

Ingeniería Superior de Informática. Curso 3º.
Ampliación de Estructura de Computadores. Examen Final. TEORIA.
13 de Septiembre de 2006

Nombre: _____

DNI: _____

- P1. (a) Definir los tiempos de propagación (*Clk-to-Q*), *setup*, *hold* y sesgo de reloj (*clock-skew*).
(b) Justifica la desigualdad $Clk-toQ + \text{camino con retardo mínimo} - \text{clock-skew} > hold$

R1.

(a)

Propagación:

Setup:

Hold:

Sesgo de reloj:

(b)

- P2. Dibujar un diagrama temporal que muestre cómo se ejecuta el siguiente código en un procesador MIPS suponiendo que son aplicables todas las anticipaciones posibles, las paradas (*stall*) “lw” cuando sean necesarias y el retardo de salto de un ciclo. Usar flechas para mostrar la anticipación de datos entre instrucciones

```
1.      lw      $1, 100($2)
2.      addi $3, $1, 1
3.      beq     $0, $0, label
4.      lui     $6, 12
5.      add     $1, $2, $3
6. label: lw     $2, 400($1)
7.      subi   $6,$3, 200
8.      lw     $12, 64($6)
9.      sw     $12, 20($5)
```

R2.

- P3. Un sumador con detección de acarreo finalizado es un sumador que genera una señal indicando, de manera asíncrona, que los bits de suma **ya están estables**. La unidad básica es un sumador con las mismas entradas x_i e y_i y salida s_i , pero con los acarreos modificados. En vez de un acarreo de entrada y otro de salida hay dos de cada. El acarreo $c_{i+1}^0 = 1$ indica que **ya se sabe** que el acarreo de salida es 0, por ejemplo si x_i e y_i son los dos 0, c_{i+1}^0 es inmediatamente 1 ya que no puede haber acarreo de salida. Por su parte, el acarreo $c_{i+1}^1 = 1$ indica que **ya se sabe** que el acarreo de salida es 1.
- a) Escribir las ecuaciones de las tres salidas c_{i+1}^0 , c_{i+1}^1 y s_i en función de las entradas.
- b) Si definimos una señal para cada etapa $CC_i = c_{i+1}^0 + c_{i+1}^1$, ¿cuál serían las ecuaciones de la señal CC asíncrona que indica que la suma ya se ha acabado?
- c) Sumar 00110 y 00010 con este sumador y comparar su retardo con el del sumador con propagación de arrastres tradicional.

R3

***Ingeniería Superior de Informática. Curso 3º.
Ampliación de Estructura de Computadores. Examen Final. TEORIA.
13 de Septiembre de 2006***

Nombre: _____

DNI: _____

- P4. (a) ¿En qué se basa el algoritmo de división binaria sin restauración (DBSR)? ¿Qué ventajas tiene respecto al algoritmo básico de división con restauración (DBCR)?
- (b) ¿En qué situaciones el algoritmo DBSR requiere un paso adicional final de corrección del resto y en qué consiste dicho paso?
- (c) Divide 011000 entre 110 (ambos enteros sin signo) mediante DBSR indicando claramente las operaciones de cada paso.

R4.

P5. (a) Multiplicar los números siguientes representados en coma flotante IEEE, usando el algoritmo de multiplicación de coma flotante.

$$3fe00000_{16} * 40400000_{16}$$

(b) Convertir el resultado a formato decimal

R5.

Ingeniería Superior de Informática. Curso 3º.
Ampliación de Estructura de Computadores. Examen Final. PROBLEMAS.
13 de Septiembre de 2006

Nombre: _____

DNI: _____

P1.- Considerar la ruta de datos MIPS multi-ciclo estudiada durante el curso. Queremos incorporar al repertorio básico una nueva instrucción, la instrucción TSWAP (test-and-swap) que utiliza un formato inmediato y que en ensamblador escribimos así:

TSWAP R2, R1, *offset*

El comportamiento de la instrucción es el siguiente: si el valor de R1 es cero se produce una bifurcación relativa al CP con desplazamiento *offset* y si el valor de R1 es distinto de cero se intercambian los valores de los registros R1 y R2

- a) Modificar la ruta de datos si es preciso y rediseñar la unidad de control para que contemple esta nueva instrucción.
- b) Mostrar el diagrama de estados optimizado.

P2.- (a) Dibujar una ruta de datos que permita realizar la multiplicación de dos números representados en C2 utilizando el algoritmo de Booth de raíz-4 (re-codificación por pares de bits)

(b) Realizar paso a paso la multiplicación de $A * B$ usando esa ruta de datos

$A = 01011010$; $B = 01011010$

(c) Diseñar una celda de control y una celda CASS para aplicar el método anterior (algoritmo Booth raíz-4) a un multiplicador en array

Nota.- Las funciones que deben diseñarse son

Control: $F(a_{i-1}, a_i, a_{i+1}) = (s/r, x1, x2)$

CASS: $G(s/r, x1, x2, b_i, b_{i-1}, pp0, c0) = (s/r, x1, x2, b_i, b_{i-1}, pp1, c1)$