



# Planificaciones

8641 - Sistemas Digitales

Docente responsable: ALVAREZ NICOLAS

## OBJETIVOS

El objetivo de la materia es aprender a diseñar arquitecturas digitales utilizando como herramienta los lenguajes descriptores de hardware. Dichas arquitecturas incluyen a los sistemas digitales base tales como sumadores, multiplicadores, divisores, a los utilizados para el cálculo de funciones trascendentes y a los bloques para procesamiento de señales. Además, se busca lograr la elección de la arquitectura adecuada en base al objetivo de diseño, como puede ser la optimización de la velocidad, el bajo consumo, o la optimización de los recursos.

## CONTENIDOS MÍNIMOS

### PROGRAMA SINTÉTICO

HDL como método de descripción de sistemas digitales (utilización de FPGA).

Arquitecturas de suma, multiplicación, división, CORDIC, unidad de punto flotante. Cálculo de las funciones seno, coseno, arcocoseno, arcotangente, e hiperbólicas utilizando el algoritmo de CORDIC. Cálculo del logaritmo natural. FFT radix-2, radix-2<sup>2</sup>. Especificación de procesadores RISC para sistemas embebidos. RISC 16/32.

### PROGRAMA ANALÍTICO

1. Problemática del diseño digital. HDL como solución a estos problemas. VHDL: origen y razones de su existencia. Introducción mediante el ejemplo de un diseño de un contador de 2 bits. Relación de la entidad de diseño con el encapsulado. El tipo como transporte de la información. Modo de comportamiento: descripción de un circuito en base a su comportamiento, lista de sensibilidad. Encadenado. Entidad de simulación. Modo de comportamiento: suspensión de un proceso, sentencia wait. Descripción de un karnaugh con if-then-else. Especificación de un circuito secuencial utilizando la sentencia case.

2. Conjunto de señales como un arreglo unidimensional. Variable utilizada como auxiliar en la descripción por comportamiento. Instrucción loop: forma general, ejemplo de una compuerta and. Instrucciones next y exit.

Instrucciones concurrentes, process, diseño top-down, instrucción block, instanciación de componentes, instrucción generate, ejemplo de un registro de desplazamiento. Declaración de tipos: enumeración, rangos discretos: enteros, rangos continuos, real, tipos físico, tipos compuestos: estructura y arreglos: con rango fijo y rango abierto. tipos predefinidos. Forma general de la entidad de diseño: Generic como forma de pasar parámetros a la entidad. Ejemplo : descripción completa de un registro de desplazamiento con un parámetro indicando la cantidad de retardos. Funciones: declaración, encabezado, instrucción, return.

3. Procedimientos. Modos: in, out e inout. Operadores lógicos: and, or, xor, nand, nor. Operadores aritméticos: +, -, \*, /, \*\*. Paquete y cuerpo del paquete. Declaración y uso de bibliotecas, paquetes y componentes. Retardos: inerciales para compuertas y de transporte para líneas de transmisión, ejemplos de uso. Forma de onda, elemento de forma de onda, driver de una señal, transacciones, eventos. Ejemplos de una compuerta and. Regla de simulación para circuitos inerciales.

4. Ejemplos de equivalencia entre process y asignación de señales. Regla completa para simulación de señales, ejemplos de reloj. Función de resolución: necesidad: lógica cableada, 1 fuerte 0 débil, firma. Declaración de señales resueltas. Subtipo para señales resueltas. Asignación de señales concurrentes: asignación condicional, equivalencia con el if. Asignación de señal seleccionada, equivalencia con el case. Case. Paquetes: encabezado y cuerpo del paquete. Bibliotecas: uso, acceso a sus componentes. Atributos de los objetos: RANGE, LOW, HIGH, LEFT, RIGHT, EVENT. Ejemplo: Descripción de un contador de N bits con modelo de comportamiento usando variables y atributos.

5. Modo puertos: in, out, buffer e inout. Declaración de configuración. Biblioteca IEEE, paquetes std\_logic\_1164. Tipos std\_logic, std\_logic\_vector, std\_logic\_arith, std\_logic\_unsigned y std\_logic\_signed, numeric\_std. Dimensión de arreglos mediante una constante en los puertos de una entidad. Diagrama básico de relación de las instrucciones. Lógica discreta. Ejemplo de una hoja de datos, tensión de alimentación, niveles lógicos, tiempos de propagación.

6. PLD, CPLD, FPGA.

7. Suma, multiplicación, división, CORDIC, ejemplos.

8. Multiplicador/Divisor con restauración y sin restauración. Números enteros. Multiplicación de números con un

solo operando que puede tomar valores negativos. Multiplicación de números signados: Algoritmo de Booth. Unidad de punto flotante.

9. Cálculo de las funciones seno, coseno, arcocoseno, arcotangente, e hiperbólicas. Cálculo del logaritmo natural.

10. UART.

11. NCO. Oscilador controlado numericamente.

12. Arquitecturas FFT radix-2, radix-2<sup>2</sup>.

13. Arquitecturas RISC en sistemas embebidos.

## **BIBLIOGRAFÍA**

"Computer Architecture A Quantitative Approach", Hennessy & Patterson

"VHDL: Hardware description and design", Roger Lipsett, Carl Schaffer & Cary Ussery

"VHDL: Analysis and Modeling of Digital Systems", Zainalabedin Navabi

"FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version", Pong P. Chu

## **RÉGIMEN DE CURSADA**

Metodología de enseñanza

Clases Teóricas alternadas con clases prácticas y prácticas de laboratorio

Modalidad de Evaluación Parcial

Evaluación por entrega de trabajos prácticos

## CALENDARIO DE CLASES

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<1> 09/03 al 14/03	Lenguaje VHDL	Lenguaje VHDL				
<2> 16/03 al 21/03	Lenguaje VHDL	Lenguaje VHDL				
<3> 23/03 al 28/03	Lógica programable	Lógica programable				
<4> 30/03 al 04/04	Lógica programable	Lógica programable	Implementación de unidades básicas de procesamiento			
<5> 06/04 al 11/04	Sistemas de Entrada Salida		Sistemas de Entrada y Salida			
<6> 13/04 al 18/04	Aritmética					
<7> 20/04 al 25/04	Aritmética de punto flotante	Aritmética				
<8> 27/04 al 02/05	Cordic	Cálculo de rotaciones y diversas funciones trigonométricas				
<9> 04/05 al 09/05	UART					
<10> 11/05 al 16/05	Sigma Delta		Diseño de un voltímetro digital		Aritmética de punto flotante	
<11> 18/05 al 23/05	NCO		Implementación de un NCO y verificación			
<12> 25/05 al 30/05	IP Cores		Inclusión de diversos IP cores en un diseño propio			
<13> 01/06 al 06/06	FFT	FFT				
<14> 08/06 al 13/06	RISC		Manejo de un procesador embebido en una FPGA			
<15> 15/06 al 20/06	RISC		Manejo de un procesador embebido en una FPGA			
<16> 22/06 al 27/06	Implementación de filtros digitales		Implementación de filtros digitales		Receptor serie con visualización VGA	

## CALENDARIO DE EVALUACIONES

### Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º	10	14/05	16:00	L4
2º	16	25/06	16:00	L4
3º				
4º				
Observaciones sobre el Temario de la Evaluación Parcial				
Lenguajes descriptores de hardware y lógica programable. Arquitecturas de procesamiento digital.				
Otras observaciones				
Los temas se evaluarán con dos trabajos prácticos de entrega obligatoria en cada una de las fechas establecidas.				