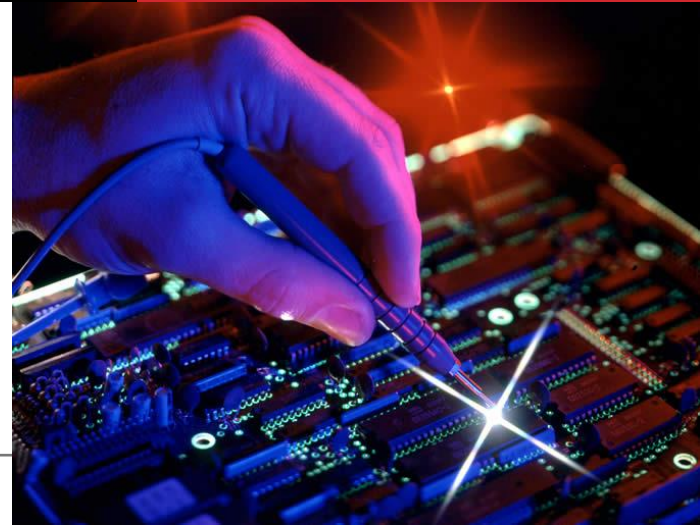




# **Advanced Computer Architecture & Advanced Microprocessor System**

## **VIVADO BASED ARTIX 7 FPGA BOARD USER GUIDE**

**Dennis A. N. Gookyi**





# CONTENTS

## ❖ Design Example





# DESIGN EXAMPLE

## ❖ Verilog code and Testbench

### □ Gates design

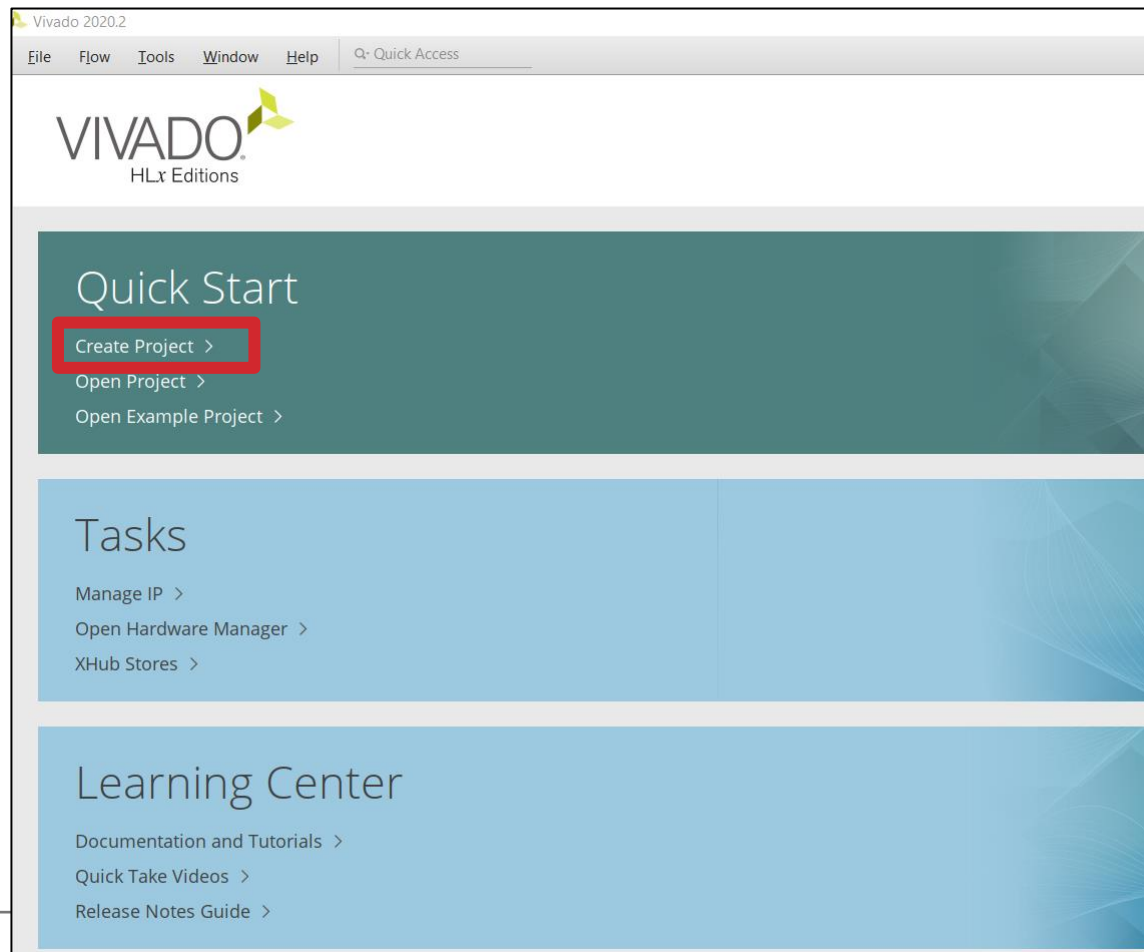
- AND gate
- OR gate
- XOR gate





# DESIGN EXAMPLE

- ❖ Vivado project
  - Creating an new project

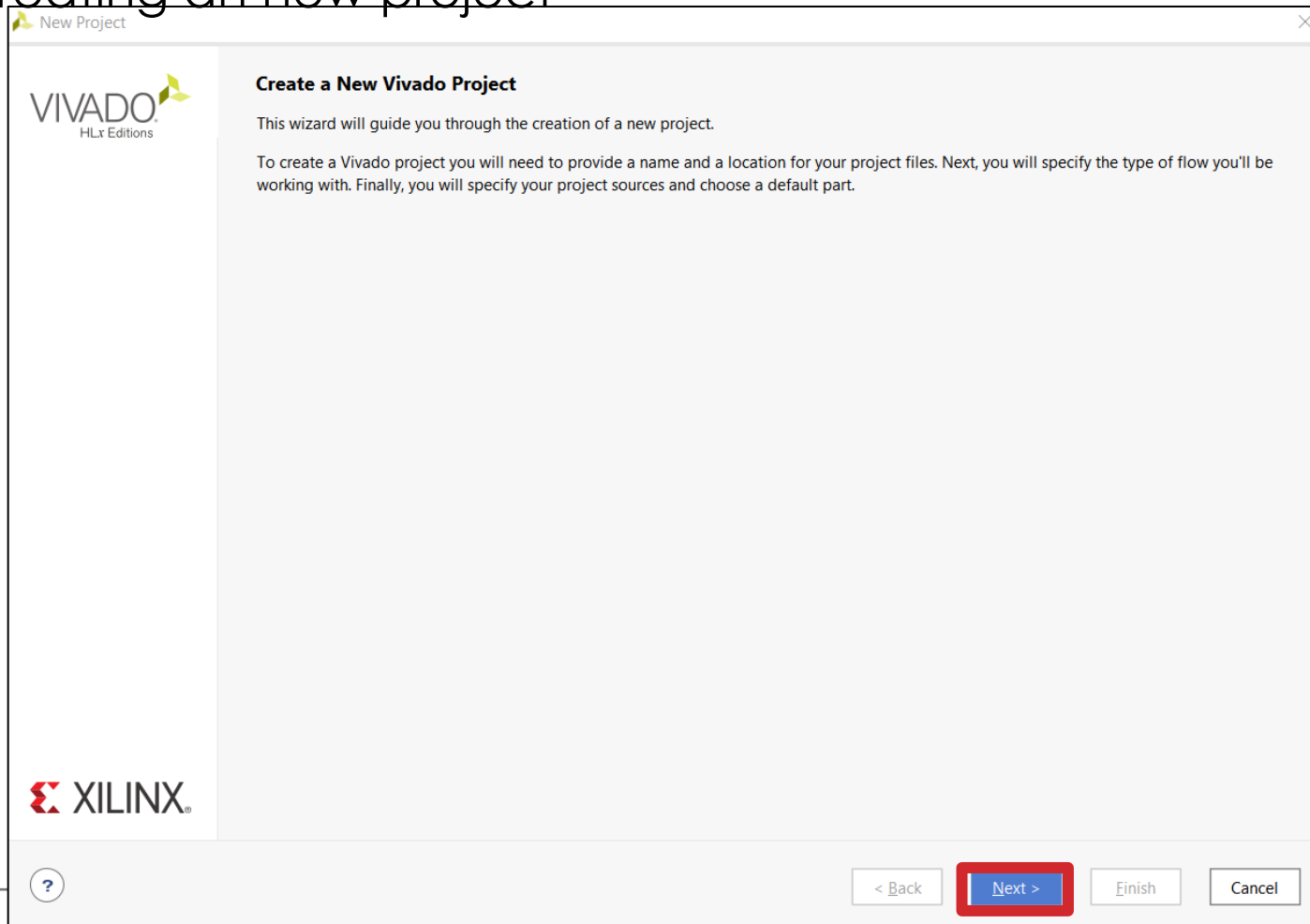




# DESIGN EXAMPLE

## ❖ Vivado project

### □ Creating an new project





# DESIGN EXAMPLE

## ❖ Vivado project

### □ Creating an new project

New Project

**Project Name**  
Enter a name for your project and specify a directory where the project data files will be stored.

Project name:

Project location:

☒ Create project subdirectory

Project will be created at: D:/vivado\_projects/GATES

Project name

- must start with an alphabet
- can include "\_" and numbers

Location

- location of the project
- subfolders can be created





# DESIGN EXAMPLE

## ❖ Vivado project

### □ Creating an new project

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, XST or ISE Project File.

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

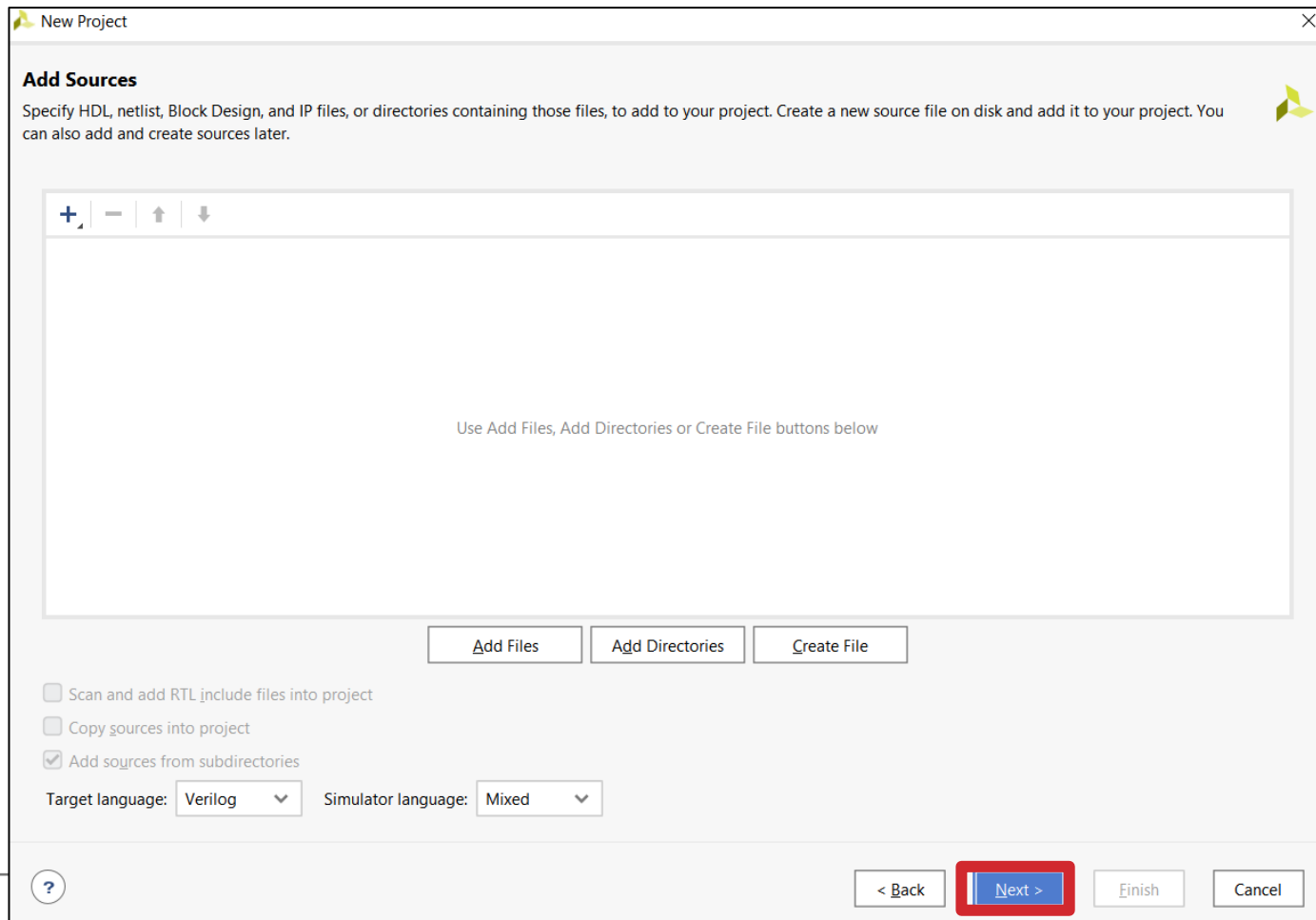
? < Back **Next >** Finish Cancel



# DESIGN EXAMPLE

## ❖ Vivado project

### □ Creating an new project



New Project

**Add Sources**

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

Use Add Files, Add Directories or Create File buttons below

Add Files Add Directories Create File

☐ Scan and add RTL include files into project

☐ Copy sources into project

☒ Add sources from subdirectories

Target language: Verilog Simulator language: Mixed

? < Back Next > Finish Cancel

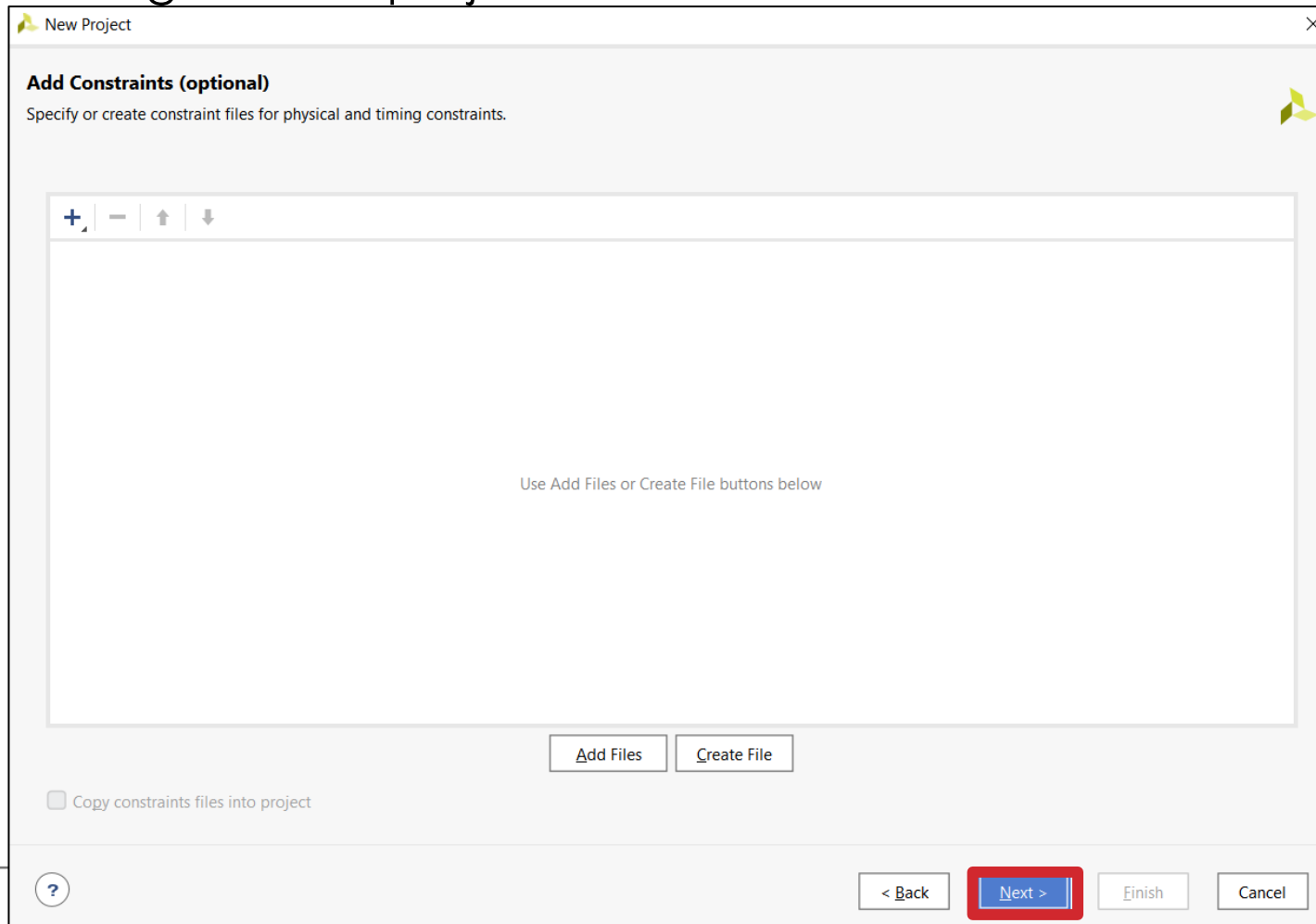




# DESIGN EXAMPLE

## ❖ Vivado project

### □ Creating an new project





# DESIGN EXAMPLE

- ❖ Vivado project
  - Creating an new project

**FPGA Device:**  
**Xc7a35tftg256-1**

New Project

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

**Parts** | Boards

[Reset All Filters](#)

Category: All Package: ftg256 Temperature: All Remaining  
Family: Artix-7 Speed: -1 Static power: All Remaining

Search: Q

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb Transceivers	GTPE2 Transceive
xc7a15tftg256-1	256	170	10400	20800	25	0	45	0	0
xc7a35tftg256-1	256	170	20800	41600	50	0	90	0	0
xc7a50tftg256-1	256	170	32600	65200	75	0	120	0	0
xc7a75tftg256-1	256	170	47200	94400	105	0	180	0	0
xc7a100tftg256-1	256	170	63400	126800	135	0	240	0	0

