Sistemas de Numeração

Nesta unidade, apresentaremos os sistemas de numeração que auxiliam o estudo das técnicas digitais e sistemas de computação.

A partir do sistema decimal, estudaremos os sistemas: binário, octal, hexadecimal e o método de conversão entre esses sistemas.

Para assimilar os conteúdos desta lição, é necessário que você conheça perfeitamente o sistema decimal.

Sistemas de Numeração

Dos sistemas de numeração existentes, os mais utilizados são o **decimal**, o **binário** e o **hexadecimal.**

Sistema de Numeração Decimal

O sistema de numeração decimal utiliza dez símbolos / algarismos para a sua codificação. São eles: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**. Assim, a **base** desse sistema é **dez**.

Com esses dez algarismos, é possível representar qualquer grandeza numérica graças à características do **valor de posição**. Desse modo, temos:

- Números que representam as **unidades**: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Números que representam as **dezenas**: 10, 11, 12, 13, 14, 15 ...; nos quais o número da posição **1** indica uma **dezena** e o outro dígito, a unidade.
- Números que representam as **centenas**: 110, 111, 112, 113, 114, 115, 116 ..., nos quais o valor de posição **1** indica a **centena**, seguida pela dezena e pela unidade. Assim, por exemplo, o número 385 indica:

centenas		Dezenas		Unidades ;
↓		↓		↓
3	+	8	+	5
↓		↓		↓
3 . **100**	+	8 . **10**	+	5 . **1**
↓		↓		↓
300	+	80	+	5

ou seja:

= 385

O número 385 também pode ser expresso por meio de uma **potência de base dez**:

3		8		5
↓		↓		↓
3 . **100**		8 . **10**		5 . **1**
↓		↓		↓
$3 . 10^2$	+	$8 . 10^1$	+	$5 . 10^0$

ou seja:

= 385

Observação:
A potência da base 10 indica o valor da posição do número.

Sistema de Numeração Binária

O sistema de numeração binária é empregado em circuitos lógicos digitais. Esse sistema possui apenas dois algarismos: **0** e **1**. Por isso, sua base é **dois** (dois dígitos).

Cada dígito ou algarismo binário é chamado de **bit** (do inglês "binary digit", ou seja dígito binário). Um bit é, pois, a menor unidade de informação nos circuitos digitais.

A tabela a seguir mostra a correspondência entre números dos sistemas decimal e binário.

Decimal	Binário
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

Decimal	Binário
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001
18	10010
...	...

Empregando a propriedade do valor de posição do dígito, podemos representar qualquer valor numérico com os dígitos **0** e **1**.

Como a base da numeração binária é 2, o valor de posição é dado pelas potências de base 2, como mostra o quadro a seguir.

Potências de base 2 ➔	2^4	2^3	2^2	2^1	2^0
Valor de posição ➔	16	8	4	2	1
Número Binário ➔	1	0	0	1	1

O valor da posição é indicado pelo **expoente da base** do sistema numérico. Esse valor aumenta da direita para a esquerda. O valor da posição do bit **mais significativo** (de maior valor) será a base elevada a n -1 (n = número de dígitos).

Por exemplo, 101011 é um número binário de 6 bits. Ao aplicar a fórmula (n^{n-1}), temos 6 - 1 = 5. Assim, o bit mais significativo terá como valor de posição 2^5 ($2^{6-1} = 2^5$).

Número Binário ➔	1	0	1	0	1	1
Valor de posição ➔	2^5	2^4	2^3	2^2	2^1	2^0
	MSB *					**LSB ****

Observação:
MSB * - do inglês "most significant bit", ou seja, bit mais significativo.
LSB ** - do inglês "least significant bit", ou seja, bit menos significativo.

A base é o elemento **diferenciador** entre um número do sistema binário e um número do sistema decimal. Portanto, **101** por ser um número base **2**, é lido **um, zero, um**. Já **101**, por ser um número de base **10**, é ligado como **cento e um.**

Sistema de numeração	Número	Leitura do número
Decimal ➔	**101**	**cento e um**
Binário ➔	**101**	**um, zero, um**

Conversão de números do Sistema Binário para o Sistema Decimal

Para converter um número binário em decimal, deve-se multiplicar cada bit pelo seu valor de posição (que é indicado pala potência de base) e somar os resultados.

Exemplo

Na conversão de 1010_2 para o sistema decimal, procede-se da seguinte forma:

Número Binário ➔	1	0	1	0	
Potência de 2 ➔	2^3	2^2	2^1	2^0	
Valor de posição ➔	1 . 8	0 . 4	1 . 2	0 . 1	
No. Decimal ➔	8 +	0 +	2 +	0	$= 10_{10}$

Portanto, $1010_2 = 10_{10}$

Observe a seguir uma tabela das potências de base 2.

Potência	**Decimal**	**Potência**	**Decimal**	**Potência**	**Decimal**
2^{-9}	0,001953125	2^0	1	2^9	512
2^{-8}	0,00390625	2^1	2	2^{10}	1024
2^{-7}	0,0078125	2^2	4	2^{11}	2048
2^{-6}	0,015625	2^3	8	2^{12}	4096
2^{-5}	0,03125	2^4	16	2^{13}	8192
2^{-4}	0,0625	2^5	32	2^{14}	16384
2^{-3}	0,125	2^6	64	2^{15}	32768
2^{-2}	0,25	2^7	128	2^{16}	65768
2^{-1}	0,5	2^8	256	2^{17}	1311072

Obs.: Lembrete matemático:

Função Exponencial Negativa: $a^{-n} = \left(\frac{1}{a}\right)^n = \frac{1}{a^n}$, assim:

Exemplo 1: $2^{-1} = \left(\frac{1}{2}\right)^1 = \frac{1}{2^1} = 0{,}5$

Exemplo 2: $2^{-5} = \left(\frac{1}{2}\right)^5 = \frac{1}{2^5} = \frac{1}{32} = 0{,}03125$

Conversão de números do Sistema Decimal para o Sistema Binário (Método prático)

A conversão de números do sistema decimal para o sistema binário é realizada efetuando-se divisões sucessivas do número decimal por **2** (base do sistema binário).

Exemplo:

29	2			
09	14	2		
1	**0**	7	2	
		1	3	2
			1	**1**

O número binário é formado pelo **quociente** da ultima divisão e os **restos** das divisões sucessivas da direita para a esquerda: $29_{10} = 11101_2$.

Observação:
Todo número decimal **par**, ao ser convertido para binário, termina em **zero**. Por outro lado, todo o número decimal **ímpar** ao ser convertido para binário, termina em **um**.

Números Fracionários

Todo número fracionário decimal tem uma parte inteira (à esquerda da vírgula) e uma parte fracionária à direita da vírgula.

