



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, 2024

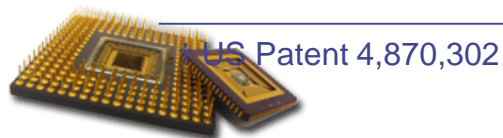


Target Technology of Digital Labs

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†.



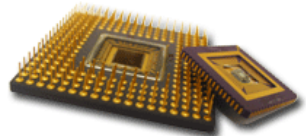
US Patent 4,870,302



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

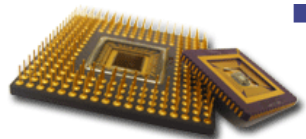
- Convert from the HDL programs or schematics to a netlist file that define a list of circuit blocks and how they are connected

◆ 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.

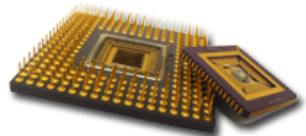




Vivado Circuit Debug Flow

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

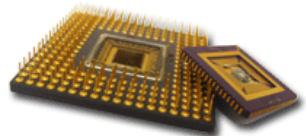
Mat 1

- ◆ Install a copy of Vivado ML Edition - 2024.1 Full Product Installation onto your computer.

- You can download it from:

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

- ◆ The installation requires about 60 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:

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. There is an option to install Power Design Manager for power estimation of Versal, UltraScale+, and Kria products.

☒ Vivado

Includes the full complement of Vivado Design Suite tools for design, including C-based design with Vitis High-Level programming. Complete device support, cable driver, and Document Navigator included. Users can also install Vitis Programmable Logic in MATLAB and Simulink. Users can select to install the Vitis Embedded Development which option to install Power Design Manager for power estimation of Versal, UltraScale+, and Kria products.

☐ Vitis Embedded Development

The Vitis Embedded Development is a standalone embedded software development package for creating, building, and applications for AMD FPGA processors. It includes a new Vitis IDE with its new backend Vitis Server, as well as the and program_flash.

☐ 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 A

☐ Hardware Server

Installs hardware server and JTAG cable drivers for remote debugging.

☐ Power Design Manager (PDM)

Copyright © 1986-2022 Xilinx, Inc. All rights reserved.
Copyright © 2022-2024 Advanced Micro Devices, Inc. All rights reserved.



Choose "Vivado" and "ML Standard"

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. Users can select to install the Vitis Embedded Development which is an embedded software development package. If you have been using AMD System Generator for DSP, you can continue development using Vitis Model Composer. There is an option to install Power Design Manager for power estimation of Versal, UltraScale+, and Kria products.

☐ 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. Users can select to install the Vitis Embedded Development which is an embedded software development package. If you have been using AMD System Generator for DSP, you can continue development using Vitis Model Composer. There is an option to install Power Design Manager for power estimation of Versal, UltraScale+, and Kria products.

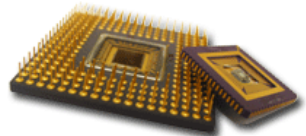
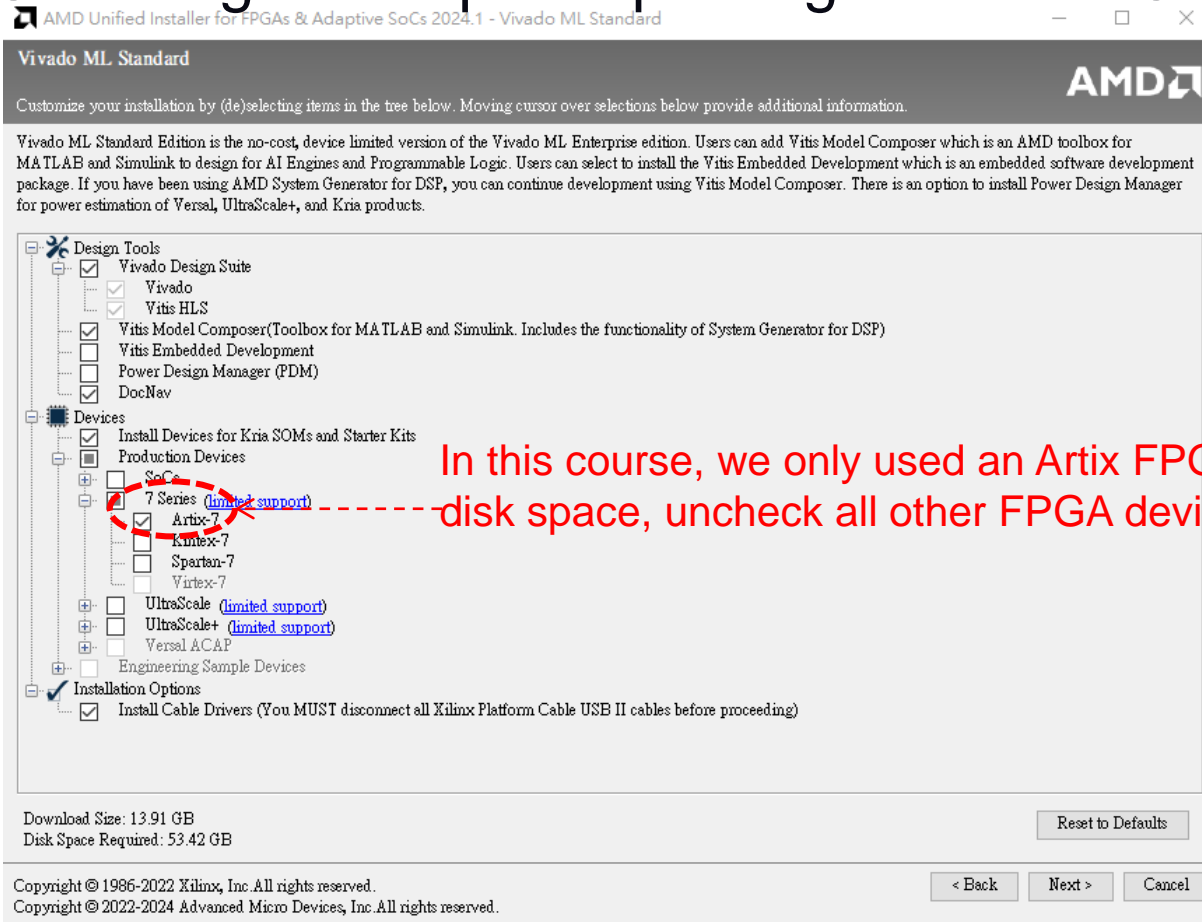
Copyright © 1986-2022 Xilinx, Inc. All rights reserved.
Copyright © 2022-2024 Advanced Micro Devices, Inc. All rights reserved.



