

# Índice

<b>1. TIPO</b>	<b>1</b>
<b>2. NOMBRE</b>	<b>1</b>
<b>3. UNIDAD ACADÉMICA EJECUTORA</b>	<b>1</b>
<b>4. COORDINADOR</b>	<b>1</b>
<b>5. DOCENTES QUE PARTICIPAN</b>	<b>1</b>
<b>6. PROFESIONALES A LOS QUE ESTÁ ORIENTADO</b>	<b>2</b>
<b>7. CONDICIONES DE INSCRIPCIÓN</b>	<b>2</b>
<b>8. OBJETIVOS Y JUSTIFICACIÓN</b>	<b>2</b>
<b>9. DURACIÓN Y ORGANIZACIÓN</b>	<b>2</b>
<b>10. METODOLOGÍA DE DICTADO</b>	<b>2</b>
<b>11. METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN</b>	<b>2</b>
<b>12. CANTIDAD MÍNIMA Y MÁXIMA DE ALUMNOS ADMITIDOS</b>	<b>2</b>
<b>13. NECESIDADES DE INFRAESTRUCTURA</b>	<b>2</b>
<b>14. CONTENIDOS</b>	<b>2</b>
<b>15. BIBLIOGRAFÍA</b>	<b>3</b>

## **1. TIPO**

Curso de posgrado, nivel Doctorado.

## **2. NOMBRE**

Análisis, Diseño y Verificación en Lógica Programable

## **3. UNIDAD ACADÉMICA EJECUTORA**

Facultad de Ingeniería, Universidad Nacional de Córdoba.

## **4. COORDINADOR**

Dra. Graciela Corral-Briones

## **5. DOCENTES QUE PARTICIPAN**

Dr. Eduardo Boemo, Universidad Autónoma de Madrid.

Dr. Gustavo Sutter, Universidad Autónoma de Madrid.

Dr. Elías Todorovich, Universidad Nacional del Centro de la Provincia de Buenos Aires.

Se adjuntan los CVs de los docentes

## **6. PROFESIONALES A LOS QUE ESTÁ ORIENTADO**

Ingenieros y físicos experimentales de todas las especialidades relacionadas con: Sistemas Digitales, Procesamiento de Señales y Redes de Comunicaciones.

## **7. CONDICIONES DE INSCRIPCIÓN**

Es condición poseer título de ingeniero u otro título universitario a ser considerado por el coordinador. Es necesario también tener conocimientos básicos sobre Sistemas Digitales y Lenguajes de Descripción de Hardware.

El curso no será arancelado.

## **8. OBJETIVOS Y JUSTIFICACIÓN**

El objetivo principal del curso es promover en los estudiantes habilidades para analizar, desarrollar y verificar un proyecto sobre lógica programable empleando las últimas técnicas, herramientas y lenguajes disponibles a la fecha. Este curso también servirá para la adquisición de recursos metodológicos de gran importancia que capacitarán a los asistentes para introducirse en el mundo de la lógica programable y las FPGAs (Field Programmable Gate Array).

## **9. DURACIÓN Y ORGANIZACIÓN**

El curso está programado para tener una extensión total de 40 (cuarenta) horas. Será impartido de manera intensiva a lo largo de una semana desde el lunes 4 al viernes 8 de abril de 2011. El horario de las clases será de 8:00 a 12:00 hs y 13:30 a 17:30.

## **10. METODOLOGÍA DE DICTADO**

Se realizarán clases presenciales teórico/prácticas. Las prácticas de laboratorio ocuparán aproximadamente el 50 % de las horas del curso. Para las prácticas se necesita un laboratorio con computadoras modernas y el software de diseño instalado.

## **11. METODOLOGÍA DE EVALUACIÓN Y APROBACIÓN**

La evaluación se hará con un trabajo final, que podrá ser una pequeña aplicación en FPGA, una monografía vinculada a alguno de los temas de análisis de tiempo o consumo de potencia, o la aplicación de las técnicas de verificación modernas desarrolladas en el curso a un diseño digital.

## **12. CANTIDAD MÍNIMA Y MÁXIMA DE ALUMNOS ADMITIDOS**

Se define un mínimo de 5 alumnos y, debido a la disponibilidad de laboratorios, un máximo de 20.

## **13. NECESIDADES DE INFRAESTRUCTURA**

La parte presencial del curso se podrá dictar en las instalaciones (aulas y laboratorios) de la FCEFyN - UNC.

## **14. CONTENIDOS**

### **Fundamentos de Diseño Digital (20hs)**

- Retardos en circuitos integrados: Retardo intrínseco y extrínseco. Glitches. Skew de reloj. Fallos de Sincronización. Arboles de reloj. Skew máximo admisible.

