

## TEMA 1

# FUNDAMENTOS DEL DISEÑO DEL HARDWARE DIGITAL

- 1.1. Introducción
- 1.2. Lenguajes para la descripción de hardware
- 1.3. Ciclo de diseño de los circuitos digitales
- 1.4. Tecnologías de circuitos integrados
- 1.5. Propiedades de los circuitos digitales
- 1.6. Test de los circuitos
- 1.7. Representaciones y niveles de abstracción
- 1.8. Conceptos básicos a través de un ejemplo
- 1.9. Dos simuladores de VHDL'93: VeriBest y ModelSim
- 1.10. Lecturas recomendadas
- 1.11. Ejercicios de autocomprobación
- 1.12. Soluciones de los ejercicios

## OBJETIVOS DOCENTES

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Discutir la finalidad de los lenguajes para la descripción del hardware (HDL) y algunas de las principales ventajas que presenta su uso.
- Discutir el ciclo de diseño del hardware digital y el papel que desempeñan en el ciclo de diseño los HDL.
- Discutir y comparar las características de las siguientes tecnologías de fabricación de circuitos integrados: full-custom ASIC, standard cell ASIC, gate array ASIC, dispositivos programables en campo (FPGA, CPLD, PROM, PAL y PLA), y componentes estándar de pequeño y medio tamaño.
- Discutir las siguientes propiedades de los circuitos digitales: el retardo de los dispositivos, su ejecución concurrente, la marginalidad en el diseño y la fortaleza de las señales.
- Discutir el propósito y los fundamentos del test en diseño y manufactura, así como los conceptos: modelo de fallos, cobertura del test y calidad del test.
- Discutir la utilidad y composición de los bancos de pruebas.
- Discutir y comparar los niveles de abstracción y representación de los sistemas digitales.
- Instalar en su propio ordenador y realizar las operaciones básicas de manejo de algún entorno de simulación de VHDL'93 de su elección. Estas operaciones básicas incluyen al menos la edición de modelos VHDL, su depurado usando el debugger, su simulación y la visualización de los resultados.

## TEMA 2

# CONCEPTOS BÁSICOS DE VHDL

- 2.1. Introducción
- 2.2. Unidades de diseño
- 2.3. Entity
- 2.4. Architecture
- 2.5. Asignaciones concurrentes
- 2.6. Sentencia generate
- 2.7. Bloque process
- 2.8. Código secuencial
- 2.9. Descripción de la estructura
- 2.10. Parametrización
- 2.11. Señales, variables y constantes
- 2.12. Tipos de datos y operadores
- 2.13. Atributos
- 2.14. Librerías
- 2.15. Assert
- 2.16. Subprogramas
- 2.17. Paquetes
- 2.18. Lecturas recomendadas
- 2.19. Ejercicios de autocomprobación
- 2.20. Soluciones de los ejercicios

## OBJETIVOS DOCENTES

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Discutir la finalidad de las *unidades de diseño* de VHDL **entity**, **architecture**, **package** y **configuration**.
- Describir la interfaz de los circuitos digitales mediante *unidades de diseño* **entity**. Discutir las características y las diferencias entre los modos **in**, **out** e **inout** de los *puertos*.
- Discutir la sintaxis y la finalidad de las sentencias concurrentes simple, condicional y de selección. Dibujar el diagrama conceptual del hardware a que da lugar la síntesis de estas sentencias y el correspondiente circuito al nivel de puertas lógicas.
- Discutir la sintaxis y la finalidad de la sentencia **generate**.
- Discutir la sintaxis y la finalidad de los bloques **process** y de las sentencias secuenciales de asignación a señal, a variable, **if**, **case** y del bucle **for**. En casos sencillos, dibujar el diagrama conceptual del hardware a que da lugar la síntesis del bloques **process**, y el correspondiente circuito compuesto de puertas lógicas y biestables.
- Discutir cómo se realiza el modelado del retardo en VHDL.
- Definir en VHDL la estructura de un circuito mediante instanciación y conexión de otros circuitos. Asimismo, discutir la utilidad de la parametrización en la descripción de un circuito.
- Discutir las principales características de las señales, variables y constantes en VHDL. Discutir la diferencia entre señales y variables en lo que respecta al retardo en la asignación de los nuevos valores.
- Discutir cuáles son los tipos predefinidos de VHDL y sus operadores básicos. Asimismo, discutir la finalidad, los operadores básicos y las funciones de conversión de los tipos de dato **std\_logic**, **std\_logic\_vector**, **unsigned** y **signed**. Finalmente, discutir las características de los tipos de dato **time** y **string**, así como de los tipos enumerados.
- Discutir qué son los atributos en VHDL y la finalidad de algunos de ellos.

