

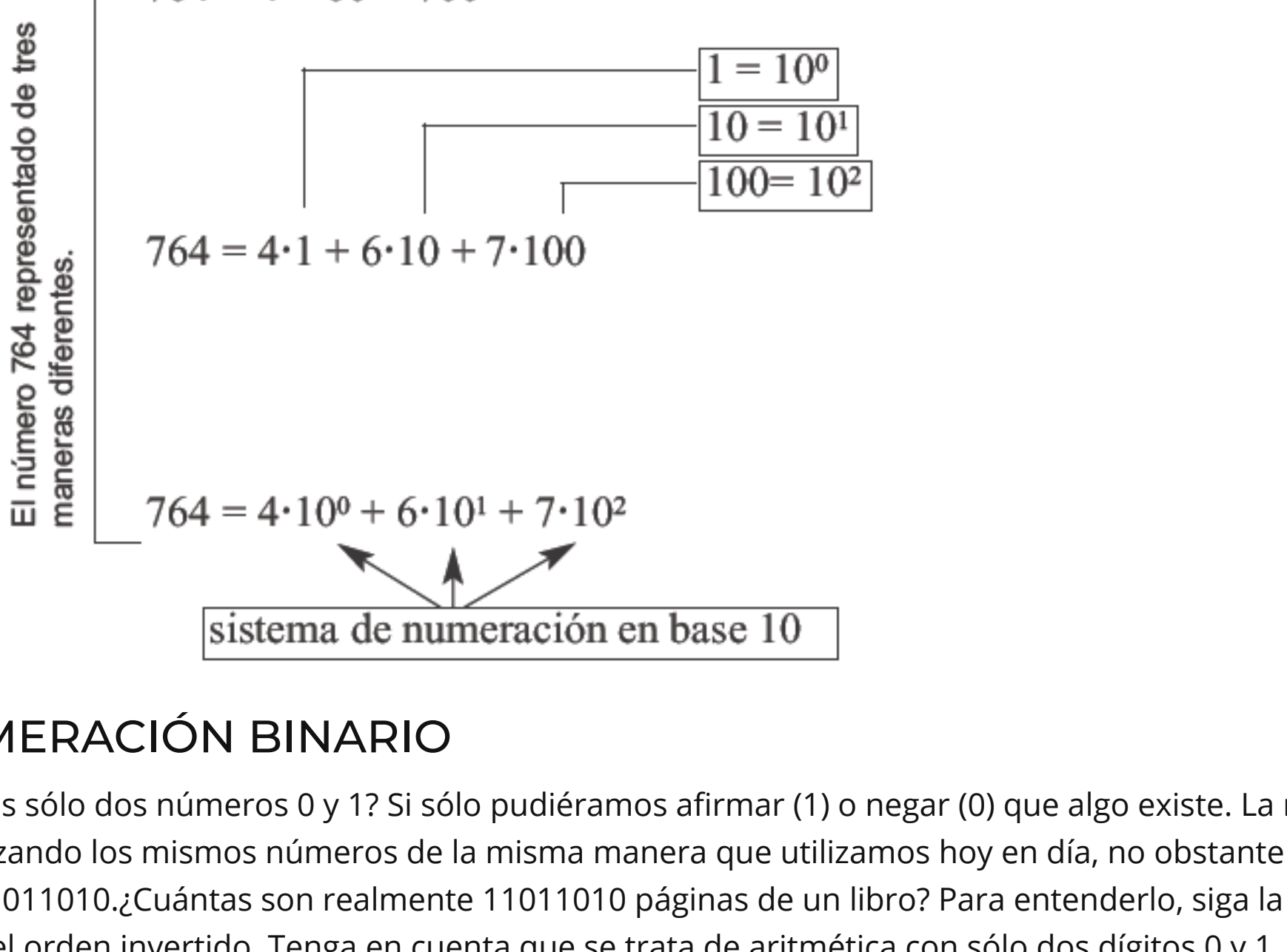
# MikroElektronika books

## Book: Microcontroladores PIC – Programación en C con ejemplos

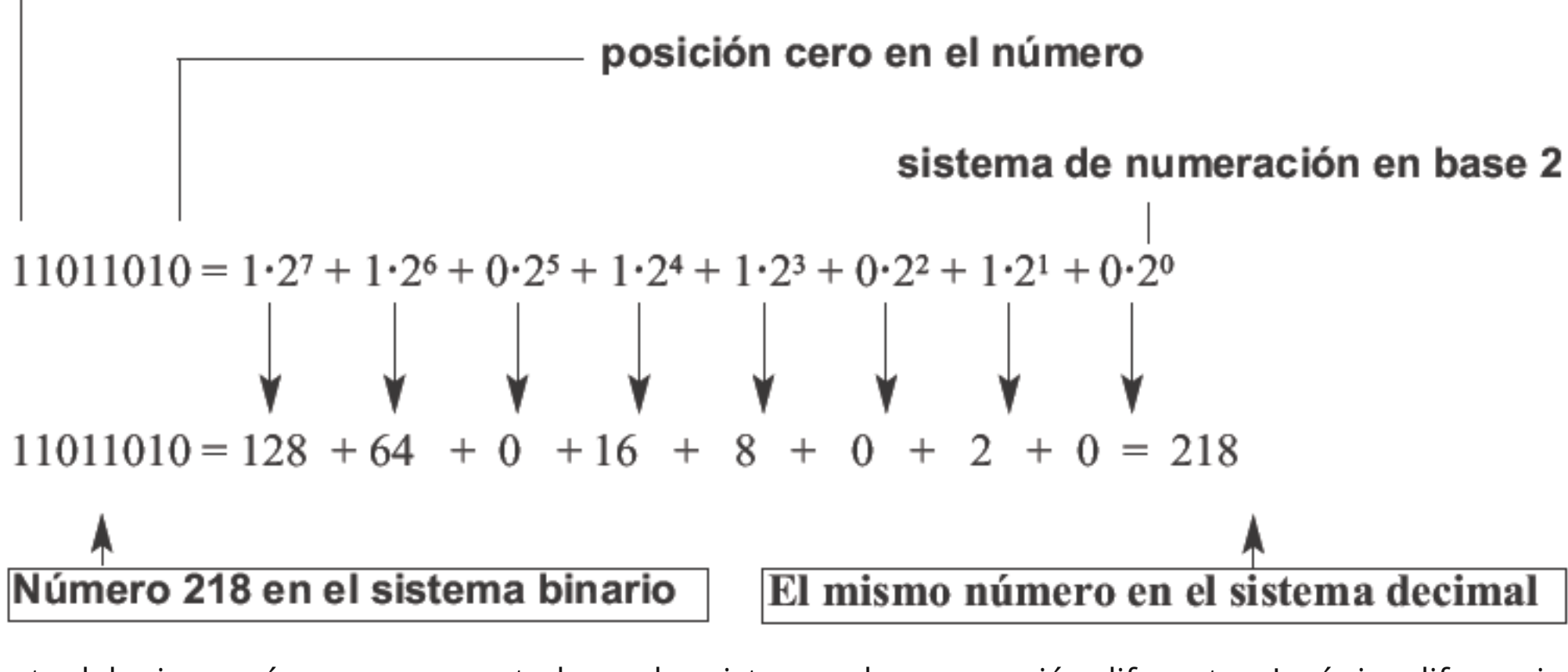
Table of Contents

### 1.2 Números, Números, Números...

¡La matemática es una gran ciencia! Todo es tan lógico y simple... El universo de los números se puede describir con sólo diez dígitos. No obstante, ¿realmente tiene que ser así? ¿Necesitamos exactamente esos 10 dígitos? Por supuesto que no, es sólo cuestión del hábito. Acuérdesse de las lecciones de la escuela. Por ejemplo, ¿qué significa el número 764? Cuatro unidades, seis decenas y siete centenas. ¡Muy simple! ¿Se podría expresar de una forma más desarrollada? Por supuesto que sí:  $4 + 60 + 700$ . ¿Aún más desarrollado? Sí:  $4 \cdot 1 + 6 \cdot 10 + 7 \cdot 100$ . ¿Podría este número parecer un poco más “científico”? La respuesta es sí otra vez:  $4 \cdot 10^0 + 6 \cdot 10^1 + 7 \cdot 10^2$ . ¿Qué significa esto realmente? ¿Por qué utilizamos exactamente estos números 100, 101 y 102 ? ¿Por qué es siempre el número 10? Es porque utilizamos 10 dígitos diferentes (0, 1, 2...8, 9). En otras palabras, es porque utilizamos el sistema de numeración en base 10, es decir el sistema de numeración decimal.



