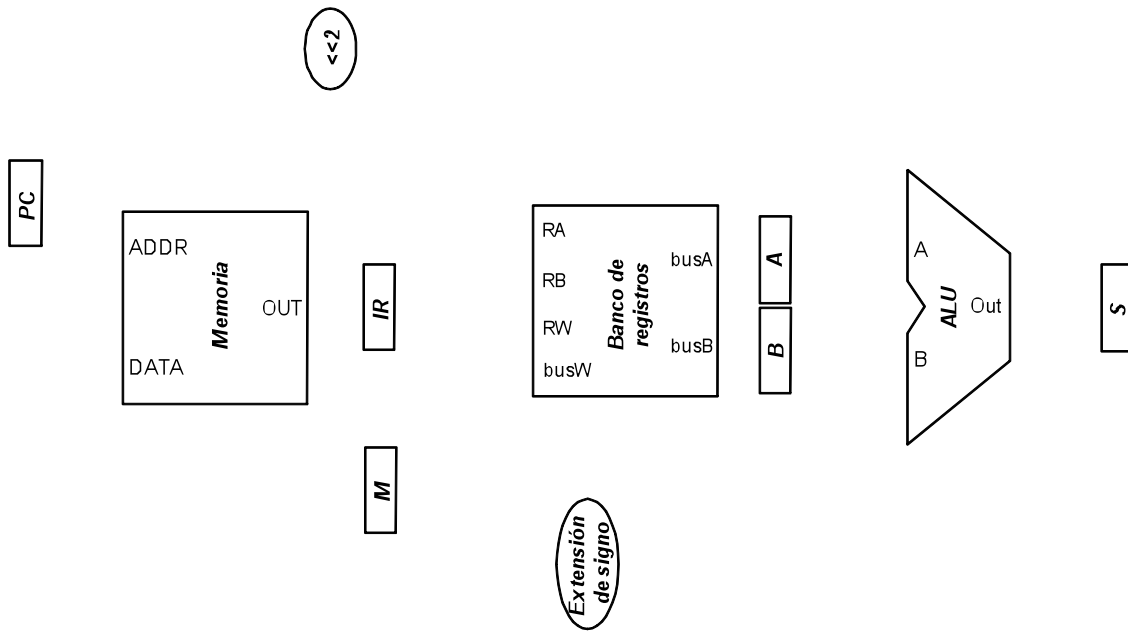


Ingeniería Superior de Informática. Curso 3º
Ampliación de Estructura de Computadores
Examen Final. 27 de Junio de 2008

Apellidos, Nombre: _____ DNI: _____

Problema 1. (2 puntos) Sea la ruta de datos multiciclo de la figura. Todas las entradas a las unidades funcionales están etiquetadas. Los registros intermedios sólo tienen una entrada y una salida (la señal de reloj se considera implícita). Este problema no trata de la unidad de control, por lo que las señales de control no se muestran en ningún caso.



Se pide:

- a) Completar la tabla para describir a nivel de transferencia de registros las instrucciones MIPS que se dan en la tabla suponiendo que estas se ejecutan en la ruta de datos dada.

jal dir: $S31 \leftarrow PC+4$, $PC \leftarrow PC[31-28] \cup 4 \times \text{dir}$

add	lw	sw	jal
$IR \leftarrow \text{MEM}[PC]$	$IR \leftarrow \text{MEM}[PC]$	$IR \leftarrow \text{MEM}[PC]$	$IR \leftarrow \text{MEM}[PC]$

- b) Algunos componentes deben reutilizarse durante la ejecución de alguna instrucción. Cablear la ruta de datos para que soporte las cuatro instrucciones, añadiendo sólo multiplexores cuando se necesiten. Etiquetar adecuadamente buses especiales (p.e. rs, rt, PC[31-28], etc.). Recuerda que no se deben incluir señales de control en los multiplexores, ya que se supone que estas se generan correctamente.
- c) Anotar en la tabla, al lado de la instrucción, el CPI de la misma, indicando al lado de cada transferencia con qué ciclo se corresponde.
- d) Suponiendo los siguientes retardos: memoria 50ns, registros (lectura) 25ns, registros (escritura) 15ns, ALU 30ns y que el resto de los retardos se consideran nulos, calcular el tiempo de ejecución de este procesador para un programa que consta de 400000 sumas, 250000 cargas, 250000 almacenamientos, y 100000 saltos.

Problema 2. (2 puntos) Considere el siguiente fragmento de código ejecutado en la ruta de datos segmentada. El valor inicial de \$4 es 396:

```
Loop: lw $1, 0($2)
      addi $1, $1, 1; Suma valor inmediato
      sw $1, 0($2)
      addi $2, $2, 4
      addi $4, $4, -4
      bnez $4, Loop; Salta si no es cero
```

Se supone que no hay anticipación de operandos ni reordenación de código, que un dato se puede escribir en el banco de registros y leer su nuevo valor en el mismo ciclo, y que el destino del salto se resuelve en la etapa de memoria. Se pide:

- a) Escriba en la tabla el momento de ejecución de cada una de las fases que se ejecutan en una iteración del bucle, incluyendo la carga de la segunda iteración.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
lw \$1,0(\$2)	F	D	X	M	W																	
addi \$1,\$1,1		F																				
sw \$1,0(\$2)																						
addi \$2,\$2,4																						
addi \$4,\$4,-4																						
bnez \$4,Loop																						
lw \$1,0(\$2)																						

- b) ¿En cuántos ciclos se ejecuta el bloque de código (todas las iteraciones, no sólo una)?

Problema 3. (2 puntos) Construir un sumador de 32 bits conectando con propagación de arrastres dos sumadores de 16 bits. De estos el que calcula los 16 bits menos significativos se diseña con sumadores de anticipación de arrastres de 4 bits conectados a su vez con anticipación de arrastres. El que calcula los 16 más significativos se diseña con sumadores de cuatro bits conectados con puenteo de arrastres. De estos sumadores de 4 bits el primero está diseñado con anticipación de arrastres y los otros tres con propagación.

- a) Dibujar el sumador
b) Explicar cuál es el retardo indicando claramente cómo se obtiene, mostrando en la ruta de datos los retardos intermedios.

Problema 4. (1 punto) Algoritmo de CORDIC

- a) Calcula por CORDIC la aproximación al $\sin(10^\circ)$ y al $\cos(10^\circ)$.
b) ¿Crees que la precisión obtenida es buena? Justifica tu respuesta.

- Tabla de arco-tangentes de que disponemos:

Arco	Tangente
45	2^0
26.56	2^{-1}

- $x_{i+1} = x_i \pm y_i \times 2^{-i}$ (Positivo para rotación sentido horario)
- $y_{i+1} = y_i \pm x_i \times 2^{-i}$ (Positivo para rotación sentido antihorario)
- Vector inicial: $z = (0.10011011, 0)$

Problema 5. (3 puntos) Aritmética en punto flotante.

- a) Construir una ruta de datos (especificando la anchura de todas y cada una de las interconexiones) que permita hacer una multiplicación de números en punto flotante representados en el estándar IEEE-754. Los números ocuparán 8 bits de los cuales 4 son de exponente.
b) Indicar claramente cuántos registros se necesitarán y de qué tamaño.
c) Diseñar los sumadores que se necesiten mediante Anticipación de arrastres, indicando las ecuaciones lógicas de TODOS los bits de salida.
d) Diseñar los multiplicadores que se necesiten con multiplicadores secuenciales lo más sencillos posibles.
e) ¿Se necesitarán bits de guarda, redondeo y sticky? Explicar dónde y cuándo.