



## SEGMENTACIÓN

### Obligatorios:

1. Sea un procesador segmentado con cinco etapas como el visto en clase que tiene las siguientes características:
  - Se puede escribir y leer en el banco de registros en el mismo ciclo de reloj.
  - No tiene técnicas para reducir o eliminar paradas en caso de riesgos de datos (NO tiene implementado cortocircuito).
  - Los saltos se resuelven en la etapa ID. Después del salto siempre se busca la instrucción siguiente y en caso de que se produzca el salto se anula la instrucción buscada.

Se ejecuta en dicho procesador una aplicación con las siguientes características:

- El 20 % de las instrucciones son saltos condicionales y el 40 % de los saltos se realizan. No hay dependencias de datos en las instrucciones de salto.
- El 18% de las veces, las instrucciones  $I_{i+1}$  tienen una dependencia de LDE con las instrucciones  $I_i$  (el 30 % de éstas corresponden a instrucciones de load).
- El 6% de las veces las instrucciones  $I_{i+2}$  tienen dependencias de LDE con la instrucción  $I_i$  (el 30 % de éstas corresponden a instrucciones de load), y en esos casos nunca hay dependencias entre  $I_{i+1}$  e  $I_{i+2}$ .

Calcular:

- a) Los ciclos por instrucción de dicho procesador.
- b) Los ciclos por instrucción si al procesador se le añade cortocircuito.
- c) El speed-up del segundo caso frente al primero.

2. Sea un procesador segmentado de cinco etapas con las siguientes características:

- Tiene el HW para la detección de riesgo y generación de la parada en decodificación.
- Se puede escribir y leer el registro en el mismo ciclo de reloj.
- Tiene implementada la técnica de forwarding (cortocircuito).
- Los saltos condicionales se resuelven en decodificación y se solucionan parando el pipeline.

Supongamos que a este procesador se le realiza la siguiente modificación: se unen las etapas de ejecución y memoria. Como consecuencia de esta modificación el ciclo de reloj aumenta un 35%.

Hallar la relación de velocidades entre la versión original del procesador y la modificada cuando se ejecuta una aplicación en la que en el 25% de los casos existen dependencias LDE entre un load y la instrucción que le sigue (esta instrucción nunca es un salto condicional) y que el 33% son saltos condicionales.

```
ADD  R2,R2,R1
SUB  R6,R6,#4
SW   10(R6),R2
BNEZ R6,BUCLE
ADD  R1,R1,#1
SUB  R3,R3,R7
```

Se supone que el valor inicial de R6 es 2000.

- a) Construye el diagrama de tiempo correspondiente a la primera iteración del bucle, indicando sobre el diagrama los cortocircuitos que se activan.
- b) Calcula el valor del CPI
- c) Si el computador trabaja con una frecuencia de 1 GHz, determina el rendimiento en MIPS.

6. Sea un MIPS 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 (tiene implementado cortocircuito).
- 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.
- Las unidades funcionales del procesador son:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	2	sí
FP MUL	1	5	sí
Int ALU	1	1	No

Sabiendo que el siguiente fragmento de código se ejecuta sobre dicho procesador

```
LD      F10, 0(R1)
MULD    F4, F0, F10
LD      F12, 0(R2)
ADD     F2, F12, F4
LD      F4, 8(R1)
MULD    F12, F4, F12
LD      F12, 16(R1)
```

- a) Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas.
- b) Determinar el CPI

7. 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 ejecución.
- 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.

3. El siguiente fragmento de código se ejecuta en un MIPS segmentado de cinco etapas:

```
sub  R1,R2,R3
add  R4,R5,R6
sub  R5,R4,R8
add  R7,R2,R3
add  R9,R7,R3
lw   R1,10(R6)
add  R3,R1,R4
sub  R6,R7,R8
```

Suponiendo que se puede escribir un dato en el banco de registros y leer su nuevo valor en el mismo ciclo, calcular el número de ciclos necesarios para ejecutar este código en los siguientes casos:

- a) No existe la posibilidad de anticipar operandos (NO tiene implementado cortocircuito) ni de reordenar el código.
- b) Existe anticipación de operandos (tiene implementado cortocircuito), pero NO existe la posibilidad de reordenación de código.
- c) No hay anticipación de operandos (NO tiene implementado cortocircuito) pero existe la posibilidad de reordenación de código. Reordenar el código para conseguir que el número de ciclos sea mínimo.

