# Introducción a los Sistemas Lógicos y Digitales

## EJERCICIOS RESUELTOS

### *T.P. Nº 1: SISTEMAS NUMÉRICOS*

**1)** Convertir $1962_{10}$ a base 16:

$$\begin{array}{llll} 1962 & \lfloor 16 & & \\ 036 & 122 & \lfloor 16 & \\ \ \ 042 & \underline{10} & \underline{7} & \\ \ \ \ \underline{10} & & & \end{array} \Rightarrow \boxed{1962_{10} = 7AA}$$

Verificación:
$7AA = 7 \times 16^2 + A \times 16^1 + A \times 16^0 = 1962_{10}$

- Convertir $DABA_{16}$ a base 10:

$DABA_{16} = D \times 16^3 + A \times 16^2 + B \times 16^1 + A \times 16^0 =$
$= 13 \times 16^3 + 10 \times 16^2 + 11 \times 16^1 + 10 \times 16^0 = 55994_{10}$

- Convertir $FOCA_{16}$ a base 2:
Reemplazando cada dígito hexadecimal por su equivalente binario queda:

$$\boxed{FOCA_{16} = 1111\ 0000\ 1100\ 1010_2}$$

- Convertir $AF9_{16}$ a base 8:
Reemplazando cada dígito hexadecimal por su equivalente binario y luego agrupando de a 3 bits (comenzando por el bit menos significativo) queda:

$$\boxed{AF9_{16} = 1010\ 1001\ 1111 = 101\ 010\ 011\ 111 = 5237_8}$$

**2)** Convertir 31,25 a binario:
Primero convertimos la parte entera (31) a binario:

$$\begin{array}{lllll} 31 & |\underline{2} & & & \\ 11 & 15 & |\underline{2} & & \\ \ \underline{1} & \underline{1} & 7 & |\underline{2} & \\ & & \underline{1} & 3 & |\underline{2} \\ & & & \underline{1} & \underline{1} \end{array} \Rightarrow 31 = 11111$$

Cálculo de la parte fraccionaria (0,25):

$0{,}25 = b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + .... + b_{-n} \cdot 2^{-n}$

$2 \times 0{,}25 = \mathbf{0}{,}5 = b_{-1} + b_{-2} \,.\, 2^{-1} + .... + b_{-n} \,.\, 2^{-n+1} \quad \Rightarrow \quad \mathbf{b_{-1} = 0}$

$2 \times 0{,}5 = \mathbf{1} = b_{-2} + .... + b_{-n} \,.\, 2^{-n+2} \quad \Rightarrow \quad \mathbf{b_{-2} = 1}$

La parte fraccionaria es 0,01.

Por lo tanto: $\boxed{31{,}25_{10} = 11111{,}01_2}$

- Convertir 8AC,DFE a binario:

Se reemplaza cada dígito hexadecimal por su equivalente binario, respetando la posición de la coma:

$\boxed{8AC{,}DFE_{16} = 1000\ 1010\ 1100\ ,\ 1101\ 1111\ 1110_2}$

- Convertir $7122{,}77_8$ a binario:

Se reemplaza cada dígito octal por su equivalente binario:

$\boxed{7122{,}77_8 = 111\ 001\ 010\ 010\ ,\ 111\ 111_2}$

**3)** Convertir a binario y resolver: $776_8 - 335_8$

$$
\begin{array}{rcr}
 & & {\scriptstyle 0\,1} \\
776_8 & = & 1\,1\,1\ \ 1\,1\,1\ \ 1\,\cancel{1}\,0 \\
-\ \ 335_8 & = & -\ \underline{0\,1\,1\ \ 0\,1\,1\ \ 1\,0\,1} \\
 & & 1\,0\,0\ \ 1\,0\,0\ \ 0\,0\,1 \quad = \quad \mathbf{441_8}
\end{array}
$$

- Realizar la operación $876_{10}$ / $32_{10}$ en binario:

876 = 1101101100
32 = 100000

```
 1101101100   |100000
- 100000       11011,011
 0101101
-  100000
   00110110
-    100000
     0101100
    -  100000
       00110000
      -   100000
          0100000
        -   100000
     resto:      0
```

Por lo tanto:

$\boxed{876_{10} / 32_{10} = 11011{,}011_2}$

- $707_8 - 356_8$

$$
\begin{array}{rcl}
707_8 & = & 111\ 000\ 111 \\
- & & - \\
356_8 & = & \underline{011\ 101\ 110} \\
 & & 011\ 011\ 001 \quad = \quad \mathbf{331_8}
\end{array}
$$

**4)** Convertir a BCD y sumar**:** **- 107 + 33**

$$
\begin{array}{rcllll}
107 & = & 0001 & 0000 & 0111 & \text{(BCD)} \\
+ & & + & & & \\
33 & = & \underline{0000} & \underline{0011} & \underline{0011} & \text{(BCD)}
\end{array}
$$

$$
\underbrace{\mathbf{0001}}_{\mathbf{1}} \quad \begin{array}{c} 0011 \\ + \\ \underline{\boxed{1}} \\ \underbrace{\mathbf{0100}}_{\mathbf{4}} \end{array} \quad \begin{array}{c} 1010 \\ + \\ \underline{0110} \\ \underbrace{\mathbf{0000}}_{\mathbf{0}} \end{array}
$$

$1010 \leftarrow$ es mayor que 9, le sumo 6 (0110)

Rta.: 0001 0100 0000 = 140

**5) a)** Expresar en binario:

$4096_{10}$ en 11 bits: ¡no se puede! ya que $4096 = 2^{12} = 1000000000000$ (13 bits)

$16_{10}$ en 8 bits: 00010000

$128_{10}$ en 8 bits: 10000000 (¡OJO! Sin signo)

**5) b)** Operar en signo y módulo

e) C + A = - 71 + (- 127) = - 71 - 127

Como ambos son negativos, sumo los módulos y al resultado le agrego el signo negativo:

$$
\begin{array}{rcl}
71_{10} & = & 1000111 \\
127_{10} & = & \underline{1111111} \\
 & & 11000110 \quad \text{(módulo)}
\end{array}
$$

Por lo tanto: -71 - 127 $= \underbrace{\mathbf{1}}_{\mathbf{sg}} \ \underbrace{\mathbf{11000110}}_{\mathbf{módulo}}$

**6)** Realizar operaciones en complemento a uno.

**a) - 127 + (- 1)**

Hay que hallar los números en complemento a uno a partir del número positivo correspondiente y luego realizar la suma:

$$
\begin{array}{llllll}
+127 = 01111111 & \Rightarrow & -127 & = & \overset{1\leftarrow\ 0\leftarrow}{1\,0\,0\,0\,0\,0\,0\,0} & \text{(CA1) 8 bits} \\
 & & & & + & \\
+\quad 1 = 00000001 & \Rightarrow & -1 & = & \underline{1\,1\,1\,1\,1\,1\,1\,0} & \text{(CA1) 8 bits} \\
 & & & & \mathbf{1 \leftarrow 0\,1\,1\,1\,1\,1\,1\,0} &
\end{array}
$$

Sumamos dos números negativos y el resultado dio positivo, por lo tanto hubo OVERFLOW, en consecuencia el resultado no se puede representar en 8 bits, es INCORRECTO. Hubo un CARRY = 1.

**b) + 100 - ( - 48)**

Primero transformamos la resta en suma:

+ 100 + (CA1 (-48)) = +100 + 48

+48 = 00110000 ⇒ - 48 = 11001111 (se invierten todos los bits)
+100 = 01100100

$$
\begin{array}{lll}
+100 & : & \overset{0\leftarrow\ 1\leftarrow}{0\,1\,1\,0\,0\,1\,0\,0} \\
 & & + \\
+48 & : & \underline{0\,0\,1\,1\,0\,0\,0\,0} \\
 & & \mathbf{1\,0\,0\,1\,0\,1\,0\,0}
\end{array}
$$

hay OVERFLOW
resultado INCORRECTO
CARRY = 0

La suma de dos números positivos dio negativo, por lo tanto hubo OVERFLOW, el resultado es INCORRECTO y el CARRY = 0.
**Nota:** podemos detectar si hubo OVERFLOW analizando los CARRYs de los 2 bits más significativos MSB y (MSB-1) en la suma. Si son iguales (ambos "0" ó ambos "1") significa que NO HAY OVERFLOW. Si son distintos ("0" y "1" ó "1" y "0") significa que HAY OVERFLOW.

