

Responde cada pregunta en una hoja distinta. Tiempo disponible: 3 horas

1. (2.5 puntos) Un algoritmo A se ejecuta en un computador que incorpora un procesador multinúcleo P_{multi} . El código de A que puede ejecutarse en paralelo se ejecuta simultáneamente en todos los núcleos del procesador, mientras que la parte secuencial del algoritmo sólo se ejecuta en uno. Aplicando Amdahl, se obtiene que, frente a un procesador mononúcleo P_{mono} , la aceleración global que ofrece P_{multi} viene dada por la siguiente expresión:

$$\frac{1}{(1-0,6) + \frac{0,6}{8}}$$

Teniendo en cuenta esta información, responda a las siguientes cuestiones:

- ¿Cuántos núcleos posee el procesador P_{multi} ?
- ¿Qué fracción del tiempo de ejecución del algoritmo A se emplea en la ejecución de código secuencial? Expresa la fracción solicitada en porcentaje.
- ¿Cuál sería la mejora de prestaciones que podríamos llegar a obtener para el algoritmo A si aceleráramos todo lo posible la ejecución del código de A que puede ejecutarse en paralelo? Expresa la mejora solicitada en porcentaje.
- Se plantea la compra de un compilador C capaz de producir código máquina para el algoritmo A con un mayor grado de paralelización. De hecho, el compilador consigue que el computador con P_{multi} ejecute el código que genera para A un 130 % más rápido que si se utiliza el procesador P_{mono} . Indica si, para dicho código, la fracción del tiempo de ejecución que se dedica a la ejecución de código secuencial es *mayor, menor o igual* que la que se dedicaba antes de considerar C. Razona y cuantifica tu respuesta.
- Como alternativa al compilador C, se plantea mantener el compilador ya existente y adquirir un nuevo procesador multinúcleo $P_{multi-nuevo}$. Se sabe que:
 - Los núcleos de $P_{multi-nuevo}$ funcionan a una frecuencia que es f veces superior a la del núcleo de P_{mono} .
 - $P_{multi-nuevo}$ ejecuta 1,3 veces más rápido que P_{mono} el código secuencial del algoritmo A.

Teniendo esta información en cuenta, ¿cuántos núcleos deberá incorporar $P_{multi-nuevo}$ y a qué frecuencia deberán funcionar dichos núcleos para que, cuando se use este procesador, se acelere globalmente la ejecución del algoritmo A un 136.36 % con respecto a utilizar el procesador P_{mono} ?

Solución:

Cambiar solución para el apartado e)

- ¿Cuántos núcleos posee P_{multi} ?

8

- ¿Qué fracción del tiempo de ejecución del algoritmo A se emplea en la ejecución de código secuencial?

$F = 1 - 0,6 = 0,4$, es decir, el 40 %.

- ¿Cuál sería la mejora de prestaciones que podríamos llegar a obtener para el algoritmo A si aceleráramos todo lo posible la ejecución de la fracción de código que, según los datos que aporta la expresión mostrada, puede ejecutarse en paralelo?

$S_{global} = \frac{1}{0,4} = 2,5$, es decir, un 150 %.

- Indica si, para el código máquina generado por el compilador C para el algoritmo A, la fracción del tiempo de ejecución que se dedica a la ejecución de código secuencial es mayor, menor o igual que la que se dedicaba antes de considerar C.

Para alcanzar una aceleración global del 130 %, y teniendo en cuenta que P_{multi} no sufre modificación alguna, lo que debe ocurrir es que el compilador C debe conseguir paralelizar un porcentaje mayor del código del algoritmo A. Esto se traducirá en que durante la ejecución del algoritmo el procesador pasará una fracción del tiempo de ejecución mayor ejecutando código en paralelo, y por tanto una menor porción de dicho tiempo ejecutando código secuencial.

Numéricamente la fracción de tiempo dedicado a la ejecución de código secuencial puede calcularse como $1 - F$ siendo F la fracción de tiempo que se beneficia de la mejora, es decir, de poder ejecutarse en paralelo.

$2,3 = \frac{1}{(1-F) + \frac{F}{8}}$ de donde despejando $F = 0,6459$ con lo que se pasa a dedicar aproximadamente un 64.6 % del tiempo ejecutando código en paralelo y un 35.4 % ($1 - 0,646 = 0,354$) del tiempo ejecutando código secuencial.

Por tanto, hemos pasado de dedicar un 40 % del tiempo a ejecutar código secuencial a dedicar un 35.4 % del tiempo, es decir, la fracción de tiempo es *menor*.