< Back Next > Finish Cancel

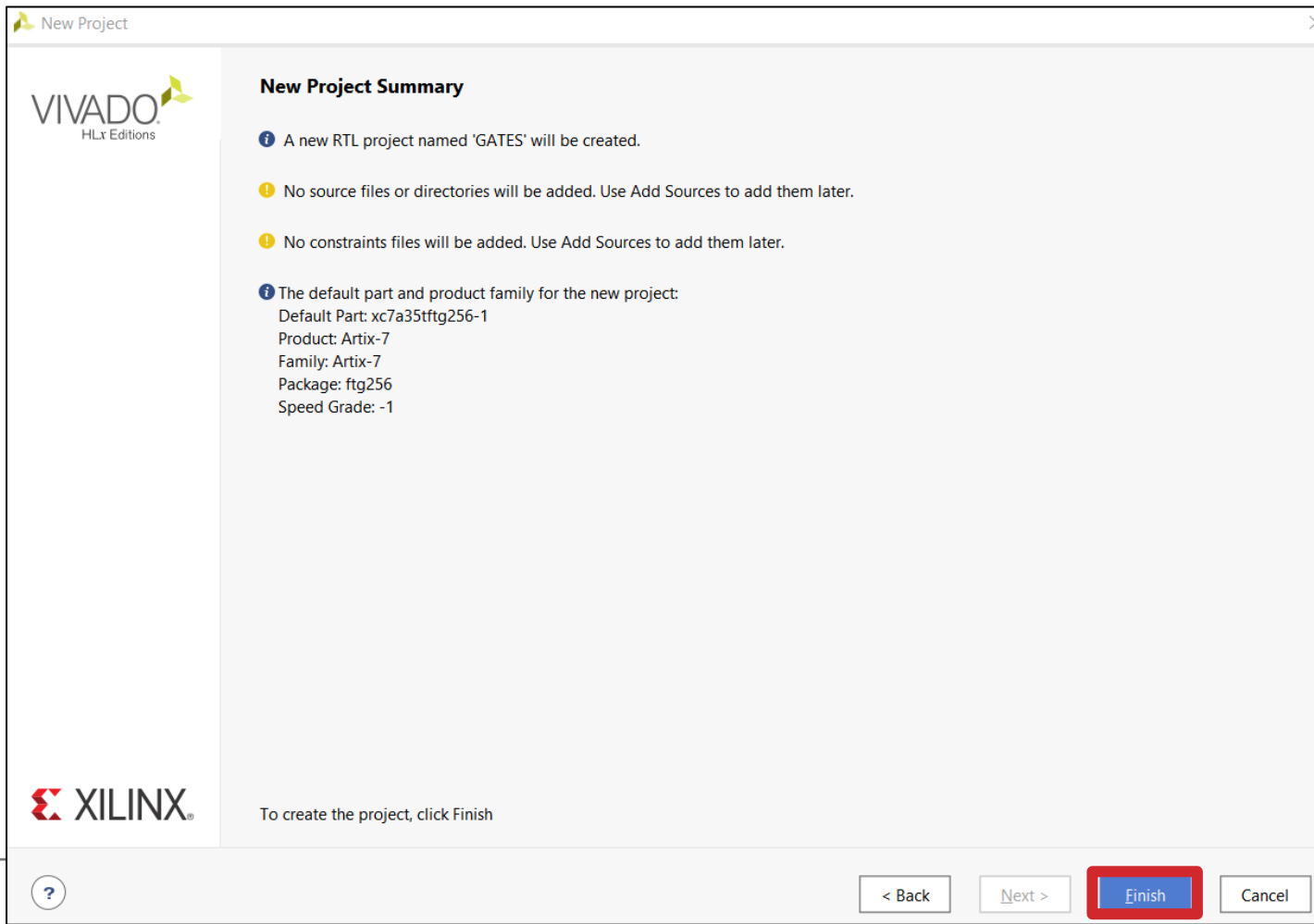




# DESIGN EXAMPLE

## ❖ Vivado project

### □ Creating an new project



# DESIGN EXAMPLE

## Vivado project

GATES - [D:/vivado\_projects/GATES/GATES.xpr] - Vivado 2020.2

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

Flow Navigator PROJECT MANAGER - GATES

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

- 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: GATES

Project location: D:/vivado\_projects/GATES

Product family: Artix-7

Project part: xc7a35tftg256-1

Top module name: Not defined

Target language: Verilog

Simulator language: Mixed

Synthesis Implementation

Status: Not started

Messages: No errors or warnings

Part: xc7a35tftg256-1

Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2020)
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2020)

# DESIGN EXAMPLE

## Vivado project

GATES - [D:/vivado\_projects/GATES/GATES.xpr] - Vivado 2020.2

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

Flow Navigator PROJECT MANAGER - GATES

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

- 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: GATES

Project location: D:/vivado\_projects/GATES

Product family: Artix-7

Project part: xc7a35tftg256-1

Top module name: Not defined

Target language: Verilog

Simulator language: Mixed

Synthesis Implementation

Status: Not started

Messages: No errors or warnings

Part: xc7a35tftg256-1

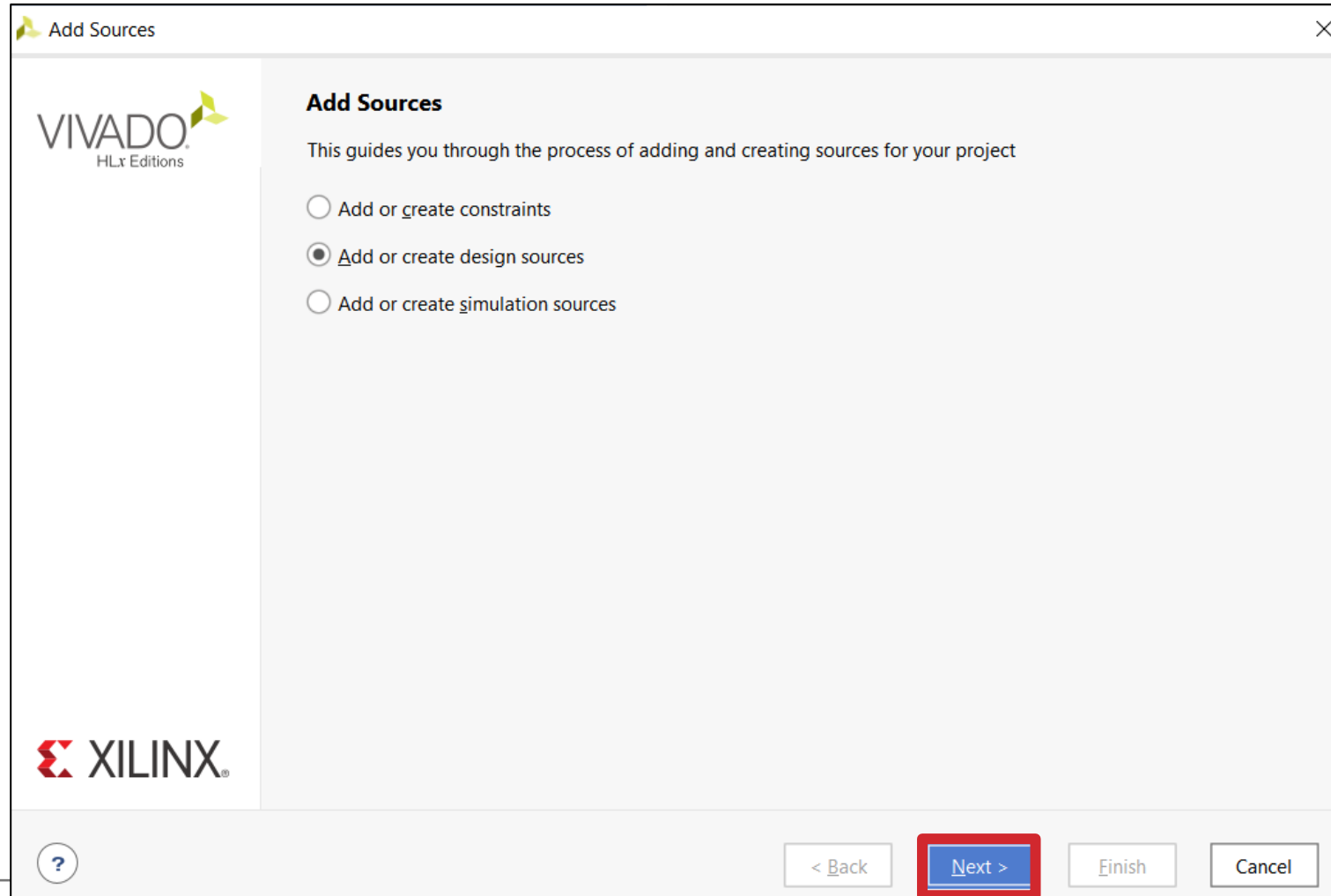
Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2020)
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2020)



# DESIGN EXAMPLE

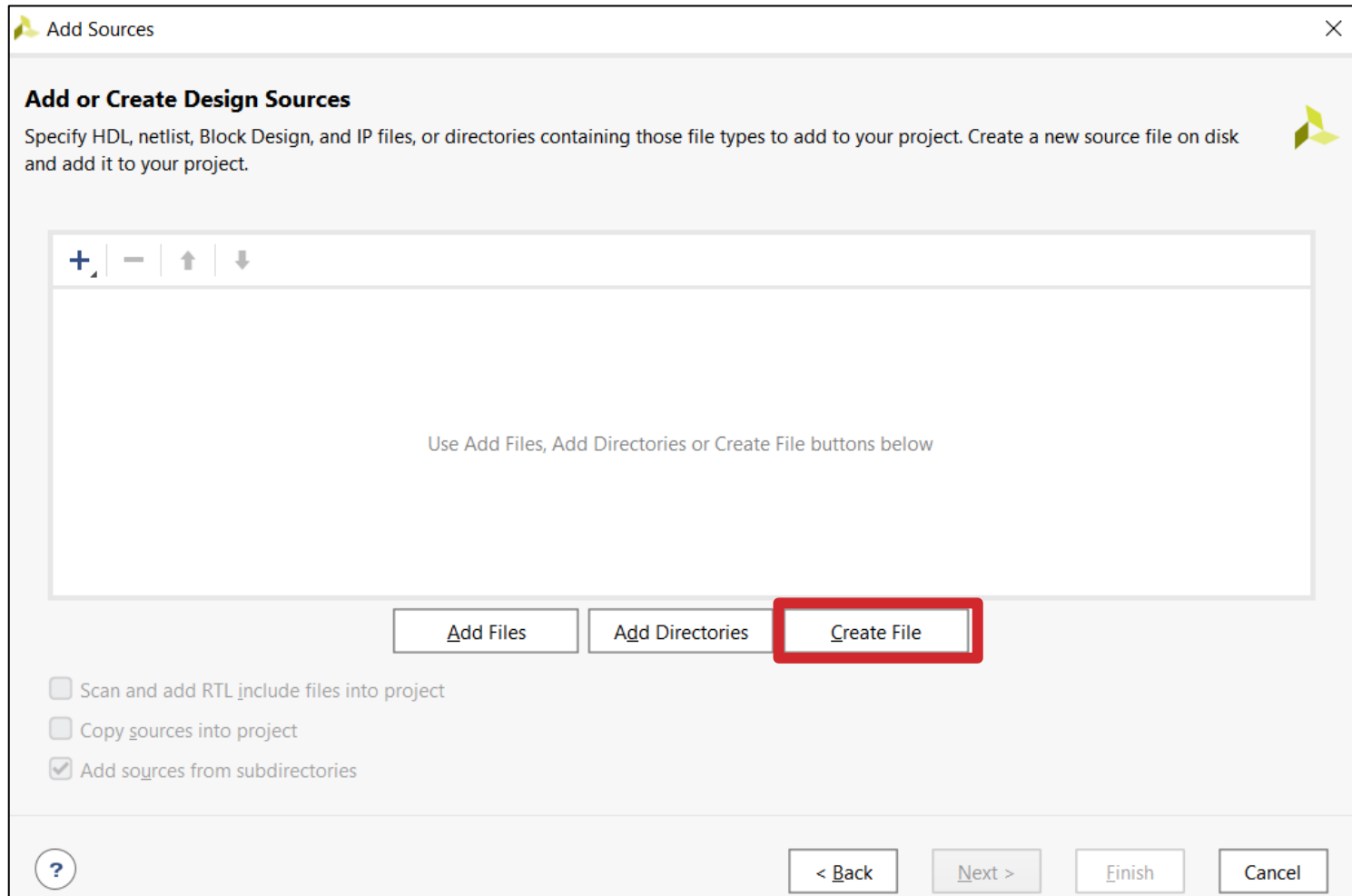
## ❖ Vivado project





# DESIGN EXAMPLE

## ❖ Vivado project





# DESIGN EXAMPLE

## ❖ Vivado project

Create Source File

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

File type: Verilog

File name: gate

File location: <Local to Project>

?

OK

Cancel







# DESIGN EXAMPLE

## ❖ Vivado project

**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	gate.v	xil_defaultlib	<Local to Project>

**Buttons:** Add Files, Add Directories, Create File

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


**Navigation:** < Back, Next >, **Finish**, Cancel





# DESIGN EXAMPLE

## ❖ Vivado project

 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	

?

OK

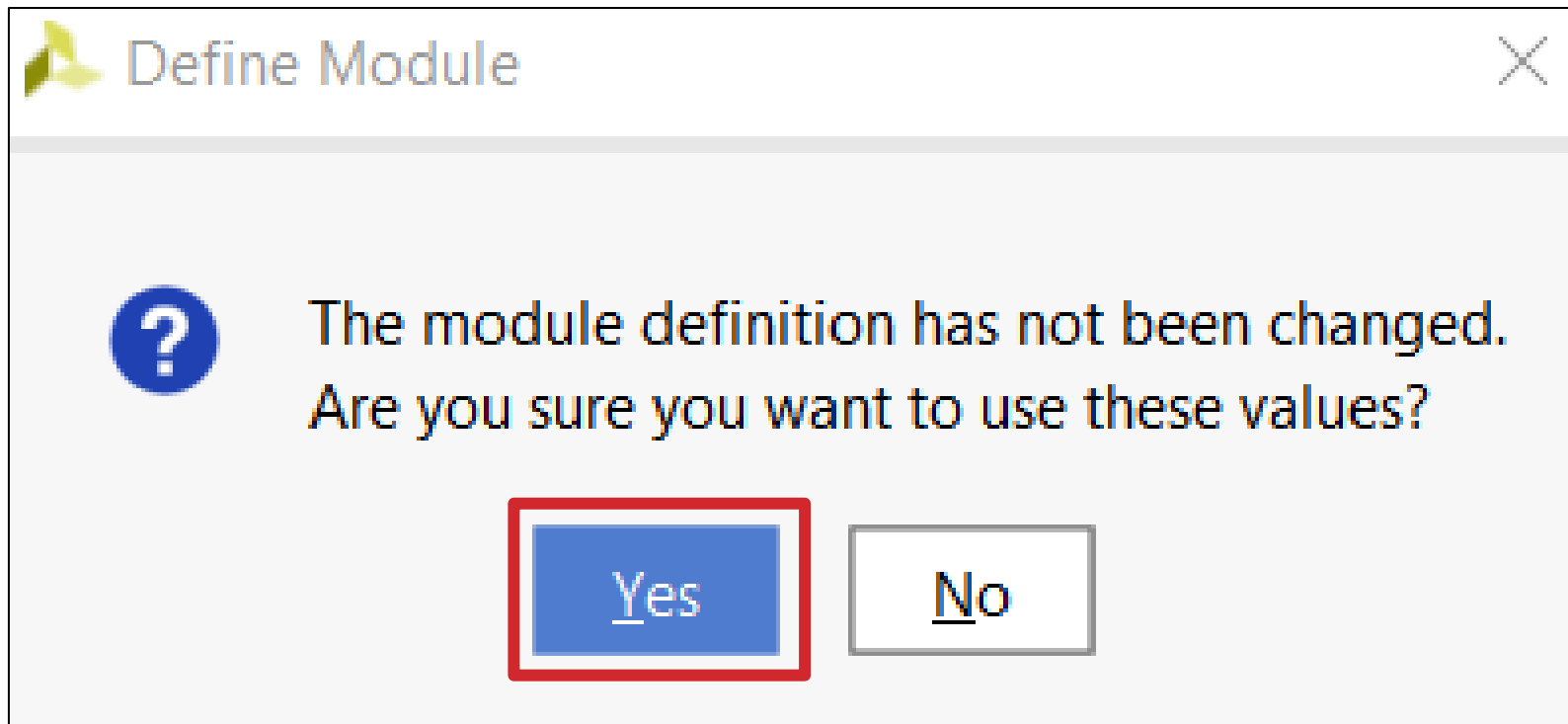
Cancel





# DESIGN EXAMPLE

## ❖ Vivado project





# DESIGN EXAMPLE

## ❖ Vivado project

PROJECT MANAGER - GATES

**Sources**

Design Sources (1)  
gate (gate.v)

Constraints

Simulation Sources (1)  
sim\_1 (1)

