
Guía básica de Xilinx Vivado: Software: Xilinx Vivado 2024.2

Contenido

1.	Objetivos	2
2.	Placa de desarrollo Basys3	2
3.	Pasos del proceso	3
3.1.	Creación de un proyecto en el entorno de diseño Xilinx Vivado 2024.2.	4
3.2.	Creación de diseños HDL dentro del proyecto	8
3.3.	Simulación	10
3.4.	Restricciones	13
3.5.	Generación del fichero de programación	15
3.6.	Programación de la FPGA	15

1. Objetivos

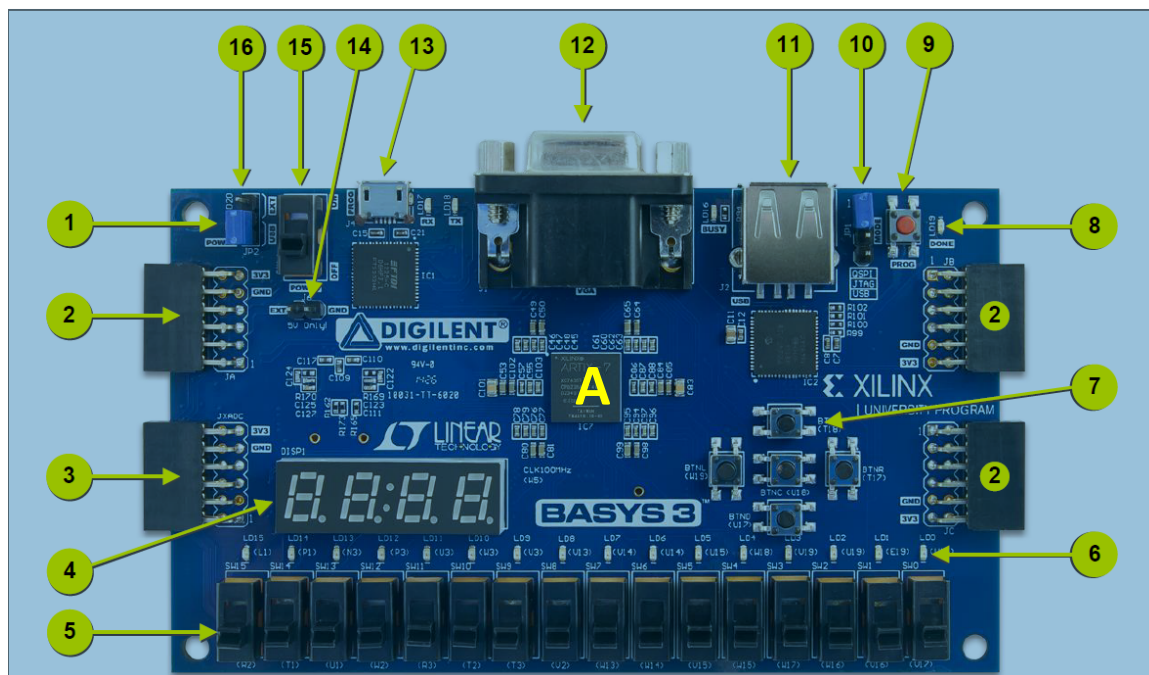
Adquirir una visión general de la herramienta de diseño Vivado Design Suite de Xilinx y recorrer el flujo de diseño.

Familiarizar al alumno con el entorno de diseño de Xilinx Vivado 2024.2, las herramientas de síntesis, verificación e implementación.

Este documento servirá de guía en los laboratorios para la realización de las prácticas.

2. Placa de desarrollo Basys3

Para la realización física de los sistemas digitales diseñados se utiliza la placa de desarrollo de sistemas digitales Basys3 de Digilent que contiene una FPGA de Xilinx.



Además del circuito integrado correspondiente a la FPGA (A), dispone de un oscilador de 100 MHz (conectado a un determinado pin de la FPGA), un chip de memoria (IC6 en el reverso de la placa, 32Mb flash no volátil, conectada a la FPGA mediante un bus SPI dedicado), otros CIs (reguladores, USB_UART bridge, ...) y diferentes puertos I/O y periféricos.

Los puertos y periféricos disponibles incluyen:

15	Encendido/apagado de la placa
16	Jumper de selección de fuente de alimentación. Las opciones son: USB y externa.

14	Alimentación externa. Debe de ser de 5V
13	Puerto USB de alimentación, programación (JTAG) y comunicación serie FPGA-PC
1	Led que indica que la placa está alimentada
10	Jumper de selección de modo de programación. Las opciones son: JTAG, memoria flash y USB
4	4 displays siete segmentos que permiten visualizar salidas del diseño
5	16 conmutadores (switches). Se pueden usar para controlar los valores de las entradas de los diseños
6	16 leds. Se pueden usar para ver los valores de las salidas del diseño
7	5 pulsadores. Entradas del diseño. Pulsados generan un pulso de 1 lógico
9	Botón para resetear la programación de la FPGA
11	Conector USB. Puede usarse para conectar un pendrive con un archivo de configuración para la FPGA, un ratón o un teclado
12	Conector VGA
2, 3	Conectores Pmod. Para expandir la Basys3 (conectar otros componentes)

Puede encontrarse una descripción más detallada en la página web de Xilinx.

Esta FPGA es una Artix-7. La familia Artix-7 usa una tecnología de 28nm. La siguiente tabla muestra los recursos disponibles en distintos dispositivos de esta familia. En particular, se ha resaltado la que usaremos.

