

**Plan 95 Adecuado**

**PROYECTO DE CÁTEDRA 2014**

**TÉCNICAS DIGITALES I**

**Departamento:** Electrónica

**Asignatura:** Técnicas Digitales I

**Carrera:** Ingeniería Electrónica (R)

**Bloque:** Tecnologías básicas.

**Área:** Técnicas digitales

**Nivel:** 3. **Tipo:** anual      Hs/Sem.:4      Hs/Anuales.128

**Año de dictado** 2014

Director de Cátedra: Consejo Departamental de Ingeniería Electrónica

Asociados: Ing. Juan Collazo y Rubén López

Adjuntos: Ing. Pablo Caldirola, Jerónimo Atencio y Salvador Tropea

Auxiliares: Ings. Susana Canel, Juan Grinberg, Roberto Simone, Lucio Martínez Garbino, Federico Roux, Marcelo Volpi

**Planificación de la asignatura**

Fundamentación de la materia dentro del plan de estudios.

Las técnicas digitales son uno de los contenidos expresamente indicados por la Resolución Ministerial 1232/2001 como necesarios para la carrera de Ingeniería Electrónica, lo que no hace sino formalizar una exigencia largamente reconocida por los diversos diseños curriculares de la carrera de Ingeniería Electrónica en las distintas Universidades argentinas.

Si bien Técnicas Digitales I está definida como una asignatura del bloque de Tecnologías básicas en el plan de estudios vigente, tal vez por el enfoque introductorio que se pretendía en el momento de la elaboración del plan 95 (es decir hace más o menos 20 años), es evidente que a la luz de la evolución tecnológica experimentada en dicho lapso, la cual no solo se limita a los procesos tecnológicos de construcción de circuitos integrados que permiten en la actualidad disponer a costos accesibles de circuitos lógicos programables de muy alta densidad construidos con transistores de 22 nm, el diseño de circuitos lógicos ha ganado un enorme y muy diverso campo de acción estimulado también por el acompañamiento de potentes herramientas CAD, las cuales se basan en su descripción y que permiten su implementación en los Lenguajes de Descripción de Hardware. Por tal motivo este extenso y diverso campo de aplicaciones (que puede llegar hasta el procesamiento Digital de Señales describiendo el hardware en las herramientas mencionadas), hace que debamos revisar el criterio del plan de estudios, e intentar a lo largo del ciclo lectivo, mutar de manera gradual pero sostenida, el clásico enfoque introductorio (apto tal vez para los temas más básicos que se abordan inicialmente) que lleva a un perfil

de asignatura de tecnología Básica, a un enfoque orientado al diseño de sistemas digitales, pasando gradualmente al perfil de una asignatura de Tecnología Aplicada.

En realidad, Técnicas Digitales I, II y III son nombres de fantasía, que bien podrían ser reemplazados por Diseño de Sistemas Digitales, Sistemas Embebidos y Procesadores de Propósito General.

Propósito/Objetivo

De acuerdo con la Ordenanza 1077 que crea el actual Plan de estudios 1995 adecuado, el objetivo de la asignatura es:

Proveer al alumno de conocimientos de lógica simbólica, circuitos combinacionales y secuenciales e introducirlo al Hardware básico de los microprocesadores.

Dicho objetivo se enmarca en los siguientes objetivos fijados por la misma Ordenanza para el Área de Técnicas Digitales

- Adquirir herramientas matemáticas para el estudio de sistemas con variables discretas, con conocimiento fluido de la informática.
- Analizar los principios de operación de circuitos binarios básicos.
- Adquirir capacidad para integrar circuitos en sistemas digitales.
- Adquirir y aplicar la capacidad para el diseño de sistemas basados en microprocesadores, con sus interfases digitales y analógicas.
- Capacitarse en el diseño de instrumental digital, como así también en el procesamiento y la transmisión de señales digitales.

Se considera algo imprecisa y escueta la redacción del objetivo de la asignatura tal como se encuentra en la citada Ordenanza, y se propone como objetivo más detallado que, tras la aprobación del curso, al haber abordado los aspectos teórico-prácticos relacionados con las bases de los circuitos digitales y lenguajes de descripción de hardware, los alumnos hayan desarrollado las siguientes competencias:

- Comprender los aspectos relacionados con circuitos combinacionales básicos y máquinas de estado secuenciales sincrónicas.
- Comprender el funcionamiento de las memorias utilizadas en los sistemas digitales que se emplearán en los sistemas que se abordarán en las asignaturas subsiguientes del Área.
- Comprender la estructura interna y funcionamiento de los dispositivos de lógica programable de alta escala de integración (VLSI), así como los estándares de interconexión de las diferentes tecnologías circuitos integrados digitales.
- Manejar fluidamente los lenguajes de descripción de hardware y sus herramientas de desarrollo para simular sistemas digitales e implementarlos sobre los dispositivos de lógica programable señalados en el apartado anterior.
- Diseñar utilizando lenguajes de descripción de hardware los diferentes módulos digitales componentes de un sistema de cómputo, relacionando las competencias en descripción de hardware con el funcionamiento interno de un sistema de cómputo (embebido o de alta performance)
- Comprender el funcionamiento interno y los componentes más simples de un sistema embebido a través de su descripción.
- Analizar el efecto de la frecuencia en una señal digital, introduciendo el concepto de integridad de señal.

- Comprender los códigos de verificación de integridad de información necesarios para asegurar la correcta transmisión de información en un sistema digital.

### **Contenidos mínimos (programa sintético)**

De acuerdo con la Ordenanza 1077 que crea el actual Plan de estudios 1995 adecuado, el programa sintético de la asignatura es el siguiente:

- 1 Lógica combinacional.
- 2 Lógica secuencial.
- 3 Estructura de buses.
- 4 Introducción a las memorias semiconductoras.
- 5 Introducción a los lenguajes descriptores de hardware.

Programa analítico

De acuerdo al programa sintético recién mencionado, y a los objetivos que se expresan en este Plan de Cátedra y en el plan específico del Área Técnicas Digitales de la carrera, se implementará el siguiente programa analítico.