Utility Sources

Hierarchy Libraries Compi

**Source File Property**

gate.v

Enabled

Location: D:/vivado\_projects/GATES

General Properties

**Project Summary** x **gate.v** x

D:/vivado\_projects/GATES/GATES.srcs/sources\_1/new/gate.v

**Editor**

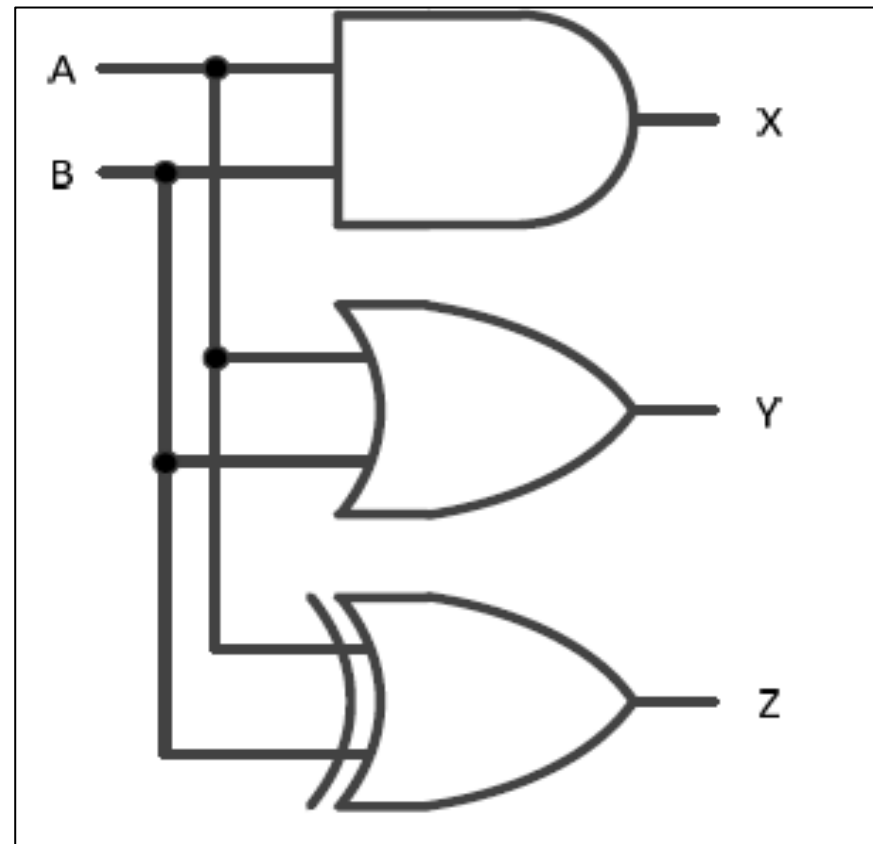
```
5 //  
6 // Create Date: 07/25/2023 02:26:43 PM  
7 // Design Name:  
8 // Module Name: gate  
9 // Project Name:  
0 // Target Devices:  
1 // Tool Versions:  
2 // Description:  
3 //  
4 // Dependencies:  
5 //  
6 // Revision:  
7 // Revision 0.01 - File Created  
8 // Additional Comments:  
9 //  
0 ///////////////////////////////////////////  
1  
2  
3 module gate(  
4  
5 );  
6 endmodule  
7
```





# DESIGN EXAMPLE

- ❖ Vivado project
  - Verilog code and Testbench
    - **AND-OR-XOR** module



AND-OR-XOR Schematic



# DESIGN EXAMPLE

- ❖ Vivado project
  - Verilog code **AND-OR-XOR** module

PROJECT MANAGER - GATES

Sources

- Design Sources (1)
  - gate (gate.v)
- Constraints
- Simulation Sources (1)
  - sim\_1 (1)
- Utility Sources

Hierarchy Libraries Compi

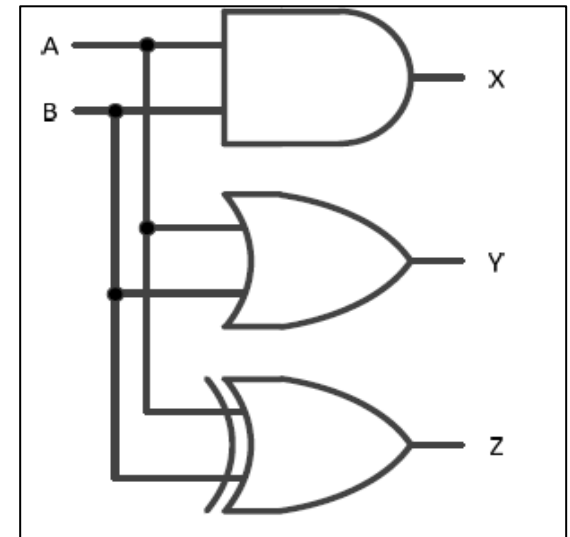
Source File Properti

- gate.v
- Enabled

Project Summary gate.v

D:/vivado\_projects/GATES/GATES.srcs/sources\_1/new/

```
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////
21
22
23 module gate(A, B, X, Y, Z);
24     //Input and output declaration
25     input A, B;
26     output X, Y, Z;
27     //Gate Declaration
28     assign X = A & B;
29     assign Y = A | B;
30     assign Z = A ^ B;
31 endmodule
```

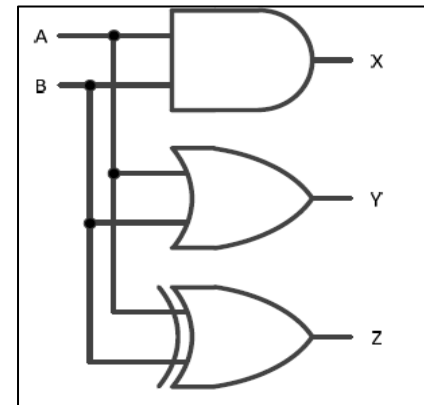


# DESIGN EXAMPLE

- ❖ Vivado project
- ❑ Logic synthesis

The screenshot displays the Vivado IDE interface for a project named "GATES". The "Flow Navigator" on the left shows the project workflow, with the "SYNTHESIS" step highlighted in a red box and the "Run Synthesis" button visible. The "PROJECT MANAGER - GATES" window shows the "Sources" tab with "gate.v" selected. The "Source File Property" window for "gate.v" shows it is "Enabled". The "Project Summary" window shows the project name "gate.v" and the source file path "D:/vivado\_projects/GATES/GATES.srscs/sources\_1/new/g". The "Source Code" window displays the Verilog code for the "gate" module, which implements a logic circuit with three outputs: X (AND), Y (OR), and Z (XOR).

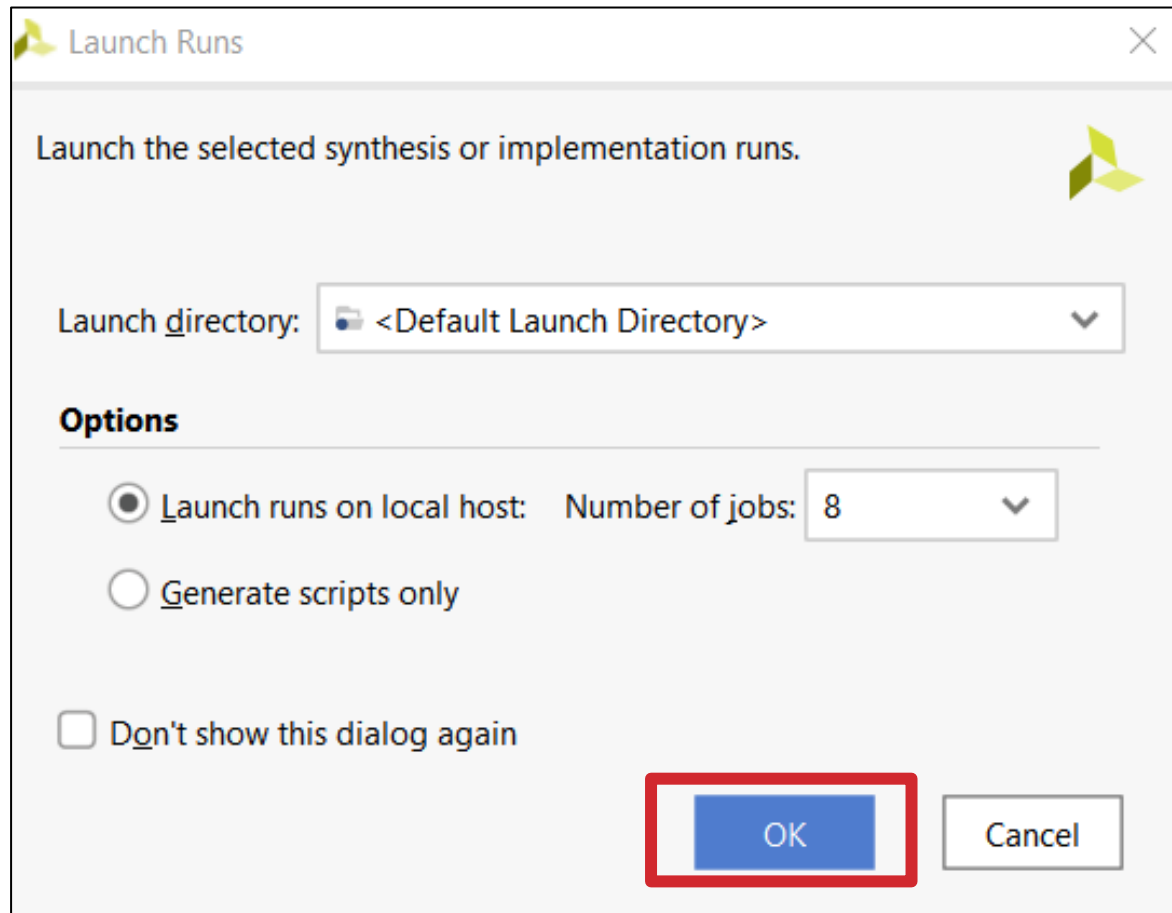
```
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////
21
22
23 module gate(A, B, X, Y, Z);
24     //Input and output declaration
25     input A, B;
26     output X, Y, Z;
27     //Gate Declaration
28     assign X = A & B;
29     assign Y = A | B;
30     assign Z = A ^ B;
31 endmodule
32
```





# DESIGN EXAMPLE

- ❖ Vivado project
  - Logic synthesis

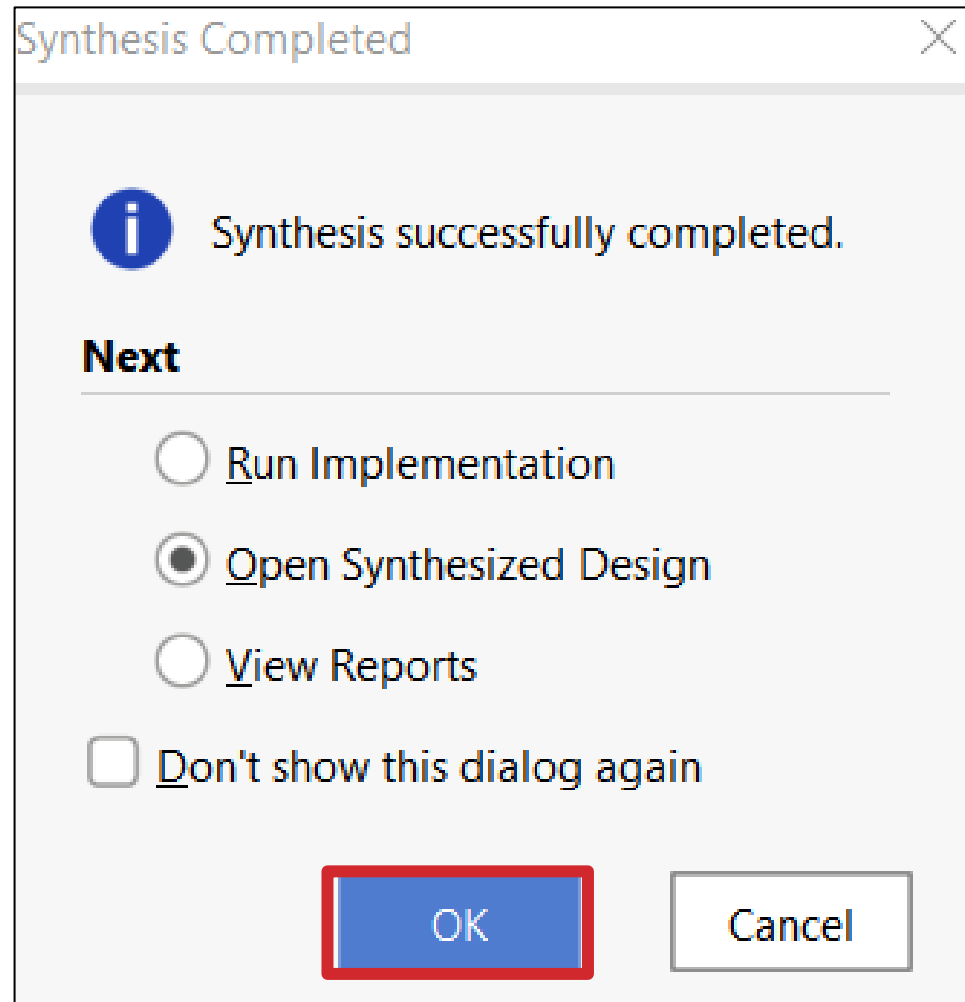






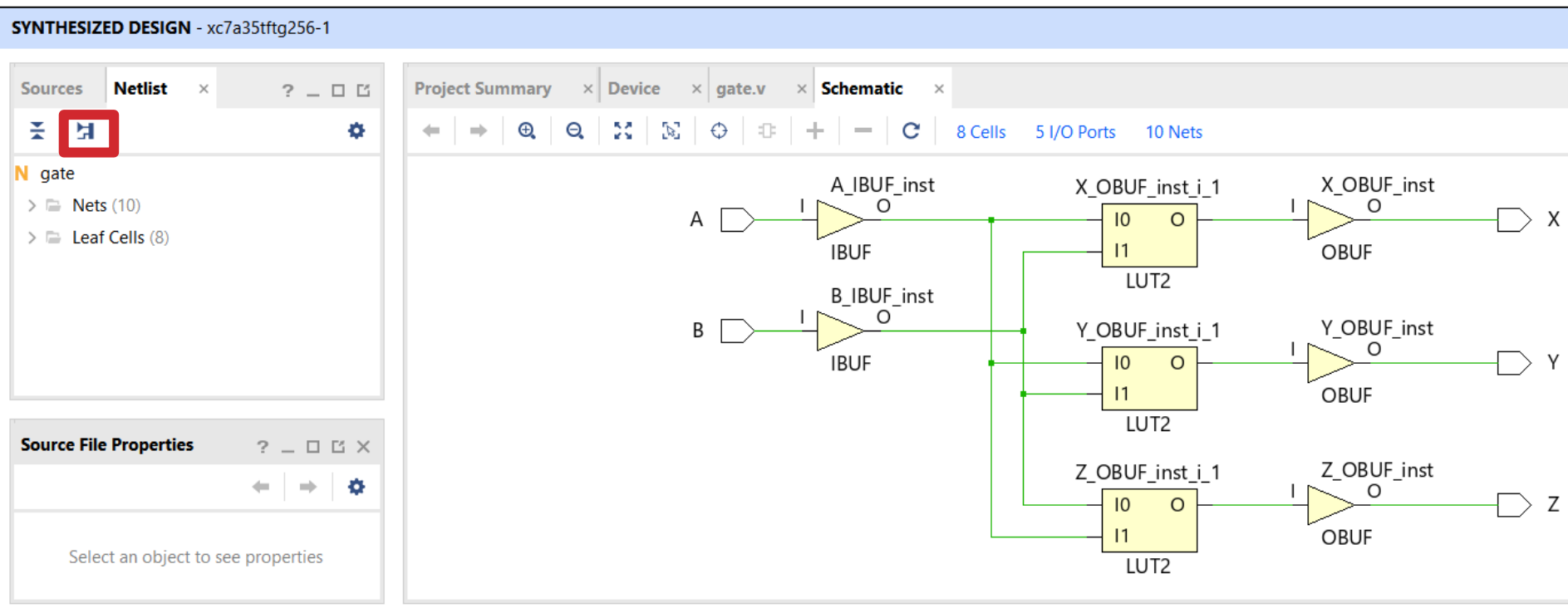
# DESIGN EXAMPLE

- ❖ Vivado project
  - Logic synthesis



# DESIGN EXAMPLE

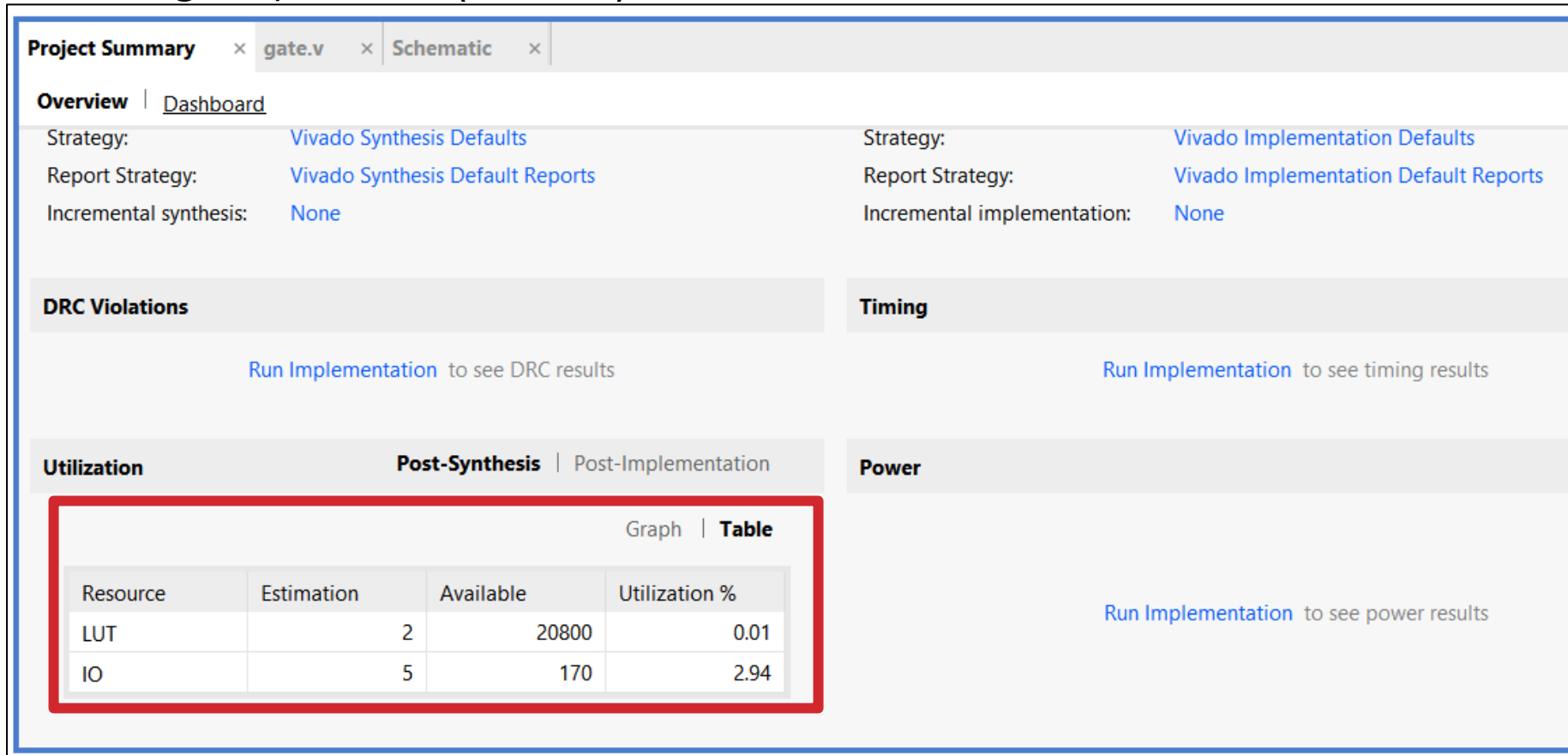
- ❖ Vivado project
  - Logic synthesis (Schematic)



# DESIGN EXAMPLE

## ❖ Vivado project

### □ Logic synthesis (Results)



The screenshot shows the Vivado IDE interface with the 'Project Summary' window open. The window has tabs for 'gate.v' and 'Schematic'. The 'Overview' tab is selected, showing a 'Dashboard' with various project settings and results sections.

