



Introduction to Vivado

Lan-Da Van and Chun-Jen Tsai

Department of Computer Science

National Yang Ming Chiao Tung University

Taiwan, R.O.C.

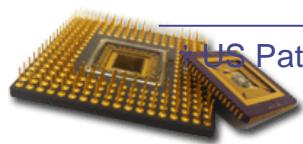
Fall, 2025



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[†].



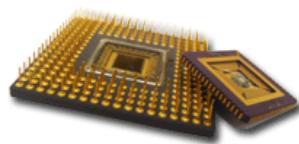
US Patent 4,870,302



Xilinx Vivado Design Suite

Mat 1

- ◆ Xilinx has two different EDA tools for FPGA-based digital system designs.
 - Vivado Design Suite
 - ◆ Only for 7th-generation FPGAs and above
 - ◆ Unified IDE for both “SoC” and “digital circuit” designs
 - ISE Design Suite
 - ◆ For 7th- 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

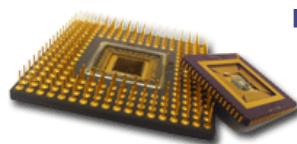
- Convert from the HDL programs or schematics to a netlist file that define a list of circuit blocks and how they are connected

◆ 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.

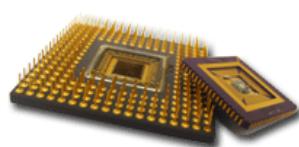




Vivado Circuit Debug Flow

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.



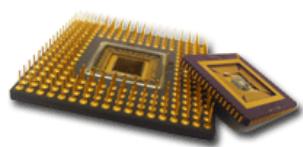


Install Your Own Vivado Design Suite

Mat 1

- ◆ Install a copy of Vivado ML Edition - 2025.1 Full Product Installation onto your computer.
 - You can download it from:
<https://www.Xilinx.com/support/download.html>

- ◆ The installation requires about 82 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 FPGAs & Adaptive SoCs Installer 2025.1 - 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.

Vivis i

Vivado i Choose “Vivado” and “ML Standard”

Vivis Embedded Development i

BootGen i

Lab Edition i

Hardware Server i

Power Design Manager (PDM) i

On-Premises Install for Cloud Deployments i

PetaLinux i

Documentation Navigator (Standalone) i

AMD Unified FPGAs & Adaptive SoCs Installer 2025.1 - 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 © 2022-2025 Advanced Micro Devices, Inc. All rights reserved.

< Back
Next >
Cancel

Copyright © 1986-2022 Xilinx, Inc. All rights reserved.
Copyright © 2022-2025 Advanced Micro Devices, Inc. All rights reserved.

< Back Next > Cancel

7



Vivado Installation Guide (2/2)

Mat 1

◆ Selecting the required packages and FPGA devices:

AMD Unified FPGAs & Adaptive SoCs Installer 2025.1 - 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.

Selected Devices

- Install Devices for Kria SOMs and Starter Kits
- Artix-7 FPGAs

Filter Devices

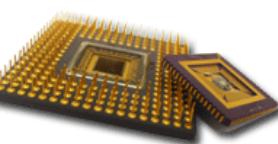
- Design Tools
 - Vivado Design Suite
 - Vivado
 - Vitis HLS
 - Vitis Networking P4
 - Vitis Model Composer(A toolbox for Simulink)
 - Vitis Embedded Development
 - Power Design Manager (PDM)
 - DocNav
 - Devices
 - Install devices for Alveo and edge acceleration platforms
 - Install Devices for Kria SOMs and Starter Kits
 - 7 Series
 - Virtex-7 FPGAs
 - Kintex-7 FPGAs
 - Artix-7 FPGAs
 - Spartan-7 FPGAs
 - SoCs
 - UltraScale
 - UltraScale+
 - Versal Adaptive SoCs
 - Engineering Sample Devices
 - Installation Options
 - Acquire or Manage a License Key

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

Download Size: 22.13 GB Disk Space Required: 82.07 GB [Reset to Defaults](#)

Copyright © 1986-2022 Xilinx, Inc. All rights reserved.
Copyright © 2022-2025 Advanced Micro Devices, Inc. All rights reserved.

[< Back](#) [Next >](#) [Cancel](#)



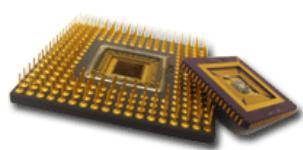
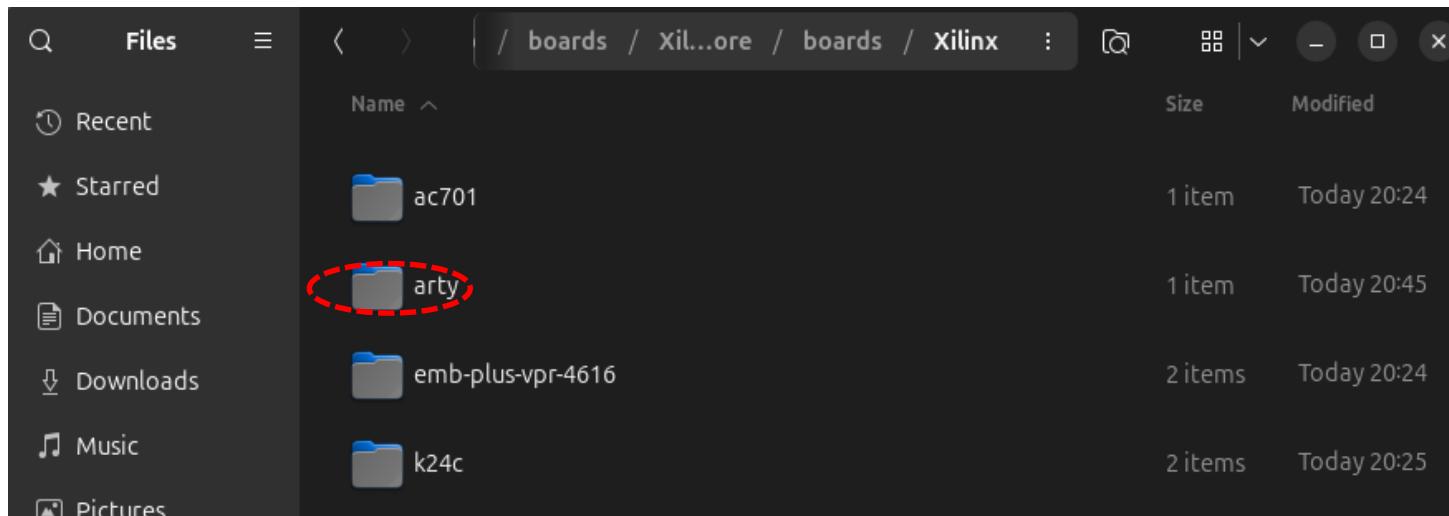


Installation of Arty Board Definitions

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/2025.1/data/xhub/boards/
XilinxBoardStore/boards/Xilinx

