

Fecha: 18 de octubre de 2019  
Duración: 90 minutos

### Pregunta 1. Rendimiento (2 pts.)

#### Parte 1

Considere dos implementaciones distintas del mismo conjunto de instrucciones (ISA). Las instrucciones pueden ser divididas en cuatro clases de acuerdo con su CPI: clase A, B, C y D. La implementación P1 tiene una tasa de reloj de 2,5 GHz y CPIs de 1, 2, 3 y 3, y la implementación P2, una tasa de reloj de 3 GHz y CPIs de 2, 2, 2 y 2, para las clases A, B, C y D, respectivamente. Dado un programa con un total de un millón de instrucciones divididas en las clases como sigue: 10% clase A, 20% clase B, 50% clase C, y 20% clase D.

- A) (0,4 pts.) ¿Cuál es más rápida, P1 o P2?  
B) (0,4 pts.) ¿Cuál es el CPI global para P1 y P2?  
C) (0,4 pts.) Encuentre el número de ciclos de reloj que requieren P1 y a P2 para ejecutar el programa.

#### Parte 2

Considere el siguiente programa escrito en lenguaje C. Se pide comparar el desempeño de un procesador MIPS monociclo con tasa de reloj de 1 GHz y un procesador MIPS con *pipeline* de 5 etapas, como el discutido en clases, con tasa de reloj de 3 GHz. Asuma que el procesador con *pipeline* dispone de adelantamiento completo (*forwarding*), predicción perfecta de bifurcaciones y que las direcciones en saltos incondicionales están disponibles en la etapa IF.

```
int i;
for (i=0; i<10000; i=i+1)
    tangananica[i] = tangananana[i] - 10;
```

- D) (0,4 pts.) Determine el CPI para ambos procesadores.  
E) (0,4 pts.) Calcule el tiempo de ejecución para ambos procesadores.

$$A) \text{ tiempo CPU} = \frac{\# \text{ Instr.} \times \text{CPI}}{\text{tasa reloj}} = \frac{\sum \# \text{ Instr.}_i \times \text{CPI}_i}{\text{tasa reloj}}$$

$$\text{tiempo P1} = (1 \times 10^5 \times 1 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 3 + 2 \times 10^5 \times 3) / 2,5 \times 10^9$$

$$= 10,4 \times 10^{-4} \text{ s}$$

$$\text{tiempo P2} = (1 \times 10^5 \times 2 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 2 + 2 \times 10^5 \times 2) / 3 \times 10^9$$

$$= 6,66 \times 10^{-4} \text{ s}$$

→ Es más rápida P2

$$B) \text{ CPI} = \frac{\text{tiempo CPU} \times \text{tasa reloj}}{\# \text{ Instr.}}$$

$$\text{CPI(P1)} = \frac{10,4 \times 10^{-4} [\text{s}] \times 2,5 \times 10^9 [\text{Hz}]}{1 \times 10^6} = 2,6$$

$$\text{CPI(P2)} = \frac{6,66 \times 10^{-4} [\text{s}] \times 3 \times 10^9 [\text{Hz}]}{1 \times 10^6} = 2,0$$

$$C) \quad \# \text{ciclos} = \text{CPI} \times \# \text{Instr}$$

$$\# \text{ciclos (P1)} = 2,6 \times 10^6$$

$$\# \text{ciclos (P2)} = 2 \times 10^6$$

D)

Notar que el programa incluye, dependiendo del compilador, más o menos las sgtes instrucciones:

	beq	(verificar si seguir iterando)
	lw	(cargar tangana [i])
!burbuja necesaria!	add	(sumar -10 a lo cargado arriba)
	sw	(guardar en tangana [i])
	addi	(actualizar índice)
	j	(saltar para nueva iteración)

CPI monociclo = 1      pues cada instrucción se ejecuta

$$\text{CPI pipeline} = \frac{\begin{array}{l} \text{ciclos para} \\ \text{llenar pipeline} \end{array} 4 + \begin{array}{l} \text{en 1 ciclo} \\ \text{ciclos por} \\ \text{cada instr.} \end{array} 6 \times 10000 + \begin{array}{l} \text{burbuja} \\ \text{ciclos} \end{array} 1 \times 10000}{\begin{array}{l} 6 \times 10000 \\ \text{total instr.} \end{array}} \approx \frac{7}{6}$$

