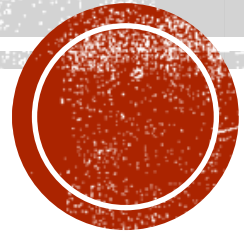
SISTEMAS DIGITAIS

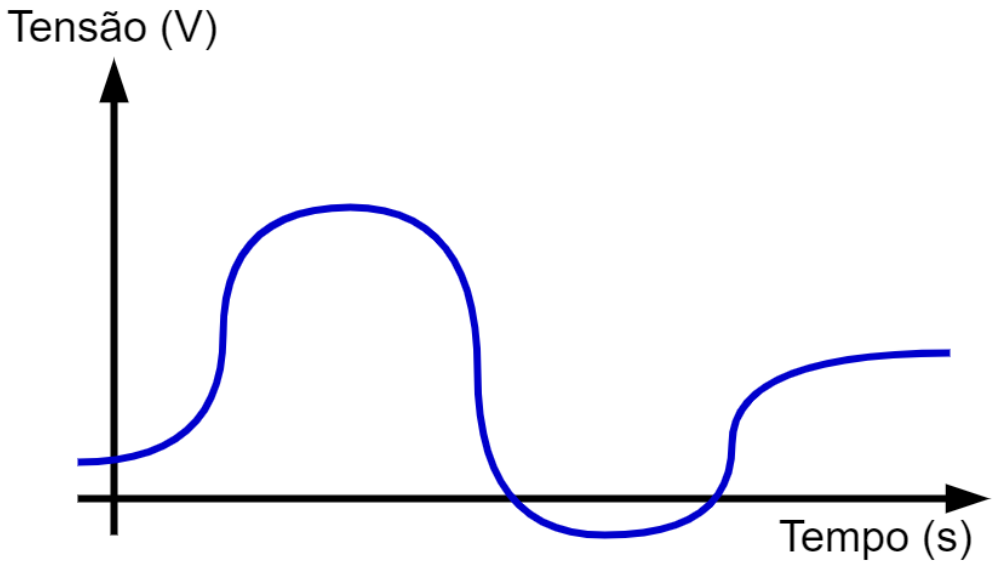
UFCD 0825 – Tipologias de Redes

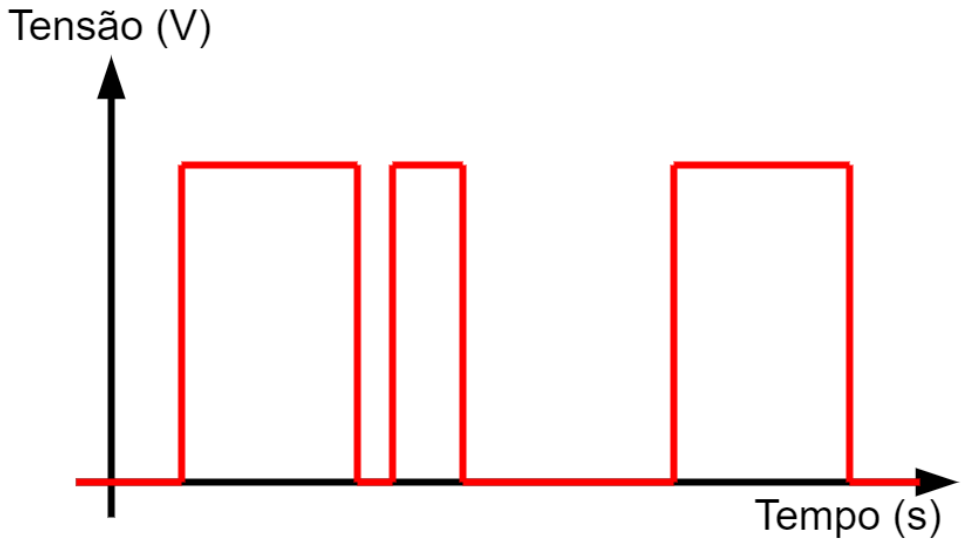


Prof. Nuno Ramos

Conceitos Introdutórios

- Os Sistemas digitais ou circuitos digitais, ou ainda circuitos lógicos;
- São definidos como circuitos eletrônicos que empregam a utilização de sinais elétricos em apenas dois níveis de tensão de forma a definir uma representação de valores binária.
- Os sinais elétricos transportados por estes circuitos são chamados de sinais digitais ou sinais binários





O que é um Sistema Digital?

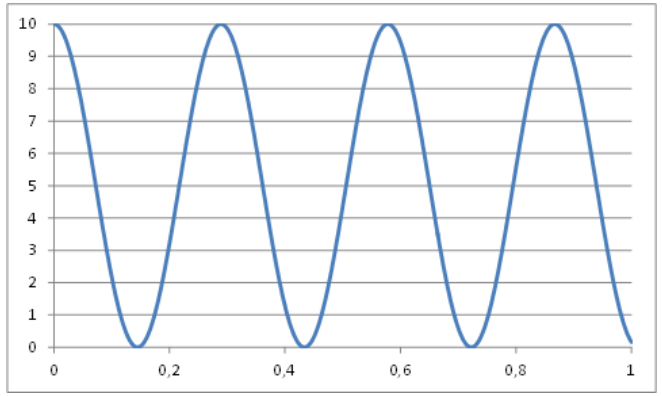
- **Um sistema que trabalha com sinais digitais**
  (em oposição aos sistemas analógicos, que trabalham com sinais analógicos)

