

Exámenes

2022-UT2A (2.1,2.2,2.3) Prueba de seguimiento (Castellano)

[Volver a la Lista de Exámenes](#)

Parte 1 de 6 / 3.5 Puntos

Preguntas 1 de 13

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)
```

- ✓ A. 1 stalls, WBaMEM
- ✓ B. 0 stalls, WBaEX
- ✓ C. 0 stalls, MEMaMEM
- ✓ D. 0 stalls, WBaMEM

Respuesta correcta: D

Preguntas 2 de 13

0.7 Puntos. Puntos descontados por fallo: 0.21

Indica cuál de los siguientes casos se ocasiona un riesgo estructural:

- ✓ A. 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.

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

Respuesta correcta: B

Preguntas 3 de 13

0.7 Puntos. Puntos descontados por fallo: 0.21

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

- ✓ A. Las instrucciones anteriores a la excepción se cancelan.
- ✓ B. Las instrucciones posteriores a la excepción terminan correctamente.
- ✓ C. Todas las demás respuestas son correctas.
- ✓ D. Se puede identificar la instrucción causante de la excepción.

Respuesta correcta: D

Preguntas 4 de 13

0.7 Puntos

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

A. -----

l.d f0, 0(r0)

×

s.d f0, 20(r0)

add r1, r1, r2

B. -----

dadd r1,r2,r3

and r20,r2,r3

```
ld r3, 100(r1)
```

C. -----

```
l.d f0, 0(r0)
```

```
add r1, r1, r2
```

```
s.d f0, 20(r0)
```

D. -----

```
dadd r1, r2, r3
```

```
dadd r4, r1, r5
```

```
dsub r6, r4, r7
```

Respuesta correcta: B, C

Preguntas 5 de 13

0.7 Puntos. Puntos descontados por fallo: 0.21

Teniendo en cuenta la ruta de datos segmentada del procesador MIPS en las etapas IF, ID, EX, M y WB, y teniendo en cuenta que el ciclo de reloj es de 10 ns, indica qué respuesta es CIERTA:

- ✓ A. La aceleración que se obtendría, en comparación a la ruta de datos sin segmentar, es de 5, independientemente de la duración de cada etapa
- ✓ B. El tiempo de lectura o escritura en el banco de registros no puede ser inferior a 10 ns
- ✓ C. La suma de retardos del registro de segmentación y la etapa más lenta no puede superar los 10 ns
- ✓ D. Todas las etapas deben tener un retardo idéntico a 10 ns

Respuesta correcta: C

Parte 2 de 6 / 2.1 Puntos

Preguntas 6 de 13

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 iniciación (initiation rate) de $\frac{1}{4}$ de ciclo, podemos afirmar que:

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

Respuesta correcta: A

Preguntas 7 de 13

0.7 Puntos. Puntos descontados por fallo: 0.21

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

- ✓ A. Para evitar riesgos WAW, deben insertarse ciclos de parada.
- ✓ B. Todos los riesgos estructurales se pueden resolver al utilizar dos bancos de registros diferenciados de coma flotante y de enteros.
- ✓ C. La ruta de datos podría introducir riesgos estructurales en la etapa WB entre una instrucción que escriba en los registros enteros y otra que escriba en los de coma flotante.
- ✓ D. La ruta de datos no tiene necesidad de insertar ciclos de parada ya que todos los riesgos se pueden resolver por cortocircuitos.

Respuesta correcta: A

Preguntas 8 de 13

0.7 Puntos. Puntos descontados por fallo: 0.21

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
```

- ✓ A. 3 stalls
- ✓ B. 4 stalls
- ✓ C. 1 stalls
- ✓ D. 0 stalls

Respuesta correcta: C

Parte 3 de 6 / 1.4 Puntos

Preguntas 9 de 13

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 13

0.7 Puntos. Puntos descontados por fallo: 0.21

En un procesador con predicción dinámica de salto, si el salto finalmente es "no tomado" y la predicción falla:

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

Respuesta correcta: D

Parte 4 de 6 / 1.0 Puntos

Preguntas 11 de 13

1.0 Puntos

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 6 / 1.0 Puntos

Preguntas 12 de 13

1.0 Puntos

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

```

loop:      l.d f0, 0(r1)
           mul.d f1, f0, f0
           s.d f1, 0(r2)
           dadd r1, r1, 8
           dadd r3, r3, -8
           dadd r2, r2, 8
           bnez r3, loop

```

El procesador resuelve los riesgos de datos mediante ciclos de parada y cortocircuitos, mientras que los riesgos de control los resuelve con la técnica *predict-not-taken*, actualizando el PC en la etapa ID. El procesador dispone de un multiplicador segmentado multiciclo con $T_{ev}=5$.

Aplica la técnica de loop unrolling con el fin de reducir al máximo los ciclos de parada por riesgos de datos, así como incrementar 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 6 / 1.0 Puntos

Preguntas 13 de 13

1.0 Puntos

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      # Siguiente iteración

```

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

El código se ejecuta en un procesador segmentado de 5 etapas el cual resuelve todos los riesgos 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 3 ciclos de parada.

Indica cuantos ciclos de penalización en total introducirá cada instrucción de salto en la ejecución del código anterior para el caso de un vector que contenga elementos con los valores "0 0 1 0 0 0 1 0 0 1 0 0

0"

a) Ciclos de penalización bnez t2, sigue: ~~X~~18 ciclos

b) Ciclos de penalización bnez t0, loop: ☒ 6 ciclos

Respuesta correcta: 9, 6