



PROBLEMAS DE ESTRUCTURA DE COMPUTADORES

SEGMENTACIÓN

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 EX. Después del salto siempre se busca las dos siguientes instrucciones y en caso de que se produzca el salto se anulan las instrucciones buscadas.

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) El CPI 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. El siguiente fragmento de código se ejecuta en un RISC-V segmentado de cinco etapas:

```
sub  x1, x2, x3
add  x4, x5, x6
sub  x5, x4, x8
add  x7, x2, x3
add  x9, x7, x3
lw   x1, 10(x6)
add  x3, x1, x4
sub  x6, x7, x8
```

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.

0x

3. Sobre la estructura del computador RISC-V segmentado, se ejecuta la siguiente secuencia de instrucciones:

```
add    x1, x2, x3
sub     x4, x2, x6
andi    x5, x4, 3
lw      x6, 4(x2)
add     x5, x2, x4
```

Si la instrucción add está colocada en la dirección de memoria 0x0000_2000 y el contenido de los registros es:

```
x1=0x0000_0005
x2=0x0000_0004
x3=0x0000_0002
x4=0x0000_0007
x6=0x0000_0006
PC=0x0000_2000
```

Conociendo que:

ALUControl2:0	Function
000	add
001	subtract
010	and
011	or
101	SLT

ImmSrc1:0	Tipo instrucción
00	I-Type
01	S-Type
10	B-Type
11	J-Type

a) Rellenar las siguientes tablas con los valores de las señales y registros justo antes de producirse el quinto flanco de carga del reloj:

RegWriteD		RegWriteE		RegWriteM		RegWriteW	
ResultSrcD		ResultSrcE		ResultSrcM		ResultSrcW	
MemWriteD		MemWriteE		MemWriteM			
JumpD		JumpE					
BranchD		BranchE					
ALUSrcD		ALUSrcE					
ImmSrcD		ALUControlE					
ALUControlD		ForwardAE					
		ForwardBE					

PCPlus4F		RD1E	
PCPlus4D		RD2E	
PCPlus4E		SrcAE	
PCPlus4M		SrcBE	
PCPlus4W		ALUResultM	

4. Hacer lo mismo que en el ejercicio anterior para el siguiente código:

```
add    x1, x2, x3
lw     x6, 4(x2)
sub     x4, x2, x6
andi   x5, x2, 3
add     x5, x2, x4
```

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

- Posee anticipación de operandos (tiene implementado cortocircuito).
- 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, en ese caso la instrucción que entraría en IF se queda en parada en esa etapa hasta que se libere la memoria).
- Se puede escribir y leer en el banco de registros en el mismo ciclo de reloj.

Supongamos que este computador ejecuta el siguiente programa:

```
BUCLE:  lw      x2, 4(x6)
        lw      x3, 8(x6)
        sub     x2, x2, x3
        add     x2, x2, x1
        addi    x6, x6, -4
        sw      x2, 10(x6)
        bne     x6, x0, BUCLE
        addi    x1, x1, 1
        sub     x3, x3, x7
```

Se supone que el valor inicial de x6 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 tiempo de ejecución.

6. Sea un RISC-V segmentado 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).
- 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	Nº de etapas	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

```
flw      f10, 0(x1)
fmul.s   f4, f0, f10
flw      f12, 0(x2)
fadd.s   f2, f12, f4
flw      f4, 8(x1)
fmul.s   f12, f4, f12
flw      f12, 16(x1)
```

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 RISC-V segmentado 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.
 - 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 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.

```
flw    f1, 0(x1)
flw    f2, 0(x2)
fmul.s f2, f2, f1
fmul.s f3, f3, f1
fadd.s 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.

```
fmul.s f1, f2, f3
fadd.s f1, f1, f2
fsw    f1, 0(x1)
```

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

```
fmul.s f1, f2, f1
fmul.s f1, f3, f1
fmul.s f5, f4, f2
```

- d. Calcular el CPI de la ejecución completa del siguiente código, siendo x2=32 al comienzo de la ejecución.

