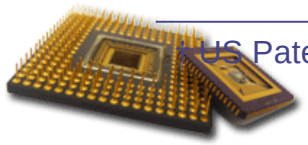
Introduction to Vivado

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

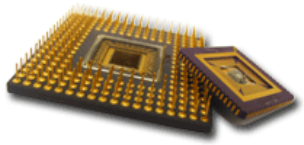
- Digital circuits can be implemented in different ways.
  - Circuits Boards
    - Circuit board design using standard IC parts (e.g., 74SLxx)
  - Application Specific ICs
    - Full-custom and Cell-based IC designs
  - Programmable logics
    - Field Programmable Gate Array (FPGA) design
- Here, we 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

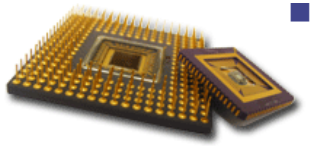
Xilinx Vivado Design Suite

- 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.



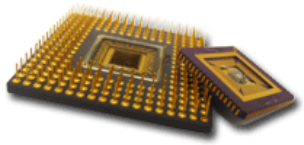
Vivado Circuit Implementation Flow

- 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

- 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.

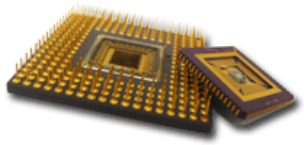


Install Your Own Vivado Design Suite

- Install a copy of Vivado ML Edition - 2025.1 Full Product Installation onto your computer.
  - You can download it from:

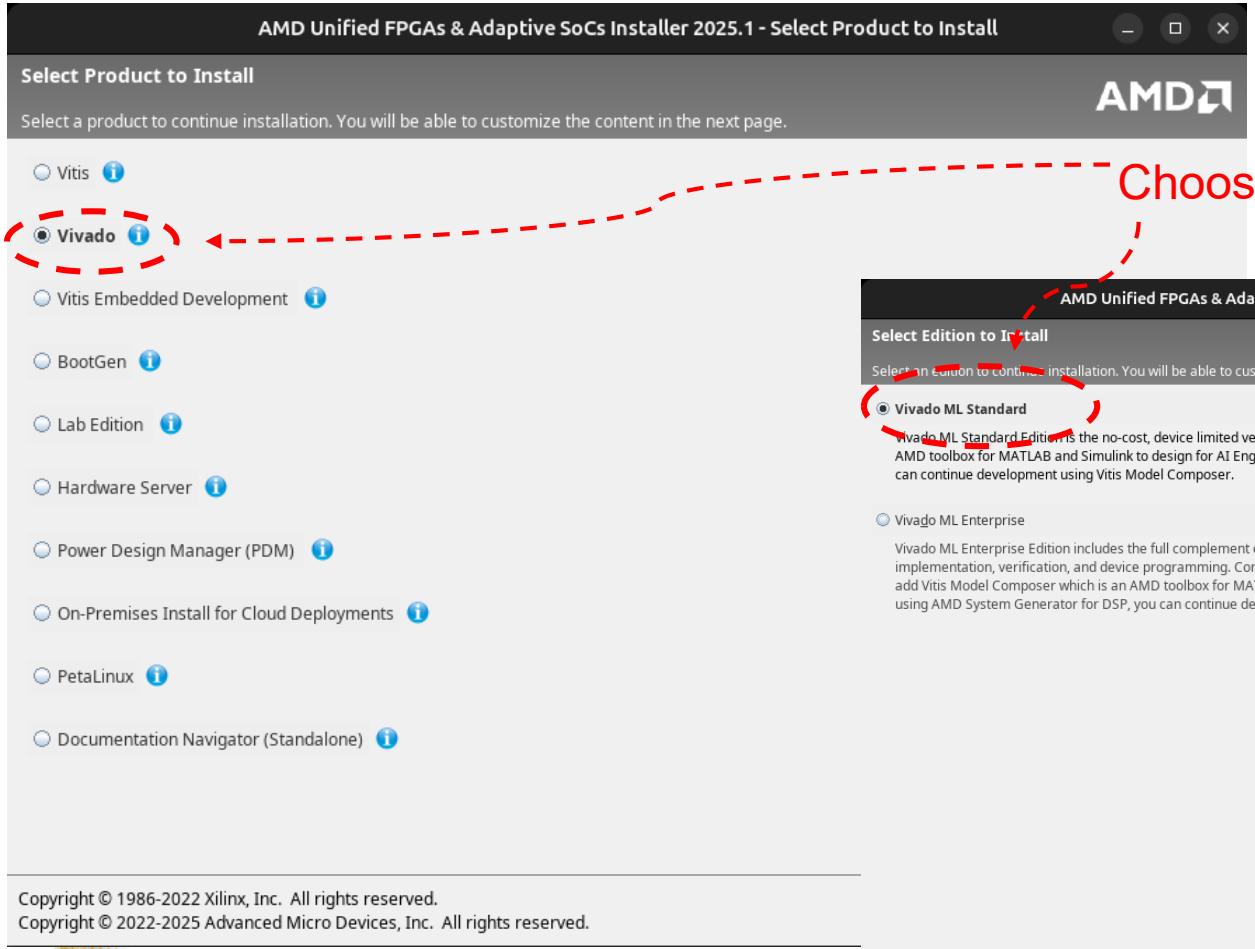
    https://www.Xilinx.com/support/download.html

- The installation requires about 82 GB of disk space.
  - Please install the “Web Installer” version and register online for a free license.

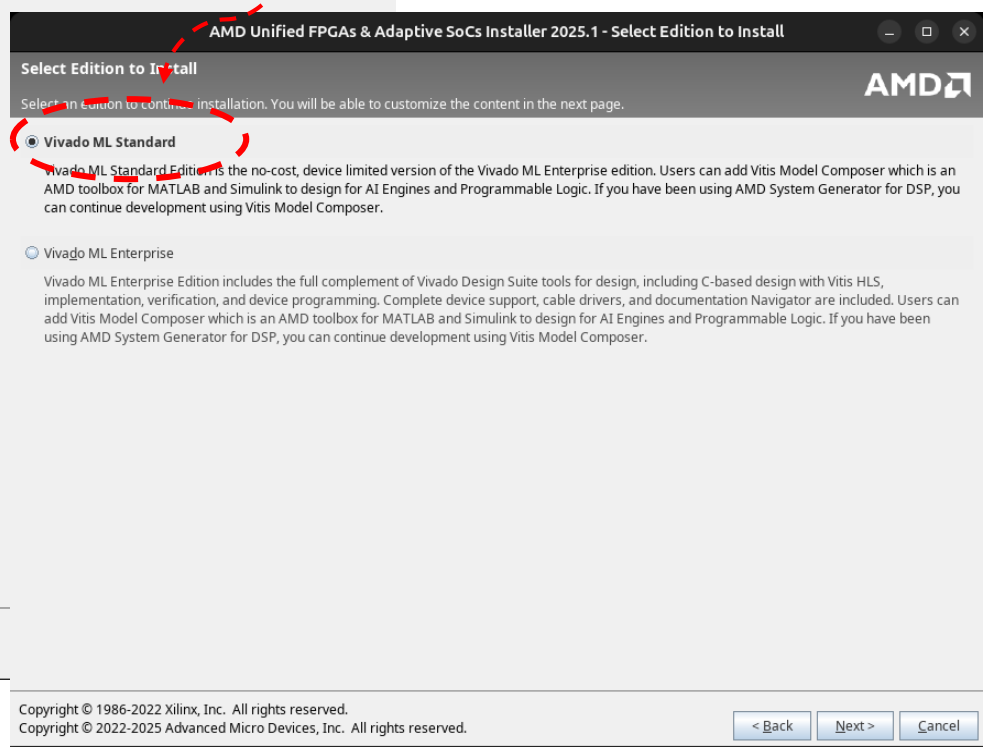


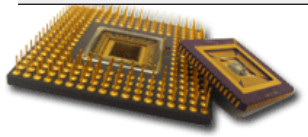
Vivado Installation Guide (1/2)

- You must select the right version upon installation:



