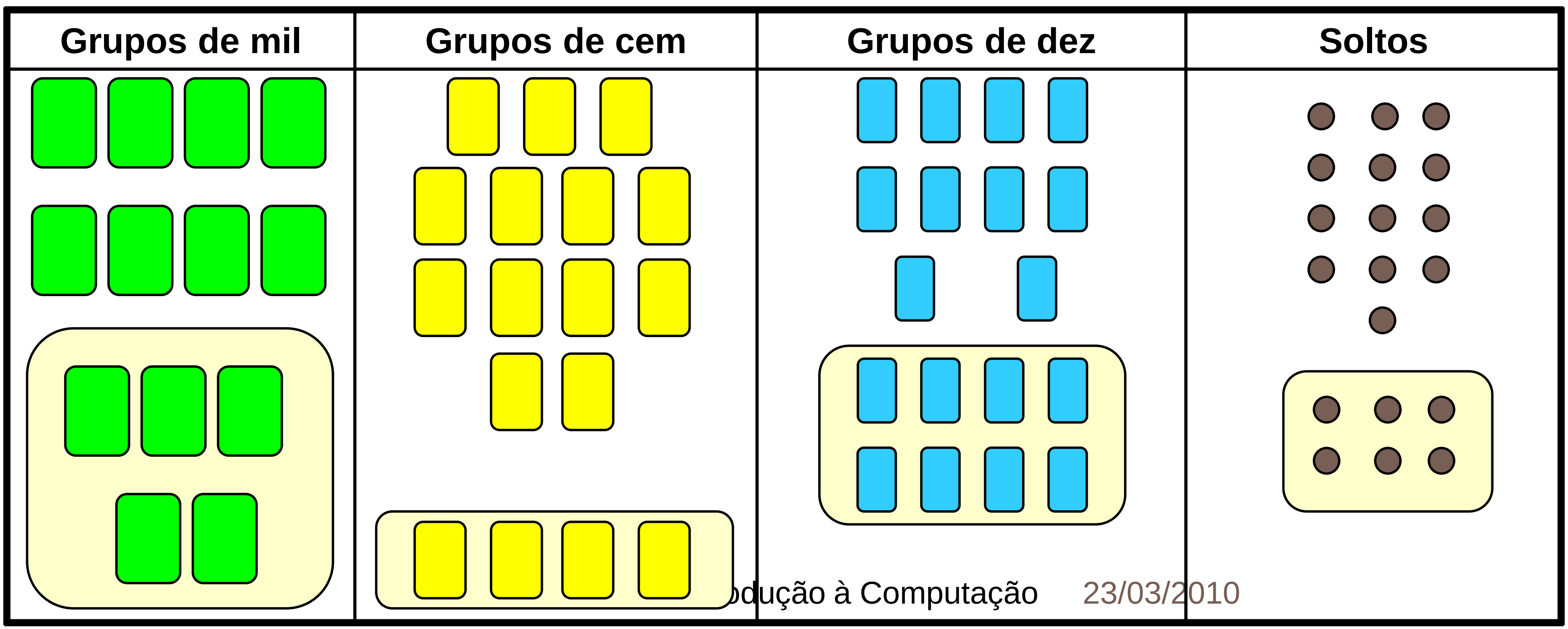
Ricardo Viana / Jorge Macêdo

Subtração Binária

- A subtração binária é realizada exatamente como subtração decimal.
- Portanto, antes realizarmos a subtração binária vamos revisar a subtração decimal.
- Você sabe que se 5486 é subtraído de 8303, a diferença 2817 é obtida.

Subtração Binária

Empréstimo		7	12	9	13
Minuendo		~~8~~	~~3~~	~~0~~	~~3~~
Subtraendo	-	5	4	8	6
Diferença		2	8	1	7