```
loop: fadd.s f7, f7, f6
      andi  x6, x2, 1
      beq   x6, x0, if
      addi  x3, x3, 2
      add   x5, x3, x5
if:    fadd.s f7, f7, f1
      addi  x2, x2, -1
      add   x4, x3, x5
      bne   x0, x2, loop
      fsw   f7, 256(x1)
```

8. 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:

```
addi    x12, x0, 4  
addi    x13, x0, 100  
addi    x1, x0, 0  
loop_n : addi    x2, x0, 0  
loop_k : flw     f3, 0(x3)          ; lee de memoria 4 bytes (load de un float)  
        flw     f4, 0(x4)          ; lee de memoria 4 bytes (load de un float)  
        fmul.s  f6, f3, f4         ; mult. en punto flotante (de dos float)  
        flw     f5, 0(x5)          ; lee de memoria 4 bytes (load de un float)  
        fadd.s  f5, f6, f5         ; suma en punto flotante (de dos float)  
        fsw     f5, 0(x5)          ; escribe en memoria 4 bytes (store de un float)  
        addi    x3, x3, 4          ; actualizo puntero a filter  
        addi    x4, x4, 4          ; actualizo puntero a input  
        addi    x2, x2, 1  
        blt     x2, x12, loop_k    ; salta a loop_k si x2 < 4  
        addi    x5, x5, 4          ; avanzamos el puntero sobre out  
        addi    x3, x3, -16        ; x3 vuelve a apuntar a filter[0]  
        addi    x4, x4, -12        ; x4 apunta al siguiente input[n]  
        addi    x1, x1, 1  
        blt     x1, x13, loop_n    ; salta a loop_n si r1 < 100  
        addi    x0, x0, 0  
        addi    x0, x0, 0
```

Se supone que:

- Inicialmente x3 apunta a filter[0], x4 apunta a input[0] y x5 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.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADD/ SUB	1	2	Sí
FP MUL	1	3	Sí
INT ALU	1	1	No

- a) Rellenar 25 ciclos del diagrama instrucción-tiempo correspondiente al comienzo del bucle interno con $k=3$ y $n=0$.
- b) A partir del apartado anterior, estimar el número de ciclos que tardará en ejecutarse el código completo y determinar el CPI. Indicar claramente los cálculos realizados para llegar a los resultados.

9. Sea un procesador RISC-V segmentado con las siguientes características:

- Se puede escribir un dato en el banco de registros y leer su nuevo valor en el mismo ciclo.
- Los saltos se resuelven en la etapa de ejecución y se cancelan las siguientes instrucciones 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 x1 y x2 contienen inicialmente el valor 4.
- Se dispone de las siguientes unidades funcionales:

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

En este procesador se ejecuta el siguiente fragmento de código:

```
L0:  fadd.s  f2, f4, f0
      fsw    f2, 0(x1)
      fdiv.s  f5, f4, f0
      fadd.s  f4, f0, f2
      fmul.s  f2, f2, f4
      flw     f4, 0(x1)
      fadd.s  f0, f2, f4
      fadd.s  f8, f6, f8
      addi    x2, x2, -1
      addi    x1, x1, 1
      bne     x2, x0, L0
      fadd.s  f0, f8, f2
      fadd.s  f2, f0, f8
```

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

10. Sea un RISC-V 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.
- Los saltos se resuelven en la etapa de ejecución y se cancelan las siguientes instrucciones si el salto es tomado.
- 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.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	3	Sí
FP SUB	1	3	Sí
FP MUL	1	4	Sí
FP DIV	1	5	No
INT ALU	1	1	No

En este procesador se ejecuta el siguiente fragmento de código

```
addi    x3, x0, 3
L1 : fsub.s  f2, f6, f8
fsub.s  f4, f8, f6
fsw     f4, 0(x3)
fdiv.s  f2, f4, f8
fadd.s  f2, f8, f8
subi    x3, x3, 1
fdiv.s  f6, f4, f8
fmul.s  f4, f2, f6
fsub.s  f10, f2, f6
flw     f4, 0(x3)
fadd.s  f0, f4, f2
bne     x3, x0, L1
fmul.s  f4, f2, f2
add     x3, x3, x3
```

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