Vivado Installation Guide (2/2)

Mat 1

◆ Selecting the required packages and FPGA devices:





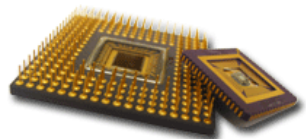
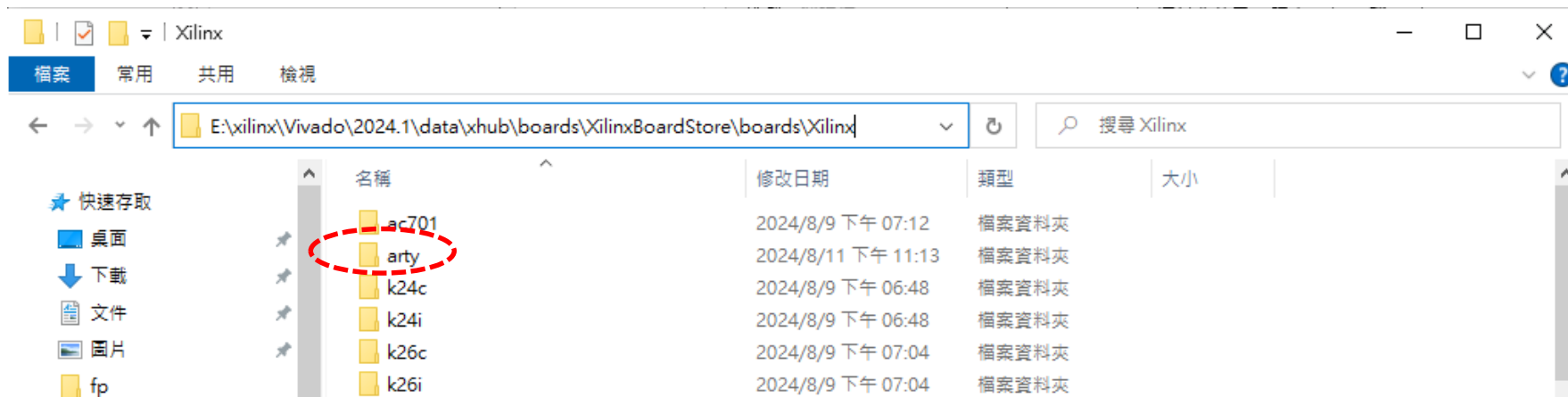
Installation of Arty Board Definitions

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/2024.1/data/xhub/boards/
XilinxBoardStore/boards/Xilinx

