

# Ejercicios procesador segmentado

JUAN MARCELO ARPI QUIZHPE<sup>1</sup>, Andrés Meneses<sup>1</sup>, and magalimejiap<sup>1</sup>

<sup>1</sup>Affiliation not available

July 2, 2020

## Problema 1

El siguiente fragmento de código se ejecuta en el nanoMIPS segmentado:

```

1      ADD      R5, R4, R2
2      LW       R1, 0, (R5)
3      AND      R6, R1, R3
4      ADD      R8, R6, R1
5      OR       R5, R4, R9
6      SW       R6, 0 (R0)
7      SW       R8, 100 (R0)
    
```

Calcular el número de ciclos necesarios para ejecutar el código y justificarlo con un diagrama de ejecución

a) Si no existe posibilidad de adelantar operandos ni de reordenar el código

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| 1 | F | D | X | M | W |   |   |   |   |    |    |    |    |    |    |    |    |
| 2 |   | F | D | D | D | X | M | W |   |    |    |    |    |    |    |    |    |
| 3 |   |   | F | F | F | D | D | D | X | M  | W  |    |    |    |    |    |    |
| 4 |   |   |   |   |   | F | F | F | D | D  | D  | X  | M  | W  |    |    |    |
| 5 |   |   |   |   |   |   |   |   | F | F  | F  | D  | X  | M  | W  |    |    |
| 6 |   |   |   |   |   |   |   |   |   |    |    | F  | D  | X  | M  | W  |    |
| 7 |   |   |   |   |   |   |   |   |   |    |    |    | F  | D  | X  | M  | W  |

17 ciclos y 6 paros

b) Si se puede utilizar adelantamiento de operandos

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| 1 | F | D | X | M | W |   |   |   |   |    |    |    |
| 2 |   | F | D | X | M | W |   |   |   |    |    |    |
| 3 |   |   | F | D | D | X | M | W |   |    |    |    |
| 4 |   |   |   | F | F | D | X | M | W |    |    |    |
| 5 |   |   |   |   |   | F | D | X | M | W  |    |    |
| 6 |   |   |   |   |   |   | F | D | X | M  | W  |    |
| 7 |   |   |   |   |   |   |   | F | D | X  | M  | W  |

12 ciclos y 1 paro

c) Si además de permitir adelantamiento, se permite reordenar el código para que el número de ciclos sea mínimo, ¿cuál sería la mejor ordenación? ¿cuántos ciclos serían necesarios para ejecutar este código?

1 ADD R5, R4, R2

2 LW R1, 0, (R5)

7 SW R8, 100(R0)

3 AND R6, R1, R3

4 ADD R8, R6, R1

5 OR R5, R4, R9

6 SW R6, 0(R0)

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|----|----|
| 1 | F | D | X | M | W |   |   |   |   |    |    |
| 2 |   | F | D | X | M | W |   |   |   |    |    |
| 3 |   |   | F | D | X | M | W |   |   |    |    |
| 4 |   |   |   | F | D | X | M | W |   |    |    |
| 5 |   |   |   |   | F | D | X | M | W |    |    |
| 6 |   |   |   |   |   | F | D | X | M | W  |    |
| 7 |   |   |   |   |   |   | F | D | X | M  | W  |

## Problema 2

El siguiente fragmento de código se ejecuta en el nanoMIPS segmentado con resolución de riesgos mediante adelantamiento y predicción de salto no tomado:

```

1      ADDI      R10,R0,#0
2      ADDI      R11,R0,#4000
3  bucle:LW      R1,A(R10)
4      BEQ      R1,R0,salto
5      LW       R2,B(R10)
6      ADD      R3,R1,R2
7      SW       R1,B(R10)
8      SW       R3,A(R10)
9  salto:ADDI    R10,R10,#4
10     BNE      R10,R11,bucle