#### ***Unidad 1 – Códigos***

Enfoque algorítmico de la codificación binaria para bloques de bits. Códigos continuos y cíclicos: Gray. Rápida revisión de codificación de las magnitudes binarias y de los números enteros signados: convenios SYM, Binario desplazado, Complemento a 1 y 2. Suma y resta de magnitudes y enteros. Códigos BCD natural empaquetado y desempaquetado, ASCII y siete segmentos. Descripción algorítmica orientada a su futura implementación en VHDL de Códigos detectores y correctores de errores de Distancia mínima 2, 3 y 4, Checksum, y CRC.

#### ***Unidad 2 - Álgebra de conmutación***

Variables y funciones lógicas. Principales postulados de (Boole, Huntington). Principio de dualidad. Teorema general de los minitérminos y de los maxitérminos. Leyes de De Morgan y de Shannon. Tablas de verdad, Simplificación de funciones lógicas. El mapa de Karnaugh. Su uso para representar y simplificar funciones lógicas sencillas. Expresiones tipo producto de sumas y suma de productos. Redundancias. Introducción a las Look-up tables como método alternativo. La ROM como circuito combinacional. Referencia introductoria a la estructura de un Dispositivo lógico programable.

#### ***Unidad 3 - Lógica combinacional con compuertas***

Compuertas básicas: AND, OR, NOT, y XOR. Tablas de verdad y compuertas derivadas (NAND, NOR). BUFFER, y de transmisión. Simbologías. Implementación de circuitos combinacionales básicos mediante tablas de verdad y Mapas de Karnaugh. Redundancias, don't care. Limitaciones de este método para implementación de sistemas de complejidad media y alta.

#### ***Unidad 4 - Circuitos combinacionales esquemáticos***

Diseño de circuitos combinacionales usando el modo esquemático de la herramienta de síntesis (ISE/Quartus). Vistas RTL y tecnológica. Vista del ChipPlanner. Introducción al concepto de Banco de pruebas (Testbench). Herramientas Isim y Model Sim. Simulación de comportamiento y temporal. Demoras y respuestas transitorias. Glitches. El tercer estado de la salida (HI-Z). Bloques combinacionales básicos a nivel RTL (multiplexor, decodificador, demultiplexor, codificador con y sin prioridad, comparador, sumador, restador, generador de paridad, barrel shifter). Uso de la herramienta TimeQuest Timing Analyzer para determinar el tiempo de propagación en caminos combinacionales. Diseño

combinacional a nivel RTL. Circuitos iterativos de 1 dimensión. Ejemplo con un incrementador. Circuitos iterativos de 2 dimensiones. Ejemplo con un multiplicador combinacional. Demos de Place and Route.

### ***Unidad 5 – Dispositivos lógicos programables y configurables***

Dispositivos programables completos (PROM, EPROM y EEPROM). Dispositivos programables incompletos (PLA, PAL y GAL-PALCE). Diferencia entre una CPLD y FPGA. Introducción y estructura de una FPGA. Arquitectura básica. LUTs. Concepto de recurso de ruteo.

### ***Unidad 6 - Introducción a VHDL***

Lenguajes descriptores de hardware. Génesis, evolución, ventajas, aplicaciones. El lenguaje VHDL. Características lexicográficas y de sintaxis. Lógica multinivel. Concepto de código concurrente. VHDL. Entidad. Arquitecturas. Objetos (signal, variable y constant). Tipos de datos std\_logic y std\_logic\_vector, Tipos de datos definidos por el usuario. Empleo de operadores predefinidos (lógicos y relacionales). Asignaciones a señal simple, condicionada y seleccionada. Descripción de circuitos iterativos (sentencias GENERATE e IF GENERATE). Vectores, indexación y concatenación. Bibliotecas y paquetes estandarizados (IEEE – 1164 y Numeric\_Std). Procesos. Sentencias IF, CASE y FOR LOOP. Descripción por flujo de datos, por comportamiento y estructural. Componentes y su instanciación. Descripción de circuitos genéricos (parametrizables). Verificación de la operación de circuitos descritos en VHDL utilizando bancos de prueba y software de simulación.

### ***Unidad 7 - Circuitos combinacionales***

Empleo de sentencias when-else y with-select y Signals en la implementación VHDL de los circuitos combinacionales básicos: Multiplexores, Demultiplexores, Decodificadores, Codificadores con y sin prioridad, Sumador completo de 1 bit, Comparadores de magnitud, Circuito conversor de binario a Gray y de Gray a binario, circuitos detectores y correctores de error: paridad, checksum, CRC, y Hamming, Decodificador BCD a 7 segmentos, y Buffer Tri-State. Verificación de los recursos utilizados en la FPGA por cada uno de los circuitos anteriores. Uso de process para descripción de circuitos combinacionales. Variables. Diferencia entre una variable y una signal. Lista de sensibilidad. Uso de las sentencias if, case, for, y loop. Código concurrente versus secuencial.

### ***Unidad 8 - Circuitos aritméticos***

Operaciones aritméticas de números en CA2 (suma, resta, multiplicación, división) Comparador de números en CA2 Implementación de circuitos aritméticos. Biblioteca numeric\_std. Esbozo de los sumadores multioperandos. Árboles de Wallace y Dadda. Sumadores rápidos. Carry look-ahead y conexiones en árbol. Esbozo de los sumadores Brent-Kung, Ladner-Fischer, Kogge-Stone y Ling. Carry Skip, Carry Select y su aplicación a los FPGA. Algoritmo de Booth. Evaluación de funciones matemáticas mediante tablas. Operaciones con un operando constante.

