Aula 10 – Aritmética Binária

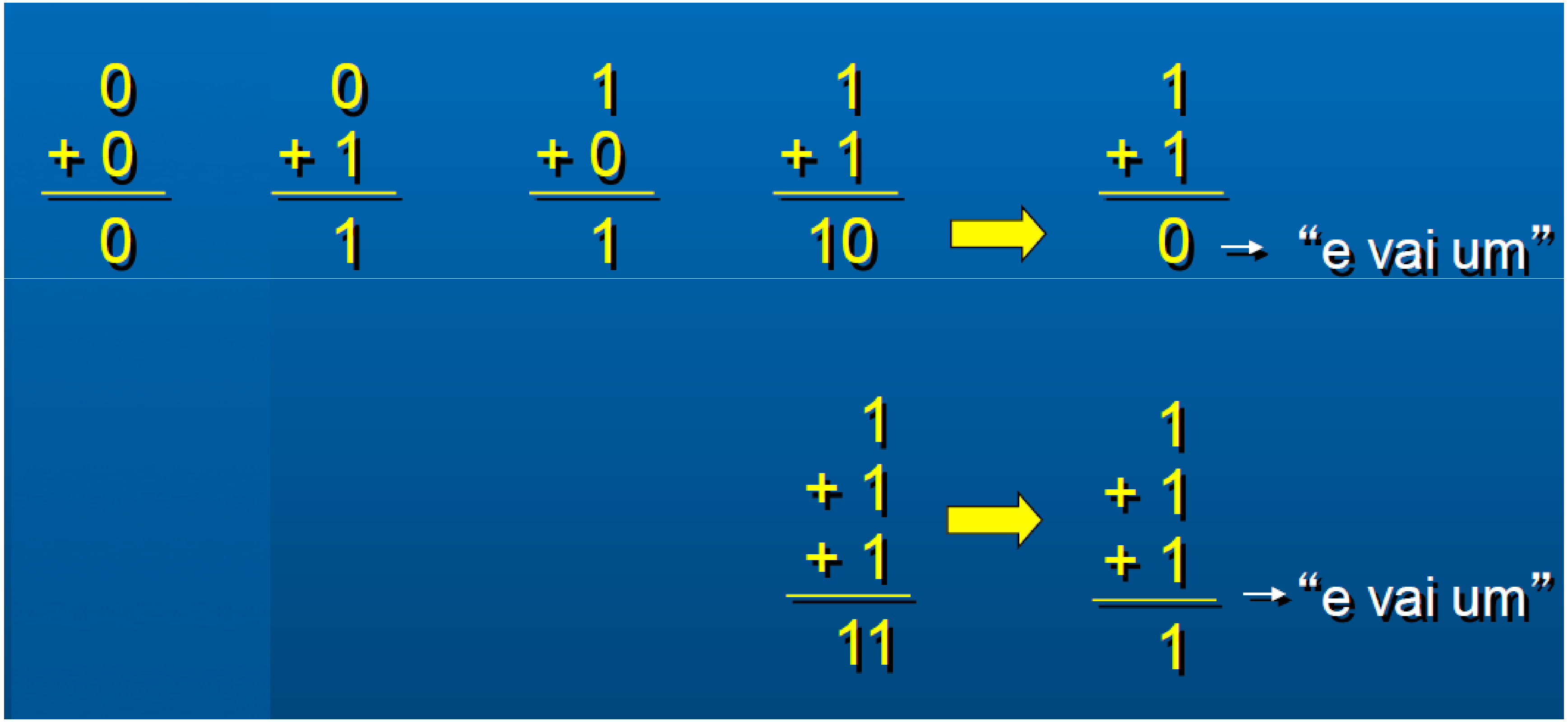
Prof. Dr. Emerson Carlos Pedrino
024376 – Circuitos Digitais
DC/UFSCar
www.dc.ufscar.br/~emerson

Soma de 2 Números Binários

Álgebra Booleana (OR)	Aritmética (+)
0 + 0 = 0	0 + 0 = 0
0 + 1 = 1	0 + 1 = 1 + 0 = 1
1 + 0 = 1	1 + 1 = 0 e "vai um" = (10)
1 + 1 = 1	1 + 1 + 1 = 1 e "vai um" = (11)

"Carry"

Soma de 2 Números Binários



Exemplos

N° 1

$$
\begin{array}{ccccccc}
1 & 1 & & 1 & 1 & & \\
 & 1 & 1 & 0 & 0 & 1 & \\
 & & 1 & 0 & 1 & 1 & + \\
\hline
1 & 0 & 0 & 1 & 0 & 0 &
\end{array}
$$

Conferindo:

$$
\begin{array}{rl}
25 & \\
11 & + \\
\hline
36 &
\end{array}
$$

Exemplos

N° 2

$$
\begin{array}{rrrrrl}
1 & 1 & 1 & 1 & & \\
 & 1 & 0 & 0 & 1 & \\
 & 1 & 1 & 1 & 1 & + \\
\hline
1 & 1 & 0 & 0 & 0 &
\end{array}
$$

Conferindo:

$$
\begin{array}{rl}
09 & \\
15 & + \\
\hline
24 &
\end{array}
$$

Exemplos

N° 3

```
 1 1 1   1
   1 1 , 0 1 1
   1 0 , 1 1 0 +
 -------------
 1 1 0 , 0 0 1
```

Conferindo:

```
  3,375
  2,750 +
  -----
  6,125
```

Soma de 2 Números BCDs

$$\begin{array}{r} 52 \\ 43\ + \\ \hline 95 \end{array} \quad \left\{ \quad \begin{array}{cccccccccc} 0 & 1 & 0 & 1 & & 0 & 0 & 1 & 0 & \\ 0 & 1 & 0 & 0 & & 0 & 0 & 1 & 1 & + \\ \hline 1 & 0 & 0 & 1 & & 0 & 1 & 0 & 1 & \end{array} \right.$$

