Introducción a los Sistemas Lógicos y Digitales

Trabajo Práctico Nº 7

CIRCUITOS ARITMÉTICOS

Ejercicio nº 1

Expresar los números en complemento a uno y realizar las operaciones indicadas. Detectar en cada operación si se produce transporte (carry), desborde (overflow) y/o error.

a) Ejemplo resuelto:

- 127 + (- 1)

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

$+127 = 01111111 \Rightarrow -127 = 10000000$ (CA1) 8 bits

$+1 = 00000001 \Rightarrow -1 = 11111110$ (CA1) 8 bits

1← 0 1 1 1 1 1 1 0

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.

Nota: es posible 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.

b) (+68) – (+93) en base 10
c) 00101101 – 00101011
d) FE91 – 0D25

Ejercicio nº 2

Expresar los siguientes números en complemento a dos y realizar las operaciones indicadas. Detectar en cada operación si se produce transporte (carry), desborde (overflow) y/o error.

a) Ejemplo resuelto:

(+ 127) + (- 2)

+127 = 01111111
+ 2 = 00000010 ⇒ (-2) = 11111110

01111111 (+127)
+
11111110 (-2)

se descarta → **1** ← 01111101

Rta.: 01111101 (125_{10})

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

b) (–127) + (–1) en base 10
c) FEF – AB6 en base 16
d) 01101011 + 10010101

Ejercicio nº 3

Dados los siguientes números en punto flotante según norma IEEE754, convertirlos en formato decimal:

a) Ejemplo resuelto:

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:

- 1,10101011111111111111111 x 2 $^{-100}$

b) 0 01111110 10110010 0
c) 1 00011110110 1100110110 0

Ejercicio nº 4

a) Sintetizar un **sumador ripple carry de 4 bits**. Calcular los tiempos de retardo desde que se estabilizan las entradas hasta que las salidas también sean estables.

b) Idem para un **sumador look ahead carry de 4 bits**. Comparar los tiempos de retardo con los del caso a).
c) Idem para un **sumador carry select de 4 bits**. Comparar los tiempos de retardo con los dos sumadores anteriores.

Ejercicio nº 5

Diseñar un sumador de dos números binarios A y B sin signo tipo **carry-select** de **4** bits basado en dos sumadores tipo **ripple-carry** de **2** bits y lógica adicional.
Calcular el retardo en todas las salidas incluyendo el del carry de la última etapa, si se realiza la suma: A + B = 001 + 111.
Dibujar un diagrama de tiempos para mostrar en esa condición la evolución de todas las señales (considerar todas las compuertas con un retardo tpd).

Ejercicio nº 6

Implementar un **sumador serie sin signo de 4 bits** con entrada de carga paralelo y salida con conversión a modo paralelo para conectar en un microprocesador (usar registros de desplazamiento).

Ejercicio nº 7

Sintetizar un **multiplicador binario de 4 x 4 bits sin signo** utilizando el **algoritmo de Booth**.

Ejercicio nº 8

Sintetizar un circuito **multiplicador por 24 utilizando barrel shifters y un sumador.**