### ***Unidad 9 – Circuitos secuenciales***

Sistemas secuenciales. Latches SR y D. Flip-flops. Amo esclavo. Entradas de fuerza. Disparo por flanco. Parámetros temporales, ventana de muestreo (apertura temporal) y metaestabilidad. Lectura sincrónica de las entradas de control. Análisis y síntesis con componentes estándar. Síntesis con FPGA usando los modos esquemáticos y HDL del sintetizador. Código en 2 bloques para separar los flip-flops (registros) de la lógica de estado futuro. Máxima frecuencia de operación. Restricciones temporales (Timing constraints): setup y hold, skew y jitter . Clock skew dentro y fuera de un chip. Uso de la herramienta

TimeQuest Timing Analyzer para determinar la máxima frecuencia de operación de un circuito, determinación de violaciones al tiempo de setup y hold, e imponer, en base a estas mediciones, restricciones a un diseño.

Circuitos secuenciales regulares y no regulares. Contadores de modo simple y multimodo *sin conversores de salida* (binarios, Gray y BCD Natural). Clock enable y gated clock. Registros PIPO y de desplazamiento (SISO, SIPO, PISO y multimodo).

Estudio de los registros de acuerdo a su capacidad de almacenamiento: registros individuales, bancos de registros, RAM. Instanciación de la SRAM interna de un FPGA y análisis del diagrama de tiempos para su acceso.

Contadores basados en registros (Anillo, Johnson y LFSR – Linear Feedback Shift Register). Sincronizadores de entradas de control asincrónicas. Fast Input y fast output registers en un FPGA. Introducción a los PLL de un FPGA.

Biestables redispares como osciladores. Operación de osciladores MOS como el CD4538.

### ***Unidad 10 – Circuitos secuenciales sincrónicos - Maquinas de estado.***

Análisis de circuitos secuenciales sincrónicos (máquinas de estado o FSM). Máquinas Mealy y máquinas Moore, ventajas y desventajas relativas. Tabla de estados y diagrama de estados. Síntesis de circuitos secuenciales sincrónicos. Reducción del número de estados, asignación de los mismos.

El problema de las entradas asincrónicas. Metaestabilidad. Sincronizadores. Descripción VHDL de las máquinas de estado.

Implementación (softcore) en FPGA de un microprocesador RISC sencillo.

### ***Unidad 11 – Camino de datos***

Metodología de transferencia de registros. Transformación de un pseudocódigo en una máquina de estados con camino de datos. Operaciones básicas a nivel RTL. Microoperaciones y microcódigo. Camino de datos. Diagramas RTL del camino de datos. Camino de control. Análisis de performance. Optimización para reducir el tiempo de procesamiento. Ejemplos de circuitos aritméticos multiplicadores y divisores. Extractores de raíz cuadrada. Interfaces de comunicaciones UART, SPI, I2C y PS2. Conversores A/D. Breve estudio de una síntesis usando una tabla de microcódigo. Microsecuenciador.

### ***Unidad 12 - Bancos de prueba avanzados***

Procedimientos y funciones. Manejo de archivos, lectura y escritura de líneas, lectura y escritura de elementos a una línea. Ejemplos. Manejo de tipos IEEE en archivos. Demoras en señales, modelos por transporte e inercial, asignación generalizada. Circuitos típicos: generador de reloj, generador de reset, esqueleto de un banco de pruebas sincrónico. Ejemplos de bancos de prueba: boundary testing, comparación con un modelo de referencia en archivo, estímulos y referencia desde un archivo, uso de entrada y salida estándar para verificación con *pipes*.

### ***Unidad 13 – Softcores***

Introducción a los conceptos de microprocesador, microcontrolador y soft/hard-core. Conceptos básicos para comprender la arquitectura de una CPU: memoria, registros, ALU, instrucciones y opcodes, pipeline, modos de direccionamiento.

Generalización y simplificación de una máquina de estados con camino de datos. Análisis de un firmware simple que multiplica 2 magnitudes binarias.

Registro acumulador y contador. Memoria de programa. Registro de instrucciones. Búsqueda (fetch) y ejecución. Diagrama de estados. Criterios para asignar códigos de operación y minimizar el tiempo del camino de datos. Puertos de entrada y salida. Concepto de bootloading.

Softcore de ejemplo con memorias de programa (13 bits) y datos separados (8 bits), un único registro, dos modos de direccionamiento y ciclos de máquina de 3 pulsos de reloj. Formato de las instrucciones, operaciones (load, store, jump, and, xor, add y sub). Diagrama a nivel de flip-flops, registros, multiplexores, compuertas, etc. de toda la CPU y su implementación en VHDL. Banco de prueba de la CPU, formas de onda.

Partición de un diseño digital en hardware y software.

Introducción al procesador NIOS II. SOPC Builder. NIOS EDS y Eclipse.

### **Unidad 14 – Tecnologías y Estándares de Hardware Digital**

Breve referencia a tecnología TTL a modo referencial del origen de lo actual, solo a nivel de hojas de datos y su interpretación.

Articulación Horizontal con Dispositivos Electrónicos: El transistor NMOS y CMOS como switch. Compuertas NMOS y CMOS. Diferentes implementaciones.

Característica de transferencia de un inversor. Parámetros de tensión. Márgenes de inmunidad al ruido (estático y dinámico). Tensión de umbral. Dispositivos con entrada Schmitt-trigger – histéresis. Parámetros de corriente. Condiciones para la interconexión de dispositivos. Parámetros temporales. Estimación de la potencia disipada en dispositivos CMOS y su dependencia de la frecuencia y la tensión de alimentación.

Estándar JESD8 – LVTTTL y LVCMOS. Interoperabilidad e interpretación de hojas de datos. Rise time y ancho de banda de una señal digital. Influencia de la inductancias parásitas en las pistas de alimentación y de datos. Ringing. Fuentes de alimentación interna y de los bloques de E/S, de un FPGA. Capacitores de desacoplo y filtros de PLL. Entradas de reloj y osciladores integrados (CAN oscillators). Conexiones para programación JTAG y propietarias. Memorias de configuración.

Interpretación de las hojas de datos de dispositivos FPGA.

### **Metodología de Enseñanza.**

Descripción de las actividades teóricas y prácticas

#### **Teoría**

Las clases teóricas consistirán en exposiciones por parte del profesor, complementada con recursos de práctica siempre que su disponibilidad lo permita y oportunidad respecto del tema lo amerite. Se procurará estimular en el alumno la participación en clase que estimule su curiosidad por aprender y el desarrollo del necesario juicio crítico indispensable para un Ingeniero de excelencia.