```

Debes mostrar el diagrama de ejecución y el número de ciclos totales para los siguientes casos:

a) Caso 1: el elemento de A al que se accede es igual a 0

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 3  | F | D | X | M | W |   |   |   |   |
| 4  |   | F | D | D | X | M | W |   |   |
| 5  |   |   |   |   |   |   |   |   |   |
| 6  |   |   |   |   |   |   |   |   |   |
| 7  |   |   |   |   |   |   |   |   |   |
| 8  |   |   |   |   |   |   |   |   |   |
| 9  |   |   | F | F | D | X | M | W |   |
| 10 |   |   |   |   | F | D | X | M | W |

9 ciclos y 1 paro

b) Caso 2: el elemento de A al que se accede es distinto de 0

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|
| 3  | F | D | X | M | W |   |   |   |   |    |    |    |    |    |
| 4  |   | F | D | D | X | M | W |   |   |    |    |    |    |    |
| 5  |   |   | F | F | D | X | M | W |   |    |    |    |    |    |
| 6  |   |   |   | F | D | D | X | M | W |    |    |    |    |    |
| 7  |   |   |   |   | F | F | D | X | M | W  |    |    |    |    |
| 8  |   |   |   |   |   |   | F | D | X | M  | W  |    |    |    |
| 9  |   |   |   |   |   |   |   | F | D | X  | M  | W  |    |    |
| 10 |   |   |   |   |   |   |   |   | F | D  | X  | M  | W  |    |

14 ciclos y 2 pasos

### Problema 3

El siguiente código se ejecuta en el nanoMIPS segmentado con adelantamiento y resolución de saltos en la etapa D.

```

                                addi    $s0, $r0, 1000
                                addi    $t0, $r0, 0
1   loop:    lw      $t1, A($t0)
2           lw      $t2, B($t0)
3           sub     $t3, $t1, $t2
4           beq    $t3, $r0, else
5           sw     $t2, A($t0)
6           b      end
7   else:    lw      $t4, C($t0)
8           add    $t1, $t1, $t4
9           sw     $t1, A($t0)
10          sw     $t1, B($t0)
11   end:    addi    $t0, $t0, 4
12          bne   $t0, $s0, loop

```

Mostrar el diagrama de ejecución (indicando el tipo de dependencia entre las instrucciones y los parones introducidos) y calcular el CPI medio teniendo en cuenta que el 65% de las veces  $A[i] = B[i]$ . ¿Qué mejora se obtendría en el código si se emplea predicción de salto estática con salto no tomado?.

1. Con salto

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| 1  | F | D | X | M | W |   |   |   |   |    |    |    |    |    |    |    |    |
| 2  |   | F | D | X | M | W |   |   |   |    |    |    |    |    |    |    |    |
| 3  |   |   | F | D | D | X | M | W |   |    |    |    |    |    |    |    |    |
| 4  |   |   |   | F | F | D | X | M | W |    |    |    |    |    |    |    |    |
| 5  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 6  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 7  |   |   |   |   | F | D | X | M | W |    |    |    |    |    |    |    |    |
| 8  |   |   |   |   |   | F | D | D | X | M  | W  |    |    |    |    |    |    |
| 9  |   |   |   |   |   |   | F | F | D | X  | M  | W  |    |    |    |    |    |
| 10 |   |   |   |   |   |   |   | F | D | D  | X  | M  | W  |    |    |    |    |
| 11 |   |   |   |   |   |   |   |   | F | F  | D  | X  | M  | W  |    |    |    |
| 12 |   |   |   |   |   |   |   |   |   |    | F  | D  | X  | M  | W  |    |    |

17 ciclos y 3 paros.

**Penalización media por riesgos de control utilizando salto tomado:**

$$\text{Pen} = 0.65 * 1 + 0.35 * 3 = 1.7 \text{ ciclos}$$

**Cálculo del CPI:**

$$CPI = \frac{\text{NumCiclos} + \text{CiclosPen}}{\text{Instrucciones}}$$

$$CPI = \frac{17 \text{ ciclos} + 1.7 \text{ ciclos}}{12 \text{ instrucciones}} = 1,56$$