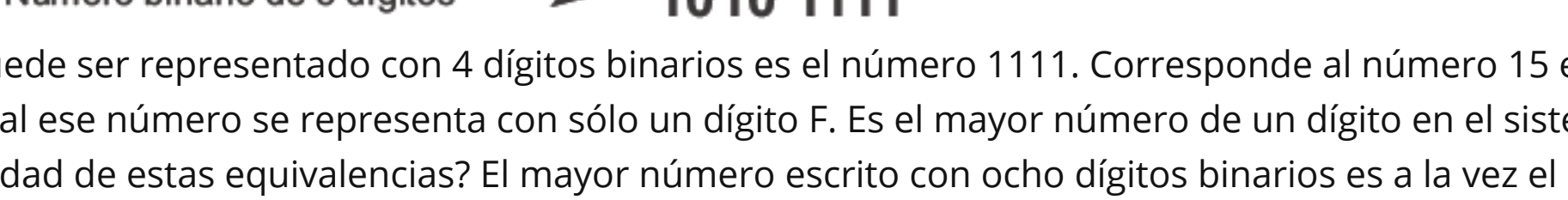
¿Qué pasaría si utilizáramos sólo dos números 0 y 1? Si sólo pudiéramos afirmar (1) o negar (0) que algo existe. La respuesta es “nada especial”, seguiríamos utilizando los mismos números de la misma manera que utilizamos hoy en día, no obstante ellos parecerían un poco diferentes. Por ejemplo: 11011010. ¿Cuántas son realmente 11011010 páginas de un libro? Para entenderlo, siga la misma lógica como en el ejemplo anterior, pero en el orden invertido. Tenga en cuenta que se trata de aritmética con sólo dos dígitos 0 y 1, es decir, del sistema de numeración en base 2 (sistema de numeración binario).



Evidentemente, se trata del mismo número representado en dos sistemas de numeración diferentes. La única diferencia entre estas dos representaciones yace en el número de dígitos necesarios para escribir un número. Un dígito (2) se utiliza para escribir el número 2 en el sistema decimal, mientras que dos dígitos (1 y 0) se utilizan para escribir aquel número en el sistema binario. ¿Ahora está de acuerdo que hay 10 grupos de gente? ¡Bienvenido al mundo de la aritmética binaria! ¿Tiene alguna idea de dónde se utiliza? Excepto en las condiciones de laboratorio estrictamente controladas, los circuitos electrónicos más complicados no pueden especificar con exactitud la diferencia entre dos magnitudes (dos valores de voltaje, por ejemplo), si son demasiado pequeños (más pequeños que unos pocos voltios). La razón son los ruidos eléctricos y fenómenos que se presentan dentro de lo que llamamos “entorno de trabajo real” (algunos ejemplos de estos fenómenos son los cambios imprevisibles de la tensión de alimentación, cambios de temperatura, tolerancia a los valores de los componentes etc...). Imagínese una computadora que opera sobre números decimales al tratarlos de la siguiente manera: 0=0V, 1=5V, 2=10V, 3=15V, 4=20V... 9=45V! ¿Alguien dijo baterías? Una solución mucho más fácil es una lógica binaria donde 0 indica la ausencia de voltaje, mientras que 1 indica la presencia de voltaje. Simplemente, es fácil de escribir 0 o 1 en vez de “no hay voltaje” o “hay voltaje”. Mediante el cero lógico (0) y uno lógico (1) la electrónica se enfrenta perfectamente y realiza con facilidad todas las operaciones aritméticas. Evidentemente, se trata de electrónica que en realidad aplica aritmética en la que todos los números son representados con sólo dos dígitos y donde sólo es importante saber si hay voltaje o no. Por supuesto, estamos hablando de electrónica digital.

### SISTEMA DE NUMERACIÓN HEXADECIMAL

En el principio del desarrollo de las computadoras era evidente que a la gente le costaba mucho trabajar con números binarios. Por eso, se estableció un nuevo sistema de numeración, que utilizaba 16 símbolos diferentes. Es llamado el sistema de numeración hexadecimal. Este sistema está compuesto de 10 dígitos a los que estamos acostumbrados (0, 1, 2, 3... 9) y de seis letras del alfabeto A, B, C, D, E y F. ¿Cuál es el propósito de esta combinación aparentemente extraña? Basta con mirar cómo todo en la historia de los números binarios encaja perfectamente para lograr una mejor comprensión del tema.



El mayor número que puede ser representado con 4 dígitos binarios es el número 1111. Corresponde al número 15 en el sistema decimal. En el sistema hexadecimal ese número se representa con sólo un dígito F. Es el mayor número de un dígito en el sistema hexadecimal. ¿Se da cuenta de la gran utilidad de estas equivalencias? El mayor número escrito con ocho dígitos binarios es a la vez el mayor número de dos dígitos en el sistema hexadecimal. Tenga en cuenta que una computadora utiliza números binarios de 8 dígitos. ¿Acaso se trata de una casualidad?

