Apêndice 2

Sistemas de numeração

Bit e byte

BIT é a contração de **BI**nary digi**T**, que significa dígito binário. Bit é a menor unidade de informação do computador.

A seguir podemos observar a correspondência entre as unidades de informação, ou seja, como se mede a informação:

Unidade de medida	Prefixo*	Valor	Quantidade de caracteres	Base binária
1 byte	B	8 bits	1	2^{0}
1 Kilobyte	kB	1024 B	1.024	2^{10}
1 Megabyte	MB	1024 kB	1.048.576	2^{20}
1 Gigabyte	GB	1024 MB	1.073.741,824	2^{30}
1 Terabyte	TB	1024 GB	1.099.511.627.776	2^{40}
1 Petabyte	PB	1024 TB	1.125.899.906.842.620	2^{50}

* Os prefixos estão de acordo com o SI – SISTEMA INTERNACIONAL DE UNIDADES

Um quilo, nas medidas que usamos cotidianamente, representa o número 1000, que é resultado de uma potência de base 10 (decimal), ou seja, $10^{3} = 1000$. Sendo assim, por exemplo, um quilo de queijo é igual a 1000 gramas de queijo.

Um **byte** é uma estrutura fundamentada no código binário, ou seja, na base 2 (binário). Portanto, quando queremos um quilo de bytes, temos que elevar a base 2 a algum número inteiro até conseguir atingir a milhar. Como não há um número inteiro possível que atinja exatamente o valor 1000, então pegamos o próximo número superior a 1000, que é o 1024, ou seja, $2^{10} = 1024$. O número 2 elevado à nona potência é inferior a 1000 ($2^{9} = 512$).

Sistemas de numeração

Este ponto do nosso estudo é importante, pois a utilização de códigos em sistemas com bases diferentes de 10 (decimal) é comum no ambiente de processamento de dados.

Nós estamos acostumados com o sistema decimal (base 10) de numeração, onde temos dez números. Vai de 0 a 9 e depois aumentamos uma casa e repetimos os dois primeiros números, que é o 10. Depois continuamos a repetir o número 1 e trocamos o 0 por 1 até 9, que são os números 11, 12, 13, 14, 15, 16, 17, 18 e 19. Depois vem o 20 e assim por diante.

Com o sistema binário (base 2) é a mesma coisa, só que temos apenas dois números, o zero (0) e o um (1). Vai de 0 a 1 e depois aumentamos uma casa e repetimos os dois primeiros números, resultando no 10. Depois vem o número 11 e como não tem outros algarismos, depois vem o 100, 101, 110, 111, 1000 e assim por diante.

Com o sistema hexadecimal (base 16) é a mesma coisa, só que temos 16 números. Vai de 0 a 9, de A a F e depois aumentamos uma casa e repetimos os dois primeiros números, que é o 10. Depois continuamos a repetir o número 1 e trocamos o 0 por 1 até 9, e depois de A a F, que são os números 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E e 1F. Depois vem o 20 e assim por diante.

A seguir temos a tabela de conversão dos sistemas de numeração:

Decimal	Binário	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Sistema Decimal

Para entender este sistema, basta decompor qualquer número inteiro em potência de base dez. Vejamos um exemplo:

$$\begin{aligned} 1024_{(10)} &= 1*(10)^3 + 0*(10)^2 + 2*(10)^1 + 4*(10)^0 \\ &= 1*1000 + 0*100 + 2*10 + 4*1 \\ &= 1000 + 0 + 20 + 4 \end{aligned}$$

Logo, pode-se concluir que o número nada mais é que o conjunto de coeficientes das potências de 10. É importante observar que, por ser base decimal, os dígitos disponíveis são: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.

Sistema Binário

Neste sistema, a base é 2 e os dígitos disponíveis são 0 e 1.
Este sistema de numeração constitui o alfabeto interno dos computadores.

Sistema Hexadecimal

A base deste sistema é 16 e os dígitos disponíveis são: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F. Este sistema simplifica a representação dos números, porque diminui a quantidade de algarismos.