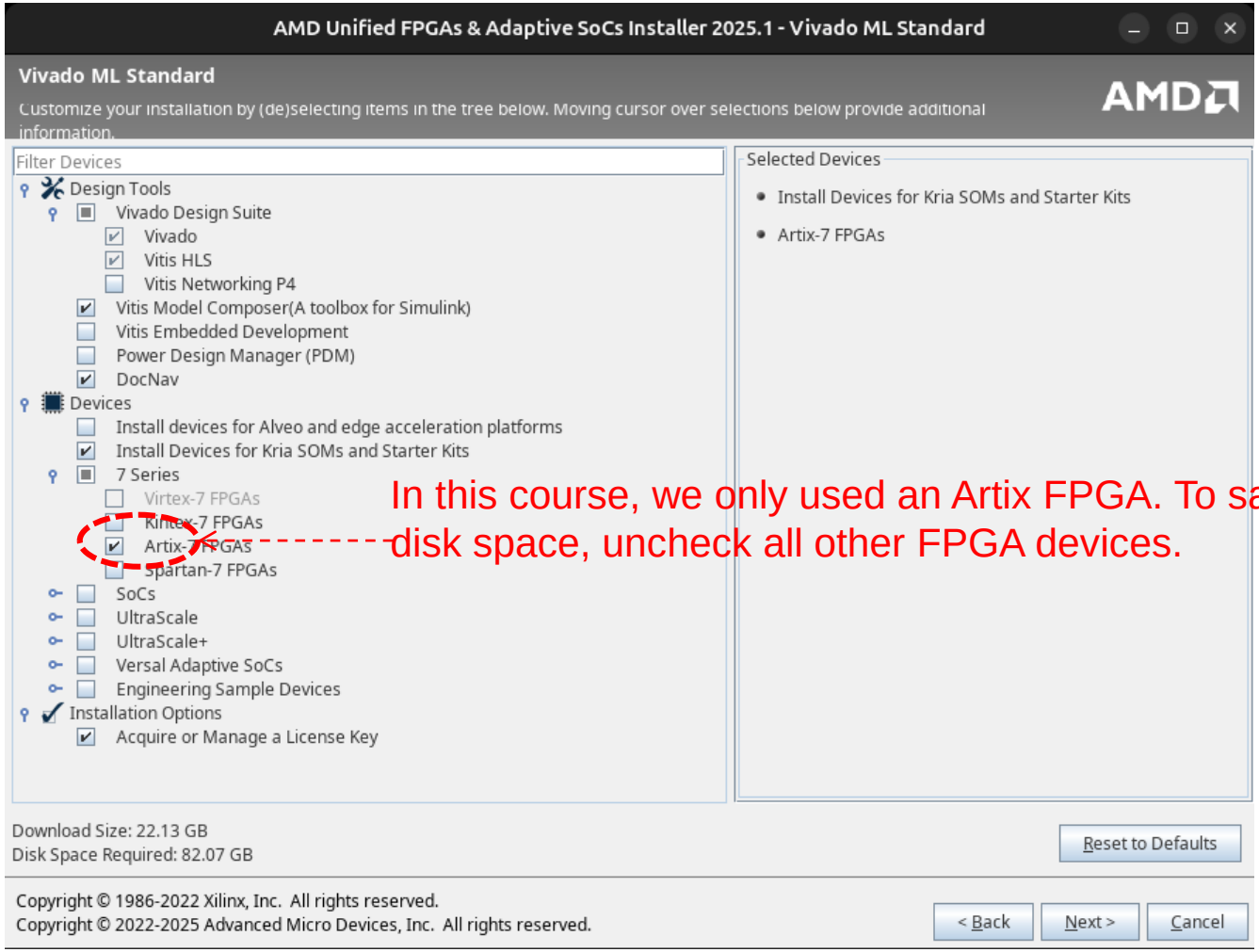
Choose "Vivado" and "ML Standard"

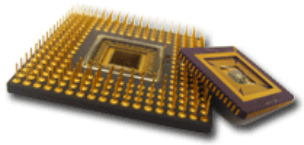




Vivado Installation Guide (2/2)

- Selecting the required packages and FPGA devices:

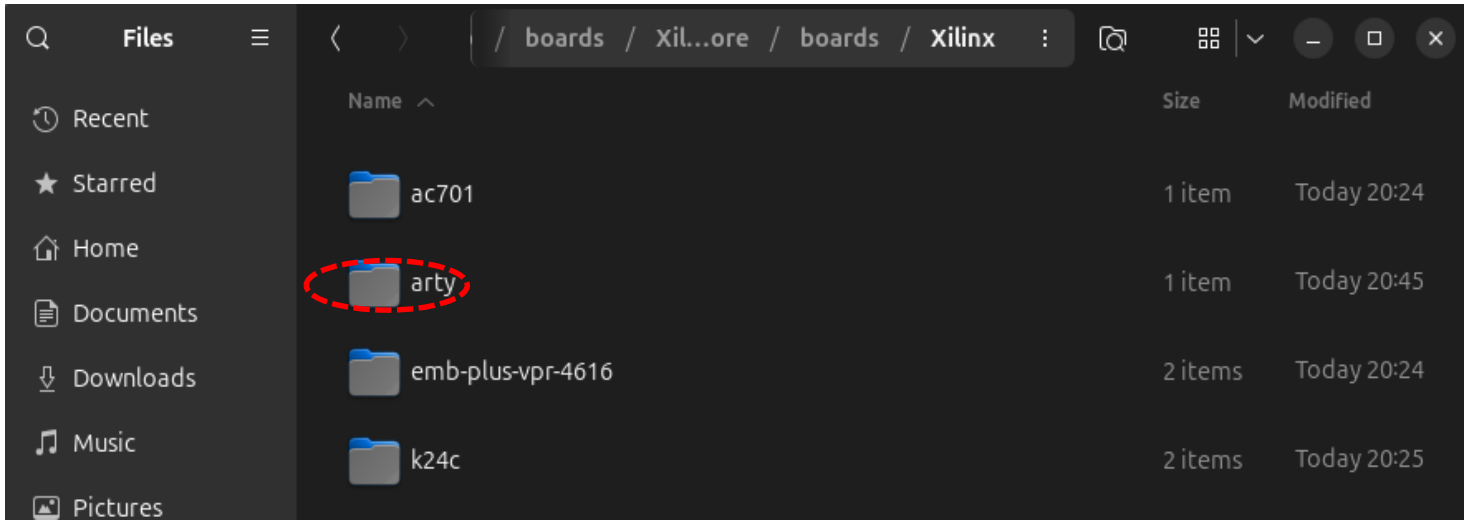


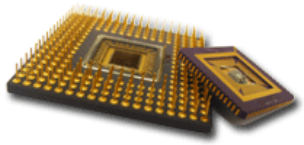


Installation of Arty Board Definitions

- 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:

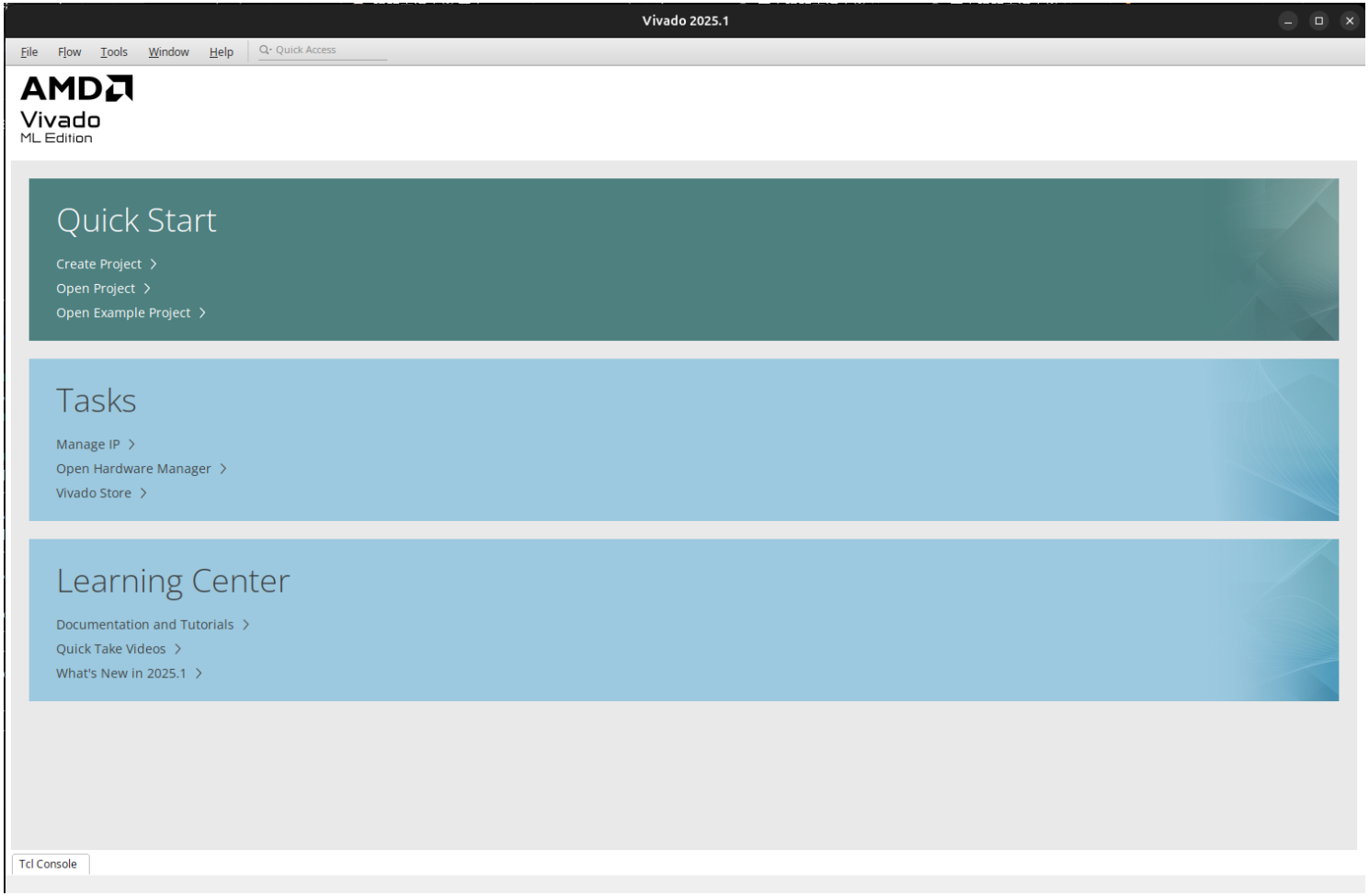
C:/<INST_DIR>/Vivado/2025.1/data/xhub/boards/XilinxBoardStore/boards/Xilinx