- e) ¿Cuántos núcleos deberá incorporar $P_{multi-nuevo}$ y a qué frecuencia deberán funcionar dichos núcleos para que, cuando se use este procesador, se acelere globalmente la ejecución del algoritmo A un 136.36 % con respecto a utilizar el procesador P_{mono} ?

Sea n el número de núcleos disponibles en $P_{multi-nuevo}$ y f el factor de mejora asociado al aumento de la frecuencia de reloj en dichos núcleos. La mejora relativa al número de núcleos disponibles afectará sólo a la ejecución del código paralelizable del programa, mientras que la relativa al incremento de la frecuencia afectará a todo el programa. Por tanto la mejora aplicada al código secuencial será de f , y la aplicada al código paralelizable será de $f \times n$.

Respecto a n , podemos calcularlo sabiendo que:

$$\frac{1}{\frac{1-0,6}{1,3} + \frac{0,6}{n \times 1,3}} = 2,3636$$

de donde deducimos que $n = 4$.

Por tanto $P_{multi-nuevo}$ dispondrá de 4 núcleos que deberán funcionar, al menos, a una frecuencia un 30 % superior a la de los núcleos de P_{mono} , para alcanzar la mejora global especificada.

□

2. (2.5 puntos) Un programa compilado para un procesador MIPS segmentado en las cinco etapas habituales ejecuta 200 millones de instrucciones con la siguiente distribución:

Tipo	%
Carga	20
Almacenamiento	5
Aritméticas	60
Salto	15

El procesador dispone del hardware necesario para evitar cualquier tipo de riesgo estructural. Los riesgos de datos se resuelven mediante cortocircuitos e insertando ciclos de parada cuando éstos no están disponibles. Los riesgos de control se resuelven mediante un BTB que obtiene la predicción en la fase de búsqueda de la instrucción y siempre acierta. En la implementación considerada, la condición de salto se calcula en ID y el PC se actualiza en la etapa EX.

El CPI de todas las instrucciones es 1, excepto en el 25 % de las instrucciones aritméticas, en las cuales el CPI será 2 debido a la inserción de un ciclo de parada para resolver el riesgo causado por una dependencia de datos con una instrucción de carga inmediatamente anterior.

Asumiendo que la frecuencia del procesador es de 500MHz, responda a las siguientes cuestiones:

- Determine el CPI medio de las instrucciones y calcule el tiempo de ejecución del programa, en segundos, en el procesador segmentado.
- Se plantea sustituir en el procesador original todas las instrucciones aritméticas que presentan conflicto de datos con las cargas que las preceden, por otras nuevas, que denominaremos *ALUmem* porque pueden operar directamente con un operando en memoria. Así el compilador podrá utilizar, siempre que sea necesario, las instrucciones *ALUmem* y prescindir de las instrucciones de carga que generan el conflicto de datos.

Sabiendo que las instrucciones *ALUmem* tienen un CPI de 3 y que la modificación propuesta obliga a reducir la frecuencia del procesador a 450MHz, ¿se mejoran o empeoran las prestaciones con respecto a las ofrecidas por el procesador segmentado original? Expresa tu respuesta indicando cuál es la opción más rápida y en qué porcentaje acelera esta opción con respecto a la más lenta.

Solución:

OJO. En el apartado b) las load son 20 -15 = 5. Recalcular ecuaciones.

a) CPI medio y tiempo de ejecución en el procesador segmentado:

$$\overline{CPI} = \frac{20 \cdot 1 + 5 \cdot 1 + 60 \cdot (0,25 \cdot 2 + 0,75 \cdot 1) + 15 \cdot 1}{100} = 1,15$$

$$T_{ejec} = I \cdot \overline{CPI} \cdot \frac{1}{f} = 200 \cdot 10^6 \cdot 1,15 \cdot \frac{1}{500 \cdot 10^6} = 0,46s$$

b) Nueva distribución de instrucciones:

Tipo	Instr.	CPI
Carga	20-15=5	1
Almacenamiento	5	1
Aritméticas	60-15=45	1
Saltos	15	1
Nuevas ALU	15	3
Total inst	85	

$$\overline{CPI}_{modif} = \frac{5 \cdot 1 + 5 \cdot 1 + 45 \cdot 1 + 15 \cdot 1 + 15 \cdot 3}{85} = \frac{115}{85} = 1,353$$

$$I_{modif} = 0,85 \cdot I_{orig}$$

$$T_{ejec} = I_{modif} \cdot \overline{CPI}_{modif} \cdot \frac{1}{f} = 0,85 \cdot 200 \cdot 10^6 \cdot 1,353 \cdot \frac{1}{450 \cdot 10^6} = 0,5111s$$

El procesador original es

$$\frac{0,5111}{0,46} = 1,1117$$

veces mejor que la propuesta de modificación propuesta, es decir, un 11.17 %, con lo que la propuesta no resulta de interés.

□

3. **(2.5 puntos)** Se dispone de un procesador MIPS con ejecución fuera de orden y especulación hardware basada en el algoritmo de Tomasulo. Las instrucciones atraviesan las siguientes etapas: IF (búsqueda de instrucciones), I (decodificación y lanzamiento de las instrucciones), En (ejecución en el operador multiciclo correspondiente), WB (escritura en el bus común de datos) y C (confirmación de las instrucciones). El ROB tiene 32 entradas, identificándose la primera de ellas como entrada #0. El procesador dispone de un predictor de saltos del tipo *Branch Target Buffer* (BTB) de 2 bits que ofrece la predicción al final de la etapa IF.

Las características de las unidades funcionales (operadores) son las siguientes:

	Nº Operadores	Latencia	Características
Carga/Almacenamiento	1	2	No segmentada; 4 buffers de lectura y 4 de escritura
Suma/Resta CF	1	2	Segmentada; 4 estaciones de reserva
Multiplicador CF.	1	4	Segmentada; 4 estaciones de reserva
Enteros/Saltos	1	1	4 estaciones de reserva

Se pretende evaluar el comportamiento del procesador ante el siguiente bucle, el cual realiza el producto escalar de dos vectores $q = \vec{x} \cdot \vec{y}$.

```
.data
;Vectores
x: .double 0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7
y: .double 0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7
;Resultado
q: .double 0

.text
; r10 apunta a x
```

```

; r11 apunta a y
loop: l.d f0,0(r10)
      l.d f1,0(r11)
      mul.d f0,f0,f1
      add.d f20,f0,f20
      dadd r10,r10,#8
      dadd r11,r11,#8
      dadd r13,r13,#1
      bne r13,r14,loop
      s.d f20,q($gp)
      trap #0

```

El diagrama instrucciones–tiempo que se muestra a continuación refleja la secuencia de instrucciones ejecutada en la última iteración del bucle:

		Ciclo																														
PC	Instruccion	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85					
loop	l.d f0,0(r10)	IF	I	-	-	AC	L1	L2	-	-	WB	-	-	-	-	-	-	-	-	C												
4100	l.d f1,0(r11)		IF	I	-	-	-	AC	L1	L2	-	-	WB	-	-	-	-	-	-	-	C											
4104	mul.d f0,f0,f1			IF	I	-	-	-	-	-	-	-	-	M1	M2	M3	M4	WB	-	-	-	C										
4108	add.d f20,f0,f20				IF	I	-	-	-	-	-	-	-	-	-	-	-	-	A1	A2	WB	-	C									
4112	dadd r10,r10,8					IF	I	E1	-	-	-	-	-	WB	-	-	-	-	-	-	-	-	-	-	C							
4116	dadd r11,r11,8						IF	I	-	E1	-	-	-	-	WB	-	-	-	-	-	-	-	-	-	-	C						
4120	dadd r13,r13,1							IF	I	-	E1	-	-	-	-	WB	-	-	-	-	-	-	-	-	-	C						
salto	bne r13,r14,loop								IF	I	-	-	-	-	-	-	E1	-	WB	-	-	-	-	-	-	-	C					
loop	l.d f0,0(r10)									IF	I	-	-	-	AC	L1	L2	-	-	WB	-	-	-	-	-	-	x					
4100	l.d f1,0(r11)										IF	I	-	-	-	AC	-	L1	L2	-	-	WB	-	-	-	-	x					

Se solicita:

a) Considera la instrucción `add.d f20, f0, f20` e indica el ciclo de reloj en el cual:

- 1) Se copia el valor del primer operando fuente a su estación de reserva.
- 2) Se copia el valor del segundo operando fuente a su estación de reserva.
- 3) Se almacena el resultado de la ejecución en el ROB.
- 4) Se almacena el resultado en el registro destino.

b) ¿Cuál es el número de ciclos consumido por una iteración cuando el predictor acierta? ¿Y cuando falla?

c) Con el objeto de mejorar las prestaciones, se diseña la siguiente generación del procesador para que sea superescalar de **2 vías, duplicando** el número de unidades funcionales disponibles. La fase IF proporciona dos instrucciones alineadas en una dirección múltiplo de 8. Dibuja el diagrama instrucciones-tiempo de una iteración **intermedia** (en la que el predictor acierta). Refleja en el diagrama **las instrucciones de esa iteración y las dos primeras de la siguiente**. Considera, para simplificar, que al inicio de la iteración el ROB y las estaciones de reserva están vacías, y que los registros no tienen marcas.