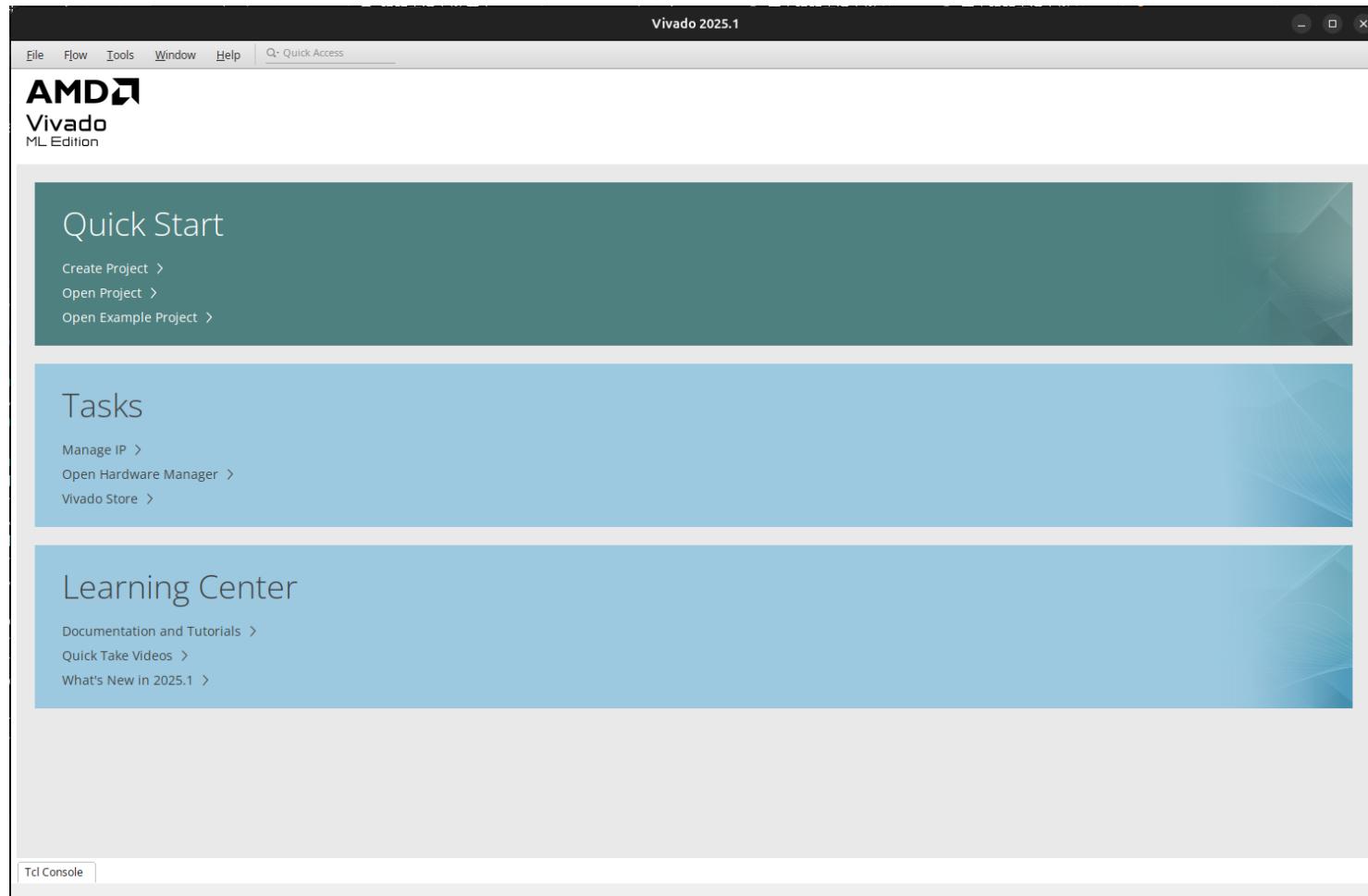




Launch Vivado 2025.1

Mat 1

- ◆ Double-click the Vivado 2025.1 icon on the desktop:





Create a New Project in Vivado

Mat 1

The screenshot shows the Vivado 2025.1 software interface. The top menu bar includes File, Flow, Tools, Window, Help, and Q. Quick Access. A red circle highlights the 'New...' option under the 'Project' dropdown in the 'File' menu. Below the menu bar, there are sections for Tasks and Learning Center, each with a list of links. At the bottom, there is a toolbar with icons for Tcl Console and Create a new project, and a decorative image of two integrated circuits.

Vivado 2025.1

File Flow Tools Window Help Q. Quick Access

Project > **New...**

Checkpoint >

Constraints >

Simulation Waveform >

IP >

Import >

Exit

Create Project >

Open Project >

Open Example Project >

Tasks

Manage IP >

Open Hardware Manager >

Vivado Store >

Learning Center

Documentation and Tutorials >

Quick Take Videos >

What's New in 2025.1 >

Tcl Console

Create a new project



Select Project Type

Mat 1

New Project

Project Type
Specify the type of project to create.

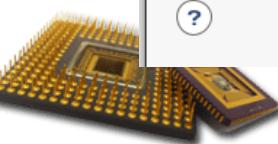
RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
 Do not specify sources at this time
 Project is an extensible Vitis platform

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.



?

Back Next Finish Cancel



Select the Target FPGA Board

Mat 1

New Project

Default Part
Choose a default AMD part or board for your project.

Parts | **Boards**

To fetch the latest available boards from git repository, click on 'Refresh' button. [Dismiss](#)

Reset All Filters

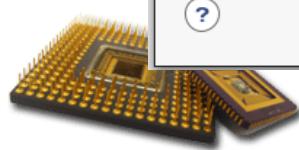
Vendor: All Name: All Board Rev: Latest

Search: (8 matches)

Display Name	Preview	Status	Vendor	File Version	Part
Arty Z7-10		Download	diligentinc.com	1.1	
Arty Z7-20		Download	diligentinc.com	1.1	
Arty		Installed	diligentinc.com	1.1	xc7a35ticsg324-1L

Refresh Catalog was last updated on 08/28/2025 8:49:11 PM

? Back Next Finish Cancel





Add a New HDL Source Code

Mat 1

Screenshot of the Vivado 2025.1 Project Manager interface for a project named "project_1".

The left sidebar shows the Flow Navigator with various sections like PROJECT MANAGER, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, IMPLEMENTATION, and PROGRAM AND DEBUG. The "Add Sources" option under PROJECT MANAGER is highlighted with a red oval.

The main area displays the PROJECT MANAGER - project_1 window. It includes:

- Sources:** Shows Design Sources, Constraints, Simulation Sources (with sim_1 selected), and Utility Sources.
- Properties:** A panel showing "Select an object to see properties".
- Project Summary:** Overview of the project settings:
 - Project name: project_1
 - Project location: /home/rty78452/Desktop/project_1
 - Product family: Artix-7
 - Project part: Arty (xc7a35tic96-1L)
 - Top module name: Not defined
 - Target language: Verilog
 - Simulator language: Mixed
 - Target Simulator: Vivado Simulator
- Board Part:** Details about the Arty board:
 - Display name: Arty
 - Board part name: digilentinc.com:arty:part0:1.1
 - Board revision: C.0
 - Connectors: No connections
 - Repository path: /tools/Xilinx/2025.1/data/xhub/boards
 - URL: www.digilentinc.com/Arty
- Design Runs:** A table showing the status of design runs:

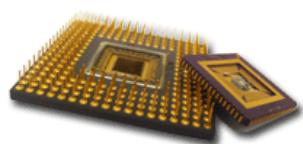
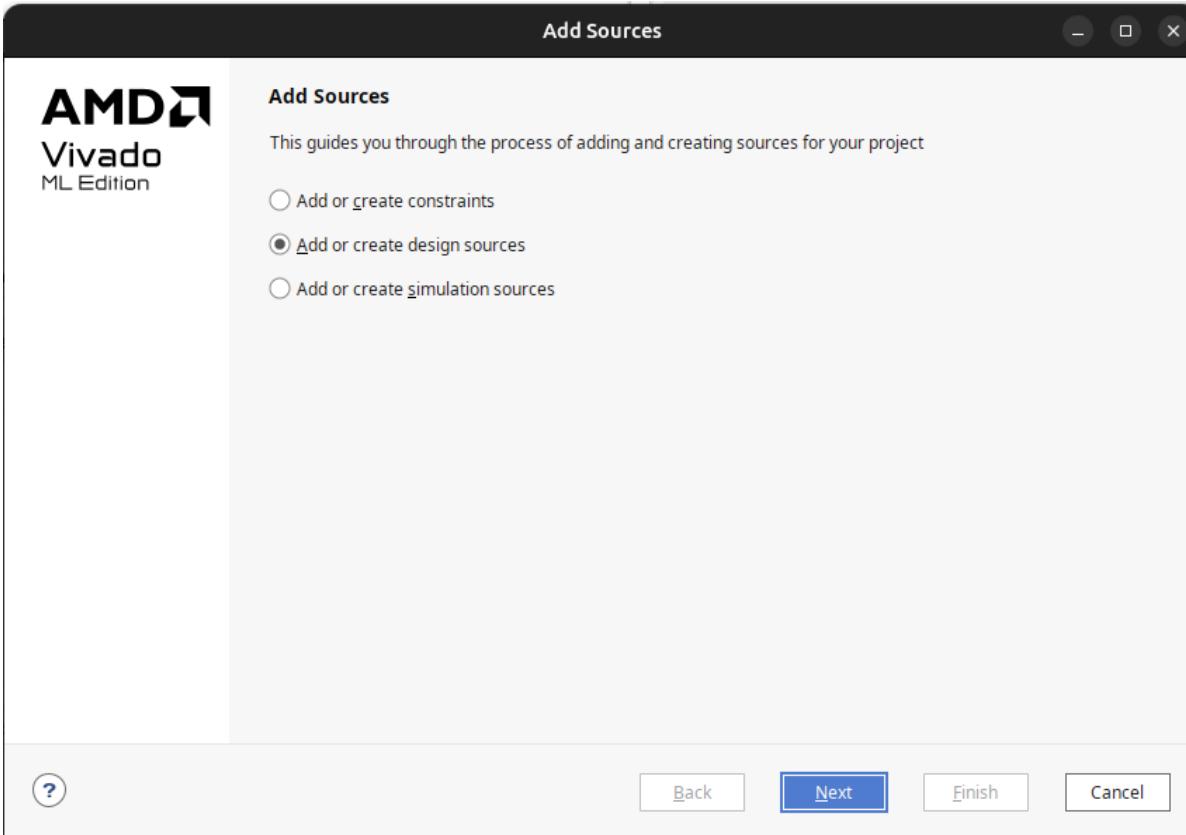
Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP
synth_1	constrs_1	Not started																
impl_1	constrs_1	Not started																



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 b

File type: Verilog

File name: FullAdder

File location: <Local to Project>

OK Cancel

Add Files Add Directories Create File

Scan and add RTL include files into project

Copy sources into project

Add sources from subdirectories

? Back Next Finish Cancel



Confirm to Create the Verilog Module

Mat 1

- ◆ Let's create a design source from scratch!

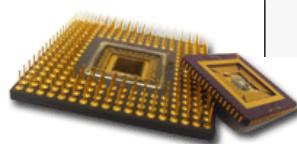
Add Sources

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.

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

Scan and add RTL include files into project
 Copy sources into project
 Add sources from subdirectories





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.

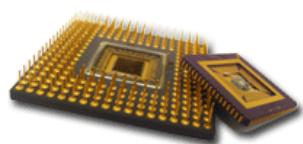
Module Definition

Module name: FullAdder

I/O Port Definitions

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

OK Cancel





An Empty Verilog Template is Created

Mat 1

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

The screenshot shows the Vivado 2025.1 interface with the following details:

- Project Manager - project_1**: The current project is named "project_1".
- Sources**: The "FullAdder" module is listed under "Design Sources". A red dashed circle highlights this entry.
- Properties**: A message says "Select an object to see properties".
- Project Summary**:
 - Project name: project_1
 - Project location: /home/rty78452/Desktop/project_1/project_1.xpr
 - Product family: Artix-7
 - Project part: Arty (xc7a35ticsg324-1L)
 - Top module name: Not defined
 - Target language: Verilog
 - Simulator language: Mixed
 - Target Simulator: Vivado Simulator
- Board Part**:
 - Display name: Arty
 - Board part name: digilentinc.com:arty:part0:1.1
 - Board revision: C.0
 - Connectors: No connections
 - Repository path: /tools/Xilinx/2025.1/data/xhub/boards
 - URL: www.digilentinc.com/Arty
- Design Runs**:

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



Type in the HDL Source Code

Mat 1

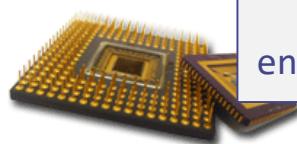
- ◆ The complete code for a 4-bit full adder is as follows:

```
// ----- 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
```

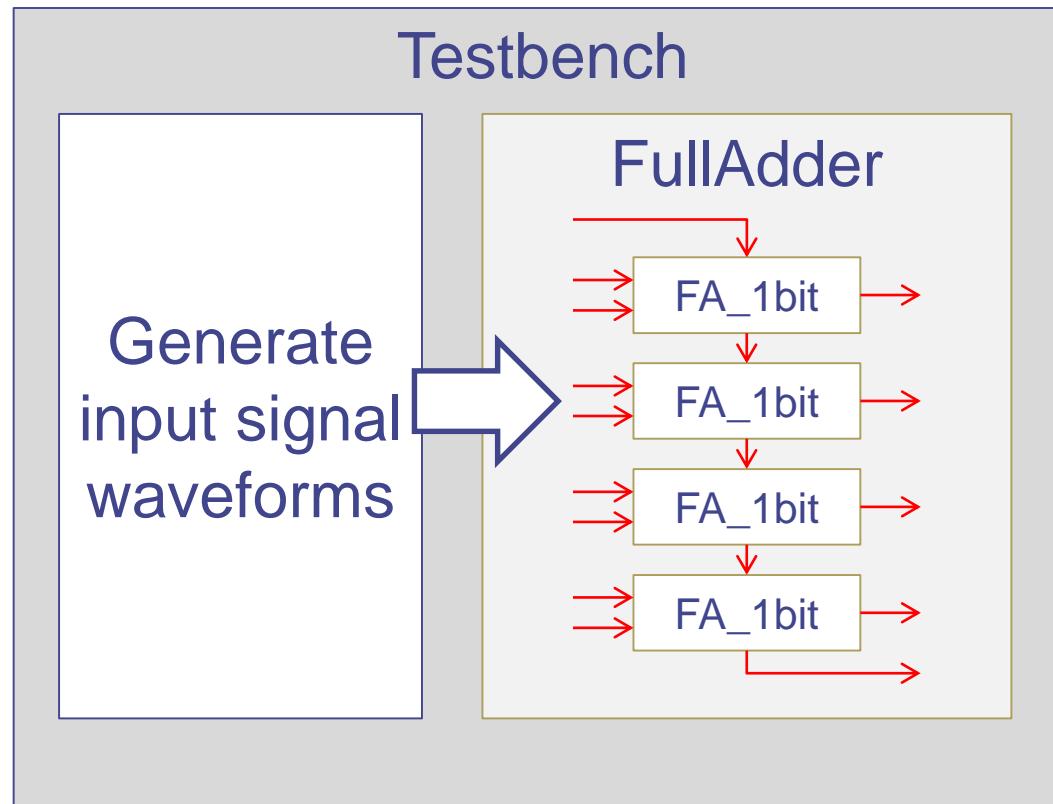




TestBench Design

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.

