



Universidad de Buenos Aires



Diseño, validación e implementación de una arquitectura RISC

por

Luciano César Natañe

Tesis presentada para optar al Título de

Ingeniero Electrónico

por la

Facultad de Ingeniería de la Universidad de Buenos Aires

Director:

Ing. Nicolás Alvarez

Co-Director:

Ing. Octavio Alpago

Miembros del Jurado:

Ing. XXXXXXXXXXXXXXXXXXXXXXXX

Ing. XXXXXXXXXXXXXXXXXXXXXXXX

Ing. XXXXXXXXXXXXXXXXXXXXXXXX

Calificación: _____

Fecha: _____

Universidad de Buenos Aires, Facultad de Ingeniería

Diseño, validación e implementación de una arquitectura RISC

por

Luciano César Natale

Resumen

El presente trabajo constituye la Tesis de Grado necesaria para obtener el título de Ingeniero Electrónico de la Facultad de Ingeniería de la Universidad de Buenos Aires.

El objetivo de este trabajo es el diseño, la validación y la implementación de una arquitectura RISC con el objetivo de generar un núcleo de procesamiento, sintetizable en FPGA, altamente configurable y suficientemente flexible y sencillo para ser utilizado en distintas aplicaciones dentro del ámbito de la investigación en los Laboratorios de Microelectrónica y de Sistemas Embebidos. Se presenta la teoría e historia necesaria para poder explicar las decisiones de diseño adoptadas en el desarrollo de la arquitectura. Se presentan vectores de prueba para las validaciones posteriores. Se verifica el diseño generado mediante emuladores. Se implementa el diseño en un lenguaje de descripción de hardware. Se sintetiza en FPGA el diseño y se contrastan los resultados con las emulaciones realizadas. Finalmente se analizan los resultados obtenidos, se contrastan los mismos contra trabajos similares, y se proponen trabajos futuros.

Agradecimientos

Agradecimientos Lucho.

Dedicatoria Lucho

Índice

1. Introducción al trabajo de tesis	1
1.1. Objetivo	2
1.2. Alcance	3
1.3. Organización del trabajo	4
2. Marco teórico: arquitectura de procesadores	5
2.1. Perspectiva histórica	6
2.2.	7

Índice de Tablas

Índice de Figuras

2.1.	Esquema simplificado de un “sistema de procesamiento”	6
------	---	---

Capítulo 1

Introducción al trabajo de tesis

El presente trabajo se encuentra enfocado en el contexto del diseño de hardware digital. El mismo fue motivado por la necesidad de contar un con 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 ¹. La arquitectura a desarrollar será del tipo RISC².

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 arquitec-

¹ *Field Programmable Gate Array*: dispositivo electrónico de compuertas lógicas reprogramables en campo

² *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.

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.

tura 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.

1.1. Objetivo

La Tesis tiene como objetivo principal el diseño, la validación e implementación de una arquitectura RISC y su conjunto de instrucciones.

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

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.

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 diseñaran 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).

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.

1.2. Alcance

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.

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.

1.3. Organización del trabajo

En esta sección se describe la organización de la presente tesis. Con el objetivo de que la misma sea autocontenida, los primeros capítulos se ocupan de presentar las bases o conocimientos necesarios para comprender la totalidad del trabajo.

El desarrollo de la tesis se organiza de la siguiente forma:

- En el capítulo 2 se presentará la teoría general de las arquitecturas de procesadores y una revisión histórica sobre el tema. Se estudiará la diferenciación entre los universos de procesadores CISC y RISC y se justificará la elección de diseñar una arquitectura RISC para la tesis. Se presentarán las técnicas de diseño de arquitecturas estudiadas. Además se presentarán reseñas de otras arquitecturas actuales y sus decisiones de diseño, para luego contrastarlas con los objetivos perseguidos por el presente trabajo.
- En el capítulo 3 se presentará la especificación completa de la arquitectura diseñada, explicitando los criterios y las decisiones de diseño tomadas. Además se presentará el diseño de los vectores de prueba que se utilizarán para validar las implementaciones de la arquitectura.
- En el capítulo 4 se desarrollarán las implementaciones del emulador de la arquitectura y del *IP Core* en RTL. Dicho RTL cumplirá con ciertas condiciones de portabilidad y legibilidad del código, para que el mismo sea efectivamente un IP core. Se evaluará y validará el *IP Core* utilizando simuladores. Se explicitarán las decisiones de diseño necesarias para pasar de la abstracción del diseño a la implementación real.
- En el capítulo 5 se validarán las implementaciones del capítulo 4 mediante los vectores de prueba diseñados para el capítulo 3. El *IP Core* será sintetizado para distintos dispositivos FPGA. Se analizará en cada caso el consumo de recursos utilizados, máxima frecuencia de operación y la potencia consumida.
- En el capítulo 6 se extraerán las conclusiones pertinentes sobre los resultados obtenidos y se propondrán futuras mejoras de la arquitecturas a partir del análisis realizado.