### CÓDIGO BCD

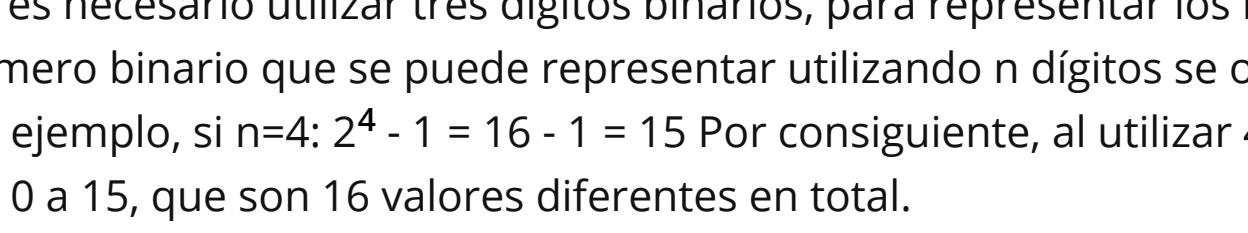
El código BCD (**B**inary-**C**oded **D**ecimal - Código binario decimal) es un código binario utilizado para representar a los números decimales. Se utiliza para que los circuitos electrónicos puedan comunicarse con los periféricos utilizando el sistema de numeración decimal o bien utilizando el sistema binario dentro de “su propio mundo”. Consiste en números binarios de 4 dígitos que representan los primeros diez dígitos (0, 1, 2, 3...8, 9). Aunque cuatro dígitos pueden hacer 16 combinaciones posibles en total, el código BCD normalmente utiliza a las primeras diez.

### CONVERSIÓN DE SISTEMAS DE NÚMERACIÓN

El sistema de numeración binario es el que utilizan los microcontroladores, el sistema decimal es el que nos resulta más comprensible, mientras que el sistema hexadecimal presenta un balance entre los dos. Por eso, es muy importante aprender cómo convertir los números de un sistema de numeración a otro, por ejemplo, cómo convertir una serie de ceros y unos a una forma de representación comprensible para nosotros.

### CONVERSIÓN DE NÚMEROS BINARIOS A DECIMALES

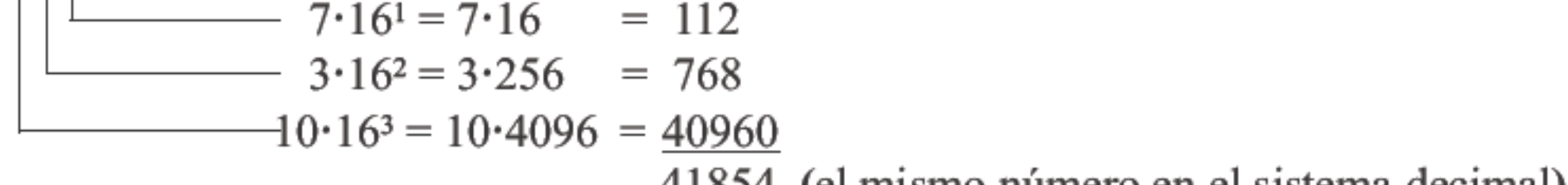
Los dígitos en un número binario tienen ponderaciones diferentes lo que depende de sus posiciones dentro del número que están representando. Además, cada dígito puede ser 1 o 0, y su ponderación se puede determinar con facilidad al contar su posición empezando por la derecha. Para hacer una conversión de un número binario a decimal es necesario multiplicar las ponderaciones con los dígitos correspondientes (0 o 1) y sumar todos los resultados. La magia de la conversión de un número binario a decimal funciona de maravilla... ¿Tiene duda? Veamos el siguiente ejemplo:



Cabe destacar que es necesario utilizar sólo dos dígitos binarios para representar a todos los números decimales de 0 a 3. Por consiguiente, para representar los números de 0 a 7 es necesario utilizar tres dígitos binarios, para representar los números de 0 a 15 - cuatro dígitos etc. Dicho de manera sencilla, el mayor número binario que se puede representar utilizando n dígitos se obtiene al elevar la base 2 a la potencia n. Luego, al resultado se le resta 1. Por ejemplo, si n=4:  $2^4 - 1 = 16 - 1 = 15$  Por consiguiente, al utilizar 4 dígitos binarios, es posible representar los números decimales de 0 a 15, que son 16 valores diferentes en total.