- **Sinal analógico:**
  - Quantidade do mundo real medida continuamente no tempo
  - O valor medido pertence ao conjunto dos números reais

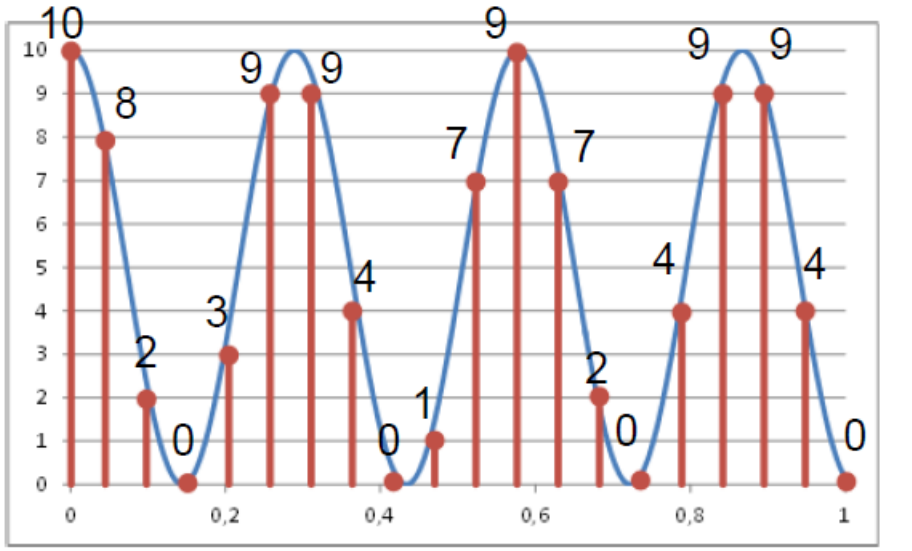
- **Sinal digital:**
  - Quantidade do mundo real medida em intervalos de tempo discretos
  - A medição (valor) pertence ao conjunto dos números racionais

Sinal Analógico vs Sinal Digital

- Medido continuamente no tempo
- As medições são valores reais



- Medições discretas com valores racionais



10, 8, 2, 0, 3, 9, 9, 4, 0, 1, 7, 9, 7, 2, 0, 4, 9, 9, 4, 0,...

- Os computadores representam os sinais digitais apenas por 0's e 1's!

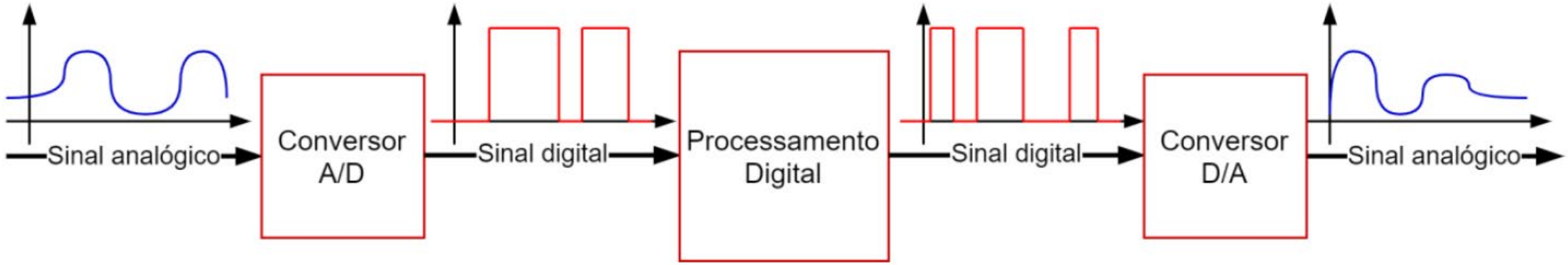
Porquê usar sinais digitais?

- Circuitos digitais são:
  - Consideravelmente baratos
  - Mais fáceis de desenhar que os circuitos analógicos
  - Permitem realizar cálculos avançados
  - Permitem guardar informação facilmente
  - Insensíveis a ruído

- As principais vantagens dos sinais digitais são:
  - Os sinais digitais são muito mais imunes a distorções, ruídos e interferências.
  - Os circuitos digitais são mais confiáveis e robustos.
  - Os circuitos digitais são fáceis de projetar e mais baratos.
  - A implementação de hardware em circuitos digitais é mais flexível.

Conversores Analógicos para Digitais e Vice-Versa

- Os sinais do mundo físico são analógicos, é necessário convertê-los para sinais digitais e vice-versa sempre que os sinais digitais tenham que interagir com os sinais do meio físico.
- A maioria dos sistemas digitais é utilizado para processar sinais oriundos de sensores analógicos, como por exemplo, microfones, sensores de temperatura, sensores de luminosidade etc.
- Sinais são primeiro convertidos de analógicos para digitais para só então serem processados.
- Este processamento é então realizado por circuitos digitais projetados para este fim.
- Os resultados deste processamento são também sinais digitais, que devem então ser convertidos em sinais analógicos, utilizando um conversor digital analógico.
- Os sinais analógicos resultantes podem então ser enviado para atuadores, como por exemplo, alto falantes, motores etc.



