

# Migración de prácticas a una plataforma de hardware reconfigurable en la asignatura de Tecnología y Organización de Computadores.

Jorge Manuel Machado Cano<sup>1</sup>, Eduardo Ros Vidal<sup>1</sup>, Manuel Rodríguez-Álvarez<sup>1</sup>,  
Jesús González Peñalver<sup>1</sup>, Pedro Jesús Martín Smith<sup>1</sup>

**Resumen**—En este trabajo se presenta la asignatura **Tecnología y Organización de Computadores correspondiente al primer curso del Grado en Ingeniería en Informática de la Universidad de Granada. Asimismo se propone una posible migración de la realización de las prácticas de dicha asignatura utilizando la placa IceZUM Alhambra y se explican ejemplos sobre la construcción de modelos lógicos con la misma.**

**Palabras clave**—Tecnología y organización de Computadores, FPGA, IceZUM, hardware abierto, modelos lógicos.

## I. INTRODUCCIÓN

LA asignatura de Tecnología y Organización de Computadores (TOC) se imparte en el segundo cuatrimestre del primer curso de la titulación de Grado en Ingeniería Informática de la Universidad de Granada [1]. Es una asignatura de las denominadas “básicas” en dicho plan de estudios. Los objetivos de dicha asignatura son:

- Conocer la organización y componentes básicos de un computador.
- Identificar los factores que determinan las prestaciones básicas de un computador.
- Comprender la conveniencia de describir un computador en diferentes niveles de abstracción para facilitar su comprensión, su diseño y su utilización.
- Conocer las distintas formas básicas de representación de la información en un computador.
- Aplicar técnicas básicas de análisis y diseño de sistemas digitales.
- Comprender las diferentes formas de representar el comportamiento de un sistema digital.
- Estimar las prestaciones de sistemas combinacionales y secuenciales.
- Comprender el funcionamiento de los diferentes bloques combinacionales y secuenciales básicos que forman parte de la mayoría de los sistemas digitales, e identificar claramente la función que realizan.
- Conocer la organización de los sistemas diseñados en el nivel de transferencia de registros, incluyendo la organización y diseño de un computador sencillo, comprendiendo la misión del camino de datos y de la unidad de control, y su interacción.
- Deducir las operaciones de transferencia entre registros que puedan realizarse en un camino de datos dado.

Para que el estudiante consiga cubrir los objetivos de la asignatura, ésta se ha estructurado en un temario que comprende impartir teoría al grupo grande de clase y una serie de seminarios y prácticas para grupos pequeños. Éstos son:

### A. TEMARIO TEÓRICO:

- Tema 1. Introducción. Conceptos básicos.
- Tema 2. Unidades funcionales de un computador
- Tema 3. Análisis y diseño de sistemas combinacionales.
- Tema 4. Análisis y diseño de sistemas secuenciales.
- Tema 5. Sistemas en el nivel transferencia entre registros (RTL)

### B. TEMARIO PRÁCTICO:

- 1) *Seminarios:*
  - S1. Sistemas de numeración usuales en Informática.
  - S2. Representación de información multimedia.
  - S3. Álgebra de Boole y funciones de conmutación.
  - S4. Minimización de funciones de conmutación.
  - S5. Introducción al manejo de un simulador y de un entrenador lógico.
- 2) *Prácticas de laboratorio:*
  - P1. Análisis y diseño de circuitos combinacionales con puertas lógicas.
  - P2. Diseño de circuitos aritméticos.
  - P3. Diseño de una unidad aritmético-lógica sencilla.
  - P4. Funcionamiento de codificadores/decodificadores y multiplexores/demultiplexores.
  - P5. Funcionamiento de biestables básicos y de registros.
  - P6. Implementación y funcionamiento de contadores y generadores de secuencias.
  - P7. Análisis y diseño de sistemas secuenciales síncronos.
  - P8. Descripción a nivel RT de un computador sencillo.

El temario detallado se puede consultar en la Guía Docente de la asignatura [2].

La ubicación de la asignatura Tecnología y Organización de Computadores y su relación con otras asignaturas del Plan de Estudios del Grado de Ingeniería en Informática de la Universidad de Granada se muestra en la Figura 1.

<sup>1</sup>Departamento de Arquitectura y Tecnología de Computadores. E.T.S. de Ingenierías Informática y de Telecomunicación. Universidad de Granada, email: {eros, manolo, jesusgonzalez, pmartin}@ugr.es.

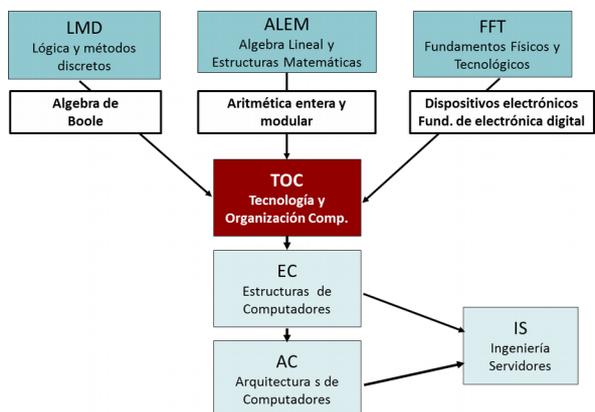


Fig. 1. Ubicación de la asignatura dentro del plan de estudios y su relación con otras asignaturas.

Las asignaturas Álgebra Lineal y Estructuras Matemáticas y Fundamentos Físicos y Tecnológicos se imparten en primer cuatrimestre de primer curso. La asignatura de Lógica y Métodos Discretos, se imparte en segundo cuatrimestre, simultáneamente con Tecnología y Organización de Computadores. Las asignaturas de Estructura y Arquitectura de Computadores se imparten en segundo curso en primer y segundo cuatrimestre, respectivamente.

## II. DESCRIPCIÓN DE LAS PRÁCTICAS DE LA ASIGNATURA

Para el desarrollo e implementación de la mayor parte de las prácticas de la asignatura en la actualidad se utiliza un software de simulación de circuitos lógicos denominado *Logic Works* [3] desarrollado por la empresa *Capilano Computing Ltd*. Las nociones básicas de funcionamiento de dicho software se explican al estudiante en el Seminario 5. Además de este software se utiliza un dispositivo entrenador lógico en el que los estudiantes pueden montar y cablear circuitos lógicos. El software *Logic Works* aunque no es un software libre en el sentido de disponer el usuario del código fuente del mismo en abierto, al ser un software de simulación lógica con fines educativos es de bajo coste (*shareware*) viniendo incluido como material suplementario (*Logic Works v4*) en varios libros de los que hay disponibles en la Biblioteca de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada. Otro inconveniente de este simulador es que sólo funciona en plataformas *Windows*, por lo que los estudiantes que usen otros sistemas operativos tendrían que utilizarlo creando una máquina virtual *Windows* o emulador como el *Wine*. No obstante, el software de simulación *Logic Works* es muy intuitivo para su utilización, dispone de un gran número de librerías con componentes, tanto individuales como familias lógicas y permite, de una forma muy rápida crear el esquemático del circuito y realizar la simulación lógica de sistemas tanto simples como complejos (incluyendo la posibilidad de realizar diseños modulares y escalables, así que como la simulación y validación de funcionalidad mediante ficheros de estímulos y cronogramas de salida). De hecho, para la práctica 8 de la asignatura se realiza, utilizando este software, la

simulación de un computador sencillo denominado CS1 [4] que permite la ejecución de pequeños programas.

No obstante lo anterior, una posible mejora que se podría plantear para las prácticas de la asignatura sería poner al estudiante en contacto con una plataforma física programable y de *hardware* reconfigurable en la que desarrollar y testear sus diseños. Por parte de la dirección del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada y de los profesores implicados en la asignatura de Tecnología y Organización de Computadores se ha planteado barajar varias posibilidades de utilización de plataformas reconfigurables para el desarrollo de las prácticas. Una de las plataformas que se ha estudiado ha sido la denominada *IceZUM Alhambra* [5] y su herramienta de desarrollo *IceStudio*.

## III. MOTIVACIÓN DE LA INCLUSIÓN DE SISTEMAS RECONFIGURABLES DE BAJO COSTE.

Hasta relativamente hace poco tiempo, las placas de desarrollo basadas en FPGA (*Field Programmable Gate Array*) tenían un alto coste. Esto hacía relativamente inviable que una persona particular pudiera desarrollar aplicaciones usando este tipo de dispositivos. Además, las FPGAs existentes debían ser programadas con herramientas de desarrollo complejas y de coste elevado. Afortunadamente, esta situación está cambiando y cada vez existen más dispositivos de desarrollo de bajo coste, entre ellos los que están basados en FPGA. Un ejemplo es la placa *IceZUM Alhambra* (Figura 2) y su herramienta de desarrollo *IceStudio*.



Fig. 2. Placa *IceZUM Alhambra* versión 1.1 (<https://github.com/FPGAwards/icezum/wiki>).

La utilización de estos dispositivos de bajo coste permite acercar el *hardware* reconfigurable al usuario y tienen una gran utilidad a nivel didáctico, pues permiten a quien las usa volcar directamente sus diseños en una plataforma física y ver cómo pueden funcionar en ella. Dichas plataformas se podrían utilizar con fines didácticos y así poder acercar más el mundo del *hardware* reconfigurable y en general los conceptos de tecnología de sistemas digitales, a los estudiantes de primer curso de la asignatura de Tecnología y Organización de Computadores. Esto debe facilitar el aprendizaje de la base tecnológica de los sistemas

digitales y metodologías de diseño y análisis de sistemas combinatoriales y secuenciales (mediante implementaciones y evaluaciones sencillas en una FPGA). Además, el uso de este tipo de sistemas, motiva al estudiante al desarrollo de otro tipo de aplicaciones que se pueden salir del marco de las prácticas de la asignatura, como se verá más adelante.

Además, *IceZUM* es placa basada en FPGA diseñada con los mismos patrones que *Arduino*, por lo que se puede utilizar para su ampliación todos los “escudos” (*shields*) que existen en el mercado para *Arduino* y es un dispositivo de los denominados de *Open Hardware*. Por otra parte, *IceStudio* es el Entorno de Desarrollo Integrado (IDE) que permite programar la FPGA. Aunque se podrían utilizar lenguajes de descripción *hardware*, realmente no es necesario para realizar muchos diseños, entre los que se encuentran los necesarios para cubrir las competencias de la asignatura de Tecnología y Organización de Computadores. Sin embargo, existen componentes como los biestables, que se necesitarían implementar explícitamente a partir de puertas lógicas (ya que no existen en las bibliotecas actuales del entorno de desarrollo) para que los estudiantes pudieran realizar las prácticas correspondientes a los sistemas secuenciales.

#### IV. DESCRIPCIÓN DEL SISTEMA ICEZUM ALHAMBRA

El *hardware* de la placa *IceZUM Alhambra* cuenta con los siguientes dispositivos:

- **FPGA:** *Lattice iCE40HX1K-TQ144* [6]. En esta FPGA hay 1280 Celdas lógicas, formadas cada una de ellas por una *Look-up-table* (LUT) y un *flip-flop*, 64 Kbits de memoria RAM y un PLL. Esta es una FPGA realmente pequeña y de bajo coste, con recursos lógicos reducidos pero suficientes para la realización de sistemas de nivel de complejidad reducido.
- **RELOJ:** Reloj MEMS a 12 MHz.
- **INTERRUPTOR:** Cuenta con un pequeño interruptor para desactivar los pines.
- **REGULADOR DE VOLTAJE:** Permite alimentar la placa con un rango de 6 a 17 voltios.
- **CORRIENTE DE ENTRADA:** Permite hasta una corriente de 3 amperios, útil para alimentar actuadores como, por ejemplo, motores de robots.
- **PINES:** Cuenta con 20 pines de Entrada/Salida a 5V y 10 a 3.3V.
- **INTERFAZ:** Programación mediante un USB micro-B, FTDI 2232H para programación e interfaz UART con un PC.
- **LEDS:** La placa cuenta con 8 leds que el usuario puede utilizar en sus programas.
- **PULSADORES:** Dispone de dos pulsadores utilizables por el usuario.
- **ANALÓGICO:** Cuenta con cuatro entradas analógicas conectadas a un convertidor analógico/digital ADC QFN16, con una precisión de 12 bits, conectado a la FPGA con un bus I2C propio.
- **PROTECCIÓN:** La placa cuenta con protección contra cortocircuitos y polaridad inversa.

Además de todo esto, el fabricante proporciona datos sobre la placa, tales como el *pinout*, el PCF (fichero donde aparecen los pines restringidos de la placa y su uso), la hoja de características técnicas de la placa (*datasheet*), o reglas de diseño de la placa.

Por otra parte, el entorno de desarrollo que suministra el fabricante de la placa es el denominado *IceStudio* [7] Cabe destacar de él su máxima simplicidad, ofreciendo una interfaz que permite instalar todas las herramientas necesarias, tanto para la síntesis de HDL, como los *drivers* FTDI (*Future Technology Devices International*, circuito USB 2.0 a UART) necesarios para programar la *IceZUM*.

La misma interfaz ofrece una biblioteca de objetos ya creados, tales como:

**ELEMENTOS BÁSICOS:** Pines de entrada, de salida, comentarios, constantes y módulos con las entradas y salidas programables en *Verilog*.

**BIT:** Se puede añadir un bit tanto a 1 como a 0.

**CONFIGURACIÓN:** En este menú desplegable se puede encontrar configuraciones *pull-up* para los pines y *buffers* tri-estado.

**LÓGICA:** En este menú desplegable se pueden encontrar dispositivos tales como multiplexores, demultiplexores, módulos para controlar visualizadores de 7 segmentos, todo tipo de puertas lógicas y algunos tipos de *flip-flops*.

Este entorno de desarrollo también permite explotar el potencial de los lenguajes de desarrollo *hardware* exportando a *Verilog* la traducción del diseño que se realice, que puede ser modificado manualmente. Otra gran ventaja del entorno es que permite añadir otros diseños realizados anteriormente como módulos a nuevos diseños. Además, el entorno de desarrollo permite añadir nuevas colecciones (o bibliotecas), es decir, empaquetar todos los diseños en una colección para añadirlos a los menús desplegables, con lo que se pueden añadir colecciones de otros diseñadores con todos los módulos ya implementados que se necesiten.

Este IDE está construido en *nodeJS* sobre *Icestorm*, el cual es una secuencia de herramientas creadas mediante el uso de ingeniería inversa para poder generar *bitstreams* compatibles con las FPGAs *Lattice iCE40*.

Una vez generados los diseños con *IceStudio*, se puede proceder a su simulación. Para ello hay que hacer uso de la herramienta *Icarus Verilog* [8], que sintetiza y simula el diseño exportado anteriormente. Antes de esto también hay que exportar el *testbench* (fichero de test) en *IceStudio*, que es otro archivo en *Verilog* necesario para declarar los estímulos de entrada para el diseño. Una vez completado el proceso, se puede analizar el comportamiento del diseño mediante un analizador temporal, como, por ejemplo *GTKWave*. Con este software y el sistema *IceZUM*, se pueden generar los diseños, simularlos, y cargarlos y ejecutarlos en la placa.

## V. DESCRIPCIÓN DE MÓDULOS AUXILIARES DESARROLLADOS

Puesto que la herramienta *IceStudio* se encuentra en una etapa temprana de desarrollo, la biblioteca de componentes que ofrece está bastante limitada, por lo que será necesario desarrollar algunos componentes en *Verilog*. Aunque esto pueda parecer un inconveniente, dado que los estudiantes de primer curso seguramente desconocen los denominados lenguajes de descripción *hardware* (*Hardware Description Language*, HDL), y aprender un lenguaje de este tipo no se puede abordar en un cuatrimestre junto con los contenidos propios de la asignatura, como *IceStudio* permite encapsular diseños para ser usados como módulos en otros diseños más complejos, este problema se puede solventar diseñando previamente los módulos que los estudiantes puedan necesitar para llevar a cabo las prácticas propuestas en la asignatura. De esta forma, los estudiantes sólo tendrán que seleccionar y conectar los módulos necesarios para implementar la solución de cada una de las prácticas.

