

Sistemas numéricos

...

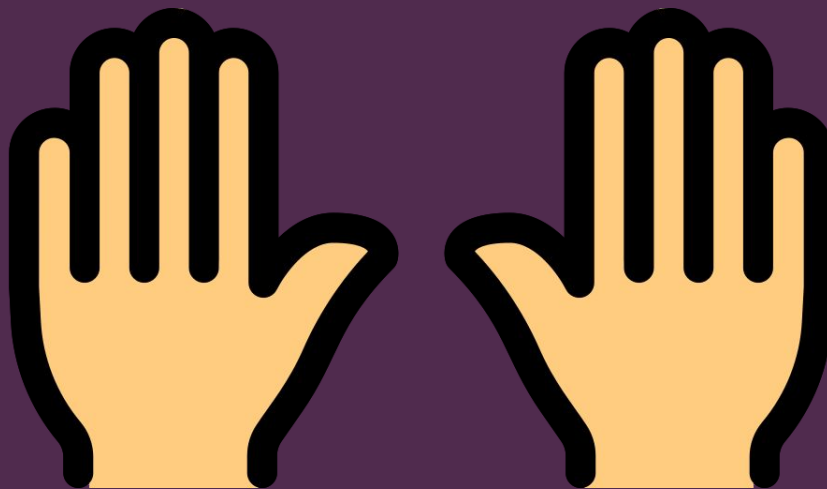
Bitabit Extra

Hay varios sistemas numéricos

- Los humanos usamos el sistema **decimal**, la base es **10**
- Para máquinas es más conveniente usar otros sistemas
 - **Binario** - base **2**
 - **Octal** - base **8**
 - **Hexadecimal** - base **16**
- Se indicará el sistema agregando d, b, o, h si es necesario
 - 100d
 - 100b
 - 100o
 - 100h

Sistema decimal (de a Diez)

- Tenemos 10 dedos para contar
- Creamos 10 símbolos llamados **Unidades**
 - 0
 - 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
 - 8
 - 9



Y para contar más allá de 10?

- Agregamos un dígito a la izquierda
- Le llamamos **Decenas** (por 10)

Valor	Decenas *10	Unidades *1	
18	1	8	1 *10 + 8 = 18
29	2	9	2 *10 + 9 = 29
99	9	9	9 *10 + 9 = 99

Y para contar más allá de 100?

- Agregamos otro dígito a la izquierda
- Le llamamos **Centenas** (por 100)

Valor	Centenas *100	Decenas *10	Unidades *1	
118	1	1	8	$1*100 + 1*10 + 8 = 118$
429	4	2	9	$4*100 + 2*10 + 9 = 429$
647	6	4	7	$6*100 + 4*10 + 7 = 647$
899	8	9	9	$8*100 + 9*10 + 9 = 899$

Y más allá?

- Cada nuevo dígito se multiplica por 10

Valor	Factor	Factorizado	Potencia
Unidades	* 1	* 1	* 10^0
Decenas	* 10	* 10	* 10^1
Centenas	* 100	* $10*10$	* 10^2
Miles	* 1000	* $10*10*10$	* 10^3
Decenas de Miles	* 10000	* $10*10*10*10$	* 10^4
Centenas de Miles	* 100000	* $10*10*10*10*10$	* 10^5
Millones	* 1000000	* $10*10*10*10*10*10$	* 10^6

En forma general

$$\begin{array}{c} N \mid \dots \mid M \mid C \mid D \mid U \\ \hline 10^n \mid \dots \mid 10^3 \mid 10^2 \mid 10^1 \mid 10^0 \end{array}$$

Dígitos **menos** significativos ->
<- Dígitos **más** significativos

Ejemplo: 7253

M | C | D | U

10^3 | 10^2 | 10^1 | 10^0

7 | 2 | 5 | 3

$$\begin{array}{ccccccc} 7 * 10^3 & + & 2 * 10^2 & + & 5 * 10^1 & + & 3 * 10^0 \\ 7 * 1000 & + & 2 * 100 & + & 5 * 10 & + & 3 * 1 \end{array}$$

7000 +

200 +

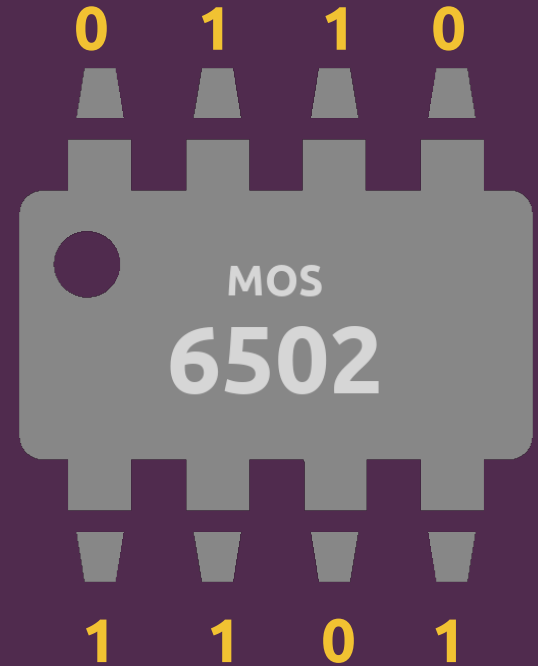
50 +

3 =

7253 .

