LICENCIATURA EN INFORMATICA

PROGRAMADOR UNIVERSITARIO

INGENIERÍA EN INFORMÁTICA

ELEMENTOS DE COMPUTACIÓN Y LÓGICA

2023

Lic. Mariela A. Velázquez - Dr. Leonardo Albarracín

Universidad Nacional de Tucumán

2020 - AÑO DEL GENERAL MANUEL BELGRANO

Contenido

Unidad 1. Proposiciones Simples y Compuestas. Representación simbólica. Conectivos lógicos: Negación, Conjunción, Disyunción, Condicional, Bicondicional. Tablas de verdad. Jerarquía de conectivos. Clasificación de fórmulas lógicas por su significado: Tautologías, contradicciones y contingencias. Implicación y equivalencia lógica. Leyes Lógicas

Unidad 2. Razonamientos. Componentes: premisas, conclusión. Reglas de inferencia. Métodos de demostración: directo e indirecto. Validez de un razonamiento. Consistencia de Premisas.

Unidad 3. Lógica de Predicados. Funciones proposicionales. Universo del discurso. Representación simbólica. Cuantificadores: Cuantificador universal y cuantificador existencial. Alcance de un cuantificador. Variables libres y variables vinculadas. Proposiciones categóricas. Negación de proposiciones cuantificadas. Equivalencia de proposiciones cuantificadas universalmente y existencialmente. Razonamientos. Reglas de especificación universal y existencial. Reglas de generalización universal y existencial.

Unidad 4. Álgebras Booleanas y circuitos combinatorios. Propiedades de los circuitos combinatorios. Funciones Booleanas.

Unidad 5. Sistemas de numeración binario. Conversiones. Suma y resta de binarios. Sistema binario: números negativos. Sistema de numeración octal. Conversiones. Sistema de numeración hexadecimal.

Unidad 6. Diseño de Algoritmos. La programación como una metodología. Diseño de Algoritmos. Métodos de refinamientos sucesivos. Lenguaje de Diseño de programas. La programación estructurada. Estructuras algorítmicas fundamentales. Formas de reducción de complejidad: secuenciación, análisis por casos, análisis Iterativo. Generalización del concepto de procedimiento: Acciones parametrizadas, funciones.

Unidad N° 5: Sistema de numeración binario. Conversiones. Suma y resta de binarios. Sistema binario: números negativos. Sistema de numeración octal. Sistema de numeración hexadecimal.

