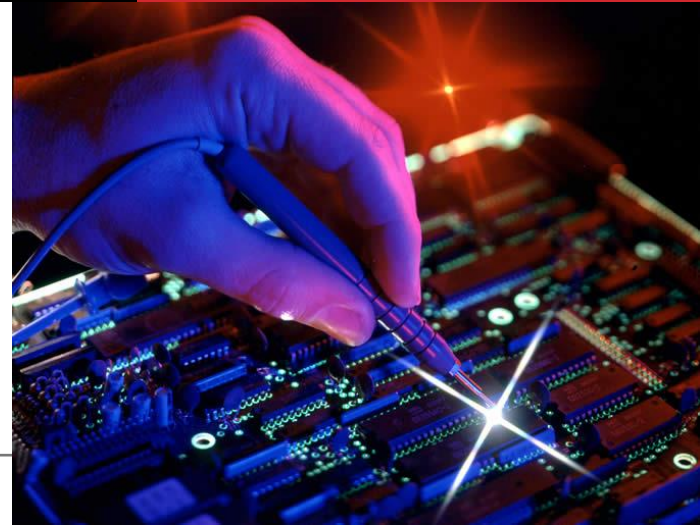




Advanced Computer Architecture & Advanced Microprocessor System

VIVADO DESIGN EXAMPLE

Dennis A. N. Gookyi





CONTENTS

❖ VIVADO Design Example

□ 4-bit Adder Design

- Half Adder Design (Using XOR and AND gates)
- Full Adder Design (Using 2 Half Adders)
- 4-bit Adder Design (Using 4 Full Adders)

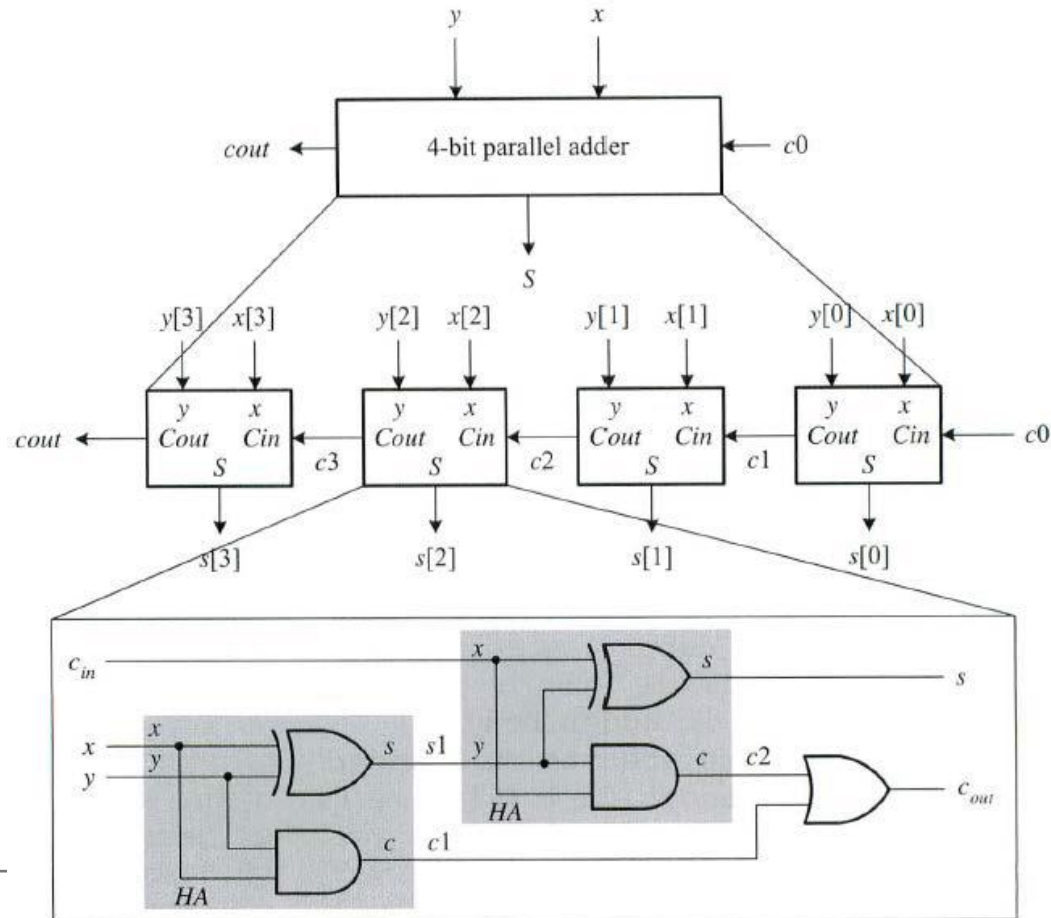


DESIGN EXAMPLE

❖ Design Example

□ 4-bit Adder and Testbench

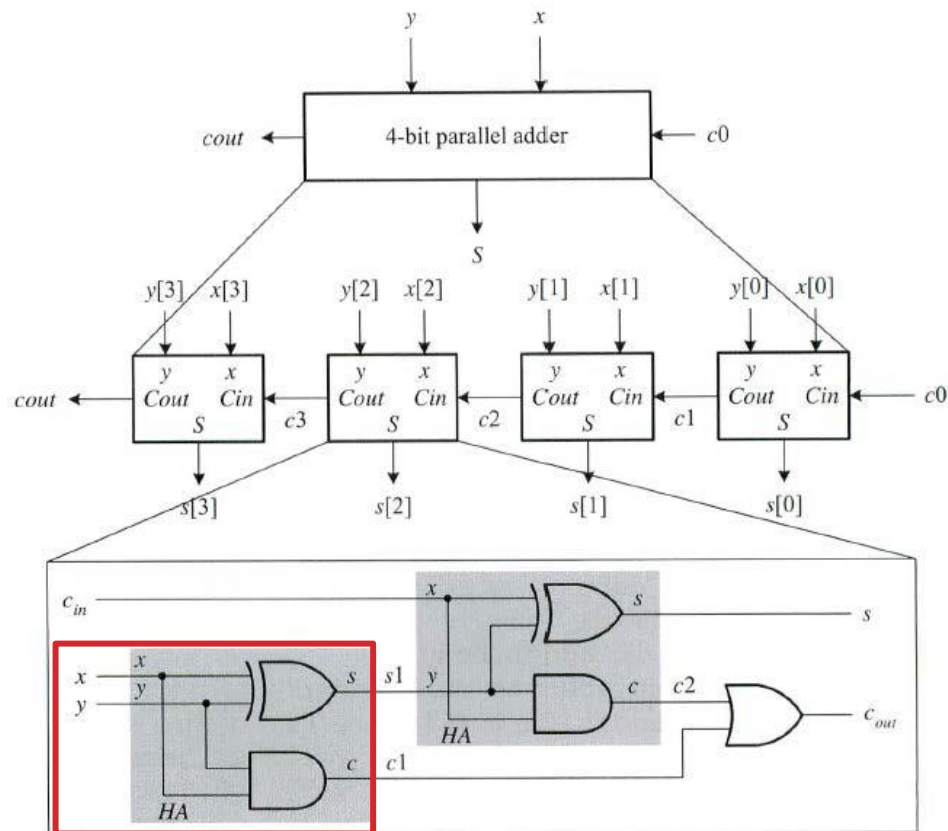
- The Gate-level hierarchical description of the 4-bit adder:



DESIGN EXAMPLE

❖ Verilog code and Testbench

□ Half Adder Design and Truth Table



Truth Table

x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

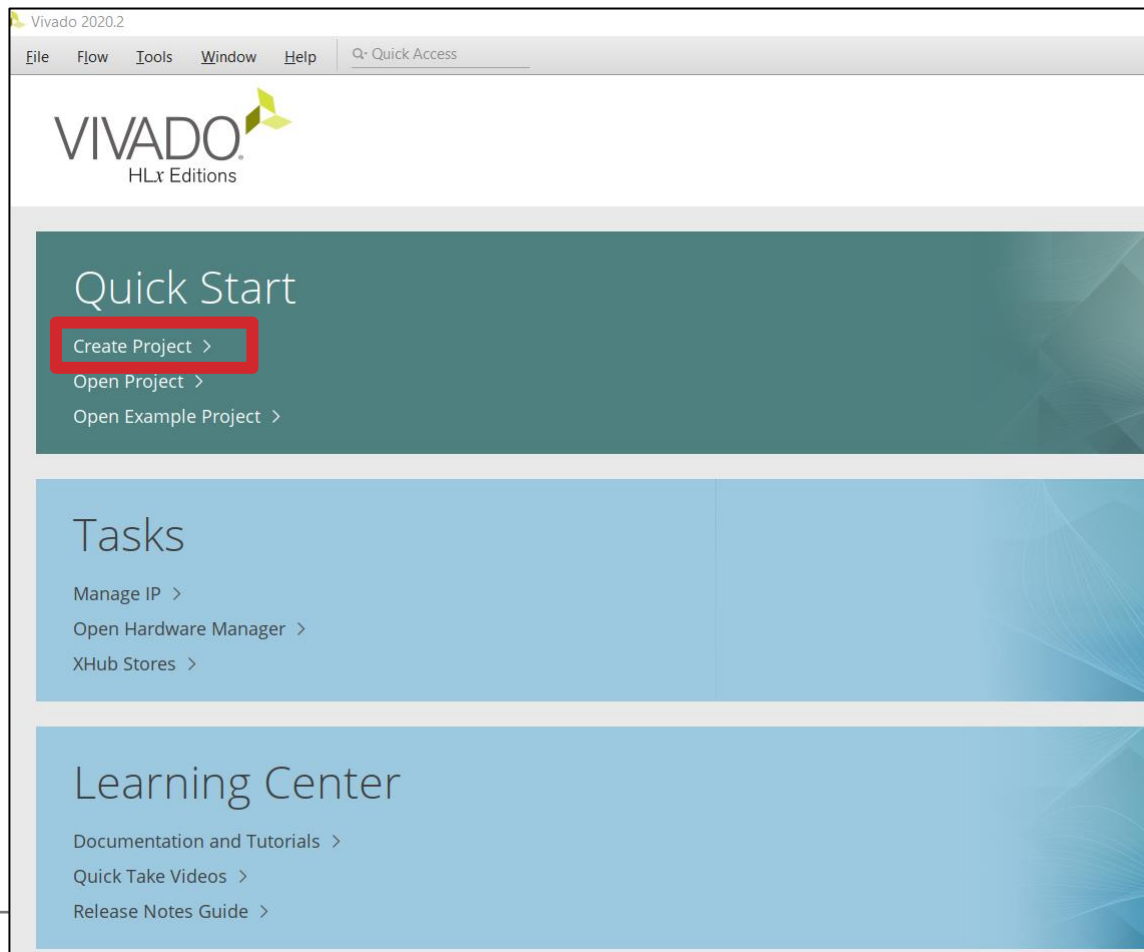
half_adder





DESIGN EXAMPLE

- ❖ Vivado project
 - Creating a new project

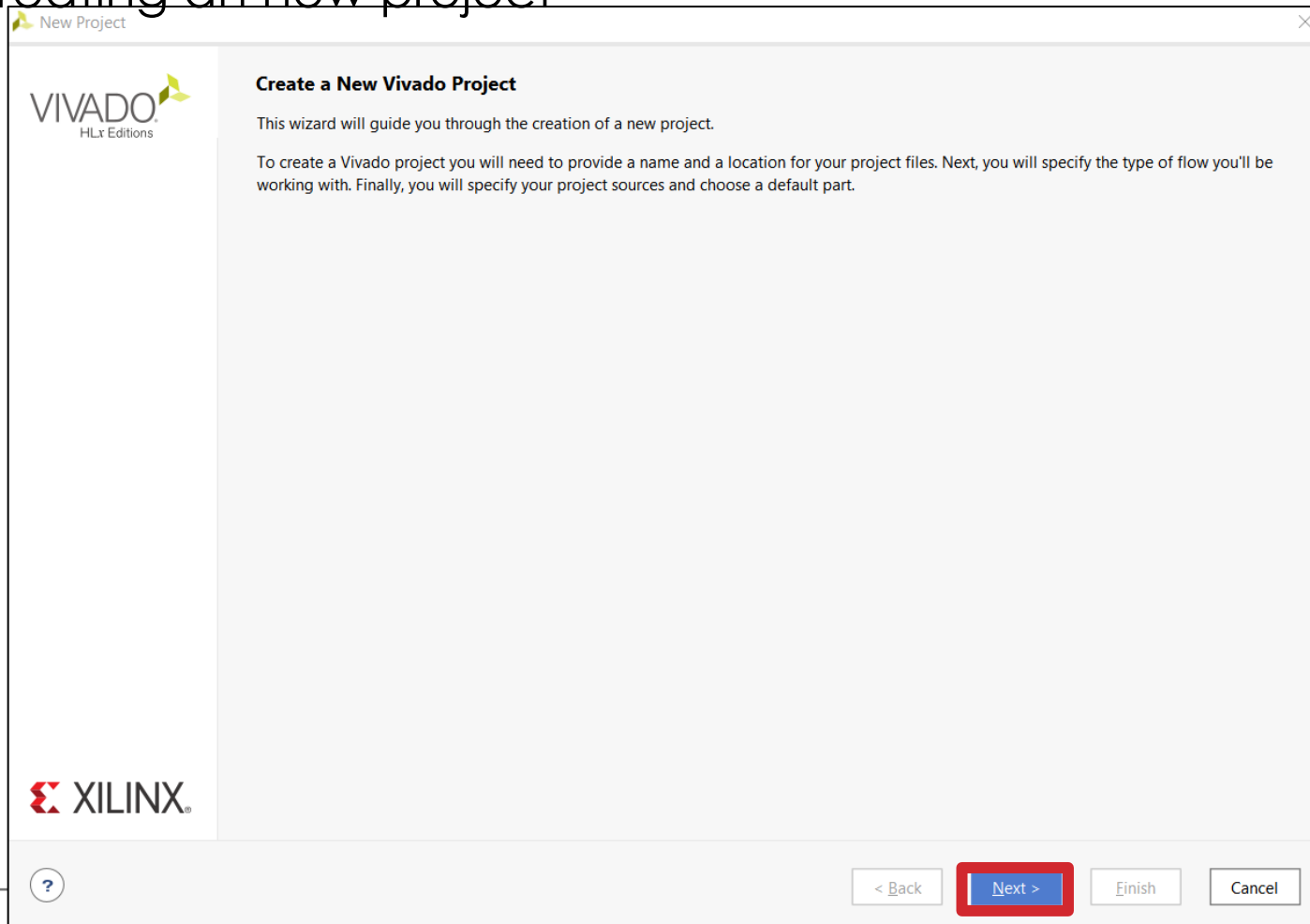




DESIGN EXAMPLE

❖ Vivado project

□ Creating an new project





DESIGN EXAMPLE

❖ Vivado project

□ Creating an new project

The screenshot shows the 'New Project' dialog box in Vivado. The 'Project Name' field is filled with 'FourBitAdder'. The 'Project location' is 'D:/vivado_projects'. The 'Create project subdirectory' checkbox is checked. The 'Project will be created at' path is 'D:/vivado_projects/FourBitAdder'. The 'Next >' button is highlighted with a red box.

New Project

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

Project name: FourBitAdder

Project location: D:/vivado_projects

☒ Create project subdirectory

Project will be created at: D:/vivado_projects/FourBitAdder

< Back Next > Finish Cancel

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

The screenshot shows the 'New Project' dialog box in Vivado. The title bar says 'New Project'. Below the title bar, there is a section titled 'Add Sources' with a description: '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.' Below this description is a large empty rectangular area for listing sources. At the bottom of this area, there is a text prompt: 'Use Add Files, Add Directories or Create File buttons below'. Below the prompt are three buttons: 'Add Files', 'Add Directories', and 'Create File'. At the bottom of the dialog, there are three checkboxes: 'Scan and add RTL include files into project', 'Copy sources into project', and 'Add sources from subdirectories'. The 'Add sources from subdirectories' checkbox is checked. Below the checkboxes are two dropdown menus: 'Target language: Verilog' and 'Simulator language: Mixed'. At the bottom right of the dialog, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted with a red rectangle.

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

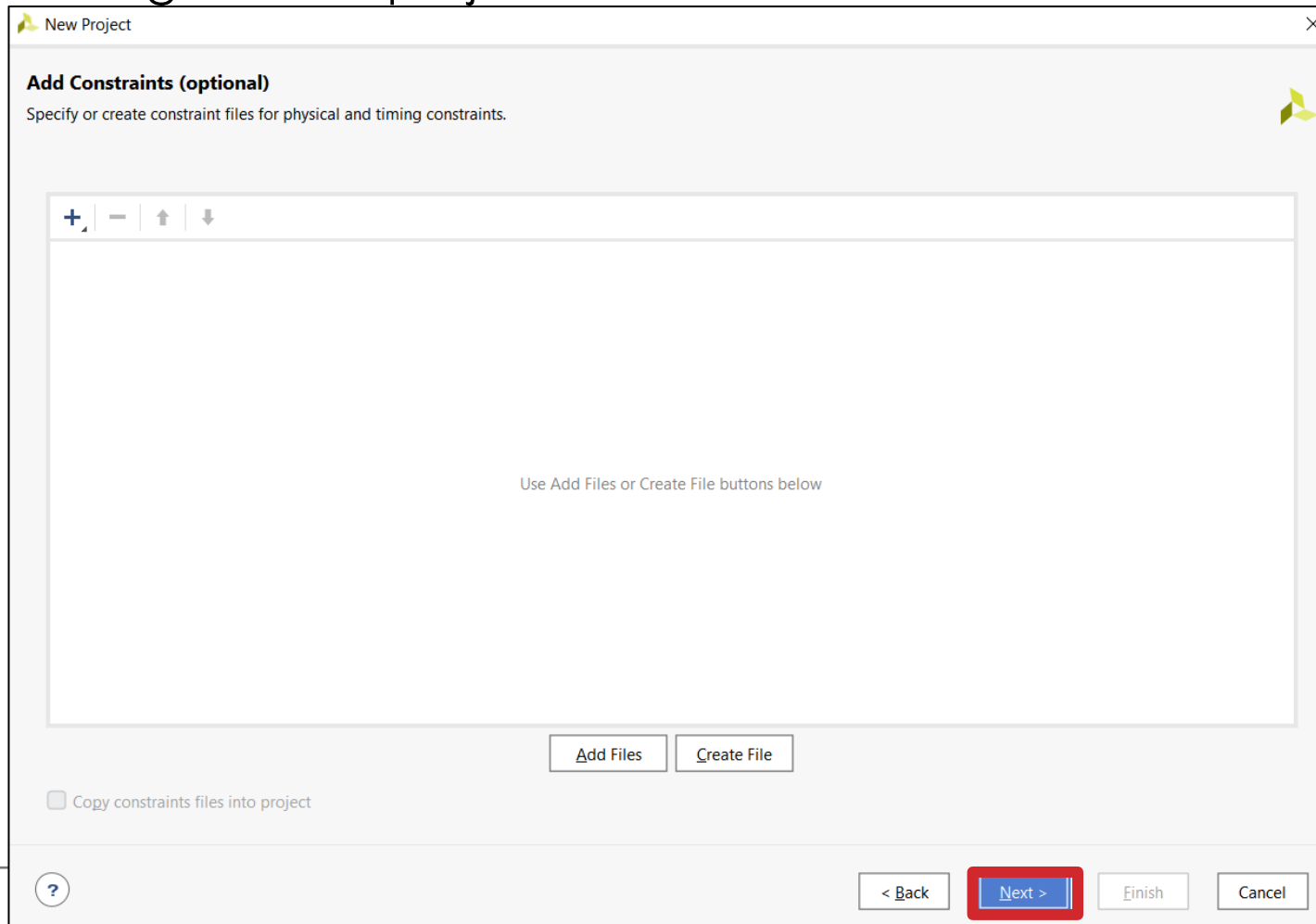




DESIGN EXAMPLE

❖ 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

The screenshot shows the 'New Project Summary' dialog in Vivado. The dialog has a light gray background. On the left side, there is a vertical bar containing the 'VIVADO HLx Editions' logo at the top and the 'XILINX' logo at the bottom. The main area of the dialog contains the following text:

New Project Summary

- ❏ A new RTL project named 'FourBitAdder' will be created.
- ❏ No source files or directories will be added. Use Add Sources to add them later.
- ❏ No constraints files will be added. Use Add Sources to add them later.
- ❏ The default part and product family for the new project:
 - Default Part: xc7a35tftg256-1
 - Product: Artix-7
 - Family: Artix-7
 - Package: ftg256
 - Speed Grade: -1

At the bottom of the dialog, there is a text label: 'To create the project, click Finish'. Below the dialog, there is a footer bar with a help icon (a question mark in a circle) on the left and four buttons on the right: '< Back', 'Next >', 'Finish' (which is highlighted with a red border), and 'Cancel'.





DESIGN EXAMPLE

Vivado project

FourBitAdder - [D:/vivado_projects/FourBitAdder/FourBitAdder.xpr] - Vivado 2020.2

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

Flow Navigator PROJECT MANAGER - FourBitAdder

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

Project Manager - FourBitAdder

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: FourBitAdder
Project location: D:/vivado_projects/FourBitAdder
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

Tcl Console Messages Log Reports **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

FourBitAdder - [D:/vivado_projects/FourBitAdder/FourBitAdder.xpr] - Vivado 2020.2

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

Flow Navigator PROJECT MANAGER - FourBitAdder

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

AM AND DEBUG

PROJECT MANAGER - FourBitAdder

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: FourBitAdder
Project location: D:/vivado_projects/FourBitAdder
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

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

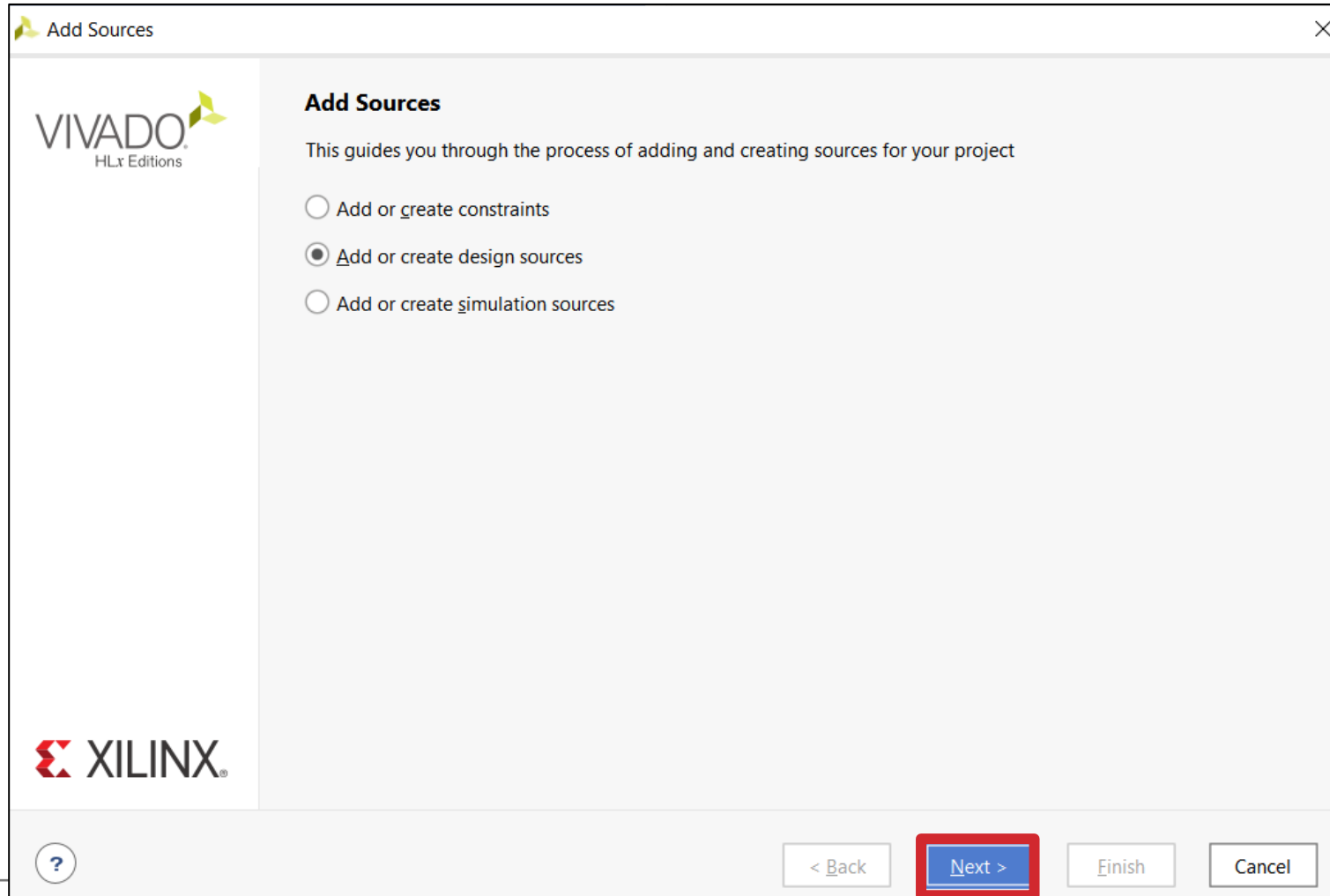
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2020)
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation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