Sistema Binario

- Sólo tiene dos símbolos: **0** y **1**
- Se usa en la electrónica digital
- Es la representación más simple:
 - On / Off
 - Sí / No
 - Activo / Inactivo



0, 1... y después?

- Se agrega un nuevo dígito a la izquierda
- En vez de multiplicar por 10, multiplicamos por **2**

Valor		*2		*1		
-----+-----+-----						
00		0		0		0 * 2 + 0 = 0d
01		0		1		0 * 2 + 1 = 1d
10		1		0		1 * 2 + 0 = 2d
11		1		1		1 * 2 + 1 = 3d

Y más allá aún?

- Se siguen agregando dígitos a la derecha
- Nuevamente multiplicados por 2

Valor		*4		*2		*1	
-----+-----+-----+-----+-----							
010		0		1		0	0 *4 + 1 *2 + 0 = 2d
101		1		0		1	1 *4 + 0 *2 + 1 = 5d
011		0		1		1	0 *4 + 1 *2 + 1 = 3d
111		1		1		1	1 *4 + 1 *2 + 1 = 7d

Y si se agregan más bits?

- Cada nuevo dígito se multiplica por 2

Bit	Factor	Factorizado	Potencia
0	* 1	* 1	* 2 ⁰
1	* 2	* 2	* 2 ¹
2	* 4	* 2*2	* 2 ²
3	* 8	* 2*2*2	* 2 ³
4	* 16	* 2*2*2*2	* 2 ⁴
5	* 32	* 2*2*2*2*2	* 2 ⁵

En forma general el valor de cada bit es...

bX | ... | b3 | b2 | b1 | b0

2^x | ... | 2^3 | 2^2 | 2^1 | 2^0

Bits **menos** significativos ->

<- Bits **más** significativos

Ejemplo número binario 100101

b5 | b4 | b3 | b2 | b1 | b0

2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0
1 | 0 | 0 | 1 | 0 | 1

$$\begin{aligned} &2^5 + 2^2 + 2^0 \Rightarrow \\ &32 + 4 + 1 = \\ &37 \end{aligned}$$

