
Práctica 5: Diseño y realización de circuitos secuenciales mediante circuitos programables (II)

Introducción

En esta práctica se diseñará y simulará un circuito secuencial con subsistemas dentro de un entorno informático y además se programará para comprobar su funcionalidad mediante el encendido de displays 7 segmentos. **Es esencial que, antes de acudir al laboratorio, el alumno haya leído atentamente y comprendido toda la parte introductoria de la práctica 5.**

Objetivos

La práctica comprende tres partes:

- Parte 1: Estudio teórico
En esta parte de la práctica se plantea el diseño a realizar. Habrá de ser realizada por el alumno antes de acudir al laboratorio.
- Parte 2: Estudio experimental
Consiste en la introducción, simulación y programación del diseño.

NOTA: Esta práctica se realizará de manera individual, cada alumno en un ordenador.

Flujo de diseño con herramientas de CAD

Los pasos a seguir para realizar un diseño con la mayoría de las herramientas informáticas de diseño electrónico (CAD) son muy parecidos, mostrándose de forma gráfica en la Figura 1. Estos pasos son:

- En primer lugar, hay que introducir el diseño. En la práctica que se plantea este diseño se realizará mediante un entorno de captura de esquemas, poniendo e interconectando los símbolos de los distintos componentes mediante un entorno gráfico.
- En segundo lugar, hay que crear los patrones o estímulos que se le van a introducir al circuito con objeto de comprobar su funcionamiento.

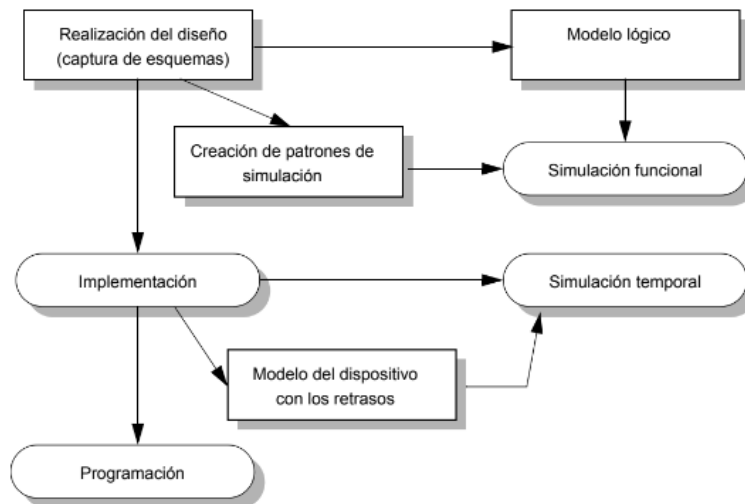


Figura 1: Esquema de la realización de un diseño mediante herramientas de CAD.

- En tercer lugar, se simula el circuito utilizando los patrones creados anteriormente. Normalmente el propio entorno de diseño asocia internamente el modelo lógico de cada celda al entorno de simulación. De esta forma lo que se tiene es una simulación funcional, donde no se tienen en cuenta los retrasos. Con esta simulación se comprueba que el diseño realizado tiene la funcionalidad deseada.
- Una vez comprobado funcionalmente el diseño se pasa a su implementación. En este caso se genera la forma interna en la que habrá de programarse el dispositivo para conseguir la funcionalidad deseada.
- Una vez implementado el diseño se realiza la simulación temporal. Del resultado de la implementación y del modelo que tiene la herramienta del dispositivo, se extrae el comportamiento temporal del dispositivo, de forma que junto con los patrones anteriormente creados pueda realizarse una simulación temporal. En esta simulación, además de comprobar que los retrasos no afectan al comportamiento pueden medirse los retrasos.
- Finalmente, una vez comprobada la funcionalidad, se pasa al último paso: la programación. En este paso se vuelca la información sobre el dispositivo y se comprueba el funcionamiento de forma experimental.

Parte 1: Estudio teórico

El objetivo es diseñar un circuito secuencial que cuente de forma ascendente en formato BCD y que muestre la cuenta en un display 7 segmentos. El diseño constará de dos circuitos:

- Un contador BCD
- Un convertidor de código de BCD a 7-segmentos.

Para el contador BCD se va a utilizar una celda de librería (CD4RE), pero el decodificador BCD a 7-segmentos vamos a construirlo.

Diseñe un convertidor de BCD a 7-segmentos. Siga los siguientes pasos, escribiéndolos en hojas aparte:

- Dibuje la tabla de verdad del circuito. Para ello tenga en cuenta que cada segmento del display se enciende cuando hay un '0' en su entrada.
- Diseñelo utilizando un decodificador de 4 a 16 (con salidas activas en alta) y puertas OR. Debe saber que cada salida del decodificador se corresponde con un mintermino de la función. Una a cada puerta OR los minterminos correspondientes al segmento que ha de encenderse.

