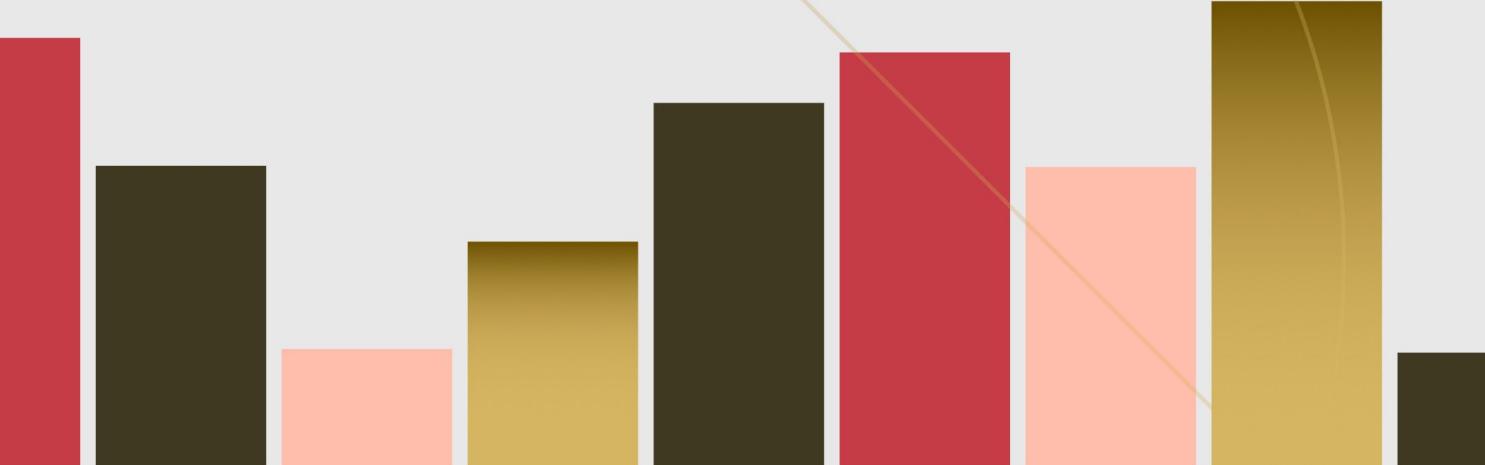


# EJERCICIOS RESUELTOS



QuesoViejo\_



**Problema 1**

Un mismo programa se ejecuta en dos computadores diferentes,

1.- ¿Qué CPU ejecuta el programa más rápido?

2.- ¿Cuánto más rápida es?

	T	CPI
CPU <sub>A</sub>	250 ps	2
CPU <sub>B</sub>	500 ps	1,2

Se supone que el programa ejecuta el mismo número de instrucciones máquina en las dos CPUs.

1.- Calculo los tiempos de ejecución

$$T_{CPU_A} = NI \cdot CPI_A \cdot T_A = NI \cdot 2 \cdot 250 = 500NI$$

$$T_{CPU_B} = NI \cdot CPI_B \cdot T_B = NI \cdot 1,2 \cdot 500 = 600NI$$

Como  $500NI < 600NI$ , entonces  $T_{CPU_A} < T_{CPU_B}$ , luego en la máquina A tarda menos tiempo

2.- Veamos cuánto más rápida es:

$$\frac{T_{CPU_B}}{T_{CPU_A}} = \frac{600NI}{500NI} = 1,2$$

La máquina A es 1,2 veces más rápida que la B

**Problema 2**

Un mismo programa → NI constante se ejecuta en dos máquinas diferentes. Sabemos que la máquina B necesita 1,2 veces más ciclos de reloj que la máquina A para ejecutarlo.

$$CPP_B = 1,2 \cdot CPP_A$$

1.- ¿Qué frecuencia de reloj tendría que tener la máquina B?

	F	T <sub>CPU</sub>
CPU <sub>A</sub>	2 GHz	10 s
CPU <sub>B</sub>	?	6 s

$$1. \quad T_{CPU_A} = \frac{CPP_A}{F_A} \quad T_{CPU_B} = \frac{CPP_B}{F_B}$$

$$CPP_A = T_{CPU_A} \cdot F_A$$

$$CPP_B = T_{CPU_B} \cdot F_B = 1,2 \cdot CPP_A$$

$$CPP_A = 10 \cdot 2 = 20$$

$$CPP_B = 6 \cdot F_B = 1,2 \cdot 20 = 24$$

Por lo tanto  $6 \cdot F_B = 24$

Luego

$$F_B = \frac{24}{6} = 4$$

**SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.**

**Problema 3**

Un diseñador de compiladores tiene que decidir entre dos secuencias de código para una máquina en particular. Esta máquina tiene tres tipos de instrucciones máquina diferentes: A, B, C. Las secuencias requieren el siguiente número de instrucciones:

	A	B	C	Total
Secuencia 1	2	1	2	5
Secuencia 2	4	1	1	6

Por otro lado, el CPI para cada tipo de instrucción es el siguiente:

	A	B	C
CPI	1	2	3

- 1.- ¿Qué secuencia de código ejecuta mayor número de instrucciones?
- 2.- ¿Qué secuencia de código es más rápida?
- 3.- ¿Cuál es el CPI<sub>MEDIO</sub> para cada secuencia?

1.- La secuencia 2 ejecuta más instrucciones que la secuencia 1

QuesoViejo\_

2.- Tendremos que calcular los tiempos de CPU de cada secuencia.

$$T_{CPU_1} = CPP_1 \cdot F$$

$$T_{CPU_2} = CPP_2 \cdot F$$

$$CPP_1 = \sum_{i=1}^C NI_i \cdot CPI_i$$

$$CPP_2 = \sum_{i=1}^C NI_i \cdot CPI_i$$

$$CPP_1 = 2 \cdot 5 + 5 \cdot 2 + 2 \cdot 3 = 10$$

$$CPP_2 = 4 \cdot 5 + 5 \cdot 2 + 1 \cdot 3 = 9$$

$$T_{CPU_1} = 10F$$

$$T_{CPU_2} = 9F$$

Por lo tanto  $9F < 10F$ , luego  $T_{CPU_2} < T_{CPU_1}$ , es decir, la secuencia 2 es más rápida.

QuesoViejo\_



3.-  $CPI_{global} = CPI_{medio} \circ CPI_{Global}$

$$CPI_{global1} = \frac{CPI_1}{N\bar{I}_1} = \frac{10}{5} = 2$$

$$CPI_{global2} = \frac{CPI_2}{N\bar{I}_2} = \frac{9}{6} = 1'5$$

QuesoViejo\_

QuesoViejo\_

### Problema 1

Sean dos implementaciones diferentes de un mismo repertorio de instrucciones. Este repertorio tiene cuatro tipos de instrucciones A, B, C y D donde:

	F	CPI <sub>A</sub>	CPI <sub>B</sub>	CPI <sub>C</sub>	CPI <sub>D</sub>
CPU <sub>1</sub>	1,5 GHz	1	2	3	4
CPU <sub>2</sub>	2 GHz	2	2	2	2

Sea un programa con  $10^6$  instrucciones donde el 10% es del tipo A, el 20% de tipo B, 50 % de tipo C y el 20% de tipo D.

- 1.- ¿Qué implementación es más rápida?
- 2.- ¿Cuál es el CPI<sub>GLOBAL</sub> de cada implementación?
- 3.- ¿Cuántos ciclos tarda el programa en cada implementación?

1.- Debemos calcular los tiempos de CPU de cada implementación:

$$T_{CPU_1} = \frac{NI \cdot CPI_{global1}}{F_1}$$

$$T_{CPU_2} = \frac{NI \cdot CPI_{global2}}{F_2}$$

$$CPI_{global1} = \frac{\sum_{i=1}^D NI_i \cdot CPI_{i,1}}{NI}$$

$$CPI_{global2} = \frac{\sum_{i=1}^D NI_i \cdot CPI_{i,2}}{NI}$$

$$CPI_{global1} = \frac{0'3 \cdot 10^6 \cdot 1 + 0'2 \cdot 10^6 \cdot 2 + 0'5 \cdot 10^6 \cdot 3 + 0'2 \cdot 10^6 \cdot 4}{10^6}$$

$$CPI_{global2} = \frac{2 \cdot 10^6 \cdot (0'3 + 0'2 + 0'5 + 0'2)}{10^6}$$

$$CPI_{global2} = 2$$

$$CPI_{global1} = 2'8$$

Por lo tanto:

$$T_{CPU_1} = \frac{10^6 \cdot 2'8}{1'5 \cdot 10^9} = 1'867 \cdot 10^{-3}$$

$$T_{CPU_2} = \frac{10^6 \cdot 2}{2 \cdot 10^9} = 10^{-3}$$

Como  $10^{-3} < 1'867 \cdot 10^{-3}$ ,  $T_{CPU_2} < T_{CPU_1}$ , luego la segunda implementación es más rápida.

SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.

2 = Calculadas en el apartado 1

$$CPI_{global1} = 2'8$$

$$CPI_{global2} = 2$$

3 = Calculamos los Ciclos por Programa (CPP)

$$T_{CPU_1} = \frac{CPP_1}{F}$$

$$T_{CPU_2} = \frac{CPP_2}{F}$$

$$CPP_1 = 5'867 \cdot 10^{-3} \cdot 5'5 \cdot 10^9$$

$$CPP_2 = 50^{-3} \cdot 2 \cdot 10^9$$

$$CPP_1 = 2800000$$

$$CPP_2 \approx 2^6$$

**Problema 2**

Sea un programa con los siguientes tipos de instrucciones y CPIs:

	Aritméticas (A)	Almacenamiento (B)	Carga (C)	Saltos (D)	Total
Nº instrucciones	500	50	100 <b>50</b>	50	700 <b>650</b>
CPI	1	5	5	2	

1.- ¿Cuál es el tiempo de ejecución en un procesador de **2 GHz**?

2.- Calcula el  $CPI_{global}$

\* 3.- Si el número de instrucciones de carga se reduce a la mitad ¿en qué factor se incrementa la velocidad? ¿cuál es el nuevo  $CPI_{global}$ ?

