



# Introduction to Vivado

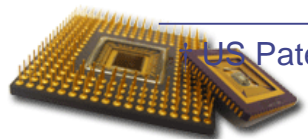
Lan-Da Van and Chun-Jen Tsai  
Department of Computer Science  
National Yang Ming Chiao Tung University  
Taiwan, R.O.C.  
*Fall, 2023*



# Target Technology of Digital Labs

Mat 1

- ◆ Digital circuits can be implemented in different ways.
  - Circuits Boards
    - ◆ Circuit board design using standard IC parts (e.g., 74SLxx)
  - Application Specific ICs
    - ◆ Full-custom and Cell-based IC designs
  - Programmable logics
    - ◆ Field Programmable Gate Array (FPGA) design
  
- ◆ Here, we use Xilinx FPGAs for circuit implementation.
  - Xilinx is the largest FPGA manufacturing company in the world.
  - Ross Freeman, the co-founder of Xilinx, invented the very first FPGA in 1985†.



US Patent 4,870,302



# Xilinx Vivado Design Suite

Mat 1

- ◆ Xilinx has two different EDA tools for FPGA-based digital system designs.
  - Vivado Design Suite
    - ◆ Only for 7<sup>th</sup>-generation FPGAs and above
    - ◆ Unified IDE for both “SoC” and “digital circuit” designs
  - ISE Design Suite
    - ◆ For 7<sup>th</sup>- and older generations of FPGAs
    - ◆ ISE EDK for SoC designs
    - ◆ ISE Project Navigator for digital circuit designs
- ◆ In this course, we use the Vivado Design Suite for digital circuit design.





# Vivado Circuit Implementation Flow

Mat 1

## ◆ Step 1: Design Entry

- Input your circuit design using Hardware Description Language (HDL), such as Verilog or VHDL

## ◆ Step 2: Synthesis

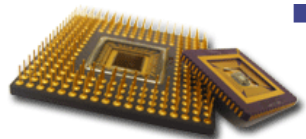
- Convert from the HDL programs or schematics to a netlist file that define a list of circuit blocks and how they are connected

## ◆ Step 3: Mapping

- Determine what FPGA resource will be used to implement which part of the netlist

## ◆ Step 4: Place-and-Route

- Determine physical location and routing of the circuit resource
- A “\*.bit” file will be generated for the FPGA device.





# Vivado Circuit Debug Flow

Mat 1

- ◆ Your design may not be perfect in the first try!
  - Circuit debugging is done via “simulation” or “signal probing”.
- ◆ Vivado supports several simulation types. In particular:
  - Behavioral simulation
    - ◆ Functional simulation before synthesis; assumes zero delay
  - Post implementation functional simulation
    - ◆ Functional simulation after synthesis; assumes zero delay
  - Post implementation timing simulation
    - ◆ Simulate signal switching of your circuit with exact signal delays on the target devices
    - ◆ Also called “post-sim”
- ◆ Vivado Logic Analyzer can analyze runtime signals.





# Install Your Own Vivado Design Suite

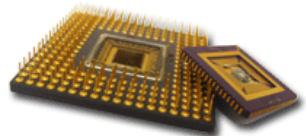
Mat 1

- ◆ Install a copy of Vivado ML Edition - 2023.1 Full Product Installation onto your computer.

- You can download it from:

<https://www.Xilinx.com/support/download.html>

- ◆ The installation requires about 130 GB of disk space.
  - Please install the “Web Installer” version and register online for a free license.





# Vivado Installation Guide (1/2)

Mat 1

◆ You must select the right version upon installation:

**AMD Unified Installer for FPGAs & Adaptive SoCs 2023.1 - Select Product to Install**

Select Product to Install

Select a product to continue installation. You will be able to customize the content in the next page.

- ☐ Vitis  
Installs Vitis Core Development Kit for embedded software and application acceleration development on AMD platforms. Vitis installation includes Vivado Design Suite. Users can also install Vitis Model Composer to design for AI Engines and Programmable Logic in MATLAB and Simulink.
- ☒ **Vivado**  
Includes the full complement of Vivado Design Suite tools for design, including C-based design with Vitis High-Level Synthesis programming. Complete device support, cable driver, and Document Navigator included. Users can also install Vitis Model Logic in MATLAB and Simulink.
- ☐ BootGen  
Installs Bootgen for creating bootable images targeting AMD SoCs and FPGAs.
- ☐ Lab Edition  
Installs only the Vivado Lab Edition. This standalone product includes Vivado Design Programmer, Vivado Logic Analyzer
- ☐ Hardware Server  
Installs hardware server and JTAG cable drivers for remote debugging.
- ☐ Power Design Manager (PDM)  
Installs only the Power Design Manager (PDM). Power Design Manager is a standalone design tool used to estimate power the Xilinx Power Estimator (XPE) file exchange format for importing data from Vivado and XPE.
- ☐ Documentation Navigator (Standalone)  
Documentation Navigator (DocNav) provides access to AMD FPGAs & Adaptive SoCs technical documentation both on the installation without Vivado Design Suite.

**AMD Unified Installer for FPGAs & Adaptive SoCs 2023.1 - Select Edition to Install**

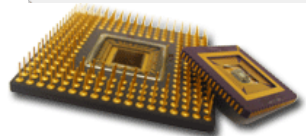
Select Edition to Install

Select an edition to continue installation. You will be able to customize the content in the next page.

- ☒ **Vivado ML Standard**  
Vivado ML Standard Edition is the no-cost, device limited version of the Vivado ML Enterprise edition. Users can add Vitis Model Composer which is an AMD toolbox for MATLAB and Simulink to design for AI Engines and Programmable Logic. If you have been using AMD System Generator for DSP, you can continue development using Vitis Model Composer.
- ☐ Vivado ML Enterprise  
Vivado ML Enterprise Edition includes the full complement of Vivado Design Suite tools for design, including C-based design with Vitis HLS, implementation, verification, and device programming. Complete device support, cable drivers, and documentation Navigator are included. Users can add Vitis Model Composer which is an AMD toolbox for MATLAB and Simulink to design for AI Engines and Programmable Logic. If you have been using AMD System Generator for DSP, you can continue development using Vitis Model Composer.

Copyright © 1986-2022 Xilinx, Inc. All rights reserved.  
Copyright © 2022-2023 Advanced Micro Devices, Inc. All rights reserved.

