

MATERIAL DE LABORATORIO PARA EL DISEÑO DE UN PROCESADOR

C. CUENCA Y F. QUINTANA

Departamento de Informática y Sistemas. Universidad de Las Palmas de G.C. España.

email: {ccuenca, fquintana}@dis.ulpgc.es

Se presenta el material docente de laboratorio desarrollado para el estudio del diseño del procesador en una asignatura optativa de 4.5 créditos de la titulación de Ingeniería Técnica en Informática de Sistemas. Este material docente consta de una implementación inicial simplificada del procesador MIPS R2000 y de un libro de prácticas. Las prácticas propuestas abarcan la modificación de la ruta de datos y de la unidad de control para la inclusión de nuevas instrucciones y el tratamiento de excepciones. El alumno trata con unidades de control definidas mediante una máquina de estados finitos y mediante un microprograma almacenado en memoria.

1. Introducción

El estudio de los fundamentos de los computadores está presente en muchos planes de estudios de las carreras de ingeniería. En esta materia, el estudio del procesador es un ítem característico, que la mayoría de la bibliografía de referencia cubre con la presentación detallada de un procesador simplificado que incluye el repertorio de instrucciones, la ruta de datos (*RD*) y la especificación de la unidad de control (*UC*). Una aplicación directa de este tema es la incorporación de nuevas instrucciones mediante las modificaciones oportunas de la *RD* y de la *UC*. A pesar del carácter eminentemente práctico de esta aplicación el estudiante practica estos conocimientos en clases de problemas, en lugar de hacerlo en clases de laboratorio.

Esta situación ocurre por el escaso número de créditos prácticos disponibles en las asignaturas, y la ausencia de un diseño de partida dentro de un entorno de desarrollo económico, a la vez que complejo. El bajo número de créditos prácticos favorece una selección de contenidos de más alto nivel para el laboratorio (lenguaje ensamblador). El material docente presentado proporciona este entorno de desarrollo económico, a la vez que suficientemente complejo como para profundizar en el diseño a bajo nivel.

Esta comunicación presenta el material docente desarrollado para la asignatura optativa de segundo curso Diseño de Sistemas Digitales de la titulación de Ingeniería Técnica en Informática de Sistemas de la Escuela Universitaria de Informática de la Universidad de Las Palmas de Gran Canaria. Esta asignatura añade 4.5 créditos (1.5T+ 3.0P) a la enseñanza relacionada con la arquitectura y tecnología de los computadores y con la utilización del material desarrollado para la impartición de la misma, se afronta el estudio del diseño del procesador desde un punto de vista práctico de laboratorio.

El diseño de partida captura una versión simplificada del procesador MIPS R2000 que incluye la implementación de algunas instrucciones del repertorio de instrucciones. Este procesador es actualmente uno de los procesadores más comunes para ejemplarizar el diseño del procesador o como base para el estudio de otras materias. Medrano y Catalán [1] proponen la utilización del MIPS R2000 como guía del aprendizaje del lenguaje VDHL. Gil y Guzmán [2] presentan también una realización básica del procesador MIPS R2000 donde los módulos están realizados con la biblioteca SDLC++ escrita en el lenguaje C++. Brorsson [3] propone el entorno de desarrollo MipsIt para la simulación de una implementación segmentada con memoria cache. Finalmente, DLXview [4] es un simulador interactivo de una implementación también segmentada del procesador básico MIPS R2000. Todas estas herramientas cubren apartados esenciales en el estudio del procesador, pero ninguna de ellas

permite el estudio del diseño estructural del procesador hasta el nivel detalle de las puertas lógicas, bien porque el diseño de los componentes estructurales están construidos con lenguajes de alto nivel (VHDL, C++, etc.), o bien porque las materias para la cual están propuestas son de más alto nivel (segmentación, memoria cache, etc.).

Esta presentación está organizada en tres secciones. La sección 2 describe el material docente para laboratorio, incluida una descripción detallada de los trabajos prácticos a realizar. La sección 3 describe la implantación docente de la asignatura donde este material ha sido utilizado. Finalmente, en la sección 4 se presentan las conclusiones de este trabajo.

2. Material docente

Este material docente consta de una implementación inicial simplificada del procesador MIPS R2000 y de un libro de prácticas [5]. El inicio de la realización práctica a partir de una implementación de partida de complejidad media, en lugar de comenzar con un proyecto desde cero, permite ahondar en el estudio del procesador desde las primeras horas de enseñanza, soslayando temas más propios de los circuitos digitales como el diseño de multiplexores, unidades aritmético-lógicas, bancos de registros, etc., que ya han sido estudiados en asignaturas previas.

2.1. Implementación de partida del procesador

La implementación de partida está construida con la herramienta profesional *Xilinx® Foundation Series* [6] ofertada en un programa específico para universidades, al cual hay adscritos más de 1800 universidades de todo el mundo, que incluye una licencia campus anual de bajo coste. La implementación de partida corresponde al procesador descrito por Patterson y Hennessy [7] e incluye los esquemas del camino de datos descrito con todos los detalles hasta el nivel de los circuitos digitales, así como la especificación de la *UC* mediante una máquina de estados finitos capturada mediante uno de los módulos de desarrollo de *Xilinx®*. La herramienta de desarrollo permite la simulación detallada del procesador hasta el nivel de señales lógicas.

2.2. Descripción de las prácticas

El alumno realiza un total de cuatro prácticas para ahondar en el estudio del diseño del procesador.

La primera práctica presenta el procesador básico. El estudiante trata con diseños complejos de circuitos digitales mediante la navegación a través de las jerarquías de diseño constituyentes de los módulos de alto nivel. El desarrollo práctico consiste en averiguar cuál es el programa almacenado en memoria mediante la simulación y análisis de los valores de las señales en la *RD* y en la *UC*.

La segunda práctica trata la inclusión de las instrucciones de suma inmediata (*addi*), bifurcación a un procedimiento (*jal*) y bifurcación a través de un registro (*jr*).

La tercera práctica trata las excepciones. El desarrollo práctico consiste en incluir el tratamiento de excepciones para desbordamiento aritmético e instrucción desconocida. El alumno añade también las instrucciones necesarias para escribir una rutina de tratamiento de excepciones, tales como retorno de excepción (*rfe*) y mover desde y hacia el coprocesador 0 (*mfc0* y *mtc0*).

La cuarta y última práctica aborda la temática de las unidades de control microprogramadas. La implementación de partida del procesador básico incorpora una *UC* microprogramada completamente funcional para la *RD* de partida. El desarrollo práctico consiste en realizar los cambios necesarios en el formato de las microinstrucciones y del microprograma para que la *UC* microprogramada funcione en la *RD* ampliada tras la incorporación de nuevas instrucciones y del tratamiento de excepciones.

Las prácticas incluyen también un conjunto de propuestas adicionales de realización optativa que trata los temas de inclusión de instrucciones con número de ciclos de ejecución variables, tratamiento vectorizado de excepciones e inclusión de módulos de memoria cache.

Práctica 1: Camino de datos y unidad de control

Esta práctica presenta todos los detalles de la implementación propuesta del procesador MIPS R2000 simplificado para utilizar con la herramienta profesional *Xilinx® Foundation Series*. Familiariza al alumno con las diferencias existentes entre una propuesta de diseño teórica y otra práctica. Además, el alumno comprueba directamente la aplicación al diseño del procesador de los conocimientos adquiridos durante el estudio de materias previas relacionadas con los bloques digitales esenciales (multiplexores, decodificadores, ALUs, registros, memorias, etc.).

El desarrollo práctico consiste en localizar los componentes más significativos de la ruta de datos (registro contador de programa *PC*, registros temporales, banco de registros, etc.) y averiguar cuál es el programa almacenado en la memoria del computador mediante el análisis de los valores de las señales en la *RD* y en la *UC*. Para ello, el alumno tiene que marcar los componentes más significativos y trazar las señales más relevantes de la *RD* y de la *UC*.

La Figura 1 muestra el diagrama de alto nivel de partida para la realización de las prácticas. El bloque izquierdo, de nombre “DLXSTUC”, encapsula la *UC* descrita mediante una máquina de estados finitos, y el bloque de la derecha, de nombre “DLXDP”, encapsula la *RD*. Dos son las únicas señales visibles en este nivel de complejidad, la señal de reloj (*CLK*) y una señal de reset (*RST*).

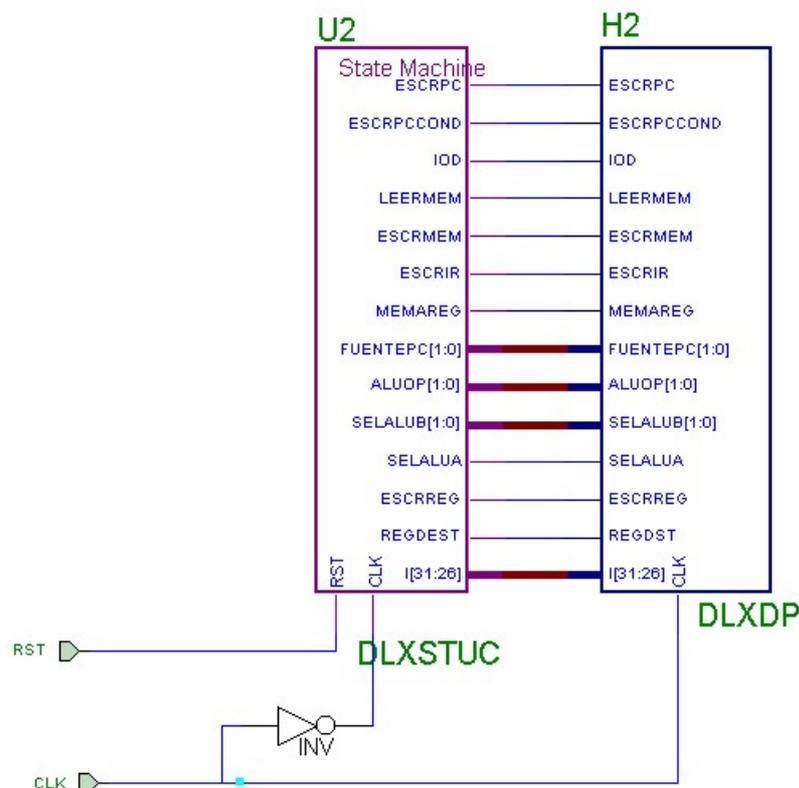


Figura 1. Diagrama de alto nivel de la implementación simplificada del procesador MIPS R2000. El bloque de la izquierda “DLXSTUC” encapsula la definición de la UC mediante una máquina de estados finitos mientras que el bloque de la derecha “DLXDP” es el módulo correspondiente a la RD.

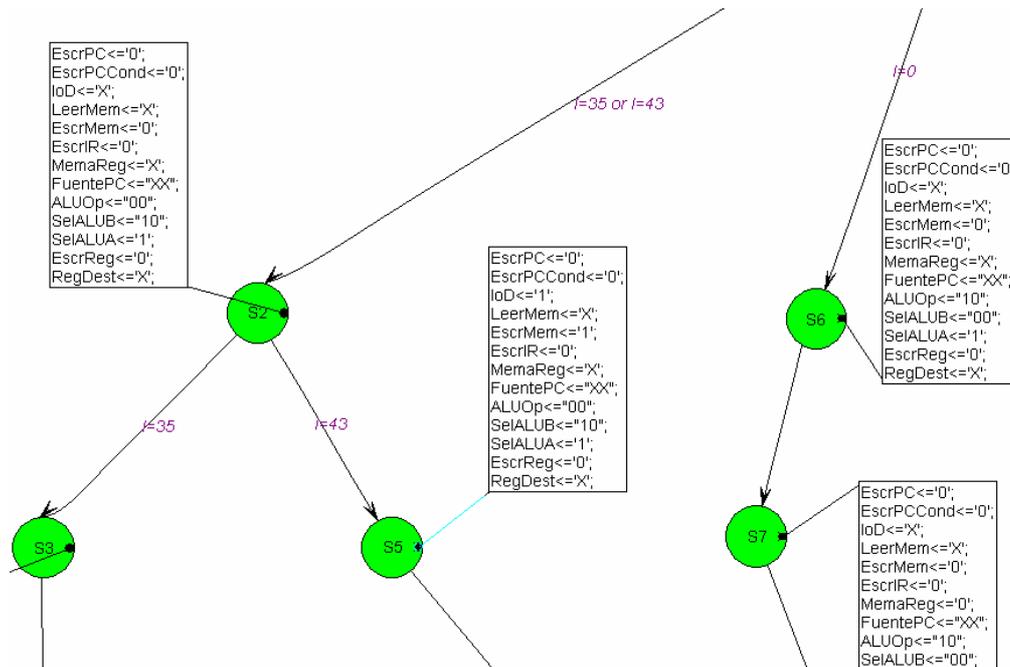


Figura 3. Vista parcial de la máquina de estados. Las acciones a realizar durante cada estado son especificadas mediante una sintaxis similar a la del lenguaje VHDL. Los arcos con etiquetas corresponden a transiciones condicionadas entre estados.

La Figura 4 muestra el resultado obtenido después de la simulación de la primera instrucción. Las señales de estímulo son la señal de reset *RST* (que lleva a la máquina de estados finitos al estado S0) y la señal de reloj *CLK*. En el cronograma están señalados los diferentes ciclos de la instrucción (IF, ID, EX, MEM y WB) y los valores de las principales señales y buses de la *RD* y de la *UC*.

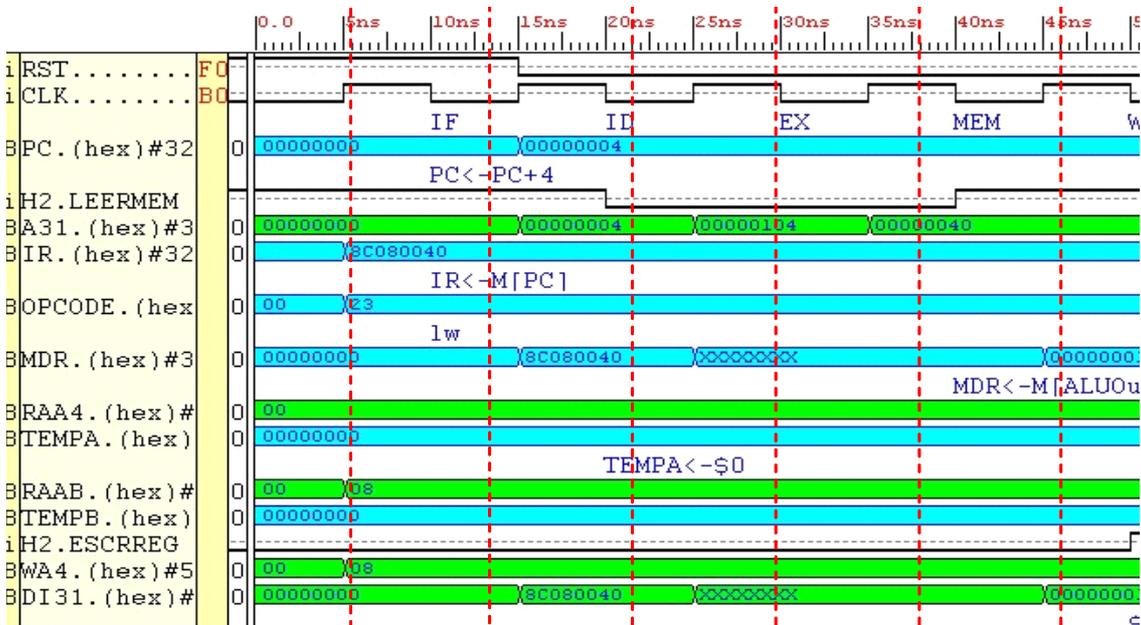


Figura 4. Cronograma resultado de la simulación de la primera instrucción almacenada en la memoria. La señal de *RST* ocasiona que la máquina de estados finitos comience en el estado S0.

Práctica 2: Inclusión de nuevas instrucciones

El repertorio de instrucciones inicial del procesador simulado es bastante limitado. Tiene sólo instrucciones básicas de carga y almacenamiento, aritmético-lógicas, de salto y bifurcación.

En esta práctica, el alumno realiza las modificaciones necesarias para proporcionar un soporte básico para el manejo de procedimientos. Añade las instrucciones *jal* (jump and link) y *jr* (jump register). También, el alumno implementa la instrucción *addi* (add immediate) para proporcionar algún manejo de constantes inmediatas.

La información proporcionada al alumno para la realización práctica coincide con la descripción en el manual del lenguaje ensamblador de MIPS. De esta manera, la incorporación de nuevas instrucciones es realizada conforme a la descripción real del procesador.

Un apartado importante de esta realización práctica es la depuración del diseño propuesto. El alumno codificará en lenguaje máquina un programa que llame a un procedimiento y retorne del mismo y que además realice una suma con un dato inmediato. La herramienta de diseño proporciona un módulo para describir los contenidos de una memoria mediante un fichero de texto (ver la Figura 5) donde las líneas con pares *dirección:contenido* (ver la Figura 6) describen el contenido de una dirección de memoria.

