

Exámenes

UT2A (2.1,2.2,2.3) Examen (Castellano)

[Volver a la Lista de Exámenes](#)

Parte 1 de 7 -

3.5 Puntos

Preguntas 1 de 14

0.7 Puntos

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

- ☒ `l.d F0, 0(R0)`
`add R1, R1, R2`
`s.d F0, 20(R0)`
- ☒ `dadd r1,r2,r3`
`and r20,r2,r3`
`ld r3,100(r1)`
- ☐ `dadd R1, R2, R3`
`dadd R4, R1, R5`
`dsub R6, R4, R7`
- ☐ `l.d F0, 0(R0)`
`s.d F0, 20(R0)`
`add R1, R1, R2`

Respuesta correcta: A, B

Preguntas 2 de 14

0.7 Puntos

Importante: utiliza únicamente el punto como separador decimal.

Si un procesador segmentado calcula la dirección y condición de salto así como la escritura del PC en la fase 2 del ciclo de instrucción, cuando se emplea el predict-not-taken para resolver los riesgos del control y el salto es efectivo, se cancelan ☒ 1 instrucciones.

Respuesta correcta: 1,

Preguntas 3 de 14

0.7 Puntos. Puntos descontados por fallo: 0.21

Indica en cuál/es de los siguientes casos se ocasiona un conflicto estructural:

- ☐ Una instrucción accede a memoria para leer o escribir un dato y el procesador tiene memorias cache separadas de datos e instrucciones
- ☒ Una instrucción accede a memoria para leer o escribir un dato y el procesador tiene una memoria cache unificada de datos e instrucciones
- ☐ Dos instrucciones aritméticas calculan su operación en la misma etapa en ciclos consecutivos
- ☐ Dos instrucciones acceden en el mismo ciclo de reloj al banco de registros, una para realizar una escritura de un resultado y la otra para leer operandos. El banco de registros tiene un puerto de lectura por operando y un puerto de escritura

Respuesta correcta: B

Preguntas 4 de 14

0.7 Puntos. Puntos descontados por fallo: 0.21

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:

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

```
sd r1, 200(r11)
```

- ☐ 0 stalls, WBaEX
- ☐ 0 stalls, MEMaMEM
- ☐ 1 stalls, WBaMEM
- ☒ 0 stalls, WBaMEM

Respuesta correcta: D

Preguntas 5 de 14

0.7 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

Respuesta correcta: disminuye

Parte 2 de 7 -

2.1 Puntos

Preguntas 6 de 14

0.7 Puntos

Dado el fragmento de código MIPS que se muestra a continuación:

```

1 bnez r1,loop
2 l.d f0,100(r10)
3 add.d f4,f0,f2
4 s.d f4,100(r10)
5 l.d f0,200(r10)
6 sub.d f4,f0,f3
7 s.d f4,200(r10)

```

relaciona cada par de instrucciones con un tipo de dependencia:

A. i4 y i6

B. i3 y i6

C. i2 y i3

D. i1 y i6



C

1. Dependencia de datos



A

2. Antidependencia



B

3. Dependencia de salida



D

4. Dependencia de control

Respuesta correcta: 1:C, 2:A, 3:B, 4:D

Preguntas 7 de 14

0.7 Puntos. Puntos descontados por fallo: 0.21

Asumiendo una ruta de datos con un operador multiciclo de multiplicación en coma flotante con tiempo de evaluación (o latencia) de 4 ciclos y tasa de inicio (issue rate) de $\frac{1}{4}$ ciclos, podemos afirmar:

- ☐ El operador no está segmentado y tiene un tiempo de ejecución de un cuarto de ciclo.
- ☐ El operador está segmentado y permite introducir a ejecución una operación cada cuatro ciclos.
- ☒ El operador no está segmentado.
- ☐ El operador podrá ejecutar al mismo tiempo cuatro instrucciones, pero cada una de ellas en una etapa distinta.

Respuesta correcta: C

Indica el número de ciclos de parada que aplicaría el procesador MIPS segmentado para resolver el riesgo generado por la secuencia de instrucciones mostrada. Considera que las latencias del multiplicador y del sumador son 4 y 3, respectivamente:

```
mul.d f0, f1, f2
```

```
add.d f0, f3, f4
```

-
- ☐ 3 stalls
 - ☐ 4 stalls
 - ☒ 1 stalls
 - ☐ 0 stalls

Respuesta correcta: C

Preguntas 9 de 14

0.7 Puntos. Puntos descontados por fallo: 0.21

Indica cual de las siguientes afirmaciones es CIERTA:

- ☐ Un predictor de saltos de dos bits con saturación predice salto efectivo solamente cuando el contador satura
- ☐ Un predictor de saltos actualiza el contador (estado) en el momento de hacer la predicción
- ☐ Un predictor de saltos de dos bits acierta el doble de veces que un predictor de un bit
- ☒ Un predictor de saltos de dos bits con histéresis, después de dos saltos consecutivos no efectivos de la misma instrucción predecirá salto no efectivo para esa instrucción

Respuesta correcta: D

Preguntas 10 de 14

0.7 Puntos. Puntos descontados por fallo: 0.21

Un predictor de dos niveles (global=4,local=3) obtiene la predicción de una instrucción de salto:

- ☐ teniendo en cuenta el comportamiento del salto en cuestión y de la última instrucción de salto ejecutada
- ☐ necesita 2^3 tablas para almacenar la predicción de cuatro bits.
- ☐ teniendo en cuenta únicamente el comportamiento del salto en cuestión
- ☒ teniendo en cuenta el comportamiento del salto en cuestión y de las cuatro últimas instrucciones de salto ejecutadas

Respuesta correcta: D

Parte 4 de 7 -

1.0 Puntos

Preguntas 11 de 14

1.0 Puntos

Importante: utiliza únicamente el punto como separador decimal.

