

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/291335556>

Sistemas numéricos

Article · January 2016

CITATIONS

0

READS

1,809

1 author:



Hector Antonio Villa-Martinez
Universidad de Sonora (Unison)

19 PUBLICATIONS 12 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Development of statistical tools for mobile devices. Fourth Stage. [View project](#)



Statistics to Go [View project](#)

Sistemas de numeración y aritmética binaria

Héctor Antonio Villa Martínez
Programa de Ciencias de la Computación
Universidad de Sonora

Este reporte consta de tres secciones. Primero, la Sección 1 presenta los sistemas de numeración más utilizados en el diseño de sistemas digitales. En seguida, la Sección 2 muestra los principales métodos para convertir números de un sistema de numeración a otro. Por último, la Sección 3 explica como se realizan las cuatro operaciones aritméticas básicas (suma, resta, multiplicación y división) en binario.

1. Sistemas de numeración

El sistema de numeración de uso común, conocido como sistema decimal o base 10, utiliza diez dígitos (0, 1, ..., 9) para representar números. El sistema decimal es posicional porque el valor de un número se obtiene multiplicando cada dígito por la potencia de 10 que le corresponde a la posición del dígito dentro del número. La potencia se encuentra asignando la posición cero al dígito que está a la izquierda del punto decimal y luego numerando el resto de los dígitos, ascendentemente hacia la izquierda y descendentemente hacia la derecha. Así el valor del número decimal 727.53 es:

$$\begin{aligned} 727.53 &= 7 \times 10^2 + 2 \times 10^1 + 7 \times 10^0 + 5 \times 10^{-1} + 3 \times 10^{-2} \\ &= 700 + 20 + 7 + 0.5 + 0.03 \\ &= 727.53 \end{aligned}$$

De la misma forma se pueden representar números en otros sistemas de numeración. En el diseño de sistemas digitales los sistemas de numeración

más utilizados, además del sistema decimal, son el sistema binario o base 2, el sistema octal o base 8 y el sistema hexadecimal o base 16. Cada uno de estos sistemas se presentará a continuación.

El sistema binario emplea solo dos dígitos, 0 y 1, para representar números. Como la base del sistema es 2, cada dígito binario se multiplica por la potencia de 2 que le corresponde:

$$\begin{aligned} 11001.011_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ &= 16 + 8 + 1 + 0.25 + 0.125 \\ &= 25.375_{10} \end{aligned}$$

Nótese como para evitar confusiones entre números en bases distintas se acostumbra marcar la base como un subíndice del número.

El sistema octal, por otro lado, utiliza ocho dígitos, 0, 1, ..., 7, para representar números y cada dígito octal se multiplica por la potencia de 8 que le toca:

$$\begin{aligned} 142.4_8 &= 1 \times 8^2 + 4 \times 8^1 + 2 \times 8^0 + 4 \times 8^{-1} \\ &= 64 + 32 + 2 + 0.5 \\ &= 98.5_{10} \end{aligned}$$

Por último, como el sistema hexadecimal necesita de dieciséis dígitos, se utilizan los diez dígitos decimales para representar dígitos entre 0 y 9 y las primeras seis letras del alfabeto para los dígitos del 10 al 15. De esta forma, A representa 10_{10} , B representa 11_{10} , C representa 12_{10} , D representa 13_{10} , E representa 14_{10} y F representa 15_{10} . Parecido a los casos anteriores, cada dígito hexadecimal se multiplica por la potencia correspondiente de la base, en este caso 16:

$$\begin{aligned} D6C.5AF_{16} &= 13 \times 16^2 + 6 \times 16^1 + 12 \times 16^0 + 5 \times 16^{-1} + 10 \times 16^{-2} + 15 \times 16^{-3} \\ &= 3328 + 96 + 12 + 0.3125 + 0.0390625 + 0.00366210938 \\ &= 3436.35522460938_{10} \end{aligned}$$

Como un ejemplo adicional, el Cuadro 1 lista los primeros 32 números enteros en base 10, 2, 8 y 16.

