

3) Analizar el siguiente programa con el simulador MIPS64:

```
.data
A:   .word 1
B:   .word 6

.code

ld r1, A(r0)
ld r2, B(r0)

loop: dsll r1, r1, 1
      daddi r2, r2, -1
      bnez r2, loop
      halt
```

a) Ejecutar el programa con Forwarding habilitado y responder:

- ¿Por qué se presentan atascos tipo RAW?
- Branch Taken es otro tipo de atasco que aparece. ¿Qué significa? ¿Por qué se produce?
- ¿Cuántos CPI tiene la ejecución de este programa? Tomar nota del número de ciclos, cantidad de instrucciones y CPI.

b) Ejecutar ahora el programa deshabilitando el Forwarding y responder:

- ¿Qué instrucciones generan los atascos tipo RAW y por qué? ¿En qué etapa del cauce se produce el atasco en cada caso y durante cuántos ciclos?
- Los Branch Taken Stalls se siguen generando. ¿Qué cantidad de ciclos dura este atasco en cada vuelta del lazo 'loop'?

Comparar con la ejecución con Forwarding y explicar la diferencia.

- ¿Cuántos CPI tiene la ejecución del programa en este caso? Comparar número de ciclos, cantidad de instrucciones y CPI con el caso con Forwarding.

c) Reordenar las instrucciones para que la cantidad de RAW sea '0' en la ejecución del programa (Forwarding habilitado)

d) Modificar el programa para que almacene en un arreglo en memoria de datos los contenidos parciales del registro r1 ¿Qué significado tienen los elementos de la tabla que se genera?

Respuestas:

3)

a)

- Se presentan stalls de tipo RAW porque la instruccion bnez r2, loop necesita r2 pero ese valor no ha sido calculado en daddi r2, r2, -1, por lo tanto debe esperar que la misma ejecute la etapa X para luego tener su operando adelantado con el forwarding.

- Los Branch Taken Stalls son atascos provocados por una instruccion de salto, los mismos se dan cuando se comienza a cumplir con la etapa de Fetch (F) de la siguiente instruccion al salto y resulta que el salto se efectua por lo que se debe dejar de cumplir con la misma y cumplir con la instruccion de destino del salto.

- El programa con forwarding activado tarda 36 ciclos de reloj, cumple 21 instrucciones y el CPI es 1,714.

b)

- Las instrucciones que generan los atascos de tipo RAW son:

* dsll r1, r1, 1: La misma necesita utilizar el valor en r1 el cual todavia no fue cargado. El mismo solo sucede en la primera vez que se cumple con la instruccion. El atasco se produce en D (Decode) ya que en esta etapa es en la que se necesita conocer los valores de los operandos. Se produce un unico atasco de tipo RAW.

* bnez r2, loop: La misma necesita utilizar el valor en r2 el cual todavia no ha sido cargado en la etapa W (Write back) de la instruccion daddi r2, r2, -1. El atasco se produce en D (Decode) ya que en esta etapa es en la que se necesita conocer los valores de los operandos. Se producen 2 atascos de tipo RAW ya que debe esperar a las etapas X (Execute) y M (Memory Access) de daddi r2, r2, -1 (No debe esperar a la etapa W ya que en ese mismo ciclo es posible primero escribir el dato en el registro y luego leerlo en la segunda parte del ciclo).

- Los Branch Taken Stalls se siguen generando pero ahora por cada salto que realicemos tendremos 3 ciclos de parada.

- El programa sin forwarding activado tarda 43 ciclos de reloj, cumple 21 instrucciones y el CPI es 2,048. Exceptuando la cantidad de instrucciones que cumple al ejecutar el programa con forwarding tardaremos menos ciclos de reloj y el CPI sera menor.