

# PROGRAMACIÓN DE DISPOSITIVOS LÓGICOS PROGRAMABLES A TRAVÉS DEL ENTORNO XILINX

Rosado A., Bataller M., Soria E., Martínez M.,  
Francés J.V., Magdalena J.R., Serrano A.J.  
GPDS. Dpto. Electrónica e Informática. Universidad de Valencia.  
C/ Dr. Moliner, 50. 46100 Burjassot. Valencia. ESPAÑA

## RESUMEN

*En la docencia de la lógica programable, la disponibilidad de un entorno completo de diseño a la hora de la realización de prácticas en laboratorio es un factor importante a tener en cuenta. Con el entorno de programación que ofrece XILINX, se ofrece al alumno la realización de un recorrido completo a través de las etapas necesarias para la programación de estos dispositivos, desde su diseño inicial, ya sea a través de captura de esquemas o con ABEL-HDL, hasta su comprobación en la placa de pruebas. Todo esto se realiza de una forma guiada de tal forma que se aprenden las técnicas de diseño con un número reducido de prácticas y se puede analizar con detalle la implementación realizada.*

## 1. INTRODUCCIÓN.

La docencia relacionada con los temas relativos a la lógica programable se enmarca dentro de la asignatura de Diseño de Circuitos y Sistemas Electrónicos perteneciente al cuarto curso de Ingeniería Electrónica que se imparte en la Universitat de València, y como complemento a esta asignatura se dispone de su correspondiente laboratorio. Al tratarse de una asignatura en la que no sólo se habla de dispositivos programables, sino que se incluyen temas relacionados con dispositivos ASIC y a medida, no es posible dedicar un gran número de sesiones a la lógica programable, por lo que se ha optado por la elección de un sistema de prácticas guiadas en las que el alumno ya dispone de un material inicial y en el que no se realizan labores de diseño lógico, para así poder disponer de ese tiempo para el aprendizaje del manejo del programa así como las diversas posibilidades de las que se dispone para controlar el proceso de implementación del diseño, tanto en lo referente a la simulación como al emplazamiento, enrutado y análisis de tiempos de propagación.

El principal objetivo de estas prácticas consiste en la familiarización del alumno con el flujo de diseño necesario para la programación de estos dispositivos (figura 1). Esto es así dado que para poder cursar esta asignatura, es necesario tener conocimientos previos de diseño digital, y por tanto, para los alumnos se asume un nivel que permite la comprensión del diseño que se va a realizar.

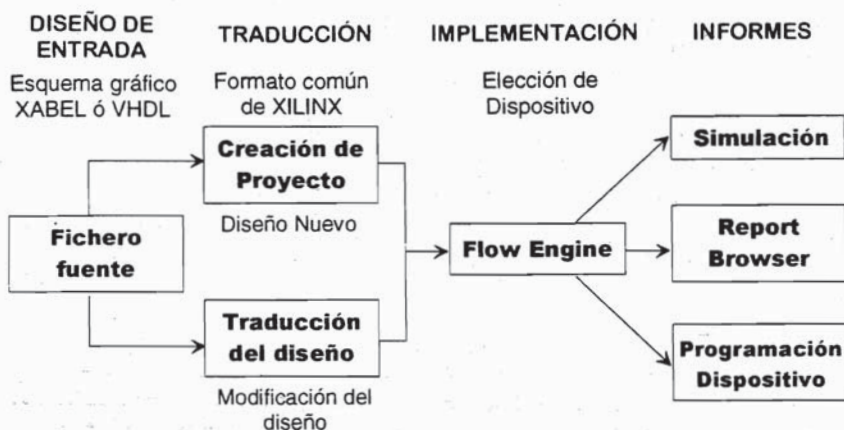


Figura 1. Flujo de diseño para los dispositivos lógicos programables.

La captura esquemática y simulación se realiza a través de Workview Office 7.3 de Viewlogic, y una vez finalizado, empleando Xilinx XACT Step 6.0 (próximamente se empleará la versión M1.3) se procede a la implementación física sobre un dispositivo FPGA que posteriormente será programado desde el PC a través del cable Xchecker proporcionado por Xilinx y que comunica el PC con la placa de pruebas.

## 2. DESCRIPCIÓN DE LA PRÁCTICA.

La práctica, originariamente propuesta por XILINX, se debe realizar a lo largo de dos sesiones de 4 horas de duración cada una, en ella se propone la implementación de un mini-procesador con una pila de datos y un conjunto reducido de instrucciones que permitirán la lectura de datos externos de entrada a través de interruptores y la salida de resultados que se visualizarán en los diodos LED de la placa de pruebas que XILINX proporciona. La lista de instrucciones se muestra en la tabla de la página siguiente.

Los esquemas que representan el diseño se proporcionan al alumno completados en un 90%, de tal forma que el manejo de la herramienta de captura de esquemas se aprende de forma rápida debiendo completar el diseño sin la necesidad de introducir el esquema jerárquico correspondiente a todo el sistema y que le llevaría gran parte de la práctica. Pero con un nivel de diseño suficiente como para apreciar el método a seguir y familiarizar al alumno con los menús y opciones más frecuentemente utilizadas en la introducción del diseño mediante la captura esquemática de Viewlogic [1,2], y que incluye la creación de esquemas jerárquicos con creación de símbolos de librería propios.

Un factor importante consiste en la familiarización del alumno con la inclusión de características específicas del dispositivo al que va destinado el diseño en esta etapa esquemática, ya que resultan de gran comodidad y permiten controlar el proceso de implementación posterior. Además, se indican las características específicas que debe contener un diseño FPGA para poder ser aceptado por el sintetizador del diseño (inclusión de buffers y PAD's específicos). Como ejemplo de estas características, se propone la asignación

directa de unas entradas y salidas a determinadas patillas del dispositivo a emplear, paso obligatorio pues el diseño se probará sobre una placa de pruebas donde las salidas y entradas están fijadas; y también se propone la especificación de algunos atributos como la inicialización de registros y restricciones de tiempo específicas entre determinadas señales. Uno de estos esquemas se proporciona en formato ABEL-HDL [3] del que posteriormente se genera un símbolo para insertar en el esquema y así entrar a formar parte del esquema jerárquico que constituye el diseño. Se hace especial hincapié en esta posibilidad que ofrece el entorno, pues en determinadas situaciones como la implementación de máquinas de estado, resulta de gran comodidad. En posteriores sesiones se profundizará en el lenguaje ABEL-HDL, por lo que se emplaza a los alumnos a reconsiderar el diseño posteriormente.

A continuación se procede a la simulación del sistema, y para ello, también se le proporciona al alumno un fichero de comandos de simulación de Workview Office [4] incompleto, pero a través del cual pueden apreciar la mecánica de funcionamiento. De esta forma, se indica qué resultados se deben obtener para conseguir así la completa simulación del diseño (comprobación de las instrucciones que se muestran en la Tabla 1). Esto permite que el alumno construya los comandos de simulación de forma más intuitiva y adquiera los conocimientos básicos sobre el lenguaje de comandos de simulación en un tiempo relativamente corto.

Interruptor	2	3	4	5	6	7	8	Operación
0	0	0					Dato	SUMA entre interruptores y registro
0	0	1					Dato	AND entre interruptores y registro
0	1	0					Dato	OR entre interruptores y registro
0	1	1					Dato	XOR entre interruptores y registro
1	0	0					Dato	RESTA el valor de los interruptores de los del registro
1	0	1	X	X	X	X		PUESTA A CERO del registro
1	1	0					Dato	CARGA del registro con el valor de los interruptores
1	1	1	0	0	0	0	X	SUMA entre la pila y el registro
1	1	1	0	0	1	1	X	AND entre la pila y el registro
1	1	1	0	1	0	0	X	OR entre la pila y el registro
1	1	1	0	1	1	1	X	XOR entre la pila y el registro
1	1	1	1	0	0	0	X	RESTA el valor de la pila del registro
1	1	1	1	0	1	1	X	APILAR (PUSH) el valor del registro en la pila
1	1	1	1	1	1	0	X	DESAPILAR (POP) el valor de la pila en el registro
1	1	1	1	1	1	1	X	NOP no realiza ninguna operación

Tabla 1. Especificación de funcionamiento del sistema a diseñar.

Una vez realizada la simulación funcional (pre-layout), se procede al empleo del programa de optimización, emplazamiento y conexionado de XILINX (XACT Step 6.0) [5], donde se indica al alumno que se empleará un dispositivo XC4003A-6PC84C, y del que previamente se ha estudiado su arquitectura interna, distribución de células lógicas, recursos de conexionado, etcétera. En esta etapa se hace un recorrido a través de las diferentes ventanas que permiten el control de la implementación y optimización, indicando los aspectos más relevantes del

proceso y mostrando el efecto que la selección de los parámetros produce sobre la lógica interna del circuito integrado

Así, una vez finalizado el proceso de traducción e implementación, se debe realizar un análisis minucioso de dicho proceso, esto se observa a través de las diferentes herramientas que ofrece el programa y permiten rastrear las señales y modificar su posición o trazado, analizar tiempos de propagación, nivel de ocupación, etc. Se analiza con profundidad la herramienta llamada "**Report Browser**" que es donde se muestran los diferentes resultados de la implementación. Para que el alumno se familiarice con este tipo de resultados que a veces resultan bastante densos, se proponen cuestiones cortas que instan a repasar cada uno de los ficheros de resultados; estas cuestiones están directamente relacionadas con el nivel de ocupación del dispositivo, propagación de señales a través de las diferentes células lógicas, estudio de prestaciones de velocidad donde se incluye el empleo de la herramienta "**Timing Analyzer**", y como parte final, se aportan ciertas indicaciones del manejo de la herramienta "**Floorplanner**" donde se puede realizar un emplazamiento manual de cada una de las células de las que consta el diseño y se explica el procedimiento a realizar para generar el fichero de programación de la memoria de almacenamiento del programa de configuración de la FPGA ("**PROM File Formatter**").

Como conclusión de la práctica, se procede a la comprobación del diseño sobre la placa de pruebas. En este paso, el programa "**Hardware Debugger**" se encarga de conectarse a la placa de pruebas [6] a través del cable Xchecker y descargar el programa sobre el dispositivo. Todos estos pasos se describen detalladamente para que el alumno siga un ritmo de trabajo continuado y que permita la finalización de la práctica en el tiempo establecido.

Con el dispositivo ya programado, el alumno debe realizar una comprobación de funcionamiento en la que mediante pulsadores en la placa deben aplicarse las mismas entradas que se habían simulado previamente y así mostrar al alumno la relación directa entre la etapa previa de comprobación software y la de comprobación hardware.

### 3. CONCLUSIONES.

Como resumen final, podemos decir que se ha tratado de realizar una práctica en la que se pretende que el alumno conozca el proceso de implementación de dispositivos programables, para ello se ha elegido un entorno comercial que permita acercar al alumno a los programas que utilizará posteriormente en la industria, y que debido a su gran potencia no siempre es posible abordar en unas sesiones de prácticas reducidas como es nuestro caso. Para ello se ha recurrido a unas sesiones guiadas en la que se proporciona al alumno gran cantidad de información previamente confeccionada para así poder centrar la práctica en aspectos más relevantes de la lógica programable, sin necesidad de perder un gran número de horas con la familiarización del entorno e introducciones muy específicas a las herramientas de soporte a la implementación. Como factor añadido, se comprueba que el alumno se familiariza pronto con el entorno y asimila con rapidez los conceptos principales involucrados en el proceso de diseño de lógica programable. La práctica propuesta toma su base de un diseño de XILINX, sobre el cual se han realizado diferentes modificaciones modulares para abarcar de forma completa las etapas de diseño como la inclusión de módulos ABEL-HDL, análisis a través del analizador de tiempos, emplazamiento manual de células y generación de ficheros intermedios para facilitar la labor de los alumnos (ficheros de restricciones, esquemáticos y de simulación incompletos)

Además, el proceso se lleva a cabo hasta su parte final, en la que el alumno, además de realizar procesos de control interno de emplazamiento y conexionado, puede apreciar

realmente el dispositivo programado y trabajar con él físicamente mediante una placa de evaluación que XILINX proporciona.

Como apartado final, se plantea un cuestionario de preguntas para comprobar que los conceptos se han asimilado y el alumno ha obtenido verdadero provecho de la práctica: esto es así, pues la experiencia nos ha indicado que el alumno obtiene importantes conclusiones y desarrolla una buena panorámica de la programación de dispositivos programables de tipo FPGA con entornos de diseño potentes y que se utilizan industrialmente.

El tiempo necesario para la finalización del proceso ha sido de dos sesiones de cuatro horas, basándose la primera práctica en el entorno esquemático y de simulación (herramientas Viewlogic), para finalizar la segunda práctica con la implementación en el dispositivo, programación física y análisis de resultados. Este tiempo resulta ser bastante corto para el manejo de herramientas de este tipo y sin experiencia previa sobre dispositivos programables, pero la dinámica propuesta ha permitido que el alumno obtenga buenos resultados y el aprovechamiento sea alto.

#### **4. BIBLIOGRAFÍA.**

- [1] Xilinx Development Systems. "Viewlogic Tutorials". Xilinx. 1997.
- [2] R. James Duckworth. "Workview Office Student Edition". Prentice-Hall. 1997.
- [3] DATA I/O. "ABEL Design Software: User Manual". DATA I/O Corporation. 1990.
- [4] Xilinx Development Systems. "Viewlogic Interface / Tutorial Guide". Xilinx. 1997.
- [5] Xilinx Development Systems. "Design Manager/Flow Engine Reference/User Guide". Xilinx. 1997.
- [6] Xilinx Development Systems. "XACT Hardware & Peripherals Guide". Xilinx. 1994.