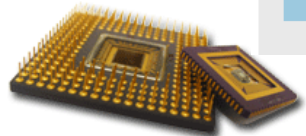
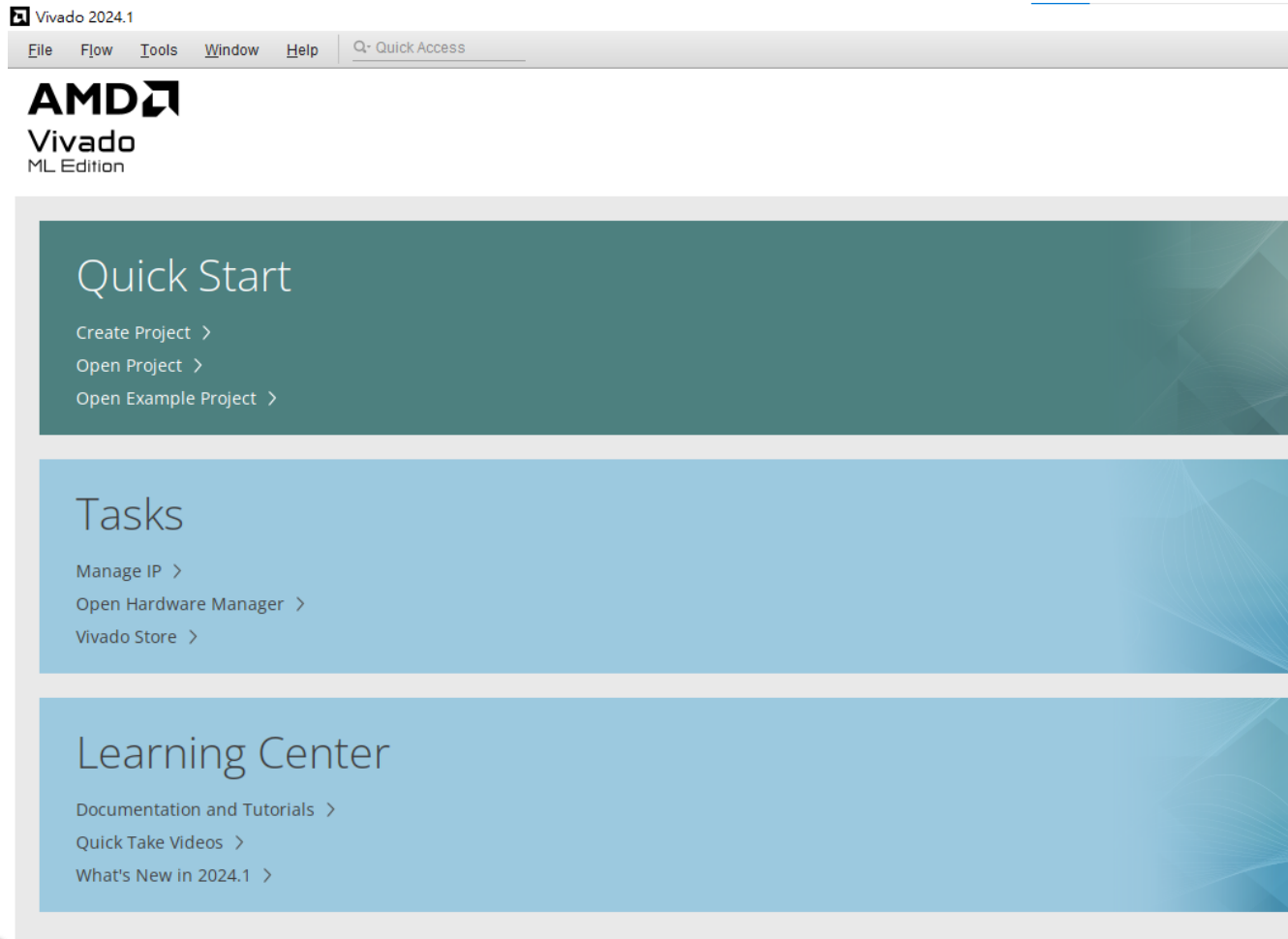




Launch Vivado 2024.1

Mat 1

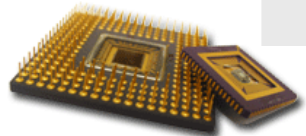
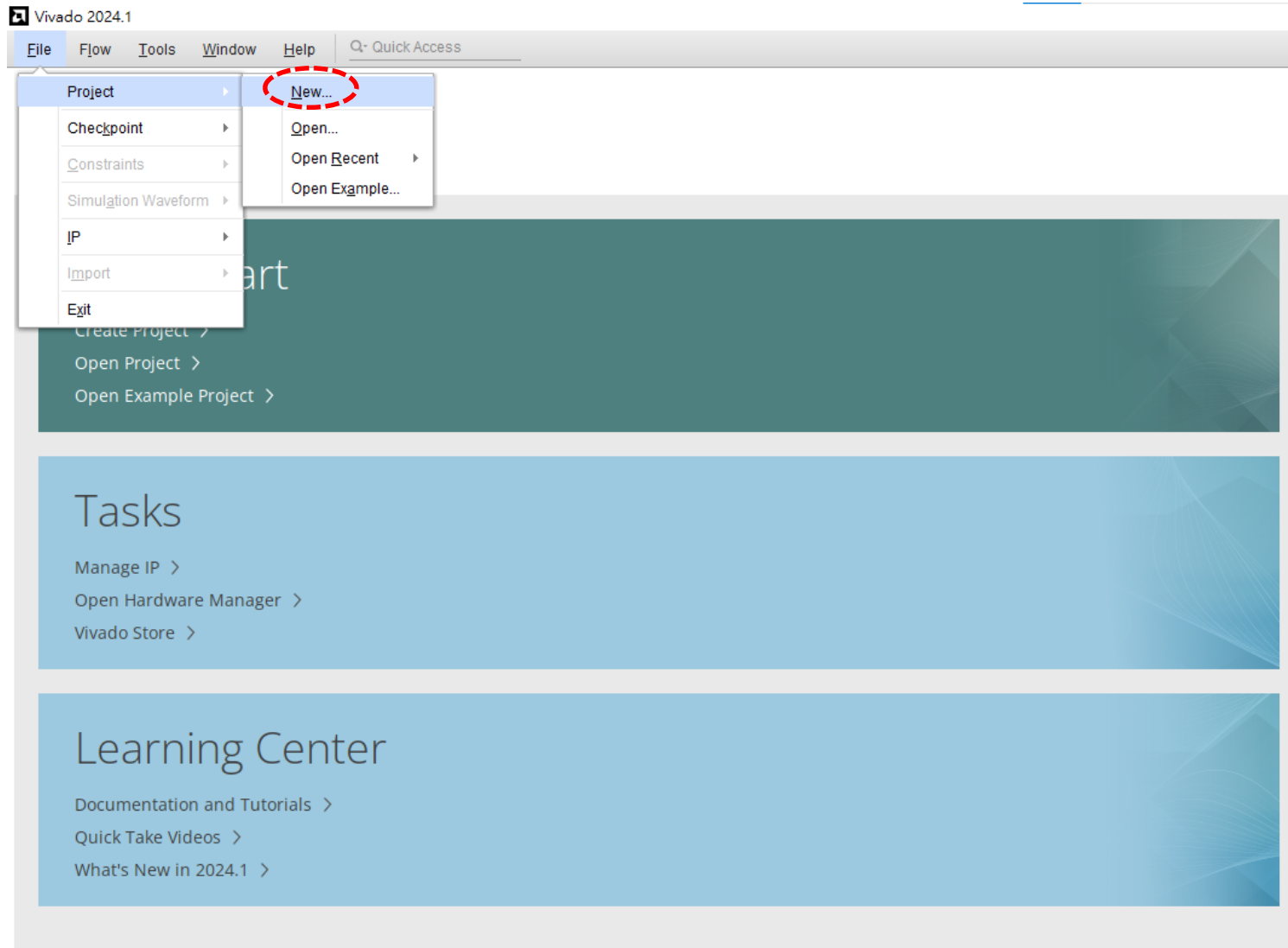
◆ Double-click the Vivado 2024.1 icon on the desktop:





Create a New Project in Vivado

Mat 1





Select Project Type

Mat 1

New Project [X]

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.

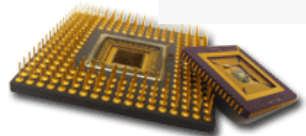
☐ Do not specify sources at this time

☐ **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

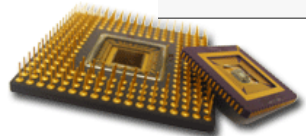
i To fetch the latest available boards from git repository, click on 'Refresh' button. [Dismiss](#)

[Reset All Filters](#)

Vendor: Name: Board Rev:

Display Name	Preview	Status	Vendor	File Version	Part	I/O Pin Count	Board Re
Arty		Installed	digilentinc.com	1.1	xc7a35ticsg324-1L	324	C.0
Artix-7 AC701 Evaluation Platform Add Companion Card Connections		Installed	xilinx.com	1.4	xc7a200tbg676-2	676	1.1
Kria K24C SOM Add Companion Card Connections		Installed	xilinx.com	1.0	Commercial temperature grade K24 SOM	530	Rev_A01

?





Add a New HDL Source Code

Mat 1

demo - [F:/Desktop/demo/demo.xpr] - Vivado 2024.1

File Edit Flow Tools Reports Window Layout View Help Q: 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 Elaborated Design
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
- PROGRAM AND DEBUG
 - Generate Bitstream

PROJECT MANAGER - demo

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: demo

Project location: F:/Desktop/demo

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

Repository path: [E:/xilinx/Vivado/2024.1/data/xhub/boards](#)