DESIGN EXAMPLE

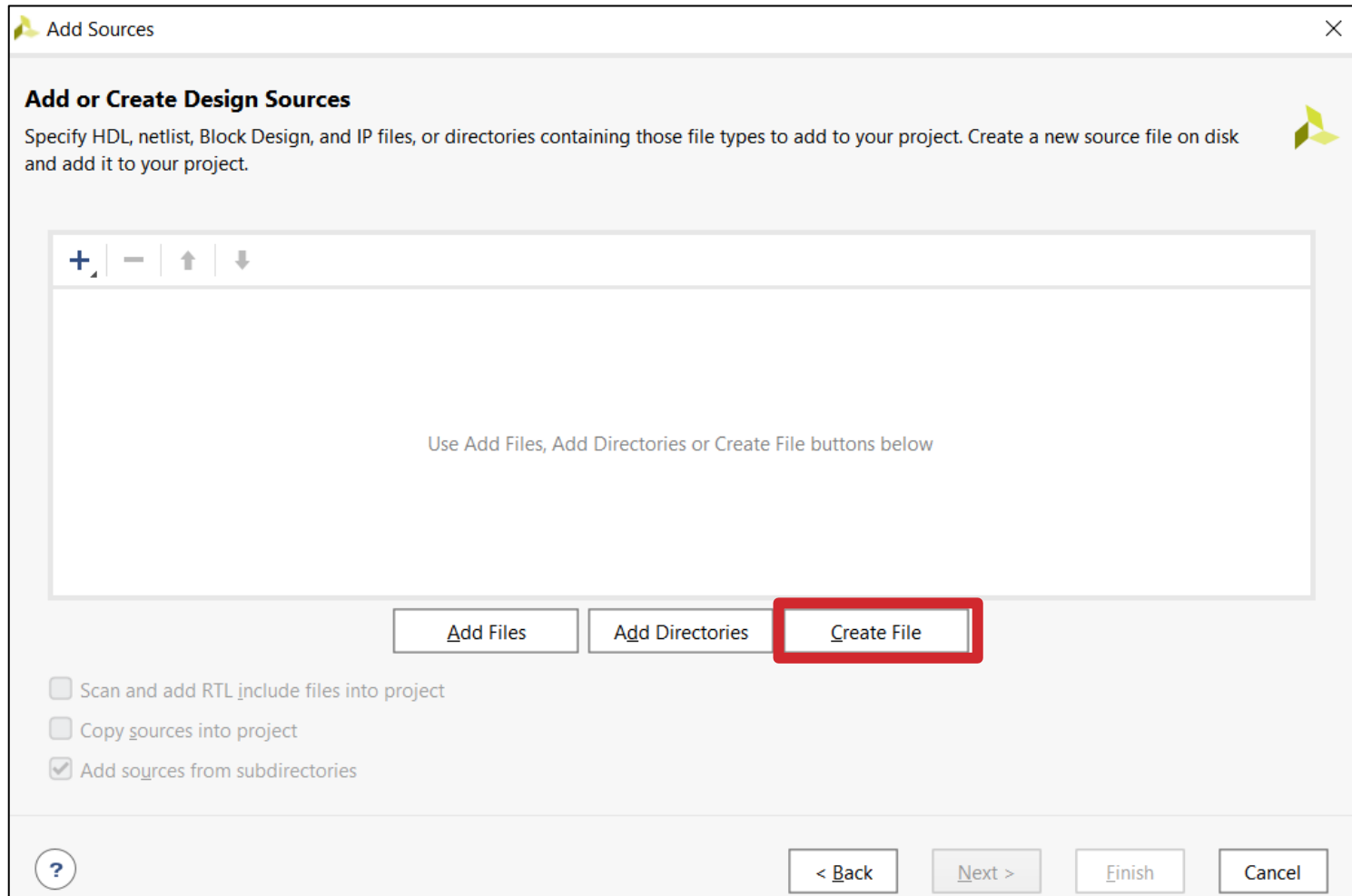
❖ Vivado project





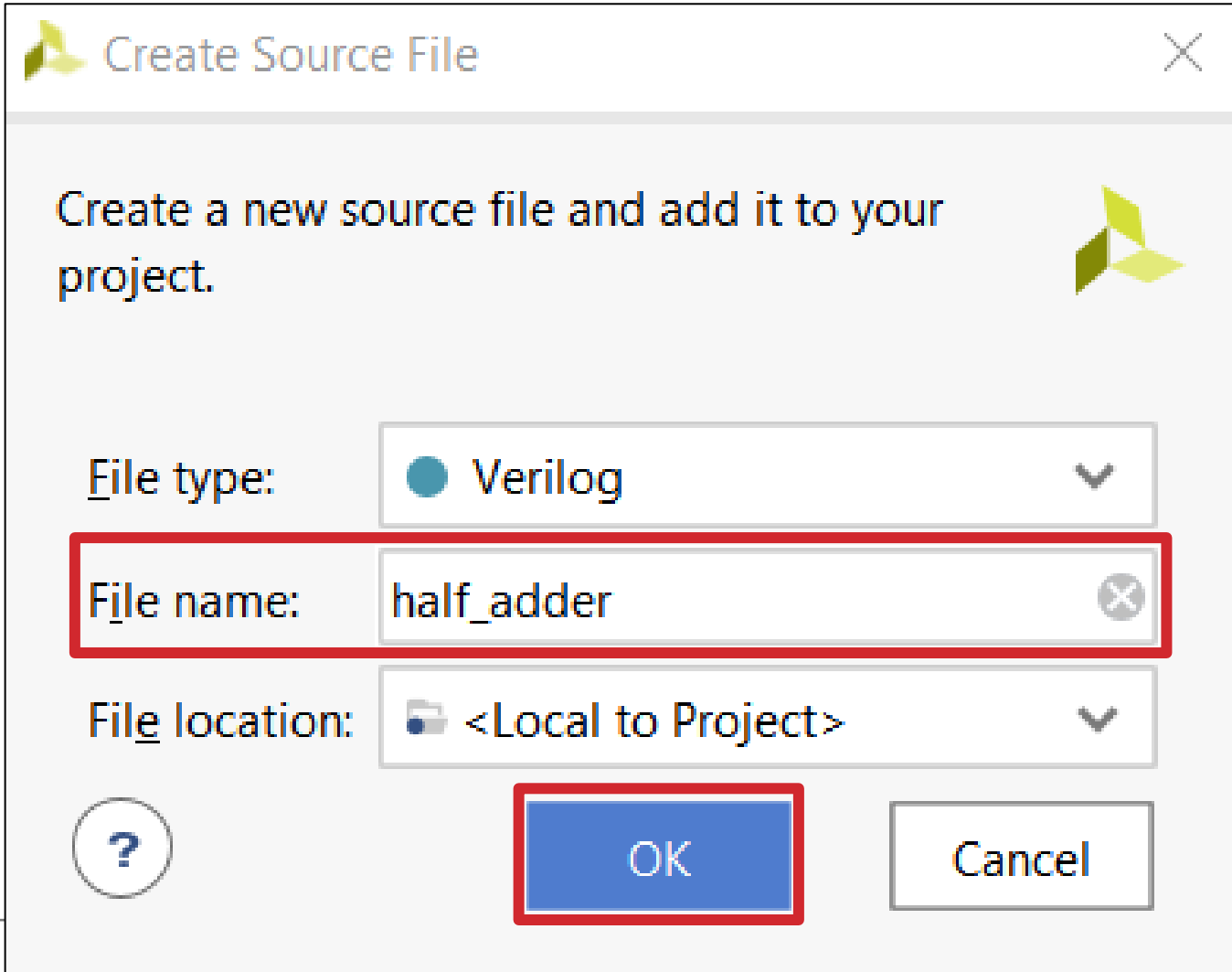
DESIGN EXAMPLE

❖ Vivado project



DESIGN EXAMPLE

❖ Vivado project



The image shows the 'Create Source File' dialog box in the Vivado IDE. The dialog has a title bar with the Vivado logo and a close button. The main text says 'Create a new source file and add it to your project.' with the Vivado logo to the right. There are three input fields: 'File type:' with a dropdown menu showing 'Verilog', 'File name:' with the text 'half_adder', and 'File location:' with a dropdown menu showing '<Local to Project>'. The 'File name:' field is highlighted with a red rectangle. At the bottom, there is a help icon (a circle with a question mark), an 'OK' button (highlighted with a red rectangle), and a 'Cancel' button.

Create Source File

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

File type: Verilog

File name: half_adder

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	half_adder.v	xil_defaultlib	<Local to Project>

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








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

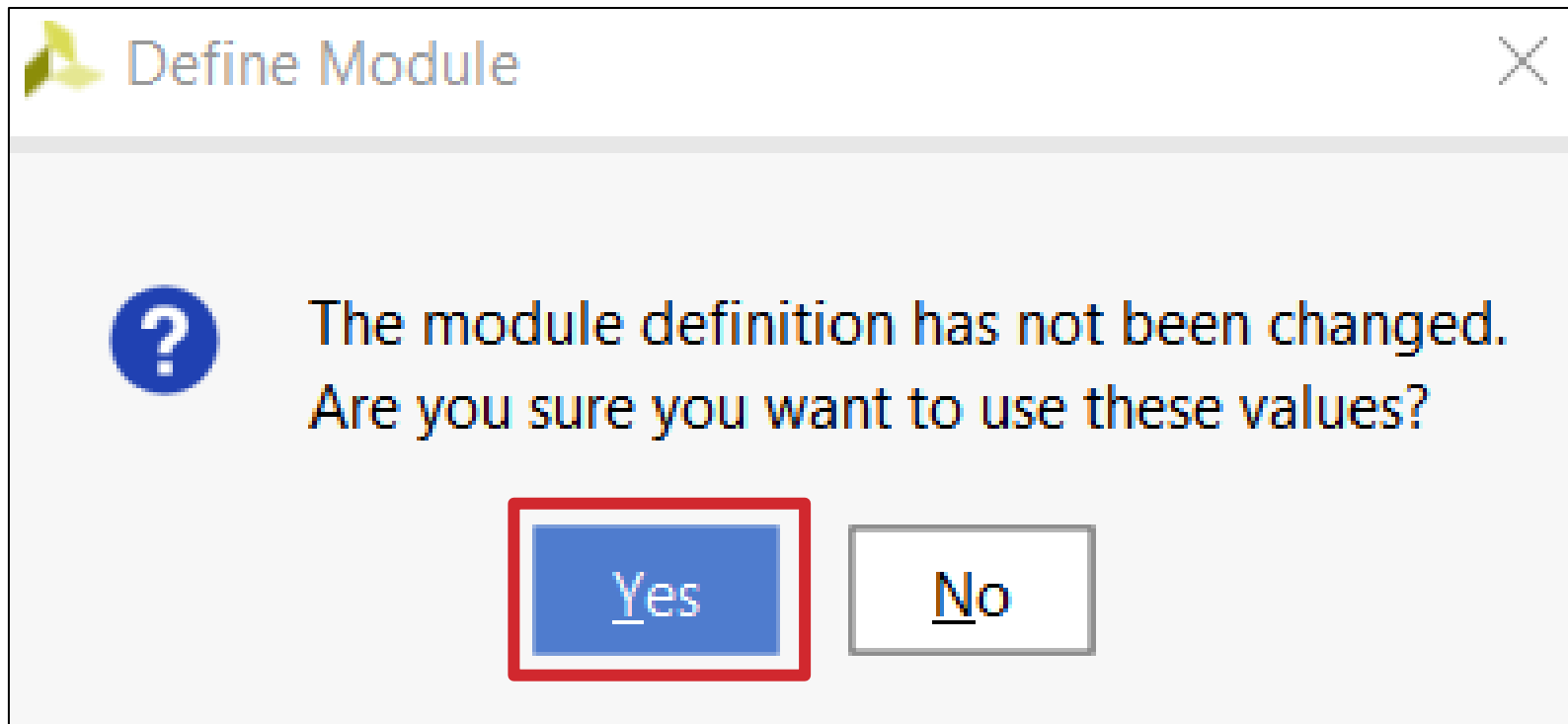
Cancel





DESIGN EXAMPLE

❖ Vivado project





DESIGN EXAMPLE

❖ Vivado project

PROJECT MANAGER - FourBitAdder

Sources

- Design Sources (1)
 - half_adder (half_adder.v)**
- Constraints
- Simulation Sources (1)
 - sim_1 (1)
- Utility Sources

Hierarchy Libraries Compile Order

Source File Properties

half_adder.v

☒ Enabled

Location: D:/vivado_projects/FourBitAdder/FourBitAdder.srscs/sources_1/new

Type: Verilog

Library: xil_defaultlib

Size: 0.5 KB

General Properties

Project Summary x half_adder.v x

D:/vivado_projects/FourBitAdder/FourBitAdder.srscs/sources_1/new/half_adder.v

Editor

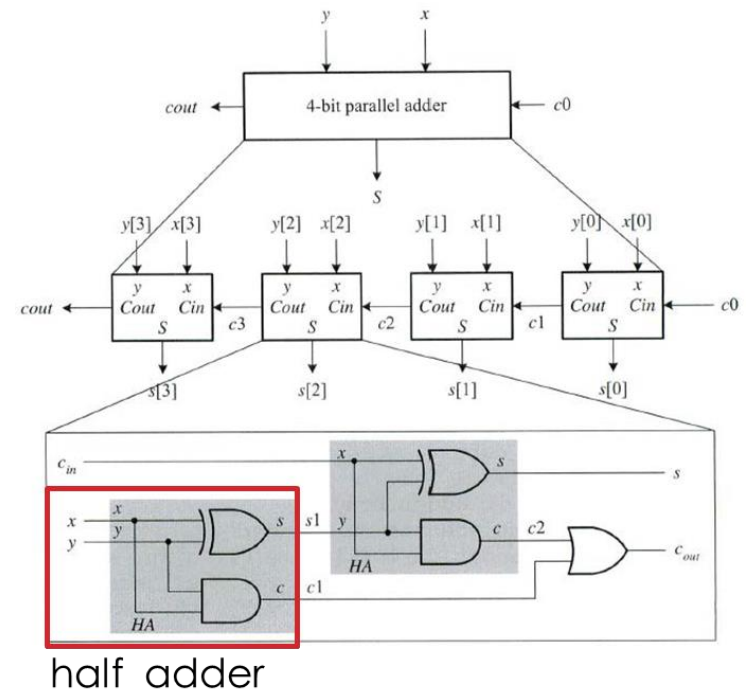
```
4 // Engineer:
5 //
6 // Create Date: 08/08/2023 12:17:10 PM
7 // Design Name:
8 // Module Name: half_adder
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////
21
22
23 module half_adder(
24
25 );
26 endmodule
27
```