4. Sobre la estructura del computador MIPS segmentado, se ejecuta la siguiente secuencia de instrucciones:

```
ADD R1, R2, R3
SUB R4, R2, R3
AND R5, R2, R3
OR  R6, R2, R3
```

Si la instrucción ADD está colocada en la dirección de memoria 00002000 (Hex), y el contenido de los registros es:

```
R1=00000005
R2=00000004
R3=00000002
PC=00002000
```

- a) Señalar el contenido de los siguientes registros al cabo de 4 ciclos de reloj: IF/ID.pc, ID/EX.pc, EX/MEM.pc, ID/EX.A, ID/EX.B, EX/MEM.ALUout, MEM/WB.ALUout, ID/EX.rd, EX/MEM.rd, MEM/WB.rd.
- b) Indicar también el contenido de cada uno de los registros que almacena el control.

5. Supongamos un computador como el MIPS segmentado en cinco etapas con las siguientes características:

- Posee anticipación de operandos (tiene implementado cortocircuito).
- Utiliza saltos retardados con un "delay-slot" de un ciclo.
- Posee una sola memoria cache para el almacenamiento de instrucciones y datos. Por esta razón no resulta posible leer una instrucción y realizar la lectura o escritura de un dato en el mismo ciclo de reloj (no puede coincidir etapa MEM de un load o store con otra instrucción en etapa IF).
- Las escrituras en el banco de registros se hacen en la primera mitad de la fase WB, mientras que las lecturas se hacen en la segunda mitad de la fase WB.

Supongamos que este computador ejecuta el siguiente programa:

```
BUCLE:  LW   R2,4(R6)
        LW   R3,8(R6)
        SUB  R2,R2,R3
```

- 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	2	No
FP MUL	1	3	No
INT ALU	1	1	No

- a. Completar el diagrama instrucción-tiempo para el siguiente fragmento de código, indicando claramente los cortocircuitos realizados, las paradas y sus causas.

```
LD      F1, 0(R1)
LD      F2, 0(R2)
MULD    F2, F2, F1
MULD    F3, F3, F1
ADD     F4, F2, F1
```

- b. Completar el diagrama instrucción-tiempo para el siguiente fragmento de código, indicando claramente los cortocircuitos realizados, las paradas y sus causas.

```
MULD    F1, F2, F3
ADD     F1, F1, F2
SD      F1, 0(R1)
```

- c. Completar el diagrama instrucción-tiempo para el siguiente fragmento de código, indicando claramente los cortocircuitos realizados, las paradas y sus causas.

```
MULD    F1, F2, F1
MULD    F1, F3, F1
MULD    F5, F4, F2
```

- d. Calcular el CPI de la ejecución completa del siguiente código, siendo R0=0 y el R2=32 al comienzo de la ejecución.

```
loop:  ADDD  F7, F7, F6
      AND   R6, R2, #1
      BEQ   R6, R0, if
      ADDI  R3, R3, #2
      ADDI  R5, R3, R5
      if:  ADDD  F7, F7, F1
      SUBI  R2, R2, #1
      ADDI  R4, R3, R5
      BNE   R0, R2, loop
      SD    F7, 256(R1)
```

8. El siguiente fragmento de código se ejecuta en un MIPS con segmentación de 7 etapas:

```
LOOP: LD      F2, 0(R1)
      MULD    F4, F2, F0
      LD      F6, 0(R2)
      ADDD    F6, F4, F6
      SD      F6, 0(R2)
      ADDI    R1, R1, #8
      SGTI    R3, R1, DONE
      BEQZ    R3, LOOP
      ADDI    R2, R2, #8
```

Suponiendo que:

- Tanto la memoria de datos como la de instrucciones están segmentadas en dos etapas
- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo
- Se dispone de lógica de cortocircuito
- La detección de todo tipo de riesgos (estructurales y LDE) y generación de paradas se realiza en la etapa de decodificación
- Los riesgos EDE se resuelven mediante inhibición de escritura
- Los saltos se resuelven en la etapa de decodificación
- Se emplea un *branchdelay slot* (salto retardado) de una instrucción
- Una inst. aritmética y un store pueden coexistir en las etapas de MEM y WB
- Las unidades funcionales tienen las siguientes características:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	4	sí
FP MUL	1	5	sí
Int ALU	1	1	No

- a) Determinar el CPI teniendo en cuenta que done=0x1000 y r1=0x0100 al inicio de la ejecución.  
b) Reordenar/modificar el código para que el CPI sea mínimo.