Se podría argumentar que el compilador reordene instrucciones para evitar burbuja insertando addi entre lw, add. En tal caso, CPI  $\approx$  1

$$E) \quad \text{tiempo monociclo} = \frac{1 \times 6 \times 10^4}{1 \times 10^9} = 6 \times 10^{-5} \text{ s}$$

$$\text{tiempo pipeline} = \frac{\frac{7}{6} \times 6 \times 10^4}{3 \times 10^9} = \frac{7}{3} \times 10^{-5} \text{ s}$$

**Pregunta 2. Procesador Monociclo (2 pts.)**

Considere el procesador MIPS monociclo visto en clases (Figura 1). Suponga que en determinado ciclo de reloj se está ejecutando la instrucción '10101100011000100000000000010100' que está almacenada en la dirección de memoria de instrucciones 0x000001A0. Asuma que la memoria de datos contiene sólo ceros y que los registros del procesador tienen los siguientes valores al inicio del ciclo en que se realiza "fetch" de esta instrucción:

R0	R1	R2	R3	R4	R5	R6	R8	R12	R31
0	-1	2	-3	-4	10	6	8	2	-16

- A) (0,2 pts.) ¿Cuáles son las salidas de los bloques "sign-extend" y el "shift left 2" de más arriba en la Figura 1?
- B) (0,3 pts.) ¿Cuáles son los valores de *todas* las entradas de la unidad ALU Control?
- C) (0,3 pts.) ¿Cuál es el nuevo valor del contador de programa al terminar la ejecución de esta instrucción? Además, marque en la Figura 1 el camino de datos por el cual este valor es determinado.
- D) (0,4 pts.) Para cada multiplexor, indique los valores de salida.
- E) (0,4 pts.) Para la ALU y las dos unidades de suma (Add), ¿cuáles son los valores de entrada?
- F) (0,4 pts.) ¿Cuáles son los valores de *todas* las entradas del archivo de registros (unidad "Registers")?

Notar que instrucción es "sw" pues opcode = 101011 = 43 (Tabla 2)

- A) "Sign-extend" : 00...010100 (32 bits)  
 El "immediate" es positivo → se concatenan 16 ceros a la izquierda  
 "shift left 2" : 0001100010...01010000 (28 bits)  
 A los 26 bits de más a la derecha se le agregan 2 ceros a la derecha

- B) ALUOp : 00 (se deduce de Tabla 1)  
 Instruction [5-0] : 010100

- C) Nuevo valor PC : 0x000001A4 (simplemente sumar 4)  
 \* ver camino en Fig.

- D) mux WrReg : 2 o 0 (RegDst es "don't care" para sw)  
 mux ALU : 20  
 mux Mem/ALU : X  
 mux Branch : PC+4 (parte c)  
 mux Jump : PC+4 (parte c)

E)

ALU : -3 y 20

Add (PC+4) : PC , 4

Add (Branch) : PC+4 ,  $20 \times 4$

F)

Read Register 1 :  $\phi\phi\phi11 = 3$

Read Register 2 :  $\phi\phi\phi1\phi = 2$

Write Register :  $\phi\phi\phi\phi\phi = 0$

Write Data : 0 (mem de datos con ceros)

Reg Write : 0 (no se escribe nada)