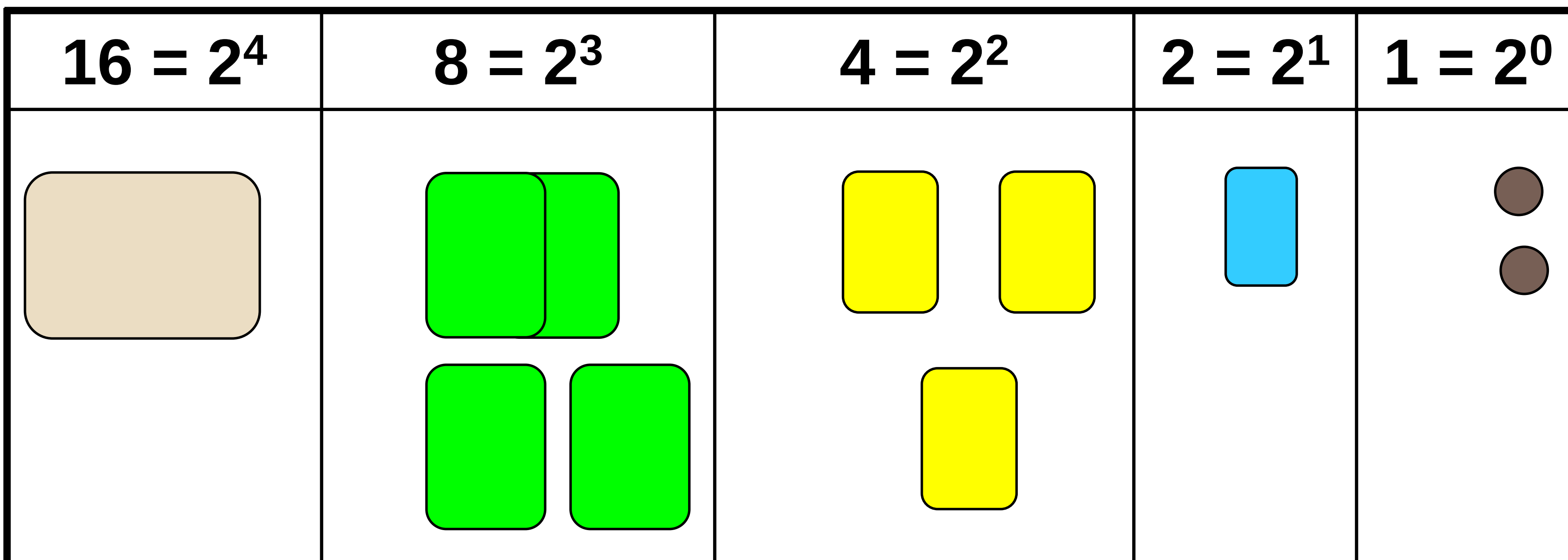
Subtração Binária

- Quando se subtrai um número binário de outro, usa-se o mesmo método descrito para subtração decimal.
- A figura a baixo resume as quatro regras para subtração binária.

```
1.   0 - 0 = 0
2.   1 - 1 = 0
3.   1 - 0 = 1
4.   0 - 1 = 1        empresta 1
```

Subtração Binária

Empréstimo	0	10	10		
Minuendo	1	1	0	1	1
Subtraendo	−	1	1	0	1
Diferença		1	1	1	0



Subtração Binária

- Na primeira coluna, 1 de 1 resulta 0 (regra 2). Então, 0 de 1 na segunda coluna resulta 1 (regra 3).
- Na terceira coluna, 1 de 0 necessita de um empréstimo da quarta coluna. Assim, 1 de 10_2 resulta 1 (regra 4).
- O minuendo na quarta coluna é agora 0, devido ao empréstimo. Portanto, um empréstimo é necessário da quinta coluna, de maneira que 1 de 10_2 na quarta coluna resulta 1 (regra 4).
- Devido ao empréstimo anterior, o minuendo na quinta coluna é agora 0 e o subtraendo é 0 (não existe), de modo que 0 de 0 resulta 0 (regra 1).
- O 0 na quinta coluna não é mostrado na diferença pois, não é um bit significativo. Assim a diferença entre 11011_2 e 1101_2 é 1110_2.

Subtração Binária

- Quando um empréstimo ("borrow") é necessário, 1 é obtido do próximo bit de ordem superior que possui 1.
- Aquele bit então, torna-se 0 e a todos os bits pulados (bits de valor 0) damos o valor 1.

Subtração Binária

- Para ilustrar o processo de subtração binária, vamos subtrair 1101_2 de 11011_2 .

Empréstimo		0	10	10	0	0
Minuendo		1	1	0	1	1
Subtraendo	–		1	1	0	1
Diferença			1	1	1	0

- Ex: subtraia 00100101_2 de 11000100_2.

Empréstimo		0	0	1	1	1	10	1	10
Minuendo		1	1	0	0	0	1	0	0
Subtraendo	–	0	0	1	0	0	1	0	1
Diferença		1	0	0	1	1	1	1	1

Exrcícios

- Realize as subtrações:
  - 10010_2 de 11111_2
  - 1_2 de 10000_2
  - 11011_2 de 100010_2
  - 10_2 de 101_2
  - 1010_2 de 1110_2
  - 10111010_2 de 11101110_2

Números Negativos

- Até agora, nós temos examinado aritmética binária usando números sem sinal.
- Entretanto, quando você realiza algumas operações aritméticas com um microprocessador, você deve estar capacitado a expressar números, **positivos e negativos**.
- Ao longo dos anos, três métodos foram desenvolvidos para representar número com sinal.
- Destes, apenas um método prosperou. Os dois métodos antigos serão examinados primeiro, e a seguir o sistema que é usado hoje em dia.