Conversão de sistemas de numeração

A seguir, veremos como fazer a conversão de Binário para Decimal, de Hexadecimal para Decimal, de Binário para Hexadecimal, de Hexadecimal para Binário, de Decimal para Binário e de Decimal para Hexadecimal.

Conversão de Binário para Decimal

Para fazer esta conversão, utilizaremos a soma das multiplicações das potências de base 2.

Vejamos um exemplo:

$10.000.000.000_{(2)}$
$= 1*(2)^{10}+0*(2)^{9}+0*(2)^{8}+0*(2)^{7}+0*(2)^{6}+0*(2)^{5}+0*(2)^{4}+0*(2)^{3}+0*(2)^{2}+0*(2)^{1}+0*(2)^{0}$
$= 1*1024+ 0*512+ 0*256+ 0*128+ 0*64+ 0*32+ 0*16+ 0*8+ 0*4+ 0*2+ 0*1$
$= 1024 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0$
$= 1024_{(10)}$

Como se pode observar, cada um dos algarismos em binário foi multiplicado por 2 (base 2 = binário) e depois, da direita para a esquerda, o número 2 foi elevado a partir de zero (0), sendo acrescido de um (1).

Vejamos outro exemplo:

$110110_{(2)}$
$= 1*(2)^5+1*(2)^4+0*(2)^3+1*(2)^2+1*(2)^1+0*(2)^0$
$= 1*32+ 1*16+ 0*8+ 1*4+ 1*2+ 0*1$
$= 32 + 16 + 0 + 4 + 2 + 0$
$= 54_{(10)}$

Conversão de Hexadecimal para Decimal

Para fazer esta conversão, utilizaremos a soma das multiplicações das potências de base 16.

Vejamos um exemplo:

$400_{(16)} = 4*(16)^2 + 0*(16)^1 + 0*(16)^0$
$= 4*256 + 0*16 + 0*1$
$= 1024 + 0 + 0$
$= 1024_{(10)}$

Como se pode observar, cada um dos algarismos em hexadecimal foi multiplicado por 16 (base 16 = hexadecimal) e depois, da direita para a esquerda, o número 16 foi elevado a partir de zero (0), sendo acrescido de um (1).

Vejamos outro exemplo:

$17A5_{(16)} = 1*(16)^3 + 7*(16)^2 + A*(16)^1 + 5*(16)^0$
$= 1+ 4096 + 7*256 + 10*16 + 5*1$
$= 4096 + 1792 + 160 + 5$
$= 6053_{(10)}$

Note, na conversão acima, que o número **A** em hexadecimal foi substituído pelo número **10** em decimal para que a conta fosse feita. Foi utilizado a tabela de conversão dos sistemas de numeração.

Conversão de Binário para Hexadecimal

Para fazer esta conversão, divide-se o número binário em grupos de quatro algarismos, começando da direita para a esquerda. Depois, utilizando a tabela de conversão de sistemas de numeração, substitui-se, então, cada um dos grupos de quatro algarismos por seu correspondente hexadecimal.

Vejamos um exemplo:

1101 0100 1111 $0011_{(2)}$
D 4 F $3_{(16)}$

Caso o último grupo da esquerda não contenha quatro algarismos, basta complementar com zeros. Um número em decimal corresponde a quatro algarismos em binário. Lembre-se de que zeros à esquerda não são significativos.

Vejamos outro exemplo:

O número 1111001111 em binário possui dez algarismos, portanto receberá dois zeros à esquerda para que possamos separar em grupos de quatro algarismos:

0011 1100 $1111_{(2)}$
3 C $F_{(16)}$ ← utilizando a tabela de conversão de sistemas de numeração

Conversão de Hexadecimal para Binário

Nesta conversão, basta fazer o inverso do que fizemos na conversão de Binário para Hexadecimal. Utilizando a tabela de conversão de sistemas de numeração, substitui-se, então, cada número hexadecimal pelo seu correspondente em binário.