Tcl Console Messages Log Reports Design Runs

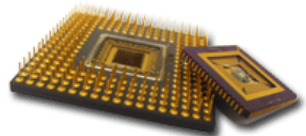
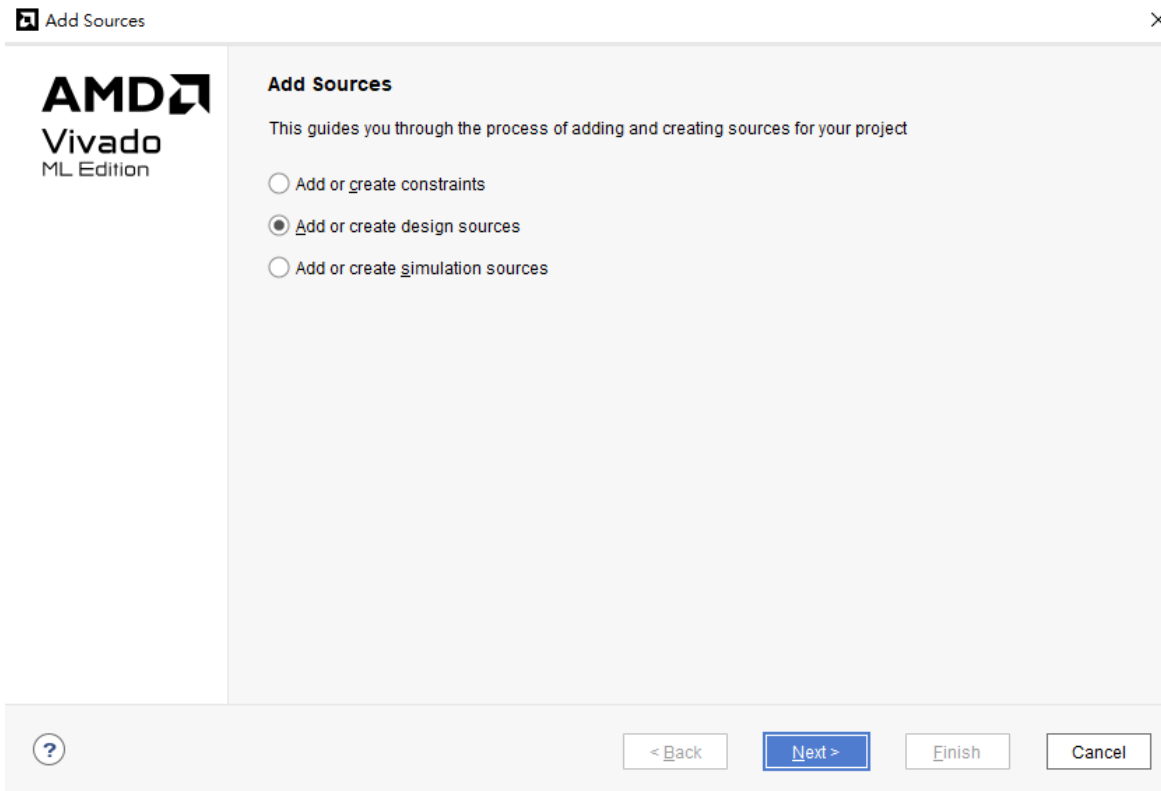
Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM
synth_1	constrs_1	Not started															
impl_1	constrs_1	Not started															



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.

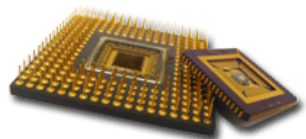
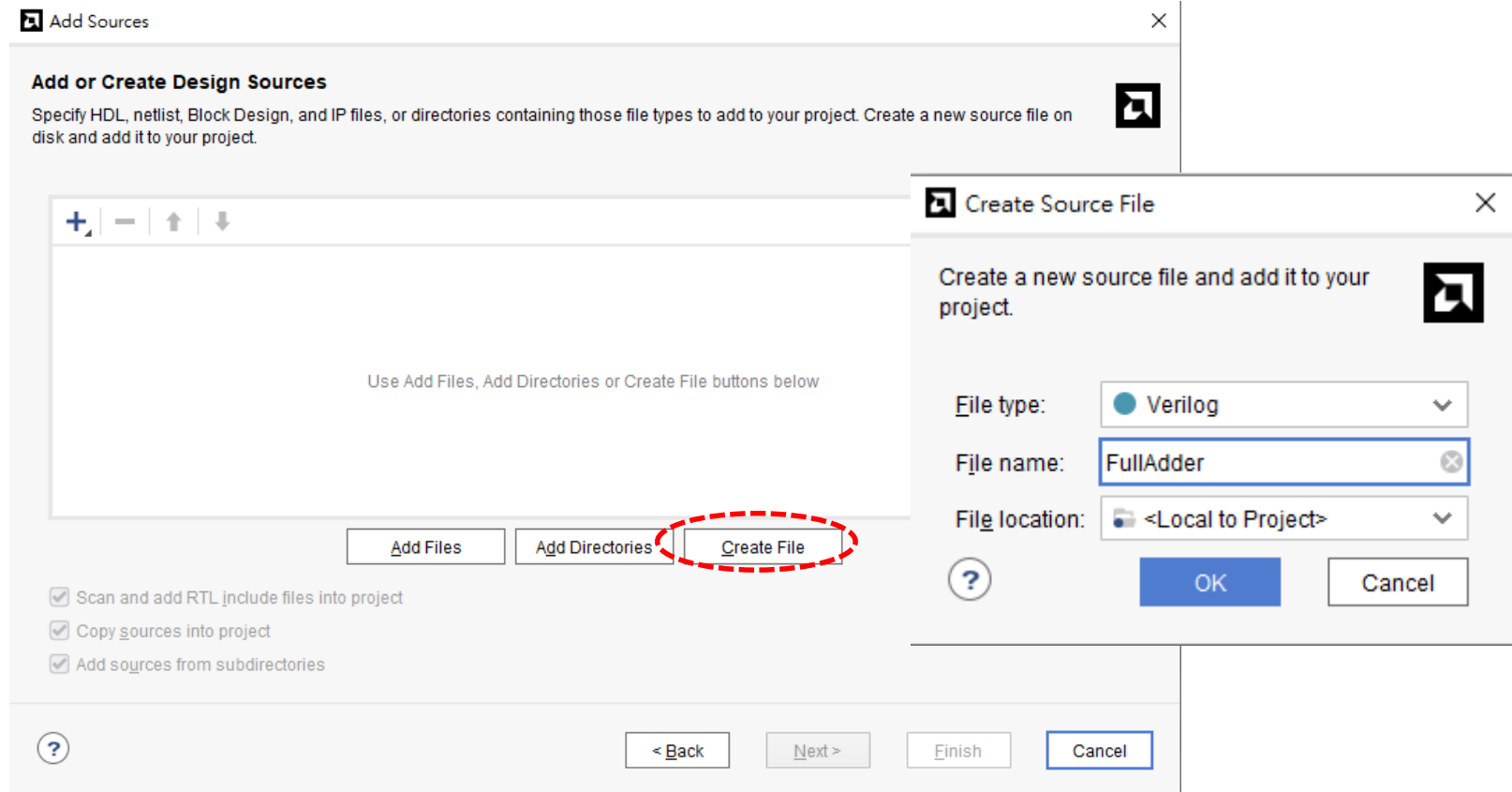