→ Frecuencia

$$1.- T_{CPU} = \frac{NI \cdot CPI_{global}}{F} =$$

$$CPI_{global} = \frac{\sum_{i=1}^D NI_i \cdot CPI_i}{NI} = \frac{500 \cdot 1 + 50 \cdot 5 + 100 \cdot 5 + 50 \cdot 2}{700} = 1'928$$

$$T_{CPU} = \frac{700 \cdot 1'928}{2 \cdot 10^9} = \frac{1350}{2 \cdot 10^9} = 6'75 \cdot 10^{-7}$$

2.- Calculado en el apartado 1

$$CPI_{global} = 1'928$$

3.- Vuelvo a calcular  $T_{CPU}$  y  $CPI_{global}$ ; los llamo igual añadiéndoles un \*

$$CPI_{global}^* = \frac{\sum_{i=1}^D NI_i^* \cdot CPI_i}{NI^*} = \frac{500 \cdot 1 + 50 \cdot 5 + 50 \cdot 5 + 50 \cdot 2}{650} = \frac{1500}{650} = 2'308$$

$$T_{CPU}^* = \frac{NI^* \cdot CPI_{global}^*}{F} = \frac{650 \cdot 2'308}{2 \cdot 10^9} = \frac{1500}{2 \cdot 10^9} = 5'5 \cdot 10^{-7}$$

$$\frac{T_{CPU}}{T_{CPU}^*} = \frac{6'75 \cdot 10^{-7}}{5'5 \cdot 10^{-7}} = 1'227 \Rightarrow \text{Ahora es } 1'227 \text{ veces más rápido}$$

**Problema 3**

Sean dos programas diferentes con el siguiente número de instrucciones:

	Aritméticas	Almacenamiento	Carga	Saltos
Programa 1	1000 500	400	100	50
Programa 2	1500 750	300	100	100
CPI	1 1 1 10 2 10 10 2 10 3 3 3			

Valores CPI en cada apartado

1.- Suponiendo que las instrucciones aritméticas necesitan 1 ciclo, las de carga y

almacenamiento 10 ciclos y los saltos 3 ciclos. ¿Cuál es el tiempo de ejecución de cada programa en un procesador de 3 GHz? → F

2.- Suponiendo que las instrucciones de aritméticas necesitan 1 ciclo, las de carga y almacenamiento 2 ciclos y los saltos 3 ciclos. ¿Cuál es el tiempo de ejecución de cada programa en un procesador de 3 GHz?

3.- Suponiendo que las instrucciones aritméticas necesitan 1 ciclo, las de carga y almacenamiento 10 ciclos y los saltos 3 ciclos. ¿Cuál es la aceleración de un programa si el número de instrucciones aritméticas se reduce a la mitad?

$$1.- T_{CPU_3} = \frac{CPI_3}{F} = \frac{\sum N_{I_3} \cdot CPI_3}{F} = \frac{1000 \cdot 1 + 400 \cdot 10 + 100 \cdot 10 + 50 \cdot 3}{3 \cdot 10^9} = \frac{6350}{3 \cdot 10^9} = 2'05 \cdot 10^{-9}$$

$$T_{CPU_2} = \frac{CPI_2}{F} = \frac{\sum N_{I_2} \cdot CPI_2}{F} = \frac{1500 \cdot 1 + 300 \cdot 10 + 100 \cdot 10 + 50 \cdot 3}{3 \cdot 10^9} = \frac{5800}{3 \cdot 10^9} = 1'93 \cdot 10^{-9}$$

$$2.- T_{CPU_3} = \frac{2150}{3 \cdot 10^9} = 7'16 \cdot 10^{-9}$$

$$T_{CPU_2} = \frac{2600}{3 \cdot 10^9} = 8'66 \cdot 10^{-9}$$

$$3.- T_{CPU_3} = \frac{5650}{3 \cdot 10^9} = 1'883 \cdot 10^{-9}$$

$$T_{CPU_2} = \frac{5050}{3 \cdot 10^9} = 1'683 \cdot 10^{-9}$$

**SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.**

**Problema 4**

Sea un programa con  $10^6$  instrucciones y dos procesadores diferentes:

	F	CPI <sub>global</sub>
CPU <sub>1</sub>	4 GHz	1,25
CPU <sub>2</sub>	3 GHz	0,75

1.- Es un error considerar que el procesador de frecuencia más elevada siempre va a tener mejores prestaciones. **¿Qué ocurre en este problema?**

2.- Otro error habitual se produce al utilizar la medida MIPS para comprobar las prestaciones de dos procesadores, y considerar que el procesador con el MIPS más elevado es siempre el que tiene mejores prestaciones. **¿Qué ocurre en este problema?**

$$1.- \text{Prestaciones} = \frac{1}{T_{\text{CPU}}}$$

$$T_{\text{CPU}_1} = \frac{NI \cdot CPI_{\text{global},1}}{F_1} = \frac{10^6 \cdot 1,25}{4 \cdot 10^9} = 0,3125 \cdot 10^{-3}$$

$$T_{\text{CPU}_2} = \frac{NI \cdot CPI_{\text{global},2}}{F_2} = \frac{10^6 \cdot 0,75}{3 \cdot 10^9} = 0,25 \cdot 10^{-3}$$

$$\text{Prestaciones}_1 = \frac{1}{0,3125 \cdot 10^{-3}} = 3,2 \cdot 10^3$$

Tiene mejores prestaciones el de frecuencia menor elevada

$$\text{Prestaciones}_2 = \frac{1}{0,25 \cdot 10^{-3}} = 4 \cdot 10^3$$

$$2.- \text{MIPS} = \frac{NI}{T_{\text{CPU}} \cdot 10^6}$$

$$\text{MIPS}_1 = \frac{10^6}{0,3125 \cdot 10^{-3} \cdot 10^6} = 3,2 \cdot 10^3$$

En este problema sí se cumple que el procesador con mayor MIPS tiene mejores prestaciones.

$$\text{MIPS}_2 = \frac{10^6}{0,25 \cdot 10^{-3} \cdot 10^6} = 4 \cdot 10^3$$

**Problema 5**

Un programa que se ejecuta en cuatro computadores diferentes.

NÚCLEOS	Nº de instrucciones por núcleo			CPI		
	Aritméticas	Carga/almac.	Saltos	Aritméticas	Carga/almac.	Saltos
COMPUTADOR <sub>1</sub>	1	2560	1280	256	1	4
COMPUTADOR <sub>2</sub>	2	1280	640	128	1	4
COMPUTADOR <sub>3</sub>	4	640	320	64	1	4
COMPUTADOR <sub>4</sub>	8	320	160	32	1	4

1.- ¿Cuál es el número total de instrucciones que se ejecutan en cada núcleo de cada computador?

2.- ¿Cuál es el número total de instrucciones que se ejecutan en cada computador?

3.- Si la frecuencia de cada núcleo es de 2 GHz. ¿Cuánto tarda en ejecutarse el programa en cada computador?

1.- Computador 1: 4096 Instrucciones/núcleo

Computador 2: 2048 Instrucciones/núcleo

Computador 3: 1024 Instrucciones/núcleo

Computador 4: 512 Instrucciones/núcleo

2.- 4096 en cada uno.

$$3.- \text{Computador 1: } T_{\text{cpu}_1} = \frac{\sum NI_i \cdot CPI_i}{F} = \frac{2056 \cdot 5 + 1280 \cdot 4 + 256 \cdot 1}{2 \cdot 10^9}$$

$$T_{\text{cpu}_1} = 3'716 \cdot 10^{-6} \text{ s}$$

$$\text{Computador 2: } T_{\text{cpu}_2} = \frac{1280 \cdot 5 + 640 \cdot 4 + 128 \cdot 1}{2 \cdot 10^9} = 1984 \cdot 10^{-6} \text{ s}$$

$$\text{Computador 3: } T_{\text{cpu}_3} = \frac{640 \cdot 5 + 320 \cdot 4 + 64 \cdot 1}{2 \cdot 10^9} = 9192 \cdot 10^{-7} \text{ s}$$

$$\text{Computador 4: } T_{\text{cpu}_4} = \frac{320 \cdot 5 + 160 \cdot 4 + 32 \cdot 1}{2 \cdot 10^9} = 4'96 \cdot 10^{-7} \text{ s}$$

**Problema 6**

Un programa que se ejecuta en dos computadores diferentes.

	NÚCLEOS	Nº de instrucciones por núcleo				CPI			
		Aritméticas Ent.	Aritméticas PF	Carga/ Almacen.	Saltos	Aritméticas Ent.	Aritméticas PF	Carga/ Almacen.	Saltos
COMPUTADOR 1	1	$2000 * 10^6$	$560 * 10^6$	$1280 * 10^6$	$256 * 10^6$	$1 \left  \begin{array}{l} 0'6 \\ 0'6 \end{array} \right.$	$1 \left  \begin{array}{l} 0'6 \\ 0'6 \end{array} \right.$	$4 \left  \begin{array}{l} 2'8 \\ 2'8 \end{array} \right.$	$2 \left  \begin{array}{l} 3'4 \\ 3'4 \end{array} \right.$
COMPUTADOR 2	8	$240 * 10^6$	$80 * 10^6$	$160 * 10^6$	$32 * 10^6$	$1 \left  \begin{array}{l} 0'6 \\ 0'6 \end{array} \right.$	$1 \left  \begin{array}{l} 0'6 \\ 0'6 \end{array} \right.$	$4 \left  \begin{array}{l} 2'8 \\ 2'8 \end{array} \right.$	$2 \left  \begin{array}{l} 3'4 \\ 3'4 \end{array} \right.$

Si la frecuencia de cada núcleo es de 2 GHz ¿Cuánto mejora el tiempo de ejecución si el CPI de las instrucciones ENT y PF se reduce en un 40% y EL CPI de las instrucciones CARGA/ALMAEN. y SALTO se reduce en un 30%?

