

Práctica 4:

Diseño y realización de circuitos secuenciales mediante circuitos programables (I)

Introducción

En esta práctica se diseñará y simulará un circuito secuencial con la ayuda de un entorno informático y además se realizará este diseño mediante un circuito programable comprobando su funcionalidad mediante el encendido de diodos LED. **Es esencial que, antes de acudir al laboratorio, el alumno haya leído atentamente y comprendido todo el contenido de esta práctica.**

Objetivos

La realización de diseños electrónicos en general, y de diseños digitales en particular, se apoya fuertemente en programas informáticos, denominados comúnmente **herramientas de CAD**. Estos programas permiten la introducción de circuitos usando varios métodos, siendo los más habituales las descripciones mediante lenguajes de descripción de hardware (HDLs) y las capturas de esquemas (método a utilizar en esta práctica). La verificación de estas descripciones se lleva a cabo, en todos los casos, empleando simuladores.

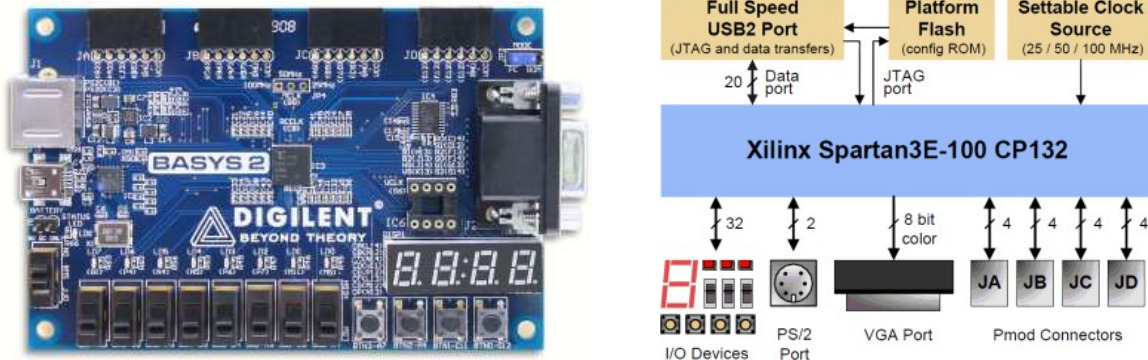
El objetivo de esta práctica es que el alumno tome contacto con una de las herramientas informáticas utilizadas para el diseño de circuitos digitales y, más que aprender todos los detalles de su utilización, aprenda cual es la metodología de diseño de circuitos digitales. Para ello diseñará un pequeño circuito y comprobará mediante simulación su correcto funcionamiento.

En esta práctica va a utilizarse el entorno de diseño Xilinx ISE WebPack. Xilinx es uno de los fabricantes más importantes de dispositivos programables de alta capacidad (CPLDs y FPGAs) y suministra un entorno propio para el diseño y programación de dichos dispositivos. Dentro de este entorno pueden realizarse todas las tareas necesarias para el diseño y programación de un dispositivo: introducción del diseño (bien por captura de esquemas o bien mediante una descripción HDL), síntesis, mapeado sobre el dispositivo y, en su caso, programación del dispositivo. Para la simulación integra dentro del entorno el simulador ISim. Con él pueden realizarse tanto simulaciones funcionales (donde se consideran a las puertas como dispositivos ideales, y por lo tanto no se tienen en cuenta los retrasos) como temporales (teniendo en cuenta los retrasos que tendría el circuito una vez programado).

El entorno ISE WebPack es gratuito y que puede obtenerse en Internet en la dirección de Xilinx: www.xilinx.com.

Una vez comprobada mediante simulación la corrección del diseño se pasará a una implementación en un circuito programable. El dispositivo que se va a utilizar es una FPGA Spartan3E XC3S100 de

Xilinx con una capacidad interna de 100.000 puertas equivalentes. Este dispositivo está montado en una placa de circuito impreso en la cual, además, del circuito se dispone de conmutadores, pulsadores y leds con los que generar entradas y visualizar salidas. La programación se realiza mediante la conexión de un cable del PC a la placa en la que está montado el circuito.



