

## **INTERNET EN LA DOCENCIA. VHDL, LENGUAJE DE DESCRIPCIÓN DE HARDWARE.**

**Cárdenas Bonelli, Juan Carlos**

Tlf: 94 427 53 90 / e-mail: juanky.cardenas@mailcity.com

**Jiménez Iglesias, Daniel**

Tlf: 94 433 81 85 / e-mail: danieljimenez@mailcity.com

**Oleagordía Aguirre, Iñigo**

Tlf: 94 421 94 10 / e-mail: jtpolagi@lg.ehu.es

**Sánchez Moronta, Manuel**

Tlf: 94 421 94 10 / e-mail: jtpsamom@lg.ehu.es

### **Resumen**

*El libro electrónico es un tutorial sobre VHDL, estando creado específicamente para ser publicado en Internet. Mediante la exposición clara y precisa del lenguaje y los ejemplos desarrollados y comentados incluidos se consigue un aprendizaje rápido y eficaz del alumno sin necesidad de un tutor, ya sea desde el disco duro o de la Red.*

### **1. INTRODUCCIÓN.**

El proyecto consiste en un libro electrónico sobre VHDL en el cual se dispone de todo lo necesario para poder aprender este lenguaje rápida y claramente. El tutorial ofrece un desarrollo completo de la sintaxis del lenguaje con ejemplos comentados, aparte de una serie de prácticas totalmente comentadas y desarrolladas listas para su implementación física en un dispositivo de lógica programable. Se incluye además la posibilidad de simular todas las prácticas incluidas.



### 3. PARTE TEÓRICA.

Esta es la parte principal del libro, en la cual se explica de una forma detallada y precisa la sintaxis del lenguaje, desde los orígenes de VHDL hasta cómo implementar físicamente nuestros propios diseños, pasando por todas las sentencias y la simulación.

La parte teórica está dividida en 6 lecciones, que a su vez están divididas en varios temas. Los títulos de las lecciones son los siguientes:

- Introducción
- Unidades Básicas de diseño
- Objetos, tipos de datos y operaciones
- Gramática del lenguaje I (sentencias secuenciales)
- Gramática del lenguaje II (sentencias concurrentes)
- Interface, implementación y compilador

Disponemos, para hacer de una navegación más sencilla, de multitud de hipervínculos que nos llevan al índice de la lección, a otras lecciones y a los ejemplos de la parte práctica. De esta forma conseguimos que a cada exposición teórica, que incluye su propio ejemplo, le complemente una práctica, con el objetivo de asentar los conocimientos perfectamente. Además, al principio de la lección hay hipervínculos a cada tema y al final los hay a la lección anterior y siguiente, para facilitar aún más el uso del libro.

El aspecto de una lección cualquiera es el siguiente:

## Introducción

### 1 Breve reseña histórica

### 2 VHDL

### 3 Qué es un PLD (Dispositivo de Lógica Programable)

### 4 Conociendo el lenguaje VHDL

### 5 Sentencias Concurrentes y secuenciales

### Breve reseña histórica



A mediados de los años setenta se produce una fuerte evolución en los procesos de fabricación de los circuitos integrados, y junto a las tecnologías bipolares, surge la MOS (*metal oxide semiconductor*).

Imagen 2. Aspecto de la parte teórica

## 4. PARTE PRÁCTICA

En esta parte se exponen 14 prácticas reales, listas para compilar, simular y grabar en un dispositivo de lógica programable. Cada práctica se ha descrito tratando de usar un estilo de programación distinto, para cubrir cada una de las sentencias y peculiaridades del lenguaje expuestas en la parte teórica.

El objetivo de este apartado es poder dar al alumno herramientas para poder manejar el lenguaje y crear sus propios diseños al poco tiempo de haber empezado a usar el libro electrónico, incrementado de esta manera el interés por VHDL y por el libro. Para conseguir esto se han escogido formas de programar los ejemplos, que aunque no son las más profesionales, sí son las más pedagógicas, y se ha dispuesto de tres formas de ayuda:

- Exposición de la operatividad de todo el conjunto.
- Separación de sentencias por bloques
- Explicación del código del lenguaje línea a línea

De esta forma se evita que el alumno tenga dudas sobre lo que acaba de aprender, al estar todo perfectamente estructurado.