En lo posible se tratará de organizar las clases en secciones teóricas seguidas de prácticas que apliquen directamente los conceptos vistos anteriormente. Por otra parte, se procurará que cada tema haga referencia y se sustente en los temas ya abordados tanto en el dictado de la asignatura como en las asignaturas previas relacionadas. Al respecto cada vez que se requiera de un tema que corresponda a otra asignatura se requerirá a los alumnos que lo traigan debidamente refrescado lectura mediante, para la clase en la que se aplicará.

#### **Práctica**

Para fortalecer la uniformidad de dictado en los seis cursos de la asignatura, se trabajará con una guía de Trabajos Prácticos unificada. Los problemas y requerimientos de diseño que componen la misma serán elaborados en conjunto por todos los docentes de la cátedra (profesores y auxiliares), quienes enviarán sus propuestas a la Dirección de Cátedra quien las unificará y les dará coherencia en una Guía que estará disponible al inicio del ciclo lectivo. La Guía deberá contar con una variada cantidad de problemas. Algunos de los problemas servirán de mero entrenamiento intelectual, mientras que otros abordarán aspectos más relacionados a la vida profesional.

La Guía deberá tener un capítulo por cada capítulo del plan analítico, en donde los problemas propuestos sean de complejidad creciente. Se recomienda para el caso de los Trabajos Prácticos más asimilables a requerimientos de la vida profesional, que éstos sean desagregados en una serie de problemas interrelacionados (cada problema se basa en el resultado del anterior) para llegar a la

solución final. Ese último Trabajo Práctico que contenga la solución requerida al problema completo, será el entregable que se le requiera al alumno.

La resolución de los trabajos prácticos es responsabilidad individual de cada alumno.

En el caso de conceptos teóricos cuya entidad no amerite tiempo de exposición por parte del profesor, se dejan para que el alumno los aborde resolviendo un problema de la guía.

Se cubrirán los siguientes capítulos en la guía de Trabajos Prácticos:

*TP 1 – Códigos.*

Problemas clásicos de cambio de sistema de numeración, operaciones aritméticas con números binarios, números en Ca1 y Ca2.

Problemas sobre conversión de binario a BCD y viceversa. Código Gray. Bits de paridad. Conversión D/A y A/D.

*TP 2 – Álgebra de Boole y mapeo tecnológico.*

Comienza con problemas sobre simplificación de expresión usando solo las leyes del álgebra combinacional. También se incluyen problemas sobre dualidad de funciones. Luego se aborda la simplificación con mapas de Karnaugh. Implementar funciones simples con conmutadores ideales. Mapeo tecnológico multinivel con celdas predefinidas. Ingreso de código en ISE o Quartus para analizar su vista tecnológica. Implementación de funciones muy simples con LUTs. Implementación de funciones en una ROM.

*TP 3 – Circuitos combinacionales esquemáticos.*

Problemas de diseño de circuitos esquemáticos con mapas de Karnaugh. Síntesis y simulación en ISE+ISim o Quartus+Model Sim: MUX, DECO, Conversor bin-Gray, Gray-bin, comparadores, generadores de paridad, barrel shifter. Circuitos iterativos: incrementador y multiplicador. Duplicador y divisor por 2 bcd, conversor bin-bcd, bcd-bin. Sumadores e indicadores de desborde.

*TP 4 – Introductorio – VHDL.*

Problemas de codificación de una entidad. Operaciones elementales. Decodificación bcd-7seg. Desplazamientos y rotaciones con el operador de concatenación. Multiplexores. Codificadores con prioridad. Process y memoria inferida. Aritmética de suma e incremento. Operador multiplicación con FPGAs con y sin multiplicadores nativos.

*TP 4 – Laboratorio – VHDL (diseños completos, con testbench y pruebas en kit).*

Problemas para inferir estructuras tipo árbol. Circuitos aritméticos de suma y resta con indicadores de desborde. Incrementador iterativo Gray. Comparadores. Decodificación bcd-7seg en lógica positiva y negativa. Codificador con prioridad. Uso de variables en un tribunal de mayorías. Duplicador y divisor por 2 bcd, conversor bin-bcd, bcd-bin. Ensayo de los problemas en los kits DE1 de Altera y Spartan 3E de Xilinx.

*TP 5 – Circuitos secuenciales.*

Problemas de análisis de circuitos de clock gating. Aplicación de las tablas de transición y excitación de los flip-flops. Síntesis y simulación en ISE+ISim y Quartus+ModelSim de los distintos tipos de flip-flops: con reset asíncrono, síncrono, enable. Análisis de las vistas RTL, tecnológicas y FPGA Editor o ChipPlanner. Bancos de registros. Usog de SRAM interna del FPGA.

*TP 6 – Parte 1 – Máquinas de estados esquemáticas (diseño clásico).*

Problemas de análisis de un contador bidireccional. Generador/Verificador de CRC. Método de las particiones para reducir la cantidad de estados. Detector de patrones con máquinas Mealy y Moore. Estudio de un sincronizador. Decodificador NRZI de una línea USB. Decodificador Manchester. Árbitro de un recurso con señales request y grant.

*TP 6 – Parte 2 - Máquinas de estados en VHDL (diseño y testbench).*

Problemas de diseño de un generador/verificador de CRC. Uso de atributos en la herramienta ISE o Quartus para la asignación de estados. Templates de sincronizadores. Generador de patrones binarios. Antirrebote de un pulsador. Máquina de estados de una alarma de automóvil. Síntesis y simulación con ISE+ISim y Quartus+ModelSim.

*TP 7 – Parte 1 – Contadores (diseño clásico).*

Problemas de diseño de contadores BCD. Contador de horas, minutos y segundos. Contadores de módulos arbitrarios. Análisis de un máquina de estados basada en un contador.

