

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

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

Preguntas 2 de 14

0.7 Puntos. Puntos descontados por fallo: 0.21

Una unidad segmentada se compone de 4 etapas de retardos 20ns,20ns,11ns,13ns. Los registros intermedios tienen un retardo de 2ns y el desfase del reloj es como máximo de 1ns. El periodo de reloj mínimo debería ser:

- ☐ 64 ns
- ☐ 21 ns
- ☒ 23 ns
- ☐ 20 ns

Respuesta correcta: C

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

- ✗ ☐ Dos instrucciones aritméticas calculan su operación en la misma etapa en ciclos consecutivos
- ☐ Una instrucción accede a memoria para leer o escribir un dato y el procesador tiene memorias cache separadas de datos e instrucciones
- ☐ 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
- ✓ ☒ Una instrucción accede a memoria para leer o escribir un dato y el procesador tiene una memoria cache unificada de datos e instrucciones

Respuesta correcta: D

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


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

Respuesta correcta: B, D

Preguntas 5 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,

Parte 2 de 7 -

2.1 Puntos

Preguntas 6 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. Considera que las latencias del multiplicador y del sumador son 7 y 4, respectivamente:

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

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

-
- ☐ 7 stalls, WBaA1
 - ☐ 6 stalls, M7aA1
 - ☒ 6 stalls, WBaA1
 - ☐ 0 stalls, WBaA1

Respuesta correcta: C

Preguntas 7 de 14

0.7 Puntos. Puntos descontados por fallo: 0.21

En el procesador MIPS con operaciones multiciclo:

-
- ☐ Se producen riesgos estructurales por emplear operadores multiciclo segmentados.
 - ☐ Se producen riesgos estructurales por escritura simultánea en el banco de registros por parte de dos o más instrucciones multiciclo de coma flotante.
 - ☒ Todas las anteriores son correctas.
 - ☐ Se producen riesgos estructurales por la ejecución simultánea de instrucciones enteras y de coma flotante.

Respuesta correcta: B

Preguntas 8 de 14

0.7 Puntos. Puntos descontados por fallo: 0.21

Asumiendo una ruta de datos segmentada con operadores multiciclo de multiplicación (TE=4, IR=1) y de suma (TE=2, IR=1) y con banco de registros de enteros y coma flotante separados, indica qué afirmación es CIERTA:

- ☐ La ruta de datos podría introducir conflictos estructurales en la etapa WB entre una instrucción que escriba en los registros enteros y otra que escriba en los de coma flotante.
- ☒ La ruta de datos puede generar conflictos WAW que lleven a una ejecución errónea
- ☐ Todos los conflictos estructurales se pueden resolver al utilizar dos bancos de registros diferenciados de coma flotante y de enteros
- ☐ La ruta de datos no tiene necesidad de insertar ciclos de parada ya que todos los conflictos se pueden resolver por cortocircuitos

Respuesta correcta: B

Parte 3 de 7 -

1.4 Puntos

Preguntas 9 de 14

0.7 Puntos

Sea un procesador con un predictor de saltos de 2 bits con saturación, donde se ejecuta un código con una instrucción de salto que implementa un bucle que realiza 100 iteraciones. Ten en cuenta que el procesador realiza una predicción *predict-not-taken* ante la ausencia de historia de la instrucción de salto, que el procesador no tiene historia reciente de la instrucción de salto, y que la primera vez que se inserta un salto en la tabla su estado será *strongly-taken* o *strongly-not taken*, según proceda. En estas condiciones, durante la ejecución del bucle, se generarán en total ☒ 2 fallos de predicción.

Respuesta correcta: 2|dos

Preguntas 10 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 histéresis, después de dos saltos consecutivos no efectivos de la misma instrucción predecirá salto no efectivo para esa instrucción
- ☐ 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 saturación predice salto efectivo solamente cuando el contador satura
- ☐ Un predictor de saltos actualiza el contador (estado) en el momento de hacer la predicción

Respuesta correcta: A

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 ciclos de parada, que calcula la condición de salto en la etapa 2 del ciclo de instrucción y que modifica el PC en la etapa 2, 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 = ✓ 1.29**Respuesta correcta:** 1.29

Parte 5 de 7 -

1.0 Puntos


Preguntas 12 de 14

1.0 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 1 ciclo de parada, mientras que la `s.d` inserta 3 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 incluirse en el nuevo cuerpo del bucle?  4

Respuesta correcta: 4,

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 1 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 "00 100000000000"

a) Ciclos de penalización `bnez t2, sigue`: ✓ 1 ciclos

b) Ciclos de penalización `bnez t0, loop`: ✓ 2 ciclos

Respuesta correcta: 1,2

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

[Downloads.zip](#)(2.930,89 KB)

- [PoliformaT](#)
- [UPV](#)
- [Powered by Sakai](#)
- Copyright 2003-2020 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.