Cuadro 1: Los primeros 32 números enteros en base 10, 2, 8 y 16.

base 10	base 2	base 8	base 16	base 10	base 2	base 8	base 16
0	0	0	0	16	10000	20	10
1	1	1	1	17	10001	21	11
2	10	2	2	18	10010	22	12
3	11	3	3	19	10011	23	13
4	100	4	4	20	10100	24	14
5	101	5	5	21	10101	25	15
6	110	6	6	22	10110	26	16
7	111	7	7	23	10111	27	17
8	1000	10	8	24	11000	30	18
9	1001	11	9	25	11001	31	19
10	1010	12	A	26	11010	32	1A
11	1011	13	B	27	11011	33	1B
12	1100	14	C	28	11100	34	1C
13	1101	15	D	29	11101	35	1D
14	1110	16	E	30	11110	36	1E
15	1111	17	F	31	11111	37	1F

2. Conversión entre sistemas de numeración

Esta sección presenta los principales métodos para convertir números de una base a otra. Como el sistema decimal es el sistema de numeración de uso común, primero se presentan los métodos para convertir de base 10 a otra base (Subsección 2.1) y de otra base a base 10 (Subsección 2.2). Después se explica la conversión entre dos bases arbitrarias (Subsección 2.3). Por último, se muestra una forma rápida para convertir entre base 2 y bases 8 y 16 (Subsección 2.4).

2.1. De base 10 a otra base

La conversión de un número en base 10 a otra base, r , se realiza mediante dos procedimientos separados, uno para obtener la parte entera del número en base r y otro para obtener la parte fraccionaria.

La parte entera del número en base r se obtiene dividiendo la parte entera del número decimal sucesivamente entre r , guardando en cada paso el residuo, y volviendo a dividir el cociente entre r . Esta división continúa hasta que el cociente sea cero. En este momento, la parte entera del número en base r se forma considerando los residuos en el orden inverso en que se obtuvieron. Es decir, el dígito más significativo de la parte entera es el último residuo que se generó y el dígito menos significativo es el primer residuo.

La parte fraccionaria del número en base r se obtiene multiplicando la parte fraccionaria del número decimal sucesivamente por r , guardando en cada paso la parte entera del resultado, y volviendo a multiplicar la parte fraccionaria del resultado por r . Esta multiplicación continúa hasta que la parte fraccionaria del resultado sea cero o se obtenga la precisión deseada. En este momento, la parte fraccionaria del número en base r se forma considerando los enteros en el mismo orden en que se obtuvieron.

Como ejemplo considere convertir 26.375 de base 10 a base 2.

La parte entera se convierte así:

$$26 \div 2 \rightarrow \text{cociente} = 13, \text{ residuo} = 0 \text{ (dígito menos significativo)}$$

$$13 \div 2 \rightarrow \text{cociente} = 6, \text{ residuo} = 1$$

$$6 \div 2 \rightarrow \text{cociente} = 3, \text{ residuo} = 0$$

$$3 \div 2 \rightarrow \text{cociente} = 1, \text{ residuo} = 1$$

$$1 \div 2 \rightarrow \text{cociente} = 0, \text{ residuo} = 1 \text{ (dígito mas significativo)}$$

Y la parte fraccionaria así:

$$0.375 \times 2 = 0.75 \rightarrow \text{entero} = 0 \text{ fracción } 0.75 \text{ (primer dígito fraccionario)}$$

$$0.75 \times 2 = 1.5 \rightarrow \text{entero} = 1 \text{ fracción } 0.5$$

$$0.5 \times 2 = 1.0 \rightarrow \text{entero} = 1 \text{ fracción } 0.0 \text{ (último dígito fraccionario)}$$

$$\text{En conclusión, } 26.375_{10} = 11010.011_2.$$

En ocasiones nos encontramos con fracciones decimales que no se pueden representar en otras bases de manera exacta. Para estos casos es necesario definir una precisión o detener el proceso si la fracción en base r es cíclica. Un ejemplo de esta última situación se genera al convertir 34.2 de base 10 a base 8.

Primero se convierte la parte entera:

$$34 \div 8 \rightarrow \text{cociente} = 4, \text{ residuo} = 2$$

$$4 \div 8 \rightarrow \text{cociente} = 0, \text{ residuo} = 4$$

Y luego la parte fraccionaria:

$$0.2 \times 8 = 1.6 \rightarrow \text{entero} = 1, \text{ fracción } 0.6$$

$$0.6 \times 8 = 4.8 \rightarrow \text{entero} = 4, \text{ fracción } 0.8$$

$$0.8 \times 8 = 6.4 \rightarrow \text{entero} = 6, \text{ fracción } 0.4$$

$$0.4 \times 8 = 3.2 \rightarrow \text{entero} = 3, \text{ fracción } 0.2$$

$$0.2 \times 8 = 1.6 \rightarrow \text{entero} = 1, \text{ fracción } 0.6$$

$$0.6 \times 8 = 4.8 \rightarrow \text{entero} = 4, \text{ fracción } 0.8$$

En este momento nos podemos dar cuenta que la fracción octal es cíclica. Por lo tanto, 34.2_{10} se puede representar como 42.1463_8 con 4 dígitos de precisión o más precisamente como $42.\overline{1463}_8$.