Choose "Vivado" and "ML Standard"





# Vivado Installation Guide (2/2)

Mat 1

## ◆ Selecting the required packages and FPGA devices:

AMD Unified Installer for FPGAs & Adaptive SoCs 2023.1 - Vivado ML Standard

**Vivado ML Standard**

Customize your installation by (de)selecting items in the tree below. Moving cursor over selections below provide additional information.

Vivado ML Standard Edition is the no-cost, device limited version of the Vivado ML Enterprise edition. Users can add Vitis Model Composer which is an AMD toolbox for MATLAB and Simulink to design for AI Engines and Programmable Logic. If you have been using AMD System Generator for DSP, you can continue development using Vitis Model Composer.

**Design Tools**

- ☒ Vivado Design Suite
  - ☒ Vivado
  - ☒ Vitis HLS
- ☒ Vitis Model Composer (Toolbox for MATLAB and Simulink. Includes the functionality of System Generator for DSP)
- ☒ DocNav

**Devices**

- ☒ Install Devices for Kria SOMs and Starter Kits
- ☒ Production Devices
  - ☒ SoCs
    - ☒ 7 Series (limited support)
      - ☒ Artix-7
      - ☐ Kintex-7
      - ☐ Spartan-7
      - ☐ Virtex-7
    - ☐ UltraScale (limited support)
    - ☒ UltraScale+ (limited support)
    - ☐ Versal ACAP
  - ☐ Engineering Sample Devices
- ☒ Installation Options
  - ☒ Install Cable Drivers (You MUST disconnect all Xilinx Platform Cable USB II cables before proceeding)

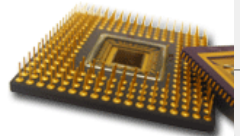
Download Size: 21.18 GB  
Disk Space Required: 72.46 GB

Reset to Defaults

Copyright © 1986-2022 Xilinx, Inc. All rights reserved.  
Copyright © 2022-2023 Advanced Micro Devices, Inc. All rights reserved.

< Back   Next >   Cancel 8

In this course, we only used an Artix FPGA. To save disk space, uncheck all other FPGA devices.







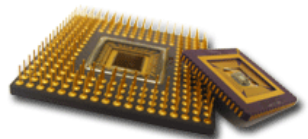
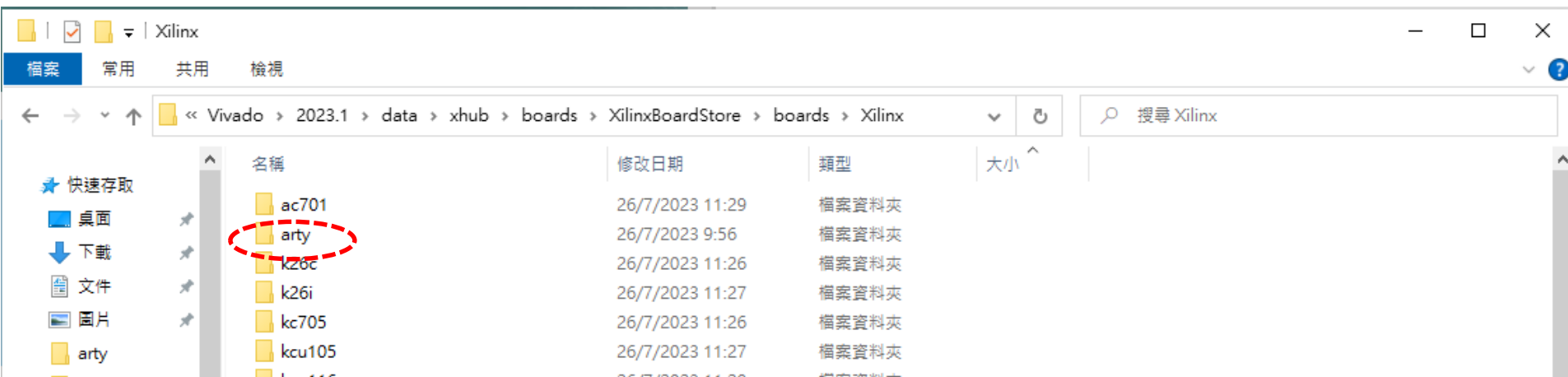
# Installation of Arty Board Definitions

Mat 1

❖ After the installation of Vivado, you must install the board definition file of Arty:

- Download arty.zip from E3.
- Unzip arty.zip to the following directory:

C:/<INST\_DIR>/Vivado/2023.1/data/xhub/boards/  
XilinxBoardStore/boards/Xilinx

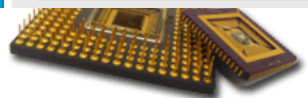
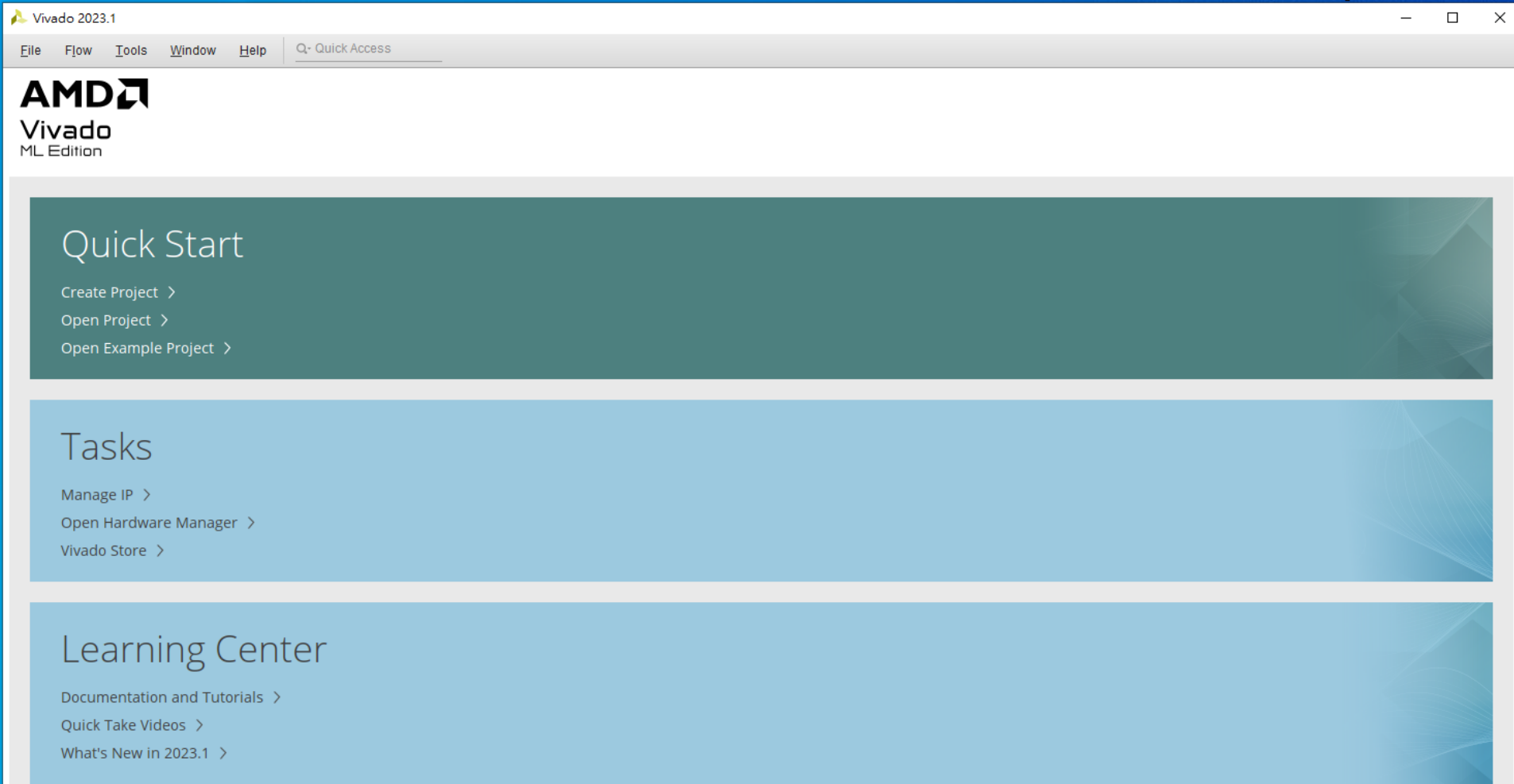




