

# **Circuitos lógicos programables**

## **Práctica 5**

### **Depuración de hardware**

## Introducción

En esta práctica se utilizará el sistema basado en UART de prácticas anteriores. Se aplicará la característica Mark Debug y se utilizará el bloque Integrated Logic Analyzer (ILA) para depurar el hardware.

## Objetivos

- Utilizar el bloque Integrated Logic Analyzer (ILA) como herramienta de depuración
- Utilizar la característica Mark Debug para depurar señales
- Utilizar las herramientas de depuración para depurar el sistema

## Procedimiento

### Descripción del Sistema

El sistema consiste en un receptor serie asincrónico (UART) que recibe caracteres y muestra la representación binaria de la parte baja del carácter en 4 Leds. Cuando se presiona un botón, se representa la parte alta del caracter. En la Figura 1 se ve el diagrama en bloques del sistema.

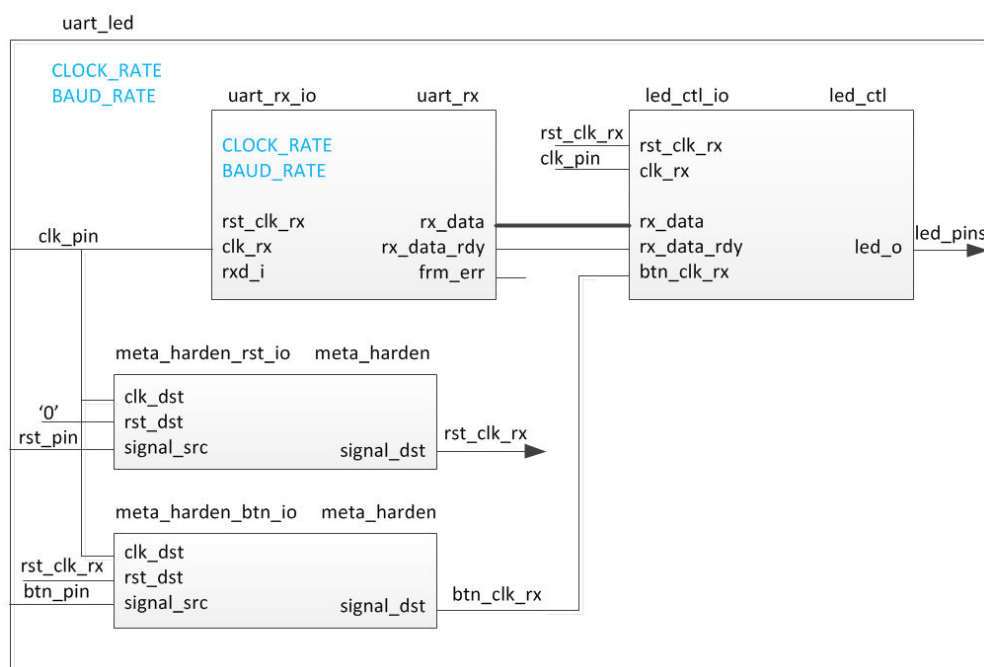
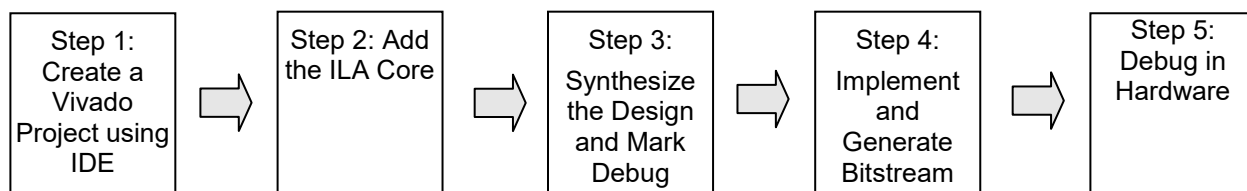


