

Responde cada pregunta en una hoja distinta. Tiempo disponible: 2h

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.
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
Saltos	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)
```

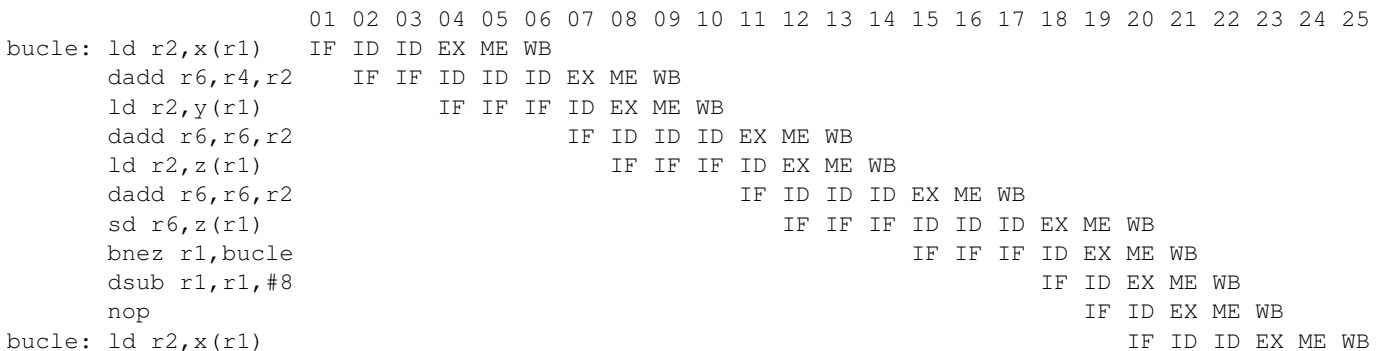
Dicha modificación simplifica la ruta de datos y por tanto permite aumentar la frecuencia de reloj un 10 %. Responda a las siguientes preguntas razonando y justificando todas sus respuestas:

- 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.

### 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:



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?
- ¿Cuántos ciclos de parada se insertan en total durante la ejecución de una iteración del bucle?
- Asumiendo que todas las iteraciones son iguales y que el número de iteraciones es elevado, ¿cuál es el CPI del bucle?
- 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).

