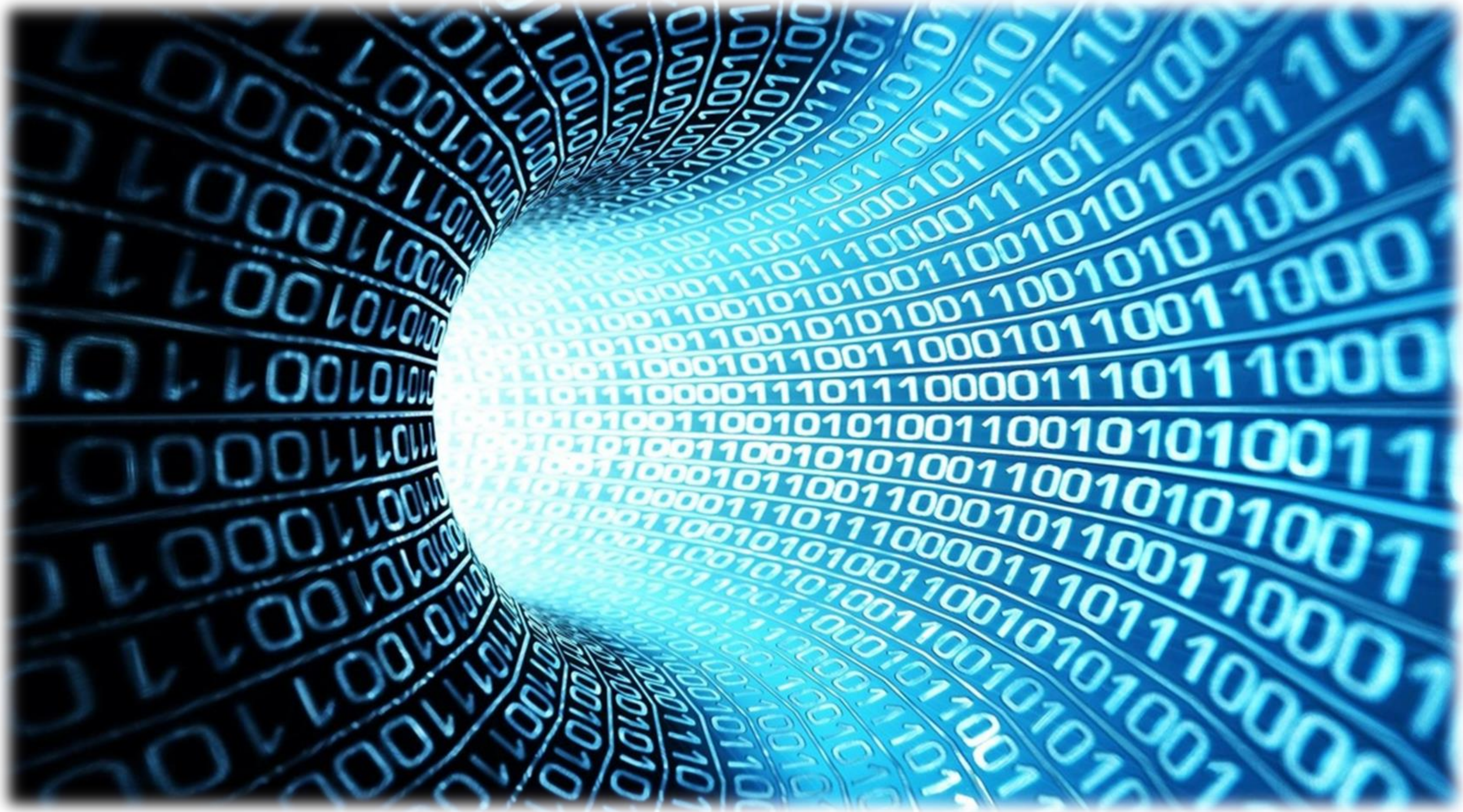
Organização e Arquitetura de Computadores



Fonte da imagem: https://cutt.ly/D4jVvQY

OBJETIVOS (continuação)

→ Compreender os conceitos do que seria a aritmética computacional:

- Representação de números - ✓;
- Conversões entre bases - ✓.

→ Como trabalhar com a aritmética não decimal ou aritmética binária - ✓.

→ Aritmética Octal, Hexadecimal - ✓.

→ Representação Numérica:

- Binário mais significativo e menos significativo - ✓;
- Conhecer os números fracionários na arquitetura de computadores - ✓;
- Ser capaz de realizar a representação numérica computacional - ✓;
- Forma dos complementos de 1 e de 2 de um número binário - ✓.

→ Divisão e Multiplicação.

→ Ponto Fixo e Ponto Flutuante.

Adição Hexadecimal

Organização e Arquitetura de Computadores

Sistema Binário – Hexadecimal

TABELA DE CONVERSÃO DE BASES Do valor "0" ao valor "15"			
Decimal Base 10	Binário Base 2	Octal (3 bits) Base 8	Hex. (4 bits) Base 16
0	000	0	0
1	001	1	1
2	010	2	2
3	011	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	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

TABELA DE CONVERSÃO DE BASES Do valor "16" ao valor "31"			
Decimal Base 10	Binário Base 2	Octal (3 bits) Base 8	Hex. (4 bits) Base 16
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F

TABELA DE CONVERSÃO DE BASES Do valor "32" ao valor "47"			
Decimal Base 10	Binário Base 2	Octal (3 bits) Base 8	Hex. (4 bits) Base 16
32	100000	40	20
33	100001	41	21
34	100010	42	22
35	100011	43	23
36	100100	44	24
37	100101	45	25
38	100110	46	26
39	100111	47	27
40	101000	50	28
41	101001	51	29
42	101010	52	2A
43	101011	53	2B
44	101100	54	2C
45	101101	55	2D
46	101110	56	2E
47	101111	57	2F

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	**Sem transporte**
Valor igual a 16 = o resultado será = 0	→	**Com transporte de 1 ou vai 1**
Valor maior que 16 = Diferença da soma parcial	→	**Com transporte de 1 ou vai 1**

Valor menor que 16:

→ Se o resultado da soma parcial for **menor** que o hexadecimal **F**, então o "**resultado final é igual ao resultado da soma sem vai 1**".

