

Responde cada pregunta en una hoja distinta. Tiempo disponible: 1h45m

1. (2,5 puntos) Una empresa utiliza un computador valorado en 4000€ para la ejecución de dos programas A y B (que no se ejecutan concurrentemente).

Se monitoriza el sistema y se constata que:

- El programa A se ejecuta el 25 % del tiempo y el resto lo hace el programa B.
- El programa A dedica el 80 % de su tiempo a realizar cálculos en coma flotante, mientras que el programa B no realiza ningún cálculo de este tipo y sólo ejecuta instrucciones de enteros dedicadas a otros menesteres.
- El programa B emplea el 40 % de su tiempo de ejecución accediendo a disco, mientras que el programa A no realiza ningún acceso.

Considerando el sistema completo, se pide:

- a) Obtener la fracción de uso del disco.
- b) Calcular la máxima aceleración que se obtendría cambiando sólo el disco del computador.
- c) Determinar si el cambio del disco existente por otro 3 veces más rápido, y que cuesta 800 €, es interesante desde el punto de vista de la relación prestaciones–coste.
- d) Finalmente, se añade una GPU para mejorar el rendimiento en coma flotante en un factor de 10 ¿en qué porcentaje se mejorarían las prestaciones del computador respecto a su configuración original si integráramos a la vez en el mismo un disco 3 veces más rápido y la GPU?

**Solución:**

- a) Obtener la fracción de uso del disco.

El disco se utiliza el 40 % del tiempo de ejecución de B, que es el 75 % del total.

$$F_{disco} = 0,4 \times 0,75 = 0,3$$

Por lo tanto, la fracción de uso del disco es del 30 % del tiempo total de ejecución del sistema.

- b) Calcular la máxima aceleración que se obtendría cambiando sólo el disco del computador.

Si cambiáramos el disco por uno infinitamente más rápido, el 30 % del tiempo total de ejecución dejaría de importar, porque pasaría a ser 0. Por tanto:

$$S_{max-disco} = \frac{1}{1 - F_{disco}} = \frac{1}{1 - 0,3} = \frac{1}{0,7} = 1,4286$$

con lo que la aceleración máxima a la que podríamos aspirar cambiando sólo el disco del computador será del 42.86 %.

- c) Determinar si el cambio del disco existente por otro 3 veces más rápido, y que cuesta 800 €, es interesante.

Si  $S_{disco} = 3$  la aceleración global sería:

$$S' = \frac{1}{1 - F_{disco} + \frac{F_{disco}}{S_{disco}}} = \frac{1}{1 - 0,3 + \frac{0,3}{3}} = \frac{1}{0,7 + 0,1} = \frac{1}{0,8} = 1,25$$

Esto se hará a costa de aumentar el precio del computador en 800 €, así pues

$$\Delta C = \frac{4000 + 800}{4000} = \frac{4800}{4000} = 1,2$$

Por tanto, incrementaremos el coste un 20 % para obtener una mejora del 25 % de las prestaciones, con lo que el cambio será interesante.

- d) Finalmente, se añade una GPU para mejorar el rendimiento en coma flotante en un factor de 10 ¿en qué porcentaje se mejorarían las prestaciones del computador respecto a su configuración original si integráramos a la vez en el mismo un disco 3 veces más rápido y la GPU?

Para obtener la mejora de prestaciones se utilizará la ley de Amdahl con dos mejoras simultaneas:

$$S = \frac{1}{1 - F_{disco} - F_{cf} + \frac{F_{disco}}{S_{disco}} + \frac{F_{cf}}{S_{cf}}}$$

Claramente la aceleración de la coma flotante es  $S_{cf} = 10$ . Por otro lado, el tiempo que el sistema utilizará la coma flotante es del 80 % de la ejecución del programa A, que se ejecuta el 25 % del tiempo total. Por lo tanto:

$$F_{cf} = 0,8 \times 0,25 = 0,2$$

Así pues:

$$S = \frac{1}{1 - 0,3 - 0,2 + \frac{0,3}{3} + \frac{0,2}{10}} = \frac{1}{0,5 + 0,1 + 0,02} = \frac{1}{0,62} = 1,6129$$

Por lo que el sistema mejoraría sus prestaciones en un 61,29 %.

