

PLATAFORMA PARA LA ENSEÑANZA Y PRÁCTICA DE FUNDAMENTOS DE COMPUTADORES BASADA EN UN PROCESADOR RISC SIMPLE

J.OLIVER, M.PRIM

*Departament de Microelectrònica y Sistemes Electrònics. Escola Tècnica Superior d'Enginyeria.
Universitat Autònoma de Barcelona. España.*

Este artículo presenta una plataforma para el aprendizaje de Fundamentos de Computadores basada en el procesador SR3C, un procesador simple de tres buses derivado del procesador Simple RISC Computer.

La plataforma consta del procesador descrito en VHDL, el ensamblador y un simulador asociado. La descripción del procesador en VHDL facilita su implementación en cualquier sistema de desarrollo basado en circuitos programables.

El uso de la plataforma está indicado fundamentalmente en la enseñanza de titulaciones informáticas, electrónicas y de telecomunicaciones.

1. Introducción

RISC es una filosofía de diseño de procesadores que está a favor de conjuntos de instrucciones pequeños y simples que toman menor tiempo para ejecutarse. La operación básica de la CPU es cargar dos números en registros, sumarlos y almacenar el resultado en otro registro, finalmente, toma el resultado del último registro y lo devuelve a la memoria principal. El concepto arquitectural que se esconde bajo la filosofía RISC es pues el proceso simple basado en datos que son cargados en registros dispuestos dentro de la CPU.

Las características que generalmente son encontradas en los diseños RISC son:

- Separación de las instrucciones de acceso a memoria de las instrucciones de operación aritmética.
- Codificación uniforme de instrucciones. El código de operación y los registros con que se opera se encuentran en los mismos bits dentro de cada instrucción.
- El modo de direccionamiento es simple y se encuentra especificado en la misma instrucción.
- El conjunto de registros es homogéneo, facilitando las tareas de ensamblaje y compilación.
- Trabaja con tipos de datos simples.

Esta estructura y operatividad simple de los procesadores RISC los hace adecuados en la enseñanza de fundamentos de computadores. El procesador SR3C[1] que se describe en este artículo es un ejemplo de procesador RISC simple que puede utilizarse en la enseñanza. SR3C es un procesador de tres buses derivado del procesador Simple Risc Computer (SRC[2]).

Además se ha desarrollado un ensamblador y un simulador, lo que lo hace especialmente indicado en la enseñanza de Fundamentos de Computadores. Además, SR3C se encuentra descrito en VHDL, lo que permite que también sea utilizado en asignaturas avanzadas. Puede ser útil tanto en asignaturas de arquitectura de computadoras, como núcleo para procesadores con pipeline, como en asignaturas de diseño de circuitos en los que se puede utilizar como núcleo en la construcción de DSPs.

El artículo se presenta dividido en tres apartados fundamentales. En el apartado 2 se hace una breve reseña y comparativa del procesador respecto a distintos procesadores RISC que se pueden encontrar como cores en el mercado (normalmente en herramientas de desarrollo) y que son utilizados en programas docentes de titulaciones informáticas, electrónicas y de telecomunicaciones. El apartado 3 está dedicado al procesador SR3C. Se describen las características arquitecturales del procesador y su repertorio de instrucciones. En el apartado 4 se describe la plataforma que rodea al procesador, compuesta por el ensamblador y el simulador. Finalmente se exponen las conclusiones del trabajo.

2. Antecedentes

La estructura simple que presentan los procesadores RISC ha hecho que en la última década hayan aparecido en la literatura distintos procesadores RISC que, además de utilizarse en ámbitos comerciales, algunos de ellos presentan plataformas que se utilizan en distintos ámbitos de las enseñanzas informáticas, electrónicas e ingenieras (especialmente en telecomunicaciones). Entre los procesadores RISC más conocidos se encuentran los procesadores DLX, MicroBlaze y PicoBlaze, NiosII y Nios, AVR, y el propio SRC. A las propuestas comerciales que presentan estos procesadores se han desarrollado procesadores más simples que son utilizados en el aprendizaje de fundamentos de computadores.