Exemplo:

parte inteira	parte fracionária
1 0 5	, 2 5

No exemplo dado, a parte fracionária (0,25) possui duas casas decimais. O valor de posição da primeira casa após a vírgula corresponde aos décimos

$$0,\mathbf{2}5 = \frac{2}{10} = \frac{2}{10^{+1}} = 2.10^{-1}$$

O valor da segunda posição após a vírgula corresponde aos centésimos:

$$0,2\mathbf{5} = \frac{5}{100} = \frac{5}{100^{+1}} = 5.10^{-2}$$

Assim, o número $105,25_{10}$ tem os seguintes valores de posição:

$$\begin{array}{ccccccccc} 1 & & 0 & & 5 & & 2 & & 5 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 1.10^{2} & + & 0.10^{1} & + & 5.10^{0} & + & 2.10^{-1} & + & 5.10^{-2} \end{array}$$

No sistema binário, a parte fracionária é análoga ao do sistema decimal.

Exemplo:

parte inteira	parte fracionária
1 0 1	, 1 1 $_2$

O valor de posição da primeira casa da parte fracionária será:

$$0,\mathbf{1}1 = \frac{1}{2^{+1}} = 1.2^{-1}$$

O valor de posição da segunda casa após a virgula será:

$$0, 1\mathbf{1} = \frac{1}{2^{+2}} = 1.2^{-2}$$

Assim os valores de posição do número 101,11 serão:

$$\begin{array}{lcccccc} \text{Valor de Posição:} & 2^{+2} & 2^{+1} & 2^{0} & & 2^{-1} & 2^{-2} \\ & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ \text{Número binário:} & 1 & 0 & 1 & , & 1 & 1 \end{array}$$

Conversão de números do Sistema Binário Fracionário em números do Sistema Decimal Fracionário

Já vimos que para converter um número binário em decimal devemos multiplicá-lo pelo valor de posição a base. Observe, a seguir, o valor de posição da parte fracionária dos seguintes números:

$$2^{-1} = \frac{1}{2^{+1}} = \frac{1}{2} = 0{,}5 \qquad \text{ou} \qquad 2^{-4} = \frac{1}{2^{+4}} = \frac{1}{16} = 0{,}0625$$

Veja a seguir o procedimento da conversão de binário fracionário em decimal fracionário. A titulo de exemplo, vamos converter o número $1001{,}01_2$ (binário fracionário) em número decimal:

1°Parte: Faz-se a conversão da parte inteira do número (1001,)

Número binário	➔	1		0		0		1		
		↓		↓		↓		↓		
valor de posição	➔	1.2^{+3}		0.2^{+2}		0.2^{+1}		1.2^{0}		
		↓		↓		↓		↓		
n° decimal	➔	8	+	0	+	0	+	1	=	9

2°Parte: A seguir faz-se a conversão da parte fracionária (...,01) da seguinte maneira:

Binário Fracionário	➔	1001,	0		1		
			↓		↓		
valor de posição	➔		0.2^{-1}		1.2^{-2}		
			↓		↓		
			0. 0,5	+	1. 0,25		
			↓		↓		
Decimal fracionário	➔		0	+	0,25	=	0,25

Assim, 0,25 é a parte decimal fracionária. Portanto. o número $1001{,}01_2$ equivale a $9{,}25_{10}$.

Outro exemplo:

1		0		1	,	1		0		1
↓		↓		↓		↓		↓		↓
$1 . 2^{+2}$	+	$0 . 2^{+1}$	+	$1 . 2^{0}$	+	$1 . 2^{-1}$	+	$0 . 2^{-2}$	+	$1 . 2^{-3}$
↓		↓		↓		↓		↓		↓
1 . 4	+	0 . 2	+	1 . 1	+	1 . 0,5	+	0 . 0,25	+	1 . 0,125
↓		↓		↓		↓		↓		↓
4	+	0	+	1	+	0,5	+	0	+	0,125
		↓						↓		
		5			+			0,625		
					$5{,}625_{10}$					

Portanto, $101{,}101_2 = 5{,}625_{10}$

Conversão de números do Sistema Decimal Fracionário em números do Sistema Binário Fracionário

Como já vimos, para converter um número decimal inteiro em binário, basta dividi-lo por 2, sucessivamente. O número binário será dado pelos restos das divisões e o quociente da última divisão.

Exemplo 1: Converter o número decimal fracionário: $9{,}25_{10}$ em número binário fracionário.

1° Parte: Trata-se primeiramente a parte inteira do número:

9	2		
1	4	2	
	0	2	2
		0	**1**

Portanto, o número: $9_{10} = 1001_2$

2° Parte: Para converter a parte fracionária do número, deve-se fazer o inverso, ou seja, multiplicá-la sucessivamente por 2 até que o resultado seja 0, ou que se obtenham oito dígitos.

Assim, temos:

0,25	0,50
x 2	x 2
0,50	**1**,00

1, ➔ 2° algarismo após a virgula

0, ➔ 1° algarismo após a virgula

Portanto, o número: $,25_{10} = 01_2$, e $9{,}25_{10} = 1001{,}\ 01_2$

Os algarismos a esquerda da vírgula na multiplicação constituirão os dígitos binários fracionários.

Exemplo 2: Converter o número decimal fracionário: $23{,}35_{10}$ em número binário fracionário.

1° Parte: Conversão da parte inteira: 23,

23	2			
03	11	2		
1	**1**	5	2	
		1	2	2
			0	**1**

$\rightarrow 23_{10} = 10111_2$

2° Parte: Conversão da parte fracionária: 0,35

a. 0,35 . 2 = **0**,70
b. 0,70 . 2 = **1**,40 (considere para multiplicação apenas a parte fracionária)
c. 0,40 . 2 = **0**,80

d. 0,80 . 2 = **1**,60
e. 0,60 . 2 = **1**,20
f. 0,20 . 2 = **0**,40
g. 0,40 . 2 = **0**,80
h. 0,80 . 2 = **1**,60 Assim: $0,35_{10} = 01011001..._2$ Portanto: $23,35_{10} = 10111,01011001_2$

Observação:
Observe que o número 0,80 é urna repetição. Isso significa que se trata de urna dízima periódica, o que pode ser indicado por três pontinhos (...).

Exercícios de Fixação: LISTA 01
(Observação: Entregar as resoluções em folha avulsa, devidamente identificada e datada).