Vejamos um exemplo:

E 7 5 $A_{(16)}$
1110 0111 0101 $1010_{(2)}$

Vejamos outro exemplo:

1 B $2_{(16)}$
0001 1011 $0010_{(2)}$

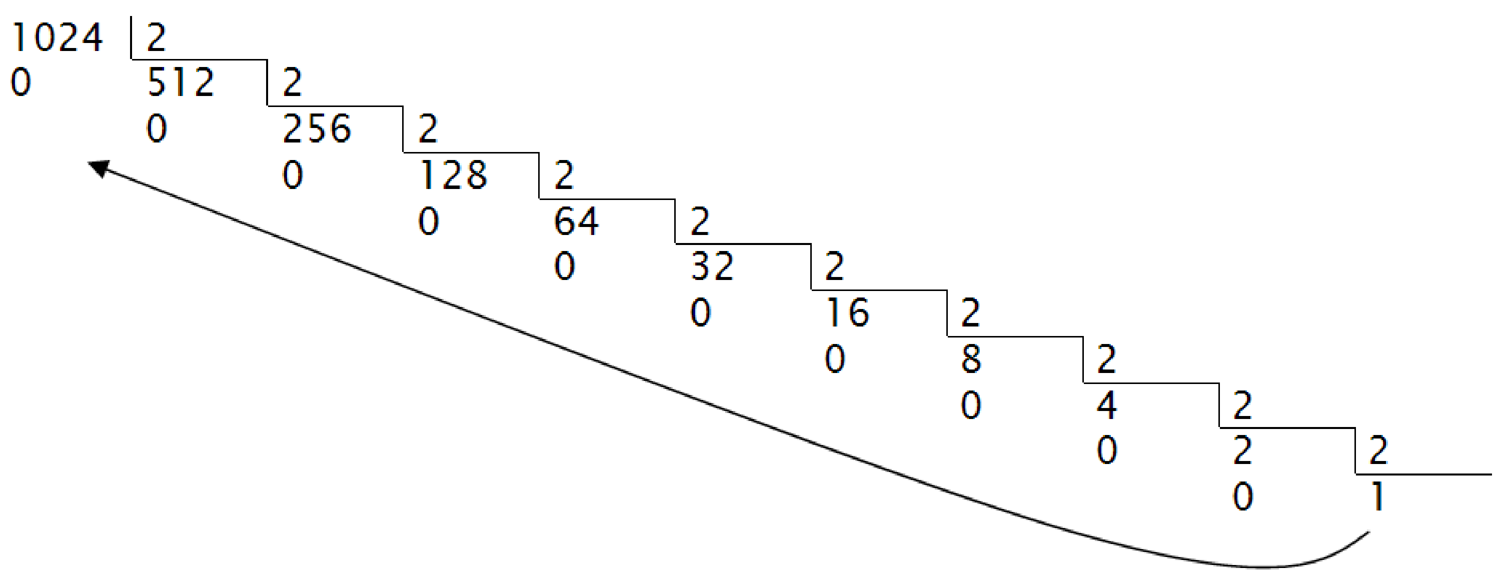
Decimal	Binário	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