A modo de comparación, a continuación se exponen las características más importantes de estos procesadores.

2.1. Procesadores DLX y DLXS

El procesador DLX es un procesador teórico basado en arquitectura RISC diseñado por Hennessy y Patterson [3]. El DLX es un procesador simple basado en una arquitectura de carga/almacenamiento de 32 bits. Pensado principalmente para propósitos educativos, se utiliza ampliamente en cursos de nivel universitario sobre arquitectura de computadores. Sus principales características son:

- Tiene cuatro tipos de instrucciones: almacenamiento, aritméticas, salto/bifurcación y punto flotante.
- El repertorio de instrucciones consta de 64 instrucciones básicas de 32 bits.
- El repertorio de instrucciones es fácilmente decodificable.
- Permite un diseño de segmentación (pipelining) eficiente.
- El procesador tiene 32 registros de propósito general de 32 bits. Adicionalmente, hay un conjunto de registros de punto flotante que permiten la realización de operaciones en simple y doble precisión.

Este procesador ha sido utilizado por otros autores en sus libros para explicar conceptos relacionados con la arquitectura de computadores o el diseño de sistemas digitales (Ashenden[4], descrito en VHDL). El DLXS[5] es una variante simple de este procesador diseñado en tres buses sin ofrecer pipeline que se utiliza para la enseñanza de fundamentos de computadores.

2.2. Los procesadores MicroBlaze y PicoBlaze

El procesador MicroBlaze[6] es un core procesador descrito en software de arquitectura RISC de 32 bits diseñado por Xilinx y optimizado para funcionar en sus FPGAs. Las características fundamentales de este procesador son:

- Reconfigurable por plataforma.
- Consta de 32 registros de propósito general de 32 bits, algunos de los cuales realizan tareas específicas.
- Las instrucciones son de 32 bits con tres operandos y dos tipos de direccionamiento. Se pueden clasificar en instrucciones aritméticas, lógicas, de salto, de carga/almacenamiento y especiales.
- Basa su funcionamiento en una arquitectura pipeline de 5 etapas, siendo el número de ciclos para completar cada instrucción igual al número de etapas pipeline que emplea.
- Está implementado con una arquitectura de memoria tipo Harvard, por lo que instrucciones y datos ocupan espacios de direccionamiento diferentes.

A nivel educativo suele ser recomendable trabajar con el procesador PicoBlaze [7], mucho más sencillo y menos potente. Es un microcontrolador compacto basado en arquitectura RISC de 8 bits, diseñado y optimizado para ser sintetizado en FPGAs. Es de código abierto (en VHDL sintetizable). El procesador incluye 16 registros de 8 bits cada uno, puede ejecutar programas de hasta 1024 instrucciones y posee sistema de interrupciones.

2.3. Procesadores Nios II y Nios

El procesador Nios II [8], sucesor del Nios, es un procesador de arquitectura RISC de 32 bits y de tipo softcore desarrollado por Altera y optimizado para funcionar en sus FPGAs. Las características fundamentales de este procesador son:

- Conjunto de instrucciones y espacio de direcciones de 32 bits.
- Tiene 32 Registros de propósito general de 32 bits y soporta hasta 32 interrupciones.
- Posibilidad de incorporar multiplicación y barrel shifter por hardware.
- Proporciona acceso a múltiples periféricos dentro del propio chip e interfaces para memorias y periféricos externos.
- Es un procesador soft-core configurable.
- Arquitectura Harvard.
- Respecto al tipo de instrucciones son parecidas a las proporcionadas por el procesador DLX.
- Incorpora pipeline.

Aunque es un procesador comercial se utiliza en fines educativos en asignaturas avanzadas. Para asignaturas de iniciación a los procesadores suele ser recomendable trabajar con su predecesor Nios.