Figura 1. Sistema basado en UART

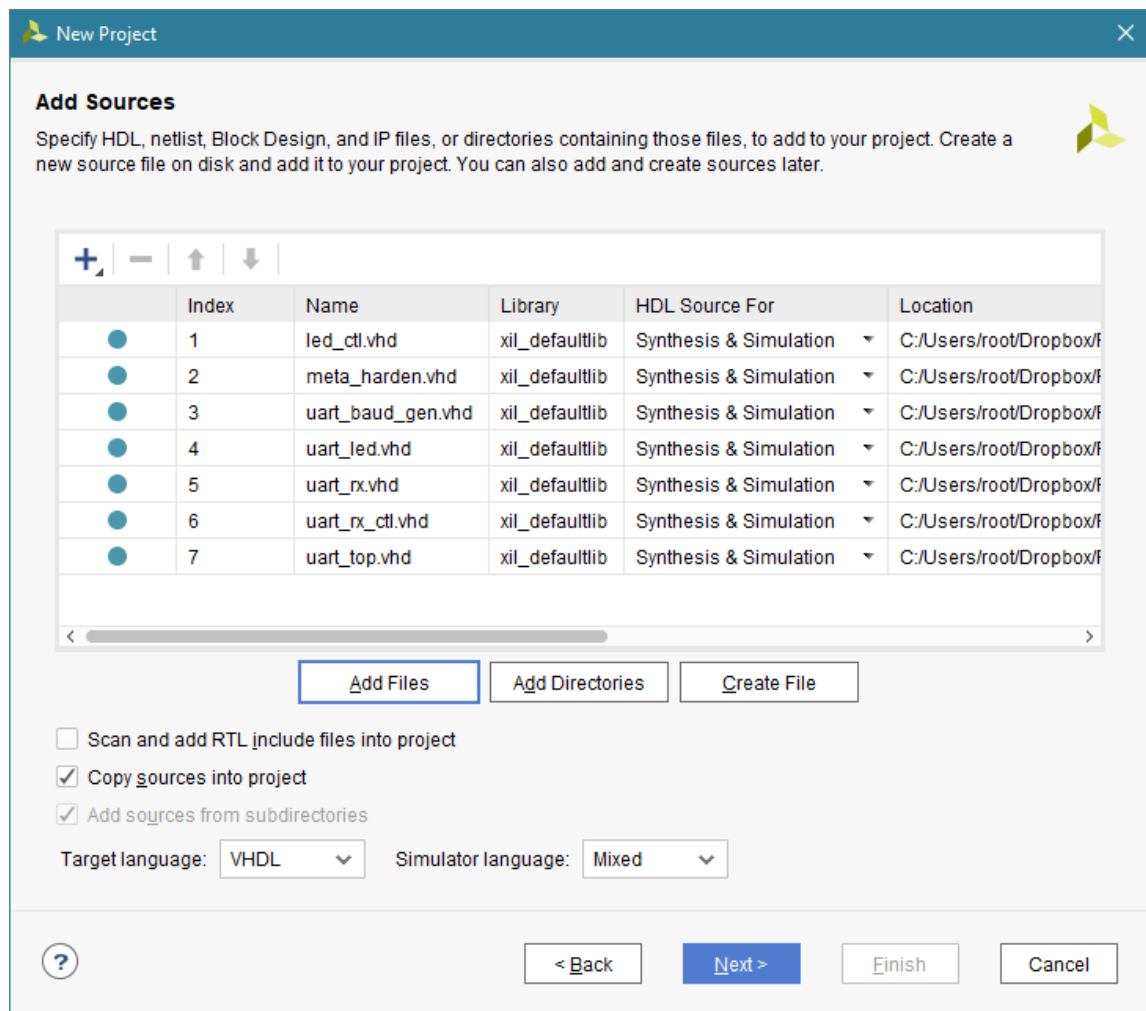
## Hoja de Ruta de la Práctica



## Crear un proyecto de Vivado

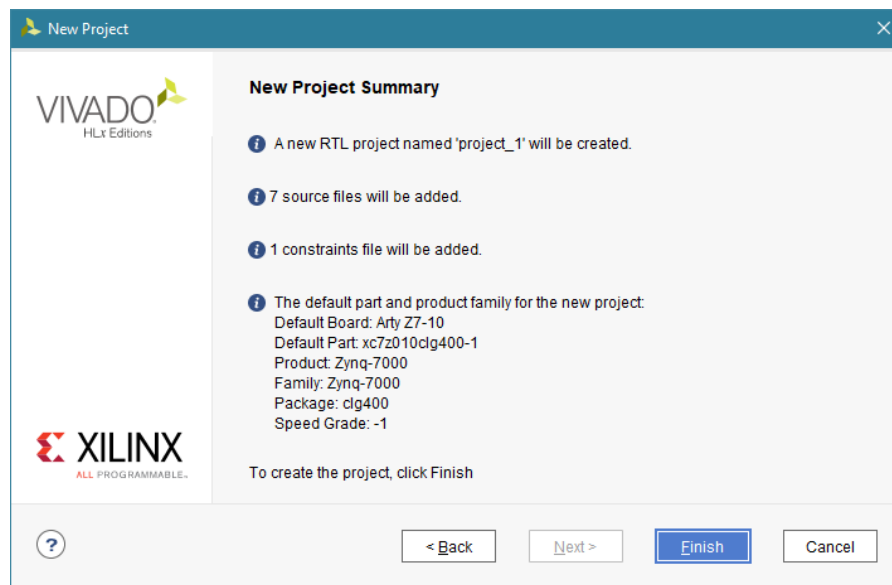
## Paso 1

- 1-1. Iniciar la herramienta Vivado y crear un proyecto con la placa Arty Z7. Agregar los archivos HDL y crear los archivos de restricciones `uart_led_pins_ArtyZ7.xdc` y `uart_led_timing_ArtyZ7.xdc`.
- 1-1-1. Iniciar la herramienta Vivado
- 1-1-2. Presionar **Create Project** para iniciar el asistente. Aparecerá el cuadro de diálogo **Create A New Vivado Project**. Presionar **Next**.
- 1-1-3. Presionar el botón de navegación del campo **Project location** del formulario **New Project**, elegir la ubicación del proyecto y presionar **Select**.
- 1-1-4. Ingresar el nombre del proyecto en el campo **Project name**. Verificar que la opción **Create Project Subdirectory** esté seleccionada. Presionar **Next**.
- 1-1-5. Seleccionar la opción **RTL Project** en el formulario **Project Type**, y presionar **Next**
- 1-1-6. Verificar que esté seleccionada la opción VHDL en el menú desplegable **Target Language** y Mixed en **Simulator Language**, en el formulario **Add Sources**
- 1-1-7. Presionar el botón **Add Files** y agregar los archivos VHDL (`led_ctl.vhd`, `meta_harden.vhd`, `uart_baud_gen.vhd`, `uart_led.vhd`, `uart_rx.vhd`, `uart_rx_ctl.vhd` and `uart_top.vhd`), presionar **OK**, y después presionar **Next** para pasar al formulario **Add Constraints**



*Figura 2. Agregado de los archivos fuente*

- 1-1-8. Presionar el botón **Add File** y agregar los archivos de restricciones `uart_led_timing_ArtyZ7.xdc` y `uart_led_pins_ArtyZ7.xdc`, y luego presionar **Ok**. Después presionar **Next**.
- 1-1-9. En el formulario **Default Part**, usar la opción **Boards**. Seleccionar la placa **Arty Z7-10**. Presionar **Next**. Debería verse el resumen de configuración del proyecto.



*Figura 3. Resumen del proyecto*

**1-1-10.** Presionar **Finish** para crear el proyecto.

## Agregar el core ILA

## Paso 2

- 2-1-1. Hacer click en **IP Catalog** debajo de **Project Manager** en el panel **Flow Navigator**.
- 2-1-2. El catálogo será mostrado en una pestaña.
- 2-1-3. Expandir la carpeta **Debug & Verification** ► **Debug** y hacer doble-click en **ILA**.

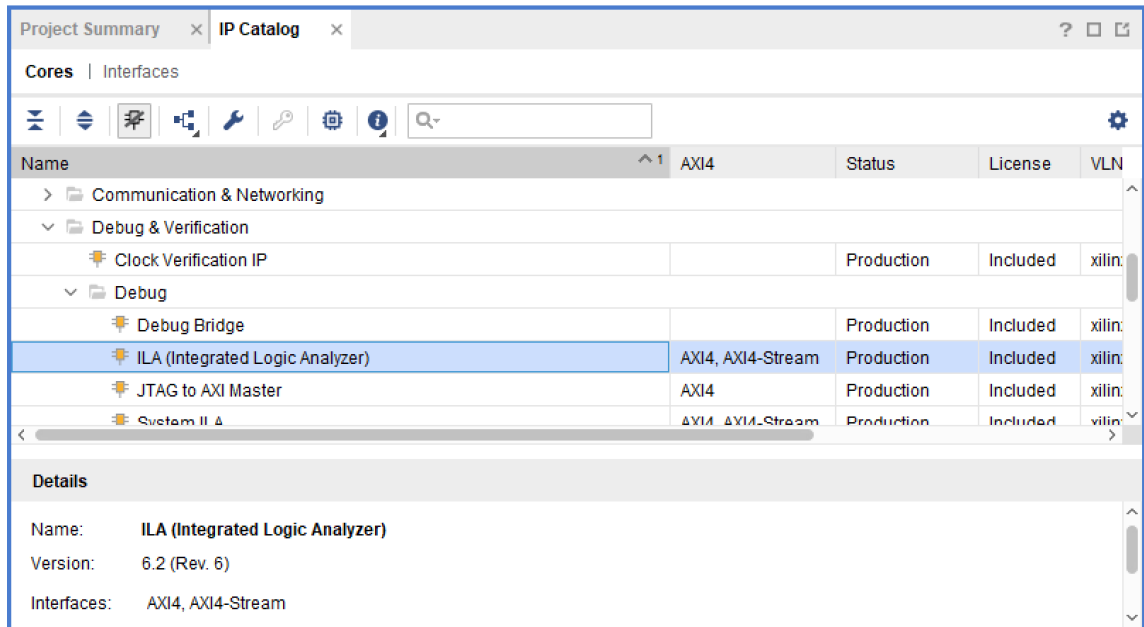


Figura 4. ILA en el catálogo IP

Esta práctica conectará el core ILA al puerto LED.

- 2-1-4. Cambiar el nombre del component a **ila\_led**.
- 2-1-5. Cambiar el número de puntas de prueba (**Number of Probes**) a **2**.