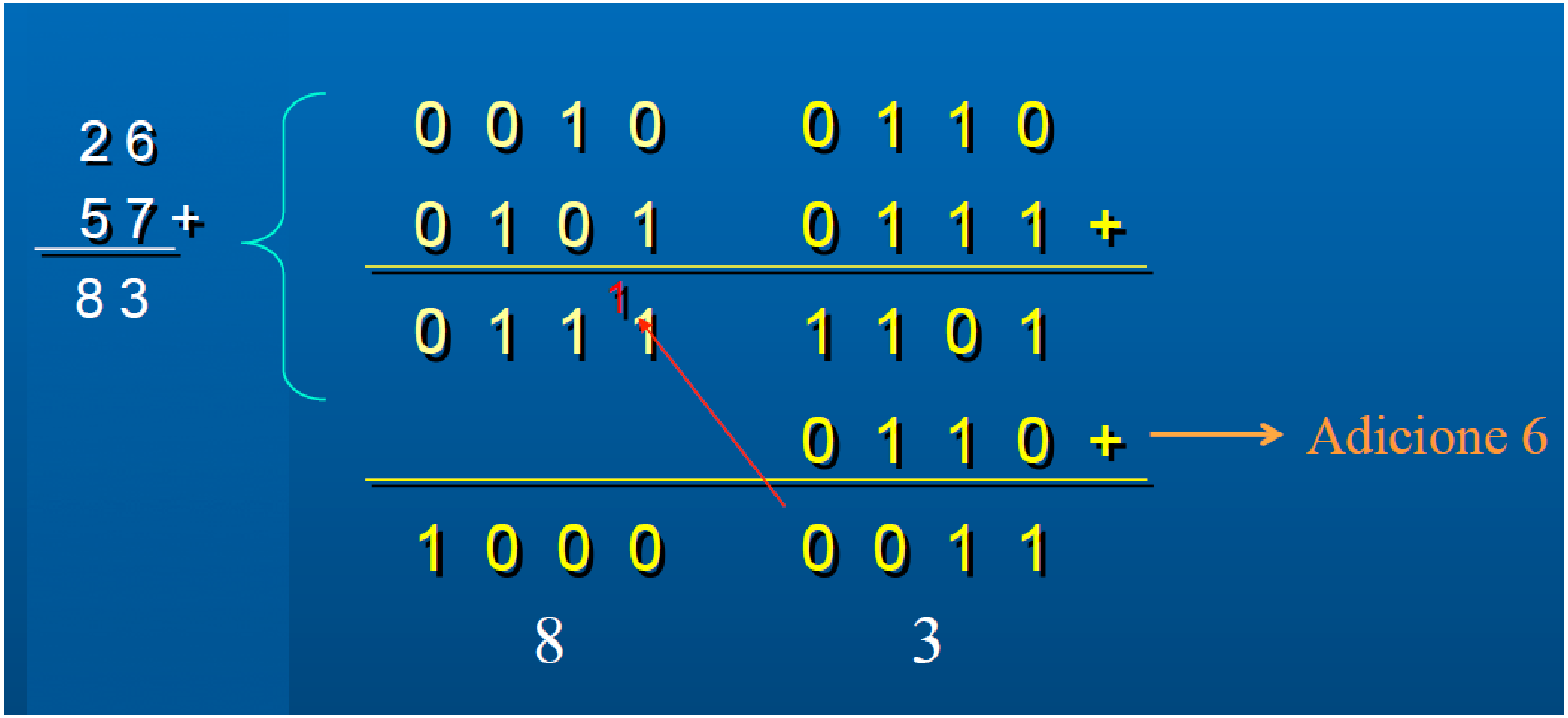
Soma de 2 Números BCDs

Quando a soma for maior do que 9?

$$
\begin{array}{r}
6 \\
7\ + \\
\hline
13
\end{array}
\qquad
\left\{
\begin{array}{cccc l}
0 & 1 & 1 & 0 & \\
0 & 1 & 1 & 1 & + \\
\hline
1 & 1 & 0 & 1 &
\end{array}
\right.
\qquad
\begin{array}{l}
\longrightarrow \text{BCD para 6} \\
\longrightarrow \text{BCD para 7} \\
\longrightarrow \textbf{BCD Inválido!}
\end{array}
$$

Somar 6 (0110) ao resultado final para corrigir o código BCD inválido e adicionar o "carry" que será gerado!

Exemplo


26
57 +
83
0 0 1 0 0 1 1 0
0 1 0 1 0 1 1 1 +
1
0 1 1 1 1 1 0 1
0 1 1 0 +
Adicione 6
1 0 0 0 0 0 1 1
8 3

Subtração Binária

Aritmética (-)

0 - 0 = 0

1 - 1 = 0

1 - 0 = 1

0 - 1 = 1 e "empresta um"

"Borrow"

Subtração Binária

$$\begin{array}{r} 0 \\ -\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ -\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ -\ 1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ -\ 1 \\ \hline 1 \end{array} \qquad \Rightarrow \qquad \begin{array}{rl} 10 & (2) \\ -\ \ 1 & (1) \\ \hline 1 & (2-1=1) \end{array}$$

"empresta um"
(2 – 1 = 1)

