

EXPERIENCIAS EDUCATIVAS SOBRE LENGUAJES DE DESCRIPCIÓN DE HARDWARE

José Ignacio Artigas, Luis Angel Barragán, José Ignacio García e Isidro Urriza.
Dpto. Ingeniería Electrónica y Comunicaciones. Universidad de Zaragoza.
María de Luna 3, 50015 Zaragoza.
Tlfno. 976761948, fax 976762111, email jartigas@posta.unizar.es.

RESUMEN

Este trabajo está basado en la experiencia en la enseñanza de lenguajes de descripción de hardware (HDLs) en distintas asignaturas pertenecientes a varias titulaciones. Se pretende establecer criterios para la elección del HDL e incluso de la herramienta a utilizar, en función de los conocimientos previos del alumno y del contenido y distribución temporal de la asignatura.

1. INTRODUCCIÓN

La mayor parte de las herramientas de diseño asistido por ordenador (CAD) para PLDs y ASICs utilizan un HDL para capturar el diseño digital, aunque también aceptan diseños creados con programas de captura de esquemas. Muchos libros de texto de diseño digital ya dedican al menos un capítulo a este tema ([1-4]. Los libros de Wakerly y Floyd utilizan ABEL, Hayes presenta VHDL y Nelson utiliza VHDL y PDL. Los libros dedicados al diseño microelectrónico también introducen metodologías de diseño basadas en HDLs [5]. Además existen otros libros dedicados exclusivamente al diseño con HDLs [6-8]

El objetivo de este trabajo es establecer criterios para la elección del HDL e incluso de la herramienta a utilizar, en función de los conocimientos previos del alumno, del contenido y distribución temporal de la asignatura y de los recursos disponibles.

2. CRITERIOS DE COMPARACIÓN

De entre todos los HDLs, en este trabajo nos centramos en ABEL y en VHDL. En la tabla 1 se comparan ambos lenguajes según distintos parámetros.

	ABEL	VHDL
Origen	DATA I/O	IEEE estándar
Representaciones de diseño	Comportamental	Comportamental, Estructural
Nivel de abstracción	Ecuaciones, RTL, MEF	RTL, MEF, Sistema
Tipo de implementación	PLD, FPGA	PLD, FPGA, ASIC
Tiempo de aprendizaje	Bajo	Alto
Tipos de datos para síntesis	MVL	MVL, entero, enumerado
Tipos de datos para simulación	MVL	Anteriores + real. <i>string, file</i>
Complejidad del diseño	Baja	De baja a alta

Tabla 1. HDLs

Como conclusiones se puede establecer que ABEL es un lenguaje orientado a diseños más sencillos que VHDL lo que se refleja en un menor número de tipos de datos y en un nivel de abstracción más bajo. Todo ello, hace que se agilice el aprendizaje del alumno. Sin embargo, para el diseño de sistemas más complejos se requiere un lenguaje con una mayor potencia de modelado como VHDL, que además es un estándar IEEE.

A la hora de realizar las correspondientes prácticas de laboratorio deberemos de elegir una herramienta de trabajo. Estas herramientas se pueden clasificar en específicas y genéricas. Las primeras permiten el desarrollo de diseños sólo para sus PLDs. Las genéricas permiten diseñar independientemente del dispositivo y después asociar el diseño al dispositivo deseado.

De acuerdo con nuestra experiencia y teniendo en cuenta que ABEL está orientado a diseños sencillos hemos seleccionado tres herramientas:

- ABEL 4.0 de Data I/O que resulta sencilla de aprender y utilizar y que se ejecuta en DOS.
- Los sistemas de iniciación de Synario (Starter Kits) [12, 13] ofrecidos gratuitamente por varios fabricantes de PLD.
- Xilinx Student Edition [9] que incluye el libro "The Practical Xilinx Designer Lab Book" [10] y el software de Xilinx Foundation Series F1.3 para CPLDs y FPGAs de la familia XC4000.

Para realizar diseños con VHDL hemos elegido también tres herramientas:

- Warp2 de Cypress, herramienta sencilla y barata. La versión 3.5 tiene pocos requerimientos de máquina. La versión 4 está incluida con el libro "VHDL for Programmable Logic" [11], es más completa y requiere más máquina.
- Foundation Express F1.4 de Xilinx, que se puede obtener gratuitamente como actualización de "Xilinx Student Edition".
- Synopsys, herramienta mucho más compleja que permite diseño microelectrónico.

	Abel 4.0	Synario (Starter Kits)	Xilinx Student Edition	Xilinx Found. Express F1.4	Warp2 V3.5 y V4.0	Synopsys
Captura del diseño	ABEL	ABEL, esquemas	ABEL, esquemas	ABEL, VHDL y esquemas	VHDL	VHDL, Esquemas
Orientado a	PLD	PLD, FPGA	CPLD, FPGA	CPLD, FPGA	PLD, FPGA	CPLD, FPGA, ASIC
Tipo de herramienta	Genérica	Genérica	Específica	Específica	Específica	Genérica
Simulación	Funcional	Funcional	Funcional, temporal	Funcional, temporal	Funcional	Funcional, temporal
Fichero de simulación	Ecuac. lógicas, JEDEC	Ecuac. lógicas	Netlist	Netlist	JEDEC	VHDL
Generación de estímulos	Vectores de test	Vectores de test y formas de onda	Formas de onda, teclado	Formas de onda, teclado	Formas de onda	Entorno de test
Plataforma	DOS, Windows	Windows	Windows	Windows	Windows	Unix
Coste de la plataforma	Bajo	Medio	Medio	Medio	Medio	Alto
RAM (MB)	< 1	16	32	32	8 - 16	64
Disco Duro (MB)	2	15 - 35	150 - 250	150 - 275	8 - 100	> 1GB
Coste de la herramienta	Bajo	Gratis	Bajo (\$87)	Actualización gratuita de F1.3	Bajo (\$54)	Muy Alto
Complejidad de utilización	Baja	Media	Media	Media	Media	Muy Alta

Tabla 2. Herramientas

En la tabla 2 se comparan las seis herramientas citadas anteriormente, en función de una serie de parámetros.

3. ELECCIÓN SEGÚN CARACTERÍSTICAS DE LA ASIGNATURA

A la hora de elegir un HDL y herramienta se debe tener en cuenta las características de la asignatura, que hemos clasificado de la siguiente forma:

3.1 Contenido de la asignatura.

Por ejemplo, para una asignatura básica de electrónica digital en la que se introducen los dispositivos lógicos programables lo mejor es utilizar ABEL ya que es más sencillo y está más cercano a la arquitectura de los PLDs. Sin embargo, para una asignatura de microelectrónica no tiene sentido elegir ABEL. En particular, se puede elegir según la tecnología a utilizar y según la complejidad de los diseños a realizar.

- *Tecnología.* Las dos primeras herramientas de ABEL valoradas permiten trabajar con PLDs y CPLDs, aunque existen versiones más completas de Synario que también incluyen FPGAs. Warp2 permite trabajar con PLDs y CPLDs (familias c34x, c37x, c37xi); además la versión 4.1 incluye FPGAs (familias pASIC380 y Ultra38000). El problema es que sólo soporta CPLDs y FPGAs de Cypress. Xilinx Student Edition F1.3 permite diseñar con CPLDs (toda la familia XC9500) y FPGAs (familia XC4000 hasta la 4008 de 8.000 puertas equivalentes) de Xilinx. La actualización a F1.4 aumenta la complejidad de las FPGAs hasta la XC4010 (10.000 puertas), e incluye toda la nueva familia SPARTAN (hasta 40.000 puertas). En contraposición, Synopsys puede trabajar con dispositivos programables y ASICs de cualquier fabricante suponiendo que se disponga de las librerías correspondientes.
- *Complejidad de los diseños.* Synopsys permite modelar diseños mucho más complejos. Esto se refleja en las múltiples posibilidades que ofrece para implementar entornos de simulación. Sin embargo, en las demás herramientas sólo se puede simular código sintetizable por lo que no se pueden implementar entornos de test. Con las herramientas de Xilinx, es posible realizar simulaciones con el entorno integrado que proporciona estímulos procedentes de varios contadores internos, así como manejar señales mediante el teclado (asociar teclas a señales de entrada). Sin embargo no incluye simulador de VHDL. En Warp2 los estímulos de entrada de un diseño son dibujados a mano y en el resto de herramientas se utilizan tablas de vectores de test y/o formas de onda, según la herramienta.