Valor igual a 16:

→ Se o resultado da soma parcial for **menor** que o hexadecimal **F** e **resultar em 0 (zero)**, teremos um **resultado igual á 0 (zero) com um vai 1**.

Valor maior que 16:

→ Se o resultado da soma parcial **for subtraído do hexadecimal máximo F** e resultar em um valor **maior que 0 (zero)**, o resultado final é igual a diferença do hexadecimal **F menos** o valor que está sendo somado com **vai um**.

Organização e Arquitetura de Computadores

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

B = 11

11 + 5 = 16 (segunda regra)

Resultado = **0** e vai 1.

1º

$$
\begin{array}{r}
{\scriptstyle 1} \\
3\,A\,9\,4\,3\,\boxed{B} \\
+\;\; 2\,3\,B\,7\,D\,\boxed{5} \\
\hline
0
\end{array}
$$

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

2º

```
        1 1
  3 A 9 4 3 B
+ 2 3 B 7 D 5
-------------
          1 0
```

1 + 3 = **4** e **D** = 13

13 + 4 = 17 - 16 (F) = **1**

(terceira regra)

Resultado = **1** e vai 1.

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

3º

```
      1 1
   3 A 9 4 3 B
+  2 3 B 7 D 5
  ------------
         C 1 0
```

1 + 4 **=** 5 + 7 **=** 12 **= C**

(primeira regra)

Resultado = **C**.

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

B = 11

9 + 11 = **20** - 16 (F) = **4**

(terceira regra)

Resultado = Fica **4** e vai 1.

4º

```
   1    1 1
  3 A 9 4 3 B
+ 2 3 B 7 D 5
-------------
      4 C 1 0
```

Organização e Arquitetura de Computadores

Sistema Binário – Hexadecimal – Adição

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

5º

1 1 1

3 A 9 4 3 B
+ 2 3 B 7 D 5

E 4 C 1 0

A = 10

1 + 10 (A) + 3 = 14 = **E**

(primeira regra)

Resultado = **E**.

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

6º

1 1 1

3 A 9 4 3 B
\+ 2 3 B 7 D 5

5 E 4 C 1 0

3 + 2 = 5 (primeira regra)

Resultado = **5**.

Resposta:

3 A 9 4 3 B = 3839035
\+ 2 3 B 7 D 5 = 2340821

5 E 4 C 1 0 = 6179856

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Soma Hexadecimal na forma de Tabela

Col_6	Col_5	Col_4	Col_3	Col_2	Col_1	
1	1		1	1		
	5	F	9	C(12)	C (12)	
	B(11)	9	2	B(11)	4	
1	1	8	C	8	0	

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Soma Hexadecimal na forma de Tabela

Col_7	Col_6	Col_5	Col_4	Col_3	Col_2	Col_1
	1	1	1	13+4=17		
	7	A(10)	C(12)	D(13)	3	4
+	7	B(11)	F(15)	4	C(12)	B(11)
	F	6	C(12)	1	F	F

1+12+15=28
28-16=12=C e vai 1
para a prox. col.

REGRAS PARA SOMA HEXADECIMAL		
Valor menor que 16 = valor da soma	→	Sem transporte
Valor igual a 16 = o resultado será = 0	→	Com transporte de 1 ou vai 1
Valor maior que 16 = Diferença da soma parcial	→	Com transporte de 1 ou vai 1

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Soma Hexadecimal na forma de Tabela

Col_7	Col_5	Col_4	Col_3	Col_2	Col_1	
		1	1	15+12=27 27-16=11		
		2	(3C) = F	F	1	
+			A	(1B) = C	2	
		3	A	B	3	
		1+15+10=26-16=10=A		11+1=12=C		

Segunda Col.: F=15 e 1B = 12 = C
Terceira Col.: 3C = 15 = F

Regras para a Subtração hexadecimal:

1. **Minuendo - Subtraindo = Diferença**;
2. A operação é realizada algarismo por algarismo;
3. Se o algarismo do minuendo **for menor que o** subtraindo, **acrescentamos** ao minuendo **um valor igual ao da base 16**, ou seja, **16 emprestados do minuendo a sua esquerda**.
4. O minuendo da esquerda para a representar um valor de uma base 16 a menos, por exemplo: se a esquerda tivermos um **hexadecimal "E"**, este passará a representar um **hexadecimal "D"**, e **passará a direita um valor de 16** para concluir a subtração da coluna.
5. Se o minuendo **for maior que o** subtraindo apenas efetuamos a subtração.
6. Não possui tabela de regras para aplicação.

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

8 é **menor** que o hexa "**A=10**", então precisamos emprestar 1 (16) a direita e o hexa **"E" passa para "D"**.

16 + 8 = **24** e "A" = **10**

24 – 10 = 14 = "**E**"

1º

D 16

4 C 7 B ~~E~~ $\boxed{8}$ → Minuendo

\- 1 E 9 2 7 $\boxed{A}$ → Subtraindo

E

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

2º

D = 13

13 - 7 = **6**

D 16

$$
\begin{array}{rcccccc}
 & 4 & C & 7 & B & \cancel{E} & 8 \\
- & 1 & E & 9 & 2 & 7 & A \\
\hline
 & & & & & 6 & E
\end{array}
$$

4C7BE8 → **Minuendo**

1E927A → **Subtraindo**

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

3º

B = 11

11 - 2 = **9**

D 16

4 C 7 $\boxed{B}$ ~~E~~ 8 → **Minuendo**

\- 1 E 9 $\boxed{2}$ 7 A → **Subtraindo**

9 6 E

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

7 é **menor** que 9, então precisamos emprestar 1 (16) a direita e o hexa **"C" passa para "B"**.

