

# **Circuitos lógicos programables**

## **Práctica 1**

### **Ciclo de desarrollo con Vivado**

## Introducción

En esta práctica se utiliza el IDE Vivado para crear un sistema básico a partir de una descripción HDL para la placa Arty Z7. Se simulará, sintetizará e implementará el sistema con las opciones por defecto; y finalmente se generará el archivo de configuración y se configurará la FPGA para verificar el funcionamiento del sistema.

## Objetivos

Después de completar esta práctica será capaz de:

- Crear un proyecto en Vivado utilizando un código fuente HDL para un dispositivo ZYNQ en la placa Arty Z7-10
- Usar un archivo de restricciones predefinido (Xilinx Design Constraint (XDC) file) para restringir las ubicaciones de los pines de I/O
- Simular el sistema usando el simulador del IDE Vivado
- Sintetizar e Implementar el sistema
- Generar el archivo de configuración (bitstream)
- Configurar el dispositivo ZYNQ usando el bitstream y verificar la funcionalidad del sistema

## Procedimiento

Esta práctica está separada en pasos que consisten en sentencias generales que proveen información sobre las instrucciones detalladas que le siguen. Siga estas instrucciones detalladas para avanzar dentro de esta práctica.

Esta práctica está compuesta por 6 pasos principales: crear un proyecto usando Vivado, simular el sistema usando el simulador, sintetizar el sistema, implementar el sistema, realizar una simulación temporal y, finalmente, verificar la funcionalidad en hardware.

## Descripción del Diseño

El sistema consiste en algunas entradas que realizan ciertas funciones lógicas y sus resultados se muestran en LEDs (Figura 1).

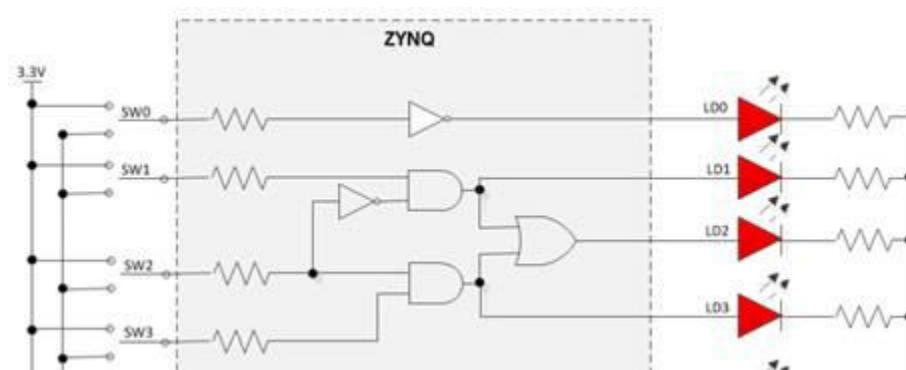
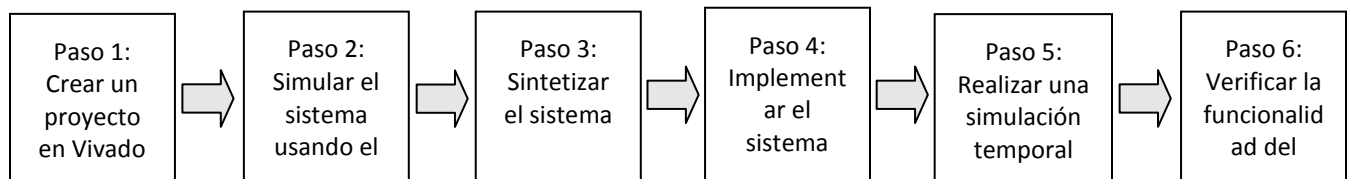


Figura 1. Sistema a implementar

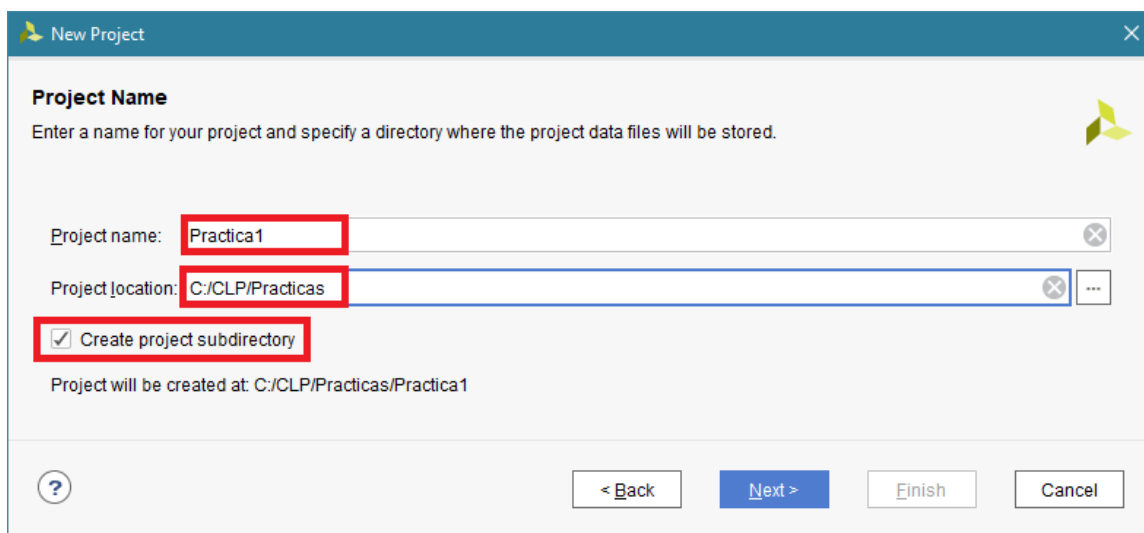
## Flujo General para esta práctica



## Crear un Proyecto en Vivado

### Paso 1

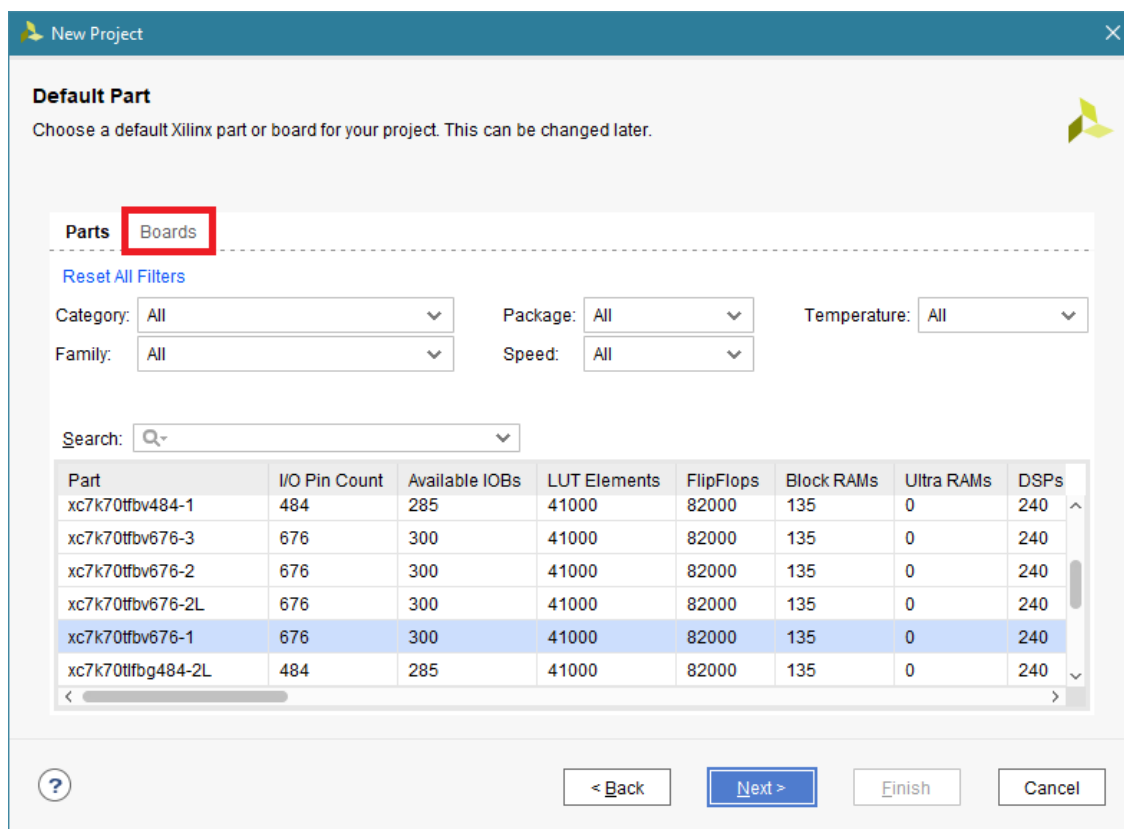
- 1-1. **Iniciar el IDE Vivado y crear un proyecto para la placa Arty Z7-10 usando el HDL VHDL. Usar los archivos lab1.vhd y lab1\_ArtyZ7\_10.xdc.**
- 1-1-1. Descomprimir los archivos de descripción de la placa Arty Z7-10. Pegarlos en el directorio C:\Xilinx\Vivado\2018.1\data\boards\board\_files (ver procedimiento en el campus)
- 1-1-2. Abrir el IDE Vivado: **Start ► All Programs ► Xilinx Design Tools ► Vivado2018.1 ► Vivado2018.1**
- 1-1-3. Presionar **Create Project** para iniciar el asistente. Aparecerá el diálogo *Create A New Vivado Project* dialog box. Presionar **Next**.
- 1-1-4. Presionar el botón Browse del campo *Project location* del formulario **New Project**, navegar a la ubicación de la practica (p.ej. **C:/CLP/Practicas**), y presionar **Select**.
- 1-1-5. Escribir **Practica1** en el campo *Project name*. Verificar que el recuadro *Create Project Subdirectory* este seleccionado. Presionar **Next**.