Fotografía y diagrama de bloques de la placa a utilizar.

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 1
Consiste en la introducción y simulación del correspondiente diseño.
- Parte 3: Estudio experimental 2
Consiste en la programación del circuito y su verificación visual del funcionamiento. Esta última parte se hará guiada por el profesor de prácticas.

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.
- 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.

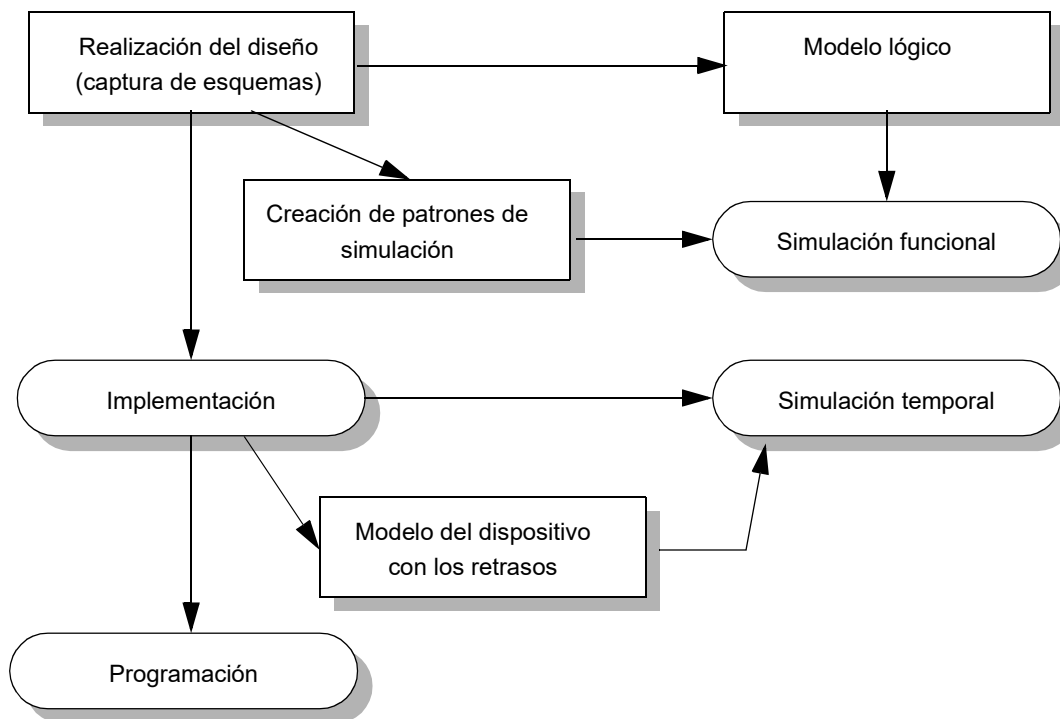


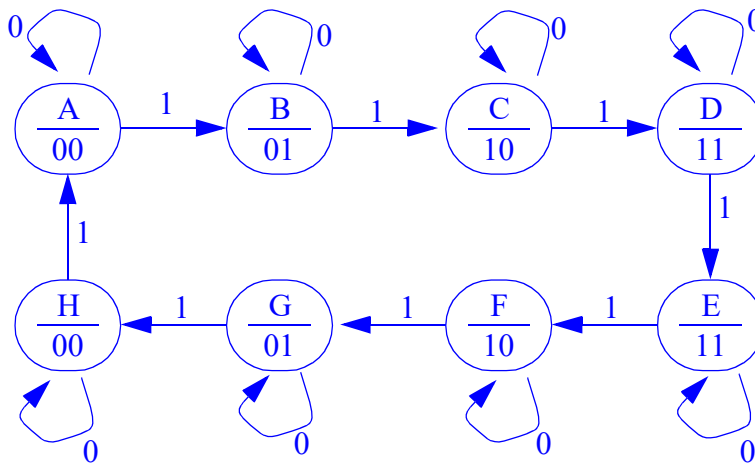
Figura 1: Esquema de la realización de un diseño mediante herramientas de CAD.

