

Organización del computador

Sistemas de representación

Organización

- ➤ Recordemos que la organización de un computador refiere al diseño específico de sus componentes, y como estas se coordinan para llevar a cabo una tarea
- ➤ Un factor decisivo en como se estructuran las componentes (y cómo estas están implementadas a través de circuitos) es cómo está representada la información
- ➤ Los sistemas modernos (a partir de The First Draft) utilizan el sistema binario (de donde proviene la palabra **bit** - **binary digit**)

Sistemas de numeración

Magnitudes vs. Números

- ➤ Lo que conocemos como **números** son formas de expresar **magnitudes** que son preexistentes (i.e. los dedos de las manos de un ser humano siempre fueron diez, aun antes de que existiera el número 10, X, etc.)
- ➤ Usar el **sistema decimal** condiciona la forma en la que comprendemos las magnitudes pues es la forma en la que las racionalizamos, por ejemplo, en la escritura.
- ➤ Las culturas mesopotámicas usaban el sistema duodecimal (base 12), introducido por los astrónomos (12 meses solares, 12 horas de día, 12 horas de noche, etc.) acabó por regir el comercio a través de la docena, la gruesa, etc.

Sistemas de numeración

Sistemas de numeración

- > Un **sistema de numeración** es un conjunto de símbolos y un conjunto de reglas que permiten combinarlos de forma que es posible representar información, en particular, **magnitudes**
- > Entre los sistemas de numeración, una clase de suma importancia son los **sistemas posicionales**

Representación

$$(a_n \dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots a_{-m})_b = a_n \cdot b^n + \dots + a_2 \cdot b^2 + a_1 \cdot b + a_0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + \dots + a_{-m} \cdot b^{-m}$$

- > b es un entero mayor que 1 y para todo $-m \leq i \leq n$, vale que $0 \leq a_i < b$
- > el punto que aparece entre a_0 y a_{-1} se denomina **punto fraccionario** separando la parte entera de un número de su arte fraccionaria

Sistemas de numeración

- ➤ **Sistema decimal:** $b = 10$ y los dígitos pertenecen al conjunto $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$; es el sistema de numeración de uso cotidiano
- ➤ **Sistema binario:** $b = 2$ y los dígitos pertenecen al conjunto $\{0, 1\}$; es el sistema de numeración sobre el que basa la construcción de componentes de electrónica digital
- ➤ **Sistema octal:** $b = 8$ y los dígitos pertenecen al conjunto $\{0, 1, 2, 3, 4, 5, 6, 7\}$
- ➤ **Sistema hexadecimal:** $b = 16$ y los dígitos pertenecen al conjunto $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$; muy utilizado en la representación compacta tanto del contenido de la memoria de una computadora, como de las direcciones de sus palabras, pues 1 dígito hexadecimal = 2 bytes.

Sistemas de numeración

Cambio de base

- > Una representación de una magnitud en un sistema de numeración puede ser modificada a partir de cambiar la base, preservando la magnitud
- > Si observamos que para nosotros las magnitudes son racionalizadas en base 10:

De representación a magnitud

$$(a_n \dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots a_{-m})_{10} = a_n \cdot 10^n + \dots + a_2 \cdot 10^2 + a_1 \cdot 10 + a_0 + a_{-1} \cdot 10^{-1} + a_{-2} \cdot 10^{-2} + \dots + a_{-m} \cdot 10^{-m}$$

||

$$(b_p \dots b_2 b_1 b_0 . b_{-1} b_{-2} \dots b_{-q})_c = b_p \cdot c^p + \dots + b_2 \cdot c^2 + b_1 \cdot c + b_0 + b_{-1} \cdot c^{-1} + b_{-2} \cdot c^{-2} + \dots + b_{-q} \cdot c^{-q}$$

Sistemas de numeración

Cambio de base

- > Una representación de una magnitud en un sistema de numeración puede ser modificada a partir de cambiar la base, preservando la magnitud
- > Si observamos que para nosotros las magnitudes son racionalizadas en base 10:

De representación a magnitud

$$(307)_{10} = 3 \cdot 10^2 + 0 \cdot 10 + 7$$

||

$$(102101)_3 = 1 \cdot 3^5 + 0 \cdot 3^4 + 2 \cdot 3^3 + 1 \cdot 3^2 + 0 \cdot 3 + 1$$

||

$$(100110011)_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2 + 1$$

Sistemas de numeración

Cambio de base (parte entera)

- Vimos cómo pasar de cualquier base a base 10,
- Ahora solo resta poder pasar de base 10 a cualquier base:

Restas sucesivas:

$$\begin{array}{r} 104 \\ -81 = 3^4 \cdot 1 \\ \hline 23 \\ -0 = 3^3 \cdot 0 \\ \hline 23 \\ -18 = 3^2 \cdot 2 \\ \hline 5 \\ -3 = 3^1 \cdot 1 \\ \hline 2 \\ -2 = 3^0 \cdot 2 \\ \hline 0 \end{array}$$

$$(104)_{10} = (10212)_3$$

Restos de cocientes:

$$\begin{array}{r} 104 \overline{) 3} \\ 2 \quad 34 \overline{) 3} \\ 1 \quad 11 \overline{) 3} \\ 2 \quad 3 \overline{) 3} \\ 0 \quad 1 \end{array}$$

$$(104)_{10} = (10212)_3$$

Sistemas de numeración

Cambio de base (parte fraccionaria)

- > Vimos cómo pasar de cualquier base a base 10,
- > ahora solo resta poder pasar de base 10 a cualquier base:

Restas sucesivas:

$$\begin{array}{r} 0.4304 \\ -0.4000 = 5^{-1} \cdot 2 \\ \hline 0.0304 \\ -0.0000 = 5^{-2} \cdot 0 \\ \hline 0.0304 \\ -0.0240 = 5^{-3} \cdot 3 \\ \hline 0.0064 \\ -0.0064 = 5^{-4} \cdot 4 \\ \hline 0.0000 \end{array}$$

$$(0.4304)_{10} = (0.2034)_5$$

Restos de cocientes:

$$\begin{array}{r} 0.4304 \\ * 5 \\ \hline 2.1520 \\ * 5 \\ \hline 0.7600 \\ * 5 \\ \hline 3.8000 \\ * 5 \\ \hline 4.0000 \end{array}$$

$$(0.4304)_{10} = (0.2034)_5$$

Sistemas de numeración

Cambio de base (parte fraccionaria)

- > No toda parte fraccionaria puede ser cambiada de base en forma exacta resultando en una expresión finita:

$$(0.2)_3 = 2 \cdot 3^{-1} = 2/3$$

- > Por ello, las partes fraccionarias solo pueden ser estimadas con cierta precisión (i.e. con una cantidad de decimales)

Sistemas de numeración

Magnitudes con signo

Magnitud signada

- > Se utiliza el dígito más a la izquierda para representar el signo
 - el número $(a_n a_{n-1} \dots a_1 a_0)_b$ es positivo si $a_n = 0$ y negativo en caso contrario
 - $(a_{n-1} \dots a_1 a_0)_b$ representa la magnitud
 - permite representar las magnitudes $[-(b^{n-1}-1), b^{n-1}-1]$
- > La suma es igual que la operación clásica incluyendo el concepto de **acarreo** y como se cuenta con una cantidad fija de dígitos puede provocar **overflow**

Ejemplo en base 10:

$$\begin{array}{r} 1101001 \quad \leftarrow \text{acarreo} \\ 08234158 \\ 01947129 \\ \hline \text{overflow} \rightarrow 10181287 \end{array}$$

Sistemas de numeración

Magnitudes con signo

Magnitud signada

- > Se utiliza el dígito más a la izquierda para representar el signo
 - el número $(a_n a_{n-1} \dots a_1 a_0)_b$ es positivo si $a_n = 0$ y negativo en caso contrario
 - $(a_{n-1} \dots a_1 a_0)_b$ representa la magnitud
 - permite representar las magnitudes $[-(b^{n-1}-1), b^{n-1}-1]$
- > La suma es igual que la operación clásica incluyendo el concepto de **acarreo** y como se cuenta con una cantidad fija de dígitos puede provocar **overflow**

Ejemplo en base 2:

$$\begin{array}{r} 1101001 \\ 01101001 \\ 00101101 \\ \hline 10010110 \end{array}$$

← acarreo

overflow →

The diagram illustrates a binary addition in base 2. The numbers 1101001, 01101001, and 00101101 are added. The result 10010110 is shown with a horizontal line above it. The first digit '1' of the result is circled in red, and an arrow labeled 'overflow' points to it. An arrow labeled 'acarreo' (carry) points to the first digit of the top number.

Sistemas de numeración

Magnitudes con signo

Sistema de complemento

- > Se utilizan los números más grandes para representar los números negativos
- > El complemento de un número n se obtiene restando n al número más grande que puede representarse
Si el número más grande representable es b^{k-1} entonces $-n$ se codifica en complemento como $b^{k-1}-n$

Ejemplo en base 10:

$$(-52)_{10} = (999)_{10} - (52)_{10} = (947)_{10}$$

Sistemas de numeración

Magnitudes con signo

Sistema de complemento

- > Se utilizan los números más grandes para representar los números negativos
- > El complemento de un número n se obtiene restando n al número más grande que puede representarse
Si el número más grande representable es b^{k-1} entonces $-n$ se codifica en complemento como $b^{k-1}-n$

