

CURSO DE POSTGRADO:**Diseño Digital Avanzado**

AÑO: 2019	CUATRIMESTRE: Segundo
CARGA HORARIA: 75 horas	No. DE CRÉDITOS: 3
CARRERA: Doctorado en Ciencias de la Ingeniería	
RESPONSABLE ACADÉMICO: Dr. Ariel L. Pola	

OBJETIVOS:

El objetivo del curso es capacitar al alumno en conceptos avanzados de diseño digital en FPGA y ASIC, de manera de aplicarlos sobre un sistema digital de procesamiento de señales. Además, se busca desarrollar competencia tales como la utilización de un variado espectro de técnicas de diseño y mapeo de algoritmos sobre FPGAs/ASICs usando un lenguaje de descripción de hardware (HDL). Como así también se interiorizará y podrá verificar los distintos circuitos utilizando técnicas de chequeo que le darán criterios para determinar la viabilidad de un proyecto dado ya sea en términos de velocidad, área, verificación y potencia.

PROGRAMA:

Unidad I: INTRODUCCIÓN A VERILOG COMO LENGUAJE DE DISEÑO DE HARDWARE (10 hs).

Introducción. Historia. Flujo de Diseño. Síntesis Lógica. Módulos. Partición del diseño y diseño jerárquico. Valores lógicos y tipos de datos. Los cuatro niveles de abstracción (switch, gate, dataflow y behavioral). Tareas y funciones. Aritmética signada. Verificación en diseño de hardware. Cobertura de código.

Unidad II: MODELO DE IMPLEMENTACIÓN DE SISTEMAS (15 hs).

Flujo de Diseño en Sistemas. Principios. Requerimientos y especificaciones de sistema. Guías de codificación para descripciones comportamentales a alto nivel. Comparación entre arquitecturas de punto fijo y punto flotante. Repaso representaciones numéricas. Representación en complemento a 2. Formatos de punto flotante y punto fijo. Conversión de números de punto flotante a punto fijo. Operaciones en ambos sistemas. Saturación y Overflow. Redondeo y truncamiento. Soporte de Python para punto fijo. Formas de Filtros Digitales. Cuantización de los coeficientes de filtros IIR y FIR. Generación de vectores para verificación de código RTL.

Unidad III: MAPEO DE ARQUITECTURAS DEDICADAS (10 hs).

Sistemas discretos de tiempo real. Sistemas síncronos. Redes de procesamiento tipo Kahn para modelar aplicaciones de streaming. Métodos para representar Sistemas DSP. Diagramas de bloque. Gráficos de flujo de señal. Diagramas de flujo de datos. Single, multi-rate y homogeneous SDFGs. Gráficos de control de flujo. Maquinas de estado finitas. Medidas de rendimiento: Periodo de iteración, periodo de muestro y velocidad de transmisión, latencia, disipación de potencia. Arquitecturas dedicadas.

Unidad IV: BLOQUES BÁSICOS DE DISEÑO EN FPGA Y ASIC (20 hs).

Procesadores embebidos y unidades aritméticas en FPGAs. Instanciación de los mismos. Mapeo óptimo para una tecnología dada. Bloques básicos de diseño en ASICs: Sumadores básicos. Half y Full adder. Ripple carry adder. Sumadores rápidos: Carry Look-ahead Adder, Hybrid Ripple Carry and Carry Look-ahead Adder, Binary Carry Look-ahead Adder, Carry Skip Adder, Conditional Sum Adder, Carry Select Adder,

Hybrid Adders. División con Barrel Shifters. Sumadores Carry Save (CSA) y Compresores. Multiplicadores paralelos. Generación de productos parciales. Reducción de productos parciales. Multiplicadores seccionados. Optimización de Compresores. Contadores de uno o múltiples columnas. Multiplicadores signados en complemento a dos. Eliminación de extensión de signo. Propiedad de cadena. Multiplicador modificado de Booth. Árboles de compresión para sumas multi-operando. Algoritmos para transformar CSA. Multiplicación por constantes. Representación canónica de dígito signado. Arquitecturas dedicadas de filtros FIR en forma directa, transpuesta e híbrida. Aritmética distribuida.

Unidad V: TRANSFORMACIONES PARA ALTA VELOCIDAD, BAJO CONSUMO Y OPTIMIZACIÓN DE ÁREA (10 hs).