**Project Summary** × **gate.v** × **Schematic** ×

**Overview** | [Dashboard](#)

Strategy: [Vivado Synthesis Defaults](#) Strategy: [Vivado Implementation Defaults](#)  
Report Strategy: [Vivado Synthesis Default Reports](#) Report Strategy: [Vivado Implementation Default Reports](#)  
Incremental synthesis: [None](#) Incremental implementation: [None](#)

**DRC Violations**

[Run Implementation](#) to see DRC results

**Timing**

[Run Implementation](#) to see timing results

**Utilization** **Post-Synthesis** | Post-Implementation

[Graph](#) | **Table**

Resource	Estimation	Available	Utilization %
LUT	2	20800	0.01
IO	5	170	2.94

**Power**

[Run Implementation](#) to see power results



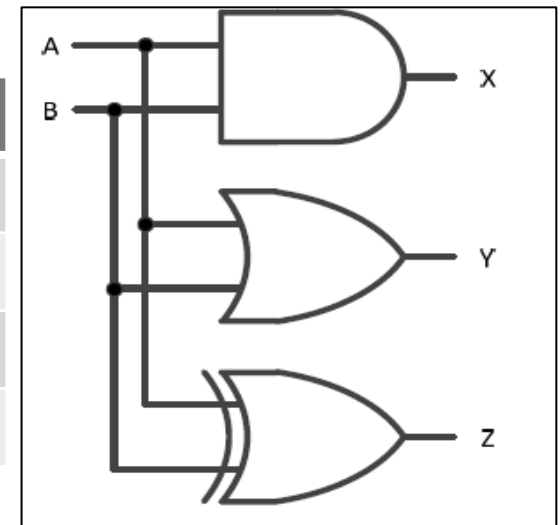


# DESIGN EXAMPLE

## ❖ Vivado project

- Testbench for **AND-OR-XOR** module (Test Vectors)

A	B	X	Y	Z
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



# DESIGN EXAMPLE

- ❖ Vivado project
  - Creating a Testbench

The screenshot displays the Vivado IDE interface with the following components:

- Flow Navigator:** A sidebar on the left containing project management tasks such as Settings, Add Sources, Language Templates, IP Catalog, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, and SYNTHESIS.
- PROJECT MANAGER - GATES:** The central workspace showing the project hierarchy. The **Sources** tab is active, listing Design Sources (1) including **gate (gate.v)**, Constraints, Simulation Sources (1) including **sim\_1 (1)**, and Utility Sources. A red box highlights the **+** icon in the Sources toolbar.
- Properties:** A panel at the bottom showing the properties for the selected **gate.v** source, including a checkbox for **Enabled** and a **Location** field.
- Code Editor:** The rightmost pane displays the **gate.v** source code. It includes a Project Summary section with metadata and a Verilog module definition for **gate** with inputs **A** and **B**, and outputs **X**, **Y**, and **Z**.

```
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 gate(A, B, X, Y, Z);
24     //Input and output declaration
25     input A, B;
26     output X, Y, Z;
27     //Gate Declaration
28     assign X = A & B;
29     assign Y = A | B;
30     assign Z = A ^ B;
31 endmodule
32
```

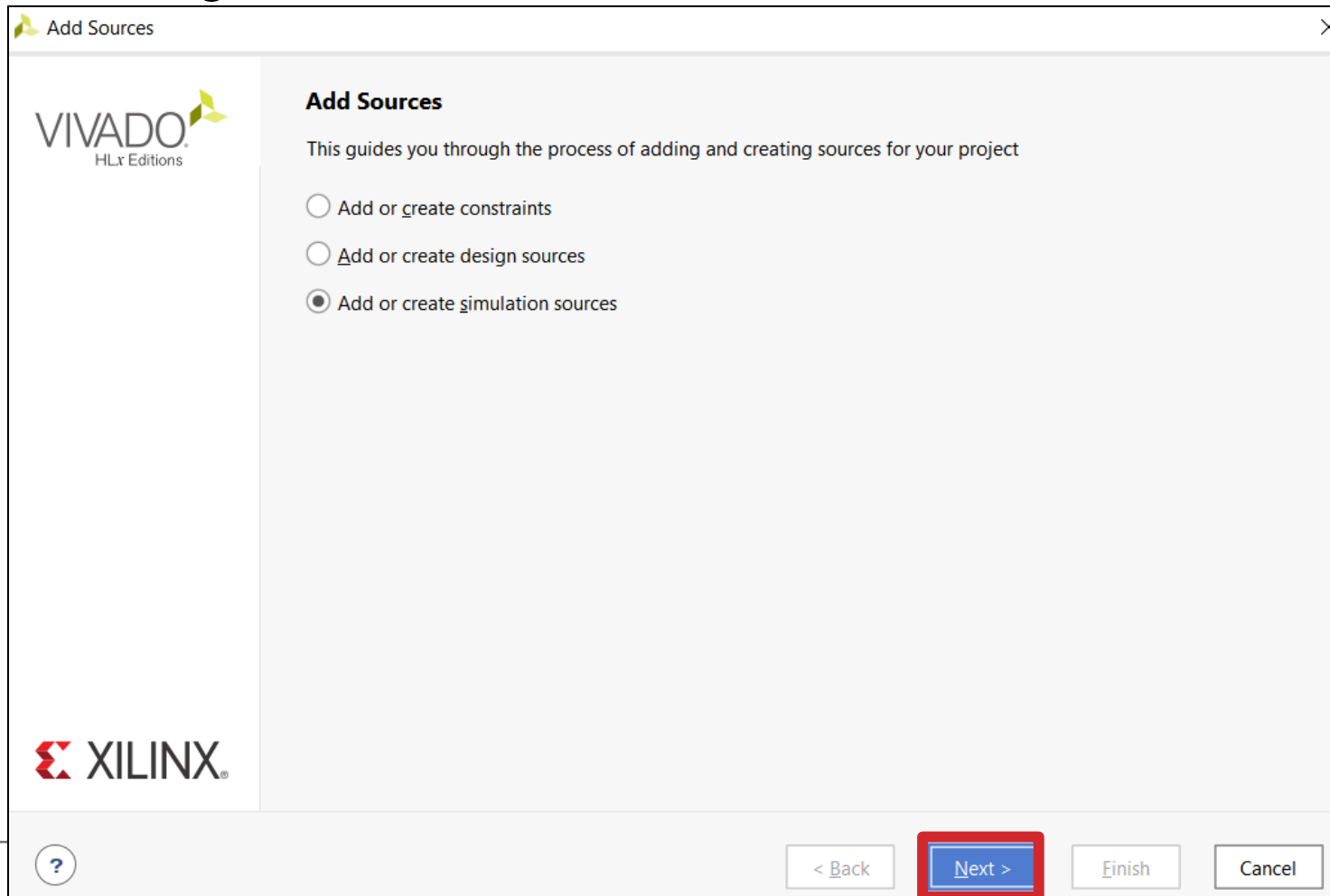




# DESIGN EXAMPLE

## ❖ Vivado project

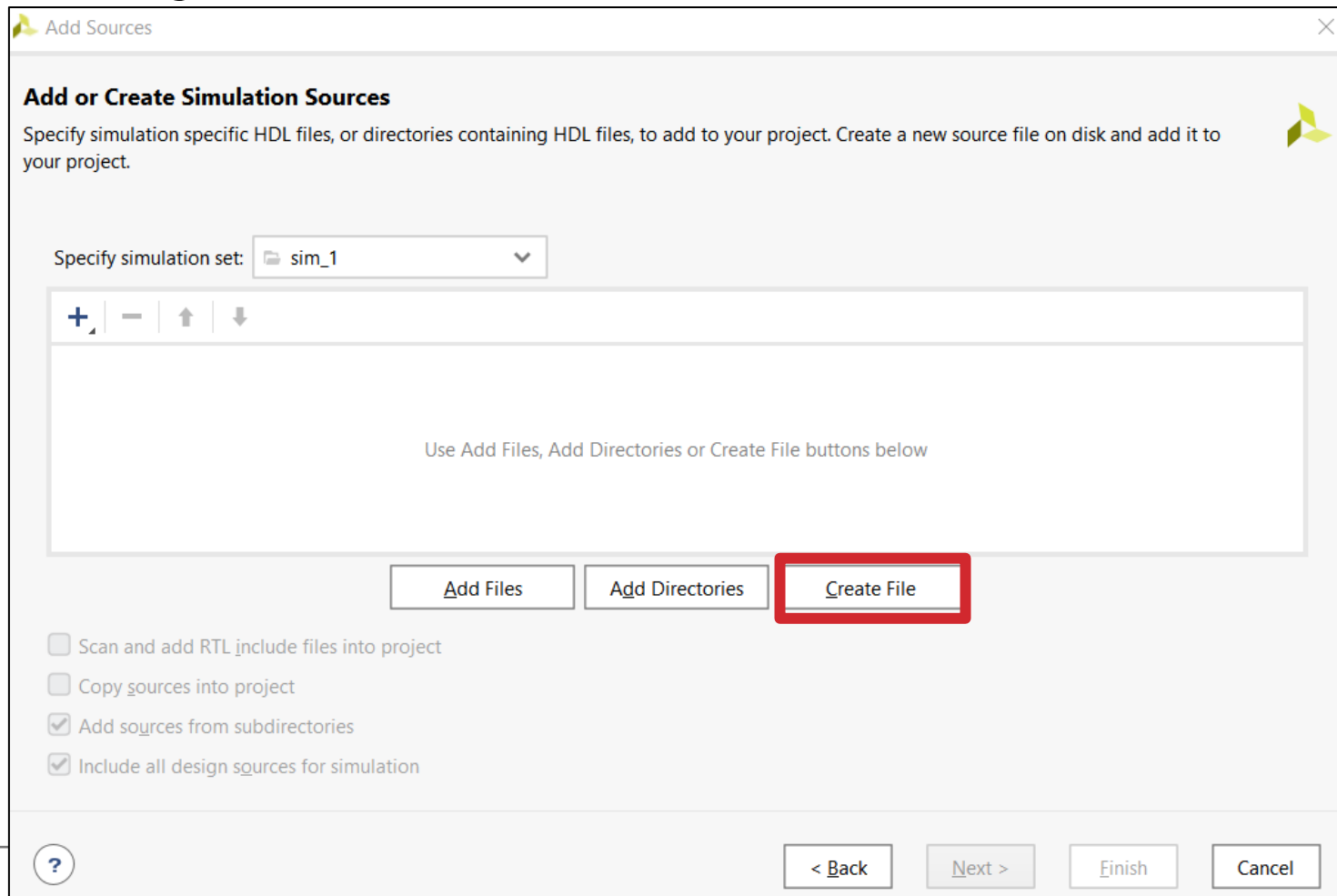
### □ Creating a Testbench





# DESIGN EXAMPLE

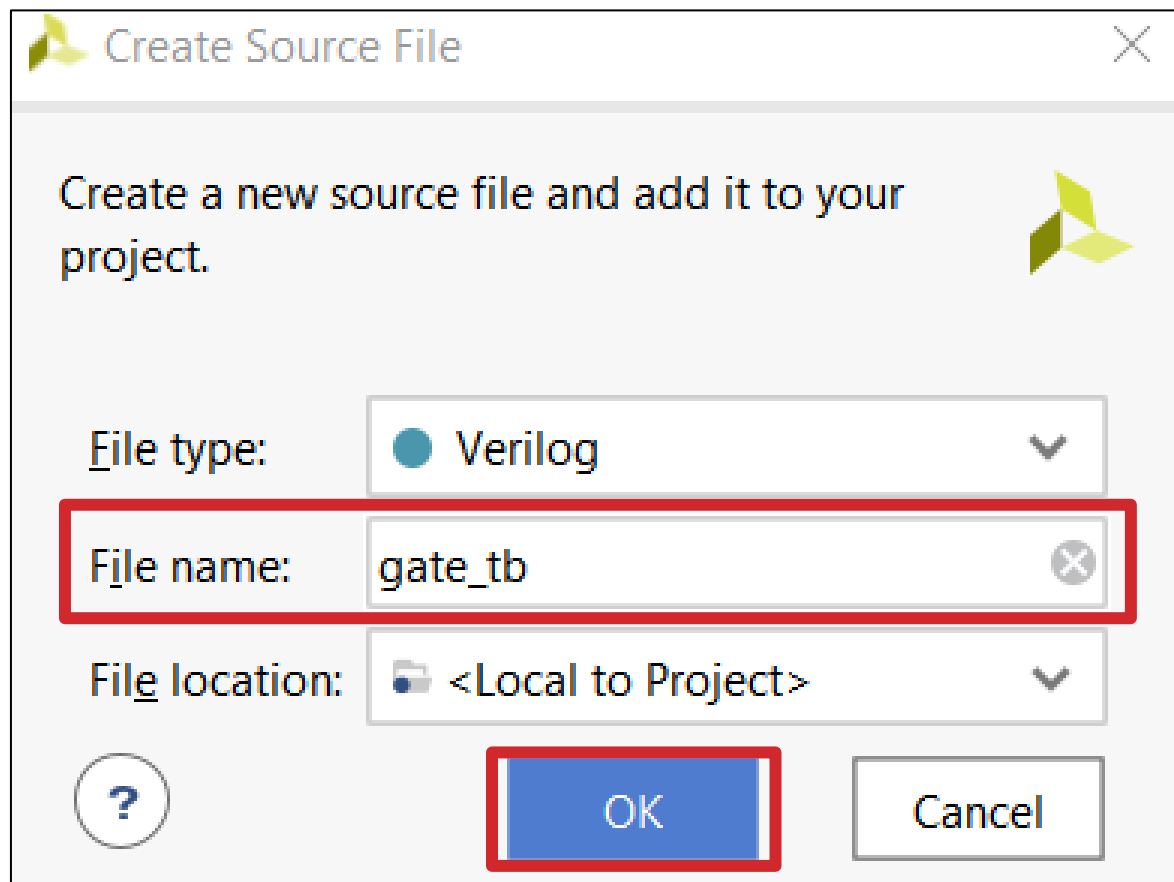
- ❖ Vivado project
- Creating a Testbench





# DESIGN EXAMPLE

- ❖ Vivado project
  - Creating a Testbench






# DESIGN EXAMPLE

## ❖ Vivado project





### □ Creating a Testbench


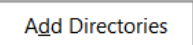
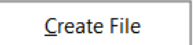
 Add Sources

**Add or Create Simulation Sources**


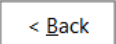
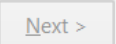
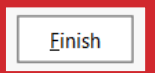
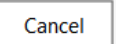
Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project.