The screenshot shows the 'New Project' dialog box in Vivado. The 'Project Name' field is filled with 'Practica1'. The 'Project location' field is filled with 'C:/CLP/Practicas'. The 'Create project subdirectory' checkbox is checked. Below these fields, it says 'Project will be created at: C:/CLP/Practicas/Practica1'. At the bottom, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Figura 2. Formulario para el nombre del proyecto y su ubicación

- 1-1-6. Seleccionar la opción **RTL Project** en el formulario *Project Type* y presionar **Next**.
- 1-1-7. Como en este momento no se van a agregar archivos fuente ni restricciones presionamos dos veces **Next**.
- 1-1-8. Seleccionar la pestaña **Boards**



*Figura 3. Selección del kit*

**1-1-8.1.** En el cuadro **Vendor** seleccionar Digilentinc.com

**1-1-8.2.** Seleccionar la placa Arty Z7-10, presionar **Next**. Se debería observar una ventana como la siguiente

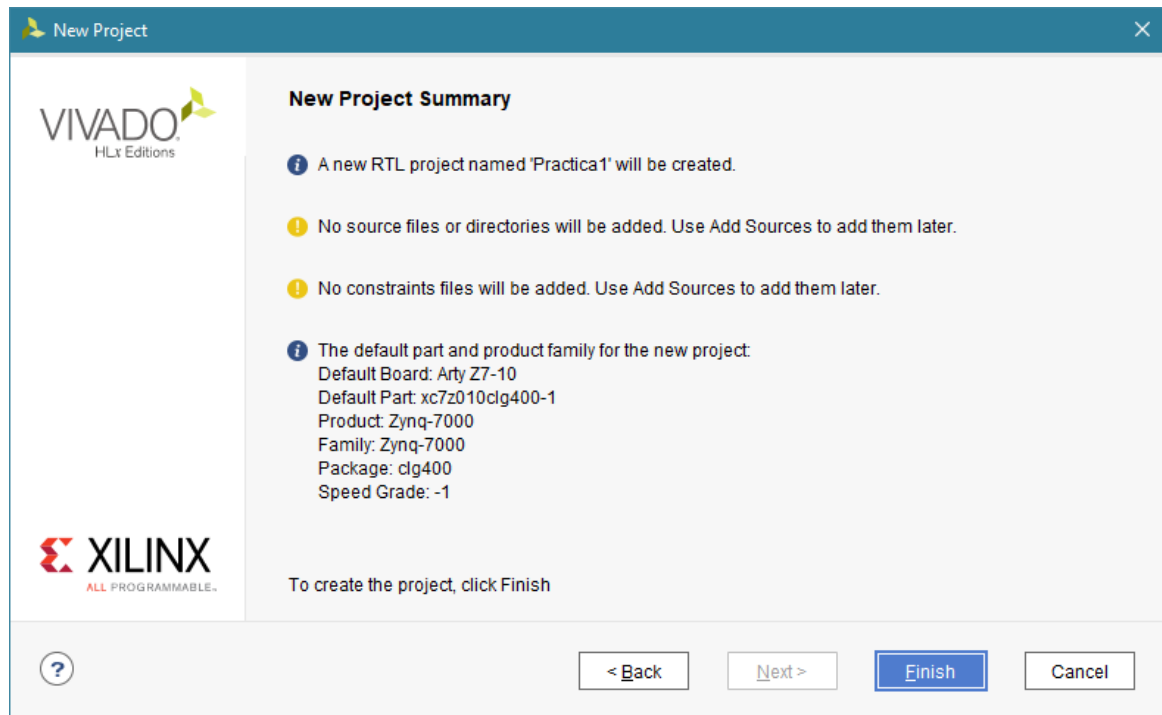


Figura 4 . Resumen de la configuración del proyecto

### 1-1-8.3. Presionar **Finish**

**1-1-9.** En la sección *Project Manager* del panel **Flow Navigator**, seleccionar *Add Sources*. En el cuadro de diálogo seleccionar *Add or create design sources*, presionar **Next**. En el cuadro de diálogo, presionar el botón **Add Files...** Navegar hasta el archivo *lab1.vhd*, presionar **OK**.

Verificar que esté seleccionado **Copy sources into project**. Presionar **Finish**.

**1-1-10.** Nuevamente, en la sección *Project Manager* del panel **Flow Navigator**, seleccionar *Add Sources*. En el cuadro de diálogo seleccionar *Add or create constraints*, presionar **Next**. En el cuadro de diálogo, presionar el botón **Add Files...** Navegar hasta el archivo *lab1\_ArtyZ7\_10.xdc*, presionar **OK**.

Verificar que esté seleccionado **Copy sources into project**. Presionar **Finish**.

*Nota: Este archivo de restricciones asigna los pines de I/O de la FPGA a los botones y leds de la placa. Esta información se puede obtener de los diagramas esquemáticos de la placa o del manual.*

**1-1-11.** Estructura del proyecto:

Usando el Explorador de Archivos, analizar el directorio del proyecto. Allí se crearon los directorios .cache y .srcs, al igual que el archivo .xpr. El directorio .cache es la ubicación física de la base de datos del proyecto. A su vez, se crean dos directorios, constr\_1 and sources\_1 dentro del directorio .srcs; dentro de ellos están los archivos .xdc (restricciones) y .vhd (código fuente HDL).

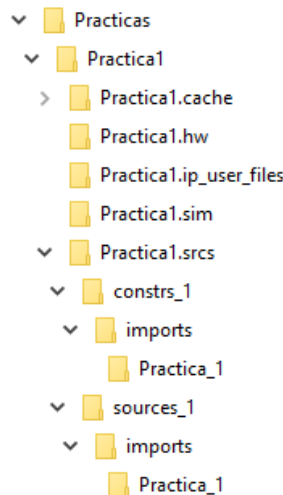


Figura 5. Estructura de archivos del proyecto

## 1-2. Abrir el código fuente lab1.vhd y analizar su contenido.

1-2-1. En el panel *Sources* seleccionar y presionar dos veces el archivo **lab1.vhd** para abrir el archivo en modo texto.

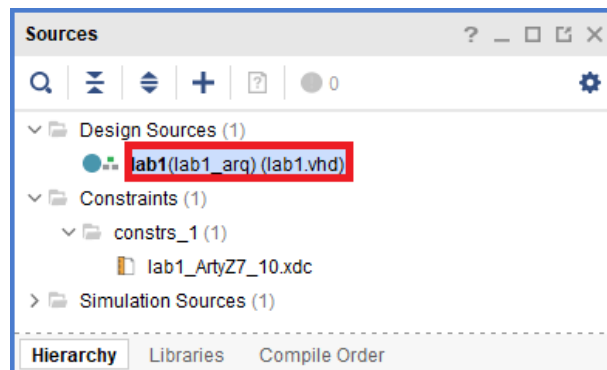


Figura 6. Abriendo el archivo de código fuente

1-2-2. Notar que el código VHDL de las líneas 9 y 10 definen la inclusión de la biblioteca IEEE. Las líneas 1 a 7 son comentarios describiendo el módulo

1-2-3. La línea 12 define el comienzo del componente (con la palabra reservada **entity**) y la línea 30 define su final (con la palabra reservada **end architecture**).