A origem dos números

- No sistema de numeração utilizado no dia a dia, usamos um sistema com dez símbolos para representar os números existentes.
- Esses símbolos vão de 0 a 9 e representam o sistema de numeração decimal, precisamente por conter 10 símbolos diferentes, denominados algarismos.
- Para os números superiores a 9 é usada uma convenção de escrita, que atribui um significado diferente ao local onde é colocado o novo dígito.
- Por exemplo, em virtude das posições ocupadas o número 6903 tem um significado numérico calculado da forma:

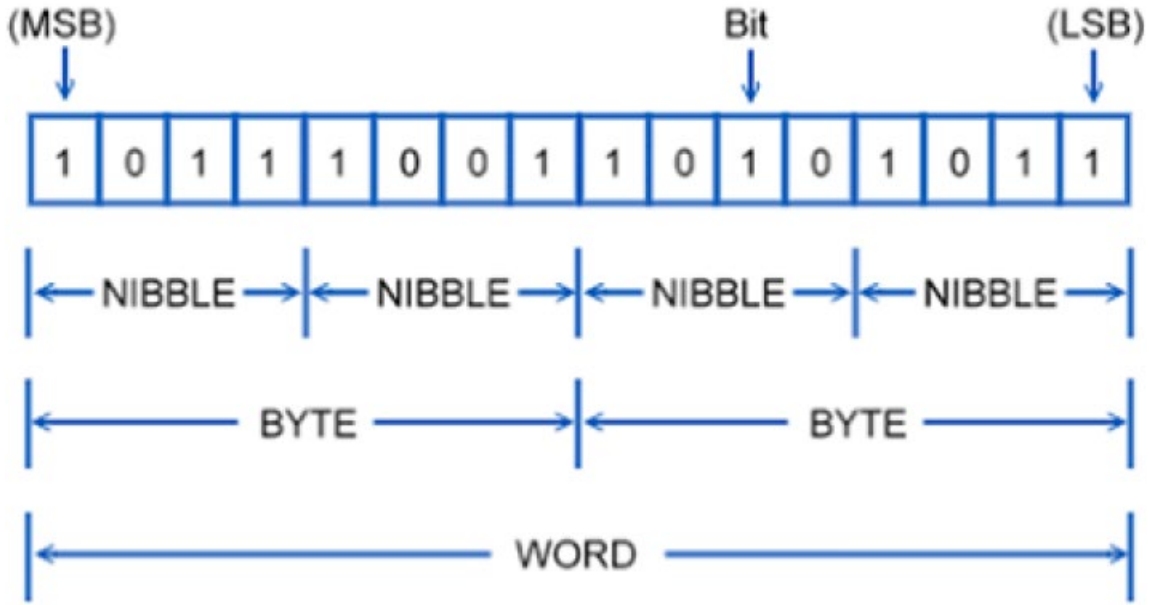
$$6903 = 6 \times 1000 + 9 \times 100 + 0 \times 10 + 3$$

- Ou colocando sob a forma de potências de 10:

$$6903 = 6 \times 10^3 + 9 \times 10^2 + 0 \times 10^1 + 3 \times 10^0$$

Sinais digitais Vs bit (Binary digit)

- A nomenclatura utilizada para os sinais digitais é o bit (Binary digit). A seguir são apresentadas as principais notações utilizadas para grupos de bits.
  - Um sinal composto por apenas um bit é um sinal binário único.
  - Um sinal composto por quatro Bits é chamado de nibble.
  - Um sinal composto por oito Bits é chamado de byte.
  - Um sinal composto por dezesseis Bits é chamado de word.



Bytes

- Um bit é raro aparecer sozinho.
- Os primeiros computadores a aparecer no mundo, trabalhavam com conjuntos de 8 bits.
- A este conjunto de 8 bits convencionou-se chamar byte (tradução octeto).
- À medida que os computadores iam evoluindo tecnologicamente, o conjunto de bits com que trabalhavam também o foi aparecendo outros nomes para conjuntos de bits diferentes.

Número de Bits	Nome
4	Nibble
8	Byte
16	Word (palavra)
32	Double Word
64	Long Word

Potenciação em base binária

- Com dois bits poderemos representar 22 números, ou seja, 4.
- Com três bits poderemos representar 23 números, ou seja 8.
- Com um byte, ou seja, 8 bits, poderemos então representar 28 números. Fazendo as contas, obtemos 256 números diferentes, com apenas 8 bits.
- Assim, poderemos construir uma tabela de potenciação com os valores possíveis de obter em base binária, para determinado número de bits.

Número de bits	Potenciação	Quantidade
1	2^1	2
2	2^2	4
3	2^3	8
4	2^4	16
5	2^5	32
6	2^6	64
7	2^7	128
8	2^8	256
9	2^9	512
10	2^{10}	1024

Contagem de Bytes

MEDIDA	SIMBOLOGÍA	EQUIVALENCIA	EQUIVALENCIA EN BYTES
Byte	b	8 bits	1
kilobyte	KB	1024 bytes	1 024
megabyte	MB	1024 KB	1 048 576
gigabyte	GB	1024 MB	1 073 741 824
terabyte	TB	1024 GB	1 099 511 627 776
petabyte	PB	1024 TB	1 125 899 906 842 624
exabyte	EB	1024 PB	1 152 921 504 606 846 976
zettabyte	ZB	1024 EB	1 180 591 620 717 411 303 424
yottabyte	YB	1024 ZB	1 208 925 819 614 629 174 706 176
brontobyte	BB	1024 YB	1 237 940 039 285 380 274 899 124 224
geopbyte	GEB	1024 BB	1 267 650 600 228 229 401 496 703 205 376

Sistemas de numeração

- Vamos estudar neste ponto quatro bases de numeração: binária , octal , decimal , hexadecimal .