*TP 7 – Parte 2 – Contadores en VHDL ( diseño y testbench ).*

Problemas de diseño de contadores con enable y con reset sincrónico. Contadores de módulo arbitrario. Generador de PWM. Timer de 1 segundo con decodificación bcd-7seg. Síntesis y simulación con ISE+ISim y Quartus+ModelSim.

*TP 8 – Registros.*

Problemas de registros con desplazamiento bidireccional, carga paralelo y retención. Análisis de un circuito con clock skew ( violación del tiempo de hold ). Registros SIPO. LFSR. Análisis de una máquina de estados basada en un registro. Scrambler.

*TP 9 – Estándares digitales.*

Problemas sobre cálculo de consumo de circuitos CMOS. Cálculo de inmunidad al ruido. Determinación de formas de onda de las salidas de circuitos considerando su tiempo de propagación. Entradas schmitt-trigger. Circuitos internos. Compatibilidad entre familias.

*TP 10 – Datapath ( en vhdl, algoritmo, RTL, vhdl, testbench y kit ).*

Problemas de diseño de multiplicadores por sumas reiteradas y por sumas y desplazamientos. Diseños de divisores por restas sucesivas y restas y desplazamiento. Diseño de extractor de raíz cuadrada. Diseños de transmisores y receptores UART, SPI, I2C y PS/2. Diseño de conversor A/D por contador ascendente y por aproximaciones sucesivas. Síntesis y simulación con ISE+ISim y Quartus+ModelSim. Ensayo de los problemas en los kits DE1 de Altera y Spartan 3E de Xilinx.

*TP 11 – Softcores.*

Problemas de diseño de un procesador c/arquitectura Harvard. Diseño de instrucciones LOAD, ADD, SUB, STORE, INC, DEC, JUMP, JNZ. Diseño de un decodificador de código de operación para optimizar en velocidad el datapath. Diseño de puertos de entrada y salida. Uso de un compilador suministrado por el docente. Implementación de un procesador NIOS con el SOPC Builder y el NIOS EDS Eclipse. Implementación del bootloader. Síntesis y simulación con Quartus+ModelSim. Ensayo de los problemas en los kits de Altera, y Xilinx.

En caso que se verificara que no es temporalmente posible resolver la completitud de la guía, podría eximirse a los alumnos de resolver algunos trabajos prácticos, como el TP 7 – 1 y/o el TP 8 y/o TP 9. Esta decisión se tomará en conjunto y cada docente deberá informar a la dirección de cátedra acerca de esta limitación. En caso que se elimine la obligatoriedad de entrega de estos Trabajos Prácticos, el docente entregará en ese caso problemas resueltos de esa guía como también brindará solución a determinados problemas de otras guías como material de apoyo.

En lo posible se dedicará tiempo de clase para la resolución de los problemas, dejando para el hogar su finalización y la elaboración de los informes. Respecto a esto último, serán en formato electrónico, el cual será subido a un enlace del aula virtual o repositorio svn o git que utilice cada curso.

En ese mismo repositorio / aula virtual el docente actualizará un listado por cada TP, con los alumnos que aprobaron o adeudan correcciones. También subirá un documento por alumno con las observaciones realizadas para cada trabajo práctico.

La resolución de cada trabajo práctico consta de un informe en formato pdf, con una carátula y la resolución en sí misma. Se admitiría un escaneo de una resolución manuscrita a condición de que sea prolija. Varios trabajos prácticos requieren además subir un proyecto en la herramienta de síntesis Quartus o ISE, asignada por el docente.

Como se mencionó, la resolución de estos problemas tiene un aspecto cooperativo. Resulta imposible que un alumno resuelva las guía de problemas completas y que los docentes la corrijan. Para ello se le asigna a cada alumno un problema de la guía de modo que en conjunto una clase haya resuelto toda la guía.

Si es posible, cuando 2 alumnos tienen asignado un mismo problema (en el caso de un curso muy numeroso) se le asignará una herramienta diferente a cada alumno (Quartus o ISE) para que se aprecien las diferencias entre ambas. Y para cada alumno en particular se buscará que cambie de herramienta en cada trabajo práctico de modo que haya utilizado en forma equilibrada ambas herramientas durante el año de cursada.



Las resoluciones aprobadas de los trabajos prácticos son subidas al aula virtual de modo que sirvan de material suplementario de estudio para los exámenes.

Todos los Trabajos Prácticos deben estar aprobados antes de la última fecha de examen final de diciembre.

Los informes deberán estar aprobados antes de la última fecha de llamado a examen del mes de diciembre. Sólo puede quedar pendiente para el mes de febrero la recuperación de exámenes parciales.

La proporción del *tiempo de clase* asignado a la realización de trabajos prácticos de aula es del orden del 40% como mínimo. El tiempo efectivo de realización de trabajos prácticos es mayor que el que surge de esta cifra, porque los problemas que se someten a los alumnos no pueden resolverse totalmente en el tiempo asignado en el aula, el que requiere ser complementado con tarea en el hogar. El tiempo de aula es utilizable por los alumnos para avanzar en la realización del práctico, la discusión grupal del enfoque o las alternativas, y la consulta con el profesor.

### Proyectos

La asignatura pretende que el alumno tenga una sólida formación de prototipado en FPGAs. Para ello se solicita a los alumnos que realicen proyectos anuales en forma grupal. La integración de los grupos es decidida por los propios alumnos y se prefiere que no tengan más de 4 integrantes y no haya más de 6 grupos, todo dependiendo la cantidad de asistentes al curso.

El tema del proyecto siempre consiste en realizar un diseño sobre un FPGA, el cual es asignado por el docente. La intención es que el mismo este totalmente funcional al finalizar la cursada y que todos los integrantes hayan participado en su confección.

Se asignan tareas a cada integrante de los grupos para hacer un seguimiento de la evolución del proyecto por parte del docente. A su vez se dedica tiempo cada 3 clases, aproximadamente, para que los alumnos soliciten instrumental, y comenten con los docentes las experiencias y dificultades que han tenido.