Marco con un asterisco (\*) los tiempos al cambiar los CPI

$$T_{CPU_1} = \frac{\sum NI_i \cdot CPI_i}{F} = \frac{8192 \cdot 10^6}{2 \cdot 10^9} = 4'096 s$$

$$T_{CPU_1}^* = \frac{\sum NI_i \cdot CPI_i}{F} = \frac{54784 \cdot 10^6}{2 \cdot 10^9} = 2'7392 s$$

$$\frac{T_{CPU_1}}{T_{CPU_1}^*} = 1'4953 \Rightarrow \text{Ahora es } 1'4953 \text{ veces más rápido.}$$

$$T_{CPU_2} = \frac{\sum NI_i \cdot CPI_i}{F} = \frac{1024 \cdot 10^6}{2 \cdot 10^9} = 0'512 s$$

$$T_{CPU_2}^* = \frac{\sum NI_i \cdot CPI_i}{F} = \frac{640 050 268'8}{2 \cdot 10^9} = 0'32 s$$

$$\frac{T_{CPU_2}}{T_{CPU_2}^*} = 1'6 \Rightarrow \text{Ahora es } 1'6 \text{ veces más rápido}$$

SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.

**Problema 1**

Sea una caché de correspondencia directa con 16KB de datos y bloques de 4 palabras. La palabra es de 32 bits.

- 1.- ¿Cuántos bloques tendrá la caché?
- 2.- ¿Cuántos bits tendrá el índice?
- 3.- ¿Cuántos bits tendrá el campo etiqueta?
- 4.- ¿Cuál es el tamaño total de la caché? Aquí hay que tener presente tanto la capacidad destinada a datos como la destinada a etiquetas.

1)  $n^{\circ} \text{ bloques} = \frac{\text{Capacidad caché}}{\text{Capacidad bloque}} = \frac{16 \text{ KB}}{4 \cdot 32 \text{ b}} = \frac{16 \cdot 2^{10} \text{ B}}{16 \text{ B}}$

$n^{\circ} \text{ bloques} = 2^{10} \text{ Bloques}$

2) Como tenemos  $2^{10}$  Bloques necesitamos 10 bits de índice

3) Índice = 10b

Block offset : 1 Bloque = 4 palabra = 2 b

Byte offset = 1 Palabra = 32b = 4 B =  $2^2$  B = 2b para dirección

bits etiqueta =  $32 - 10 - 2 \cdot 2 = 18 \text{ b}$

4) Para datos : 16kB

Para etiquetas: Tamaño etiqueta  $\cdot$  nBloques =

$$18 \text{ b} \cdot 2^{10} = 18432 \text{ B} = 2304 \text{ B}$$

Total: 16kB + 2304B = 18688B

## Problema 2

Sea una caché de correspondencia directa con 64 bloques, donde el tamaño del bloque es de una palabra (palabra = 32 bits).

1.- ¿Qué capacidad tendrá cada bloque de la memoria caché?

2.- ¿Cuántos bits tendrá el índice?

3.- ¿Cuántos bits tendrá el campo etiqueta de la memoria caché?

4.- Se está enviando un bloque de memoria principal a memoria caché: ¿En qué posición de la memoria caché se debería emplazar el bloque si la dirección del primer byte en memoria principal es  $1200_{10}$ ?

5.- La dirección  $1200_{10}$  tiene 32 bits porque el bus de direcciones tiene 32 bits. Escribe esos 32 bits e identifica cada una de las partes de la dirección: etiqueta, índice, BLOCKoffset y BYTEoffset. ¿El valor que has obtenido en la parte que identifica el índice coincide con el obtenido en la cuestión 4?

6.- Se está enviando un bloque de memoria principal a memoria caché: ¿En qué posición de la memoria caché se debería emplazar el bloque si la dirección de ese bloque en memoria principal es  $9_{10}$ ?

$$1) 1 \text{ palabra} = 32 \text{ b} = 4 \text{ B}$$

$$2) 64 \text{ bloques} = 2^6 \text{ bloques} \Rightarrow 6 \text{ b índice}$$

$$3) \text{Índice: } 6 \text{ b}$$

$$\text{Block offset} = 0$$

$$\text{Byte offset: } 1 \text{ palabra} = 4 \text{ B} = 2^2 \text{ B} = 2 \text{ b}$$

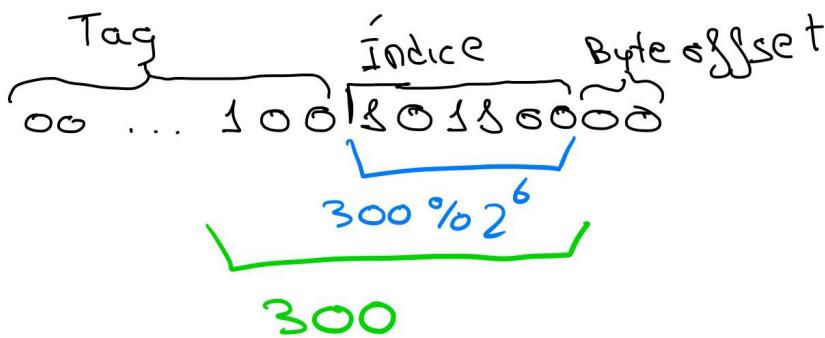
$$\text{Bits etiqueta} = 32 - 6 - 2 = 24 \text{ b}$$

$$4) \text{Dirección del Bloque} = \frac{\text{Dirección MP}}{\text{Bytes por Bloque}} = \frac{3208}{4} = 300$$

$$\text{Posición caché} = \text{Dir. Bloque \% n\# Bloques} = 300 \% 64 = 44$$

#### 4) Otra manera de calcularlo

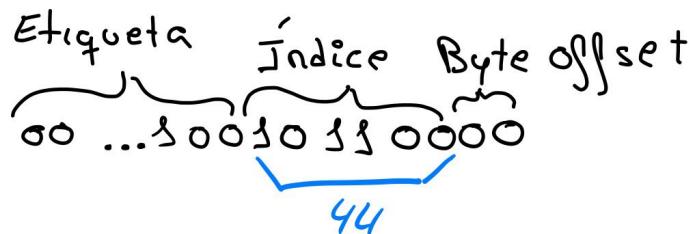
$$\begin{array}{r}
 5200_{10} \\
 \text{---} \\
 0 \quad 600 \quad | \quad 2 \\
 | \quad 300 \quad | \quad 2 \\
 | \quad 150 \quad | \quad 2 \\
 | \quad 75 \quad | \quad 2 \\
 | \quad 1 \quad 37 \quad | \quad 2 \\
 | \quad 18 \quad | \quad 2 \\
 | \quad 9 \quad | \quad 2 \\
 | \quad 4 \quad | \quad 2 \\
 | \quad 2 \quad | \quad 2 \\
 | \quad 0 \quad | \quad 1 \\
 \hline
 \end{array}$$



$$\text{Índice} = 10011000_2 = 44_{10}$$

$$5) \quad \$200_{10} =$$

Coincide



$$6) \quad \text{Bloque en caché} = \frac{\text{Dir Bloque en MP}}{\text{nº bloques caché}} \quad \text{MOD}$$

$$= 9_{10} \% 44_{10} = 9_{10}$$

Se coloca en la posición  $9_{10} = 001001_2$  (este sería el índice)

QuesoViejo\_

**SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.**

**Problema 3**

Sea una caché de correspondencia directa con 64 bloques, donde el tamaño del bloque es de cuatro palabras (palabra = 32 bits).

- 1.- ¿Qué capacidad tendrá cada bloque de la memoria caché?
- 2.- ¿Cuántos bits tendrá el índice?
- 3.- ¿Cuántos bits tendrá el campo etiqueta de la memoria caché?
- 4.- Se está enviando un bloque de memoria principal a memoria caché: ¿En qué posición de la memoria caché se deberíaemplazar el bloque si la dirección del primer byte en memoria principal es  $1200_{10}$ ?
- 5.- La dirección  $1200_{10}$  tiene 32 bits porque el bus de direcciones tiene 32 bits. Escribe esos 32 bits e identifica cada una de las partes de la dirección: etiqueta, índice, BLOCKoffset y BYTEoffset. ¿El valor que has obtenido en la parte que identifica el índice coincide con el obtenido en la cuestión 4?
- 6.- Se está enviando un bloque de memoria principal a memoria caché: ¿En qué posición de la memoria caché se deberíaemplazar el bloque si la dirección de ese bloque en memoria principal es  $9_{10}$ ?

Caché de 64 bloques. 1 bloque = 4 palabras

\* Si no se dan datos concretos, 1 palabra = 32b

se puede suponer que el ancho del bus de direcciones coincide con el de datos

$$1.- \text{ 4 palabras} \cdot \frac{32b}{1 \text{ palabra}} = 128b = 16B$$

2.- Hay 64 bloques  $\Rightarrow$  64 elementos que direccionar  $= 2^6$  elementos  $\Rightarrow$  6 bits de índice

3.- La etiqueta son los bits del bus de direcciones que no se usan ni para índice ni para byte / block offset

Etiqu	Índice	Block offset	Byte offset
-------	--------	--------------	-------------

$$\text{Etiqu} = 32 - 6 - 2 - 2 = 22$$

6 2 2  
QuesoViejo\_

Block offset:  
1 bloque = 4 palabras  $= 2^2 = 2^2$   
1 palabra = 4 bytes  $= 2^2 = 2^2$



4.- Dir primer byte = \$200<sub>16</sub> →

Paso 5: Obtener la dirección del bloque

Aplicamos la fórmula:

$$\text{Dirección Bloque} = \frac{\text{Dirección Byte}}{\text{Nº Bytes del bloque}}$$

$$\text{Dir bloque} = \frac{\$200}{4 \cdot 4}$$

→ dirección a nivel Byte  
→ nº de Bytes de 1 palabra  
Nº palabras de s' bloque

$$\text{Dir bloque} = 75$$

Paso 2: Obtener el índice

$$75 \bmod 64 = 11 \rightarrow \text{Índice}$$

5.- \$200 → 10010110000<sup>10</sup>  
1200-1024 = 176  
176-128 = 48  
48-32 = 16  
16 - 16 = 0

Etiqueta	Índice	Block offset	Byte offset
00...	01 001011	00	00

11  
11

6 = Dir bloque = 9<sub>10</sub>

$$\text{Índice} = 9 \bmod 64 = 9$$

**SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.**

**Problema 4**

El procesador tiene que acceder a estas direcciones del sistema de memoria y en este orden:

22, 26, 22, 26, 16, 3, 16, 18

$\rightarrow 2 \rightarrow 3 \rightarrow$  índice e

Sea una caché de correspondencia directa con 8 bloques de un byte. La caché inicialmente está vacía.