# Launch Vivado 2023.1

Mat 1

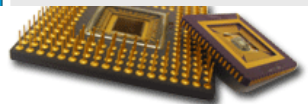
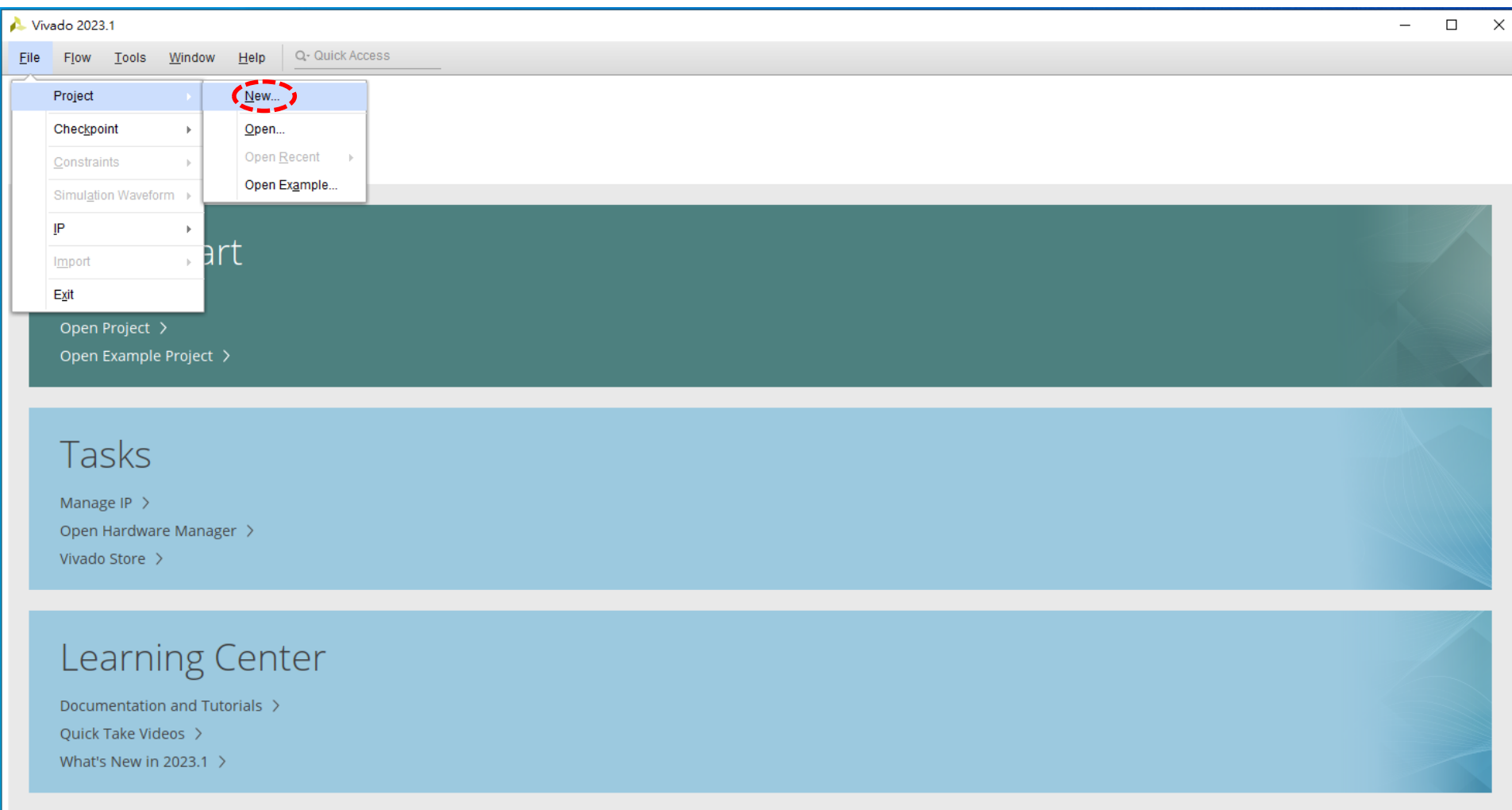
◆ Double-click the Vivado 2023.1 icon on the desktop:





# Create a New Project in Vivado

Mat 1





# Select Project Type

Mat 1

New Project

**Project Type**

Specify the type of project to create.

☒ **RTL Project**  
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☒ **Do not specify sources at this time**

☐ Project is an extensible Vitis platform


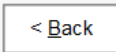
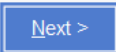
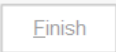

☐ **Post-synthesis Project**  
You will be able to add sources, view device resources, run design analysis, planning and implementation.

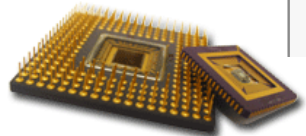
☐ Do not specify sources at this time

☐ **I/O Planning Project**  
Do not specify design sources. You will be able to view part/package resources.