2. Sin salto

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|
| 1  | F | D | X | M | W |   |   |   |   |    |    |    |    |
| 2  |   | F | D | X | M | W |   |   |   |    |    |    |    |
| 3  |   |   | F | D | D | X | M | W |   |    |    |    |    |
| 4  |   |   |   | F | F | D | X | M | W |    |    |    |    |
| 5  |   |   |   |   |   | F | D | X | M | W  |    |    |    |
| 6  |   |   |   |   |   |   | F | D | X | M  | W  |    |    |
| 7  |   |   |   |   |   |   |   |   |   |    |    |    |    |
| 8  |   |   |   |   |   |   |   |   |   |    |    |    |    |
| 9  |   |   |   |   |   |   |   |   |   |    |    |    |    |
| 10 |   |   |   |   |   |   |   |   |   |    |    |    |    |
| 11 |   |   |   |   |   |   |   | F | D | X  | M  | W  |    |
| 12 |   |   |   |   |   |   |   |   | F | D  | X  | M  | W  |

13 ciclos y 1 paro

## Problema 4

El siguiente código se ejecuta en un nanoMIPS segmentado con adelantamiento, pero sólo para los riesgos de datos.

```

                                addi    $s0, $r0, 1000
                                addi    $t0, $r0, 0
1   loop:                       lw      $t1, A($t0)
2                               lw      $t2, B($t0)
3                               sub     $t3, $t1, $t2
4                               beq     $t3, $r0, else
5                               sw      $t2, A($t0)
6                               b       end
7   else:                       lw      $t4, C($t0)
8                               add     $t1, $t1, $t4
9                               sw      $t1, A($t0)
10                              sw      $t1, B($t0)
11  end:                       addi    $t0, $t0, 4
12                              bne    $t0, $s0, loop

```

Mostrar el diagrama de ejecución (indicando el tipo de dependencia entre las instrucciones y los parones introducidos) y calcular el CPI medio teniendo en cuenta que el 65% de las veces  $A[i] = B[i]$ . ¿Qué posibles técnicas, tanto hardware como software, se podrían utilizar para reducir el número de parones? Enumerar y explicar brevemente.

a) Con salto

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| 1  | F | D | X | M | W |   |   |   |   |    |    |    |    |    |    |    |    |
| 2  |   | F | D | X | M | W |   |   |   |    |    |    |    |    |    |    |    |
| 3  |   |   | F | D | D | X | M | W |   |    |    |    |    |    |    |    |    |
| 4  |   |   |   | F | F | D | X | M | W |    |    |    |    |    |    |    |    |
| 5  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 6  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 7  |   |   |   |   | F | D | X | M | W |    |    |    |    |    |    |    |    |
| 8  |   |   |   |   |   | F | D | D | X | M  | W  |    |    |    |    |    |    |
| 9  |   |   |   |   |   |   | F | F | D | X  | M  | W  |    |    |    |    |    |
| 10 |   |   |   |   |   |   |   | F | D | D  | X  | M  | W  |    |    |    |    |
| 11 |   |   |   |   |   |   |   |   | F | F  | D  | X  | M  | W  |    |    |    |
| 12 |   |   |   |   |   |   |   |   |   |    | F  | D  | X  | M  | W  |    |    |

Penalización media por riesgos de control utilizando salto tomando:

$$\text{Pen} = 0.65 \cdot 1 + 0.35 \cdot 3 = 1.7 \text{ ciclos}$$

Cálculo del CPI:

$$CPI = \frac{\text{NumCiclos} + \text{CiclosPen}}{\text{Instrucciones}}$$

$$CPI = \frac{17 \text{ ciclos} + 1.7 \text{ ciclos}}{12 \text{ Instrucciones}} = 1.56$$

b) Sin salto

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|
| 1  | F | D | X | M | W |   |   |   |   |    |    |    |    |
| 2  |   | F | D | X | M | W |   |   |   |    |    |    |    |
| 3  |   |   | F | D | D | X | M | W |   |    |    |    |    |
| 4  |   |   |   | F | F | D | X | M | W |    |    |    |    |
| 5  |   |   |   |   |   | F | D | X | M | W  |    |    |    |
| 6  |   |   |   |   |   |   | F | D | X | M  | W  |    |    |
| 7  |   |   |   |   |   |   |   |   |   |    |    |    |    |
| 8  |   |   |   |   |   |   |   |   |   |    |    |    |    |
| 9  |   |   |   |   |   |   |   |   |   |    |    |    |    |
| 10 |   |   |   |   |   |   |   |   |   |    |    |    |    |
| 11 |   |   |   |   |   |   | F | D | X | M  | W  |    |    |
| 12 |   |   |   |   |   |   |   | F | D | X  | M  | W  |    |