Base	Alfabeto															
binária	0	1														
octal	0	1	2	3	4	5	6	7								
decimal	0	1	2	3	4	5	6	7	8	9						
hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Sistema decimal

- O sistema decimal é o mais conhecido sistema de numeração e é amplamente empregado em todo o mundo.
- Este sistema utiliza um conjunto de 10 caracteres ou símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9) para representar ou números.
- Estes caracteres são também chamados de dígitos.
- Como são 10 caracteres que compõe a base do sistema decimal, este sistema é também conhecido como sistema de base 10.

10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}
5	4	3	2 ,	7	8	9

$$5 * 10^3 + 4 * 10^2 + 3 * 10^1 + 2 * 10^0 + 7 * 10^{-1} + 8 * 10^{-2} + 8 * 10^{-3} = 5432,789$$

Sistema binário

- O sistema binário segue a mesma lógica, porém ele utiliza um conjunto de apenas 2 caracteres ou símbolos (0 e 1) para representar ou números.
- Este sistema de numeração é chamado de sistema de base 2.
- Quando a base de um número é diferente da base 10 é comum colocar-se o número da base subscrita no final do número, assim o número anterior seria 1000110_2.

2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	32	16	8	4	2	1
1	0	0	0	1	1	0

$$1000110_2 = 1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 = 70_{10}$$

Sistema octal

- O sistema octal possui este nome porque utiliza um conjunto de 8 caracteres ou símbolos (0, 1, 2, 3, 4, 5, 6 e 7) para representar ou números.
- Assim este sistema de numeração é chamado de sistema de base 8.

8^3	8^2	8^1	8^0	8^{-1}	8^{-2}
4	2	0	5 ,	4	7

$$4205,47_8 = 4 * 8^3 + 2 * 8^2 + 0 * 8^1 + 5 * 8^0 + 4 * 8^{-1} + 7 * 8^{-2} = 2181,609375_{10}$$

- O sistema octal é importante para os estudos de sistemas digitais pois os números (0 a 7) podem ser representados por 3 bits, assim a representação binária de números em formato octal é facilitada.

4			2			0			5			4			7		
1	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1	1	1

Sistema hexadecimal

- O sistema hexadecimal é parecido com o sistema octal, porém utiliza 4 bits cara cada caractere.
- Utiliza um conjunto de 16 caracteres ou símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F) para representar ou números.
- Sistema de numeração é chamado de sistema de base 16.
- Neste sistema as letras A, B, C, D, E e F, ou seja, A=10, B=11, C=12, D=13, E=14 e F=15.

16^2	16^1	16^0	16^{-1}	16^{-2}
3	A	4 ,	F	2

$$3A4,F2_{16} = 3 * 16^2 + A * 16^1 + 4 * 16^0 + F * 16^{-1} + 2 * 16^{-2} = 932,9453125_{10}$$

- O sistema hexadecimal é importante para os estudos de sistemas digitais pois os números (0 a F) podem ser representados por 4 bits, assim a representação binária de números em formato hexadecimal é facilitada.

Hexadecimal	3		A		4		F		2	
Decimal	3		10		4		15		2	
Binário	0 0 1 1		1 0 1 0		0 1 0 0		1 1 1 1		0 0 1 0	

Decimal, Binário, Octal e Hexadecimal

Decimal	Binário	Octal	Hexa
0	0	0	0
1	1	1	1
2	10	2	2
3	11	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
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
Pesos →	16 8 4 2 1	8 1	16 1

Exemplos de Conversões – para Decimal

Pretende-se converter para decimal o seguinte número em binário $10110101_{(2)}$.

Resolução

Primeiro reescrevemos o número e por cima de cada dígito, numeramos a sua posição. Começa-se sempre no zero, numerando da direita para a esquerda.

Pos	7	6	5	4	3	2	1	0
	1	0	1	1	0	1	0	1$_{(2)}$

Para este número, o limite do somatório da fórmula geral será entre 0 e 7. A base na qual estamos a trabalhar é a base binária, pelo que, na fórmula, temos base = 2.
Finalmente, o dígito será o correspondente à posição. Preenchendo a fórmula geral, temos:

$$\sum_{Pos=0}^{7} dígito * 2^{Pos}$$

A primeira iteração será:

Pos	7	6	5	4	3	2	1	0
	1	0	1	1	0	1	0	1$_{(2)}$

1×2^0

Agora temos de fazer o mesmo para os restantes dígitos, somando-os como se mostra a seguir.

$$\sum_{Pos=0}^{7} dígito * 2^{Pos}$$

$$= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 + 0 \times 2^6 + 1 \times 2^7 = 181_{(10)}$$

Que número representa $2374_{(8)}$ em decimal?

Resolução

Seguindo a lógica do exercício anterior, comecemos por numerar as posições:

Pos	3	2	1	0
	2	3	7	4$_{(8)}$

Base = 8
Pos → de 0 a 3.