El componente CD4RE es un contador BCD de 4 bits con clock enable y reset síncrono:

- Dibuje un cronograma en el que se aprecie el funcionamiento del contador sabiendo que su tabla de funcionamiento es la siguiente:

Inputs			Outputs					
R	CE	C	Q3	Q2	Q1	Q0	TC	CEO
1	X	↑	0	0	0	0	0	0
0	1	↑	Inc	Inc	Inc	Inc	TC	CEO
0	0	X	No change	No change	No change	No change	TC	0
0	1	X	1	0	0	1	1	1

$TC = Q3 * !Q2 * !Q1 * Q0$
 $CEO = TC * CE$

Parte 2: Estudio Experimental

1. Introducción del diseño del convertidor de código

El primer paso consiste en introducir el diseño. Para ello se siguen los siguientes pasos:

- 5.1. Arranque el entorno de diseño haciendo doble clic sobre el icono del escritorio: “**ISE Design Suite 14.7**”.
- 5.2. Crear un nuevo proyecto (en esta herramienta los diseños se agrupan dentro de “proyectos”):

File -> New Project...

En la ventana que aparece se rellena la siguiente información:

Project Name: <nombre_del_alumno> (No introducir espacios ni acentos ni ñ)

Project Location: c:\ED

Top-Level Module Type: Schematic

Se pulsa el botón “siguiente”. En la ventana que aparece nos aseguramos de introducir la siguiente información:

Device Family: Spartan 3E

Device: XC3S100E

Package: CP132

Speed Grade: -4

Generated Simulation Language: VHDL

Pulsamos “next” y después “Finish”.

Se nos abre la ventana principal del software.

- 5.3. Una vez creado el proyecto lo primero que tenemos que hacer es añadir un diseño nuevo. En primer lugar vamos a crear el convertidor de BCD a 7-segmentos. Para ello ejecutamos el comando:

Project -> New Source...

En la parte izquierda seleccionamos “Schematic” y en la derecha le damos nombre:

File Name: dec7seg

Presionamos “siguiente” y “finalizar”.

También puede hacerse colocando el ratón encima del nombre del proyecto, y presionando el botón derecho. Se nos abre un menú en el que escogemos la opción “**New Source ...**”.

- 5.4. Tras crear el diseño, se nos abre automáticamente la ventana de captura de esquemas. Los comandos básicos dentro de este entorno son (recordatorio):

- Para colocar una puerta lógica: **Add -> Symbol**
(se selecciona en la sección “symbol” que aparece a la izquierda)
- Para dibujar una conexión entre dos puertas: **Add -> wire**
- Para poner una conexión de entrada o salida: **Add -> I/O Marker**
(se coloca automáticamente de entrada o de salida)
- Para salvar el diseño: **File -> Save**

- Nombrar una línea de conexión (*wire*): picar dos veces sobre el cable y poner el nombre en la ventana que se abre.
- Para hacer un zoom sobre una zona: pulsar F9 y hacer un rectángulo en la zona a visualizar.
- Para ver todo el diseño: pulsar F6.
- Para dejar de aplicar un comando: presionar la tecla “Escape”.

Los componentes a utilizar para la realización del diseño son:

Elemento	Categoría	Símbolo
Decodificador 4 a 16	Decoder	d4_16e
Puerta OR de 2 entradas	Logic	or2
Puerta OR de 3 entradas	Logic	or3
Puerta OR de 4 entradas	Logic	or4
Puerta OR de 6 entradas	Logic	or6
Tensión a '1' lógico	General	vcc

- 5.5. Colocamos e interconectamos todos los componentes. En la entrada “E” del decodificador colocamos un ‘1’ lógico con el elemento “vdd”. Ponemos además un I/O Marker en cada entrada y salida. Nos aseguramos que tienen la polaridad correcta (entrada o salida) y le cambiamos los nombres (les ponemos unos que nos aporten información). El cambio de nombres se hace en la ventana que aparece tras picar dos veces sobre el “I/O Marker”.

