

Problemas-Evaluacion-Tema-5-Segm...



irenecasrod



Arquitecturas Avanzadas de Procesadores



4º Grado en Ingeniería Informática



Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Máster

Online en Ciberseguridad

Nº1 en España según El Mundo



Hasta el 46%
de beca



Mejor Máster
según el
Ranking de
ELMUNDO

Para ser el mejor hay que aprender
de los mejores.

IMEF

Smart Education

Deloitte

Infórmate

Consigue Empleo o Prácticas

Matricúlate en IMF y accede sin coste a nuestro servicio de Desarrollo Profesional con más de 7.000 ofertas de empleo y prácticas al mes.



PROBLEMAS TEMA 5. SEGMENTACIÓN Y RIESGOS (RELACIÓN PARA SU EVALUACIÓN)

- 1) Las siguientes secuencias de instrucciones hacen uso del camino de datos segmentado del procesador MIPS:

```
lw    $1, 40($2)
add   $2, $3, $3
add   $1, $1, $2
sw    $1, 20($2)
```

Secuencia 1

```
add   $1, $2, $3
sw    $2, 0($1)
lw    $1, 4($2)
add   $2, $2, $1
```

Secuencia 2

- a) Detectar los posibles riesgos que se producen en las dos secuencias.

lw (\$1), 40(\$2)	add (\$1), \$2, \$3
add (\$2), \$3, \$3	sw \$2, 0(\$1)
add (\$1), (\$1), (\$2)	lw (\$1), 4(\$2)
sw (\$1), 20(\$2)	add \$2, \$2, (\$1)
Secuencia 1	Secuencia 2

- b) Considerando que no hay anticipación ni detección de riesgos, insertar el menor número de *nop* para asegurar una ejecución correcta para cada una de las secuencias.

Secuencia 1

Postescritura y lectura no simultánea	Postescritura y lectura simultánea
lw \$1, 40(\$2)	lw \$1, 40(\$2)
add \$2, \$3, \$3	add \$2, \$3, \$3
NOP	NOP
NOP	NOP
NOP	add \$1, \$1, \$2
add \$1, \$1, \$2	NOP
NOP	NOP
NOP	sw \$1, 20(\$2)
NOP	
sw \$1, 20(\$2)	



Secuencia 2

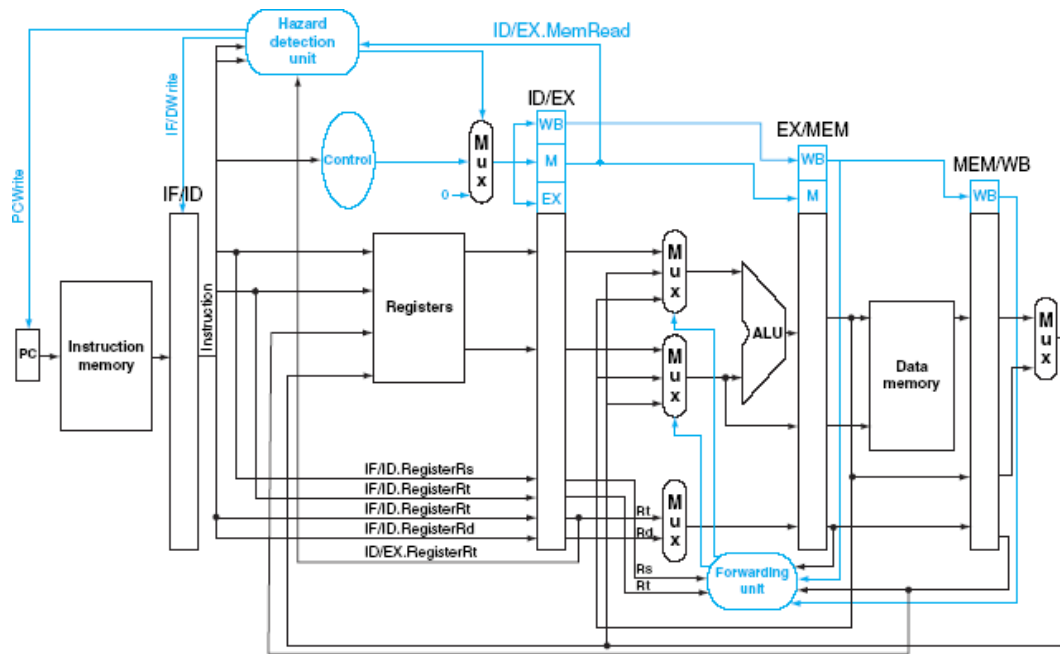
Postescritura y lectura no simultánea	Postescritura y lectura simultánea
add \$1, \$2, \$3	add \$1, \$2, \$3
NOP	NOP
NOP	NOP
NOP	sw \$2, 0(\$1)
sw \$2, 0(\$1)	lw \$1, 4(\$2)
lw \$1, 4(\$2)	NOP
NOP	NOP
NOP	add \$2, \$2, \$1
NOP	
add \$2, \$2, \$1	

- c) Se permite reorganización de código, pudiendo usar el registro \$7 como registro temporal, insertar el menor número de *nop* para asegurar una ejecución correcta para cada una de las secuencias.
- d) Teniendo en cuenta las secuencias iniciales de código, si el procesador tiene anticipación, pero se ha olvidado implementar la unidad de control de riesgos, ¿qué ocurre al ejecutar las secuencias 1 y 2 de instrucciones?

En el caso en que se necesite un dato de una instrucción previa y éste se encuentre en la ALU, se podría anticipar antes de que sea escrito en el registro destino. Por eso, en un procesador que solo tiene unidad de anticipación, este riesgo se puede resolver a nivel de hardware sin que el compilador tenga que insertar NOP ni provocar retrasos. Es sólo en estos casos cuando la unidad de control de riesgos no haría falta.

Sin embargo, la anticipación no puede resolver aquellas situaciones en las que se realizará la carga de un registro (lw) y el dato a cargar es un operando de la instrucción siguiente, como ocurre en estas secuencias de código. Por tanto, se necesitaría la *unidad de control de riesgos*, que opere en la etapa ID, de manera que pueda insertar una burbuja entre la carga y la instrucción que usará el dato a cargar.

- e) Indicar teniendo en cuenta la siguiente figura, cómo trabajaría la unidad de detección y anticipación en la secuencia 1, indicando todos los ciclos de la secuencia.



Tiempo (en ciclos de reloj) →

CC 1 CC 2 CC 3 CC 4 CC 5 CC 6 CC 7 CC 8

Orden de
ejecución
del programa
(en instrucciones)

