



Universidad
de Huelva

*Departamento de Ingeniería Electrónica, de Sistemas
Informáticos y Automática*

ELECTRÓNICA DIGITAL

PRÁCTICAS DE LABORATORIO (Parte I)

CURSO 2000-2001

NORMAS para las prácticas de E.D.

ASISTENCIA

Las prácticas tendrán lugar en el Laboratorio de Redes (Sotano del edificio V.R.C), siendo obligatoria la asistencia.

Al principio de cada cuatrimestre se entregará la lista de prácticas de laboratorio a realizar por el alumno. Este deberá realizarlas a lo largo de las sesiones de prácticas en el orden dado. Una vez realizadas serán evaluadas por el profesor de prácticas, que resolverá cualquier duda presentada en la práctica. Se aconseja traer la práctica preparada de casa, como mínimo su diseño.

El alumno podrá realizar grupos de dos personas como máximo para el montaje de las prácticas, debiendo indicárselo al profesor de prácticas, que llevará un seguimiento de cada grupo. Sin embargo, cada alumno entregará una memoria de prácticas individual (no se admitirán fotocopias ni copias de prácticas de los alumnos del grupo) al final de cada cuatrimestre en fechas que fijara el profesor de prácticas.

EVALUACION

a) Las prácticas han de ser aptas para aprobar la asignatura.

Las prácticas de laboratorio se calificarán aptas, y muy bien en función de:

- Nivel técnico y calidad de redacción de la memoria de prácticas.
- Revisión del trabajo de practicas funcionando por el profesor de prácticas.

En caso de que las prácticas se encuentren muy bien, podrá suponer la subida de 1 punto en la nota final si esta es mayor de 4.

Para aprobar la asignatura será necesario tener al menos 3 puntos en el examen de Teoría, media ≥ 5 entre Teoría y Problemas y prácticas de laboratorio APTAS.

b) Memoria de prácticas

Se entregará una memoria impresa de las prácticas realizadas, junto con un disco que contendrá los ficheros fuente de los diseños o programas si ha lugar. En la memoria debe aparecer claramente el nombre del alumno. Cada memoria de practicas debe estar redactada mediante procesador de textos o equivalente y contendrá como mínimo los siguientes apartados:

- 1.- Portada donde aparezca claramente nombre, curso y fecha.
- 2.- Índice del documento.
- 3.- Un epígrafe para cada práctica, donde aparezca:
 - Enunciado.
 - Diseño del circuito con esquemas.
 - Comentarios sobre el montaje y/o problemas presentados y como se han resuelto.
 - Respuesta a las cuestiones planteadas en el enunciado.
 - Posibilidades de ampliación/mejora si ha lugar.
- 4.- Anexos donde aparezcan fotocopias y/o documentación técnica si se considera necesario. Bibliografía que se ha empleado.

Todos los programas y diseños han de aparecer comentados, tanto los del disco como los de la memoria. Se valorará la claridad en este aspecto. Por imperativos legales la memoria no se devolverá al alumno.

PRÁCTICA Nº 1

DISEÑO DE UN PROCESADOR DIGITAL COMBINACIONAL I

OBJETIVOS

Introducir el manejo práctico en laboratorio de circuitos TTL y placas de montaje prototipo

Realizar el diseño a partir de la interconexión de bloques funcionales estándar M.S.I.

CONCEPTOS BÁSICOS

El diseño de procesadores digitales combinacionales permite obtener un resultado fiable con un tiempo de diseño reducido. Se evitan en gran medida las simplificaciones booleanas y se introducirán siempre que sea posible circuitos de escala de integración media o alta.

La filosofía de diseño cambia en estos procesadores, ya que nos basamos en la idea de que 'cada parte del circuito realiza una misión concreta y aislable del resto'.

MATERIAL

Entrenador lógico.

Placa breadboard de montaje e hilos de conexión.

Componentes TTL.

PROCEDIMIENTO

Se desea realizar un circuito que reciba dos números de dos bits y que realice las siguientes operaciones:

- 1) Si ambos números son iguales la salida debe ser cero.
- 2) Si los números son distintos, la salida será la suma de ambos.