2.4. El procesadores AVR

El procesador AVR32 [9] pertenece a la compañía Atmel. Posee una arquitectura innovadora y un diseño completamente síncrono y sintetizable, proporcionándose como *Intellectual Property*. Asimismo se definen varias microarquitecturas para intentar así llegar a un rango más amplio de aplicaciones, lo que obliga al ingeniero a elegir la arquitectura para cada aplicación. Entre sus principales características se encuentran:

- Dispone de 15 registros de propósito general y un stack pointer.
- El conjunto de instrucciones del AVR32 incluye los tipos de instrucciones compactas y extendidas. Las primeras tienen una longitud de 16 bits mientras que las extendidas tienen una longitud de 32 bits.
- Ello obliga a que el conjunto de instrucciones del procesador incluya formatos muy diferentes dependiendo de la instrucción, lo que complica el ensamblaje y la compilación.
- La arquitectura AVR32 define un acelerador hardware de Java como opción en forma de máquina virtual de Java hardware.
- Los últimos procesadores incluyen hasta 3 etapas pipeline.

Al igual que con los otros procesadores existen procesadores más simples de 8 bits que trabajan con instrucciones de 16 bits y disponen de 32 registros de propósito general de 8 bits.

2.5. El procesadores SRC

El procesador SRC (Simple RISC Computer), como el DLX, es un procesador teórico introducido por Heuring y Jordan en 1997 y que se encuentra descrito en [2]. Como su nombre indica, se basa en una arquitectura RISC de 32 bits. Los autores proporcionan la descripción en notación de transferencia de registros del procesador. Entre las características principales del procesador están:

- Incorpora 32 registros de propósito general de 32 bits cada uno.
- Aunque se puede trabajar con datos de 32 bits, la memoria principal está organizada como un vector de 8 bits. Ello permite direccionar hasta 4 Gb de memoria.
- El código de operación de cada instrucción es de 5 bits y de tamaño fijo. El procesador original introducido en [2] utiliza 28 códigos de operación diferentes, de los cuales cinco son utilizados para dar soporte al sistema de interrupciones.
- Las instrucciones del SRC incluyen carga/almacenamiento, salto condicional e incondicional a posiciones de memoria almacenadas en registros, aritméticas/lógicas y de desplazamiento
- No dispone de ninguna unidad de soporte para números en coma flotante, por lo que su diseño se simplifica bastante.

Aunque el procesador SRC sólo está descrito en detalle en su arquitectura de un único bus interno, se especifica la extensión del mismo a una versión con pipeline de cinco etapas.

El procesador SRC, en su variante de un bus único, se ha llegado a sintetizar en un ASIC [11].

3. El procesador SR3C

3.1. Características principales.

Del análisis que se ha realizado de los principales procesadores, y de las prestaciones que cada procesador ofrece en su plataforma, se deducen unas consideraciones importantes en el uso en plataformas educativas básicas basadas en procesadores RISC:

- El rango del número de bits con el que trabajan va desde los 8 bits de los procesadores más simples a los 32 bits de los procesadores complejos. 8 bits implican arquitecturas simples que son adecuadas para la enseñanza, mientras que 32 bits se acerca a las propuestas comerciales actuales.
- Todos los procesadores trabajan con un banco de registros, el mínimo número de los cuales es 32. Este parece ser, pues, un número mínimo adecuado para trabajar con procesadores RISC
- Aunque se trabaje con palabras de memoria de 8 o 16 bits, es recomendable que el registro de instrucciones tenga un ancho mínimo de 32 bits. Ello simplifica la nomenclatura de las instrucciones en tanto que permite que en una misma instrucción puedan decodificarse tanto la instrucción como los registros o operandos con que se trabaja.
- Todos los procesadores utilizan una entrada/salida mapeada en memoria.
- Todos los procesadores ofrecen la posibilidad de pipelining.
- Además, algunos de estos procesadores ofrecen la posibilidad de trabajar con unidades aritméticas específicas o en punto flotante.