- Discutir qué finalidad tienen las librerías en VHDL y conocer las librerías más comúnmente usadas.
- Discutir la utilidad de los procedimientos y funciones de VHDL.
- Discutir la finalidad de las sentencias **assert** y **report**.

## TEMA 3

# SIMULACIÓN DEL CÓDIGO VHDL

- 3.1. Introducción
- 3.2. Procesamiento del código VHDL
- 3.3. Orden de compilación
- 3.4. Drivers
- 3.5. Inicialización
- 3.6. Atributos de las señales
- 3.7. El retardo delta
- 3.8. Gestión de la cola de transacciones del driver
- 3.9. Ejemplo: simulación de un circuito sencillo
- 3.10. Lecturas recomendadas
- 3.11. Ejercicios de autocomprobación
- 3.12. Soluciones de los ejercicios

## OBJETIVOS DOCENTES

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Discutir las etapas *análisis*, *elaboración* y *ejecución*, de que consta el procesamiento y simulación del código VHDL realizado por las herramientas de CAD.
- Discutir el orden en el cual debe compilarse el código VHDL, atendiendo a las relaciones existentes entre las diferentes unidades de diseño.
- Discutir los conceptos *driver* y *función de resolución*.
- Discutir en qué consiste la *inicialización* y cómo se realiza.
- Discutir qué son los *atributos* de las señales y los conceptos *evento*, *señal activa*, *transacción* y *señal implícita*.
- Discutir cómo se realiza la simulación de las asignaciones con retardo  $\delta$ , y cómo se combina la simulación de asignaciones con retardo explícito y asignaciones con retardo  $\delta$ .
- Discutir cómo se gestionan las colas de transacciones de los *drivers*.
- Realizar manualmente, paso a paso, la simulación de diseños sencillos, indicando cómo se realiza la inicialización, cómo va avanzando el reloj de la simulación, qué eventos se producen y cómo se gestionan las colas de transacciones planificadas. Saber comprobar el resultado de las simulaciones realizadas manualmente, usando para ello algún simulador de VHDL.

## TEMA 4

# DISEÑO DE LÓGICA COMBINACIONAL

- 4.1. Introducción
- 4.2. Diseño para síntesis de lógica combinacional
- 4.3. Funciones lógicas
- 4.4. Multiplexor de 4 entradas
- 4.5. Restador completo de 1 bit
- 4.6. Sumador completo de 1 bit
- 4.7. Unidad aritmético lógica
- 4.8. Lecturas recomendadas
- 4.9. Ejercicios de autocomprobación
- 4.10. Soluciones de los ejercicios



## OBJETIVOS DOCENTES

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Discutir qué características debe reunir un diseño para que el circuito resultante de su síntesis sea combinacional.
- Diseñar circuitos lógicos combinacionales empleando VHDL, tales como funciones lógicas, multiplexores, sumadores y restadores binarios, y ALUs, describiendo el comportamiento y/o la estructura del circuito.
- Describir un mismo circuito combinacional usando diferentes representaciones (comportamental, estructural o mixta) y empleando diferentes tipos de sentencias, tanto concurrentes como secuenciales.
- Programar y simular bancos de prueba para circuitos combinacionales, y analizar los resultados de la simulación.