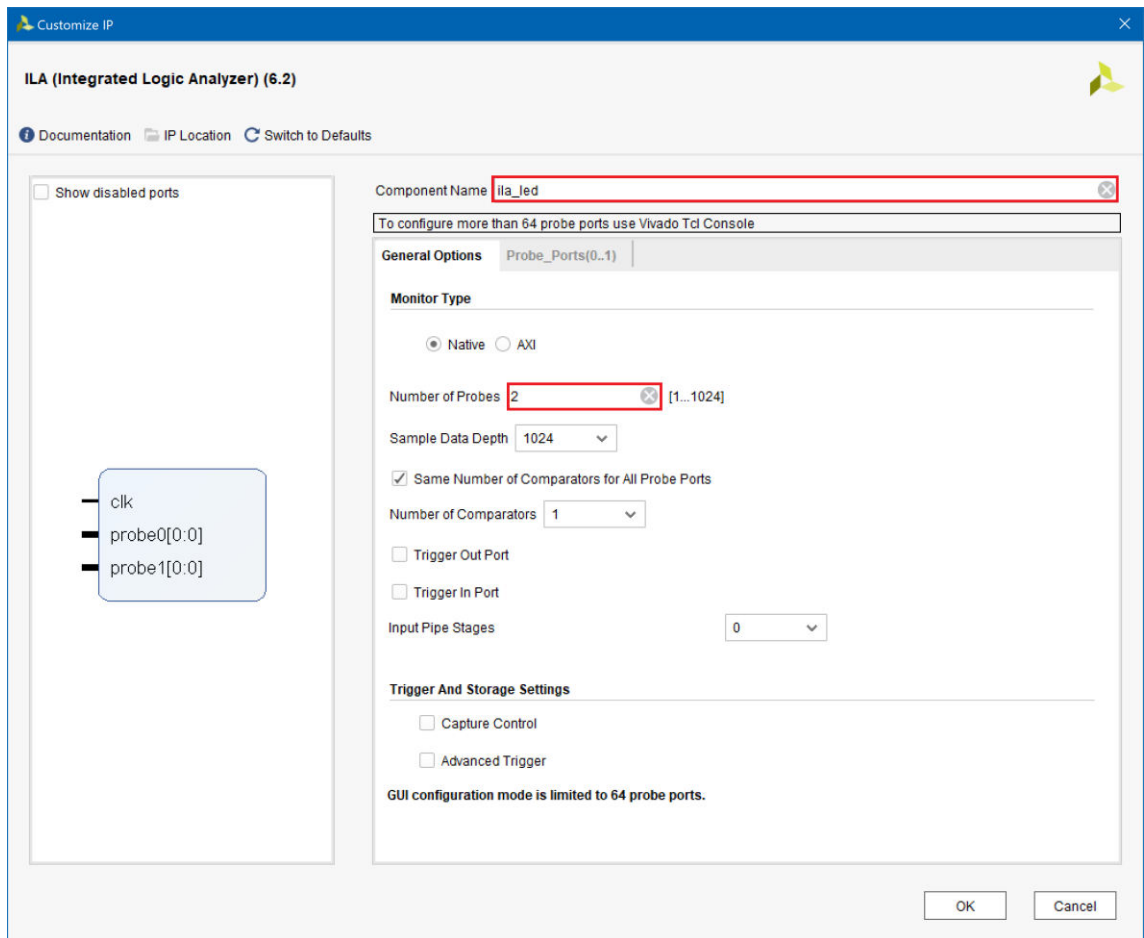


Figura 5. Estableciendo el nombre del componente y el número de puntas de prueba

- 2-1-6.** Seleccionar la pestaña **Probe Ports** y cambiar el ancho del puerto PROBE1 a **4**, dejando el ancho de PROBE0 en **1**.

General Options		Probe_Ports(0..1)		
Probe Port	Probe Width [1..4096]		Number of Comparators	Probe Trigger or Data
PROBE0	1	✕	1	DATA AND TRIGGER
PROBE1	4	✕	1	DATA AND TRIGGER

Figura 6. Estableciendo los anchos de las puntas de prueba

- 2-1-7.** Hacer click en **OK**. Nuevamente hacer click en OK para generar el directorio asociado a la IP.

Aparecerá el cuadro de diálogo **Generate Output Products**.

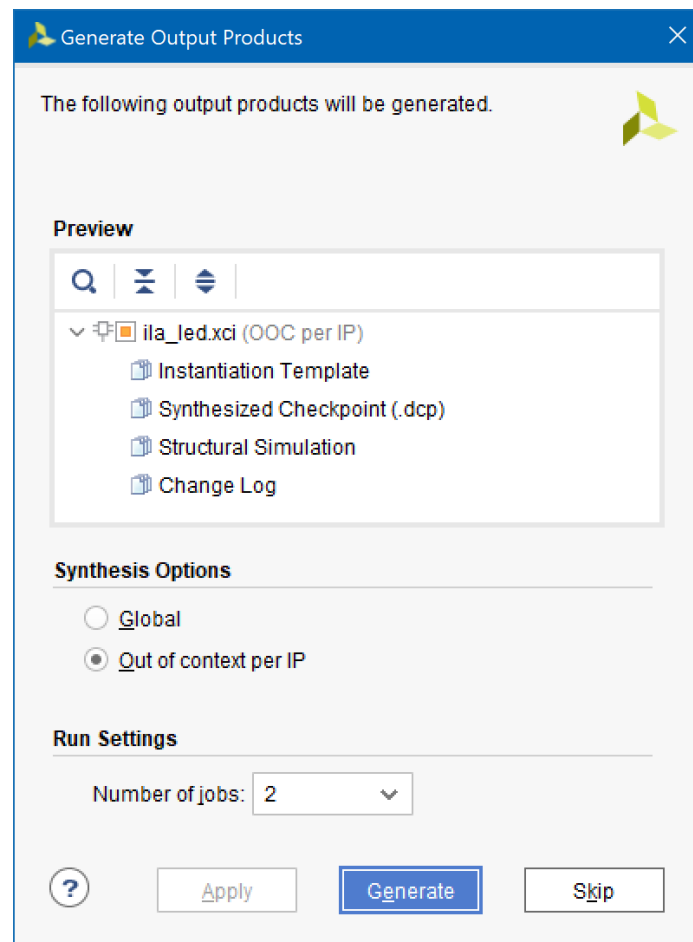


Figura 7. Ventana de generación de archivos

- 2-1-8.** Hacer click en el botón **Generate** para generar el core incluyendo el template de instanciación. Hacer click en **OK** en el cuadro de diálogo de advertencia. Notar que el core es agregado a la vista *Design Sources*.



Figura 8. Nuevo core ILA agregado al diseño

- 2-1-9.** Seleccionar la pestaña **IP Sources**, expandir **IP(1) ▶ ila\_led ▶ Instantiation Template**, y hacer doble-click en **ila\_led.vho** para ver la plantilla de instanciación.



