

Relacion-de-problemas-del-tema-2...



L0R3N



Estructura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

ZERO AZÚCAR
**#ZERO
PALABRAS**

DEMASIADO BUENO PARA
EXPLICARLO CON PALABRAS

Coca-Cola
Real Magic™

REAL MAGIC, COCA-COLA ZERO son marcas registradas de The Coca-Cola Company.



quieres trabajar
en Wuolah??

TE BUSCAMOS

campus virtual
ETSI Informática

CV ► ETSI Informática ► Mis asignaturas en este Centro ► Curso académico 2019-2020 ► Grado en Ingeniería del Software ► Estructura de Computadores (2019-20. Grupo C, Gr. Ingeniería del Software, Gr. Ingeniería de Computadores, Gr. Ingeniería Informática) ► Tema 2: Mejora del Rendimiento del Procesador con la Segmentación ► Relación de problemas del tema 2 (Segmentación)

Comenzado el	sábado, 23 de noviembre de 2019, 17:53
Estado	Finalizado
Finalizado en	sábado, 23 de noviembre de 2019, 17:53
Tiempo empleado	6 s
La puntuación	83,00/83,00
Calificación	10,00 de 10,00 (100%)

Pregunta 1
Correcta
Puntúa 1,00 sobre 1,00
Marcar pregunta

Sea el programa "suma1" dado por el siguiente código MIPS:

```
#1      sub $5, $0, $0
#2 suma: lw $10, 1000($20)
#3      add $5, $5, $10
#4      addi $20, $20, -4
#5      bne $20, $0, suma
```

Los distintos apartados se resolverán en clase a modo de ejemplo.

Calcula el número de ciclos en la implementación monociclo cuando el número de iteraciones es 1000:

Pregunta 2
Correcta
Puntúa 8,00 sobre 8,00
Marcar pregunta

Considerar la estructura segmentada en cinco etapas del procesador MIPS, con hardware para la detección de riesgos por dependencias de datos e **inserción de burbujas** para garantizar la correcta ejecución de los programas. Respecto a las dependencias de control, el controlador implementa la suposición de *salto no realizado*. Todos los saltos se resuelven en la etapa de decodificación. El banco de registros permite la lectura y escritura simultánea de un mismo registro sin conflicto. Sea el siguiente programa, donde se han numerado las instrucciones para facilitar referenciarlas:

```
#1      addi $5, $0, 0
#2 etiq: lw $11, 3000($22)
#3      add $5, $5, $11
#4      and $11, $11, $5
#5      sw $11, 3000($22)
#6      subi $22, $22, 4
#7      bne $22, $0, etiq
#8      subi $22, $22, 4
#9      sw $5, 3000($22)
```

a) Realiza un diagrama temporal multiciclo donde se vea la evolución del código a través del cauce segmentado. Mostrar tanto la posibilidad de éxito en el salto como la contraria. De acuerdo con este diagrama:

- Para resolver los riesgos por dependencias de datos, ¿cuántas burbujas ha sido necesario introducir por cada iteración del bucle? ¿y cuántas cuando se sale del mismo?
- ¿Cuántos ciclos se pierden cada vez que en la instrucción de salto (#7) se falla en la predicción?
- Para ejecutar correctamente la instrucción #3 ha sido necesario
- ¿Cuántas veces ha sido útil la anticipación dentro del banco de registros para ahorrar ciclos por cada iteración del bucle? ¿y cuándo se sale del mismo?

b) Suponer que el bucle se efectúa 1000 veces, es decir, el salto tiene éxito 999 veces seguidas y a continuación no tiene éxito:

- La instrucción #8 se ha cancelado veces
- ¿Cuántas instrucciones han terminado de ejecutarse?
- ¿Cuántos ciclos han transcurrido en total?
- ¿Cuál sería el CPI para ese trozo de código?

c) Suponer ahora que se activan todos los cortocircuitos vistos en clase. Vuelve a realizar el diagrama temporal multiciclo y responde a las preguntas siguientes:

- Para resolver los riesgos por dependencias de datos, ¿cuántas burbujas ha sido ahora necesario introducir en cada iteración del código?
- Para ejecutar correctamente la instrucción #3 ha sido necesario
- Para ejecutar correctamente la instrucción #5 ha sido necesario
- ¿Cuántos ciclos han transcurrido en total (1000 iteraciones)?
- ¿Cuál sería el nuevo CPI?

Pregunta 3
Correcta
Puntúa 8,00 sobre 8,00
Marcar pregunta

Considerar la estructura segmentada en cinco etapas del procesador MIPS, con hardware para la detección de riesgos por dependencias de datos e **inserción de burbujas** para garantizar la correcta ejecución de los programas. Respecto a las dependencias de control, el controlador implementa la suposición de *salto no realizado*. Todos los saltos se resuelven en la etapa de decodificación. El banco de registros permite la lectura y escritura simultánea de un mismo registro sin conflicto. Sea el siguiente programa, donde se han numerado las instrucciones para facilitar referenciarlas:

sin ánimo
de lucro,
chequea esto:



tú puedes
ayudarnos a
llevar
WUOLAH
al siguiente
nivel
(o alguien que
conozcas)

WUOLAH

```
#1      or $3, $0, $2
#2  ite:  add $4, $2, $3
#3      and $7, $3, $4
#4      beq $3, $2, fin
#5      bne $4, $7, ite
#6  fin:  lw $3, 100($7)
```

Realiza un diagrama temporal multiciclo donde se vea la evolución del código a través del cauce segmentado. Ten en cuenta que a partir de la propia semántica del programa se puede deducir la resolución de los saltos condicionales. Contesta las siguientes preguntas:

- Para resolver los riesgos por dependencias de datos ¿cuántas burbujas ha sido necesario introducir en la ejecución completa del código, hasta que se termina de ejecutar la instrucción #6? ✓
- Para resolver los riesgos por dependencias de control ¿cuántos ciclos de pérdida han sido necesarios? ✓
- ¿Cuántas veces se ha ejecutado el salto de la instrucción #5? ✓
- ¿Cuántas instrucciones se han cancelado por fallos en la predicción? ✓
- ¿Cuántas instrucciones han terminado de ejecutarse? ✓
- ¿Cuántos ciclos han transcurrido en total? ✓
- CPI resultante: ✓

