



Práctica 2: ALU: Circuitos Aritméticos

1. Analice a nivel compuertas los costos en términos de área y tiempo de cálculo de un sumador *ripple carry*.

- i) Para 4 bits
- ii) Para N bits

2. Analice a nivel de compuertas los costos en términos de área y tiempo de cálculo de un sumador *carry look-ahead*.

- i) Para 4 bits
- ii) Para N bits

3. Basado en los resultados de los ejercicios 1 y 2 compare los tiempos de cálculo de ambos sumadores si se conocen los tiempos de las siguientes compuertas elementales: $T_{and2} = 3ns$; $T_{or2} = 2.5ns$; $T_{xor2} = 4ns$; $T_{andK} = 5ns$; $T_{orK} = 4ns$ (para K entradas de la compuerta entre 3 y 5)

- a) Considere para la resolución que se posee compuertas *and* y *or* de hasta 5 entradas
- b) Considere para la resolución que se posee solamente compuertas *and* y *or* de 2 entradas

4. Realice un sumador binario secuencial de 8 bits (iteración en el tiempo). Utilice para la implementación registros de desplazamiento, flips-flops y:

- i) un sumador completo de 1 bit.
- ii) un sumador completo de 2 bits.
- iii) un sumador completo de 4 bits.

Determine cantidad de ciclos que involucra el cálculo de la suma de operandos de 8 bits, frecuencia de reloj y tiempo total de cálculo asumiendo T_{and2} , T_{or2} , T_{xor2} , T_{sReg} y T_{pReg} los tiempos de las 3 compuertas necesarias y los tiempos de escritura y lectura de registros respectivamente.

5. Realice el circuito completo de un sumador/restador de 4 bits que soporte operaciones en aritmética de cero desplazado (CD).

- a) Analice las condiciones de rebalse para esta aritmética e impleméntelas en el circuito pedido.
- b) compare el diseño obtenido con el sumador en C'2 del ejercicio 1-i) en función del área utilizada y del tiempo de cálculo

6. Realice las siguientes multiplicaciones usando para ello

a) El algoritmo de multiplicación con operandos en complemento a la base

b) El algoritmo de multiplicación con recodificación de Booth para el multiplicador:

considere multiplicando y multiplicador de 8 bits

a)
$$\begin{array}{r} -40 \\ 13 \end{array}$$

b)
$$\begin{array}{r} 56 \\ -8 \end{array}$$

c)
$$\begin{array}{r} -1 \\ -10 \end{array}$$

d)
$$\begin{array}{r} 85 \\ -85 \end{array}$$

7. Analice costos de multiplicadores *carry-save* y *ripple-carry* de 4 bits. Ventajas y desventajas.

8. Utilice el algoritmo de división con restauración para dividir dos valores naturales A (de 5 dígitos) y B (2 dígitos)

$$\begin{array}{l} A = 4567 \\ B = 62 \end{array}$$

$$\begin{array}{l} A = 12345 \\ B = 5 \end{array}$$

$$\begin{array}{l} A = 69 \\ B = 13 \end{array}$$

Repita los cálculos anteriores para obtener precisiones fraccionarias de dos dígitos adicionales

9. Utilice el algoritmo de división sin restauración para dividir dos valores fraccionarios normalizados A (de 8 bits) y B (5 bits) con precisión fraccionaria de 6 bits

$$\begin{array}{l} A = 0,00001110 \\ B = 0,11010 \end{array}$$

$$\begin{array}{l} A = 7,75_{(10)} \\ B = 8,5_{(10)} \end{array}$$

$$\begin{array}{l} A = 0,110111 \\ B = 0,0011 \end{array}$$

10. Calcular, utilizando el método de Newton-Raphson los inversos de los siguientes valores con 32 bits de precisión y suponiendo un valor inicial (en LUT) de 4 bits.

a) $28_{(10)}$ calcular en base 10

b) $9_{(10)}$ calcular en base 2

c) $0,011_{(2)}$ calcular en base 2 usando la aproximación del método

11. Analice de que parámetros depende el tamaño (ancho de celda y número de celdas) de la LUT usada para la implementación del método de Newton-Raphson para la obtención del inverso de un número. De la misma manera determine cuales deben ser los valores de cada celda.

En función de lo anterior, muestre la LUT necesaria para implementar el algoritmo para el inverso de un número con 32 bits de precisión fraccionaria final y 4 bits de precisión fraccionaria de X_0 (primera aproximación).