DESIGN EXAMPLE

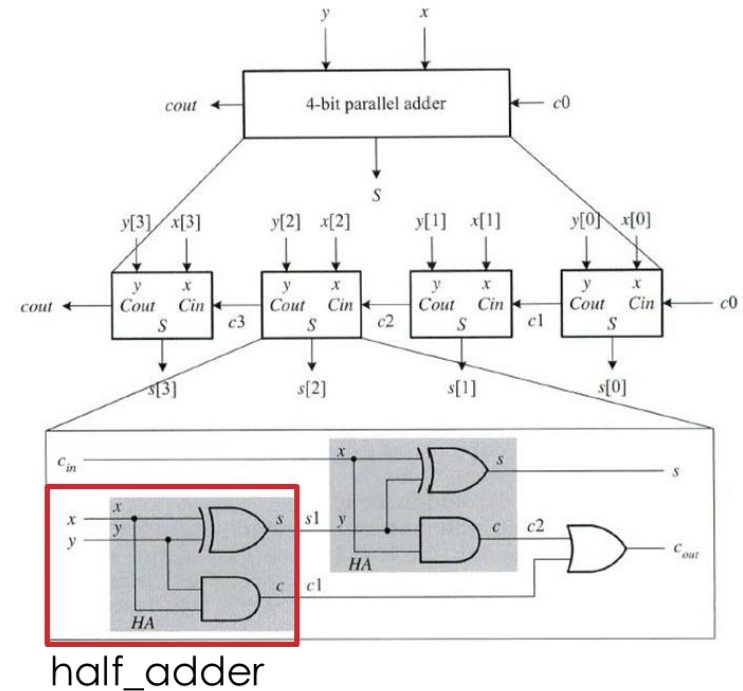
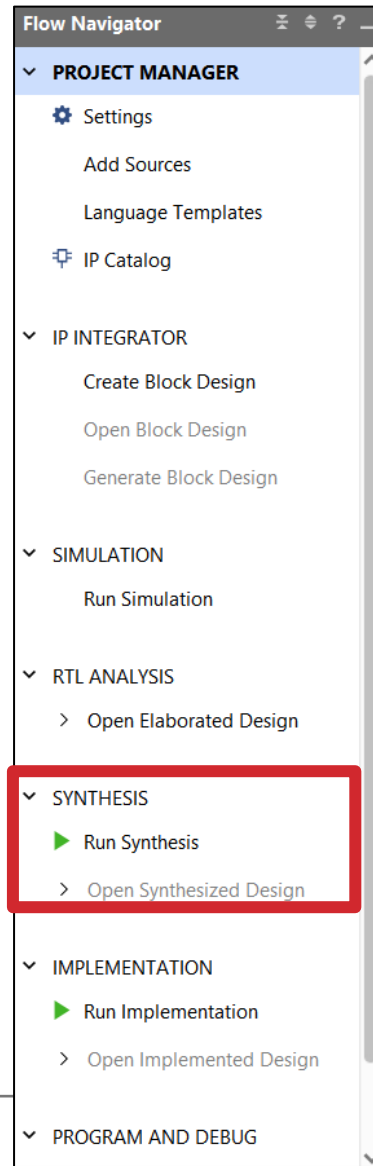
- ❖ Vivado project
 - Verilog code for Half Adder Module

```
23 module half_adder(x, y, s, c);  
24  
25 //input/output declaration  
26 input x, y;  
27 output s, c;  
28  
29 //primitive gate instantiation  
30 xor(s, x, y);  
31 and(c, x, y);  
32  
33 endmodule
```