Exercícios:

1. Represente os números abaixo utilizando potencia de 10
   a) 843 = ...
   b) 1238 =..
   c) 2048 = ...

2. Converta para decimal os seguintes números binários:
   a) $1110_{(2)}$ = ... $_{(10)}$
   b) $101010_{(2)}$ = ... $_{(10)}$
   c) $100100_{(2)}$ = ... $_{(10)}$
   d) $100000_{(2)}$ = ... $_{(10)}$
   e) $110110_{(2)}$ = ... $_{(10)}$
   f) $11011101_{(2)}$ = ... $_{(10)}$
   g) $1010111_{(2)}$ = ... $_{(10)}$

3. Converta para binário os números decimais abaixo. Utilize o processo prático.
   a) $88_{(10)}$ = ... $_{(2)}$
   b) $51_{(10)}$ = ... $_{(2)}$
   c) $33_{(10)}$ = ... $_{(2)}$
   d) $97_{(10)}$ = ... $_{(2)}$
   e) $15_{(10)}$ = ... $_{(2)}$
   f) $42_{(10)}$ = ... $_{(2)}$
   g) $1045_{(10)}$ = ... $_{(2)}$
   h) $128_{(10)}$ = ... $_{(2)}$

4. Converta os números binários fracionários abaixo em números decimais.
   a) $111,11_{(2)}$ = ... $_{(10)}$
   b) $1101,011_{(2)}$ = ... $_{(10)}$
   c) $11101,1011_{(2)}$ = ... $_{(10)}$
   d) $1010,11_{(2)}$ = ... $_{(10)}$
   e) $1110,001_{(2)}$ = ... $_{(10)}$
   f) $10011,111_{(2)}$ = ... $_{(10)}$

5. Converta para a numeração binária os seguintes números decimais fracionários:
   a) $12,375_{(10)}$ = ... $_{(2)}$
   b) $4,8_{(10)}$ = ... $_{(2)}$
   c) $10,25_{(10)}$ = ... $_{(2)}$
   d) $21,75_{(10)}$ = ... $_{(2)}$
   e) $15,875_{(10)}$ = ... $_{(2)}$
   f) $5,6_{(10)}$ = ... $_{(2)}$
   g) $21,35_{(10)}$ = ... $_{(2)}$
   h) $12,721_{(10)}$ = ... $_{(2)}$

Bom Estudo.

Sistema de Numeração Octal

O sistema de numeração octal tem a base 8. Os oito símbolos que constituem a numeração octal são os seguintes algarismos: **0, 1, 2, 3, 4, 5, 6, 7.**

Este sistema é empregado em computação e em mapeamento de memórias de máquinas digitais que utilizam palavras de 4 ou 8 bits.

A tabela a seguir mostra a relação entre numeração decimal e a octal.

Decimal	Octal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Decimal	Octal
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

Decimal	Octal
16	20
17	21
18	22
19	23
20	24
21	25
22	26
. . .	. . .

Pela tabela, é possível observar que a contagem recomeça a cada 8 dígitos.

Os valores de posição da numeração octal serão as potências de base 8. Observe o quadro a seguir.

Potências de base 8 ➔	8^3	8^2	8^1	8^0
Valores de posição ➔	512	64	8	1

Conversão de números do Sistema Octal para o Sistema Decimal

A conversão de um número octal é realizada de mesmo modo como nos sistemas já estudados. Ou seja, multiplicando-se cada dígito octal por seu valor de posição e somando-se os resultados.

Exemplo: Converter o número 137_8 em um número decimal.

No. Octal ➔	1		3		7	
Potências de 8 ➔	8^2		8^1		8^0	
Valor de posição ➔	1 . 64		3 . 8		7 . 1	
No. Decimal ➔	64	+	24	+	7	= 95_{10}

Portanto, $137_8 = 95_{10}$

Conversão de números do Sistema Decimal para o Sistema Octal

Para converter um número decimal em octal, executam-se divisões sucessivas do número decimal por 8, que é a base do sistema octal. O número octal será dado pelo último quociente e pelos restos das divisões.

Exemplo:

1236	8		
43	154	8	
36	74	19	8
4	2	3	2

O último quociente e os restos das divisões resultarão no número octal.
Portanto, $\mathbf{1236_{10} = 2324_{8}}$.

Conversão de números do Sistema Octal para o Sistema Binário

A tabela a seguir mostra a correspondência entre o sistema octal e o binário.

0ctal	Binário	Octal	Binário	Octal	Binário
0	000	10	1000	20	10000
1	001	11	1001	21	10001
2	010	12	1010	22	10010
3	011	13	1011	23	10011
4	100	14	1100	24	10100
5	101	15	1101	25	10101
6	110	16	1110	26	10110
7	111	17	1111	...	...

Pela tabela é possível observar que cada código octal (dígitos de 0 a 7) corresponde três dígitos binários. Desse modo, para converter cada algarismo do número octal no número binário correspondente. Esse número binário terá três dígitos.

Exemplo: Converter o número 4625_{8} para binário.

Dígitos octais ➔	4	6	2	5
Dígitos binários ➔	100	110	010	101

Portanto, $4625_{8} = 100110010101_{2}$

Conversão de números do Sistema Binário para o Sistema Octal

Para converter um número binário em octal, basta separar o número binário, da direita para a esquerda, em grupos de três bits. Em seguida, converte-se cada grupo no algarismo octal correspondente.

Observação
Se não for possível formar um grupo de 3 bits, completa-se o grupo com zeros, ou seja: 10011, por exemplo, daria 010011.

Exemplo: Converter 1100101_2 para o sistema octal

Dígitos binários →	001	100	101
Digito octal →	1	4	5

Portanto, $1100101_2 = 145_8$

Exercícios de Fixação: LISTA 02
(Observação: Entregar as resoluções em folha avulsa, devidamente identificada e datada).

Exercícios:

1. Converta os números Octais abaixo em números Decimais:
   a. $246_{(8)}$ = ... $_{(10)}$
   b. $1176_{(8)}$ = ... $_{(10)}$
   c. $74_{(8)}$ = ... $_{(10)}$
   d. $325_{(8)}$ = ... $_{(10)}$
   e. $123_{(8)}$ = ... $_{(10)}$

2. Converta os números Decimais abaixo em números do sistema Octal:
   a. $298_{(10)}$ = ... $_{(8)}$
   b. $512_{(10)}$ = ... $_{(8)}$
   c. $174_{(10)}$ = ... $_{(8)}$
   d. $327_{(10)}$ = ... $_{(8)}$
   e. $4096_{(10)}$ = ... $_{(8)}$

3. Converta em binário os seguintes números Octais:
   a. $447_{(8)}$ = ... $_{(2)}$
   b. $632_{(8)}$ = ... $_{(2)}$
   c. $70_{(8)}$ = ... $_{(2)}$
   d. $5227_{(8)}$ = ... $_{(2)}$

4. Converta para o sistema octal os seguintes números binários:
   a. $10101_{(2)}$ = ... $_{(8)}$
   b. $10010100_{(2)}$ = ... $_{(8)}$
   c. $111011001_{(2)}$ = ... $_{(8)}$
   d. $101110110_{(2)}$ = ... $_{(8)}$

Bom Estudo.

Sistema de Numeração Hexadecimal

O sistema de numeração hexadecimal tem a base 16. Os dezesseis símbolos que constituem a numeração hexadecimal são os seguintes algarismos e letras:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Este sistema é empregado em computação e em mapeamento de memórias de máquinas digitais que utilizam palavras de 4, 8 ou 16 bits.