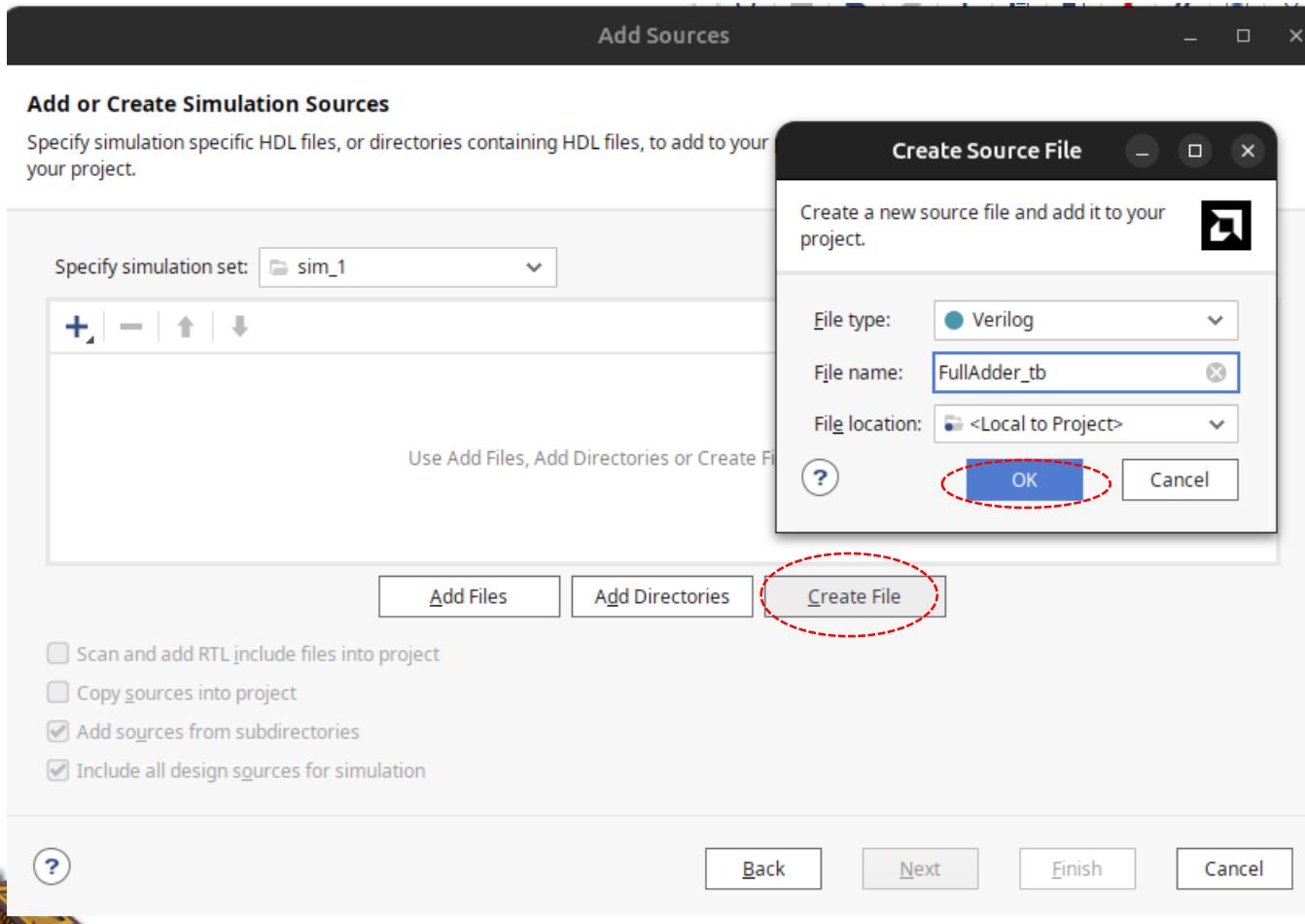




Create the Testbench Source Code

Mat 1

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

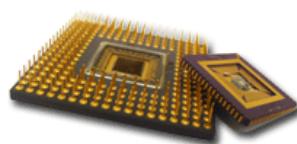
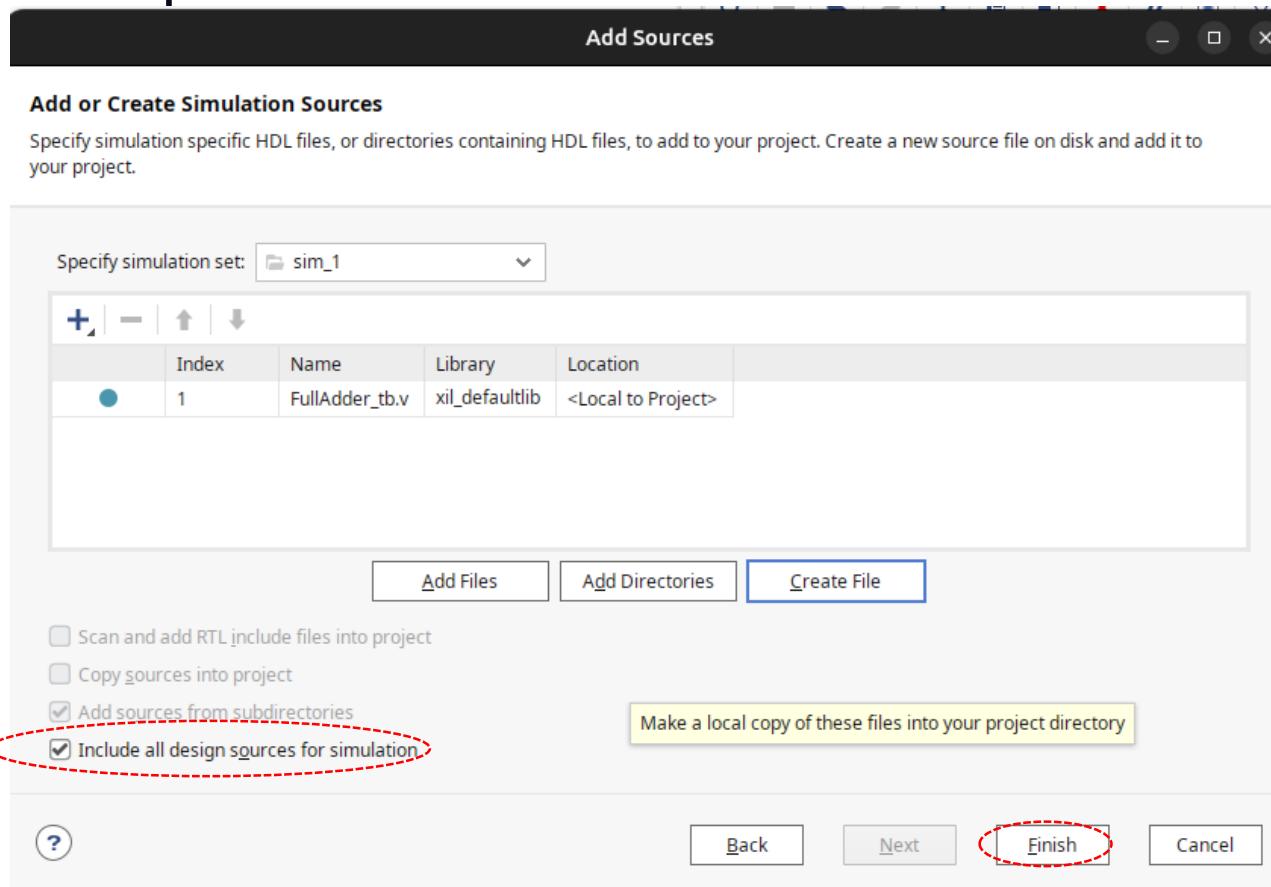