A continuación, se describen los diferentes módulos que ha sido necesario desarrollar para poder realizar las prácticas propuestas actualmente en la placa *IceZUM*:

- 4-BIT-AND y 3-BIT-AND: Se han diseñado estos módulos (que son fácilmente generables mediante puertas AND de 2 entradas), para simplificar los diseños.
- 4-BIT-OR y 3-BIT-OR: De la misma manera, también se han diseñado estos módulos.
- FLIP-FLOP D: Se ha diseñado este biestable con entrada de reset asíncrona y activo por flanco de subida de la señal de reloj. Su descripción *Verilog* se detalla en la Figura 3.
- FLIP-FLOP JK: Biestable con entrada de reset asíncrona y activo por flanco de subida. Su descripción *Verilog* se detalla en la Figura 4.
- FLIP-FLOP T: Biestable con entrada de reset asíncrona y activo por de subida de la señal de reloj.
- SLOW CLOCK Y SLOWER CLOCK: Se trata de dos divisores de reloj para que los alumnos puedan ver de forma cómoda cómo actúan sus diseños.

```
reg q = 0;

always @(posedge clk_in or posedge R)
begin
    if(R)
        q = 0;
    else
        q = D;
end

assign Q = q;
```

Fig. 3. Código Verilog de un biestable tipo D.

```
reg q = 1'b0;

always @(posedge C or posedge R)
begin
    if(R)
        q <= 1'b0;
    else if(J && K)
        q <= ~q;
    else if (J && !K)
        q <= 1'b1;
    else if(K && !J)
        q <= 1'b0;
end

assign {Q, Qneg} = {q, ~q};
```

Fig. 4. Código Verilog de un biestable tipo JK.

## VI. PROPUESTA DE ADAPTACIÓN DE LAS PRÁCTICAS EN LA NUEVA PLATAFORMA

En total en la asignatura de Tecnología y Organización de Computadores se desarrollan las ocho prácticas descritas en la sección I de Introducción:

- P1 Análisis y diseño de circuitos combinacionales con puertas lógicas.
- P2 Diseño de circuitos aritméticos.
- P3 Diseño de una unidad aritmético-lógica sencilla.
- P4 Funcionamiento de codificadores/decodificadores y multiplexores/demultiplexores.
- P5 Funcionamiento de biestables básicos y de registros.
- P6 Implementación y funcionamiento de contadores y generadores de secuencias.
- P7 Análisis y diseño de sistemas secuenciales síncronos.
- P8 Descripción a nivel RT de un computador sencillo.

Se ha realizado la implementación de las siete primeras prácticas mediante su adaptación al sistema *IceZUM Alhambra*. Una vez realizadas las prácticas en la nueva plataforma se ha comprobado que no se requieren conocimientos adicionales en ningún lenguaje de descripción *hardware*, exceptuando la generación de baterías de pruebas para simular los diseños.

A continuación se detallan las adaptaciones de las prácticas 3 (Diseño de una unidad aritmético-lógica sencilla) y 6, apartado 2, (Diseño de un generador de secuencias) y los circuitos desarrollados para la plataforma *IceZUM*.

1. Práctica 3. Unidad aritmético-lógica de 4 bits: En esta práctica el estudiante realiza el diseño de una Unidad Aritmético Lógica de 4 bits. En las Figuras 5 a 10 se puede observar la estructura modular y jerárquica de la realización de la práctica.

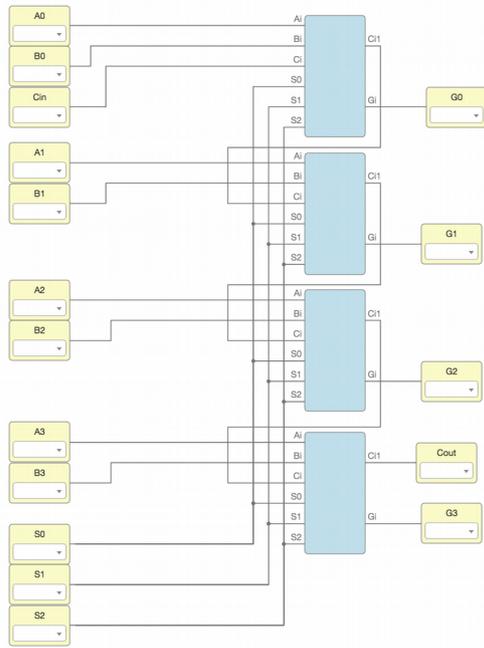


Fig. 5. Primer nivel ALU. Cada uno de los bloques azules representa una ALU de 1 bit.