- 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

Diseñar un circuito secuencial, en forma de máquina de Moore, con una entrada (además de la entrada de reloj) y dos salidas, que realice la siguiente funcionalidad:

- Cuando la entrada está a '1' el circuito cambiará las salidas de la siguiente forma: 00, 01, 10, 11, 11, 10, 01, 00, 00, 01, ... (y así sucesivamente).
- Cuando la entrada está a '0' el circuito se queda en el estado de cuenta en el que se encuentre.
- Dibujar el diagrama de estados (máquina de Moore).



- Obtener la tabla de estados correspondiente.

	x			
s		0	1	
A	A	B	00	
B	B	C	01	
C	C	D	10	
D	D	E	11	
E	E	F	11	
F	F	G	10	
G	G	H	01	
H	H	A	00	
		NS	Z_1Z_0	

	x			
$q_2q_1q_0$		0	1	
000	000	001	00	
001	001	010	01	
010	010	011	10	
011	011	100	11	
100	100	101	11	
101	101	110	10	
110	110	111	01	
111	111	000	00	
		$Q_2Q_1Q_0$	Z_1Z_0	

	x			
$q_2q_1q_0$		0	1	
000	000	001	00	
001	001	010	01	
011	011	100	11	
010	010	011	10	
110	110	111	01	
111	111	000	00	
101	101	110	10	
100	100	101	11	
		$Q_2Q_1Q_0$	Z_1Z_0	

- Realizar el diseño utilizando biestables JK disparados por flanco de subida y con señal de puesta a cero asíncrona activa en alto.

x $q_2q_1q_0$	0	1	
000	0-	0-	
001	0-	0-	
011	0-	1-	
010	0-	0-	
110	-0	-0	
111	-0	-1	
101	-0	-0	
100	-0	-0	

J_2K_2

$$J_2 = xq_1q_0$$

$$K_2 = xq_1q_0$$

x $q_2q_1q_0$	0	1	
000	0-	0-	
001	0-	1-	
011	-0	-1	
010	-0	-0	
110	-0	-0	
111	-0	-1	
101	0-	1-	
100	0-	0-	

J_1K_1

$$J_1 = xq_0$$

$$K_1 = xq_0$$

x $q_2q_1q_0$	0	1	
000	0-	1-	00
001	-0	-1	01
011	-0	-1	11
010	0-	1-	10
110	0-	1-	01
111	-0	-1	00
101	-0	-1	10
100	0-	1-	11

J_0K_0

$$J_0 = x$$

$$K_0 = x$$

z_1z_0

$$z_1 = \overline{q_2}q_1 + q_2\overline{q_1} = q_2 \oplus q_1$$

$$z_0 = \overline{q_2}q_0 + q_2\overline{q_0} = q_2 \oplus q_0$$

- Proponer unos patrones de simulación con los que verificar adecuadamente el diseño. En estos patrones habrá de tenerse en cuenta la señal de reloj.

Parte 2: Estudio Experimental 1

1. Introducción del diseño

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

- 1.1. Arranque el entorno de diseño haciendo doble clic sobre el icono del escritorio: “ISE Design Suite 14.7”.
- 1.2. 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 en blanco 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:

The screenshot shows the ISE Design Suite software interface. On the left, there are four labels with arrows pointing to specific areas of the interface:

- Comandos** points to the menu bar.
- Zona de componentes** points to the Hierarchy pane.
- Zona de procesos** points to the Processes pane.
- Mensajes** points to the Console window.

The Design Summary window is open, showing the following information:

division Project SI	
Project File:	laboratorio12.xise
Module Name:	division
Target Device:	xc3s500e-4fg320
Product Version:	ISE 12.1
Design Goal:	Timing Performance
Design Strategy:	Performance with Physical Synthesis
Environment:	System Settings

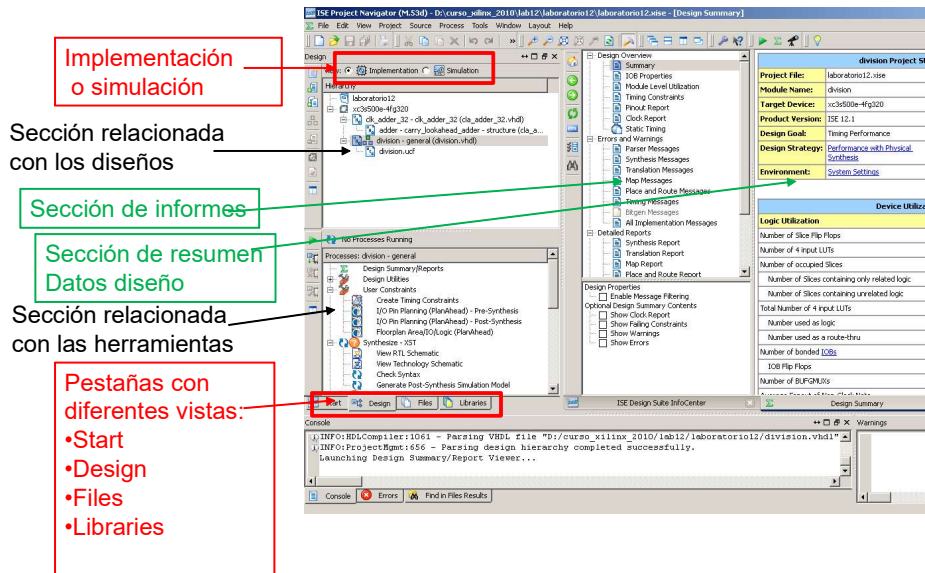
Device Utilize	
Logic Utilization	
Number of Slice Flip Flops	
Number of 4 input LUTs	
Number of occupied Slices	
Number of Slices containing only related logic	
Number of Slices containing unrelated logic	
Total Number of 4 input LUTs	
Number used as logic	
Number used as a route-thru	
Number of bonded IOBs	
IOB Flip Flops	
Number of BUFGMUXs	

The Console window at the bottom shows the following messages:

```

INFO:HDLCompiler:1061 - Parsing VHDL file "D:\curso_xilinx_2010\lab12\laboratorio12\division.vhdl"
INFO:ProjectMgmt:656 - Parsing design hierarchy completed successfully.
Launching Design Summary/Report Viewer...
    
```

En la siguiente ventana se ofrece más información.



1.3. Una vez creado el proyecto lo primero que tenemos que hacer es añadir un diseño nuevo. Para ello ejecutamos el comando:

Project -> New Source...

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

File Name: cto_secuencial

Presionamos “**next**” y “**finish**”.

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 ...**”.

1.4. Tras crear el diseño, se nos abre automáticamente el entorno de captura de esquemas.

Los comandos básicos dentro de este entorno son:

- Para colocar una puerta lógica: **Add -> Symbol**
(se selecciona en la sección “symbol” que aparece en la zona de 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”.

El proceso a seguir es el siguiente: primero se colocan los componentes, después se interconectan y finalmente se colocan los I/O Markers.

Los componentes a utilizar para la realización del diseño son:

Elemento	Categoría	Símbolo
Biestable JK	Flip_Flop	fjkc
Puerta AND de 2 entradas	Logic	and2
Puerta AND de 3 entradas	Logic	and3
Puerta XOR de 2 entradas	Logic	xor2

Como los biestables tienen una señal de reset asíncrono activo en alto. Estas señales las conectaremos a una misma entrada que llamaremos “reset”.

- 1.5. Para visualizar las salidas en 4 diodos led, de forma que sólo se active uno a la vez, conectamos las salidas del circuito a la entrada de un decodificador 2 a 4. El decodificador posee una entrada de habilitación activa en alto, por lo que, para que esté siempre habilitado, conectaremos a un ‘1’ lógico (símbolo “vcc”)

Elemento	Categoría	Símbolo
Decodificador 2a 4	Decoder	d2_4e
‘1’ lógico	General	vcc

- 1.6. Finalmente en cada una de las entradas y en las salidas colocamos un “I/O Marker”. Nos aseguramos que tiene 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”.

2. Simulación funcional

A continuación hay que realizar una simulación funcional. La generación de patrones ha de hacerse obligatoriamente utilizando el lenguaje de descripción de hardware VHDL. Los pasos que hay que seguir son los siguientes:

- 2.1. Añadimos una nueva “source” que va a ser la encargada de generar los patrones de entrada del diseño. Para ello añadimos una nueva fuente con el comando:

Project -> New Source...

En la parte izquierda seleccionamos “**VHDL Test Bench**” y en la derecha le damos nombre:

File Name: tb_cto_secuencial

Presionamos “**next**” y nos aseguramos que en la ventana que aparece está seleccionado “cto_secuencial” como “Associate Source”. Después presionamos “next” y “**finish**”.

- 2.2. Se nos abre una plantilla del fichero de test_bench. Hay que modificar este fichero poniendo las siguientes líneas en la zona “User Define Section”:

- Generar la señal de reloj (señal periódica de 50 Mhz de periodo):

```
clk: PROCESS
```

```

BEGIN
    reloj <= '1'; wait for 10 ns;
    reloj <= '0'; wait for 10 ns;
END PROCESS;

```

- Generar una señal de reset, en alto durante 55ns y después en bajo:
- Generar valores de '0' y '1' en la entrada para comprobar que hace el contador:

```

reset <= '1', '0' after 55 ns;
entrada <= '1',
          '0' after 400 ns,
          '1' after 450 ns,
          '0' after 600 ns;

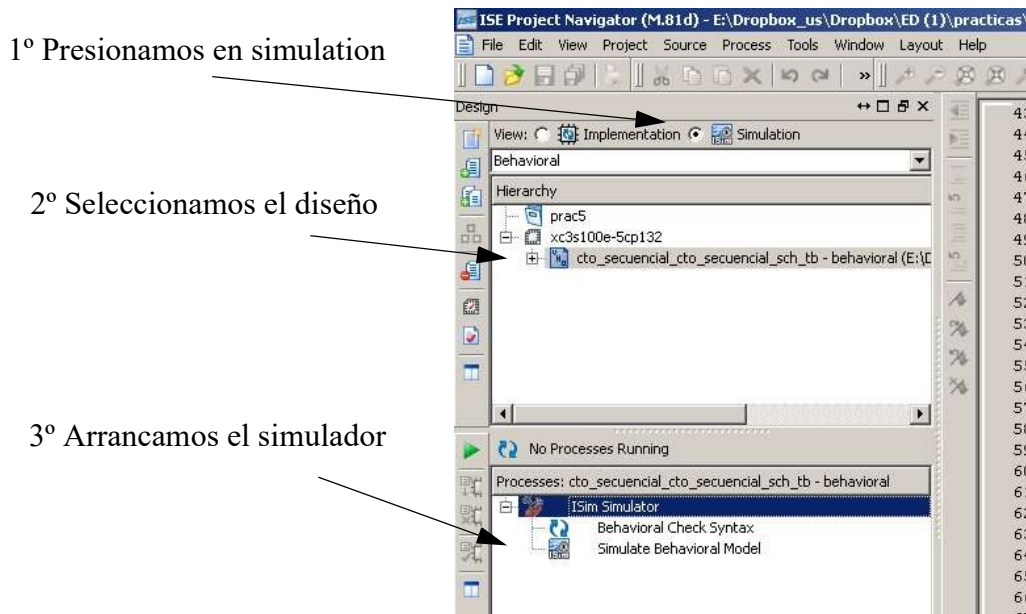
```

Debe introducir más cambios en el valor de entrada.

2.3. Una vez realizados todos los cambios salvamos el fichero.

2.4. 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”.



2.5. 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.