Confirm the Creation of the Testbench

Mat 1

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

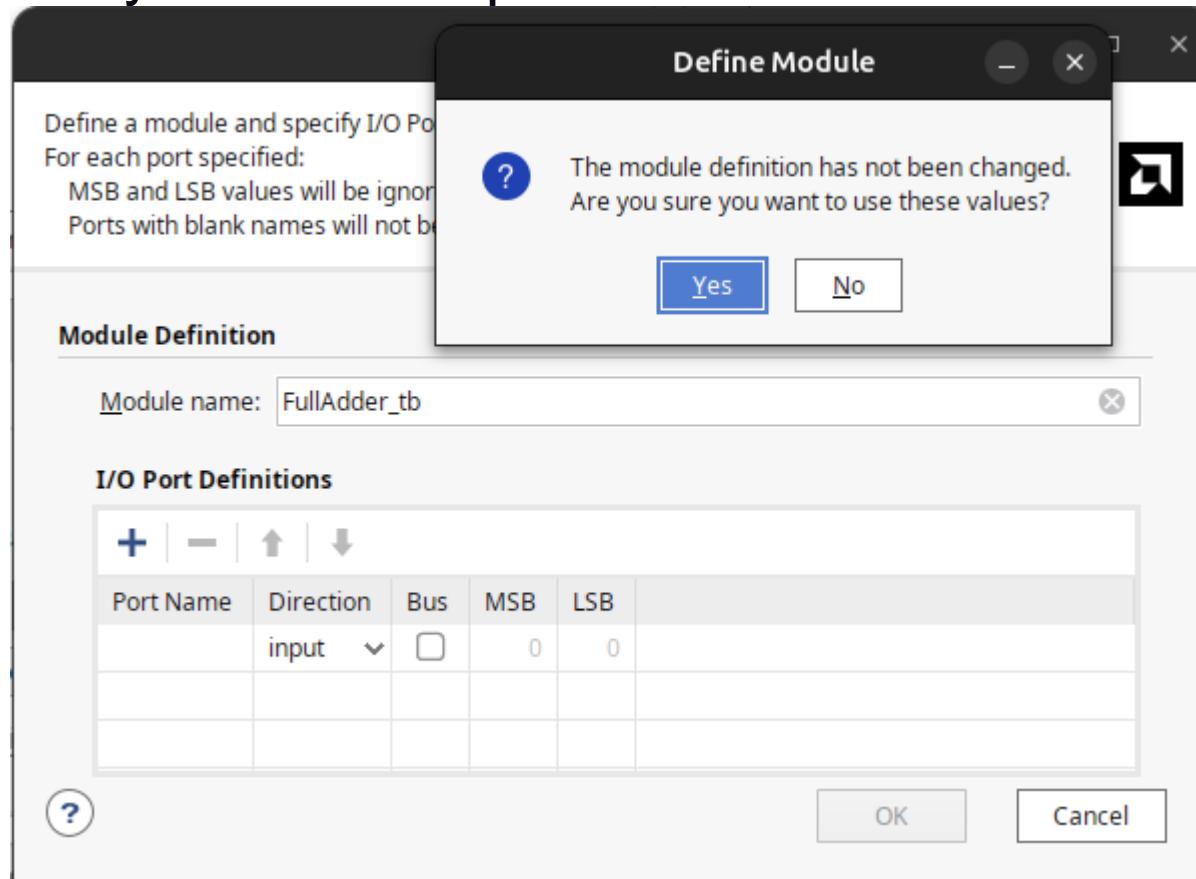




Create the Testbench Template

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

Screenshot of the Vivado 2025.1 Project Manager interface showing a FullAdder design.

