

Parte 2 de 6 -

Preguntas 5 de 10

1 Puntos. Puntos descontados por fallo: 0.3

En el procesador MIPS con operaciones multiciclo:

- ☐ A. Se producen riesgos estructurales por emplear operadores multiciclo segmentados.
- ☐ B. Se producen riesgos estructurales por escritura simultánea en el banco de registros por parte de dos o más instrucciones multiciclo.
- ☐ C. Se producen riesgos RAW, WAW y WAR.
- ☒ D. Todas las anteriores son correctas.

[Borra selección](#)

Preguntas 6 de 10

1 Puntos. Puntos descontados por fallo: 0.3

Indica el número de ciclos de parada y el cortocircuito que aplicaría el procesador MIPS segmentado para resolver los riesgos de datos generados por la secuencia de instrucciones mostrada. Considera que las latencias del multiplicador y del sumador son 7 y 4, respectivamente:

`mul.d $0, $1, $2`

`add.d $0, $0, $3`

- ☒ A. 6 stalls, WBaA1
- ☐ B. 6 stalls, M7aA1
- ☐ C. 0 stalls, WBaA1
- ☐ D. 7 stalls, WBaA1

[Borra selección](#)

Parte 3 de 6 -

Preguntas 7 de 10

1 Puntos. Puntos descontados por fallo: 0.3

En un procesador con predicción dinámica de salto, si la predicción falla:

- ☐ A. Se cancelan las instrucciones buscadas antes de la de salto.
- ☐ B. Todas las anteriores son correctas.
- ☐ C. Se comienza la búsqueda de instrucciones en la dirección destino del salto.
- ☒ D. Se actualiza el estado del predictor.

[Borra selección](#)

Parte 4 de 6 -

Preguntas 8 de 10

1 Puntos

[Pulse para ver instrucciones adicionales](#)

Teniendo en cuenta la ruta de datos del procesador MIPS segmentada en cinco etapas (IF, ID, EX, M, WB), que aplica todos los cortocircuitos posibles para resolver conflictos de datos, que resuelve los conflictos de control mediante la técnica del salto retardado, que calcula la dirección de salto y modifica el PC en la etapa ID, y que no tiene ningún conflicto estructural, calcula el CPI para una iteración del siguiente código (redondea a dos decimales):

`loop: ld $3, 0($2)`

`ld $4, 0($3)`

`dadd $1, $1, $4`
`daddi $0, $0, -1`
`bne $2, $0, loop`

`daddi $2, $2, 8`

`sd $1, 0($0)`

CPI =

Parte 5 de 6 -

Preguntas 9 de 10

1 Puntos

Importante: utiliza únicamente el punto como separador decimal.

Un bucle ejecuta la siguiente secuencia de instrucciones en el procesador MIPS:

```
loop:  l.d f0, X(r10)
      add.d f0,f0,f10
      s.d f0,X(r10)
      dadd r10,r10,#8
      dadd r11,r11,#-1
      bnez r11,loop
```

Durante la ejecución de una iteración, la instrucción `add.d` inserta un ciclo de parada, mientras que la `s.d` inserta 2 ciclos de parada. Si aplicamos la técnica `loop-unrolling` para eliminar los ciclos de parada, incrementando al mínimo el número de instrucciones ¿cuántas iteraciones del bucle original deberían hacerse en el nuevo cuerpo del bucle?

Parte 6 de 6 -

Preguntas 10 de 10

1 Puntos

Importante: utiliza únicamente el punto como separador decimal.

Un procesador dispone de un predictor de saltos BTB que obtiene la predicción en la primera fase del ciclo de instrucción. La condición, dirección de salto y la escritura del PC se realiza en la fase 3 del ciclo de instrucción. Sabiendo que los saltos son efectivos en el 80% de los casos y que el predictor acierta en el 94% de los saltos, calcula el CPI medio de las instrucciones de salto:

Guardar

Enviar para calificar

[PoliformaT](#) [UPV](#)

- [Powered by Sakai](#)
- Copyright 2003-2018 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.

► Información de SW:
► Hora del servidor: