



PROBLEMAS DE FUNDAMENTOS DE COMPUTADORES II

TEMA 7

Problemas básicos:

1. Dado el siguiente fragmento de código y suponiendo que el valor inicial de los registros **t1** y **t2** es 11 y 22 respectivamente,

```
addi t1, t2, 5
add  t3, t1, t2
addi t4, t1, 15
```

Se pide:

- El valor final de los registros tras ejecutarlo en el procesador multiciclo.
 - El valor final de los registros tras ejecutarlo en el procesador segmentado sin gestión de conflictos.
 - Insertar en el programa las instrucciones **nop** necesarias para que el anterior procesador obtenga un resultado correcto.
 - El valor final de los registros al ejecutarlo en el procesador segmentado.
2. Dado el siguiente fragmento de código y suponiendo que el valor inicial de los registros **t1** y **t2** es 11 y 22 respectivamente,

```
addi t1, t2, 5
add  t3, t1, t2
addi t4, t1, 15
add  t5, t1, t1
```

Se pide:

- El valor final de los registros tras ejecutarlo en el procesador multiciclo.
 - El valor final de los registros tras ejecutarlo en el procesador segmentado sin gestión de conflictos.
 - Insertar en el programa las instrucciones **nop** necesarias para que el anterior procesador obtenga un resultado correcto.
 - El valor final de los registros tras ejecutarlo en el procesador segmentado sin gestión de conflictos, pero con capacidad de escribir en el BR a mitad de ciclo y leer el valor escrito durante la segunda mitad de ciclo.
 - Inserte en el programa las instrucciones **nop** necesarias para que el anterior procesador obtenga un resultado correcto.
 - El valor final de los registros tras ejecutarlo en el procesador segmentado.
3. Suponiendo que el siguiente programa se ejecuta en el procesador segmentado, indique en qué etapa del pipeline se encuentra cada instrucción en los 5 primeros ciclos de ejecución, así como los registros leídos y escritos del BR en cada uno de ellos. Dibuje también el diagrama de ejecución del programa.

```
xor  s1, s2, s3
addi s0, s3, -4
lw   s3, 16(s7)
sw   s4, 20(s1)
```

```
or t2, s0, s1
```

4. Suponiendo que el siguiente programa se ejecuta en el procesador segmentado, indique en qué etapa del pipeline se encuentra cada instrucción en los 7 primeros ciclos de ejecución. Dibuje también el diagrama de ejecución del programa.

```
addi s1, zero, 11
lw s2, 25(s0)
add s3, s3, s4
or s4, s1, s2
lw s5, 16(s2)
```

5. Dado el siguiente fragmento de código:

```
addi t1, t2, 5
add t3, t1, t2
addi t4, t1, 15
add t5, t3, t2
```

Inserte en él las instrucciones **nop** que en cada caso sean necesarias para que se obtenga un resultado correcto, así como dibuje el diagrama de ejecución en:

- El procesador segmentado sin gestión de conflictos.
 - El procesador segmentado sin gestión de conflictos, pero con capacidad de escribir en el BR a mitad de ciclo y leer el valor escrito durante la segunda mitad de ciclo.
 - El procesador segmentado.
6. Dado el siguiente fragmento de código:

```
add x7, x5, x8
lw x6, 8(x7)
lw x5, 0(x5)
or x6, x7, x6
sw x6, 0(x7)
```

Inserte en él las instrucciones **nop** que en cada caso sean necesarias para que se obtenga un resultado correcto, así como dibujar el diagrama de ejecución en:

- El procesador segmentado sin gestión de conflictos.
 - El procesador segmentado sin gestión de conflictos, pero con capacidad de escribir en el BR a mitad de ciclo y leer el valor escrito durante la segunda mitad de ciclo.
 - El procesador segmentado.
- En el último caso, simule el valor que en los 7 primeros ciclos de reloj toman los registros de segmentación Rs1E, Rs2E, RdM, BRwrM, RdW y BRwrW así como las señales ForwardA y ForwardB.
7. Suponiendo que el siguiente programa se ejecuta en el procesador segmentado, indique en qué etapa del pipeline se encuentra cada instrucción en los 5 primeros ciclos de ejecución. Dibuje también el diagrama de ejecución del programa.

```
lw t1, 0(t0)
add t5, t2, t1
addi t0, t0, 1
sub t3, t5, t0
```

Repita el ejercicio reordenando el código para evitar la parada obligada por el conflicto de datos existente entre la primera y segunda instrucción.

8. Dado el siguiente fragmento de código:

```
lw  t1, 0(t0)
add t5, t2, t1
add t2, t1, t4
```

Inserte en él las instrucciones **nop** que en cada caso sean necesarias para que se obtenga un resultado correcto, así como dibuje el diagrama de ejecución en:

- El procesador segmentado sin gestión de conflictos.
 - El procesador segmentado sin gestión de conflictos, pero con capacidad de escribir en el BR a mitad de ciclo y leer el valor escrito durante la segunda mitad de ciclo.
 - El procesador segmentado con gestión parcial de conflictos (escritura de BR a mitad de ciclo + unidad de anticipación).
 - El procesador segmentado.
 - En este último caso, indique también en qué etapa del pipeline se encuentra cada instrucción en los 5 primeros ciclos
9. Suponiendo que el siguiente programa se ejecuta en el procesador segmentado, indique en qué etapa del pipeline se encuentra cada instrucción en 10 primeros ciclos de ejecución. Indique también el diagrama de ejecución del programa.

```
addi s1, zero, 11
lw    s2, 25(s1)
lw    s5, 16(s2)
add   s3, s2, s5
or    s4, s3, t4
and   s2, s3, s4
```

10. Dado el siguiente fragmento de código:

```
addi s1, zero, 52
addi s0, s1, -4
lw    s3, 16(s0)
sw    s3, 20(s0)
xor   s2, s0, s3
or    s2, s2, s3
```

Indique el diagrama de ejecución y en qué etapa del pipeline se encuentra cada instrucción en los 7 primeros ciclos en:

- El procesador segmentado.
 - El procesador segmentado con anticipación optimizada, es decir, con la capacidad de anticipar datos a la etapa MEM desde la etapa WB para tratar las dependencias de datos **lw** → **sw**,
11. Dado el siguiente fragmento de código:

```
lw  s3, 0(s4)
add s5, s4, s3
add s2, s3, s6
and s1, s1, s2
lw  s5, 0(t3)
sw  s5, 0(t3)
or  s2, s2, s5
```

Indique el diagrama de ejecución en:

- El procesador segmentado.
- El procesador segmentado con anticipación optimizada, es decir, con la capacidad de

anticipar datos a la etapa MEM desde la etapa WB para tratar las dependencias de datos **lw** → **sw**,

12. Dado el siguiente fragmento de código y suponiendo que el valor inicial de los registros **t1** y **t2** es 11 y 22 respectivamente,

```
    beq zero, s0, L1
    addi t1, t1, -1
    addi t2, t1, -1
L1:
    addi t3, t1, 1
    addi t4, t2, 1
```

Suponiendo que **s0** contiene un valor distinto de 0, se pide calcular:

- El valor final de los registros tras ejecutarlo en el procesador multiciclo.
- El valor final de los registros tras ejecutarlo en el procesador segmentado sin gestión de conflictos de control.
- Insertar en el programa las instrucciones **nop** necesarias para que el anterior procesador obtenga un resultado correcto. Dibujar el diagrama de ejecución resultante.
- Dibujar el diagrama de ejecución y en qué etapa del pipeline se encuentra cada instrucción en los 7 primeros ciclos en el procesador segmentado con gestión de conflictos de control mediante parada.
- Dibujar el diagrama de ejecución y en qué etapa del pipeline se encuentra cada instrucción en los 7 primeros ciclos en el procesador segmentado (con gestión de conflictos de control mediante predicción de salto no tomado).

Suponiendo que **s0** contiene un 0, se pide:

- El valor final de los registros tras ejecutarlo en el procesador multiciclo.
- El valor final de los registros tras ejecutarlo en el procesador segmentado sin gestión de conflictos de control.
- Insertar en el programa las instrucciones **nop** necesarias para que el anterior procesador obtenga un resultado correcto. Dibujar el diagrama de ejecución resultante.
- Dibujar el diagrama de ejecución y en qué etapa del pipeline se encuentra cada instrucción en los 7 primeros ciclos en el procesador segmentado con gestión de conflictos de control mediante parada.
- Dibujar el diagrama de ejecución y en qué etapa del pipeline se encuentra cada instrucción en los 7 primeros ciclos en el procesador segmentado (con gestión de conflictos de control mediante predicción de salto no tomado).

13. Suponiendo que el siguiente programa se ejecuta en el procesador segmentado indique en qué etapa del pipeline se encuentra cada instrucción en los 8 primeros ciclos de ejecución. Dibuje también el diagrama de ejecución del programa.

```
    jal x0, L1
    addi t1, x0, 5
    add t3, t1, t2
L1:
    sw t4, 0(t3)
```

14. Suponiendo que el valor inicial del registro `t0` es 1, dibuje el diagrama de ejecución en el procesador segmentado del siguiente fragmento de código:

```
    addi s0, x0, 0
L1:  beq  t0, x0, L2
     add  s0, s0, t0
     addi t0, t0, -1
     jal  x0, L1
L2:  add  s0, s0, s2
     sw   s0, 0(gp)
```

15. Suponiendo que el valor cargado por la segunda instrucción en el registro `t0` es 1, dibuje el diagrama de ejecución en el procesador segmentado del siguiente fragmento de código:

```
    addi s0, x0, 0
    lw   t0, 4(gp)
L1:  beq  t0, x0, L2
     add  s0, s0, t0
     addi t0, t0, -1
     jal  x0, L1
L2:  lw   s1, 0(gp)
     add  s1, s0, s1
     sw   s1, 0(gp)
```