Exemplos

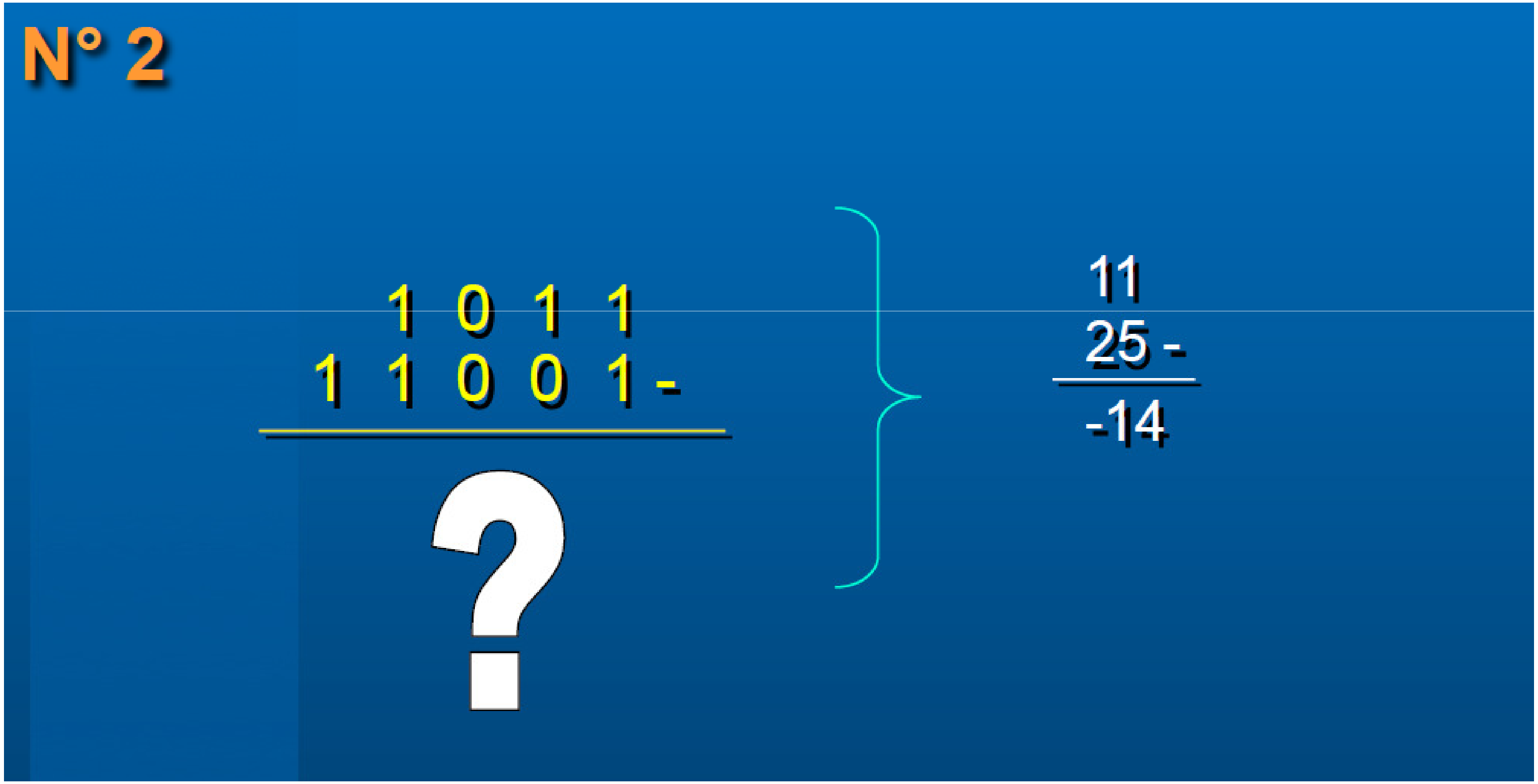
N° 1

$$
\begin{array}{r}
1\;1\;0\;0\;1 \\
1\;0\;1\;1\;- \\
\hline
0\;1\;1\;1\;0
\end{array}
$$

Conferindo:

$$
\begin{array}{r}
25 \\
11\;- \\
\hline
14
\end{array}
$$

Exemplos



Números Negativos

BIT DE SINAL

Bit mais significativo (MSB) = indicador de sinal:

- *se MSB = 0* ➡ +
- *se MSB = 1* ➡ -

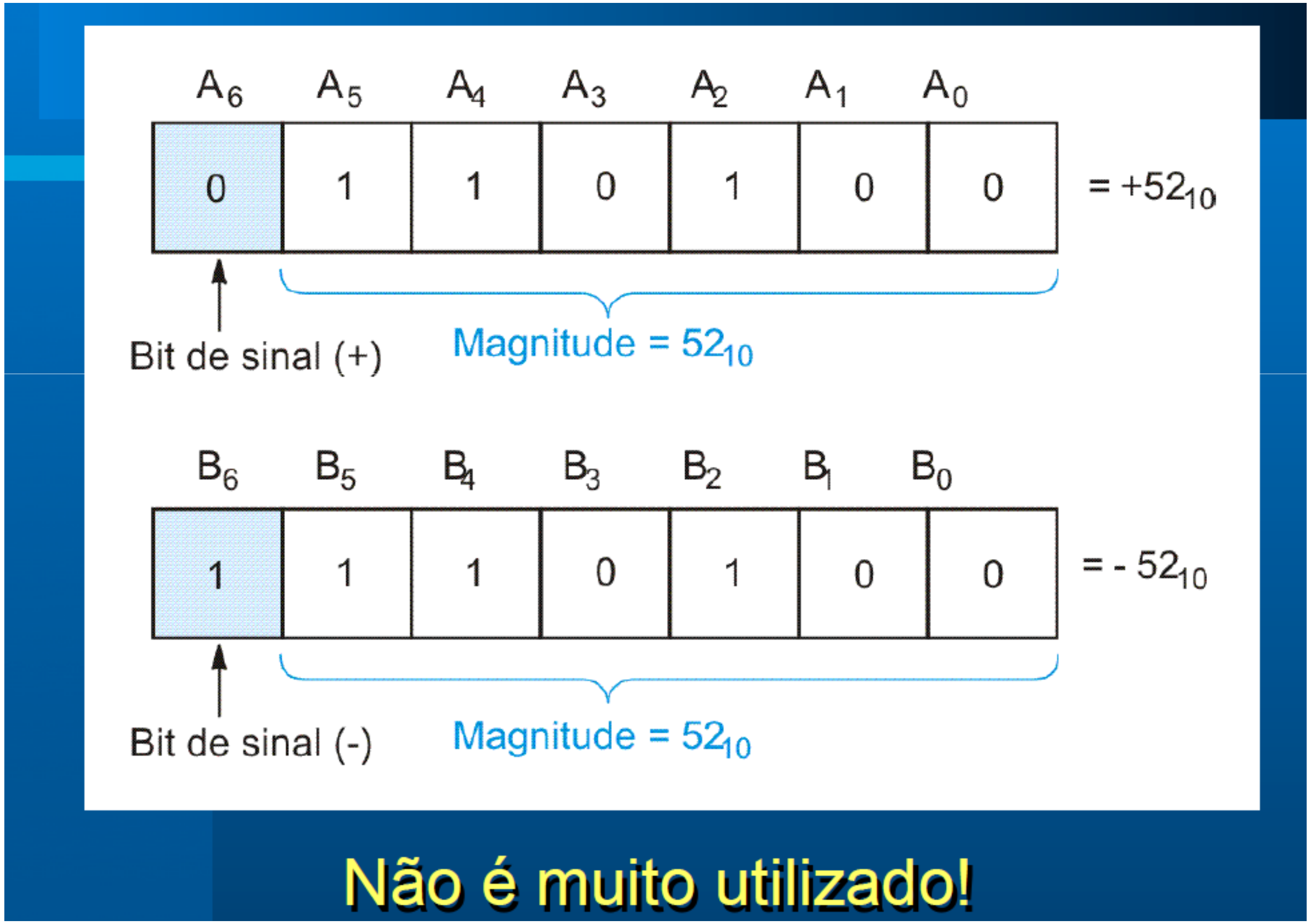
Portanto, o número binário pode ser representado por:

SINAL / MAGNITUDE

MSB — (n-1) bits restantes

Sistema Sinal-Magnitude

Sistema Sinal-Magnitude



Complemento de 1

Em binário:

Complemento ➡ $(2^n - 1)$ - número

➡ substituem-se todos os “0” por “1” e vice-versa

Comp. de 10110 = **01001**

$(2^5 - 1) = 32 - 1 = 31$ ➡ 11111 (31) - 10110 (22) = 01001 (9)

Complemento de 1

Comp. de 11011010 = **00100101**

$(2^8 - 1) = 255$ ➔ 11111111 (255) - 11011010 (218) = 00100101 (37)

Complemento de 2

Em binário:

Complemento de 2 ➡ (2^n) - número

➡ substituem-se todos os "0" por "1" e vice-versa

➡ soma-se "1" ao resultado

$\boxed{\text{Comp. de 2 de 10110} = \mathbf{01010}}$

$\boxed{(2^5) = 32 \Rightarrow 100000\ (32) - 10110\ (22) = 01010\ (10)}$

Complemento de 2

Comp. de 2 de 11011010 = **00100110**

$(2^8) = 256 \Rightarrow$ 100000000 (256) - 11011010 (218) = 00100110 (38)

Representação de Números em Complemento de 2

Complemento de 2

Exemplos

+13 ⇨ 01101

- 9 ⇨ 01001 ⇨ 10110 + 1 ⇨ 10111

- 8 ⇨ 01000 ⇨ 10111 + 1 ⇨ 11000

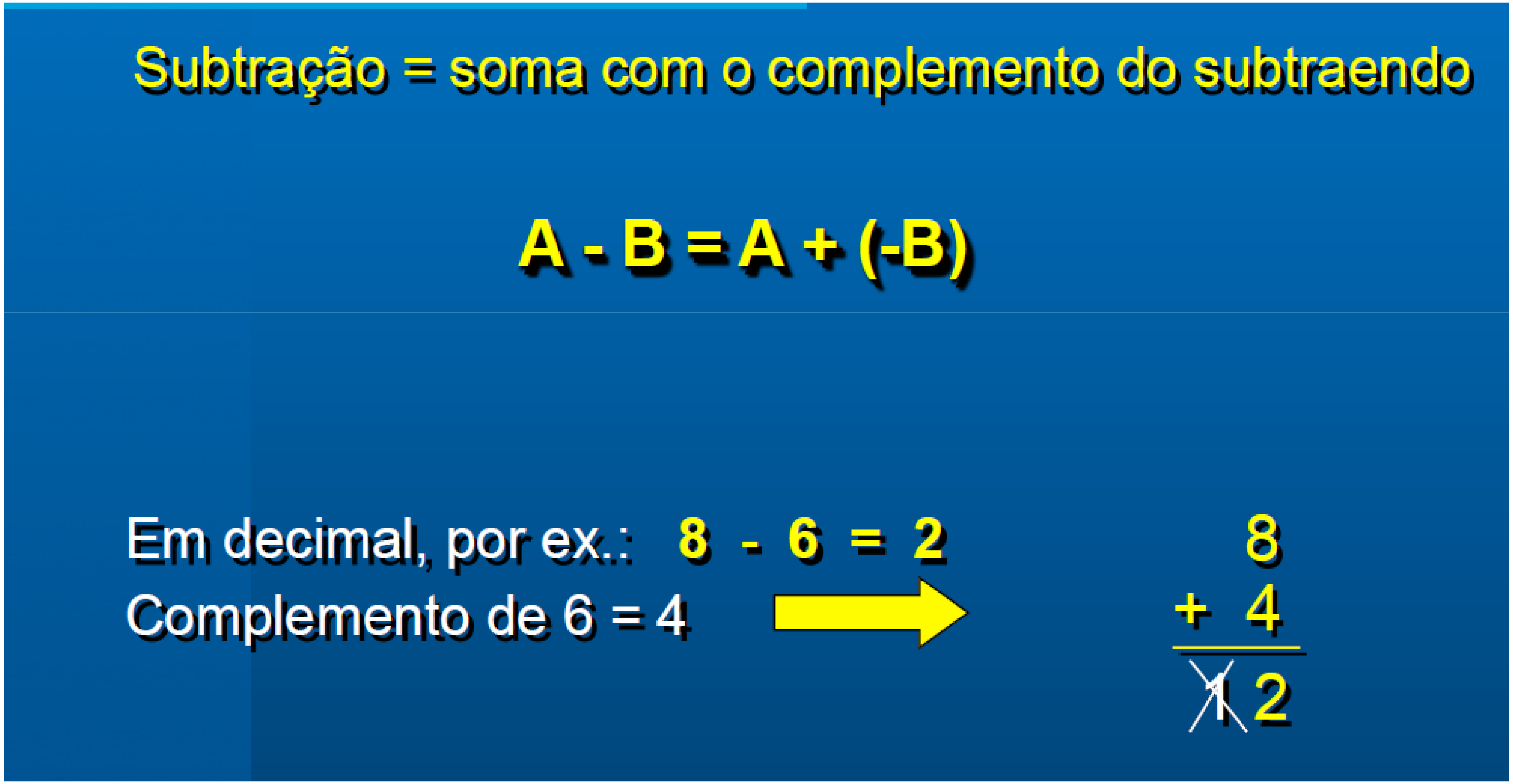
Exemplos

$01100 \Rightarrow +12$

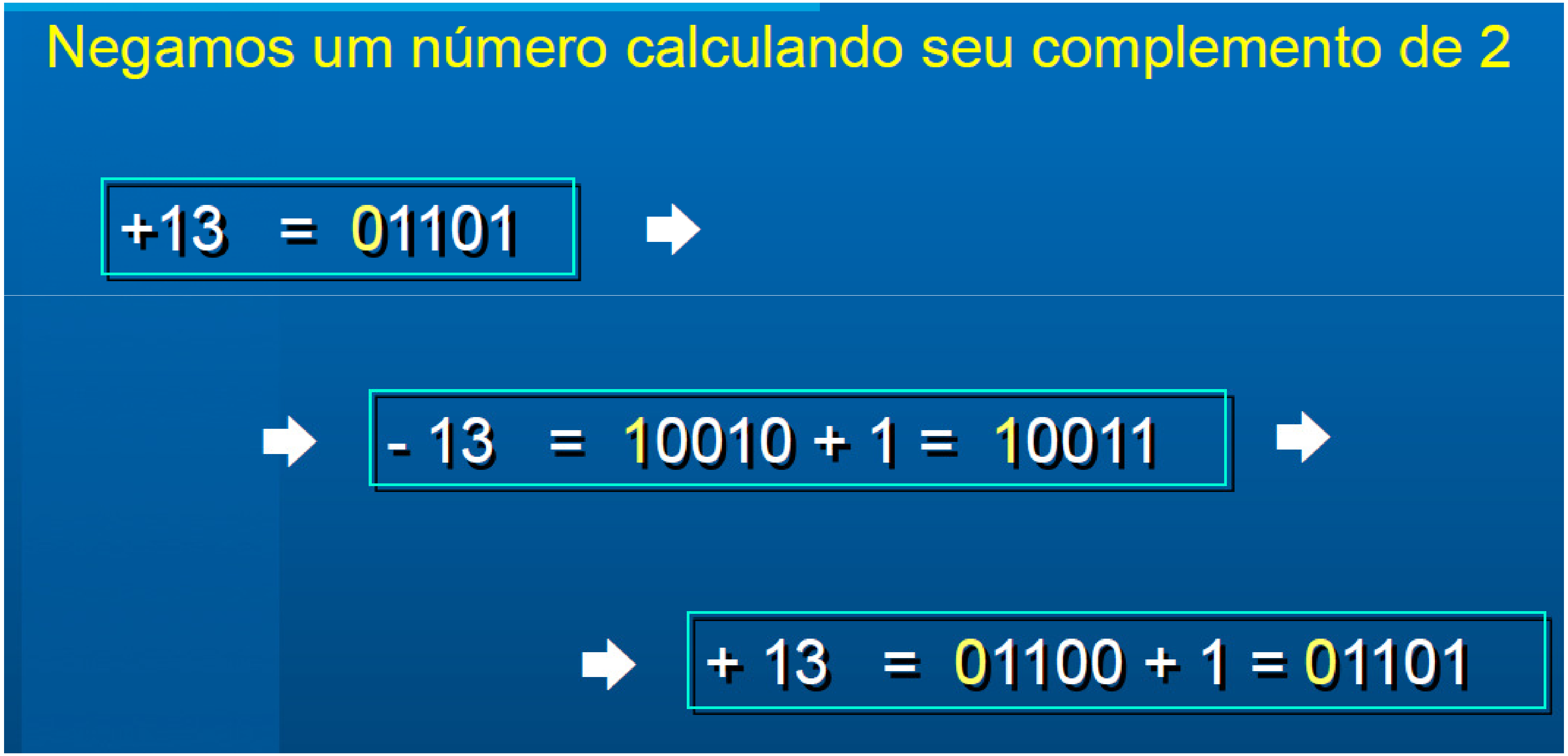
$11010 \Rightarrow 00101 + 1 \Rightarrow 0\overbrace{0110}^{6} \Rightarrow -6$

