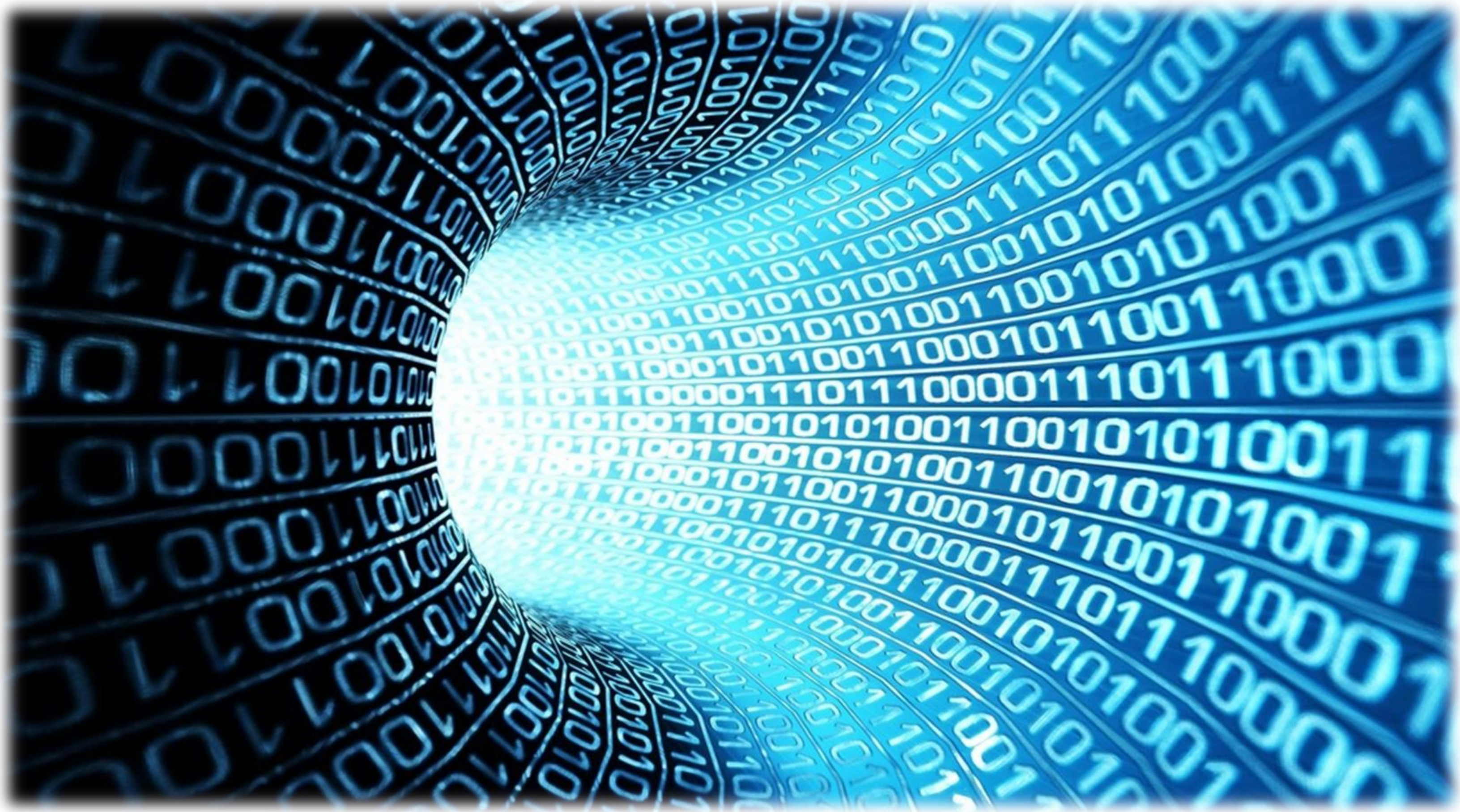
Organização e Arquitetura de Computadores



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.

Na a*ula 04 e 05 vimos as formas de cálculos para a “Conversão de Bases e Aritmética I (soma e subtração)”*.

Hoje analisaremos:

☺ A vírgula binária (MSB e LSB),

☺ A representação de números binários com **sinal** negativo e positivo”;

☺ As formas de “complementos de 1 e 2”.

