

1. (3.5 puntos) Un procesador dispone de ocho núcleos idénticos. Para mejorar las prestaciones del procesador los arquitectos están pensando en dos opciones de diseño; en ambas, **el área debe ser la misma** que la del procesador original.

La primera opción consiste en sustituir todos los núcleos originales por núcleos que ocupan la mitad de espacio pero cuyas prestaciones son un 70 % las de los núcleos originales.

La segunda opción consiste en dedicar parte del espacio a un núcleo un 41 % más rápido. En concreto, este núcleo ocupa doble área que un núcleo original. El resto del área está ocupada por núcleos originales.

Se pide:

- Suponiendo que una aplicación A tiene una fracción de tiempo paralelizable del 80 % ($F=0.8$) cuando se ejecuta sobre 1 solo núcleo, calcula la aceleración del procesador original con 8 núcleos respecto a la ejecución en 1 solo núcleo.
- Calcula la aceleración o deceleración de la opción 1 sobre el procesador original (con 8 núcleos) para la aplicación A.
- Calcula la aceleración o deceleración de la opción 2 sobre el procesador original (con 8 núcleos) para la aplicación A. Asume que la fracción de tiempo paralelizable se reparte a partes iguales entre los núcleos independientemente de la potencia de cómputo del mismo, y la fracción secuencial la ejecuta el núcleo más rápido.
- Cuando una aplicación B se ejecuta sobre los 8 núcleos en el procesador original se alcanza un speedup de 1.7, respecto a si la misma aplicación se ejecuta sobre el mismo procesador pero utilizando solo 4 núcleos. Calcula la fracción de tiempo paralelizable de la aplicación B con 4 núcleos.

Solución:

- a) Aceleración o deceleración de la opción 1: procesador de 16 núcleos. Aceleración sistema original sobre un núcleo:

$$S'_{8nucleos} = \frac{1}{1 - F + \frac{F}{S}} = \frac{1}{1 - 0,8 + \frac{0,8}{8}} = \frac{1}{0,2 + 0,1} = 3,33$$

- b) Aceleración o deceleración de la opción 1: procesador de 16 núcleos.

Aceleración con 16 núcleos sobre un núcleo original:

$$S'_{16nucleos} = \frac{1}{\frac{1-F}{S_{secuencial}} + \frac{F}{S_{paralela}}} = \frac{1}{\frac{1-0,8}{0,7} + \frac{0,8}{0,7 \times 16}} = \frac{1}{0,285 + 0,071} = 2,81$$

Lo que significa que hay deceleración, en este caso el sistema original es un 18 % más rápido.

$$S'_{orig \text{ sobre } 16nucleos} = \frac{3,33}{2,81} = 1,18$$

- c) Calcula la aceleración o deceleración de la opción 2: uso de 1 núcleo doble rápido.
Aceleración procesador con núcleo doble rápido sobre un núcleo original:

$$S'_{1nucleo \text{ rapido}} = \frac{1}{\frac{1-F}{S_{secuencial}} + \frac{F}{S_{paralela}}} = \frac{1}{\frac{1-0,8}{1,41} + \frac{0,8}{7}} = \frac{1}{0,142 + 0,114} = 3,90$$

Por tanto, la mejora del procesador con el núcleo doble rápido sobre el original es:

$$S'_{1nucleo-rapido \text{ sobre original}} = \frac{3,90}{3,33} = 1,17$$

- d) Fracción paralela de la aplicación A cuando se ejecuta con 4 núcleos.

$$S'_{8nucleos \text{ sobre } 4nucleos} = \frac{1}{1 - F + \frac{F}{S}} \rightarrow 1,7 = \frac{1}{1 - F + \frac{F}{2}}$$

De donde $F=0.8235$.

2. (3 puntos) En un computador Load/Store se pretende mejorar las prestaciones basandose en los resultados obtenidos en un benchmark. De dichos resultados se han obtenido las siguientes estadísticas para las distintas categorías de instrucciones:

Operación	%	CPI
ALU	38	1
LOAD	26	2
STORE	11	2
Salto	25	2

En el análisis realizado se descubre que un porcentaje de los modos *Desplazamiento* utilizados en las instrucciones de acceso a memoria (load/store) tienen un valor de cero en el campo inmediato del formato I. Para mejorar las prestaciones se plantea eliminar el modo Desplazamiento en todas las instrucciones de acceso a memoria y dejar solo el modo Registro Indirecto para estas instrucciones. Esta modificación permitirá reducir el CPI de las instrucciones Load/Store a 1 ciclo de reloj. El inconveniente es que requiere la utilización de una instrucción aritmética adicional cada vez que el campo inmediato es distinto de cero. En concreto, la instrucción load

```
ld r1, 100(r10)
```

se sustituirá por:

```
daddi r10, r10, #100
ld r1, (r10)
```

mientras que la load

```
ld r1, 0(r10)
```

se sustituirá por:

```
ld r1, (r10)
```

Dicha modificación simplifica la ruta de datos y por tanto permite aumentar la frecuencia de reloj un 10 %. Responda a las siguientes preguntas razonando y justificando todas sus respuestas:

- Calcule el CPI promedio del procesador original.
- Sabiendo que se ha observado un valor distinto de cero en el campo inmediato en un 47 % de las load y un 74 % de las store, indique como quedaría la tabla de porcentajes de instrucciones después de realizar la modificación.
- Calcule el CPI promedio del procesador modificado.
- Cuantifique si la mejora propuesta es adecuada o no calculando la aceleración obtenida respecto al procesador original.

Solución:

- Calcule el CPI promedio del procesador original. $CPI_{orig} = 0,38 \cdot 1 + 0,26 \cdot 2 + 0,11 \cdot 2 + 0,25 \cdot 2 = 1,62$.
- Sabiendo que se ha observado un valor distinto de cero en el campo inmediato en un 47 % de las load y un 74 % de las store, indique como quedaría la tabla de porcentajes de instrucciones después de realizar la modificación.

El código generado para el procesador modificado añadirá una instrucción ALU para cada instrucción de acceso a memoria que tenga un valor distinto de cero del código original. Por tanto el nuevo porcentaje de instrucciones ALU será:

$$I_{ALU_{mod}} = I_{ALU_{orig}} + 26 \cdot 0,47 + 11 \cdot 0,74 = 38 + 26 \cdot 0,47 + 11 \cdot 0,74 = 58,36$$

La tabla correspondiente al porcentaje de instrucciones del procesador modificado quedaría:

Operación	%	CPI
ALU	58.36	1
LOAD	26	1
STORE	11	1
Salto	25	2
		120.36

c) Calcule el CPI del procesador modificado.

$$CPI_{mod} = \frac{0,5836 \cdot 1 + 0,26 \cdot 1 + 0,11 \cdot 1 + 0,25 \cdot 2}{1,2036} = 1,2077.$$

d) Cuantifique si la mejora propuesta es adecuada o no calculando la aceleración obtenida respecto al procesador original.

Como nos dicen en el enunciado que la frecuencia de reloj aumenta un 10 % tenemos:

$$f_{mod} = 1,10 f_{orig} \Rightarrow \frac{1}{T_{mod}} = 1,10 \cdot \frac{1}{T_{orig}} \Rightarrow T_{orig} = 1,10 \cdot T_{mod}$$

$$S = \frac{I_{orig} \cdot CPI_{orig} \cdot T_{orig}}{I_{mod} \cdot CPI_{mod} \cdot T_{mod}} = \frac{I_{orig} \cdot 1,62 \cdot 1,10 T_{mod}}{1,2036 \cdot I_{orig} \cdot 1,2077 \cdot T_{mod}} = 1,2259$$

Si que es interesante la propuesta ya que la modificación consigue mejorar las prestaciones un 22.59 % con respecto al procesador original.

□

3. (3.5 puntos)

En un procesador MIPS básico segmentado en las 5 etapas habituales (IF, ID, EX, M, WB) se ejecuta un bucle para realizar la siguiente operación: $\vec{Z} = a + \vec{X} + \vec{Y} + \vec{Z}$.

El diagrama instrucciones-tiempo de una iteración cualquiera del bucle es el siguiente:

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
bucle: ld r2,x(r1)	IF	ID	ID	EX	ME	WB																			
dadd r6,r4,r2		IF	IF	ID	ID	ID	EX	ME	WB																
ld r2,y(r1)			IF	IF	IF	ID	EX	ME	WB																
dadd r6,r6,r2						IF	ID	ID	ID	EX	ME	WB													
ld r2,z(r1)							IF	IF	IF	ID	EX	ME	WB												
dadd r6,r6,r2								IF	ID	ID	ID	EX	ME	WB											
sd r6,z(r1)									IF	IF	IF	ID	ID	ID	EX	ME	WB								
bnez r1,bucle										IF	IF	IF	ID	ID	ID	EX	ME	WB							
dsub r1,r1,#8												IF	ID	EX	ME	WB									
nop														IF	ID	EX	ME	WB							
bucle: ld r2,x(r1)																			IF	ID	ID	EX	ME	WB	