Table 1: XA Artix-7 FPGA Device-Feature Table

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices ⁽²⁾	Block RAM Blocks ⁽³⁾			CMTs ⁽⁴⁾	PCIe ⁽⁵⁾	GTPs	XADC Blocks	Total I/O Banks ⁽⁶⁾	Max User I/O ⁽⁷⁾
		Slices ⁽¹⁾	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)						
XA7A12T	12,800	2,000	171	40	40	20	720	3	1	2	1	3	150
XA7A15T	16,640	2,600	200	45	50	25	900	5	1	4	1	5	210
XA7A25T	23,360	3,650	313	80	90	45	1,620	3	1	4	1	3	150
XA7A35T	33,280	5,200	400	90	100	50	1,800	5	1	4	1	5	210
XA7A50T	52,160	8,150	600	120	150	75	2,700	5	1	4	1	5	210
XA7A75T	75,520	11,800	892	180	210	105	3,780	6	1	4	1	6	285
XA7A100T	101,440	15,850	1,188	240	270	135	4,860	6	1	4	1	6	285

3. Pasos del proceso

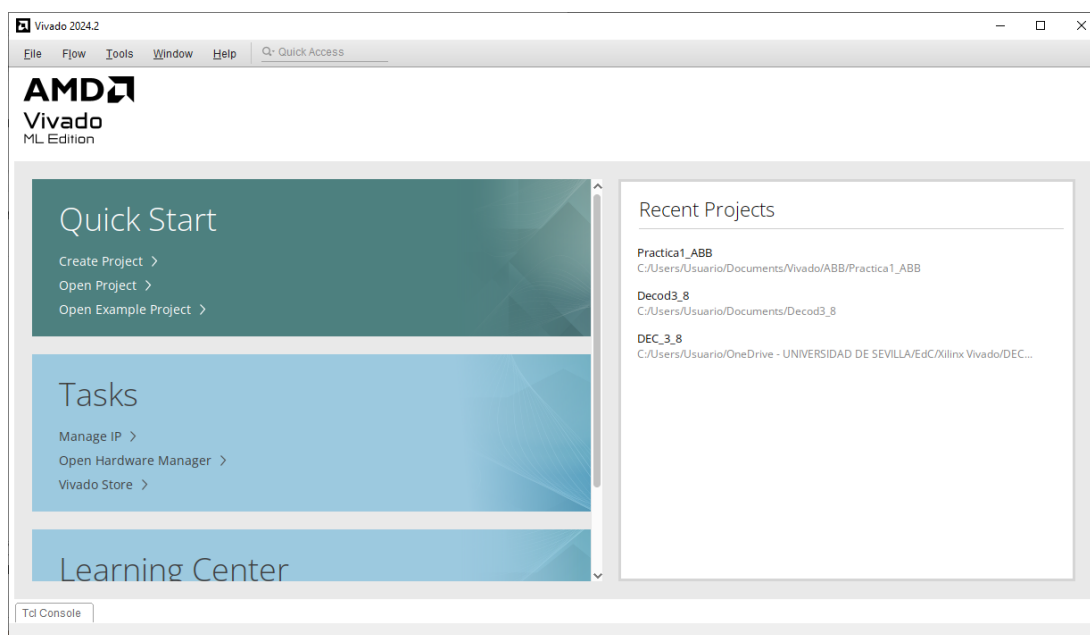
Son varios los pasos a seguir para crear y convertir un código HDL a un diseño que funciona en una FPGA.

Estos pasos son:

1. Abrir el entorno de diseño Xilinx Vivado 2024.2
2. Crear un nuevo proyecto.
3. Añadir/Crear un diseño HDL al proyecto.
4. Simular el diseño (funcional).
5. Generar el fichero de programación bitstream.
6. Programar el dispositivo en una placa Basys3 (Digilent).

3.1. Creación de un proyecto en el entorno de diseño Xilinx Vivado 2024.2.

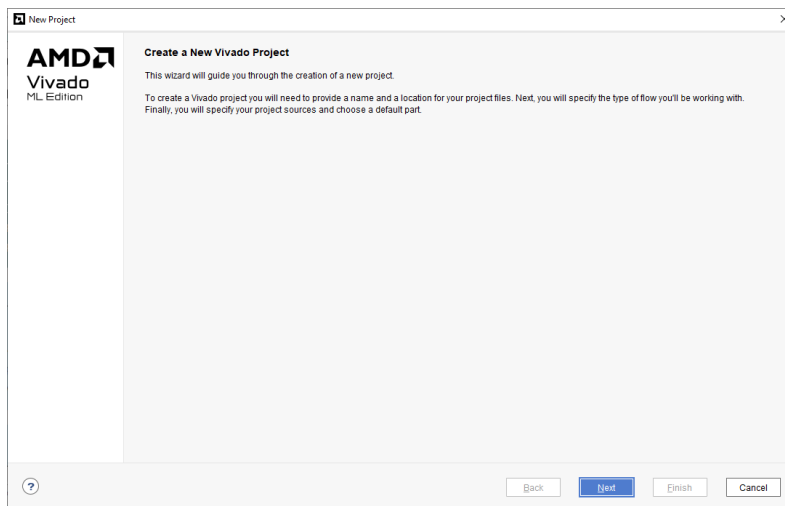
Se abre el entorno de diseño haciendo doble click en el icono “Vivado2024.2” que se encuentra en el escritorio. Esto abre la ventana principal del entorno, desde la que arranca todo el proceso de diseño:



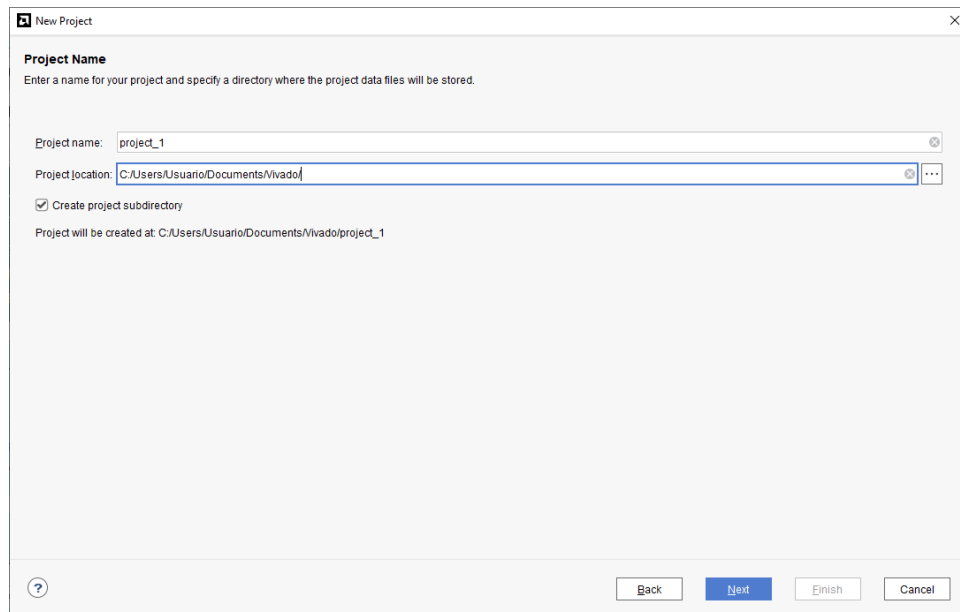
Desde el área **Quick Start** se puede crear un proyecto nuevo, abrir un proyecto existente o crear un proyecto utilizando alguna de las plantillas disponibles. En esta guía se va a crear un proyecto nuevo.