No todas las características anteriores son esenciales en una arquitectura para un procesador educativo en el que debe prevalecer la simplicidad frente a la velocidad de proceso. De los procesadores anteriores, el procesador SRC cumple ya con las especificaciones adecuadas para el desarrollo de una plataforma educativa simple basada en un procesador RISC descrita en VHDL que pueda usarse en distintas etapas de la enseñanza de los fundamentos de computadores. Así, la implementación del SR3C se ha realizado tomándolo como patrón. Por consiguiente, el conjunto de prestaciones del procesador SR3C, y en comparación con las prestaciones de su procesador padre (el SRC), son:

- El número máximo de instrucciones, como en el SRC, es de 32.
- Los tipos de direccionamiento permitidos, como en el SRC, son 5: inmediato, directo, indirecto, indexado y relativo.
- El número de registros, como en el SRC, es de 32.
- Se trabaja totalmente con instrucciones, buses y datos de 32 bits. Así el número de ciclos de lectura/escritura de instrucciones y datos en memoria pasa a ser de 1 ciclo. Ello conduce a que el bus de direcciones es de 32 bits. Originalmente, el SRC trabajaba con un bus de 8 bits.
- Se ha propuesto una arquitectura de 3 buses, con lo que se aumenta la velocidad de procesador. Toda microinstrucción aritmético/lógica del procesador puede ejecutarse en un ciclo, puesto que los dos operandos y el resultado pueden leerse y almacenarse en un registro en un mismo ciclo. Aunque el SRC se desarrolló inicialmente bajo una arquitectura de 1 bus, también se proponía su extensión a una arquitectura de 3 buses.

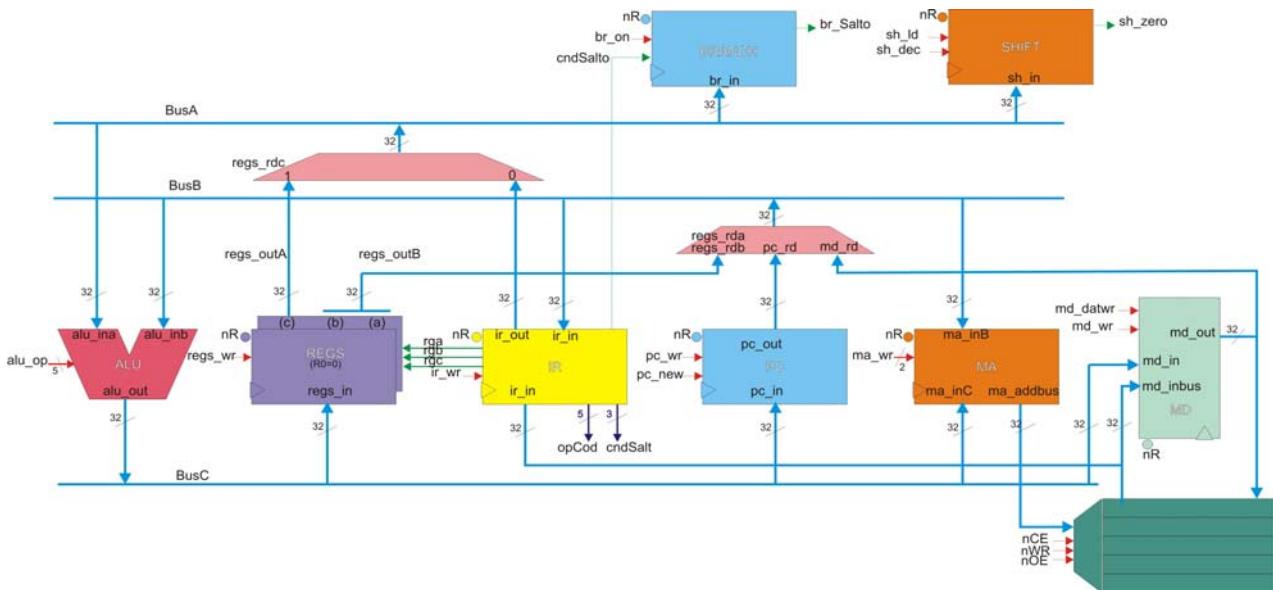


Figura 1. Unidad de proceso del SR3C.

- Al igual que la arquitectura SRC se han especificado un conjunto propio de instrucciones para el tratamiento de interrupciones.

