AMPLIACIÓN DE ESTRUCTURA DE COMPUTADORES
21 DE JUNIO DE 2002
Problemas

1.- El siguiente fragmento de código se ejecuta en un MIPS segmentado:
add $1, $2, $3
sub $4, $5, $6
and $5, $3, $1
sw $4, 10($6)
lw $4, 12($7)
beq $7, $4, 1
and $10, $7, $13
or $14, $15, $16
Suponiendo que un dato se puede escribir en el banco de registros y leer su nuevo valor en el mismo ciclo. Mostrar sobre el esquema de la figura el momento de ejecución de cada una de las fases de las instrucciones,
(1,5 puntos)

a) Suponiendo que el salto no se realiza.

Add	IF	ID	EX	M	WB														
Sub																			
And																			
Sw																			
Lw																			
Beq																			
And																			
Or																			

b) Suponiendo que el salto sí se realiza.

Add	IF	ID	EX	M	WB														
Sub																			
And																			
Sw																			
Lw																			
Beq																			
Or																			

c) Tratar de reordenar el código para minimizar el número de ciclos de ejecución.

2.- a) Diseñar un multiplicador combinacional recodificado que siga el algoritmo de Booth, como el visto en clase, para multiplicar números de 5 bits.
b) Indicar cuáles son las ecuaciones que definen los dos tipos de celdas utilizados.
c) Multiplicar 10100*11010
(2 puntos)

3.- Supongamos un formato IEEE-754 reducido, con 12 bits, de los cuales 5 son de exponente, determinar:
a) ¿Cuál es el número, en decimal, más próximo a cero y positivo representable?. (Puede dejarse indicado como potencias de 2.)
b) ¿Cuál es el mayor número representable?
c) Realizar a-b, con los pasos indicados en clase, si:
a= 001011001100
b= 001011010000
d) Realizar a*b con los operandos anteriores, siguiendo el algoritmo visto en clase.
(2 puntos)

AMPLIACIÓN DE ESTRUCTURA DE COMPUTADORES

21 DE JUNIO DE 2002
Teoría

1.- Indicar si las siguientes afirmaciones son verdaderas o falsas, indicando el porqué:

a) Los multiplicadores de Waugh-Booley son más rápidos que los de Pezaris dado que utilizan sumadores más simples.
b) Para sumar números creados uniendo sumadores carry-save en árboles de Wallace, el tiempo de cálculo no depende del tamaño de los números ya que, al calcular el carry y la suma por separado, cada etapa necesita sólo dos niveles lógicos.
c) Para calcular un seno mediante el método CORDIC, se itera sumando o restando ángulos cuya tangente es 2^{-i}, hasta que se llega al ángulo deseado. Por tanto el número de iteraciones requerido dependerá del ángulo deseado.
d) El resultado de una resta en punto flotante es: 0,111000. Y g= 0, r= 1, s= 0. Por tanto el resultado una vez redondeado, por cualquiera de los métodos excepto el método hacia + infinito, es: 1,110000.
e) El estándar del IEEE nº 754, indica que el resultado de cualquier operación en Punto Flotante que siga este estándar dará el mismo resultado que si se hiciera con precisión total y después se redondeara, esto implica que para hacer la multiplicación de dos números con **n** bits de mantisa, se debe disponer de un registro intermedio para almacenar el resultado con **2n-1** bits, y se redondea a partir del contenido de este registro.

(2 puntos en total)

2.- Indicar brevemente qué son los tiempos de setup, hold, clk_to_Q, y skew.
(1 punto)

3.- Diseñar un sumador de 15 bits, a partir de módulos sumadores de 3 bits, construidos mediante anticipación de arrastres, e interconectados mediante puenteo de arrastres.
Indicar en qué instante, medido en etapas lógicas atravesadas, se conoce cada uno de los bits de suma. Indicar claramente, o en el dibujo o mediante las ecuaciones lógicas, de dónde provienen los retardos indicados.
(1,5 puntos)