A tabela a seguir mostra a relação entre numeração dos sistemas de numeração decimal e hexadecimal.

Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal
0	0	8	8	16	10	24	18
1	1	9	9	17	11	25	19
2	2	10	A	18	12	26	1A
3	3	11	B	19	13	27	1B
4	4	12	C	20	14	28	1C
5	5	13	D	21	15	29	1D
6	6	14	E	22	16	30	1E
7	7	15	F	23	17	31	1F
						. . .	. . .

Pela tabela, é possível observar que a contagem recomeça a cada 16 dígitos.

Os valores de posição da numeração hexadecimal serão as potências de base 16. Observe o quadro a seguir.

Potências de base 16 ➔	16^3	16^2	16^1	16^0
Valores de posição ➔	4096	256	16	1

Conversão de números do Sistema Hexadecimal para o Sistema Decimal

A conversão de um número hexadecimal é realizada de mesmo modo como nos sistemas já estudados. Ou seja, multiplicando-se cada dígito hexadecimal por seu valor de posição e somando-se os resultados.

Exemplo:
Converter o número $1A8_{16}$ em um número decimal.

No. Hexadecimal ➔	1		A		8	
Potências de 16 ➔	16^2		16^1		16^0	
Valor de posição ➔	1 . 256		10 . 16		8 . 1	
No. Decimal ➔	256	+	160	+	8	= 424_{10}

Portanto, $1A8_{16} = 424_{10}$

Conversão de números do Sistema Decimal para o Sistema Hexadecimal

Para converter um número decimal em hexadecimal, executam-se divisões sucessivas do número decimal por 16, que é a base do sistema hexadecimal. O número hexadecimal será dado pelo último quociente e pelos restos das divisões.

Exemplo:

12412	16		
12	775	16	
	7	48	16
		0	**3**

O último quociente e os restos das divisões resultarão no número hexadecimal. Contudo, em número hexadecimal não existe o número 12. Na tabela já mostrada, vemos que a letra **C** em hexadecimal equivale ao número **12** decimal. Portanto, pela conversão, obtivemos o número **307C**. Portanto, **12412_{10} = $307C_{16}$**.

Conversão de números do Sistema Hexadecimal para o Sistema Binário

A tabela a seguir mostra a correspondência entre o sistema hexadecimal e o binário.

Hexadecimal	Binário
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Hexadecimal	Binário
10	1 0000
11	1 0001
12	1 0010
. . .	. . .
19	1 1001
1A	1 1010
. . .	. . .
1F	1 1111
20	10 0000
21	10 0001
22	10 0010
. . .	. . .
29	10 1001
2A	10 1010
. . .	. . .
2F	10 1111

Hexadecimal	Binário
30	11 0000
31	11 0001
32	11 0010
. . .	. . .
FF	1111 1111
100	1 0000 0000
101	1 0000 0001
. . .	. . .
1FF	1 1111 1111
200	10 0000 0000
201	10 0000 0001
. . .	. . .
2FF	10 1111 1111
. . .	. . .
FFF	1111 1111 1111
1000	1 0000 0000 0000

Pela tabela é possível observar que o cada código hexadecimal (caracteres de 0 a F), corresponde quatro dígitos binários. Desse modo, para converter cada algarismo ou letra do número hexadecimal no número binário correspondente. Esse número binário terá quatro dígitos.

Exemplo: Converter o número FACA $_{16}$ para binário.

Dígitos hexadecimais ➔	F	A	C	A
Dígitos binários ➔	1111	1010	1100	1010

Portanto, FACA $_{16}$ = 1111101011001010 $_{2}$

Conversão de números do Sistema Binário para o Sistema Hexadecimal

Para converter um número binário em hexadecimal, basta separar o número binário, da direita para a esquerda, em grupos de quatro bits. Em seguida, converte-se cada grupo no algarismo hexadecimal correspondente.

Observação:
Se não for possível formar um grupo de 4 bits, completa-se o grupo com zeros, ou seja: 10011, por exemplo, daria 00010011.

Exemplo: Converter 101001101_2 para o sistema hexadecimal

Dígitos binários	➔	0001	0100	1101
Decimal	➔	1	4	13
Hexadecimal	➔	1	4	D

Na numeração hexadecimal não existe o número 13; em seu lugar usa-se a letra D. Portanto, o resultado da conversão será: $101001101_2 = 14D_{16}$.

Exercícios de Fixação: LISTA 03
(Observação: Entregar as resoluções em folha avulsa, devidamente identificada e datada).

Exercícios:

1. Converta em decimais os seguintes números hexadecimais:
   a. BEBE $_{(16)}$ = ... $_{(10)}$
   b. 5A $_{(16)}$ = ... $_{(10)}$
   c. 49F $_{(16)}$ = ... $_{(10)}$
   d. A1E $_{(16)}$ = ... $_{(10)}$
   e. 26C $_{(16)}$ = ... $_{(10)}$
   f. 291 $_{(16)}$ = ... $_{(10)}$
   g. CADE $_{(16)}$ = ... $_{(10)}$
   h. 7C $_{(16)}$ = ... $_{(10)}$
   i. 0F $_{(16)}$ = ... $_{(10)}$
   j. 36D $_{(16)}$ = ... $_{(10)}$

2. Converta em hexadecimais os seguintes números decimais:
   a. 329 $_{(10)}$ = ... $_{(16)}$
   b. 2327 $_{(10)}$ = ... $_{(16)}$
   c. 284 $_{(10)}$ = ... $_{(16)}$
   d. 1238 $_{(10)}$ = ... $_{(16)}$
   e. 59 $_{(10)}$ = ... $_{(16)}$
   f. 5438 $_{(10)}$ = ... $_{(16)}$

3. Converta em binário os seguintes números hexadecimais:
   a. 31C $_{(16)}$ = ... $_{(2)}$
   b. AF3 $_{(16)}$ = ... $_{(2)}$
   c. D1347 $_{(16)}$ = ... $_{(2)}$
   d. A091 $_{(16)}$ = ... $_{(2)}$
   e. 16BE $_{(16)}$ = ... $_{(2)}$
   f. CE1A $_{(16)}$ = ... $_{(2)}$

4. Converta em hexadecimais os seguintes números binários:
   a. 11011 $_{(2)}$ = ... $_{(16)}$
   b. 1101011100 $_{(2)}$ = ... $_{(16)}$
   c. 100110010011 $_{(2)}$ = ... $_{(16)}$
   d. 1111010111 $_{(2)}$ = ... $_{(16)}$
   e. 10011011 $_{(2)}$ = ... $_{(16)}$
   f. 1101010 $_{(2)}$ = ... $_{(16)}$
   g. 10110111010 $_{(2)}$ = ... $_{(16)}$
   h. 100010011011 $_{(2)}$ = ... $_{(16)}$

Bom Estudo

Operações Aritméticas do Sistema Binário

Para facilitar a compreensão de circuitos lógicos e aritméticos, tais como **somadores** e **subtratores** é necessário estudar as operações aritméticas de adição, subtração e multiplicação de números binários.

