

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

any

Mi poliformaT

ETC-18-19

Gpr

Sar

AIC

Sitios

Miguel Angel

poli **[format]** AIC > **EXÁMENES**

«

Inicio

Guía Docente

Anuncios

Recursos

Tareas

**Exámenes**

Calificaciones

Calendario

Grupos

Videoapuntes

Simuladores MIPS

UT2(2.1;2.2;2.3) Seguimiento

[Volver a la Lista de Exámenes](#)

CPI =  $Tej / I \rightarrow Tej+1/I+1 \rightarrow 5000/2000=2.5, 5020/2020=disminuye$

Parte 1 de 6 -

3.5/ 4.0 Puntos

Preguntas 1 de 10

1.0/ 1.0 Puntos

En el procesador MIPS segmentado, las instrucciones de carga insertan 1 ciclo de parada si la instrucción siguiente consume el dato leído de la memoria. Si el compilador, en esos casos, coloca instrucciones NOP entre ambas instrucciones, el valor del CPI obtenido por el programa (aumenta/disminuye/no cambia) disminuye

Preguntas 2 de 10

1.0/ 1.0 Puntos. Puntos descontados por fallo: 0.3

Respecto al salto retardado en el MIPS, indica cuál de las siguientes afirmaciones es CIERTA

☒ A. El compilador puede insertar instrucciones en el hueco del salto siempre que no modifiquen el comportamiento correcto del código

☐ B. Utilizar el salto retardado lleva en el caso de que la instrucción de salto no salte a una pérdida de prestaciones al insertarse ciclos de parada

☐ C. El salto retardado implica que el procesador incluirá un número de ciclos de parada para asegurar que la ejecución del código sea correcta

☐ D. El hueco (delay slot) del salto retardado es siempre de 3 ciclos, independientemente de en qué etapa se escriba el PC

Preguntas 3 de 10

1.0/ 1.0 Puntos. Puntos descontados por fallo: 0.3

Si un procesador soporta un comportamiento preciso frente a las excepciones:

☐ A. Las instrucciones posteriores a la excepción terminan correctamente.

☐ B. Las instrucciones anteriores a la excepción se cancelan.

☒ C. Se puede identificar la instrucción causante de la excepción.

☐ D. Todas las anteriores son correctas.

Preguntas 4 de 10

0.5/ 1.0 Puntos

Indica en cuáles de los siguientes fragmentos de código se aplicaría un cortocircuito WBaEX:

☒ A.

l.d F0, 0(R0)  
  
add R1, R1, R2  
  
s.d F0, 20(R0)

☒ B.

dadd r1,r2,r3  
  
and r20,r2,r3  
  
ld r3,100(r1)

☒ C.

l.d F0, 0(R0)  
  
s.d F0, 20(R0)  
  
add R1, R1, R2

☐ D.

dadd R1, R2, R3  
  
dadd R4, R1, R5  
  
dsub R6, R4, R7

Parte 2 de 6 -

0.7/ 2.0 Puntos

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.

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

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.

[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 $t3, 0($t2)`

`ld $t4, 0($t3)`

`dadd $t1, $t1, $t4`  
`daddi $t0, $t0, -1`  
`bezt $t0, loop`

`daddi $t2, $t2, 8`

`sd $t1, 0($t0)`

CPI = 1.57

**Importante:** utiliza únicamente el punto como separador decimal.

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

1. `addi $t0, $t0, 1`

```
loop:  l.d r0, 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? 4

Parte 6 de 6 -

0.0/ 1.0 Puntos

Preguntas 10 de 10

0.0/ 1.0 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: 2.25

[PollformaT](#) [UPV](#)

- [Powered by Sakai](#)
- Copyright 2003-2019 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:

<

>