Solución:

a) Considera la instrucción `add.d f20, f0, f20` e indica el ciclo de reloj en el cual:

- 1) Se copia el valor del primer operando fuente a su estación de reserva. Ciclo 76.
- 2) Se copia el valor del segundo operando fuente a su estación de reserva. Ciclo 64.
- 3) Se almacena el resultado de la ejecución en el ROB. Ciclo 79.
- 4) Se almacena el resultado en el registro destino. Ciclo 81.

b) ¿Cuál será el número de ciclos consumido por una iteración cuando el predictor acierta? ¿Y cuando falla?.

- Acierto: Del ciclo 60 al 67 ($67 - 60 + 1$) = 8 ciclos
- Fallo: Del ciclo 60 al 85 ($85 - 60 + 1$) = 26 ciclos

c) Con el objeto de mejorar las prestaciones, se diseña la siguiente generación del procesador para que sea superescalar de **2 vías, duplicando** el número de unidades funcionales disponibles. La fase IF proporciona dos instrucciones alineadas en una dirección múltiplo de 8. Dibuja el diagrama instrucciones-tiempo de una iteración **intermedia** (en la que el predictor acierta). Refleja en el diagrama **las instrucciones de esa iteración y las dos primeras de la siguiente**. Considera, para simplificar, que al inicio de la iteración el ROB y las estaciones de reserva están vacías, y que los registros no tienen marcas.

PC	Instruccion	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
loop	l.d f0,0(r10)	IF	I	AC	L1	L2	WB	C											
4100	l.d f1,0(r11)	IF	I	AC	L1	L2	WB	C											
4104	mul.d f0,f0,f1		IF	I	-	-	-	M1	M2	M3	M4	WB	C						
4108	add.d f20,f0,f20		IF	I	-	-	-	-	-	-	-	-	A1	A2	WB	C			
4112	dadd r10,r10,8			IF	I	E1	-	WB	-	-	-	-	-	-	-	C			
4116	dadd r11,r11,8			IF	I	E1	-	WB	-	-	-	-	-	-	-	-	C		
4120	dadd r13,r13,1				IF	I	E1	-	WB	-	-	-	-	-	-	-	C		
salto	bne r13,r14,loop				IF	I	-	-	-	E1	WB	-	-	-	-	-	-	C	
loop	l.d f0,0(r10)					IF	I	-	AC	L1	L2	WB	-	-	-	-	-	C	
4100	l.d f1,0(r11)					IF	I	-	AC	L1	L2	-	WB	-	-	-	-	-	C

□

4. (2.5 puntos)

Un computador tiene los siguientes componentes:

- Un procesador *load/store* semejante al MIPS, con ejecución en orden, un ancho de palabra de 64 bits y que trabaja a una frecuencia de reloj de 3.0 GHz. Se ha medido que el valor medio del CPI es 1.7, y que el 15 % de las instrucciones que ejecuta son lecturas de memoria, y el 5 % son escrituras.
- Una memoria cache de dos niveles. El primer nivel L1 tiene una cache de instrucciones y de datos separadas. El tiempo de acierto es de 1 ciclo para las caches L1 y de 5 ciclos para la cache L2. Las tasas de fallos locales son de 1 % para la cache L1 de instrucciones, 2.5 % para la cache L1 de datos y 35 % para la cache L2. El tamaño de bloque es de 128 Bytes en ambos niveles.
- La memoria principal es SDRAM DDR4 y sus parámetros temporales son $CL - t_{RCD} - t_{RP} = 10 - 12 - 12$ a una frecuencia de reloj de 1200 MHz.

Con la carga estandar ejecutada en el computador se ha estimado la tasa de aciertos de fila abierta, $TAbf$, es un 90 %. Calcula justificando las respuestas:

- El tiempo medio necesario para traer un bloque desde la memoria principal a la cache L2 expresado en ciclos de reloj de procesador y en nanosegundos.
- El tiempo de ejecución de un *benchmark* con 200 millones de instrucciones, suponiendo que el tiempo medio de acceso a un bloque en memoria principal es de 20 ns.
- La tasa de fallos global de la jerarquía de cache.
- El tiempo de ejecución si se introdujera un tercer nivel de cache con un tiempo de acierto de 8 ciclos y una tasa de fallos del 50 %, suponiendo que el tiempo medio de acceso a un bloque en memoria principal es de 20 ns.