Se selecciona **Create New Project**.

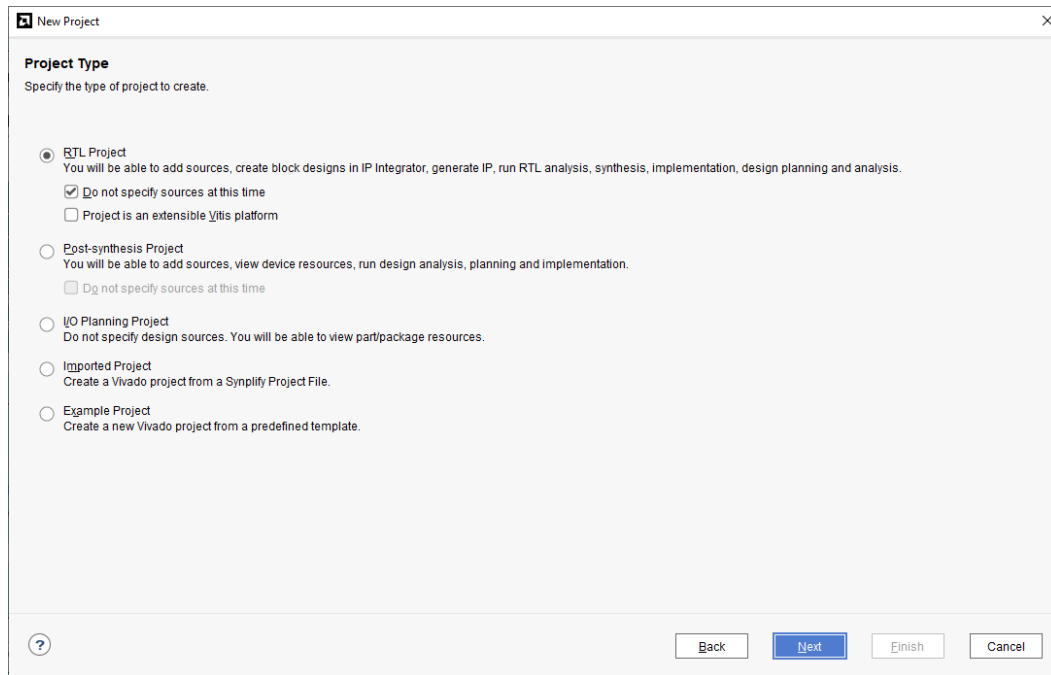
Se pulsa **Next** en el asistente para la creación de proyectos que se abre:



A continuación, se solicita un nombre y una localización para el nuevo proyecto y después se pulsa **Next**:



A continuación solicita el tipo de proyecto. Se selecciona RTL Project y se marca **“Do not specify sources at this time”**.



A continuación, hay que pulsar **Next**.

El siguiente paso es especificar la FPGA (Part) o placa (Board) con la que se trabaja y sobre la que se implementará el diseño.

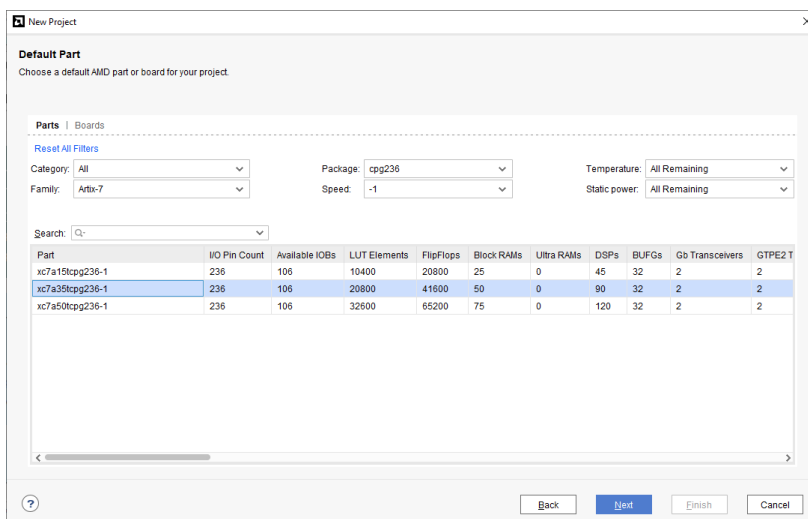
Se selecciona **parts** y el dispositivo que tiene la placa Basys3 es (**XC7A35T-1CPG236**). Éste se identifica utilizando los desplegables indicando los siguientes valores:

Family: Artix-7

Device: xc7a35t

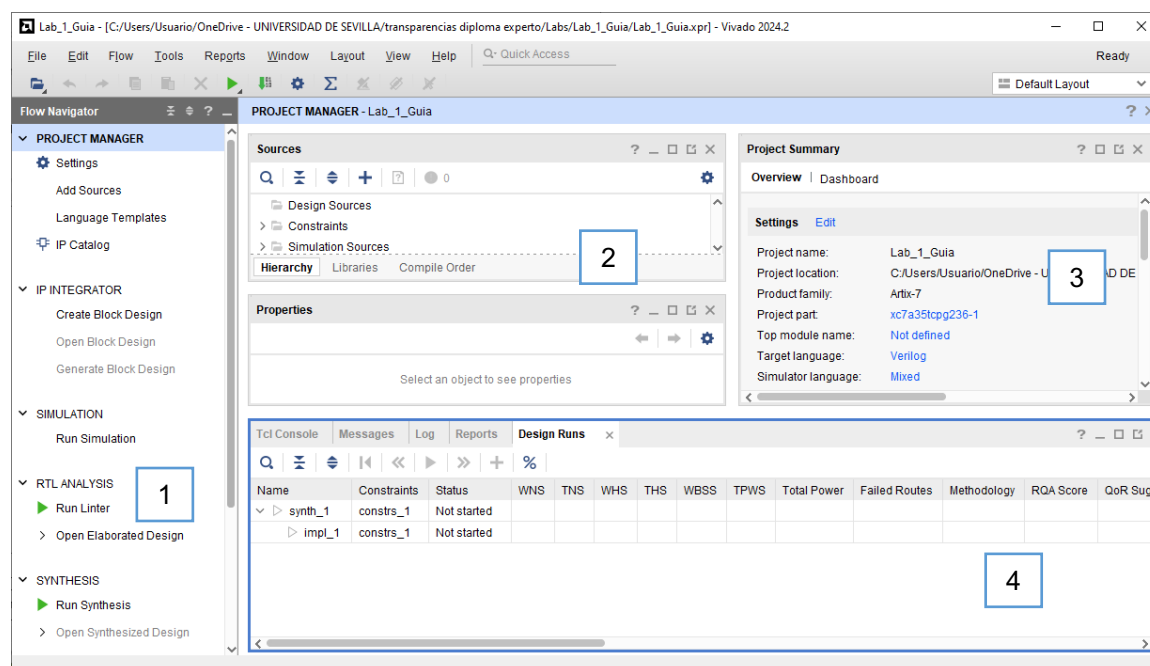
Package: cpg236

Speed Grade: -1



A continuación, se pulsa **Next** y posteriormente **Finish** en la ventana donde se muestra un resumen del proyecto creado.

Tras ello, se abre la interfaz para la gestión de diseños. Permite recorrer el flujo de diseño y analizar los resultados obtenidos.



Se distinguen 4 áreas diferenciadas:

- 1.- **Flow Navigator.** Contiene el **Project manager** y las etapas del flujo de diseño (básicamente simulación, síntesis, implementación y programación del dispositivo).
- 2.- **Fuentes del proyecto** (parte del **Project manager**). En la ventana **Hierarchy** contiene las fuentes del diseño organizadas por “tipos”: descripciones de diseños (Design Sources), testbenches (Simulation Sources) y otras que serán necesarias en distintos pasos del proceso de

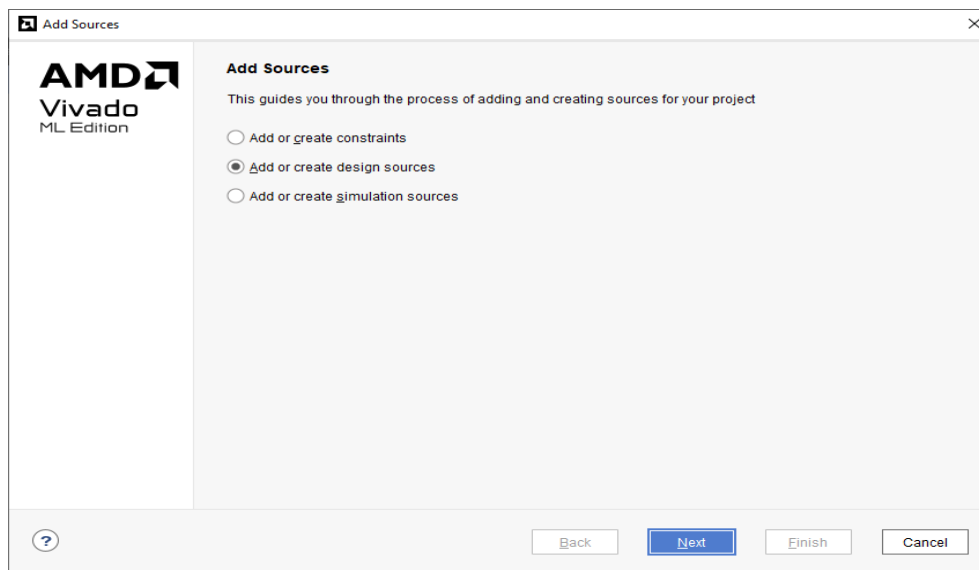
diseño (Constraints u otros).

Existen subcarpetas para organizar las fuentes de simulación y los constraints.

- 3.- **Área de visualización.** Visualización de los ficheros fuentes del proyecto y determinados resultados.
- 4.- **Comunicación** con la herramienta: aparecerán los mensajes que vaya dando la herramienta.

3.2. Creación de diseños HDL dentro del proyecto

Se selecciona **Add Sources** en el **Project Manager** de la ventana **Flow Navigator** o en el **Menu File**