### **Metodología de Evaluación.**

#### Modalidad (tipo, cantidad, instrumentos)

Los exámenes parciales son individuales, y consisten en preguntas teórico prácticas que ante una especificación requieren su resolución práctica con su correspondiente justificación teórica, la cual deberá ser lo más exhaustiva posible. La evaluación consiste en dos exámenes parciales:

- El 1º parcial trata sobre las unidades 1 a 9, pudiendo esta última ser abarcada en forma total o parcial, quedando los temas que no hayan llegado a dictarse para el segundo parcial. El grado de completitud de la Unidad 9 se trata con esta flexibilidad en virtud de no contar todos los cursos con la misma cantidad de clases en virtud de feriados y demás eventualidades, que varían de un ciclo lectivo al siguiente.
- El 2º parcial abarca el resto de los temas. Para este ciclo lectivo se propondrá que aquellos cursos que tengan capacidad de implementarlo adecuadamente, cambien el parcial en modalidad clásica por un proyecto integrador individual que consista en la implementación de un softcore mínimo como el detallado en el programa analítico, con algunas mejoras que el alumno deberá proponer y la integración de algún(os) del(os) dispositivo(s) periférico(s) mencionados en el programa analítico

2

propuesto, tales como interfaces UART, SPI, o I C. El entregable es el código VHDL ejecutando correctamente en el kit que el alumno seleccione. Además deberá estar acompañado de la documentación correspondiente, que constará en un informe técnico junto con un capítulo en el que se describe el funcionamiento del proyecto con referencia permanente a los contenidos teóricos de la asignatura.

#### Recuperatorios:

Los recuperatorios se tomarán tras la finalización de las clases en noviembre y antes de la fecha tope que fije la Facultad para dar por aprobados los trabajos prácticos. Se tendrán dos oportunidades de

recuperación por cada parcial, una en Diciembre y la otra en Febrero/Marzo. Las mismas estarán intercaladas con las tres fechas de finales de Noviembre-Diciembre y con las correspondientes al período Febrero-Marzo.

A fin de establecer un punto adicional de homogeneización de contenidos, el tema de cada recuperatorio será el mismo para todos los cursos y deberá estar elaborado en forma rotativa por los docentes con curso a cargo, quienes lo enviará para su validación a la Dirección de Cátedra, de acuerdo con la mecánica de interacción entre docentes de la asignatura que se detallará más adelante.

Requisitos de regularidad:

La Facultad establece como requisito para no perder la condición de alumno regular un mínimo de asistencia, cuyo control está a cargo de la bedelía.

Requisitos de aprobación:

Para la aprobación de los trabajos prácticos se debe aprobar todos los informes de trabajos prácticos y los dos exámenes parciales aprobados.

Para la aprobación de la materia se debe aprobar (con una calificación mínima de 4) un examen final que tendrá características similares a los parciales en cuanto al formato, pero cuyos casos a resolver tendrán un sesgo integrador de todos los contenidos. La capacidad de diseñar una solución para el requerimiento propuesto en el tema será una de las competencias más valoradas junto con el dominio conceptual y teórico práctico de los contenidos.

Al igual que en las instancias de recuperación de parciales, el tema del examen final será elaborado en forma rotativa por los docentes con curso a cargo con la previa supervisión y aprobación por parte de la Dirección de Cátedra.

Recursos didácticos a utilizar como apoyo a la enseñanza.

Proyectors y recursos de laboratorio en la medida de su disponibilidad por parte del Departamento de Electrónica o de la Facultad.

Articulación horizontal y vertical con otras materias

La articulación vertical se implementa fundamentalmente a través de las reuniones de Directores de Cátedra del área de Técnicas Digitales (asignaturas Informática I y II, y Técnicas Digitales I, II y III).

La articulación vertical hacia los niveles anteriores del plan de estudios se apoya en las cátedras de Informática 1, de la que se toman los conocimientos de sistemas de numeración, y de Informática 2, en donde es muy útil la experiencia de los alumnos en el manejo de un entorno de desarrollo integrado (IDE) complejo como el Eclipse o QT Designer, y especialmente el manejo desde el software de varios periféricos básicos de comunicación como UART, SPI, I2C, por ejemplo, los cuales estamos abordando en nuestra asignatura a nivel de diseño de hardware con lo cual completamos el análisis de estos dispositivos. Por otra parte, los conocimientos de lenguajes de programación y la práctica intensiva de desarrollo de software adquiridos a través de ambos cursos, resultan útiles a la hora de elaborar los bancos de prueba, y son importantes como marco conceptual para abordar el tema datapath y procesadores por software.

La articulación vertical hacia los niveles siguientes de la carrera es el desafío de esta asignatura, y consiste en brindarle a la asignatura Técnicas Digitales 2 alumnos experimentados en el diseño de los bloques básicos de los sistemas embebidos que utilizarán en el microcontrolador NXP1679, muchos de los cuales han sido descritos en VHDL lo cual ha permitido comprender su funcionamiento. También para que los puedan emplear con un herramienta de integración de IP (intelectual properties) como ser Qsys+NIOS IDE, en el diseño de SOCs ( System On-a-Chip ). Como producto de la última reunión de área se ha implementado además el diseño de un softcore sencillo a efectos que Técnicas Digitales II pueda partir de la base del conocimiento de la estructura completa de un microprocesador, y su set de instrucciones. De este modo haremos los mejores esfuerzos para establecer una base de despegue más amplia a Técnicas Digitales II de modo de permitirle a esta asignatura adquirir mayor vuelo de aplicaciones, utilizando el resto de los periféricos avanzados que dispone el NXP1679, ya que entre Informática II y Técnicas Digitales I se han cubierto los periféricos más sencillos. Este requerimiento se estableció en la última reunión de Área.

La articulación horizontal debería hacerse a través de la coordinación de la materia integradora de tercer nivel, Teoría de los circuitos I, con la que compartimos temas de interés como es el caso del

comportamiento transitorio frente a excitaciones escalón de circuitos RC. Este punto debe estar presente en los análisis de Integridad de señal en alta frecuencia, y tiempos de establecimiento de señales lógicas. Pero también existen temas comunes con la asignatura Dispositivos electrónicos (dispositivos MOS), lo cual sería de gran ayuda si pudiesen analizar el funcionamiento de una puerta lógica y un bit de memoria dinámica en todos los cursos.