3.2. La unidad de proceso

La figura 1 muestra la unidad de proceso que configura el procesador SR3C. Sus principales características, aparte de las ya introducidas en el apartado 3.1 son:

- La decodificación de la instrucción, en el registro de instrucciones contiene los registros o operandos con los que se trabaja. La conexión directa entre memoria y registro de instrucciones optimiza en un ciclo la ejecución de la instrucción.

- Consta de una unidad aritmético-lógica que realiza las operaciones de suma, resta, operaciones lógicas, desplazamiento de un bit a izquierda y derecha y transparencia.

- Contiene un registro de direccionamiento y un registro de datos para la comunicación con la memoria.

- Existe un módulo específico de cálculo de salto o bifurcación.

- Contiene un módulo específico para desplazamiento de datos que agiliza las instrucciones de desplazamiento.

- La memoria se considera como una caché.

- La figura 1 muestra una multiplexación de datos para con los buses. Ello se debe a que el procesador se ha descrito en VHDL sintetizable, por lo que puede materializarse en cualquier dispositivo programable. Aunque la especificación del procesador con buses clarifica el diseño, la introducción de

componentes tri-state (o de alta impedancia) en un circuito programable no es en absoluto recomendable. De ahí que el diseño se haya especificado con los multiplexores que forman parte de la unidad de proceso.

- Finalmente la unidad de control muestra las señales de control que determinan la comunicación con la unidad de proceso.

3.3. La unidad de control

La unidad de control puede especificarse tanto como máquina de estados finitos como microprogramada. Para simplicidad, se ha escrito en VHDL como máquina de estados finitos.

Debido a la propia arquitectura del procesador, el número de ciclos que requiere cada instrucción es variable. En promedio, cada instrucción del procesador SRC se ejecuta en 2.7 ciclos básicos, siendo las operaciones aritméticas las que menos ciclos de reloj requieren y las interrupciones las que más.

3.4. Repertorio de instrucciones

El conjunto de instrucciones del SR3C se encuentra dividido en siete categorías distintas dependiendo de su funcionalidad, estando formado por las siguientes instrucciones:

Instrucciones de Carga

ld ra, c2 ¹ ;	Carga directa: $R[ra]=M[c2]$
ld ra, c2(rb) ;	Carga indexada($rb \neq 0$): $R[ra]=M[c2+R[rb]]$
la ra, c2 ;	Carga directa con desplazamiento directo: $R[ra]=c2$
la ra, c2(rb) ;	Carga directa con desplazamiento indexado: $R[ra]=c2+R[rb]$
ldr ra, c1 ;	Carga relativa: $R[ra]=M[PC+c1]$
lar ra, c1 ;	Carga dirección relativa: $R[ra]=PC+c1$

Instrucciones de Almacenamiento

st ra, c2 ;	Almacenamiento directo: $M[c2]=R[ra]$
st ra, c2(rb) ;	Almacenamiento indexado($rb \neq 0$): $M[c2+R[rb]]=R[ra]$
str ra, c1 ;	Almacenamiento relativo: $M[PC+c1]=R[ra]$

Instrucciones Aritméticas

add ra, rb, rc ;	Suma en complemento a la base: $R[ra]=R[rb]+R[rc]$
addi ra, rb, c2 ;	Suma inmediata en complemento a la base: $R[ra]=R[rb]+c2$
sub ra, rb, rc ;	Resta en complemento a la base: $R[ra]=R[rb]-R[rc]$
neg ra, rc ;	Negación: $R[ra]=-R[rc]$
and ra, rb, rc ;	AND lógica: $R[ra]=R[rb] \text{ and } R[rc]$

¹ ra, rb y rc representan registros. c1 y c2 representan valores en complemento a la base.