**2-1-10.** Declarar el componente `ila_led` incluyendo en el archivo `uart_top.vhd` el texto correspondiente existente en la plantilla.

```
component ila_led
  port (
    clk : in std_logic;
    probe0 : in std_logic_vector(0 downto 0);
    probe1 : in std_logic_vector(3 downto 0)
  );
end component;
```

**2-1-11.** Instanciar `ila_led` en el diseño copiando las líneas correspondientes de la plantilla y pegándolas en el archivo `uart_top.vhd` en la parte descriptiva de la arquitectura (entre el `begin` y el `end`).

```
your_instance_name : ila_led
  port map (
    clk => clk,
    probe0 => probe0,
    probe1 => probe1
  );
```

**2-1-12.** Cambiar la etiqueta de instancia `your_instance_name` por `ila_led_i0`.

**2-1-13.** Crear una señal auxiliar `led_pins_aux`.

```
signal led_pins_aux: std_logic_vector(3 downto 0);
```

**2-1-14.** Conectarla a al puerto `led_pins` del componente `uart_led` (en reemplazo de `led_pins`)

**2-1-15.** Asignar a `led_pins` la señal recién creada.

```
led_pins <= led_pins_aux;
```

**2-1-16.** Cambiar los siguientes nombres de puertos en el código VHDL para conectar el `ila` a las señales existentes en el diseño:

<code>Clk =&gt; CLK</code>	➔	<code>clk =&gt; clk_pin</code>
<code>probe0 =&gt; PROBE0</code>	➔	<code>probe0 =&gt; rx_data_rdy_out</code>
<code>probe1 =&gt; PROBE1</code>	➔	<code>probe1 =&gt; led_pins_aux</code>

```
86  ila_led_i0 : ila_led
87      PORT MAP (
88          clk => clk_pin,
89          probe0 => rx_data_rdy_out,
90          probe1 => led_pins_aux
91      );
```

Figura 9. Instanciado del core ILA en `uart_top.vhd`

**2-1-17.** Seleccionar **File ► Save File**.

Notar que la instancia del core ILA está en la jerarquía del diseño.

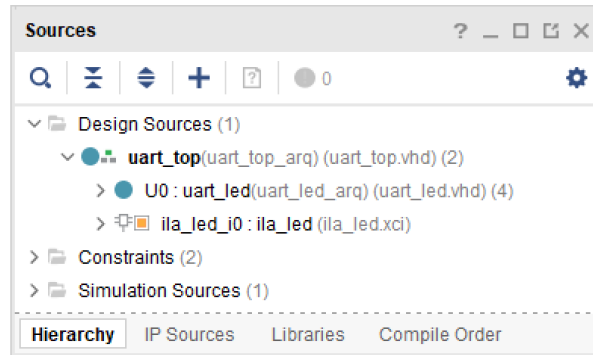


Figura 10. Core ILA agregado al diseño

## Sintetizar el Diseño y Mark Debug

## Paso 3

**3-1. Sintetizar el diseño. Abrir el diseño sintetizado. Observar el esquemático. Agregar Mark Debug en el bus rx\_data entre las instancias uart\_rx\_i0 y led\_ctl\_i0.**

**3-1-1.** Hacer click en **Run Synthesis** debajo de **Synthesis** en el panel **Flow Navigator**. Presionar **Save** para guardar el proyecto si es requerido.

El proceso de síntesis correrá sobre uart\_top.vhd y todos los archivos jerárquicos debajo de él. Cuando el proceso se complete se mostrará un cuadro de diálogo indicando **Synthesis Completed**.

**3-1-2.** Seleccionar la opción **Open Synthesized Design** y presionar **OK**.

**3-1-3.** Hacer click sobre **Schematic** debajo de **Synthesized Design** de la sección **Synthesis** del panel **Flow Navigator** para ver el diseño sintetizado en una vista de esquemático (si es que no se abrió al finalizar el proceso de síntesis).

**3-1-4.** Expandir el componente **U0** y seleccionar el bus **rx\_data** ente las instancias **uart\_rx\_i0** y **led\_ctl\_i0**, hacer botón-derecho, y seleccionar **Mark Debug**.

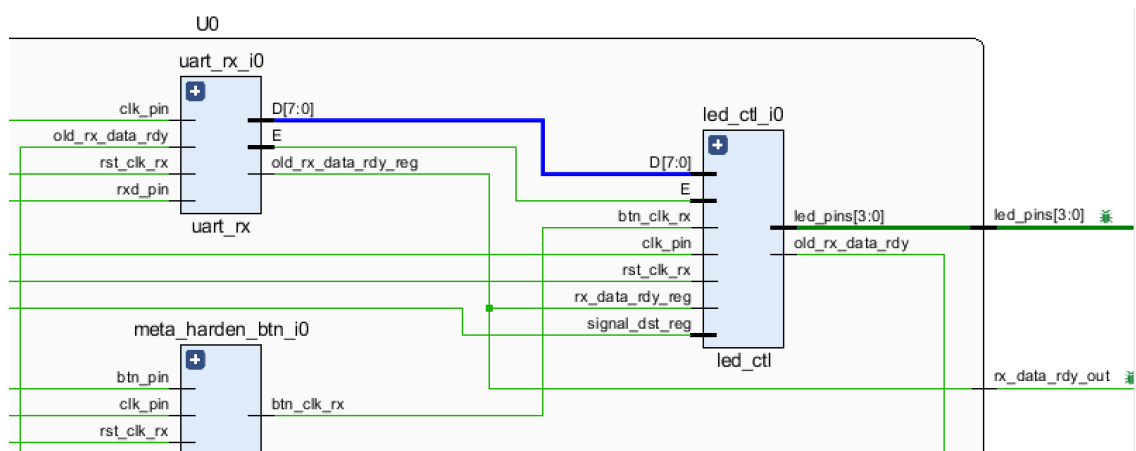
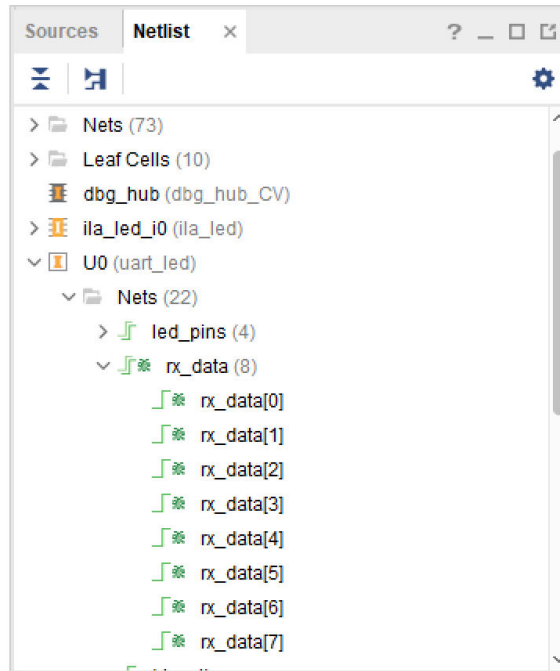


Figura 11. Marcando un bus para depuración

**3-1-5.** Seleccionar **File ► Constraints ► Save**.

**3-1-6.** Hacer click en **OK**, y nuevamente **OK** para usar **uart\_led\_timing\_ArtyZ7.xdc** como target.

**3-1-7.** Seleccionar el panel **Netlist** y notar que las nets que están marcadas/asignadas para depuración tienen un icono de debug al lado de ellas.



*Figura 12. Redes con iconos de depuración*

**3-1-8.** Seleccionar el layout **Debug** o **Layout ► Debug (desde el menu)** .

Notar que la pestaña **Debug** es visible en el panel consola mostrando grupos de redes asignadas y no asignadas.

