

# **CREACION DE APLICACIONES PARTICIONADAS HW/SW CON FPGAS SYLLABUS**

**Curso Introductorio: Introducción al VHDL, entorno de simulación modelsim y system generator (12 Horas)**

## **Primera Sesión (3 horas)**

1. Introducción
2. Niveles de Abstracción y representación
3. Estructura básica de un archivo VHDL
  - 3.1 Modelamiento Comportamental
  - 3.2 Concurrencia
  - 3.3 Descripción estructural
4. Elementos léxicos de VHDL

## **Segunda Sesión (3 horas)**

5. Objetos en VHDL
  - 5.1 Constante
  - 5.2 Variable
  - 5.3 Señal
6. Tipos de datos
7. Operadores
8. Modelamiento comportamental: Sentencias secuenciales
  - 8.1 Sentencias de asignación: señal o variable
  - 8.2 Sentencias condicionales: IF y CASE.
  - 8.3 Los Bucles: FOR y WHILE
  - 8.4 Ejemplos con ModelSim

## **Tercera Sesión (3 horas)**

9. Sentencias concurrentes.
  - 9.1 Modelamiento Flujo de datos: Sentencias de Asignación de valor a señal
  - 9.2 Asignación Incondicional
  - 9.3 Asignación condicional
  - 9.4 Asignación seleccionada
  - 9.5 Ejemplos con ModelSim
10. Modelamiento Estructural
11. Descripción de Máquinas de Estados Finitos en VHDL
  - 11.1 Maquinas de estados finitos
  - 11.2 Maquina de estados de Moore
  - 11.3 Maquina de estados de Mealy
  - 11.4 Aplicaciones de Maquinas de Estados Finitos

## **Cuarta Sesión (3 horas)**

12. System Generator
  - 12.1 Descripción del System Generator.
  - 12.2 Bloques del System Generator
  - 12.3 Creación de Modelos en System Generator.
  - 12.4 Herramientas y Uso de Herramientas de System Generator.
  - 12.5 Co-Simulación Hardware.
13. Laboratorios con System Generator.

## **Curso Principal: Creación de aplicaciones particionadas HW/SW con FPGAs (34 horas)**

**Abelardo Jara Berrocal**

1. Introducción a los FPGAs de Xilinx: las familias Spartan3, Virtex 2 Pro y Virtex 4 y al flujo de diseño ASICs. Arquitectura de los FPGAs. Descripción de las CLBs de los FPGAs. (1 hora)
  
2. Introducción a la síntesis de alto nivel HLS (high level synthesis) (5 horas)
  - 2.1 Descripción del flujo de implementación: Compilación, Síntesis, Place and Route, Bitstream generation.
  - 2.2 Niveles de arquitectura, lógico y geométrico. Gráfico Y de Kuhn.
  - 2.3 Vistas de modelo y estructural
  - 2.4 Descripción del flujo de síntesis: Planificación (Scheduling), Enlazado (Binding), Síntesis de arquitectura y optimización, síntesis lógica, mapeo a tecnología (Technology mapping)
  - 2.5 Repaso del lenguaje de descripción de HW: VHDL
    - 2.5.1 Modelamiento de circuitos combinacionales
    - 2.5.2 Modelamiento de circuitos secuenciales (maquinas de estado)
  - 2.6 Niveles de modelamiento en VHDL: Behavioral, nivel RTL o gate level. Recomendaciones para modelar a nivel RTL (register transfer level).
  - 2.7 Diseño de sistemas usando cuadros ASM
  
3. Introducción al simulador Modelsim de Mentor Graphics (2 horas)
  - 3.1 Como escribir un testbench para nuestro modelo VHDL
  - 3.2 LABORATORIO 1: Creación de testbenchs y simulación usando Modelsim
  
4. Introducción a las herramientas de síntesis Xilinx ISE (3 horas)
  - 4.1 Archivos de entrada VHDL
  - 4.2 Uso de buffering y DLL para reducir clock skew
  - 4.2 Análisis de timing estático (critical path)
  - 4.3 Estableciendo constraints en tiempo y pines. El archivo .ucfLABORATORIO 2: Implementación de un modular PWM (modulación de ancho de pulso)
  
5. LABORATORIO 3: Implementación del procesador RISC MIPS en un Virtex II Pro (4 horas)
  - 5.1 Creación de un procesador a medida de nuestra aplicación
  - 5.2 La arquitectura del procesador RISC MIPS
  - 5.3 Interface para entrada de teclado PS/2
  - 5.4 Interface para salida de video VGA. Creación de una ROM con caracteres.
  - 5.5 Programación en Assembler de ejemplo: Algoritmo de ordenación Bubblesort y Shellsort
  - 5.6 Integración del sistema completo de lectura de datos y mostrarlos ordenados en el monitor VGA.

6. Técnicas de optimización en diseño con FPGAs para velocidad: definición de throughput. Pipelining, paralelismo, balanceo de lógica, desdoblamiento de bucle (loop unrolling) (2 horas)

7. Técnicas de optimización en diseño con FPGAs para área: compartición de recursos (Resource sharing) (1 hora)

8. Técnicas de optimización en diseño con FPGAs para potencia. Uso de desdoblamiento de bucle (loop unrolling), apagado de lógica (logic shutdown), apagado de reloj (clock gating) y codificación óptima de la máquina de estados (FSM) (2 horas)

9. Usando el Coregen de Xilinx: Creación de memorias y de FIFOs (2 horas)

10. Descripción de plataformas de computación reconfigurable (reconfigurable computers). Las tarjetas Nallatech. Uso del profiling de SW para particionado de aplicaciones HW/SW. Estimación del costo de mapear una parte de la aplicación en HW (enviar esa función al FPGA): tiempo de ejecución y tiempo de comunicación. Comunicación entre la parte de nuestra aplicación en SW y la parte de nuestra aplicación en HW. (3 horas)

11. LABORATORIO 4: Implementación de un generador de señales en FPGA. Creación de una UART para comunicar nuestro SW con el HW. Creación de una interface con la memoria SRAM de la tarjeta. Uso del códec de audio, de la interface RS232 y del banco de memoria SRAM. (3 horas).

12. Teoría de implementación de un filtro FIR en VLSI. LABORATORIO 5: Implementación de un filtro FIR de coeficientes variables en FPGAs. Uso del códec de audio y de la interface RS232. (2 horas).

13. Introducción al diseño modular en FPGAs de Xilinx. Uso de las herramientas del flujo de diseño modular desde línea de consola. Uso de floorplanning y de bus macros para comunicarse entre nódulos. Introducción a la reconfiguración parcial dinámica en Virtex FPGAs. Uso del FPGA Editor de Xilinx.

LABORATORIO 6: Creación de un diseño con una ALU reconfigurable en tiempo de ejecución. (5 horas)