**7)** Realizar operaciones en complemento a 2.

- **(+ 127) + (- 2)**

+127 = 01111111
\+ 2 = 00000010 $\Rightarrow$ ( -2) = 11111110

$$\begin{array}{rll} & 01111111 & (+127) \\ + & & \\ & \underline{11111110} & (-2) \end{array}$$

se descarta $\rightarrow$ **1** $\leftarrow$ 01111101

$\boxed{\text{Rta.: } 01111101 \quad (125_{10})}$

En este caso hay CARRY = 1, no hay OVERFLOW y el resultado es CORRECTO.

Para 16 bits:

$$\begin{array}{rll} & 0000\ 0000\ 0111\ 1111 & (+127) \\ + & & \\ & \underline{1111\ 1111\ 1111\ 1110} & (-2) \end{array}$$

se descarta $\rightarrow$ **1** $\leftarrow$ $\boxed{0000\ 0000\ 0111\ 1101}$

Hay CARRY=1, no hay OVERFLOW, resultado CORRECTO.

- **FFE - AC0 en C. a 2**

FFE : 1111 1111 1110
AC0 : 1010 1100 0000

Como se trata de una resta, la transformamos en suma hallando el C. a 2 de AC0:

-AC0 : 0101 0100 0000 (C.a 2 de AC0)

El próximo paso es realizar la suma de ambos operandos:

$$
\begin{array}{rr}
 & {}^{1}\leftarrow{}^{1}\leftarrow 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0 \\
+ & \\
 & \underline{0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0} \\
1 \leftarrow & \boxed{0\,1\,0\,1\,0\,0\,1\,1\,1\,1\,1\,0}
\end{array}
$$

Hay CARRY=1, no hay OVERFLOW, resultado CORRECTO.

**8)** Pasar a decimal:

**8.1)** 11110101 (C. a 1)

Es un número negativo. Para hallar el módulo aplicamos la definición:

$$N^* = 2^n - 1 - N$$

$$
\begin{array}{rr}
 & 1\,0\,0\,0\,0\,0\,0\,0\,0 \\
- & \\
 & \underline{\phantom{0\,0\,0\,0\,0\,0\,0\,0}1} \\
 & 1\,1\,1\,1\,1\,1\,1\,1 \\
- & \\
 & \underline{1\,1\,1\,1\,0\,1\,0\,1} \\
 & 0\,0\,0\,0\,1\,0\,1\,0 \quad \rightarrow \quad 10_{10}
\end{array}
$$

Por lo tanto, el resultado final será: $\boxed{1\,1\,1\,1\,0\,1\,0\,1 = -\,10_{10}}$

Otra forma de obtener el número decimal negativo es:

**1 1 1 1 0 1 0 1** $= -\,(2^7 - 1) + 117 = -127 + 117 = -\,10_{10}$

**8.2)** Pasar de C. a 1 a C. a 2 y viceversa:

Los números positivos no cambian (su representación es la misma en signo y módulo, C. a 1 y C. a 2).

a) **1 0 1 1 1 0 1 0 (en C.a 1 a C.a 2)**

$$\left.\begin{array}{l} N_{C2} = 2^n - N \\ N_{C1} = 2^n - N - 1 \end{array}\right\} \qquad N_{C2} = N_{C1} + 1$$

Aplicando esta propiedad:

```
    1 0 1 1 1 0 1 0    ( C. a 1 )
+                 1
    ---------------
    1 0 1 1 1 0 1 1    ( C. a 2 )
```

**b) 1 0 0 0 0 0 0 0 ( en C. a 2 )**

Este número es el -128 en complemento a 2, pero en 8 bits en complemento a 1 no es posible representarlo.

**9)**

| decimal | exponentes (binario desplazado) | exponentes |
|---|---|---|
| 255 | 11111111 | reservado |
| 254 | 11111110 | + 127 |
| ...... | ................ | ........ |
| 246 | 11110110 | + 119 |
| ...... | ................ | ........ |
| 129 | 10000001 | + 2 |
| 128 | 10000000 | + 1 |
| 127 | 01111111 | 0 |
| 126 | 01111110 | -1 |
| ..... | ............... | ..... |
| 21 | 00010101 | -106 |
| ...... | ................ | ........ |
| 1 | 00000001 | -126 |
| 0 | 00000000 | reservado |

**a)** Representar **A = 1 10001010 11001010010111111000000** en complemento a 2 con formato **(8,4).**

Para obtener el exponente, a 10001010 le restamos el bias o desplazamiento 01111111, obteniendo el número 00001011 **(+11)**.