1-2-4. Las líneas 14 y 15 definen los puertos de entrada y de salida, mientras que las líneas 22 a 28 definen la lógica.

### 1-3. Abrir el archivo .xdc y analizar el contenido.

- 1-3-1. En la ventana *Sources*, abrir la carpeta *Constraints* y presionar dos veces el archivo *lab1\_ArtyZ7\_10.xdc* para abrir el archivo en modo texto.

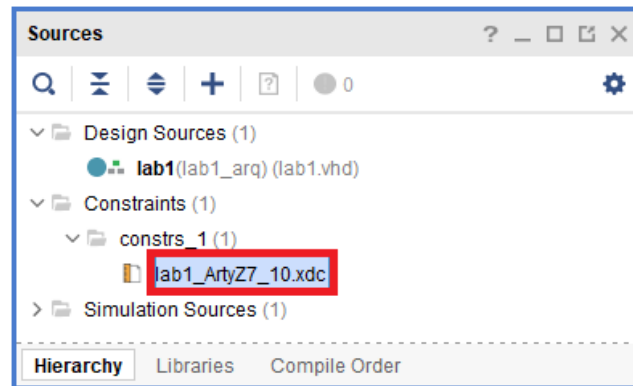


Figure 7. Abriendo el archivo de restricciones

- 1-3-2. Las líneas 5 a 8 definen las ubicaciones de los pines conectados a los botones de entrada [3:0] y las líneas 13 a 16 definen las ubicaciones de los pines conectados a los LEDs [3:0].

### 1-4. Realizar un análisis RTL del código fuente.

- 1-4-1. En el *Flow Navigator*, en la sección *RTL Analysis* expandir *Open Elaborated Design* y presionar en **Schematic**. Seleccionar **OK** en el cuadro de diálogo.

El sistema es elaborado, y se genera una vista lógica.

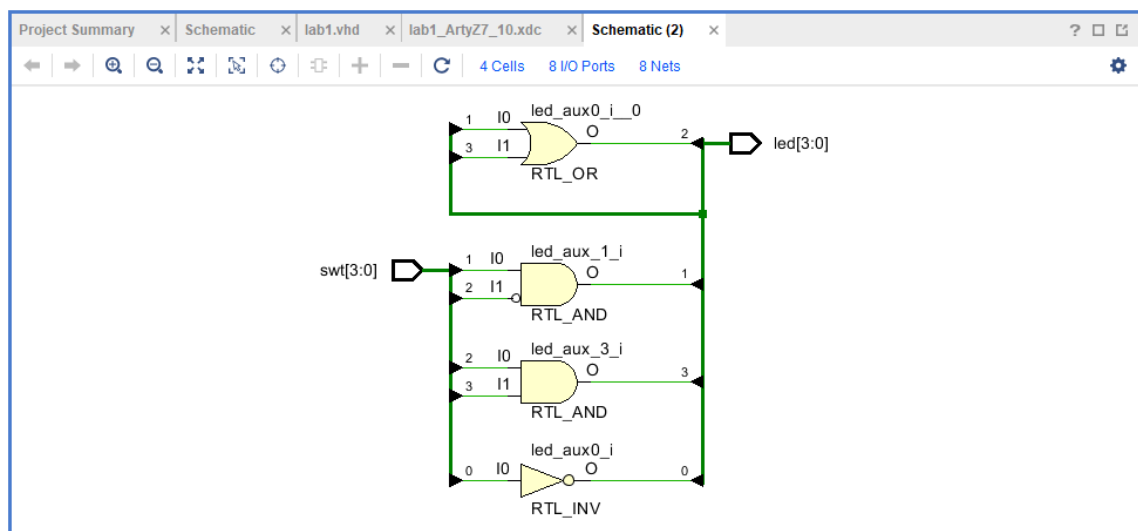


Figura 8. Vista lógica del sistema

## Simular el sistema usando el simulador de Vivado

## Paso 2

### 2-1. Agregado del archivo de pruebas lab1\_tb.vhd .

- 2-1-1. En la sección *Project Manager* del panel **Flow Navigator**, seleccionar *Add Sources*. En el cuadro de diálogo seleccionar *Add or create Simulation Sources*, presionar **Next**. En el cuadro de diálogo, presionar el botón **Add Files...** Navegar hasta el archivo *lab1\_tb.v*, presionar **OK**.

Verificar que esté seleccionado *Copy sources into project*. Presionar **Finish**.

- 2-1-2. Seleccionar la pestaña *Sources* y expandir *Simulation Sources* .

El archivo **lab1\_tb.vhd** está agregado en el grupo *Simulation Sources*, y **lab1.vhd** está ubicado en la jerarquía como instancia DUT (device under test).

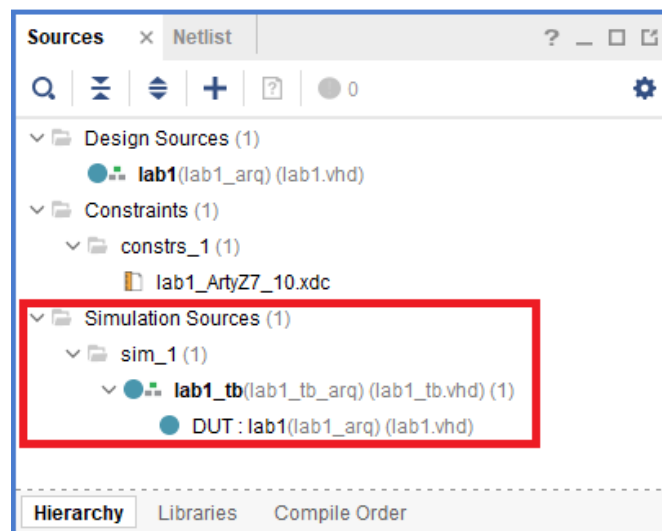


Figura 9. Jerarquía de archivos en la simulación

- 2-1-3. Usando el explorador de Archivos verificar que se creó el directorio **sim\_1** al mismo nivel que *constrs\_1* y *sources\_1* dentro de *Practica1.srcs* y que hay una copia de *lab1\_tb.vhd* en **Practica1.srcs ► sim\_1 ► imports ► lab1**.

- 2-1-4. Presionar dos veces en **lab1\_tb** en el panel *Sources* para ver su contenido

En la línea 17 se declara el componente que va a ser probado. En la línea 30 se implementa una función que genera los valores esperados de salida de acuerdo a la entrada. Las líneas 51 a 67 generan las señales de entrada y comparan la salida esperada con lo que genera el DUT. La función **report** imprime los resultados de las comparaciones durante la simulación.

### 2-2. Simular el sistema por 200 ns usando el simulador de Vivado.

- 2-2-1. En el menú **Flow Navigator**, en la sección *Project Manager*, seleccionar *Settings*.

Aparecerá el formulario **Project Settings**; seleccionar **Simulation** para ver sus propiedades.



- 2-2-2.** Seleccionar la pestaña **Simulation**, y establecer la propiedad **Simulation Run Time** en 200 ns. Asimismo seleccionar *Mixed* como *Simulator language*. Presionar **OK**.

