

Diseño, validación e implementación de una arquitectura RISC

Tesista

Luciano César Natale, *Padrón Nro. 86.659*
luchonat@gmail.com

Director

Ing. Nicolás Álvarez, *Profesor Adjunto*
nalvare2001@yahoo.com.ar

Co-director

Ing. Octavio Alpago, *JTP*
oalpago@gmail.com

1 de abril de 2016

1. Objeto y área de la tesis

El objetivo principal de este trabajo consiste en diseñar, validar e implementar una arquitectura RISC¹. El diseño incluirá la completa especificación funcional de la arquitectura y su conjunto de instrucciones. El diseño se validará mediante el desarrollo en software de un emulador de la arquitectura; la implementación se realizará en el lenguaje de descripción de hardware Verilog, y se sintetizará en un dispositivo FPGA².

El interés en este desarrollo proviene de la necesidad de contar con un núcleo de procesamiento altamente configurable y suficientemente flexible y sencillo para distintas aplicaciones dentro del ámbito de la investigación en los Laboratorios de Microelectrónica y de Sistemas Embebidos; sintetizable en FPGA.

El área profesional de relevancia del trabajo de tesis es el diseño de sistemas digitales y su aplicación como base de proyectos de investigación que necesiten de un núcleo de procesamiento configurable y sencillo.

¹ *Reduced Instruction Set Computer*. Técnica de diseño de unidades de procesamiento basada en el hecho de que un conjunto de instrucciones simple provee una mayor performance al ser combinado con una arquitectura capaz de ejecutar dichas instrucciones en algunos pocos ciclos de máquina.

² *Field Programmable Gate Array*. Dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada ‘in situ’ mediante un lenguaje de descripción especializado.

2. Introducción, antecedentes y actualidad

Desde la aparición de los microprocesadores a mediados de los años 70, la tendencia fue el aumento de la complejidad de las arquitecturas, generando un efecto de “bola de nieve”, al ir superponiendo capas sobre un núcleo central. Existió, entonces, una reacción adversa a esta tendencia. Por ejemplo, la arquitectura experimental de IBM 801; y también en Berkeley, Patterson y Ditzel fueron los primeros en acuñar el término RISC, para describir una nueva clase de arquitectura que deshacía el camino del resto de las arquitecturas hasta el momento, conocidas, en contraposición, como CISC³. A partir de este antecedente, los principales fabricantes de microprocesadores han lanzado al mercado sus propias implementaciones basadas en los principios establecidos en IBM y Berkeley.

El concepto de las arquitecturas RISC se basa, principalmente, en el hecho de que al simplificar la lógica necesaria para la ejecución de una instrucción permite aumentar la frecuencia de operación de las compuertas que componen la lógica. Además, es posible dividir la ejecución de las instrucciones en etapas sencillas y consecutivas, permitiendo de esta manera implementar fácilmente optimizaciones como, por ejemplo, una arquitectura de *pipeline*⁴. Es por esto que el conjunto de instrucciones es sencillo, permitiendo solamente operaciones básicas entre registros internos del microprocesador. El trabajo realizado por cada instrucción, en general, es menor que el generado por una instrucción CISC, pero se hace de manera sencilla y rápida. Es importante notar que no solamente la ganancia radica en poder aumentar la frecuencia de operación de la lógica, sino que estas condiciones facilitan el desarrollo de diseños de bajo consumo, característica muy valorada en el nicho de los sistemas embebidos.

El mercado de los sistemas embebidos es excesivamente amplio y está inserto en todas las industrias. En un automóvil, por ejemplo, podemos encontrar microprocesadores en el sistema de frenos, en la central de inyección electrónica, en el sistema de entretenimiento y navegación, etc. La otra arista de vital importancia para el mercado de los sistemas embebidos, es el de los dispositivos móviles, donde se vuelve vital el requerimiento de bajo consumo. Estamos viviendo la revolución de IoT⁵, que se trata básicamente de sistemas embebidos autónomos que están conectados a “la nube” y pueden ser monitoreados y controlados remotamente a través de *Internet*.