Suponer ahora que **se activan todos los cortocircuitos** vistos en clase. Vuelve a realizar el diagrama temporal multiciclo y responde a las preguntas siguientes:

- Para ejecutar correctamente la instrucción #2 ha sido necesario ✓
- Para ejecutar correctamente la instrucción #3 ha sido necesario ✓
- ¿Cuántos ciclos han transcurrido en total? ✓
- CPI resultante: ✓

Pregunta 4

Correcta

Puntúa 6,00 sobre 6,00

✓ Marcar pregunta

Considerar la estructura segmentada en cinco etapas del procesador MIPS, con hardware para la detección de riesgos por dependencias de datos e **inserción de burbujas** para garantizar la correcta ejecución de los programas. Tampoco hay anticipación en el banco de registros. Supongamos que se han fusionado la memoria de instrucciones y la de datos en una sola memoria común. Como consecuencia aparece un nuevo tipo de riesgo cuando dos instrucciones intentan, en el mismo ciclo, acceder a la memoria (una de ellas durante su fase de búsqueda de instrucción y la otra durante la fase de acceso a memoria). Para resolver el riesgo la unidad de detección de riesgos tiene que ser capaz de detectarlo. La resolución consistirá en retrasar la ejecución de la nueva instrucción (no dejando que entre en la fase de búsqueda de instrucción) mientras la instrucción anterior termina de acceder a memoria.

Rellena un diagrama multiciclo que muestre la evolución temporal del siguiente código en este nuevo procesador segmentado:

```
#1      lw $8, 3000($7)
#2      sub $4, $4, $5
#3      sw $5, 3000($8)
#4      add $3, $4, $5
#5      and $5, $4, $5
#6      sub $9, $4, $0
#7      or $8, $8, $5
#8      muli $5, $8, 100
```

Contesta las siguientes preguntas:

- ¿Cuántas burbujas han sido necesarias introducir en la ejecución completa del código, hasta que se termina de ejecutar la instrucción #8? ✓
- ¿Cuántas de estas burbujas han sido debidas a los conflictos por acceder a memoria? ✓
- ¿Cuántos ciclos ha tardado el código completo en ejecutarse? ✓
- Calcula el CPI: ✓

Pregunta 5

Correcta

Puntúa 6,00 sobre 6,00

✓ Marcar pregunta

Sea una arquitectura RISC con un conjunto de instrucciones similar al del procesador MIPS visto en clase. Su camino de datos presenta una segmentación en 3 etapas y una única vía de acceso a memoria, común para instrucciones y datos. Esto hace que las instrucciones se ejecuten según se muestra a continuación:

Instrucción	1	2	3	4	5
I1	IFD	REX	MEW		
I2		IFD	REX	MEW	
I3			IFD	REX	MEW

donde

- IFD es la etapa de búsqueda y decodificación de instrucción.
- REX es la etapa de búsqueda de operandos, ejecución de operación ó resolución de condiciones de salto y cálculo de dirección para operando destino ó salto.
- MEW es la etapa de acceso a memoria para instrucciones de carga/almacenamiento, escritura de resultados en registros para instrucciones aritmético/lógicas y actualización del PC para instrucciones de salto.

Además no podrá leerse en un mismo ciclo un registro que va a ser escrito en dicho ciclo (es decir, no hay anticipación en el banco de registros). Bajo estas condiciones, responder a las siguientes cuestiones:

a) Analiza qué tipo de riesgos pueden darse:

- ¿Se pueden producir riesgos de datos RAW? ✓
- ¿Son posibles los riesgos de control? ✓
- ¿Se pueden resolver totalmente los riesgos que puedan existir? ✓



Cerveceros de España recomienda el consumo responsable.

Cuando disfrutas de tu gente y de la cerveza,
con cabeza, disfrutas el doble.



**UNA GRAN CERVEZA.
UNA GRAN RESPONSABILIDAD.**

b) Supóngase que se añade una unidad de detección de riesgos que detiene la ejecución de una instrucción cuando detecta un riesgo y además el controlador aplica la suposición de salto no realizado. Representar en un diagrama multiciclo la evolución del cauce para el siguiente trozo de código.

```
#1      sw $5, 3000($7)
#2      or $5, $4, $5
#3      sub $8, $4, $7
#4      lw $4, 3000($8)
#5      bne $7, $0, etiq
#6      add $8, $8, $5
#7 etiq: sw $5, 3000($8)
```

Considera las dos posibilidades de resolución del salto y responde a las siguientes preguntas:

- Cuando el salto no se efectúa:
 - ¿cuántos ciclos se detiene el cauce debido a riesgos de datos? ✓
 - ¿cuántos ciclos se detiene el cauce debido a riesgos estructurales? ✓
 - ¿cuántos ciclos se pierden debido a los riesgos de control? ✓
 - ¿Cuántos ciclos tardó en ejecutarse el código? ✓
 - ¿Cuántas instrucciones se ejecutan? ✓
 - ¿Cuál es el CPI resultante? ✓
- Cuando el salto se efectúa:
 - ¿cuántos ciclos se detiene el cauce debido a riesgos de datos? ✓
 - ¿Cuántos ciclos tardó en ejecutarse el código? ✓
 - ¿Cuántas instrucciones se ejecutan? ✓
 - ¿Cuál es el CPI resultante? ✓

Pregunta 6

Correcta

Puntúa 4,00 sobre 4,00

✓ Marcar pregunta

Betta, una empresa dedicada a la fabricación de microprocesadores segmentados tipo MIPS, quiere saber si tendría éxito un nuevo procesador antes de empezar a fabricarlo. Hasta ahora, con el modelo M1, trataban los riesgos de control con *detención del cauce*. En el nuevo modelo, el M2, han introducido una mejora en el hardware, la de suponer los saltos no realizados. Ambas estrategias se corresponden fielmente con las vistas en clase para el MIPS. **La terminación del salto es en la etapa de memoria.**

De los programas que usan como bancos de prueba, tienen la siguiente información en cuanto a la probabilidad con lo que aparece cada tipo de instrucción:

- 4% de llamadas a procedimientos.
- 3% de saltos incondicionales.
- 25% de saltos condicionales, de los cuales en el 42% la condición que se evalúa es cierta.
- El resto son operaciones aritmético-lógicas y de transferencia de datos.

Se ha utilizado el mismo compilador tanto para M1 como para M2, pero en M2 se ha conseguido doblar la frecuencia de funcionamiento del procesador. Los riesgos de datos no hay que tenerlos en cuenta, ya que van resueltos en el mismo procesador.