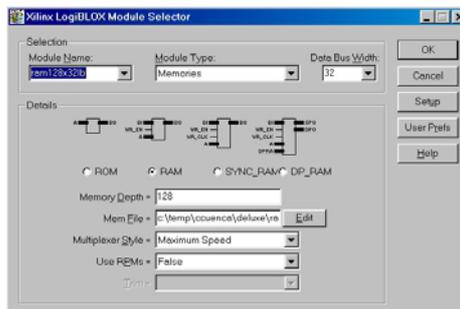


Figura 5. Ventana para la edición de los contenidos de la memoria.

```
DATA
00:8c080040;          lw $8, 0x40($0)
04:8c090044;          lw $9, 0x44($0)
08:8c0d0048;          lw $13, 0x48($0)
0C:00005020;          add $10, $0, $0
10:016b5822;          sub $11, $11, $11
14:00007022;          sub $14, $0, $0
18:8d0c0050;          bucle:lw $12, 80($14)
1C:016c5820;          add $11, $11, $12
20:01485020;          add $10, $10, $8
24:01cd7020;          add $14, $14, $13
28:11490001;          beq $10, $9, fin
2C:08000006;          j bucle
30:ac0b004c;          fin: sw $11, 0x4C($0
40:00000001;          constante 1
44:00000002;          variable N
48:00000004;          constante 4
4C:ffffffff;          variable suma
50:00000007;          primer
54:00000005;          segundo componente
```

Figura 6. Contenidos de la memoria RAM. Estos contenidos son definidos en un fichero de texto de acuerdo con la sintaxis *dirección:contenido*.

Tras ello, el alumno depurará la ejecución del mismo con la visualización y traza de las señales más significativas.

Práctica 3: Incorporación de excepciones

El alumno realiza las modificaciones de la *RD* y de la *UC* para el tratamiento de las excepciones por el procesador. Tiene que incluir el registro contador de programa de excepciones *EPC*, para guardar la dirección de la instrucción causante de la excepción, y el registro de causa de excepción *Cause*, para almacenar la causa de la excepción, definidos por la arquitectura del procesador MIPS R2000 para el tratamiento de las excepciones.

Además, el alumno aprende a escribir rutinas de manejo de excepciones. Las excepciones a manejar son las causadas por la ejecución de una instrucción con código de operación desconocido o por la ejecución de una instrucción aritmética que produce un desbordamiento aritmético. Para ello, incorporará del repertorio de instrucciones del procesador MIPS R2000 las instrucciones *rfe* (return from exception), *mfco* (move from coprocessor 0) y *mtc0* (move to coprocessor 0). Este apartado consiste en escribir una rutina de tratamiento de excepción que mueva los registros *EPC* y *Cause* a un registro de propósito general. El registro *Cause* debe ser escrito con un valor significativo para anular la causa de la excepción, y el registro *EPC* debe ser incrementado en 4 para que la ejecución a continuación de una instrucción *rfe* evite una nueva ejecución de la instrucción causante de la excepción.

Práctica 4: Microprogramación

La última práctica aborda el tema de la microprogramación. El diseño de partida incluye también una unidad de control microprogramada. El alumno comienza la práctica con la sustitución del módulo correspondiente a la *UC* descrito mediante una máquina de estados por el módulo correspondiente a la *UC* microprogramada (ver la Figura 7), ambos funcionalmente equivalentes.

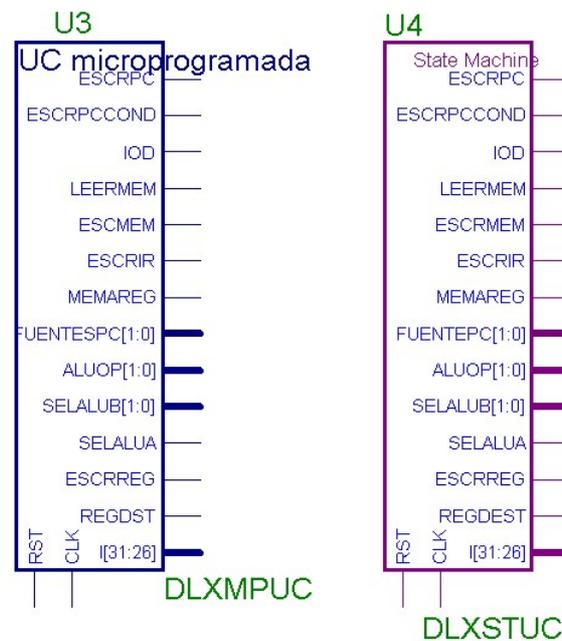


Figura 7. El diseño de partida proporciona dos UC funcionalmente equivalentes, pero de diseño estructural diferente. “DLXMPUC” es una UC microprogramada que contiene el microprograma y la lógica de secuenciación necesarios para la ejecución de las instrucciones elementales implementadas.

