

PRÁCTICA 4

Carlos Raso Alonso

Introducción a los ordenadores.

Objetivos

En esta práctica se pretendía que entendamos bien cómo funciona un procesador que use pipeline, más concretamente, cómo funciona el pipeline que utiliza el procesador 5-stage processor incorporado en el simulador ripes. Para ello debemos ver cómo este procesador divide cada instrucción en una serie de etapas (5)

Ejercicios

1) ¿Cuál es el estado de cada una de las 5 etapas del pipeline en los ciclos 6 y 8?

ciclo 6: WB: lw, MEM: addi, EX: add, ID: add, IF: addi

ciclo 8: WB: add, MEM: add, EX: addi, ID: blt, IF: auipc

2) ¿Qué señales de control se activan en el ciclo 4? ¿A qué instrucciones del código corresponden?

La única que se activa es el “write enable” del conjunto de registros. Corresponde a la microinstrucción MEM de la primera instrucción addi, es decir, lo que consigue esta señal de control en este momento es que el resultado de la instrucción addi se escriba en el registro especificado en la instrucción (en este caso el registro a2).

3) ¿Cuáles son los valores de las salidas de los multiplexores señalados en la figura 7?

-8 el que está conectado al registro inmediato y el valor del PC el otro multiplexor. Esto es así ya que está en la fase de ejecución la instrucción bgt a7, zero, loop, la cual, tras realizar la comparación entre 0 y a7 en la decodificación de esta instrucción y haber visto que se tenía que hacer un salto ya que se cumplía la condición pedida, en la fase de ejecución debe alterar el valor de PC para ir a la etiqueta pedida. En este caso la etiqueta está 8 posiciones de memoria detrás de la posición actual de PC y por eso en la ALU entra desde el inmediato el valor de -8 y también entra el valor de PC para que se le sume el valor entrante desde el inmediato.

4) ¿Por qué los valores aparecen en este orden?

Sería indiferente por qué multiplexor entra cada valor sin embargo entran en este orden por que el valor de PC solo puede salir por uno de los dos multiplexores directamente desde el registro PC y el valor del registro del inmediato solo puede salir por uno de los multiplexores también, concretamente, por el que no puede salir el PC.

5) Justifica el estado del pipeline en el ciclo 9

La justificación del estado del pipeline en el ciclo 9 está en las dos preguntas anteriores. Está relacionado con la implementación de la instrucción de branch bgt

6) Cuando no se produce el salto, ¿cuántos ciclos tarda en ejecutarse la instrucción bgt a7, zero, loop?

Tarda 5 ciclos, 1 por cada microinstrucción

7) Cuando se está ejecutando la instrucción de salto pero el salto no se produce, ¿a qué posición apunta la memoria de instrucciones?

Apunta a la instrucción que tendría que apuntar normalmente siguiendo el flujo de ejecución del programa, 0x24. En este caso apunta a la instrucción addi x10, x10, -28 que es la posición que está dos posiciones de memoria delante de la posición de memoria de la instrucción de salto (ya que la instrucción de salto tras haber pasado su etapa de fetch, ha pasado la etapa de decode mientras la instrucción en la posición de memoria inmediata -auipc x10 0x10000- estaba en la etapa de fetch)

8) Cuando se produce el salto, ¿Cuántos ciclos tarda en ejecutarse la instrucción bgt a7, zero, loop?

Tarda los mismos ciclos que tarda cuando no se produce el salto, 5 ciclos, uno por instrucción

9) Cuando se está ejecutando la instrucción de salto y el salto se produce, ¿a qué posición apunta la memoria de instrucciones?

En el momento en que se está ejecutando la instrucción de salto si la condición de salto se satisface se activa el bit de control del multiplexor que va al PC para que este albergue la nueva instrucción a procesar en vez de la inmediata (a la que se accede sumando 4 al PC). Esto hará que el siguiente ciclo la IM tenga el valor de la instrucción a la que se ha saltado. Sin embargo, cuando la instrucción de salto está en la fase de ejecución la IM todavía alberga la instrucción siguiente que se tendría que hacer sin tener en cuenta el salto, su valor será el correcto, como ya se ha dicho, en el siguiente ciclo de reloj.

Conclusiones

Tras hacer la práctica he aprendido cómo un microprocesador con pipeline divide cada instrucción en etapas más pequeñas. En el caso de el procesador 5 stage del ripes he aprendido cuál es cada una de esas etapas (instruction fetch, instruction decode, execution, memory access y write back) y cómo las implementa.