Operação Aritmética de Adição

A operação de adição de números binários é idêntica à do sistema decimal. O sistema binário, como já sabemos, possui apenas dois algarismos: 0 e 1. Para a realização da soma, existem as seguintes condições

0 + 0 = **0**
1 + 0 = **1**
0 + 1 = **1**
1 + 1 = **0 e vai 1** = 10 (um, zero)

Observação:
Na condição 1 + 1 = 10 (um, zero) está exemplificada a **regra de transporte** na qual 1 é transportado para a coluna seguinte, ou seja, "vai um".

Por exemplo, a soma de $110_2 + 101_2$, de acordo com essas regras é realizada do seguinte modo:

```
1*
  1 1 0 +
  1 0 1
 -------
1 0 1 1
```

Obs.: * Transporte ou vai-um. Assim, $110_2 + 101_2 = 1011_2$

Operação Aritmética de Subtração

O processo de subtração binária é igual ao de subtração decimal. As regras da subtração binária são:

0 - 0 = **0**
1 - 1 = **0**
1 - 0 = **1**
0 - 1 = **1 e "empresta um"**

Observação:

Na condição 0 - 1 = 1 está exemplificada a regra de transporte na qual 1 é **emprestado** da coluna seguinte.

Veja, por exemplo, a subtração de 110_2 - 10_2:

$$\begin{array}{cccc} & 1 & 1 & 0_2 \\ & & 1 & 0_2 \;\; - \\ \hline & 1 & 0 & 0_2 \end{array}$$

Assim, $110_2 - 10_2 = 100_2$

Veja, agora, um exemplo com "empresta um":

$$\begin{array}{cccccccc} & & {\scriptstyle 1} & {\scriptstyle 1} & & {\scriptstyle 1} & & \\ & 1 & 0 & 0 & 1 & 0 & 1 & _2 \\ - & & & 1 & 0 & 1 & 0 & _2 \\ \hline & 0 & 1 & 1 & 0 & 1 & 1 & _2 \end{array}$$

→ transporte ou empréstimo de 1.

Observação: $11011_2 = 27_{10}$; $1010_2 = 10_{10}$; $100101_2 = 37_{10}$;

Assim, $100101_2 - 1010_2 = 11011_2$

Para visualização de cada uma das passagens parciais da operação de subtração, acompanhe os passos descritos a seguir:

1° Passo

$$\begin{array}{cccccccc} & 1 & 0 & 0 & 1 & 0 & \boxed{\mathbf{1}} & _2 \\ - & & & 1 & 0 & 1 & \boxed{\mathbf{0}} & _2 \\ \hline & & & & & & \mathbf{1} & _2 \end{array}$$

Obs.: 1 - 0 = 1

2° Passo

$$\begin{array}{cccccccc} & 1 & 0 & 0 & 1 & \boxed{\mathbf{0}} & 1 & _2 \\ & & & & {\scriptstyle 1} & & & \\ - & & & 1 & 0 & \boxed{\mathbf{1}} & 0 & _2 \\ \hline & & & & & \mathbf{1} & 1 & _2 \end{array}$$

Obs.: 0 - 1 = 1 e empresta 1

3° Passo

$$\begin{array}{cccccccc} & 1 & 0 & 0 & 1 & 0 & 1 & _2 \\ & & & & {\scriptstyle 1} & & & \\ & & & & {\scriptstyle 1} & & & \\ - & & & 1 & \boxed{\mathbf{0}} & 1 & 0 & _2 \\ \hline & & & & & 1 & 1 & _2 \end{array}$$

Obs.: 1 + 0 = 1 (operação com o empresta 1)

4° Passo

$$\begin{array}{cccccccc} & 1 & 0 & 0 & \boxed{\mathbf{1}} & 0 & 1 & _2 \\ & & & & {\scriptstyle 1} & & & \\ & & & & {\scriptstyle 1} & & & \\ - & & & 1 & 0 & 1 & 0 & _2 \\ \hline & & & & \mathbf{0} & 1 & 1 & _2 \end{array}$$

Obs.: 1 – 1 = 0 (operação com o resultado do 3° Passo)

5° Passo

$$\begin{array}{cccccccc} & 1 & 0 & \boxed{\mathbf{0}} & 1 & 0 & 1 & _2 \\ & & & & {\scriptstyle 1} & & & \\ & & {\scriptstyle 1} & & {\scriptstyle 1} & & & \\ - & & & \mathbf{1} & 0 & 1 & 0 & _2 \\ \hline & & & \mathbf{1} & 0 & 1 & 1 & _2 \end{array}$$

Obs.: 0 - 1 = 1 e empresta 1

6° Passo

$$\begin{array}{cccccccc} & 1 & 0 & 0 & 1 & 0 & 1 & _2 \\ & & {\scriptstyle 1} & & & & & \\ & & {\scriptstyle 1} & & {\scriptstyle 1} & & & \\ - & & \boxed{\scriptstyle 0} & 1 & 0 & 1 & 0 & _2 \\ \hline & & & 1 & 0 & 1 & 1 & _2 \end{array}$$

Obs.: 1 + 0 = 1 (operação com o empresta 1)

7° Passo

$$\begin{array}{cccccccc} & 1 & \boxed{\mathbf{0}} & 0 & 1 & 0 & 1 & _2 \\ & & {\scriptstyle 1} & & & & & \\ & {\scriptstyle 1} & {\scriptstyle 1} & & {\scriptstyle 1} & & & \\ - & & {\scriptstyle 0} & 1 & 0 & 1 & 0 & _2 \\ \hline & & \mathbf{1} & 1 & 0 & 1 & 1 & _2 \end{array}$$

Obs.: 0 – 1 = 1 e empresta 1 (operação com o resultado do 6° Passo)

8° Passo								
		1	0	0	1	0	1	2
		1	1					
		1	1		1			
	-	0	0	1	0	1	0	2
			1	1	0	1	1	2

Obs.: 1 + 0 = 1

9° Passo								
		1	0	0	1	0	1	2
		1	1					
		1	1		1			
	-	0	0	1	0	1	0	2
		0	1	1	0	1	1	2

Obs.: 1 – 1 = 0

Assim, $100101_2 - 1010_2 = 11011_2$

Operação Aritmética de Subtração pelo “Complemento”

A subtração de números binários pode ser efetuada pela **soma do complemento**. Esse método possui três variações:

Método 1: • Soma simples do complemento;
Método 2: • Soma do complemento de 1;
Método 3: • Soma do complemento de 2.