Para asegurar que los temas hayan sido abordados en el año lectivo es que los temas en donde se produce articulación se proponen abordar sobre el final del curso

### Cronograma estimado de clases.

Semana	1er cuatrimestre	2º cuatrimestre
1	Códigos + Lab Quartus	Máquinas de estado + TP 6-1+Proyectos
2	Códigos + TP 1+ Lab ISE+Proyectos	Máquinas de estado + TP 6-1
3	Álgebra conmutación + TP 1 + Lab Quartus esquemáticos + testbench.	Máquinas de estado + TP 6-2
4	Álgebra conmutación + TP 2 + Lab ISE esquemáticos + testbench	Máquinas de estado + TP 6-2+Proyectos
5	Combinacionales esquemáticos + TP 3+Proyectos	Máquinas de estado + TP 6-2
6	Combinacionales esquemáticos + TP 3	Contadores + TP 7-2
7	VHDL + TP 4 -1+Proyectos	Registros + TP 8+Proyectos
8	VHDL + TP 4 -1	Camino de datos + TP 10
9	VHDL + TP 4 - 2+Proyectos	Camino de datos + TP 10
10	VHDL + TP 4 -2+Lab VHDL	Softcores + TP 11+ Proyectos
11	Estándares digitales + TP 9	Procesadores por software + TP 11
12	Laboratorio estándares digitales	Procesadores por software + TP 11
13	PLD+Proyectos	TP's pendientes +Terminación de proyectos
14	Secuenciales + TP 5	Terminación de los trabajos prácticos
15	Secuenciales VHDL + TP 5	Laboratorio multivibradores
16	1er Parcial	2do Parcial

### Bibliografía

#### Indispensable

- **Fundamentos de lógica digital con VHDL. Stephen Brown. Zvonko Vranesic. Mc. Graw Hill 2ed.**  
Cumple con uno de los requisitos más exigentes para el objetivo de esta asignatura dentro del área: el enfoque de VHDL orientado a síntesis, simulación y prototipado, pero sin perder de vista el diseño clásico. Orientado para las herramientas de Altera no por ello pierde generalidad. Uno de los pocos textos donde se indaga en el funcionamiento del sintetizador. Escrito por 2 conocidos expertos en el tema de los FPGA que han publicado gran cantidad de papers sobre el tema.

- **RTL Hardware Design Using VHDL. Pong Chu. Wiley.**  
Sin duda un libro indispensable a la hora de aclarar porque VHDL no es un lenguaje de programación, su prédica consiste en mostrar diagramas RTL de cada circuito diseñado. Como adicional, el enfoque a nivel RTL permite que el lector diseñe a nivel conceptual en lugar de quedar enredado en mapas de Karnaugh y kilométricas tablas de transiciones. Un buen punto de referencia para abordar el tema de las máquinas de estado con datapath. Por su abordaje general no ejemplifica con una herramienta de síntesis en particular. Este libro debe complementarse con un texto sobre diseño clásico.

- **Volnei A. Pedroni. Circuit Design and Simulation with VHDL. Second Edition (2010) Ed. Mit Press**

Trata de manera muy completa los aspectos de VHDL y sus aplicaciones al diseño y simulación de circuitos aplicables en la industria. Se enfoca solamente en el lenguaje. Requiere algún conocimiento

previo de sistemas digitales, por tal motivo, a pesar de ser un excelente texto para estudiar y aprender el lenguaje, no es lo más aconsejable para principiantes. Sin embargo una vez abordados los temas iniciales de la asignatura es un excelente libro de referencia para el lenguaje el workbench y sus aplicaciones.

Volnei A. Pedroni se ha doctorado en Ingeniería Eléctrica en el California Institute of Technology, y es profesor de Ingeniería Electrónica en Brasil, en la Federal University of Technology.

### **Básica**

- ***FPGA Prototyping by VHDL Examples. Xilinx Spartan 3-E Version. Pong Chu. Wiley.***

A diferencia de *RTL Hardware Design Using VHDL* este libro se basa en el kit Spartan 3E y la herramienta ISE de Xilinx. Todos los ejemplos y simulaciones se puede bajar al kit comprobar su funcionamiento. Todo el tema de VHDL se supone conocido y es repasado en los primeros capítulos. Tiene el acierto de hacer diseños válidos para cualquier FPGA y aquellos tópicos solo válidos para Xilinx están debidamente remarcados. La segunda parte del libro describe el procesador PicoBlaze de Xilinx. Esta parte tal vez escape a los alcances de la cátedra. Los códigos de ejemplo de interfaces UART, PS/2, el controlador VGA c/ROM de caracteres son excelentes ejemplos de códigos tanto para el alumno como para el profesor.

- ***Embedded SOPC Design with NIOS II Processor and VHDL Examples. Pong Chu. Wiley.***

Siguiendo la línea de *FPGA Prototyping by VHDL Examples* se enfoca el texto en el kit DE1 de Altera y la herramienta Quartus. Luego de una introducción similar al anterior, el libro introduce el procesador NIOS II. Si bien escapa a los temas vistos en la cátedra, el lector podrá encontrar ejemplos de diseño de interfaces PS/2, SPI, I2C, controladores VGA fácilmente comprensibles para el alumno o el docente.

- ***Rapid Prototyping of Digital Systems. Quartus II Edition. Hamblen. Hall. Furman. Springer.***

Este libro escrito por 3 profesores universitarios se destaca por un capítulo llamado el diseño de 15 minutos. Con un ejemplo simple cubre todos los pasos del diseño: ingreso del código, síntesis, corrección de errores, programación y simulación. Esto es el punto más sobresaliente de este libro y es la parte que más vale la pena leer. Tiene la particularidad de cubrir VHDL y Verilog, lo cual sirve al lector para comprobar la similitud de ambos lenguajes y desmitificar la complejidad de una eventual migración. Con numerosos ejemplos como controladores VGA, teclado PS/2 y mouse, dedica un capítulo al tema de softcores. Es una muy buena guía para el docente sobre cómo enfocar el tema. Este texto tiene su talón de Aquiles: no separa adecuadamente la lógica de estado futuro de la inferencia de los registros. El código obtenido no es el más claro, a diferencia de los libros del Dr. Pong Chu. Esto puede inducir a pensar que codificar en VHDL es como programar en lenguaje C. Por lo tanto debe ser advertido el lector acerca de esta deficiencia.