$10001 \Rightarrow 01110 + 1 \Rightarrow 0\overbrace{1111}^{15} \Rightarrow -15$

Subtração como Soma de Complementos



Negação


Negamos um número calculando seu complemento de 2
+13 = 01101
- 13 = 10010 + 1 = 10011
+ 13 = 01100 + 1 = 01101

Exemplos

Ex1:

$$\begin{array}{r} 51 \\ 18\ - \\ \hline \end{array} \qquad \begin{array}{r} 0\;1\;1\;0\;0\;1\;1 \\ 0\;0\;1\;0\;0\;1\;0\ - \\ \hline \end{array} \longrightarrow \text{comp. 2: 1101110}$$

$$\begin{array}{r} 0\;1\;1\;0\;0\;1\;1 \\ 1\;1\;0\;1\;1\;1\;0\ + \\ \hline \cancel{1}\;\underbrace{0\;1\;0\;0\;0\;0\;1}_{\text{Resultado final (+33)}} \end{array}$$

Desprezado quando estiver à esquerda do bit de sinal

Exemplos

Ex2:

18	0 0 1 0 0 0 1 0		
51 -	0 1 1 0 0 1 1	→	comp. 2: 1001101

```
  0 0 1 0 0 0 1 0
    1 0 0 1 1 0 1 +
  ---------------
    1 0 1 1 1 1 1  →  0100001
```

$\underbrace{0100001}_{33}$

Resultado final (-33)

Observação

A quantidade de 0´s colocados à esquerda de um número positivo ou 1´s colocados à esquerda de um número negativo não altera seu valor

+13 = 01101 ➡ +13 = 00001101

- 13 = 10010 + 1 = 10011 ➡ - 13 = 1110011

Usado para representar qualquer número binário, positivo ou negativo, com o número de bits desejado.