16 + 7 = **23**

23 – 9 = 14 = "**E**"

4º

B 16 D 16

4 ~~C~~ 7 B ~~E~~ 8 → **Minuendo**

- 1 E 9 2 7 A → **Subtraindo**

E 9 6 E

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

O hexa "B" é **menor** que o hexa "E", então precisamos emprestar 1 (16) a direita e o hexa **"4" passa para "3"**.

11 + 16 = **27**

27 – 14 = 13 = "**D**"

5º

16

3 B 16 D 16

~~4~~ ~~C~~ 7 B ~~E~~ 8 → **Minuendo**

\- 1 E 9 2 7 A → **Subtraindo**

D E 9 6 E

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

6º

$$
\begin{array}{rcccccc}
 & & 16 & & & & \\
 & 3 & B\ 16 & & & D\ 16 & \\
 & \boxed{\cancel{4}} & \cancel{C} & 7 & B & \cancel{E} & 8 \\
- & \boxed{1} & E & 9 & 2 & 7 & A \\
\hline
 & 2 & D & E & 9 & 6 & E
\end{array}
$$

4 C 7 B E 8 → Minuendo

1 E 9 2 7 A → Subtraindo

3 – 1 = **2**

Resposta:

$$
\begin{array}{rl}
 & 4\,C\,7\,B\,E\,8 = 5012456 \\
- & 1\,E\,9\,2\,7\,A = 2003578 \\
\hline
 & 2\,D\,E\,9\,6\,E = 3008878
\end{array}
$$

Tabela Hexadecimal

Símbolo	Valor absoluto
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

	Subtração Hexa usando tabela						
Colunas ⇒	7	6	5	4	3	2	1
		16			16	16	
	~~0~~=0	1	E	~~9~~ = 8	~~2~~ = 1	7	A
	-	4	C	7	B	E	8
	0	D	2	1	6	9	2

1. Coluna: A – 8 = 2
2. Coluna: (16 + 7) – E = 9
3. Coluna: (16 + 1) – B = 6
4. Coluna: 8 – 7 = 1
5. Coluna: E – C = 2
6. Coluna: (16 + 1) – 4 = D
7. Coluna: ϕ = ϕ

Multiplicação e Divisão Binária

Multiplicação Binária de Números inteiros sem Sinal

O processo de multiplicação é feita manualmente, complexo e demorado, onde:

→ Envolve a **geração** de produtos **parciais**, um resultado para cada dígito do multiplicador que serão somados ao final na obtenção do resultado final.

→ Caso o ***bit* do multiplicador seja** ϕ, o valor parcial **é** ϕ **.**

→ Se o ***bit* do multiplicador for** 1, o **produto parcial será o próprio multiplicando**.

→ Cada produto parcial é **deslocado** um dígito a esquerda em relação ao anterior.

→ A resultante "**N**" será obtida somando todos os produtos parciais.

→ A multiplicação de números inteiros de "**n *bits***", sendo "**n**" usado parra indicar a posição e o cálculo do expoente da base, ***n – 1***.

→ A fórmula usada no cálculo, já treinada e estuda, é "$N = n * b^{n-1}$ **ou** $N = n * b^{\wedge}$", está última desenvolvida pelo professor.

Um exemplo de como estamos acostumados a multiplicar:

Multiplicação Binária						
Multiplicando				3	2	4
Multiplicador	x			2	2	3
Primeiro Produto Parcial				9	7	2
Segundo Produto Parcial	+		6	4	8	
Terceiro Produto Parcial		6	4	8		
Produto Final		7	2	2	5	2

Na multiplicação de base 10 o **multiplicando** multiplica cada dígito do **multiplicador** para após as multiplicações, somar os **produtos parciais e obter o valor final**.

Simples!

Na multiplicação de base 2 os conceitos são parecidos, calcula-se os valores do **multiplicando** *bit* a *bit* pelo **multiplicador** e soma-se os valores para obter a **Resultante**.

REGRAS DA MULTIPLICAÇÃO BINÁRIA			
Regra 1	0 * 0 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
Regra 2	0 * 1 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
Regra 3	1 * 0 = 0	⇒	Com carry ou vai um para a próxima coluna/posição.
Regra 4	1 * 1 = 1	⇒	Com carry ou vai um para a próxima coluna/posição.

REGRAS DA SOMA BINÁRIA – TABELA PADRÃO		
0 + 0 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
0 + 1 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 0 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 1 = 0	⇒	Com carry ou vai um para a próxima coluna/posição.
1 + 1 + 1 = 1	⇒	Com carry ou vai um para a próxima coluna/posição.

Então pode-se dizer que a multiplicação de base 2 é um processo de deslocamento e soma, assim como na base 10, onde para cada ***bit* do multiplicador** é usado para multiplicar cada ***bit* do multiplicando**, mas para verificar qual regra será aplicada vamos analisar um exemplo usando as regras da multiplicação e depois com as regras da soma:

Multiplicação de números inteiros binários sem Sinal				
	$(1011)_2$		Multiplicando $(11)_{10}$	Multiplicação de 4 *bits*
X	$(1101)_2$		Multiplicador $(13)_{10}$	
+	1011		**Produtos Parciais:** como os binários são de 4 bits não há necessidade de se igualar o número de bits do multiplicador ou do multiplicando.	
	0000			
	1011			
	1011			
	$(10001111)_2$		**Produto = $(143)_{10}$**	

Assim, com base no cálculo do slide anterior não serão usadas as “**Regras da Multiplicação Binária**” por não necessitar, mas será usado as “**Regras da Soma Binária**”.