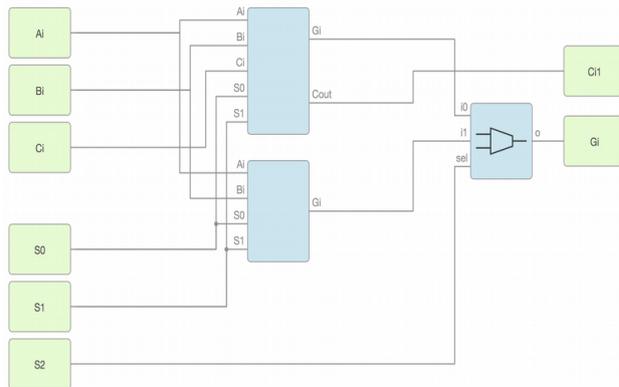


Fig. 6. Segundo nivel ALU. El bloque azul superior es la unidad aritmética, y el inferior la unidad lógica. El multiplexor decide qué circuito se utiliza.

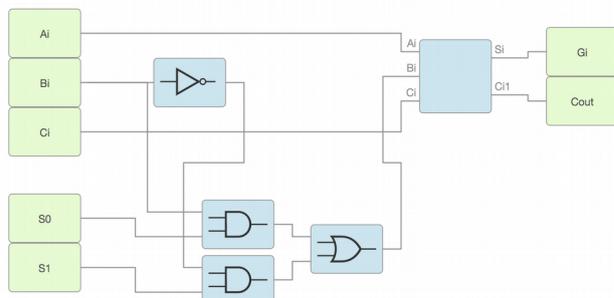


Fig. 7. Tercer nivel ALU. Unidad aritmética. El bloque azul contiene un sumador completo de un bit.

2. Práctica 6. Generador de secuencias: En esta práctica se realiza, entre otros, el diseño de un generador de secuencias que, cíclicamente, genera la secuencia de salidas 0,1,3,0,2. En la Figura 11 se puede observar el diseño de dicho sistema secuencial. En la parte

inferior de dicha figura se utilizan tres biestables de tipo T y la lógica de la función de excitación de los mismos. En la parte superior se implementa la lógica de la función de salida.

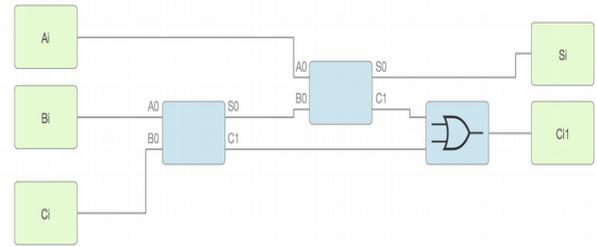


Fig. 8. Cuarto nivel ALU. Sumador completo de un bit. Cada uno de los bloques azules contiene un semisumador de 1 bit.

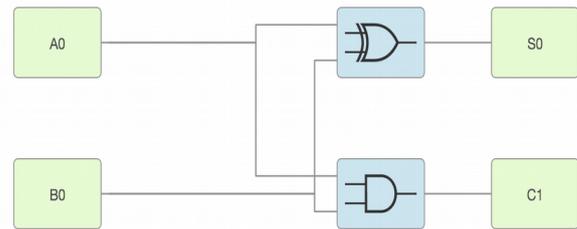


Fig. 9. Quinto nivel ALU. Semisumador de 1 bit a partir de puertas lógicas.