Exemplos

Ex3:

$$
\begin{array}{r} 5 \\ 25\ - \\ \hline \end{array}
\qquad
\begin{array}{r} 1\,0\,1 \\ 1\,1\,0\,0\,1\ - \\ \hline \end{array}
=
\begin{array}{r} 0\,0\,0\,1\,0\,1 \\ 0\,1\,1\,0\,0\,1\ - \\ \hline \end{array}
$$

Preencha com zeros à esquerda até que os bits de sinal fiquem alinhados

$$
\begin{array}{r} 0\,0\,0\,1\,0\,1 \\ 1\,0\,0\,1\,1\,1\ + \\ \hline 1\,0\,1\,1\,0\,0 \end{array}
\longrightarrow
\underbrace{010100}_{20}
$$

Resultado final (-20)

Exemplos

Ex4:

$$\begin{array}{r} 25 \\ 5\ - \\ \hline \end{array} \qquad \begin{array}{r} 11001 \\ 101\ - \\ \hline \end{array} = \begin{array}{r} 011001 \\ 000101\ - \\ \hline \end{array}$$

Preencha com zeros à esquerda até que os bits de sinal fiquem alinhados

$$\begin{array}{r} 011001 \\ 111011\ + \\ \hline \cancel{1}010100 \end{array} \longrightarrow \text{Resultado final (+20)}$$

Desprezado quando estiver à esquerda do bit de sinal

Faixa Completa Representada

FAIXA COMPLETA DE VALORES QUE PODEM SER REPRESENTADOS

$$-2^{(n-1)} \Rightarrow +(2^{(n-1)} - 1)$$

sendo n o número de bits

Exemplo

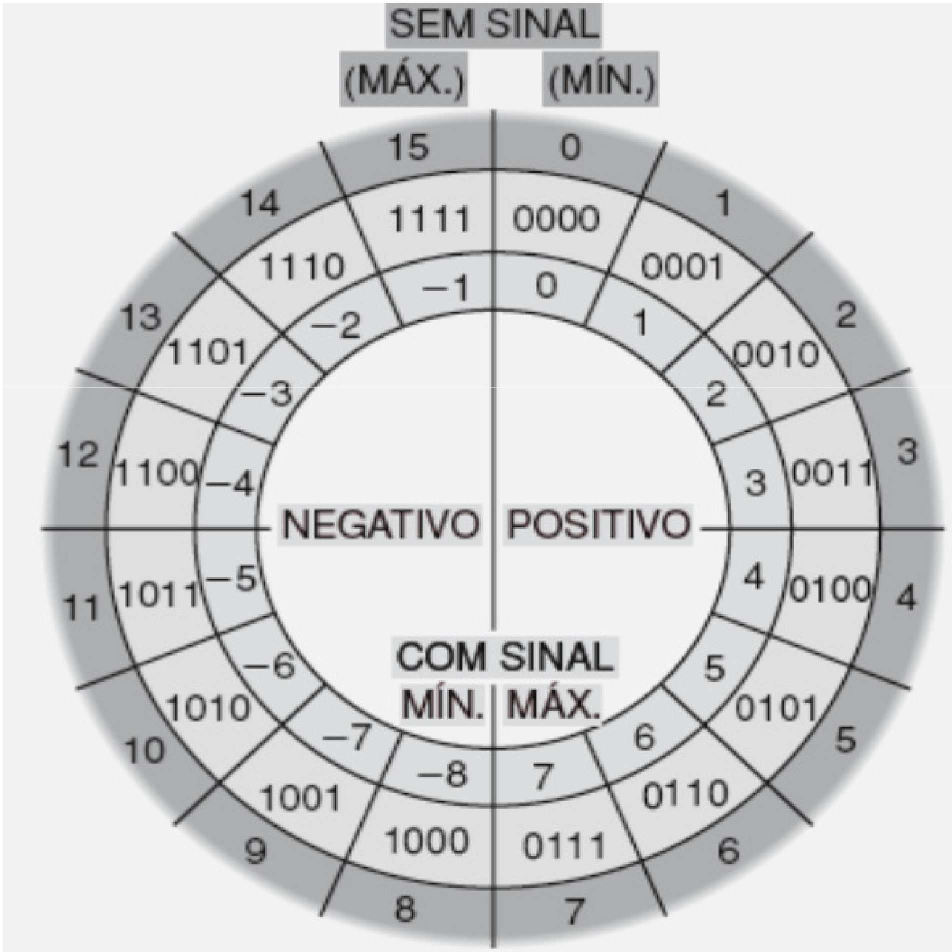
"Quantos números **com sinal** podem ser representados utilizando 4 bits?"

$-(2^{n-1}) \Rightarrow +(2^{n-1}-1) = 1000\ (-8) \Rightarrow 0111\ (+7)$

"Quantos números **sem sinal** podem ser representados utilizando 4 bits?"