REGRAS DA MULTIPLICAÇÃO BINÁRIA			
Regra 1	0 * 0 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
Regra 2	0 * 1 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
Regra 3	1 * 0 = 0	⇒	Com carry ou vai um para a próxima coluna/posição.
Regra 4	1 * 1 = 1	⇒	Com carry ou vai um para a próxima coluna/posição.

REGRAS DA SOMA BINÁRIA – TABELA PADRÃO		
0 + 0 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
0 + 1 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 0 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 1 = 0	⇒	Com carry ou vai um para a próxima coluna/posição.
1 + 1 + 1 = 1	⇒	Com carry ou vai um para a próxima coluna/posição.

- Já foi estudado como **somar e subtrair números na notação em complemento de dois** com valores **positivo**, ⊕, por exemplo: $(1001)_2 + (0011)_2 = (1100)_2$.
- **Quando** os valores de base 2 **booleanos** assumem **valores positivos** é fácil de se resolver, basta fazer a soma, por exemplo: $(9)_{10}$ $(1001)_2$ + $(3)_{10}$ $(0011)_2$ = $(12)_{10}$ $(1100)_2$.
- **Vimos** também que ao representar os números **em complementos de dois**, por exemplo na soma também é possível calcular valores **negativos**, θ, como: $(-7)_{10}$ $(1001)_2$ + $(3)_{10}$ $(0011)_2$ com um resultado igual à $(-4)_{10}$ $(1100)_2$.

Simples, mas essa forma de cálculo para um binário booleano de base 2 **não funciona** na multiplicação. Para uma melhor compreensão vamos a uma **explicação** mais detalhada.

Organização e Arquitetura de Computadores

Sistema Binário – Multiplicação Binária de Números em Complemento de Dois

Binário com Sinal em Compl. de 2 (C2)

BASE 10 (b_{10})	BASE 2 (b_2)		
	- Bit de Sinal - MSB - Binário	Novo bit de Sinal e Complemento	
		C1	C2
$+8_{10}$	01000	10111	11000
$+7_{10}$	0111	1000	1001
$+6_{10}$	0110	1001	1010
$+5_{10}$	0101	1010	1011
$+4_{10}$	0100	1011	1100
$+3_{10}$	0011	1100	1101
$+2_{10}$	0010	1101	1110
$+1_{10}$	0001	1110	1111
0	0000	1111	10000
-1_{10}	1111	0000	0001
-2_{10}	1110	0001	0010
-3_{10}	1101	0010	0011
-4_{10}	1100	0011	0100
-5_{10}	1011	0100	0101
-6_{10}	1010	0101	0101
-7_{10}	1001	0110	0111
-8_{10}	11000	00111	01000

Multiplicação de números inteiros binários sem Sinal

	$(1011)_2$		Multiplicando $(11)_{10}$	Multiplicação de 4 *bits*
X	$(1101)_2$		Multiplicador $(13)_{10}$	
+	1011		**Produtos Parciais:** como os binários são de 4 bits não há necessidade de se igualar o número de bits do multiplicador ou do multiplicando.	
	0000			
	1011			
	1011			
	$(10001111)_2$		**Produto = $(143)_{10}$**	

Na multiplicação de $(11)_{10}$ $(1011)_2$ **X** $(13)_{10}$ $(1101)_2$ temos $(143)_{10}$ $(10001111)_2$. Mas e se os valores estivessem em complemento de 2 negativos?

Então, teríamos **conforme tabela** $(-5)_{10}$ $(1011)_2$ **X** $(-3)_{10}$ $(1101)_2$, = $(-113)_{10}$ ou $(0\,1\,1\,1\,0\,0\,0\,1)_2$, como na multiplicação o **multiplicador** e **multiplicando** **negativos**, não funcionam, **então**, passa-se o complemento de 2 para **positivos** e obter um *bit de sinal* positivo, ou seja, ***bit* zero**.

REGRAS DA SOMA BINÁRIA – TABELA PADRÃO		
0 + 0 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
0 + 1 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 0 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 1 = 0	⇒	Com carry ou vai um para a próxima coluna/posição.
1 + 1 + 1 = 1	⇒	Com carry ou vai um para a próxima coluna/posição.

Treino – Multiplicar os valores binários:

$(0\ 1\ 1\ 0\ 1\ 1)_2\ (27)_{10}$ **X** $(0\ 1\ 0\ 1)_2\ (5)_{10}$

Bit de Sinal		Multiplicação Binária								
		0		1	1	0	1	1	$(27)_{10}$	Multiplicando
X		0				1	0	1	$(5)_{10}$	Multiplicador
+										
									$(135)_{10}$	

REGRAS DA SOMA BINÁRIA – TABELA PADRÃO		
0 + 0 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
0 + 1 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 0 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 1 = 0	⇒	Com carry ou vai um para a próxima coluna/posição.
1 + 1 + 1 = 1	⇒	Com carry ou vai um para a próxima coluna/posição.