- Diseño de Alta Velocidad I - Pipelining: Paralelismo y pipelining. Latencia y throughput. Métodos heurísticos de segmentación. Líneas equitemporales. Registros de skewing y deskewing. Granularidad y Profundidad de Lógica. Funciones de costo. Comunicación global y local. Compromiso entre área, velocidad, y latencia. Ley de los Rendimientos Decrecientes. Segmentación e interconexión. Wave pipelining: Fundamentos. Velocidad máxima de operación. Bandas oscuras. Skewed-clock wave pipeline.
- Diseño de Bajo Consumo: Técnicas aplicables a FPGAs: Consumo en circuitos CMOS. Medición de consumo. Potencia y Energía. Influencia de la actividad espuria. Dependencia de los datos. Potencia de sincronización y off-chip. Revisión de técnicas LPD aplicables al diseño FPGA. ATP en FPGAs. Consumo en microprocesadores. Verificación Térmica.

## **Tecnología FPGA y síntesis de alto nivel(20hs)**

- Dispositivos FPGAs actuales. Repaso de familias clásicas, y nuevos dispositivos. LUTs, slices, CLB's, BRAMs, nuevos bloques dedicados.
- Aspectos avanzados en ISE, Análisis de tiempos usando analizador estático de tiempos. Ejemplos de utilización con FPGA editor y Timming Analyzer.
- Síntesis desde alto nivel. Introducción a la computación reconfigurable. Ideas claves de las FPGAs como aceleradores de algoritmos, ideas de reprogramabilidad, evolución, aplicaciones y herramientas. Ejemplos de placas para aceleración en computación de altas prestaciones (HPRC - High Performace Reconfigurable Computing). PCI / PCIe boards, insocket acceleration. Síntesis desde lenguajes de alto nivel (High-Level Languages HLL). Por qué usar HLL? Evolución, tendencias y panorama. Modelos de computación basado en streams. Ejemplos con Impulse-C.

## **Introducción a la Verificación Funcional Moderna (20hs)**

- Simuladores de HDLs y Verificación Funcional. Verificación tradicional. Diseño de testbenches: la unidad bajo test (DUT), drivers y monitores. Análisis de cobertura de código. Ejemplos y casos de estudio.
- HVLs (Hardware Verification Languages. Nuevas metodologías de verificación. Verification Frameworks (VMM, OVM). Verificación Basada en Aserciones (ABV). El lenguaje de aserciones PSL. Uso PSL en herramientas EDA. Verificación Dirigida por la Cobertura (CDV). Metodología de cobertura funcional. Assertion Patterns. Verification Libraries. Open Verification Library (OVL).

## **15. BIBLIOGRAFÍA**

1. Harry D. Foster, Adam C. Krolnik, David J. Lacey, Assertion-Based Design, 2nd edition, Springer, 2004. (ISBN: 1402080271).
2. Harry D. Foster, Adam C. Krolnik, Creating Assertion-Based IP, Springer US, 2009, (ISBN: 1441942181).
3. Bruce Wile, John Goss, Wolfgang Roesner, Comprehensive Functional Verification: The Complete Industry Cycle, Morgan Kaufmann, 2005. (ISBN: 0127518037)
4. Michael Keating and Pierre Bricaud, Reuse Methodology Manual for System-on-a-Chip Designs, 3rd Edition, Kluwer Academic Publishers, 2002. (ISBN: 1-4020-7141-8).
5. J. Bergeron, Writing Testbenches Using SystemVerilog, Springer, 2006. (ISBN-10: 0-387-29221-7).
6. M. Mintz, R. Ekendahl, Hardware verification with SystemVerilog: An object-oriented framework, Springer, 2007. (ISBN-10: 0-387-71738-2).
7. Cadence Designs Systems and Mentor Graphics Inc., Open Verification Methodology User Guide, Version 2.0, 2008, available at <http://www.ovmworld.org>
8. Glasser, Mark, Open Verification Methodology Cookbook, Springer, 2009, (ISBN: 978-1-4419-0967-1)

9. S. Iman, Step-by-step functional verification with SystemVerilog and OVM, Hansen Brown Publishing, May 2008. (ISBN-10: 0-9816562-1-8).
10. Ashenden, Peter, The designer's guide to VHDL, 2nd Edition, Morgan Kaufmann, 2002. (ISBN: 1558606742).
11. Ashenden, Peter J. and Jim Lewis, VHDL-2008 just the new stuff, Amsterdam, Elsevier Morgan Kaufmann, 2008. (ISBN: 9780123742490).
12. Sutherland, Stuart , SystemVerilog for design a guide to using SystemVerilog for hardware design and modeling, 2nd ed. Springer, 2006. (ISBN: 0387333991).
13. Bhasker, Jayaram, A Verilog HDL primer, Star Galaxy, 2005. (ISBN: 0965039161).
14. IEEE Standard for System Verilog-Unified Hardware Design, Specification, and Verification Language, IEEE Std 1800-2009, 2009.