PROJECT MANAGER - project_1

Sources

- Design Sources (1)
 - FullAdder (FullAdder.v) (4)
- Constraints
- Simulation Sources (2)
 - sim_1 (2)
 - FullAdder (FullAdder.v) (4)
 - FullAdder_tb (FullAdder_tb.v)
- Utility Sources

Source File Properties

FullAdder.v

Enabled

Location: /home/rty78452/Desktop/project_1/project_1.srcs/sources_1/new

Type: Verilog

Library: xil_defaultlib

Design Runs

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

FullAdder.v

```

15 // Revision:
16 // Revision 0.01 - File Created
17 // Additional Comments:
18 //
19 ///////////////////////////////////////////////////
20
21
22 module FullAdder(A, B, Cin, S, Cout);
23   input [3:0] A, B;
24   input Cin;
25   output [3:0] S;
26   output Cout;
27   wire [2:0] t;
28
29
30 FA_1bit(.A(A[0]), .B(B[0]), .Cin(Cin), .S(S[0]), .Cout(t[0]));
31 FA_1bit(.A(A[1]), .B(B[1]), .Cin(t[0]), .S(S[1]), .Cout(t[1]));
32 FA_1bit(.A(A[2]), .B(B[2]), .Cin(t[1]), .S(S[2]), .Cout(t[2]));
33 FA_1bit(.A(A[3]), .B(B[3]), .Cin(t[2]), .S(S[3]), .Cout(Cout));
34 endmodule
35
36 module FA_1bit(A, B, Cin, S, Cout);
37   input A, B, Cin;
38   output S, Cout;
39
40   assign S = Cin ^ A ^ B;
41   assign Cout = (A&B)|(Cin&B)|(Cin&A);
42 endmodule
43

```



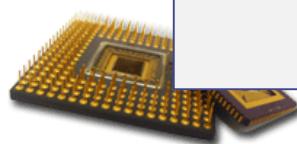
The Sample Testbench Code

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

The screenshot shows the Vivado 2025.1 Project Manager interface. The left sidebar contains various project management, simulation, and analysis tools. In the center, the 'PROJECT MANAGER - project_1' window is open, displaying the 'Sources' section with files like 'FullAdder.v' and 'FullAdder_tb.v'. Below it is the 'Source File Properties' window for 'FullAdder.v'. On the right, the 'Project Summary' tab for 'FullAdder.v' is active, showing details such as Project name: project_1, Project location: /home/rty78452/Desktop/project_1/project_1.xpr, and Target Simulator: Vivado Simulator. At the bottom, the 'Design Runs' table shows two entries: 'synth_1' and 'impl_1', both in the 'Not started' status. A red circle highlights the 'Run Behavioral Simulation' option in the context menu, with the word 'Click!' overlaid in red text.

project_1 - [/home/rty78452/Desktop/project_1/project_1.xpr] - Vivado 2025.1

File Edit Flow Tools Reports Window Layout View Help Quick Access

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

- Generate Bitstream
- Open Hardware Manager

PROJECT MANAGER - project_1

Sources

- Design Sources (1)
 - FullAdder (FullAdder.v) (4)
- Constraints
- Simulation Sources (2)
 - sim_1 (2)
 - FullAdder (FullAdder.v) (4)
 - FullAdder_tb (FullAdder_tb.v)
- Utility Sources

Hierarchy Libraries Compile Order

Source File Properties

Run Behavioral Simulation Click!

Run Post-Synthesis Functional Simulation

Run Post-Synthesis Timing Simulation

Run Post-Implementation Functional Simulation

Run Post-Implementation Timing Simulation

Library: xil_defaultlib

Size: 0.5 KB

General Properties

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 DSP

Project Summary FullAdder.v FullAdder_tb.v

Overview Dashboard

Settings Edit

Project name: project_1

Project location: /home/rty78452/Desktop/project_1/project_1.xpr

Product family: Artix-7

Project part: Artix (xc7a35ticsg324-1L)

Top module name: FullAdder

Target language: Verilog

Simulator language: Mixed

Target Simulator: Vivado Simulator

Board Part

Display name: Artix

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

Board revision: C.0

Connectors: No connections

Repository path: /tools/Xilinx/2025.1/data/xhub/boards

URL: www.digilentinc.com/Arty

Board overview Artix



Vivado Simulator Window

Mat 1

project_1 - [/home/rty78452/Desktop/project_1/project_1.xpr] - Vivado 2025.1

Flow Navigator SIMULATION - Behavioral Simulation - Functional - sim_1 - FullAdder_tb

File Edit Flow Tools Reports Window Layout View Run Help

Quick Access

Simulation time duration

Zoom waveform to fit window

Scope Sources Objects Protocol Ins Untitled 1

FullAdder.v FullAdder_tb.v Untitled 1

FullAdder_tb FullAdder_tb Verilog M

A[3:0] 6 Array
B[3:0] 1 Array
Cin 1 Logic
S[3:0] 8 Array
Cout 0 Logic

1,000,000 ns

0.000 ns 500.000 ns

0 5 f 0 6
0 a 1 f 1
Cin
8 0 f 0 8
Cout

Tcl Console Messages Log

```
# } # run 1000ns
INFO: [USF-XSim-96] XSim completed. Design snapshot 'FullAdder_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
```

Type a Tcl command here

Sim Time: 1 us