Dentro del universo de las arquitecturas RISC, actualmente se destacan dos: MIPS y ARM. La primera, fue desarrollada por un grupo de investigadores de la Universidad de Stanford (entre ellos John L. Hennessy, pionero del concepto RISC junto a David Patterson, coautores de la bibliografía más relevante del área). Esta arquitectura, por su sencillez, es la predilecta al momento del desarrollo de cursos enfocados en la enseñanza de arquitectura de computadoras. Si bien MIPS posee gran relevancia académica, es muy popular en el mercado de los microprocesadores en sistemas embebidos como equipos de telecomunicaciones, decodificadores de TV digital, y consolas de entretenimiento, con ejemplos muy conocidos como *Nintendo* y *PlayStation*. ARM, por otro lado, ha ganado una importante porción del mercado de los sistemas embebidos (con un gran aporte de los dispositivos móviles), basando su modelo de negocios en la venta de la propiedad intelectual (IP, *intellectual property*) del diseño de los microprocesadores a las empresas que finalmente producen el microprocesador.

3. Desarrollo previsto de la tesis

3.1. Teoría, enfoque y métodos a utilizar

El enfoque de la tesis se basará en un desarrollo teórico del conjunto de instrucciones y de las características de la arquitectura; y en el desarrollo práctico del emulador y la implementación en lenguaje descriptor de hardware.

El concepto central detrás del desarrollo será el de **ortogonalidad**. Esto implica, por una parte, que los bloques constructivos de la arquitectura que se repiten sean independientes e indiferenciables entre sí. Por otra parte, los formatos de las instrucciones, en la medida de lo posible, se diseñarán de manera tal que se pueda mantener el mismo ancho de campo para los datos inmediatos y los desplazamientos (excepto en los casos donde es explícitamente conveniente agrandarlos sin penalizar la complejidad del diseño).

³ *Complex Instruction Set Computer*. Técnica de diseño de unidades de procesamiento basadas en el hecho de que el conjunto de instrucciones debe ser lo más poderoso posible.

⁴ Técnica de diseño de arquitecturas de computadoras en la que se segmenta la ejecución de las instrucciones en múltiples etapas, permitiendo que múltiples instrucciones estén ejecutándose en paralelo.

⁵ Es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet.

El objetivo perseguido va a ser el de mantener la sencillez y la ortogonalidad, favoreciendo así la simplificación de la implementación. Se trabajará en el desarrollo de la definición de la arquitectura y su conjunto de instrucciones en favor de este objetivo. Se definirá la interfaz física para la conectividad con periféricos, los tipos de datos que maneja la arquitectura, la cantidad y tipos de registros internos, el acceso a memoria de programa y de datos con su organización y modo de direccionamiento, la interfaz con la ALU⁶ y la FPU⁷, mecanismos de manejos de excepciones e interrupciones, modos de operación y manejo de periféricos. Luego se definirá el conjunto de instrucciones que ejecutará la arquitectura.

Una vez definida la arquitectura y su conjunto de instrucciones, se procederá a diseñar los vectores de prueba para poder validar las implementaciones. Se desarrollará un emulador de la arquitectura que deberá validar los vectores de prueba diseñados. Una vez concluida esta etapa, se implementará a nivel RTL el diseño en *Verilog*. Este diseño será validado mediante simulaciones y utilizando dispositivos programables. Se validará también contra los vectores de prueba. Se analizarán los recursos utilizados en dispositivos FPGA. Se realizará un análisis comparativo entre la arquitectura desarrollada y otras arquitecturas RISC.

3.2. Estudios conexos

Asignaturas y otros estudios previstos o previos que son relevantes al desarrollo de la tesis:

- **Sistemas digitales:** asignatura que abarca la teoría de técnicas de diseño de hardware digital y codificación de sistemas digitales, así como también su síntesis y medición.
- **Organización de computadoras:** asignatura en la cual se cubren los aspectos teóricos del diseño de arquitecturas.

3.3. Alcance proyectado para la tesis

Como resultados a obtener de la tesis se tienen los siguientes:

- Especificación completa de la arquitectura
- Vectores de prueba
- Emulador de la arquitectura
- *IP Core* codificado en el lenguaje *Verilog* de la arquitectura completa
- Resultado de los vectores de prueba tanto en el emulador como en el *IP Core*
- Análisis comparativo entre la arquitectura desarrollada y otras arquitecturas RISC
- Proposición de trabajos futuros y/o mejoras.