□

2. (3 puntos) Se tiene un MIPS64 segmentado funcionando a 3 GHz que ejecuta un programa P con la siguiente distribución de instrucciones:

Tipo	%	CPI
Cargas	24 %	1.2
UAL	50 %	1.5
Almacenamientos	14 %	1
Salto	12 %	1.1

Se plantea añadir nuevas instrucciones SIMD que operen simultáneamente sobre las dos porciones de 32 bits de cada registro de coma flotante de 64 bits. De esta manera, en los programas, se pueden sustituir secuencias de dos instrucciones que trabajan sobre números en coma flotante de simple precisión por una instrucción SIMD, como en los siguientes ejemplos:

Las secuencias:	son sustituidas por:
l.s f0,d(rb)	l.ps f0,d(rb)
l.s f1,d+4(rb)	
...	
add.s f2,f0,f10	add.ps f2,f0,f10
add.s f3,f0,f11	
...	
s.s f0,d(rb)	s.ps f0,d(rb)
s.s f1,d+4(rb)	

Con la incorporación de las nuevas instrucciones se ha medido que se pueden reemplazar el 80 % de las instrucciones de carga, el 40 % de las UAL y el 60 % de los almacenamientos. Sin embargo, la inclusión de las nuevas instrucciones afecta al CPI de las instrucciones de carga y UAL (tanto las iniciales como las SIMD), pasando a ser de 1.4 y 1.8, respectivamente. Adicionalmente, debido a la necesidad de decodificar nuevas instrucciones, la frecuencia de reloj se reduce en un 5 %.

Se pide:

- CPI medio del programa P en el procesador original.
- Si el programa P ejecuta  $n$  instrucciones en el procesador original, calcular su tiempo de ejecución en ciclos y en segundos.
- Dado un programa P con  $n = 100$  instrucciones en el que se realizará la sustitución de instrucciones SIMD, indicar:
  - El número de instrucciones ejecutadas de carga escalar y carga SIMD (l.ps).
  - El número de instrucciones ejecutadas ALU escalar y ALU SIMD (por ejemplo, add.ps).
  - El número de instrucciones ejecutadas de almacenamiento escalar y almacenamiento SIMD (s.ps).
  - El número total de instrucciones ejecutadas en el procesador con instrucciones SIMD.
- CPI medio del programa P en el procesador con instrucciones SIMD.
- Tiempo de ejecución del programa P en el procesador con instrucciones SIMD, en ciclos y en segundos.
- La aceleración alcanzada, en su caso, al utilizar el procesador con instrucciones SIMD.

**Solución:**

- CPI medio en el procesador original Hay que hacer la media ponderada de las frecuencias y CPI individuales:  
 $CPI = 1,31$
- Tiempo de ejecución del programa P en el procesador original en función de su número  $n$  de instrucciones.  
Como la frecuencia es de 3GHz, el periodo de reloj es  $T = \frac{1}{3 \cdot 10^9} = 3,33 \cdot 10^{-10} = 0,33 \text{ ns}$   
 $T_{ejecucion} = I \times CPI \times T = n \times 1,31 \times 0,33 = 0,47n \text{ ns}$
- Número de instrucciones ejecutadas en el procesador con instrucciones SIMD en relación al original.  
Cuando se puede hacer la modificación, se reducen a la mitad las instrucciones en los tres casos implicados:
  - Instr. carga: Se puede aplicar al 80 %, que son  $24 \cdot 80 \% = 19,2$ . Se reducen a  $\frac{19,2}{2} = 9,6$  instrucciones SIMD y se quedan  $24 - 19,2 = 4,8$  sin modificar. En total:  $9,6 + 4,8 = 14,4$  instr. carga.
  - Instr. aritméticas: Se puede aplicar al 40 %, que son  $50 \cdot 40 \% = 20$ . Se reducen a  $\frac{20}{2} = 10$  instrucciones SIMD y se quedan  $50 - 20 = 30$  sin modificar. En total:  $10 + 30 = 40$  instr. aritméticas.
  - Instr. almac.: Se puede aplicar al 60 %, que son  $14 \cdot 80 \% = 8,4$ . Se reducen a  $\frac{8,4}{2} = 4,2$  instrucciones SIMD y se quedan  $14 - 8,4 = 5,6$  sin modificar. En total:  $4,2 + 5,6 = 9,8$  instr. almac.

La siguiente tabla muestra cómo queda la cuenta de instrucciones:

Tipo	#	CPI
Cargas:	14,4 %	1.4
UAL	40 %	1.8
Almac.	9,8 %	1
Salto	12 %	1.1
Total	76,2	

Por lo tanto, el la inclusión de instrucciones SIMD reduce el número de instrucciones ejecutadas  $I' = 0,762I$

d) CPI medio en el procesador con instrucciones SIMD.

