



**ESTRUCTURA DE COMPUTADORES**  
**Examen Final - Junio de 2018**

Nombre \_\_\_\_\_ DNI \_\_\_\_\_  
Apellidos \_\_\_\_\_ Grupo \_\_\_\_\_

1.- (4 puntos) Sea un procesador segmentado de cinco etapas con las siguientes características:

- Un dato se puede leer y escribir en el banco de registros en el mismo ciclo de reloj.
- Existe anticipación de operandos. El cortocircuito se realiza siempre desde los registros del pipeline, y existe cortocircuito a la etapa ID para las instrucciones de salto, que se resuelven en dicha etapa.
- La detección de riesgos LDE y generación de paradas se realiza en la etapa de decodificación.
- Los riesgos estructurales referidos a memoria se detectan y se resuelven mediante espera en la última etapa de cada unidad funcional.
- Los riesgos de EDE entre dos instrucciones A y B tal que A precede a B se resuelven mediante inhibición de escritura de la instrucción A.
- No se implementan saltos retardados, sino que se busca siempre la siguiente instrucción al salto y se cancela su ejecución si el salto es tomado.
- Las unidades funcionales de las que dispone el procesador son las siguientes:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	3	Sí
FP MUL	1	5	Sí
Int ALU	1	1	No

El siguiente código calcula los elementos del vector `double C[128]` a partir de los elementos de los vectores `double A[128]` y `double B[128]` (recuerda que un elemento en punto flotante de doble precisión ocupa 8 bytes). Los vectores se encuentran almacenados en memoria de forma consecutiva (A-B-C) y la dirección de comienzo del vector A es `0x80001000`.

LOOP:

```
LD      F2, 0(R6)
AND     R2, R4, R9
BEQ     R2, R0, ELSE
IF:     ADDD  F6, F2, F2
        MULD  F6, F8, F6
        BEQ   R0, R0, END
ELSE:   LD    F4, 1024(R6)
        MULD  F6, F2, F4
        ADD   R10, R10, R2
        ADDD  F10, F2, F4
END:    SD    F6, 2048(R6)
        ADD   R6, R6, R3
        ADD   R4, R4, R9
        BNE   R4, R5, LOOP
        SUB   R4, R4, R4
```

R6=0x80001000  
R3=8  
R5=128  
R4=0  
R9=1  
R10=0  
R7=-1  
F8=5.5  
R0 =0 siempre en la  
arquitectura MIPS

El contenido inicial de los registros es el siguiente:

- (2 puntos) Representa el diagrama instrucción-tiempo para la primera y segunda iteración e indica los cortocircuitos realizados. Indica claramente las paradas y sus causas.
- (1 punto) Calcula el CPI de la ejecución completa del código.
- (1 punto) Si el procesador utilizara saltos retardados, ¿qué instrucciones llevarías al *delay-slot* de cada uno de los saltos y qué efectos tendría?

2.- (3 puntos) En un sistema con procesador ARM y una memoria de 64KB se quiere analizar el comportamiento del siguiente fragmento de un programa.

```
.global start
.equ N, 12

.data
A: .word N valores separados por comas
B: .word N valores separados por comas

.bss
C: .space 4*N

.text
    ....
    mov R3, #N
    mov R4, #0
    mov R5, #0
    mov R2, #0
for:  cmp R2, R3
     bge fin
     ldr R4, [R0, R2, LSL#2]
     ldr R5, [R1, R2, LSL#2]
     add R4, R4, R5
     str R4, [R7, R2, LSL#2]
     add R2, R2, #1
     b for
fin:  ....
```

El programa se enlaza ubicando la sección .data a partir de la dirección 0x0090 y la sección .bss a continuación de la sección .data, y que el fragmento de la sección .text en la que se encuentra la primera instrucción (mov R3, #N) se ubica en la dirección 0x01400

R0, R1 y R7 contienen las direcciones de A, B y C respectivamente

Responde de forma razonada a las siguientes preguntas:

- (0.5 puntos) Obtener el rango de direcciones que ocupa el fragmento de programa y cada uno de los arrays. ¿Cuántos bloques de memoria principal ocupan? ¿Qué bloques son?
- (1.5 puntos) El sistema dispone de caches separadas de datos (D\$) e instrucciones (I\$) de 32 bytes cada una, ambas con política de emplazamiento directo, y políticas de escritura directa (write through) con asignación en escritura (write allocate), y tamaño de bloque de 16 bytes. Obtener los fallos que se producirán en la cache de datos (D\$) y en la cache de instrucciones (I\$) al ejecutar el fragmento del programa. Indicar, además del número total de fallos, para qué bloques de memoria principal se producen los fallos.
- (1 punto) Ahora añadimos al sistema de memoria una cache de segundo nivel unificada para datos e instrucciones (L2\$) de 128 bytes, asociativa por conjuntos de 2 vías, con reemplazamiento LRU y políticas de escritura directa (write through) con asignación en escritura (write allocate). Indicar el número de fallos que se producen en L2 tras ejecutar el fragmento del programa. (El tamaño de bloque es también de 16 bytes).

3.- (3 puntos) En una empresa dedicada al análisis de datos tienen que utilizar un algoritmo que requiere sumar todos los elementos de cada columna de una matriz de 512x512 elementos. El código C de dicho algoritmo es el siguiente:

```
for (j = 0; j < 512; j++)
    for (i = 0; i < 512; i++)
        red[j] = red[j] + A[i][j];
```

Sabiendo que:

- En C los elementos de una matriz se emplazan en memoria consecutivos por filas
- El procesador es de 32 bits (direcciones virtuales de 32 bits).
- El array de enteros A[512][512] se ubica en el mapa virtual de memoria del proceso a partir de la dirección 0x00458000, y el array de enteros red[512] se ubica justo a continuación de A.
- El sistema operativo (SO) maneja páginas de 4KB con reemplazo LRU.

se pide:

- (0.75 puntos) Indicar qué páginas de memoria virtual ocupan los arrays A y red, indicando qué elementos de cada array están en cada página.
- (1.25 puntos) Si el SO le asigna al proceso 8 marcos de página y sabiendo que dos de ellos están ocupados por la página de código y la página de pila, accedidos constantemente y que por tanto nunca son elegidos para el reemplazo, y que los 6 restantes no contienen inicialmente ninguna página de A ni de red; indicar razonadamente cuantos fallos de página se producirán al ejecutar el código y el número de accesos a A y red que no suponen fallo.
- (1 punto) Razone si es posible o no reducir significativamente los fallos de página del proceso con alguna modificación en su código, indicando en su caso que transformación se haría y cuantificando la mejora esperada.