Ejemplo Aplicado

Gráfico que representa la letra A

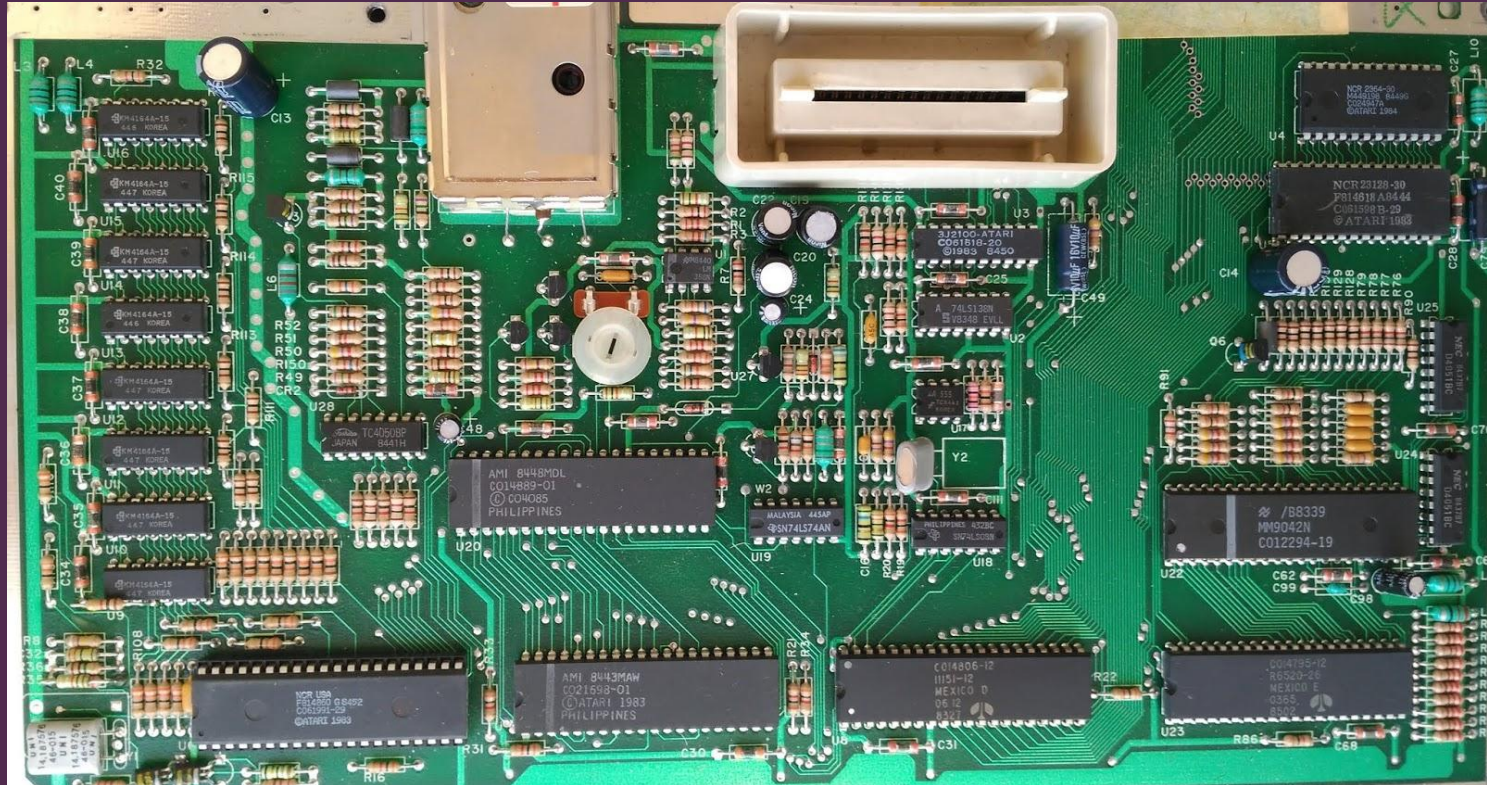


bit
76543210

16318421
2426
8

00000000	= 0	= 0
00011000	= 16 + 8	= 24
00111100	= 32 + 16 + 8 + 4	= 60
01100110	= 64 + 32 + 4 + 2	= 102
01100110	= 64 + 32 + 4 + 2	= 102
01111110	= 64 + 32 + 16 + 8 + 4 + 2	= 126
01100110	= 64 + 32 + 4 + 2	= 102
00000000	= 0	= 0

Bits a nivel electrónico



Rangos y valores interesantes

Rangos del sistema decimal

El rango depende de la cantidad de dígitos

X	1	dígito	:	10	elementos	($10^1 = 10$)
XX	2	dígitos	:	$10*10$	elementos	($10^2 = 100$)
XXX	3	dígitos	:	$10*10*10$	elementos	($10^3 = 1000$)
X..X	N	dígitos	:	$10*...*10$	elementos	(10^n)

Para **N** dígitos decimales el rango es **10^n**

Rangos del sistema binario

El rango depende de la cantidad de dígitos / bits

X	1 bit	:	2 elementos	($2^1 = 2$)
XX	2 bits	:	2*2 elementos	($2^2 = 4$)
XXX	3 bits	:	2*2*2 elementos	($2^3 = 8$)
X..X	N bits	:	2*...*2 elementos	(2^n)

Para **N** dígitos o bits el rango es 2^n

Algunos valores interesantes

Si para **N** dígitos o bits el rango es 2^n

Con **8** bits tenemos $2^8 = 256$ valores posibles (1 byte)

Con **10** bits tenemos $2^{10} = 1024$ valores posibles (kilo, mega, giga, tera, etc)

Con **16** bits tenemos $2^{16} = 65536$ valores posibles (64 kilobytes)

Otros sistemas numéricos

Sistema HexaDecimal

- En vez de usar 2 o 10, funciona en base a **16** elementos
- Se origina al dividir 8 bits en dos grupos de **4** bits llamados nibbles
- Con **4** bits tenemos $2*2*2*2 = 2^4 = \mathbf{16}$ valores

10010010

1001

0010

nibble

nibble

Símbolos HexaDecimales

Ahora se necesitan 16 símbolos

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

A = 10d

B = 11d

C = 12d

D = 13d

E = 14d

F = 15d

Bytes en formato HexaDecimal

Cada byte es representado por 2 dígitos hexadecimales

10010010

1001	0010
9	2

10010010b = 92h

Otros ejemplos

Byte 10100100

$$1010 \quad 0100 = 128 + 32 + 4 = 164$$

$$\begin{array}{c} A \quad 4 \\ 10 * 16 + 4 = 160 + 4 = 164 \end{array}$$

Hex = A4

Byte 11111111

$$1111 \quad 1111 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

$$\begin{array}{c} F \quad F \\ 15 * 16 + 15 = 240 + 15 = 255 \end{array}$$

Hex = FF

Direcciones en formato HexaDecimal

Con 2 bytes o **16** bits direccionamos 64 KBytes (2^{16})

Necesitamos 4 dígitos hexadecimales

Valor		16^3	16^2	16^1	16^0		
	-----+					-----+	
1FFF		1	F	F	F		$1*16^3 + 15*16^2 + 15*16^1 + 15 = 8191$
2000		2	0	0	0		$2*16^3 + 0*16^2 + 0*16^1 + 0 = 8192$
D1F0		D	1	F	0		$13*16^3 + 1*16^2 + 14*16^1 + 0 = 53744$
FFFF		F	F	F	F		$15*16^3 + 15*16^2 + 15*16^1 + 15 = 65535$

Sistema Octal

- En base a 8 al usar sólo 3 bits
- Se usa en Unix para describir permisos de archivos
- Cada bit es un permiso
- *man chmod*

Gracias!
Repasa **lentamente**
(con café, obvio)