☐ **Imported Project**  
Create a Vivado project from a Synplify Project File.

☐ **Example Project**  
Create a new Vivado project from a predefined template.





# Select the Target FPGA Board

Mat 1

New Project ×

**Default Part**

Choose a default Xilinx part or board for your project.

Parts **Boards**

[Reset All Filters](#)

Vendor:  Name:  Board Rev:

Search:

Display Name	Preview	Status	Vendor	File Version	Part	I/O Pin C
Arty		Installed	digilentinc.com	1.1	xc7a35ticsg324-1L	324
Artix-7 AC701 Evaluation Platform <a href="#">Add Companion Card</a> <a href="#">Connections</a>		Installed	xilinx.com	1.4	xc7a200tfbg676-2	676



# Add a New HDL Source Code

Mat 1

Lab1 - [D:/Vivado\_2023/Lab1/Lab1.xpr] - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Help Q- Quick Access

Ready

Default Layout

Flow Navigator

PROJECT MANAGER - Lab1

**PROJECT MANAGER**

- Settings
- Add Sources**
- Language Templates
- IP Catalog

**IP INTEGRATOR**

- Create Block Design
- Open Block Design
- Generate Block Design

**SIMULATION**

- Run Simulation

**RTL ANALYSIS**

- Run Linter
- Open Elaborated Design

**SYNTHESIS**

- Run Synthesis
- Open Synthesized Design

**IMPLEMENTATION**

- Run Implementation
- Open Implemented Design

**PROGRAM AND DEBUG**

**Sources**

- Design Sources
- Constraints
- Simulation Sources
  - sim\_1
- Utility Sources

Hierarchy Libraries Compile Order

**Properties**

Select an object to see properties

**Project Summary**

Overview | Dashboard

**Settings** Edit

Project name: Lab1  
 Project location: D:/Vivado\_2023/Lab1  
 Product family: Artix-7  
 Project part: Artix (xc7a35ticsg324-1L)  
 Top module name: Not defined  
 Target language: Verilog  
 Simulator language: Mixed

**Board Part**

Display name: Arty  
 Board part name: digilentinc.com:arty:part0:1.1  
 Board revision: C.0  
 Connectors: No connections

Tcl Console Messages Log Reports **Design Runs**

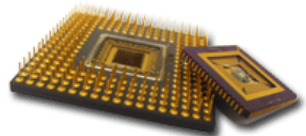
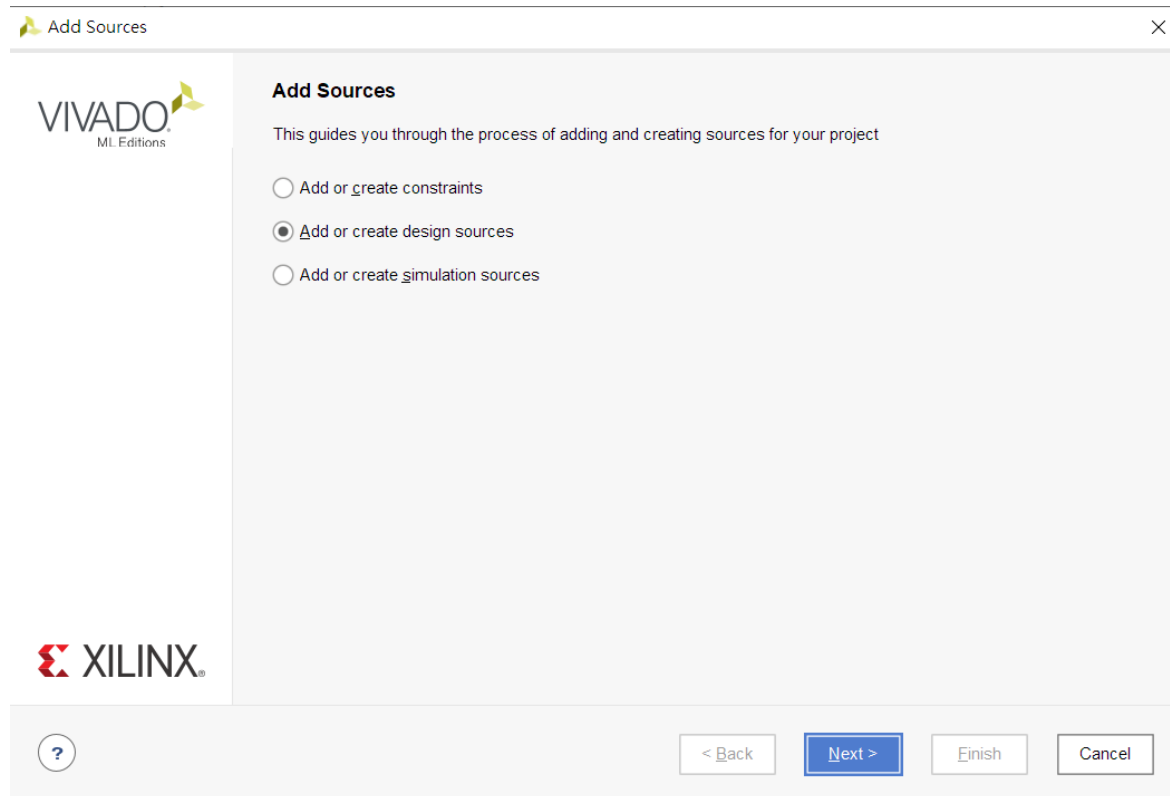
Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM
synth_1	constrs_1	Not started															
impl_1	constrs_1	Not started															



# Specifying the Source Type to Create

Mat 1

- ◆ There are several types of source files in a circuit design project: design sources, constraint sources, simulation sources, and memory sources, etc.





# Create a 4-bit Full Adder Design

Mat 1

◆ Let's create a design source from scratch!

**Add Sources**

**Add or Create Design Sources**

Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk and add it to your project.

Use Add Files, Add Directories or Create File buttons below

☐ Scan and add RTL include files into project

☐ Copy sources into project

☐ Add sources from subdirectories

**Create Source File**

Create a new source file and add it to your project.