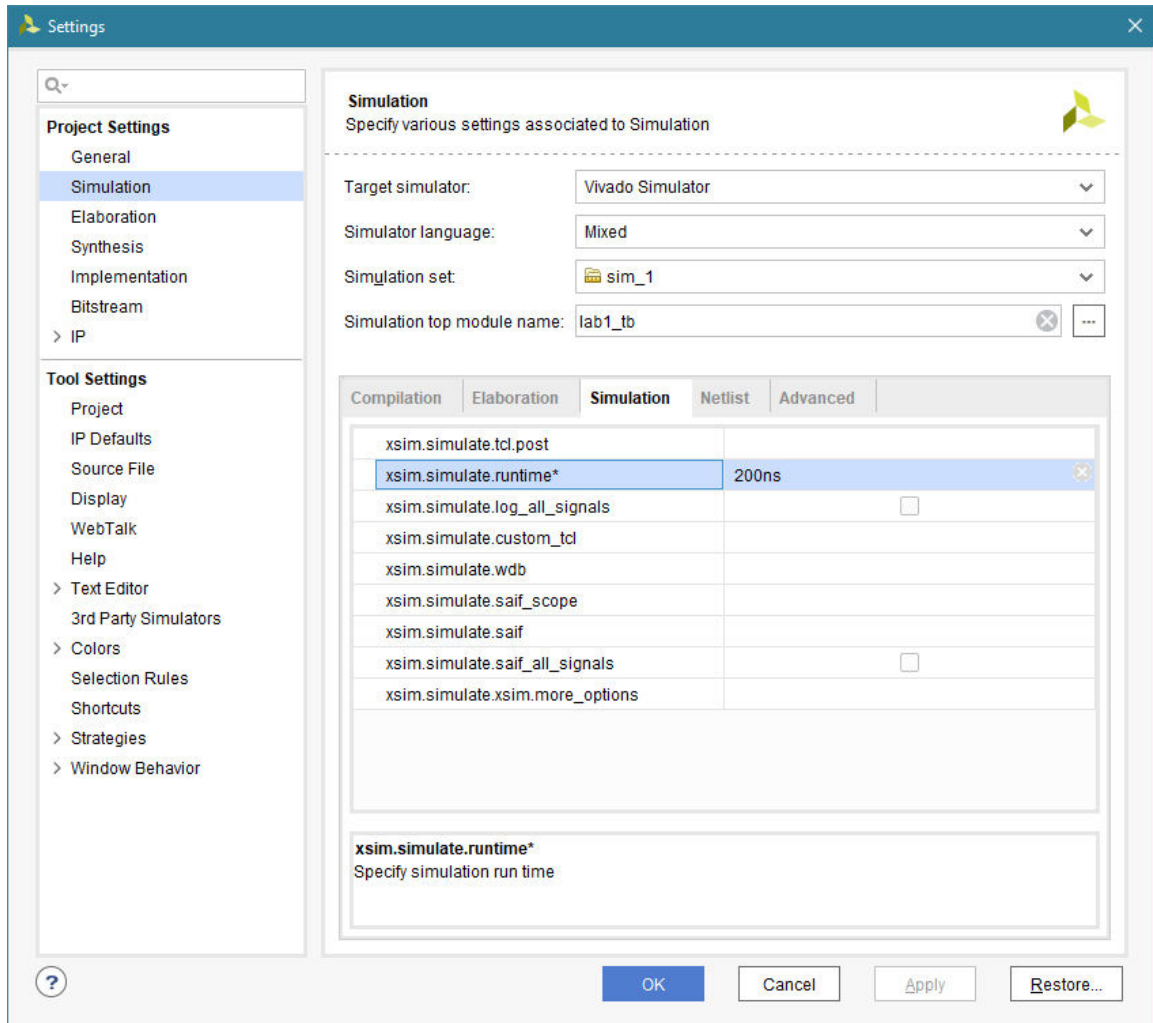


Figura 10. Estableciendo el tiempo de simulación

- 2-2-3.** En el menú **Flow Navigator**, en la sección *Simulation* seleccionar **Run Simulation ► Run Behavioral Simulation**.

El archivo de pruebas y los fuentes se compilan y se ejecuta el simulador de Vivado. Se debería ver una salida similar a esta:

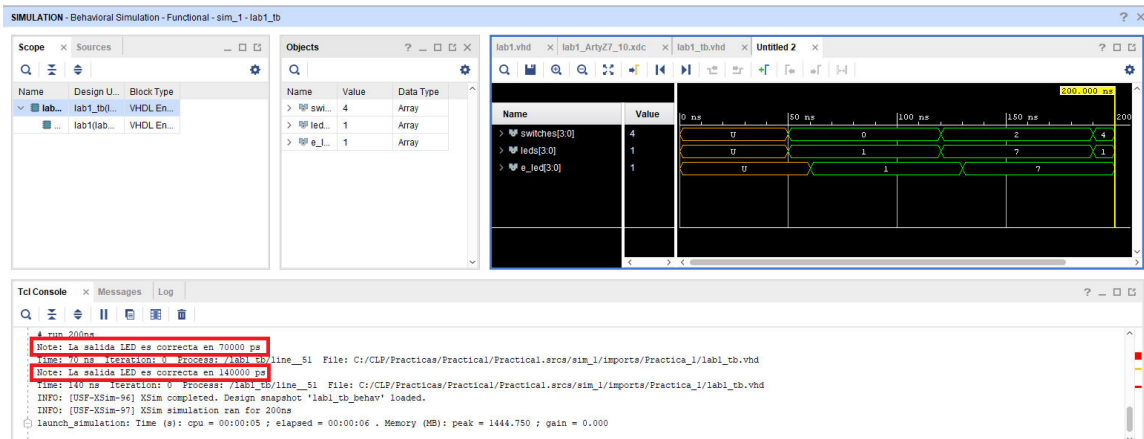


Figura 11. Salida del simulador

En la salida del simulador hay 4 vistas principales: (i) *Scope*, donde se muestra la jerarquía de la simulación, (ii) *Objects*, donde se muestran las señales del top-level, (iii) la ventana de formas de onda, y (iv) la *Tcl Console*, donde se ven los mensajes de la simulación. Notar que como el archivo de pruebas tiene autochequeo, los resultados se muestran a medida que se ejecuta la simulación.

En la estructura de archivos, el directorio **Practica1.sim** se crea debajo del directorio **Practica1**, junto con otros subdirectorios.

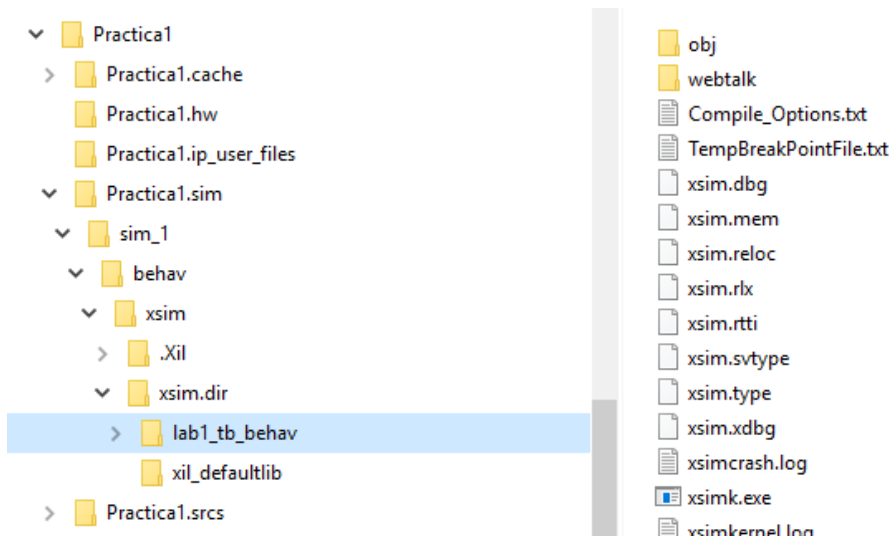


Figura 12. Estructura de directorios después de la simulación funcional

#### 2-2-4. Presionar el botón *Zoom Fit* para ver la forma de onda entera

Verificar los cambios en la salida a medida que cambia la entrada.

También se puede poner la ventana de simulación flotante presionando el botón *Float* en la parte superior derecha. Esto permite una ventana más ancha para ver las formas de onda. Para volver a poner la ventana en la interface gráfica, presionar el botón *Dock Window*.



Figure 13. Botón Float



Figure 14. Botón Dock Window

## 2-3. Cambiar el formato de las señales.

Seleccionar **i[31:0]** en la ventana de formas de onda, presionar el botón derecho, seleccionar *Radix*, y después seleccionar *Unsigned Decimal* para ver el índice del ciclo for in formato *integer*. De la misma manera cambiar el formato de **switches[3:0]** a *Hexadecimal*. Cambiar los formatos de **leds[3:0]** y **e\_led[3:0]** a *binary* para ver cada bit de salida.

## 2-4. Agregar señales para monitorear señales de más bajo nivel y continuar la simulación por 500 ns.

2-4-1. En la ventana *Scope* expandir la instancia **lab1\_tb** y seleccionar la instancia **dut**.

Las señales **swt[3:0]** y **led[3:0]** aparecerán en la ventana *Objects*.