andi ra, rb, c2 ; AND lógica inmediata: $R[ra]=R[rb]$ and c2
 or ra, rb, rc ; OR lógica: $R[ra]=R[rb]$ or $R[rc]$
 ori ra, rb, c2 ; OR lógica inmediata: $R[ra]=R[rb]$ or c2
 not ra, rc ; NOT: $R[ra]=\text{not}(R[rc])$
 xor ra, rb, rc ; XOR lógica: $R[ra]=R[rb]$ xor $R[rc]$
 nxor ra, rb, rc ; NXOR lógica: $R[ra]=R[rb]$ nxor $R[rc]$
 nand ra, rb, rc ; NAND lógica: $R[ra]=R[rb]$ nand $R[rc]$
 nor ra, rb, rc ; NOR lógica: $R[ra]=R[rb]$ nor $R[rc]$

Instrucciones de Salto

br rb, rc ; Salto a $R[rb]$ si $R[rc]$ cumple condición
 brl ra, rb, rc ; Salto con link a $R[rb]$ si $R[rc]$ cumple la condición $R[ra]=PC$
 Las condiciones de salto son: Nunca, Siempre, Cero, No cero, Mayor o cero y Negativo

Instrucciones de Desplazamiento

shr ra, rb, rc ; Desplaza $R[rb]$ a derecha en $R[ra]$, $R[rc]$ bits
 shr ra, rb, count ; Desplaza $R[rb]$ a derecha en $R[ra]$, count bits
 shra ra, rb, rc ; Desplaza Arit. $R[rb]$ a derecha en $R[ra]$, $R[rc]$ bits
 shra ra, rb, count ; Desplaza Arit. $R[rb]$ a derecha en $R[ra]$, count bits
 shl ra, rb, rc ; Desplaza $R[rb]$ a izquierda en $R[ra]$, $R[rc]$ bits
 shl ra, rb, count ; Desplaza $R[rb]$ a izquierda en $R[ra]$, count bits
 shc ra, rb, rc ; Desplaza $R[rb]$ circularmente en $R[ra]$, $R[rc]$ bits
 shc ra, rb, count ; Desplaza $R[rb]$ circularmente en $R[ra]$, count bits

Instrucciones de Interrupción.

svi ra, rb ; Guarda los reg. II e IPC en Ra y Rb respectivamente.
 ri ra, rb ; Vuelca los reg. ra y rb en II e IPC respectivamente.
 een ; Habilita las interrupciones. IE=1
 edi ; Deshabilita las interrupciones. IE=0
 rfi ; Fin de rutina de interrupción. PC=IPC e IE=1

Otras Instrucciones

nop ; No operación
 stop ; Parar ejecución

Por consiguiente, el procesador dispone de los tipos de direccionamiento inmediato, directo, indirecto, relativo y desplazado.

4. Plataforma del SR3C

4.1. Lenguaje de descripción

El procesador se ha descrito en lenguaje de alto nivel VHDL, por lo que es posible su implementación en cualquier plataforma de desarrollo basada en circuitos programables, inclusive es posible su integración.

Asimismo, cualquier modificación que se desee puede ser incorporada al procesador. De hecho, la incorporación de interrupciones aún se están comprobando y se está trabajando en el acoplo a un bus estándar como Wishbone [11].

4.2. Ensamblador

Paralelamente al diseño del procesador se ha desarrollado un ensamblador (actualmente acoplado al simulador) que facilita la codificación de las instrucciones a lenguaje máquina.

La figura 2 es un ejemplo de ensamblaje a código máquina. Corresponde a la codificación del algoritmo de Fibonacci en el SR3C.

El ensamblador facilita la programación del procesador por parte de los estudiantes, más si se tiene en cuenta que la codificación de instrucciones en procesadores de 32 bits es tediosa: cada instrucción contiene diversos campos y, algunos de estos campos se codifican de forma distinta según la instrucción. Esto puede observarse fácilmente tomando, por ejemplo, las instrucciones *la r5, 8*; y *add r2, r2, r3*;

La primera instrucción corresponde a una instrucción de carga directa con desplazamiento directo. En este caso, al registro r5 se le carga el valor 8. Por consiguiente, la codificación de la instrucción se establece de acuerdo a la figura 3.

En cuanto a la instrucción suma, la operación realiza la suma de los contenidos de los registros r2 y r3 y el resultado es devuelto al registro r2. En este caso, la codificación de la instrucción se establece de acuerdo a la figura 4.