- (Método 1) **Subtração por soma simples do complemento -** Para realizar a subtração por soma simples do complemento, procede-se da seguinte forma:

  1° Passo: Determina-se o **complemento do minuendo** (transformando o 1 em 0 e o 0 em 1);
  2° Passo: Soma-se o subtraendo;
  3° Passo: Determina-se o complemento do resultado.

Exemplo:
Subtrair 0010_2 de 0111_2

1° Passo: Determina-se o complemento do minuendo (transformando o 1 em 0 e o 0 em 1);
Minuendo = 0111_2 → Complemento = 1000_2

2° Passo: Soma-se o complemento do minuendo ao subtraendo;

1	0	0	0		(Complemento)
0	0	1	0	+	(Subtraendo)
1	0	1	0		(resultado)

3° Passo: Determina-se o complemento do resultado.
Resultado = 1010 → Complemento = **0101**

Portanto, o resultado é **0101_2.**

Pode-se provar a exatidão desse resultado comparando-se com o da subtração decimal:

$0\ 1\ 1\ 1_2$		→	7_{10}
–	$0\ 0\ 1\ 0_2$	→	$- 2_{10}$
		→	
$0\ 1\ 0\ 1_2$			5_{10}

- (Método 2) **Subtração por soma do complemento de 1 -** Esse método de subtração segue a seguinte seqüência:

  1° Passo: determina-se o **complemento de 1** do subtraendo, transformando-se o 0 em 1 e o 1 em 0;

  2° Passo: efetua-se a **soma** do minuendo com o complemento de 1 do subtraendo;

  3° Passo: soma-se o vai-um ao bit menos significativo.

Exemplo:

Subtrair 0110_2 de 1101_2.

1° Passo: determina-se o complemento de 1 do subtraendo, transformado-se o 0 em 1 e o 1 em 0; (Obs.: Subtraendo = 0110_2 ➔ Complemento = 1001_2)

2° Passo: efetua-se a soma do minuendo com o complemento de 1 do subtraendo;

$$\begin{array}{rl} 1\ 1\ 0\ 1 & \text{(Minuendo)} \\ +\ 1\ 0\ 0\ 1 & \text{(Complemento do Subtraendo)} \\ \hline 1\ 0\ 1\ 1\ 0 & \text{(resultado)} \end{array}$$

↓

vai-um

3° Passo: soma-se o vai-um ao bit menos significativo.

$$\begin{array}{r} 0\ 1\ 1\ 0 \\ +\quad\quad\ 1 \\ \hline 0\ 1\ 1\ 1 \end{array}$$

Portanto, **0111** é o resultado final.

Pode-se comprovar esse resultado, comparando-o com o obtido na subtração decimal.

$$\begin{array}{r} 1\ 1\ \ \ \ \\ 1\ 1\ 0\ 1_2 \\ -\ 0\ 1\ 1\ 0_2 \\ \hline 0\ 1\ 1\ 1_2 \end{array} \quad \begin{array}{c} \\ ➔ \\ ➔ \\ ➔ \end{array} \quad \begin{array}{r} 13_{10} \\ -\ 6_{10} \\ \hline 7_{10} \end{array}$$

Observação

Se o **subtraendo** tiver **menos** dígitos do que o **minuendo**, deve-se completar com zeros as posições que faltarem antes de completar o subtraendo. Por exemplo:

$$\begin{array}{r} 1\ 0\ 0\ 1\ 1_2 \\ -\quad\ 0\ 1\ 1_2 \\ \hline \end{array} \qquad \begin{array}{r} 1\ 0\ 0\ 1\ 1_2 \\ -\ 0\ 0\ 0\ 1\ 1_2 \\ \hline \end{array} \qquad \begin{array}{r} 1\ 1\ 1\ \ \ \ \ \ \\ 1\ 0\ 0\ 1\ 1_2 \\ -\ 1\ 1\ 1\ 0\ 0_2 \\ \hline 1\ 0\ 1\ 1\ 1\ 1 \\ 1+ \\ \hline 1\ 0\ 0\ 0\ 0_2 \end{array}$$

O resultado pode ser provado se comparado com o resultado da operação executada com números decimais:

$$\begin{array}{r} 1\ 0\ 0\ 1\ 1_2 \\ -\quad 0\ 1\ 1_2 \\ \hline 1\ 0\ 0\ 0\ 0_2 \end{array} \qquad \begin{array}{r} 19_{10} \\ -\ 3_{10} \\ \hline 16_{10} \end{array}$$

- (Método 3) **Subtração por soma do complemento de 2 -** O método de subtração pela soma do complemento de 2 segue a seguinte seqüência:

  1° Passo: Determina-se o **complemento de 1** do **subtraendo**;
  2° Passo: **Soma-se 1** ao subtraendo (complemento de 1) a fim de obter o **complemento de 2;**
  3° Passo: Soma-se o minuendo com o complemento de 2 do subtraendo;
  4° Passo: Ignora-se o vai-um do resultado da soma.

Exemplo:
Efetuar a seguinte subtração: 1101_2 - 0110_2

1° Passo: Determina-se o complemento de 1 do subtraendo;
Subtraendo = 0110_2 ➔ Complemento = 1001_2

2° Passo: Soma-se 1 ao complemento de 1 do subtraendo (resultado do passo 1) a fim de obter o complemento de 2;

$$\begin{array}{rl} 1\ 0\ 0\ 1 & \text{(Subtraendo)} \\ +\quad\quad 1 & \\ \hline 1\ 0\ 1\ 0 & \text{(Complemento de 2)} \end{array}$$

3° Passo: Soma-se o minuendo com o complemento de 2 do subtraendo;

$$\begin{array}{rl} 1\ 1\ 0\ 1 & \text{(Minuendo)} \\ +\ 1\ 0\ 1\ 0 & \text{(Complemento de 2)} \\ \hline \mathbf{1}\ 0\ 1\ 1\ 1 & \end{array}$$

↓
vai-um

4° Passo: Ignora-se o vai-um do resultado da soma. (resultado final)

= **0 1 1 1**

Como o vai-um é ignorado, o resultado de 1101_2 - 0110_2 = $\mathbf{0111_2}$.