Hay que hacer la media ponderada de las frecuencias y CPI individuales, pero teniendo en cuenta que el número de instrucciones ya no suma 100:

$$CPI = 1,51$$

e) Tiempo de ejecución del programa en el procesador con instrucciones SIMD.

Como la frecuencia se reduce en un 5 %,  $f' = 0,953GHz = 2,85$ , y el periodo de reloj es  $T' = \frac{1}{2,85 \cdot 10^9} = 3,51 \cdot 10^{-10} = 0,351 \text{ ns}$

$$T'_{ejecucion} = I' \times CPI' \times T' = 0,762n \times 1,51 \times 0,351 = 0,4n \text{ ns}$$

f) La aceleración alcanzada, en su caso, por parte del procesador con instrucciones SIMD.

El procesador con instrucciones SIMD es más rápido, puesto que reduce el tiempo de ejecución. La aceleración alcanzada es:

$$S = \frac{0,437n}{0,4} = 1,08 \text{ veces (un 8 % más rápido).}$$

□

3. (2,5 puntos) El siguiente diagrama instrucciones–tiempo corresponde a una iteración intermedia de un bucle que se ejecuta en un procesador MIPS segmentado:

	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
loop: ld r12,0(r1)	IF	ID	EX	ME	WB												
ld r14,0(r2)		IF	ID	EX	ME	WB											
dadd r12,r10,r12			IF	ID	EX	ME	WB										
dadd r14,r12,r14				IF	ID	EX	ME	WB									
sd r14,0(r3)					IF	ID	EX	ME	WB								
dadd r1,r1,#8						IF	ID	EX	ME	WB							
dadd r2,r2,#8							IF	ID	EX	ME	WB						
dadd r3,r3,#8								IF	ID	EX	ME	WB					
seq r5,r4,r1									IF	ID	EX	ME	WB				
beqz r5,loop										IF	ID	EX	ME	WB			
nop											IF	ID	X				
nop												IF	X				
loop: ld r12,0(r1)													IF	ID	EX	ME	WB

Responde a las siguientes preguntas:

- ¿Cómo resuelve la unidad de ejecución de instrucciones los riesgos de control? Explícalo e indica el nombre de la técnica utilizada.
- ¿En qué etapa de segmentación actualizan los saltos el valor del PC en el procesador considerado? No te limites a indicar cuál es la etapa, explica qué indicios te llevan a dicha conclusión.
- Se añade al procesador una BTB que emplea un predictor de dos bits con saturación. Cuando se introduce un nuevo salto en la BTB, el estado del predictor depende del comportamiento del salto. Si el salto “Salta”, el estado del predictor será el estado 11, en el que predice que el salto “Salta”, mientras que si “No Salta”, el estado del predictor será el estado 00, en el que se predice que el salto “No Salta”. Si el salto no está en la tabla, se predice que “No Salta”.
  - Indica el número de ciclos consumido en una iteración cuando el predictor acierta.
  - Indica el número de ciclos consumido en una iteración cuando el predictor falla.
  - Suponiendo que la tabla BTB está inicialmente vacía ¿cuántos ciclos requerirá el bucle para procesar un vector de 100 elementos?
  - Si la BTB se organiza como una tabla completamente asociativa y contara con 1024 entradas, ¿Cuántos bits del PC deberá almacenar cada una de las entradas del BTB?

**Solución:**

- ¿Cómo resuelve la unidad de ejecución de instrucciones los riesgos de control? Explícalo e indica el nombre de la técnica utilizada.

Utilizando la técnica del Predict-not-taken. Esto se aprecia en el ciclo 42 del diagrama IT, en el cuál se cancelan las 2 instrucciones que entraron en el pipeline, y que siguen al salto, para comenzar a ejecutar de nuevo la primera instrucción del bucle.