Binário Mais e Menos Significativo

Valores Posicionais →	**8**	**4**	**2**	**1**		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
Expoente de "b^" →	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
	↓	↓	↓	↓		↓	↓	↓
Valor Binário →	1	0	1	1	•	1	0	1
	↑				↑			↑
	MSB				**Vírgula Binária**			**LSB**
Valores posicionais de Base 2 ⇒ Binário com vírgula								

Fonte: Adaptada pelo autor (2023)

→ **MSB (*Most Significant Bit*):** Posição mais a esquerda do *bit* binário **mais significativo**.

→ **LSB (*Less Significant Bit*):** Posição mais a direita do *bit* binário **menos significativo**.

O **ponto**, considerado a vírgula binária, possui a mesma função da vírgula decimal: separar a parte **inteira** do número de sua parte **fracionária**.

Valores Posicionais →	8	4	2	1		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
Expoente de "b^" →	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
	↓	↓	↓	↓		↓	↓	↓
Valor Binário →	1	0	1	1	•	1	0	1
	↑				↑			↑
	MSB	Vírgula Binária						LSB
Valores posicionais de Base 2 ⇒ Binário com vírgula								

Fonte: Adaptada pelo autor (2023)

→ À esquerda da vírgula binária estão as potências de 2 com expoente positivo (+).

→ À direita da vírgula binária estão as potências de 2 com expoente negativo (-) com os valores antes da vírgula/ponto.

→ **IMPORTANTE**: Não existe expoente **zero negativo**.

Para descobrir, precisamos somar os produtos do valor de cada dígito (0 ou 1) pelo seu respectivo valor posicional (**peso**).

Passo 1 – Analisar os valores posicionais:

Valores Posicionais →	8	4	2	1		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
Expoente de "b^" →	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
	↓	↓	↓	↓		↓	↓	↓
Valor Binário →	1	0	1	1	•	1	0	1
	↑				↑			↑
	MSB				Vírgula Binária			LSB
Valores posicionais de Base 2 ⇒ Binário com vírgula								

Fonte: Adaptada pelo autor (2023)

→ Valores positivo ⇒ 2^3 = 8 | 2^2 = 4 | 2^1 = 2 | 2^0 = 1

→ Valores negativos ⇒ 2^{-1} = ½ = 0,5 | 2^{-2} = ¼ = 0,25 | 2^{-3} = ⅛ = 0,125

Entendo o que foi feito:

Ao valores foram convertidos para a linguagem decimal, $\mathbf{b_{10}}$, com base nos expoentes positivo e negativos.

Passo 2 – Calcular:

Valores Posicionais ⟶	**8**	**4**	**2**	**1**		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
Expoente de "b^" ⟶	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
	↓	↓	↓	↓		↓	↓	↓
Valor Binário ⟶	1	0	1	1	•	1	0	1
	↑				↑			↑
	MSB				Vírgula Binária			LSB
Valores posicionais de Base 2 ⇒ Binário com vírgula								

Fonte: Adaptada pelo autor (2023)

⇒ **N = n * b^**

⇒ $N = (1011.101)_2 = ((1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (1 * 2^0)) + ((1 * 2^{-1}) + (0 * 2^{-2}) + (1 * 2^{-3})) =$

⇒ $N = (1011.101)_2 = (8 + 0 + 2 + 1) + (0{,}5 + 0 + 0{,}125) =$

⇒ $N = (1011,.101)_2 = 11 + 0{,}625 = 11{,}625$

⇒ $\mathbf{N = (1011.101)_2 = (11{,}625)_{10}}$

Relembrando a Soma

Na soma de valores binários **com vírgula é igual** ao processo de soma sem vírgula com as mesmas regras da soma binária.

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.

Somar o binário de base 2 com vírgula => $(11,011)_2 + (10,110)_2$											
Transporte ou Vai um				1	1	1		1			Decimal
Parcelas da Soma					1	1	,	0	1	1	$3,375_{10}$
Transporte ou Vai um					0	0		1			
Parcelas da Soma	+				1	0	,	1	1	0	$2,75_{10}$
Somatória				1	1	0	,	0	0	1	$6,125_{10}$

A soma de $(11,011)_2 + (10,110)_2 = (110,001)_2$ ou $(6,125)_{10}$

Representação de números binários com Sinal negativo / positivo.

Como sabemos, os computadores e as calculadoras trabalham com **números negativos e positivos**, necessários para os cálculos que engloba os valores "Reais".