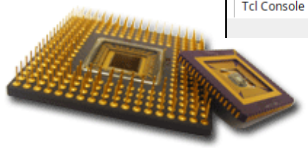




Launch Vivado 2025.1

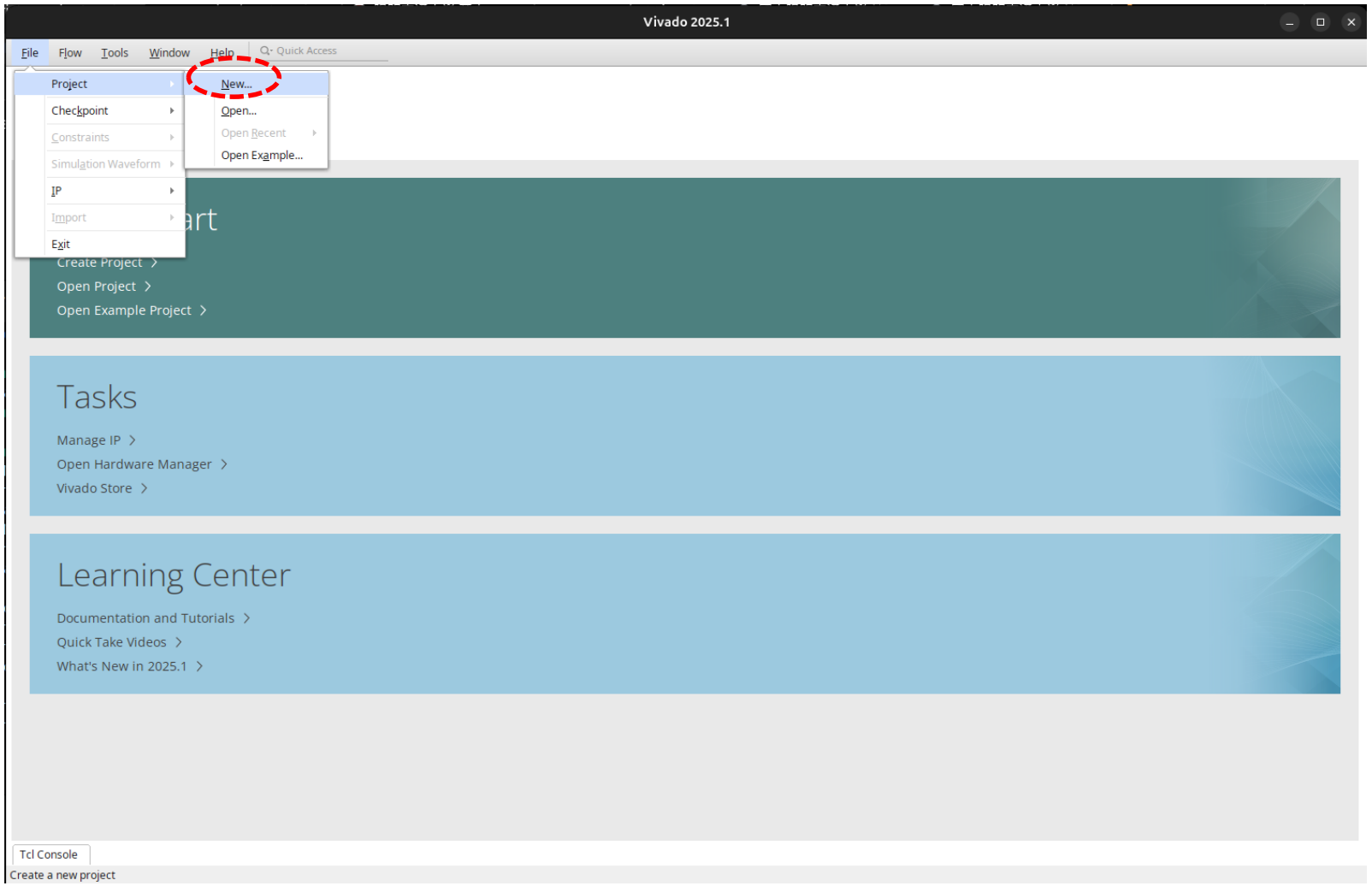
- Double-click the Vivado 2025.1 icon on the desktop:

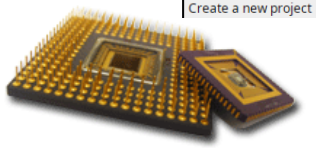




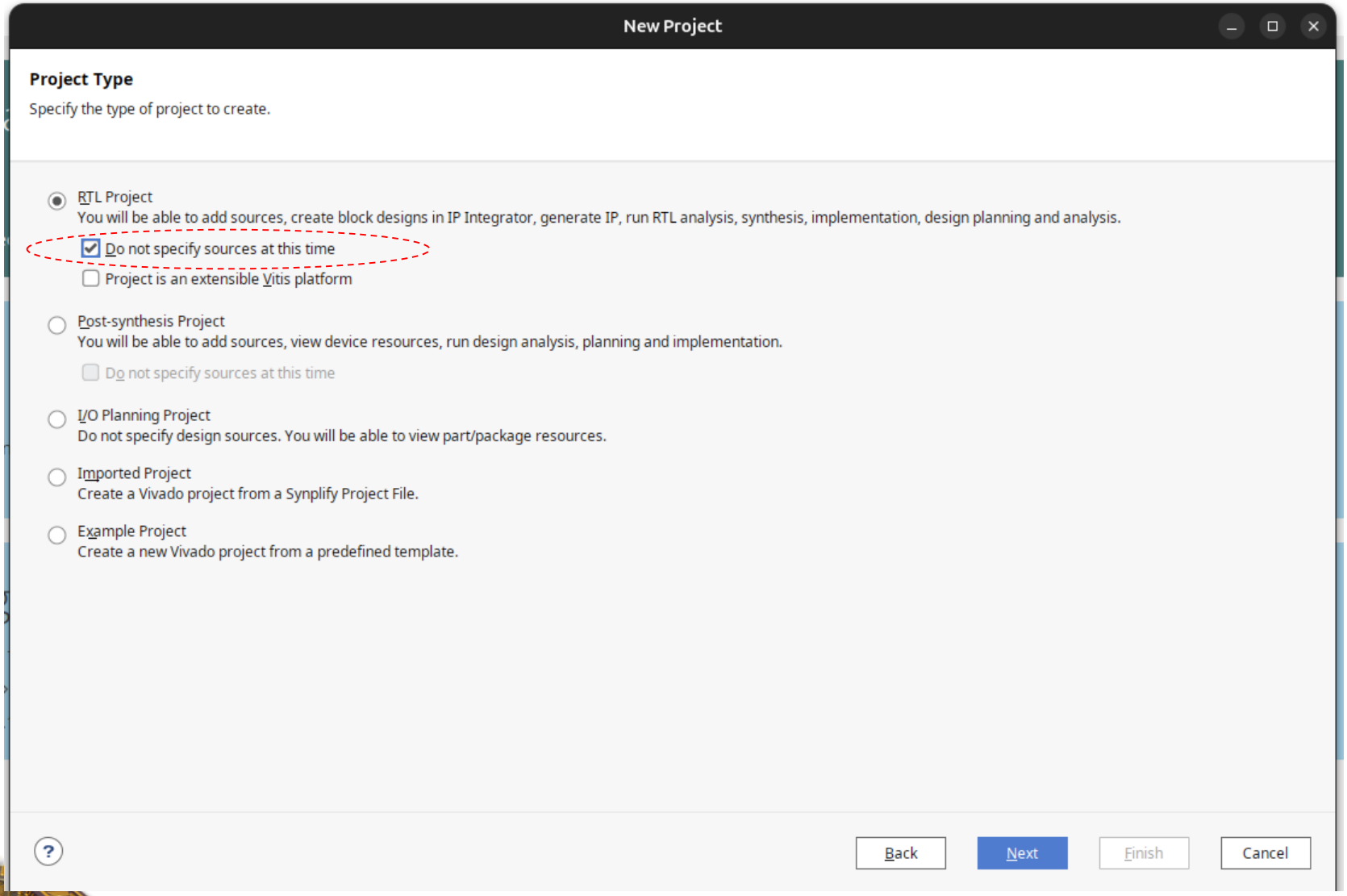
Create a New Project in Vivado

Mat 1

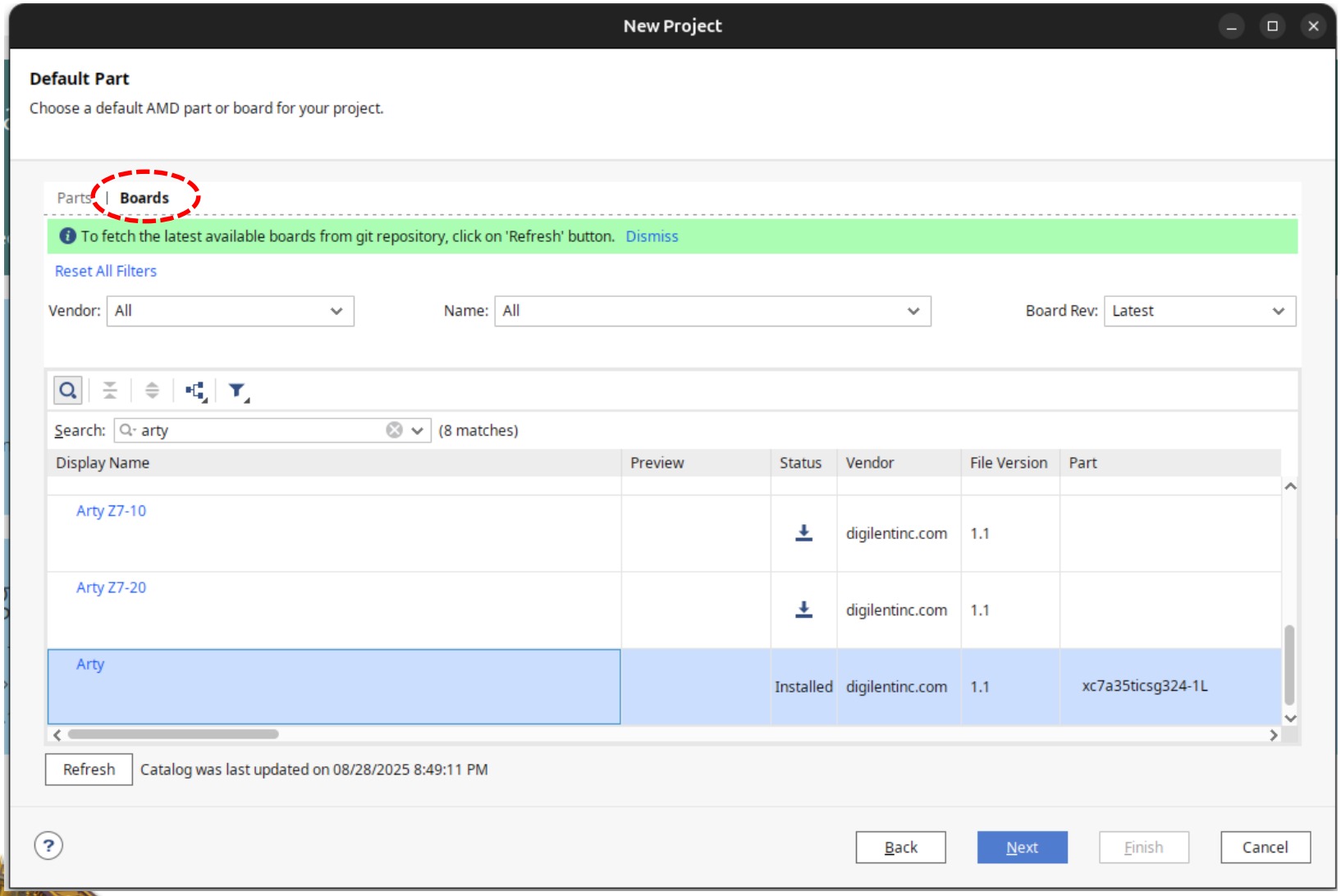




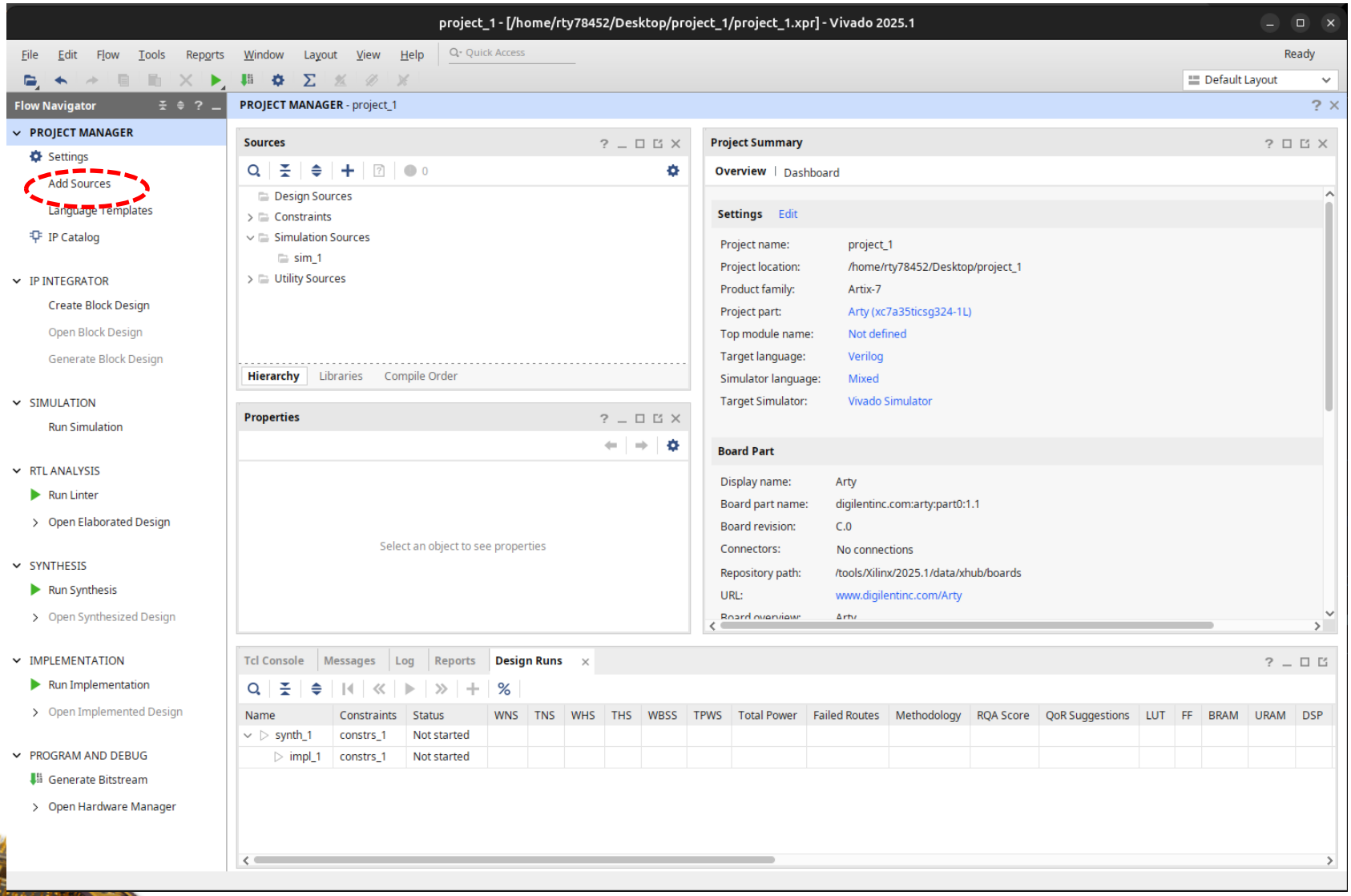
Select Project Type



Select the Target FPGA Board

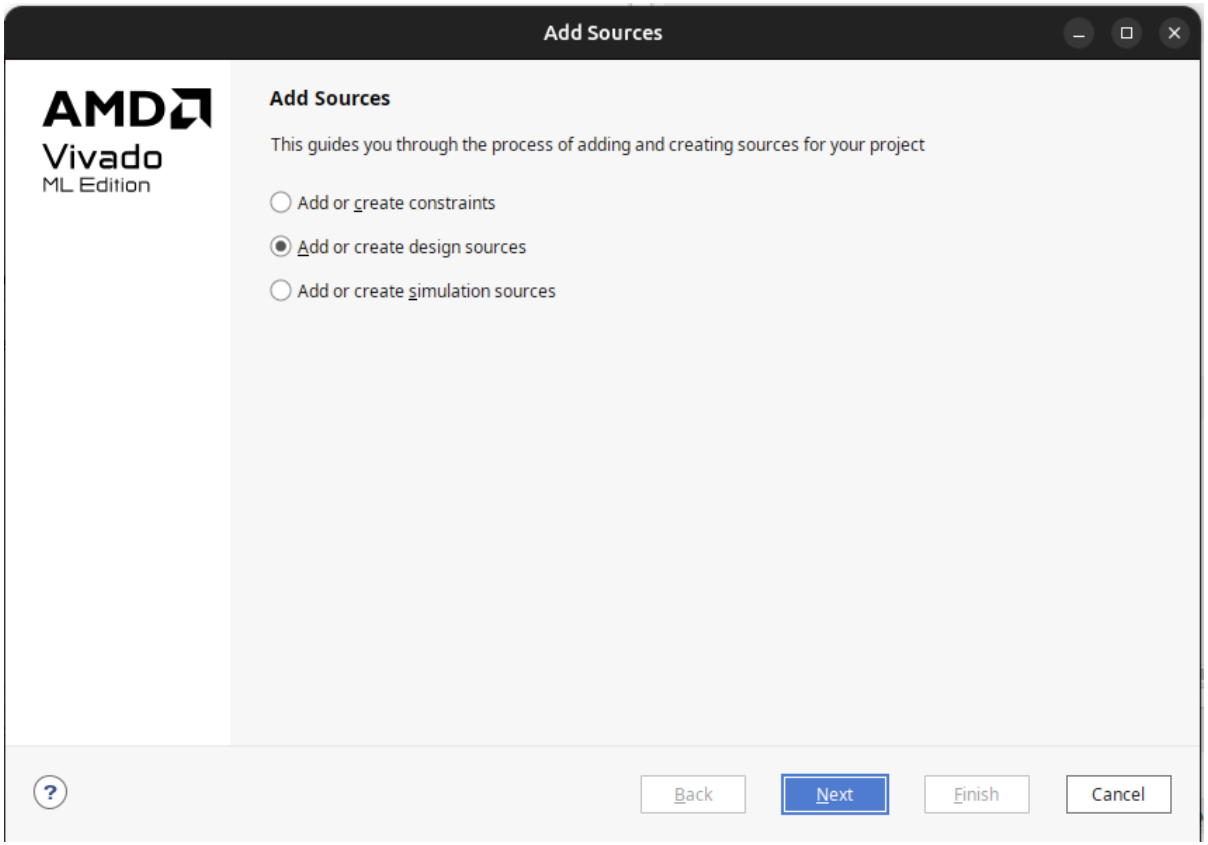


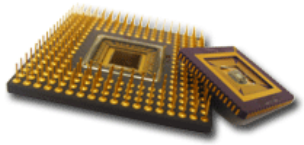
Add a New HDL Source Code