Con todos estos datos, responder a las siguientes cuestiones:

a) ¿Qué funcionalidad debe realizar el hardware que se ha añadido en M2 para el tratamiento de saltos?

Si falla la predicción (salto realizado) deben eliminarse del cauce las instrucciones en IF, ID, EX ✓

c) Calcular el CPI de ambos procesadores para los programas de prueba (redondea a 2 decimales).

CPI de M1: ✓

CPI de M2: ✓

d) Por una encuesta de mercado, dados los costes adicionales que tiene la fabricación de M2, éste solo tendría éxito si ejecutara los programas al menos en la mitad de tiempo que su antecesor, el M1. Suponiendo que no te interesa la ruina de Betta, ¿aconsejarías a ésta la fabricación de M2 o esperar a que se mejore el diseño?

La aceleración que se obtiene con M2 es ✓ (redondeado a 2 decimales) y por lo tanto ✓ recomiendo su fabricación.

Pregunta 7

Correcta

Puntúa 4,00 sobre 4,00

✓ Marcar pregunta

Considerar la estructura segmentada en cinco etapas del procesador MIPS, con hardware para la detección de riesgos por dependencias de datos y **todos los caminos de anticipación (cortocircuitos) habilitados**. Respecto a las dependencias de control, el controlador implementa la *suposición de salto no realizado*. **Todos los saltos se resuelven en la etapa de memoria.** El banco de registros permite la lectura y escritura simultánea de un mismo registro sin conflicto. Sea el siguiente programa, donde se han numerado las instrucciones para facilitar referenciarlas:

```
#1 cambia: xor $4, $4, $5
#2          xor $5, $4, $5
#3          xor $4, $4, $5
#4          jr $31
#5 inicio: lw $4, 1000($0)
#6          lw $5, 1004($0)
#7          beq $4, $5, sigue
#8          jal cambia
#9 sigue:  sw $4, 1008($0)
```

Se sabe que las posiciones de memoria 1000, 1004 y 1008 almacenan palabras cuyo valor es 0000BEB0, 0000CAFE y 0000DED0, respectivamente (en hexadecimal). Realizar el diagrama temporal multiciclo teniendo en cuenta que el programa comienza su ejecución en la línea etiquetada con inicio y termina cuando se completa la última instrucción en la posición etiquetada con sigue. Responde a las siguientes preguntas.

- Indica cómo se han solucionado los riesgos por dependencias de datos:
 - Se ha habilitado un cortocircuito EX-EX entre #1 y #2: ✓

WUOLAH

- Se ha habilitado un cortocircuito EX-EX entre #1 y #3: ✓
 - Se ha habilitado un cortocircuito M-EX entre #1 y #3: ✓
 - Se ha habilitado un cortocircuito EX-EX entre #2 y #3: ✓
 - Se ha habilitado un cortocircuito M-EX entre #2 y #3: ✓
 - Se ha habilitado un cortocircuito EX-EX entre #5 y #7: ✓
 - Se ha habilitado un cortocircuito M-EX entre #5 y #7: ✓
 - Se ha habilitado un cortocircuito EX-EX entre #6 y #7: ✓
 - Se ha habilitado un cortocircuito M-EX entre #6 y #7: ✓
- ¿Cuántas burbujas se introducen por los riesgos por dependencias de datos? ✓
 - ¿Cuántos ciclos se pierden debido a los riesgos por dependencias de control? ✓
 - Calcula el CPI (deja indicada la forma en que lo calculas): $CPI = \text{numerador} / \text{denominador}$
 - numerador = ✓
 - denominador = ✓

Pregunta 8

Correcta

Puntúa 4,00 sobre 4,00

Marcar pregunta

Considerar la estructura segmentada en cinco etapas del procesador MIPS, con hardware para la detección de riesgos por dependencias de datos y **todos los caminos de anticipación** (cortocircuitos) habilitados. Respecto a las dependencias de control, el controlador implementa la *suposición de salto no realizado*. Todos los **saltos** se resuelven por completo en la etapa de **decodificación**. El banco de registros permite la lectura y escritura simultánea de un mismo registro sin conflicto. Sea el siguiente programa, donde se han numerado las instrucciones para facilitar referenciarlas:

```
#1      ori $2, $0, 1000h
#2 loop: lw $1, 2800h($2)
#3      sub $4, $1, $0
#4      jal rotar
#5      sw $7, 7800h($2)
#6      sw $1, C800h($2)
#7      subi $2, $2, 4
#8      bne $2, $0, loop
#9      ...
#20 rotar: add $10, $4, $4
#21      muli $7, $10, 2
#22      jr $31
```

Realiza el diagrama temporal multiciclo desde que el programa comienza su ejecución en la instrucción #1 hasta que se termina de ejecutar completamente por segunda vez la instrucción #2. Responde a las siguientes preguntas.

a) Señala cuáles de las siguientes dependencias verdaderas se producen entre las instrucciones del código propuesto:

- dependencia verdadera de #2 con #1 debido a \$2: ✓
- dependencia verdadera de #5 con #1 debido a \$2: ✓
- dependencia verdadera de #6 con #2 debido a \$1: ✓
- dependencia verdadera de #3 con #2 debido a \$1: ✓
- dependencia verdadera de #7 con #1 debido a \$2: ✓
- dependencia verdadera de #8 con #7 debido a \$2: ✓
- dependencia verdadera de #20 con #3 debido a \$4: ✓
- dependencia verdadera de #21 con #20 debido a \$10: ✓
- dependencia verdadera de #5 con #21 debido a \$7: ✓

b) Tras la realización del diagrama multiciclo (recuerda que se termina de ejecutar completamente por segunda vez la instrucción #2):

- Indica cómo se han solucionado los riesgos por dependencias de datos:
 - Se ha habilitado un cortocircuito EX-EX entre #1 y #2: ✓
 - Se ha habilitado un cortocircuito EX-EX entre #2 y #3: ✓
 - Se ha habilitado un cortocircuito M-EX entre #2 y #3: ✓
 - Se ha habilitado un cortocircuito EX-EX entre #3 y #20: ✓
 - Se ha habilitado un cortocircuito M-EX entre #3 y #20: ✓
 - Se ha habilitado un cortocircuito EX-EX entre #20 y #21: ✓
 - Se ha habilitado un cortocircuito M-EX entre #20 y #21: ✓
 - Se ha habilitado un cortocircuito EX-EX entre #21 y #5: ✓
 - Se ha habilitado un cortocircuito M-M entre #21 y #5: ✓
 - Se ha habilitado un cortocircuito EX-EX entre #7 y #8: ✓
 - Se ha habilitado un cortocircuito EX-ID entre #7 y #8: ✓