Para asegurar que el proyecto de tesis incluya todos los trabajos requeridos, los procesos a completar se describen en un plan de trabajo en la siguiente sección.

⁶ *Arithmetic Logic Unit*. Bloque constructivo encargado de realizar las operaciones aritmético lógicas sobre los datos.

⁷ *Floating Point Unit*. Bloque constructivo encargado de realizar las operaciones en punto flotante sobre los datos.

3.4. Plan de trabajo

La duración total del trabajo se estima en un año y se considera que el mismo estará compuesto por las siguientes etapas:

- **Investigación bibliográfica:** recolección de libros, *papers*, trabajos de tesis, y fuentes de investigación con el objetivo de obtener el entendimiento teórico requerido y conocer el estado del arte en el tema a trabajar.
- **Introducción al trabajo de tesis:** comprender la teoría de las arquitecturas RISC. Describir el contenido teórico requerido para exponer los conceptos de funcionamiento del hardware a desarrollar.
- **Análisis de arquitecturas existentes:** analizar las diferentes arquitecturas existentes con sus pro y contras. Obtener de este análisis las relaciones de compromiso asumidas en ellas para poder aplicarlas en la arquitectura a desarrollar.
- **Diseño de la arquitectura y su conjunto de instrucciones:** realizar la especificación funcional de la arquitectura y su conjunto de instrucciones.
- **Validación:** desarrollar los vectores de pruebas que permitirán validar la arquitectura.
- **Emulación:** desarrollar el emulador de la arquitectura y validar los vectores de pruebas.
- **Implementación:** desarrollar el *IP core* en un ambiente de simulación y, en la medida que sea posible, en campo. Realizar la síntesis del mismo para distintos dispositivos FPGA, medir recursos utilizados, máxima frecuencia de operación y potencia consumida. Validar los vectores de pruebas.
- **Conclusiones y trabajos a futuro:** se extraerán las conclusiones pertinentes sobre los resultados obtenidos y se propondrán, si correspondiese, futuras mejoras de la arquitectura.
- **Preparación del informe final:** se consolidará la documentación con la memoria de la tesis la cual contendrá el resultado de todo el trabajo realizado. Se revisará la tesis por el Director, Co-director y por los pares antes de enviarla al Jurado.
- **Preparación de la presentación y defensa de la tesis:** se elaborarán diapositivas para la presentación con los objetivos, alcance, introducción a las arquitecturas RISC, desarrollo de la arquitectura, resultados obtenidos, conclusiones y trabajos a futuro. La misma será posteriormente utilizada para la defensa.

En la figura 1 se encuentra el plan propuesto para la formación del tesista y el cumplimiento de los objetivos.

Actividad	Mes											
	1	2	3	4	5	6	7	8	9	10	11	12
Investigación bibliográfica	■											
Introducción al trabajo de tesis		■										
Análisis de arquitecturas existentes			■	■								
Diseño de la arquitectura y su conjunto de instrucciones				■	■	■	■					
Validación								■	■	■		
Emulación									■	■		
Implementación										■	■	
Conclusiones y trabajos a futuro											■	
Preparación del Informe Final		■	■	■	■	■	■	■	■	■	■	
Preparación de la presentación y defensa de la Tesis												■

Figura 1: Diagrama de las actividades a realizar por el tesista.

4. Bibliografía

Referencias

- [1] David A. Patterson & John L. Hennessy, “Computer Organization and Design - The Hardware/Software Interface”, Fourth Edition - ARM Edition, Morgan Kaufmann, 2010.
- [2] John L. Hennessy & David A. Patterson, “Computer Architecture - A Quantitative Approach”, Fourth Edition, Morgan Kaufmann, 2010.
- [3] Miles J. Murdocca & Vincent P. Heuring “Principles of Computer Architecture”, First Edition, Prentice Hall, 1999.
- [4] John F. Wakerly, “Diseño Digital - Principios y Prácticas”, Tercera Edición, Pearson, 2010.

Buenos Aires, 1 de abril de 2016

Sr. Luciano César Natale
Tesisista

Ing. Nicolás Álvarez
Prof. Adjunto, Director

Ing. Octavio Alpago
JTP, Co-director