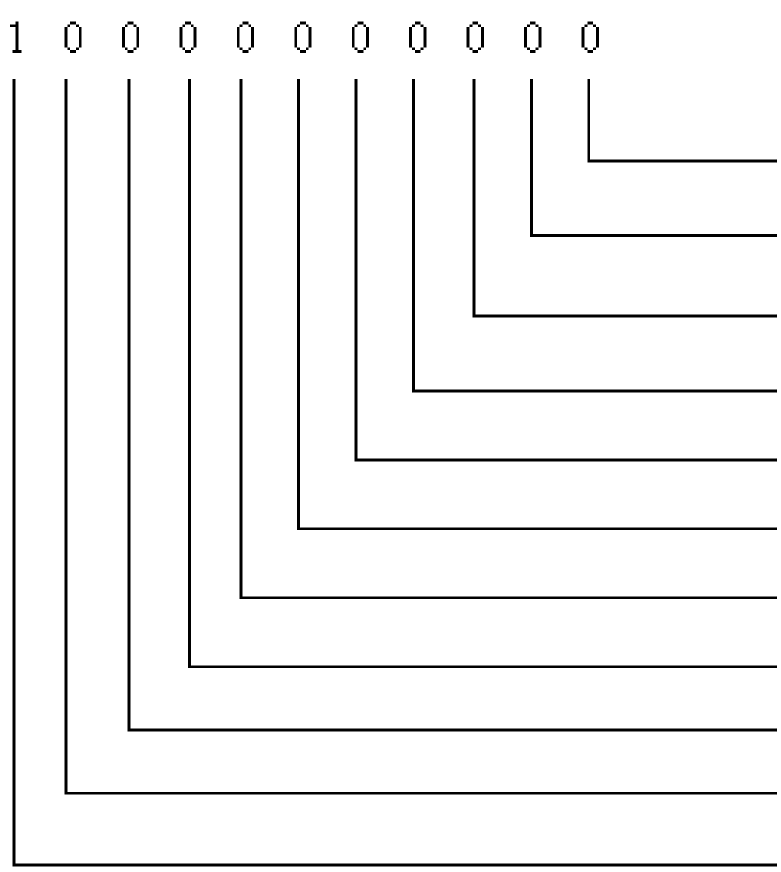
Conversão de Decimal para Binário

Para fazer esta conversão, basta realizar sucessivas divisões por dois e, a partir daí, devemos pegar sempre o último quociente, que será sempre 1, e ele será o primeiro número binário, e depois pegamos todos os restos na ordem da última divisão para a primeira.