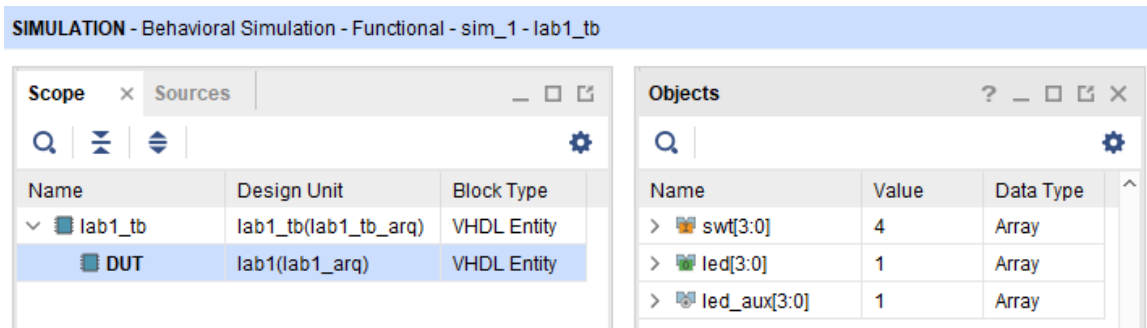


Figure 15. Selección de señales de más bajo nivel

2-4-2. Seleccionar **swt[3:0]** y **led[3:0]** y arrastrarlas a la ventana de formas de onda para monitorear esas señales de más bajo nivel.

2-4-3. En la barra superior del simulador, escribir 500 en la ventana *simulation run time*, presionar en el botón desplegable de las unidades y seleccionar *ns* dado que queremos continuar la simulación por 500 ns (700 ns en total), y presionar el botón de avance (▶).

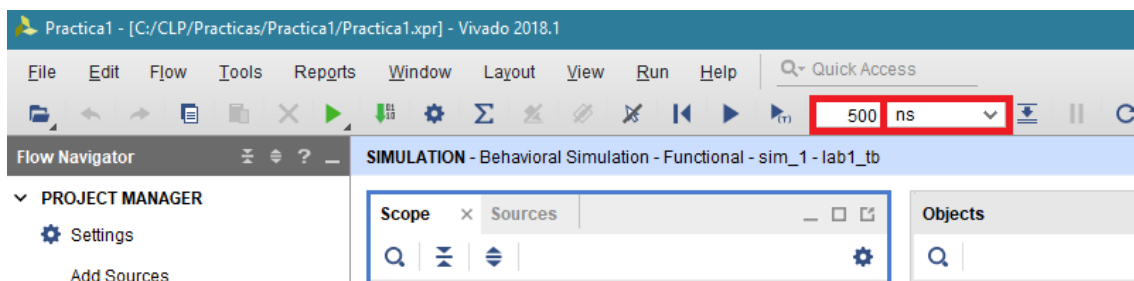


Figure 16. Cambiando el tiempo de simulación

La simulación continuará por otros 500 ns.

- 2-4-4.** Presionar en el botón *Zoom Fit* y verificar la salida (poner la ventana de formas de onda flotante para aumentar su tamaño).

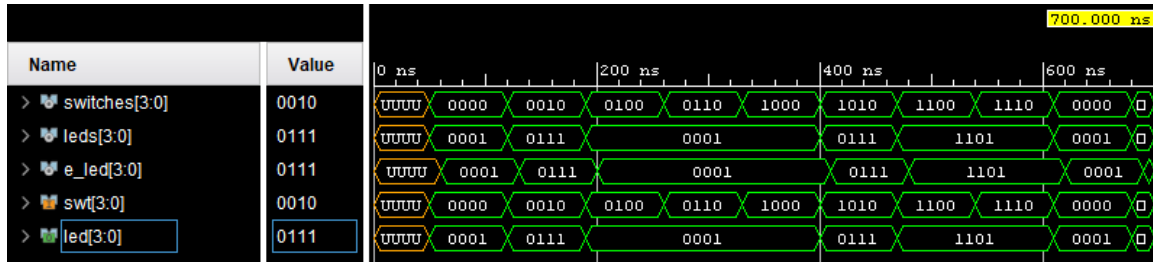


Figura 17. Simulación ejecutada por otros 500 ns

En la ventana *Tcl Console* se observa como la salida se muestra a medida que el archivo de prueba usa la función report.

```
Tcl Console x Messages Log
[Icons]
run 500 ns
Note: La salida LED es correcta en 210000 ps
Time: 210 ns Iteration: 0 Process: /lab1_tb/line__51 File: C:/CLP/Practicas/Practical/Practical.srcs/sim_1/imports/Practica_1/lab1_tb.vhd
Note: La salida LED es correcta en 280000 ps
Time: 280 ns Iteration: 0 Process: /lab1_tb/line__51 File: C:/CLP/Practicas/Practical/Practical.srcs/sim_1/imports/Practica_1/lab1_tb.vhd
Note: La salida LED es correcta en 350000 ps
Time: 350 ns Iteration: 0 Process: /lab1_tb/line__51 File: C:/CLP/Practicas/Practical/Practical.srcs/sim_1/imports/Practica_1/lab1_tb.vhd
Note: La salida LED es correcta en 420000 ps
Time: 420 ns Iteration: 0 Process: /lab1_tb/line__51 File: C:/CLP/Practicas/Practical/Practical.srcs/sim_1/imports/Practica_1/lab1_tb.vhd
Note: La salida LED es correcta en 490000 ps
Time: 490 ns Iteration: 0 Process: /lab1_tb/line__51 File: C:/CLP/Practicas/Practical/Practical.srcs/sim_1/imports/Practica_1/lab1_tb.vhd
Note: La salida LED es correcta en 560000 ps
Time: 560 ns Iteration: 0 Process: /lab1_tb/line__51 File: C:/CLP/Practicas/Practical/Practical.srcs/sim_1/imports/Practica_1/lab1_tb.vhd
Note: La salida LED es correcta en 630000 ps
Time: 630 ns Iteration: 0 Process: /lab1_tb/line__51 File: C:/CLP/Practicas/Practical/Practical.srcs/sim_1/imports/Practica_1/lab1_tb.vhd
Note: La salida LED es correcta en 700000 ps
Time: 700 ns Iteration: 0 Process: /lab1_tb/line__51 File: C:/CLP/Practicas/Practical/Practical.srcs/sim_1/imports/Practica_1/lab1_tb.vhd
```

Figura 18. Ventana *Tcl Console* luego de ejecutar la simulación por otros 500 ns

- 2-4-5.** Cerrar el simulador mediante **File ->Close Simulation**.

- 2-4-6.** Presionar **OK** y luego **Discard** para cerrar la simulación sin guardar los resultados.

---

## Sintetizar el sistema

## Paso 3

---

### 3-1. Sintetizar el sistema con la herramienta de síntesis y analizar el resumen de proyecto (Project Summary).

**3-1-1.** En el menú **Flow Navigator**, en la sección *Synthesis*, seleccionar **Run Synthesis**. Presionar **OK**.

El proceso de síntesis se realizará en el archivo lab1.vhd (y sus sub-archivos dentro de la jerarquía, si existen). Cuando se completa el proceso de síntesis, aparece el cuadro de diálogo **Synthesis Completed** con tres opciones.

**3-1-2.** Seleccionar la opción **Open Synthesized Design** y presionar **OK**, ya que queremos ver la salida del proceso de síntesis antes de pasar a la etapa de implementación.