Para que a leitura negativa ou positiva acrescenta-se a esquerda do binário **mais** um *bit*, denominado ***bit* de sinal**, onde:

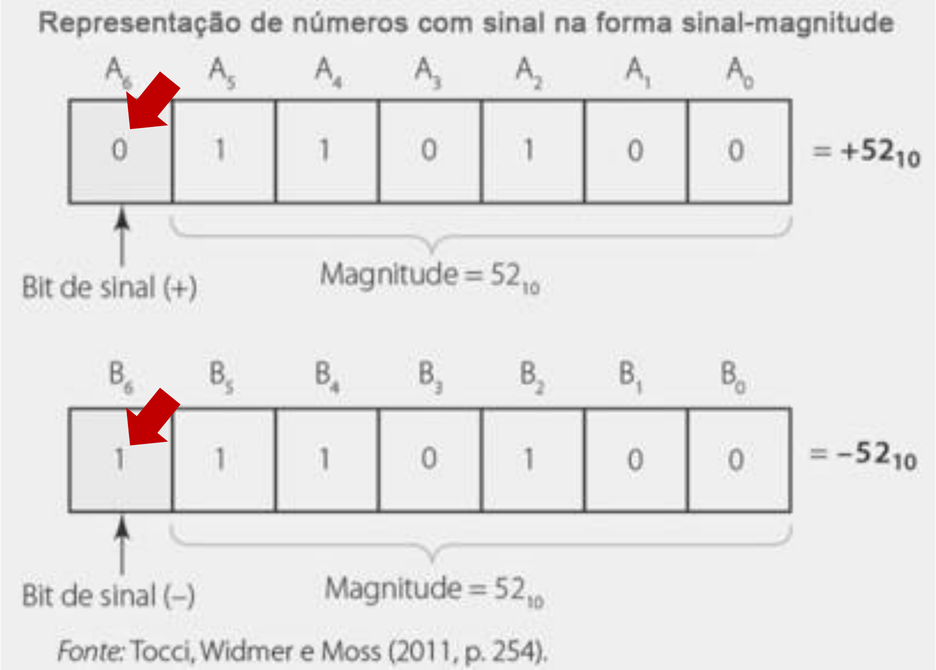
- Se o valor for "**0**" no ***bit* de sinal**, representará um **valor positivo**;
- Se o valor for "**1**" no ***bit* de sinal**, representará um **valor negativo**.

Representação de números com sinal na forma sinal-magnitude

Fonte: Tocci, Widmer e Moss (2011, p. 254).

Exemplo:

1. O **registrador A** contém os *bits* **0**110100$_2$, o ***bit* 0** mais à esquerda do **MSB**, "**A$_6$**" é um *bit* de sinal **positivo** ao binário (110100)$_2$ que está sendo **armazenado pelo registrador A**.
2. Ao contrário, no **registrador B** o armazenamento binário assume uma posição de valor negativa devido ao ***bit* 1** a esquerda do **MSB**, "**B$_6$**".



Chamamos esse modelo de representação de "Sistema Sinal-Magnitude".

Complementos de 1 e 2 de um binário

Na aritmética binária para a representação de números binários com sinal, os cálculos são elaborados através dos "**Sistemas de Complementos de 2 e 1**", sendo o de complemento de 2 **o mais utilizado** e o de complemento de 1 o mais simples de calcular.

Complemento de 1:

→ O complemento de 1 de um valor binário é obtido pela substituição de cada *bit* do número binário através de seu complemento, se for "**0**" troca-se para "**1**" e **vice-versa**, por exemplo:

$(1\ 0\ 1\ 1\ 0\ 1)_2$	=> Número binário **original**
↓ ↓ ↓ ↓ ↓ ↓	=> **Onde** for 0 (zero) no original passa a ser 1.
$(0\ 1\ 0\ 0\ 1\ 0)_2$	=> Complementando cada bit para obter o **complemento de 1**

Complemento de 2:

→ Já, o complemento de 2 de um número binário é mais trabalhoso que o complemento de 1, onde:

1. Inicialmente faz-se o complemento de 1 do número binário.
2. Após o complemento de 1 **adiciona-se 1 *bit* na posição do *bit* menos significativo**, mais a direita, para se somado e completar o cálculo.