Create a 4-bit Full Adder Design

Mat 1

◆ Let's create a design source from scratch!





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

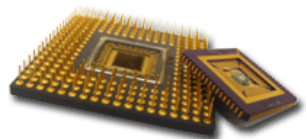
Create Source File

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

File type: Verilog

File name: FullAdder

File location: <Local to Project>





Confirm to Create the Verilog Module

Mat 1

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

Define Module [X]

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

? 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 2024.1 interface with the following components:

- Flow Navigator:** A sidebar on the left with sections for PROJECT MANAGER, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, IMPLEMENTATION, and PROGRAM AND DEBUG.
- PROJECT MANAGER - demo:** The main workspace area containing:
 - Sources:** A list of sources including Design Sources (with 'FullAdder (FullAdder.v)' circled in red), Constraints, Simulation Sources (1), and Utility Sources.
 - Properties:** A panel below Sources with a message: "Select an object to see properties".
 - Project Summary:** A panel on the right showing project details:
 - Project name: demo
 - Project location: F:/Desktop/demo
 - Product family: Artix-7
 - Project part: Arty (xc7a35tcs324-1L)
 - Top module name: Not defined
 - Target language: Verilog
 - Simulator language: Mixed
 - Board Part:** A section below Project Summary showing board details:
 - Display name: Arty
 - Board part name: digilentinc.com:arty:part0:1.1
 - Board revision: C.0
 - Connectors: No connections
 - Repository path: E:/xilinx/vivado/2024.1/data/xhub/boards
 - Design Runs:** A table at the bottom showing the status of synthesis and implementation 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															