DESIGN EXAMPLE

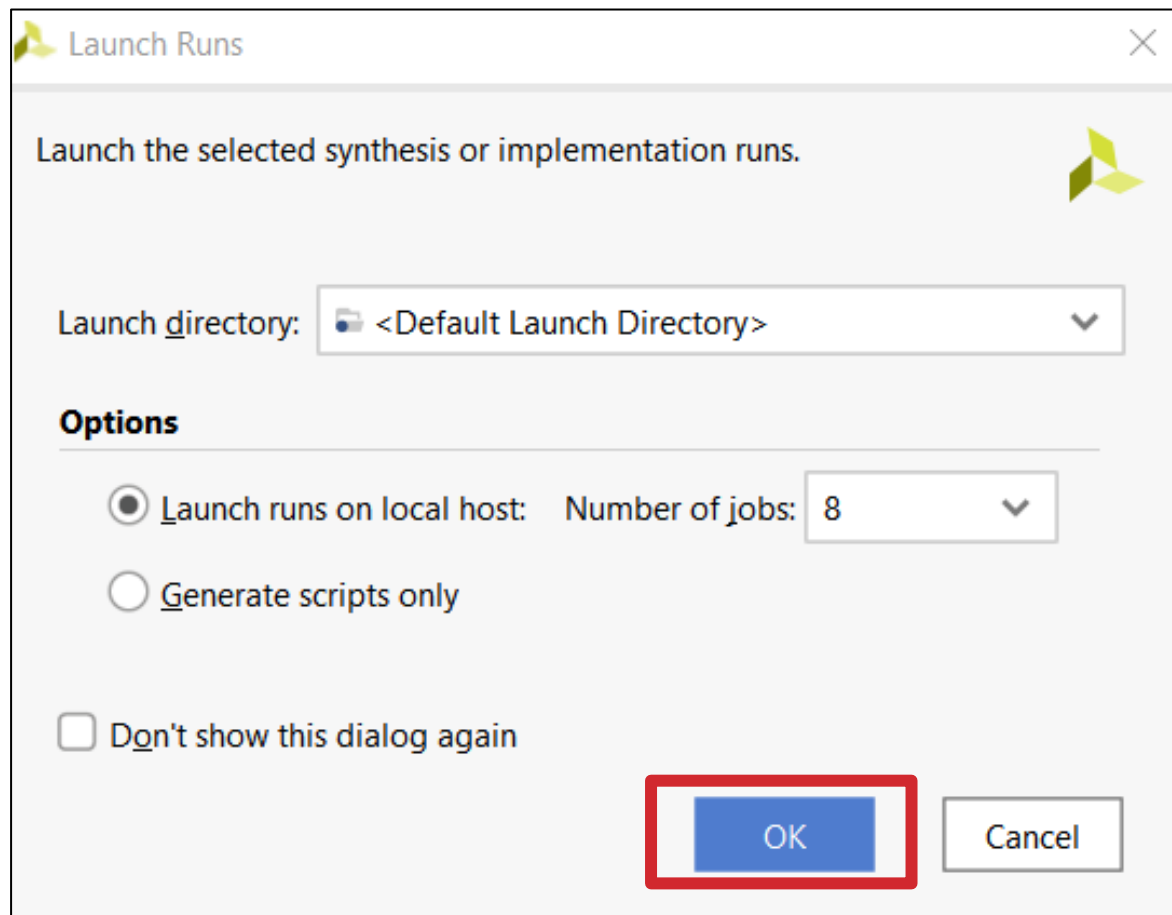
- ❖ Vivado project
 - Logic synthesis





DESIGN EXAMPLE

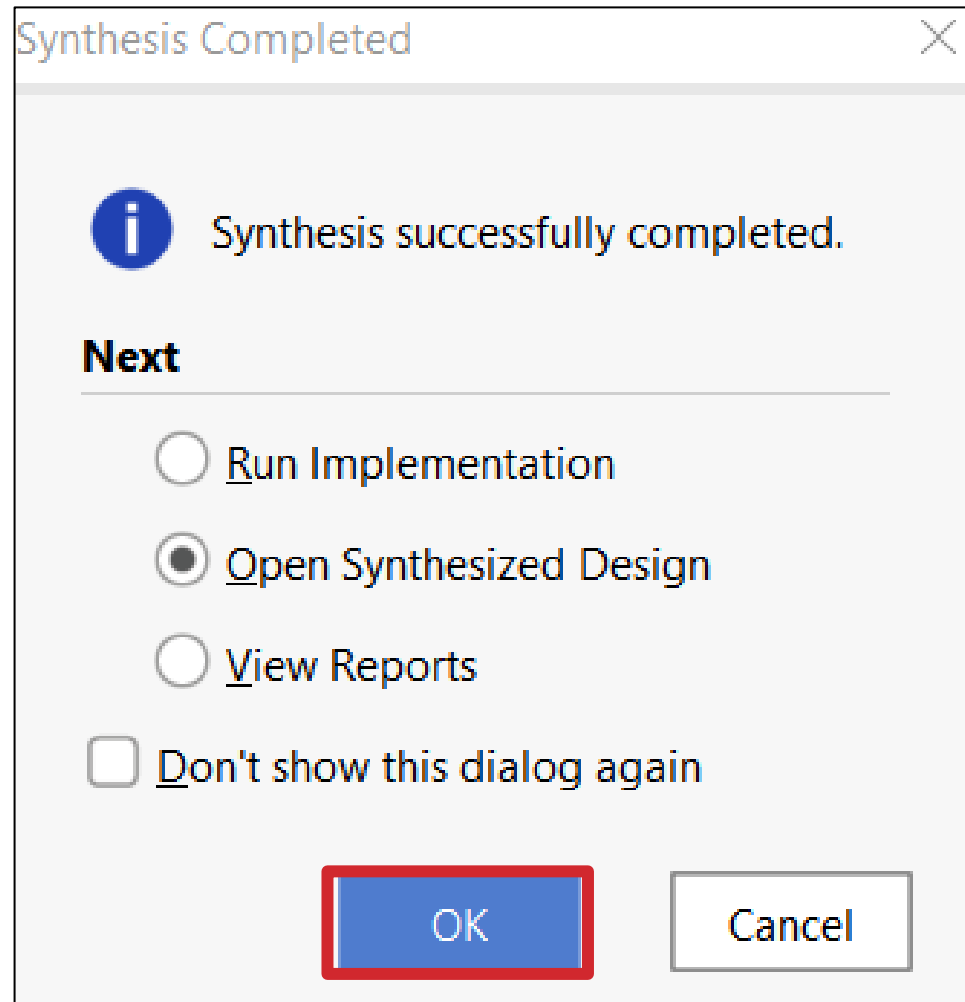
- ❖ Vivado project
 - Logic synthesis





DESIGN EXAMPLE

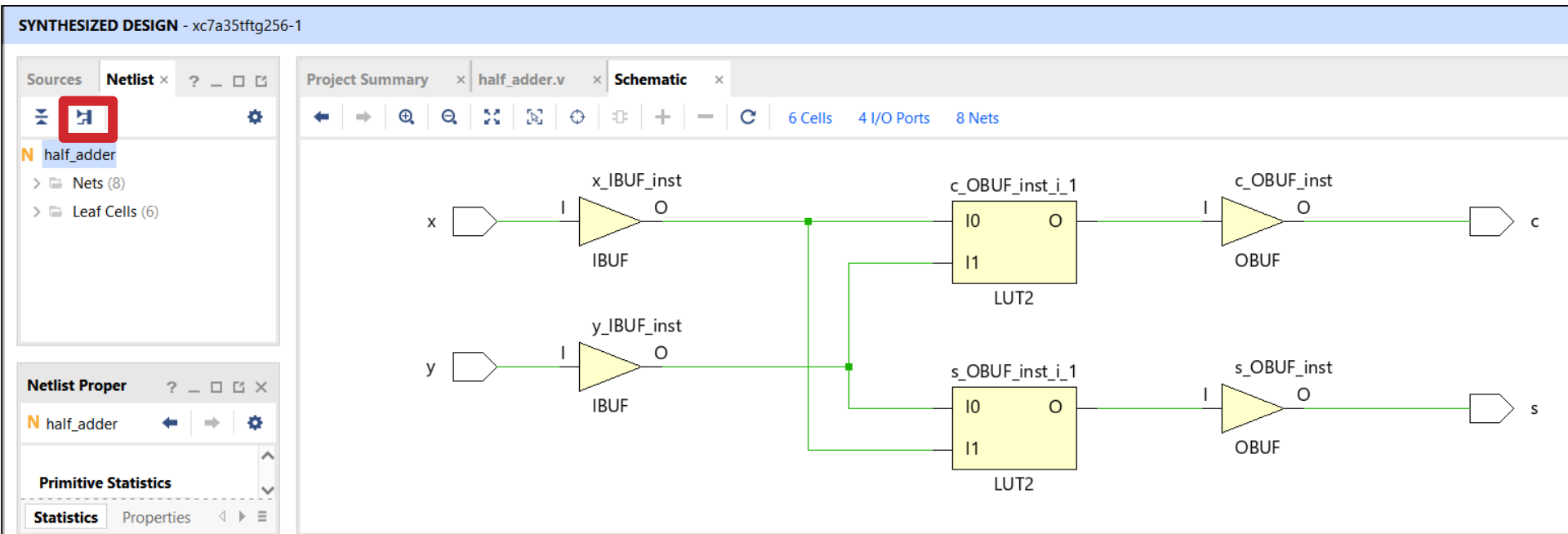
- ❖ Vivado project
 - Logic synthesis





DESIGN EXAMPLE

- ❖ Vivado project
 - Logic synthesis (Schematic)



DESIGN EXAMPLE

- ❖ Vivado project
 - Logic synthesis (Results)

SYNTHESIZED DESIGN - xc7a35tftg256-1

Sources Netlist × ? _ □ □

half_adder

- > Nets (8)
- > Leaf Cells (6)

Project Summary × half_adder.v × Schematic ×

Overview | Dashboard

Synthesis

Status: ✓ Complete

Messages: No errors or warnings

Part: xc7a35tftg256-1

Strategy: [Vivado Synthesis Defaults](#)

Report Strategy: [Vivado Synthesis Default Reports](#)

Incremental synthesis: None

Implementation

Status: Not started

Messages: No errors or warnings

Part: xc7a35tftg256-1

Strategy: [Vivado Implementation Defaults](#)

Report Strategy: [Vivado Implementation Default Reports](#)

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	1	20800	0.01
IO	4	170	2.35

Power

[Run Implementation](#) to see power results

Netlist Proper ? _ □ □ ×

half_adder ← → ⚙

Primitive Statistics

Primitive type	Count
27	istics

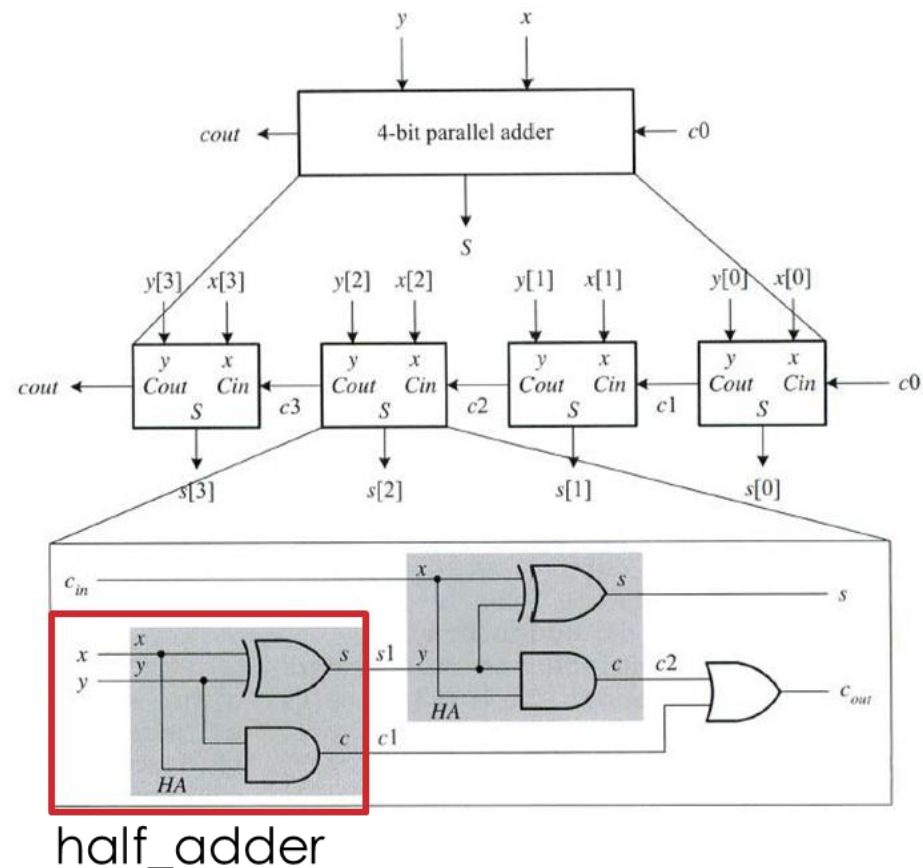
Properties < > ≡



DESIGN EXAMPLE

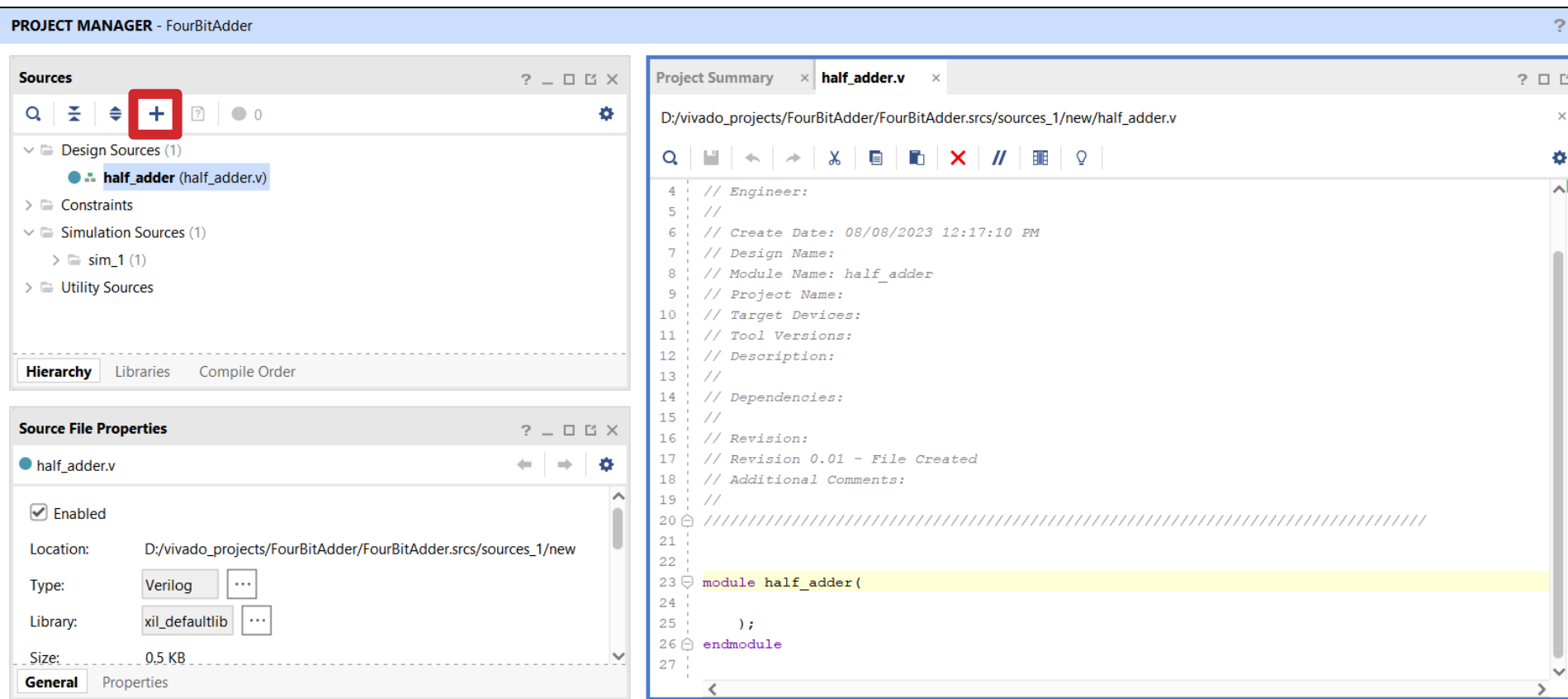
- ❖ Vivado project
 - Half Adder Testbench (Test Vectors)

x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



DESIGN EXAMPLE

- ❖ Vivado project
 - Creating a Testbench



The screenshot displays the Vivado Project Manager and Source File Properties windows. The Project Manager window, titled "PROJECT MANAGER - FourBitAdder", shows the "Sources" tab with a red box highlighting the "+" icon for adding new sources. The "Design Sources (1)" section lists "half_adder (half_adder.v)". The "Source File Properties" window, titled "Source File Properties", shows the properties for "half_adder.v". The "General" tab is selected, showing the file is "Enabled", located at "D:/vivado_projects/FourBitAdder/FourBitAdder.srscs/sources_1/new", with a type of "Verilog" and a library of "xil_defaultlib". The "Size" is 0.5 KB. The "Project Summary" window, titled "Project Summary x half_adder.v", shows the file path "D:/vivado_projects/FourBitAdder/FourBitAdder.srscs/sources_1/new/half_adder.v" and the Verilog code for the "half_adder" module.