A partir del diagrama, responda a las siguientes cuestiones:

- ¿Con qué otra instrucción del bucle tiene un riesgo la instrucción `ld r2,x(r1)` que causa la inserción de un ciclo de parada en el ciclo 21? ¿A qué se debe este riesgo?
- ¿Cuál es el registro involucrado en la inserción de los ciclos de parada en los ciclos 15 y 16? ¿Para qué se usa su contenido?
- ¿Qué técnica se utiliza para resolver los riesgos de control y cuáles son sus características?

- d) ¿Cuántos ciclos de parada se insertan en total durante la ejecución de una iteración del bucle?
- e) Asumiendo que todas las iteraciones son iguales y que el número de iteraciones es elevado, ¿cuál es el CPI del bucle?
- f) El bucle anterior se adapta para trabajar con vectores de datos en coma flotante de doble precisión, sustituyéndose su código por el siguiente:

```
bucle:  l.d f2,x(r1)
        add.d f6,f4,f2
        l.d f2,y(r1)
        add.d f6,f6,f2
        l.d f2,z(r1)
        add.d f6,f6,f2
        s.d f6,z(r1)
        bnez r1,bucle
        dsub r1,r1,#8
        nop
        trap #0
```

Este código se ejecuta en un procesador MIPS con operador de suma de coma flotante multiciclo con latencia 3 y las siguientes características:

- Resolución de riesgos de datos por cortocircuitos.
- Bancos de registros de enteros y de coma flotante independientes con dos puertos de lectura y uno de escritura cada uno.

Teniendo en cuenta que en este procesador, las instrucciones de enteros y de carga/almacenamiento siguen utilizando las 5 etapas habituales, mientras que las instrucciones de suma en coma flotante pasan por las siguientes etapas IF,ID,A1,A2,A3,WB, dibuja en la hoja adjunta el diagrama correspondiente a la ejecución de las 7 primeras instrucciones del bucle (hasta la `s.d f6,z(r1)` inclusive).

Solución:

- a) ¿Con qué otra instrucción del bucle tiene un riesgo la instrucción `ld r2,x(r1)` que causa la inserción de un ciclo de parada en el ciclo 21? ¿A qué se debe este riesgo?

La instrucción `dsub r1,r1,#8` es la otra instrucción involucrada en el riesgo que se resuelve insertando ciclos de parada. El riesgo se produce porque `dsub r1,r1,#8` produce el contenido de `r1` que la instrucción `ld r2,x(r1)` usa para calcular su dirección efectiva.

- b) ¿Cuál es el registro involucrado en la inserción de los ciclos de parada en los ciclos 15 y 16? ¿Para qué se usa su contenido?

El registro involucrado es el `r6` y su contenido lo usa la instrucción `sd r6,z(r1)` para almacenarlo en memoria.

- c) ¿Qué técnica se utiliza para resolver los riesgos de control y cuáles son sus características?

Delay-slot con latencia 2.

- d) ¿Cuántos ciclos de parada se insertan en total durante la ejecución de una iteración del bucle?

9

- e) Asumiendo que todas las iteraciones son iguales y que el número de iteraciones es elevado, ¿cuál es el CPI del bucle?

$$CPI = 1 + \frac{9}{10} = \frac{19}{10} = 1,9.$$

f) Diagrama instrucciones-tiempo con vectores de datos en coma flotante de doble precisión:

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21
bucle: l.d f2,x(r1)	IF	ID	EX	ME	WB																
add.d f6,f4,f2		IF	id	ID	A1	A2	A3	WB													
l.d f2,y(r1)			if	IF	id	ID	EX	ME	WB												
add.d f6,f6,f2					if	IF	id	ID	A1	A2	A3	WB									
l.d f2,z(r1)						if	IF	id	ID	EX	ME	WB									
add.d f6,f6,f2							if	IF	id	ID	A1	A2	A3	WB							
s.d f6,z(r1)								if	IF	id	ID	EX	ME	WB							
bnez r1,bucle									if	IF	ID	EX	ME	WB							
dsub r1,r1,#8													IF	ID	EX	ME	WB				
nop																IF	ID	EX	ME	WB	
bucle: l.d f2,x(r1)																	IF	ID	EX	ME	WB

□