Ejemplo en base 2:

$$(-10)_2 = (111)_2 - (010)_2 = (101)_2$$

Sistemas de numeración

Magnitudes con signo

Sistema de complemento

- Permite resolver la resta de dos números usando solo sumas y la obtención de complemento

Ejemplo en base 10:

$$\begin{aligned}(167)_{10} - (52)_{10} &= (167)_{10} - (52)_{10} + (1000)_{10} - (1000)_{10} \\&= (167)_{10} - (52)_{10} + [(999)_{10} + (1)_{10} - (1000)_{10}] \\&= (167)_{10} + [(999)_{10} - (52)_{10}] + (1)_{10} - (1000)_{10} \\&= (167)_{10} + (947)_{10} + (1)_{10} - (1000)_{10}\end{aligned}$$

$$\begin{aligned}(167)_{10} - (52)_{10} &= (167)_{10} + [(999)_{10} - (52)_{10}] \\&= (167)_{10} + (947)_{10} \\&= (1114)_{10}\end{aligned}$$

Se suma descartando el acarreo

$$(114)_{10} + (1)_{10} = (115)_{10}$$

Se adiciona 1

Sistemas de numeración

Magnitudes con signo

Complemento a 1

- > Especializa lo visto en base 10 para números binarios
- > El **complemento a 1** de un número es invertir los bits que lo componen:

$$(-101)_2 = (1111)_2 - (0101)_2 = (1010)_2$$

Los números “más grandes” del rango se distinguen de los “más chicos” por tener el bit más significativo en 1 en lugar de 0,

- > El rango representado es el mismo que en signed magnitude (si no tomamos en cuenta el bit que determina si un número es “grande” o “chico” nos quedan $n-1$ bits para la magnitud y uno para el signo)

Sistemas de numeración

Magnitudes con signo

Complemento a 2

- > El **complemento a 2** es similar al complemento a 1 salvo que se se calcula en relación al número más chico mayor que el máximo representable:

$$(-101)_2 = (10000)_2 - (0101)_2 = (1011)_2$$

- > A diferencia de complemento a 1, el rango representado es $[-2^{n-1}, 2^{n-1}-1]$
- > Se computa como el complemento a 1 y se le suma 1 al resultado

Sistemas de numeración

Magnitudes con signo

Representación decimal de un número en complemento a 2

–> Para magnitudes **positivas** se evalúa el polinomio en base 2

–> Para magnitudes **negativas**:

1. Se invierten los bits

2. Se adiciona 1

3. Se convierte a decimal como si fuera una magnitud positiva

4. Se coloca el signo -

Complemento a 2

Sistemas de numeración

Magnitudes con signo

Suma de números en complemento a 2

- La suma se realiza como una suma binaria clásica
- Cambia la detección de **overflow** si el acarreo sobre el bit más significativo es igual al acarreo fuera de la representación, no hay overflow, si fueran distintos sí lo hay:

no overflow

acarreo

1 ←

$$\begin{array}{r} 1101001 \\ 01101001 \\ 10101101 \\ \hline 00010110 \end{array}$$

overflow

acarreo

0 ←

$$\begin{array}{r} 1101001 \\ 01101001 \\ 01101101 \\ \hline 11010110 \end{array}$$

Sistemas de numeración

Magnitudes con signo

Multiplicación de números enteros en complemento a 2

- La multiplicación binaria se realiza como sucesión de sumas desplazadas tal como en base 10 pero resulta muy fácil: si el bit es 1 se suma el multiplicando y se desplaza, si es 0 solo se desplaza:

Multiplicando	Multiplicando	Producto parcial
1001	1101	
10010	1101	00001001
100100	1101	00001001
1001000	1101	00101101
	1101	01110101

Sistemas de numeración

Magnitudes con signo

División de números enteros en complemento a 2

- > La división se realiza a partir de realizar sustracciones sucesivas de divisor al dividendo
- > En el caso de la división entera el resultado se expresa como el par (cociente, resto)
- > En el caso de la división no entera depende de la representación y lo veremos más adelante cuando veamos la representación de los números reales

Sistemas de numeración

Magnitudes con signo

Notación exceso

- > La notación **exceso-N** es similar al complemento a 2 salvo que se assume que el número 0...00 representa la magnitud menos significativa mientras que 1...11 representa a la más significativa
- > Si se trata de representaciones de k bits $N = 2^{k-1}$ se obtiene una representación análoga a complemento a 2 pero en la que 0 está representado por el bit más significativo en 1 seguido de 0s
- > Permite representar rangos asimétricos pues N determina la representación del 0: veamos $(-101)_2$ (i.e. -5) en notación exceso-7
$$\begin{aligned}(-101)_2 &= (00111)_2 - (101)_2 = (00010)_2 \\ (-111)_2 &= (00111)_2 - (111)_2 = (00000)_2\end{aligned}$$