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

Parte 3: Estudio Experimental 2

3. Preparación del diseño

Antes de poder programar el circuito hay que realizar un último paso en el diseño. Este paso consiste en decirle a qué terminales de entrada y salida del dispositivo programable se van a conectar los terminales de entrada y salida del diseño. Esta asignación no puede hacerse en nuestro caso libremente, debido a que en la placa tanto los conmutadores como los diodos led están ya conectados a ciertos terminales del circuito. Los números de estos terminales están impresos en la propia placa (además de en la documentación correspondiente) y se resumen en la siguiente tabla:

Conmutadores de entrada		LEDs	
nombre	posición	nombre	posición
SW0 (reset)	P11	LED 0	M5
SW1 (entrada)	L3	LED 1	M11
SW2	K3	LED 2	P7
SW3	B4	LED 3	P6
SW4	G3	LED 4 (sin usar)	N5
SW5	F3	LED 5 (sin usar)	N4
SW6	E2	LED 6 (sin usar)	P4
SW7	N3	LED 7 (sin usar)	G1
Reloj (RCCLK, 1 Hz)	C8		

Para hacer la asignación de pines debemos crear un fichero de restricciones (User Constraint File):

3.1. Colocamos el ratón sobre el diseño y añadimos una nueva fuente que esta vez es del tipo “Implementation Constraint File”. Esto crea una unidad llamada “cto_secuencial2.ucf”. Picamos dos veces y se nos abre el fichero para poner las restricciones.

3.2. Rellenamos el fichero con las siguientes líneas:

```
NET "Entrada" LOC = L3;
NET "led0" LOC = M5;
NET "led1" LOC = M11;
NET "led2" LOC = P7;
NET "led3" LOC = P6;
NET "reloj" LOC = C8;
NET "reset" LOC = P11;
```

NOTA: deberá tener en cuenta los nombres puestos por el alumno en el esquemático.

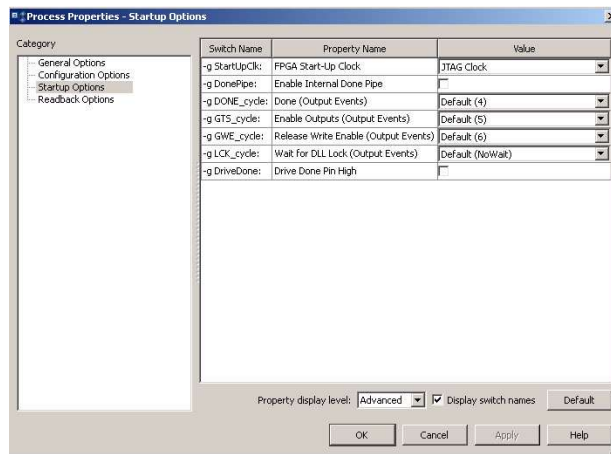
3.3. Salvamos y cerramos la ventana.

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

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

3.5. 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 Generating Programing File. Si todo termina correctamente podemos seguir adelante. En caso contrario hay que corregir los errores.

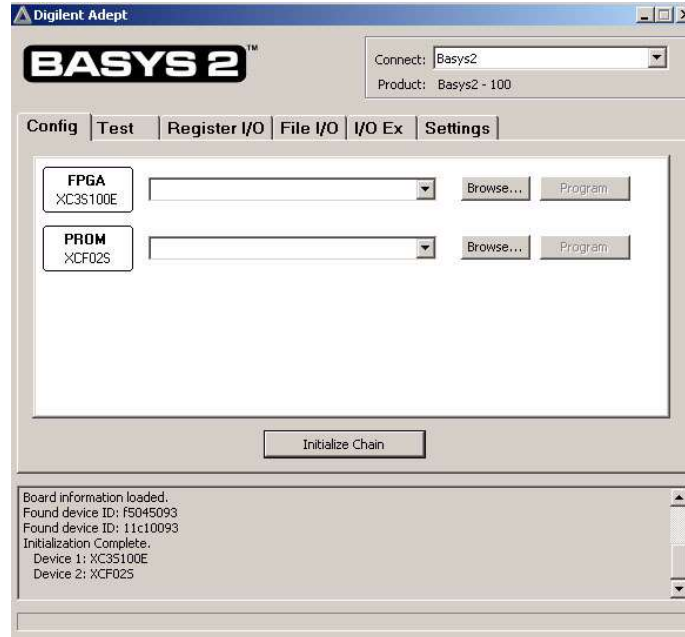
4. Programación del dispositivo y comprobación del funcionamiento

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

4.1. 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.

- 4.2. 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”.

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