9. En un procesador segmentado es necesario ejecutar el siguiente código:

```
for ( n = 0; n < 100; n++ ) {
    for ( k = 0; k < 4; k++ ) {
        out[n] += filter[k] * input[n+k];
    }
}
```

Queriendo conseguir el mayor rendimiento, se decide realizar la siguiente implementación en ensamblador:

```
loop_n: ADDI    R2,R0,#0
loop_k: LD      F3,0(R3)    ; lee de memoria 4 bytes (load de un float)
        LD      F4,0(R4)    ; lee de memoria 4 bytes (load de un float)
        MULS    F6,F3,F4    ; mult. en punto flotante (de dos float)
        LD      F5,0(R5)    ; lee de memoria 4 bytes (load de un float)
        ADDS    F5,F6,F5    ; suma en punto flotante (de dos float)
        SD      0(R5),F5    ; escribe en memoria 4 bytes (store de un float)
```

Opcionales:

10. Supongamos un procesador segmentado de cinco etapas en el que se ejecutan 2 aplicaciones diferentes. La aplicación A tiene un 17% de instrucciones con una penalización de 3 ciclos de reloj y un 12% de instrucciones con una penalización de 2 ciclos de reloj. La aplicación B tiene un 25% de instrucciones con 1 ciclo de penalización.

- a) ¿Cuál es el CPI de cada una de las aplicaciones?  
b) ¿Cuál es el CPI promedio del procesador?  
c) ¿Cuál es el speedup frente al procesador sin segmentar?  
d) ¿Cuál es la eficacia frente al procesador segmentado ideal?

11. Sea un procesador segmentado con UFs multiciclo del cual se sabe que los riesgos de LDE tienen una penalización media de 3 ciclos y que los riesgos de EDE tienen una penalización media de 4 ciclos. Se ejecuta en dicho procesador una aplicación en la que hay un 35% de riesgos LDE y un 8% de riesgos EDE.

- a) ¿Cuál es el CPI del procesador?  
b) Se modifica el procesador de manera que los riesgos EDE se eliminen inhabilitando la escritura de la primera instrucción. ¿Cuál es el nuevo CPI?  
c) ¿Cuál es el speedup obtenido con la modificación?

12. Se tiene el siguiente fragmento de código del MIPS.

```
OR R4, R8, R9
(100 instrucciones con dependencias internas de datos, pero no detención de pipeline)
ADD R5, R6, R7
OR R4, R1, R6
(100 instrucciones con dependencias internas de datos, pero no detención de pipeline)
LW R10, 20(R4)
ADD R8, R9, R10
SUB R6, R8, R1
OR R1, R3, R5
BEQ R1,R2,1000
ADD R1, R5, R6
```

- a) Cuánto tardaría en ejecutarse, si cuando hay un conflicto de control se espera a que se solucione y se supone que el destino de salto se conoce en la fase ID y la comparación de salto también. Suponer que no hay anticipación de operandos.  
b) Si se tuviesen saltos retardados, ¿cómo se podría rellenar el hueco de salto para disminuir el tiempo de ejecución? ¿Cuál sería este tiempo?  
c) ¿Y si además de saltos retardados tuviésemos anticipación de operandos?

El destino del salto es la instrucción OR R4, R1, R6, y el salto se repite 100 veces.