Treino – Multiplicar os valores binários:

$(0\ 1\ 1\ 1\ 0\ 0\ 1\ )_2\ (57)_{10}$ **X** $(0\ 1\ 0\ 1\ 0\ )_2\ (10)_{10}$

Bit de Sinal					Multiplicação Binária							
			0		1	1	1	0	0	1	$(57)_{10}$	Multiplicando
X			0				1	0	1	0	$(10)_{10}$	Multiplicador
+												
											$(570)_{10}$	

REGRAS DA SOMA BINÁRIA – TABELA PADRÃO		
0 + 0 = 0	⇒	Sem carry ou vai um para a próxima coluna/posição.
0 + 1 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 0 = 1	⇒	Sem carry ou vai um para a próxima coluna/posição.
1 + 1 = 0	⇒	Com carry ou vai um para a próxima coluna/posição.
1 + 1 + 1 = 1	⇒	Com carry ou vai um para a próxima coluna/posição.

Treino – Multiplicar os valores binários:

$(1\,0\,0\,0\,1\,0\,0\,)_2$ $(-60)_{10}$ **X** $(1\,0\,0\,0\,1\,)_2$ $(-15)_{10}$

Bit de Sinal						Multiplicação Binária							
				0		1	1	1	1	0	0	$(60)_{10}$	Multiplicando
X				0				1	1	1	1	$(15)_{10}$	Multiplicador
+													
0	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**$(900)_{10}$**	

Divisão Binária de Números inteiros sem Sinal

→ A divisão binária é o reverso da multiplicação e também **requer o lápis e o papel**.

→ A ideia é saber o números de vezes que um valor poderia ser subtraído por outro.

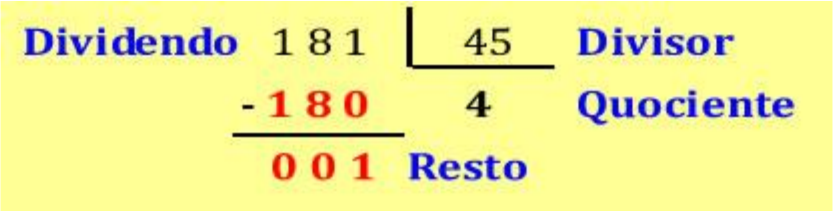
→ Chamamos esse processo de "**divisão longa**“, por exemplo, ao dividir **181 por 45**, teremos um **quociente inteiro de valor 4 com resto 1**.

→ A divisão binária poderá ser um valor exata ou não, com vírgula, seria como em uma divisão decimal, poderá gerar uma “**dizima periódica**”.

→ **Não** possui uma tabela para auxiliar ao resultado (quociente).

→ A divisão binária poderá ser um processo simples se a base for dois (2).

→ Na divisão binária temos no **quociente** apenas 0 ou 1.

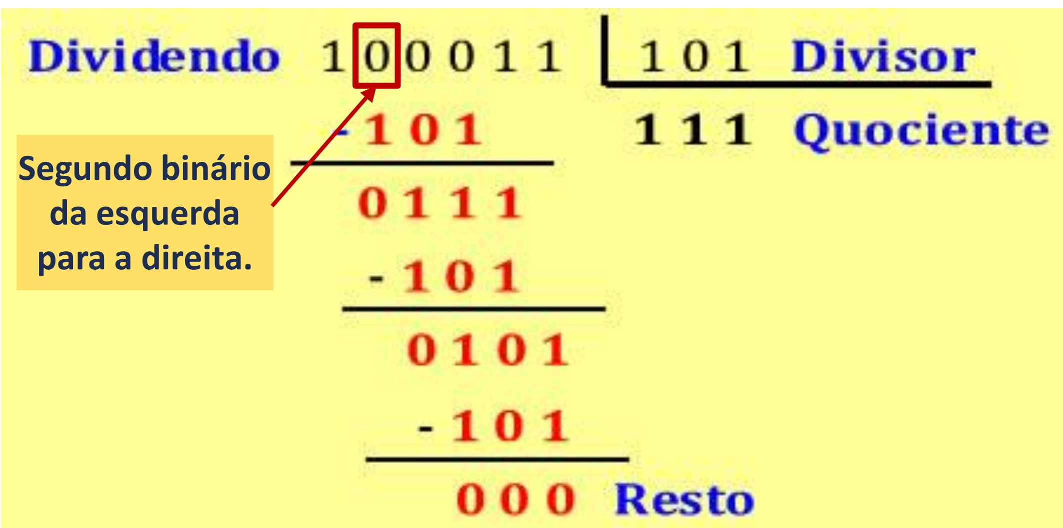


→ É necessário **subtrair o divisor** do valor do **dividendo selecionado** e o **transporte** do próximo ***bit* mais significativo** do **dividendo** para o atual **resto**.

→ Segue as regras da subtração binária.

Exemplo 01:

- Dividir $(100011)_2$ por $(101)_2$;
- Sempre iniciar **a partir do segundo binário da esquerda para direita no Dividendo**.



REGRAS DA SUBTRAÇÃO BINÁRIA – TABELA PADRÃO		
0 – 0 = 0	⇒	Sem empréstimo ou empresta 1.
1 – 1 = 0	⇒	Sem empréstimo ou empresta 1.
1 – 0 = 1	⇒	Sem empréstimo ou empresta 1.
0 – 1 = 1	⇒	Com empréstimo ou empresta 1 da coluna a esquerda para a coluna da direita.

Resposta: $(111)_2$ com resto igual a zero.

Exemplo 01:

- Dividir $(101010)_2$ por $(110)_2$;
- Sempre iniciar **a partir do segundo binário da esquerda para direita no Dividendo**.

REGRAS DA SUBTRAÇÃO BINÁRIA – TABELA PADRÃO		
0 – 0 = 0	⇒	Sem empréstimo ou empresta 1.
1 – 1 = 0	⇒	Sem empréstimo ou empresta 1.
1 – 0 = 1	⇒	Sem empréstimo ou empresta 1.
0 – 1 = 1	⇒	Com empréstimo ou empresta 1 da coluna a esquerda para a coluna da direita.