Escrevendo os dígitos da direita para a esquerda, temos:

$$2374_{(8)} = \sum_{Pos=0}^{3} dígito * 8^{Pos} = 4 \times 8^0 + 7 \times 8^1 + 3 \times 8^2 + 2 \times 8^3 = 1276_{(10)}$$

Como se pode constatar, a única diferença entre o método de resolução anterior (binário → decimal) e este é a base. Neste caso particular, a base é 8, enquanto no anterior a base era 2. Todo o resto se mantém.

Qual o número decimal a que corresponde $3D6_{(16)}$?

Resolução

Mais uma vez, começamos por numerar as posições.

Pos	2	1	0
	3	*D*	6

Base = 16
Pos → de 0 a 2.

Escrevendo os dígitos da direita para a esquerda, temos:

$$3D6_{(16)} = \sum_{Pos=0}^{2} dígito * 16^{Pos} = 6 \times 16^0 + D \times 16^1 + 3 \times 16^2$$

Parece haver um problema na expressão anterior. Como vamos multiplicar letras por números? A resposta é muito simples: através de uma tabela de correspondência entre os valores hexadecimais e decimais, a conversão é imediata.

Exemplos de Conversões – Decimal para 2, 8 e 16

Converta o número $463_{(10)}$ para binário.

Resolução

$N.°_{(10)} = 463_{(10)}$

Base = 2.

Recorrendo às três expressões, temos, assim:

1.ª Iteração	2.ª Iteração	3.ª Iteração
$463_{(10)} \div 2 = 231{,}5$	$231 \div 2 = 115{,}5$	$115 \div 2 = 57{,}5$
$231 \times 2 = 462$	$115 \times 2 = 230$	$57 \times 2 = 114$
$463 - 462 = 1 \rightarrow$ Resto	$231 - 230 = 1 \rightarrow$ Resto	$115 - 114 = 1 \rightarrow$ Resto
4.ª Iteração	**5.ª Iteração**	**6.ª Iteração**
$57 \div 2 = 28{,}5$	$28 \div 2 = 14$	$14 \div 2 = 7$
$28 \times 2 = 56$	$14 \times 2 = 28$	$7 \times 2 = 14$
$57 - 56 = 1 \rightarrow$ Resto	$28 - 28 = 0 \rightarrow$ Resto	$14 - 14 = 0 \rightarrow$ Resto
7.ª Iteração	**8.ª Iteração**	**9.ª Iteração**
$7 \div 2 = 3{,}5$	$3 \div 2 = 1{,}5$	Como $1 < 2$ então paramos e fica o 1 como resto
$3 \times 2 = 6$	$1 \times 2 = 2$	Resto → 1
$7 - 6 = 1 \rightarrow$ Resto	$3 - 2 = 1 \rightarrow$ Resto	

O resultado escreve-se do dígito mais significativo (MSB – Most Significant Bit) para o menos significativo (LSB – Least Significant Bit). Escrevemos o resultado a partir do resto da 9.ª iteração para a 1.ª iteração

$\overset{MSB}{1}\ 1100111\ \overset{LSB}{1}_{(2)}$.

Então, $463_{(10)} = 111001111_{(2)}$.

Converta o número $524_{(10)}$ para octal.

Resolução

$N.°_{(10)} = 524_{(10)}$

Base = 8.

1.ª Iteração	2.ª Iteração	3.ª Iteração
$524_{(10)} \div 8 = 65{,}5$	$65 \div 8 = 8{,}125$	$8 \div 8 = 1$
$65 \times 80 = 520$	$8 \times 8 = 64$	$1 \times 8 = 8$
$524 - 520 = 4 \rightarrow$ Resto	$65 - 64 = 1 \rightarrow$ Resto	$8 - 8 = 0 \rightarrow$ Resto
4.ª Iteração		
Como $1 < 8$, então paramos e fica o 1 como resto.		
Resto → 1		

O resultado, mais uma vez, é dado pela leitura invertida das iterações. Neste caso, começamos no resto da 4.ª iteração e terminamos na 1.ª iteração: $524_{(10)} = 1014_{(8)}$.

Converta o número $1081_{(10)}$ para hexadecimal.

Resolução

$N.°_{(10)} = 1081_{(10)}$

Base = 16.

1.ª Iteração	2.ª Iteração	3.ª Iteração
$1081_{(10)} \div 16 = 67{,}5625$	$67 \div 16 = 4{,}1875$	Como $4 < 16$, então paramos e fica o 4 como resto.
$67 \times 6 = 1072$	$4 \times 16 = 64$	Resto → 4
$1081 - 1072 = 9 \rightarrow$ Resto	$67 - 64 = 3 \rightarrow$ Resto	