El signo de A es **negativo** (1).

Por lo tanto: **A = - 1,11001010010111111000000 x 2** $^{11}$

Que pasado a punto fijo es: **A = - 111001010010,111111000000** (que no es posible representar en formato (8,4). Podemos hacerlo en formato (16,4), quedando:

**A = - 0000111001010010,1111**

Finalmente, hallando el complemento a 2 queda:

$\boxed{\mathbf{A = 1111000110101101{,}0001}}$ (formato 16,4)

**b)** Representar **A = 1 10000011 1101001010 ... 0**

Restando el bias el exponente resulta: 00000100 ( + 4)

Por lo tanto **A = -1,1101001010 ... 0 x 2** $^{4}$ que en formato (8,4) resulta:

**A = - 00011101,0010**

Hallando el complemento a 2 resulta:

$\boxed{\mathbf{A = 11100010{,}1110}}$

**10)** Convertir a formato decimal.

**1 00011011 1010101 ... 1**

El signo del número es negativo.
Exponente: 00011011 = +27
Restándole el bias queda: + 27 - 127 = -100

Por lo tanto, el resultado definitivo será:

$\boxed{\mathbf{- 1{,}10101011111111111111111 \times 2^{-100}}}$

**11)** A = - 1,10 ... 0 x 2 $^{65}$

11000000 = + 192
exponente: 192 - 127 = + 65

B = + 1,00101 ... 1 x 2 $^{-33}$

01011110 = + 94
exponente: 94 - 127 = - 33

C = + 1,1101001 ... 1 x 2 $^{-107}$

00010100 = +20
20 - 127 = -107

**a) A x C + B**

A x C = -1,10 ... 0 x 2 $^{65}$ x 1,1101001 ... 1 x 2 $^{-107}$ =

$$= -1{,}10 \ldots 0 \times 1{,}1101001 \ldots 1 \times 2^{-42}$$

$$\begin{array}{rr}
 & 1{,}1101001 \ldots 1 \\
\times & 1{,}10 \ldots 0 \\
\hline
 & 111010011111111111111111 \\
+ & 111010011111111111111111\phantom{1} \\
\hline
 & 10{,}10111101111111111111101
\end{array}$$

Por lo tanto A x C = $- 10{,}10111101111111111111101 \times 2^{-42} =$
$= - 0{,}00000001010111101111111111111101 \times 2^{-33}$

Como es un número negativo, lo representamos en complemento a 2:

A x C (c. a 2) = $1{,}11111110101000010000000000000011 \times 2^{-33}$

**A x C + B** = $1{,}11111110101000010000000000000011 \times 2^{-33} + 1{,}00101 \ldots 1 \times 2^{-33}$

$$\begin{array}{rll}
 & 1{,}11111110101000010000000000000011 & \times 2^{-33} \\
+ & 1{,}0010111111111111111111 & \times 2^{-33} \\
\hline
 & \boxed{\mathbf{11{,}00101110101000001111111000000011 \times 2^{-33}}} &
\end{array}$$

**12) a)** 1 00000000 10010 ... 0 = **- 0,10010 ... 0**

**b)** 0 11111111 0 ...........0 = $+\infty$

**13) (Ejercicio optativo)**. Realizar la siguiente operación sin pasar a binario.

$\mathbf{724_8 - 762_8}$

Hallamos el complemento a la base 8 del $762_8$ y luego sumamos:

C.a 8 (762) = 015 + 1 = 016 (ya que el complemento a la base 8 es igual al C.a 7 +1)

Por lo tanto nos queda: $724_8 + 016_8 = \mathbf{742_8}$

Que es un número negativo, ya que en octal los números del 0 al 3 son positivos y los números del 4 al 7 son negativos. Entonces:

$742_8 = - (035 + 1) = \mathbf{-36_8}$

En definitiva: $\boxed{\mathbf{724_8 - 762_8 = -36_8}}$