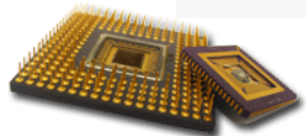
File type: Verilog

File name: FullAdder

File location: <Local to Project>

OK Cancel

< Back Next > Finish Cancel



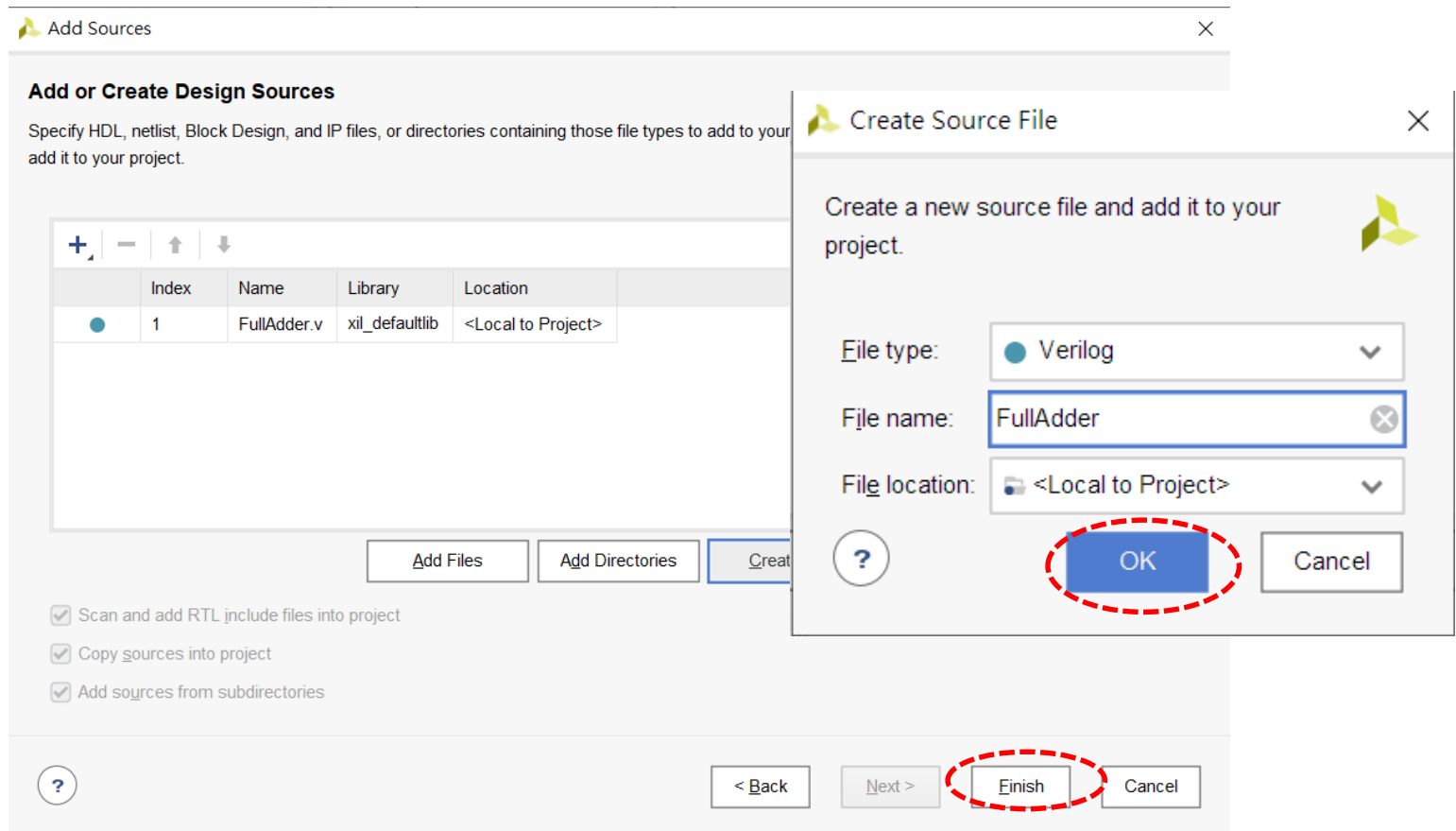




# Confirm to Create the Verilog Module

Mat 1

◆ Let's create a design source from scratch!

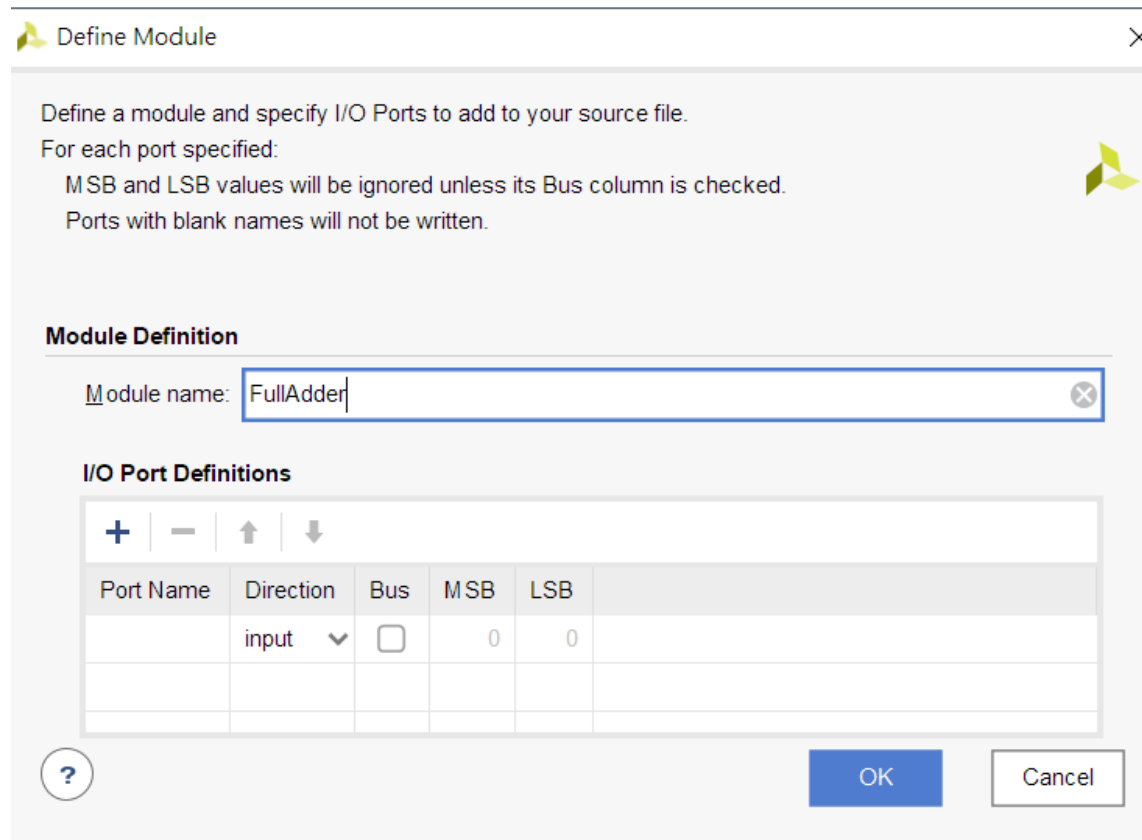




# Confirm to Create the Verilog Module

Mat 1

- ◆ You can define your ports here or do it in the HDL code:

 Define Module


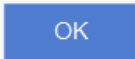
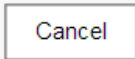
Define a module and specify I/O Ports to add to your source file.  
For each port specified:  
MSB and LSB values will be ignored unless its Bus column is checked.  
Ports with blank names will not be written.