**3-1-3.** Seleccionar la pestaña **Project Summary** y ver las distintas secciones.

Si no se ve la pestaña **Project Summary**, seleccionar **Layout ► Default Layout**. Maximizar y poner flotante la pestaña para ver toda la información

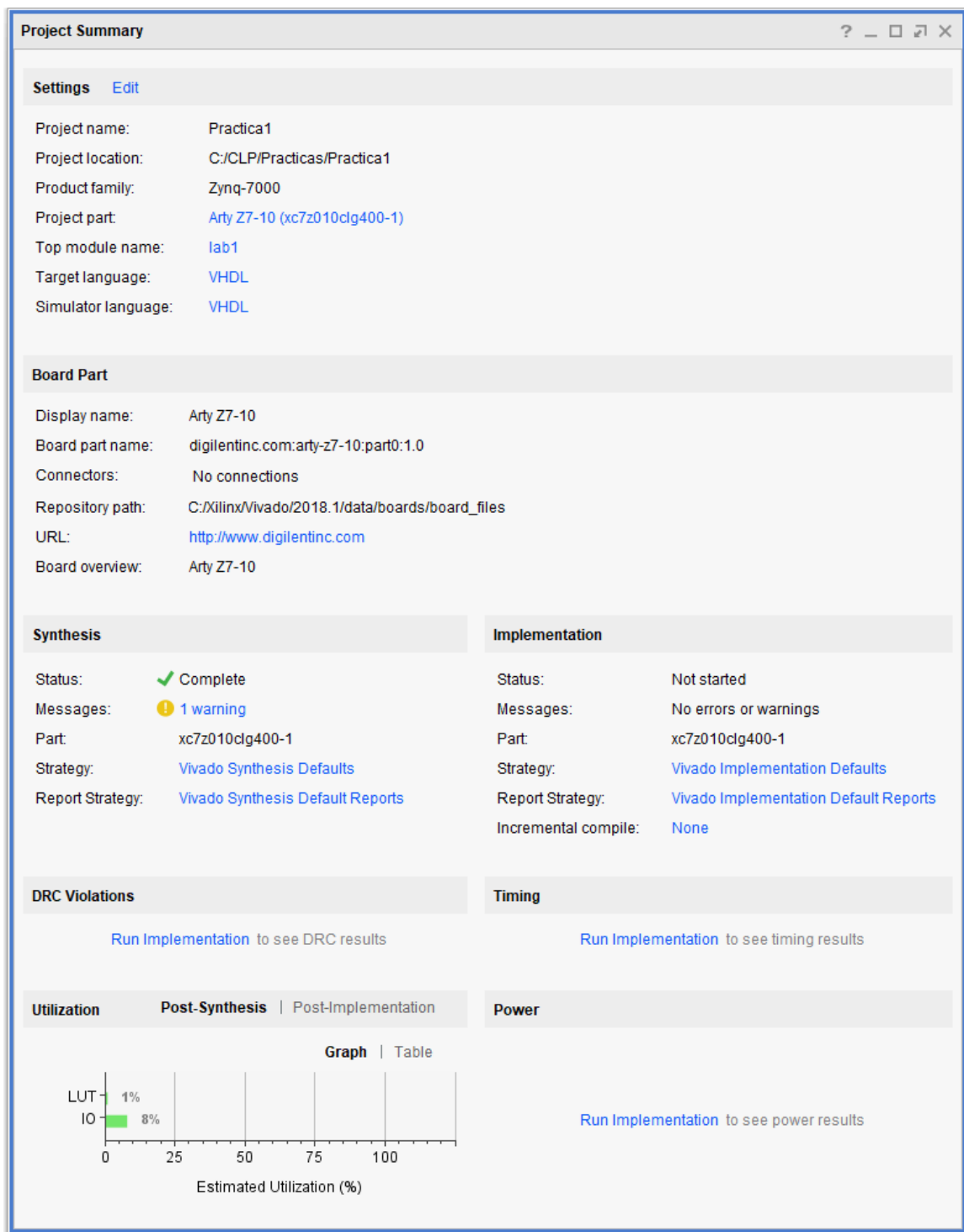


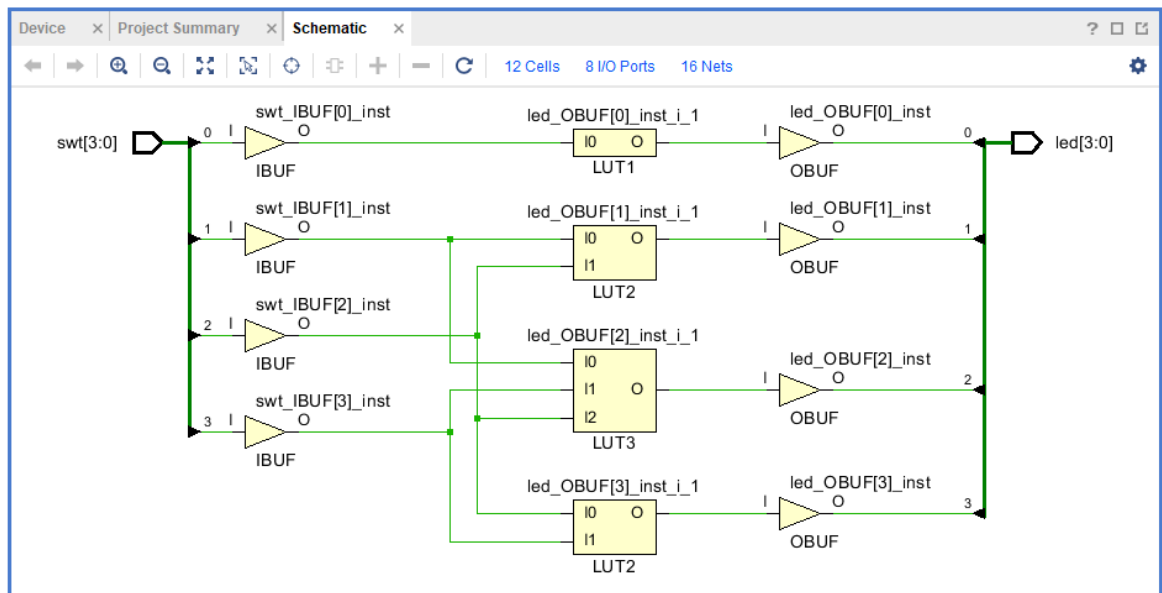
Figura 19. Pestaña Project Summary

Presionando en algunos enlaces se puede ver qué información proveen, y algunos permiten cambiar las opciones de síntesis.

**3-1-4.** En la sección **Utilization**, donde se encuentra el gráfico, presionar en el botón **Table**.

La estimación es que se utilizarán 3 LUTs y 8 IOs (4 de entrada y 4 de salida).

**3-1-5.** En el **Flow Navigator**, bajo **Synthesis** (expandir **Open Synthesized Design**, si es necesario), presionar en **Schematic** para ver el sistema sintetizado en una vista esquemática.



*Figura 20. Diagrama esquemático del sistema sintetizado*

Notar que los IBUFs y OBUFs se instancian en forma automática (se agregan) al sistema, dado que las entradas y salidas son acondicionadas (buffered). Las compuertas lógicas se implementan con LUTs (las compuertas de 1 entrada se muestra como LUT1, las de 2 entradas se muestran como LUT2, y las compuertas de 3 entradas como LUT3). Las 4 compuertas del análisis RTL se mapean a 4 LUTs en la salida del proceso de síntesis.

Mediante el explorador de archivos, verificar que el directorio **Practica1.runs** fue creado dentro del directorio **Practica1**. Dentro del directorio **runs**, se creó el directorio **synth\_1**, el cual tiene distintos archivos relacionados con el proceso de síntesis.

## Implementación del sistema

## Paso 4

### 4-1. Implementar el sistema con las opciones por defecto y analizar la salida del resumen de proyecto (Project Summary output).

#### 4-1-1. En el menú **Flow Navigator**, en la sección **Implementation**, seleccionar **Run Implementation**. Presionar **OK**.

Se ejecutará el proceso de implementación en el sistema sintetizado. Cuando se complete el proceso, aparecerá el cuadro de diálogo **Implementation Completed** con tres opciones.

#### 4-1-2. Seleccionar **Open implemented design** y presionar **OK**, ya que queremos ver el sistema implementado en la pestaña de vista de dispositivo.

#### 4-1-3. Si aparece un cuadro de diálogo, presionar **Yes**, para cerrar el sistema sintetizado.

Se abrirá la vista del sistema implementado.

#### 4-1-4. En el panel **Netlist**, seleccionar una de las redes (p.ej. led\_OBUF[2]). Ver que se muestra la red en la región de reloj X1Y1 en la pestaña de vista de dispositivo (puede ser necesario hacer zoom para verla).