Es necesario representar la salida en un visualizador de 7 segmentos. Se dispone del siguiente material:

- 74154 (Decodificador de 4 a 16 líneas)
- 74157 (Multiplexor cuádruple de 2 a 1 líneas)
- 7483 (Sumador de cuatro bits con acarreo)
- 7420 (Puertas NAND de cuatro entradas)
- 7400 (Puertas NAND de dos entradas)

¿ De qué forma es posible detectar la igualdad de las dos combinaciones de entrada con el material disponible ?

Diseñar el circuito completo y montarlo. Comprobar su funcionamiento.

PRÁCTICA Nº 2

MANEJO DE SOFTWARE DE CAPTURA DE ESQUEMATICOS

OBJETIVOS

Manejo de programa de captura de esquemáticos.
Generación de documentación a partir del esquemático.

CONCEPTOS BÁSICOS

En la actualidad es básico el conocimiento del manejo del ordenador personal, siendo una herramienta muy útil en muchas facetas del trabajo diario. Dada la profusión de su uso existen en el mercado gran variedad de herramientas de diseño electrónico que el alumno debe conocer y aprovechar. El auge de Internet también ha permitido tener acceso a gran cantidad de documentación que nos ofrecen los fabricantes, convirtiéndose en una herramienta más.

MATERIAL

Software de captura de esquemáticos PROTEL 99.
Acceso a Internet.

PROCEDIMIENTO

El circuito realizado en la práctica anterior se introducirá en el programa comenzando por la colocación de los componentes en la hoja de trabajo y su interconexión.

Tras entrar el esquemático el alumno comprobará mediante un test de reglas eléctricas que no existe ningún error en su conexionado.

A continuación el alumno generará una lista de materiales y un netlist (lista de conexiones) del diseño.

Por último, el alumno obtendrá a través de los servicios de búsqueda de Internet o del que ofrezca cualquier fabricante de componentes los "data sheets" (hojas de características) de todos los componentes que intervengan en el diseño y los adjuntará al diseño que entregará al profesor en un disco flexible.

¿ Que ventajas y desventajas aporta el uso de ordenador frente a trabajar "a mano" ?

¿ Como emplearías la documentación generada a la hora de realizar un informe del diseño ?

PRÁCTICA Nº 3

DISEÑO DE UN DECODIFICADOR DE DIRECCIONES MEDIANTE SOFTWARE DE DISEÑO DE PLD

OBJETIVOS

Manejo de programa de diseño de lógica programable PLD.
Uso básico del lenguaje de programación de PLD, CUPL.

CONCEPTOS BÁSICOS

El sistema CAD capaz de programar PLD nos facilita el trabajo reduciendo las ecuaciones lógicas de nuestro sistema combinacional y generando un fichero JEDEC con la lista de fusibles de nuestro PLD. Una vez generado éste se emplea un programador convencional de dispositivos para su programación.

MATERIAL

Software de síntesis de PLD PROTEL 99. Programador de dispositivos. Dispositivos PLD 22V10. Entrenador Digital.

PROCEDIMIENTO

Se pide diseñar un decodificador de direcciones para un computador con un Megabyte de memoria (desde la posición 00000 a la FFFFF). La organización de la memoria se representa en la tabla siguiente.

| Dirección de Memoria | | Longitud | Uso | Señal |
|----------------------|---------|----------|---------------------|-------|
| Hexadecimal | Decimal | | | |
| 0000..9FFFF | 0K | 640K | Memoria del Sistema | SMEM |
| A000..AFFFF | 640K | 64K | Area EGA | EGA |
| B0000..B0FFF | 704K | 4K | Area MDA | MDA |
| B1000..B7FFF | 708K | 28K | Dispositivo 0 | DIS0 |
| B8000..BBFFF | 736K | 16K | Area CGA | CGA |
| BC000..BFFFF | 752K | 16K | Dispositivo 1 | DIS1 |
| C0000..C7FFF | 768K | 32K | Dispositivo 2 | DIS2 |
| C8000..CBFFF | 800K | 16K | Disco Fijo | DISK |
| CC000..CFFFF | 816K | 16K | IO Periféricos | IO |
| D0000..DFFFF | 832K | 64K | Memoria Expandida | EXP |
| E0000..EFFFF | 896K | 128K | Memoria ROM | ROM |

Comparar las ecuaciones resueltas por el compilador para cada uno de los ficheros fuente e indicar si coinciden o existe alguna diferencia.