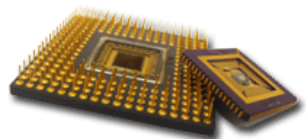
**Module Definition**

Module name:

**I/O Port Definitions**

Port Name	Direction	Bus	MSB	LSB
	input	<input type="checkbox"/>	0	0





# An Empty Verilog Template is Created

Mat 1

◆ You can type in your Verilog code in the editor window:

Lab1 - [D:/Vivado\_2023/Lab1/Lab1.xpr] - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Help Q- Quick Access

Ready

Default Layout

Flow Navigator

PROJECT MANAGER - Lab1

Sources

Design Sources (1)

FullAdder (FullAdder.v)

Constraints

Simulation Sources (1)

sim\_1 (1)

Utility Sources

Hierarchy Libraries Compile Order

Properties

Select an object to see properties

Project Summary

Overview | Dashboard

Settings Edit

Project name: Lab1

Project location: D:/Vivado\_2023/Lab1

Product family: Artix-7

Project part: Arty (xc7a35ticsg324-1L)

Top module name: Not defined

Target language: Verilog

Simulator language: Mixed

Board Part

Display name: Arty

Board part name: digilentinc.com:arty:part0:1.1

Board revision: C.0

Connectors: No connections

Tcl Console Messages Log Reports Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM
synth_1	constrs_1	Not started															
impl_1	constrs_1	Not started															



# Type in the HDL Source Code

Mat 1

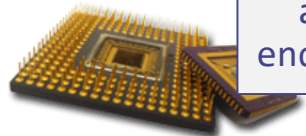
◆ The complete code for a 4-bit full adder is as follows:

```
// ----- A four-bit full adder -----
module FullAdder(A, B, Cin, S, Cout);
    input [3:0] A, B;
    input Cin;
    output [3:0] S;
    output Cout;
    wire [2:0] t;

    FA_1bit FA0(.A(A[0]), .B(B[0]), .Cin(Cin), .S(S[0]), .Cout(t[0]));
    FA_1bit FA1(.A(A[1]), .B(B[1]), .Cin(t[0]), .S(S[1]), .Cout(t[1]));
    FA_1bit FA2(.A(A[2]), .B(B[2]), .Cin(t[1]), .S(S[2]), .Cout(t[2]));
    FA_1bit FA3(.A(A[3]), .B(B[3]), .Cin(t[2]), .S(S[3]), .Cout(Cout));
endmodule

// ----- A 1-bit full adder -----
module FA_1bit(A, B, Cin, S, Cout);
    input A, B, Cin;
    output S, Cout;

    assign S = Cin ^ A ^ B;
    assign Cout = (A & B) | (Cin & B) | (Cin & A);
endmodule
```

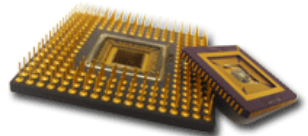
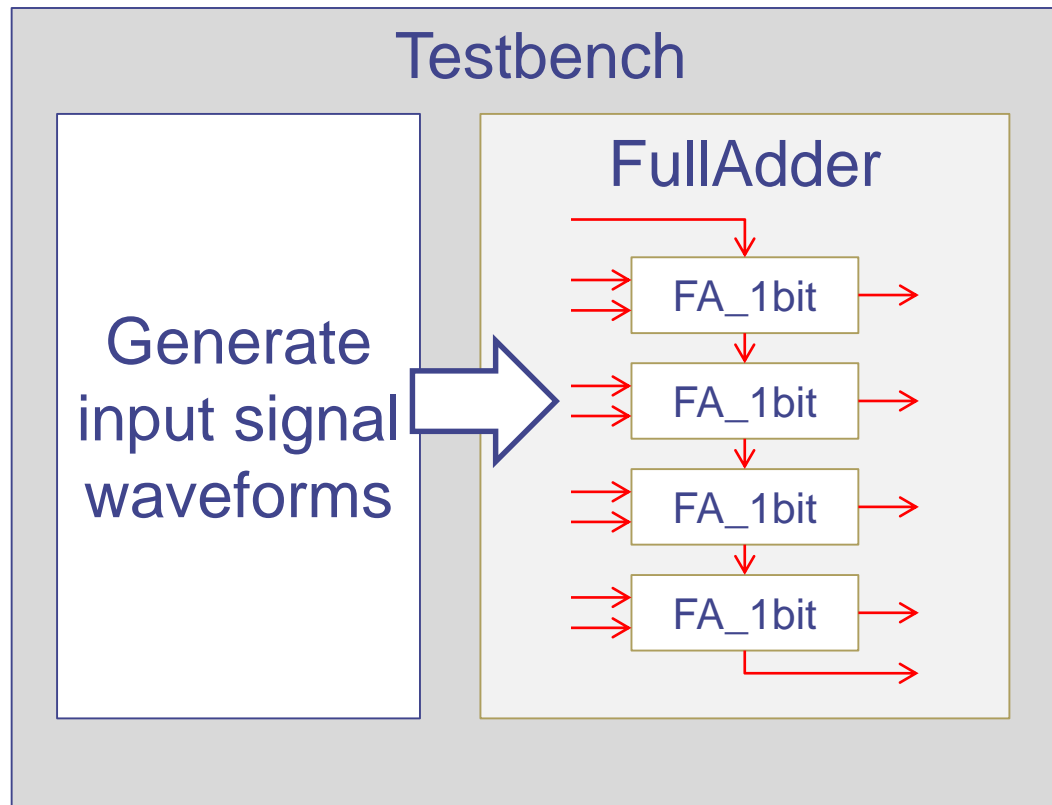




# TestBench Design

Mat 1

- ◆ You must create a testbench to generate input signals that can feed into your circuit module, such that you can analyze the output to verify its correctness.