16. Dado el siguiente fragmento de código:

```
// for( sum=0, i=0; i!=N; i++ )
//   sum = sum + a[i];
//
// a[]->s0, N->s1, i->s2, sum->s3
//
.equ N, ...
...
addi s1, zero, N
addi s2, zero, 0
addi s3, zero, 0
for:
beq  s2, s1, efor
slli t0, s2, 2
add  t0, s0, t0
lw   t0, 0(t0)
add  s3, s3, t0
addi s2, s2, 1
jal  x0, for
efor:
sw   s3, 0(gp)
...
```

Suponiendo que se ejecuta en el procesador segmentado calcule:

- El número de instrucciones que se ejecuta.
- El número de ciclos que tarda en ejecutarlas
- El CPI resultante.

¿Tardaría menos si se reordena el código? En caso afirmativo, indique un código más eficiente y calcule el nuevo CPI.

17. Supongamos que sobre el procesador RISC-V segmentado, se ejecuta un programa de 500 instrucciones distribuidas de la siguiente manera:

- El 20% corresponden a instrucciones de **lw**, y de éstas, en la mitad de los casos vienen seguidas de una instrucción aritmética que lee el registro sobre el que escribe la instrucción **lw**.
- El 15% son instrucciones de **sw**.
- El 25% son instrucciones **beq**, en las que la condición evaluada se cumple en el 70% de los casos.
- El 5% se corresponden con instrucciones **jal**.
- El 35% restante son instrucciones aritmético-lógicas.

Suponiendo que el procesador funciona a una frecuencia de 1.5 GHz, calcula el CPI obtenido en la ejecución del programa, así como el tiempo de ejecución de este.

Problemas adicionales:

18. Dado el siguiente fragmento de código:

```
        addi s4, zero, 2
L1:     lw    s0, 0(s2)
        lw    s1, 0(s2)
        add  s3, s0, s1
        sw   s3, 0(s2)
        add  s2, s3, s6
        and  s1, s1, s2
        beq  s4, zero, L1
        addi s2, s2, 1
        or   s3, s2, s4
```

Indique el diagrama de ejecución en:

- a) El procesador segmentado.
- b) Un procesador segmentado que resuelve todos los conflictos por parada y es capaz de escribir de BR a mitad de ciclo.

19. Suponiendo que el siguiente fragmento de código se ejecuta en el procesador segmentado, dibuje el diagrama de su ejecución tanto si la condición evaluada por la instrucción **beq** es cierta como si no lo es.

```
        lw  t1, 0(s1)
        lw  t2, 0(s2)
        beq t1, t2, else
        add t3, t1, t2
        jal zero, eif:
else:
        sub t3, t1, t2
eif:
        and t4, t3, t1
        or  t5, t3, t2
```

20. Dado el siguiente fragmento de código:

```
    addi s1, zero, 1
L1:  sub  s5, s5, s2
    addi s2, s2, 1
    addi s1, s1, -1
    beq  s1, zero, L1
    and  s5, s2, s3
    or   s3, s3, s4
    andi s4, s3, s2
```

Indique el diagrama de ejecución de las 2 primeras iteraciones.

21. Discuta las modificaciones que sería necesario realizar en la ruta de datos del procesador segmentado para incorporar al repertorio de una nueva instrucción de tipo-R que lea una palabra de memoria usando un registro base y un desplazamiento variable almacenado en un registro índice, **lwi rd, rs1, rs2**:

$$\{ rd \leftarrow \text{Mem}[rs1 + rs2] \}$$

Analice los riesgos que se producirían y la manera de resolverlos.

22. Discuta las modificaciones que sería necesario realizar en la ruta de datos del procesador segmentado para añadir una nueva instrucción de tipo-R que intercambie el valor contenido en 2 registros, **swap rs1, rs2**:

$$\{ rs1 \leftarrow rs2, rs2 \leftarrow rs1 \}$$

Analice los riesgos que se producirían y la manera de resolverlos.

23. Si pudiera reducir el retardo de solo uno los módulos del procesador segmentado diseñado con la biblioteca CMOS de 90 nm, ¿cuál elegiría para obtener un tiempo de ciclo menor? ¿cuánta reducción tendría sentido realizar? ¿cuál sería el nuevo tiempo de ciclo?

24. Si pudiera dividir una de las etapas del procesador segmentado diseñado con la biblioteca CMOS de 90 nm en otras 2 de igual retardo, ¿cuál elegiría para obtener un tiempo de ciclo menor? ¿cuál sería el nuevo tiempo de ciclo?

25. Si se redujera un 30% el retardo de la ALU, un 10% el retardo del extensor de signo y 20% los retardos de lectura/escritura del banco de registros del procesador segmentado diseñado con la biblioteca CMOS de 90 nm, calcule

- El retardo del camino crítico en cada etapa del *pipeline*.
- El speed-up obtenido al ejecutar un programa de 100 millones de instrucciones.