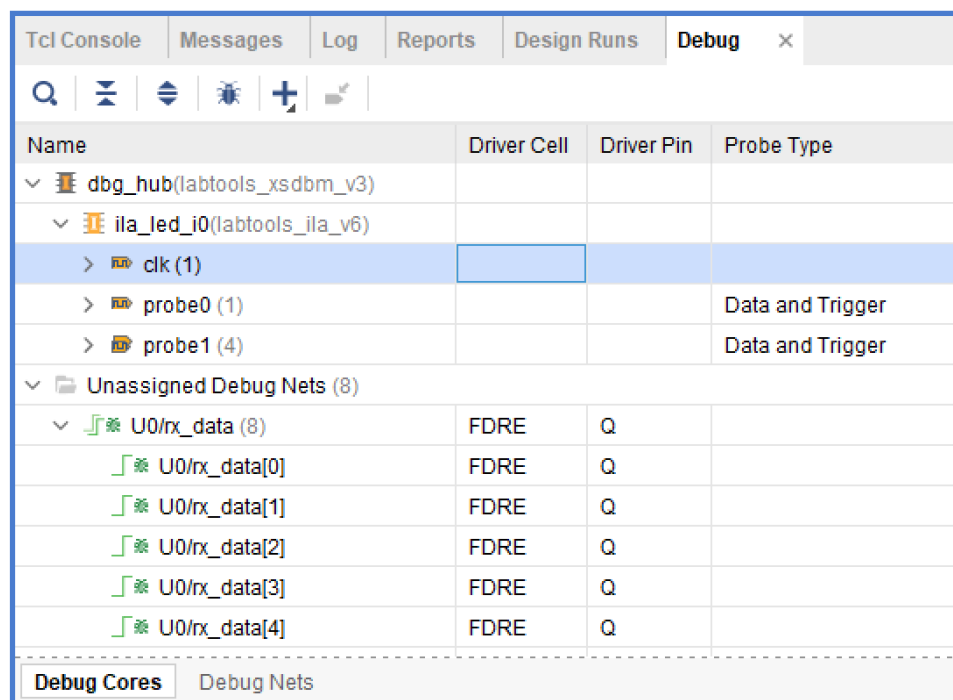


Figura 13. Pestaña Debug mostrando redes asignadas y no asignadas

**3-1-9.** Hacer botón-derecho sobre **Unassigned Debug Nets** y seleccionar la opción **Set up Debug....**

**3-1-10.** En el wizard **Set Up Debug** presionar **Next**.

Notar que se lista *rx\_data*, con el Clock Domain como *clk\_pin\_IBUF\_BUFG*.

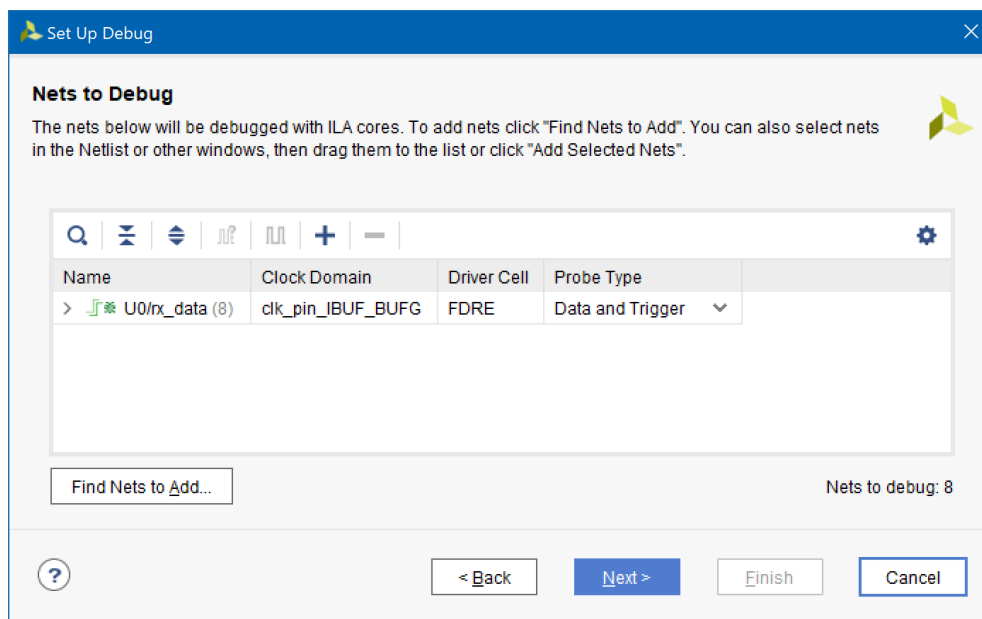


Figura 14. Redes restantes después de remover las redes asignadas en el wizard Set Up Debug

**3-1-11.** Hacer click en **Next** y nuevamente en **Next** (dejando toda la configuración por defecto), luego en **Finish**.

**3-1-12.** En el **Synthesized Design Schematic**, hacer click sobre la red a la salida del BUFG para el pin de entrada clk\_pin. Ubicarse encima con el puntero del mouse para que aparezca la info asociada y notar que el nombre es clk\_pin\_IBUF\_BUFG. Esta es la red de reloj seleccionada antes para red de debug.

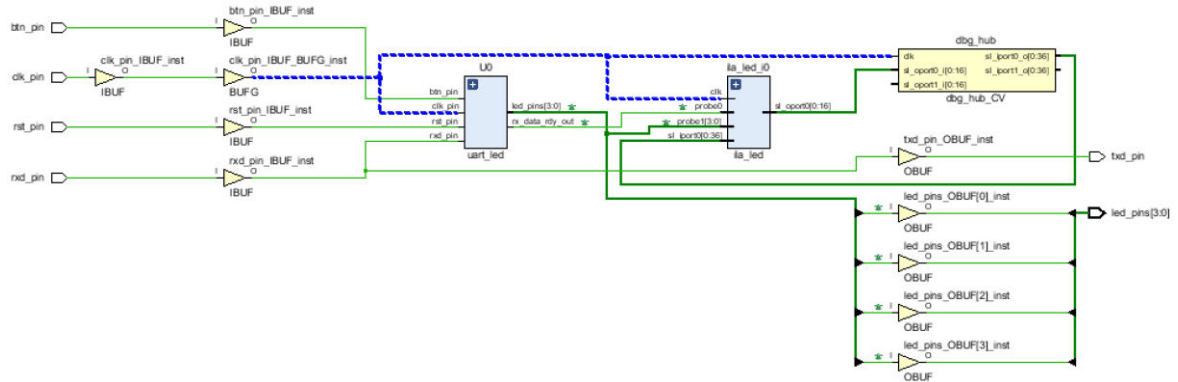


Figura 15. Ubicando clk\_pin\_IBUF\_BUFG en el diseño

**3-1-13.** Hacer click en el botón derecho del mouse sobre **uart\_led\_pins\_ArtyZ7.xdc** en el panel **Sources** y seleccionar **Set as Target Constraint File**. Esto guardará los cambios sobre el archivo.

**3-1-14.** Seleccionar **File ► Constraints ► Save** y presionar **OK** y luego **OK** nuevamente.

**3-1-15.** Abrir el archivo **uart\_led\_pins\_ArtyZ7.xdc** y notar que las redes de debug han sido agregadas al final. Si el archivo estaba abierto se lo deberá actualizar presionando el link **reload**.

**3-1-16.** Realizar este paso si la síntesis figura como Out-of-date en la pestaña **Design Runs**. Hacer botón derecho sobre **synth\_1** y seleccionar **Force Up-to-Date** para asegurarse de que el proceso de síntesis no sea corrido nuevamente, ya que el diseño no ha cambiado.