2.2. De otra base a base 10

La conversión de números de otra base a base 10 fue presentada informalmente en los ejemplos de la Sección 1. Formalmente, para convertir un número en base r , formado por n dígitos enteros y m fraccionarios $a_{n-1}a_{n-2}\dots a_1a_0.b_0b_1\dots b_{m-2}b_{m-1}$, a base 10, se aplica la siguiente ecuación:

$$N = a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \dots + a_1 \times r^1 + a_0 \times r^0 + b_0 \times r^{-1} + b_1 \times r^{-2} + \dots + b_{m-2} \times r^{-(m-1)} + b_{m-1} \times r^{-m}$$

Donde N es el número en base 10.

Como ejemplo, considere la conversión del número hexadecimal $25A.CC$ a base 10.

$$\begin{aligned}
N &= 2 \times 16^2 + 5 \times 16^1 + 10 \times 16^0 + 12 \times 16^{-1} + 12 \times 16^{-2} \\
&= 512 + 80 + 10 + 0.75 + 0.046875 \\
&= 602.796875_{10}
\end{aligned}$$

2.3. Conversión de base r a base s

El método más sencillo para convertir un número de base r a base s es utilizar la base 10 como base intermedia. Esto es, el número original en base r se convierte a base 10 como se explicó en la Sección 2.2 y luego el número decimal se convierte a base s siguiendo el procedimiento visto en la Sección 2.1.

2.4. Conversión rápida entre base 2 y bases 8 y 16

Para el caso particular de convertir números de base 2 a base 8 o 16 y viceversa, existen métodos de conversión directos para evitar utilizar la base 10 como base intermedia. A continuación se describen primero los métodos de conversión entre base 2 y base 8, y en seguida los métodos de conversión entre base 2 y base 16.

Para convertir un número de base 2 a base 8 se sigue el siguiente procedimiento. Primero, si el número de dígitos en la parte entera o en la parte fraccionaria del número binario no es múltiplo de tres, se agregan ceros suficientes a la izquierda de la parte entera o a la derecha de la parte fraccionaria cuidando de no alterar el valor del número. Segundo, los dígitos del número binario se agrupan en bloques de tres dígitos. Finalmente, el número octal se obtiene cambiando cada bloque de tres dígitos binarios por un dígito octal de acuerdo al Cuadro 2. Como ejemplo, se muestra la conversión del número binario 1011.01 a octal.

$$\begin{aligned}
1011.01_2 &= 001011.010 \quad \text{Después de agregar ceros a la izquierda y derecha.} \\
&= \boxed{001}\boxed{011}\boxed{.010} \quad \text{Después de agrupar los dígitos binarios.} \\
&= 13.2_8 \quad \text{Después de cambiar los dígitos usando el Cuadro 2.}
\end{aligned}$$

La conversión de un número de base 8 a base 2 sigue el procedimiento inverso. Cada dígito octal se cambia por tres dígitos binarios de acuerdo al

Cuadro 2: Conversión entre base 2 y base 8.

Binario	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Cuadro 2. Por ejemplo, la conversión del número octal 242.52 a binario es como sigue.

$$242.52_8 = \boxed{010} \boxed{100} \boxed{010} . \boxed{101} \boxed{010} \quad \text{Después de cambiar los dígitos usando el Cuadro 2.}$$

$$= 10100010.10101_2 \quad \text{Después de eliminar los ceros superfluos.}$$

La conversión de base 2 a base 16 es similar a la conversión de base 2 a base 8 descrita arriba. Las únicas diferencias son que los bloques tienen cuatro bits binarios y que cada bloque se cambia por un dígito hexadecimal usando el Cuadro 3. En el ejemplo que sigue, se muestra la conversión del número binario 101110.01 a hexadecimal.

$$101110.01_2 = 00101110.0100 \quad \text{Después de agregar los ceros necesarios.}$$

$$= \boxed{0010} \boxed{1110} . \boxed{0100} \quad \text{Después de agrupar los dígitos binarios.}$$

$$= 2E.4_{16} \quad \text{Después de cambiar los dígitos usando el Cuadro 3.}$$

Finalmente, para convertir un número de base 16 a base 2 cada dígito hexadecimal se sustituye por cuatro dígitos binarios de acuerdo al Cuadro 3. Así, la conversión del número hexadecimal 72.E5C a binario es como se muestra a continuación.

$$72.E5C_{16} = \boxed{0111} \boxed{0010} . \boxed{1110} \boxed{0101} \boxed{1100} \quad \text{Después de cambiar los dígitos usando el Cuadro 3.}$$

$$= 1110010.1110010111_2 \quad \text{Después de eliminar los ceros superfluos.}$$

Cuadro 3: Conversión entre base 2 y base 16.