Divisão Binária de Números inteiros sem Sinal

Obs.: Quando os primeiros valores do dividendo forem menor que o divisor o cálculo se iniciará a partir do segundo *bit* “1” ou “**0**”. sucessivamente.

```
            1
Dividendo   1  0  1  0  1  0 | 1  1  0   →  Divisor
            -  1̶  1  0        ---------
                               1  1  1   →  Quociente
            0  1  0  0  1
               -  1̶  1  0
                  0  1  1  0
               -     1  1  0
                  ----------
                     0  0  0  →  Resto
```

Resposta: $(111)_2$ com resto igual a zero.

REGRAS DA SUBTRAÇÃO BINÁRIA – TABELA PADRÃO		
0 – 0 = 0	⇒	Sem empréstimo ou empresta 1.
1 – 1 = 0	⇒	Sem empréstimo ou empresta 1.
1 – 0 = 1	⇒	Sem empréstimo ou empresta 1.
0 – 1 = 1	⇒	Com empréstimo ou empresta 1 da coluna a esquerda para a coluna da direita.

Exemplo 02:

→ Dividir o binário $(100101)_2$ / $(100)_2$

Divisão Binária de Números inteiros sem Sinal

Dividendo		1	0	0	1	0	1	1	0	0	→	**Divisor**
	-	1	0	0				**1**	**0**	**0**	**1** →	**Quociente**
		0	0	0	**1**	**0**	**1**					
				-	1	0	0					
					0	0	1	→	Resto			

Os zeros foram acrescidos para se baixar os valores do "dividendo

Resposta: $(1001)_2$ com resto de "1". Caso o resto fosse considerado a resposta seria $(1001.10)_2$.

Quando não for possível trabalhar com **dividendos** que possuem um valor muito baixo para a próxima subtração acrescentamos um "0" no quociente e desce-se ao próximo bit do dividendo, e caso ainda não seja possível a subtração, a sequência de zeros no quociente deverá ser repetida, até ser possível efetuar a subtração com o divisor.

REGRAS DA SUBTRAÇÃO BINÁRIA – TABELA PADRÃO		
0 – 0 = 0	⇒	Sem empréstimo ou empresta 1.
1 – 1 = 0	⇒	Sem empréstimo ou empresta 1.
1 – 0 = 1	⇒	Sem empréstimo ou empresta 1.
0 – 1 = 1	⇒	Com empréstimo ou empresta 1 da coluna a esquerda para a coluna da direita.

Exemplo 03:

→ Dividir o binário $(100110010)_2$ / $(10001)_2$

Divisão Binária de Números inteiros sem Sinal

↓ ↓ ↓ ↓

Dividendo 1 0 0 1 1 0 0 1 0 | 1 0 0 0 1 → **Divisor**

\- 1 0 0 0 1   **1 0 0 1 0** → **Quociente**

0 0 0 **1** **0** 0 0 1

1 0 0 0 1

0 0 0 0 0 0 → Resto

Os zeros foram acrescidos para se baixar os valores do “dividendo

Resposta: $(10010)_2$ com resto igual a zero.

Soma e Divisão com Octais

Tabela de Octal		
Decimal	**Binário**	**Octal**
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Soma Octal

→ A operação de soma de dois ou mais números em base 8 é efetuada de modo semelhante à soma decimal.

→ Vamos a alguns exemplos:

Exemplo A – Calcular $(3657)_8 + (1741)_8 = (5620)_8$:

Soma Octal					
"vai 1" →		1	1	1	
		3	6	5	7
	+	1	7	4	1
		5	6	2	0

Vamos compreender o cálculo!

Tabela de Octal		
Decimal	**Binário**	**Octal**
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Compreendendo o resultado de $(3657)_8$ **+** $(1741)_8$ **=** $(5620)_8$:

1) **7 + 1 é igual a 8,** na aritmética octal não temos o algarismo 8, o máximo é 7, **então**, emprega-se o conceito posicional, elevando o valor de 1 a próxima coluna da esquerda. Essa movimentação apresentará um resultado igual à 0 (zero) na coluna 1 e "**vai 1**" para a coluna 2 a esquerda.

Soma Octal

"vai 1" →			**1**	
	3	6	5	7
+	1	7	4	1
	5	6	2	0

Tabela de Octal		
Decimal	**Binário**	**Octal**
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Soma Octal

Compreendendo o resultado de $(3657)_8$ + $(1741)_8$ = $(5620)_8$:

2) **1 + 5 + 4 é igual a 10**, utilizaremos o mesmo conceito do passo 1, onde teremos uma soma total de 10, precisando ser descontado o valor da base 8, onde **10 - 8 = 2**, e novamente temos “**vai 1**” para a próxima coluna 3 da esquerda.

Soma Octal					
“vai 1” →			1	1	
		3	6	5	7
	+	1	7	4	1
		5	6	2	0

Soma Octal

Compreendendo o resultado de $(3657)_8 + (1741)_8 = (5620)_8$:

3) **1 + 6 + 7 é igual a 14**, novamente utilizamos o mesmo conceito do passo 1. Desta vez teremos: 14 onde será desconta a base 8, **14 - 8 = 6**, e novamente "**vai 1**" para a próxima coluna 4 da esquerda.

Tabela de Octal		
Decimal	**Binário**	**Octal**
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Soma Octal

"vai 1" →		**1**	**1**	**1**	
		3	6	5	7
	+	1	7	4	1
		5	6	2	0

Soma Octal

Compreendendo o resultado de $(3657)_8$ **+** $(1741)_8$ **=** $(5620)_8$:

4) **1 + 3 + 1 é igual a 5**, como o número está abaixo do valor da base octal a soma se mantém e não teremos "**vai 1**".

- A resultante "**N**" será igual à $(5620)_8$.