Specify simulation set: sim\_1

	Index	Name	Library	Location
   	1	gate_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


    





# DESIGN EXAMPLE

- ❖ Vivado project
  - Creating a Testbench

 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

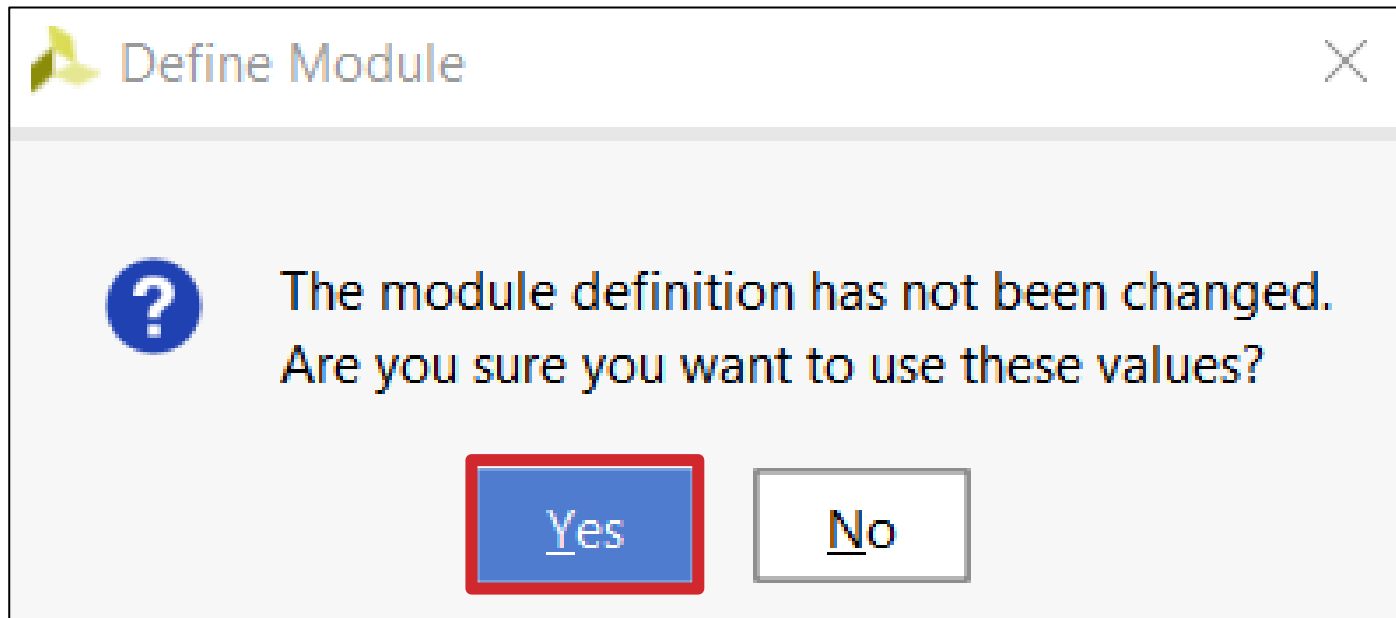
 OK Cancel





# DESIGN EXAMPLE

- ❖ Vivado project
  - Creating a Testbench



# DESIGN EXAMPLE

- ❖ Vivado project
  - Creating a Testbench

The screenshot displays the Vivado Project Manager interface for a project named 'GATES'. The 'Sources' pane on the left shows the project hierarchy: 'Design Sources (1)' containing 'gate (gate.v)', 'Constraints', 'Simulation Sources (2)' containing 'sim\_1 (2)' which includes 'gate (gate.v)' and 'gate\_tb (gate\_tb.v)', and 'Utility Sources'. The 'gate\_tb (gate\_tb.v)' file is highlighted with a red rectangle. Below the Sources pane is the 'Source File Property' window for 'gate\_tb.v', showing it is 'Enabled' and its 'Location' is 'D:/vivado\_projects/GATES'. The 'Project Summary' pane on the right shows the file path 'D:/vivado\_projects/GATES/GATES.srsrcs/sim\_1/new/gate\_tb.v' and a list of metadata fields. The 'Code' pane on the right shows the Verilog testbench code for 'gate\_tb'.

```
4 // Engineer:
5 //
6 // Create Date: 07/25/2023 03:23:45 PM
7 // Design Name:
8 // Module Name: gate_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 gate_tb(
24
25 );
26 endmodule
27
```



# DESIGN EXAMPLE

- ❖ Vivado project
  - Creating a Testbench

PROJECT MANAGER - GATES

Sources

- Design Sources (1)
  - gate (gate.v)
- Constraints
- Simulation Sources (1)
  - sim 1 (1)
    - gate\_tb (gate\_tb.v) (1)
      - uut : gate (gate.v)
- Utility Sources

Hierarchy Libraries Compi

Source File Property

gate\_tb.v

Enabled

Location: D:/vivado\_projects/G/

Project Summary

gate.v gate\_tb.v

D:/vivado\_projects/GATES/GATES.srscs/sim\_1/new/gate\_tb.v

```
23 module gate_tb();
24     //Inputs
25     reg A;
26     reg B;
27     //Outputs
28     wire X;
29     wire Y;
30     wire Z;
31     //Instantiate the Unit Under Test (UUT)
32     gate uut (
33         .A(A),
34         .B(B),
35         .X(X),
36         .Y(Y),
37         .Z(Z)
38     );
39     initial begin
40         //Specify input stimulus
41         A = 0; B = 0; #100; //Wait 100ns
42         A = 0; B = 1; #100; //Wait 100ns
43         A = 1; B = 0; #100; //Wait 100ns
44         A = 1; B = 1; #100; //Wait 100ns
45     end
46 endmodule
```

A	B	X	Y	Z
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



# DESIGN EXAMPLE

- ❖ Vivado project
- Running Simulation

The screenshot displays the Vivado IDE interface for a project named "GATES". The "Flow Navigator" on the left shows the project structure, with the "SIMULATION" tab highlighted by a red box. The "PROJECT MANAGER - GATES" window in the center shows the "Sources" tab, where the "gate\_tb (gate\_tb.v) (1)" file is selected. The "Source File Property" window at the bottom shows the "gate\_tb.v" file with the "Enabled" checkbox checked. The "Project Summary" window on the right shows the "gate\_tb.v" file, which contains the following code:

```
23 module gate_tb();
24     //Inputs
25     reg A;
26     reg B;
27     //Outputs
28     wire X;
29     wire Y;
30     wire Z;
31     //Instantiate the Unit Under Test (UUT)
32     gate uut(
33         .A(A),
34         .B(B),
35         .X(X),
36         .Y(Y),
37         .Z(Z)
38     );
39     initial begin
40         //Specify input stimulus
41         A = 0; B = 0; #100; //Wait 100ns
42         A = 0; B = 1; #100; //Wait 100ns
43         A = 1; B = 0; #100; //Wait 100ns
44         A = 1; B = 1; #100; //Wait 100ns
45     end
46 endmodule
```

# DESIGN EXAMPLE

- ❖ Vivado project
- Running Simulation

The screenshot displays the Vivado IDE interface for a project named 'GATES'. The 'Flow Navigator' on the left shows the 'SIMULATION' tab selected, with the 'Run Simulation' option highlighted. A context menu is open, showing options like 'Run Behavioral Simulation', 'Run Post-Synthesis Functional Simulation', and 'Run Post-Synthesis Timing Simulation'. The 'Sources' window in the center shows the project hierarchy, including 'Design Sources' (gate.v), 'Constraints', 'Simulation Sources' (sim\_1), and 'Utility Sources'. The 'Project Summary' window on the right shows the file path 'D:/vivado\_projects/GATES/GATES.srsrcs/sim\_1/new/gate\_tb.v' and the Verilog code for the testbench.

```
module gate_tb();  
    //Inputs  
    reg A;  
    reg B;  
    //Outputs  
    wire X;  
    wire Y;  
    wire Z;  
    //Instantiate the Unit Under Test (U  
    gate uut (  
        .A(A),  
        .B(B),  
        .X(X),  
        .Y(Y),  
        .Z(Z)  
    );  
    initial begin  
        //Specify input stimulus  
        A = 0; B = 0; #100; //Wait 100ns  
        A = 0; B = 1; #100; //Wait 100ns  
        A = 1; B = 0; #100; //Wait 100ns  
        A = 1; B = 1; #100; //Wait 100ns  
    end
```

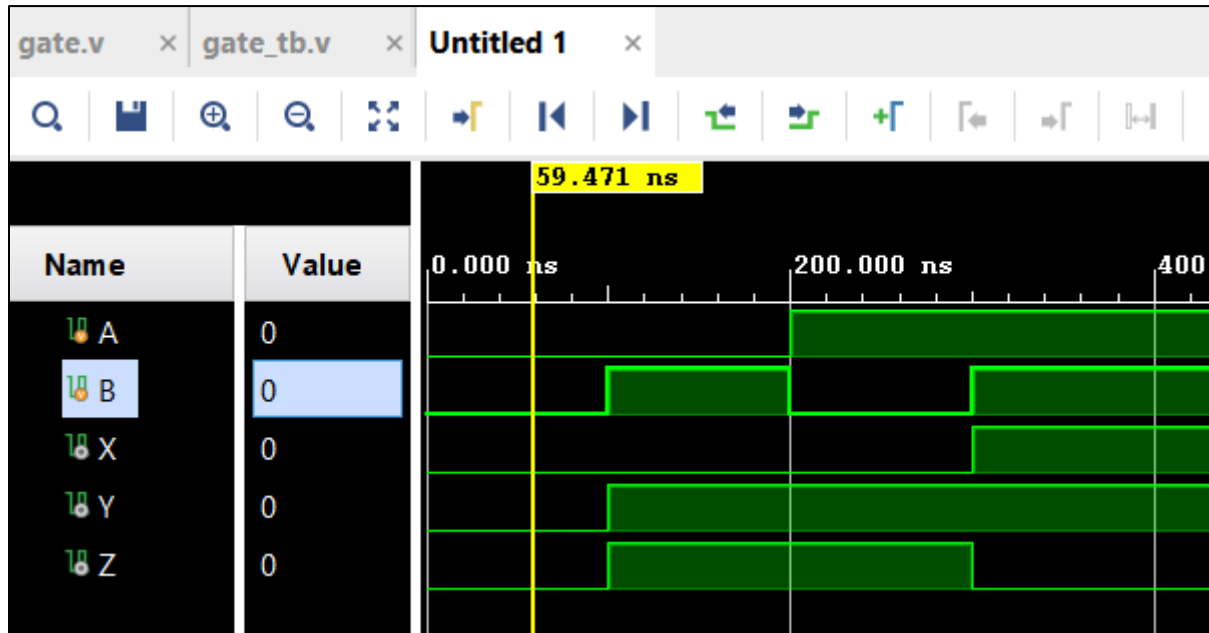


# DESIGN EXAMPLE

- ❖ Vivado project
  - Running Simulation

```
//Specify input stimulus  
A = 0; B = 0; #100;//Wait 100ns  
A = 0; B = 1; #100;//Wait 100ns  
A = 1; B = 0; #100;//Wait 100ns  
A = 1; B = 1; #100;//Wait 100ns
```

A	B	X	Y	Z
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



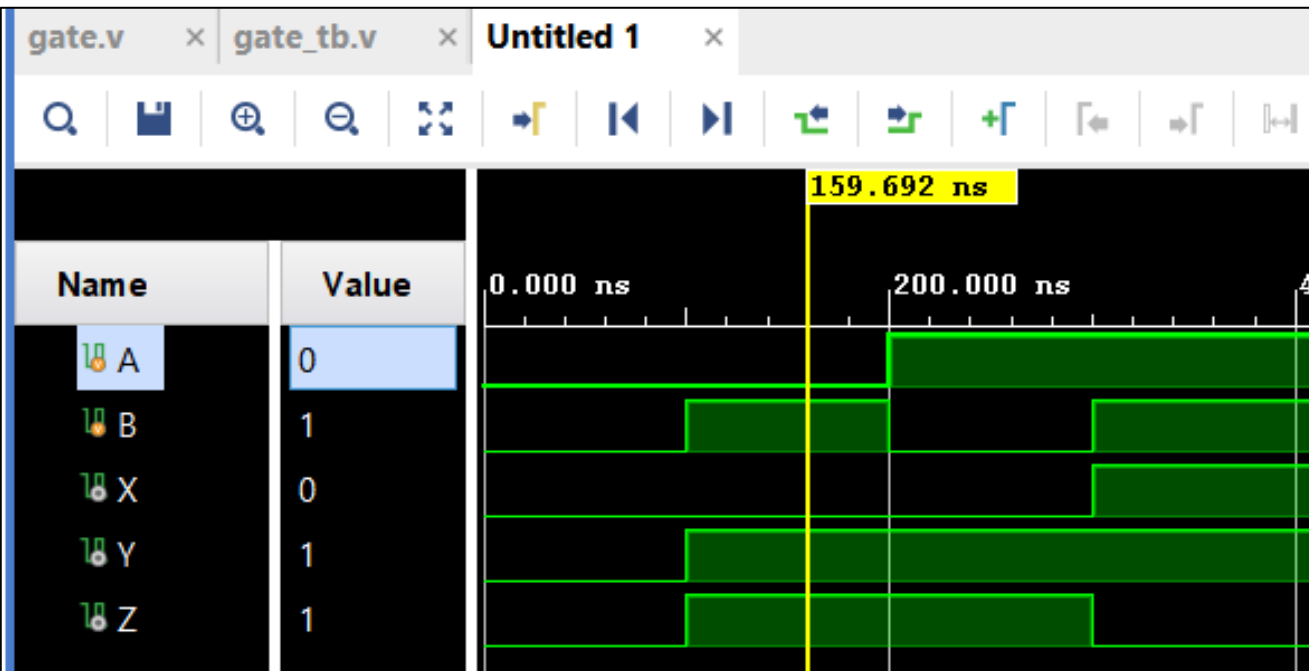




# DESIGN EXAMPLE

- ❖ Vivado project
  - Running Simulation

```
//Specify input stimulus  
A = 0; B = 0; #100;//Wait 100ns  
A = 0; B = 1; #100;//Wait 100ns  
A = 1; B = 0; #100;//Wait 100ns  
A = 1; B = 1; #100;//Wait 100ns
```



A	B	X	Y	Z
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



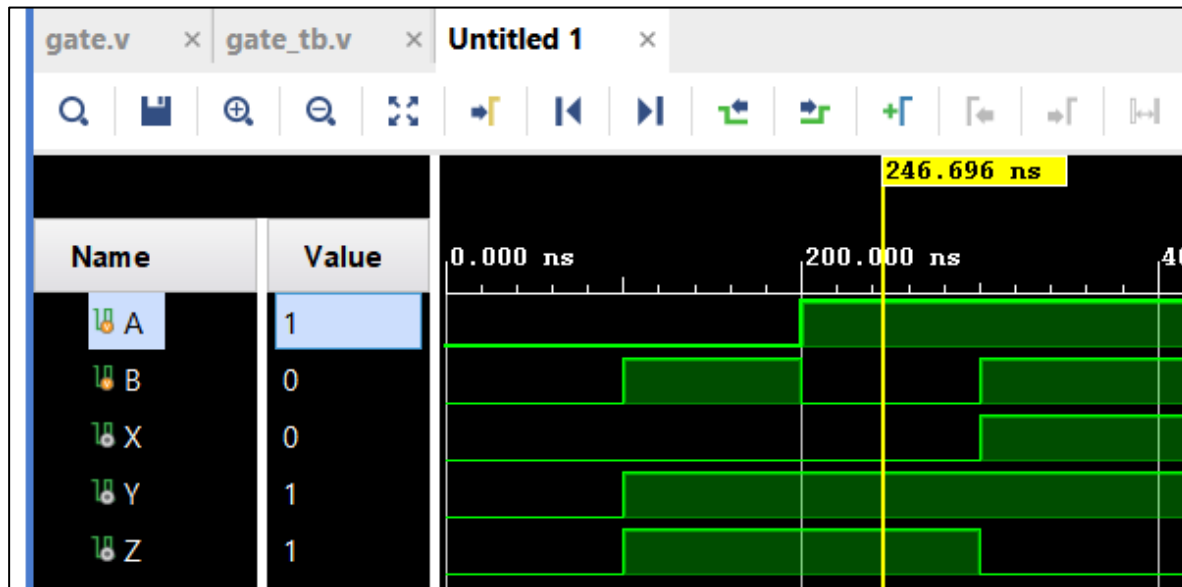


# DESIGN EXAMPLE

- ❖ Vivado project
  - Running Simulation

```
//Specify input stimulus  
A = 0; B = 0; #100;//Wait 100ns  
A = 0; B = 1; #100;//Wait 100ns  
A = 1; B = 0; #100;//Wait 100ns  
A = 1; B = 1; #100;//Wait 100ns
```

A	B	X	Y	Z
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

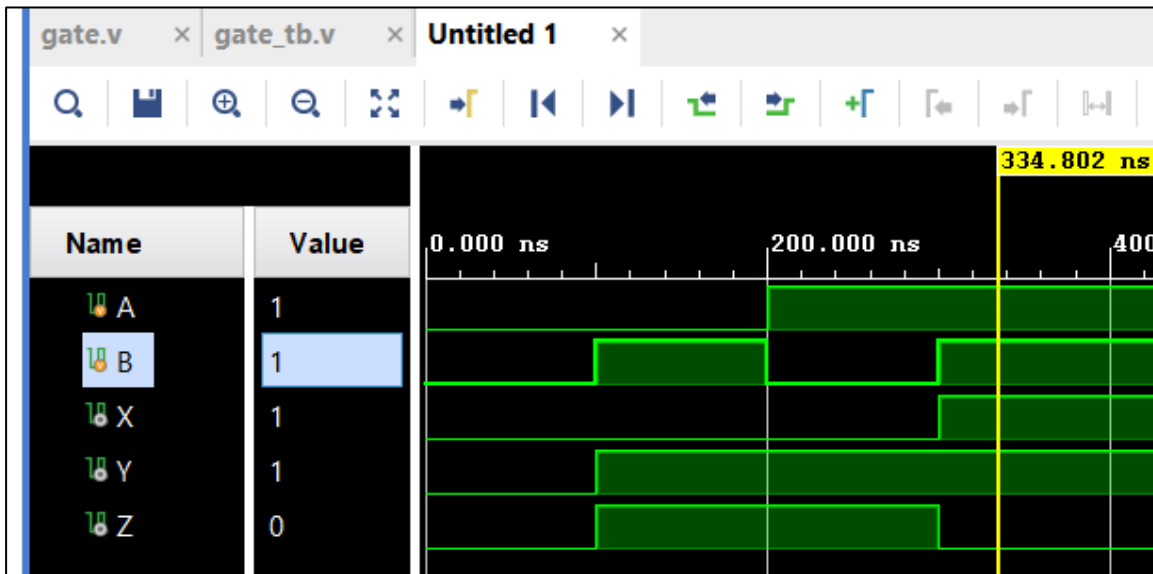




# DESIGN EXAMPLE

- ❖ Vivado project
  - Running Simulation

```
//Specify input stimulus  
A = 0; B = 0; #100;//Wait 100ns  
A = 0; B = 1; #100;//Wait 100ns  
A = 1; B = 0; #100;//Wait 100ns  
A = 1; B = 1; #100;//Wait 100ns
```