De manera similar a la primera práctica, un trabajo preliminar del alumno consiste en la familiarización con esta nueva unidad de control. La navegación jerárquica a través de los diferentes niveles estructurales de la *UC* proporcionará este conocimiento. La Figura 8 muestra el diagrama de bloques de más alto nivel de la *UC* microprogramada a estudiar.

modificaciones en el diseño sin haber desarrollado previamente una propuesta que tenga en cuenta todos los aspectos, pronto se dan cuenta de que esta forma de trabajar, poco sistemática, no les lleva a buen puerto. Es por esto que nos ha resultado una buena práctica docente concertar una serie de entrevistas previas al comienzo de la realización práctica a las que el alumno acude con una propuesta inicial, por escrito, de modificación de la *RD* y la *UC*. En estas entrevistas se discuten y argumentan las decisiones de diseño que se han tomado, lo que facilita el trabajo posterior de los alumnos.

La utilización de este material de prácticas ha permitido a los estudiantes afianzar los conocimientos relacionados con el diseño del procesador e incluso ha ayudado a alumnos con dificultades previas en esta temática a la superación posterior de los correspondientes exámenes de otras asignaturas.

Los alumnos que optan por esta asignatura son los alumnos con vocación para el estudio de las materias relacionadas con el diseño del procesador. La elección de esta asignatura, de entre un amplio abanico de asignaturas optativas de las más diferentes temáticas y de diferentes grados de dificultad, conforma un perfil de alumno interesado en el estudio minucioso del funcionamiento de un procesador. Durante los años de implantación, no ha ocurrido ningún caso de alumno que no superase la asignatura, dándose incluso el caso de que la mayoría de ellos realizan los trabajos optativos propuestos.

4. Conclusiones

Se ha presentado un material docente de prácticas de laboratorio sobre el diseño del procesador para una asignatura optativa de 1,5 créditos teóricos y 3 créditos prácticos de la titulación de Ingeniería Técnica en Informática de Sistemas. Este material, consistente en una implementación inicial de un procesador básico (realizado en una herramienta de bajo coste) y un libro de prácticas, permiten llegar a un alto nivel de detalle en el estudio del diseño del procesador.

Las realizaciones prácticas abarcan la modificación de una ruta de datos multiciclo, la modificación de unidades de control definidas mediante una máquina de estados finitos o con un microprograma. Las prácticas propuestas incluyen también el estudio de las excepciones.

A diferencia de otros materiales docentes, el material docente propuesto está orientado al diseño de bajo nivel. Los alumnos trabajan con bloques construidos a partir de puertas lógicas, en lugar de utilizar bloques implementados con bibliotecas construidas a partir de lenguajes de alto nivel. Una ventaja importante la proporciona el hecho de partir de un diseño inicial de complejidad media que evita tener que dedicar tiempo a temas más propios de asignaturas previas. De esta manera, el material propuesto enlaza directamente con la presentación usual del diseño del procesador en un curso previo de fundamentos de computadores. La puesta en marcha de este tipo de prácticas ha sido exitosa y ha tenido gran aceptación en el alumnado. El material desarrollado está disponible para uso docente universitario.

Referencias

- [1] *El diseño de un procesador MIPS como guía para el aprendizaje de VHDL*. C.T. Medrano, C. Catalán, A. Blesa, F. Serna y F. Martínez. Universidad de Zaragoza. TAAE 2000, Barcelona, 13-15. Septiembre. 2000
- [2] *Una simulación del procesador MIPS R2000*. J.A. Gil, J.L. Guzmán y V.G. Ruiz. Universidad de Almería. TAAE 2000, Barcelona, 13-15. Septiembre. 2000
- [3] Mats Brorsson, MipsIt. *A Simulation and Development Environment Using Animation for Computer Architecture Education*. Department of Microelectronics and Information Technology, KTH, Royal Institute of Technology Electrum 229, SE-164 40 Kista, Sweden
- [4] *DLXView*, 12 de junio de 2006 <<http://yara.ecn.purdue.edu/~teamaaa/dlxview/>>
- [5] C. Cuenca y F. Quintana. *Diseño de Sistemas Digitales*. Editorial Universidad de Las Palmas de Gran Canaria (2000). <<http://serdis.dis.ulpgc.es/~itis-dsd>>

[6] D. Van den Bout. *The Practical Xilinx® Designer Lab Book*. Editorial Prentice Hall (1999).

<<http://www.xilinx.com>>

[7] D. Patterson y J. Hennessy. *Computer Organization and Design*. Editorial Morgan Kaufmann (2005)