Specifying the Source Type to Create

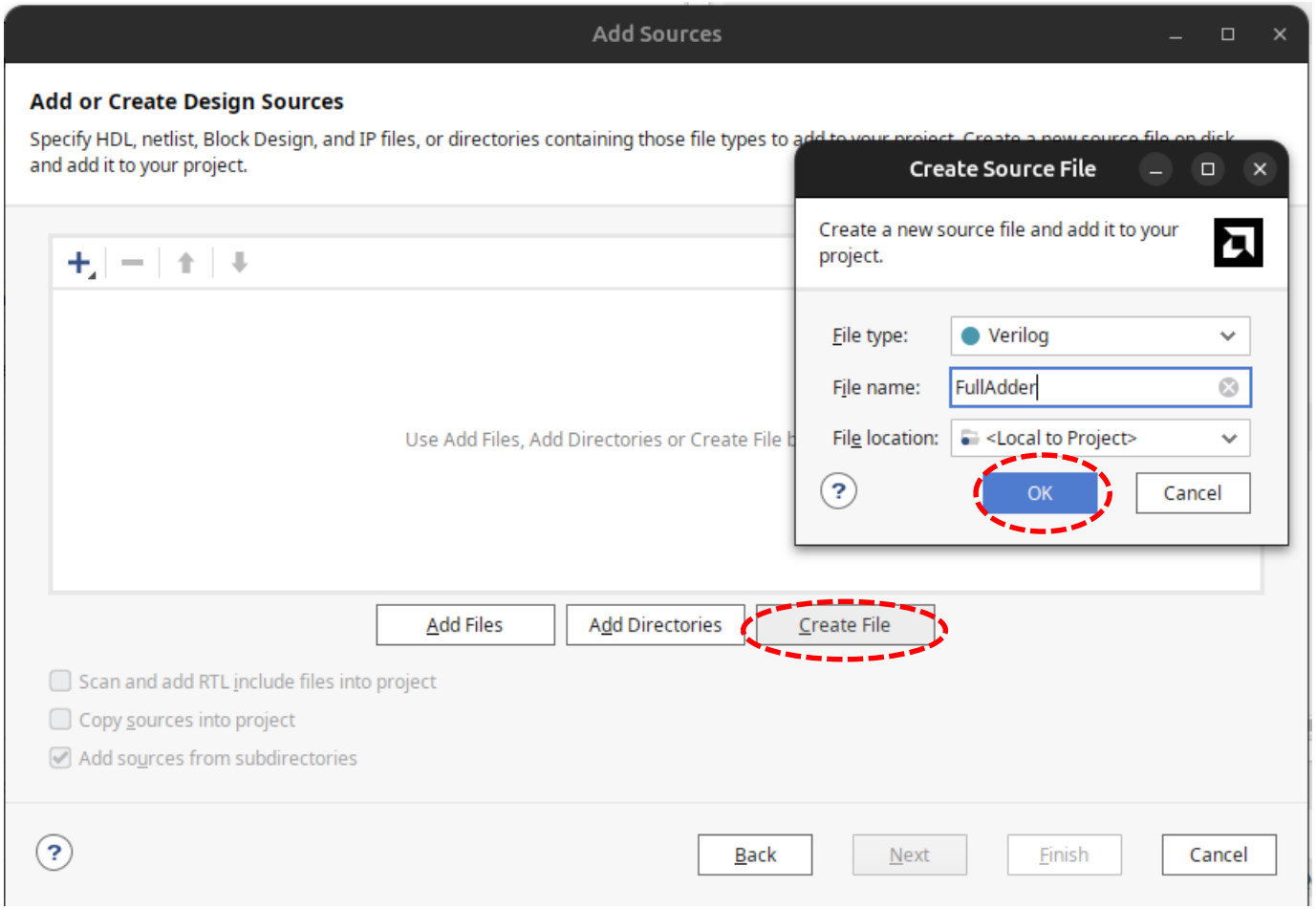
- There are several types of source files in a circuit design project: design sources, constraint sources, simulation sources, and memory sources, etc.

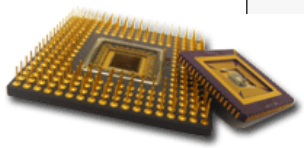




Create a 4-bit Full Adder Design

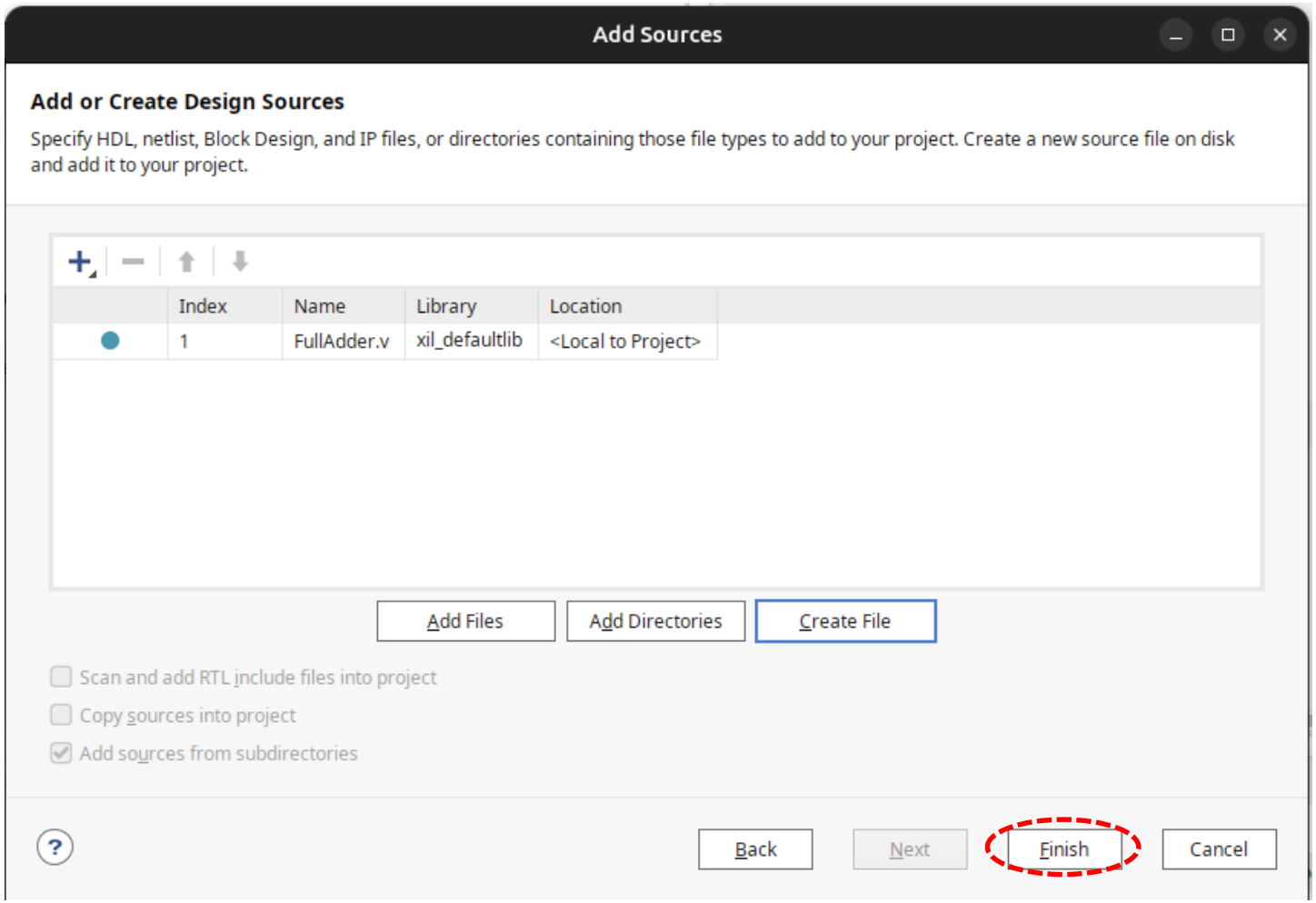
- Let's create a design source from scratch!