Realizamos una simulación funcional siguiendo los siguientes pasos:

- 5.6. Añadimos una nueva “source” que contenga los patrones de simulación. Para ello ponemos el ratón sobre el nombre del componente “dec7seg”, presionamos el botón derecho del ratón y en el menú que aparece seleccionamos “New Source...”. En la ventana que aparece seleccionamos en la izquierda “VHDL Test Bench” y como nombre de fichero introducimos: “tb_dec7seg”. Presionamos “next” y como Associate Source seleccionamos “dec7seg”. Presionamos “next” y “finish”.
- 5.7. Introducimos los patrones de simulación. Esto nos abre un fichero de texto (un test_bench en VHDL) en el que hay que introducir los patrones de simulación. Modifique el fichero para que quede de la siguiente forma:

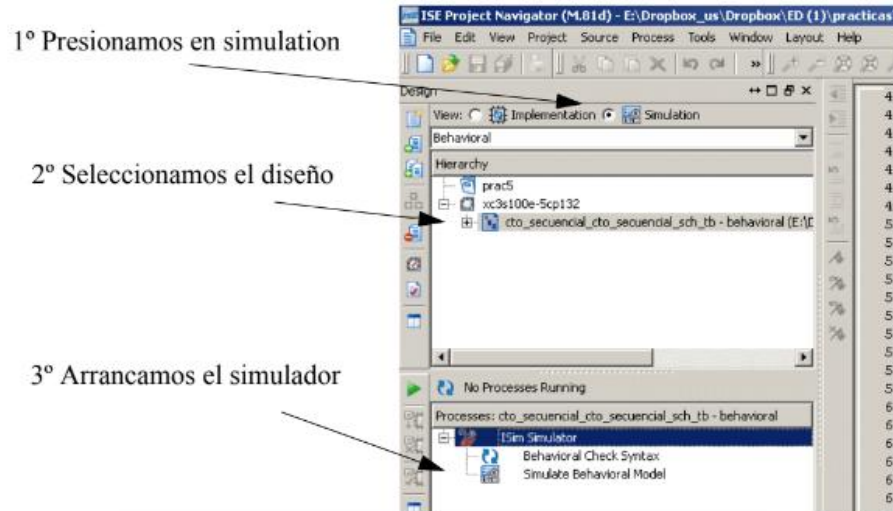
```
-- *** Test Bench - User Defined Section ***
tb : PROCESS
    variable i: unsigned(3 downto 0) := "0000";
BEGIN
    wait for 50 ns;
    i := i + 1;
    a0 <= i(0);
    a1 <= i(1);
    a2 <= i(2);
    a3 <= i(3);

END PROCESS;
-- *** End Test Bench - User Defined Section ***
```

5.8. Salvamos el fichero.

5.9. Arrancamos el simulador

Una vez salvado los patrones de simulación, presionamos en la zona “View” (parte superior izquierda) sobre “simulation”, después seleccionamos el diseño (tb_cto_secuencial) y finalmente arrancamos el simulador picando dos veces sobre “Simulate Behavioral Model”.



5.10. El simulador automáticamente muestra las entradas y salidas. Para ver todos los patrones ponga el cursor sobre la ventana “Waves” y presione F6.

Compruebe que la simulación produce los resultados esperados.

Finalmente avise al profesor para que lo compruebe.

5.11. Una vez comprobado por el profesor puede cerrar el simulador.

5.12. Una vez comprobado el correcto funcionamiento del circuito le creamos un símbolo. Para ello seleccionamos la vista “Implementation” (parte superior izquierda), seleccionamos el componente “dec7seg” y en la parte de “Processes” en la zona de “Design Utilities” ejecutamos “Create Schematic Symbol”.

6. Introducción del diseño completo

6.1. Creamos un segundo componente al que llamaremos “contador_modulo10”. Para ello seguimos los pasos:

Project -> New Source...

En la parte izquierda seleccionamos “Schematic” y en la derecha le damos nombre:

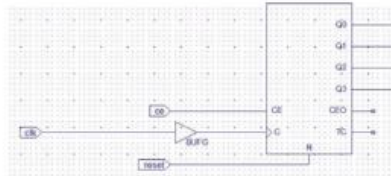
File Name: contador_modulo_10

Presionamos “siguiente” y “finalizar”.

6.2. Introducimos el esquemático. Los componentes a instanciar son tres:

Elemento	Categoría	Símbolo
Contador BCD	Counter	cd4re
Convertidor BCD a 7 segmentos		dec7seg
Buffer	Buffer	bufg

NOTA: Para que funcione la señal de reloj en pin que no es de reloj hay que colocar dos buffers, un “bufg” tal como se muestra en la figura:



6.3. Colocamos los dos componentes principales y los interconectamos. Finalmente les ponemos los I/O Makers en las entradas y salidas y les ponemos nombres. El display 7 segmento tiene una señal de activación. Esta señal la activaremos con un conmutador. Como no se puede conectar directamente una entrada a una salida ponemos un buffer.

Una vez terminado el diseño lo salvamos.