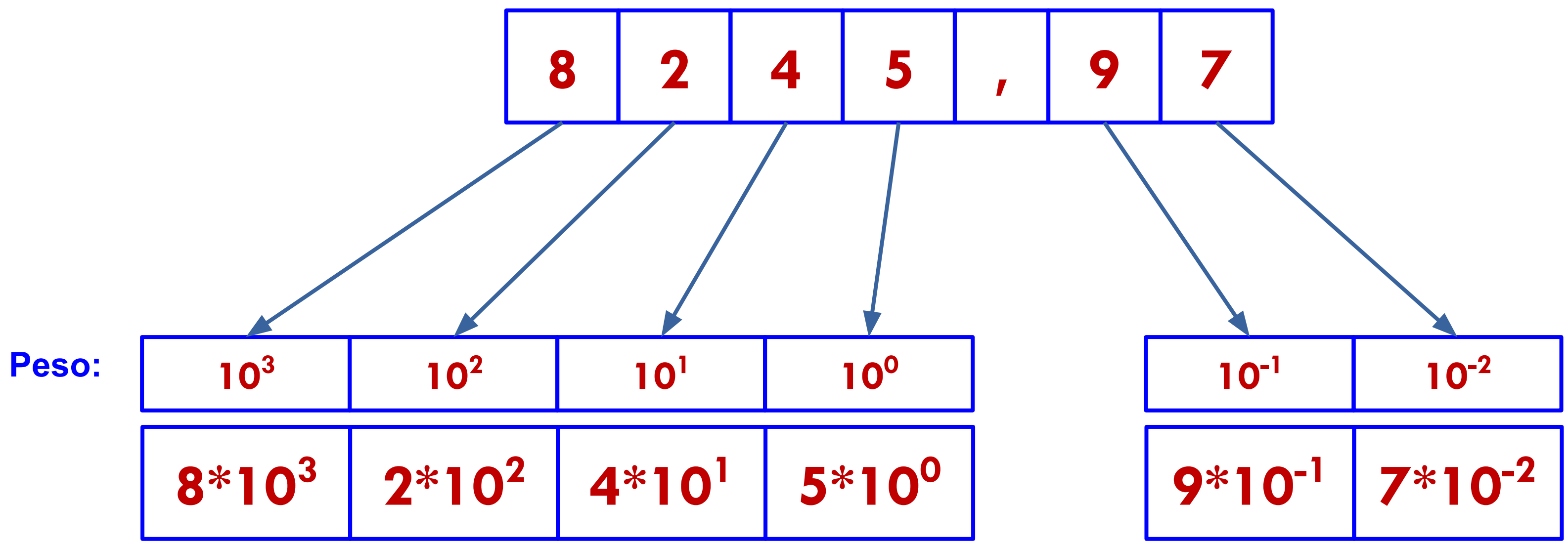
Sistema de Numeración

- Un sistema de numeración es un **conjunto de símbolos y reglas** que permiten representar datos numéricos.
- Los sistemas de numeración actuales son sistemas posicionales, ya que un símbolo tiene distinto valor según la posición que ocupa en la cifra.

- El sistema de **numeración decimal se compone de diez símbolos o dígitos** (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9) a los que otorga un valor dependiendo de la posición que ocupen en la cifra: unidades, decenas, centenas, etc. El valor de cada dígito está asociado al de una potencia de base 10 y un exponente igual al peso correspondiente de acuerdo a la posición que ocupa. En el caso de números con decimales, los exponentes de las potencias serán negativos.

Sistema de Numeración

Por ejemplo, el número **8245,97** se calcularía como:



= 8000 + 200 + 40 + 5 + 0,9 + 0,07

= 8245,97

SISTEMA DE BINARIO

Las computadoras trabajan con un sistema que **utiliza sólo dos valores** para manipular cualquier tipo de información. Esto quiere decir que todas las operaciones que la computadora hace, son realizadas utilizando sólo dos valores, que son los dígitos “0” (cero) y “1” (uno).

En las computadoras estos ceros “0” y unos “1” son llamados **dígitos binarios o solamente “bit”**, la cual es una conjunción de dos palabras de la lengua inglesa: **“binary digit”.**

Sistema de Binario

- El sistema de numeración binario utiliza sólo **dos dígitos**: el cero (0) y el uno (1).
- Cada dígito tiene distinto valor dependiendo de la posición que ocupe.
- El valor de cada posición es el de una potencia de base 2, elevada a un exponente igual al peso correspondiente de acuerdo a la posición que ocupa.

Conversión del sistema de numeración binario a decimal

Convertir el número binario 1011 a base decimal:

1	0	1	1

$1*2^3$	$0*2^2$	$1*2^1$	$1*2^0$

$$= 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$$

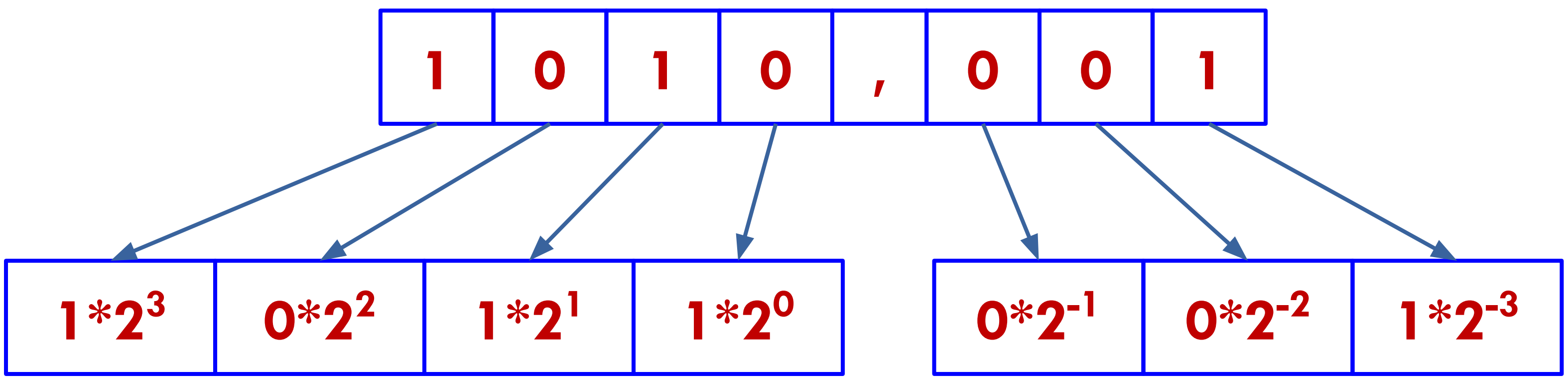
$$= 8 + 0 + 2 + 1 = (11)_{10}$$

Para expresar que ambas cifras describen la misma cantidad lo escribimos:

$$(1011)_2 = (11)_{10}$$

Conversión del sistema de numeración binario a decimal

Convertir el número binario $(1010{,}001)_2$ a base decimal:



$$= 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0 + 0*2^{-1} + 0*2^{-2} + 1*2^{-3}$$

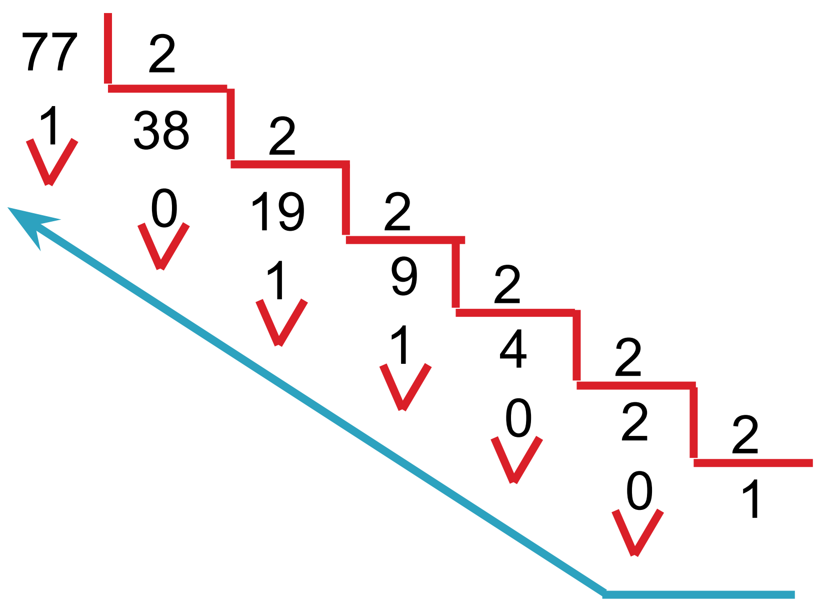
$$= 8 + 0 + 2 + 0 + 0 + 0 + 0.125 = (10{,}125)_{10}$$

Para expresar que ambas cifras describen la misma cantidad lo escribimos:

$$(1010{,}001)_2 = (10{,}125)_{10}$$

Conversión del sistema de numeración decimal a binario

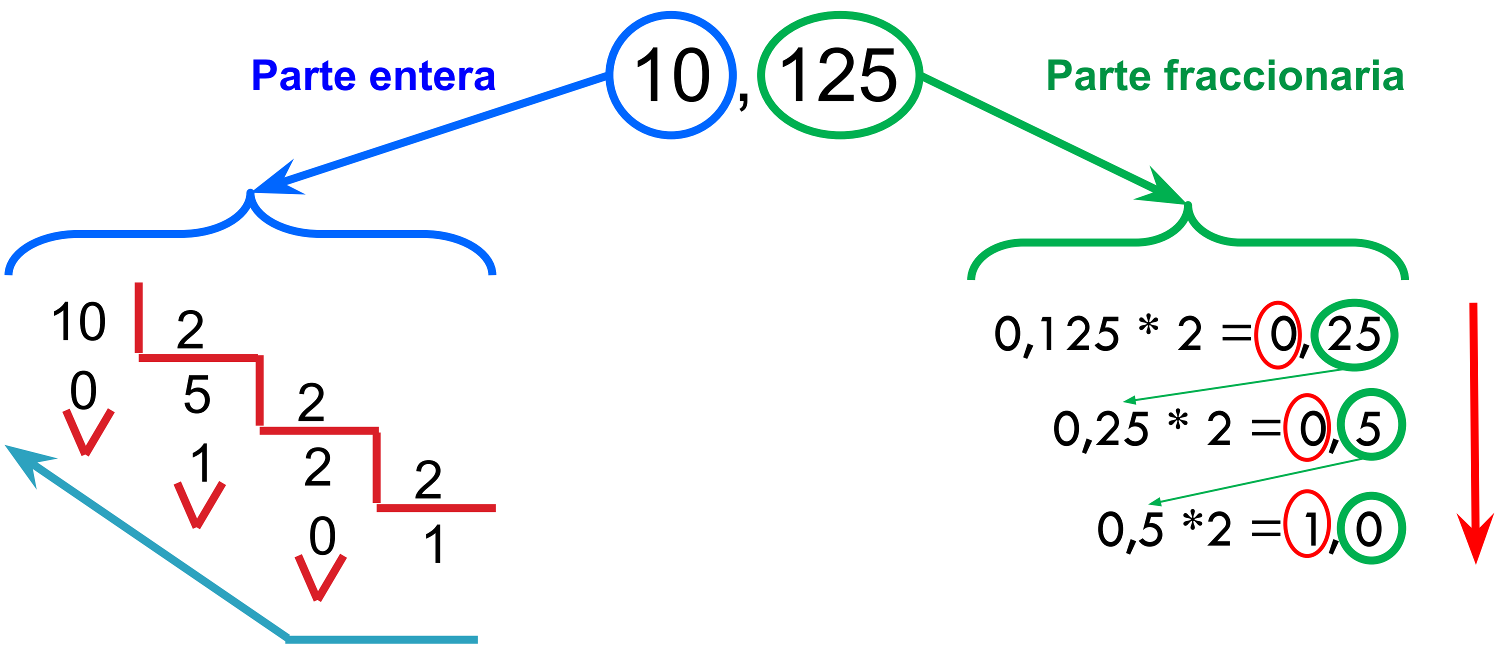
Para convertir un número decimal al sistema binario, se realizan **divisiones sucesivas por 2** y escribir los restos obtenidos en cada división en orden inverso al que han sido obtenidos. Por ejemplo, para convertir al sistema binario el número 77_{10} haremos una serie de divisiones que arrojarán los restos siguientes:



Luego podemos componer la cifra binaria $(77)_{10}$ = $\mathbf{(1001101)_2}$

Conversión del sistema de numeración decimal a binario

Ahora ¿qué sucede si tenemos un número fraccionario? ¿Lo podemos representar en binario?. Por ejemplo, para convertir al sistema binario el número $(10{,}125)_{10}$ haremos las siguientes operaciones:



Luego podemos componer la cifra binaria $(10{,}125)_{10}$ = **$(1010{,}001)_2$**

Conversión del sistema de numeración decimal a binario

Ejercicio:

Convierta $(5,1)_{10}$ a base 2.

5 , 1

101

0,1 * 2 = **0**,2
0,2 * 2 = **0**,4
0,4 * 2 = **0**,8
0,8 * 2 = **1**,6
0,6 * 2 = **1**,2
0,2 * 2 = **0**,4
0,4 * 2 = **0**,8
0,8 * 2 = **1**,6
0,6 * 2 = **1**,2
0,2 * 2 = **0**,4

$$(5,1)_{10} = (101,0\ 0011\ 0011\ 0....)_2$$

Conversión del sistema de numeración decimal a binario

La cantidad de dígitos necesarios para representar un número en el sistema binario es mayor que en el sistema decimal para representar el número 77, en el sistema decimal está compuesto tan sólo por dos dígitos, han hecho falta siete dígitos en binario.

Para representar números grandes harán falta muchos más dígitos. Por ejemplo, para representar números mayores de 255 se necesitarán más de ocho dígitos, porque $2^8 = 256$, por tanto, 255 es el número más grande que puede representarse con ocho dígitos.

Conversión del sistema de numeración decimal a binario

Como regla general, con n dígitos binarios pueden representarse un máximo de 2^n, números. El número más grande que puede escribirse con n dígitos es : $2^n – 1$.

Con cuatro bits, por ejemplo, pueden representarse un total de 16 números, porque $2^4 = 16$ y el mayor de dichos números es el 15, porque $2^4-1 = 15$.