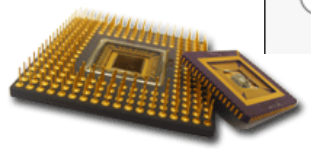




Confirm to Create the Verilog Module

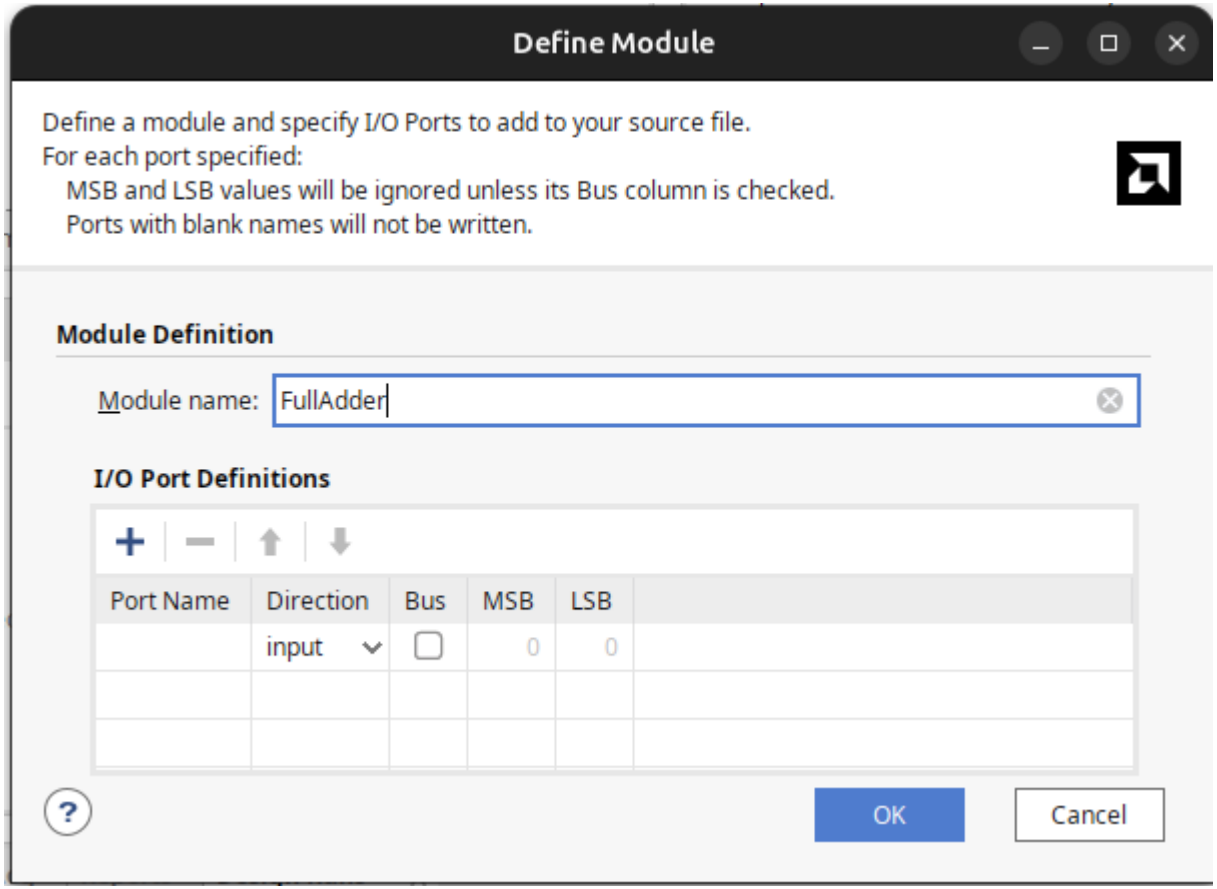
- Let's create a design source from scratch!

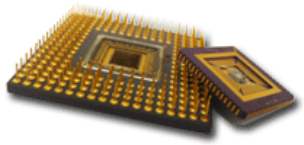




Confirm to Create the Verilog Module

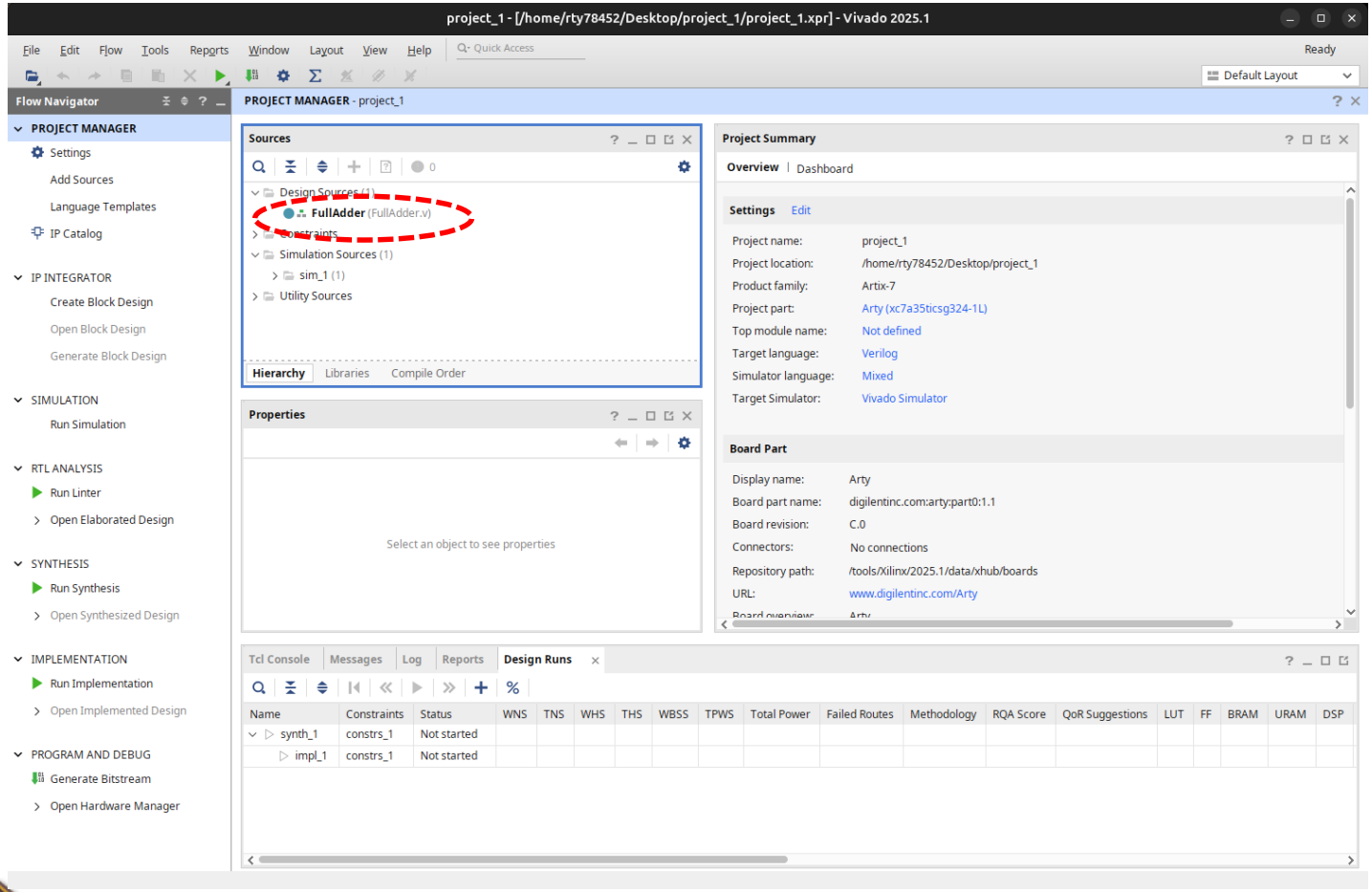
- You can define your ports here or do it in the HDL code:

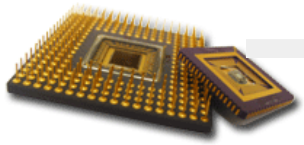




An Empty Verilog Template is Created

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

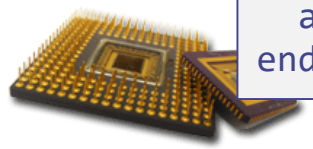




Type in the HDL Source Code

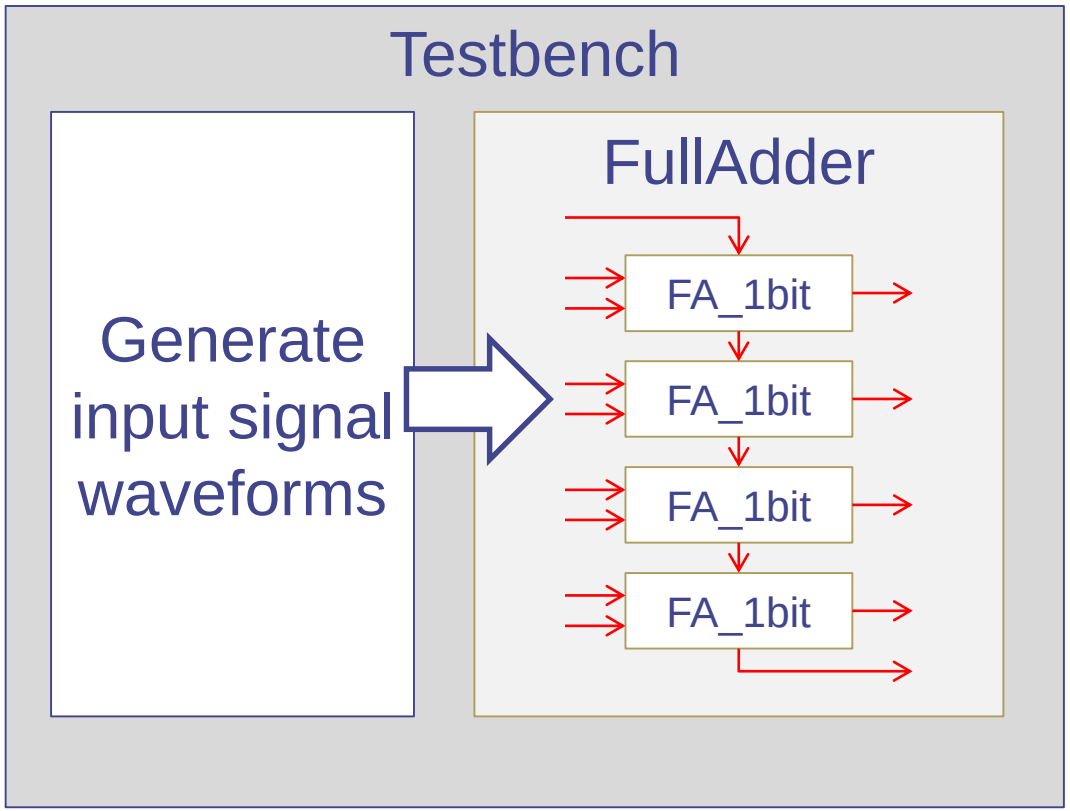
- The complete code for a 4-bit full adder is as follows:

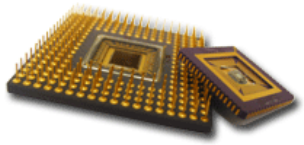
```
// ------------- A four-bit full adder ----------------------------
module FullAdder(A, B, Cin, S, Cout);
  input [3:0] A, B;
  input Cin;
  output [3:0] S;
  output Cout;
  wire [2:0] t;

  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

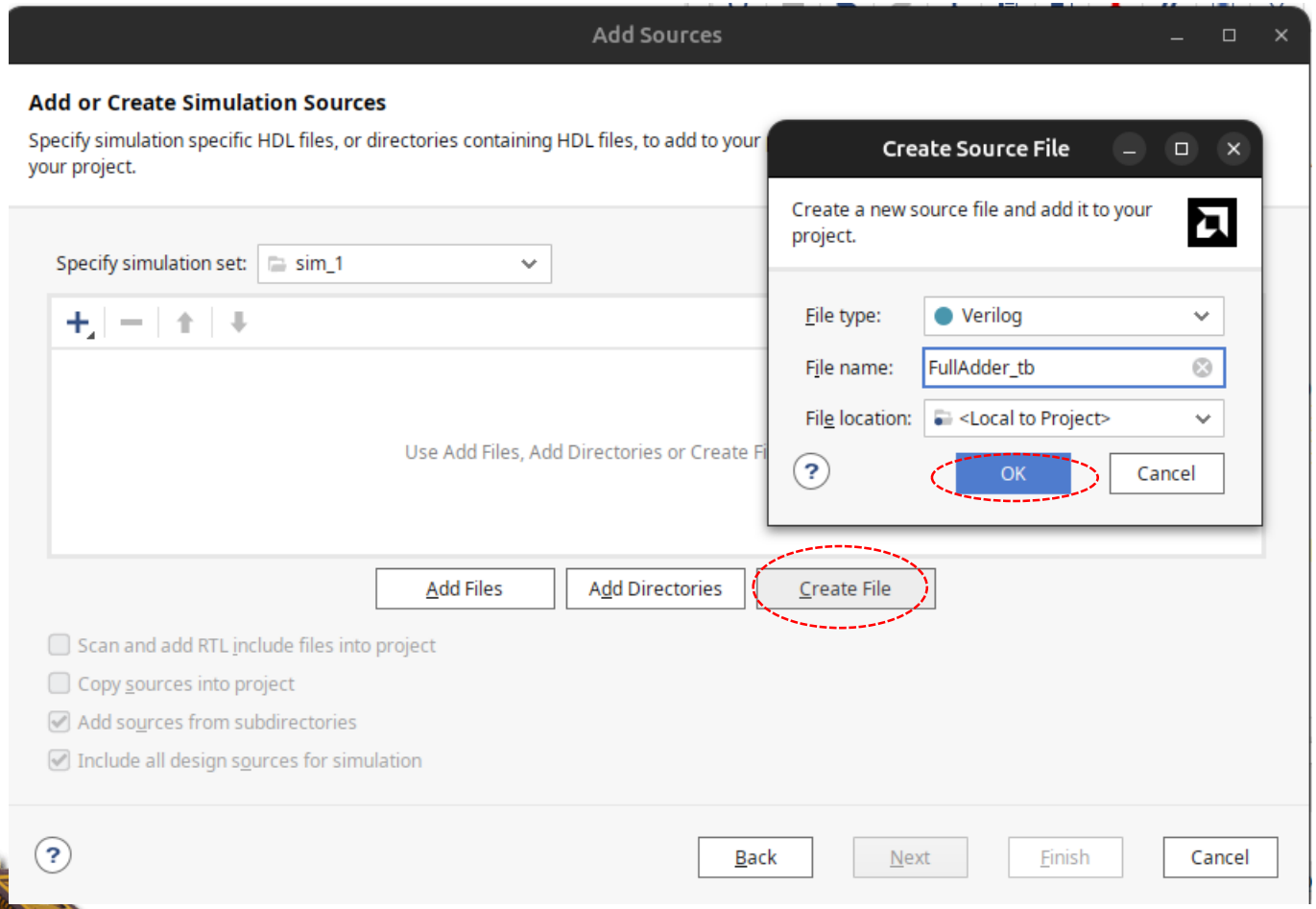
- You must create a testbench to generate input signals that can feed into your circuit module, such that you can analyze the output to verify its correctness.





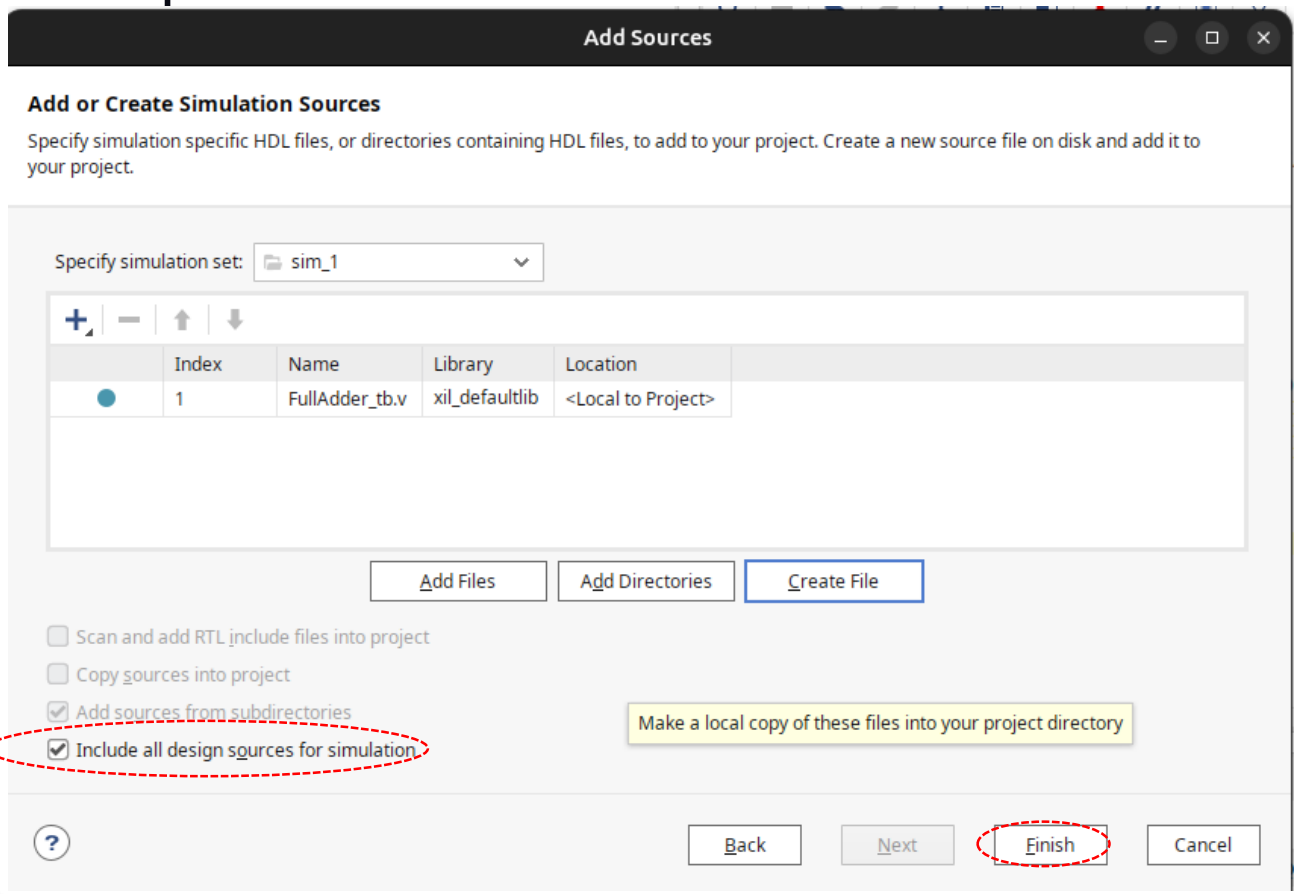
Create the Testbench Source Code

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



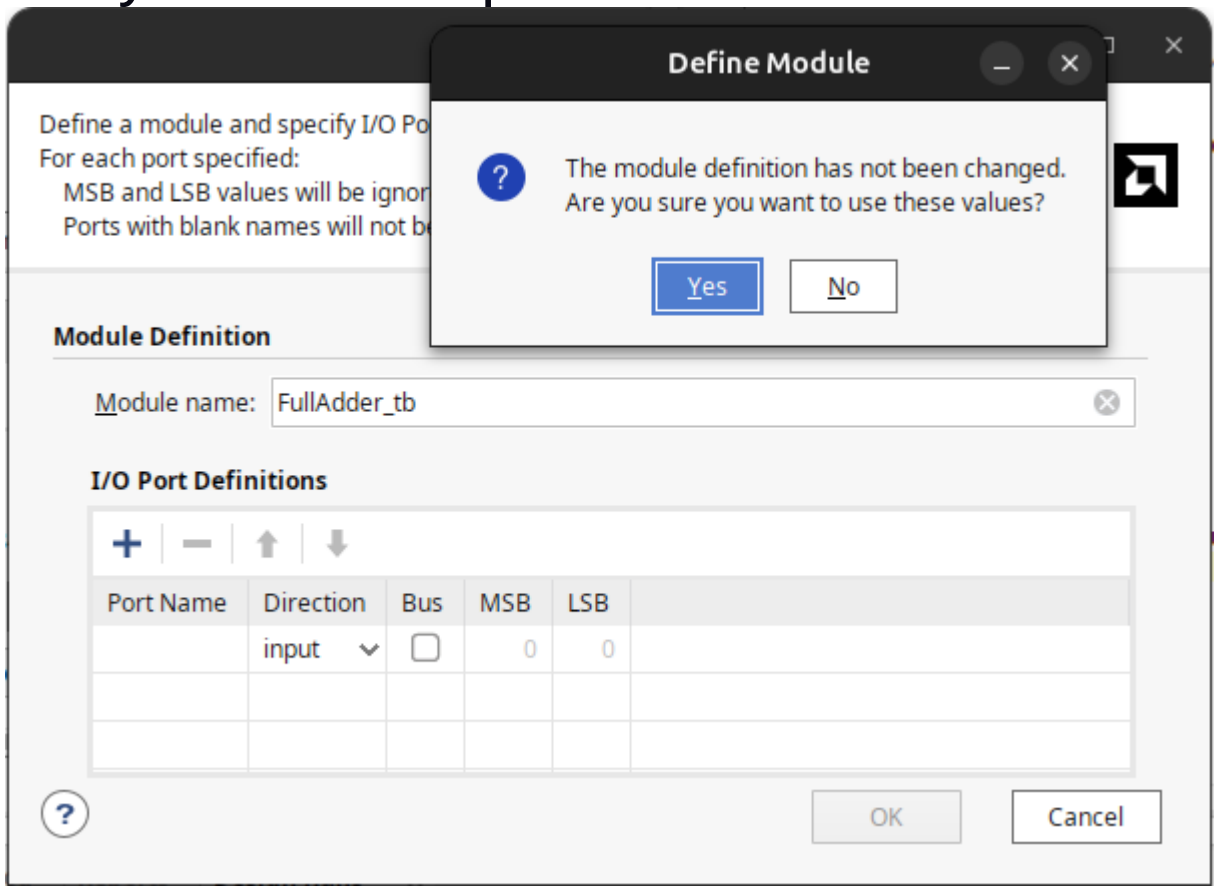
Confirm the Creation of the Testbench

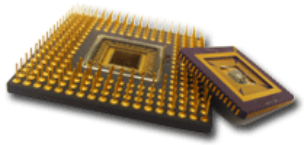
- Here, we include the design sources into the simulation set so that we can test the modules under development.



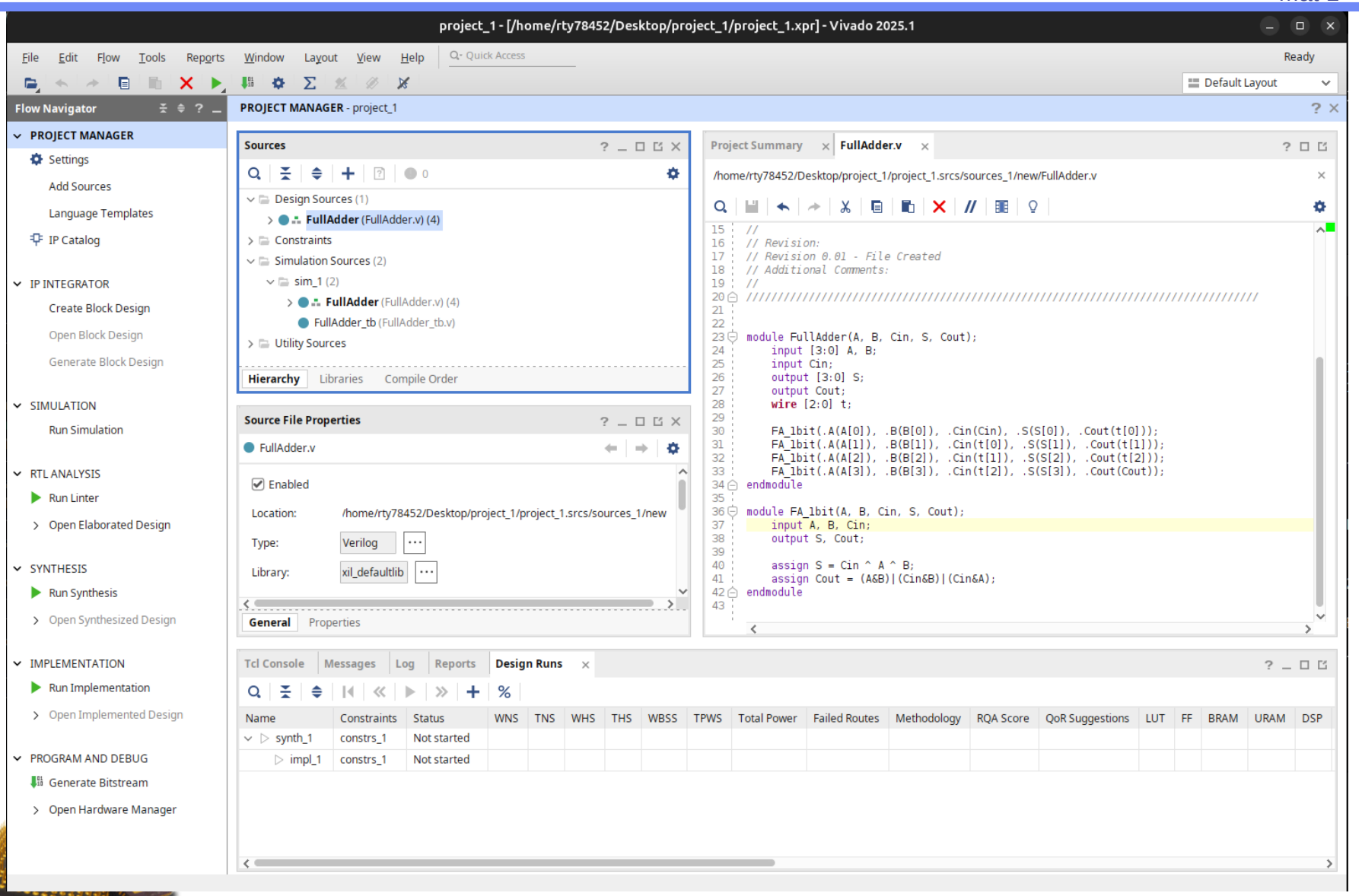
Create the Testbench Template

- 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

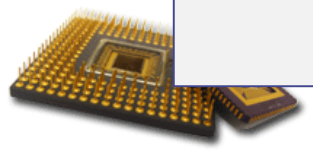


The Sample Testbench Code

- 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

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

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

Default Layout

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 Elabora

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

78452/Desktop/project_1/project_1.srcs/sim_1/new

Library: xil_defaultlib

Size: 0.5 KB

General Properties

Project Summary | FullAdder.v | FullAdder_tb.v *

Overview | Dashboard

Settings Edit

Project name:	project_1
Project location:	/home/rty78452/Desktop/project_1
Product family:	Artix-7
Project part:	Arty (xc7a35ticsg324-1L)
Top module name:	FullAdder
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

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
synth_1	constrs_1	Not started																
impl_1	constrs_1	Not started																

Vivado Simulator Window