---

## Implementar y Generar el Bitstream

## Step 4

---

### 4-1. Generar el bitstream.

- 4-1-1. Hacer click en **Generate Bitstream** para ejecutar la implementación y el proceso de generación del bit.
- 4-1-2. Hacer click en **Yes** para ejecutar el proceso de implementación y luego **OK**.
- 4-1-3. Cuando el proceso de generación del bitstream haya terminado correctamente, un cuadro con tres opciones aparecerá. Seleccionar la opción **Open Hardware Manager** y presionar **OK**.

## Depurar en Hardware

## Paso 5

### 5-1. Conectar la placa. Conectar el adaptador usb-serie. Abrir una sesión de hardware, y programar la FPGA.

- 5-1-1. Asegurarse que el cable Micro-USB esté conectado al conector JTAG PROG de la placa.
- 5-1-2. Conectar el adaptador usb-serie a la computadora. Verificar el puerto COM al que se mapea.
- 5-1-3. Conectar los 3 cables de Tx, Rx y GND entre el conector USB-serie y el puerto JA de la placa ArtyZ7: TxD en el pin 1; RxD en el pin 2 y GND en el pin 5

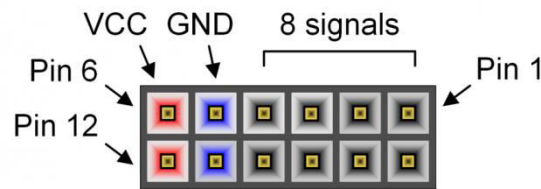


Figura 16. Vista de un PMOD

Nota: Primero conectar la placa ArtyZ7 a la computadora, luego conectar el adaptador usb-serie.

- 5-1-4. Hacer click en el link **Open target**, luego en **Auto Connect**.
- 5-1-5. El estado de la sesión de hardware cambia de Unconnected al nombre del servidor y el dispositivo es resaltado. Notar que el estado indica que no está programado.
- 5-1-6. Seleccionar el dispositivo y verificar que el **uart\_top.bit** es seleccionado como el archivo de programación en la pestaña **General**. Notar también que hay una entrada en el campo **Probes file**.

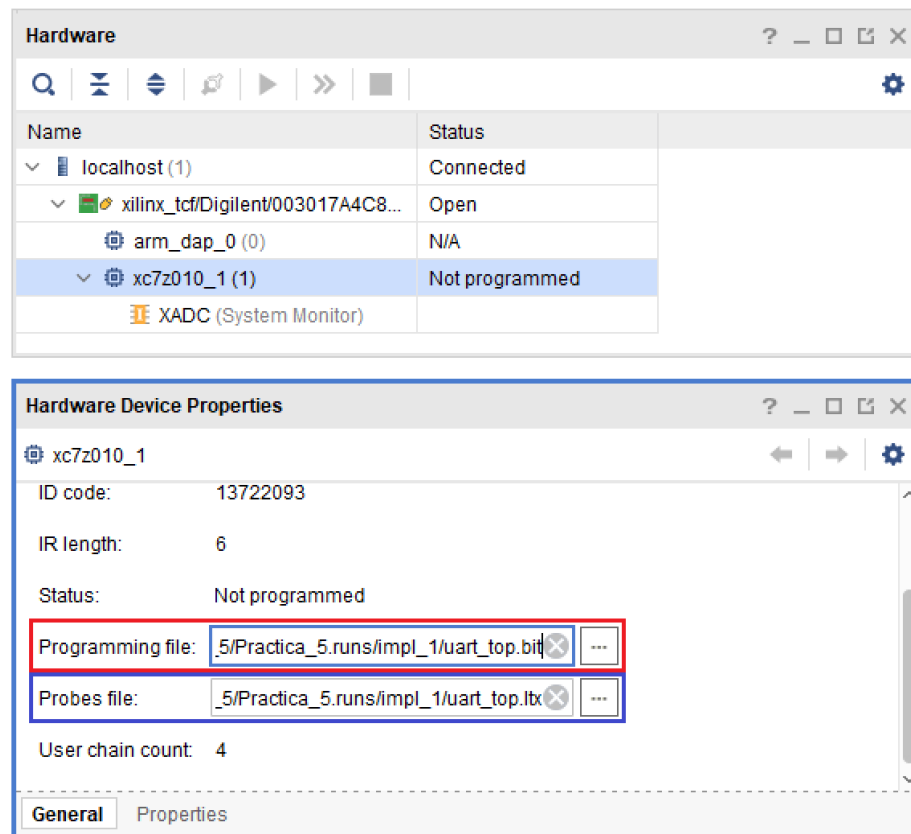


Figura 17. Verificación del archivo de configuración

## 5-2. Iniciar una terminal tal como TeraTerm o HyperTerminal. Seleccionar el Puerto COM apropiado. Configurar el puerto COM en 115200. Programar la FPGA y verificar la funcionalidad.

5-2-1. Iniciar una terminal tal como TeraTerm o HyperTerminal.

5-2-2. Seleccionar un Puerto COM apropiado.

5-2-3. Configurar el baud rate en 115200.

5-2-4. Hacer click-derecho en la FPGA, y seleccionar **Program Device...** y presionar **Program**.

El archivo de configuración será descargado y el led DONE se encenderá indicando que la FPGA ha sido configurada. La ventana **Debug Probes** también se abrirá, si no lo hace seleccionar **Window ► Debug Probes**.

5-2-5. En la ventana **Debug Probes** notar que hay dos cores de depuración, hw\_ila\_1 and hw\_ila\_2.



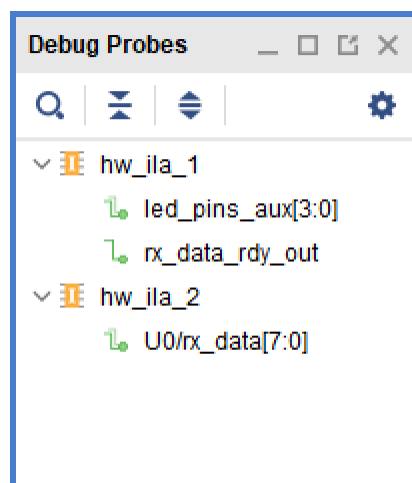


Figura 18. Debug probes

La ventana de estado de la sesión de hardware también se abre mostrando que la FPGA fue programada conteniendo dos cores ila en estado idle.