Código ensamblador	Código máquina (posición de memoria, lenguaje máquina)
.org 0;	
main: lar r31, loop;	0 "0011011111000000000000000000010"
la r5, 8;	1 "00101001010000000000000001000"
la r1, num1;	2 "00101000010000000000000001111"
loop: ld r2, 0(r1);	3 "000010001000001000000000000000"
ld r3, 1(r1);	4 "000010001100001000000000000001"
add r2, r2, r3;	5 "0110000010000100001100000000000"
st r2, 2(r1);	6 "000110001000001000000000000010"
addi r1, r1, 1;	7 "011010000100001000000000000001"
addi r5, r5, -1;	8 "0110100101001011111111111111111"
brnz r31, r5;	9 "010000000111110010100000000011"
stop;	10 "111110000000000000000000000000"
.org 15;	
num1: .dc 1;	15 "000000000000000000000000000001"
num2: .dc 1;	16 "000000000000000000000000000001"

Figura 2. Código ensamblador y código máquina del algoritmo de Fibonacci.

El número de ciclos por instrucción requeridos por el procesador va de 3 a 5, con un CPI promedio de 3,7 ciclos/instrucción. Mejoras en el ciclo de búsqueda de instrucción permiten reducir el CPI en un ciclo. Una primera implementación en una Cyclone de Altera proporciona una frecuencia de trabajo de 9 MIPS, con una frecuencia de reloj de 20MHz.

La descripción en VHDL del procesador permite su uso en asignaturas avanzadas de fundamentos de computadores y circuitos digitales en las que el procesador puede introducirse como core en plataformas de desarrollo educativas basadas en circuitos programables.

Referencias

- [3] V.P. Heuring, H.F. Jordan. *Computer Systems Design and Architecture*. Prentice Hall. 2004
- [4] J. Gimbert. *Análisis y diseño de un procesador RISC para adquisición y proceso de datos*. Memoria del Proyecto fin de carrera. Universitat Autònoma de Barcelona. 2007.
- [5] J. L. Hennessy and D. Patterson. *Computer Architecture: A quantitative approach*. Morgan Kaufmann, San Mateo, 1990.
- [6] P. J. Ashenden. *The Designer's Guide to VHDL*. Morgan Kaufmann, San Francisco, second edition, 2002.
- [7] M. Gumm. *VLSI Design Course Notes: VHDL-Modeling and Synthesis of The DLXS RISC Processor*. University of Stuttgart, Germany, Diciembre 1995.
- [8] *MicroBlaze Processor Reference Guide*. Xilinx, Inc, Septiembre 2006. http://www.xilinx.com/ise/embedded/mb_ref_guide.pdf.
- [9] *PicoBlaze 8-bit Embedded Microcontroller User Guide*. Xilinx, Inc, Noviembre 2005. <http://www.xilinx.com/bvdocs/userguides/ug129.pdf>.
- [10] *Nios 3.0 CPU Data Sheet*. Altera, corp, Noviembre 2004. http://www.altera.com/literature/ds/ds_nios_cpu.pdf.
- [11] *AVR32 Architecture Document*. Atmel, Febrero 2006. http://www.atmel.com/dyn/resources/prod_documents/doc32000.pdf.
- [12] A. Kim, G. R. Weistroffer, D. M. Grammer, and R. H. Klenke. *The vcu src ii: A full-custom vlsi 32-bit risc processor*. University/Government/Industry Microelectronics Symposium, 2001. Proceedings of the Fourteenth Biennial, pages 201–204, 17-20 Junio 2001.
- [13] *Wishbone System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores* (Revision B.3). OpenCores.org. 7 de Septiembre 2002.
- [14] Página web oficial del programador de Python: <http://www.python.org>
- [15] Referencia a Modelsim: http://www.xilinx.com/products/design_resources/design_tool/index.htm
- [16] *UP3-1C12 Education Kit Reference Manual*. Cyclone Edition. Version 01.00. 2004.