

**Ingeniería Superior de Informática. Curso 3º.**  
**Ampliación de Estructura de Computadores.**  
**Examen Final. TEORIA.**  
**17 de Junio de 2005**

1. Diseñar un sumador de 1 bit y un restador de 1 bit. ¿Cuál es la diferencia entre ambos? ¿Cómo se define cada tipo de celda 0, 1, 2 y 3 usadas en el multiplicador de Pezaris, como sumadores o como restadores?

R1.

2. La multiplicación secuencial se realiza inspeccionando los bits del multiplicador de derecha a izquierda (de menos a más significativo). ¿Qué desventaja tendría hacerlo al contrario – de izquierda a derecha?  
Aplicar el segundo método al ejemplo  $1011 \times 0111$  tanto en multiplicación con signo como en multiplicación en C'2.

R2.

3. ¿Qué inconvenientes presenta admitir el tratamiento de números des-normalizados en las operaciones de multiplicación/división en coma flotante?

R3.

**Ingeniería Superior de Informática. Curso 3º.**  
**Ampliación de Estructura de Computadores.**  
**Examen Final. TEORIA.**  
**17 de Junio de 2005**

4. Supongamos que los dos fragmentos de código ensamblador siguientes han sido generados por sendos compiladores (C1 y C2) para un mismo código fuente.

Compilador 1	Compilador 2
<pre> add \$t0, \$zero, \$zero add \$t1, \$a1, \$zero loop: bge \$t1, \$a2, retorno sll \$t2, \$t1, 2 add \$t2, \$t2, \$a0 lw \$t3, 0(\$t2) add \$t0, \$t0, \$t3 add \$t1, \$t1, \$a3 j loop retorno: add \$v0, \$t0, \$zero jal \$ra </pre>	<pre> add \$v0, \$zero, \$zero sll \$a2, \$a2, 2 sll \$a3, \$a3, 2 add \$t1, \$a1, \$zero bge \$t1, \$a2, retorno loop: add \$t2, \$t1, \$a0 lw \$t3, 0(\$t2) add \$v0, \$v0, \$t3 add \$t1, \$t1, \$a3 blt \$t1, \$a2, loop retorno: jal \$ra </pre>

- a. Inicialmente \$a1=0, \$a2=5, \$a3=1. Determinar la frecuencia de cada tipo de instrucción rellenando la tabla siguiente:

Tipo de instrucción	CPI	Frecuencia en C1	Frecuencia en C2
ALU Aritmética			
ALU lógica	3		
Memoria ("load")			
Memoria ("sw")			
Bifurc. Cond. SI			
Bifurc. Cond. NO			
Bifurc. Incond.			
Total			

(Suponemos que la instrucción "sll" – shift logical left – se ejecuta en 3 ciclos. El registro \$zero se corresponde con \$0 y siempre vale 0)

- b. Calcular el CPI de cada programa usando las frecuencias obtenidas en a)  
c. Si P1 se ejecuta en una máquina de 500 MHz y P2 en una de 400 MHz ¿qué combinación h/w-s/w es más rápida? ¿Cuál es la ganancia?

R4.

**Ingeniería Superior de Informática. Curso 3º.**  
**Ampliación de Estructura de Computadores.**  
**Examen Final. TEORIA.**  
**17 de Junio de 2005**

5. El formato básico de las microinstrucciones puede permitir combinaciones que la ruta de datos no pueda admitir y otras que no se usan en el repertorio.
- Pon un ejemplo de cada caso para el formato horizontal del MIPS visto en clase.
  - Pon un ejemplo de cada caso para el formato vertical del MIPS visto en clase.
  - La posibilidad de aparición de estas codificaciones no deseables, ¿están relacionadas con el tipo de formato elegido? Justifica la respuesta.

R5.

6. El desenrollado de bucles es una técnica que permite elevar la velocidad de ejecución de códigos iterativos. Para el siguiente ejemplo en una arquitectura pipeline,

Código de bucle original	Código de bucle desenrollado una vez
<pre>bucle: lw \$t0, 0(\$s1)       add \$t0, \$t0, \$s2       sw \$t0, 0(\$s1)       addi \$s1, \$s1, -4       bne \$s1, \$zero, bucle</pre>	<pre>bucle: lw \$t0, 0(\$s1)       add \$t0, \$t0, \$s2       sw \$t0, 0(\$s1)       lw \$t1, -4(\$s1)       add \$t1, \$t0, \$s2       sw \$t1, -4(\$s1)       addi \$s1, \$s1, -8       bne \$s1, \$zero, bucle</pre>

comparar la diferencia de rendimiento, mediante contabilización de ciclos de ejecución, para los tres casos siguientes:

- bucle original
  - bucle desenrollado
  - bucle desenrollado y reordenado para evitar bloqueos si es posible
- (Suponer que la instrucción bne toma la decisión de salto en la etapa ID, y se dispone de circuitería de anticipación para resolver dependencias de datos).