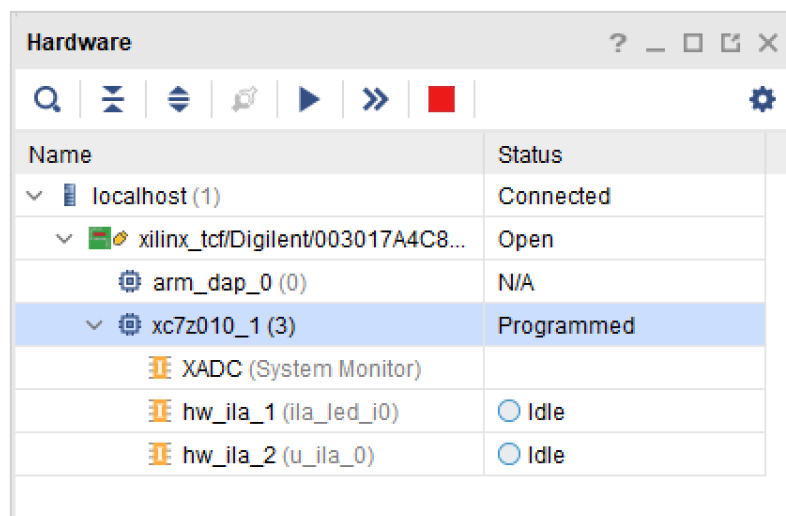


Figura 19. Estado de la sesión de Hardware

**5-2-6.** Presionar sobre el botón **Run Trigger Immediate** para ver las señales en la ventana de formas de onda.

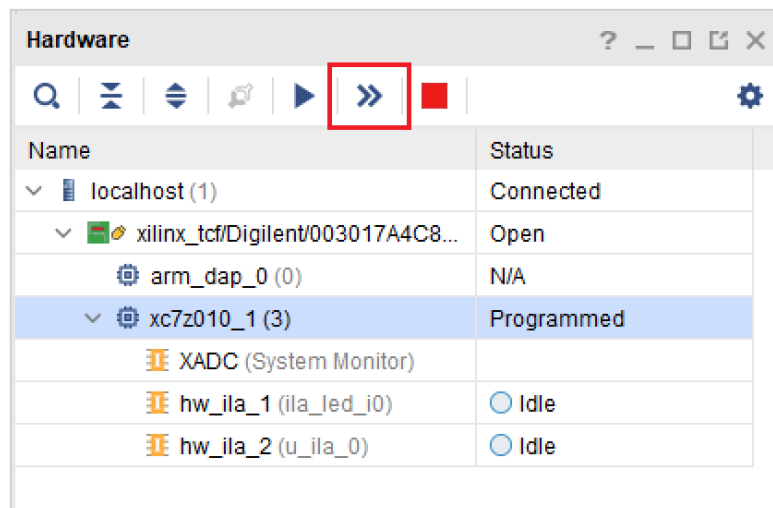


Figura 20. Abriendo la ventana de formas de onda

Existen dos pestañas, una para cada ila; una para el core ila instanciado y otra para el creado por el método MARK\_DEBUG.

### 5-3. Establecer las condiciones para el trigger para disparar ante una escritura sobre el puerto led (`rx_data_rdy=1`) y para posicionarlo en 512. Armar el trigger.

5-3-1. Agregar un trigger para `rx_data_rdy_out` presionando el botón + en la ventana **Trigger Setup**.

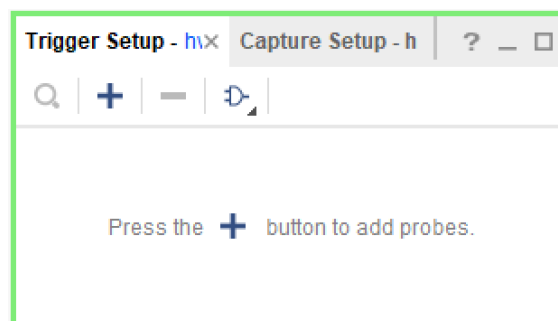


Figura 21. Agregando un trigger

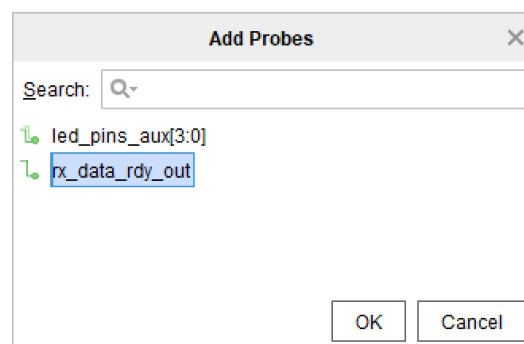


Figura 22. Seleccionando una señal a sensor

- 5-3-2.** En la ventana **Trigger Setup**, configurar el campo **Operator** == (equal) y cambiarel valor de X a 1. Hacer click en **OK**.

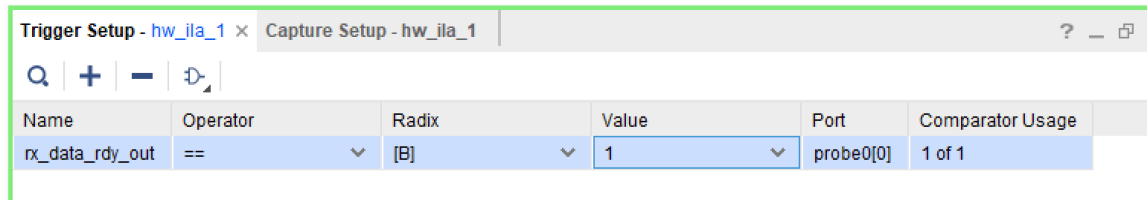


Figura 23. Estableciendo la condición de disparo

- 5-3-3.** Establecer la posición del trigger del *hw\_ila\_1* en **512**.

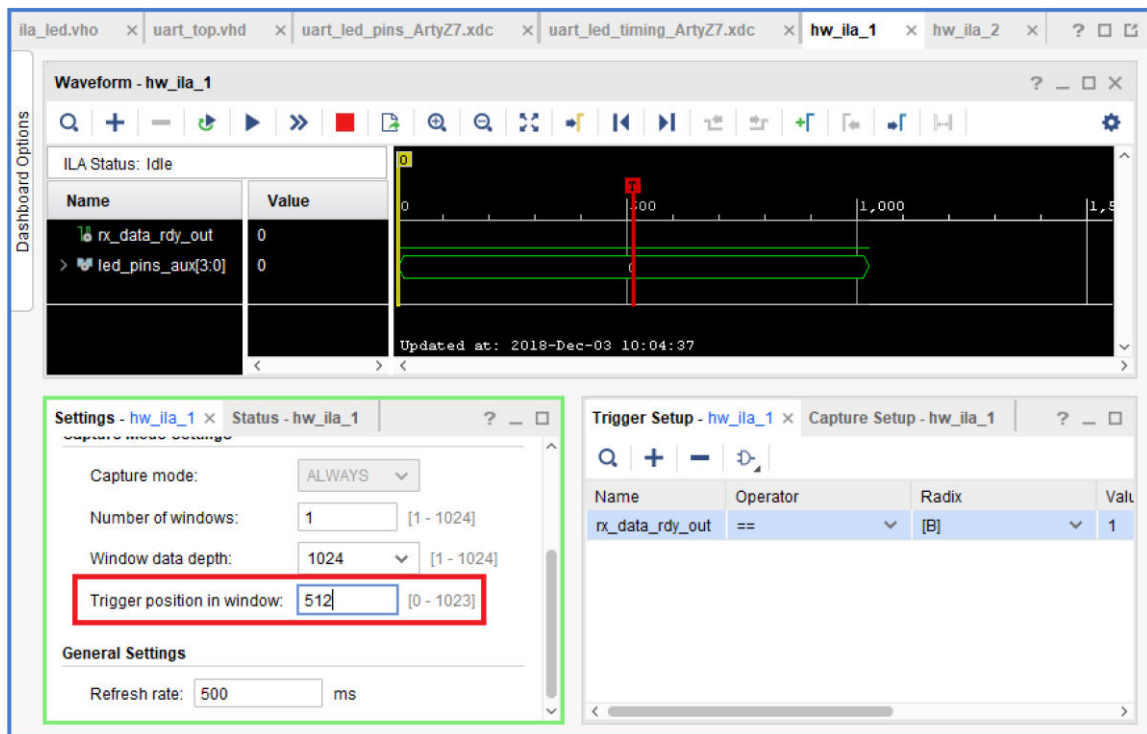


Figura 24. Estableciendo la posición del trigger

- 5-3-4.** De manera similar, establecer la posición del trigger del *hw\_ila\_2* en **512**.
- 5-3-5.** Seleccionar el *hw\_ila\_1* en la ventana **Hardware** y hacer click en el botón **Run Trigger**.