Solución:

- El tiempo medio necesario para traer un bloque desde la memoria principal a la cache L2 expresado en ciclos de reloj de procesador y en nanosegundos.

$$\begin{aligned}
 PF_{L2} = T_{MP} &= (1 - 0,9) \cdot (10 + 12 + 12) + 0,9 \cdot (10) + \frac{128B}{16B} = 20,4 \text{ ciclos memoria} \\
 &= 20,4 \cdot \frac{1}{1,2} = 17 \text{ ns} \\
 &= 20,4 \cdot \frac{3,0}{1,2} = 51 \text{ ciclos CPU}
 \end{aligned}$$

- El tiempo de ejecución de un *benchmark* con 200 millones de instrucciones, suponiendo que el tiempo medio de acceso a un bloque en memoria principal es de 20 ns.
El tiempo medio de acceso a memoria principal es de 20 ns, y esto corresponderá a la PF_{L2} al haber 2 niveles de cache. Lo pasamos a ciclos de procesador y calculamos PF_{L1} ,

$$20 \cdot 10^{-9} * 3 \cdot 10^9 = 60 \text{ ciclos CPU}$$

$$PF_{L1} = 1 + 0,35 \times 60 = 26 \text{ ciclos CPU}$$

el tiempo de ejecución de calcula,

$$\begin{aligned} T_{ex} &= 200 \cdot 10^6 \text{ (I)} \times 1,7 \text{ (CPI)} \times \frac{1}{3 \cdot 10^9} \text{ (T)} + \\ &200 \cdot 10^6 \text{ (I)} \times 1 \text{ (API}_I\text{)} \times 0,01 \text{ (TF}_I\text{)} \times 26 \text{ (PF}_{L1}\text{)} \times \frac{1}{3 \cdot 10^9} \text{ (T)} + \\ &200 \cdot 10^6 \text{ (I)} \times (0,15 + 0,05) \text{ (API}_D\text{)} \times 0,025 \text{ (TF}_D\text{)} \times 26 \text{ (PF}_{L1}\text{)} \times \frac{1}{3 \cdot 10^9} \text{ (T)} \\ &= 0,1393 \text{ seg} \end{aligned}$$

c) La tasa de fallos global de la jerarquía de cache.

$$TF_G = TF_{L1} \times TF_{L2}$$

hay que calcular la tasa de fallo combinada para la $L1$ ya que nos la dan separada para datos e instrucciones,

$$TF_{L1} = TF_I \cdot \frac{1}{1 + 0,2} + TF_D \cdot \frac{0,2}{1 + 0,2} = 0,0125$$

y así,

$$TF_G = 0,0125 \times 0,35 = 0,0044 \rightarrow 0,44 \%$$

d) El tiempo de ejecución si se introdujera un tercer nivel de cache con un tiempo de acierto de 8 ciclos y una tasa de fallos del 50 %.

En este caso la PF_{L2} cambia y esto afecta a la PF_{L1} . Se calculará como,

$$PF_{L1} = TA_{L2} + TF_{L2} \times PF_{L2}$$

$$PF_{L2} = TA_{L3} + TF_{L3} \times PF_{L3}$$

donde $PF_{L3} = 20 \text{ ns}$ (tiempo de acceso medio a memoria principal) que corresponden a 60 ciclos de CPU como hemos calculado antes. Así,

$$PF_{L2} = 8 + 0,5 \times 60 = 38 \text{ ciclos de CPU}$$

$$PF_{L1} = 5 + 0,35 \times 38 = 18,3 \text{ ciclos de CPU}$$

recalculando,

$$\begin{aligned} T_{ex} &= 200 \cdot 10^6 \text{ (I)} \times 1,7 \text{ (CPI)} \times \frac{1}{3 \cdot 10^9} \text{ (T)} + \\ &200 \cdot 10^6 \text{ (I)} \times 1 \text{ (API}_I\text{)} \times 0,01 \text{ (TF}_I\text{)} \times \mathbf{18,3} \text{ (PF}_{L1}\text{)} \times \frac{1}{3 \cdot 10^9} \text{ (T)} + \\ &200 \cdot 10^6 \text{ (I)} \times (0,15 + 0,05) \text{ (API}_D\text{)} \times 0,025 \text{ (TF}_D\text{)} \times \mathbf{18,3} \text{ (PF}_{L1}\text{)} \times \frac{1}{3 \cdot 10^9} \text{ (T)} \\ &= 0,1316 \text{ seg} \end{aligned}$$

□

Apellidos y Nombre:	
----------------------------	--

Ejercicio 3 Diagrama instrucciones-tiempo.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

[illegible]