Exemplo de Complemento de 2 do Binário 101101_2	
$(1\ 0\ 1\ 1\ 0\ 1)_2$	=> Número binário **original**
$(0\ 1\ 0\ 0\ 1\ 0)_2$	=> Complementando cada bit para obter o **complemento de 1**
+ 1	=> **Adicionando 1** para obter o complemento de 2
$(0\ 1\ 0\ 0\ 1\ 1)_2$	=> **Complemento de 2** do número binário original

Representação de números com sinal em complemento de 2

A representação de números com sinal no sistema de complemento de 2 trabalha os cálculos da seguinte forma:

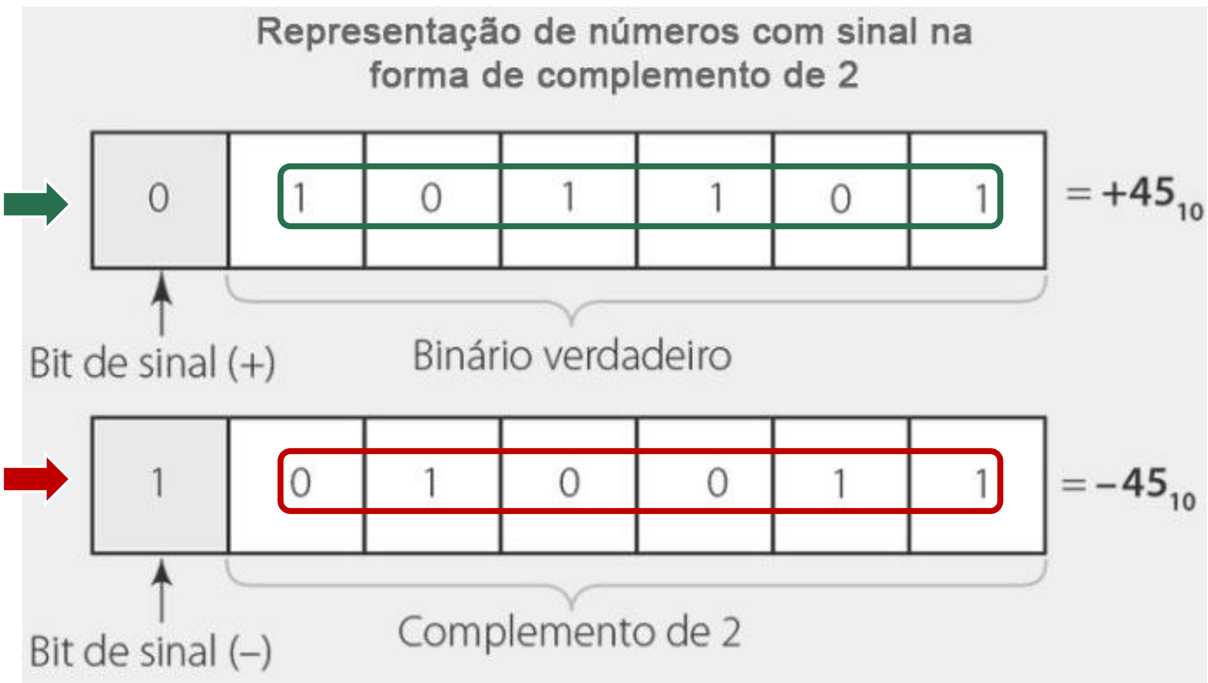
→ **Se o número for positivo:** a magnitude é representada na forma binária direta, e **um *bit* de sinal 0** será colocado a esquerda do MSB (binário mais significativo).

→ **Se o número for negativo:** a magnitude é representada na forma do complemento de 2, e **um *bit* de sinal 1** será colocado a esquerda do MSB (binário mais significativo).

⇒ Caso o complemento de 2 seja positivo receberá ***um bit de sinal 0*** a esquerda do MSB.

VAMOS ANALISAR NA FORMA GRÁFICA

$(101101)_2$	=> Número binário **original.**
$(010010)_2$	=> Complementando cada bit para obter o **complemento de 1.**
+ 1	=> **Adicionando 1** para obter o complemento de 2.
$(010011)_2$	=> **Complemento de 2** do número binário original.



Fonte: Tocci, Widmer e Moss (2011, p. 255).

Exemplo de cálculo de sinal em complemento de 2 para o decimal $(+9)_{10}$

Modelo 01 – Complemento de 2 Positivo:

1. Passa-se o decimal $(+9)_{10}$ para o formato binário de **8 *bits*** ou $(00001001)_2$;
2. Faz-se o complemento de 2 obtendo como resultado o valor de $\mathbf{(11110111)_2}$ **ou** $\mathbf{(-9)_{10}}$.