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

Cuadro 4: Suma de dos dígitos binarios.

x	y	$x + y$	Acarreo
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

3. Aritmética binaria

En los sistemas digitales las operaciones aritméticas se realizan en binario porque es más fácil diseñar circuitos digitales para realizar aritmética binaria que aritmética decimal. Estas operaciones aritméticas incluyen la suma (Sección 3.1), resta (Sección 3.2), multiplicación (Sección 3.3) y división (Sección 3.4). Las operaciones se hacen del mismo modo que en el sistema decimal, pero debido a la sencillez del sistema binario, pueden hacerse algunas simplificaciones que facilitan la realización de las operaciones.

3.1. Suma

La suma de dos dígitos binarios, x y y (ver Cuadro 4), involucra el concepto de *acarreo* (carry en inglés). El acarreo se presenta cuando se suma $1 + 1$ cuyo resultado, 2 en decimal, se escribe con dos dígitos en binario (10). En este caso la suma es 0 y se pasa (o se acarrea) 1 a la siguiente columna.

Como ejemplo considere la suma binaria de 11001 y 1101.

Cuadro 5: Resta de dos dígitos binarios.

x	y	$x - y$	Préstamo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\begin{array}{r}
 \textcolor{red}{110010} \leftarrow \text{acarreo} \\
 11001 \\
 + \quad 1101 \\
 \hline
 100110
 \end{array}$$

Los acarreos, presentados en rojo, muestran si en la columna anterior hubo acarreo (valor de 1) o no (valor de 0).

3.2. Resta

La resta de números binarios es similar a la resta decimal que conocemos. Cuando al restar dos números binarios nos encontramos con la operación $0 - 1$ se genera un *préstamo* de la siguiente columna (ver Cuadro 5). Al continuar con la siguiente columna hay que tomar en cuenta el préstamo, y si es necesario volver hacer un préstamo de la columna que sigue. Si el préstamo se propaga hasta la última columna entonces el resultado de la resta es un número negativo. La representación en binario de números negativos se presenta en [1]. Para una explicación más extensa del concepto de préstamo, consultar Roth [2, p. 11].

Como ejemplo considere la resta binaria de 111001 y 1011.

$$\begin{array}{r}
 \textcolor{red}{011100} \leftarrow \text{préstamos} \\
 111001 \\
 + \quad 1011 \\
 \hline
 101110
 \end{array}$$

Los préstamos, presentados en rojo, muestran si en la columna anterior hubo un préstamo (valor de 1) o no (valor de 0).

Cuadro 6: Multiplicación de dos dígitos binarios.

x	y	$x \times y$
0	0	0
0	1	0
1	0	0
1	1	1

3.3. Multiplicación

En la multiplicación de números binarios sólo es necesario recordar la tabla de multiplicación que se exhibe en el Cuadro 6.

Como ejemplo se presenta la multiplicación de 1111 por 1101. Nótese como para evitar sumas de varios dígitos binarios se hacen sumas parciales, denotadas en rojo, después de cada dos números generados. El producto final se muestra en azul.

$$\begin{array}{rcl}
 & 1111 & \\
 \times & 1101 & \\
 \hline
 & 1111 & \text{Primer producto parcial} \\
 & 0000 & \text{Segundo producto parcial} \\
 \hline
 & 01111 & \text{Primera suma parcial} \\
 & 1111 & \text{Tercer producto parcial} \\
 \hline
 & 1001011 & \text{Segunda suma parcial} \\
 & 1111 & \text{Cuarto producto parcial} \\
 \hline
 & 11000011 & \text{Producto final}
 \end{array}$$

3.4. División

La división binaria es más sencilla que la división decimal porque solo hay dos posibles valores para el cociente: 0 si el divisor es menor que el dividendo, y 1 en caso contrario. Como ejemplo a continuación se muestra la división de 10010001 entre 1011.

$$\begin{array}{r}
 1101 \\
 1011 \overline{) 10010001} \\
 \underline{1011} \\
 1110 \\
 \underline{1011} \\
 1101 \\
 \underline{1011} \\
 10
 \end{array}$$

Al finalizar la división, se obtiene que el cociente es 1101 y el resto o residuo es 10.

Referencias

- [1] Héctor Antonio Villa Martínez. Representación de números en binario. Technical report, Universidad de Sonora, Hermosillo, Sonora, 2008.
- [2] Charles H. Roth Jr. *Fundamentos de diseño lógico*. Thomson, México, DF, quinta edición, 2005.