

Planificación de Cátedra

Carrera: Ingeniería Electrónica

Asignatura: Técnicas Digitales I

Nivel: 3ro.

Bloque: Tecnologías Básicas.

Área: Técnicas Digitales.

Dictado anual.

Horas semanal: 4 Horas año: 128

Programa analítico.

Unidad 1. Introducción a los Sistemas digitales y códigos binarios. (Cap. I)

Objetivo específico:

Presentar el marco de referencia para el estudio de las próximas unidades. Introducir los diversos sistemas binarios adecuados para representar información en sistemas digitales. Introducir al alumno en los sistemas de códigos.

Contenido:

Introducción a los Sistemas Digitales. Metodología de diseño basada en HDL (Hardware Description Language). Características analógicas del diseño digital. Definición de código. Código BCD. Otros códigos decimales: 2,4,2,1 y exceso 3. Código de Grey. Código ASCII y de 7 segmentos. Códigos detectores de errores. Bit de Paridad. Códigos detectores y correctores de error de distancia 2,3 y 4. Código de Hamming. Códigos CRC. Checksum. Códigos para transmisión de datos en serie y almacenamiento.

Justificación:

La Unidad 1 comienza con la introducción a la materia y tiene como objetivo permitir que el alumno tenga una visión global del conjunto, para saber de antemano de que se tratará el curso. Se introducen todos los temas sobre los que se trabajará en el desarrollo de la asignatura. Se deberá asimilar que un Sistema Digital es una interconexión de módulos digitales. Para entender la operación de cada uno de los módulos, es necesario tener un conocimiento básico de circuitos digitales y su función lógica.

A partir de la definición de “código” se confrontará al estudiante con modos distintos de presentar la realidad. Se introducen los códigos usuales: BCD, ASCII en sus varias modalidades. Los conceptos de paridad, de chequeo cíclico de redundancia y distancia se introducen a los efectos de abrir senda para los sistemas de comunicaciones. Se presenta el código Gray como ejemplo de contigüidad, a los fines de introducir las ventajas de codificación contigua desde lo técnico.

Demandará ocho horas de dictado.

Unidad 2. Algebra de Boole. (Cap. 2)

Objetivo específico:

Desarrollar el álgebra de conmutación y suministrar los fundamentos que permitirán optimizar circuitos simples y entender el propósito de algoritmos usados por herramientas de software para optimizar circuitos complejos.

Introducir Lenguaje de descripción de hardware (HDL) que se usa en modernas herramientas de diseño.

Contenido:

Definición. Postulados. Principio de dualidad. Operaciones lógicas y su analogía mecánica con llaves. Compuertas básicas: AND, OR, NOT, y XOR. Tablas de verdad y compuertas derivadas (NAND, NOR).

Teoremas del Álgebra de Boole. Teorema de De Morgan. Teorema del consenso.

Concepto del lenguaje de descripción de Hardware SystemVerilog. Elementos del lenguaje. Declaración de un módulo. Instanciación. Sentencia de asignación continua: ASSIGN. Operadores bitwise y de reducción. Variables internas. Representación de números.

Justificación.

Los métodos matemáticos que simplifican circuitos se basan principalmente en el álgebra de Boole.

Demandará ocho horas de dictado.

Unidad 3. Funciones lógicas y su minimización. (Cap. 3)

Objetivo específico:

Entender la descripción matemática subyacente y la solución al problema de minimizar un conjunto grande de ecuaciones booleanas. Entender la importancia del tema y preparar para realizar diseño manual de circuitos simples. Preparar para el uso adecuado de las herramientas de diseño moderno.

Contenido:

Función del Álgebra de Boole. Concepto de Minitérminos y Maxitérminos. Forma canónica. Obtención de funciones desde la tabla de verdad. Expresión matemática de la Función. Función complemento. Expansión a la forma canónica, métodos tabulares y algebraicos. Conceptos básicos de minimización. Diagramas de Karnaugh para 2, 3, 4 y 5 variables. Condiciones no importa/no sucede. Aplicación de las funciones lógicas. Referencia introductoria a la estructura de un Dispositivo lógico programable. Descripción de funciones lógicas en SystemVerilog. Introducción a las herramientas de CAD. Retardos en las compuertas. Introducción al concepto de Banco de pruebas (Testbench). Simulador ModelSim. Simulación de comportamiento y temporal. Demoras y respuestas transitorias. Glitches. Síntesis.