Processo descrito pelo Modelo 01	
$(0\ 0\ 0\ 0\ 1\ 0\ 0\ 1)_2$	=> Binário de 8 bits representando $\mathbf{+9_{10}}$ devido ao MSB
$(1\ 1\ 1\ 1\ 0\ 1\ 1\ 0)_2$	=> Obtendo o **complemento de 1** igual a $\mathbf{-9_{10}}$ devido ao MSB
+ 1	=> **Adicionando 1** para obter o complemento de 2
$\mathbf{(1\ 1\ 1\ 1\ 0\ 1\ 1\ 1)_2}$	=> **Complemento de 2** de $\mathbf{+9_{10}}$ será $\mathbf{-9_{10}}$ devido ao MSB

Exemplo de cálculo de sinal em complemento de 2 para o número decimal $(-9)_{10}$

Modelo 02 – Complemento de 2 Negativo:

1. Iniciar pelo resultado da representação negativa, **(-9_{10} ou 11110111_2)**, e obter o valor inverso do complemento de 2, que é **00001001_2 ou $+9_{10}$**.

   $\Rightarrow$ Útil para verificar se o resultado está correto.

Processo descrito pelo Modelo 02	
$(1\ 1\ 1\ 1\ 0\ 1\ 1\ 1)_2$	=> Binário de 8 bits representando **-9_{10}** devido ao MSB
$(0\ 0\ 0\ 0\ 1\ 0\ 0\ 0)_2$	=> Obtendo o **complemento de 1** igual a **$+9_{10}$** devido ao MSB
+ 1	=> **Adicionando 1** para obter a **negação** do complemento de 2
$(0\ 0\ 0\ 0\ 1\ 0\ 0\ 1)_2$	=> **Complemento de 2** de **-9_{10}** será **$+9_{10}$** devido ao MSB

Binário com sinal usando complemento de 2

→ O sistema de complemento de 2, no geral, se trata de um processo simples de transformação como no de complemento de 1, **mas requer mais atenção e raciocínio lógico**.

→ Poderão existir alguns casos especiais em complemento de 2 que necessitará de um pouco mais de raciocínio lógico, nesse sentido, sempre que um número com sinal **tiver o valor de 1 em seu *bit* de sinal** e **todos** os *bits* de **magnitude forem = 0**, sua equivalência em decimal será *-2 com expoente N* ou ***-2^N^***, onde ***N*** será o número de *bits* da magnitude, por exemplo:

- $1000_2 = -2^3 = -8_{10} \Rightarrow$ Sinal = *1* e magnitude "*000*", o expoente de N = "*3*".
- $10000_2 = -2^4 = -16_{10} \Rightarrow$ Sinal = *1* e magnitude "*0000*", o expoente de N = "*4*".
- $100000_2 = -2^5 = -32_{10} \Rightarrow$ Sinal = *1* e magnitude "*00000*", o expoente de N = "*5*".
- $1000000_2 = -2^6 = -64_{10} \Rightarrow$ Sinal = *1* e magnitude "*000000*", o expoente de N = "*6*".
- $10000000_2 = -2^7 = -128_{10} \Rightarrow$ Sinal = *1* e magnitude "*0000000*", o expoente de N = "*7*".

→ A tabela, ao lado, procura relacionar todos os números com sinal que podem ser representados com **quatro *bits*** utilizando o sistema de complemento de 2 com **três *bits* de magnitude ou N = 3**.

→ A sequência se inicia em $\mathbf{-2^N = -2^3}$ $= -8_{10} = 1000_2$ e;

→ Termina em $\mathbf{+2^N - 1}$ $= +2^3 - 1 = +7_{10} = 0111_2$.

IMPORTANTE:

Não se esqueça, as magnitudes abrangem além de "**N = 3**", conforme apresentado no slide 18.

$1000_2 = -2^3 = -8_{10}$

$10000_2 = -2^4 = -16_{10}$

$100000_2 = -2^5 = -32_1$

$1000000_2 = -2^6 = -64_{10}$

$10000000_2 = -2^7 = -128_{10}$

Tabela: Adaptada pelo autor.
Fonte: Tocci, Widmer e Moss (2011, p. 258).

Adição e Subtração no sistema de complemento de 2

As próximas informações demonstram e analisam como as **operações booleanas de adição e subtração** são realizadas pelas máquinas digitais que utilizam a representação em complemento de 2 para números negativos.

Será impossível o uso de uma calculadora digital, sendo necessário o uso de lápis e papel para os cálculos.

TABELA ou MULETA?

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

A adição no sistema de complemento de 2 **observa** como o ***bit* de sinal de cada número** realizará a operação da "**soma**" binária **subdivididos em quatro casos.**

» **Caso I:** Adição de **dois números positivos**, o cálculo será feito diretamente pela tabela da soma:

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.

