

TRABAJO PRÁCTICO ESPECIAL

Arquitectura de computadoras 1 - 2021



Integrantes:

[249843 - Coria, Santiago]

santiagocoria@live.com.ar

[248110 - Aguiar, Ismael]

aguiarismael.ai@gmail.com

Introducción

En el trabajo se implementó el procesador MIPS segmentado visto en la teoría, en el cual para la primera etapa de *instrucción fetching* se realizaron la implementación de estos componentes:

- Program counter (PC): es un registro que asigna la dirección, la cual nos dirá qué instrucción del programa leer.
- Mux dirección de salto: Este mux tiene un selector el cual cuando sea 0 leerá la dirección que calcula el sumador y si es 1 significa que va a suceder un salto por lo tanto le asigna la dirección de salto.
- Sumador: El sumador se encarga de calcular la dirección siguiente entonces lo que hace es sumarle 4 en hexa a la dirección actual.

Luego consta de un componente más que sería instruction memory el cual proporcionó la cátedra y este almacena el programa a leer.

En la siguiente etapa se procede a decodificar la instrucción. En esta etapa aparecen tres componentes:

- Banco de registro: Este se encarga de almacenar valores en sus registros y nos permitirá hacer operaciones aritméticas junto con la ALU.
- Unidad de control: Se encargará de leer del bit 31 a 26 de la instrucción y a partir de ahí asignar los valores que sean necesarios para ejecutar la instrucción.
- Extensión de signo: se encarga de completar la instrucción para que sea de 32 bits dependiendo del valor que tenga el bit 15.

En la tercera etapa recibe el nombre de ejecución y se encarga de ejecutar la instrucción decodificada, esta instancia consta de tres componentes los cuales son:

- Sumador: se encarga de calcular la dirección de salto.
- Mux AluSrc: dependiendo de la instrucción se usará el valor que tenemos en el registro o sino se usará el valor inmediato.
- Alu control: se encarga de enviarle a la ALU el código de la operación que tiene que hacer, para instrucciones de R-type se fijará en los bit de 5 a 0 y decidirá a partir de esto que hacer.
- ALU: Se encargará de hacer las operaciones aritméticas con los dos valores que le lleguen y decidirá cual hacer a partir del código de operación que le envíe la ALU Control.

En la cuarta etapa se encargará de guardar los valores en la memoria o utilizar los datos que hay en la misma dependiendo de la instrucción además en esta etapa se hará un and entre zero que es el valor calculado por la operación que se realizó en la ALU y branch que se decodifico de la instrucción.

Por último esta la quinta etapa la cual recibe el nombre de write back y esta cuenta un mux que nos dirá de donde proviene el dato a guardar.

Además en cada etapa se utilizarán registros que almacenan valores necesarios para realizar la instrucción en cada etapa.

Experiencia

Cuando se estaba implementando el procesador los principales problemas surgieron en la etapa de decodificación cuando se estaba implementando la unidad de control ya que nos trabamos en que valor tenia que tomar cada señal, pero una vez nos pusimos a hacer bien el seguimiento de cada instrucción paso por paso que valor tenia que tomar cada señal pudimos solucionarlo y resolver el problema.

Para implementar la unidad de control se decidió utilizar una estructura de tipo case la cual se fija en los bit 31 al 26 de la instrucción y compara con cada código de instrucción definido en el enunciado del práctico a partir de esto se le asignaban sus respectivos valores a las señales.

Luego se hizo algo parecido para la ALU Control en la cual recibe el código de operación y esta sí hacía referencia a una instrucción de tipo R miraba los últimos 6 bit y si hacia match con alguno mandaba el código correspondiente a la ALU para que realice la operación y si no era de tipo R miraba cuál era su código de operación nada más y manda el código que le correspondía a la ALU.

Una vez que se terminó la implementación empezó el testeó y en este se descubrieron varios errores en los cuales se fueron solucionando casi todo menos los que se corresponden a la operación LUI, este mismo no entendemos porque se da porque el valor se calcula bien ya que hicimos el seguimiento de la señal que nos dice cual es el valor que se va a guardar en el registro pero en el momento de guardar se guarda 00000020.

Cuando se estaba realizando el testeó descubrimos que el programa en que nos habían proporcionado no hacía referencia al program1 entonces eso dificultó un poco el testeó porque pensamos que estábamos poniendo mal los códigos especiales y después cuando subieron el programa sin la operación de BEQ se nos facilitó hacer el seguimiento del mismo pese que también tenía un mini error en la instrucción del ORI porque el valor inmediato en realidad es 70.

Conclusión

Realizar este trabajo nos permitió acoplar lo visto en la teoría e implementarlo en la práctica esto en nuestro caso nos ayudó ya que pudimos comprender más a fondo cómo trabaja el procesador en cada etapa cuando realizaba las instrucciones.