13 ciclos y 1 paro

¿Qué posibles técnicas, tanto hardware como software, se podrían utilizar para reducir el número de parones? Enumerar y explicar brevemente.

Solución Software.

1. Se reordena código sin variar la lógica del programa.
2. Insertar instrucciones NOP.
3. Retrasar la ejecución de la instrucción encontrando k instrucciones que se puedan ejecutar después de la instrucción que genera la dependencia sin variar la estructura lógica del programa.

## Solución Hardware

1. Detener el pipeline: Duplicar recursos de hardware que provocan conflictos.
2. Separar memoria de instrucciones y de datos.

Adelantamiento: Pasar el resultado obtenido en una instrucción a las instrucciones que lo necesitan como operando.

## Problema 5

El siguiente código se ejecuta en el nanoMIPS segmentado con adelantamiento para riesgos de datos.

```

1  loop:      lw      $r1, 0($r1)
2             and     $r1, $r1, $r2
3             lw      $r1, 0($r1)
4             lw      $r1, 0($r1)
5             beq     $r1, $r0, loop
    
```

Indica el tipo de dependencia entre las instrucciones, el número de parones introducidos por la dependencia y entre que etapas se produce el adelantamiento. Calcular el CPI medio.

- Existe dependencia entre la instrucción 1 y 2, pues la instrucción 2 necesita el registro dónde se guarda la información en la introducción 1.
- Se dan 3 parones.
- Cálculo del CPI medio

$$CPI = \frac{Ciclos}{Instrucciones} = \frac{11}{4} = 2.75$$

| I/C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|-----|---|---|---|---|---|---|---|---|---|----|----|
| 1   | F | D | X | M | W |   |   |   |   |    |    |
| 2   |   | F | D | D | X | M | W |   |   |    |    |
| 3   |   |   | F | F | D | X | M | W |   |    |    |
| 4   |   |   |   | F | F | D | D | X | M | W  |    |
| 5   |   |   |   |   | F | F | D | D | X | M  | W  |

¿Qué mejora se obtendría en el código si se emplea adelantamiento para riesgos de control?

No se tendría mejora en el código, dado que el adelantamiento para riesgos de control en el caso de BEQ hasta la fase M no se podrá ejecutar la siguiente instrucción, que sería la etiqueta. Al generar más ciclos por ende se daría un mayor CPI.

**Si además se añade predicción de salto estática, ¿qué predicción utilizarías? ¿por qué?**

Predicción de salto tomado: Conociendo la dirección destino del salto, se puede llenar el cauce con instrucciones de dicha dirección

## Problema 6

El siguiente código se ejecuta en el nanoMIPS segmentado con adelantamiento para riesgos de datos y de control:

```
1 loop:   lw $t1, 0($s1)
2         lw $t2, 0($t1)
3         subi $t3, $t3, 1
4         beq $t3, $r0, else
5         sw $t2, A($t0)
6         b end
7 else:   lw $t4, 0($t0)
8         add $t1, $t1, $t4
9         sw $t1, 0($s1)
10        end: addi $t0, $t0, 4
11        bne $t0, $s0, loop
```

Para las dos posibles secuencias de ejecución, indica el tipo de dependencia entre las instrucciones, el número de parones introducidos por la dependencia y entre que etapas se produce el adelantamiento. Calcular el CPI medio para ambas secuencias (2 puntos). ¿Qué otra técnica hardware se podrían utilizar para reducir el número de parones? Indica cual sería el CPI en este caso.