+9	➡	0	1001_2	9_{10}	1º parcela de 4 *bits*
+4	➡	0	0100_2	4_{10}	2º parcela de 4 *bits*
+13	➡	0	$\mathbf{1101_2}$	13_{10}	Soma = +13
					bits de sinal

→ Os *bits* de sinal da 1º e da 2º linha são ambos 0.

→ O *bit* de sinal da soma também é 0, 3º linha, **indicando um valor positivo**.

→ As linha 1 e 2 possuem **4 *bits***, e necessário ao sistema do complemento de 2 para o cálculo.

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

» **Caso II:** Com **um números positivos e outro menor *e* negativo**, para o exemplo, temos uma **adição de $+9_{10}$ com -4_{10}**, onde o **-4_{10}** representa um valor de complemento de 2 negativo que será convertido para positivo via tabela e obtendo o valor de $+4_{10} = (0100)_2$.

	1				
+9			1001	9_{10}	1º parcela de 4 *bits*
-4			1100	-4_{10}	2º parcela de 4 *bits*
+5	***1***	**0**	**0101**	$+5_{10}$	Soma = +5
					A soma é positiva, então o bits de sinal é 0.
	→ Desconsiderar o ***vai 1 da soma 1 + 1***. → O resultado da soma será 00101_2 ou $+5_{10}$.				

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.**

→ Neste exemplo o *bit* de sinal da 2º linha é **1** (negativo), e que participa do processo de soma.

→ Um *carry* (***X***) é gerado na última posição da soma, mas deve ser desconsiderado.

→ Assim, a soma final é 00101_2, equivalente a $+5_{10}$.

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

» **Caso III:** Com **um número positivos e outro maior *e* negativo**, para o exemplo a soma de -9_{10} **com** $+4_{10}$:

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.

			1		
-9	➡	1	0111	-9_{10}	1º parcela de 4 *bits*
+4	➡	0	0100	4_{10}	2º parcela de 4 *bits*
-5	➡	1	1011	-5_{10}	**Soma = -5**
					bits de sinal **negativo**, gerou um ***carry 1***

Processo descrito pelo item 1 e 2 (slide 17)	
$(0\,1\,0\,0\,1)_2$	=> Binário de 4 bits representando $\mathbf{+9_{10}}$ devido ao MSB.
$(1\,0\,1\,1\,0)_2$	=> Obtendo o **complemento de 1** igual a $\mathbf{-9_{10}}$ devido ao MSB.
+ **1**	=> **Adicionando 1** para obter o complemento de 2.
$(1\,0\,1\,1\,1)_2$	=> **Complemento de 2** de $\mathbf{+9_{10}}$ será $\mathbf{-9_{10}}$ devido ao MSB.

» **Treino – Caso I: Somar $+8_{10}$ com $+6_{10}$**

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.

+8			**1**	**0**	**0**	**0**		1º parcela de 4 *bits*
+6			**0**	**1**	**1**	**0**		2º parcela de 4 *bits*
								Soma =
								bits de sinal

» **Treino – Caso II: Somar $+7_{10}$ com -6_{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.

+7		**0**	**0**	**1**	**1**	**1**		1º parcela de 4 *bits*
-6		**1**	**1**	**0**	**1**	**0**		2º parcela de 4 *bits*
								Soma =
								bits de sinal

» **Treino – Caso III: Somar -7_{10} com $+6_{10}$**

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.

-7				1 0 0 1		1º parcela de 4 *bits*
+6				0 1 1 0		2º parcela de 4 *bits*
						Soma =
						bits de sinal

Obter o complemento de 1 e 2 do resultado da Soma	
$(1\ 1\ 1\ 1)_2$	=> Binário de 4 bits representando **-1_{10}** devido ao MSB.
	=> Obtendo o **complemento de 1** igual a **$+1_{10}$** devido ao MSB.
	=> **Adicionando 1** para obter o complemento de 2.
	=> **Complemento de 2** de serão devido ao MSB.

» **Treino – Complemento de 1 para complemento de 2:**

	Processo descrito pelo Modelo 01 (slide 17)	
0	$(1\ 1\ 1\ 0)_2$	=> Binário de 4 bits representando **$+14_{10}$** devido ao MSB.
	$(0\ 0\ 0\ 1)_2$	=> Obtendo o **complemento de 1** igual a **$+14_{10}$** devido ao MSB.
	+ **1**	=> **Adicionando 1** para obter o complemento de 2.
1	$(0\ 0\ 1\ 0)_2$	=> **Complemento de 2** de **$+14_{10}$** será **-14_{10}** devido ao MSB.