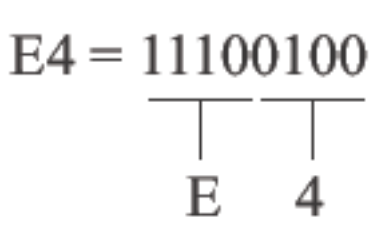
### CONVERSIÓN DE NÚMEROS HEXADECIMALES A DECIMALES

Para realizar una conversión de un número hexadecimal a decimal, cada dígito hexadecimal debe ser multiplicado con el número 16 elevado al valor de su posición. Por ejemplo:



### CONVERSIÓN DE NÚMEROS HEXADECIMALES A BINARIOS

No es necesario realizar ningún cálculo para convertir un número hexadecimal a binario. Los dígitos hexadecimales se reemplazan simplemente por los cuatro dígitos binarios apropiados. Ya que el dígito hexadecimal máximo es equivalente al número decimal 15, es necesario utilizar cuatro dígitos binarios para representar un dígito hexadecimal. Por ejemplo:



### MARCAR LOS NÚMEROS

El sistema de numeración hexadecimal, junto con los sistemas binario y decimal, se consideran los más importantes para nosotros. Es fácil realizar una conversión de cualquier número hexadecimal a binario, además es fácil de recordarlo. Sin obstante, estas conversiones pueden provocar una confusión. Por ejemplo, ¿qué significa en realidad la sentencia: “Es necesario contar 110 productos en una cadena de montaje”? Dependiendo del sistema en cuestión (binario, decimal o hexadecimal), el resultado podría ser 6, 110 o 272 productos, respectivamente. Por consiguiente, para evitar equivocaciones, diferentes prefijos y sufijos se añaden directamente a los números. El prefijo \$ o 0x así como el sufijo h marca los números en el sistema hexadecimal. Por ejemplo, el número hexadecimal 10AF se puede escribir así: \$10AF, 0x10AF o 10AFh. De manera similar, los números binarios normalmente obtienen el sufijo % o 0B. Si un número no tiene ni sufijo ni prefijo se considera decimal. Desafortunadamente, esta forma de marcar los números no es estandarizada, por consiguiente depende de la aplicación concreta. La siguiente es tabla comparativa que contiene los valores de números 0-255 representados en tres sistemas de numeración diferentes. Esto es probablemente la manera más fácil de entender lógica común aplicada a todos los sistemas de numeración.

DEC.	BINARIO				HEX.
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	0	2
3	0	0	0	0	3
4	0	0	0	0	4
5	0	0	0	0	5
6	0	0	0	0	6
7	0	0	0	0	7
8	0	0	0	0	8
9	0	0	0	0	9
10	0	0	0	0	A
11	0	0	0	0	B
12	0	0	0	0	C
13	0	0	0	0	D
14	0	0	0	0	E
15	0	0	0	0	F
16	0	0	0	0	10
17	0	0	0	0	11

253	1	1	1	1	1	0	1	FD
254	1	1	1	1	1	1	0	FE
255	1	1	1	1	1	1	1	FF

### NÚMEROS NEGATIVOS

Como ya hemos visto, para escribir un número negativo en matemáticas, basta con añadirle el prefijo “-” (signo menos). Sin embargo, en la programación, al acabarse el proceso de la compilación, se quedan sólo los números binarios, volviéndose las cosas más complicadas. Se utilizan sólo dos dígitos - 0 y 1, mientras que todos los demás números, símbolos y signos se forman por medio de las combinaciones de estos dos dígitos. En el caso de los números negativos, la solución es la siguiente: En los números negativos, el bit más significativo (el bit del extremo izquierdo) representa el signo del número (donde 0 será positivo y 1 negativo). En el caso de un número de 8 bits, para representar un valor numérico sólo quedan 7 bits. De acuerdo a este tipo de codificación el número +127 es el mayor número positivo con signo que se puede representar con 8 bits. Asimismo, hay cero (0) positivo y negativo (refiérase a la tabla de la derecha). La siguiente pregunta sería: ¿Cómo es posible saber de qué número se trata? Por ejemplo, si ponemos el número 10000001, ¿es -1 o 129? No se preocupe, de eso se encarga el compilador. Ésta es la razón por la que se declaran variables al escribir el programa. Bueno, de eso vamos a hablar a continuación.