$2^n = 16 \Rightarrow 0 \rightarrow 15$

Observação



SEM SINAL
(MÁX.)
(MÍN.)
15
0
14
1111
0000
1
1110
-1
0
0001
13
-2
1
2
1101
0010
-3
2
12
1100
-4
3
0011
3
NEGATIVO
POSITIVO
11
1011
-5
4
0100
4
-6
COM SINAL
5
1010
MÍN.
MÁX.
0101
10
-7
6
5
-8
7
1001
0110
9
1000
0111
6
8
7

Caso Especial de Complemento de 2

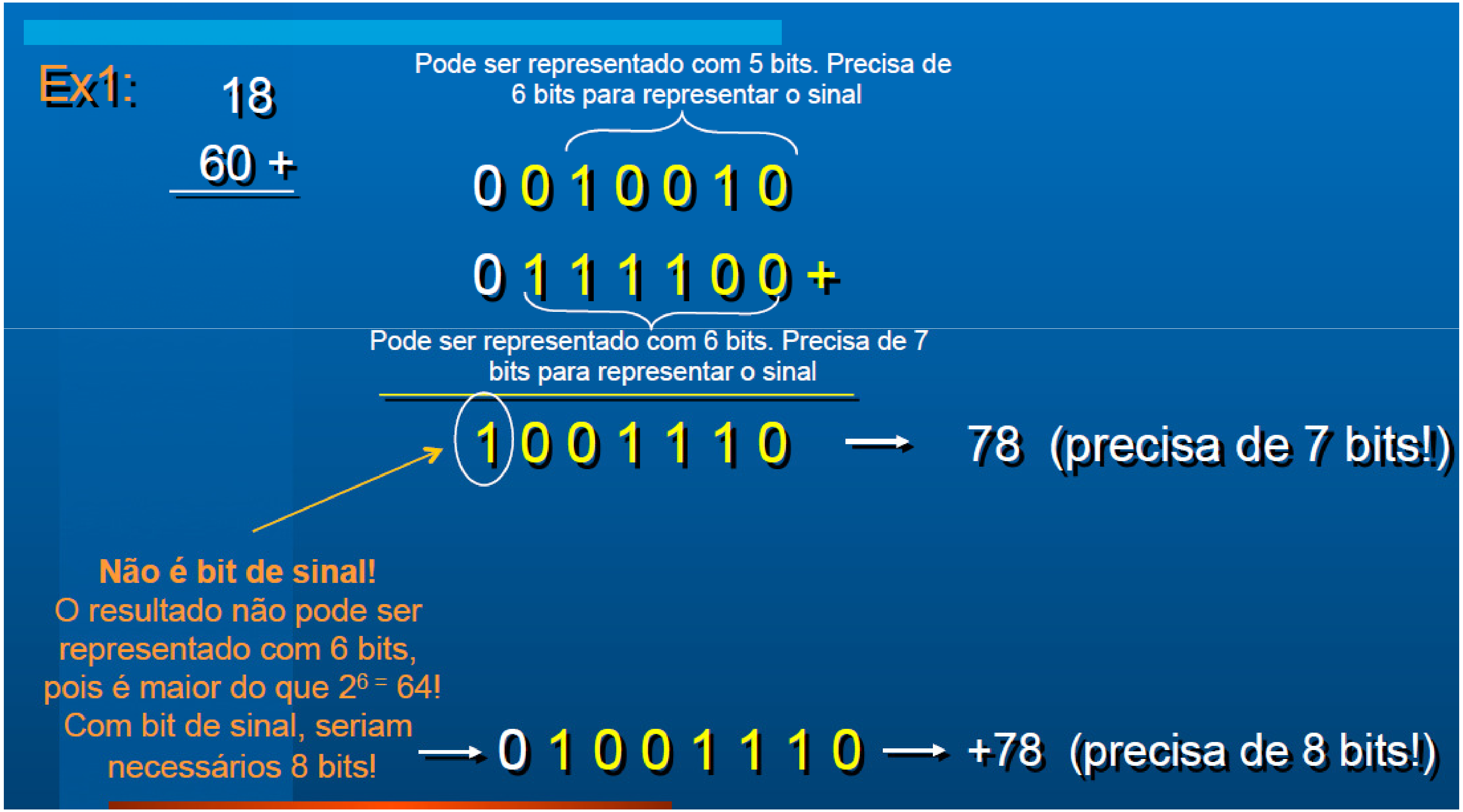
1000 ➔ 0111 + 1 ➔ 1000 ?

"Sempre que o número com sinal tiver um 1 no bit de sinal e zero em todos os outros bits, seu equivalente decimal será **-2^n**, sendo n o número de bits da magnitude"

1000 = -2^3 = - 8
10000 = -2^4 = - 16

01000 = + 8
010000 = + 16

Overflow


Ex1:
18
60 +
Pode ser representado com 5 bits. Precisa de 6 bits para representar o sinal
0 0 1 0 0 1 0
0 1 1 1 1 0 0 +
Pode ser representado com 6 bits. Precisa de 7 bits para representar o sinal
1 0 0 1 1 1 0
78 (precisa de 7 bits!)
Não é bit de sinal!
O resultado não pode ser representado com 6 bits, pois é maior do que 2⁶ = 64!
Com bit de sinal, seriam necessários 8 bits!
0 1 0 0 1 1 1 0
+78 (precisa de 8 bits!)

Overflow

Ex2:

$$
\begin{array}{r}
-18 \\
-60 \\
\hline
\end{array}
$$

$$1\ \overbrace{1\ 0\ 1\ 1\ 1\ 0}^{\text{Precisa ser representado com no mínimo 6 bits}}$$

$$\underbrace{1\ 0\ 0\ 0\ 1\ 0\ 0}_{\text{Precisa ser representado com no mínimo 7 bits}}\ +$$

$$1\ 0\ 1\ 1\ 0\ 0\ 1\ 0 \longrightarrow \text{(precisa de 8 bits!)}$$

bit de sinal!

$$\text{comp. 2: } 0\ \underbrace{1\ 0\ 0\ 1\ 1\ 1\ 0}_{78}$$

Resultado final (-78)

Overflow

- Só pode ocorrer overflow quando dois números positivos ou dois números negativos são somados. Assim, não pode haver mudança de sinal na resposta.

- Nesse caso, a necessidade de um bit extra é detectada quando o bit de sinal da resposta é diferente dos números somados.