R6.

**Ingeniería Superior de Informática. Curso 3º.**  
**Ampliación de Estructura de Computadores.**  
**Examen Final. PROBLEMAS.**  
**17 de Junio de 2005**

Nombre: \_\_\_\_\_

DNI: \_\_\_\_\_

1. 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)  
 Realizar paso a paso la multiplicación de  $A * B$  usando esa ruta de datos

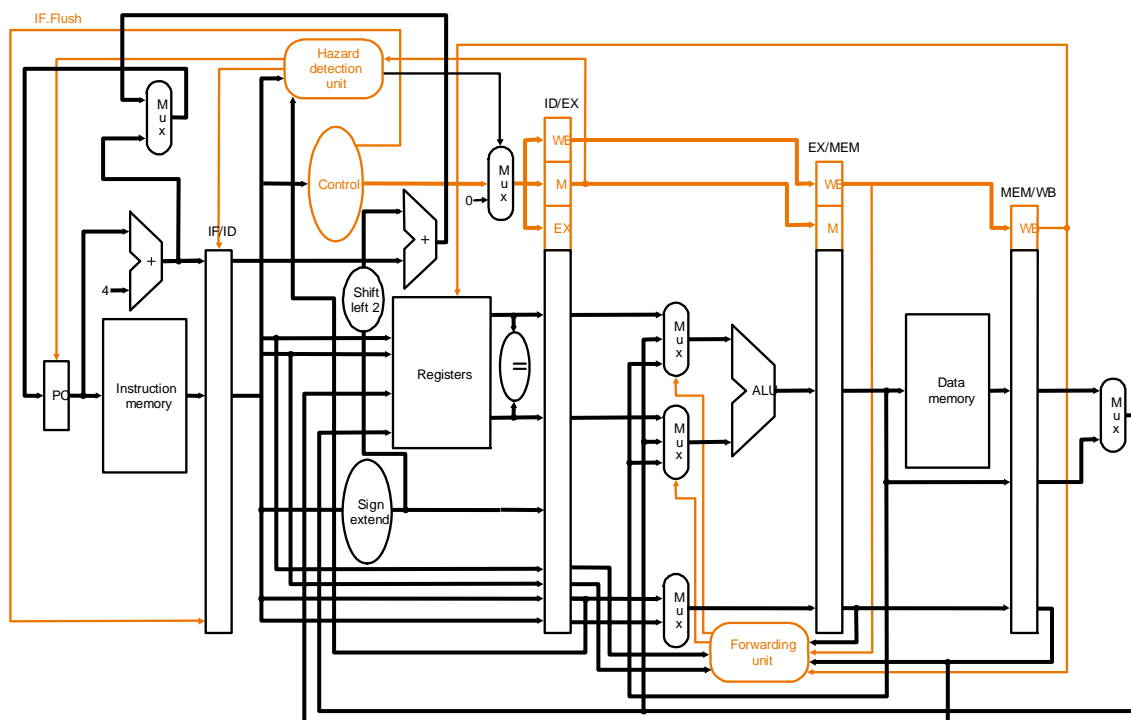
$A = 01011010$ ;  $B = 01011010$

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

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

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

2. Considerar la ruta de datos segmentada de la figura.



- a). En esta ruta de datos ¿puede ocurrir simultáneamente un intento de anulación de instrucción (*flush*) debido a un salto y un intento de parada (bloqueo o *stall*) debido a una dependencia de datos que afecta a una instrucción LW? Mostrar una secuencia de código donde se manifieste la situación descrita.
- b). Mostrar cuál es el efecto del bloqueo y de la anulación sobre el *pipeline*, indicando cómo se detectan y a qué señales y registros afectan.
- c). Las acciones realizadas en el bloqueo y anulación simultáneos, ¿colaboran o tienen efectos que entran en conflicto? Si hay colaboración, ¿cómo trabajarían juntos? Si hay conflicto, ¿qué acción debe tener prioridad?