

Proyecto de Ingeniería de Computadores

Procesador SISA

Constantino Gómez
Cristóbal Ortega

0 Índice

<u>0 Índice.....</u>	<u>2</u>
<u>1 Introducción.....</u>	<u>3</u>
<u>2 Descripción del entorno de desarrollo.....</u>	<u>4</u>
<u>3 Planificación.....</u>	<u>5</u>
<u>4 Valoración.....</u>	<u>6</u>
<u>5 Conclusión.....</u>	<u>7</u>
<u>6 Ampliación.....</u>	<u>8</u>
<u>7 Referencias.....</u>	<u>9</u>

1 Introducción

El propósito de este proyecto es aprender cómo hacer un procesador básico con instrucciones reales como las del procesador SISA, que hemos usado en distintas asignaturas durante la carrera, y así poder tener una base de como funcionan y diseñan procesadores.

La importancia del proyecto para nuestra formación es enorme, ya no porque estemos en un grado de informática, sino que además hemos elegido la especialidad de Ingeniería de computadores en la cuál vemos mucho más hardware que en otras especialidades, y por ello necesitamos como mínimo una base para poder entender cómo funcionan procesadores más avanzados o simplemente con todas las funcionalidades (BIOS, interrupciones...) para poder usarlo en un ordenador real.

La planificación que hemos intentado seguir durante el desarrollo del procesador es la sugerida en la asignatura. Siguiendola bastante bien hasta la etapa de memoria, donde nos detuvimos un tiempo para probarla en la placa físicamente. Por tanto hemos ido con 1 sesión o 2 por detrás de lo sugerido por el calendario.

Los resultados al acabar el procesador son bastantes satisfactorios ya que hemos conseguido implementar todas las funcionalidades requeridas. Aunque al final, nos surgió un bug que no hemos podido corregir, el cuál nos provoca que antes de bootar el procesador se escriba en memoria 1 o 2 bytes.

Este problema no surgía en el simulador, dónde el permiso de escritura de la memoria nunca se activaba, en la placa por las pruebas que hicimos vimos que se activa durante un momento antes del booteo.

2 Descripción del entorno de desarrollo

Para este proyecto se ha utilizado la placa DE1 del fabricante Terasic. Los componentes que hemos usado para procesador son:

- FPGA modelo Cyclone del fabricante Altera
- Memoria SRAM de 512 KB

El entorno de desarrollo software consta de varias herramientas, la herramienta principal de desarrollo es Quartus II 12.1, es un IDE que incorpora editor de texto con soporte para lenguaje VHDL, puede verificar y compilar proyectos VHDL para gran cantidad de modelos de placas de la marca Altera y permite ver esquemas de los componentes lógicos del diseño.

ModelSIM es una herramienta de simulación de circuitos VHDL, nos permite verificar hasta cierto punto que el comportamiento del diseño es correcto y en caso contrario detectar en qué parte del hardware se produce el error.

Para comprobar los resultados en la placa, poder cargar las instrucciones, limpiar la memoria y demás, usamos un software específico para la placa, el DE1 Control panel.

Para programar y compilar los distintos juegos de prueba de las instrucciones hemos usado las herramientas disponibles en la web de la asignatura (mayormente objdump y compilación)

3 Planificación de etapas

Etapla 1. Procesador base

En esta etapa nos ajustamos bastante al guión, sin realizar ningún cambio destacable

Etapla 2.1. Procesador multiciclo

Aquí seguimos el guión aunque en el diseño de grafos de estados de fetch y ejecutar lo hicimos por nuestra cuenta, más simple que el que se da en el Anexo del guión de esta etapa.

Más tarde añadimos un tercer estado de NoBoot para cuando el procesador no se ha booteado aún, en un intento de arreglar el bug de memoria descrito antes.

Etapla 2.2. Controlador de memoria

El controlador decidimos implementarlo en 2 ciclos en la escritura y 1 ciclo en la lectura. La decisión de lectura es porque no había que controlar diferentes señales, y con 1 estado nos valía. Aunque leemos 2 veces ya que no podemos ajustar el tiempo de ciclo según leemos o escribimos.

La escritura la hicimos en 2 ciclos porque había que controlar un par de señales y dejar un tiempo entre ellas.

Etapla 3. ALU

Esta etapa fue la más mecánica de todas, puesto que requiere poco diseño para implementarla.

Etapla 4. Saltos

La decisión más significativa de aquí fue no tocar el control de cargar el PC o no. Lo que hicimos fue crear un registro adicional dónde iban todas las señales de saltos o el PC + 2, y siempre cargamos ese registro como el nuevo PC.

Etapla 5. IN/OUT

Esta etapa fue también bastante mecánica, decidimos hacer la lectura en un process porque como se especifica que hay efectos colaterales en un futuro podamos implementarlos sin mucho cambio en el módulo.

4 Valoración

En general estamos contentos con el resultado final del procesador, pues funciona todo relativamente bien. Quizá la parte más entretenida de realizar hayan sido los saltos o el sistema de Entrada/Salida y por tanto estamos más satisfechos con estas partes pues hemos dedicado un tiempo mayor a realizarlas.

Por contra, la parte con la que estamos menos satisfechos es el controlador de memoria por el bug de que escriba en memoria 1 o 2 bytes antes de bootar el procesador.

Cabe destacar que cuando acabamos el controlador de memoria y lo probamos en el ModelSim funciona perfectamente, y como ya se avisó, para mayor seguridad de que funcionaba deberíamos probarlo en la placa.

Lo hicimos, y funcionaba bien, no escribía antes del booteo. Es decir que durante el desarrollo del resto del procesador algo pasó, aunque el controlador de memoria no fue tocado.

También decir que en el ModelSim, en ningún momento el permiso de escritura se pone a 1 antes de bootar.

5 Conclusión

Para concluir decir que este es un proyecto que nos ha gustado mucho, hemos “tocado” hardware por así decirlo, cosa que no hacemos en mucho en las asignaturas. Hemos aprendido un lenguaje como el VHDL, que a nuestro parecer es muy curioso.

Y hemos aprendido bastante sobre como diseñar, implementar un procesador aunque sea básico.

Quizá este proyecto debería realizar durante todo el cuatrimestre, y ver un procesador más cercano a la realidad, añadiendo interrupciones, segmentarlo quizá o añadiendo booteo.

Ya que es un proyecto muy interesante y entretenido de realizar.

6 Ampliación

Las ampliaciones para este procesador son bastantes claras:

Añadir interrupciones, ver cómo se implementan, cómo manejarlas es quizá otro punto que sería muy interesante de ver en el procesador.

Añadir soporte para periféricos, como un teclado por PS/2, crear un controlador para puertos PS/2 y así poder tener distintos periféricos conectados al procesador.

Estas son las que más nos llaman la atención, pero hay otras que también pueden estar bien de implementar, como un sistema de booteo y así poder ejecutar un SO real en el procesador, aunque antes deberíamos implementar el controlador de VGA.

7 Referencias

Todo el trabajo ha sido realizado a partir de la documentación proporcionada.