Justificación.

La minimización a nivel de compuertas es la tarea de diseño de encontrar una implementación óptima a nivel de compuertas de las funciones booleanas que describen un circuito digital. Esta tarea se entiende bien, pero es difícil de ejecutar por métodos manuales cuando la lógica tiene muchas entradas.

Demandará doce horas de dictado.

Unidad 4. Familias Lógicas. (Fin Cap 3 y Apéndice)

Objetivo específico:

Comprender lo que hay debajo de la abstracción digital.

Contenido:

Estructura básica de un transistor MOS. Familia lógica CMOS. Características: especificaciones de tensión y de corriente de entrada y de salida, margen de ruido, retardo de propagación, consumo estático y dinámico. Familia TTL: Serie 74/54 Standard, Schotky y Low Schotky como referencia histórica. Características generales. Interfase C-MOS/TTL y TTL/C-MOS. Modelado a nivel de llaves con SystemVerilog: **nmos** y **pmos**. Puerta de transmisión CMOS. Puerta de transmisión en SystemVerilog: **cmos**. Lógica de tres estados. Tipos de salida. Valores flotantes en SystemVerilog (Zs) y desconocidos o ilegales (Xs). Buffer.

Justificación.

Un sistema digital usa variables de valor discreto. Sin embargo, las variables están representadas por cantidades físicas continuas tal como el voltaje en un cable. Por lo tanto el diseñador debe reconocer una forma de relacionar el valor continuo al valor discreto en este caso definiendo los niveles lógicos.

Demandará ocho horas de dictado.

Unidad 5. Circuitos combinacionales. (Cap. 4)

Objetivo específico:

Comprender la especificación funcional de un circuito combinacional.

Contenido:

Circuitos combinacionales básicos: Multiplexores, Demultiplexores, Decodificadores, Codificadores con y sin prioridad, 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. Desplazadores. Uso del decodificador y del multiplexor como generador de funciones.

Verificación de los recursos utilizados en la FPGA por cada uno de los circuitos anteriores. El operador ternario en SystemVerilog para implementación de multiplexores. Modelado estructural. Diferencia entre una variable y una señal.

Justificación.

La lógica combinatoria a menudo se agrupa en bloques constructivos más grandes para construir sistemas más complejos. Esta es una aplicación del principio de abstracción, ocultando los detalles innecesarios a nivel de compuertas para enfatizar la función del bloque constructivo.

Demandará ocho horas de dictado.

Unidad 6. Aritmética binaria y circuitos aritméticos. (Cap 4)

Objetivo específico:

Comprender las operaciones aritméticas de números con y sin signo. Conocer las diferentes alternativas de implementación de los circuitos aritméticos en general y de los sumadores en particular.

Contenidos:

Concepto de complemento a la base y a la base-1. Operaciones aritméticas de números en CA2 (suma, resta, multiplicación, división). Implementación de circuitos aritméticos. Sumador completo de 1 bit, Sumador por propagación de acarreo. Sumadores rápidos. Carry lookahead 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. Circuitos Detectores de todos unos o ceros. Circuitos comparadores de magnitud y de igualdad. Evaluación de funciones matemáticas mediante tablas. Operaciones con un operando constante.

Justificación.

La suma forma la base para muchas operaciones de procesamiento, desde ALUs que la utilizan para efectuar la multiplicación hasta el filtrado. Como resultado, los circuitos sumadores que suman dos números binarios son de gran interés para los diseñadores de sistemas digitales. Una casi interminable variedad de arquitecturas de sumador están destinadas a servir a diferentes requisitos de velocidad / potencia / área.

Los complementos se usan en computadoras digitales para simplificar la operación de substracción y para la manipulación lógica. Simplificando las operaciones conduce a circuitos más simples y menos costosos para implementar tales operaciones.

Demandará doce horas de dictado.