- ¿Cuántas burbujas se introducen por los riesgos por dependencias de datos? ✓



LOS JUEGOS DEL CUATRI

- ¿Cuántos ciclos se pierden debido a los riesgos por dependencias de control hasta que comienza a ejecutarse por segunda vez la instrucción #2?
- Calcula el CPI (deja indicada la forma en que lo calculas): $CPI = \text{numerador} / \text{denominador}$
 - numerador =
 - denominador =

Pregunta 9

Correcta

Puntúa 4,00 sobre 4,00

Marcar pregunta

Queremos mejorar el rendimiento del procesador MIPS visto en clase, reduciendo el impacto de los riesgos de control. Para ello introducimos un comparador en la etapa ID que se encarga de realizar la comparación de los valores de los registros que se lean en el banco de registros, para el caso de instrucciones de salto condicional. Además, en esa misma etapa añadimos otra ALU que se encarga de calcular la dirección efectiva de salto, en caso de que hubiera que saltar. Por lo tanto en esta arquitectura los saltos se resuelven completamente al final de la etapa ID.

a) Con esta estrategia, ¿se eliminan totalmente los riesgos de control? Razona la respuesta.

No, porque seguimos sin conocer la dirección correcta del siguiente fetching hasta el final de la etapa de ID de la instrucción de salto

b) En esta nueva arquitectura hemos habilitado todos los caminos de anticipación de la arquitectura MIPS vista en clase: EXE-EXE, MEM-MEM, MEM-EXE. ¿Existe alguna situación en la que sea conveniente introducir un nuevo camino de anticipación? Pon un ejemplo.

Si, los nuevos caminos de anticipación EX-ID y MEM-ID cuando un salto condicional depende de una instrucción previa a distancia 1 o 2

c) Para evaluar el rendimiento de esta nueva arquitectura utilizamos un benchmark de prueba en el que el 15% de las instrucciones son saltos condicionales, de los cuales el 80% son saltos realizados. El 5% de las instrucciones son saltos incondicionales y llamadas a procedimientos. El 20% son instrucciones tipo load el 10% instrucciones tipo store y el resto instrucciones entre registros. Ignorando los riesgos por dependencias de datos, calcula la aceleración que se consigue con este benchmark, comparando la nueva arquitectura con la arquitectura del MIPS vista en clase, cuando en ambas se aplica *predicción de salto tomado*.

El CPI del nuevo procesador es: (redondea a dos decimales)

El CPI del procesador visto en clase es:

Con lo que la aceleración conseguida con el nuevo procesador es: $A = \frac{1,09}{1,03} = 1,06$

Pregunta 10

Correcta

Puntúa 6,00 sobre 6,00

Marcar pregunta

Los diseñadores del MIPS han decidido incorporar un nuevo modo de direccionamiento para especificar los operandos fuente de las instrucciones aritmético-lógicas. En concreto, se van a permitir instrucciones aritmético-lógicas en las que uno de los operandos fuente puede especificar una posición de memoria. Por ejemplo, es válida la instrucción:

add \$2, \$3, 5Ch(\$4)

donde comprobamos que el operando destino es el registro 2, el primer operando fuente es el registro 3 y el segundo operando fuente es el dato que está en la posición de memoria dada por \$4+5Ch, es decir la instrucción anterior realiza la operación $\$2 \leftarrow \$3 + \text{MEM}[\$4 + 5\text{Ch}]$.

Sin embargo, introducir nuevos modos de direccionamiento supone cambiar la arquitectura del procesador MIPS segmentado, así que sus diseñadores deciden modificarlo para que ahora incorpore 6 etapas, tal como se indica en la siguiente figura:

Instrucción	1	2	3	4	5	6	7
I1 - add \$2, \$3, 5Ch(\$4)	IF	ID	A1	M	A2	WB	
I2 - or \$6, \$0, \$0		IF	ID	A1	M	A2	WB

donde cada etapa tiene el siguiente significado:

- IF representa la etapa de búsqueda de instrucción.
- ID representa la etapa de decodificación y en ella se accede al banco de registros.
- A1 representa una nueva etapa con un sumador que se encarga de calcular una dirección efectiva de memoria, si la instrucción lo requiere. Esto ocurrirá en las instrucciones lw y sw, así como en las instrucciones aritmético lógicas en las que uno de los operandos fuentes es un dato que está en memoria (tal y como ocurre en la instrucción add -I1- en la figura anterior). En caso en que la instrucción no necesite acceder a memoria, esta etapa no tiene ningún efecto, aunque se consume el ciclo de reloj (tal y como ocurre en la instrucción I2 de esa misma figura);
- M representa la etapa de acceso a memoria.
- A2 representa la etapa de ejecución, donde se ubica la ALU del procesador.
- WB es la etapa de post-escritura, en la que se accede al banco de registros.

NOTA: Las etapas IF, ID, M, A2 y WB son idénticas a las etapas IF, ID, MEM, EX y WB del procesador MIPS visto en clase. Asume que los saltos se resuelven en A2.

a) Si suponemos anticipación en el banco de registros, ¿qué cortocircuitos propondrías para reducir los riesgos por dependencias de datos?

