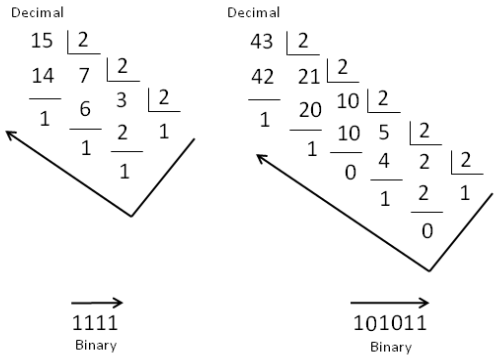
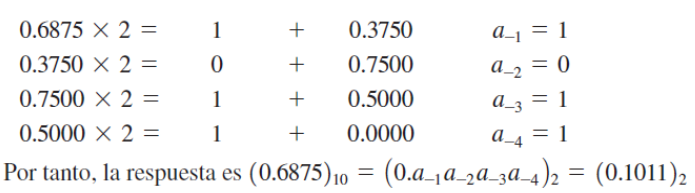
PRÁCTICO 1 - Sistemas de numeración

Tabla de equivalencia de codificación binaria - hexadecimal

Hexadecimal	Binary	Hexadecimal	Binary	Hexadecimal	Binary	Hexadecimal	Binary
0_{hex}	0000_{two}	4_{hex}	0100_{two}	8_{hex}	1000_{two}	c_{hex}	1100_{two}
1_{hex}	0001_{two}	5_{hex}	0101_{two}	9_{hex}	1001_{two}	d_{hex}	1101_{two}
2_{hex}	0010_{two}	6_{hex}	0110_{two}	a_{hex}	1010_{two}	e_{hex}	1110_{two}
3_{hex}	0011_{two}	7_{hex}	0111_{two}	b_{hex}	1011_{two}	f_{hex}	1111_{two}

Conversión de decimal a binario

Parte ENTERA	Parte FRACCIONARIA
	

Conversión de binario a decimal

El número decimal se obtiene de la sumatoria de todos los dígitos binarios multiplicado por el factor de 2^N, donde N es la posición del dígito binario en la secuencia, comenzando en 0 y creciendo de derecha a izquierda.

$$num_{10} = b_N*2^N + \ldots + b_3*2^3 + b_2*2^2 + b_1*2^1 + b_0*2^0 + b_{-1}*2^{-1} + b_{-2}*2^{-2} + b_{-3}*2^{-3} + \ldots + b_{-M}*2^{-M}$$

Cálculo del complemento a 2 de un número binario

Agregar ceros hasta completar el registro y verificar que el número se puede representar. Luego, negar bit a bit todos los dígitos del número y luego sumar 1 al resultado.

Conversión de binario negativo en representación "complemento a 2" a decimal

Obtener el complemento a 2 del número binario, convertir a decimal sin signo y agregar el signo negativo al resultado en decimal.

DATA ALIGNMENT

Double Word							
Word				Word			
Halfword		Halfword		Halfword		Halfword	
Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte

IEEE 754 FLOATING-POINT STANDARD ④

$(-1)^s \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - \text{Bias})}$
where Single Precision Bias = 127,
Double Precision Bias = 1023

IEEE 754 Symbols

Exponent	Fraction	Object
0	0	± 0
0	≠ 0	± Denorm
1 to MAX - 1	anything	± F1. Pt. Num.
MAX	0	± ∞
MAX	≠ 0	NaN

IEEE Single Precision and Double Precision Formats:

S.P. MAX = 255, D.P. MAX = 2047

S	Exponent	Fraction
31	30 23	22 0

S	Exponent	Fraction
63	62 52	51 0

Operation	Operand A	Operand B	Result indicating overflow
$A + B$	≥ 0	≥ 0	< 0
$A + B$	< 0	< 0	≥ 0
$A - B$	≥ 0	< 0	< 0
$A - B$	< 0	≥ 0	≥ 0

FIGURE 3.2 Overflow conditions for addition and subtraction.