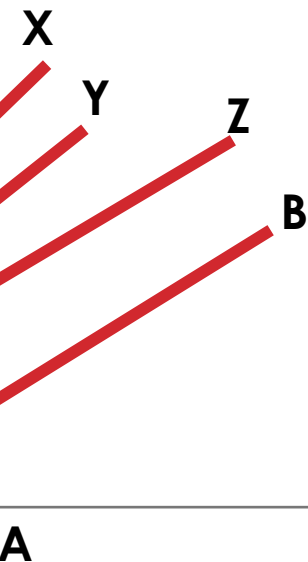
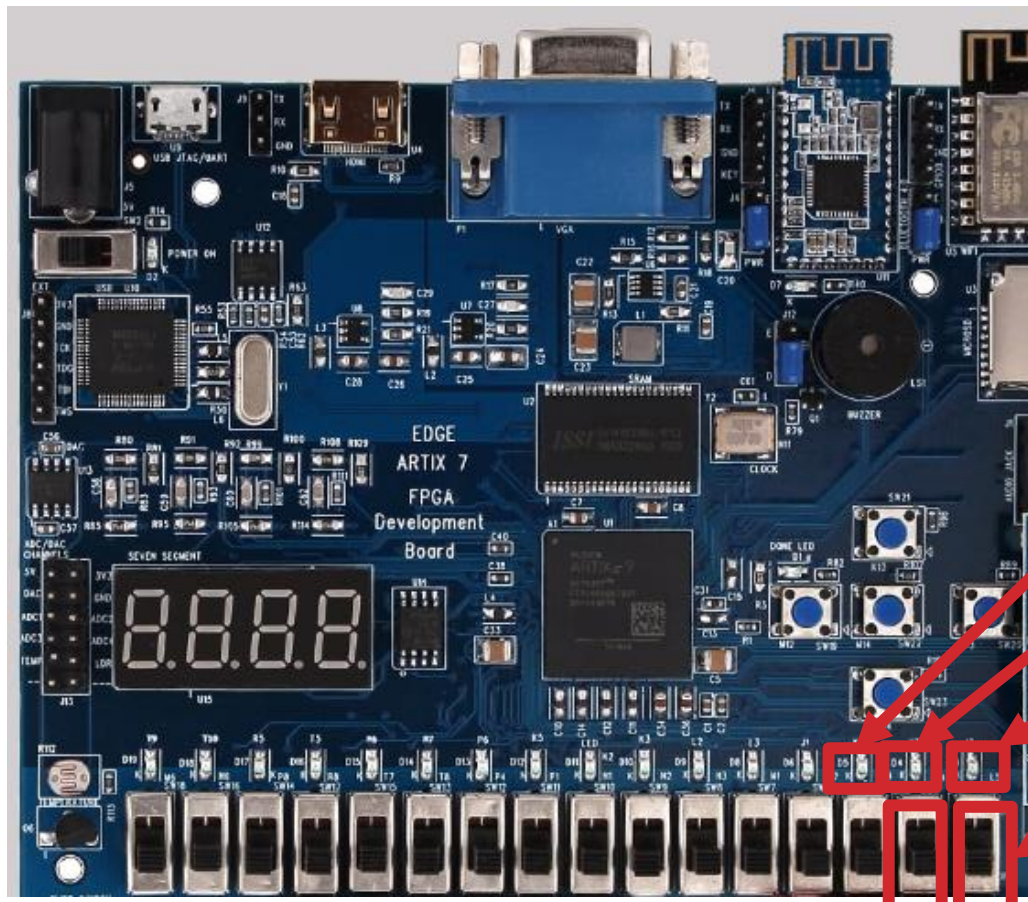
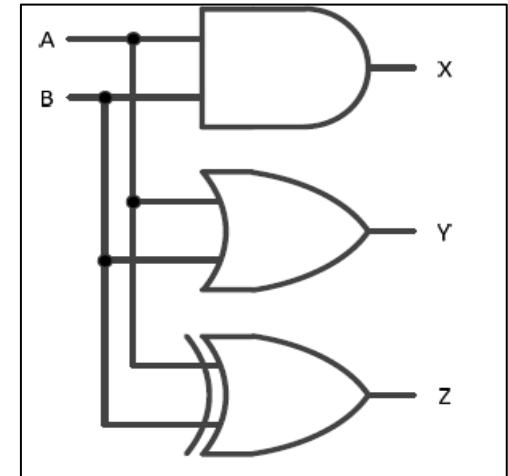
A	B	X	Y	Z
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0





# DESIGN EXAMPLE

- ❖ Vivado project
- ❑ FPGA programming (mapping)





# DESIGN EXAMPLE

- ❖ Vivado project
- FPGA programming (mapping)

```
# Switches
set_property -dict { PACKAGE_PIN L5 IOSTANDARD LVCMOS33 } [get_ports { sw[0] }];#LSB
set_property -dict { PACKAGE_PIN L4 IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];
set_property -dict { PACKAGE_PIN M4 IOSTANDARD LVCMOS33 } [get_ports { sw[2] }];
set_property -dict { PACKAGE_PIN M2 IOSTANDARD LVCMOS33 } [get_ports { sw[3] }];
set_property -dict { PACKAGE_PIN M1 IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];
set_property -dict { PACKAGE_PIN N3 IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];
set_property -dict { PACKAGE_PIN N2 IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];
set_property -dict { PACKAGE_PIN N1 IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];
set_property -dict { PACKAGE_PIN P1 IOSTANDARD LVCMOS33 } [get_ports { sw[8] }];
set_property -dict { PACKAGE_PIN P4 IOSTANDARD LVCMOS33 } [get_ports { sw[9] }];
set_property -dict { PACKAGE_PIN T8 IOSTANDARD LVCMOS33 } [get_ports { sw[10] }];
set_property -dict { PACKAGE_PIN R8 IOSTANDARD LVCMOS33 } [get_ports { sw[11] }];
set_property -dict { PACKAGE_PIN N6 IOSTANDARD LVCMOS33 } [get_ports { sw[12] }];
set_property -dict { PACKAGE_PIN T7 IOSTANDARD LVCMOS33 } [get_ports { sw[13] }];
set_property -dict { PACKAGE_PIN P8 IOSTANDARD LVCMOS33 } [get_ports { sw[14] }];
set_property -dict { PACKAGE_PIN M6 IOSTANDARD LVCMOS33 } [get_ports { sw[15] }];#MSB
```

Port Name	Pin Number
A	L4
B	L5
X	J1
Y	H3
Z	J3

```
# LEDs
set_property -dict { PACKAGE_PIN J3 IOSTANDARD LVCMOS33 } [get_ports { led[0] }];#LSB
set_property -dict { PACKAGE_PIN H3 IOSTANDARD LVCMOS33 } [get_ports { led[1] }];
set_property -dict { PACKAGE_PIN J1 IOSTANDARD LVCMOS33 } [get_ports { led[2] }];
set_property -dict { PACKAGE_PIN K1 IOSTANDARD LVCMOS33 } [get_ports { led[3] }];
set_property -dict { PACKAGE_PIN L3 IOSTANDARD LVCMOS33 } [get_ports { led[4] }];
set_property -dict { PACKAGE_PIN L2 IOSTANDARD LVCMOS33 } [get_ports { led[5] }];
set_property -dict { PACKAGE_PIN K3 IOSTANDARD LVCMOS33 } [get_ports { led[6] }];
set_property -dict { PACKAGE_PIN K2 IOSTANDARD LVCMOS33 } [get_ports { led[7] }];
set_property -dict { PACKAGE_PIN K5 IOSTANDARD LVCMOS33 } [get_ports { led[8] }];
set_property -dict { PACKAGE_PIN P6 IOSTANDARD LVCMOS33 } [get_ports { led[9] }];
set_property -dict { PACKAGE_PIN R7 IOSTANDARD LVCMOS33 } [get_ports { led[10] }];
set_property -dict { PACKAGE_PIN R6 IOSTANDARD LVCMOS33 } [get_ports { led[11] }];
set_property -dict { PACKAGE_PIN T5 IOSTANDARD LVCMOS33 } [get_ports { led[12] }];
set_property -dict { PACKAGE_PIN R5 IOSTANDARD LVCMOS33 } [get_ports { led[13] }];
set_property -dict { PACKAGE_PIN T10 IOSTANDARD LVCMOS33 } [get_ports { led[14] }];
set_property -dict { PACKAGE_PIN T9 IOSTANDARD LVCMOS33 } [get_ports { led[15] }];#MSB
```



# DESIGN EXAMPLE

## ❖ Vivado project

- ❑ FPGA programming (mapping) – Create a constraint file

**PROJECT MANAGER - GATES**

**Sources**

- Design Sources (1)
  - gate (gate.v)
- Constraints
- Simulation Sources (1)
  - sim\_1 (1)
    - gate\_tb (gate\_tb.v) (1)
    - uut : gate (gate.v)
- Utility Sources

**Source File Property**

gate.v

☒ Enabled

Location: D:/vivado\_projects/G...

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

D:/vivado\_projects/GATES/GATES.srsrcs/sources\_1/new/gate.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 gate(A, B, X, Y, Z);
24     //Input and output declaration
25     input A, B;
26     output X, Y, Z;
27     //Gate Declaration
28     assign X = A & B;
29     assign Y = A | B;
30     assign Z = A ^ B;
31 endmodule
32
```



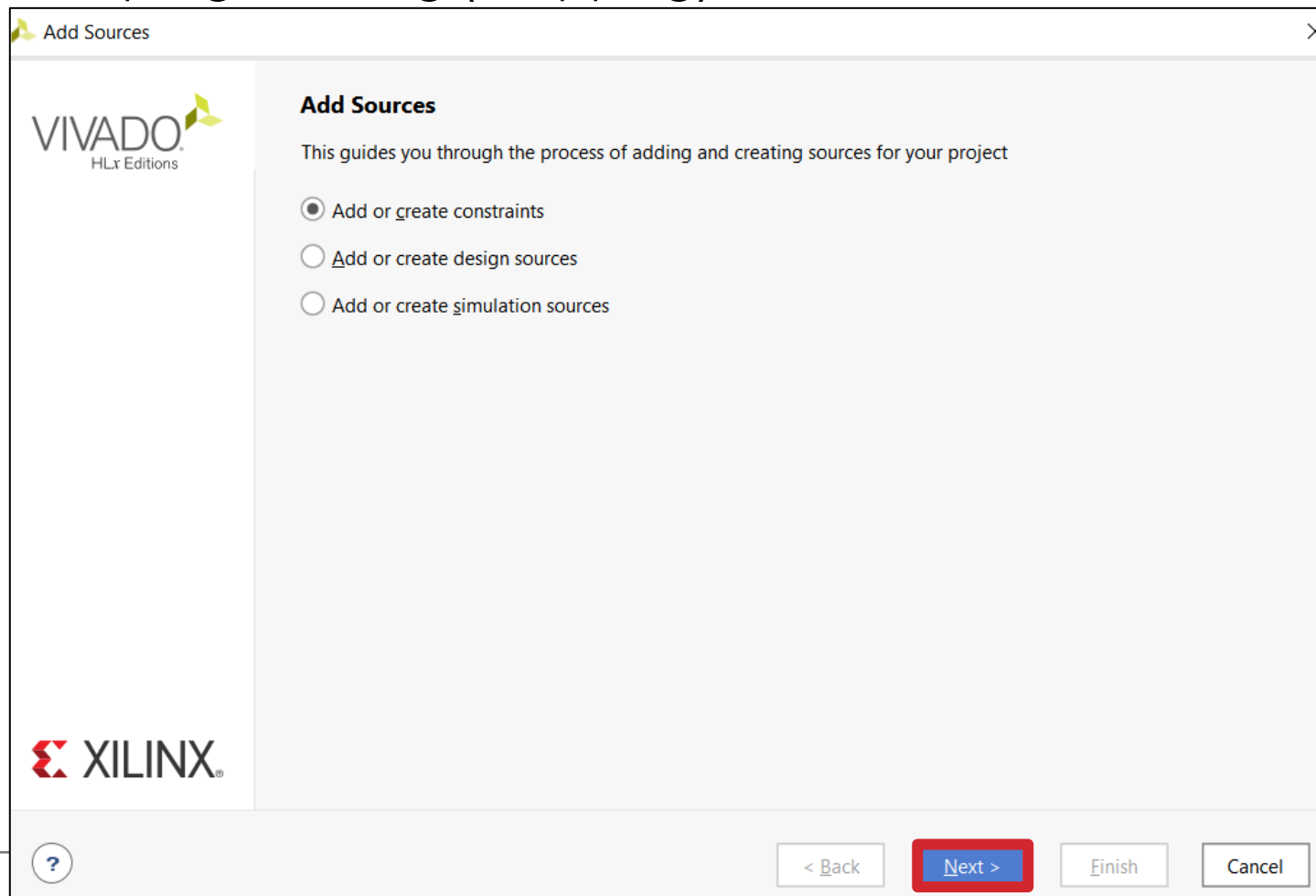




# DESIGN EXAMPLE

## ❖ Vivado project

- ❑ FPGA programming (mapping) – Create a constraint file

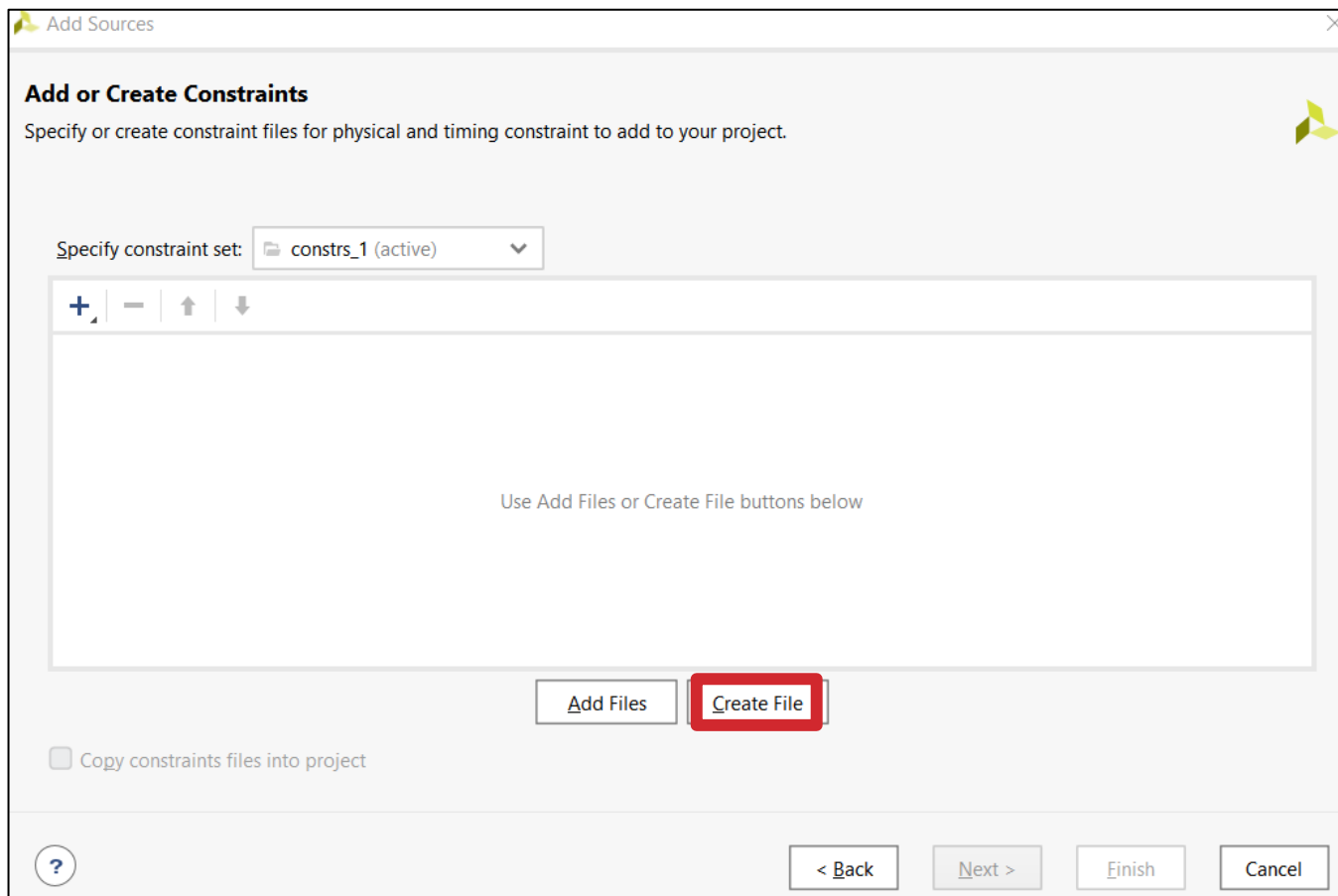




# DESIGN EXAMPLE

## ❖ Vivado project

- ❑ FPGA programming (mapping) – Create a constraint file

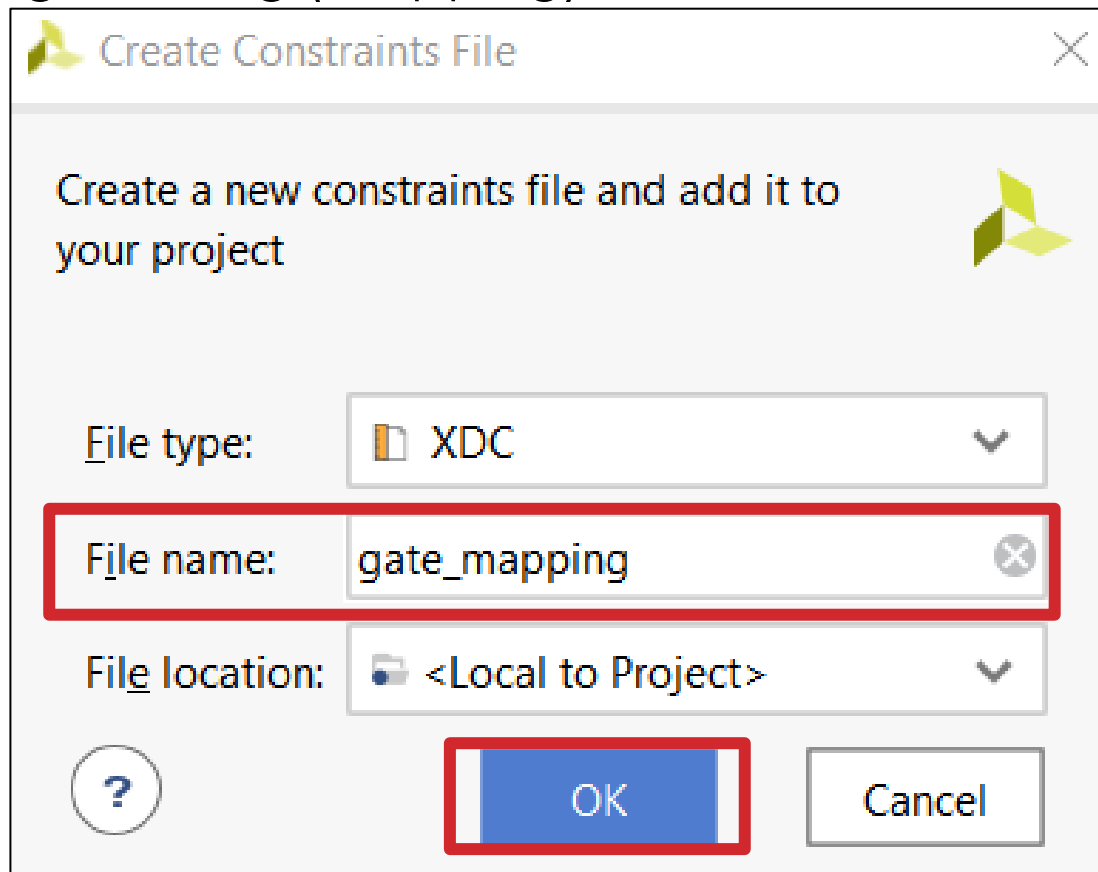






# DESIGN EXAMPLE

- ❖ Vivado project
  - FPGA programming (mapping) – Create a constraint file

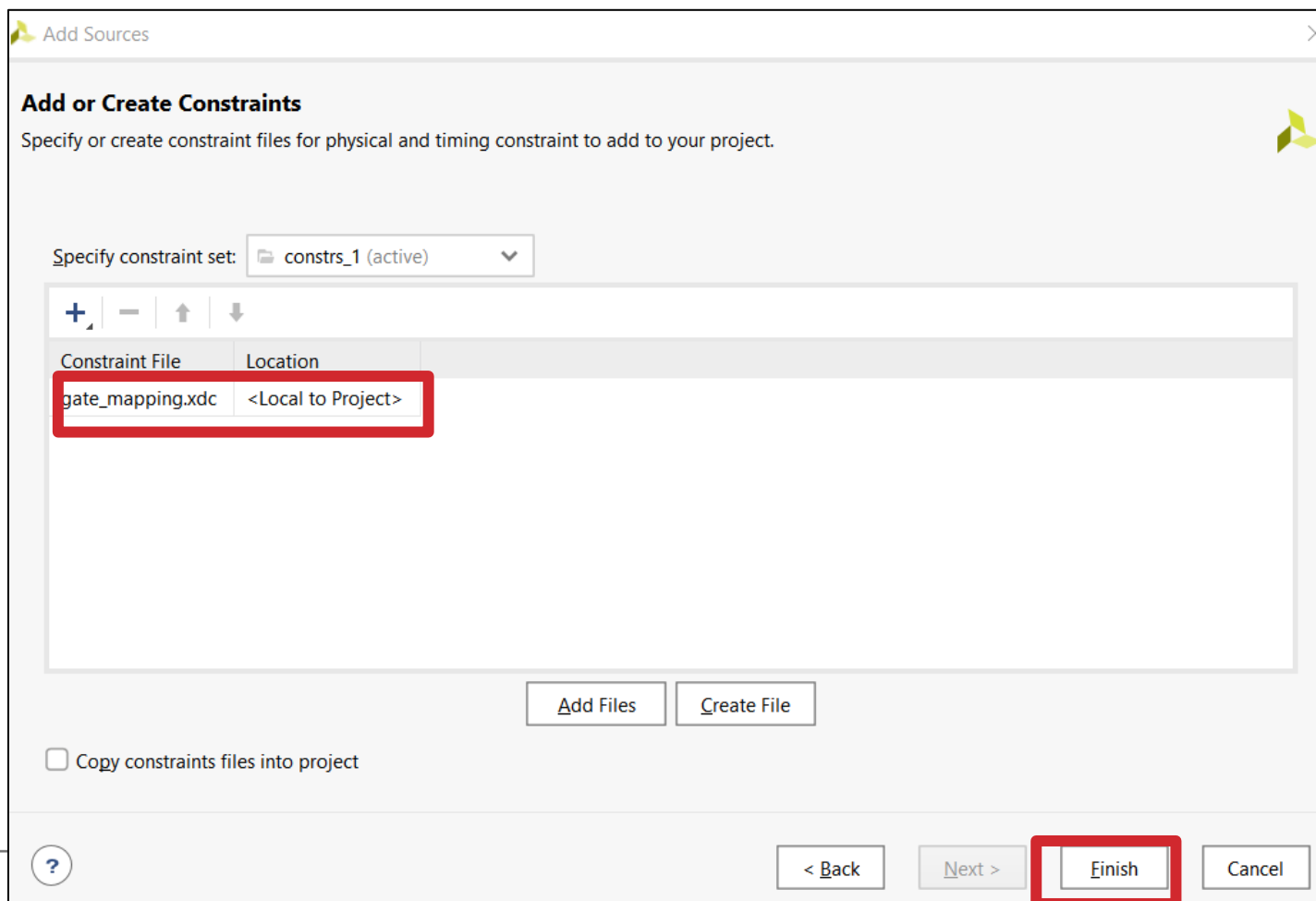




# DESIGN EXAMPLE

## ❖ Vivado project

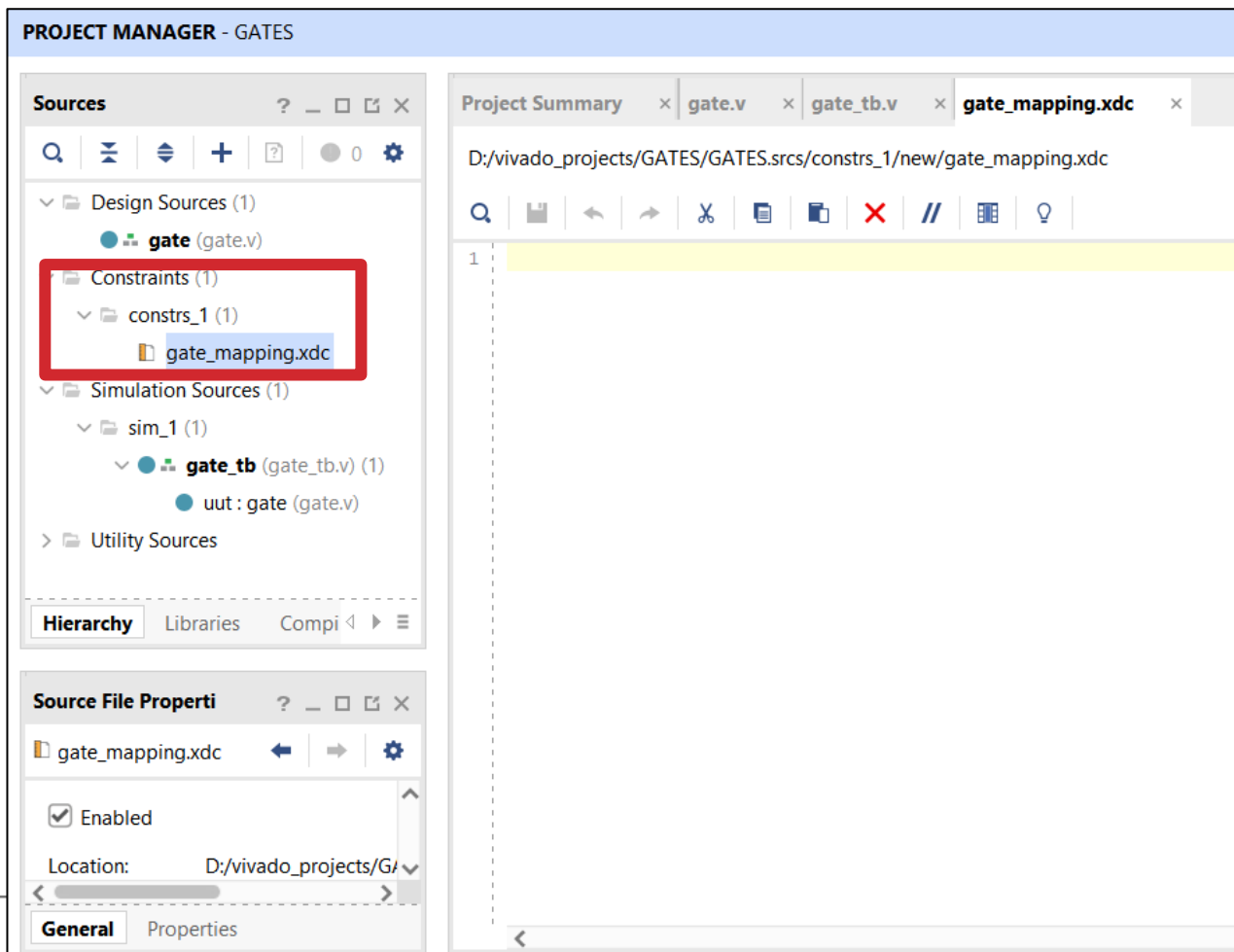
- ❑ FPGA programming (mapping) – Create a constraint file



# DESIGN EXAMPLE

## ❖ Vivado project

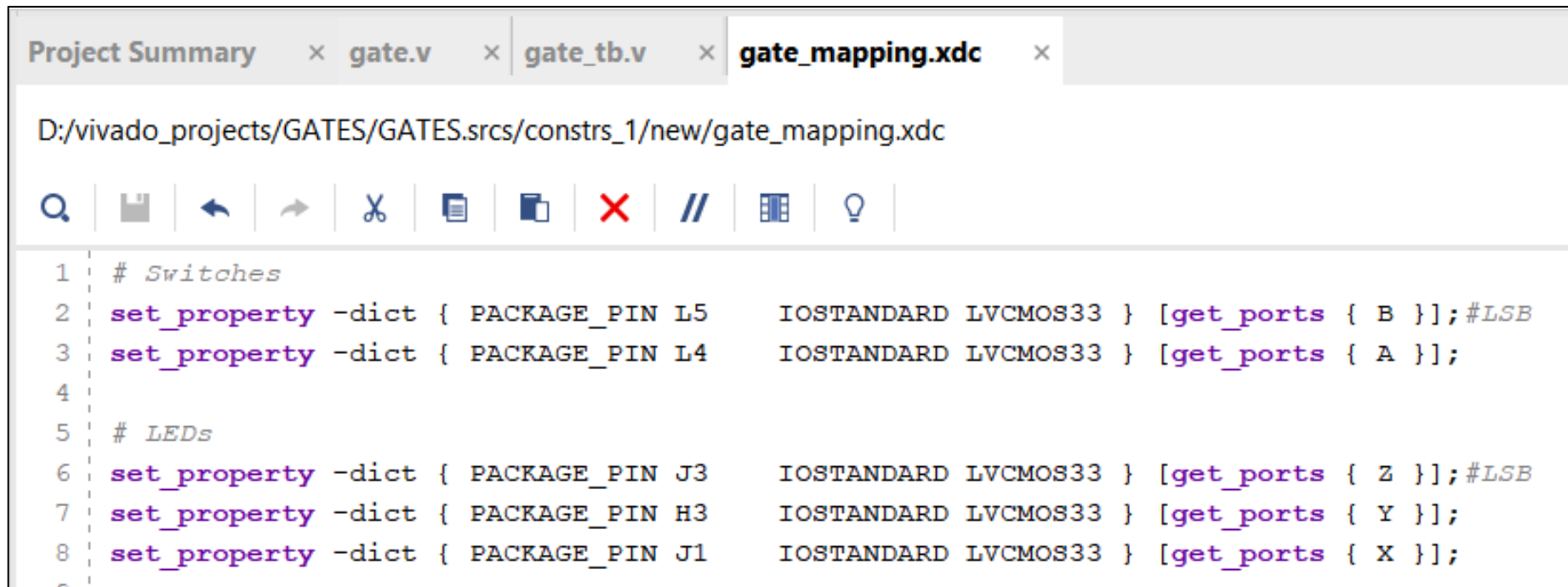
- ❑ FPGA programming (mapping) – Create a constraint file



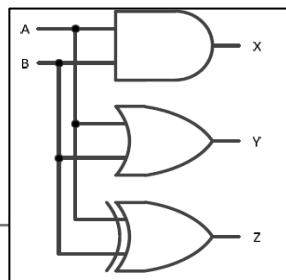
# DESIGN EXAMPLE

## ❖ Vivado project

- ❑ FPGA programming (mapping) – Create a constraint file