- cortocircuito A1-ID: No
- cortocircuito A1-A1: No
- cortocircuito A1-M: No
- cortocircuito A1-A2: No
- cortocircuito M-ID: No
- cortocircuito M-A1: Si
- cortocircuito M-M: Si
- cortocircuito M-A2: No
- cortocircuito A2-ID: No
- cortocircuito A2-A1: Si
- cortocircuito A2-M: Si
- cortocircuito A2-A2: Si

b) Suponiendo que el siguiente código se ejecuta en el nuevo procesador MIPS de 6 etapas con anticipación en el banco de registros y con el HW necesario para implementar todos los caminos de anticipación posibles

representa en una tabla la evolución del cauce para la ejecución completa del código y contesta a las siguientes preguntas:

es el momento de presentarte como tributo



```
#1      addi $9, $0, $0
#2      add $10, $0, 58($9)
#3      muli $11, $10, 4
#4      lw $3, 0($11)
#5      sw $3, 0($10)
#6      sw $3, 4($10)
#7      lw $4, 4($11)
#8      sub $9, $9, 0($4)
```

- Indica cómo se han solucionado los riesgos de datos:

- Se ha habilitado un cortocircuito A1-A1 entre #1 y #2: ✓
- Se ha habilitado un cortocircuito A2-A1 entre #1 y #2: ✓
- Se ha habilitado un cortocircuito A2-A2 entre #1 y #2: ✓
- Se ha habilitado un cortocircuito A1-A1 entre #2 y #3: ✓
- Se ha habilitado un cortocircuito A2-A1 entre #2 y #3: ✓
- Se ha habilitado un cortocircuito A2-A2 entre #2 y #3: ✓
- Se ha habilitado un cortocircuito A2-A1 entre #2 y #5: ✓
- Se ha habilitado un cortocircuito A2-A1 entre #2 y #6: ✓
- Se ha habilitado un cortocircuito A1-A1 entre #3 y #4: ✓
- Se ha habilitado un cortocircuito A2-A1 entre #3 y #4: ✓
- Se ha habilitado un cortocircuito A2-A2 entre #3 y #4: ✓
- Se ha habilitado un cortocircuito A2-A1 entre #3 y #7: ✓
- Se ha habilitado un cortocircuito M-A1 entre #4 y #5: ✓
- Se ha habilitado un cortocircuito M-M entre #4 y #5: ✓
- Se ha habilitado un cortocircuito M-A2 entre #4 y #5: ✓
- Se ha habilitado un cortocircuito A2-M entre #4 y #5: ✓
- Se ha habilitado un cortocircuito M-A1 entre #4 y #6: ✓
- Se ha habilitado un cortocircuito M-M entre #4 y #6: ✓
- Se ha habilitado un cortocircuito M-A2 entre #4 y #6: ✓
- Se ha habilitado un cortocircuito A2-M entre #4 y #6: ✓
- Se ha habilitado un cortocircuito M-A1 entre #7 y #8: ✓
- Se ha habilitado un cortocircuito M-M entre #7 y #8: ✓
- Se ha habilitado un cortocircuito M-A2 entre #7 y #8: ✓

- ¿Cuántas burbujas se introducen por los riesgos por dependencias de datos? ✓
- ¿Cuántas burbujas se introducen por los riesgos por dependencias de control? ✓
- Calcula el CPI (deja indicada la forma en que lo calculas): $CPI = \text{numerador} / \text{denominador}$
 - numerador = ✓
 - denominador = ✓

Pregunta 11

Correcta

Puntúa 4,00 sobre 4,00

Marcar pregunta

Los diseñadores del procesador MIPS deciden intentar mejorar el rendimiento de la arquitectura. En concreto, se centran en el problema de los riesgos de control. Para ello proponen realizar la carga del PC con la dirección de la instrucción a la que se salta en la misma etapa en la que se sabe si se salta o no. Por ejemplo, las instrucciones de salto incondicional saben que se salta al final de la etapa ID (decodificación), mientras que las instrucciones de salto condicional pueden saber si se salta o no al final de la etapa EX (ejecución). Para evaluar el rendimiento de la nueva máquina, deciden utilizar un conjunto de benchmarks de prueba. De estos benchmarks conocen la siguiente información:

- El 20% son instrucciones del tipo lw/sw
- El 40% son instrucciones aritmético-lógicas
- El 25% son instrucciones de salto condicional, de las que el 80% son saltos que se realizan.
- El 5% son instrucciones de salto incondicional
- El resto son instrucciones de llamadas a procedimientos

Responder a las siguientes cuestiones:

- a) Con este rediseño en la arquitectura, ¿se eliminan los riesgos de control?

No, porque si falla la predicción entran instrucciones incorrectas ✓

- b) Calcula el rendimiento de la nueva arquitectura MIPS respecto de la arquitectura clásica del MIPS visto en clase, en la que los saltos terminaban en la etapa de memoria, suponiendo que en ambos casos se aplica una **estrategia de predicción de salto no realizado**.

El CPI del nuevo procesador es: ✓ (redondea a dos decimales)

El CPI del procesador visto en clase es: ✓

Con lo que la aceleración conseguida con el nuevo procesador es: $S =$ ✓

- c) Calcula el rendimiento de la nueva arquitectura MIPS respecto de la arquitectura clásica del MIPS visto en clase, suponiendo ahora que en ambos casos se aplica una **estrategia de predicción de salto realizado**. Asumir en este caso que se dispone de una BTB en la etapa IF.

El CPI del nuevo procesador es: ✓ (redondea a dos decimales)

WUOLAH

El CPI del procesador visto en clase es:

Con lo que la aceleración conseguida con el nuevo procesador es: $S =$

d) En vista de los resultados anteriores, ¿qué estrategia de predicción es la más efectiva? ¿Por qué?

Pregunta 12

Correcta

Puntúa 6,00 sobre 6,00

Marcar pregunta

Los diseñadores del procesador MIPS se han planteado reducir el coste del sistema. Para ello deciden utilizar una memoria de datos más barata. El problema es que la latencia de esta memoria es de 2 ciclos de reloj. Para incorporar esta nueva memoria en el diseño segmentado, dividen la etapa de memoria original del MIPS (M) en dos etapas: M1 y M2. Por lo tanto, el camino de datos del nuevo procesador segmentado hace que las instrucciones se ejecuten según se muestra a continuación:

Instrucción	1	2	3	4	5	6	7
I1		IF	ID	EX	M1	M2	WB
I2			IF	ID	EX	M1	M2

En este nuevo diseño las instrucciones lw y sw completan su acceso a memoria al finalizar la etapa M2. Se ha decidido además activar todos los caminos de anticipación posibles, y que exista anticipación en el banco de registros (es decir, en un mismo ciclo puede escribirse y leerse un mismo registro). **En cuanto a los saltos, se resuelven por completo durante la etapa de decodificación y el controlador implementa la suposición de salto no realizado.** Bajo estas condiciones, responde a las siguientes cuestiones:

a) ¿Entre qué etapas pueden activarse cortocircuitos?

