ARQUITECTURA DE COMPUTADORAS 2017

Laboratorio 1 – MIPS en VHDL

1 INTEGRANTES DEL GRUPO

- Nahomy Estephanya Puente Cancho
- Agustín Andrés Mallea
- Diego Sebastián Camus

2 ESTRUCTURACIÓN DEL CÓDIGO

- DE0_NANO
  - Mips
    - Controller
      - Aludec
      - maindec
    - datapath
      - fetch
      - ifde
      - decode
      - deex
      - execute
      - exme
      - memory
      - mewb
      - writeback

3 DECISIONES DE DISEÑO

Al finalizar el practico 3 nuestro MIPS ya tenía las siguientes 5 etapas bien definidas: Fetch, Decode, Execute, Memory y WriteBack y era single cycle. Para agregar pipelining lo que hicimos fue crear 4 componentes que se corresponden a las etapas intermedias: Fetch-Decode, Decode-Execute, Execute-Memory y Memory-Writeback. Estos actúan como buffers preservando la información a través del flujo del programa utilizando internamente flipflops.

Luego conectamos las etapas entre ellas y agregamos las señales de control, todo esto dentro del componente “datapath”.

4 DIFICULTADES

Como dificultades podríamos mencionar la minuciosa tarea de conectar todos los componentes correctamente ya que son muchas señales y tuvimos que revisar varias veces porque había componentes mal conectados o directamente sin conectar.

Al realizar los ejercicios y pensando que nuestro MIPS con pipeline estaba bien nos dimos con que los testbenchs no se correspondían a lo que estábamos programando. Esto sumo dudas sobre si estábamos programando mal en MARS, o el MIPS estaba mal hecho o ambas. Afortunadamente nos dimos cuenta de que al programar los ejercicios nosotros asumíamos que contábamos con forwarding unit lo cual era incorrecto, así que corregimos los ejercicios teniendo en cuenta la ausencia de esta unidad y todo funcionó.

5 TESTBENCHS

Los testbenchs se hicieron todos utilizando el University Program VW.

El "clock" se seteo a un pulso cada 10ns, y la señal de "reset" a 1 durante los dos primeros ciclos de clock. Tenemos tres señales de salida:

- MemAddress: se corresponde a la dirección de memoria.
- MemData: se corresponde al dato en sí.
- MemEna: se corresponde a si esta habilitada o no la escritura.

Esto nos permitió analizar los programas que fuimos haciendo y que cargamos en nuestro MIPS mediante archivos del tipo MIF. Para conseguir este tipo de archivo utilizamos el programa MARS en el cual programamos en código assembler nuestra lógica y luego hacíamos un "dump" para conseguir el código del archivo "imem.mif".

Tener en cuenta que a la hora de realizar los testbenchs se debe establecer como entidad de mayor nivel al componente "mips", de otra manera los resultados serán incorrectos.

6 RESOLUCIÓN DE EJERCICIOS

6.1 EJERCICIO 1

a) Se producen hazards de datos.
b) Se presentan dos escenarios en la ocasión de utilizar reordenamiento estático para evitar los stalls en este caso.
   Si asumimos que nuestro MIPS no tiene una forwarding unit, no es posible evitar los stalls con el reordenamiento.
   En caso de poseer efectivamente la forwarding unit, es posible evitar los stalls.
c) Se puede evitar los stalls introduciendo nops.
d) De nuevo, si nuestro MIPS no posee forwarding unit (es el caso de nuestro MIPS) podríamos reescribir el código de la siguiente manera:
   - addi $t0, $zero, 1
   - Nop
   - Nop
   - Nop

- addi $t1, $t0, 2
- Nop
- Nop
- Nop
- addi $t2, $t1, 2
- Nop
- Nop
- Nop
- addi $t3, $t2, 2
- sw $t0, 0($zero)
- sw $t1, 4($zero)
- sw $t2, 8($zero)
- sw $t3, 12($zero)
- exit: beq $zero $zero exit

En el caso de poseer forwarding unit podríamos rescribirlo así:

- addi $t0, $zero, 1
- sw $t0, 0($zero)
- addi $t1, $t0, 2
- sw $t1, 4($zero)
- addi $t2, $t1, 2
- sw $t2, 8($zero)
- addi $t3, $t2, 2
- sw $t3, 12($zero)
- exit: beq $zero $zero exit

6.2 EJERCICIO 2

Para este ejercicio propusimos dos soluciones:

- La primera ubicada en la carpeta “ejercicio2” es un código que debería encender todos los 8 leds de la FPGA, solo eso.
- La segunda ubicada en la carpeta “ejercicio2extra” es un código que debería alternar el encendido de los leds (0, 2, 4, 6, 8) y (1, 3, 5, 7) de forma alternada por tiempo indefinido.