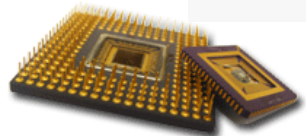
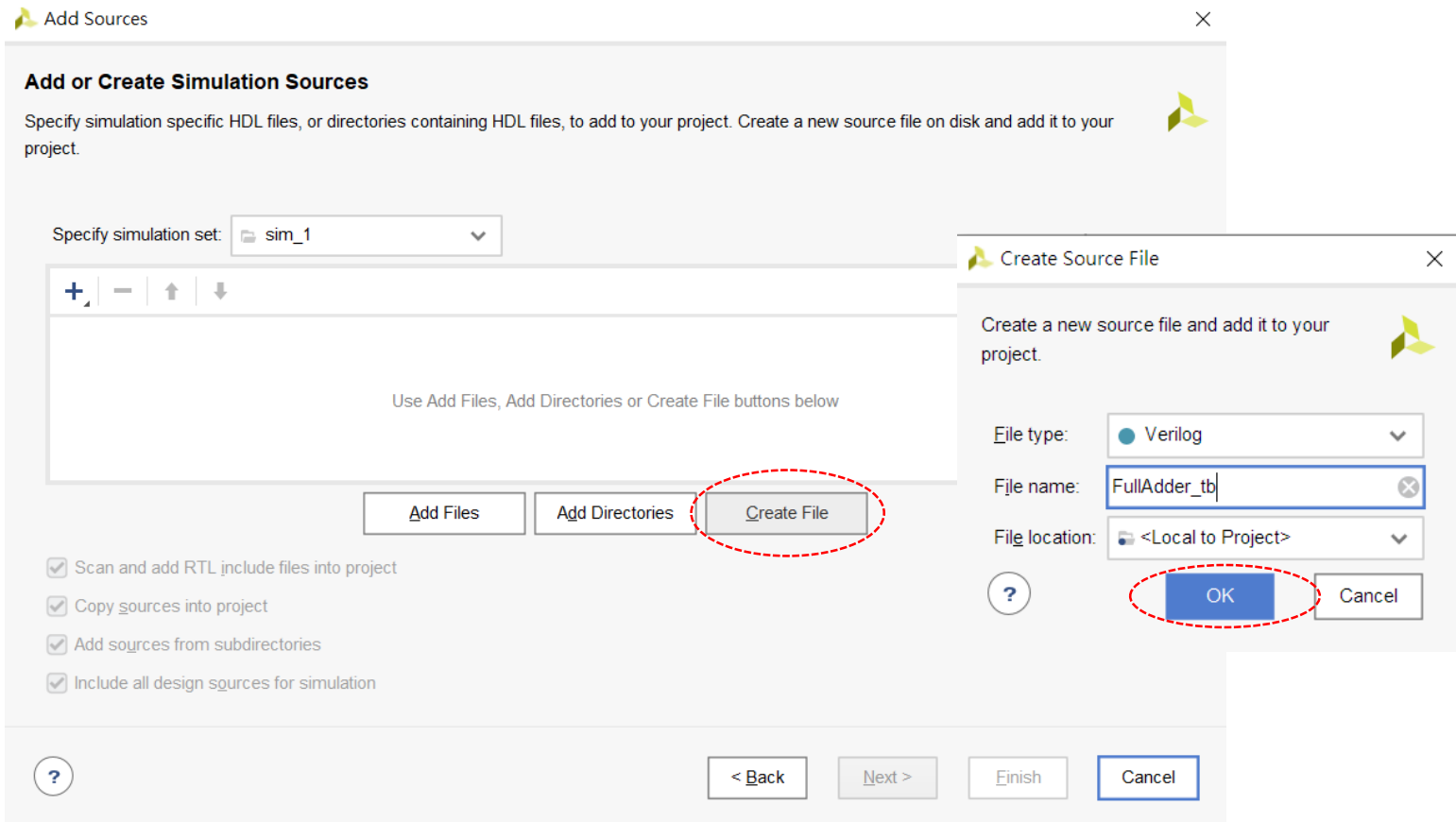




# Create the Testbench Source Code

Mat 1

- Click “Add Sources” button again, and this time, select “Add or create simulation sources”





# Confirm the Creation of the Testbench

Mat 1

- ◆ Here, we include the design sources into the simulation set so that we can test the modules under development.

**Add Sources**

**Add or Create Simulation Sources**

Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project.

Specify simulation set: sim\_1

	Index	Name	Library	Location
	1	FullAdder_tb.v	xil_defaultlib	<Local to Project>

☒ Scan and add RTL include files into project

☒ Copy sources into project

☒ Add sources from subdirectories

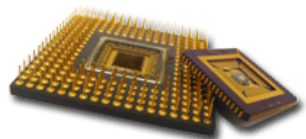
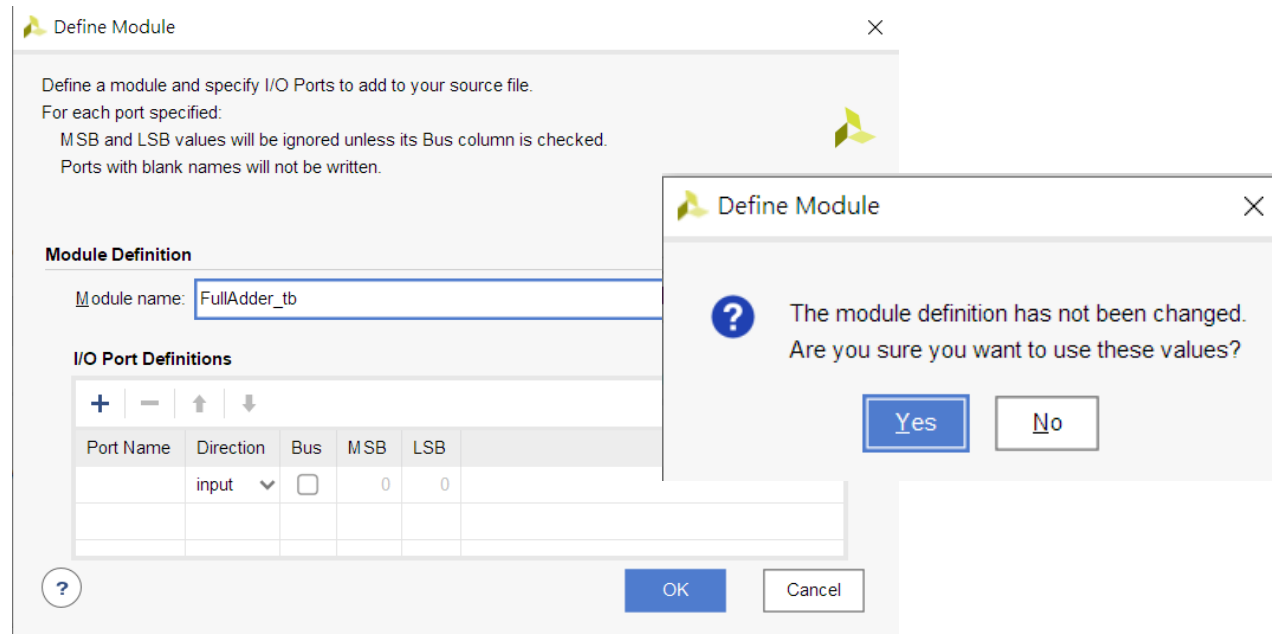
☒ Include all design sources for simulation