Capítulo 2

Marco teórico: arquitectura de procesadores

En líneas generales un procesador es un sistema que permite, por un lado ingresar instrucciones y datos obteniendo en consecuencia los resultados de operar lo indicado en las instrucciones sobre los datos. No es necesario para tal fin, definir ninguna tecnología que soporte este comportamiento; se trata más bien de un desarrollo teórico. Dicho desarrollo implica distinguir las distintas partes que lo componen. En una primera aproximación, en un sistema de procesamiento podemos distinguir cuatro componentes:

- Datos de entrada
- Instrucciones
- Unidad de procesamiento
- Datos de salida

Estos componentes se relacionan de la forma mostrada en la figura 2.1. Debe notarse que los datos de entrada y las instrucciones ingresan a la unidad de procesamiento, la cual genera datos de salida como resultado de operar las instrucciones sobre los datos de entrada.

Los datos de entrada representan el dominio sobre el cual puede operar la unidad de procesamiento. Para definir un sistema de procesamiento debemos especificar, entonces, cuál es el dominio que puede manejar. Dicha definición deberá especificar el formato y cantidad de datos que acepta la unidad de procesamiento, así como los mecanismos para ingresarlos al sistema.

Las instrucciones definirán las operaciones que la unidad de procesamiento puede realizar sobre los datos de entrada. Por lo tanto, para definir un sistema de procesamiento, debemos definir qué operaciones será capaz de realizar sobre el dominio de entrada del

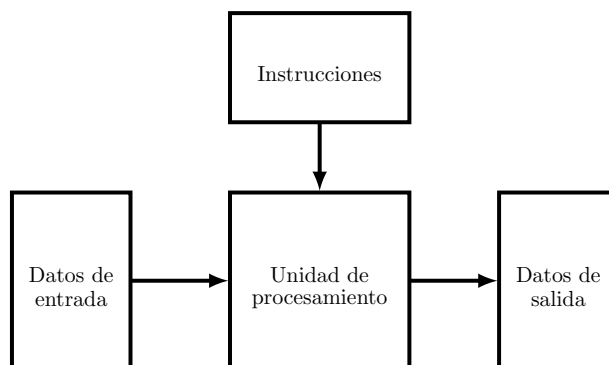


Figura 2.1: Esquema simplificado de un “sistema de procesamiento”

mismo. A su vez, se debe definir la salida esperada de las instrucciones; es por eso que al definir las instrucciones estamos definiendo intrínsecamente el dominio de salida del sistema. Es importante notar que la información sobre las instrucciones también representa una entrada para el sistema, es por eso que, como en el caso de los datos de entrada, se deberá especificar formato y cantidad que acepta la unidad de procesamiento, así como los mecanismos para ingresarlas.

Los datos de salida representan el dominio sobre el cual las instrucciones vuelcan el resultado de las operaciones realizadas sobre los datos de entrada. El dominio de salida, como fue notado antes, queda definido al definir las instrucciones, pero debe definirse el formato y cantidad de los mismos, así como los mecanismos necesarios para extraerlos del sistema.

En este contexto la unidad de procesamiento es la encargada de recibir las instrucciones y datos, ejecutar las operaciones para finalmente presentar los resultados.

2.1. Perspectiva histórica

Desde épocas remotas, el hombre se ha destacado en el mundo animal por su capacidad de modificar su entorno para resolver problemas recurrentes. Es esta capacidad la fortaleza de la especie en la naturaleza. La película “2001: Odisea del espacio”¹ narra, desde un enfoque particular, parte de la evolución del ser humano, o al menos la interpretación de los autores sobre la misma. En la misma, ubicándose temporalmente varios millones de años atrás, un clan de cavernícolas prehumanos intentan sobrevivir en condiciones extremas. Comen los pocos hierbajos que pueden encontrar en el desolado paisaje, hierbajos que para colmo han de compartir con una manada de tapires que habita la misma zona. La única fuente de agua del clan —un simple charco— les es arrebatada por un

¹2001: Odisea en el espacio es un film del año 1968 dirigida por Stanley Kubrick basada en la previa novela de Arthur C. Clarke.

clan rival. Por si fuera poco, este desdichado clan vive permanentemente amenazado por un leopardo que domina la región y que de vez en cuando caza a alguno de sus miembros. En resumen: este grupo de homínidos padece hambre, frío y miedo, y parecen condenados a una segura extinción. En ese contexto, y por motivos que no vienen al caso, aparece uno de los cavernícolas contemplando el esqueleto de un animal. Parece reflexionar sobre lo que tiene delante, como si estuviese viéndolo desde una nueva perspectiva. Hay algo nuevo en aquellos huesos. Algo que hasta entonces ni él ni ninguno de sus congéneres habían visto. Los huesos que hay tirados por el suelo pueden ser usados. El cavernícola toma el más robusto de los huesos y empieza a golpear el esqueleto; primero con precaución, más tarde con fuerza, hasta que termina consumido por un frenesí violento. Este cavernícola acaba de descubrir el primer arma —la primera herramienta— de la historia. O dicho de otro modo, acaba de aparecer el primer ser humano sobre la faz de la tierra. Gracias al uso del hueso —o de herramientas similares como palos o piedras— el clan que estaba a punto de extinguirse descubre que puede cazar a los tapires con los que convive y comérselos. Así que sus problemas de hambre han terminado. También gracias a sus armas pueden atacar al clan rival y recuperar el charco de agua, lo que soluciona también sus problemas de sed. Y deducimos que serán capaces incluso de defenderse del peligroso leopardo. Los miembros del clan ya no son prehumanos indefensos; ahora son humanos armados.

2.2.

Luciano César Natale