## TEMA 5

### REGISTROS Y MEMORIAS

- 5.1. Introducción
- 5.2. Registro de 4 bits
- 5.3. Registro multifunción
- 5.4. Registro de desplazamiento
- 5.5. Register file
- 5.6. Bus bidireccional y memorias
- 5.7. Lecturas recomendadas
- 5.8. Ejercicios de autocomprobación
- 5.9. Soluciones de los ejercicios

## OBJETIVOS DOCENTES

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Discutir la diferencia conceptual entre la operación de reset síncrono y asíncrono. Describir ambas usando VHDL.
- Diseñar usando VHDL registros con las funcionalidades siguientes: reset, carga serie y paralelo, lectura serie y paralelo, y desplazamiento de 1 bit hacia la derecha e izquierda. Programar usando VHDL bancos de pruebas para los registros que testeen las funcionalidades anteriores.
- Discutir las características de los registros siguientes: registro de desplazamiento, registro multifunción, register file y registro triestado.
- Realizar el diseño comportamental y estructural usando VHDL de un register file.
- Programar bancos de pruebas que lean los vectores de test de fichero.
- Diseñar usando VHDL el acceso al bus de varios componentes mediante buffers triestado.
- Diseñar en VHDL memorias RAM y ROM, y buses bidireccionales.

## TEMA 6

# DISEÑO DE LÓGICA SECUENCIAL

- 6.1. Introducción
- 6.2. Diseño de máquinas de estado finito
- 6.3. Síntesis de lógica secuencial
- 6.4. Flip-flop JK
- 6.5. Máquinas de estado finito de Moore
- 6.6. Máquinas de estado finito de Mealy
- 6.7. Máquinas de estado finito seguras
- 6.8. Lecturas recomendadas
- 6.9. Ejercicios de autocomprobación
- 6.10. Soluciones de los ejercicios

## OBJETIVOS DOCENTES

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Diseñar usando VHDL distintos tipos de flip-flop y sus bancos de prueba. Diseñar usando VHDL el circuito latch. Discutir la diferencia entre un flip-flop y un latch.
- Discutir buenas prácticas aplicables al diseño para síntesis de circuitos secuenciales síncronos.
- Diseñar en VHDL máquinas de estado finito de Moore y de Mealy sintetizables, realizando el diseño tanto en base a la descripción de su estructura al nivel de puertas lógicas y flip-flops, como en base a su comportamiento.
- Diseñar y programar en VHDL bancos de pruebas de circuitos secuenciales síncronos. Discutir diferencias conceptuales entre los bancos de pruebas para circuitos secuenciales síncronos y para circuitos combinacionales.
- Discutir qué son las máquinas de estado finito seguras y realizar el diseño de este tipo de máquinas usando VHDL.

## TEMA 7

# METODOLOGÍA DE TRANSFERENCIA ENTRE REGISTROS

- 7.1. Introducción
- 7.2. Operaciones de transferencia entre registros
- 7.3. Máquinas de estado finito con camino de datos
- 7.4. Descripción del programa RT usando VHDL
- 7.5. Circuito detector de secuencia
- 7.6. Control de una máquina expendedora
- 7.7. Lecturas recomendadas
- 7.8. Ejercicios de autocomprobación
- 7.9. Soluciones de los ejercicios

## OBJETIVOS DOCENTES

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Discutir los conceptos siguientes: operaciones RT, notación RT, programa RT y metodología RT.
- Discutir el ciclo de ejecución de una operación RT.
- Discutir los principios fundamentales de la programación RT.
- Dado un algoritmo, traducirlo a un programa RT.
- Describir usando VHDL programas RT sencillos y programar bancos de pruebas para testear estos diseños.
- Discutir los conceptos fundamentales de la síntesis de programas RT mediante FSMD. En particular, discutir la estructura general de la FSMD, cómo se realiza la síntesis del camino de datos y por qué la unidad de control puede sintetizarse como una FSM.