Números representables = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}

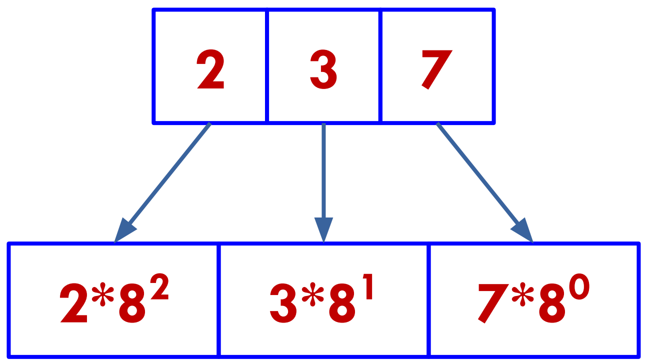
Sistema de Numeración Octal

El inconveniente de la codificación binaria es que la representación de algunos números resulta muy larga. Por este motivo se utilizan otros sistemas de numeración que resulten más cómodos de escribir: **el sistema octal y el sistema hexadecimal**. Afortunadamente, resulta muy fácil convertir un número binario a octal/hexadecimal.

En el **sistema de numeración octal**, los números **se representan mediante ocho dígitos** diferentes: 0, 1, 2, 3, 4, 5, 6 y 7. Cada dígito tiene, naturalmente, un valor distinto dependiendo del lugar que ocupen. El valor de cada una de las posiciones viene determinado por las potencias de base 8

Conversión: Sistema de numeración octal a decimal

La conversión de un número octal a decimal se realiza conociendo el peso de cada posición en una cifra octal. Por ejemplo, para convertir el número 237_8 a decimal:



$$2*8^2 + 3*8^1 + 7*8^0 = 128 + 24 + 7 = 159_{10}$$

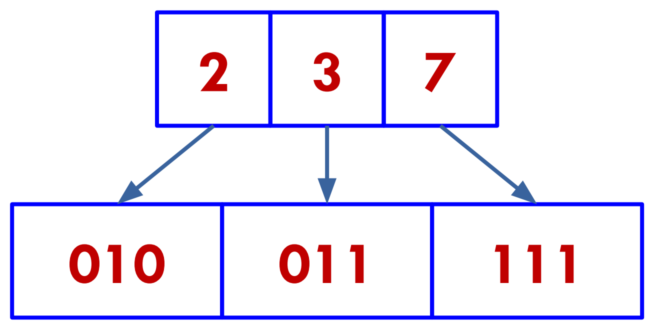
$$\mathbf{(237)_8 = (159)_{10}}$$

Conversión: Sistema de numeración octal a binario

La conversión de un número octal a binario se usa **“tres lugares”** para cada uno de esos números en base ocho.

Por ejemplo, para convertir el número 237_8 a binario:

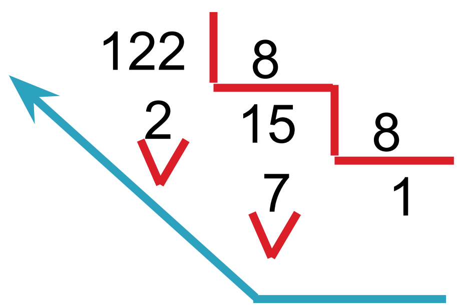
Octal	Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



$$(237)_8=(010\ 011\ 111)_2$$

CONVERSIÓN: SISTEMA DE NUMERACIÓN DECIMAL A OCTAL

La conversión de un número decimal a octal se hace con la misma técnica que ya hemos utilizado en la conversión a binario, mediante **divisiones sucesivas por 8** y colocando los **restos obtenidos en orden inverso**. Por ejemplo, para escribir en octal el número decimal 122_{10} tendremos que hacer las siguientes divisiones:



Tomando los restos obtenidos en orden inverso tendremos la cifra octal:

$$\mathbf{122_{10} = 172_8 = (001\ 111\ 010)_2}$$

Sistema de numeración hexadecimal

En el sistema hexadecimal los números se representan con dieciséis símbolos: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F**. Se utilizan los caracteres A, B, C, D, E y F representando las cantidades decimales 10, 11, 12, 13, 14 y 15. El valor de cada uno de estos símbolos depende, como es lógico, de su posición, que se calcula mediante potencias de base 16.