Teniendo en cuenta la ruta de datos del procesador MIPS segmentada en cinco etapas (IF: etapa 1 del ciclo de instrucción, ID: etapa 2, EX: etapa 3, M: etapa 4, WB: etapa 5), que aplica todos los cortocircuitos posibles para resolver conflictos de datos, que resuelve los conflictos de control mediante la técnica *predict-not-taken*, que calcula la condición de salto en la etapa 2 del ciclo de instrucción y que modifica el PC en la etapa 3, y que no tiene ningún conflicto estructural, calcula el CPI para un alto número de iteraciones del bucle en el siguiente código:

```
loop: ld r3, 0(r2)
      ld r4, 0(r3)
      sd r4, 1024(r3)
      dadd r1, r1, r4
      daddi r10, r10, -1
      daddi r2, r2, 8
      bnez r10, loop
      sd r1, 0(r11)
      <sgte+1>
      <sgte+2>
      <sgte+3>
```

CPI = ✓ 1.43**Respuesta correcta:** 1.43

Parte 5 de 7 -

1.0 Puntos

Preguntas 12 de 14

1.0 Puntos


Importante: utiliza únicamente el punto como separador decimal.

Sea el siguiente código que se ejecuta en un procesador MIPS:

```
mul.d F2, F0, F0  
beqz r1,etiqueta  
add.d F2, F1, F1
```

El procesador resuelve los conflictos de datos mediante ciclos de parada y cortocircuitos, mientras que los conflictos de control los resuelve con la técnica *predict-not-taken*, actualizando el PC en la etapa ID. El procesador dispone de una unidad de suma en coma flotante con $Tev=3$ e $IR=1/2$ y un multiplicador multiciclo con $Tev=6$ e $IR=1/2$.

Si el valor de r1 es distinto de cero, ¿cuantos ciclos de parada se introducirán en la ejecución de la instrucción

add.d?  2

Respuesta correcta: 2,

Parte 6 de 7 -

1.0 Puntos

Preguntas 13 de 14

1.0 Puntos

Importante: utiliza únicamente el punto como separador decimal.

Sea el siguiente código en ensamblador:

```
nozero:      li t0, 13          # Número de elementos del vector
             li v0, 0          # contador inicial = 0
             li t1, v          # dirección vector V
loop:        lw t2, 0(t1)       # lectura V[i]
             addi t0, t0, -1     # Decrementa elementos vector
             bnez t2, sigue     # Si V[i] es distinto de cero salta
             addi v0, v0, 1     # Incrementa contador
sigue:       addi t1, t1, 4      # Incrementa dirección vector V
             bnez t0, loop      # Siguierte iteración
```

Dicho código implementa la función *nozero* que calcula el número de elementos de un vector de 13 elementos con valor distinto a cero.

El código se ejecuta en un procesador segmentado de 5 etapas el cual resuelve todos los conflictos de datos con cortocircuitos. El procesador implementa un BTB con un predictor de dos bits con saturación. La tabla tiene 16 entradas e inicialmente está vacía. En ausencia de historia del salto se utiliza *predict-not-taken*. Cuando la información del salto se almacena por primera vez en la BTB, el estado del predictor se pone a "00" (*Strongly Not Taken*) si el salto no es efectivo y a "11" (*Strongly Taken*) en caso contrario. Un fallo de predicción ocasiona la inserción de 2 ciclos de parada.

Indica cuantos ciclos de penalización introducirá cada instrucción de salto en la ejecución del código anterior para el caso de un vector que contenga elementos con los valores "00100000000000"

a) Ciclos de penalización *bnez t2, sigue*: ~~5~~ 5 ciclos

b) Ciclos de penalización *bnez t0, loop*: ~~5~~ 5 ciclos

Respuesta correcta: 2,4

Parte 7 de 7 -

0.0 Puntos

Preguntas 14 de 14

0.0 Puntos

Adjunte el archivo con la justificación manuscrita de los tres ejercicios (las tres últimas preguntas) del examen.

Sólo se admiten manuscritos, no se admiten ficheros editados electrónicamente (p.ej. MS Word, MS Excel, ...)

Se deberá indicar a mano en la cabecera de la primera hoja del PDF el nombre del alumno y apellidos.

Sólo se admite formato PDF.

El nombre del archivo deberá ser <Apellido1_Apellido2_Nombre>.pdf

Scherle Carboneres Fabián.pdf(1.204,77 KB)

- [PoliformaT](#)
- [UPV](#)
- [Powered by Sakai](#)
- Copyright 2003-2021 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.