- Temos na soma de **-9_{10} com $+4_{10}$ um *bit* de sinal 1**, negativo, obtendo um resultado de ***-5***.
- A soma é negativa, estando em complemento de 2, representado pelos últimos **quatro *bits* (1011)$_2$** .

			1		
-9	➡	1	0111	-9_{10}	1º parcela de 4 *bits*
+4	➡	0	0100	4_{10}	2º parcela de 4 *bits*
-5	➡	1	1011	-5_{10}	**Soma = -5**
					bits de sinal **negativo**, gerou um ***carry 1***

Processo descrito pelo item 1 e 2 (slide 17)	
$(0\,1\,0\,0\,1)_2$	=> Binário de 4 bits representando **$+9_{10}$** devido ao MSB.
$(1\,0\,1\,1\,0)_2$	=> Obtendo o **complemento de 1** igual a **-9_{10}** devido ao MSB.
+ 1	=> **Adicionando 1** para obter o complemento de 2.
$(1\,0\,1\,1\,1)_2$	=> **Complemento de 2** de **$+9_{10}$** será **-9_{10}** devido ao MSB.

- Para verificar se o resultado está correto, foi calculado o completo de dois, conforme apresentado no slide 17, onde o seu valor **positivo**, conforme tabela de "Binário com Sinal usando Complemento de 2" é igual à $(00101)_2$ ou +5.

» **Caso IV:** Com **dois número negativos**, vamos considerar a seguinte adição para o exemplo, **-9_{10} com -4_{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.

	1	1	1					
-9	➡	1	0	1	1	1	-9_{10}	1º parcela de 4 *bits*
-4	➡	1 + 0	1 + 1	1	0	0	-4_{10}	2º parcela de 4 *bits*
-13	*1*	1	0	0	1	1	-13_{10}	Soma = -13_{10}
								bits de sinal gerando um ***carry* 1**
	→ **Desconsiderar o *carry* 1.** → O resultado da soma será 10011_2 ou -13_{10}.							

→ Temos novamente um resultado negativo.

→ A sua forma é de complemento de 2 com um *bit* de sinal 1, negativo no MSB.

→ Se a negação de complemento de 2 for efetuada teremos como resultado $(01101)_2 = +13_{10}$.

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

» **Treino – Caso IV:**

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.

	1	1		1	1		
-5	➡	1	1	0	1	1	1º parcela de 4 *bits*
-7	➡	1	1	0+1	0+0	1	2º parcela de 4 *bits*
-12	*1*	1	0	1	0	0	Soma = -12_{10}
							bits de sinal gerando um ***carry 1***
	→ **Desconsiderar o *carry 1*.** → O resultado da soma será (1 0100)$_2$ = -12_{10}						

→ Temos novamente um resultado negativo.

→ A sua forma é de complemento de 2 com um *bit* de sinal 1, negativo no MSB.

→ Se a negação de complemento de 2 for efetuada teremos como resultado $(01101)_2 = +13_{10}$.

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

De positivo para negativo e vice-versa

→ **Treino: $+7_{10}$ para -7_{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.

0	0	1	1	1	$= +7_{10}$	Igualando o valor a 4 bits.
	1	0	0	0		Complemento de 1 de +7
			+	1		Acrescentando +1
1	1	0	0	1	$= -7_{10}$	Obtendo o complemento de 2

O valor de $+7_{10}$ ou 0111_2, em complemento de 1 passa a ser 1000_2, para se chegar ao complemento, foi acrescido **+1** a primeira coluna do complemento de 1, como resultado de **1001_2 ou -7_{10}**.

O **bit de sinal de +7 é igual "0"** por ser positivo e o **bit de sinal de -7 é igual "1"** por ser negativo.

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

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.

De positivo para negativo e vice-versa

→ **Treino: $+4_{10}$ para -4_{10}:**

	0	1	0	0	$= +4_{10}$	Igualando o valor a 4 bits.
	1	0	1	1	C1	Complemento de 1 de
			+	1	C2	Acrescentar +1 a primeira coluna
	1	1	0	0	$= -4_{10}$	Obtendo o complemento de 2

O valor de $+4_{10}$ ou 0100_2, em complemento de 1 passa a ser 1011_2, para se chegar ao complemento, foi acrescido **+1** a primeira coluna do complemento de 1, como resultado de **1011_2 ou -4_{10}**.