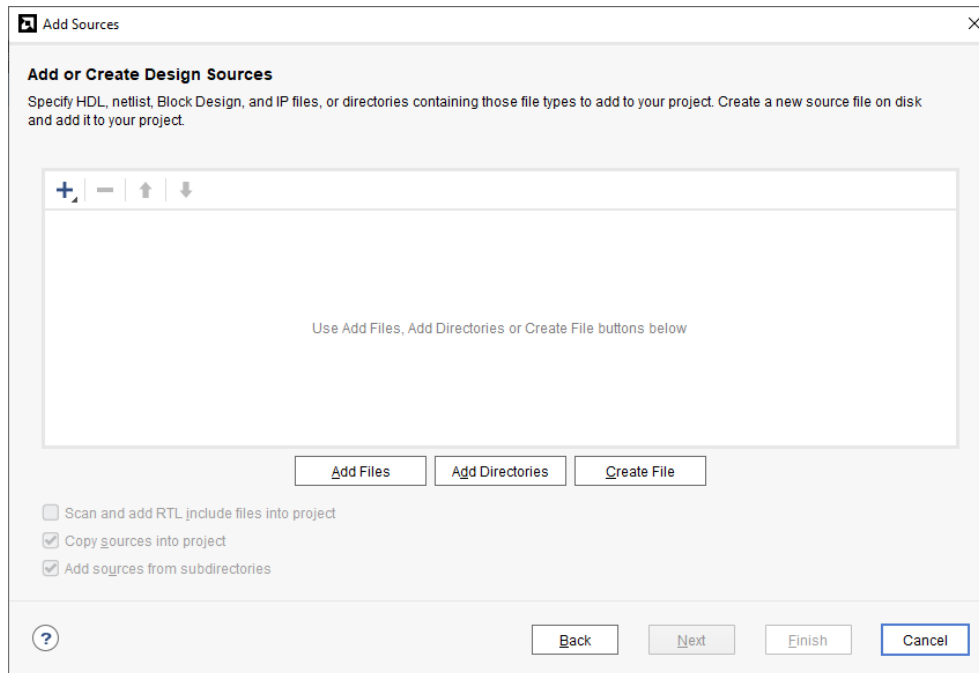


Se elige el tipo de fuente que queremos añadir o crear:

- Añadir o crear constraints (restricciones) (se explica más adelante)
- Añadir o crear fuentes de diseño
- Añadir o crear fuentes de simulación (testbench)

Se elige (**Add or create design sources**). Se pulsa **Next**.

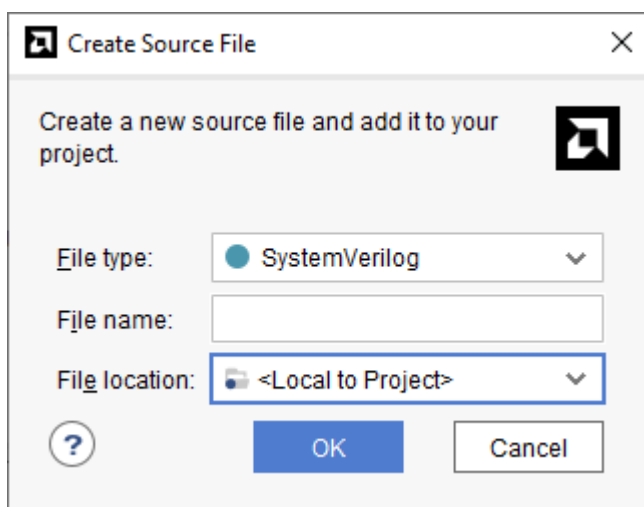
A continuación se elige entre añadir fichero con la descripción HDL del diseño, añadir directorio o crear un nuevo fichero para introducir la descripción HDL del diseño.



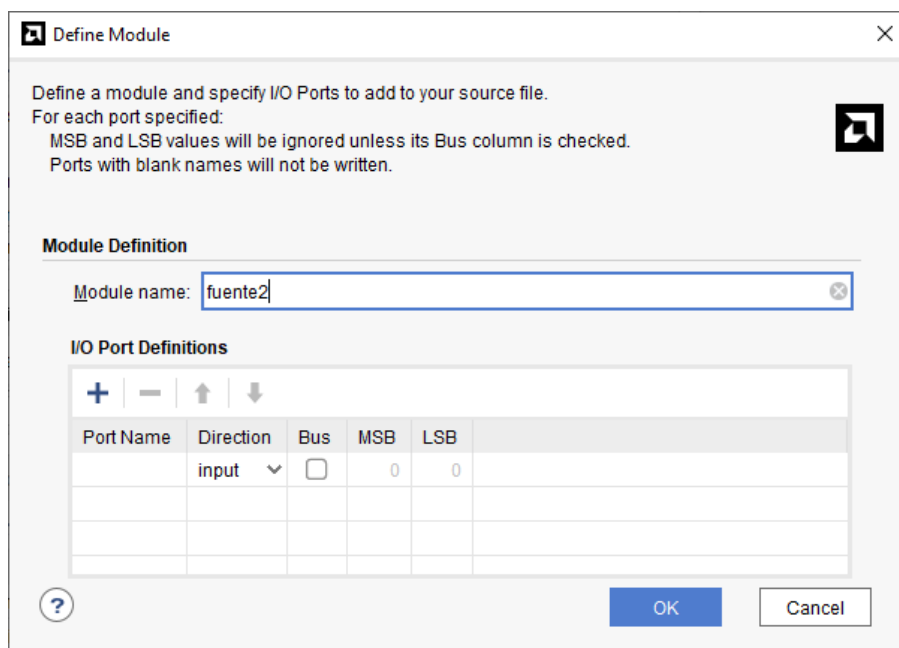
Si se dispone de un fichero con la descripción HDL del diseño se pulsa **Add file**. En la ventana emergente se elige el archivo con la descripción HDL, a continuación **Ok** y, por último **Finish**.

Se observa que el module de la descripción añadida aparece ya en **Design Sources**. Si se pica dos veces sobre el nombre del fichero, se abre en el área de visualización (ventana 3), una pestaña donde se puede leer su descripción.

Si no se dispone de fichero con la descripción HDL, en la ventana anterior se elige la opción **Create File**, aparece una ventana emergente donde se solicita el lenguaje HDL que se usará (seleccione **Verilog**), el nombre de la fuente y el directorio de trabajo donde se guardará la nueva fuente. Por último, **Finish**.



En la siguiente ventana se definen líneas de entradas, salidas y líneas bidireccionales si las hubiera así como su dimensión. A continuación **Ok**.



Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Module name:

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
	input	☐	0	0

Buttons: +, -, ↑, ↓, OK, Cancel

Se puede comprobar que existe una nueva fuente en la ventana **Sources** dentro de **Project Manager**. Al picar sobre su nombre, en el área de visualización (ventana 3), se abre una nueva pestaña en donde aparece una plantilla de descripción HDL a rellenar. Tras ello, **Guardar**, si hubiera errores sintácticos se marcarían para que sean corregidos.