Indique, para cada una de estas direcciones, la dirección en binario sabiendo que son direcciones de 5 bits, la etiqueta y el índice. Indique también si en el acceso se produce un fallo o un acierto. Cuando se produzca un fallo hay que especificar el motivo.

Dirección decimal	Dirección binaria	Etiqueta	Índice	F/A
22	10110	10	110	F
26	11010	11	010	F
22	10110	10	110	A
26	11010	11	010	A
16	10000	10	000	F
3	00011	00	011	F
16	10000	10	000	A
18	10010	10	010	F

v=0 Caché vacío  
 v=0 Caché vacío  
 v=1 y coincide con etiqueta  
 } Caché vacío v=0  
 -> Índice si, etiqueta no

Para comprobar si hay acierto, se comprueba en el bloque de caché que direcciona el índice si la etiqueta coincide o no.

El bit V en la memoria caché directa indica si el dato en ese bloque de caché es un dato válido ( $V=1$ ) o no ( $V=0$ )

Al principio la caché está vacía  $\Rightarrow$  Dato no válido

Ojo:

La caché no está "vacía", sino que los valores que tiene son basura y no sirven para nada.

### Problema 5

El procesador Intel Core i7 tiene 4 núcleos Nehalem. En este problema vamos a analizar la caché L1 de instrucciones del procesador Nehalem. Los datos que necesitamos para realizar el problema son:

- Capacidad = 32 KB
- Tamaño del bloque = 64 bytes
- Tipo de caché: asociativa por conjuntos de 4 vías
- Tamaño de palabra = 64 bits
- Direccionamiento a nivel de byte

- 1.- ¿Cuántos índices tendrá la caché?
- 2.- ¿Cuántos bits de la dirección se destinan al índice?
- 3.- ¿Cuántos bits de la dirección se destinan a la etiqueta?

Etiqueta	Índice	Block offset	Byte offset
----------	--------	--------------	-------------

$$1.- \text{Nº Índices} = \text{Nº conjuntos} = \frac{\text{Nº Bloques}}{4} = \frac{\frac{\text{Capacidad}}{\text{Tam bloq}}}{4}$$

$$\frac{32 \cdot 2^{16}}{64} = \frac{2^9}{2^2} = 2^7$$

Método hecho en clase:

$$\text{Nº índices} = \frac{\text{capacidad caché}}{\text{Capacidad conjunto}} = \frac{\text{capacidad caché}}{\text{Capacidad bloque} \cdot \text{Nº vías}}$$

$$= \frac{32 \text{ kB}}{64 \text{ B} \cdot 4} = \frac{32 \cdot 2^{16}}{64 \cdot 4} = 2^7$$

$$2.- 7$$

$$3.- \text{Índice} = 7b$$

$$\text{Block offset: } \frac{\text{Nº pal}}{\text{Bloque}} = \frac{\text{Tamaño Bloque}}{\text{Tamaño Palabra}} = \frac{64 \cdot 8b}{64b} = 8$$

Queso Viejo  $\Rightarrow 3b$  de Block offset

Byte offset: 3 palabras =  $64b = 8B = 2^3B = 3$  b byte offset

Bits Etiqueta = Bits bus direcciones - bits Indice - bits block offset  
- bits Byte offset

Suponemos ancho bus direcciones = ancho bus datos

Bits etiqueta =  $64 - 7 - 3 - 3 = 51$  bits para la etiqueta

**Problema 6**

Sea la caché L3 del procesador AMD Operon: 2MB de capacidad de datos, asociativa por conjuntos de 32 vías, bloques de 64 bytes. Palabra de 64 bits.

1.- ¿Cuántos bits tendrá el índice?

2.- ¿Cuántos bits tendrá el campo etiqueta de la memoria caché?

3.- Se está enviando un bloque de memoria principal a memoria caché: ¿En qué posición de la memoria caché se debería emplazar el bloque si la dirección del primer byte en memoria principal es  $2048_{10}$ ?

4.- Verifica si el índice obtenido en la cuestión 3 se corresponde con el índice extraído de la dirección  $2048_{10}$ .

5.- ¿Cuál es el tamaño total de la caché? *Cap dato + Cap etiqueta (no tendríamos en cuenta el bit V)*

Todas las respuestas deben estar justificadas.

$$1.- \text{Capacidad caché} = 2 \text{ MB} \quad \text{Nvías} = 32$$

$$\text{Tamaño Bloque} = 64 \text{ B} \quad \text{Tamaño palabra} = 64 \text{ b}$$

$$N^{\circ} \text{ Indicadores} = N^{\circ} \text{ Conjuntos} = \frac{\text{Capacidad caché}}{\text{Capacidad conj}} = \frac{\text{Capacidad caché}}{\text{Tam Bloq} \cdot \text{Nvías}}$$

$$= \frac{2 \text{ MB}}{32 \cdot 64 \text{ B}} = \frac{2 \cdot 2^{20} \text{ B}}{2^5 \cdot 2^6 \text{ B}} = 2^{10} \text{ índices}$$

$$\text{Bits índice} = 10 \text{ bits}$$

$$2.- \text{Bits etiq} = \text{Bits total} - \text{bits resto de cosas}$$

$$\frac{N^{\circ} \text{ Palabras}}{\text{Bloque}} = \frac{\text{Tam Bloque}}{\text{Tam palabra}} = \frac{64 \cdot 8 \text{ b}}{64 \text{ b}} = 8 = 2^3 \text{ bits para dirección}$$

$$\frac{N^{\circ} \text{ Bytes}}{\text{Palabra}} = \frac{\text{Tam palabra}}{\text{Tam Byte}} = \frac{64 \text{ b}}{8 \text{ b}} = 8 = 2^3 \text{ bits para direccionar}$$

$$\text{Bits etiq} = 64 - 50 - 3 - 3 = 48 \text{ bits}$$

3.- Dirección primer Byte =  $2048_{10}$

$$\text{Dir Bloque en MP} = \frac{2048}{64} = \frac{2^9}{2^6} = 2^3$$

$$\text{Índice} = 2^5 \bmod 2^6 = 2^5$$

4)  $2048_{10} = 100000|0000000_2$   
0000 100000

$$\text{Dir Bloque en MP} = 32 = 2^5$$

$$\text{Índice} = 2^5 \bmod N^{\circ} \text{ conjuntos caché} = 2^5 \bmod 2^{10} = 2^5$$

Se mete el bloque en el índice 32, es decir, el conjunto 32.

Dentro de ese conjunto, hay 32 bloques (32 vías). Se situará en un bloque u otro de ese conjunto según el algoritmo empleado (aleatorio, última vez usado (LRU)...)

5) Capacidad etiquetas =  $N^{\circ} \text{ Bloques Caché} \cdot \text{Tamaño etiqueta}$

$$N^{\circ} \text{ Bloques Caché} = N^{\circ} \text{ Conjuntos} \cdot N^{\circ} \text{ vías} = 2^{10} \cdot 32 = 2^{15}$$

$$\text{Capacidad etiquetas} = 2^{15} \cdot 48b = 2^{12} \cdot 48B$$

Tamaño caché = Capacidad etiquetas + Capacidad datos

$$= 2MB + 2^{12} \cdot 48B = 2240 KB$$

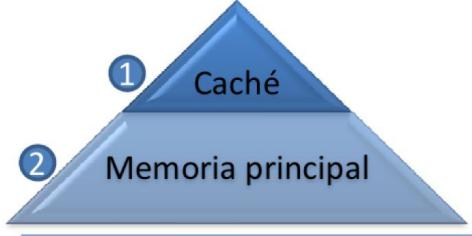
QuesoViejo\_

Problema 7

El tiempo de acceso de la memoria caché a la memoria principal es 70ns.

1.- Suponiendo que el tiempo de acierto de la caché determina la duración del ciclo de reloj. ¿Cuáles son las frecuencias de reloj de cada CPU?

2.- ¿Cuál es el tiempo medio de acceso a memoria en cada CPU? Expresarlo en nanosegundos y ciclos.



	Tamaño caché	Frecuencia de fallos caché	Tiempo de acierto caché
CPU1	1KB	11,4%	0,62ns
CPU2	2KB	8,0%	0,66ns
CPU3	8KB	4,3%	0,96ns
CPU4	16KB	3,4%	1,08ns

	Tamaño caché	Frecuencia de fallos caché	Tiempo de acierto caché	Frecuencia reloj	Tiempo acceso	Ciclos acceso
CPU1	1KB	11,4%	0,62ns	$1/0,62\text{ns} = 1,616 \text{ GHz}$	<del>86 ns</del>	<del>86</del>
CPU2	2KB	8,0%	0,66ns	$1/0,66\text{ns} = 1,515 \text{ GHz}$	$70 \cdot 0,08 + 0,66 = 626$	<del>70</del>
CPU3	8KB	4,3%	0,96ns	$1/0,96\text{ns} = 1,042 \text{ GHz}$	$70 \cdot 0,043 + 0,66 = 367 \text{ ns}$	<del>70</del>
CPU4	16KB	3,4%	1,08ns	$1/1,08\text{ns} = 0,926 \text{ GHz}$	$70 \cdot 0,034 + 0,66 = 3,64 \text{ ns}$	<del>70</del>

$$\text{Periodo: } T \quad \text{Frecuencia: } F = \frac{1}{T}$$

$T = T_{acierto}$  (porque lo dice el enunciado)

$$T_{acceso} = N^{\circ} \text{ ciclos} \cdot \text{Periodo} \quad \Leftrightarrow \quad T_{acierto} + \text{Prob fallo} \cdot T_{acceso}$$

$$\text{CPU}_3 : \text{Ciclos acceso MP} = \frac{\text{Tiempo acceso MP}}{\text{Periodo}(T)} = \frac{70\text{ns}}{0,62} =$$

$$= 112,9 \Rightarrow 113 \text{ ciclos}$$

$$\begin{aligned} \text{ciclos promedio} &= \text{ciclos caché} + \text{Prob fallo} \cdot \text{ciclos MP} \\ &= 3 + 0,384 \cdot 113 = 131,882 \text{ ciclos} \end{aligned}$$

$$T_{acceso} = 131,882 \cdot 0,62\text{ns} = 81,6 \text{ ns}$$

SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.