Sinal e Magnitude

- Usando este sistema, um número binário contém ambos, o sinal (+ ou -) e o valor do número.
- Portanto, valores positivos e negativos seriam expressos como na figura.
- O BMS do número binário indica o sinal, enquanto que os bits remanescente contém o valor do número. Como você pode ver, um bit de sinal zero indica um valor positivo, enquanto um bit de sinal um indica um valor negativo.
- Apesar deste método de se representar números negativos parecer lógico, ele pouco durou.
- Devido a ele requerer uma circuitaria aritmética complexa e lenta ele foi abandonado antes do advento do microprocessador.

Sinal e Magnitude

$+45_{10}$ = 0 0 1 0 1 1 0 1

SINAL

MAGNITUDE

-45_{10} = 1 0 1 0 1 1 0 1

$+127_{10}$ = 0 1 1 1 1 1 1 1

SINAL

MAGNITUDE

-127_{10} = 1 1 1 1 1 1 1 1

Complemento de 1

- É um outro método de representação de números negativos que se tornou popular no início da época dos computa-dores.Ele é chamado método do complemento de um.
- Usando este sistema, números positivos eram represen-tados do mesmo modo que no sistema de sinal magnitude.
- Ou seja, o BMS em qualquer número é considerado um bit de sinal. Um bit de sinal 0 representa positivo.
- Números negativos são representados pelo complemento de **um** do valor positivo, que é formado pela troca de todos os 0 por 1 e de todos os 1 por 0.

Complemento de 1

$+4_{10}$	=	0	0	0	0	0	1	0	0
-4_{10}	=	1	1	1	1	1	0	1	1
$+17_{10}$	=	0	0	0	1	0	0	0	1
-17_{10}	=	1	1	1	0	1	1	1	0
$+125_{10}$	=	0	1	1	1	1	1	0	1
-125_{10}	=	1	0	0	0	0	0	1	0
$+127_{10}$	=	0	1	1	1	1	1	1	1
-127_{10}	=	1	0	0	0	0	0	0	0

Bit de Sinal

Valor Binário

PADRÃO	Sem Sinal	Compl. De Um
0000 0000	0	+0
0000 0001	1	+1
0000 0010	2	+2
0000 0011	3	+3
-	-	-
-	-	-
0111 1100	124	+124
0111 1101	125	+125
0111 1110	126	+126
0111 1111	127	+127
1000 0000	128	-127
1000 0001	129	-126
1000 0010	130	-125
1000 0011	131	-124
-	-	-
-	-	-
1111 1100	252	-3
1111 1101	253	-2
1111 1110	254	-1
1111 1111	255	-0

Complemento de 2

- O método usado para representar números com sinal em microprocessadores é chamado complemento de **dois.**
- Números positivos são representados exatamente como eram com o método do **sinal e magnitude** e o método do **complemento de um.**
- Entretanto, números negativos são representados como complemento de dois dos números positivos.

Complemento de 2

- O complemento de dois de um número é formado tomando-se o complemento de um e somado-se um.
- Por exemplo se você trabalha com números de 8 bits e usa o sistema de complemento de dois, $+4_{10}$ é representado por 00000100_2.
- Para achar -4_{10} você deve achar o complemento de dois deste número. Você faz o complemento de **um**, o que é 11111011_2 e soma 1. Assim a representação em complemento de dois de -4_{10} é 11111100_2.

Complemento de 2

1	1	1	0	1	1	1	0	←	**Complemento de 1**
+							1		
1	1	1	0	1	1	1	1	←	**Complemento de 2**

1	1	1	1	1	0	1	1	←	**Complemento de 1**
+							1		
1	1	1	1	1	1	0	0	←	**Complemento de 2**

PADRÃO	SEM SINAL	COMPLEMENTO DE DOIS	COMPLEMENTO DE UM
0000 0000	0	+0	+0
0000 0001	1	+1	+1
0000 0010	2	+2	+2
0000 0011	3	+3	+3
.	.	.	.
.	.	.	.
0111 1100	124	+124	+124
0111 1101	125	+125	+125
0111 1110	126	+126	+126
0111 1111	127	+127	+127
1000 0000	128	-128	-127
1000 0001	129	-127	-126
1000 0010	130	-126	-125
1000 0011	131	-125	-124
.	.	.	.
.	.	.	.
1111 1100	252	-4	-3
1111 1101	253	-3	-2
1111 1110	254	-2	-1
1111 1111	255	-1	-0