Pipelining and Retiming. Conceptos. Métodos: de corte o por el teorema de la transferencia de retardos. Aplicación a sistemas con y sin realimentación. Retiming en herramientas de síntesis. Minimización del número de registros y camino crítico. Descomposición de Shannon. C-Slow Retiming. Plegado y desplegado (Unfolding y Folding) de arquitecturas. Consideraciones sobre la velocidad de muestreo. Técnicas para maximizar uso de árboles de compresión, uso efectivo de recursos de FPGAs. Técnicas de plegado aplicadas a estructuras regulares: filtros y FFTs.

Unidad VI: DISEÑO DE MÁQUINAS DE ESTADO (10 hs).

Ejemplos de arquitecturas multiplexadas en tiempo. Arquitecturas seriales a nivel de bit y palabra. Arquitecturas secuenciales. Máquinas de estado finitas. Codificación de los estados. Máquinas de Moore y Mealy. Guías para la codificación de las máquinas de estado. C-Slow Retiming. Distintas representaciones gráficas de las mismas. Ejemplos de máquinas de estado. Diseño para verificación. Metodología. Métricas de cobertura. Métodos de reducción de potencia.

BIBLIOGRAFÍA BÁSICA:

- *Digital Design of Signal Processing Systems: A Practical Approach*, S. A. Khan, Wiley, 2011.
- *VLSI Digital Signal Processing Systems*, K. Parhi, Wiley, 1999.
- *Logic Synthesis and Verification Algorithms*, G. Hachtel y F. Somenzi, Kluwer Academic Publishers, 2002.

METODOLOGÍA:

Para el dictado del curso se definen tres actividades claramente diferenciadas:

- *Teórico (T)*: exposición de los conceptos teóricos del curso por parte del docente.
- *Práctico (P)*: desarrollo de ejemplos por parte del docente y realización de problemas por parte del alumno. Dichos problemas se realizarán empleando herramientas de síntesis y simulación (Xilinx Vivado WebPack y Python) previamente introducidas por el docente.
- *Laboratorio (L)*: diseño e implementación de bloques digitales por parte del alumno con la supervisión del docente en FPGA.

ARANCELES:

El curso no tendrá ningún costo para alumnos y docentes universitarios.

MODALIDAD DE EVALUACIÓN:

Al finalizar el dictado del curso se realizará un coloquio con presentación de un trabajo final.

ACTIVIDADES DE LABORATORIO:

El curso incluye la realización de simulaciones en computadora para profundizar los conceptos teóricos dictados en las distintas unidades empleando la herramienta Python. Además, síntesis e implementación de bloques digitales en FPGA. El contenido de estas actividades se detallan más adelante.

DISTRIBUCIÓN DE CARGA HORARIA:

Para el dictado del curso se tiene previsto un total de 80 hs reloj con el docente según la siguiente distribución:

CARGA HORARIA DE CLASE CON EL DOCENTE

ACTIVIDAD	HORAS
TEÓRICO (T)	39
PRÁCTICO (P)	10
LABORATORIO (L)	26
TOTAL DE CARGA HORARIA	75

CRONOGRAMA DE CLASES:

El curso se divide en módulos diarios (clases) de 5 hs de duración total efectiva. Se tienen previsto dos modalidades para el dictado:

- *Bimestral*: este modo consiste de dos módulos (clases) semanales.
- *Cuatrimestral*: en este caso se tiene un módulo (clase) semanal.

La elección de la modalidad será definida oportunamente por el docente. En la siguiente página se presenta el cronograma detallado de las clases.