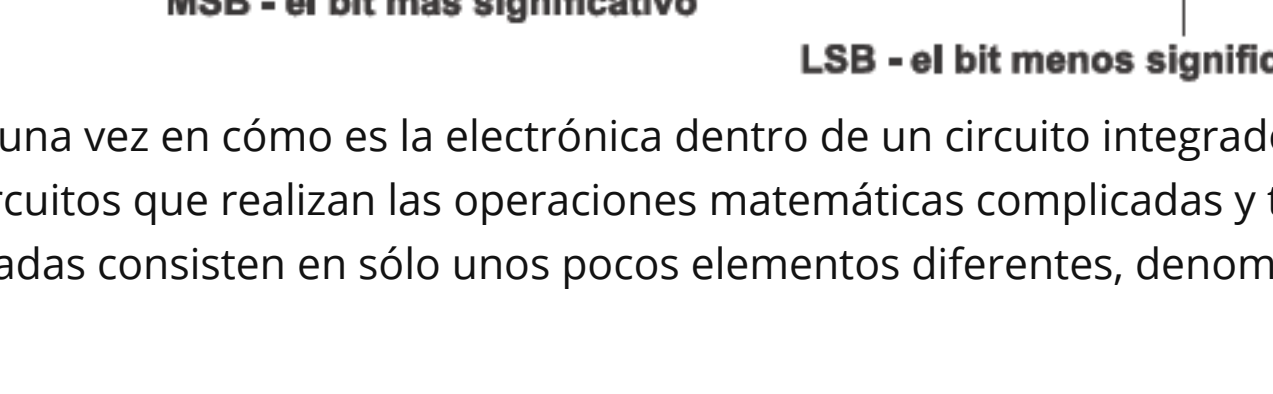
Binario	con signo	sin signo
00000000	+0	0
00000001	1	1
...	...	...
01111111	127	127
10000000	-0	128
10000001	-1	129
...	...	...
11111111	-127	255

### BIT

La teoría dice que un bit es la unidad básica de información...Vamos a olvidarlo por un momento y demostrar qué es eso en la práctica. La respuesta es - nada especial- un bit es un sólo dígito binario. Similar a un sistema de numeración decimal en el que los dígitos de un número no tienen la misma ponderación (por ejemplo, los dígitos en el número 444 son los mismos pero tienen los valores diferentes), el “significado” de un bit depende de la posición que tiene en número binario. En este caso no tiene sentido hablar de unidades, centenas etc. En los números binarios, sus dígitos se denominan el bit cero (el primer bit a la derecha), el primer bit (el segundo bit a la derecha) etc. Además, ya que el sistema binario utiliza solamente dos dígitos (0 y 1), el valor de un bit puede ser 0 o 1. No se confunda si se encuentra con un bit que tiene el valor 4, 16 o 64. Son los valores representados en el sistema decimal. Simplemente, nos hemos acostumbrado tanto a utilizar los números decimales que estas expresiones llegaron a ser comunes. Sería correcto decir por ejemplo, “el valor del sexto bit en cualquier número binario equivale al número decimal 64”. Pero todos somos humanos y los viejos hábitos mueren difícilmente. Además, ¿cómo le suena “número uno-uno-cero-uno-cero...”?

### BYTE

Un byte consiste en 8 bits colocados uno junto al otro. Si un bit es un dígito, es lógico que los bytes representen los números. Todas las operaciones matemáticas se pueden realizar por medio de ellos, como por medio de los números decimales comunes. Similar a los dígitos de cualquier número, los dígitos de un byte no tienen el mismo significado. El bit del extremo izquierdo tiene la mayor ponderación, por eso es denominado el bit más significativo (MSB). El bit del extremo derecho tiene la menor ponderación, por eso es denominado el bit menos significativo (LSB). Puesto que los 8 dígitos de un byte se pueden combinar de 256 maneras diferentes, el mayor número decimal que se puede representar con un byte es 255 (una combinación representa un cero). Un nibble o un cuarteto representa una mitad de byte. Dependiendo de la mitad del número en cuestión (izquierda o derecha), se les denomina nibbles “altos” o “bajos”, respectivamente.



Usted seguramente ha pensado alguna vez en cómo es la electrónica dentro de un circuito integrado digital, un microcontrolador o un microprocesador. ¿Cómo son los circuitos que realizan las operaciones matemáticas complicadas y toman decisiones? ¿Sabía que sus esquemas, aparentemente complicados consisten en sólo unos pocos elementos diferentes, denominados circuitos lógicos o computas lógicas?