Operação Aritmética de Multiplicação

A multiplicação de números binários é feita do mesmo modo como no sistema decimal, ou seja:

0 **.** 0 = **0**
0 **.** 1 = **0**
1 **.** 0 = **0**
1 **.** 1 = **1**

Exemplo: Multiplicar 11010_2 . 10_2

$$\begin{array}{r} 11010_2 \\ .\quad 10_2 \\ \hline 00000 \\ 11010 \\ \hline 110100_2 \end{array} \qquad \begin{array}{r} 26 \\ .\,2 \\ \hline 52_{10} \end{array}$$

Operação Aritmética de Divisão

A divisão de números binários é feita do mesmo modo como no sistema decimal, ou seja:

0 ÷ 0 = **?**
0 ÷ 1 = **0**
1 ÷ 0 = **?**
1 ÷ 1 = **1**

Exemplo: Dividir: 110000_2 . 1000_2

1° Passo:

$$\begin{array}{rl|l} & 110000 & 1000 \\ \cline{3-3} - & \mathbf{1000} & \mathbf{1} \\ \cline{2-2} & \mathbf{0100} & \end{array}$$

Obs.:
$110000_2 = 48_{10}$
$1000_2 = 8_{10}$
$110_2 = 6_{10}$

2° Passo:

$$\begin{array}{rl|l} & 1100\mathbf{0}0 & 1000 \\ \cline{3-3} & 1000 & 1\mathbf{1} \\ & 0100\mathbf{0} & \\ - & \mathbf{1000} & \\ \cline{2-2} & \mathbf{0000} & \end{array}$$

3° Passo:

$$\begin{array}{rl|l} & 11000\mathbf{0} & 1000 \\ \cline{3-3} & 1000 & 11\mathbf{0} \\ & 01000 & \\ - & 1000 & \\ \cline{2-2} & 0000\mathbf{0} & \end{array}$$

Códigos Binários

Os códigos binários utilizam os mesmos símbolos do sistema de numeração binário, ou seja, os símbolos 0 e 1 . Os símbolos 1 e 0 são códigos que podem representar números, letras do alfabeto e sinais de pontuação.

Os códigos binários mais empregados são:

- BCD 8421
- BCD-AIKEN
- Johnson
- ASCII

Código Binário BCD 8421

BCD significa “Decimal Codificado em Binário”(do inglês: “Binary Coded Decimal”). É um código que utiliza números binários para representar os dígitos de um número decimal. Cada grupo de **quatro dígitos** representa um algarismo do número decimal. Veja exemplos a seguir.

Exemplo 1:

Valor de posição			➔	8 4 2 1	8 4 2 1
Dígito binário	➔	$11\ 0111_2$	➔	0011	0111
				↓	↓
Digito decimal			➔	3	7

Portanto: $11\ 0111_2 = 37_{10}$

Exemplo 2:

Valor de posição			→	8 4 2 1	8 4 2 1	8 4 2 1
Dígito binário	→	$1000\ 0010\ 0101_2$	→	1000	0010	0101
				↓	↓	↓
Digito decimal			→	8	2	5

Portanto: $1000\ 0010\ 0101_2 = 825_{10}$

O código BCD é um código ponderado. O bit mais significativo tem peso 8 e o menos significativo tem peso 1. Observe a seguir a tabela do código binário BCD 8421

Dígito Decimal	Código BCD 8421
	Peso: 8 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Código Binário BCD – AIKEN

Esse código representa os dígitos dos números decimais. Cada dígito decimal é representado por um grupo de quatro bits, cujos pesos ou valores de posição dos bits sã: 2, 4, 2 e 1 . Veja a tabela a seguir.

Dígito Decimal	Código BCD - AIKEN
	Peso: 2 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	1 0 1 1

6	1 1 0 0
7	1 1 0 1
8	1 1 1 0
9	1 1 1 1

Os exemplos a seguir mostram como os números decimais são representados no código binário BCD – AIKEN.

Exemplo 1:

Digito decimal → 3 7

Dígito binário → 0011 1101 → 0011 1101

↓ ↓

Valor de posição → 2 4 2 1 2 4 2 1

Portanto: BCD – AIKEN = 0011 1101 é igual a 37_{10}

Exemplo 2:

Digito decimal → 8 2 5

Dígito binário → 1110 0010 1011 → 1110 0010 1011

↓ ↓ ↓

Valor de posição → 2 4 2 1 2 4 2 1 2 4 2 1

Portanto: BCD – AIKEN = 1110 0010 1011 é igual a 825_{10}

Código Binário Johnson

O código Johnson, como o código BCD, utiliza grupos de números binários para representar números decimais. Esse grupo é formado por cinco bits. Veja a tabela a seguir:

Dígito Decimal	Código Johnson
0	0 0 0 0 0
1	0 0 0 0 1
2	0 0 0 1 1
3	0 0 1 1 1
4	0 1 1 1 1
5	1 1 1 1 1
6	1 1 1 1 0
7	1 1 1 0 0
8	1 1 0 0 0
9	1 0 0 0 0

Por esse código o número decimal é assim representado:

Digito decimal → 3 7

↓ ↓

Dígito binário → 00111 11100

Portanto o código binário Johnson: 00111 11100 é igual a 37_{10}

Observação:

Neste código não são usados os valores de posição. É um código progressivo, ou seja, as numerações adjacentes (um número em relação ao seu anterior ou posterior) diferem somente de um bit.

Código Binário ASCII

ASCII significa “Código padrão americano para intercambio internacional” (do inglês: American Standard Code for International Interchange”).

Esse código padroniza a representação de símbolos gráficos (caracteres) e sinais de controle, o que possibilita a comunicação entre diferentes equipamentos. É utilizado em dispositivos de controle numérico, equipamentos industriais, etc.

É constituído de sete bits que combinados, representam 128 estados. Um oitavo bit pode ser usado, não para representar uma informação, mas para possibilitar a checagem de erros de paridade.

Veja a tabela a seguir, mostrando a estrutura de formação do código ASCII

MSB	$b_7\ b_6\ b_5$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
LSB	$b_4\ b_3\ b_2\ b_1$	Controles		Caracteres					

	0 0 0 0	NUL	DLE	SP	0	@	P	.	p
	0 0 0 1	SOH	DC1	!	1	A	Q	a	q
	0 0 1 0	SIX	DC2	“	2	B	R	b	r
	0 0 1 1	ETX	DC3	#	3	C	S	c	s
	0 1 0 0	EOT	DC4	$	4	D	T	d	t
	0 1 0 1	ENQ	NAK	%	5	E	U	e	u
	0 1 1 0	ACK	SYN	&	6	F	V	f	v
	0 1 1 1	BEL	ETB	‘	7	G	W	g	w
	1 0 0 0	BS	CAN	(	8	H	X	h	x
	1 0 0 1	HT	EM	)	9	I	Y	i	y
	1 0 1 0	LF	SUB	*	!	J	Z	j	z
	1 0 1 1	VT	ESC	+	;	K	[	k	{
	1 1 0 0	FF	FS	:	<	L	\	l	|
	1 1 0 1	CR	GS	-	=	M	]	m	}
	1 1 1 0	SO	RS	.	>	N	^	n	~
	1 1 1 1	SI	US	/	?	O	-	o	DEL

Observação: MSB - do inglês “most significant bit”, ou seja, bit mais significativo.
LSB - do inglês “least significant bit”, ou seja, bit menos significativo.

Para entender a composição do código ASCII, localize, por exemplo, na tabela dos caracteres o símbolo “2” . Os bits que representam o caractere 2 são: 011 0010 que vem a ser a combinação de:

011 = bit mais significativo (MSB) em cuja coluna o número 2 está situado.
0010 = bit menos significativo (LSB) em cuja linha o número 2 está situado.