- ***Fundamentals of Digital and Computer Design with VHDL. Richard y Michael Sandige. Mc. Graw Hill.***

Notable que este libro profundice en el uso del VHDL sin descuidar el diseño clásico, del cual hace un tratamiento adecuado a los objetivos de esta asignatura. Los abundantes diagramas RTL aseguran que el lector tenga siempre presente que está diseñando hardware e impiden que pueda considerarse a VHDL un lenguaje de programación como 'C'. Más de la mitad del libro está dedicado al diseño de un procesador. A diferencia de otros textos no esconde la descripción del soft-processor en la sintaxis del VHDL sino que analiza en profundidad sus bloques funcionales. Además de tratar numerosas instrucciones aborda el diseño de puertos de E/S, controladores de memoria de modo de concluir con un sistema simple pero completo. Lamentablemente coloca el uso de las herramientas de síntesis en un apéndice. Este libro podría ser excelente pero tiene poco acierto de diseñar divisores de reloj mediante ripple dividers, rompiendo el concepto de diseño sincrónico, en lugar emplear clock enables.

### **Complementaria**

- ***Digital Design with CPLD applications. Dueck.***

Similar al libro de *Fundamentos de lógica digital con VHDL* se queda en el uso de los CPLD y no aborda los FPGA. Abunda en ejemplos y no descuida el diseño clásico.

- ***Digital Logic and Microprocessor Design with VHDL. Enoch Hwang.***

Una excelente fuente de referencia, presenta el diseño clásico y en VHDL. Tiene la virtud de presentar desde el comienzo un sistema basado en un CPU. A partir de allí cada capítulo muestra como diseñar los elementos de ese sistema. Cubre a fondo los temas de datapath y el diseño de un microprocesador tanto en VHDL como con diseño clásico. Deja claro que este último no solo sirve para implementar detectores de secuencias de unos y ceros.

- ***Digital System Design using VHDL. Charles Roth. PWS Publishing Company.***

Aunque se requiere de conocimientos previos, pues los temas básicos los cubre sucintamente, explica con claridad el tema de datapath. Con mucho detalle se analizan los circuitos multiplicadores por sumas y desplazamientos, y los divisores por restas y desplazamientos. Trata la aritmética de punto flotante. Para explicar un procesador por software tiene el buen tino de basarse en un microcontrolador real: el MC68HC705 de Motorola. El hecho que el autor no separe la lógica de estado futuro de la inferencia de los registros no opaca sus méritos.

- ***Computer Arithmetic. Behrooz Parhami. Oxford University Press.***

Posiblemente el libro que contenga la mayor información sobre aritmética digital. Códigos, circuitos, algoritmos. Desde sumadores, multiplicadores, y funciones avanzadas, ahora incluye capítulos sobre implementación en FPGA. Lamentablemente el autor no da explicaciones detalladas por lo que se recomienda como un material avanzado.

- ***Arithmetic and Logic in Computer Systems. Mi Lu.***

Este el libro explica menos tópicos que Computer Arithmetic pero de una manera más comprensible.

- ***High Speed Digital Design. Howard Johnson. Martin Graham. Prentice Hall.***

En este libro encontrarán la explicación a fenómenos aparentemente inexplicables en los circuitos digitales cuando se trabaja en altas frecuencias. Además se abordan temas como la medición en altas frecuencias, medición de metaestabilidad, y prácticas recomendadas a la hora de diseñar circuitos impresos, utilizar conectores y elegir fuentes de alimentación para sistemas digitales.

- ***Logic and Computer Design Fundamentals. Morris Mano. Charles Kime. Prentice Hall.***

Aunque este libro desarrolle tópicos que exceden a esta materia, los conceptos básicos del diseño digital son explicados con sumo detalle. Se incluyen secciones sobre VHDL aunque no con suficiente detalle. Es una buena fuente de referencia para el tema de soft-processors en el cual se encontrarán explicaciones para el nivel deseado por la cátedra y también algunas que la exceden.

#### *Apuntes*

Jorge E. Sinderman

Técnicas Digitales - Dispositivos, circuitos, diseño y aplicaciones

Editorial CEIT – Buenos Aires 2002

ISBN 987106304-0

Jorge E. Sinderman

Técnicas Digitales - Dispositivos, circuitos, diseño y aplicaciones (2ª edición)

Editorial Nueva librería – Buenos Aires 2006

ISBN 978-987110451-2

Jorge E. Sinderman

Descripción de circuitos lógicos combinacionales usando VHDL – CEIT 2008 (R3GT1). Se planea sustituirlo por una edición 2012.

Jorge E. Sinderman

Descripción de circuitos lógicos secuenciales usando VHDL – CEIT 2008 (R3GT2). Se planea sustituirlo por una edición 2012.

Salvador Tropea

Cuaderno de Cátedra sobre Escritura de bancos de prueba.

Susana Canel

Guía para el uso del programa Quartus II (código R3AP4 del CEIT)

Susana Canel

Guía para el uso del kit de entrenamiento y desarrollo Altera (código R3AP5 del CEIT)

Páginas Web de referencia

[www.altera.com](http://www.altera.com)

[www.alteraforum.com](http://www.alteraforum.com)

[www.xilinx.com](http://www.xilinx.com)

[www.opencores.org](http://www.opencores.org)

[www.terasic.com](http://www.terasic.com)

[www.fpga4fun.com](http://www.fpga4fun.com)

[www.digilentinc.com](http://www.digilentinc.com)

<http://users.ece.gatech.edu/~hamblen/book/bookse.htm>

<http://www.fpgaarcade.com/>

---