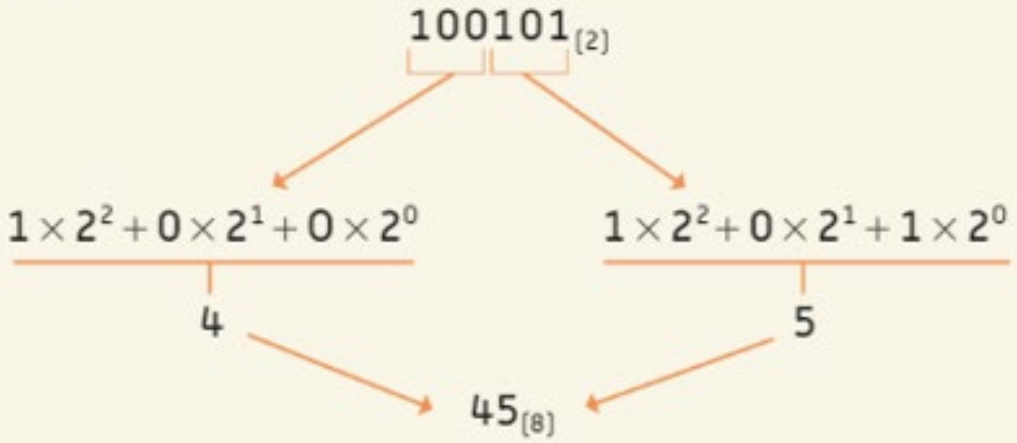
Escrevendo, mais uma vez, o resultado da última iteração para a primeira, temos:

$1081_{(10)} = 439_{(16)}$

Exemplos de Conversões – Binário Vs Octal

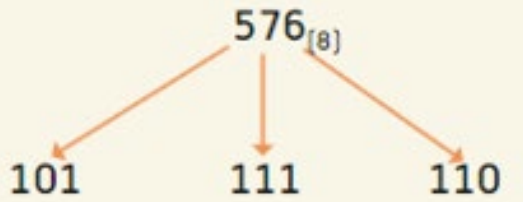
Converta o número $100101_{(2)}$ para octal.

Resolução



Converta o número $576_{(8)}$ para binário.

Resolução



Agora, cada dígito que constitui o número em octal tem de ser passado para um conjunto de 3 bits em binário. O resultado será o agrupar desses conjuntos de bits. Neste caso, o resultado será $576_{(8)} = 101111110_{(2)}$.

Exemplos de Conversões – Binário Vs Hexadecimal

Converta o número $1101101_{(2)}$ para hexadecimal.

Resolução

Primeiro agrupamos em grupos de 4 bits, da direita para a esquerda.

$$?1101101_{(2)}$$

No entanto, parece que temos um bit a menos no segundo grupo. O que fazer? Para resolver esta situação, necessitamos de acrescentar um zero à esquerda do número. Os zeros à esquerda não alteram o número, pelo que podem ser acrescentados quantos forem necessários.

$$01101101_{(2)}$$

$$0\times2^0+1\times2^1+1\times2^2+0\times2^3 \qquad 1\times2^0+0\times2^1+1\times2^2+1\times2^3$$

$$6\ D_{(16)}$$

Pretende-se converter o número $24A8_{(16)}$ para binário.

Resolução

$$24A8_{(16)}$$

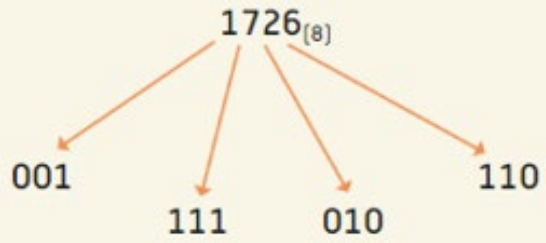
0010 0100 1010 1000

O resultado será o agrupar destes conjuntos de 4 bits. Neste caso, o resultado será $24A8_{(16)} = 0010010010101000_{(2)}$.

Exemplos de Conversões – Hexadecimal Vs Octal

Converta o número $1726_{(8)}$ para hexadecimal.

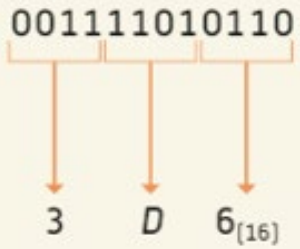
Resolução



Primeiro, passamos cada dígito do número octal para binário (ver a tabela anterior) em grupos de 3 bits. De seguida, agrupamos todos os conjuntos de 3 bits, resultando em $001111010110_{(2)}$.

Para se obter o número em hexadecimal, temos agora de agrupar este resultado em grupos de 4 bits e fazer a conversão tal e qual aprendemos anteriormente.

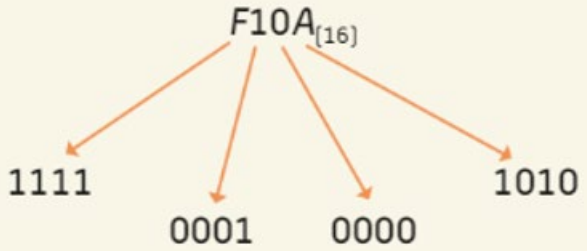
O resultado será, então, $1726_{(8)} = 3D6_{(16)}$.



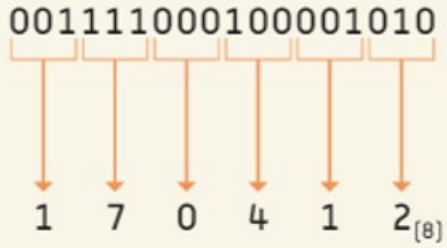
Converta o número $F10A_{(16)}$ para octal.