Calculemos, a modo de ejemplo, el valor del número hexadecimal $1A3F_{16}$:

$$1A3F_{16} = 1*16^3 + A*16^2 + 3*16^1 + F*16^0$$

$$=1*4096 + 10*256 + 3*16 + 15*1$$

$$= 6719$$

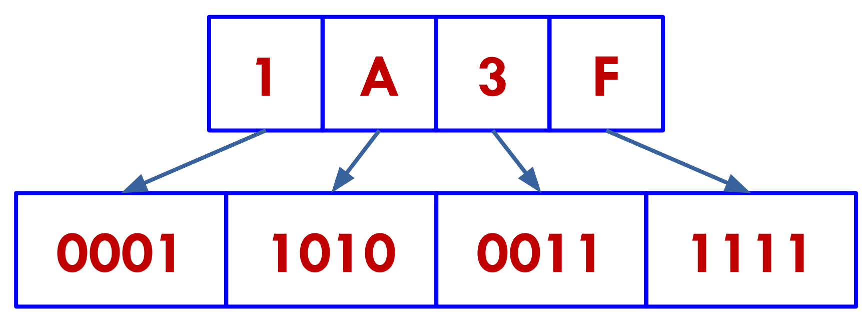
$$\mathbf{1A3F_{16} = 6719_{10}}$$

Conversión: Sistema de numeración hexadecimal a binario

La conversión de un número hexadececimal a binario se usa **"cuatro lugares"** para cada uno de esos números en base hexadecimal.

Por ejemplo, para convertir el número $1A3F_{16}$ a binario:

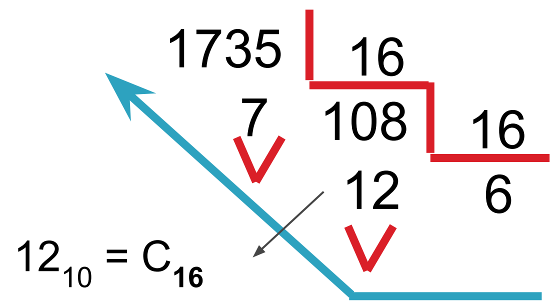
Hexa	Binario	Hexa	Binario
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111



$$(1A3F)_{16}=(0001\ 1010\ 0011\ 1111)_2$$

Conversión: Sistema de numeración decimal a hexadecimal

Ensayemos, utilizando la técnica habitual de divisiones sucesivas, la conversión de un número decimal a hexadecimal. Por ejemplo, para convertir a hexadecimal del número 1735_{10} será necesario hacer las siguientes divisiones:



De ahí que, tomando los restos en orden inverso, resolvemos el número en hexadecimal:

$$\mathbf{1735_{10} = 6C7_{16} = (0110\ 1100\ 0111)_2}$$

Sistema Binario: Operaciones aritméticas

En este curso solo nos enfocaremos en suma y resta de binarios

Suma
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

Veamos un ejemplo de sumas :

$$\begin{array}{r} 1011 \\ +\quad 101 \\ \hline 10000 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \rightarrow \end{array} \quad \begin{array}{r} (11)_{10} \\ +\quad (5)_{10} \\ \hline (16)_{10} \end{array}$$

$$\begin{array}{r} 1011 \\ +\quad 110 \\ \hline 10001 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \rightarrow \end{array} \quad \begin{array}{r} (11)_{10} \\ +\quad (6)_{10} \\ \hline (17)_{10} \end{array}$$

Sistema Binario: Operaciones aritméticas

Resta

0 - 0 = 0

1 - 0 = 1

1 - 1 = 0

0 - 1 = 1

Habiendo tomando 1 prestado al dígito anterior

Realicemos los siguientes ejemplos

1 0 0 0 1	→	$(17)_{10}$
- 0 1 0 1 0	→	- $(10)_{10}$
0 0 1 1 1	→	$(7)_{10}$

1 1 0 1 1 0 0 1	→	$(217)_{10}$
- 1 0 1 0 1 0 1 1	→	- $(171)_{10}$
0 0 1 0 1 1 1 0	→	$(46)_{10}$

Resta
0 - 0 = 0
1 - 0 = 1
1 - 1 = 0
0 - 1 = 1

Habiendo tomando 1 prestado al dígito anterior

Podríamos realizar la siguiente resta?

$$\begin{array}{r} 0\,1\,0\,0 \\ -\quad 0\,1\,0\,1 \\ \hline ? \end{array} \longleftarrow \begin{array}{r} (4)_{10} \\ -\quad (5)_{10} \\ \hline (-1)_{10} \end{array}$$

Sistema binario: Números Negativos

Al igual que el sistema de numeración decimal, el sistema binario tiene diferentes "formas" de representar un número negativo, las cuales enunciaremos a continuación.

Signo y Magnitud

En este modelo se toma el bit más significativo para representar el signo del número.