3.2 Créditos teóricos.

En asignaturas en las que se dedica un capítulo a los dispositivos lógicos programables, es más apropiada la utilización de ABEL ya que requiere un menor número de horas para la comprensión del alumno. Según nuestra experiencia, con 4 h. de teoría y ejemplos, el alumno es capaz de escribir programas en ABEL para modelar circuitos combinatoriales y máquinas de estados. Para alcanzar el mismo objetivo con VHDL hay que multiplicar por dos (o incluso más) el número de horas.

3.3 Créditos prácticos.

Debido a su complejidad, para realizar prácticas de laboratorio con Synopsys se requiere un mínimo de 12 h. para empezar a manejarse con los entornos gráficos (SGE y DESIGN ANALYZER). Lo cual implica dedicar un gran porcentaje de las horas de prácticas a aprender la herramienta más que a diseñar. En contraposición, una herramienta más sencilla como ABEL, Synario o Warp se puede utilizar con soltura en menos de 2 horas. Las herramientas de Xilinx son algo más complejas de utilizar que estas últimas, pero están más cercanas a ellas que a Synopsys.

3.4 Coste.

El coste de las herramientas es un parámetro fundamental en muchos centros de enseñanza, pensando en que suelen ser necesarias varias licencias para permitir la realización de prácticas con varios puestos de laboratorio a la vez.

Con este parámetro, de entre las herramientas de bajo coste destaca Synario debido a su política de ofrecer sistemas de diseño de iniciación gratuitos. Estos sistemas están restringidos a dispositivos concretos de ciertos fabricantes. Incluyen la captura de esquemas así como modelado en ABEL, simulación funcional y compiladores (*fitters*) para los dispositivos. Por ejemplo existe un kit para dispositivos de Vantis que trabaja con las CPLDs MACH111 y MACH211 y las PLDs 16V8 y 22V10 [12]. También está el kit de Lattice que trabaja con las GAL y CPLDs ispLSI1016 y 2032 de 44 y 48 pines [13]. Una posible ampliación de este último kit es el "ISP Synario System", que trabaja con las GAL de Lattice y las familias ispLSI, pLSI, ispGDS. Sólo cuesta \$495 con cable de programación incluido. Los sistemas completos de Synario suelen costar entre \$1800 y \$12000, según las opciones elegidas, claro que también incluyen VHDL.

Otra alternativa muy atractiva en cuanto a coste es Xilinx Student Edition, que con un precio de \$87, y una vez actualizado a F1.4, ofrece la posibilidad de trabajar con ABEL, VHDL y esquemas. Incluye las CPLDs de Xilinx (familia XC95000) y algunas de sus FPGAs (XC4000 hasta la 4010 y SPARTAN completa). Esta última familia tiene la ventaja añadida de su bajo coste, frente a otras FPGAs en el mercado. En esta herramienta se puede destacar la alta calidad de la síntesis de VHDL, que se realiza con tecnología de Synopsys (FPGA Express).

4. Bibliografía

- [1] J.F. Wakerly, "Digital Design Principles and Practices", 2ª edición, Prentice-Hall, 1994.
- [2] T.L. Floyd, "Fundamentos de Sistemas Digitales", Prentice-Hall, 1997.
- [3] J.P. Hayes, "Introducción al Diseño Lógico Digital", Addison-Wesley Iberoamericana, 1995.
- [4] V. P. Nelson et al., "Análisis y Diseño de Circuitos Lógicos Digitales". Prentice-Hall Iberoamericana, 1996.
- [5] W. Wolf, "Modern VLSI Design", Prentice-Hall, 1994.

- [6] D. Pellerin y M. Holley, "Digital Design using ABEL", Prentice-Hall, 1994.
- [7] S. Sjöholm y L. Lindh, "VHDL for Designers", Prentice-Hall, 1997.
- [8] E. Sternheim et al., "Digital Design and Synthesis with Verilog HDL", Automata Publishing Company, 1993.
- [9] Xilinx Student Edition web link, <http://www.xilinx.com/programs/xse1.htm>.
- [10] Dave Van den Bout, "The Practical Xilinx Designer Lab Book", Prentice-Hall, 1998.
- [11] K. Skrabill, J. Legenhausen, R. Wade, C. Wilner, B. Wilson, "VHDL for Programmable Logic", Addison-Wesley, 1996.
- [12] "Vantis-Synario Starter Software", http://www.vantis.com/software/synario_kit.html.
- [13] "ISP Synario Starter Software", <http://www.latticesemi.com/ftp/synario.html>.