A veces es necesario añadir varios diseños a la carpeta de proyecto y después seleccionar alguno de ellos para realizar su implementación. Es necesario asegurarse que el diseño a elegir sea puesto como “Top Module”, porque Xilinx sólo permite la implementación para el “Top Module”. Para ello, en la ventana **Sources** se pica con el botón derecho sobre el módulo que se quiere actúe como Top y se elige la opción **Set as Top** en el menú desplegable. A la izquierda del nombre del módulo se distingue el icono .

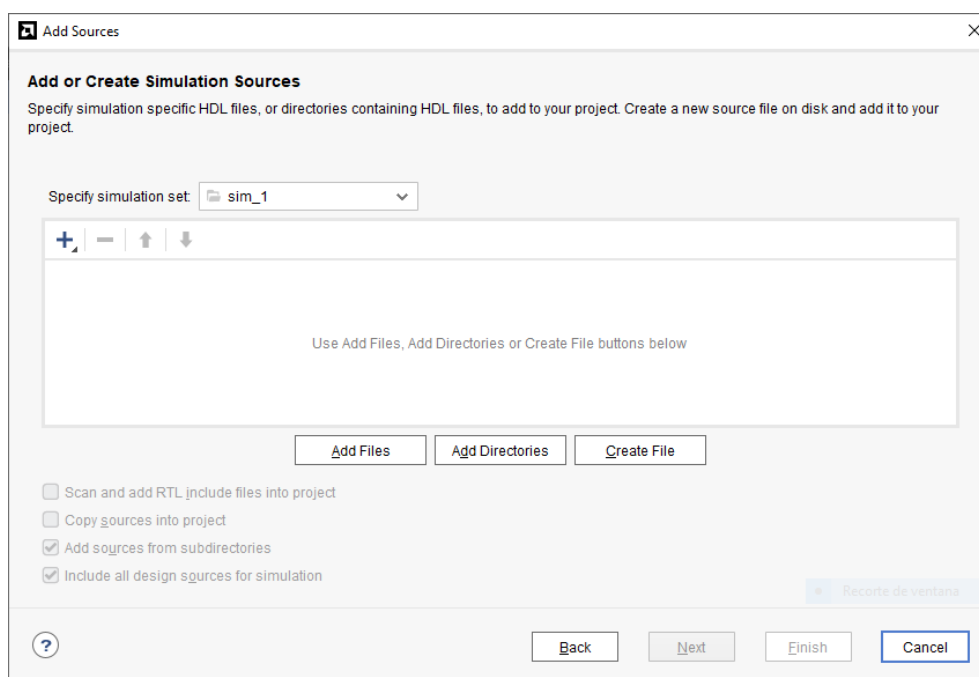
3.3. Simulación

Antes de implementar el diseño en la FPGA es conveniente hacer una simulación de comportamiento para validar la funcionalidad de la descripción HDL.

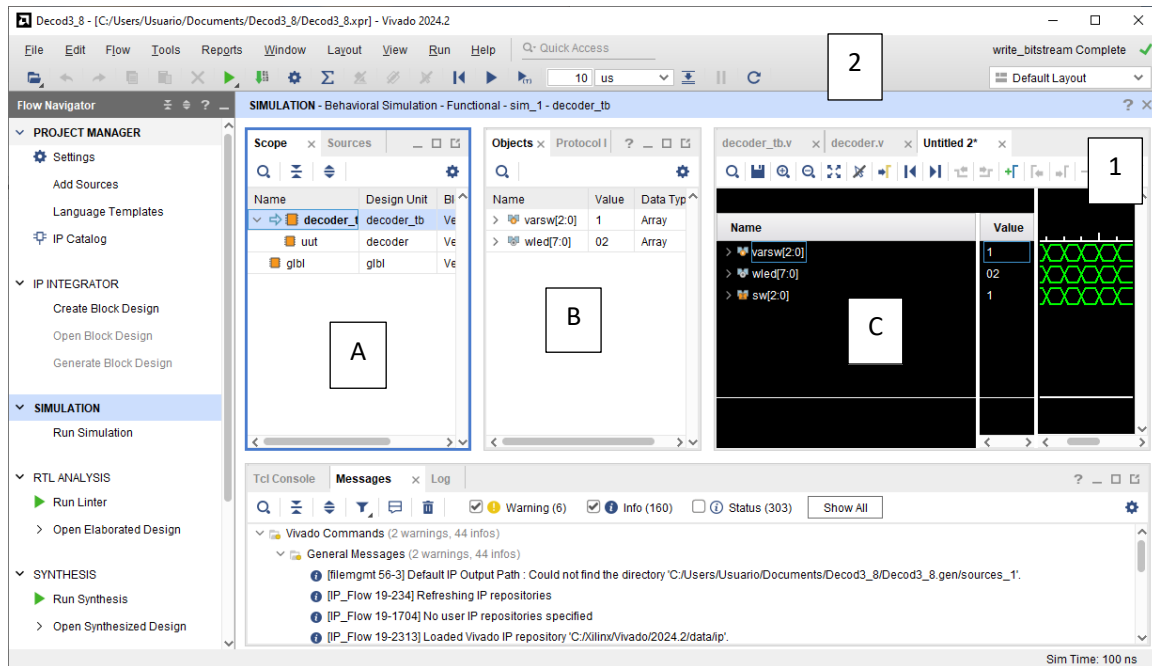
Vivado permite crear, para un mismo diseño lógico, varios conjuntos de bancos de pruebas. Se utilizará, por defecto, el conjunto `sim_1`.