CLASE	UNIDAD	TEMARIO	T (hs)	P (hs)	L (hs)
1	I	Introducción. Historia. Flujo de Diseño. Síntesis Lógica. Módulos. Partición del diseño y diseño jerárquico. Valores lógicos y tipos de datos. Los cuatro niveles de abstracción (switch, gate, dataflow y behavioral).	3	2	-
2	I	Tareas y funciones. Aritmética signada. Verificación en diseño de hardware. Cobertura de código. <i>Introducción a las herramientas de diseño. Conceptos básicos y funciones en Python como lenguaje de simulación.</i>	2	-	3
3	II	Flujo de Diseño en Sistemas. Principios. Requerimientos y especificaciones de sistema. Guías de codificación para descripciones comportamentales a alto nivel. Comparación entre arquitecturas de punto fijo y punto flotante. Repaso representaciones numéricas. Representación en complemento a 2. Formatos de punto flotante y punto fijo.	3	2	-
4	II	Conversión de números de punto flotante a punto fijo. Operaciones en ambos sistemas. Saturación y Overflow. Redondeo y truncamiento. <i>Implementación de las técnicas de conversión de punto fijo en Verilog corroborando el funcionamiento con el simulador en Python.</i>	2	-	3
5	II	Soprote de Python para punto fijo. Formas de Filtros Digitales. Cuantización de los coeficientes de filtros IIR y FIR. Generación de vectores para verificación de código RTL. <i>Evaluación del efecto de cuantización de los coeficientes en filtros FIR e IIR e implementación en lenguaje HDL.</i>	2	-	3
6	III	Sistemas discretos de tiempo real. Sistemas síncronos. Redes de procesamiento tipo Kahn para modelar aplicaciones de streaming. Métodos para representar Sistemas DSP. Diagramas de bloque. Gráficos de flujo de señal. Diagramas de flujo de datos. Single, multi-rate y homogeneous SDFGs.	3	2	-
7	III	Gráficos de control de flujo. Maquinas de estado finitas. Medidas de rendimiento: Periodo de iteración, periodo de muestro y velocidad de transmisión, latencia, disipación de potencia. Arquitecturas dedicadas. <i>Diseño e implementación de un bloque multi-rate en FPGA. Análisis de desempeño.</i>	2	-	3
8	IV	Procesadores embebidos y unidades aritméticas en FPGAs. Instanciación de los mismos. Mapeo óptimo para una tecnología dada. Bloques básicos de diseño en ASICs: Sumadores básicos. Half y Full adder. Ripple carry adder. Sumadores rápidos: Carry Look-ahead Adder, Hybrid Ripple Carry and Carry Look-ahead Adder. <i>Implementación de un procesador embebido.</i>	3	-	2
9	IV	Binary Carry Look-ahead Adder, Carry Skip Adder, Conditional Sum Adder, Carry Select Adder, Hybrid. Multiplicadores paralelos. Generación de productos parciales. Reducción de productos parciales. Multiplicadores seccionados. <i>Implementación de un procesador embebido en conjunto con algunas técnicas de sumadores.</i>	3	-	2
10	IV	Optimización de Compresores. Contadores de uno o múltiples columnas. Multiplicadores signados en complemento a dos. Eliminación de extensión de signo. Propiedad de cadena. Multiplicador modificado de Booth.	3	2	-

CLASE	UNIDAD	TEMARIO	T (hs)	P (hs)	L (hs)
11	IV	Arboles de compresión para sumas multi-operando. Algoritmos para transformar CSA. Multiplicación por constantes. Representación canónica de dígito signado. Arquitecturas dedicadas de filtros FIR en forma directa, transpuesta e híbrida. Aritmética distribuida. <i>Implementación de arquitecturas dedicadas de filtros FIR usando los criterios de diseños de multiplicadores.</i>	3	-	2
12	V	Pipelining and Retiming. Conceptos. Métodos: de corte o por el teorema de la transferencia de retardos. Aplicación a sistemas con y sin realimentación. Retiming en herramientas de síntesis. Minimización del número de registros y camino crítico. Descomposición de Shannon. C-Slow Retiming. <i>Resolución de problemas de tiempo (timing) en bloques típicos de DSP.</i>	3	2	-
13	V	Plegado y desplegado (Unfolding y Folding) de arquitecturas. Consideraciones sobre la velocidad de muestreo. Técnicas para maximizar uso de arboles de compresión, uso efectivo de recursos de FPGAs. Técnicas de plegado aplicadas a estructuras regulares: filtros y FFTs. <i>Aplicación de técnica de plegado sobre filtros.</i>	2	-	3
14	VI	Ejemplos de arquitecturas multiplexadas en tiempo. Arquitecturas seriales a nivel de bit y palabra. Arquitecturas secuenciales. Máquinas de estado finitas. Codificación de los estados. Máquinas de Moore y Mealy. Guías para la codificación de las máquinas de estado. <i>Implementación en FPGA de máquinas de Moore y Mealy.</i>	3	-	2
15	VI	C-Slow Retiming. Distintas representaciones gráficas de las mismas. Ejemplos de máquinas de estado. Diseño para verificación. Metodología. Métricas de cobertura. Métodos de reducción de potencia. <i>Diseño, implementación y verificación funcional de una FSM en conjunto con un filtro FIR.</i>	2	-	3
16	I-VI	Exámen Final	-	-	-
HORAS TOTALES			39	10	26