- cortocircuito EX-ID:
- cortocircuito EX-EX:
- cortocircuito EX-M1:
- cortocircuito EX-M2:
- cortocircuito M1-ID:
- cortocircuito M1-EX:
- cortocircuito M1-M1:
- cortocircuito M1-M2:
- cortocircuito M2-ID:
- cortocircuito M2-EX:
- cortocircuito M2-M1:
- cortocircuito M2-M2:

b) Suponiendo que la unidad de control aplica todos los cortocircuitos posibles, así como la estrategia de predicción de salto no realizado, representa en la tabla la evolución del cauce para la ejecución completa del siguiente trozo de código:

#1	addi \$2, \$0, 16
#2	buc: lw \$3, 0(\$2)
#3	lw \$4, 4(\$2)
#4	add \$5, \$3, \$4
#5	sw \$5, 100(\$2)
#6	subi \$2, \$2, 8
#7	bne \$2, \$0, buc
#8	add \$6, \$3, \$4

- Indica cómo se han solucionado los riesgos de datos:
 - Se ha habilitado un cortocircuito EX-EX entre #1 y #2:
 - Se ha habilitado un cortocircuito EX-EX entre #1 y #3:
 - Se ha habilitado un cortocircuito M1-EX entre #1 y #3:
 - Se ha habilitado un cortocircuito M2-M1 entre #1 y #3:
 - Se ha habilitado un cortocircuito EX-EX entre #2 y #4:
 - Se ha habilitado un cortocircuito M2-EX entre #2 y #4:
 - Se ha habilitado un cortocircuito EX-EX entre #3 y #4:
 - Se ha habilitado un cortocircuito M2-EX entre #3 y #4:
 - Se ha habilitado un cortocircuito EX-EX entre #4 y #5:
 - Se ha habilitado un cortocircuito M1-M1 entre #4 y #5:
 - Se ha habilitado un cortocircuito M2-M2 entre #4 y #5:
 - Se ha habilitado un cortocircuito EX-EX entre #6 y #7:
 - Se ha habilitado un cortocircuito EX-ID entre #6 y #7:

• ¿Cuántas burbujas se introducen por los riesgos por dependencias de datos?

• ¿Cuántos ciclos se pierden debido a los riesgos por dependencias de control?

• Calcula el CPI (deja indicada la forma en que lo calculas): $CPI = \text{numerador} / \text{denominador}$

- numerador =
- denominador =

Pregunta 13

Correcta

Puntúa 6,00 sobre 6,00

Marcar pregunta

Los diseñadores del procesador MIPS se centran ahora en el problema de los riesgos de control. Se han percatado de que se pierde un alto número de ciclos cuando se implementa una estrategia de predicción estática, tanto salto tomado como salto no tomado. Por eso deciden implementar una estrategia de **predicción dinámica**. Para ello, se decide incorporar en el controlador una tabla llamada BTB. La BTB sólo contiene información sobre las instrucciones de salto que se han ejecutado hasta ese momento. Esta tabla es una memoria asociativa con 3 campos: el primer campo identifica el PC de una instrucción de salto (PCInst), el segundo campo identifica la dirección destino de salto de esa instrucción (DirDest) y el tercer campo es un bit de predicción (Pred). Si este bit vale 1 es que la última vez que se ejecutó esa instrucción de salto, se saltó. Si vale 0 es porque la última vez que se ejecutó no se saltó. Se le llama bit de predicción porque depende del valor que tenga, se saltará o no al volver a ejecutarse esa instrucción.

El funcionamiento de la BTB es el siguiente:

- Durante la etapa de búsqueda de instrucción (IF)) el controlador consulta la BTB.
 - Si no hay ninguna entrada en la BTB cuyo campo PCInst coincida con el PC de la instrucción que se está buscando actualmente, es porque no se trata de una instrucción de salto, o bien es un salto que todavía no se ha ejecutado. En este caso, en el siguiente ciclo de reloj, entrará en el cauce la siguiente instrucción del programa.
 - Si, por el contrario, hay alguna entrada en la BTB cuyo campo PCInst coincida con el PC actual, es porque se trata de una instrucción de salto que se ha ejecutado antes. En este caso se chequeará el bit de predicción, Pred.
 - Si Pred=0, el controlador asumirá que hay que aplicar la estrategia de *salto no tomado*, y por lo tanto en el siguiente ciclo de reloj, entrará en el cauce la siguiente instrucción del código.
 - Si Pred=1 el controlador asumirá *salto tomado*, y en el siguiente ciclo de reloj, entrará en el cauce la instrucción destino de salto (se sabe cuál es porque en la tabla está guardada la dirección destino DirDest).
 - La instrucción continuará ejecutándose por las etapas ID, EX, M y WB. Si se trata de una instrucción de salto, en la etapa de EX se habrá calculado la condición de salto (es decir, si hay que saltar o no) y se podrá comprobar si la predicción ha sido correcta o no:
 - **Si la predicción ha sido correcta**, es decir, la predicción fue salto tomado (o salto no tomado) y la condición de salto indica que hay que saltar (o no saltar) entonces la predicción ha acertado y no hay que hacer nada más.
 - Por el contrario, **si la predicción falló**, es decir, se ejecutó salto tomado (o salto no tomado) y la condición de salto indica que no hay que saltar (o hay que saltar), en ese caso hay que actualizar el bit de predicción, Pred para que indique lo que ha ocurrido en esta ejecución de la instrucción de salto. Además, habrá que anular las instrucciones que han entrado en el cauce después de la instrucción de salto y cargar el PC con la dirección correcta. En el siguiente ciclo de reloj, por tanto, recomenzará la ejecución con la instrucción correcta que debe ejecutarse después de este salto.
 - Otra posibilidad es que sea la primera vez que se ejecuta esa instrucción de salto. Por tanto, la información de esta instrucción no estará en la BTB. En este caso, en la etapa de M habrá que actualizar la BTB con la información del PC de esta instrucción (PCInst), la dirección destino de salto (DirDest) y el bit de Predicción (Pred).