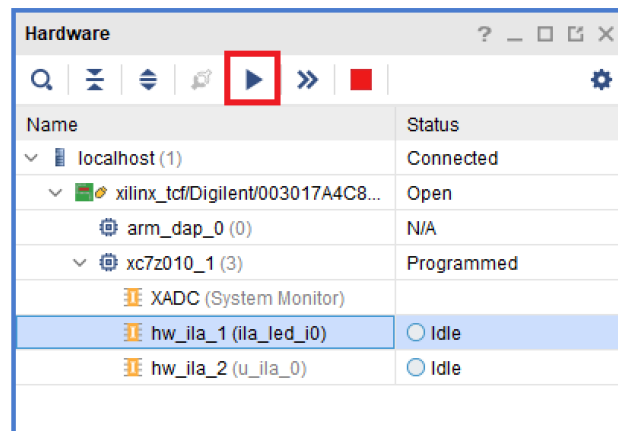


Figura 25. Armar el trigger

Observar que el core hw\_ila\_1 está armado y muestra el estado de **Waiting for Trigger**.

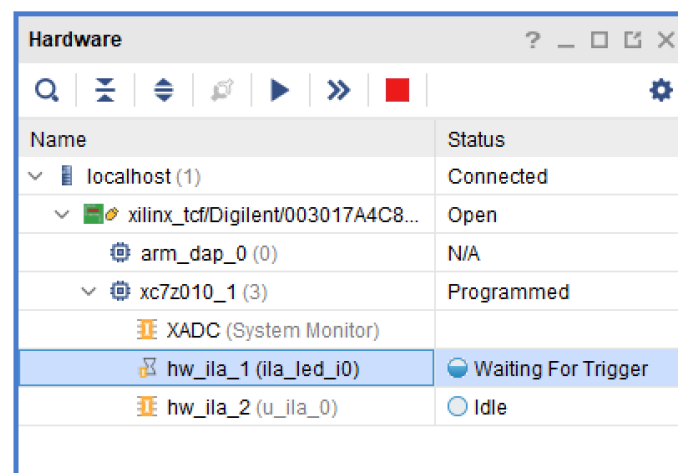


Figura 26. Analizador de Hardware corriendo en modo captura

- 5-3-6.** En la terminal serie tipear un caracter, y observar que el estado del the hw\_ila\_1 cambia a Full y luego a idle al momento en que the rx\_data\_rdy se pone en 1.
- 5-3-7.** Seleccionar la ventana **Waveform - hw\_ila\_data\_1** y ver la forma de onda. Notar que rx\_data\_rdy\_out cambia de 0 a 1 en la muestra 512 y que los bits led\_pins\_aux[3:0] también cambian desde 0 al patrón correspondiente al caracter tipeado (correspondiente a los 4 bits más bajos).

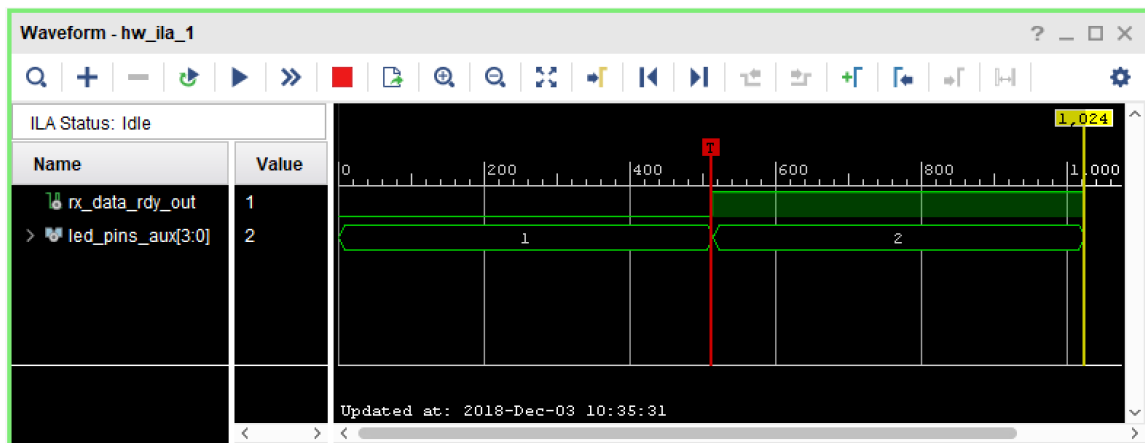


Figura 27. Ampliación de la vista de forma de onda

- 5-3-8.** Agregar a la ventana trigger la punta de prueba hw\_ila\_2 y cambiar la condición de disparo para la base de rx\_data[7:0]'s de hexadecimal a binario. Cambiar XXXX\_XXXX a **0100\_0001** (para el ASCII equivalente de A).

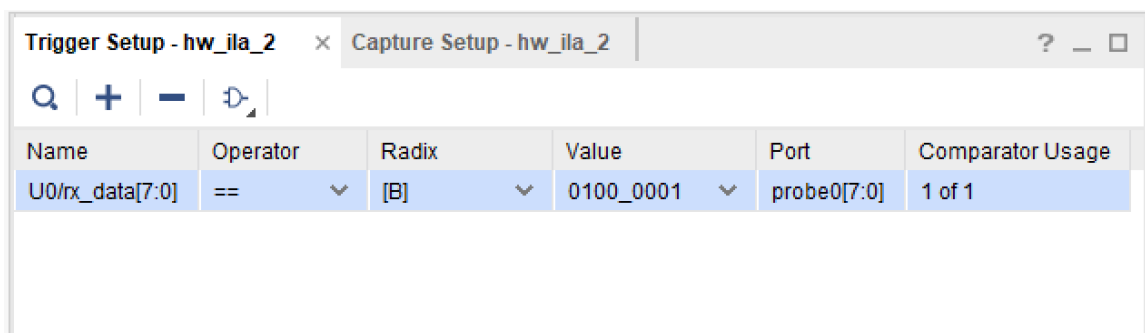


Figura 28. Estableciendo una condición de disparo para un patrón de entrada particular

- 5-3-9.** En la ventana Hardware, hacer click-derecho sobre **hw\_ila\_2** y seleccionar **Run Trigger**, y notar que el estado de hw\_ila\_2 cambia de *idle* a *Waiting for Trigger*. También notar que el estado hw\_ila\_1 no cambia de su estado idle ya que no está armado.
- 5-3-10.** Cambiar a la terminal serie y tipear U para disparar el core.
- 5-3-11.** Seleccionar la ventana de formas de onda correspondiente y verificar que muestra 41 después del trigger.



Figura 29. Segundo core ila disparado

**5-3-12.** Cerrar la terminal serie.

**5-3-13.** Desconectar el adaptador USB-serie

**5-3-14.** Seleccionar **File ► Close Hardware Manager**. Hacer click en **OK** para cerrarlo.

**5-3-15.** Cerrar el **Vivado** seleccionando **File ► Exit** y haciendo click en **OK**.