

Arquitectura de computadoras Práctica 2

Diseño y simulación de un procesador uni-ciclo basado en la arquitectura MIPS

Actividad a realizar

Implementar en Verilog un procesador basado en la arquitectura MIPS, el cual sea capaz de ejecutar las instrucciones *add, addi, sub, or, ori, and, andi, lui, nor, sll, srl, lw, sw, beq, bne, j, jal, jr* las cuales se deben apegar a la especificación del MIPS Green Sheet. Dicha implementación puede usar como punto de partida el data-path visto en clase, el cual fue diseñado para soportar algunas de las instrucciones anteriores. Esta implementación debe ser capaz de ejecutar el programa que se utilizó en la practica 1, note que esta implementación debe soportar funciones recursivas.

El módulo del MIPS_Processor debe emplear solo Verilog estructural, es decir, crear instancias de los módulos básicos, tales como la ALU o el register file y conectar dichas instancias. Las definiciones de los módulos básicos se encuentran en el archivo BasicMIPSProcessor.rar, el cual se puede descargar desde Moodle. Se puede emplear el comando assign dentro del módulo MIPS_Processor para definir buses especiales o crear salidas en el top-level.

Para poder ejecutar las pruebas se necesita que estén disponibles el archivo text.dat en el mismo subdirectorio que el proyecto de Modelsim. Este archivo contiene las instrucciones que se cargarán en la memoria de programa. El archivo text.dat se puede generar usando el programa MARS, ensamblando un programa de prueba y exportando los datos del segmento .text (segmento de programa) en formato texto hexadecimal.

En Modelsim debe poder observarse la ejecución individual de cada instrucción a través de las distintas señales (buses y señales de control) que conforman al diseño. Además, el valor de PC (0x400000) y el address (0x10010000) de la memoria de datos deben coincidir con la simulación en el MARS.

Entregables

Antes de la fecha y hora indicadas subir a Moodle los siguientes entregables. Solo es necesario que uno de los integrantes del equipo suba esta información.

Reporte:

1. Incluir en el reporte el IC, CPI, clock rate y CPU time para implementación del MIPS. En el caso del IC se tiene que especificar el porcentaje de instrucciones de tipo R I y J.
2. Incluir la micro-arquitectura propuesta (esquemático como los vistos en clase hecho un Visio o un programa equivalente), es decir, las modificaciones que permiten al MIPS ejecutar el programa de las torres de Hanói.
3. Se debe documentar las partes de los códigos de Verilog que se modificaron, estos es las instrucciones que se agregaron en las unidades de control así como la incorporación de nuevo hardware que permite la ejecución de dichas instrucciones (solo modificaciones).
4. Explicación de la razón por la cual se modificó (o agrego módulos) el modelo en verilog.
5. Se debe entregar un programa y simulación en Modelsim para cada una de las siguientes de las instrucciones, beq, bne, lw, sw, j y jr.
6. Se debe incluir los resultados de simulación para el caso de 3 discos. Es decir, segmentos de simulación donde se muestren los movimientos en la memoria de dato (ver video torres de Hanoi en modelsim).
7. Se debe incluir un diagrama de flujo donde se explique la forma en la que se implementó el programa torres de Hanói.
8. El archivo hanoi.asm con el código de la práctica 1, ya con las correcciones necesarias si no funcionó al 100%, y el archivo text.dat. Su procesador debe ser capaz de resolver el problema de las torres de Hanói sin errores.
9. La memoria de datos del MIPS a implementar deberá tener una capacidad de 256 Word.
10. Se deben incluir los archivos Verilog que se generaron para práctica.

Para el desarrollo de la práctica se debe utilizar el control de versiones git, y usa el repositorio remoto git hub. Para aprender el uso de git se recomienda hacer el siguiente tutorial o alguno equivalente.

<https://www.codecademy.com/es/learn/learn-git>

Para obtener un repositorio remoto hay que registrarse en la pagina

<https://education.github.com/>

El repositorio que se utilice para la práctica tiene que ser privado, de lo contrario se anulará la práctica.

Reglas

1. Cualquier sección del programa de torres de Hanói en el MARS, debe poder compararse con la simulación en Modelsim. De la misma manera el resultado final debe poder visualizarse en la memoria de datos en Modelsim.
2. Esta práctica deberá realizarse en equipos de 2 personas. Excepcionalmente se aceptará que se realice individualmente.
3. No se puede hacer equipo con integrantes del otro grupo de Arquitectura de Computadoras.
4. El módulo del MIPS_Processor debe emplear solo Verilog estructural. No se aceptarán prácticas que empleen en este módulo otros estilos de codificación de Verilog. Se puede emplear el comando assign dentro del módulo MIPS_Processor.v para definir buses especiales.
5. Está prohibido usar soluciones que encuentren en Internet. Los equipos que usen programas bajados de Internet y los presenten como propios, tendrán calificación de cero en esta práctica. Lo mismo aplica si se emplean secciones de un programa publicado en Internet.
6. Está prohibido compartir información con otros equipos de su grupo o del otro grupo. Las prácticas que contengan código común, ya sea en la totalidad del programa o de una sección, tendrán calificación de cero, tanto del equipo que copió, como del que compartió información.
7. Está prohibido emplear soluciones de semestres pasados. Las prácticas que contengan código de prácticas de semestres pasados, ya sea en la totalidad del programa o de una sección, tendrán calificación de cero.
8. Si no se suben los archivos (reporte y archivos .v) en tiempo y forma a moodle se tomara como practica retrasada.
9. Está prohibido utilizar cualquier otra versión de procesador MIPS que nos sea la que se encuentra en el moodle de la clase.
10. Revisa el manual del MIPS que se encuentra en moodle.

Evaluación

Estos son los elementos a evaluar y sus respectivos pesos:

1. **Funcionalidad.** Se considera que el procesador (modelo en Verilog) funciona solo si se compila sin errores en Quartus y Modelsim, además debe ser posible en este último. Este aspecto tiene un peso del 30%. Si la descripción no se puede compilar, o la simulación no termina o se aborta, se obtendrán cero puntos en este apartado. Si es posible simular la descripción en Verilog, pero contiene errores funcionales, se asignaran puntos de manera proporcional a la funcionalidad lograda. En particular, la simulación en modelsim tendrá 20% y la implementación en el FPGA 10%.
2. **Reporte.** Se evaluará claridad y sencillez en la redacción, ortografía, que se presenten las decisiones clave que tuvieron que tomar para resolver el problema, que contenga la información solicitada (diagrama de bloques, ecuaciones) y que los datos de los integrantes estén completos y sean correctos. El peso de este apartado es de 30%.
3. **Preguntas.** Durante la revisión de la práctica, el profesor hará varias preguntas a ambos integrantes del equipo. Cada pregunta va dirigida a un integrante en particular (el otro integrante no puede intervenir). Las preguntas son sobre la práctica y sobre los conceptos necesarios para realizarla. Se cuestionarán también las razones por las que decidió implementar una solución específica. Ambos integrantes deberán estar enterados de todos los aspectos de la práctica. Este apartado tiene un peso del 30%. Los puntos obtenidos dependerán de la proporción de preguntas contestadas correctamente. La nota obtenida en este apartado es individual.