- **Exemplo 1**

$1.024_{(10)} = 100\ 0000\ 0000_{(2)}$





$0*2^0 = 0*1$

$0*2^1 = 0*2$

$0*2^2 = 0*4$

$0*2^3 = 0*8$

$0*2^4 = 0*16$

$0*2^5 = 0*32$

$0*2^6 = 0*64$

$0*2^7 = 0*128$

$0*2^8 = 0*256$

$0*2^9 = 0*512$

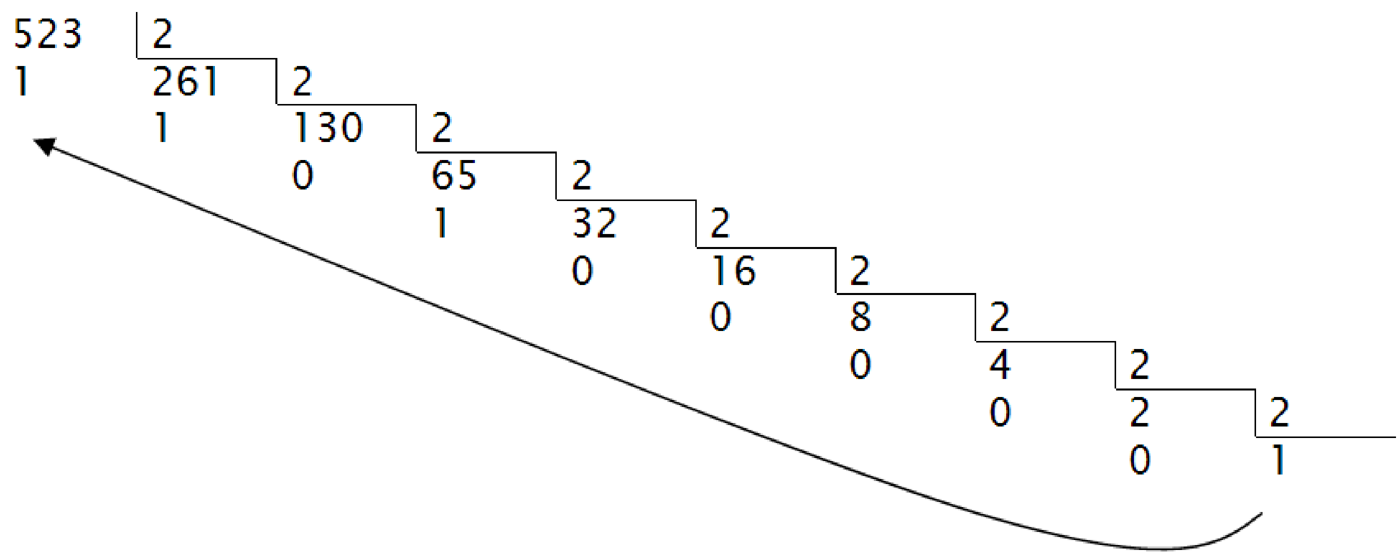
$1*2^{10} = 1024$

Total = $1024_{(10)}$

Tirando a Prova

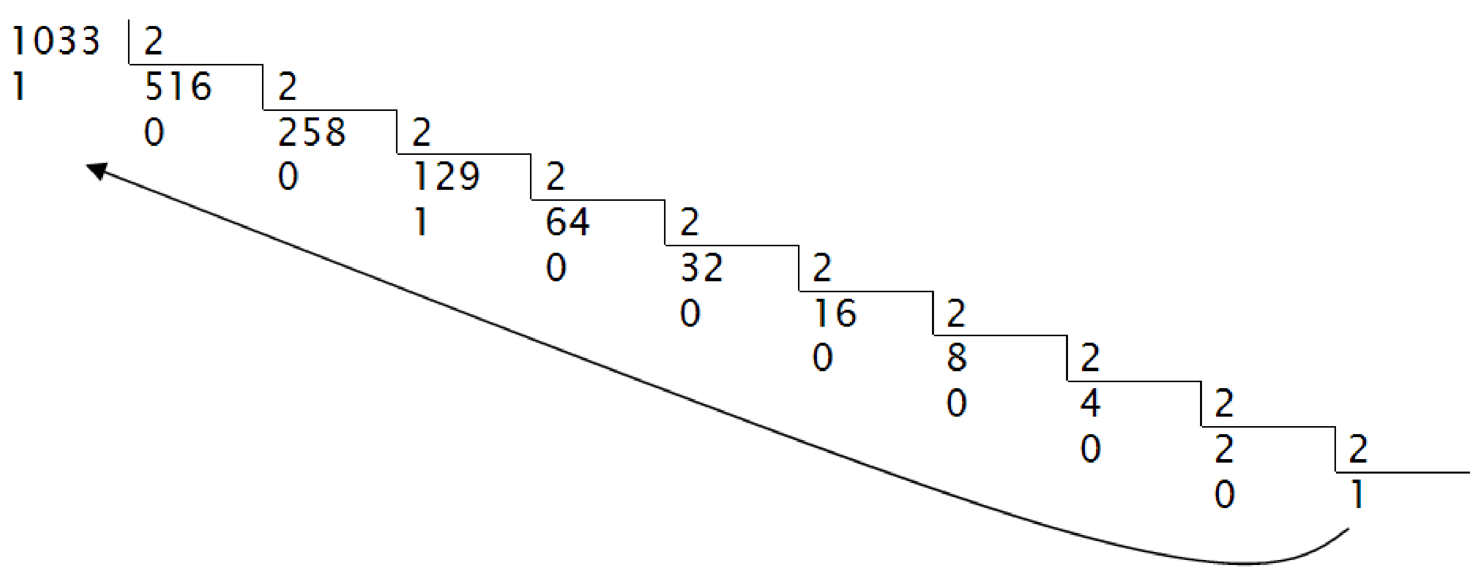
- **Exemplo 2**

$523_{(10)} = 10\ 0000\ 1011_{(2)}$



- **Exemplo 3**

$1.033_{(10)} = 100\ 0000\ 1001_{(2)}$

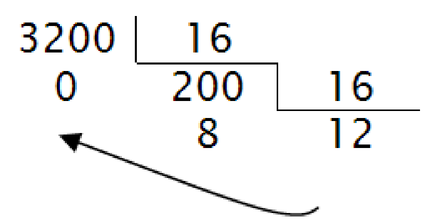


Conversão de Decimal para Hexadecimal

Para converter de decimal para hexadecimal, procede-se do mesmo modo que na conversão decimal para binário. Basta agora dividir por 16 e não mais por 2.