Unidad 7. Lógica secuencial síncrona. (Cap. 5)

Objetivo específico:

Esbozar el procedimiento formal para el análisis y el diseño de circuitos secuenciales síncronos. Presentar la estructura de compuertas del flip-flop D. Establecer la diferencia entre disparo por nivel y por flanco. Explicar el modelado comportamental en HDL SystemVerilog para los circuitos secuenciales.

Contenidos:

Sistemas secuenciales. Introducción de los conceptos de memoria y tiempo. Autómata de Mealy y de Moore. Latches SR y D. Flip-flop maestro-esclavo. Flip-flop activado por flanco. Procedimiento de análisis. Diagrama de estados. Asignación de estados. Diseño con flip-flop D. Reducción de estados. Osciladores integrados CMOS para reloj. Máxima frecuencia de operación. Restricciones temporales: setup y hold, skew y jitter. Descripción de maquinas de estado en SystemVerilog, simulación. Problemas de aplicación.

Justificación.

Hay dos tipos de circuitos secuenciales que representan a la mayoría de los diseños discretos prácticos: un **circuito secuencial realimentado** que usa compuertas comunes y lazos de realimentación para obtener memoria en un circuito lógico, por lo tanto creando bloques constructivos de circuitos secuenciales tales como latches y flip-flops que se usan en diseños de nivel superior. Una **máquina de estados síncrona con reloj** usa estos bloques constructivos, en particular flip-flops D activados por flancos, para crear circuitos cuyas entradas son examinadas y cuyas salidas cambian de acuerdo a una señal de reloj de control.

Demandará dieciséis horas de dictado.

Unidad 8. Registros y contadores. (Cap. 6)

Objetivo específico:

Presentar varios componentes de circuitos secuenciales tales como registros, registros de desplazamiento y contadores como también la respectiva descripción HDL.

Contenidos:

Introducción. Registros. Registro con carga en paralelo. Registro de desplazamiento. Registro de desplazamiento con carga en paralelo. Registro de desplazamiento bidireccional. 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 binarios síncronos up/down. Contadores binarios rápidos. Contadores basados en registros: Anillo, Johnson y LFSR (Linear Feedback Shift Register). Descripción en SystemVerilog.

Justificación.

Los registros, registros de desplazamiento y contadores son los bloques constructivos básicos a partir de los cuales se construyen sistemas digitales más complejos.

Demandaré doce horas de dictado.

Unidad 9. Memorias y Lógica programable.(Cap. 7)

Objetivo específico:

Lograr que el estudiante distinga entre los distintos tipos de tecnología de memoria, comprenda sus limitaciones en cuanto a prestaciones y conozca en detalle sus usos y aplicaciones.

Contenidos:

Conceptos básicos, clasificación de las memorias según el modo de operación, permanencia de la información, la tecnología, etc. Descripción de los pines, utilización de los buses, del habilitador del chip. Diagrama temporal de una memoria de acceso aleatorio. Memorias RAM, ROM, PROM, EPROM y EEPROM, estructura básica y descripción de las celdas básicas. Análisis de los requerimientos temporales de los sistemas. Descripción en SystemVerilog.

Lógica programable: ROM, PLA, PAL, GAL, CPLD y FPGAs. Diferencia entre una CPLD y FPGA. Introducción y estructura de una FPGA. Arquitectura básica. LUTs. Memorias de configuración.

Justificación.

Una unidad de memoria es un dispositivo en el que la información binaria se transfiere para el almacenamiento y de la que se recupera dicha información cuando es necesaria para el procesamiento. Cuando el procesamiento de datos se lleva a cabo, la información de la memoria se transfiere a los registros seleccionados en la unidad de proceso. Los resultados intermedios y finales obtenidos en la unidad de proceso se transfieren de nuevo para almacenarse en la memoria. La información binaria recibida desde un dispositivo de entrada se almacena en la memoria, y la información transferida a un dispositivo de salida se toma de la memoria. Una unidad de memoria es una colección de celdas capaces de almacenar una gran cantidad de información binaria.

La lógica programable es una de las formas que nos permite la implementación de los circuitos digitales. La otra es el diseño de los circuitos integrados de aplicación específica (ASIC).

Demandaré 12 horas de dictado.

Unidad 10. Trayectoria de los datos: unidad de proceso.