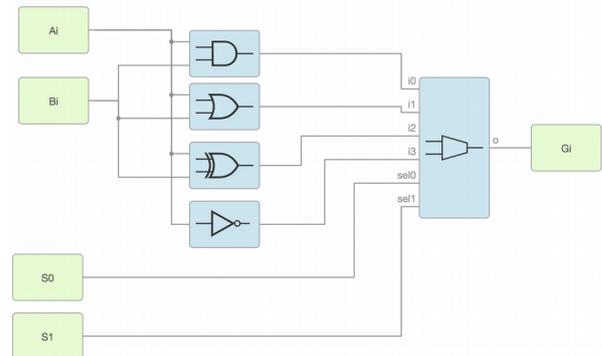


Fig. 10 Tercer nivel ALU. Unidad de operación lógica.

## VII. POSIBILIDADES DE MEJORA DE LAS PRÁCTICAS Y MOTIVACIÓN AL ESTUDIANTE

Como posible actuación de mejora y motivación del estudiante se podría considerar lo siguiente:

1. El beneficio de probar las prácticas sobre una placa física hace posible que la evaluación mediante simulación y validación experimental de todas las prácticas sea bastante sencilla, permitiendo al estudiante que compruebe fácilmente que el funcionamiento de su diseño es el correcto. En el caso de no disponer físicamente de la placa, el estudiante podría realizar su práctica utilizando el simulador. Uniendo esto, con el estudio teórico previamente realizado, el estudiante puede conocer

de una forma sencilla si el trabajo realizado es correcto.

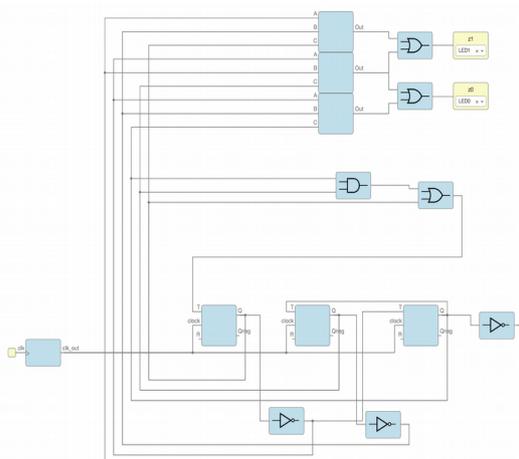


Fig. 11. Generador de secuencias.

2. Dado que la placa *IceZUM Alhambra* es compatible con escudos (*shields*) de *Arduino*, la ampliación de la misma es inmediata, permitiendo al estudiante probar una serie de experimentos no disponibles con otros simuladores de circuitos lógicos o entrenadores lógicos, por ejemplo, control de vehículos, robots, motores, etc. Ampliaciones que pueden motivar al estudiante a avanzar en conceptos que pueden serle útiles de cara a su utilización en otras asignaturas y consolidando las competencias que debe adquirir en la asignatura de Tecnología y Organización de Computadores.
3. De hecho, se está trabajando en un programa de prácticas alternativo basado en diseño de robots [9]. De momento se ha diseñado un pequeño vehículo de Braitenberg [10] basado en la placa *IceZUM* y el robot BQ Evolution [11], en el que se se han añadido dos sensores de luz, de forma que los estudiantes puedan implementar diferentes comportamientos para que el robot persiga la luz o se aleje de ella.

## VIII. CONCLUSIONES

Gracias al desarrollo e implementación de las prácticas referidas en secciones anteriores, se puede dotar al estudiante de conocimientos muy útiles sobre lógica digital que abren un mundo de posibilidades de aprendizaje y trabajo. Esto facilita la adquisición de competencias en la asignatura de Tecnología y Organización de Computadores, particularmente en el análisis y diseño de sistemas combinacionales y secuenciales. Estos módulos son elementos básicos de cualquier sistema digital y constituyen uno de los pilares de la asignatura. Además los estudiantes pueden llegar a comprender conceptos de metodologías de diseño como técnicas de diseño modular y escalable. Finalmente adquieren la capacidad para evaluar la complejidad de un diseño y su impacto en las potenciales prestaciones del sistema.

