

# Uso de dispositivos FPGA como apoyo a la enseñanza de asignaturas de arquitectura de computadores

Francisco Charte, Macarena Espinilla, Antonio J. Rivera, and Francisco Pulgar

Departamento de Informática, Universidad de Jaén.  
{fcharte,mestevez,arivera,fpulgar}@ujaen.es

**Resumen** Los estudiantes del actual Grado en Ingeniería Informática han de cursar obligatoriamente asignaturas en las que se aborda el estudio teórico de la arquitectura de un computador, dedicándose la parte práctica fundamentalmente a la programación en ensamblador usando un determinado conjunto de instrucciones y un software de emulación. En este artículo se propone complementar esa parte práctica, introduciendo el uso de dispositivos FPGA, de forma que el estudiante aprenda a diseñar un microprocesador a partir de sus componentes básicos.

**Palabras clave:** FPGA, microprocesador, arquitectura de computadores.

**Abstract** Computer Engineering students in Spanish universities have to take one or more courses devoted to their learning of computer architecture. The theoretical part of these subjects are usually focused on describing the architecture itself, while practical sessions are used to introduce assembly programming by means of a certain instruction set which runs into a software emulator. This paper proposes to supplement practical sessions, so that students learn to design a microprocessor by themselves from its basic components, by introducing the use of FPGA devices.

**Keywords:** FPGA, microprocessor, computer architecture.

## 1. Introducción

El conocimiento detallado de las herramientas con las que se trabaja es un requisito imprescindible para cualquier profesional. En el campo de la Ingeniería informática esta necesidad se traduce en el aprendizaje de lenguajes de programación, sistemas operativos, protocolos de red y un sinfín de capas de software que es necesario configurar, administrar y gestionar. Por debajo de todas esas capas se encuentra el hardware, la máquina que, con el microprocesador actuando como CPU (*Central Processing Unit*) al frente, se encarga de ejecutar el software, traduciendo conjuntos de órdenes en operaciones concretas de procesamiento de datos.

El aprendizaje de la arquitectura del computador, por tanto, ha de ser también parte esencial de la formación en Ingeniería Informática. El diseño de computadores, el desarrollo de compiladores, la programación a bajo nivel de controladores de dispositivos o el núcleo del sistema operativo, con la optimización que permite aprovechar al

máximo las características de cada máquina, se apoyan en el saber profundo del funcionamiento del computador. Por estas razones en los estudios oficiales de Grado en Ingeniería en Informática se incluyen asignaturas como *Electrónica Digital*, *Fundamentos de Arquitectura de Computadores*, *Arquitectura de Computadores* o *Microprocesadores y Microcontroladores*, entre otras.

La metodología empleada para el estudio de la arquitectura de computadores ha ido, como no podía ser de otra manera, evolucionando con el tiempo, dependiendo de los medios disponibles en cada momento. Son muchas las universidades en las que desde hace años se recurre al uso de emuladores/simuladores [1] de una determinada arquitectura, tomada como referencia, de tal forma que los estudiantes se centran fundamentalmente en escribir código en ensamblador, ejecutarlo en dichos emuladores y analizar los resultados. De esta forma el emulador, una aplicación software, se ha convertido en la herramienta de enseñanza básica, al permitir operar con una arquitectura bien conocida y en un entorno controlado.

El progresivo abaratamiento de los circuitos FPGA (*Field Programmable Gate Array*), conjuntamente con la disponibilidad de herramientas software gratuitas y en algunos casos de acceso libre para trabajar con este tipo de hardware, abre un abanico de nuevas posibilidades para la enseñanza de arquitectura de computadores que merece la pena considerar. El objetivo de este artículo es plantear cuál podría ser la adaptación de la metodología usada para impartir asignaturas como las antes citadas, evaluando las ventajas y problemas que potencialmente aportaría el uso de hardware reconfigurable, concretamente FPGA, en sustitución o complementariamente a los medios software empleados actualmente.

La estructura del presente artículo es la siguiente: en la Sección 2 se hace una somera descripción de la metodología empleada actualmente para la enseñanza de asignaturas relacionadas con la arquitectura de computadoras, centrada en la Universidad de Jaén donde los autores desarrollan su actividad. La Sección 3 describe potenciales alternativas a la metodología existente. El objetivo de la Sección 4 es la de introducir conceptos fundamentales sobre hardware reconfigurable y, más concretamente, sobre el trabajo con FPGAs, como paso previo para, en la Sección 5, realizar propuestas relativas al uso de ese tipo de hardware como herramienta de enseñanza.

## 2. Enseñanza de arquitectura de computadores

En el libro blanco de la titulación Grado en Ingeniería Informática publicado por la Agencia Nacional de Evaluación de la Calidad y Acreditación [2] (ANECA), concretamente en la categoría *Contenidos específicos de la Ingeniería Informática*, a la que se asigna entre un 35 % y un 40 % de la carga de contenidos de la titulación, se define la subcategoría 2.4: *Ingeniería de computadores*, formando parte de ella los descriptores *Fundamentos*, *Estructura y Arquitectura de computadores*. *Tecnología de Computadores*.

Por otra parte, en el Computer Engineering Curricula 2016 [3] (CE2016) se detallan cuáles deberían ser los contenidos centrados en el estudio de la arquitectura de computadores, concretamente en el módulo *CE-CAO Computer Architecture and Orga-*

nization, al que se recomienda asignar un total de 60 *core hours*<sup>1</sup>. Entre otros aspectos se destacan conocimientos como la estructura básica de un computador, los diferentes niveles de memoria, conjuntos de instrucciones con representación a nivel máquina y ensamblador, etc.

El libro blanco de la ANECA y el CE2016 están entre los recursos más importantes a la hora de diseñar y planificar los contenidos del Grado en Ingeniería Informática en el sistema universitario español. En el caso concreto de la Universidad de Jaén<sup>2</sup>, que es el tomado aquí como referencia, los contenidos correspondientes a las recomendaciones de ANECA y CE2016 son recogidos fundamentalmente en las asignaturas detalladas en las siguientes subsecciones<sup>3</sup>.

### 2.1. Fundamentos de Arquitectura de Computadores

Esta asignatura representa el primer contacto del estudiante de Grado en Ingeniería Informática con la arquitectura de un ordenador, al ser impartida durante el primer cuatrimestre del primer curso. Cuenta con una carga docente de 6 ECTS y es impartida por el profesorado del área Arquitectura y Tecnología de Computadores.

Los contenidos teóricos de la asignatura abordan los diferentes niveles de organización estructural de un computador: componente físico, electrónico, digital, transferencia de registros, etc., la evolución de los ordenadores (generaciones), su arquitectura básica (unidad aritmético-lógica, unidad de control, registros, etc.), representación de datos a bajo nivel, gestión de entrada/salida, etc.

El programa de prácticas se desarrolla esencialmente a través de la implementación en ensamblador de soluciones a problemas cuya complejidad va creciendo paulatinamente a lo largo del cuatrimestre, usando para ello el conjunto de instrucciones Intel 8085. La ejecución de estos programas se lleva a cabo con el emulador j8085sim, cuya interfaz de usuario puede apreciarse en la Figura 1.

### 2.2. Electrónica Digital

Es una asignatura impartida en el segundo cuatrimestre del primer curso de la titulación, estando a cargo del área de Tecnología Electrónica y con una carga docente de 6 ECTS. Pertenece al módulo de formación básica, junto a materias como física y matemáticas.

El objetivo de esta asignatura es transmitir al estudiante los conocimientos necesarios sobre el funcionamiento de componentes básicos de la arquitectura de un computador, incluyendo biestables, contadores, multiplexores, sumadores y otros elementos esenciales, creándolos a partir de puertas lógicas simples. Son los componentes electrónicos a partir de los cuales se diseñan la unidad aritmético lógica, unidad de control, registros y demás partes de un microprocesador, según la estructura introducida en la anterior asignatura.

<sup>1</sup> En dicho documento se denomina así a las horas dedicadas exclusivamente a introducir nuevos conocimientos, con independencia de los medios y actividades empleadas para ello.

<sup>2</sup> <http://www.ujaen.es>.

<sup>3</sup> Las guías docentes correspondientes a estas asignaturas están disponibles públicamente en la web de la Escuela Politécnica Superior de Jaén (<http://eps.ujaen.es>).

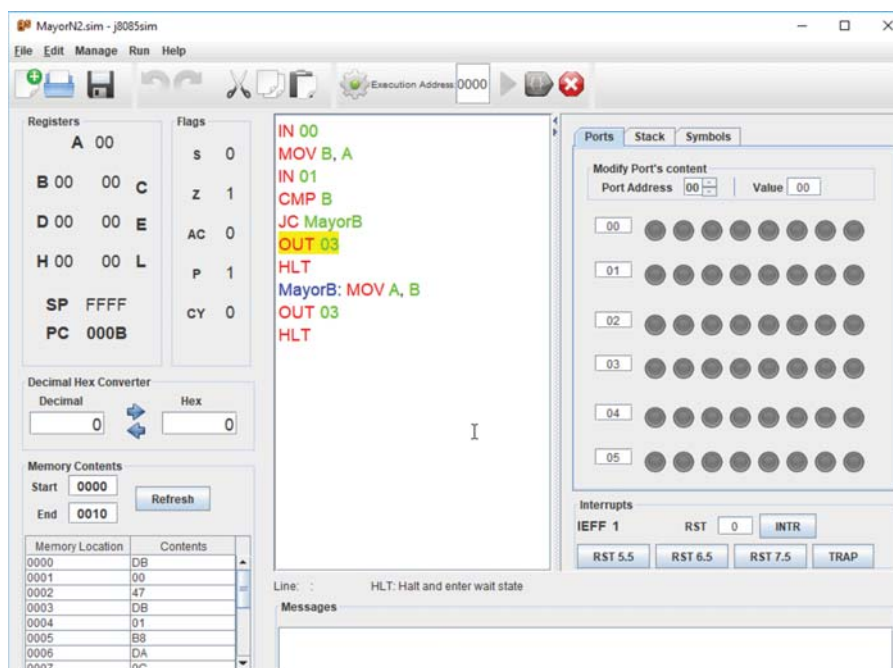


Figura 1. Simulador 8085 empleado en las sesiones prácticas.

La realización de prácticas se lleva a cabo mediante un *entrenador digital*<sup>4</sup>, un software que permite al estudiante disponer estos componentes básicos en una placa virtual y conectarlos entre sí para crear otros de mayor complejidad.

### 2.3. Arquitectura de Computadores

Sobre la base del conocimiento adquirido durante el primer curso de la titulación, en el primer cuatrimestre del segundo curso los estudiantes cursan esta asignatura que, como las dos anteriores, cuenta con una carga docente de 6 ECTS. Al igual que Fundamentos de Arquitectura de Computadores, esta también es impartida por el área de Arquitectura y Tecnología de Computadores.

El núcleo del temario teórico de esta asignatura es el paralelismo en la arquitectura de microprocesadores, desde la segmentación de cauce y los procesadores superescalares a los de tipo VLIW (*Very Large Instruction Word*) y vectoriales [4]. Los estudiantes aprenden a detectar los diferentes tipos de conflictos que aparecen durante la ejecución de código a nivel de máquina: riesgos de datos, de control y estructurales, así como los mecanismos diseñados a lo largo del tiempo para superar dichos problemas: caminos de *bypass*, predictores de saltos, emisión desordenada de instrucciones, etc.

Al igual que ocurre con las dos asignaturas previas, en esta las prácticas también se realizan mediante software de emulación. Concretamente se emplea un simulador de

<sup>4</sup> <http://www4.ujaen.es/aabarca/descargas.htm>

procesador con arquitectura DLX y segmentación de cauce y otro de tipo superescalar (véase la Figura 2) con el mismo conjunto de instrucciones. Los estudiantes aprenden a usar un conjunto de instrucciones ensamblador distinto al empleado en Fundamentos de arquitectura de computadores, más avanzado, así como a identificar las técnicas que hacen posible optimizar su ejecución: reordenación de código, desenrollado de bucles, etc.

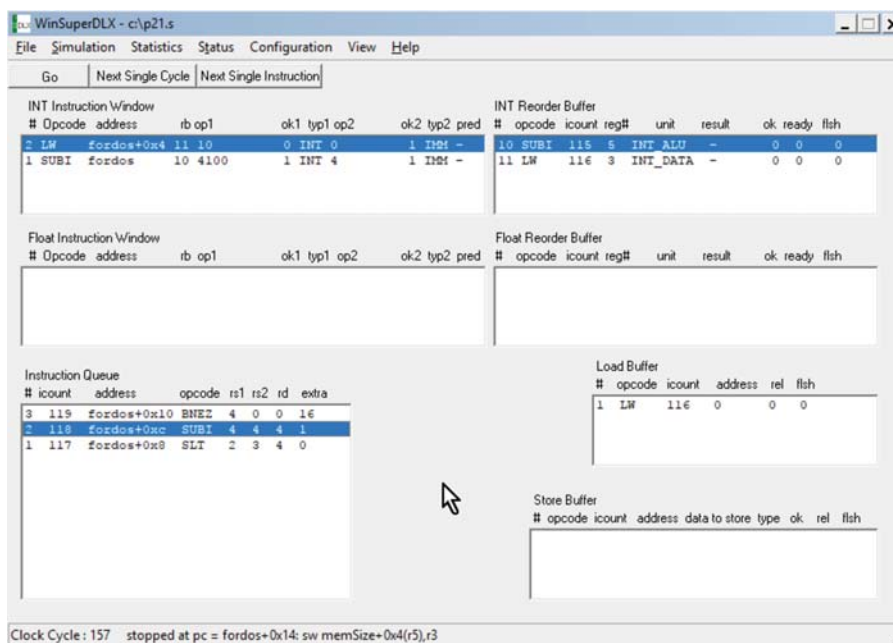


Figura 2. Simulador de procesador superescalar WinSuperDLX.

#### 2.4. Microprocesadores y Microcontroladores

A diferencia de las previas, asignaturas todas ellas obligatorias, esta es de carácter optativo y se imparte en el primer cuatrimestre del cuarto curso de la titulación, estando a cargo del área Tecnología electrónica. Como el resto, tiene asignada una carga docente de 6 ECTS.

La mayor parte del programa teórico de esta asignatura se centra en el estudio de la arquitectura de una familia de microcontroladores, incluyendo la estructura de su memoria, puertos de E/S y funcionamiento de las interrupciones. Se compara esta arquitectura con la de otros circuitos de control, como los microprocesadores, ASIC (*Application Specific Integrated Circuit*) y PLD (*Programmable Logic Device*). Tal y como se apunta en la propia guía docente, el objetivo principal es complementar los estudios abordados en las tres asignaturas citadas en los apartados anteriores.

En cuanto al trabajo práctico, este se centra en el desarrollo de soluciones simples con un microcontrolador, concretamente unas placas de tipo Arduino, e incluyen la programación de sus puertos de E/S y el trabajo con distintos tipos de periféricos, todo ello orientado a aplicaciones de adquisición de datos y de control.

### 3. Metodología actual y potenciales acciones complementarias

A partir de la descripción dada en la sección previa, que aunque corresponde a la Universidad de Jaén nos consta que es similar en cuanto a estructura y metodología a la seguida en otras universidades, pueden derivarse algunos hechos de interés:

- Los contenidos teóricos de Fundamentos de Arquitectura de Computadores y Arquitectura de Computadores se centran en el estudio de las diferentes partes de un computador, esencialmente las partes de la CPU, memoria y E/S. En contrapartida las prácticas consisten básicamente en aprender a programar en ensamblador, primero, y a optimizar dicho código usando algunas técnicas comunes, después.
- La realización de las prácticas se desarrolla casi en todos los casos mediante aplicaciones que emulan/simulan una determinada arquitectura, como la del procesador 8085 o DLX<sup>5</sup>. En general son aplicaciones considerablemente antiguas, con muchas limitaciones y que plantean a los estudiantes problemas para su uso.
- El salto desde la asignatura Electrónica Digital, en la que se conocen los componentes electrónicos más básicos, al diseño de las partes de una CPU (registros, ALU, etc.) usando dichos elementos no se encuentra definido en los contenidos posteriores, especialmente desde una perspectiva práctica que permita a los estudiantes establecer una conexión clara entre las técnicas empleadas en ambas partes.

Con el objetivo de mejorar tanto la perspectiva global sobre la materia como la experiencia adquirida por los estudiantes, planteamos una serie de potenciales cambios en la actual metodología, como por ejemplo:

- Las dificultades que plantea el uso de los actuales emuladores/simuladores representan un obstáculo que requiere un esfuerzo adicional por parte del estudiante. La sustitución de los mismos por herramientas de uso más accesible reduciría dicho esfuerzo, que podría en su lugar dedicarse a los objetivos siguientes.
- Aprender a usar el conjunto de instrucciones de una determinada arquitectura, programando en ensamblador, es un paso fundamental en el aprendizaje. Estas prácticas, no obstante, podrían complementarse incluyendo el propio diseño del computador a partir de componentes básicos, dejando en manos del estudiante el diseño una ALU básica, la correspondiente unidad de control, el conjunto de registros, buses de comunicaciones, etc.
- La configuración de procesadores avanzados como los estudiados en Arquitectura de Computadores, con cauces segmentados o arquitectura superescalar, es prácticamente la única intervención del estudiante en su funcionamiento, estableciendo por ejemplo el número de unidades funcionales, tamaño de la ventana de instrucciones, etc., todo ello mediante una interfaz gráfica de usuario. Una alternativa a este

<sup>5</sup> Una arquitectura definida con fines didácticos basada en la del procesador MIPS.

método de aprendizaje podría ser la introducción de un lenguaje de descripción de hardware, como VHDL o Verilog [5], que ofreciese mucha más flexibilidad a la hora de diseñar cada aspecto del microprocesador sobre el que se quiere trabajar.

La introducción de todos o parte de estos cambios, así como de otros en el mismo sentido, precisan de la selección de una herramienta de trabajo adecuada para los estudiantes. Podría plantearse el desarrollo de nuevo software de emulación/simulación, sin las limitaciones y problemas que plantea el actual. Otra alternativa, más cercana a la realidad en cuanto que se emplearía hardware real en lugar de software, y previsiblemente de mayor utilidad para los estudiantes, se basaría en el uso de hardware reconfigurable.

## 4. Hardware reconfigurable y FPGAs

Los circuitos integrados tradicionales, con el microprocesador al frente de una familia de la que también forman parte microcontroladores, ASIC, DSP (*Digital Signal Processing*) y otros, se caracterizan por contar con una microarquitectura hardware fija. La función de este tipo de circuitos se establece durante su diseño y queda fijada de forma permanente durante el proceso de fabricación. Algunos circuitos integrados (CI), como es el caso del microprocesador, son de propósito general. Por ello incorporan la capacidad de ser programados, ejecutando funciones definidas por software. Otros, como los ASIC y DSP, se diseñan desde un principio para satisfacer una función específica, siendo menos flexibles que un microprocesador pero ofreciendo a cambio un mejor rendimiento, mayor eficiencia energética, entre otras ventajas.

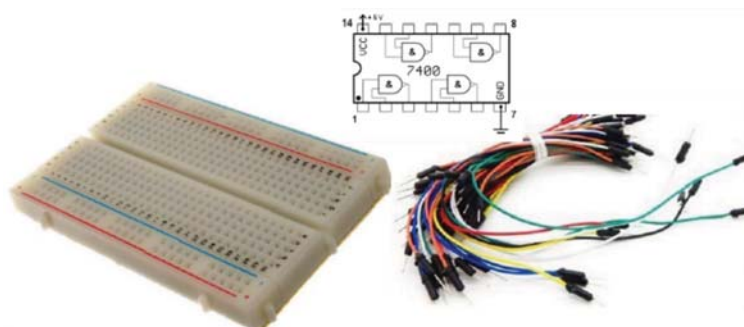
Paralelamente a los anteriores, durante las últimas tres décadas también se han desarrollado CI configurables por el usuario. Inicialmente estas soluciones, plasmadas en circuitos como las PAL (*Programmable Array Logic*) y PLA (*Programmable Logic Array*) [6], se basaban en tecnologías de fusibles, por lo que únicamente podían configurarse una vez. La disponibilidad de nuevos tipos de memorias, incluyendo las estáticas y las no volátiles, han hecho posible fabricar CI reconfigurables.

### 4.1. Introducción al hardware reconfigurable

Los orígenes del hardware reconfigurable se remontan más de medio siglo atrás, concretamente a 1959, y tienen lugar en la UCLA [7] a iniciativa del matemático John Pasta. Este plantea un desafío: crear un ordenador en el que se combine una parte de hardware fijo con otra variable configurable por el operador, naciendo así el concepto de arquitectura F+V. La *configurabilidad* se conseguía merced a unos módulos que actuaban como componentes básicos y que el operario podía instalar e interconectar en una placa base que, posteriormente, se instalaba en el ordenador en cuestión.

El diseño de circuitos digitales empleando integrados discretos, con unas pocas puertas lógicas, dispuestos sobre una placa de conexionado es algo del pasado, aunque sigue siendo una herramienta didáctica muy útil (véase Figura 3) para la comprensión de los principios más básicos del funcionamiento de un computador. Actualmente es habitual el uso de estos recursos sustituyendo los CI discretos por microcontroladores como la

Raspberry y Arduino, combinándolos con diversos sensores, leds, etc., o bien recurriendo a entrenadores digitales: aplicaciones software que simulan el funcionamiento del hardware.



**Figura 3.** Placa de prototipado, circuito integrado de la familia 7400 con cuatro puertas AND y cables para establecer conexiones.

Durante la década de los 70 aparecen los primeros circuitos integrados con arquitectura configurable, las anteriormente citadas PAL y PLA. Estos ofrecen matrices de puertas lógicas con conexiones sin establecer, de forma que el CI puede realizar distintas funciones según el proyecto en el que vayan a utilizarse. Las conexiones no se cablean, como ocurría con los integrados discretos, sino que se fijan mediante un proceso que consiste en *quemar* unos fusibles internos [8] usando para ello un hardware a medida, como podía ser una placa conectable a un PC.

En la década de los 80 el fabricante Lattice introduce los GAL (*Generic Array Logic*), análogos a los PAL pero con una tecnología de establecimiento de las conexiones basada en memoria EEPROM [9]. A diferencia de PAL y PLA, que eran CI que únicamente podían configurarse una vez, los GAL podían ser reconfigurados tantas veces como se necesitase, aportando mucha más flexibilidad.

Actualmente tanto GAL como PLA/PAL están en desuso, merced a los avances experimentados en tecnologías de hardware reconfigurable. En su lugar se recurre a integrados de tipo CPLD (*Complex Programmable Logic Device*) y FPGA. El encapsulado de estos deja atrás el clásico DIP (*Dual In-line Package*) con unas pocas decenas de pines, mientras que la tecnología de programación pasa a estar basada en memoria tipo Flash o bien SRAM (*Static RAM*). La mayor ventaja respecto a los CI ya mencionados estriba en la mayor densidad de elementos básicos reconfigurables, pasando de pocas decenas de puertas lógicas a miles o millones de ellas. Además, en general no es necesario disponer de hardware especializado para establecer la configuración de estos CI, esta se obtiene directamente de la memoria del ordenador o de la placa de entrenamiento correspondiente.



## 4.2. Estado actual de la tecnología FPGA

A pesar de ser una tecnología disponible desde hace más de 30 años, el primer circuito FPGA fue comercializado por Xilinx en 1985, la difusión de este tipo de CI, haciendo dicho producto accesible al usuario en general, es un hecho relativamente reciente. A ello han contribuido múltiples factores, entre los cuales habría que destacar la adquisición de Altera, el segundo mayor fabricante de FPGA mundial, por parte de Intel en 2015. La entrada en escena del gigante de los microprocesadores ha incrementado el nivel de competencia, provocando el lanzamiento de productos basados en FPGA con mayores capacidades y a menor precio.

En 2016 Intel presentó sus procesadores para servidores Xeon con circuitería FPGA integrada<sup>6</sup>, destinados al diseño de centros de procesamiento de datos que, como los creados por Microsoft para sus servicios Azure y Bing a partir de su proyecto Catapult [10], combinan un microprocesador clásico con la posibilidad de implementar en hardware las partes más críticas para el rendimiento y consumo del sistema. En el extremo opuesto, en cuanto a nivel de prestaciones se refiere, también encontramos la familia de procesadores Atom E600C<sup>7</sup> con FPGA integrada, producto dirigido a sistemas de tipo IoT (*Internet of Things*) y empujados.

Por otra parte Xilinx, el primer fabricante mundial de dispositivos FPGA, también cuenta con productos que aúnan en un mismo encapsulado hardware reconfigurable con núcleos de procesamiento de tipo ARM<sup>8</sup>. Ambas gamas de producto, la de Intel y la Xilinx, hacen posible la implementación de soluciones híbridas hardware-software. Esta es una tendencia que está llegando incluso a los servicios de computación en la nube, como los ofrecidos por AWS (*Amazon Web Services*)<sup>9</sup>, en los que el desarrollador cuenta con instancias de ejecución que combinan los servidores tradicionales con hardware reconfigurable de tipo FPGA. Todo ello apunta a un importante crecimiento en los próximos años de la demanda de profesionales formados en técnicas de hardware reconfigurable, por lo que su incorporación en la enseñanza de las asignaturas relacionadas con arquitectura de computadores conllevaría una mejora en la formación del estudiante de cara a su futuro.

Esta evolución del uso de hardware reconfigurable, cada vez más habitual para distintos tipos de soluciones, queda recogida en la actualización del Computer Engineering Curricula de la ACM/IEEE [11] a través de un incremento en las horas de dedicación a su estudio respecto a versiones previas de este documento, en detrimento de otras técnicas heredadas como VLSI.

## 4.3. Metodología de trabajo con hardware reconfigurable

El procedimiento de trabajo con hardware reconfigurable implica múltiples pasos específicos, estando más cerca del proceso de diseño de circuitos tipo ASIC que del

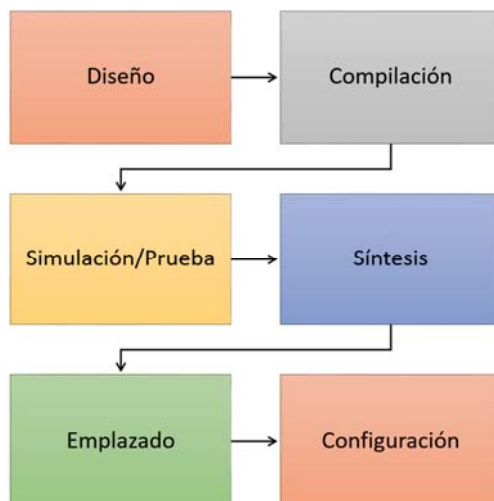
<sup>6</sup> <https://www.nextplatform.com/2016/03/14/intel-marrying-fpga-beefy-broadwell-open-compute-future>.

<sup>7</sup> <http://www.embeddedintel.com/commentary.php?article=2143>.

<sup>8</sup> <https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>

<sup>9</sup> <https://aws.amazon.com/es/blogs/aws/developer-preview-ec2-instances-f1-with-programmable-hardware/>.

desarrollo de software. Los pasos fundamentales son los presentados esquemáticamente en la Figura 4.



**Figura 4.** Pasos habituales en el flujo de trabajo con FPGA.

Se parte con el diseño de la solución que quiere implementarse en el hardware reconfigurable, tarea para la cual se puede, dependiendo de la complejidad del caso, elaborar un esquemático (véase la Figura 5) o bien describir el hardware mediante un lenguaje apropiado<sup>10</sup>.

Tras compilar el diseño se procede normalmente a probarlo, usando para ello una herramienta de simulación. Esta puede ser de bajo nivel, construyendo *waveforms* que representan los cambios en las señales a lo largo del tiempo (véase la Figura 6), o bien estar descrita en forma de *test bench* usando el propio lenguaje de descripción de hardware.

Una vez se ha confirmado el correcto funcionamiento del circuito en el entorno de simulación llega el momento de sintetizar dicho circuito, determinando el hardware de la FPGA que se empleará para implementarlo. Asimismo es necesario emplazar las señales de entrada/salida (véase la Figura 7), estableciendo los pines del integrado al que se conectarán. Los elementos conectados a dichos pines pueden estar predeterminados o no, dependiendo del hardware concreto que se emplee.

El paso final consiste en transferir la configuración generada por la herramienta de diseño/development al hardware, configurando la FPGA. Las placas de aprendizaje suelen contar con conexión USB a fin de facilitar esta tarea, pero al trabajar con FPGA integrada en el encapsulado de un microprocesador, como las mencionadas en la sección previa, el conexionado sería interno.

<sup>10</sup> Los dos estándares en cuando a lenguajes de descripción de hardware son VHDL y Verilog, estando el uso del primero más extendido en Europa y el del segundo en Estados Unidos.

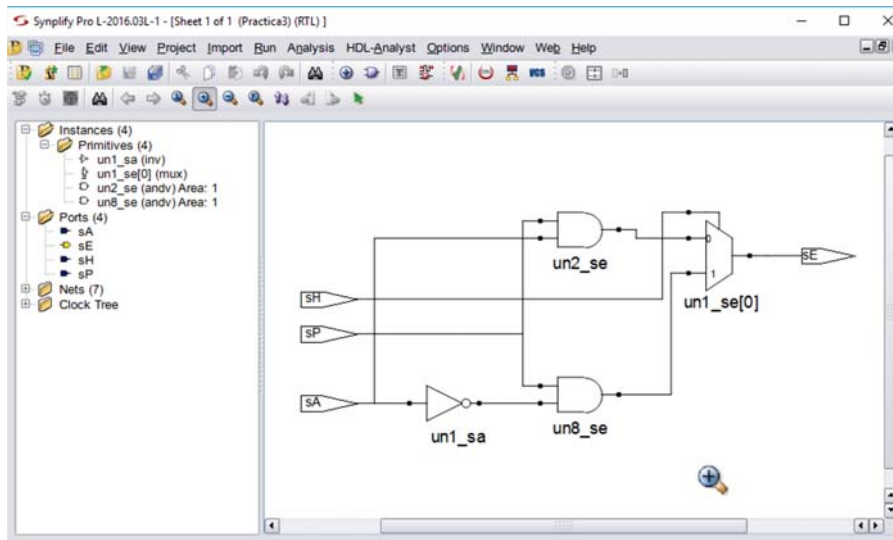


Figura 5. Diseño de un circuito simple a partir de un esquemático.

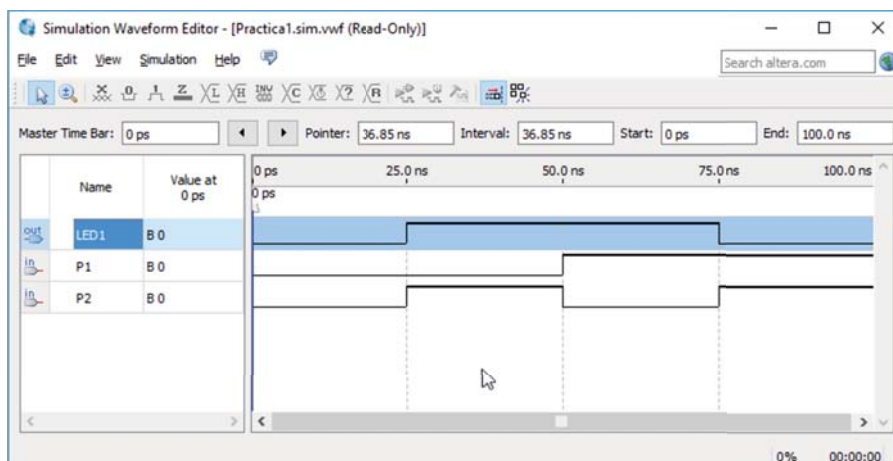


Figura 6. Se prueba el diseño mediante una herramienta de simulación.

## 5. Uso de FPGA para la enseñanza de arquitectura de computadores

En la Sección 3 se planteaban potenciales cambios en la metodología seguida hasta ahora para la enseñanza de arquitectura de computadores, apuntando al final que estas actividades requerían el uso de las herramientas adecuadas. En la presente sección se describe cómo podrían desarrollarse dichas actividades empleando como herramientas hardware reconfigurable, concretamente tarjetas de aprendizaje con circuitería FPGA.

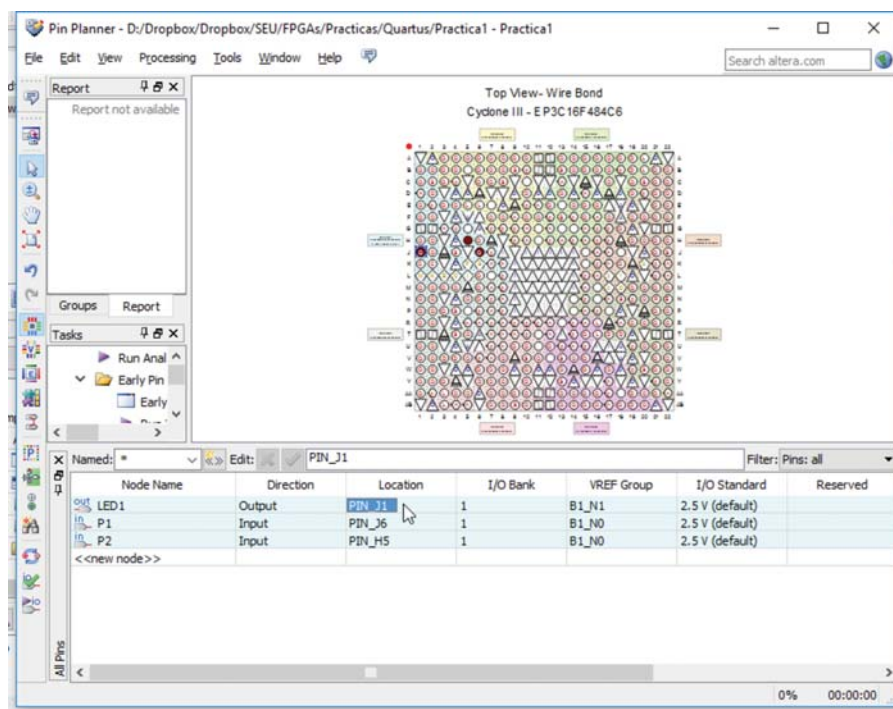


Figura 7. El emplazamiento conecta las señales de E/S del diseño a pines físicos de la FPGA.

Las FPGA actualmente se usan en asignaturas como *Implementación de algoritmos Hardware* [12], perteneciente al Grado en Ingeniería Informática (especialidad en Ingeniería de Computadores) de la Universidad de Granada, o *Programación Hardware*, perteneciente al Grado en Ingeniería Informática de la Universidad de Jaén. Se trata, por tanto, de una tecnología presente en algunas universidades españolas y con la que parte del profesorado ya está familiarizado.

### 5.1. Selección del hardware a emplear

Trabajar con hardware real, en el que la implementación de un cierto microprocesador simple, diseñado para el aprendizaje, opere como lo haría cualquier otro circuito integrado, ofrece al estudiante una visión alternativa, posiblemente complementaria, a la de los emuladores software. No obstante, la adquisición de dicho hardware implica un determinado coste que será necesario asumir. Este será proporcional al número de unidades que se precise adquirir que, a su vez, dependerá del número de estudiantes matriculados en la asignatura.

Actualmente los tres principales fabricantes de FPGA, Xilinx, Altera/Intel y Lattice, cuentan con productos dedicados al aprendizaje. Son tarjetas con una FPGA básica y algunos puertos de E/S, incluyendo leds, pulsadores, microinterruptores, etc., dependiendo de cada caso. Los precios oscilan desde los 20 euros del producto más básico de

Lattice (véase la Figura 8) hasta más de 100 euros para las placas de entrenamiento avanzadas que cuentan con salida de vídeo, pantallas de siete segmentos y LCD, etc. Las primeras son incluso accesibles para los estudiantes, que podrían adquirirlas como material propio de prácticas a título individual o en pequeños grupos.



**Figura 8.** Tarjeta FPGA para aprendizaje Lattice ICEstick iCE40HX1K.

Como se apuntaba anteriormente, en universidades como la Universidad de Jaén y la Universidad de Granada ya existen asignaturas en las que se usa hardware FPGA, por lo que esos mismos dispositivos podrían también emplearse ocasionalmente en las asignaturas correspondientes a arquitectura de computadores.

## 5.2. Diseño basado en componentes simples

Las herramientas de diseño facilitadas por los diferentes fabricantes de productos FPGA, así como las disponibles a través de licencias de código libre<sup>11</sup>, ofrecen bibliotecas de puertas lógicas y algunos componentes simples, como los codificadores, descodificadores y multiplexores, que pueden ser empleados durante las prácticas de asignaturas como Electrónica Digital.

Partiendo de esos elementos simples es posible crear biestables y, a partir de estos, unidades básicas de memoria de un bit que, a su vez, serían el pilar de la creación de registros con el ancho de palabra que interese. Análogamente, con los demás elementos básicos es posible diseñar máquinas de estados que actúen como unidad de control. Las placas de entrenamiento cuentan con una señal de reloj integrada que es posible conectar a los circuitos diseñados, permitiendo así la creación de circuitos tanto combinacionales como secuenciales.

Para la realización de este tipo de prácticas los estudiantes pueden efectuar sus diseños con esquemáticos, colocando componentes y conectándolos, o bien aprender los fundamentos más básicos de un lenguaje de descripción de hardware como VHDL. Con independencia de ello, los diseños una vez validados se sintetizan y transfieren a la placa FPGA, ejecutándose en hardware real en lugar de ser una simulación.

<sup>11</sup> <http://fpgawars.github.io/>

### 5.3. Diseño basado en módulos reutilizables

Tanto VHDL como Verilog, los dos lenguajes de descripción de hardware de uso más habitual, contemplan la posibilidad de crear bibliotecas de módulos con componentes prediseñados. Se pueden tener módulos con elementos básicos, como registros, sumadores, unidad de control, etc., de tal forma que el estudiante únicamente tiene que aprender a diseñar un microprocesador/computador “instanciando” esos componentes con la configuración que se demande. Pueden ajustar el tamaño de los registros, ancho de los buses de datos y direcciones, etc. Asimismo sería posible incluir elementos más complejos: unidades funcionales de distintos tipos, predictor de saltos, etc. De esta forma el aprendizaje teórico sobre arquitectura de computadores se reforzaría con una componente práctica, en la que es el propio estudiante el que puede diseñar su microprocesador.

Una alternativa al enfoque anterior, basado en el aprendizaje a través de la construcción de soluciones desde abajo (componentes básicos) hacia arriba (sistema completo), consistiría en facilitar al estudiante el diseño completo de un sistema, permitiéndole realizar ajustes en la configuración de este: cambiar el tamaño de la ventana de instrucciones, número de unidades funcionales de cada tipo, memoria del predictor de saltos, etc. Existen diseños libres de multitud de microprocesadores escritos en VHDL/Verilog, preparados para su uso o adaptación a las necesidades de cada asignatura, desde micros clásicos de 8 bits [13] hasta microprocesadores multinúcleo [14], pasando por micros teóricos como DLX [15]. Incluso cabe la posibilidad de sintetizar en FPGA emuladores de procesadores didácticos como WepSIM [16].