Objetivo específico:

Dar una introducción a métodos ampliamente utilizados por diseñadores de microprocesadores y otros circuitos VLSI.

Contenidos:

Ejemplo introductorio. Secuencialización de las operaciones. Máquinas algorítmicas. Circuitos pipeline. Metodología de transferencia de registros. Operaciones básicas a nivel RTL. Unidad de proceso secuencializada. Buses. Arquitectura de un bus. Arquitectura de dos buses. Diferentes tipos de realización del control. Microprograma de control. Decodificador de órdenes. Contador de programa. Microsecuenciador. Descripción en SystemVerilog.

Justificación.

Los diseñadores de sistemas digitales están acostumbrados a utilizar microprocesadores, memorias, periféricos, etc., así como a sintetizar la lógica de interfase por métodos clásicos de diseño lógico (circuitos combinacionales, máquinas secuenciales, etc.). Muchas veces basan su diseño en un microprocesador, aún cuando sólo se utiliza un conjunto reducido de sus funciones. Son circuitos muy difundidos y, por lo tanto, relativamente baratos. En cambio, en el caso de un ASIC, el costo depende, entre otras cosas, de la superficie del chip. Por tanto, integrar en un ASIC una macrocélula tipo microprocesador estándar, del cual sólo se utilizaría un número reducido de funciones, no sería rentable. En tal caso convendría diseñar un circuito totalmente adaptado a la aplicación que se pretende desarrollar. En conclusión, el diseñador de hoy tiene que conocer una serie de temas reservados hasta ahora a los diseñadores de circuitos VLSI.

Demandaré dieciséis horas de dictado.

Unidad 11. Sistemas digitales tolerantes a fallas.**Objetivo específico:**

Estudiar las técnicas para conseguir que los sistemas continúen funcionando correctamente, a pesar de fallas en su hardware o errores de software.

Contenidos:

Conceptos generales de tolerancia a fallas. Fallas, error y avería. Causa de las fallas. Caracterización de las fallas. Filosofía de diseño para combatir las fallas. Redundancia. Tipos de redundancia: hardware o física, software, informacional y temporal. Redundancia modular triple. Duplicación con comparación. Temporizadores centinela. Checksum. Detección de fallas transitorias.

Justificación.

El Diseño de sistemas tolerante a fallas ocasionales reduce el costo y mejora la performance. Son necesarios en aplicaciones críticas relacionadas a la vida humana, actividades espaciales, etc.

Demandaré ocho horas de dictado.

Bibliografía

La bibliografía de uso en la asignatura se divide en dos grupos: obligatoria y de referencia

Bibliografía básica

Digital design with an introduction to the Verilog HDL.

Morris Mano, Michael Ciletti.

Fifth edition. Pearson, 2013.

ISBN-13: 978-0-13-277420-8

Digital Design and Computer Architecture.

David Money Harris & Sarah L. Harris

2nd edition. Morgan Kaufmann (Elsevier Press), 2013.

ISBN: 978-0-12-394424-5

Digital Design. Principles and Practices.

John F. Wakerly.

Fourth edition. Prentice Hall, 2006.

ISBN 0-13-186389-4

Bibliografía de referencia.

CMOS VLSI design. A Circuits and Systems Perspective.

Neil Weste – David Money Harris

Fourth edition. Addison Wesley, 2011

ISBN 13: 978-0-321-54774-3

Digital Integrated Circuits. A design perspective.

Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic.

Second edition. Prentice Hall, 2003.

ISBN: 0-13-090996-3.

Rapid prototyping of Digital Systems.

James O. Hamblen, Tyson S. Hall, Michael D. Furman.

SOPC edition. Springer, 2008.

ISBN 978-0-387-72670-0

Programa de Trabajos Prácticos.

La propuesta de Trabajos Prácticos para la materia implica un conjunto de problemas diseñados para que a través de ellos el estudiante conecte los temas de teoría con situaciones reales. Los problemas deberán resolverse individualmente o en grupo. Los grupos tendrán no más de tres miembros, que deberán pertenecer a la misma Comisión. Los informes de trabajos grupales serán objeto de un coloquio evaluativo, con nota individual para los miembros.