#### 4-1-5. Si no está seleccionado, presionar el botón **Routing Resource** para ver el ruteo.

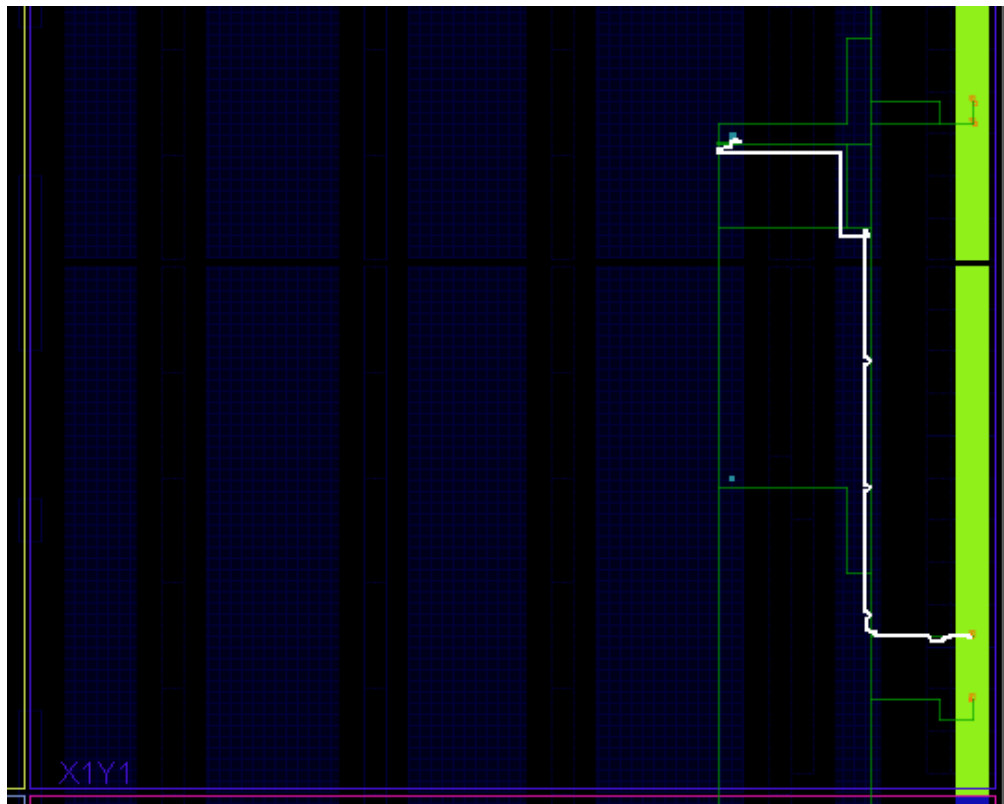


Figura 21. Vista del ruteo realizado

#### 4-1-6. Cerrar la vista del sistema implementado mediante **File ► Close Implemented Design**. Debería aparecer la pestaña **Project Summary** con los resultados.



Presionar el botón **Post-Implementation** en la parte **Utilization**. Luego presionar el botón **Table**

Se observa que la utilización de recursos es 3 LUTs y 8 IOs. También en la parte de **Timing** se puede ver que no hay definidas restricciones de temporización, ya que el circuito es puramente combinacional.

Usando el explorador de archivos, verificar que el directorio **impl\_1** se creó en el mismo nivel que **synth\_1** dentro del directorio **Practica1.runs**. El directorio **impl\_1** contiene distintos archivos de implementación, incluidos los reportes.

- 4-1-7.** En el IDE Vivado, seleccionar la pestaña **Reports** en el panel de abajo (si no está visible, presionar **Window** en la barra de menú, y seleccionar **Reports**), en la parte de **Implementation**, dentro de **impl\_1**, buscar la sección **Place Design** y presionar dos veces en **impl\_1\_place\_report\_utilization\_0**. Se abrirá el reporte mostrando la utilización de recursos. Dado que el circuito es puramente combinacional, no hay uso de registros.

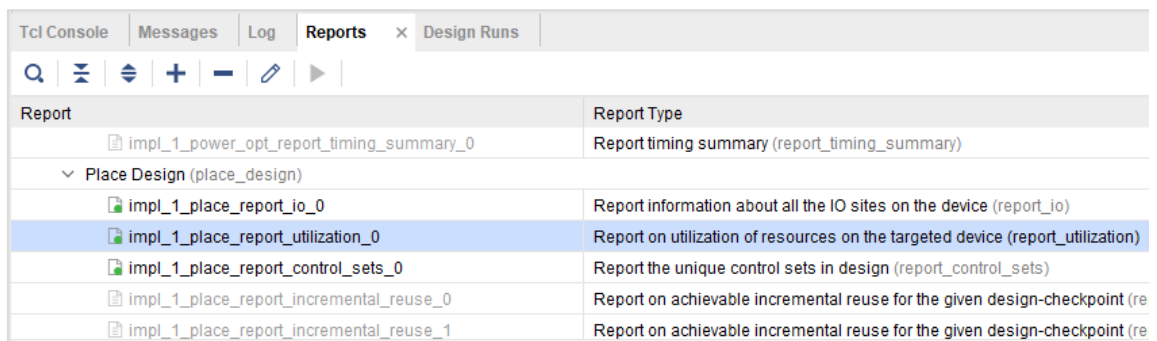


Figure 22. Apertura del reporte de utilización de recursos

## Simulación Temporal

## Paso 5

### 5-1. Run a timing simulation.

- 5-1-1.** En el menú **Flow Navigator**, en la sección **Simulation**, seleccionar Run Simulation -> Run Post-Implementation Timing Simulation. Seleccionar **Yes**.


Se iniciará el simulador de Vivado usando el sistema implementado y **lab1\_tb** como componente principal.

Usando el explorador de archivos, verificar que se creó el directorio **timing** ► **xsim** dentro de **Practica1.sim** ► **sim\_1** ► **impl**. Este directorio contiene los archivos necesarios para la simulación temporal.

- 5-1-2.** Presionar el botón **Zoom Fit** para ver las formas de onda en el rango 0 a 200 ns.

- 5-1-3.** Se pueden colocar diferentes marcadores. Presionar el botón derecho en la marca de 50 ns (hacerlo por debajo de las formas de onda, donde la entrada se fija en 0000b) y seleccionar **Markers** ► **Add Marker**.

- 5-1-4.** De la misma manera, agregar una marca alrededor de los 58ns, donde cambia la salida **(leds)**.

- 5-1-5.** También se puede agregar un marcador mediante el botón **Add Marker** (  ). Presionar el botón **Add Marker** y presionar el botón izquierdo alrededor de los 60 ns, donde cambia **e\_led**.

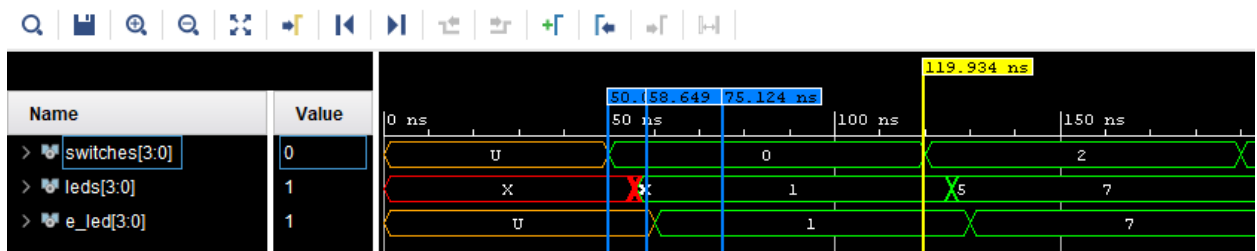


Figura 23. Agregado de marcadores

Notar que la salida **leds[3:0]** comienza a cambiar su estado en un determinado momento, y termina de hacerlo un tiempo después, lo que permite evaluar tiempos de propagación. Este tiempo dependerá de la FPGA usada.

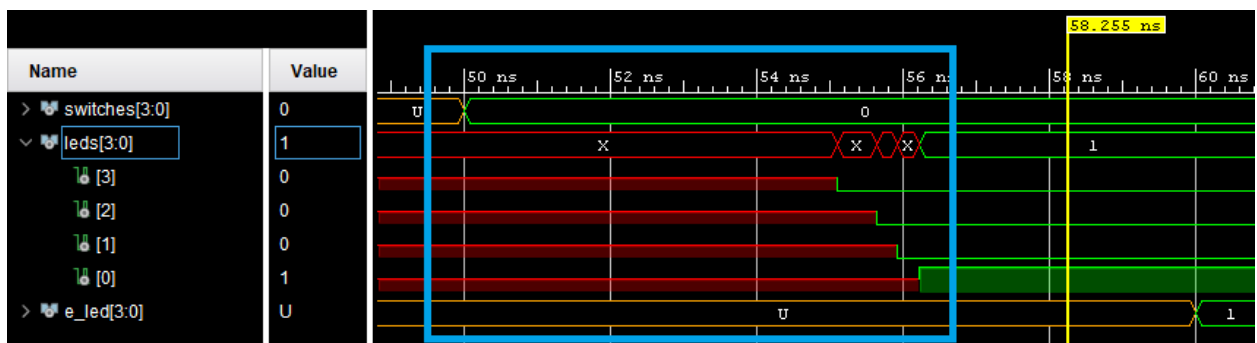


Figura 24. Visualización de retardos

- 5-1-6.** Cerrar el simulador mediante el menú **Flow Navigator**, en la sección **Simulation**, presionar el botón derecho y seleccionar **Close** (no guardar cambios).

