

Proyecto de Diseño de un Procesador MIPS-Based

1. Especificaciones:

1. La arquitectura estará basada en los principios de MIPS.
2. Se implementará el siguiente sub conjunto de instrucciones de la arquitectura MIPS:
 - Instrucciones aritméticas, lógicas y “shifts” (slides 5-7, 9-10)
 - Instrucciones load/stores (slide 11)
 - Instrucciones de mover (slide 13)
 - Todas las instrucciones de branch y brinco incondicional (slides 14-16)
 - Instrucciones de trap condicional (slide 26)
 - Las instrucciones privilegiadas *MFC0*, *MTC0* y *ERET* (slides 21 y 29)
3. Manejo de traps e interrupts:
 - El procesador tendrá una señal externa de Reset cuyo efecto es llevar al sistema al estado inicial del microprocesador que eventualmente llevará a iniciar la ejecución de la primera instrucción que estará en la localización de memoria 0. (*esto es un desvío de la arquitectura MIPS*)
 - El procesador tendrá una señal externa de Hardware Interrupt (HI) cuyo efecto es llevar al sistema la rutina de excepción mas privilegiada. Este interrupt es non-maskable. El mismo debe llevar el sistema a ejecutar la instrucción ubicada en la localización de memoria 384.
 - El procesador tendrá una señal externa de un Maskable Interrupt (MI) el cual será el de segunda mayor prioridad. Este interrupt corresponderá al interrupt IP2 del registro Cause y a IM2 del registro de Status.
 - El microprocesador generará una excepción de instrucción ilegal cuando se detecten instrucciones ilegales (que no tienen privilegio para ejecutar) o inexistentes (cualquier otra instrucción que no sea de las descritas en la parte 2).
 - Cada vez que se atiende una excepción el procesador entra en modo privilegiado. En este modo es que se pueden ejecutar las instrucciones privilegiadas.
 - Además de las instrucciones de trap y los interrupt o exepciones antes descritos solo las instrucciones de suma y resta con signo pueden generar la excepción de Overflow
 - Las localizaciones en donde deben iniciar cada una de las excepciones se indican a continuación:

Excepción	Localización Inicial
HI	384
MI	400
Instrucción ilegal	416
Trap condicional	432
Overflow	448

(esto es un desvío de la arquitectura MIPS)

4. Para demostrar el proyecto necesitan conectar el procesador a una memoria RAM de al menos 512 bytes. Esta debe estar organizada Big Endian y debe proveer acceso tanto a bytes, halfwords y words. Es imprescindible que la memoria sea accedida de manera asincrónica mediante handshaking.
5. El circuito del procesador debe ser implementado utilizando Verilog. Sin embargo, solo los multiplexers, la unidad de control, el ALU y shifters pueden ser implementados de manera behavioral. El register file debe ser implementado de manera estructural a nivel de multiplexers, registros y gates. Cualquier otro circuito debe ser representado a nivel de componentes en el datapath pero puede ser implementado de manera behavioral con Verilog.
6. La unidad de control debe ser implementada según el ejemplo discutido en clase. Debe haber una correspondencia de uno a uno entre el diagrama ASMD y el código de la unidad de control. En la unidad de control no puede haber integrado ningún tipo de registro que corresponda al datapath. La unidad de control solo recibe señales de status, reset, interrupt y clock y genera señales de control para el datapath.
7. Las especificaciones anteriores son un **requisito**.

2. Plan de Trabajo:

Para garantizar el progreso del proyecto se observará el siguiente calendario de actividades.

13 de marzo de 2015 - Informe preliminar de la Organización del Procesador

Codificación en Verilog y simulación de prueba de los siguientes componentes del datapath:

- Registro de 32 bits con “load enable” síncronico y “clear” asíncronico.
- Multiplexers de buses de 32 bits: 2 a 1 y 4 a 1
- Arithmetic Logic Unit (ALU) que dado dos números de 32 bits y un bit de carry realice cualquiera de las operaciones aritméticas, lógicas o shift. Las salidas del ALU serán un número de 32 bits que es el resultado de la operación y los cuatro bits que corresponden a los condition codes (N, Z, C, V).
- Register File tipo Register Windows (Deben preparar de antemano un diagrama de interconexión)
- Memoria RAM

Cada integrante de cada grupo será responsable del diseño y simulación de uno de los últimos tres componentes.

2 de abril de 2014 - Informe preliminar de la Organización del Procesador

El grupo de trabajo someterá una copia del diagrama de bloque de datapath del procesador, un diagrama ASMD y una tabla de las señales de control para cada estado necesario para implementar cada una de las instrucciones de la arquitectura. En el diagrama ASMD solo muestren las señales de control tipo Mealy (las que dependen de algún input del sistema. Para este informe el datapath debe estar implementado y simulado en Verilog. Deben estar preparados para explicar la operación del datapath.

9 de abril de 2015 - Show time!!! – Demostración de Simulación del Procesador e Informe Final

En este día el grupo de diseño demostrará una simulación del circuito del procesador. Deben además someter su informe final Este informe será escrito, no debe exceder 10 páginas y debe incluir lo siguiente:

1. Diagrama ASMD
3. Diagrama de bloque del datapath incluyendo un diagrama de bloque del Register File.

4. Diagrama de circuito del procesador
5. Código utilizado para implementar el circuito y la simulación

Cada integrante del grupo debe conocer la organización de su proyecto en detalle pues habrá una sección de "minutos infelices" de forma individual.

Además del informe los(as) miembros(as) de cada grupo deben entregar una evaluación de su contribución al proyecto y de la contribución de cada uno de sus compañeros(as). **Sugiera una calificación para cada uno de ellos(as).** Estas evaluaciones deben ser entregadas individualmente en sobres sellados.

Todas las entregas de documentos serán por vía electrónica.

Evaluación:

Codificación y simulación de componentes datapath	15%
Diagrama de Circuito	40%
Simulación	30%
Minutos infelices sobre implementación	15%

"Introduce Sugar Cane"