Tabela de Octal		
Decimal	**Binário**	**Octal**
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Soma Octal					
"vai 1" →		1	1	1	
		3	6	5	7
	+	1	7	4	1
		5	6	2	0

Exemplo B – Calcular $(443)_8 + (653)_8 = (1316)_8$:

→ **Primeira coluna: "3 + 3 = 6"** e não ultrapassa a base 8, **então** são somados os valores e não temos "**vai 1**" para a segunda coluna.

→ **Segunda coluna: "4 + 5 = 9"**, esta ultrapassa a base 8, então subtraímos a base 8, "**9 – 8 = 1**", para obter o resultado da segunda coluna que terá "**vai 1**" para terceira coluna.

→ **Terceira coluna: "1 + 4 + 6 = 11"**, o valor também ultrapassa base 8, então uma novamente temos uma subtração, "**11 – 8 = 3**", para obter o resultado da terceira coluna com "**vai 1**" para uma quarta coluna.

→ **Quarta coluna:** temos o "**vai 1**" e mais nada, então simplesmente descemos o "**vai 1**" para o resultado final.

→ **Resultado da soma será $(1316)_8$.**

Tabela de Octal		
Decimal	**Binário**	**Octal**
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Soma Octal

"vai 1" →	1	1		
		4	4	3
+		6	5	3
	1	3	1	6

Soma Octal

Calcular a soma Octal dos valores:

- 77657_8
- 5667_8
- **Soma = $(XXXX)_8$.**

Tabela de Octal		
Decimal	**Binário**	**Octal**
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Soma Octal

"vai 1" →

	7	7	6	5	7
+		5	6	6	7

Subtração Octal

A subtração em base 8 é relativamente mais complicada por necessitar de "**empréstimo**" ("**vem 1**") de um valor igual à base (8) obtido através do primeiro algarismo diferente de 0 (zero) existente à esquerda, por exemplo, $(7312)_8 - (3465)_8 = (3625)_8$:

Subtração Octal

							Empréstimo ou "vem 1"
		6	8			→	**2 - 4** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
			2	8		→	**0 - 6** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
				0	8	→	**2 - 5** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
Minuendo		**7**	**3**	**1**	**2**		
Subtraindo	**-**	**3**	**4**	**6**	**5**		Na **quarta coluna (cinza)** temos **6 - 3 = 3**, finalizando o cálculo em $\mathbf{(3625)_8}$.
		3	**6**	**2**	**5**		

Entendo o que o cálculo!

1. **2 – 5 não é possível**, **então,** retira-se 1 da coluna à esquerda o valor da base 8 = 8 para termos na primeira coluna **(8 + 2) - 5 = 5** e no minuendo da segunda coluna anula-se o valor de “**1**” que passa a valor “**0**”.

Subtração Octal

							Empréstimo ou “vem 1”
		6	8			→	**2 - 4** não é possível, **então** temos “**vem 1**” do **minuendo** da esquerda.
			2	8		→	**0 - 6** não é possível, **então** temos “**vem 1**” do **minuendo** da esquerda.
				0	8	→	**2 - 5** não é possível, **então** temos “**vem 1**” do **minuendo** da esquerda.
Minuendo		7	3	1	2		
Subtraindo	-	3	4	6	5		Na **quarta coluna (cinza)** temos **6 - 3 = 3**, finalizando o cálculo em **$(3625)_8$**.
		3	6	2	5		

Entendo o que o cálculo!

2. **0 – 6 não é possível, então** retira-se o valor da base 8 = 8 da coluna à esquerda, para acrescermos na segunda coluna um valor de 8 e termos **(8 + 0) – 6 e obter o resultado de 2**, já no minuendo da terceira coluna que era 3 passa a ter um valor de 2 devido ao **vem 1** para a coluna 2.

Subtração Octal							
							Empréstimo ou "vem 1"
		6	8			→	**2 - 4** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
			2	8		→	**0 - 6** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
				0	8	→	**2 - 5** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
Minuendo		7	3	1	2		
Subtraindo	-	3	4	6	5		Na **quarta coluna (cinza)** temos **6 - 3 = 3**, finalizando o cálculo em **$(3625)_8$**.
		3	6	2	5		

Entendo o que o cálculo!

3. **2 – 4 não é possível, então,** faz-se um "**vem 1**" da coluna à esquerda, para termos na terceira coluna o valor da base 8 = 8 e termos uma equação de **(8 + 2) – 4 = 6**, com o "**vem 1**" o minuendo da quarta coluna passa a ter um **valor = 6 ou (7 – 1)**.

Subtração Octal

							Empréstimo ou "vem 1"
		6	8			→	**2 - 4** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
			2	8		→	**0 - 6** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
				0	8	→	**2 - 5** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
Minuendo		7	3	1	2		
Subtraindo	-	3	4	6	5		Na **quarta coluna (cinza)** temos **6 - 3 = 3**, finalizando o cálculo em **$(3625)_8$**.
		3	6	2	5		

Entendo o que o cálculo!

4. Na quarta coluna temos agora **6** em vez de **7** no minuendo devido ao "**vem 1**" para a coluna 3. O valo de "**6**" é maior que o subtraindo "**3**" não acontecendo "**vem 1**" e apenas fazemos a subtração **finalizando o processo** e obtendo o valor de $(3\,6\,2\,5\,)_8$.