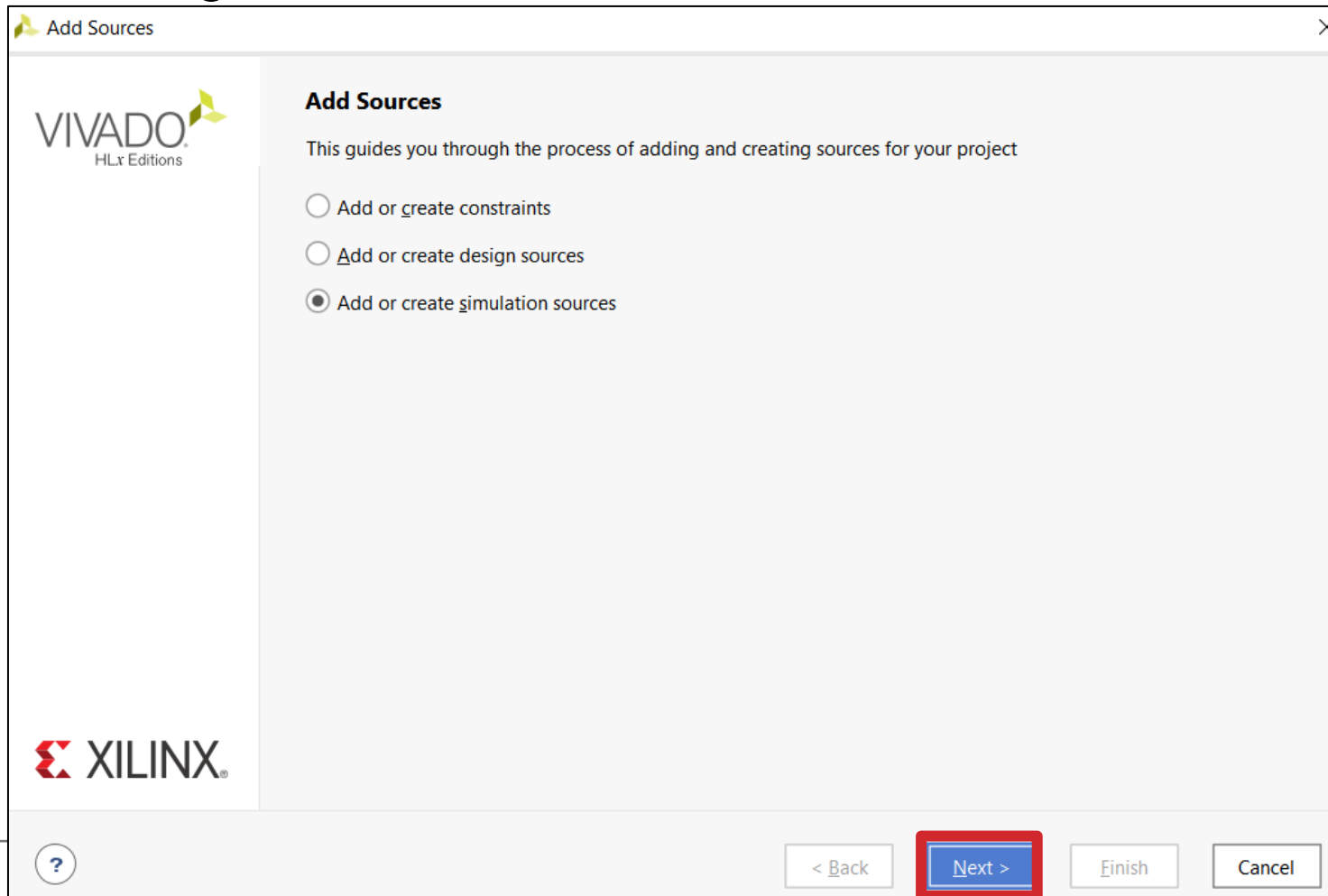
```
4 // Engineer:
5 //
6 // Create Date: 08/08/2023 12:17:10 PM
7 // Design Name:
8 // Module Name: half_adder
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module half_adder(
24
25 );
26 endmodule
27
```