Disponemos en la parte práctica de una pantalla de presentación para cada ejercicio que nos llevará, pulsando en el botón correspondiente, a cada uno de los subapartados del mismo y a la página de simulación. El aspecto de ésta página de presentación es el siguiente:

**Multiplexor**

? Un multiplexor es un dispositivo lógico que recibe información por sus dos o más entradas (de uno o mas bits de ancho) y mediante una señal de control decidimos cual de las entradas aparece reflejada en la salida; esto es, un convertidor de paralelo a serie. Si tienen una señal de "enable" esta hace que el multiplexor esté habilitado o no

↓ Elige en la lista una práctica para empezar, sin más que pinchar en el botón adecuado. Para volver pulsa el botón que está al lado del título

Los multiplexores que hemos propuesto tienen todos señal de enable menos el primero. Las arquitecturas están realizadas solo para algunas entidades

- Entidad 1 Multiplexor de un bit y dos canales sin "enable"
- Entidad 2 Cómo añadir una señal de "enable"
- Entidad 3 Como aumentar los bits de cada canal
- Entidad 4 Cómo aumentar el número de canales
- Arquitectura 1 Completar la entidad 4 con estilo algorítmico o de comportamiento
- Arquitectura 2 Completar la entidad 2 con estilo flujo de datos
- Arquitectura 3 Completar la entidad 2 con estilo estructural

**SIMULACIÓN**

Imagen 3. Aspecto del índice de la práctica del multiplexor.

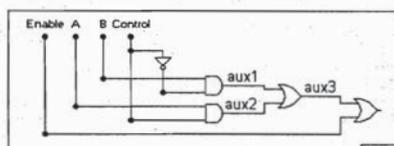
Pulsando en el botón correspondiente podremos visitar el apartado deseado, en el cual encontraremos el ejemplo con su explicación correspondiente (incluyendo dibujos, tablas y textos flotantes), aparte de botones que nos devuelven a la página de presentación y que nos permiten simular *directamente* el ejemplo que se acaba de estudiar, para comprender aún más lo expuesto. Se incluye además en cada práctica, la posibilidad de ir a la parte de la teoría que sea necesaria para poder dar un apoyo extra al alumno, en caso de que la explicación del ejemplo no le haya satisfecho.

Un ejemplo de cómo es un subapartado de las prácticas se muestra a continuación:

## Arquitectura 2: Estilo *flujo de datos* usando la entidad 2



El estilo de flujo de datos nos muestra la funcionalidad de un dispositivo mediante ecuaciones ejecutadas concurrentemente, es decir, todas a la vez. Mediante las sentencias and, xor, not, and... transformamos las entradas en las salidas. Para el ejemplo del multiplexor de dos canales de un bit con "enable" el esquema de puertas lógicas es el siguiente:



Esquema en base a puertas lógicas del multiplexor de la entidad 2

```

1  entity multi is port(
2  a,b      : in  bit;
3  control  : in  bit;
4  enable   : in  bit;
5  c        : out bit
6  );
7  end multi;
8  architecture archmul of multi is
9  signal aux1,aux2,aux3: bit;
10 begin
11 aux1 <= b and (not(control));
12 aux2 <= a and control;
13 aux3 <= aux1 or aux2;
14 c   <= enable and aux3;
15 end archmul;

```

Cabecera del programa  
 Declaración de señales  
 Empieza el programa  
 Sentencia concurrente  
 Sentencia concurrente  
 Sentencia concurrente  
 Sentencia concurrente  
 Finaliza el programa

- En la línea 9 se declaran las señales aux1, aux2 y aux3, que son señales intermedias que no forman parte de la entrada ni de la salida y que las utilizamos para hacer más cómodamente la estructuración del programa. Si estas señales, el código sería más confuso aunque igualmente válido, ya que de no usarlas deberíamos haber sustituido las líneas de la 11 a la 14 por la siguiente:

```
d <= enable and ( (b and (not(control))) or (a and control) );
```

- La explicación de las sentencias que están en la línea 11 a la 14 habla por sí solas sin más que mirar el código. Con una simple inspección comprobamos que se corresponde con el gráfico situado sobre el código.

Antes de simular

Ver código fuente

Simulación

## 5. SIMULACIÓN.

La gran ventaja de haber hecho este tutorial en HTML es que se puede hacer uso de otros programas distintos al navegador que se esté usando, es decir, se pueden lanzar aplicaciones. Si se llaman desde la misma página web las herramientas de simulación de que incluye cualquier compilador de VHDL, se puede observar el comportamiento cada uno de los ejemplos compilados que se incluyen en las prácticas, mostrando al alumno en pantalla el resultado obtenido del código fuente que acaba de estudiar.

Para simular un ejemplo tan solo se tiene que pulsar el botón adecuado para que se ejecute el programa que esté asociado a los ficheros de extensión ".jed". De esta forma, cada usuario podrá utilizar el programa al que esté más habituado, siendo un uso aún más personal. Tan sólo se recomienda que sea un programa que se ejecute con rapidez como el Nova de Cypress Semiconductor Inc.

## 6. BIBLIOGRAFÍA.

[1] Z. Navabi. "VHDL. Analysis and Modeling of Digital Systems". Ed. McGraw-Hill, Inc. 1993.

[2] K. Skahill. "VHDL for Programmable Logic". Ed. Addison-Wesley. 1996.

[3] E. Villar, L. Terés, S. Olcoz y Y. Torroja. "VHDL. Lenguaje Estándar de Diseño Electrónico". Ed McGraw-Hill. 1997.