Para realizar una simulación se tiene que disponer de una fuente que contenga un fichero con los estímulos de entrada (testbench) para comprobar la correcta funcionalidad del diseño. Para ello, se selecciona **Add Sources** en el **Project Manager** de la ventana **Flow Navigator** (o en el Menu **File**), indicando que se desea

una fuente para simulación (**Add or Create Simulation Sources**). Se pulsa **Next** y se selecciona **Add File** en la ventana emergente si el testbench se tiene disponible entre los ficheros suministrados. En la ventana emergente se selecciona dicho fichero, se pulsa **Finish** y a continuación se carga en el entorno, se puede visualizar en la ventana **Sources**. Compruebe que en la carpeta Simulation Sources aparece el testbench incorporado como **Top** y de él cuelga el módulo a simular. Picando dos veces sobre el nombre del fichero testbench se abre una nueva pestaña en el Área de visualización en el que se puede leer su contenido con los patrones de test que se van a utilizar para realizar la simulación funcional. Este proceso también se puede realizar creando un fichero nuevo (**Create File**) y a continuación se siguen los pasos indicados en el apartado 3.2.



Para lanzar la simulación, en la ventana **Flow Navigator** se pulsa con el botón izquierdo sobre la opción **Run Simulation** y se elige “**Run Behavioral Simulation**”. Si se pulsara con el botón derecho se puede entrar en **Simulation Settings** para controlar opciones de simulación. Una vez completada se abre una ventana de simulación donde se pueden identificar las siguientes partes:



A.- Ventana **Scope** (A en la figura) que permite moverse por la jerarquía de diseño. Por defecto está seleccionado el module correspondiente al testbench.

B.- Ventana **Objects** (B) que tiene las señales asociadas al módulo que esté seleccionado en **Scope**.

C. Ventana de visualización de formas de onda (pestaña **Untitled**). Inicialmente muestra las señales del testbench.

Con la barra de herramientas (1 en la figura)) se puede, entre otras cosas, cambiar el Zoom, ver simulación completa, moverse por los cambios de la señal seleccionada, ..., por defecto se simula hasta 1000ns.

Seleccionando las señales en el visualizador, con el botón derecho se puede cambiar el tipo de representación (**Radix**: binario, hexadecimal,...), el color, renombrar, expandir las líneas si son buses, ...

Para incluir líneas internas en la ventana de simulación, en la ventana **Scope**, se selecciona el módulo que se simula (uut) y, en caso de ser un diseño jerárquico, se despliega para ver sus bloques. Seleccionado el bloque de interés, en la ventana **Objectcs** aparecen las líneas de entrada/salida de ese bloque y seleccionando y arrastrando con el ratón se llevan hasta la ventana de visualización de ondas quedando incorporadas (también se pueden trasladar a la ventana de ondas al picar con el botón derecho en la línea de interés y se elige la opción **Add to Wave Window**). Posteriormente, para ver su comportamiento, hay que reiniciar la simulación accionando el icono **Restart** y **Run All** del marco superior de la ventana (2).

Si se modificara el testbench y se quisiera lanzar una nueva simulación, se acciona sobre el icono **Relaunch** del marco superior de la ventana (2).

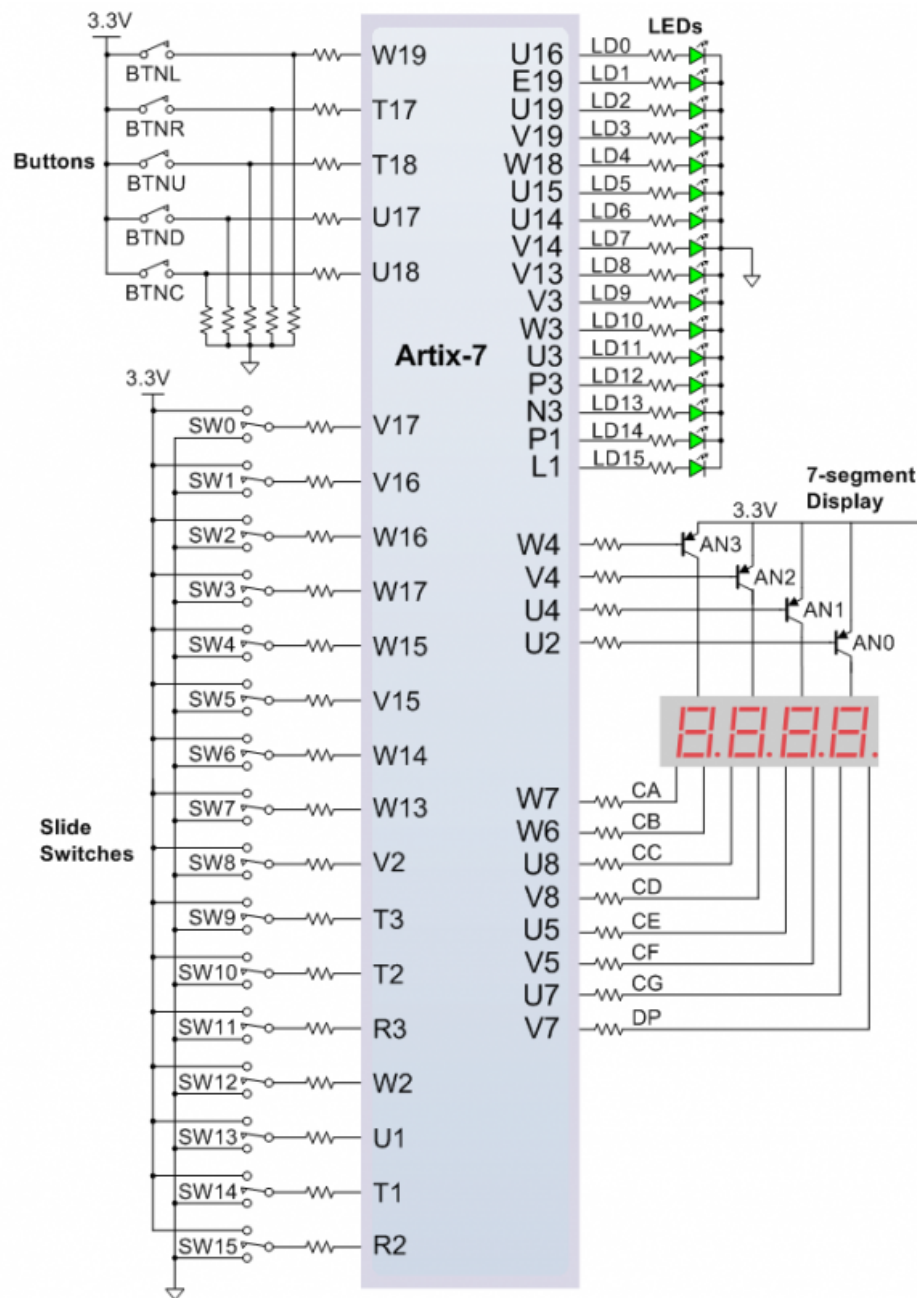
Se puede salvar (segundo icono en la barra de la ventana de visualización) la configuración del visualizador. Puede ser útil si se ha personalizado, para evitar repetir esta tarea en futuras simulaciones.

Para cerrar la simulación se pulsa X en la barra del **Behavioral Simulation View**. A continuación **Ok** y se salva o descarta la configuración del visualizador de formas de onda.

3.4. Restricciones

Antes de implementar el diseño en la placa Basys3 es necesario especificar los pines concretos de la FPGA que van a utilizarse y conectarse a las entradas y salidas del diseño, de modo que sea posible usar los interruptores, botones, leds, ... que hay en la placa.

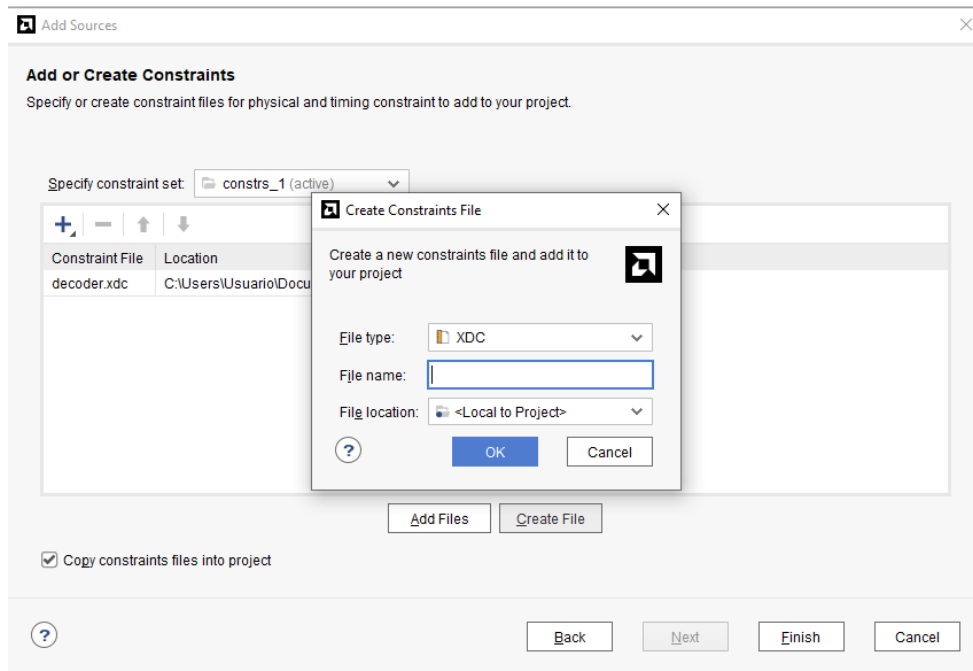
En la siguiente figura se puede comprobar la asociación de los pines de la FPGA con los dispositivos físicos de entrada/salida de la placa. Por ejemplo, puede observarse que el pin U16 está conectado al Led LD0, el pin E19 al led LD1, el pin W19 al botón izquierdo (BTNL), etc.



Para cada diseño, habrá que imponer restricciones sobre qué terminal físico se asocia a cada señal de entrada o salida. Para añadir las restricciones (constraints), se añade al proyecto un fichero de restricciones con extensión .xdc

Para la placa hay un archivo “master” suministrado por el fabricante que simplifica esta tarea (Basy3_Master.xdc). Se añade al proyecto y se edita para adecuarlo a nuestro diseño.

Para añadirlo se sigue el procedimiento antes descrito (**Add Sources**), indicando ahora **Add or create constraints**. Una vez añadido, se puede editar:



- Se elimina el símbolo de comentario (#) en las líneas del fichero asociadas a los pines que se van a usar en el diseño. Por ejemplo, switches para las entradas del diseño y leds para observar las salidas.

- Se renombran los puertos utilizados (después de la función **get_ports**) usando los nombres concretos de señal de nuestro diseño.

3.5. Generación del fichero de programación

Una vez especificadas las asociaciones con el fichero “.xdc”, se genera ahora el fichero que se volcará sobre la FPGA para su programación. Para ello, se selecciona **Generate Bitstream** en el área del **Flow Navigator**. En este punto el programa advierte de que no se ha realizado aún la implementación, pero no hay problema porque la generación de bitstream incluye los pasos previos necesarios (síntesis e implementación). Se confirma pulsando Yes. Aparece una ventana emergente en la que se pregunta en qué directorio volcar el fichero, se mantiene el “por defecto” y se pulsa **OK**. En la carpeta del proyecto aparece un fichero con extensión .bit que es el bitstream necesario para programar (configurar) la FPGA.

3.6. Programación de la FPGA

Los pasos a seguir para programar la FPGA son:

1. Se conecta la placa Basys3 al ordenador mediante el puerto USB.
2. Se comprueba que el jumper JP1 de la placa (esquina superior derecha) está en la posición **JTAG**.
3. Se enciende la placa (switch esquina superior izquierda de la placa).
4. En el área **Flow Navigator**, se pulsa **Open Hardware Manager -> Open Target -> Auto Connect**

5. Se acciona **Program Device** en **Open Hardware Manager** se abre una ventana y en ella se selecciona la FPGA usada.
6. Tras ello, en la ventana emergente se selecciona el fichero .bit generado previamente y se acciona **Program**.

A partir de ese momento, la FPGA está programada según el diseño realizado.