Suponiendo que se trabaja con el MIPS visto en clase, para el que **se activan todos los caminos de anticipación posibles**, y que en el banco de registros pueda leerse en un mismo ciclo el registro que va a ser escrito en dicho ciclo (anticipación en el banco de registros), responde a las siguientes cuestiones:

a) Representa en un diagrama multiciclo la evolución del cauce hasta que se ejecute por tercera vez la instrucción `lw $3, 0($2)` para el código que se muestra a continuación.

```
#1      addi $2,$0, 1000
#2  bucle: lw $3, 0($2)
#3      add $4, $3, $3
#4      sw $4, 1000($2)
#5      subi $2, $2, 8
#6      bne $2, $0, bucle
```

- Marca cuáles de las siguientes dependencias verdaderas se producen entre las instrucciones del código propuesto:

- dependencia verdadera de #2 con #1 debido a \$3: ☐ No ☒ Si
- dependencia verdadera de #2 con #1 debido a \$2: ☐ Si ☒ No
- dependencia verdadera de #3 con #2 debido a \$4: ☐ No ☒ Si
- dependencia verdadera de #4 con #3 debido a \$4: ☐ Si ☒ No
- dependencia verdadera de #4 con #3 debido a \$2: ☐ No ☒ Si
- dependencia verdadera de #2 con #5 debido a \$2: ☐ Si ☒ No
- dependencia verdadera de #5 con #2 debido a \$2: ☐ No ☒ Si
- dependencia verdadera de #6 con #5 debido a \$2: ☐ Si ☒ No

- En cuanto a cómo se han solucionado los riesgos de datos:

- Se ha habilitado un cortocircuito EX-EX entre #1 e #2: ☐ Si ☒ No
- Se ha habilitado un cortocircuito M-M entre #2 e #3: ☐ No ☒ Si
- Se ha habilitado un cortocircuito M-EX entre #2 e #3: ☐ Si ☒ No
- Se ha habilitado un cortocircuito EX-EX entre #3 e #4: ☐ No ☒ Si
- Se ha habilitado un cortocircuito M-EX entre #3 e #4: ☐ No ☒ Si
- Se ha habilitado un cortocircuito M-M entre #3 e #4: ☐ Si ☒ No
- Se ha habilitado un cortocircuito EX-EX entre #5 e #6: ☐ Si ☒ No
- Se ha habilitado un cortocircuito EX-ID entre #5 e #6: ☐ No ☒ Si

- ¿Ha sido necesario detener el cauce por dependencias de datos? ¿Cuántas burbujas se introducen en total?
- ¿Cuántos ciclos se pierden por fallos en la predicción del salto?

b) Sabiendo que la instrucción `bne $2,$0,bucle` se encuentra en la dirección 000A5624h, rellena la siguiente tabla indicando, para cada iteración del código anterior, el valor que los campos DirDest y Pred tendrán en la BTB cuando se comienza a ejecutar esa instrucción en la iteración correspondiente. Fijate que en la primera iteración aún no se ha ejecutado esa instrucción, por lo tanto en la BTB no hay información sobre esa instrucción.

	PCInst	DirDest (8 dígitos hexadecimales, mayúsculas)	Pred
1ª iteración	---	---	---
2ª iteración	000A5624h	<input type="text" value="000A5614"/> h <input checked="" type="checkbox"/>	<input type="text" value="1"/> <input checked="" type="checkbox"/>
3ª iteración	000A5624h	<input type="text" value="000A5614"/> h <input checked="" type="checkbox"/>	<input type="text" value="1"/> <input checked="" type="checkbox"/>

(a nosotros por suerte nos pasa)

No si antes decirte
Lo mucho que te voy a recordar

Pero me voy a graduar.
Mañana mi diploma y título he de pagar

Llegó mi momento de despedirte
Tras años en los que has estado mi
lado.

Siempre me has ayudado
Cuando por exámenes me he
agobiado

Oh Wuolah wuolilah
Tu que eres tan bonita

Última iteración	000A5624h	000A5614 ✓h	1 ▼ ✓
------------------	-----------	-------------	-------

c) Calcular el CPI de la ejecución completa del código:

El número de iteraciones es N= 125

CPI= $\frac{\text{numerador}}{\text{denominador}}$

numerador = 758

denominador = 626

Pregunta 14

Correcta

Puntúa 8,00 sobre 8,00

 Marcar pregunta

Sea la versión del procesador MIPS vista en clase, en la que la **terminación de los saltos es en la etapa de memoria**, evaluándose la **condición de salto en ejecución**, y que para la resolución de las dependencias de control implementa la suposición de *salto no realizado*. Además, para la resolución de los riesgos de datos, dispone de hardware para la implementación de las estrategias de detención y **anticipación**, y adicionalmente implementa anticipación en el banco de registros. Para el siguiente fragmento de código:

```
#1      ori $4, $0, 10
#2 loop: add $7, $4, $1
#3      beq $7, $4, fin
#4      lw $1, 100($7)
#5      j loop
#6 fin:  sw $1, 200($4)
#7      sub $5, $1, $4
```

a) Completa un diagrama multiciclo donde se muestre la evolución del código a través del cauce segmentado, suponiendo que en el salto condicional la primera vez no salta y la segunda sí:

- Marca cuáles de las siguientes dependencias verdaderas se producen entre las instrucciones del código propuesto:

- | | | |
|--|----|---|
| ◦ dependencia verdadera de #7 con #6 debido a \$1: | No | ▼ |
| ◦ dependencia verdadera de #6 con #4 debido a \$1: | Si | ▼ |
| ◦ dependencia verdadera de #2 con #1 debido a \$4: | Si | ▼ |
| ◦ dependencia verdadera de #4 con #2 debido a \$7: | Si | ▼ |
| ◦ dependencia verdadera de #4 con #3 debido a \$7: | No | ▼ |
| ◦ dependencia verdadera de #2 con #4 debido a \$1: | Si | ▼ |
| ◦ dependencia verdadera de #3 con #2 debido a \$7: | Si | ▼ |
| ◦ dependencia verdadera de #6 con #3 debido a \$4: | No | ▼ |
| ◦ dependencia verdadera de #7 con #6 debido a \$4: | No | ▼ |
| ◦ dependencia verdadera de #3 con #1 debido a \$4: | Si | ▼ |

- En cuanto a cómo se han solucionado los riesgos de datos:

- Se ha habilitado un cortocircuito EX-EX entre #1 e #2: ✓
- La anticipación en el banco de registros soluciona el riesgo entre #1 e #3: ✓
- Se ha habilitado un cortocircuito EX-EX entre #1 e #3: ✓
- Se ha habilitado un cortocircuito M-EX entre #1 e #3: ✓
- Se ha habilitado un cortocircuito M-M entre #1 e #3: ✓
- Se ha habilitado un cortocircuito EX-EX entre #2 e #3: ✓
- Se ha habilitado un cortocircuito M-EX entre #2 e #3: ✓
- Se ha habilitado un cortocircuito M-EX entre #2 e #4: ✓

- ¿Ha sido necesario detener el cauce por riesgos por dependencias de datos? ¿Cuántas burbujas se introducen en total? 0

b) Calcula el CPI suponiendo que se realizan 100 iteraciones, es decir, las 99 primeras veces la instrucción beq no salta, y en la última iteración sí (deja indicada la forma en que lo calculas).

$$\text{CPI} = \text{numerador} / \text{denominador}$$

numerador = 705 ✓

denominador = 401

Se decide modificar la arquitectura del procesador para anticipar los **saltos** en HW, de forma que **se terminen en la etapa de decodificación**.

e) Vuelve a hacer el apartado a) para esta segunda implementación del MIPS.