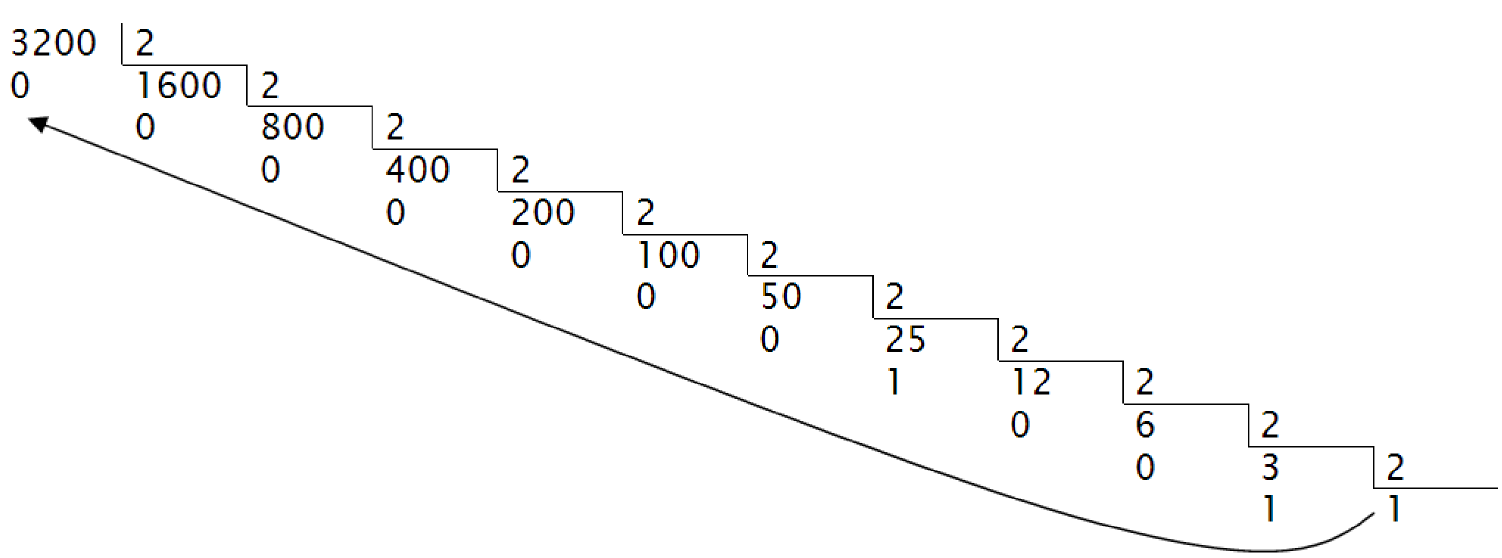
- **Exemplo**

$3200_{(10)} = C80_{(16)}$



Na tabela de conversão, $12_{(10)} = C_{(16)}$, portanto $3200_{(10)} = C80_{(16)}$.
Pegaremos sempre o último quociente, pois ele será o primeiro número hexadecimal, e depois pegamos todos os restos na ordem da última divisão para a primeira.

Outra forma de fazer esta conversão é converter primeiro de decimal para binário e, depois, converter de binário para hexadecimal utilizando a tabela de conversão dos sistemas de numeração.



$1100\ \ 1000\ \ 0000_{(2)}$
$\ \ C\ \ \ \ \ \ \ 8\ \ \ \ \ \ \ 0_{(16)}$

Forma rápida para conversão de sistemas de numeração

Para converter um número Decimal em Binário, usamos o método de sucessivas divisões por dois, pegando sempre o resto.