Do mesmo modo, o caractere “C“ será representado por: 100 0011, ou seja:

100 = bit mais significativo (MSB)
0011 = bit menos significativo (LSB)

Exercícios de Fixação: LISTA 04

(Observação: Entregar as resoluções em folha avulsa, devidamente identificada e datada).

Exercícios:

1. Descreva: Onde e como são utilizados os Códigos Binários?

   ....

   ....

2. O que significam as siglas:

   BCD : ....

   ASCII : ....

3. Complete a tabela abaixo convertendo os caracteres dados em seus respectivos códigos binários, se possível.

Caracteres	Código Binário			
	ASCII	BCD-8421	BCD-AIKEN	Johnson
7	....	....	....	....
16	....	....	....	....
22	....	....	....	....
53	....	....	....	....
38	....	....	....	....
B	....	....	....	....
a	....	....	....	....
S	....	....	....	....
?	....	....	....	....

Bom Estudo.   .

Exemplos resumidos de mudanças de base e operações aritméticas

1. Conversão de números de base **Binária** para base **Decimal**:

   $1010_{(2)}$ *resp. = 10 (10)*

   $$\begin{array}{ccccccc} 1 & & 0 & & 1 & & 0_{(2)} \\ 1 \times 2^3 & + & 0 \times 2^2 & + & 1 \times 2^1 & + & 0 \times 2^0 \\ 1 \times 8 & + & 0 \times 4 & + & 1 \times 2 & + & 0 \times 1 \\ 8 & + & 0 & + & 2 & + & 0 \end{array}$$

5. Conversão de números de base **Octal** para base **Decimal**:

   $376_{(8)}$ *resp. = 254 (10)*

   $$\begin{array}{cccccc} 3 & & 7 & & 6 & {}_{(8)} \\ 3 \times 8^2 & + & 7 \times 8^1 & + & 6 \times 8^0 & \\ 3 \times 64 & + & 7 \times 8 & + & 6 \times 1 & \\ 192 & + & 56 & + & 6 & \end{array}$$

$= 10_{(10)}$

$= 254_{(10)}$

2. Conversão de números de base **Decimal** para base **Binária**:

$47_{(10)}$ *resp. = $101111_{(2)}$*

47	2				
07	23	2			
1	03	11	2		
	1	**1**	5	2	
			1	2	2
				0	**1**

3. Conversão de números de base **Binária-Fracionária** para base **Decimal**:

$0{,}10010_{(2)}$ *resp. = $0{,}5625_{(10)}$*

0, → $0 \times 2^{0} = 0 \times 1 = 0$
1 → $1 \times 2^{-1} = 1 \times ½ = 0{,}5$
0 → $0 \times 2^{-2} = 0 \times 0{,}25 = 0$
0 → $0 \times 2^{-3} = 0 \times 0{,}125 = 0$
1 → $1 \times 2^{-4} = 1 \times 0{,}0625 = 0{,}0625$
0 → $0 \times 2^{-5} = 0 \times 0{,}03125 = 0$
Soma das parcelas: $= 0{,}5625_{(10)}$

4. Conversão de números de base **Decimal-Fracionária** para base **Binária**:

$7{,}125_{(10)}$ *resp. = 111,001*

A- Parte inteira do número: (7)

7	2	
1	3	2
	1	**1**

B- Parte fracionária do número: (0,125)

0,125	0,250	0,500
x 2	x 2	x 2
0,250	**0**,500	**1**,000

3˚ algarismo após a virgula

2˚ algarismo após a virgula

1˚ algarismo após a virgula

6. Conversão de números de base **Octal** para base **Binária**:

$536_{(8)}$ *resp. = $101011110_{(2)}$*

5	3	6
101	011	110

7. Conversão de números de base **Binária** para base **Octal**:

$11010101_{(2)}$ *resp. = $325_{(8)}$*

11	010	101
3	2	5

8. Conversão de números de base **Decimal** para base **Octal**:

$719_{(10)}$ *resp. = $1317_{(8)}$*

719	8		
79	89	8	
7	09	11	8
	1	**3**	**1**

9. Conversão de números de base **Hexadecimal** para base **Decimal**:

$238_{(16)}$ *resp. = $568_{(10)}$*

2		3		8
2×16^{2}	+	3×16^{1}	+	8×16^{0}
2 x 256	+	3 x 16	+	8 x 1
512	+	48	+	8

$= 568_{(10)}$

10. Conversão de números de base **Hexadecimal** para base **Binária**:

$ABF_{(16)}$ *resp. = $101010111111_{(2)}$*

A	B	F
1010	1011	1111

11. Conversão de números de base **Binária** para base **Hexadecimal**:

$011011110101_{(2)}$ *resp. = $6F5_{(16)}$*

0110	1111	0101
6	F	5

12. Conversão de números de base **Decimal** para base **Hexadecimal**:

$567_{(10)}$ *resp. = $237_{(16)}$*

567	16	
87	35	16

15. Operação Aritmética de Multiplicação:

$101011_{(2)}$ **x** $101_{(2)}$ = *resp. = $11010111_{(2)}$*

```
  1 0 1 0 1 1
x     1 0 1
1   1
  1 0 1 0 1 1
  0 0 0 0 0 0 +
  1 0 1 0 1 1 + +
  1 1 0 1 0 1 1 1
```

```
 43
x 5
215
```

Obs.:

0 x 0 = 0
0 x 1 = 0
1 x 0 = 1
1 x 1 = 1

16. Operação Aritmética de Divisão:

$10100_{(2)} \div 100_{(2)}$ = *resp. = $101_{(2)}$*

```
7        3        2
```

13. Operação Aritmética de Adição:

10101 $_{(2)}$ **+** 1001001 $_{(2)}$ = *resp. = 1011110* $_{(2)}$

```
 1
 1 0 1 0 1
+  1 0 0 1 0 0
 1
 ______________
 1 0 1 1 1 1 0
```

Obs.:
0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 1 e empresta 1

14. Operação Aritmética de Subtração:

11110 $_{(2)}$ **-** 1111 $_{(2)}$ = *resp. = 01111* $_{(2)}$

```
  1 1 1 1
  1 1 1 1 0
-   1 1 1 1
 ___________
  0 1 1 1 1
```

Obs.:
0 – 0 = 0
1 – 1 = 0
1 – 0 = 1
0 – 1 = 1 e empresta 1

```
 1 0 1 0 0  | 1 0 0
- 1 0 0     |______
 ______       1 0 1
 0 0 1 0 0
   - 1 0 0
   _______
     0 0 0
```

Obs.:
0 ÷ 0 = ?
0 ÷ 1 = 0
1 ÷ 0 = ?
1 ÷ 1 = 1

Operações:
1° Multiplicar o quociente pelo divisor
2° Subtrair o resultado do dividendo.