- Si el bit es 0, entonces el número es positivo, por ejemplo: $(0\,|\,0111)_2 = (7)_{10}$
- Si el bit es 1, entonces el número es negativo, por ejemplo: $(1\,|\,0111)_2 = (-7)_{10}$

Sistema binario: Números Negativos

Signo y Magnitud

Esta forma de representar tiene las siguientes características:

- El rango de enteros representables con una secuencia de N bits es:

  $[- (2^{N-1} – 1), +(2^{N-1} – 1)]$

- Existen dos configuraciones para el cero: +0 y -0
- La representación no es consistente para las operaciones aritméticas.

Sistema binario: Números Negativos

Complemento a 1

Este sistema se denomina también negación, pues la regla para obtener dicho complemento es cambiar ceros por uno y uno por ceros. Por ejemplo:

$$(20)_{10} = (0\ \ 10100)_2$$

$$(-20)_{10} = (1\ \ 01011)_2$$

Esta forma de representar tiene la siguiente dificultad, al igual que signo y magnitud:

$$0 \mid 0\,0\,0\,0\,0\,0\,0\ (+0)$$

$$1 \mid 1\,1\,1\,1\,1\,1\,1\ (-0)$$

Dos formas distintas de representar el cero.

Sistema binario: Números Negativos

Complemento a 2

El complemento a dos se pueden realizar de dos formas:

1ra opción

a) Aplicar complemento a uno, es decir, cambio cero por uno y uno por ceros

b) El número en complemento a uno, le sumamos 1 al bit de menor significación.

Por ejemplo: $(\mathbf{10100})_2$

$$
\begin{array}{r}
01011 \\
+\quad\quad 1 \\
\hline
01100
\end{array}
$$

Sistema binario: Números Negativos

Complemento a 2

2da opción

Aplicar complemento a uno, es decir, cambio cero por uno y uno por ceros, hasta llegar al 1 menos significativo

Por ejemplo: **$(10100)_2$** → **01100**

Sistema binario: Números Negativos

Restas con complemento a dos

$$\begin{array}{rr} & (4)_{10} \\ - & (5)_{10} \\ \hline & (-1)_{10} \end{array} \quad \left. \right\} \quad \begin{array}{rr} & 0\,|\,1\,0\,0 \\ - & 0\,|\,1\,0\,1 \\ \hline & ? \end{array} \longrightarrow \begin{array}{rr} & 0\,|\,1\,0\,0 \\ + & 1\,|\,0\,1\,1 \\ \hline & 1\,|\,1\,1\,1 \end{array}$$

Complemento a 2
de 5

$$\longrightarrow \quad (1\,|\,001)_2 = (-1)_{10}$$

SE REALIZA
Complemento a 2
Resultado negativo

Sistema binario: Números Negativos

Restas con complemento a dos

$$\begin{array}{r} (5)_{10} \\ -\quad (4)_{10} \\ \hline (1)_{10} \end{array} \quad \begin{array}{r} 0\,|\,1\,0\,1 \\ -\quad 0\,|\,1\,0\,0 \\ \hline ? \end{array} \longrightarrow \begin{array}{r} 0\,|\,1\,0\,1 \\ +\quad 1\,|\,1\,0\,0 \\ \hline 1\,\mathbf{0}\,|\,0\,0\,1 \end{array}$$

Complemento a 2 de 4

$\longrightarrow$ $(0|001)_2 = (1)_{10}$

NO SE REALIZA
Complemento a 2
Resultado positivo

Sistema binario: Números Negativos

Ejercicio: Realizar la siguiente operación 31 – 52 usando complemento a 2

$(31)_{10} = (11111)_2$ $(52)_{10} = (110100)_2$

$$\begin{array}{r} 0|0\,1\,1\,1\,1\,1 \\ -\quad 0|1\,1\,0\,1\,0\,0 \\ \hline ? \end{array} \xrightarrow[\text{Complemento a 2}]{} \begin{array}{r} 0|0\,1\,1\,1\,1\,1 \\ +\quad 1|0\,0\,1\,1\,0\,0 \\ \hline 1|1\,0\,1\,0\,1\,1 \end{array}$$

Se calcula el complemento 2 del resultado y recién se comprueba si es correcto.

$C2(101011) = (010101)_2 = 0*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$

$= 0 + 16 + 0 + 4 + 0 + 1 = (21)_{10}$

El resultado de la diferencia es: $(-21)_{10}$

WEBINAR

Cátedra de Elementos de Computación y Lógica

Hasta la próxima clase

p	q	p ∨ q
V	V	V
V	F	V
F	V	V
F	F	F