Subtração Octal

							Empréstimo ou "vem 1"
		6	8			→	**2 - 4** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
			2	8		→	**0 - 6** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
				0	8	→	**2 - 5** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
Minuendo		**7**	**3**	**1**	**2**		
Subtraindo	**-**	**3**	**4**	**6**	**5**		Na **quarta coluna (cinza)** temos **6 - 3 = 3**, finalizando o cálculo em $\mathbf{(3625)_8}$.
		3	**6**	**2**	**5**		

Vamos a outro exemplo, $5113_8 - 3775_8 = 1116_8$:

Subtração Octal						
						Empréstimo ou "vem 1"
	4	8			→	**0 - 7** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
		0	8		→	**0 - 7** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
			0	8	→	**3 - 5** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
Minuendo	~~5~~	~~1~~	~~1~~	3		
Subtraindo -	3	7	7	5		Na **quarta coluna (cinza)** temos **4 - 3 = 1**, finalizando o cálculo em $\mathbf{(1216)_8}$.
	1	2	1	6		

Entendo o que foi calculado

1. **3 – 5 não é possível**, **então** temos vem 1 da coluna à esquerda e termos na primeira coluna **(8 + 3) - 5 = 6** e no minuendo da segunda coluna um **valor = 0 (1 - 1) devido ao vem 1 para o coluna 2**.

Subtração Octal

						Empréstimo ou "vem 1"
	4	8			→	**0 - 7** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
		0	8		→	**0 - 7** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
			0	8	→	**3 - 5** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
Minuendo	5	1	1	3		
Subtraindo -	3	7	7	5		Na **quarta coluna (cinza)** temos **4 - 3 = 1**, finalizando o cálculo em **$(1216)_8$**.
	1	2	1	6		

Entendo o que foi calculado

2. **0 – 7 não é possível, então**, novamente temos vem 1 da coluna à esquerda para a da direita, segunda coluna, e ter-se uma equação de **(8 + 0) – 7 = 1** e no minuendo da terceira coluna um **valor = 0 (1 - 1) devido ao vem 1 para o coluna 2**.

Subtração Octal							
							Empréstimo ou "vem 1"
		4	8			→	**0 - 7** não é possível, **então** temos **"vem 1"** do **minuendo** da esquerda.
			0	8		→	**0 - 7** não é possível, **então** temos **"vem 1"** do **minuendo** da esquerda.
				0	8	→	**3 - 5** não é possível, **então** temos **"vem 1"** do **minuendo** da esquerda.
Minuendo		5	1	1	3		
Subtraindo	-	3	7	7	5		Na **quarta coluna (cinza)** temos **4 - 3 = 1**, finalizando o cálculo em **$(1216)_8$**.
		1	2	1	6		

Entendo o que foi calculado

3. **0 – 7 não é possível, então**, novamente temos vem 1 da coluna à esquerda para a da direita, terceira coluna **(8 + 0) – 7 = 1** e no minuendo da quarta coluna um **valor = 4 (5 - 1) devido ao vem 1 para o coluna 3**.

Subtração Octal

							Empréstimo ou "vem 1"
		4	8			→	**0 - 7** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
			0	8		→	**0 - 7** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
				0	8	→	**3 - 5** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
Minuendo		5	1	1	3		
Subtraindo	-	3	7	7	5		Na **quarta coluna (cinza)** temos **4 - 3 = 1**, finalizando o cálculo em **$(1216)_8$**.
		1	2	1	6		

Entendo o que foi calculado

4. Na quarta coluna não termos vem 1 e apenas resolve-se a equação de **4 - 3 = 1, finalizando o processo** e obter uma resultante de "**N**" de $(1\,2\,1\,6)_8$.

Subtração Octal						
						Empréstimo ou "vem 1"
	4	8			→	**0 - 7** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
		0	8		→	**0 - 7** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
			0	8	→	**3 - 5** não é possível, **então** temos "**vem 1**" do **minuendo** da esquerda.
Minuendo	5	1	1	3		
Subtraindo -	3	7	7	5		Na **quarta coluna (cinza)** temos **4 - 3 = 1**, finalizando o cálculo em $\mathbf{(1216)_8}$.
	1	2	1	6		

Subtração Octal

TREINO: Fazer a subtração de $(65133)_8 - (43715)_8$

Subtração Octal

							Empréstimo ou "vem 1"
						→	Não houve "vem 1".
		4	8			→	(8+1)-7=2 -> Resultado da terceira coluna
				2	8	→	(8+3)-5=6 -> Resultado da primeira coluna.
Minuendo	6	~~5~~	1	~~3~~	3		
Subtraindo -	4	3	7	1	5		Resultado é igual a 21216_8.
	2	1	2	1	6		

Bibliografia do Curso

Bibliografia Básica

TANENBAUM, A. S. Organização estruturada de computadores. 6. ed. São Paulo: Pearson Prentice Hall, 2013 (e-book).

MONTEIRO, M. A. Introdução à organização de computadores. 4. ed. Rio de Janeiro: LTC, 2002.

STALLINGS, W. Arquitetura e organização de computadores: projeto para o desempenho. 5. ed. São Paulo: Prentice-Hall, 2002.

Bibliografia Complementar

CORRÊA, A. G. D. [org.]. Organização e arquitetura de computadores. São Paulo: Pearson Education do Brasil, 2016 (e-book).

DELGADO, J.; RIBEIRO, C. Arquitetura de computadores. 5. ed. Rio de Janeiro: LTC, 2017 (e-book).

PAIXÃO, R. R. Arquitetura de computadores - PCs. São Paulo: Érica, 2014 (e-book).

WEBER, R. F. Fundamentos de arquitetura de computadores. 4. ed. Porto Alegre: Bookman, 2012 (e-book).

WIDMER, N. S.; MOSS, G. L.; TOCCI, R. J. Sistemas digitais: princípios e aplicações. 12. ed. São Paulo: Pearson Education do Brasil, 2018 (e-book).

Conteúdo elaborado por:

Prof. Ms. Celso Candido
celsoc@unicid.edu.br

Fim da Apresentação