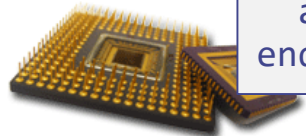


Type in the HDL Source Code

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
```

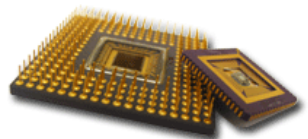
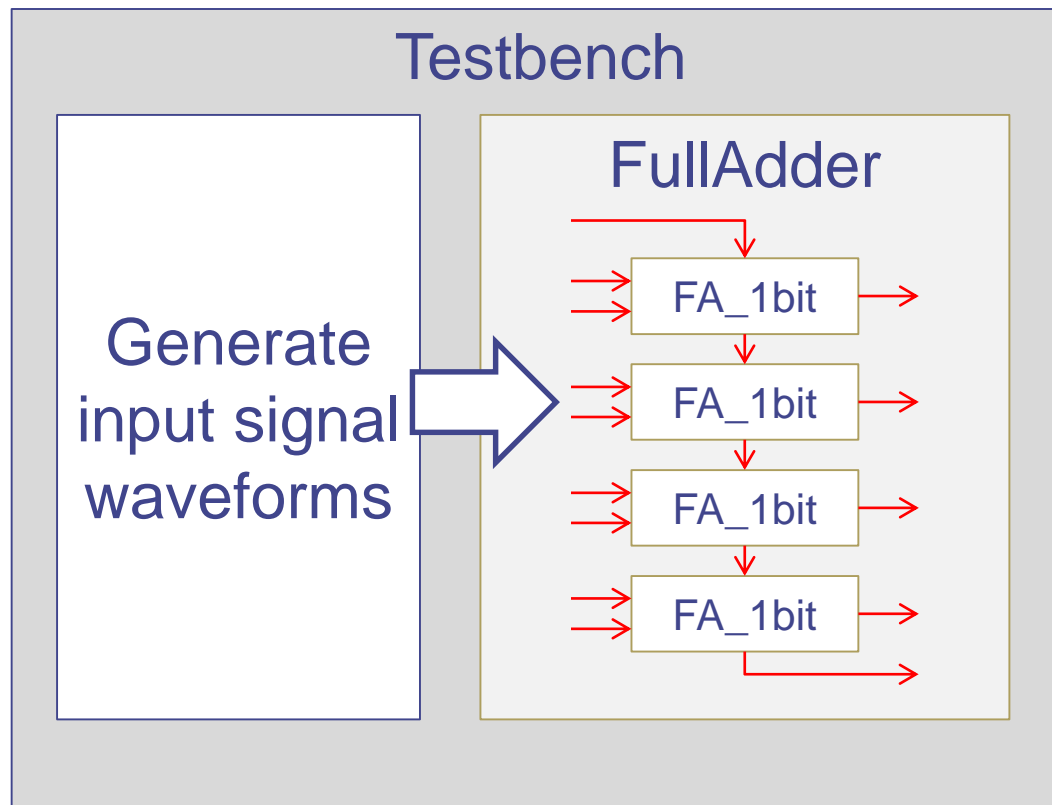




TestBench Design

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.

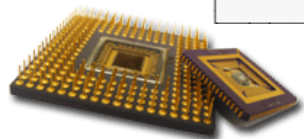
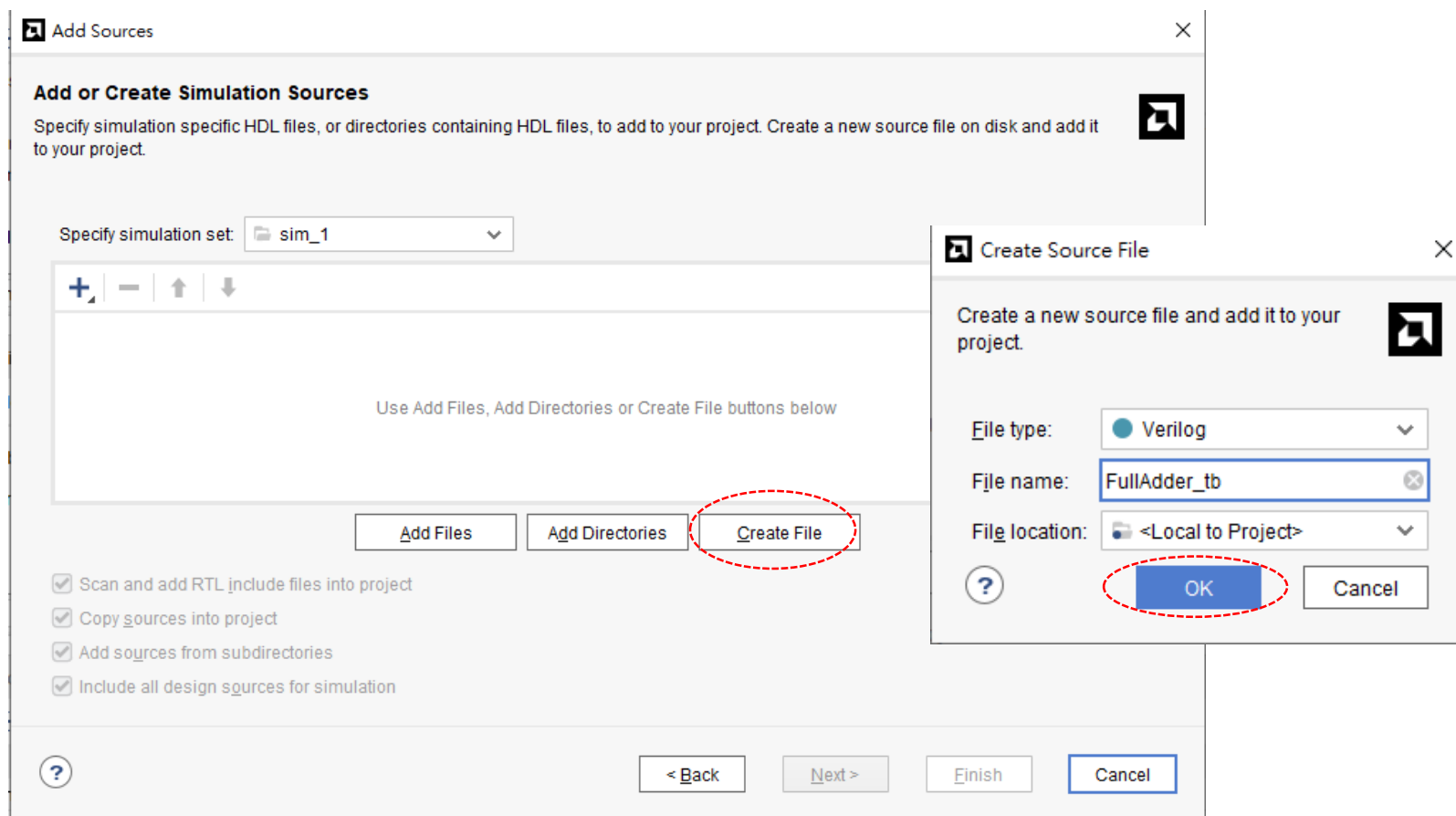




Create the Testbench Source Code

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

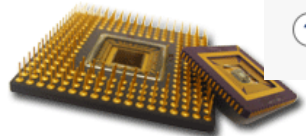
Add or Create Simulation Sources

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:

	Index	Name	Library	Location
<input checked="" type="radio"/>	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

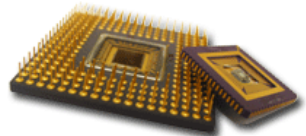
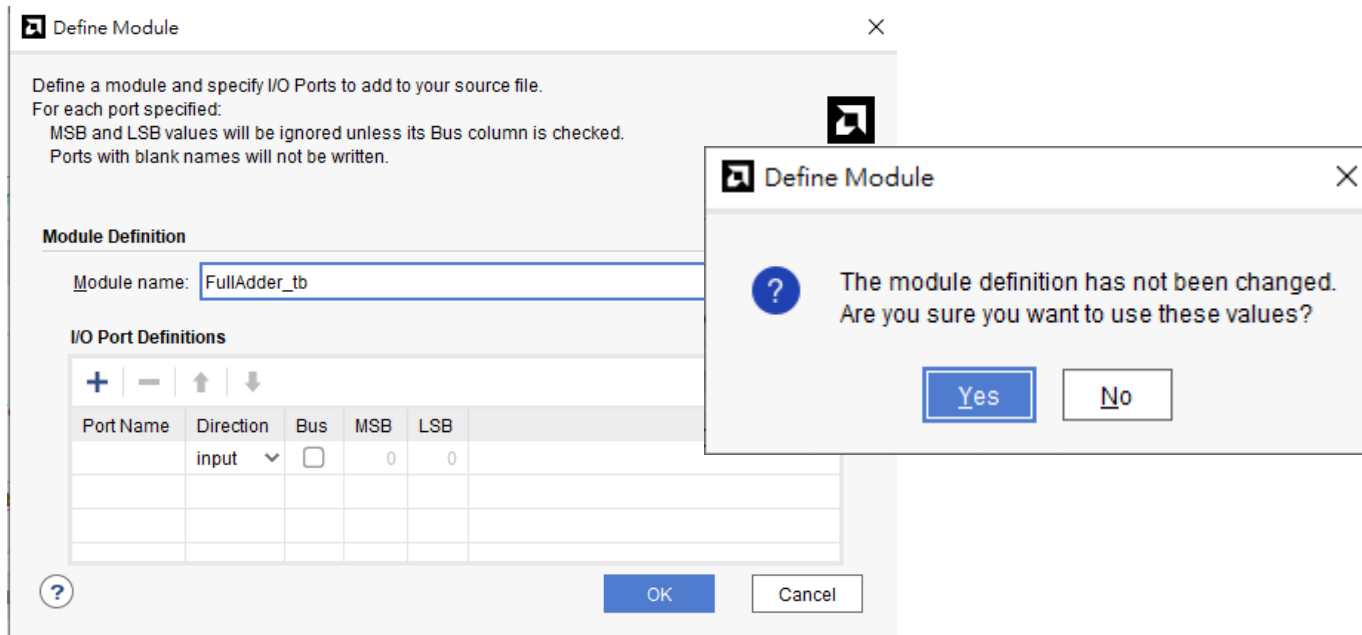




Create the Testbench Template

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

demo - [F:/Desktop/demo/demo.xpr] - Vivado 2024.1

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

Ready

Default Layout

Flow Navigator

PROJECT MANAGER - demo

Sources

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

Hierarchy Libraries Compile Order

Source File Properties

FullAdder_tb.v

☒ Enabled

Location: F:/Desktop/demo/demo.srscs/sim_1/new

Type: Verilog

Library: xil_defaultlib

General Properties

Project Summary x FullAdder.v x FullAdder_tb.v x

F:/Desktop/demo/demo.srscs/sim_1/new/FullAdder_tb.v