| I/C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|
| 1   | F | D | X | M | W |   |   |   |   |    |    |    |    |
| 2   |   | F | D | D | X | M | W |   |   |    |    |    |    |
| 3   |   |   | F | D | D | X | M | W |   |    |    |    |    |
| 4   |   |   |   | F | F | D | X | M | W |    |    |    |    |
| 5   |   |   |   |   |   | F | D | X | M | W  |    |    |    |
| 6   |   |   |   |   |   |   | F | D | X | M  | W  |    |    |
| 7   |   |   |   |   |   |   |   | F | D | X  | M  | W  |    |
| 8   |   |   |   |   |   |   |   |   | F | D  | X  | M  | W  |

13 ciclos y 2 paros

$$CPI = \frac{\text{NúmeroDeCiclos}}{\text{NúmeroInstrucciones}} = \frac{13}{8} = 1.62$$

| I/C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|
| 1   | F | D | X | M | W |   |   |   |   |    |    |    |    |
| 2   |   | F | D | D | X | M | W |   |   |    |    |    |    |
| 3   |   |   | F | D | D | X | M | W |   |    |    |    |    |
| 4   |   |   |   | F | F | D | X | M | W |    |    |    |    |
| 5   |   |   |   |   |   | F | D | X | M | W  |    |    |    |
| 6   |   |   |   |   |   |   | F | D | D | X  | M  | W  |    |
| 7   |   |   |   |   |   |   |   | F | F | D  | X  | M  | W  |

13 ciclos y 3 paros

$$CPI = \frac{\text{NúmeroDeCiclos}}{\text{NúmeroInstrucciones}} = \frac{13}{7} = 1.85$$

También se podría reordenar el código y realizar una predicción de salto estática.

## PROBLEMA 7

El siguiente código se ejecuta en el nanoMIPS segmentado con adelantamiento para riesgos de datos.

```

1 loop: lw $t1, 0($t1)
2 lw $t1, 0($t1)
3 subi $t3, $t3, 1
4 beq $t3, $zero, else
5 sw $t1, A($t3)
6 b end
7 else: lw $t4, 0($t0)
8 add $t2, $t1, $t4
9 sw $t2, B($t3)
10 end: addi $t0, $t0, 4
11 bne $t0, $s0, loop

```

Indica las dependencias de datos tipo RAW entre las instrucciones, el número de parones introducidos por la dependencia y entre qué etapas se produce el adelantamiento. Si suponemos que ya se han realizado los ciclos de llenado, calcula el CPI para cada secuencia (1,5 punto). ¿Qué mejora se obtendría en el código si se emplea adelantamiento para riesgos de control? Si además se añade predicción de salto estática, ¿qué predicción utilizarías? ¿por qué?

| I/C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| 1   | F | D | X | M | W |   |   |   |   |    |    |    |    |    |    |    |    |    |
| 2   |   | F | D | D | X | M | W |   |   |    |    |    |    |    |    |    |    |    |
| 3   |   |   | F | F | D | D | X | M | W |    |    |    |    |    |    |    |    |    |
| 4   |   |   |   | F | F | D | X | M | W |    |    |    |    |    |    |    |    |    |
| 5   |   |   |   |   | F | D | X | M | W |    |    |    |    |    |    |    |    |    |
| 6   |   |   |   |   |   | F | D | X | M | W  |    |    |    |    |    |    |    |    |
| 7   |   |   |   |   |   |   | F | D | X | M  | W  |    |    |    |    |    |    |    |
| 8   |   |   |   |   |   |   |   | F | D | D  | X  | M  | W  |    |    |    |    |    |
| 9   |   |   |   |   |   |   |   |   | F | F  | D  | X  | M  | W  |    |    |    |    |
| 10  |   |   |   |   |   |   |   |   |   | F  | D  | X  | M  | W  |    |    |    |    |
| 11  |   |   |   |   |   |   |   |   |   |    | F  | D  | X  | M  | W  |    |    |    |

Dependencias entre \$t1, \$t3, \$T2, \$t0

18 ciclos y 3 parones

$$CPI = \frac{\text{NúmeroDeCiclos}}{\text{NúmeroInstrucciones}} = \frac{18}{11} = 1.63$$