Lista de Trabajos Prácticos

Los Trabajos Prácticos en diseño lógico y circuitos digitales pueden construirse usando circuitos integrados estándar colocados en una breadboard y son de fácil armado en el laboratorio. Los experimentos más complejos deberán implementarse a través de una placa de desarrollo de FPGA que es lo deseable como una alternativa a los IC's.

Trabajo Práctico 1. Introducción a los Sistemas digitales y códigos binarios.

Diseñar un código de Hamming para codificar 5 bits de información.

Trabajo Práctico 2. Álgebra de Boole.

Obtener una función lógica en su forma canónica como suma de productos o productos de suma y aplicar el teorema de De Morgan para cambiar las compuertas con que se resuelve la función.

Trabajo Práctico 3. Funciones lógicas y su minimización.

Problemas de simplificación usando diagramas de Karnaugh. Resolución individual.

Síntesis de circuitos combinacionales sencillos a partir de álgebra de proposiciones.

Resolución grupal.

Trabajo Práctico 4. Familias Lógicas.

Modelado a nivel de llaves con SystemVerilog: **nmos** y **pmos**. Describir una compuerta lógica de dos entradas.

Puerta de transmisión en SystemVerilog: **cmos**. Describir un multiplexor de dos entradas.

Cálculo de potencia: Un celular determinado tiene una batería de 6 watt-hora (W-hr) y opera a 1.2V. Suponga que, cuando está en uso, el teléfono opera a 300MHz y el valor promedio de la capacidad en la conmutación del chip en un momento dado es 10nF. Cuando está en uso, también emite 3 W de potencia de su antena. Cuando no está en uso, la potencia dinámica cae casi a cero debido a que el procesamiento de la señal se apaga. Sin embargo, el teléfono también drena 40 mA de corriente de reposo si está en uso o no. Determine la vida de la batería del teléfono (a) si no está siendo usado, y (b) si está siendo usado continuamente.

Trabajo Práctico 5. Circuitos combinacionales.

Describe en SystemVerilog una función lógica combinacional con 6 bits de entrada N_5 - N_0 representando un entero entre 0 y 63, y dos salidas M_3 y M_5 que indica si el entero es un múltiplo de 3 o 5, respectivamente. Lleve su diseño a un PLD, CPLD o FPGA y determine cuantos recursos se emplean para la realización del mismo. Después escriba un testbench que compare las salidas de su circuito para todas las combinaciones de entrada posibles contra resultados calculados por el simulador usando su propia aritmética. El testbench debe detenerse y mostrar un mensaje de error si hay una discordancia. Compruebe su testbench colocando un error en su descripción original en SystemVerilog y corra el testbench.

Trabajo Práctico 6. Aritmética binaria y circuitos aritméticos.

Problemas sobre cambio de base y aritmética binaria. Para resolver de manera individual. Diseño de una ALU.

Trabajo Práctico 7. Lógica secuencial síncrona.

Diseño de máquinas de estado: el controlador de un tren eléctrico.

Cap. 8 del Libro: Rapid prototyping of Digital Systems. (ver bibliografía)

Trabajo Práctico 8. Registros y contadores.

El desarrollo de la parte de recepción de un enlace RS 232-C, es integrador de conocimientos de esta Unidad pues usa todos los elementos discutidos.

Trabajo Práctico 9. Memorias y lógica programable.

Cada grupo deberá buscar en Internet datos sobre una clase diferente de memoria. Los resultados de la búsqueda serán presentados en un informe. La evaluación será individual, a partir de la exposición en clase del informe realizado.

Interpretación de las hojas de datos de dispositivos FPGA.

Trabajo Práctico 10. Trayectoria de los datos: unidad de proceso.

Diseño de un monoestable no redispensible con 16 salidas: OUT0, ..., OUT15. Los tiempos de temporización correspondientes son: $N_0 T_S$, $N_1 T_S$, ..., $N_{15} T_S$. Siendo N_0 , N_1 , ..., N_{15} números enteros, disponibles dentro del sistema, y T_S el periodo de una señal de referencia S.

Trabajo Práctico 11. Sistemas digitales tolerantes a fallas.

Aplicar la técnica TMR (Triple Modular Redundancy) al módulo original correspondiente a un registro de 8 bits con habilitación de carga en paralelo.