O **bit de sinal de +4 é igual "0"** por ser **positivo** e o **bit de sinal de -4 é igual "1"** por ser **negativo**.

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

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.

De positivo para negativo e vice-versa

→ **Treino: -6_{10} para $+6_{10}$:**

1	1	0	1	0	$= -6_{10}$	Igualando o valor a 4 bits.
	0	1	0	1	0101_2	Complemento de 1 de
			+	1		Acrescentar +1 a primeira coluna
0	0	1	1	0	$= +6_{10}$	Obtendo o complemento de 2

O valor de -6_{10} ou 1010_2, em complemento de 1 passa a ser 0101_2, para se chegar ao complemento, foi acrescido **+1** a primeira coluna do complemento de 1, como resultado de **0110_2 ou -6_{10}**.

O **bit de sinal de -6 é igual "1"** por ser **negativo** e o **bit de sinal de +6 é igual "0"** por ser **positivo**.

No complemento de 1 apenas inverto os bits do binário original.

No complemento de 2 após a inversão pelo complemento de se obtém um **processo de negação do número ou valor do binário original.**

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

SUBTRAÇÃO NO SISTEMA DE COMPLEMENTO DE 2

Subtração em Complemento de 2

→ A operação de subtração no sistema de complemento de 2 **envolve as regras da soma**.

→ Por apresentar característica parecidas a **soma**, o sistema de complemento de 2 é **considerado como um dos métodos mais utilizados na subtração**.

→ Esse processo permite que a adição e a subtração possam ser realizadas pelas mesmas regras.

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.

As regra da soma binária se aplica a subtração
no sistema de complemento de 2.

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.

→ Ao efetuar a subtração de um número binário é necessário seguir alguns procedimentos:

1. **Faz-se a operação de negação do subtraendo –** A intenção é alterar o **subtraendo** para o valor equivalente **com sinal oposto**.
2. **Adiciona o número obtido ao minuendo –** O resultado da adição representará a **diferença entre o subtraendo e o minuendo**.

→ Para que a operação seja bem sucedida é necessário que os dois **binários booleanos** possuam o mesmo número de *bits* em suas representações, e **devendo acontecer para todas** as operações aritméticas em complemento de 2.

→ **Exemplo:** Calcular a operação onde: **+4** (subtraendo) ou (**0** 0100)$_2$, onde o "**0**" mais a esquerda, *bit* de sinal, representa um valor positivo ⊕, que **deverá ser subtraído** de **+9** (minuendo) ou (**0** 1001)$_2$, também com *bit* de sinal = "**0**".

→ Antes de iniciar precisamos **negar** (⊖) o **subtraendo,** (**0** 0100)$_2$, para obter seu negativo = (**1** 1100)$_2$, para isso será calculado o complemento de 1 e 2.

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.

1	0	1001	$(+9)_{10}$	**Minuendo** de 4 *bits* *(0 1001)$_2$*
	1	1100	$(-4)_{10}$	**Subtraendo** de 4 *bits* *(1 1100)$_2$*
X	0	0101	$+5_{10}$	Soma = **+5**
				bits de sinal gerando um ***carry 1***

→ Desconsiderar o ***carry 1***.

→ O resultado da soma será 00101_2 ou $+5_{10}$.

Entendo:

→ Ao negar-se o **subtraendo** pelo cálculo em complemento de 2, torna-se igual a $(-4)_{10}$.

→ Após a negação faz-se a soma $(+9)_{10}$ com $(-4)_{10}$, situação já vista no "**Caso II**".

→ Para o **próximo** exemplo, agora teremos $(+9)_{10}$ ou $(01001)_2$, sendo **subtraído** de $(-4)_{10}$ ou $(1\ 1100)_2$.

→ Nesta operação de soma e subtração precisamos **negar** o $(+9)_{10}$, subtraendo, para obter $(-9)_{10}$ ou $(1\ 0111)_2$, através do complemento de 1 e 2, e somar com o minuendo $(-4)_{10}$.

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.

1	1	1+1 1 0 0	$(-4)_{10}$	**Minuendo** de 4 *bits* $(11100)_2$
	1	0+0 1 1 1	$(-9)_{10}$	**Subtraendo** de 4 *bits* $(10111)_2$
X	1	0 0 1 1	-13_{10}	Soma = -13 = $(1\ 0011)_2$
				bits de sinal gerando um ***carry 1***

→ Desconsiderar o ***carry 1***.
→ O resultado da soma será 10011_2 ou -13_{10}.

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