$$CPU_2: F = \frac{1}{0'66 \cdot 50^9} = 1'53 \cdot 50^9 \text{ Hz} = 1'53 \text{ GHz}$$

$$\text{Ciclos acceso a MP} = \frac{70}{0'66} = 106'06 \rightarrow 107 \text{ ciclos}$$

$$Nº \text{ ciclos promedio acceso} = \frac{\text{Ciclos acceso cache}}{\text{cache}} + \frac{\text{Penalización fallo de cache}}{\text{Periodo}}$$

$$\text{Penalización fallo cache} = \text{Prob fallo} \cdot \text{ciclos acceso a MP}$$

$$Nº \text{ ciclos promedio} = 106'08 \cdot 107 = 9'56 \xrightarrow{\text{Ciclos promedio que tarda al ser un promedio puede dar decimales}} \text{Si hablamos de ciclos concretos se redondea hacia arriba}$$

↑ Periodo

$$\text{Tiempo promedio} = T \cdot 9'56 = 0'66 \cdot 9'56 = 6'33 \text{ ns}$$

CPU 3 :

$$\text{Ciclos acceso MP} = \frac{T_{\text{acceso}} \cdot MP}{\text{Periodo}(T)} = \frac{70 \text{ ns}}{0'96 \text{ ns}} = 72'93 \downarrow \\ 73$$

$$\text{Ciclos promedio} = \text{Ciclos acierto} + \text{Prob fallo} \cdot \text{Ciclos fallo}$$
$$106'043 \cdot 73 = 4'139$$

$$T_{\text{promedio}} = \text{Ciclos promedio} \cdot \text{Periodo} = 4'139 \cdot 0'96 = 3'97$$

**Problema 8**

Sea una memoria caché asociativa por conjuntos con las siguientes características:

- Capacidad = 512 KB
- Tamaño del bloque = 64 bytes
- Tipo de caché: asociativa por conjuntos de 16 vías
- Tamaño de palabra = 64 bits = 8 Bytes
- Direccionamiento a nivel de byte

*Sí no dan ancho bus de direcciones suponemos que es igual que el de datos*

1.- ¿Cuántos bits de la dirección se destinan al índice?

2.- ¿Cuántos bits de la dirección se destinan a la etiqueta?

3.- Se está enviando un bloque de información de memoria principal a memoria caché, ¿en qué posición de la memoria caché se debería emplazar el bloque si la dirección de su primer byte en memoria principal es  $1024_{10}$ ?

4.- Verifica si el índice obtenido en la cuestión anterior coincide con el extraído de la dirección  $1024_{10}$

5.- Se está enviando un bloque de información de memoria principal a memoria caché, ¿en qué posición de la memoria caché se debería emplazar el bloque si la dirección de ese bloque en memoria principal es  $10_{10}$ ?

$$1.- \text{ 1 conjunto} = 56 \text{ bloques} \quad \text{Capacidad 1 conjunto} = 16 \cdot 64B = 2^{10}B$$

$$\text{Nº Conjuntos} = \frac{\text{Capacidad total}}{\text{Capacidad un conjunto}} = \frac{2^9 \cdot 2^{10}B}{2^{10}B} = 2^9 \text{ conjuntos}$$

$$2^9 \text{ conjuntos} \Rightarrow 9 \text{ bits para el índice}$$

$$2.- \text{ Bits etiqueta} = \text{Bits totales} - \text{Bits índice} - \text{Bits block offset} \\ - \text{Bits Byte offset}$$

$$\text{Bits block offset} : 1 \text{ bloque} = 64 \text{ Bytes}$$

$$1 \text{ palabra} = 64 \text{ bits}$$

$$\text{Nº palabras por bloque} = \frac{\text{capacidad bloque}}{\text{capacidad palabra}} = \frac{2^6 \cdot 2^3}{2^6} = 2^3 = 8 \text{ bits}$$

$$\text{Byte offset} : 1 \text{ palabra} = 64b = 8B = 2^3B = 3 \text{ bits}$$

$$\text{Bits etiqueta} = 64 - 9 - 3 - 3 = 49 \text{ bits}$$

$$3.5 \text{ Dir primer byte} = 5024_{10} \quad \text{Dir Bloque} = \frac{5024}{64} = 78$$

Nº Bytes en 1 bloque ✓

Aplicando la fórmula: Índice = Dir bloque MOD nº conjuntos

$$\text{Índice cache} = 78 \text{ MOD } 2^4 = 2^4 \text{ MOD } 2^4 = 2^4$$

$$\text{Índice} = 5024 \text{ MOD}$$

$$5024_{10} = \underbrace{1\ 5\ 0\ 0\ 0\ 0}_{2^4=16} | 0\ 0\ 0 | 0\ 0\ 0$$

Etiqueta	Índice	Block	Byte
49b	9b	3b	3b

Quiero índice. Quito Block y Byte al dividir entre  $2^6$

Quito Etiqueta con MOD  $2^9$

En este caso coincide porque los bits de etiqueta están a 0

SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.

# Tema 3

## Problema 1

Se va a ejecutar la instrucción 0x012A5825.

- 1.- ¿Qué entradas recibe la unidad de control?
- 2.- ¿Qué valor/es tiene que decodificar la unidad de control?
- 3.- ¿Qué salidas proporciona la unidad de control?
- 4.- ¿Para qué sirven esas salidas?
- 5.- ¿Qué valores deben tener esas salidas para que se ejecute correctamente la instrucción?

0x052A5825

1. - 0000 0000 0040 1010 | 0100 1000 0000 0101  
2. - Aritmético-lógica \$9 \$10 \$11 0 OP = 37 => Or

Instrucción	OP	RS	R2	R3	Shamt	Función				
	6	5	5	5	5	6	ALU operation	MemRead	MemWrite	
OR	37	5=0	1	1	0	0	1	0	0	

3. - (Mirar en tabla) 0001  
4. - Escribe en reg  
    Usa reg destino  
    Accede directo al reg  
    No es un Salto

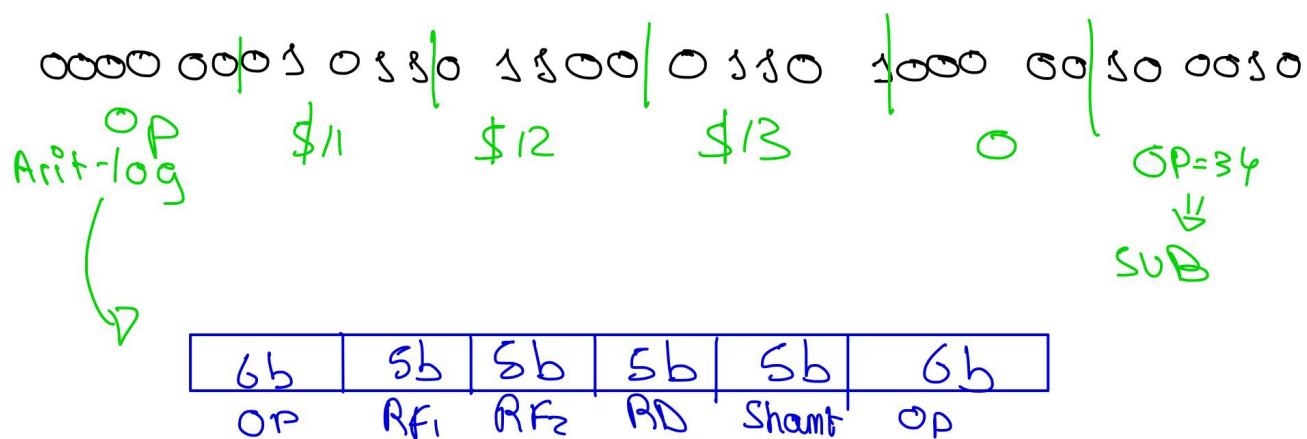
## Problema 2

Se va a ejecutar la instrucción 0x016C6822.

- 1.- ¿Qué tipo de instrucción es?
- 2.- ¿Cuál es el número de identificación del registro destino?
- 3.- ¿Utiliza algún registro como registro fuente? En tal caso, averigua el número de identificación.
- 4.- ¿Qué valores deben tener las señales de control para que se ejecute correctamente la instrucción?