```

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 //
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;

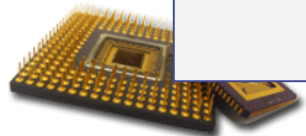
// Instantiate the Unit
// Under Test (UUT)

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

demo - [F:/Desktop/demo/demo.xpr] - Vivado 2024.1

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

Ready

Default Layout

Flow Navigator

PROJECT MANAGER - demo

Sources

Design Sources (1)

FullAdder (FullAdder.v) (4)

Constraints

Simulation Sources (1)

sim_1 (1)

FullAdder_tb (FullAdder_tb.v) (1)

Utility Sources

Hierarchy Libraries Compile Order

Source File Properties

FullAdder_tb.v

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.v

FullAdder_tb.v

F:/Desktop/demo/demo.srcs/sim_1/new/FullAdder_tb.v

```

6
7 // outputs
8 wire [3:0] S;
9 wire Cout;
10
11 // Instantiate the Unit
12 // Under Test (UUT)
13
14 FullAdder uut (
15     .A(A),
16     .B(B),
17     .Cin(Cin),
18     .S(S),
19     .Cout(Cout)
20 );
21 initial begin
22     // Initialize Inputs
23     A = 0; B = 0; Cin = 0;
24
25     // Wait 100 ns for global
26     // reset to finish
  
```

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															



Vivado Simulator Window

Mat 1

demo - [F:/Desktop/demo/demo.xpr] - Vivado 2024.1

File Edit **Flow** Tools Reports Window Layout View Run Help

Quick Access

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

SIMULATION - Behavioral Simulation - Functional - sim_1 - FullAdder_tb

Scope Sources

Name	Design Unit	Block Type
FullAdder_tb	FullAdder_tb	Verilog Mo
uut	FullAdder	Verilog Mo
gbl	gbl	Verilog Mo

Objects

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.v x FullAdder_tb.v x **Untitled 1**

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

0 5 f 0 6

0 a 1 f 1

0 f 0 8