Resolução

A diferença deste método para o anterior reside na quantidade de bits a agrupar. No método anterior, começámos por fazer a correspondência de cada dígito para 3 bits, agrupando-os de seguida em 4 bits para obter o resultado. Neste exercício, primeiro fazemos corresponder cada dígito a 4 bits, agrupando-o de seguida em grupos de 3 bits para obter o resultado final. Vejamos a resolução:



O resultado será $1111000100001010_{(2)}$. Agora temos de agrupar em grupos de 3 bits, para obter o número em octal. Assim:



$F10A_{(16)} = 170412_{(8)}$.

Resumo de Conversões

Binário

Octal
Agrupar o número em conjuntos de 3 bits e ir à tabela ver o número correspondente.

Hexadecimal
Agrupar o número em conjuntos de 4 bits e ir à tabela ver o número correspondente.

Decimal

$$\sum_{Pos+direita}^{Pos+esquerda} dígito*2^{Pos}$$

Octal

Binário
Ver na tabela o conjunto de 3 bits que corresponde a cada número octal.

Hexadecimal
1.º passo – passar para binário (grupos de 3 bits)
2.º passo – passar de binário para hexadecimal.

Decimal

$$\sum_{Pos+direita}^{Pos+esquerda} dígito*8^{Pos}$$

Decimal

Binário
$N.^{o}_{(10)} \div 2 = x$
$x_{inteiro} \longrightarrow resto = 0$
$x_{não\ inteiro} \longrightarrow resto = 1$

Octal
$N.^{o}_{(10)} \div 8 = x$
$x_{inteiro} \times 8 = y$
$N.^{o}_{(10)} - y = resto$

Hexadecimal
$N.^{o}_{(10)} \div 16 = x$
$x_{inteiro} \times 16 = y$
$N.^{o}_{(10)} - y = resto$

Hexadecimal

Binário
Ver na tabela o conjunto de 4 bits que corresponde a cada número hexadecimal.

Octal
1.º passo – passar para binário (grupos de 4 bits)
2.º passo – passar de binário para octal

Decimal

$$\sum_{Pos+direita}^{Pos+esquerda} dígito*16^{Pos}$$

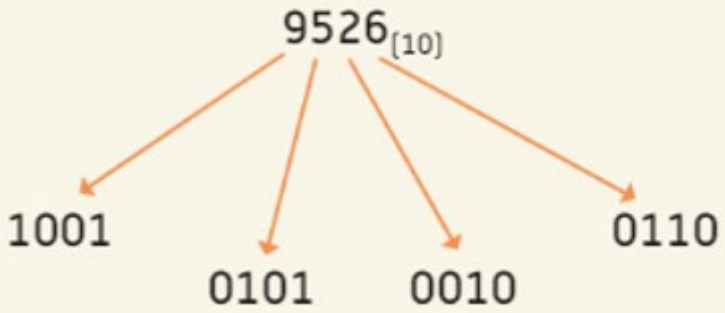
Sistema BCD (do inglês Binary Coded Decimal)

- É uma forma diferente de codificar números decimais em binário.
- Nesta codificação, cada dígito de um número decimal é codificado separadamente por uma combinação de 4 bits.

Codifique o número $9526_{(10)}$ em BCD 8421.

Resolução

Sabendo que cada dígito terá de ser codificado em 4 bits, teremos um resultado final com 16 bits. Assim, analisando cada dígito separadamente, temos:



O resultado final será o agrupar destes conjuntos de 4 bits, ou seja, $1001010100100110_{(BCD)}$.

Tabela de Valores			
Decimal	Binário	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	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

ASCII (American Standard Code for Information Interchange)

- Conhecido por associar conjuntos de 7 bits a carateres alfanuméricos.

7654321	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Código ASCII (American Standard Code for Information Interchange)

- é um código muito popular usado em todos os sistemas digitais, ele utiliza 7 bits para representar 128 caracteres;

Decimal	Binário	Símbolo	Decimal	Binário	Símbolo	Decimal	Binário	Símbolo	Decimal	Binário	Símbolo
0	0	NUL	32	100000	espaço	64	1000000	@	96	1100000	`
1	1	SOH	33	100001	!	65	1000001	A	97	1100001	a
2	10	STX	34	100010	"	66	1000010	B	98	1100010	b
3	11	ETX	35	100011	#	67	1000011	C	99	1100011	c
4	100	EOT	36	100100	$	68	1000100	D	100	1100100	d
5	101	ENQ	37	100101	%	69	1000101	E	101	1100101	e
6	110	ACK	38	100110	&	70	1000110	F	102	1100110	f
7	111	BEL	39	100111	'	71	1000111	G	103	1100111	g
8	1000	BS	40	101000	(	72	1001000	H	104	1101000	h
9	1001	HT	41	101001	)	73	1001001	I	105	1101001	i
10	1010	LF	42	101010	*	74	1001010	J	106	1101010	j
11	1011	VT	43	101011	+	75	1001011	K	107	1101011	k
12	1100	FF	44	101100	,	76	1001100	L	108	1101100	l
13	1101	CR	45	101101	-	77	1001101	M	109	1101101	m
14	1110	SO	46	101110	.	78	1001110	N	110	1101110	n
15	1111	SI	47	101111	/	79	1001111	O	111	1101111	o
16	10000	DLE	48	110000	0	80	1010000	P	112	1110000	p
17	10001	DC1	49	110001	1	81	1010001	Q	113	1110001	q
18	10010	DC2	50	110010	2	82	1010010	R	114	1110010	r
19	10011	DC3	51	110011	3	83	1010011	S	115	1110011	s
20	10100	DC4	52	110100	4	84	1010100	T	116	1110100	t
21	10101	NAK	53	110101	5	85	1010101	U	117	1110101	u
22	10110	SYN	54	110110	6	86	1010110	V	118	1110110	v
23	10111	ETB	55	110111	7	87	1010111	W	119	1110111	w
24	11000	CAN	56	111000	8	88	1011000	X	120	1111000	x
25	11001	EM	57	111001	9	89	1011001	Y	121	1111001	y
26	11010	SUB	58	111010	:	90	1011010	Z	122	1111010	z
27	11011	ESC	59	111011	;	91	1011011	[	123	1111011	{
28	11100	FS	60	111100	<	92	1011100	\	124	1111100	|
29	11101	GS	61	111101	=	93	1011101	]	125	1111101	}
30	11110	RS	62	111110	>	94	1011110	^	126	1111110	~
31	11111	US	63	111111	?	95	1011111	_	127	1111111	Delete