0x016C6822

10



Instrucción	Código operación	Código función	RegWrite	RegDst	RegSrc	PC src 1	PC src 2	ALU src	ALU operation	MemRead	MemWrit
4.-	1	1	1	0	-	-	1	0110 (Tabla)	0	0	0

→ (No afecta)

1.- Sub

2.- 13

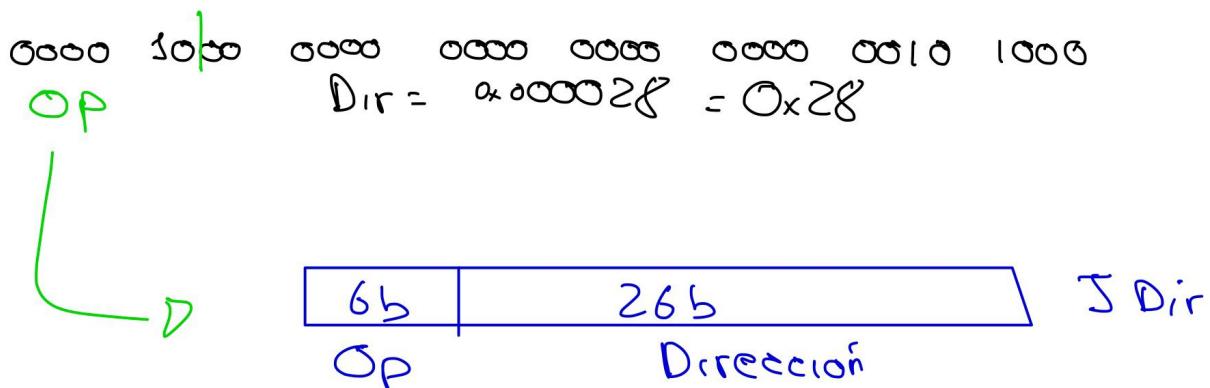
3.- Sí, \$1 y \$2

### Problema 3

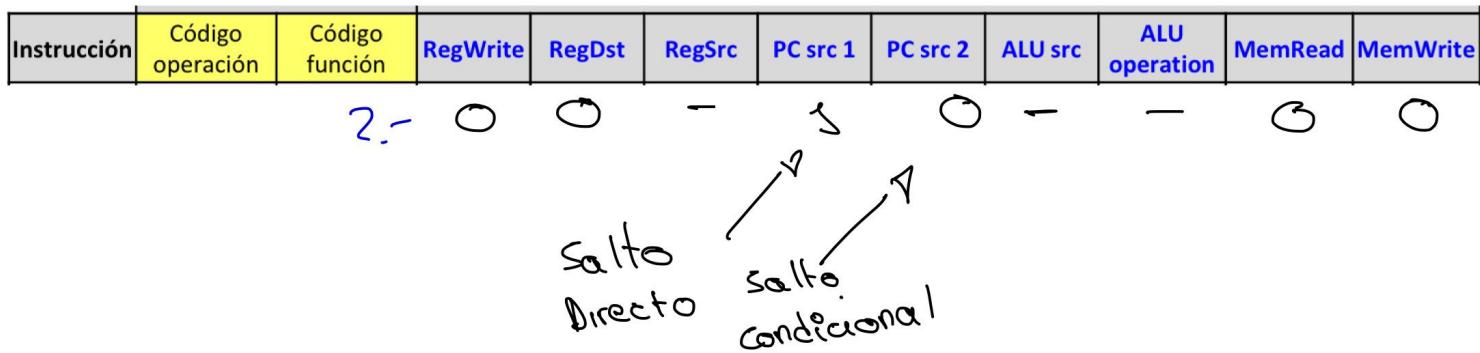
Se va a ejecutar la instrucción 0x08000028.

- 1.- Escribe la instrucción en código ensamblador.
- 2.- ¿Qué valores deben tener las señales de control para que se ejecute correctamente la instrucción?
- 3.- Si antes de ejecutarse la instrucción el contador del programa tiene el valor 0xA001E020 ¿Qué valor tendrá después de ejecutarse la instrucción?

0x08000028



1.- Instrucción: J 0x28



3.- Instrucción : 32b , dirección : 32b

Dirección Instrucción : 26b

Dirección final : 4b PC | 26b instr | 00z

Dirección final 1010|00 0000 0000 0000 0000 00101000 00  
A 0 0 0 0 0 0 0 0 A 0

QuesoViejo

**SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.**

Al final PC vale : 0xA00000A0

SS<sub>16</sub> = 05035

ZD<sub>10</sub> = 10100

OR: 55551

S  
AMOS  
NAL  
GUIR



US  
NC Y SELENIO

**Problema 4**

Se va a ejecutar la instrucción 0x8D4F0008.

1.- Escribe la instrucción en código ensamblador.

2.- ¿Qué valores deben tener las señales de control para que se ejecute correctamente la instrucción?

0x8D4F0008 =

1000 11010100 11110000 0000 0000 0000

Los 6 primeros bits nos indican el código de operación

100011 =&gt; Instrucción LW

La estructura de esta instrucción es

Cod operación	Reg Dest	Reg Fuente	Inmediato
6b	5b	5b	16b
100011	01010	01111	0000 0000 0000 1000 $(10_{10}) \Rightarrow \$t2$ $(15_{10}) \Rightarrow \$t7$

La instrucción en código ensamblador es:

LW \$t2, 0x0008(\$t7)

o se puede dejar así: LW \$t2, 0x0008(\$t7)

Cod instrucción: 100011 Cod función: -

QuesoViejo\_

Los valores de las señales de control serían:

RegWrite	RegDst	RegSrc	PCsrc3	PCsrc2	ALUSrc	ALUOP	MEMWr	MEMRead
1	0	1	0	0	1	0050 (suma)	0	1

SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.

# Tema 4 Procesador Segmentado

## Problema 1

Sea la siguiente secuencia de instrucciones:

ADD \$8, \$9, \$10 } Dependencia o riesgo de datos,  
LW \$11, 4(\$8)  
LW \$12, 8(\$8) RAW (Read after write)

Contenido de ciertos registros y posiciones de memoria:

\$8 = 0  
\$9 = 20  
\$10 = 12  
MEM[36] = 200  
MEM[40] = 100

1.- Indica las dependencias que existen entre las instrucciones.

2.- Supongamos que esta secuencia de instrucciones se ejecuta en un procesador MIPS-32 segmentado CON unidad de detección de riesgos (bloqueos) y unidad de anticipación . Representar en un diagrama qué sucede en cada ciclo de ejecución.

Los valores obtenidos en una fase hay que guardarlos en los reg de segmentación (la siguiente barra azul) ya que si no otra instrucción podría cambiarlos en el siguiente ciclo.

Los llamamos ID/EX\$, TD/EX?, EX/MEM\$, EX/MEM? ...

Al meter un valor en un registro

\$3	28 / 50	50
EX/MEM\$	43	20
	\$3 ← 50 EX/MEM\$ ← 20	

En los registros del banco de registros el valor está disponible en la segunda mitad del ciclo .

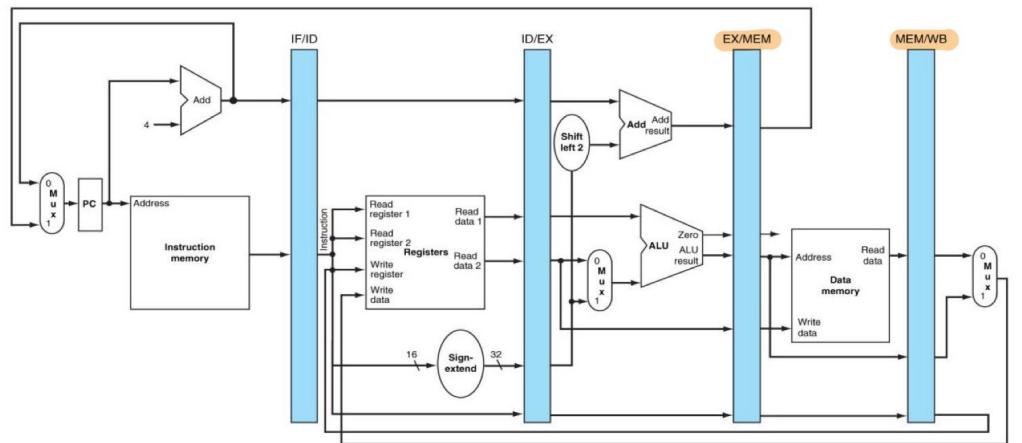
En los de segmentación no .

## PLANTILLA PARA EJERCICIOS DE SEGMENTACIÓN:

Registro	Contenido
\$8	0
\$9	20
\$10	12

Posición de memoria	Contenido
[36]	200
[40]	100

Método utilizado para resolver los riesgos de datos	
Bloqueo?	Sí
Anticipación?	Sí



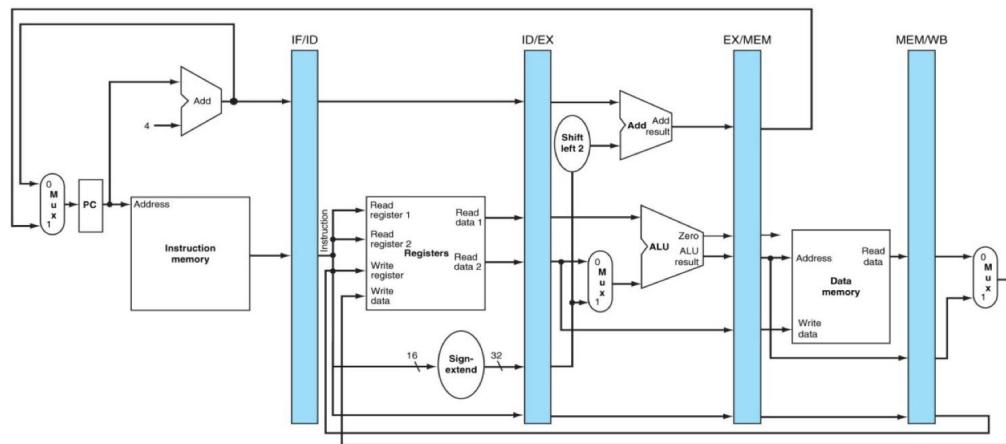
Registros	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Ciclo 6	Ciclo 7	Ciclo 8	Ciclo 9
\$8	0	0	0	0	0 / 32	32	32		
\$11	X <sup>→ No lo sabemos</sup>	X	X	X	X	X / 200	200		
\$12	X	X	X	X	X	X	X / 100		
EX/MEM	X	X	X	32	36	40	40		
MEM/WB	X	X	X	X	32	200	200		
ADD \$8,\$9,\$10	Fase Síntesis característica IF ( Cache L1 I )	ID ( Banco de registros ) ID/EX1 ← \$9 = 20 ID/EX2 ← \$10 = 12	EX ( ALU ) ID/EX3 ← \$8 = 0	MEM ( MD ) MEM/WB ← EX/MEM = 32	WB ( Regs ) \$8 ← MEM/WB = 32 Actualizar la tabla acísta	—			
LW \$11,4(\$8)	—	IF ( X ) ( L1 I )	ID ( Banco de reg ) ID/EX1 ← \$8 = 0 ID/EX2 ← 4	EX( ALU ) ID/EX3 ← \$8 = 0	MEM ( MD ) MEM/WB ← EX/MEM + INDEX2 = 32 + 4 MEM/WB ← MEM [ EX/MEM ] = MEM[36] = 200	WB ( Regs ) \$11 ← MEM/WB = 200			
LW \$12,8(\$8)	—	—	IF ( X ) instrucciones ID/EX1 ← \$8 = 0 ID/EX2 ← 8	ID ( Reg ) ID/EX3 ← \$8 = 0	EX ( ALU ) EX/MEM ← MEM + ID/EX2 = 32 + 8 = 40	MEM ( MD ) MEM/WB ← MEM[ EX/MEM ] = MEM[40] = 100	WB ( Regs ) \$12 ← MEM/WB = 100		
INSTRUCCIÓN 4									
Comentarios				\$8 no contiene todavía el valor deseado	Anticipación del valor a través de EX/MEM \$8 no contiene todavía el valor deseado	Anticipación del valor a través de MEM/WB			

## PLANTILLA PARA EJERCICIOS DE SEGMENTACIÓN:

Registro	Contenido
\$8	0
\$9	20
\$10	12

Posición de memoria	Contenido
[36]	200
[40]	100

Método utilizado para resolver los riesgos de datos	
Bloqueo?	SÍ
Anticipación?	SÍ



Registros	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Ciclo 6	Ciclo 7	Ciclo 8	Ciclo 9
\$8	0	0	0	0	0/32	32	32		
\$11	X	X	X	X	X	X/200	200		
\$12	X	X	X	X	X	X	X/100		
EX/MEM	X	X	X	32	36	40	40		
MEM/WB	X	X	X	X	32	200	100		
ADD\$8,\$9,\$10	IF(MI)	ID(REG) ID/EX1←\$9=20 ID/EX2←\$10=12	EX(ALU) EX/MEM←ID/EX1+ID/EX2= =20+12=32	MEM(MD) MEM/WB←EX/MEM=32	WB(REG) \$8←MEM/WB=32				
LW\$11,4(\$8)		IF(MI)	ID(REG) ID/EX1←\$8=0 ID/EX2←4	EX(ALU) EX/MEM←EX/MEM +ID/EX2= =32+4=36	MEM(MD) MEM/WB←MEM[EX/MEM]= =MEM[36]=200	WB(REG) \$11←MEM/WB=200			
LW\$12,8(\$8)			IF(MI)	ID(REG) ID/EX1←\$8=0 ID/EX2←8	EX(ALU) EX/MEM←MEM/WB +ID/EX2= =32+8=40	MEM(MD) MEM/WB←MEM[EX/MEM]= =MEM[40]=100	WB(REG) \$12←MEM/WB=100		
Comentarios			\$8 no contiene el valor actualizado	Anticipación a través de EX/MEM \$8 no contiene el valor actualizado	Anticipación a través de MEM/WB				

QuesoViejo\_

**SIGUE ESTUDIANDO, HASTA LLEGAR A SER QUIÉN SIEMPRE HAS QUERIDO.**

**Problema 2**

Sea la siguiente secuencia de instrucciones:

LW \$8, 4(\$9)  
ADD \$11, \$8, \$10  
SUB \$12, \$12, \$12

Contenido de ciertos registros y posiciones de memoria:

\$8 = 3  
\$9 = 16  
\$10 = 2  
\$12 = 1  
**MEM[20] = 200**

1.- Indica las dependencias que existen.

2.- Supongamos que esta secuencia de instrucciones se ejecuta en un procesador MIPS-32 segmentado CON unidad de detección de riesgos (bloqueos) y unidad de anticipación . Representar en un diagrama qué sucede en cada ciclo de ejecución.

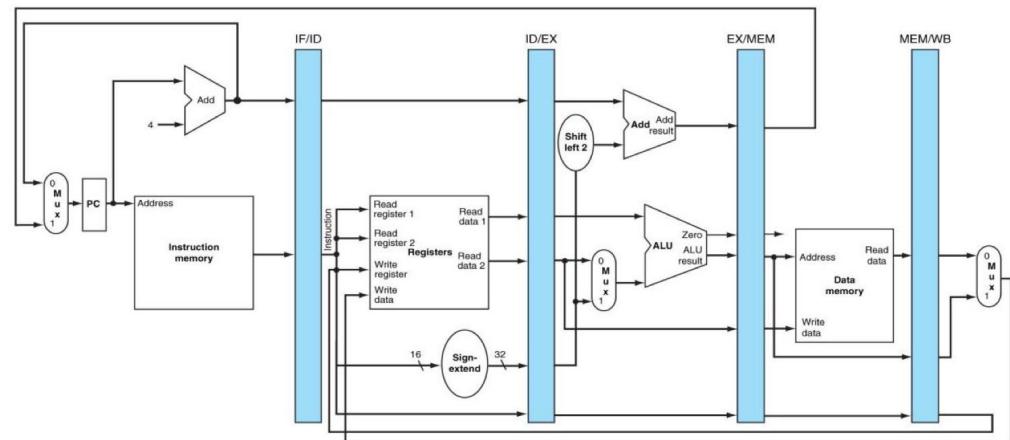


## PLANTILLA PARA EJERCICIOS DE SEGMENTACIÓN:

Registro	Contenido
\$8	3
\$9	16
\$10	2
\$12	1

Posición de memoria	Contenido
[20]	200

Método utilizado para resolver los riesgos de datos	
Bloqueo?	Sí
Anticipación?	Sí



Los dos reg que se utilizan para anticipación son EX/MEM y MEN/WB

Registros	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Ciclo 6	Ciclo 7	Ciclo 8	Ciclo 9
\$8	3	3	3	3	3/200	200	200	200	
\$9	X	X	X	X	X	X	X/202	202	
\$10	1	1	1	1	1	1	1	1	1
EX/MEM	X	X	X	20	20	202	0	0	
MEN/WB	X	X	X	X	200	200	202	0	
LW \$8,4(\$9)	IF(MI)	ID(Banco registros) ID/EX1 ← \$9 = 16 ID/EX2 ← 4	EX(ALU) EX1/EN ← INDEX1 + INDEX2 16 + 4 = 20	MEN(MD) MEN/WB ← MEN(EX/MEM) MEN[20] = 200	WB(Banco Regs) \$8 ← MEN/WB = 200				
ADD \$11,\$8,\$10		IF(MI)	ID(Banco reg) ID/EX1 ← \$8 = 3 ID/EX2 ← \$10 = 2	Bloqueo	EX(ALU) EX1/EN ← MEN/WB = 202 200 + 2 = 202	MEN(MD)	WB(Banco Regs) \$11 ← MEN/WB = 202		
SUB \$12,\$12,\$12			IF(MI)	Bloqueo	ID(Banco Reg) ID/EX1 ← \$12 = 1 ID/EX2 ← \$12 = 1	EX(ALU) EX1/EN = INDEX1 - INDEX2 = 1 - 1 = 0	MEN(MD)	WB(Banco reg) \$12 ← MEN/WB = 0	
INSTRUCCIÓN 4									
Comentarios				\$8 no contiene el valor deseado	Bloqueo ya que no se encuentra en ningún reg de segmentación el valor deseado	Anticipación del valor 200 en MEN/WB			

Se podría haber evitado el bloqueo cambiando el orden de estas instrucciones ya que sub es independiente y así solo va 2 ciclos por delante de LW, siendo innecesario el bloqueo

### Problema 3

Sea la siguiente secuencia de instrucciones:

LW \$1, 40(\$7)  
SW \$6, 8(\$1)  
ADD \$9, \$2, \$2

Dependencia RAW

1. Indica las dependencias que existen.
2. Supongamos que esta secuencia de instrucciones se ejecuta en un procesador MIPS-32 segmentado CON unidad de detección de riesgos (bloqueos) y unidad de anticipación. Representar en un diagrama qué sucede en cada ciclo de ejecución.
3. Si fuese el compilador el encargado de resolver los riesgos, ¿qué reordenación de código haría? ¿Tendría el compilador que insertar alguna instrucción NOP?

\*Instrucción NOP: No hacer nada (es un bloqueo / burbuja)

\*Los datos iniciales ya están puestos en la siguiente página.

3) Cambiaría el orden de las instrucciones 2 y 3, de esta forma no haría falta bloqueo

## PLANTILLA PARA EJERCICIOS DE SEGMENTACIÓN:

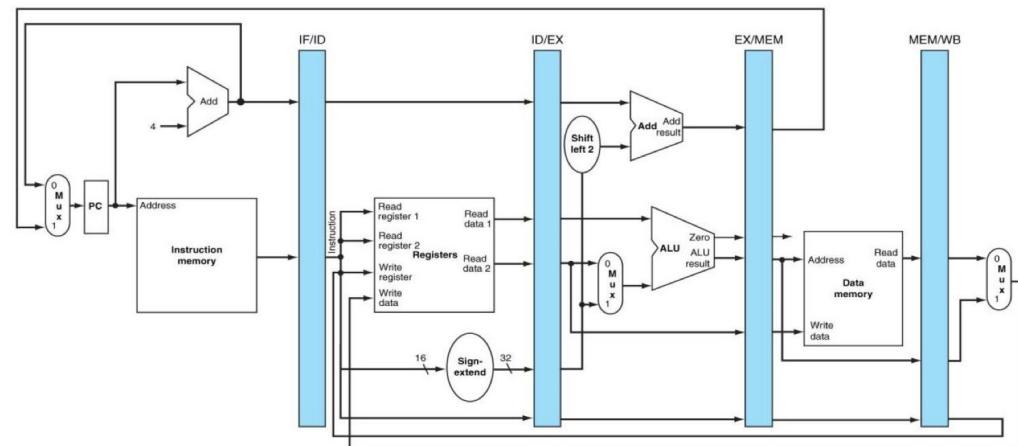
Registro	Contenido
\$7	4
\$5	16
\$2	10
\$6	1000
\$9	0

