PROGRAMADOR UNIVERSITARIO
LICENCIATURA EN INFORMÁTICA
INGENIERIA EN INFORMÁTICA
Facultad de Ciencias Exactas y Tecnología
Universidad Nacional de Tucumán

ARQUITECTURA Y ORGANIZACIÓN DE COMPUTADORAS
TRABAJO PRÁCTICO REPASO
Sistemas de Numeración – Codificación y Operaciones en Binario

Problema 1

Convertir a decimal (Base 10) los siguientes números.

a) 10000111_2
b) 01110010_2
c) 01001010_2
d) 306_8
e) 47_8
f) 751_8
g) $10F_{16}$
h) AE_{16}
i) $2D_{16}$

Problema 2

Convertir los siguientes números decimales (Base 10) a las bases indicadas.

a) 111_{10} → Binario
b) 111_{10} → Octal
c) 111_{10} →Hexadecimal
d) 67_{10} → Binario
e) 67_{10} → Octal
f) 67_{10} → Hexadecimal
g) 127_{10} → Binario
h) 127_{10} → Octal
i) 127_{10} → Hexadecimal

Problema 3

Convertir los siguientes números a las bases indicadas desde sus correspondientes bases.

a) 11010010_2 → Octal
b) 10111100_2 → Hexadecimal
c) 356_8 → Binario
d) 356_8 → Hexadecimal
e) $3F_{16}$ → Binario
f) $3F_{16}$ → Octal

Problema 4

Realizar las siguientes sumas en las bases indicadas. Explicar qué pasa en los casos que se produce rebasamiento (Overflow). Suponer que se trabaja con la cantidad de dígitos indicados.

a) 011001012 + 101110002
b) 011100102 + 110100102
c) 5558 + 3408
d) 1778 + 2468
e) 4E16 + BC16
f) 70_{16} + $8E_{16}$

Problema 5

Realizar las sumas bajo las siguientes premisas:

1) Suponer que se trabaja en binario natural de 8 dígitos (sin signo). Indicar si hay rebasamiento, comprobar los resultados haciendo la misma operación en decimal.

2) Suponer que se trabaja en complemento a dos (con signo). Indicar si hay rebasamiento, comprobar los resultad haciendo la misma operación en decimal.

a) $00111011_2 + 00011001_2$
b) $01110011_2 + 11110101_2$
c) $10110011_2 + 01011101_2$
d) $10000001_2 + 11000011_2$

Problema 6

Representar los siguientes números decimales fraccionarios (Base 10) en las bases indicadas. Suponer que queremos resolver hasta 4 cifras en la parte fraccionaria del número.

a) $41,27_{10}$ → Binario
b) $25,34_{10}$ → Binario
c) $83,15_{10}$ → Octal
d) $122,66_{10}$ → Octal
e) $58,88_{10}$ → Hexadecimal
f) $110,91_{10}$ → Hexadecimal

Problema 7

Convertir a binario natural aquellos números que no lo sean y efectuar las correspondientes restas. Comprobar las operaciones trabajando en decimal. Suponer que estamos trabajando con binarios natural de 8 bits.

a) $01111000_2 - 01000101_2$
b) $01010100_2 - 60_{10}$
c) $170_8 - 01100111_2$
d) $7F_{16} - 27_{10}$
e) $5B_{16} - 77_8$
f) $64_{16} - 00110101_2$

Problema 8

Realizar la misma operación matemática que la del ejercicio anterior, pero esta vez usando el método de complemento a dos. Es decir, codifique el sustraendo en complemento a dos y haga la correspondiente suma.

Problema 9

Representar los siguientes números decimales en notación de punto flotante simple precisión según el estándar IEEE 754 single-precision-format (32 bits: 1 bit para signo, 8 bits para exponente, 23 bits para la mantisa).

a) 27_{10}
b) -88_{10}
c) $0{,}250_{10}$
d) $-0{,}500_{10}$

Problema 10

Convertir los siguientes números decimales a su representación binaria según el tipo de dato (C data types).

a) 250_{10} → signed int (16 bits)
b) 250_{10} → signed long (32 bits)
c) 250_{10} → float (32 bits: IEEE 754 single-precision)
d) -250_{10} → signed int (16 bits)
e) -250_{10} → signed long (32 bits)
f) -250_{10} → float (32 bits: IEEE 754 single-precision)
g) $-37{,}75_{10}$ → float (32 bits: IEEE 754 single-precision)
h) $-37{,}75_{10}$ → double (64 bits: IEEE 754 double-precision)