13. Sea un procesador MIPS segmentado con las siguientes características:

- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- Se dispone de lógica de cortocircuito (*forwarding*).
- Los saltos se resuelven en la etapa de decodificación y se cancela la siguiente instrucción si el salto es tomado.
- La detección de todo tipo de riesgos y generación de paradas se realiza en la etapa de decodificación.
- En el caso de los riesgos EDE se produce una parada hasta que la instrucción ya lanzada entre en la etapa de memoria (en ese momento la segunda instrucción saldría de su etapa ID).
- Los registros r1 y r2 contienen inicialmente el valor 4.

```
ADDI    R3,R3,#4    ; actualizo puntero a filter
ADDI    R4,R4,#4    ; actualizo puntero a input
ADDI    R2,R2,#1
BLTI    R2,#4, loop_k    ; salta a loop_k si R2 < 4
ADDI    R5,R5,#4    ; avanzamos el puntero sobre out
SUBI    R3,R3,#16    ; R3 vuelve a apuntar a filter[0]
SUBI    R4,R4,#12    ; R4 apunta al siguiente input[n]
ADDI    R1,R1,#1
BLTI    R1,#100,loop_n    ; salta a loop_n si R1 < 100
```

Se supone que:

- Inicialmente R1=0, R3 apunta a filter[0], R4 apunta a input[0] y R5 apunta a out[0]
- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- Se dispone de lógica de cortocircuito (*forwarding*).
- La detección de todo tipo de riesgos (estructurales y LDE) y generación de paradas se realiza en la etapa de decodificación.
- Los riesgos EDE se resuelven mediante paradas hasta que la instrucción ya lanzada entre en la etapa de memoria.
- Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- Los saltos se resuelven en la etapa de decodificación y NO se implementan saltos retardados (se cancela la siguiente instrucción si el salto es tomado)
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADDS/ SUBS	1	2	Sí
FP MULS	1	3	Sí
INT ALU	1	1	No

- a) Rellena 25 ciclos del diagrama instrucción-tiempo correspondiente al comienzo del bucle interno con k=3 y n=0.  
b) A partir del apartado anterior, estima el número de ciclos que tardará en ejecutarse el código completo y determina el CPI. Indica claramente los cálculos que realices para llegar a los resultados.  
c) En una versión del procesador, si se implementan saltos retardados (*delay-slot*). Adapta (si es necesario) el código anterior para que siga funcionando correctamente. ¿Se obtiene algún beneficio para este código? Justifica tu respuesta.

- Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	2	No
FP MUL	1	3	No
FP DIV	1	5	No
Int ALU	1	1	No

En este procesador se ejecuta el siguiente fragmento de código:

```
L0:  ADDD    F2,F4,F0
      SD      F2,0(r1)
      DIVD    F4,F4,F0
      ADDD    F4,F0,F2
      MULD    F2,F2,F4
      LD      F4,0(r1)
      ADDD    F0,F2,F4
      ADDD    F8,F6,F8
      SUBI    r2,r2,#1
      ADDI    r1,r1,#1
      BNEZ    r2, L0
      ADDD    F0,F8,F2
      ADDD    F2,F0,F8
      end
```

- a) Representar el diagrama instrucción-tiempo e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas.  
b) Calcular los CPI (ciclos por instrucción) resultantes de la ejecución del código.

14. Sea un MIPS segmentado con las siguientes características:

- Un dato se puede escribir en el banco de registros y leer su valor en el mismo ciclo.
- Se dispone de lógica de cortocircuito (*forwarding*).
- Los saltos se resuelven en la etapa de decodificación.
- La detección de todo tipo de riesgos (estructurales y LDE) y generación de paradas se realiza en la etapa de decodificación.
- Los riesgos EDE se resuelven mediante inhibición de escritura.
- Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADDD	1	3	Sí
FP SUBD	1	3	Sí
FP MULD	1	4	Sí
FP DIVD	1	5	No
INT ALU	1	1	No

En este procesador se ejecuta el siguiente fragmento de código

```
ADDI r3,r0,#3
L1: SUBD F2,F6,F8
```

```
SUBD F4,F8,F6
SD 0(r3),F4
DIVD F2,F4,F8
ADDD F2,F8,F8
SUBI r3,r3,#1
DIVD F6,F4,F8
MULD F4,F2,F6
SUBD F10,F2,F6
LD F4,0(r3)
ADDD F0,F4,F2
BNEZ r3,L1
MULD F4,F2,F2
end
```

- a) Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas.
- b) A la vista del diagrama obtenido, y sin necesidad de desarrollar todas las iteraciones, calcular el número de ciclos que toma la ejecución completa del código.