- Quando isso ocorrer, um bit de sinal deve ser adicionado no bit mais significativo.

Multiplicação Binária

$$
\begin{array}{r}
1\ 1\ 0\ 0\ 1 \\
1\ 1\ \times \\
\hline
1\ 1\ 0\ 0\ 1 \\
1\ 1\ 0\ 0\ 1 \\
\hline
1\ 0\ 0\ 1\ 0\ 1\ 1
\end{array}
$$

$25 \times 3 = 75$

Multiplicação Binária

Multiplicamos um número binário por **2** cada vez que seus bits são rotacionados para a **esquerda** e o **zero** é colocado no bit **menos** significativo.

$$1\ 0\ 0\ 1 = 9$$

$$1\ 0\ 0\ 1\ 0 = 9 \times 2 = 18$$

$$1\ 0\ 0\ 1\ 0\ 0 = 9 \times 4 = 36$$

Divisão Binária

Dividimos um número binário por **2** cada vez que seus bits são rotacionados para a **direita** e o **zero** é colocado no bit **mais** significativo.

Atenção: Usar vírgula quando o bit menos significativo for igual a 1!

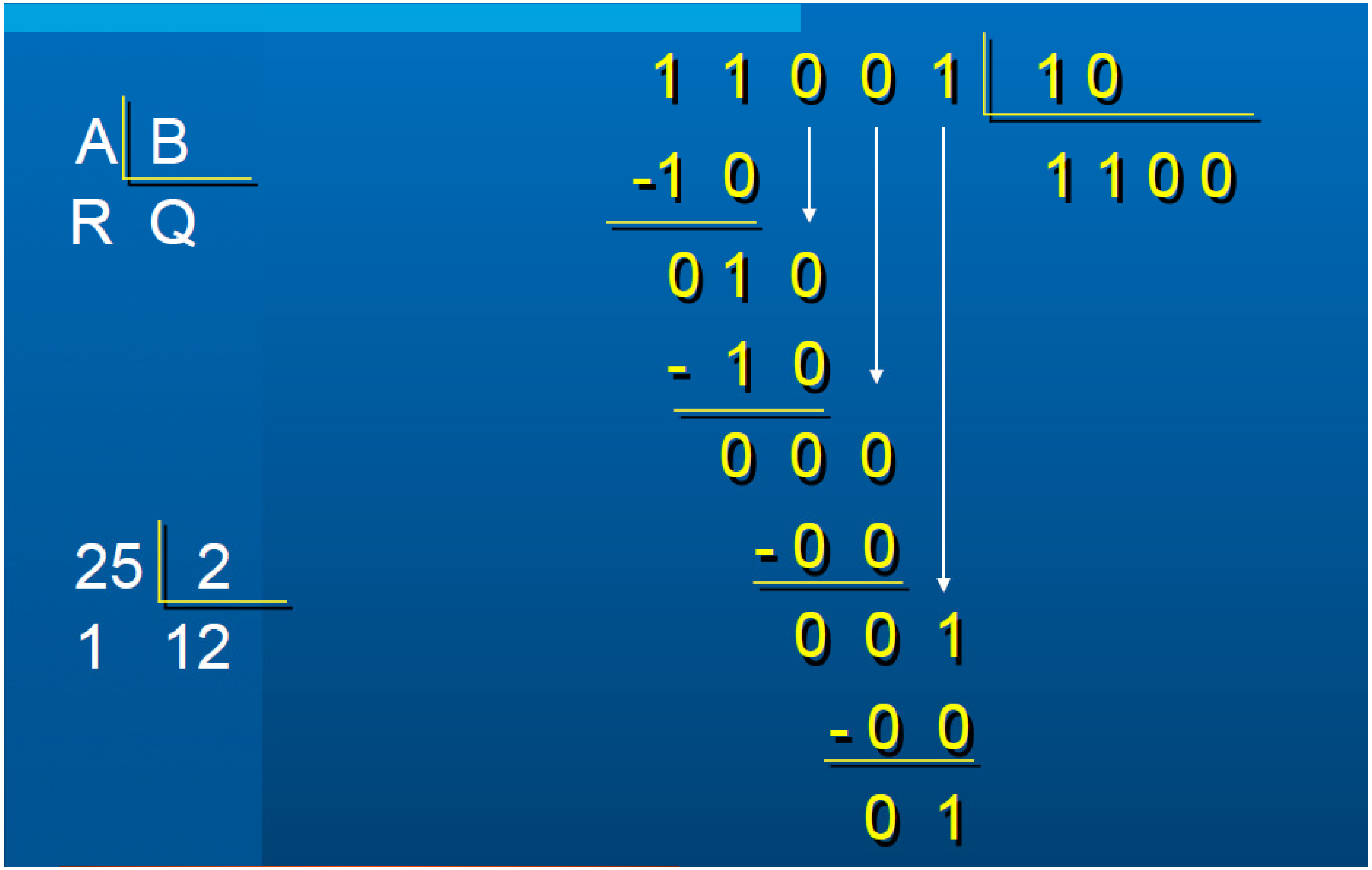
1 0 0 1 0 = 18

0 1 0 0 1 = 18 : 2 = 9

0 0 1 0 0 , 1 = 18 : 4 = 4,5

0 0 0 1 0 , 0 1 = 18 : 8 = 2,25

Divisão Binária



Divisão Binária

25 ÷ 2 = 12,5

```
1 1 0 0 1 | 1 0
-1 0        1 1 0 0,1
  0 1 0
  - 1 0
    0 0 0 1
    0 0 0 1 0
        - 1 0
    0 0 0 0 0
```

Exercícios *☺

- Fazer os seguintes exercícios:
  - De 6.1 até 6.17.

Referências

- Tocci, R. J. et al. Sistemas Digitais (princípios e aplicações), 10a Edição. Pearson, 2007.
- Vieira, M. A. C. SEL-0414-Sistemas Digitais, EESC-USP.