Tabela ASCII - versão expandida

- Esta versão expandida contempla os caracteres de 128 a 255.

Decimal	Binário	Símbolo	Decimal	Binário	Símbolo	Decimal	Binário	Símbolo	Decimal	Binário	Símbolo
128	10000000	Ç	160	10100000	á	192	11000000	└	224	11100000	Ó
129	10000001	ü	161	10100001	í	193	11000001	┴	225	11100001	ß
130	10000010	é	162	10100010	ó	194	11000010	┬	226	11100010	Ô
131	10000011	â	163	10100011	ú	195	11000011	├	227	11100011	Ò
132	10000100	ä	164	10100100	ñ	196	11000100	─	228	11100100	õ
133	10000101	à	165	10100101	Ñ	197	11000101	┼	229	11100101	Õ
134	10000110	å	166	10100110	ª	198	11000110	ã	230	11100110	µ
135	10000111	ç	167	10100111	º	199	11000111	Ã	231	11100111	þ
136	10001000	ê	168	10101000	¿	200	11001000	╚	232	11101000	Þ
137	10001001	ë	169	10101001	®	201	11001001	╔	233	11101001	Ú
138	10001010	è	170	10101010	¬	202	11001010	╩	234	11101010	Û
139	10001011	ï	171	10101011	½	203	11001011	╦	235	11101011	Ù
140	10001100	î	172	10101100	¼	204	11001100	╠	236	11101100	ý
141	10001101	ì	173	10101101	¡	205	11001101	═	237	11101101	Ý
142	10001110	Ä	174	10101110	«	206	11001110	╬	238	11101110	¯
143	10001111	Å	175	10101111	»	207	11001111	¤	239	11101111	´
144	10010000	É	176	10110000	░	208	11010000	ð	240	11110000	
145	10010001	æ	177	10110001	▒	209	11010001	Ð	241	11110001	±
146	10010010	Æ	178	10110010	▓	210	11010010	Ê	242	11110010	‗
147	10010011	ô	179	10110011	│	211	11010011	Ë	243	11110011	¾
148	10010100	ö	180	10110100	┤	212	11010100	È	244	11110100	¶
149	10010101	ò	181	10110101	Á	213	11010101	ı	245	11110101	§
150	10010110	û	182	10110110	Â	214	11010110	Í	246	11110110	÷
151	10010111	ù	183	10110111	À	215	11010111	Î	247	11110111	¸
152	10011000	ÿ	184	10111000	©	216	11011000	Ï	248	11111000	°
153	10011001	Ö	185	10111001	╣	217	11011001	┘	249	11111001	¨
154	10011010	Ü	186	10111010	║	218	11011010	┌	250	11111010	·
155	10011011	ø	187	10111011	╗	219	11011011	█	251	11111011	¹
156	10011100	£	188	10111100	╝	220	11011100	▄	252	11111100	³
157	10011101	Ø	189	10111101	¢	221	11011101	¦	253	11111101	²
158	10011110	×	190	10111110	¥	222	11011110	Ì	254	11111110	■
159	10011111	ƒ	191	10111111	┐	223	11011111	▀	255	11111111	

Ficha de Trabalho 2 - Sistemas Digitais – Bases de Numeração

- Pesquise e apresente uma tabela com contagem de Bytes.
- Pesquise e apresente uma tabela para as quatro bases de numeração: binária, octal, decimal, hexadecimal.
- Explique como funciona o sistema Binário. Poderá usar imagens para ajudar na explicação.
- Explique como funciona o sistema Octal. Poderá usar imagens para ajudar na explicação.
- Explique como funciona o sistema Hexadecimal. Poderá usar imagens para ajudar na explicação.
- Diga o que entende pelo Código ASCII (American Standard Code for Information Interchange).
- Assinale com uma cruz identificando na tabela, em que sistemas numéricos podem estar representados os seguintes números:
- Converta os valores para Bytes .
  - a) 1 kB = b) 23 kB = c) 755 MB = d) 6 GB =
- Quantas combinações consegue com:
  - a) 1 bit b) 2 Byte c) 4 bit d) 16 bit
- Complete a seguinte tabela:

Número de bits	Nome
1	bit
4	
	Byte
16	
	Double Word
64	