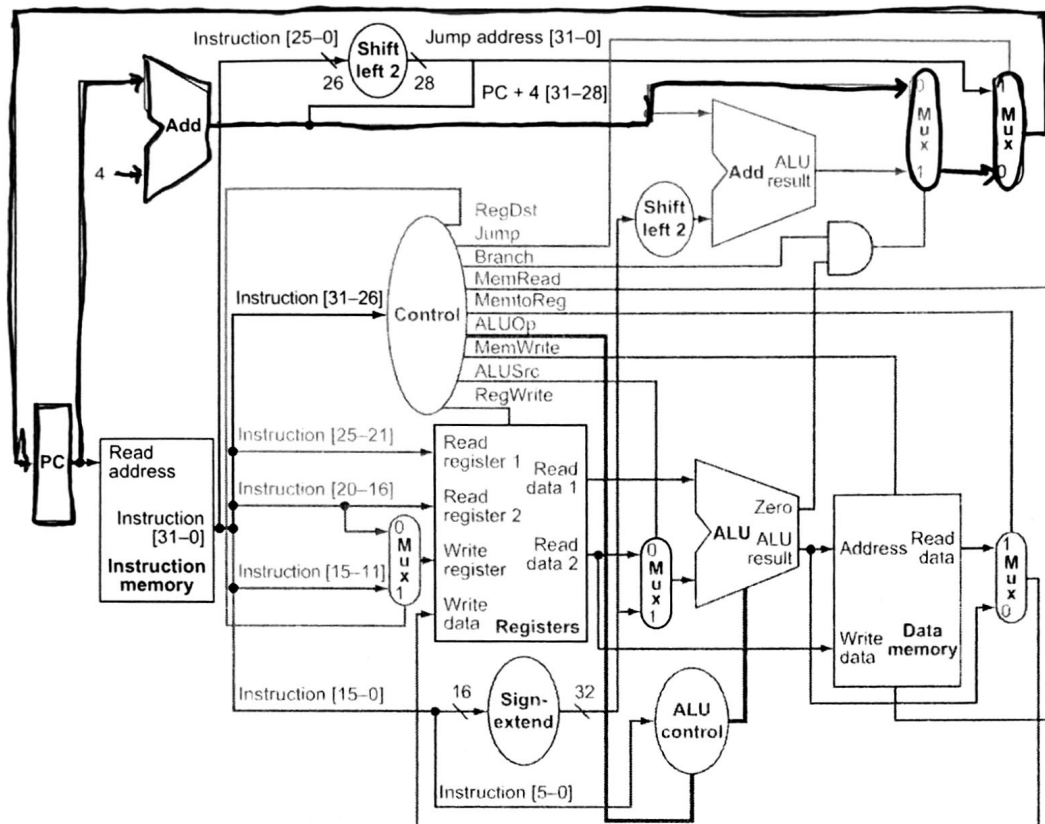


Figura 1. Diagrama de un procesador monociclo. Se ha marcado el camino de datos de la instrucción "jump".

Tabla 2. Opcode de algunas instrucciones MIPS

Instrucción	Opcode (decimal)
slr	0
add	0
sub	0
j	2
beq	4
bne	5
addi	8
lw	35
sw	43

Tabla 1. Tabla de verdad de ALU Control (X = "don't care")

ALUOp	Campo "Funct"	Salida de ALU Control	Operación de ALU
00	.XXXXX	0010	Suma
X1	.XXXXX	0110	Resta
1X	.XX0000	0010	Suma
1X	.XX0010	0110	Resta
1X	.XX0100	0000	"Y" lógico
1X	.XX0101	0001	"O" lógico
1X	.XX1010	0111	"Set on less than"

**Pregunta 3. Pipeline (2 pts.)**

Considere un procesador MIPS con *pipeline* de 5 etapas, en el que, de todas las instrucciones ejecutadas, las siguientes fracciones de instrucciones tienen un tipo específico de dependencia de datos. El tipo de dependencia es identificado por la etapa que produce el resultado, EX o MEM, y la instrucción que requiere el resultado, es decir, la primera instrucción que sigue a aquella que produce el resultado, la segunda instrucción que sigue, o ambas. Asuma que el CPI del procesador **sin** riesgos de datos es 1.

Sólo EX a 1a instr.	Sólo MEM a 1a instr.	Sólo EX a 2a instr.	Sólo MEM a 2a instr.	EX a 1era instr. y MEM a 2da instr.	Otras
5%	20%	5%	10%	10%	10%

Además, asuma que se tienen las siguientes latencias, en ps, para las etapas del *pipeline*. Note que, para la etapa EX, se entregan distintas latencias dependiendo si el procesador tiene o no adelantamiento (ad) y de qué tipo.

