

PROBLEMAS DE ARITMETICA ENTERA

1.- Analizar el retardo, en número de etapas lógicas atravesadas, de cada uno de los sumadores de 64 bits construidos como se indica a continuación:

- a) Con propagación de arrastres.
- b) Módulos sumadores de 4 bits construidos con propagación de arrastres e interconectados con selección de arrastres.
- c) Módulos sumadores de 4 bits construidos con propagación de arrastres e interconectados con puenteo de arrastres.
- d) Módulos sumadores de 4 bits construidos con anticipación de arrastres e interconectados con selección de arrastres.
- e) Módulos sumadores de 4 bits construidos con anticipación de arrastres e interconectados con puenteo de arrastres.
- f) Módulos sumadores de 4 bits con anticipación de arrastres agrupados en módulos interconectados con anticipación de arrastres y los módulos agrupados en secciones también interconectada con anticipación de arrastres.
- g) Se usan cuatro sumadores de 16 bits encadenados con propagación de arrastres. Cada uno de estos módulos de 16 bits está construido conectando mediante puenteo de arrastres cuatro sumadores de 4 bits cada uno. De estos sumadores los tres primeros (aquellos que calculan los 12 bits menos significativos de la suma) están construidos internamente con propagación de arrastres y el último con anticipación de arrastres.

2.- Realizar la siguiente suma utilizando sumadores carry-save con la estructura básica vista en clase y con árboles de Wallace.

$$A+B+C+D+E+F$$

$$A = 101010$$

$$B = 010101$$

$$C = 000111$$

$$D = 101111$$

$$E = 101100$$

$$F = 011011$$

3.- Siguiendo la estructura de un multiplicador de Booth secuencial, multiplicar $a*b$.

$$a = 10011100$$

$$b = 11110001$$

Suponiendo que el retardo asociado a la unidad de control es de 1ns, que el sumador/restador tarda 5 ns en realizar cualquier operación, que los desplazamientos implican 1 ns, y que se desprecian los tiempos de setup, hold, clk_to_q, skew, y los retardos del interconexionado. ¿Cuál es el tiempo de ejecución de este algoritmo?.

4.- Siguiendo la estructura de un multiplicador recodificado por pares de bits secuencial, multiplicar $a*b$.

$$a = 10011100$$

$$b = 11110001$$

Suponiendo que el retardo asociado a la unidad de control es de 1ns, que el sumador/restador tarda 5 ns en realizar cualquier operación, que el multiplexor tarda 0,5 ns, y la unidad de realización del C2 tarda 0,5 ns, que los desplazamientos implican 1 ns, y que se desprecian los tiempos de setup, hold, clk_to_q, skew, y los retardos del interconexionado. ¿Cuál es el tiempo de ejecución de este algoritmo?.

5.- Siguiendo la estructura de un multiplicador salva-arrastre secuencial, multiplicar $a*b$.

$$a = 10011100$$

$$b = 11110001$$

6.- Construir un multiplicador de Pezaris que multiplique números de 4 bits por números de 5 bits, usando sólo sumadores de tipo 0 y 2.

Realizar sobre esa estructura la multiplicación $A*B$, siendo $A = 0110$ y $B = 10011$.

7.- Construir un multiplicador de Baugh and Wooley que multiplique dos números de 5 bits.

Probarlo multiplicando $a*b$, siendo $a = 11001$ y $b = 01110$.

8.- Siguiendo la estructura de un multiplicador recodificado combinacional, multiplicar $a \cdot b$.

$a = 1001$

$b = 10011$

9.- En un divisor secuencial sin restauración dividir A/B .

$A = 10001111$

$B = 1010$

10.- El algoritmo de división secuencial sin restauración está pensado en realidad para números fraccionarios. Buscar algún caso en el que la división de enteros no funcione y tratar de establecer cuáles son las condiciones que deben cumplir el dividendo y el divisor para que funcione correctamente.

11.- Dividir a/b en un divisor combinacional sin restauración.

$A = 1, 110011$

$B = 0, 101$

12.- Utilizando los dos métodos de división por convergencia calcular a/b .

$a = 0, 1010011$

$b = 0, 1110111$

con ocho bits de precisión en todos los registros usados.

13.- Calcular el coseno de 35 utilizando el método CORDIC.

14.- Buscar una fórmula que me indique el retardo de un multiplicador combinacional recodificado en función del número de bits de los operandos.