15. El siguiente fragmento de código se ejecuta en un MIPS con segmentación:

```
LOOP: LD      F2, 0(R1)
      MULD    F4, F2, F0
      LD      F6, 0(R2)
      ADDD    F6, F4, F6
      SD      0(R2), F6
      ADDI    R1, R1, 8
      ADDI    R2, R2, 8
      SGTI    R3, R1, DONE
      BEQZ    R3, LOOP
      ADD     R1, R4, #5
```

Suponiendo que:

- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo
- Se dispone de lógica de cortocircuito
- Permite la coexistencia del store y de las operaciones aritméticas en las etapas de M y W
- Los saltos se resuelven en la etapa de decodificación
- Se emplea un *branchdelay slot* (salto retardado) de una instrucción
- Las unidades funcionales tienen las siguientes características:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	4	No
FP MUL	1	5	No
Int ALU	1	1	No

- a) Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas.
- b) Determinar el CPI teniendo en cuenta que done=0x1000 y r1=0x0100 al inicio de la ejecución.

16. En un MIPS con segmentación ejecutamos el siguiente fragmento de código:

```
LOOP :   DIVD F0,F4,F2
         ADDD F0,F2,F6
         DIVD F8,F8,F2
```

```
ADDI R3,R3,#1
ADDD F2,F6,F8
MULD F6,F8,F0
LD F2, 0(R3)
SD 0(R5), F6
MULD F2,F6,F8
ADDD F6,F8,F0
ADDD F0, F2, F2
SUBI R5,R5,#1
BNEZ R5,LOOP
ADDD F4,F2,F2
SUBD F6,F0,F0
END : SUBI R3, R3, #1
```

Se supone que:

- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- Se dispone de lógica de cortocircuito (*forwarding*).
- Los saltos se resuelven en la etapa EX y se espera a que se resuelvan antes de lanzar la siguiente instrucción.
- La detección de todo tipo de riesgos (estructurales y LDE) y generación de paradas se realiza en la etapa de decodificación.
- Los riesgos EDE se resuelven mediante inhibición de escritura.
- Inicialmente r5=1000.
- Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADDD	1	2	Si
FP SUBD	1	2	Si
FP MULD	1	3	Si
FP DIVD	1	4	No
INT ALU	1	1	No

Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados así como las paradas producidas y sus causas. A la vista del diagrama obtenido, indicar el número de ciclos por instrucción (CPI) en régimen estacionario.

17. Sea un MIPS segmentado con las siguientes características:

- Un dato se puede escribir y leer de un registro en el mismo ciclo de reloj.
- Los saltos se resuelven en la etapa de decodificación
- Las detecciones de riesgos de datos y generación de paradas se realiza en la etapa de decodificación.
- La detección de riesgos EDE se realiza en la etapa ID y se produce una parada hasta que la instrucción ya lanzada entre a la etapa de memoria.
- Forwarding combinacional
- Salto retardado de 1 instrucción

UF	CANTIDAD	LATENCIA	SEGMENTACIÓN
FPADD	1	2	NO
FP DIV	1	10	NO
FP MUL	1	5	NO

INTALU	1	1	NO
--------	---	---	----

Y el siguiente fragmento de código:

```
LOOP: LD F6, 0(R2)
      MULD F8,F6,F0
      ADDI R2,R2,#1
      LD F2, 0(R2)
      DIVD F8,F2,F8
      SD 0(R2), F8
      MULD F8,F2,F0
      ADD F8,F4,F6
      ADDI R3,R3, #8
      SGTI R4,R3,DONE
      BNEZ R4,LOOP
      ADD R1,R2, #5
```

- a) CPI del pipeline, suponiendo que el bucle se ejecuta muchas veces.
- b) Suponiendo que el hardware de la máquina puede inhibir la escritura en el registro destino de una instrucción ya lanzada. Explica cómo usar esta característica para reducir las penalizaciones debidas a los riesgos EDE. Calcula el nuevo valor de CPI.