IF	ID	EX (sin ad)	EX (con ad completo)	EX (ad sólo de EX/MEM)	EX (ad sólo de MEM/WB)	MEM	WB
150	100	120	150	140	130	120	100

- A) (0,4 pts.) En el caso sin adelantamiento, ¿qué fracción de ciclos corresponden a esperas debido a riesgos de datos?
- B) (0,4 pts.) En el caso con adelantamiento completo (adelantamiento de todos los resultados que se puedan adelantar), ¿qué fracción de ciclos corresponden a esperas debido a riesgos de datos?
- C) (0,4 pts.) Suponga que es muy costoso tener los multiplexores de 3 entradas necesarios para un adelantamiento completo. Se quiere decidir si es mejor adelantar sólo de EX/MEM (adelantamiento del siguiente ciclo) o sólo de MEM/WB (adelantamiento de 2 ciclos). ¿Qué caso resulta en menos ciclos de espera?
- D) (0,4 pts.) ¿Cuál es la aceleración lograda al incorporar adelantamiento completo al procesador con *pipeline* sin adelantamiento?
- E) (0,4 pts.) Suponga que se incorpora un “adelantamiento de viaje en el tiempo”, llamado McFly, que elimina **todos** los riesgos de datos. Asuma que la circuitería para el adelantamiento McFly agrega 100 ps a la latencia de la etapa EX con adelantamiento completo. ¿Es más rápido este procesador McFly que el procesador con adelantamiento completo? ¿Por cuánto?

A) Dependencias con la 1a instrucción (35% de los casos) generan 2 esperas (notar que si hay dependencia con la y 2a instrucción también requiere 2 esperas).

Dependencias sólo con 2a instrucción (15%) generan 1 espera

$$\text{Luego, } CPI = 1 + 35\% \times 2 + 15\% \times 1 = 1,85$$

$$\rightarrow \text{Porcentaje de esperas } \frac{0,85}{1,85} = 46\%$$

B) Con adelantamiento completo sólo genera 1 espera aquellos riesgos de MEM a la 1a instrucción (caso lw  $\rightarrow$  tipo R)

$$CPI = 1 + 20\% \times 1 = 1,2$$

$$\rightarrow \text{Porcentaje de esperas } \frac{0,2}{1,2} = 16,6\%$$

C)

- Con adelantamiento de EX/MEM, solo EX a 1a se resuelve ~~con~~ sin esperas, todas las otras dependencias requieren 1 espera.  

$$CPI = 1 + 20\% \times 1 + 5\% \times 1 + 10\% \times 1 + 10\% \times 1 = 1,45$$
- Con adelantamiento de MEM/WB: EX a 2a  $\rightarrow$  sin esperas  
 MEM a 1a  $\rightarrow$  1 espera  
 EX a 1a  $\rightarrow$  1 espera

$$CPI = 1 + 5\% \times 1 + 20\% \times 1 + 10\% \times 1 = 1,35$$

$\therefore$  Conviene adelantamiento de MEM/WB

D) En A), B) ya calculamos CPI

$$\begin{aligned} \text{Aceleración} &= \frac{\text{tiempo CPU s/ad}}{\text{tiempo CPU c/ad}} = \frac{CPI_{s/ad}}{CPI_{c/ad}} \quad \left( \begin{array}{l} \text{se cancela } \\ \# \text{ Instr.} \end{array} \right) \\ &= \frac{1,85}{1,20} = 1,54 \quad \left( \begin{array}{l} \text{tasa reloj,} \\ \text{pues son los mismos} \end{array} \right) \end{aligned}$$

E)

Notar que la etapa EX es la más lenta y por tanto define el tiempo de reloj mínimo.

$$\begin{aligned} \text{Tiempo por instrucción con adelantamiento} &= CPI_{c/ad} \times 150 \text{ ps} = 180 \text{ ps} \\ \text{McFly} &= CPI_{perfecto} \times 250 \text{ ps} = 250 \text{ ps} \end{aligned}$$

$$\therefore \underline{\text{No}} \text{ es más rápido el McFly. Aceleración} = \frac{180 \text{ ps}}{250 \text{ ps}} = 0,72$$