```
1  # Switches
2  set_property -dict { PACKAGE_PIN L5      IOSTANDARD LVCOS33 } [get_ports { B }];#LSB
3  set_property -dict { PACKAGE_PIN L4      IOSTANDARD LVCOS33 } [get_ports { A }];
4
5  # LEDs
6  set_property -dict { PACKAGE_PIN J3      IOSTANDARD LVCOS33 } [get_ports { Z }];#LSB
7  set_property -dict { PACKAGE_PIN H3      IOSTANDARD LVCOS33 } [get_ports { Y }];
8  set_property -dict { PACKAGE_PIN J1      IOSTANDARD LVCOS33 } [get_ports { X }];
```



Port Name	Pin Number
A	L4
B	L5
X	J1
Y	H3
Z	J3



# DESIGN EXAMPLE

## ❖ Vivado project

### ❑ FPGA programming (mapping) – Implement Design

The screenshot displays the Vivado IDE interface for a project named "xc7a35tftg256-1". The left sidebar shows the "Flow Navigator" with the "IMPLEMENTATION" tab selected and highlighted with a red box. The "Run Implementation" button is also visible. The main workspace is divided into three panes: "Sources" (showing "gate" with "Nets (10)" and "Leaf Cells (8)"), "Implementation Run Pro" (showing "impl\_1" with "Name: impl\_1"), and "Project Summary" (showing the "gate.v" file). The "Project Summary" pane displays the Verilog code for the "gate" module, which includes input/output declarations and logic assignments for X, Y, and Z. The bottom status bar shows the "Design Timing Summary" and "General Information" tabs.

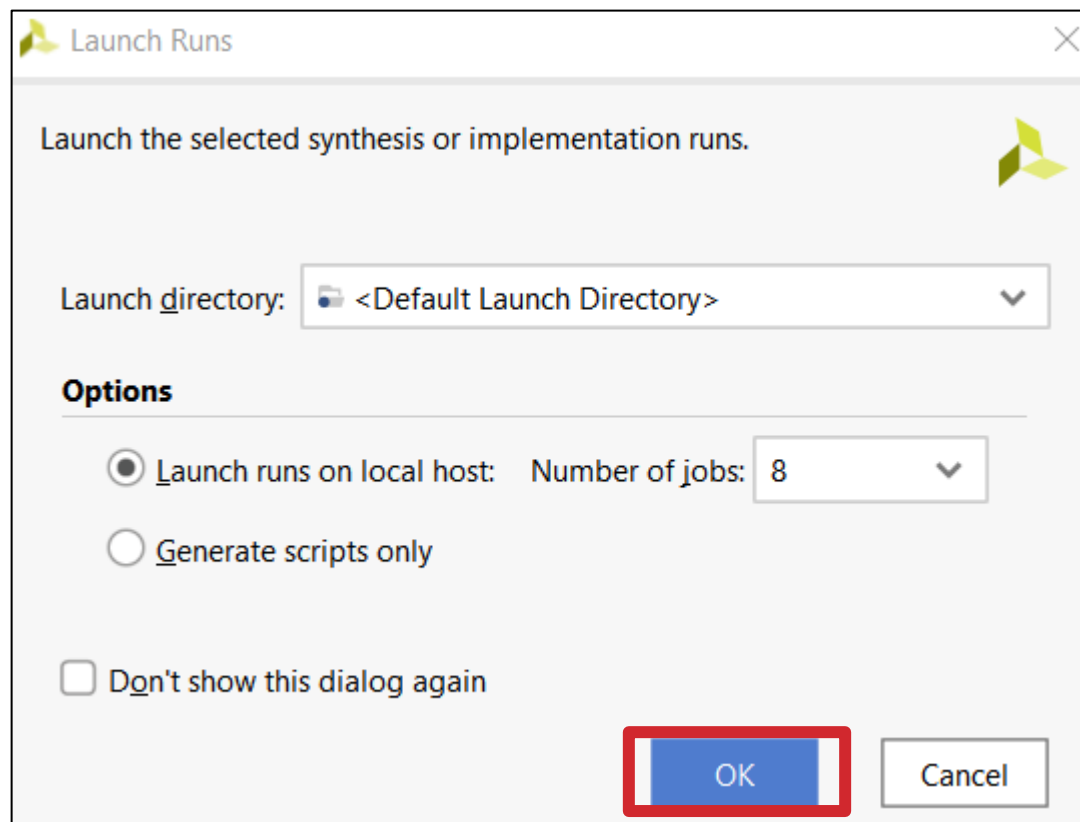
```
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////
23 module gate(A, B, X, Y, Z);
24     //Input and output declaration
25     input A, B;
26     output X, Y, Z;
27     //Gate Declaration
28     assign X = A & B;
29     assign Y = A | B;
30     assign Z = A ^ B;
31 endmodule
32
```





# DESIGN EXAMPLE

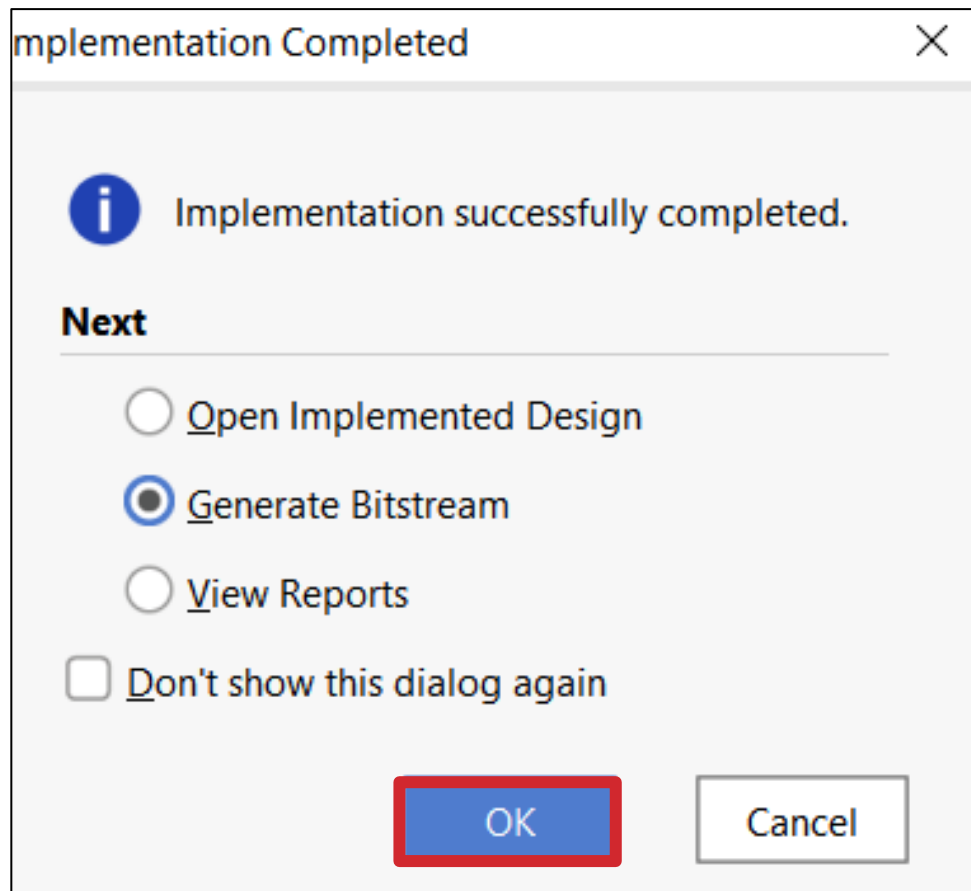
- ❖ Vivado project
  - FPGA programming (mapping) – Implement Design





# DESIGN EXAMPLE

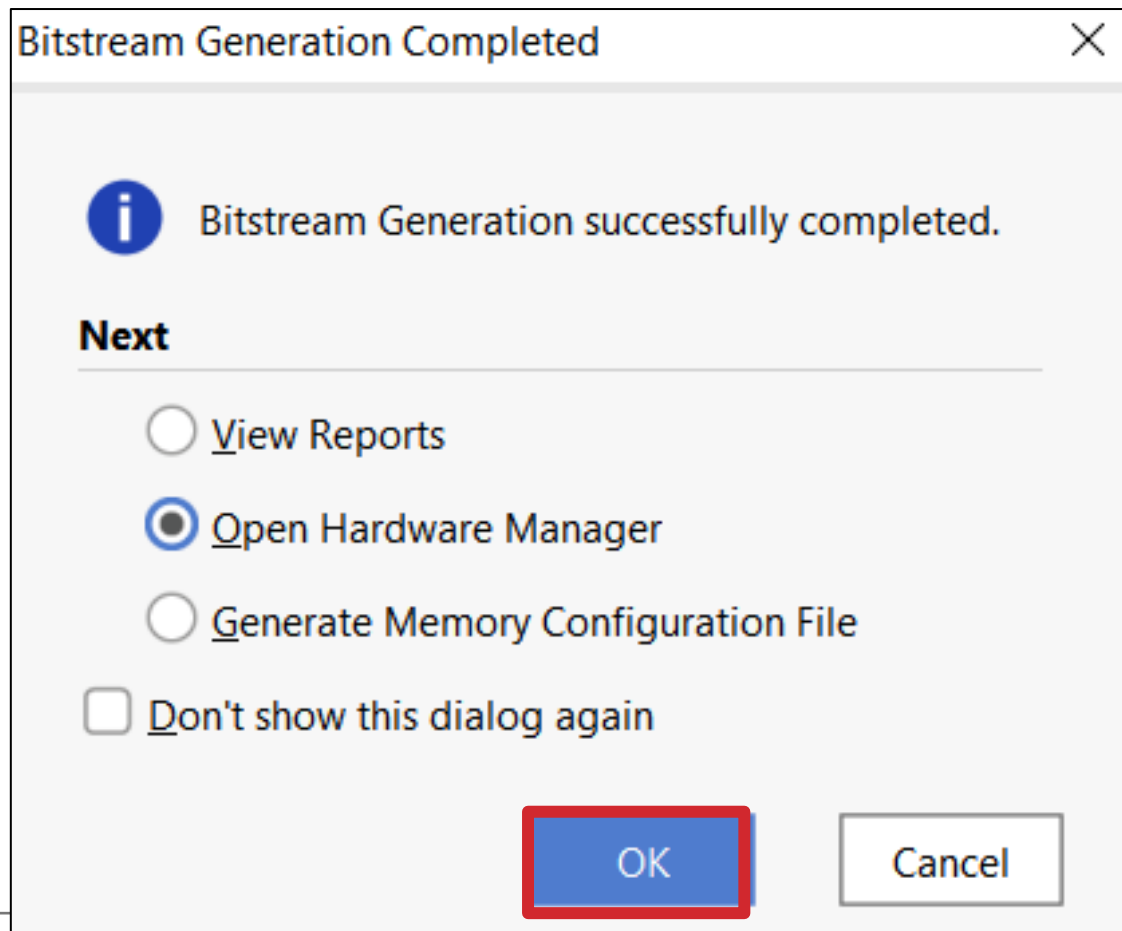
- ❖ Vivado project
  - FPGA programming (mapping) – Generate Programming File





# DESIGN EXAMPLE

- ❖ Vivado project
  - FPGA programming (mapping) – Generate Programming File





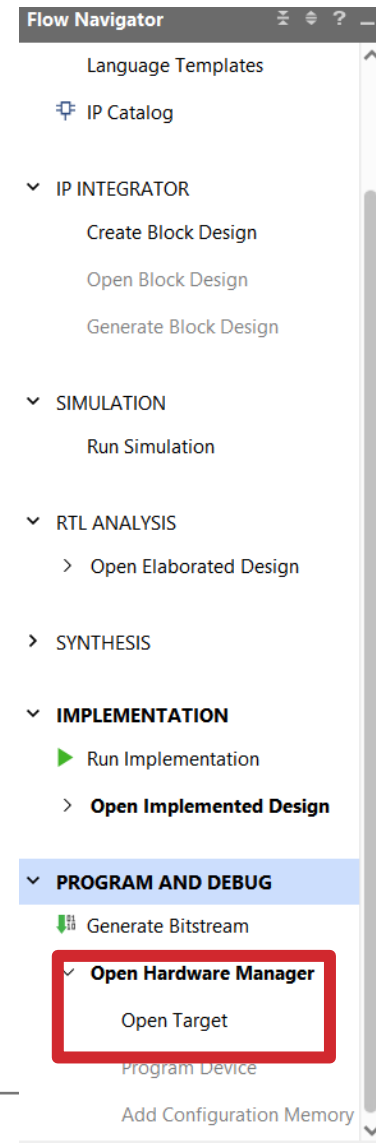


# DESIGN EXAMPLE

## ❖ Vivado project

### □ FPGA programming

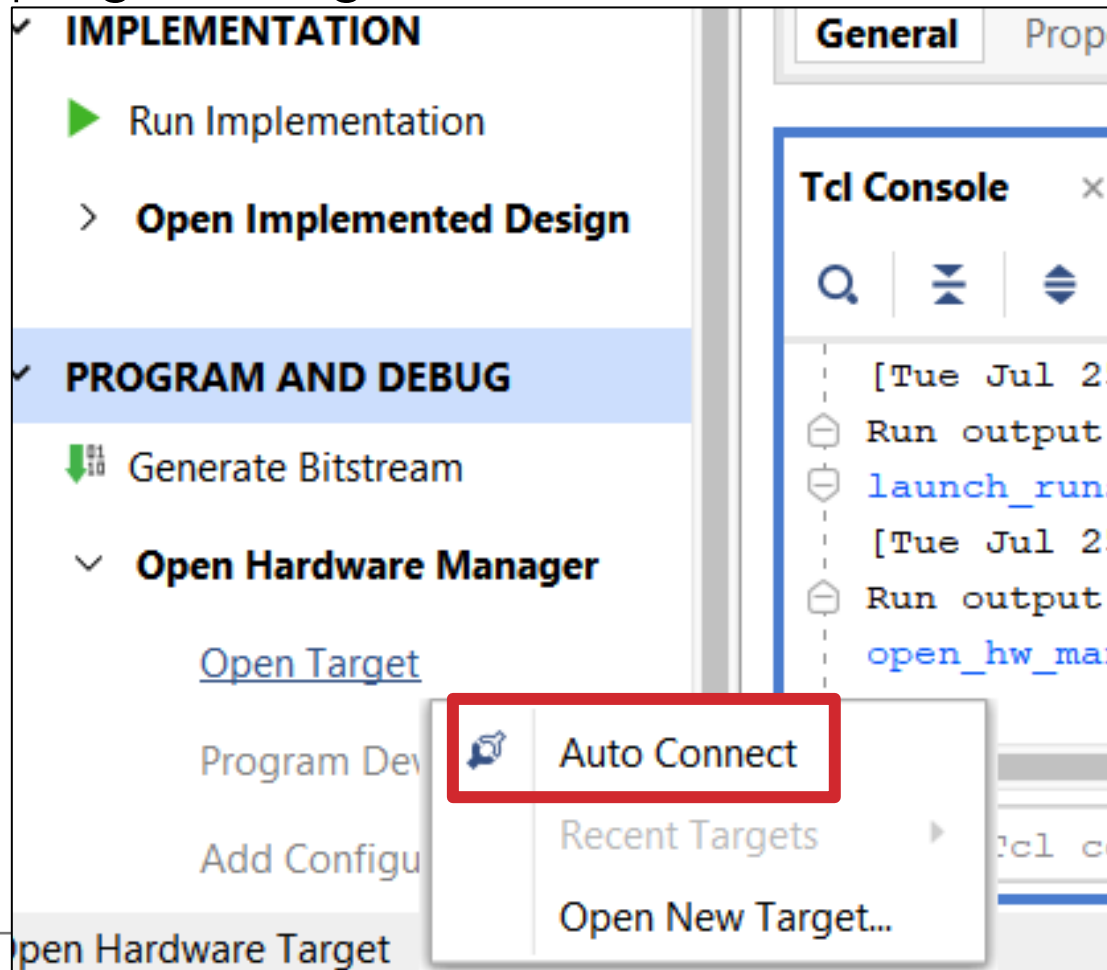
- Connect the FPGA board to the computer and click on Open Target





# DESIGN EXAMPLE

- ❖ Vivado project
  - FPGA programming



# DESIGN EXAMPLE

- ❖ Vivado project
- FPGA programming

The screenshot displays the Vivado Hardware Manager interface. At the top, a green banner indicates "There are no debug cores" with buttons for "Program device" (highlighted with a red box) and "Refresh device". Below this, the "Hardware" panel shows a tree structure:

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319	Open
xc7a35t_0 (1)	Not programmed
XADC (System Monit	

The "xc7a35t\_0 (1)" entry is highlighted with a blue background and a red border. Below the hardware tree, the "Hardware Device Properties" panel shows details for "xc7a35t\_0", including its name and tabs for "General" and "Properties".

On the right, the "gate.v" file is open, showing a Verilog module definition:

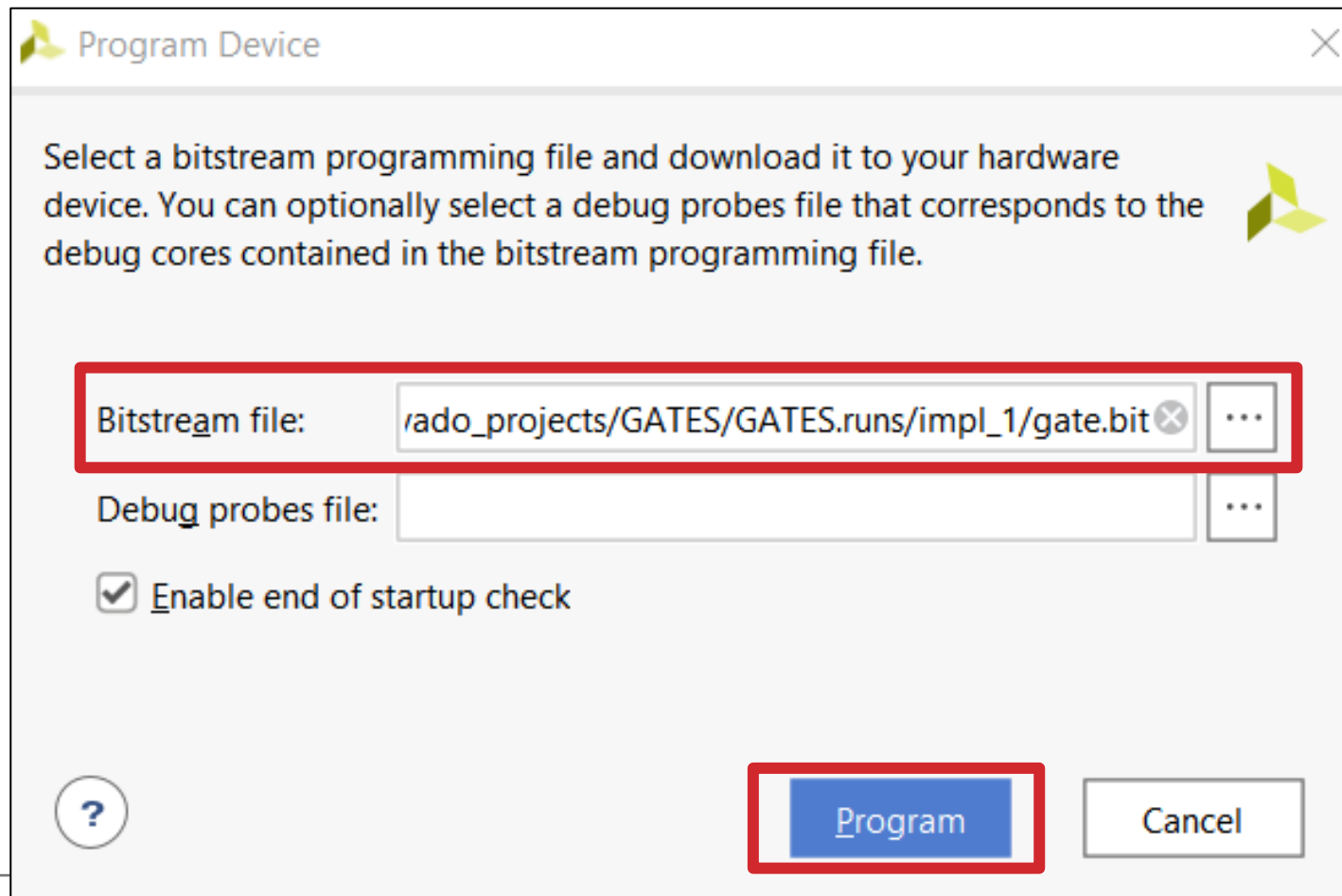
```
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////
21
22
23 module gate(A, B, X, Y, Z);
24     //Input and output declaration
25     input A, B;
```





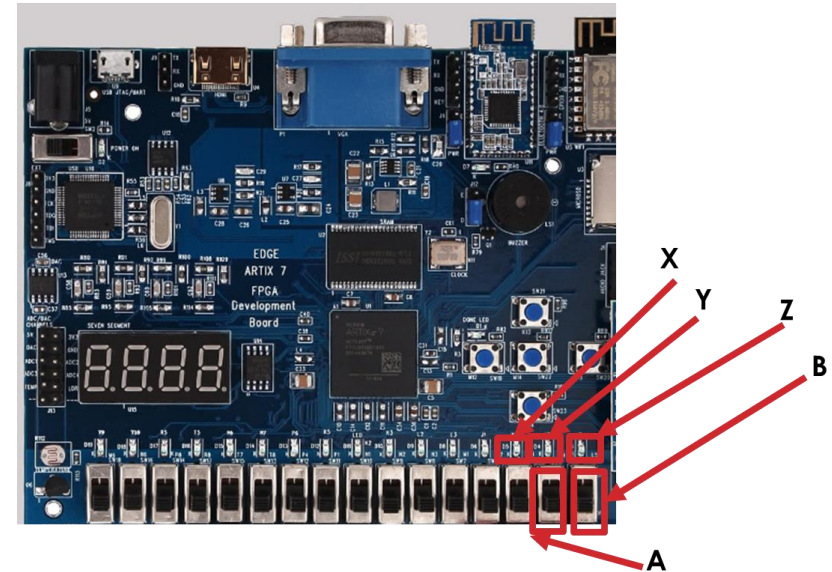
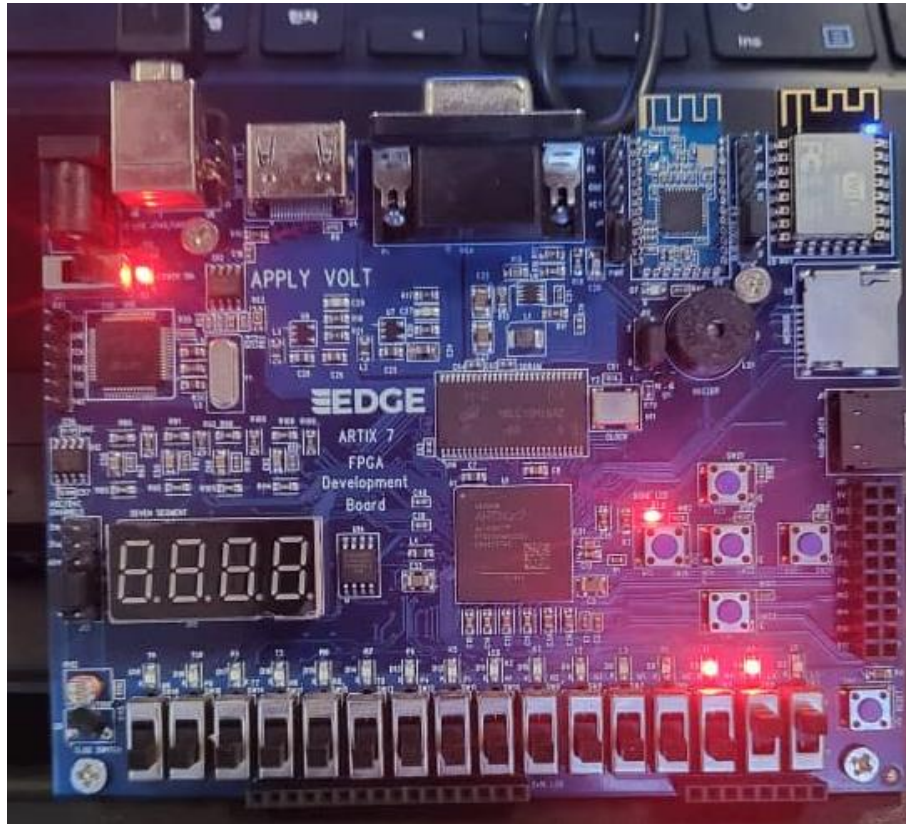
# DESIGN EXAMPLE

- ❖ Vivado project
  - FPGA programming



# DESIGN EXAMPLE

- ❖ Vivado project
- FPGA programming



A	B	X	Y	Z
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