- ¿En qué etapa de segmentación actualizan los saltos el valor del PC en el procesador considerado? No te limites a indicar cuál es la etapa, explica qué indicios te llevan a dicha conclusión.  
En la tercera etapa del pipeline, es decir, en EX. Esto se deduce del diagrama IT, concretamente en el ciclo 41 en el que el salto se toma y es en el ciclo 42 cuando comienzan a buscarse instrucciones en la dirección destino (*loop* en nuestro caso).
- Indica el número de ciclos consumido en una iteración cuando el predictor acierta.  
Si el predictor acierta, el bucle se ejecutará en 10 ciclos.
- Indica el número de ciclos consumido en una iteración cuando el predictor falla.  
Si el predictor falla, el bucle se ejecutará en 12 ciclos.
- Suponiendo que la tabla BTB está inicialmente vacía ¿cuántos ciclos requerirá el bucle para procesar un vector de 100 elementos?  
En la primera iteración del bucle, el salto no se encontrará en la BTB y se predecirá incorrectamente que el salto no se toma, posicionándose por tanto el predictor en estado 11 (“Salta”). En la segunda y sucesivas iteraciones, el predictor acertará en su predicción. Pero en la última iteración el predictor fallará, prediciendo que el salto se tomará, cuando no debería hacerse. En definitiva, el predictor fallará 2 veces (al principio y al final), con lo que el tiempo de ejecución del bucle será de  $98 * 10 + 2 * 12 = 1004$  ciclos.
- Si la BTB se organiza como una tabla completamente asociativa y contara con 1024 entradas, ¿Cuántos bits del PC deberá almacenar cada una de las entradas del BTB?  
Con independencia del tamaño de la tabla, cada entrada almacenará todos los bits del PC.

□

4. (2 puntos) Indica, para los diagramas instrucciones–tiempo mostrados, qué técnica se ha utilizado para resolver los riesgos de datos (“parada”, “cortocircuito”) y los riesgos de control (“parada”, “predict-not-taken”, “salto retardado”, “predictor BTB”). No te limites a indicar el nombre de la técnica. Explica que indicios te llevan a dicha conclusión.

a)	PC	Instrucción	1	2	3	4	5	6	7	8	9	10
	.text	slt r1,r3,r0	IF	ID	EX	ME	WB					
	4	dsub r2,r0,r3		IF	ID	EX	ME	WB				
	8	bnez r1,then			IF	ID	EX	ME	WB			
	12	nop				IF	ID	X				
	16	nop					IF	X				
	then	dadd r4,r2,r5						IF	ID	EX	ME	WB

  

b)	PC	Instrucción	1	2	3	4	5	6	7	8	9	10	11
	.text	slt r1,r3,r0	IF	ID	EX	ME	WB						
	4	dsub r2,r0,r3		IF	ID	EX	ME	WB					
	8	bnez r1,then			IF	id	ID	EX	ME	WB			
	12	nop					IF	ID	EX	ME	WB		
	16	nop						IF	ID	EX	ME	WB	
	then	dadd r4,r2,r5							IF	ID	EX	ME	WB

  

c)	PC	Instrucción	1	2	3	4	5	6	7	8
	.text	slt r1,r3,r0	IF	ID	EX	ME	WB			
	4	dsub r2,r0,r3		IF	ID	EX	ME	WB		
	8	bnez r1,then			IF	ID	EX	ME	WB	
	then	dadd r4,r2,r5				IF	ID	EX	ME	WB

  

d)	PC	Instrucción	1	2	3	4	5	6	7	8	9	10
	.text	slt r1,r3,r0	IF	ID	EX	ME	WB					
	4	dsub r2,r0,r3		IF	ID	EX	ME	WB				
	8	bnez r1,then			IF	ID	EX	ME	WB			
	12	nop				if	if					
	then	dadd r4,r2,r5						IF	ID	EX	ME	WB

### Solución:

- a) Riesgos de datos: cortocircuito. Indicios: Hay un riesgo de datos entre las instrucciones *slt* y *bnez*, pero no se insertan ciclos de parada, aplicándose un cortocircuito de entre las etapas WB a EX en el ciclo 5.  
Riesgos de control: predict-not-taken. Indicios: Se buscan dos instrucciones que siguen a la instrucción de salto, las cuales se cancelan al ser el salto efectivo.
- b) Riesgos de datos: ciclos de parada. Indicios: Hay un riesgo de datos entre las instrucciones *slt* y *bnez*, insertándose un ciclo de parada, de manera que se lee el registro involucrado en el mismo ciclo que se está escribiendo (ciclo 5).  
Riesgos de control: salto retardado. Indicios: Se buscan dos instrucciones que siguen a la instrucción de salto, las cuales no se cancelan a pesar de ser el salto efectivo.
- c) Riesgos de datos: cortocircuito.  
Riesgos de control: predictor BTB. Indicios: Se busca la instrucción destino del salto inmediatamente a continuación de la de salto, lo que implica que hay un predictor que ofrece su resultado en la etapa IF.

d) Riesgos de datos: cortocircuito.

Riesgos de control: ciclos de parada Indicios: Se detiene la búsqueda de instrucciones hasta que se obtiene el resultado de la instrucción de salto (efectivo en este caso) en el ciclo 5, comenzando la búsqueda en la dirección destino del salto en el ciclo 6.