Más allá de los conocimientos generales adquiridos en la asignatura, el *hardware* reconfigurable está cada vez más extendido y son más sistemas los que utilizan

*hardware* de este tipo, por lo que es totalmente recomendable aprender a utilizar este tipo de tecnología. Además, una forma experimental comenzando con módulos básicos y finalizando con el diseño de un procesador sencillo o módulos de control de robots motiva al estudiante hacia perfiles no exclusivamente de software en su titulación, propiciándolo a cursar asignaturas de un perfil de Ingeniería de Computadores. Aparte de todo esto, aprender diseño lógico combinacional y secuencial en un sistema real es mucho más ameno que en un simulador, donde el estudiante puede no percibir claramente la utilidad de lo que está haciendo.

Se propone la utilización de una plataforma de las denominadas de *hardware* abierto, lo que acerca aún más este mundo a las personas no profesionales. Sin embargo, hay que mencionar que la plataforma *IceZUM Alhambra* es pequeña comparada a las que se utilizan normalmente en el mundo profesional, por lo que es fácil quedarse sin recursos, lo que no evita que se puedan realizar diseños muy interesantes e integrarlos con cualquier otro sistema.

Otro punto importante a destacar es que, de momento, para poder simular los diseños realizados en *IceStudio* es necesario exportarlos a *Verilog*, y crear un *testbench* en *Verilog* que genere los estímulos, para poder visualizar posteriormente los cronogramas mediante *GTKWave*. Aunque los ficheros de *testbench* se pueden dejar preparados para los estudiantes, no deja de ser un proceso relativamente complejo. Sin embargo, dado que todas las herramientas utilizadas son de código abierto, se está planteando modificar la herramienta *IceStudio* para que permita la generación automática de los *testbenches* y la invocación a *GTKWave*, de forma que se simplifique la simulación de los diseños por parte de los estudiantes.

## AGRADECIMIENTOS

Este trabajo ha sido financiado entre el departamento de Tecnología y Arquitectura de Computadores de la Universidad de Granada y el proyecto del Ministerio de Economía y Competitividad AMIGA-6 (AYA2015-65973-C3-2-R).

## REFERENCIAS

- [1] Universidad de Granada. *Web oficial del Grado en Ingeniería Informática*. [Online]. Disponible: <http://grados.ugr.es/informatica>. Último acceso: 26/06/2017.
- [2] Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada. *Guía docente de la asignatura Tecnología y Organización de Computadores*. [Online]. Disponible: [http://grados.ugr.es/informatica/pages/infoacademica/guias\\_docentes/curso\\_actual/primer/2semestre/tecnologiayorganizaciondecomputadores](http://grados.ugr.es/informatica/pages/infoacademica/guias_docentes/curso_actual/primer/2semestre/tecnologiayorganizaciondecomputadores). Último acceso: 26/06/2017.
- [3] Capilano Computing Systems Ltd.. *Logic Works 5*. Addison Wesley, 2004.
- [4] S. Díaz Ruiz, M. C. Romero Ternero, A. J. Molina Cantero. *Estructura y Tecnología de Computadores. Teoría y problemas*. McGraw-Hill, 2009.
- [5] J. González. *IceZUM Alhambra board*. [Online]. Disponible: <https://github.com/FPGAwards/icezum/wiki>. Último acceso: 26/06/2017.

- [6] Lattice Semiconductor. *iCE40TM LP/HX Family Data Sheet*, 2017.
- [7] J. González and J. Arroyo. *IceStudio*. [Online]. Disponible: <https://github.com/FPGAwards/icestudio>. Último acceso: 26/06/2017.
- [8] Stephen Williams. *Icarus Verilog*. [Online]. Disponible: <http://iverilog.icarus.com/>. Último acceso: 26/06/2017.
- [9] K. Brands, E. Ros, J. González, M. Rodríguez, P. Martín-Smith. *IceZUM for robotics. Building low level wired behaviours for robots*, en Actas de las III Jornadas Andaluzas de Informática (JAI'2017), Canillas de Aceituno, Málaga, Septiembre 2017.
- [10] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*. The MIT Press, Cambridge Massachusetts, 1986.
- [11] BQ. *Printbot Evolution*. [Online]. Disponible: <https://www.bq.com/es/printbot-evolution>. Último acceso: 26/06/2017.