Se realizará un fichero en formato CUPL mediante el "PLD Wizard" en el que rellenaremos los campos que falten e introduciremos las ecuaciones lógicas de nuestro circuito. Hecho esto se procederá a la síntesis del circuito y la generación del fichero JEDEC que se enviará a través de la red del laboratorio al ordenador que dispone del programador, realizando la programación de este. Por último, se procederá a probarlo sobre un entrenador digital. Proporcionar:

- 1) Fichero fuente.
- 2) Ecuaciones resueltas por el compilador.
- 3) Fichero de salida del simulador.

¿ Que ventajas y desventajas aporta el uso de lógica digital frente a lógica TTL ?

Enumera los parámetros necesarios por el programa para realizar la síntesis del diseño.

PRÁCTICA Nº 4**DISEÑO DE UN GENERADOR DE PARIDAD**OBJETIVOS

Dominar la captura de esquemáticos y el proceso de compilación empleando el software CAE Foundation M1.4 de Xilinx.

CONCEPTOS BÁSICOS

El entorno Foundation M1 de Xilinx introduce al alumno en los entornos CAE de diseño de sistemas digitales de alta densidad. Este software nos va a permitir realizar el esquemático y la documentación del diseño, realizar su prueba funcional, sintetizar el diseño en un fichero con la extensión BIT que se empleara para programar el dispositivo FPGA y si es necesaria, realizar una posterior simulación temporal (teniendo en cuenta las rutas de las señales dentro del chip escogidas por el proceso de síntesis).

MATERIAL

Software de síntesis de FPGA de Xilinx Foundation F1.4. Programador de FPGA. Entrenador FPGA. Entrenador Digital.

PROCEDIMIENTO

Construir un generador de paridad de 8 bits a partir de un generador de paridad de 4 bits construido a su vez mediante puertas OR-EXCLUSIVAS. Realizar la simulación del circuito. Emplear símbolos en el esquemático.

¿Que ventajas e inconvenientes trae consigo el empleo de varios niveles de jerarquía en los diseños?

PRÁCTICA Nº 5

DISEÑO DE UN SISTEMA SÍNCRONO DE CONTROL

OBJETIVOS

Emplear el método de definición de diagramas de estado de XILINX para implementar un circuito secuencial síncrono de control.

CONCEPTOS BÁSICOS

El entorno Foundation M1 de Xilinx incorpora un editor de diagramas de estado que ayuda en la creación de sistemas secuenciales complejos. Una vez realizado el diagrama es posible simularlo y genera un fichero con la configuración de la lógica secuencial necesaria para realizar el esquemático y la documentación del diseño, realizar su prueba funcional, sintetizar el diseño en un fichero con la extensión BIT que se empleara para programar el dispositivo FPGA y si es necesaria, realizar una posterior simulación temporal (teniendo en cuenta las rutas de las señales dentro del chip escogidas por el proceso de síntesis).

MATERIAL

Software de síntesis de FPGA de Xilinx Foundation F1.4. Programador de FPGA. Entrenador FPGA. Entrenador Digital.

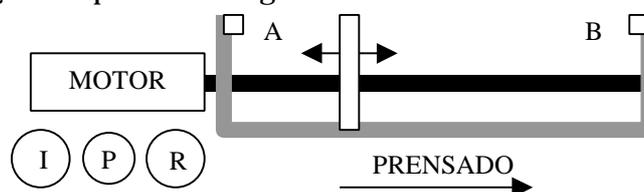
PROCEDIMIENTO

Diseñar un sistema de control para automatizar el funcionamiento de la prensa de la figura, según los siguientes criterios:

- 1) Partiendo de la posición de reposo ($A=1$) y una vez colocado el material a prensar, se activará el pulsador de inicio 'I', con lo que comenzará el prensado.
- 2) El proceso continuará hasta la activación de 'B', momento en el cual la prensa retrocederá a la posición de reposo, realizando de nuevo el ciclo indefinidamente parando el ciclo en la posición de reposo mediante una nueva pulsación de 'I'.
- 3) Si durante el funcionamiento se produjera alguna anomalía, se activaría el pulsador de parada de emergencia 'P', con lo que la prensa se pararía inmediatamente. La prensa permanecerá parada hasta activar el pulsador de rearme 'R', con lo cual reanudaría el proceso desde el punto exacto donde se interrumpió.
- 4) Para el movimiento de la prensa se dispone de una variable 'M' ($M=1$: motor activo) y otra 'S' para seleccionar el sentido del desplazamiento ($S=0$: retroceso; $S=1$: prensado).

Se pide:

- a) Resolver este problema mediante un diagrama de estados empleando el editor de diagramas de estados de XILINX.
- b) El diseño se realizará mediante la placa XESS. Se pide el fichero fuente en formato ABEL generado por el editor de diagramas de estado.
- c) ¿ Que ocurre si los pulsadores P y R se activan a la vez durante una fase de prensado ? ¿ Y los sensores A y B ? ¿ Podría decirse que bajo estas circunstancias el diseño original no funciona bien ?, ¿ cómo podría corregirse en caso necesario ?



PRÁCTICA Nº 6

DISEÑO DE UN CONTADOR PROGRAMABLE

OBJETIVOS

Realizar un circuito digital secuencial y su síntesis en FPGA a partir del esquemático del circuito.

CONCEPTOS BÁSICOS

A partir del esquema del circuito, se va a proceder a realizar su prueba funcional, sintetizar el diseño en un fichero con la extensión BIT que se empleara para programar el dispositivo FPGA y si es necesaria, realizar una posterior simulación temporal (teniendo en cuenta las rutas de las señales dentro del chip escogidas por el proceso de síntesis).

MATERIAL

Software de síntesis de FPGA de Xilinx Foundation F1.4. Programador de FPGA. Entrenador FPGA. Entrenador Digital.

PROCEDIMIENTO

Se desea diseñar un contador ascendente/descendente que tiene las siguientes características:

- Con 'UP=1' cuenta ascendente.
- Con 'UP=0' cuenta descendente.
- Tiene una línea de RESET síncrona.
- Cuenta de 0 a 12, pero se salta la cuenta '5'.
- Tiene entrada de carga síncrona mediante una señal: 'PE' (Preset Enable).

Se empleará una XC4003EPC84-3, y se asignarán las patillas de la siguiente forma:
UP:22, RESET:23, PE:28, A,B,C,D:24,25,26,27,

Se realizará una comprobación completa, tanto del funcionamiento de la línea RESET, como de la carga paralelo y de las cuentas completas hacia delante y hacia atrás. Todos estos listados se incluirán en la memoria de prácticas.

Implementar el diseño en la FPGA y observar si realiza la cuenta correctamente.

PRÁCTICA N° 7

DISEÑO DE UNA MEMORIA FIFO

OBJETIVOS

Emplear las herramientas de XILINX para crear una memoria secuencial síncrona de tipo FIFO “Primero en entrar, primero en salir”.

CONCEPTOS BÁSICOS

Las memorias síncronas de acceso secuencial se emplean en muchos sistemas de electrónica digital para almacenar datos de forma temporal, permitiendo una gran velocidad de funcionamiento. Es posible emplear celdas de memoria estática que almacenan normalmente la configuración de la FPGA como memoria para uso del ingeniero de diseño.

MATERIAL

Software de síntesis de FPGA de Xilinx Foundation F1.4. Programador de FPGA. Entrenador FPGA. Entrenador Digital.

PROCEDIMIENTO

Diseñar por un lado la RAM a base de registros (flip-flops) del dispositivo. Bastará con una RAM de 32 posiciones y palabras de 4 bits. Una vez comprobado que funciona correctamente la RAM, realizar el circuito de control a base de dos punteros (contadores) y un multiplexor que seleccione si se va a leer o escribir.

Implementar el diseño en la FPGA y realizar la prueba de introducir varios numero en orden y extraerlos en el mismo orden, comprobando de esta forma su buen funcionamiento.