- En cuanto a cómo se han solucionado los riesgos de datos:
 - Se ha habilitado un cortocircuito M-EX entre #2 e #4:
 - La anticipación en el banco de registros soluciona el riesgo entre #1 e #3:
 - Se ha habilitado un cortocircuito M-EX entre #1 e #3:
 - Se ha habilitado un cortocircuito EX-EX entre #1 e #3:
 - Se ha habilitado un cortocircuito M-EX entre #2 e #3:
 - Se ha habilitado un cortocircuito EX-EX entre #2 e #3:
 - Se ha habilitado un cortocircuito EX-ID entre #2 e #3:
 - Se ha habilitado un cortocircuito EX-EX entre #2 e #4:

- ¿Ha sido necesario detener el cauce por riesgos por dependencias de datos? ¿Cuántas burbujas se introducen en total? ✓

Calcula el CPI resultante (de nuevo para 100 iteraciones):

CPI= numerador / denominador

numerador = ✓

denominador = ✓

Pregunta 15

Correcta

Puntúa 4,00 sobre 4,00

✓ Marcar pregunta

Considerar la estructura segmentada en cinco etapas del procesador MIPS, con hardware para la detección de riesgos por dependencias de datos y **todos los caminos de anticipación (cortocircuitos) habilitados**. Respecto a las dependencias de control, el controlador implementa la suposición de *salto no realizado*. La **condición de salto y el cálculo de la dirección de salto se computan en la etapa de ejecución**, por lo que los saltos se resuelven por completo en dicha etapa. El banco de registros permite la lectura y escritura simultánea de un mismo registro sin conflicto. Sea el siguiente programa, donde se han numerado las instrucciones para facilitar referenciarlas:

```
#1      addi $9, $0, NUM
#2      sub $10, $10, $10
#3      add $11, $0, $0
#4 SA:  j SB
#5      lw $4, 8000($10)
#6 SB:  lw $3, 400($10)
#7      lw $4, 560($10)
#8      sub $3, $3, $4
#9      sw $3, 1200($10)
#10     add $11, $11, $3
#11     sw $11, 1600($10)
#12     addi $10, $10, 4
#13     bne $9, $10, SB
#14     sw $11, 400($10)
```

a) Suponiendo que la constante NUM tiene un valor de 8, realiza el diagrama temporal multiciclo hasta que termina de ejecutarse la instrucción #14 y responde a las siguientes preguntas:

- ¿Cuántas burbujas ha sido necesario introducir en total para resolver riesgos por dependencias de datos? ✓
- Para ejecutar correctamente la instrucción #8 ha sido necesario ▼
- Para ejecutar correctamente la instrucción #10 ha sido necesario ▼ ✓
- Para ejecutar correctamente la instrucción #11 ha sido necesario ▼ ✓
- La instrucción #5 se ha cancelado ✓ veces

b) Calcula el CPI:

- ¿Cuántas instrucciones han terminado de ejecutarse? ✓
- ¿Cuántos ciclos han transcurrido en total? ✓
- CPI: ✓

Pregunta 16

Correcta

Puntúa 4,00 sobre 4,00

✓ Marcar pregunta

Considerar la estructura segmentada en cinco etapas del procesador MIPS, con hardware para la detección de riesgos por dependencias de datos y los siguientes caminos de anticipación habilitados: Ejecución-Ejecución, Memoria-Ejecución, y Memoria-Memoria. Respecto a las dependencias de control, el controlador implementa la suposición de salto no realizado. El cálculo de la dirección de salto se hace en la etapa de decodificación y la evaluación de la condición de salto en la etapa de ejecución, por lo que **los saltos incondicionales se resuelven por completo en la etapa de decodificación y los saltos condicionales en la etapa de ejecución**. El banco de registros permite la lectura y escritura simultánea de un mismo registro sin conflicto.

a) Para esta arquitectura, ¿son suficientes los cortocircuitos propuestos? ▼ ✓

b) Sea el siguiente programa, donde se han numerado las instrucciones para facilitar referenciarlas:

```
#1 main: addi $1, $0, 0
#2      addi $2, $0, 100
#3 loop: lw $3, 0($2)
#4      beq $1, $3, fin
#5      add $1, $1, $3
#6      addi $2, $2, 4
#7      j loop
#8 fin:  sw $1, 100($2)
```

Se sabe que la instrucción #7 se ejecuta sólo la primera vez. Realiza el diagrama temporal multiciclo hasta que termina de ejecutarse la instrucción #8 y responde a las siguientes preguntas.

a) Indica los cortocircuitos y burbujas que has necesitado:

- ¿Cuántas burbujas ha sido necesario introducir en total para resolver los riesgos por dependencias de datos? ✓
- Para ejecutar correctamente la instrucción #4 ha sido necesario ▼ ✓
- El resultado producido por la instrucción #3 es anticipado a ✓ instrucción/es mediante un cortocircuito y a ✓ instrucción/es gracias a la anticipación en el banco de registros.

WUOLAH

- La instrucción #5 se ha cancelado ✓ veces

c) Calcula el CPI de la ejecución completa:

- ¿Cuántas instrucciones han terminado de ejecutarse? ✓
- ¿Cuántos ciclos han transcurrido en total? ✓
- CPI: ✓

Finalizar revisión

uni>ersia



Universidad de Málaga · Avda. Cervantes, 2. 29071 MÁLAGA · Tel. 952131000 · info@uma.es

© Todos los derechos reservados

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

si lees esto me debes un besito