## Generar el archivo de configuración (BitStream)

## Paso 6

### 6-1. Conectar la placa y encenderla, generar el archivo de configuración, abrir una sesión de hardware y configurar la FPGA.

#### 6-1-1. Verificar que el cable Micro-USB esté conectado al conector PROG/UART al lado del conector Ethernet.

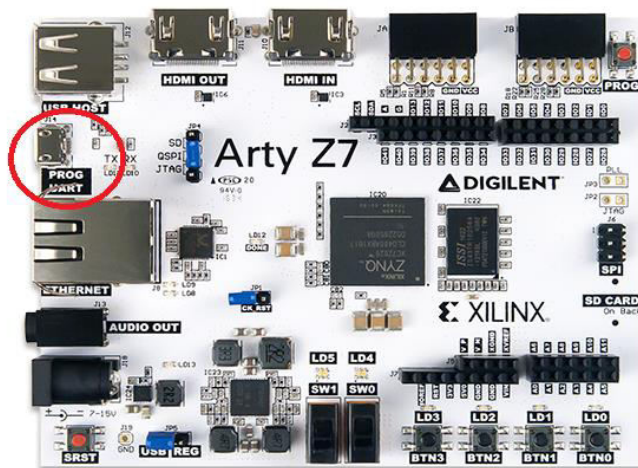


Figura 25. Conector de programación de la placa

#### 6-1-2. En el menú **Flow Navigator**, en la sección **Program and Debug**, seleccionar **Generate Bitstream**. Presionar **OK**.

Se ejecutará el proceso de generación del archivo de configuración en el sistema implementado. Cuando el proceso se complete, se abrirá el cuadro de diálogo **Bitstream Generation Completed** con 3 opciones (este proceso generará el archivo lab1.bit en el directorio impl\_1 dentro de Practica1.runs).

#### 6-1-3. Seleccionar la opción **Open Hardware Manager** y presionar **OK**.

Se abrirá la ventana Hardware Manager indicando el estado “unconnected”.

#### 6-1-4. Presionar el link **Open target** y del menú desplegable seleccionar **Auto Connect**.

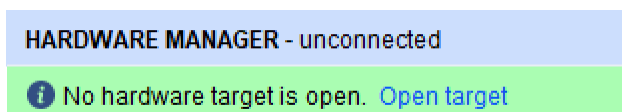


Figura 26. Hardware Manager

El estado de la sesión de hardware cambiará de Unconnected al nombre del servidor (localhost) y el dispositivo quedará seleccionado. Asimismo el estado será “Not programmed”.

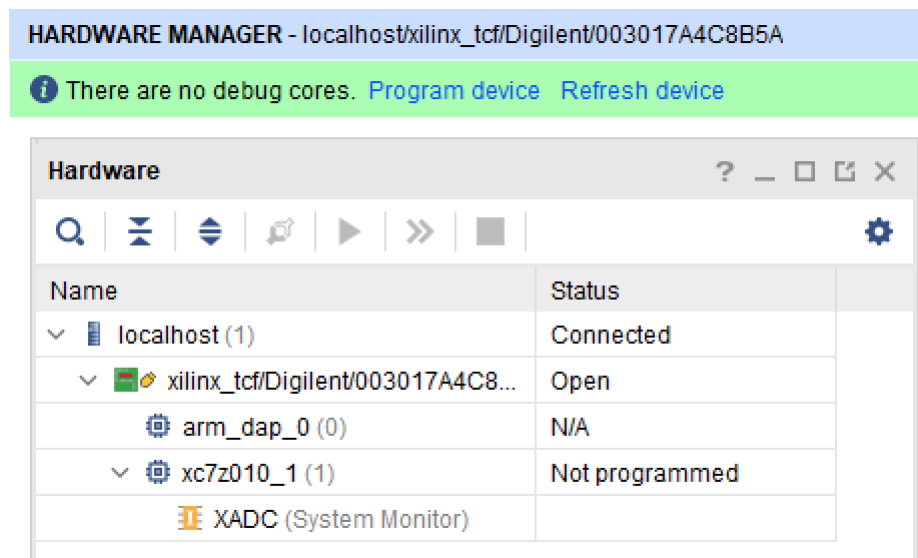


Figura 27. Sesión de hardware

Seleccionar el dispositivo y verificar que el archivo lab1.bit esté seleccionado como archivo de programación en la pestaña **General** del panel **Hardware Device Properties**.

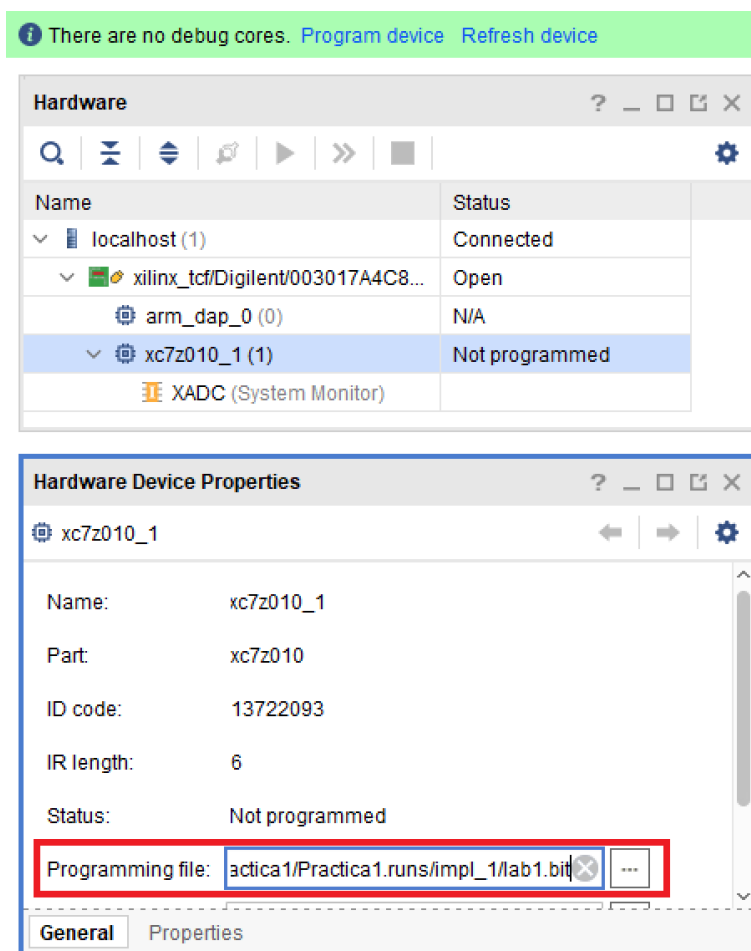
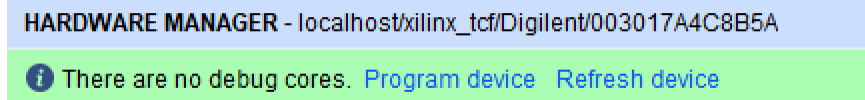


Figura 28. Verificación del archivo de configuración

**6-1-5.** Presionar en el link **Program device**.

Otra opción es presionar el botón derecho en el dispositivo y seleccionar **Program Device...**



*Figura 29. Link para programar la FPGA*

**6-1-6.** Presionar el botón **Program** para configurar la FPGA.

El led DONE se encenderá al finalizar la configuración; los otros leds cambiarán de acuerdo a la posición de los interruptores.

**6-1-7.** Verificar la funcionalidad del sistema cambiando los interruptores y observando cómo responden los leds de acuerdo al diagrama lógico.**6-1-8.** Para finalizar, desconectar la placa.**6-1-9.** Cerrar la sesión de hardware mediante **File ► Close Hardware Manager**.**6-1-10.** Presionar **OK** para finalizar la sesión.**6-1-11.** Cerrar el entorno **Vivado** mediante **File ► Exit** y presionar **OK**.