# Create the Testbench Template

Mat 1

- ◆ Hit “OK” then “Yes” to create an empty testbench template → top-level of the testbench template usually has no I/O ports







# Type in the Testbench Source Code

Mat 1

Lab1 - [D:/Vivado\_2023/Lab1/Lab1.xpr] - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Help Q: Quick Access

Ready

Default Layout

Flow Navigator

PROJECT MANAGER - Lab1

**Sources**

- Constraints
- Simulation Sources (2)
  - sim\_1 (2)
    - FullAdder (FullAdder.v)
    - FullAdder\_tb (FullAdder\_tb.v)
- Utility Sources

Hierarchy Libraries Compile Order

**Source File Properties**

FullAdder\_tb.v

☒ Enabled

Location: D:/Vivado\_2023/Lab1/Lab1.srcs/sim\_1/new

Type: Verilog

Library: viv\_defaultlib

General Properties

**Project Summary** x FullAdder\_tb.v x

D:/Vivado\_2023/Lab1/Lab1.srcs/sim\_1/new/FullAdder\_tb.v

```

9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module FullAdder_tb(
24
25 );
26 endmodule
27
  
```

**Tcl Console** Messages Log Reports **Design Runs** x

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM
synth_1	constrs_1	Not started															
impl_1	constrs_1	Not started															



# The Sample Testbench Code

Mat 1

- ◆ The template created by Vivado is an empty module; you must add test pattern generators in the module.

```
module FullAdder_tb;

// inputs
reg [3:0] A, B;
reg Cin;

// outputs
wire [3:0] S;
wire Cout;

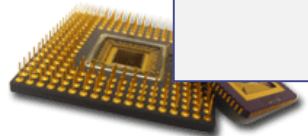
// Instantiate the Unit
// Under Test (UUT)

FullAdder uut(
    .A(A),
    .B(B),
    .Cin(Cin),
    .S(S),
    .Cout(Cout)
);
```

```
initial begin
    // Initialize Inputs
    A = 0; B = 0; Cin = 0;

    // Wait 100 ns for global
    // reset to finish
    #100;

    // Add stimulus here
    A = 4'b0101; B = 4'b1010;
    #50;
    A = 4'b1111; B = 4'b0001;
    #50;
    A = 4'b0000; B = 4'b1111;
    Cin = 1'b1;
    #50;
    A = 4'b0110; B = 4'b0001;
end
endmodule
```





# Run the Simulation

Mat 1

Lab1 - [D:/Vivado\_2023/Lab1/Lab1.xpr] - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Help Q- Quick Access

Ready

Default Layout

Flow Navigator

PROJECT MANAGER - Lab1

Sources

Constraints

Simulation Sources (2)

sim\_1 (2)

FullAdder (FullAdder.v)

FullAdder\_tb (FullAdder\_tb.v)

Utility Sources

Hierarchy Libraries Compile Order

Source File Properties

FullAdder\_tb.v

Enabled

Run Behavioral Simulation **Click!**

Run Post-Synthesis Functional Simulation

Run Post-Synthesis Timing Simulation

Run Post-Implementation Functional Simulation

Run Post-Implementation Timing Simulation

Project Summary

FullAdder\_tb.v \* FullAdder.v \*

D:/Vivado\_2023/Lab1/Lab1.srcs/sim\_1/new/FullAdder\_tb.v

```

1 timescale 1ns / 1ps
2 //////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 2023/09/12 18:50:48
7 // Design Name:
8 // Module Name: FullAdder_tb
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //

```

Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM
synth_1	constrs_1	Not started															
impl_1	constrs_1	Not started															

Vivado Simulator



# Vivado Simulator Window

Mat 1

Lab1 - [D:\Vivado\_2023\Lab1\Lab1.xpr] - Vivado 2023.1

File Edit Flow Tools Reports Window Layout View Run Help Q: Quick Access Ready

10 us

Simulation time duration

Zoom waveform to fit window

Flow Navigator

**PROJECT MANAGER**

- Settings
- Add Sources
- Language Templates
- IP Catalog

**IP INTEGRATOR**

- Create Block Design
- Open Block Design
- Generate Block Design

**SIMULATION**

- Run Simulation

**RTL ANALYSIS**

- Run Linter
- Open Elaborated Design

**SYNTHESIS**

- Run Synthesis
- Open Synthesized Design

**IMPLEMENTATION**

- Run Implementation
- Open Implemented Design

**PROGRAM AND DEBUG**

**SIMULATION - Behavioral Simulation - Functional - sim\_1 - FullAdder\_tb**

Scope Source x ? \_ □ □

Design Sources (1)

- FullAdder (FullAdder.v) (4)
  - FA0 : FA\_1bit (FullAdder.v)
  - FA1 : FA\_1bit (FullAdder.v)
  - FA2 : FA\_1bit (FullAdder.v)
  - FA3 : FA\_1bit (FullAdder.v)
- Constraints
- Simulation Sources (1)
  - sim\_1 (1)
    - FullAdder\_tb (FullAdder.v)
- Utility Sources

Object x Protocol ? \_ □ □

Name	Value	Data Type
A[3:0]	6	Array
B[3:0]	1	Array
Cin	1	Logic
S[3:0]	8	Array
Cout	0	Logic

FullAdder\_tb.v x FullAdder.v x Untitled 1 x

FullAdder\_tb.v

Name	Value
A[3:0]	6
B[3:0]	1
Cin	1
S[3:0]	8
Cout	0

0.000 ns 200.000 ns 400.000 ns 600.000 ns 800.000 ns 1,000.000 ns

28

Sim Time: 1 us