Al igual que para el aprendizaje de electrónica digital, el procedimiento de trabajo podría estar basado en el diseño esquemático, colocando y conectando bloques para formar el computador, o bien en una descripción formal en un lenguaje como VHDL. Ambas son herramientas que servirían al estudiante no solo para cursar posteriormente otras asignaturas, como las antes citadas Implementación de algoritmos hardware y Programación hardware, sino también como experiencia útil para un hipotético futuro laboral en el diseño de soluciones basadas en hardware reconfigurable.

## 6. Conclusiones

La formación de los estudiantes del Grado en Ingeniería Informática en el área de arquitectura de computadores es fundamental. En la actualidad, tal y como se ha descrito, las clases teóricas suelen dedicarse al estudio de la arquitectura y las prácticas a la programación en ensamblador. La estructura del computador descrito en teoría ya se encuentra implementada, habitualmente en un emulador software, y el estudiante se limita a utilizarla.

Nuestra propuesta tiene el objetivo de complementar la metodología seguida actualmente, ampliando los conocimientos adquiridos por los estudiantes. Para ello se recomienda la introducción del hardware reconfigurable, ya en uso en otras asignaturas que el estudiante puede cursar posteriormente de forma optativa. Parte de los recursos necesarios, así como la experiencia por parte del profesorado en este campo, ya están disponibles.

Desde nuestra perspectiva los beneficios para los estudiantes son claros. Por una parte, el diseño de la arquitectura de un computador simple, a través de esquemáticos y/o lenguajes de descripción de hardware, reforzará la comprensión que de esta materia adquieren en las clases teóricas. Por otra, adquirirán experiencia en un campo y con unas herramientas, la circuitería FPGA y el desarrollo de soluciones con VHDL/Verilog, cuya demanda de profesionales es creciente.

## Referencias

1. M. Rivas Pérez, M. Domínguez Morales, F. Gómez Rodríguez, A. Linares Barranco, G. Jiménez Moreno, A. Civit Balcells, Diseño e implementación de un simulador software basado en el procesador MIPS32, *Enseñanza y Aprendizaje de Ingeniería de Computadores* (5) (2015) 79–104.  
URL <http://hdl.handle.net/10481/36571>
2. A. N. de Evaluación de la Calidad y Acreditación, Libro blanco del título Grado en Ingeniería Informática, 2004.  
URL [http://www.aneca.es/var/media/150388/libroblanco\\_jun05\\_informatica.pdf](http://www.aneca.es/var/media/150388/libroblanco_jun05_informatica.pdf)
3. ACM/IEEE, Computer Engineering Curricula, 2016.  
URL <http://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf>
4. J. Ortega Lopera, M. Anguita López, A. Prieto Espinosa, *Arquitectura de computadores*, Thomson, 2005.
5. D. J. Smith, VHDL & Verilog Compared & Contrasted - Plus Modeled Example Written in VHDL, Verilog and C, in: *DAC*, 1996.
6. T. L. Floyd, *Digital Fundamentals*, 10/e, Pearson Education, 2011.
7. G. Estrin, Reconfigurable computer origins: the ucla fixed-plus-variable (f+v) structure computer, *IEEE Annals of the History of Computing* 24 (4) (2002) 3–9. doi:10.1109/MAHC.2002.1114865.
8. J. M. Birkner, PAL, programmable array logic, handbook, *Monolithic Memories*, 1983.
9. D. Gembris, Generic Array Logic (GAL), *Elektor Electronics* (4) (1992) 24–25.
10. A. Putnam, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J.-Y. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. M. Caulfield, A. Smith, J. Thong, P. Y. Xiao, D. Burger, E. S. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. P. Gopal, A reconfigurable fabric for accelerating large-scale datacenter services, *Communications of the ACM* 59 (11) (2016) 114–122. doi:10.1145/2996868.
11. E. Durant, J. Impagliazzo, S. Conry, R. Reese, H. Lam, V. Nelson, J. Hughes, W. Liu, J. Lu, A. McGettrick, CE2016: Updated computer engineering curriculum guidelines, in: *2015 IEEE Frontiers in Education Conference (FIE)*, IEEE, 2015, pp. 1–2. doi:10.1109/FIE.2015.7344157.
12. J. Díaz Alonso, Diseño e implementación de un simulador software basado en el procesador MIPS32, *Implementación de algoritmos en hardware: de la puerta NAND al bucle FOR* (3) (2013) 95–105.  
URL <http://hdl.handle.net/10481/26401>
13. E. Ayeh, K. Agbedanu, Y. Morita, O. Adamo, P. Guturu, FPGA Implementation of an 8-bit Simple Processor, in: *2008 IEEE Region 5 Conference*, IEEE, 2008, pp. 1–5. doi:10.1109/TPSD.2008.4562743.
14. W. Wójcik, J. Długopolski, FPGA-based multi-code processor, *Computer Science @BULLET* 14 (3). doi:10.7494/csci.2013.14.3.459.
15. P. J. Ashenden, P. J. Ashenden, 15 – Case Study: The DLX Computer System, in: *The Designer's Guide to VHDL*, 2002, pp. 373–458. doi:10.1016/B978-155860674-6/50017-4.

16. A. Calderón Mateos, F. García Carballeira, J. Prieto Cepeda, WepSIM: Simulador modular e interactivo de un procesador elemental para facilitar una visión integrada de la microprogramación y la programación en ensamblador, *Enseñanza y Aprendizaje de Ingeniería de Computadores* (6) (2016) 35-53.  
URL <http://hdl.handle.net/10481/41909>