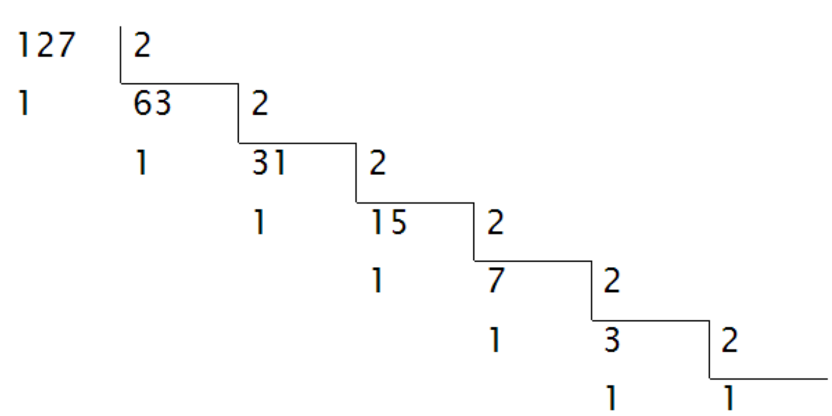
A forma rápida de conversão consiste em escrever numa linha os números da direita para a esquerda a partir do número 1 e, depois, sempre o dobro, ou seja, são as potências de 2.

Depois, na linha de baixo, coloque o número 1 abaixo do mesmo número decimal a ser convertido e preencha os demais números com 0.

- **Exemplo 1**

Qual a correspondência do número $127_{(10)}$ em binário?

$127_{(10)} = 111\ 1111_{(2)}$



Usando o esquema seguinte, encontramos uma forma rápida para fazer a conversão. Quando não houver o número a ser convertido na primeira linha, pegue o imediatamente menor, da esquerda para a direita, e complemente na mesma ordem até chegar no valor desejado.

128	64	32	16	8	4	2	1	
0	1	1	1	1	1	1	1	
	64 +	32 +	16 +	8 +	4 +	2 +	1	= 127

$$0111\ 1111_{(2)} = 127_{(10)}$$

- **Exemplo 2**

Qual a correspondência do número $4100_{(10)}$ em binário?

32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
			4096 +										4		

$$4100_{(10)} = 0001\ 0000\ 0000\ 0100_{(2)} = 1004_{(16)}$$

- **Exemplo 3**

Quanto corresponde o número decimal 4 em binário?

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	0

$$0100_{(2)} = 4_{(10)}$$

Ou seja, o número 4 em decimal corresponde ao número 100 em binário.

- **Exemplo 4**

Quanto corresponde o número decimal 108 em binário?

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	0

$$0110\ 1100_{(2)} = 108_{(10)}$$

Ou seja, o número 108 em decimal corresponde ao número 01101100 em binário.

Como não há exatamente o número 108 na linha de cima, pegamos o 64 e colocamos 1 abaixo.

Vemos quanto sobrou e continuamos a fazer a mesma coisa: 108 - 64 = 44. Como não há exatamente o número 44 na linha de cima, pegamos o 32 e colocamos 1 abaixo.

Vemos quanto sobrou e continuamos a fazer a mesma coisa: 44 - 32 = 12. Como não há exatamente o número 12 na linha de cima, pegamos o 8 e colocamos 1 abaixo.

Vemos quanto sobrou e continuamos a fazer a mesma coisa: 12 - 8 = 4. Como há exatamente o número 4 na linha de cima, colocamos 1 abaixo e terminamos.

A seguir, temos a tabela com as potências de 2 até o expoente 20 e com as potências de 16 até o expoente 5:

Potência de 2 = Valor	Potência de 16 = Valor
$2^0 = 1$	$16^0 = 1$
$2^1 = 2$	
$2^2 = 4$	
$2^3 = 8$	
$2^4 = 16$	$16^1 = 16$
$2^5 = 32$	
$2^6 = 64$	
$2^7 = 128$	
$2^8 = 256$	$16^2 = 256$
$2^9 = 512$	
$2^{10} = 1024$	
$2^{11} = 2048$	
$2^{12} = 4096$	$16^3 = 4096$
$2^{13} = 8192$	
$2^{14} = 16\ 384$	
$2^{15} = 32.768$	
$2^{16} = 65.536$	$16^4 = 65.536$
$2^{17} = 131.072$	
$2^{18} = 262.144$	
$2^{19} = 524.288$	
$2^{20} = 1.048.576$	$16^5 = 1.048.576$