Ejercicio 1:
Convertir los siguientes números en hexadecimal a binario de 32 bits:

a) 0xABCDEF00 b) 0x123456 c) 0x8E3FC581 d) 0x10A6F2B

Ejercicio 2:
Convertir los siguientes números en binario a decimal y a hexadecimal:

Binario	Decimal	Hexadecimal
$(1110011110000011)_2$		
$(10110111001101000101111)_2$		
$(10110011011011.11000010000)_2$		
$(1000111111010001111.000001101)_2$		

Ejercicio 3:
Suponiendo que se tienen registros de 16 bits, convertir a binario **sin** signo los siguientes números en base 10:

a) 123_{10} b) 59_{10} c) $255{,}46_{10}$ d) $98{,}019_{10}$

Ejercicio 4:
Suponiendo que un microprocesador utiliza registros de 8 bits y representación de números negativos en complemento a 2, muestre el contenido de estos registros al codificar en binario los siguientes números **con** signo:

a) -76_{10} b) -43_{10} c) $+64_{10}$ d) -121_{10}

Ejercicio 5:
Convertir los siguientes valores binarios de 8 bits en formato de complemento a dos a decimal:

a) 10010110 b) 11111011 c) 11100000 d) 00011110

Ejercicio 6:
Suponga que los registros A y B del microprocesador del ejercicio 4 (registros de 8 bits) contienen los valores 0x80 y 0xD0 respectivamente.

a) ¿Qué valor contiene el registro C después de ejecutar la operación C = A + B? ¿El resultado que se guarda en C es el esperado?
b) ¿Qué valor contiene el registro C después de ejecutar la operación C = A - B? ¿El resultado que se guarda en C es el esperado?
c) En base al análisis de las operaciones anteriores, ¿cuál es la ventaja de la representación de números negativos mediante su complemento a 2, por sobre la representación binaria regular + un bit de signo?

Ejercicio 7:
Expresar los siguientes números decimales en su representación binaria (negativos en complemento a 2) considerando los tamaños de los registros donde serán alojados según la tabla.

Decimal	**Binario**		
	Byte	**HalfWord**	**Word**
113			
-63			
319			
-128			
65535			
-149744			

Ejercicio 8:
Convertir los siguientes números decimales a formato IEEE 754 de precisión simple (normalizados):

a) 5678_{10} b) $306{,}59375_{10}$ c) $723{,}125_{10}$
d) $18{,}1953125_{10}$ e) -3020.993_{10} f) -0.000892_{10}

Ejercicio 9:
Convertir los siguientes en formato IEEE 754 de precisión simple (normalizados) a números decimales:

a) 1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 b
b) 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 b
c) 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 b
d) 0 0 1 1 1 0 0 1 0 0 1 0 0 1 1 0 1 0 1 1 1 0 0 1 0 0 1 1 1 1 0 1 b
e) 1 0 1 1 1 0 1 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 0 1 0 1 0 0 1 b
f) 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 b

Ejercicio 10:
Investigar cómo se escriben los símbolos especiales (“NaN”, “+Infinito”, “Infinito”, “+0”, “0”) en formato IEEE 754 de precisión simple (normalizados).

Ejercicios complementarios (para pensar)

Ejercicio 11:
Existe una representación denominada IEEE 754 de media precisión que sólo ocupa 16 bits de la siguiente forma: 1 para signo, 5 para exponente (con base +15) y 10 para mantisa. Tomar los tres números del ejercicio 8 y transformarlos a media precisión. ¡Cuidado! Puede que alguno no entre o haya que redondear. En cualquier caso escribir la diferencia entre el número obtenido en media precisión y el original en simple precisión.

Ejercicio 12:
Dar una secuencia de 3 números binarios de 32 bits que sea significativa tanto al interpretarlos como números binarios sin signo como números IEEE 754 de precisión simple. ¿Existe algún número binario de 32 bits que interpretado como binario sin signo sea igual a la interpretación IEEE 754 de simple precisión?

Ejercicio 13:
Convertir a decimal el siguiente número: 10000000000000000000000000000000 como si este se interpretará como un número:

a) Binario signado
b) Binario no signado
c) Formato IEEE 754