Posición de memoria	Contenido
E443	200



Método utilizado para resolver los riesgos de datos

Bloqueo?	Sí
Anticipación?	Sí



Registros	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Ciclo 6	Ciclo 7	Ciclo 8	Ciclo 9
\$5	16	16	16	16	16 / 200	200	200	200	200
\$6	1000	1000	1000	1000	1000	1000	1000	1000	1000
\$9	4	4	4	4	4	4	4	4 / 20	20
EX/NEM 1	-	-	-	44	44	208	20	20	20
NEM/WB 3	-	-	-	-	200	200	200	20	20
[208] 3	-	-	-	-	-	-	2000	2000	2000
LW \$3, 40(\$7)	IF(NT)	ID(Reg) ID/EX3 ← 40 ID/EX2 ← \$7 = 4	EX(ALU) EX/NEM 3 ← JDEV3 + ID/EX2 EX/NEM 3 ← 40 + 4 = 44	NEM(MD) NEM/WB 3 ← [EX/NEM 1] NEM/WB 3 ← [44] = 200	WB(Reg) \$3 ← NEM/WB 3 = 200				
SW \$6, 8(\$5)		IF(NT)	ID(Reg) ID/EX3 ← 8 ID/EX2 ← \$5 = 16 ID/EX3 ← \$6 = 1000	Bloqueo/Burbuja	EX ALU EX/NEM 3 ← JDEV3 + NEM/WB 1 EX/NEM 3 ← 31200 + 208 EX/NEM 2 ← ID/EX3 = 1000	NEM(MD) [EX/NEM 1] ← ID/EX 3 [208] ← 3000	WB(Reg)		
ADD \$9, \$2, \$2			IF(NT)	Bloqueo/Burbuja	ID(Reg) ID/EX3 ← \$2 = 30 ID/EX2 ← \$2 = 30	EX (ALU) EX/NEM 3 ← INVS1 + ID/EX2 EX/NEM 3 ← 10 + 10 = 20	NEM(MD) NEM/WB 3 ← EX/NEM 3 = 20	WB(Reg)	
Comentarios				En \$3 todavía no está el valor esperado	El valor esperado de \$3 no se puede anticipar todavía.	Anticipación del valor de \$3 a través del registro NEM/WB 3			

#### Problema 4

Sea la siguiente secuencia de instrucciones:

LW \$5, -16(\$5)  
SW \$5, -16(\$5)  
ADD \$5, \$5, \$5

1. Indica las dependencias que existen.
2. Supongamos que esta secuencia de instrucciones se ejecuta en un procesador MIPS-32 segmentado CON unidad de detección de riesgos (bloqueos) pero SIN unidad de anticipación. Representar en un diagrama qué sucede en cada ciclo de ejecución.
3. ¿Qué ocurre si al procesador se le añade una unidad de anticipación? ¿Se resuelven los riesgos?
4. Si fuese el compilador el encargado de resolver los riesgos, ¿qué reordenación del código haría? ¿Tendría el compilador que insertar alguna instrucción NOP?

4) Ninguna reordenación, ya que las instrucciones 2 y 3 tienen que leer el registro 5 y usan el valor que le mete la primera instrucción, por lo que no hay reordenación más eficiente

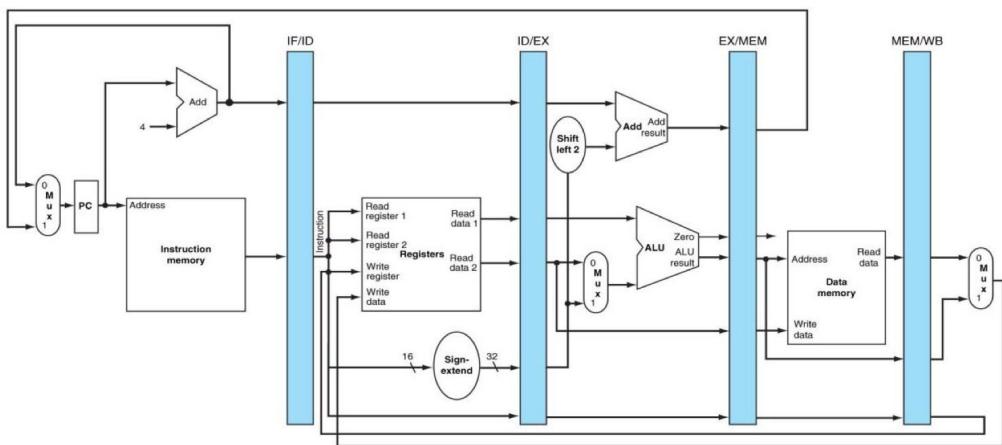
**PLANTILLA PARA EJERCICIOS DE SEGMENTACIÓN:** 2)

Registro	Contenido
\$5	200

Posición de memoria	Contenido
[200]	56
[184]	30
[34]	-

Método utilizado para resolver los riesgos de datos	
Bloqueo?	Sí
Anticipación?	No

Cloud icons indicating memory locations [200], [184], and [34].



Registros	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Ciclo 6	Ciclo 7	Ciclo 8	Ciclo 9
\$5	200	200	200	200	200 / 30	30	30	30	30 / 60
EXMEM3	-	-	-	584	584	584	54	54	54
MEMWB3	-	-	-	-	30	30	30	9	9
[34]	-	-	-	-	-	-	30	30	30
LW \$5,16(\$5)	IF(Not I)	ID(Reg) ID/EX3 ← -16 ID/EX2 ← \$5 = 200	EX(ALU) EX/MEM3 = ID/EX3 + ID/EX2 EX/EX3 = -16 + 200 = 184	MEM(MD) MEMWB3 ← [EX(MEM3)] MEMWB3 ← -184 = 30	WB(Reg) \$5 ← MEMWB3 = 30				
SW \$5,-16(\$5)		IF(Not I)	Bloqueo/Burbuja	Bloqueo/Burbuja	ID(Reg) ID/EX3 ← -16 ID/EX2 ← \$5 = 30 ID/EX3 ← \$5 = 30	EX(ALU) EX(MEM3) ← ID/EX3 + ID/EX2 EX/EX3 ← -16 + 30 = 14 EX/MEM3 ← ID/EX3 = 30	MEM(MD) [EX(MEM3)] ← EX/EX2 [EX(MEM3)] ← 30	WB(Reg)	
ADD \$5,\$5,\$5			Bloqueo/Burbuja	Bloqueo/Burbuja	IF(Not I)	ID(Reg) ID/EX3 ← \$5 = 30 ID/EX2 ← \$5 = 30	EX(ALU) EX(MEM3) ← ID/EX3 + ID/EX2 EX/EX3 ← 30 + 30 = 60	MEM(MD) MEMWB3 ← EX(MEM3) = 60	WB(Reg) \$5 ← MEMWB3 = 60
Comentarios			\$5 todavía no tiene el valor esperado por la instrucción 2	\$5 todavía no tiene el valor esperado por la instrucción 2					

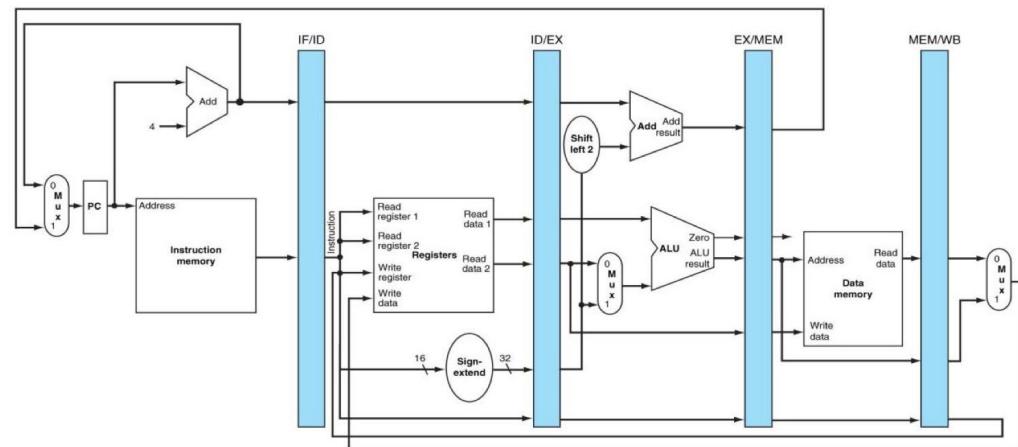
## PLANTILLA PARA EJERCICIOS DE SEGMENTACIÓN:

Registro	Contenido
\$5	200

Posición de memoria	Contenido
[200]	50
[384]	30
[34]	-

Método utilizado para resolver los riesgos de datos	
Bloqueo?	Sí
Anticipación?	Sí

3) Con anticipación



Registros	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Ciclo 6	Ciclo 7	Ciclo 8	Ciclo 9
\$5	200	200	200	200	200 / 30	30	30	30 / 60	60
EXMEMS	-	-	-	384	384	34	60	60	60
MEMWB3	-	-	-	-	36	30	30	60	60
[34]						-	30	30	30
LW \$5,-16(\$5)	IF(MI)	ID(Reg) ID/EX3 <- -16 ID/EX2 <- \$5 = 200	EX(ALU) EX/MEM3 = ID/EX3 + ISVEXT EX/EM3 <- -16 + 200 = 384	XEN(MD) XENWB3 <- [EXMEM3] MEMWB3 <- 384 = 30	WB(Reg) \$5 <- MEMWB3 = 30				
SW \$5,-16(\$5)		IF(MI)	ID(Reg) ID/EX3 <- -16 ID/EX2 <- \$5 = 200 ID/EX3 <- \$5 = 200	Bloqueo/Burbuja	EX(ALU) EX/EM3 <- ID/EX3 + XENWB3 EXMEM3 <- -16 + 30 = 14 EXMEM2 <- MEMWB3 = 30	XEN(MD) EXMEM3 <- EXMEM2 [34] <- 30	WB(Reg)		
ADD \$5,\$5,\$5			IF(MI)	Bloqueo/Burbuja	ID(Reg) ID/EX3 <- 35 = 36 ID/EX2 <- 35 = 30	EX(ALU) EX/MEM3 = ID/EX3 + ISVEXT EX/EM3 <- 30 + 30 = 60	XEM(MD)	WB(Reg) \$5 <- MEMWB3 = 60	
Comentarios			\$5 no tiene el valor esperado en la instrucción 2	No se puede anticipar \$5 todavía	Instrucción 2: Anticipa el valor de \$5 a través de MEMWB3				

QuesoViejo\_