6.4. Sobre este circuito se realizará simulación sólo si hay tiempo. Se seguirán los pasos seguidos anteriormente.

6.5. Una vez terminado el diseño generamos un fichero de restricciones en el que decimos la localización de las entradas y las salidas. Las entradas las vamos a conectar a conmutadores y a un pulsador y las salidas al display 7 segmentos. La localización de dichos pines es:

Conmutadores de entrada		LEDs	
nombre	posición	segmento	posición
SW0 (reset)	P11	a	L14
SW1 (ce)	L3	b	H12
SW2 (enable)	K3	c	N14
BTN0 (clk)	G12	d	N11
		e	P12
		f	L13
		g	M12
		AN0 (enable display)	F12

Esto se hace de la siguiente forma:

6.6. Colocamos el ratón sobre el diseño y añadimos una nueva fuente que esta vez es del tipo “Implementation Constraint File”. Este fichero lo asociamos al diseño “contador_modulo_10”.

Esto crea una unidad llamada contador_modulo_10.ucf. Se nos abre el fichero para poner las restricciones. Lo rellenamos de la siguiente forma:

```

NET "CE" LOC = L3;
NET "clk" LOC = G12;
NET "enable" LOC = K3;
NET "reset" LOC = P11;

NET "a" LOC = L14;
NET "b" LOC = H12;
NET "c" LOC = N14;
NET "d" LOC = N11;
NET "e" LOC = P12;
NET "f" LOC = L13;
NET "g" LOC = M12;

NET "enable_0" LOC = F12;
NET "clk" CLOCK_DEDICATED_ROUTE = FALSE;

```

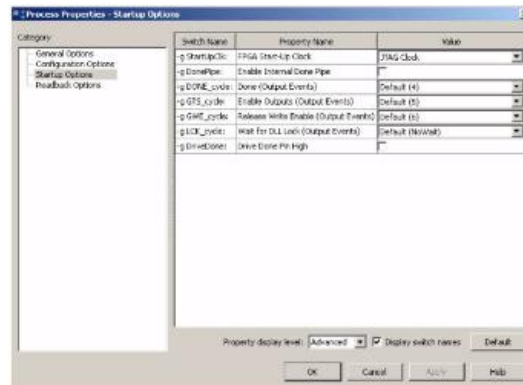
6.7. Una vez introducidos los datos salvamos y cerramos la ventana.

Nos aseguraremos que todos los pasos se han realizado correctamente realizando la implementación del circuito. Para ello:

6.8. Nos aseguramos de que en la ventana “Sources in Project” está seleccionado el diseño que se está realizando.

6.9. En la ventana “Processes for Source” nos posicionamos sobre “Generating Programming File” y pulsando el botón de la derecha, en el menú que aparece, seleccionamos “Process Properties”.

En la ventana que aparece, en la categoría “Startup Options” ponemos la variable “FPGA Start-up Clock” a “JTAG Clock”.

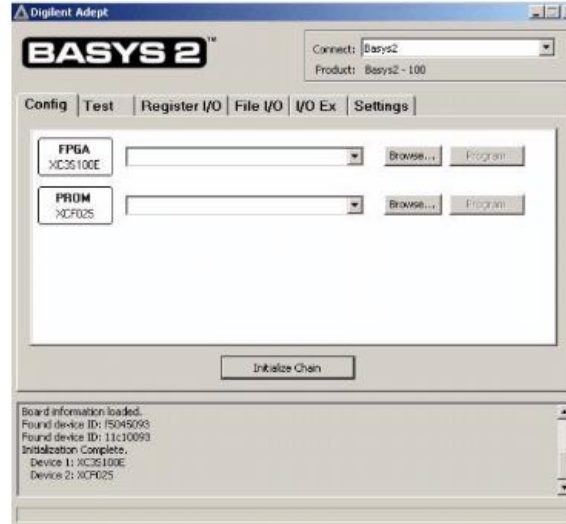


Una vez cambiado seleccionamos “Run” en Generation Programming File. Si todo termina correctamente podemos seguir adelante. En caso contrario hay que corregir los errores.

7. Programación del dispositivo y comprobación del funcionamiento

El proceso de programación del dispositivo tiene el siguiente orden de pasos:

- 7.1. En primer lugar se conecta la placa al PC mediante el cable USB.
NOTA: La placa se coge por los bordes. No se tocan los componentes de la parte superior ni de la parte inferior.
- 7.2. Se arranca el software de programación:
Inicio -> Programas -> Digilent -> Adept -> Adept



En esta ventana se reconocen los dos dispositivos que hay en la placa: la FPGA y una memoria PROM. Nosotros vamos a programar la FPGA. Buscamos el fichero .bit que ha creado el software de Xilinx, y pulsamos “Program”.

- 7.3. Finalmente avise al profesor para que compruebe el funcionamiento.