DESIGN EXAMPLE

❖ Vivado project

□ Creating a Testbench

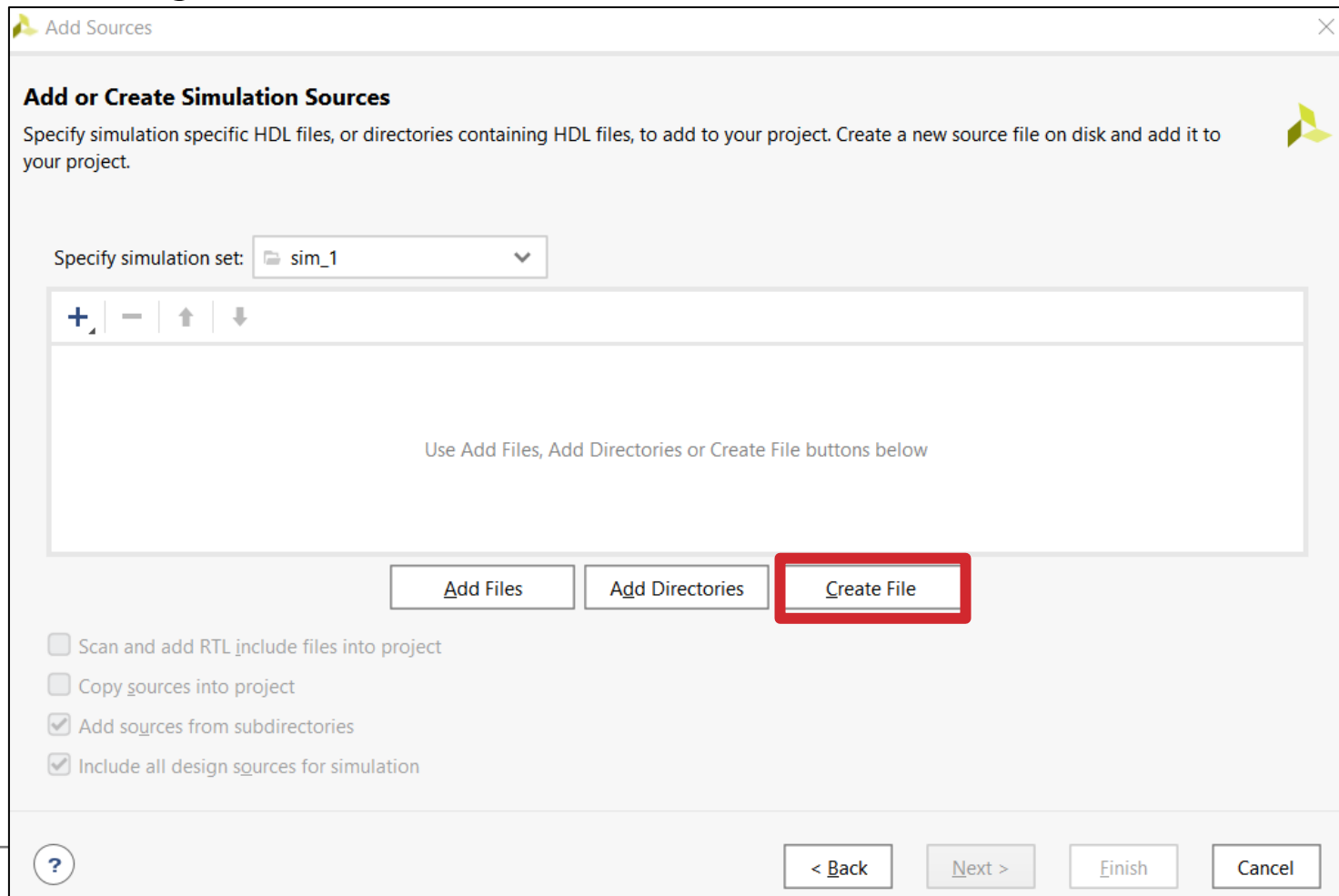




DESIGN EXAMPLE

❖ Vivado project

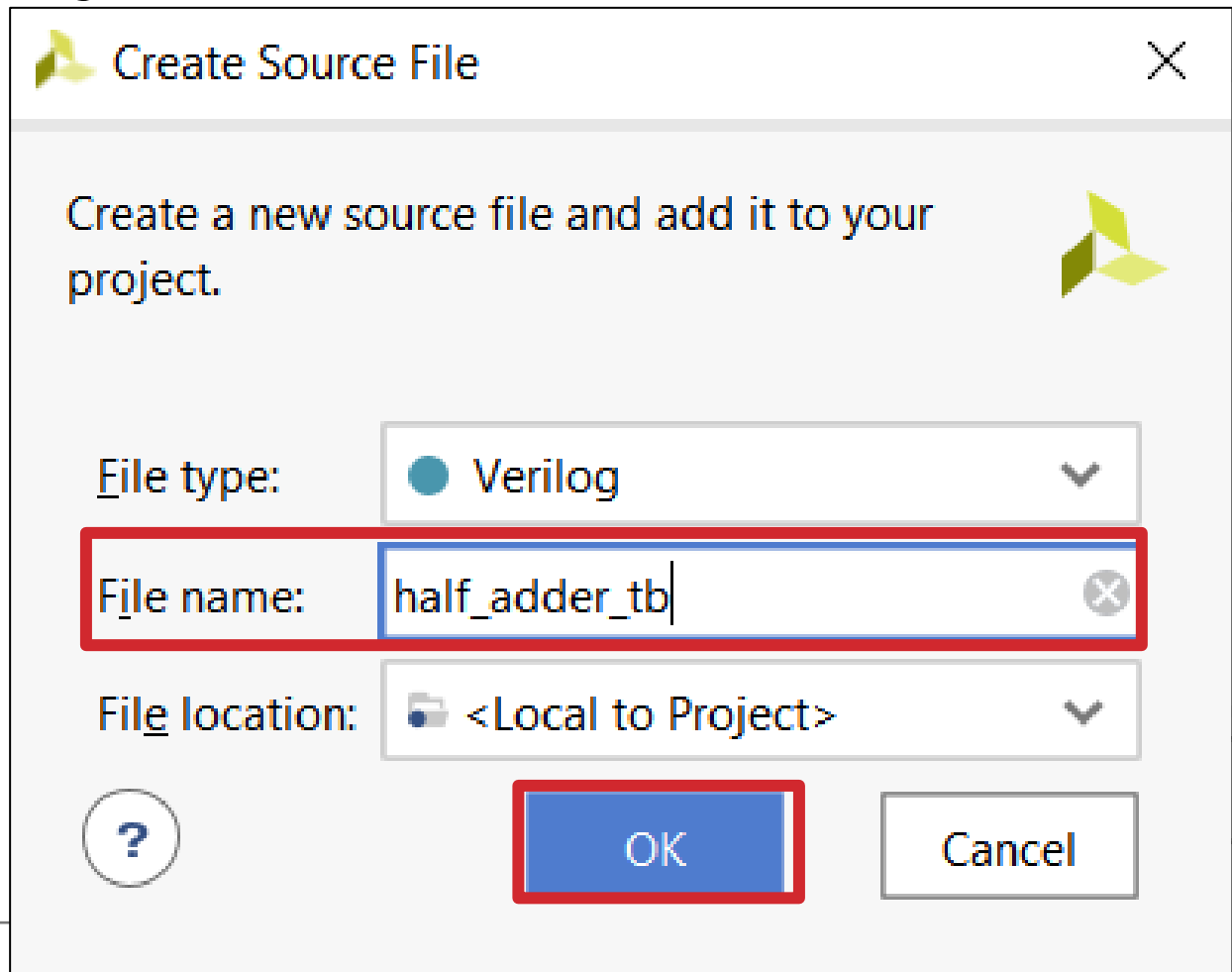
□ Creating a Testbench





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	half_adder_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

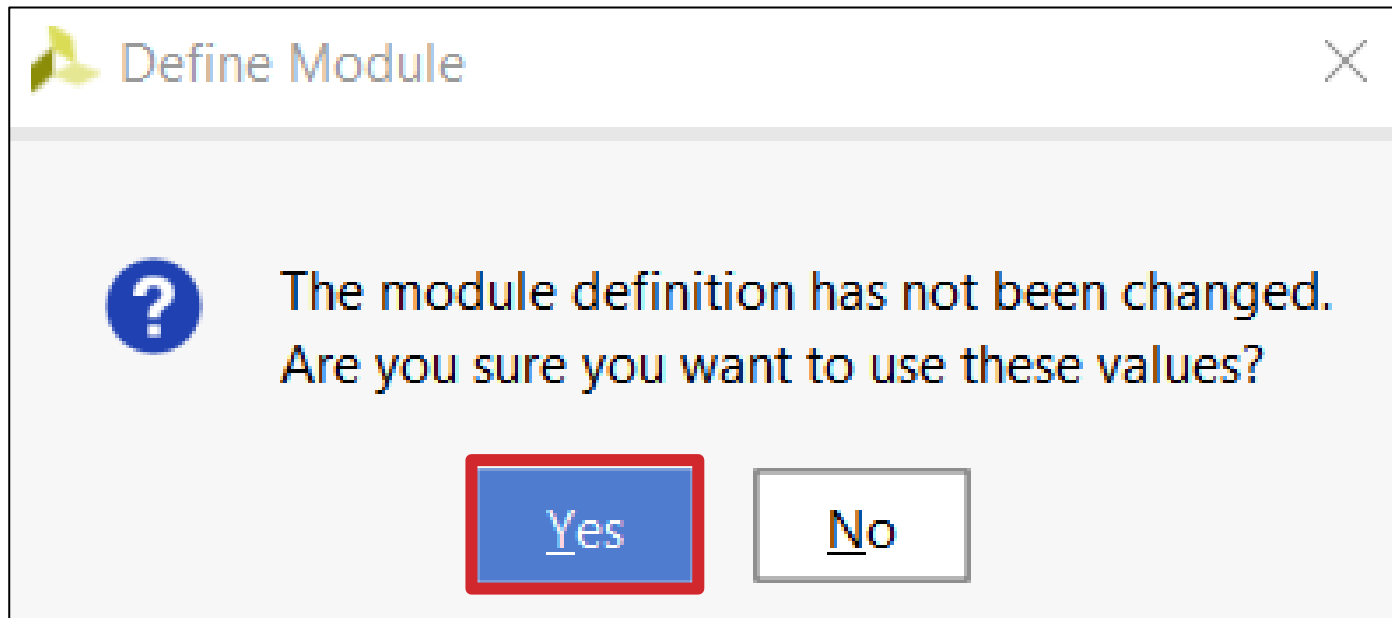
 OK Cancel





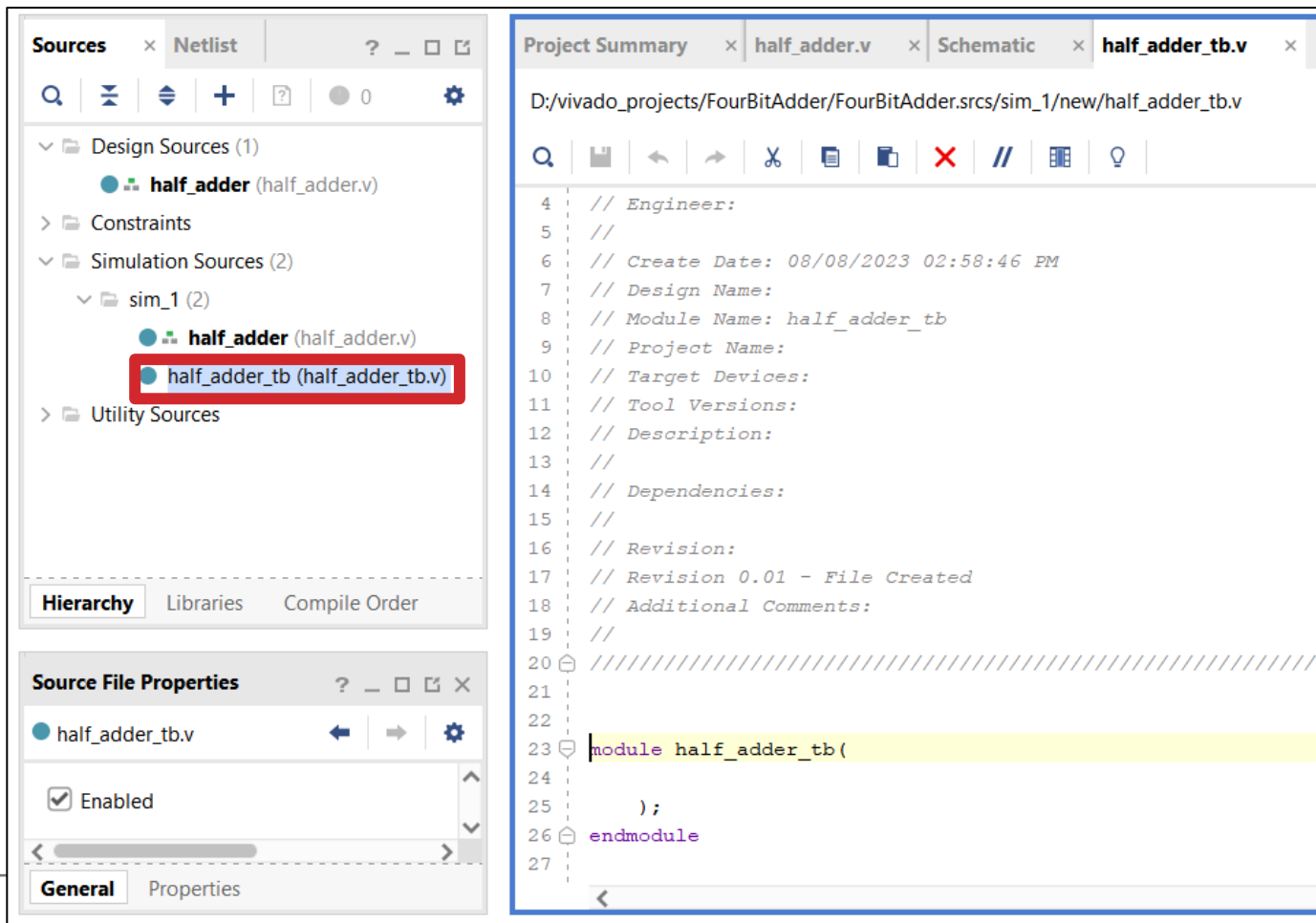
DESIGN EXAMPLE

- ❖ Vivado project
 - Creating a Testbench



DESIGN EXAMPLE

- ❖ Vivado project
 - Creating a Testbench



The screenshot displays the Vivado IDE interface. On the left, the 'Sources' window shows the project hierarchy: 'Design Sources (1)' containing 'half_adder (half_adder.v)', 'Constraints', and 'Simulation Sources (2)' containing 'sim_1 (2)' which includes 'half_adder (half_adder.v)' and 'half_adder_tb (half_adder_tb.v)'. The 'half_adder_tb (half_adder_tb.v)' file is highlighted with a red box. Below the 'Sources' window is the 'Source File Properties' window for 'half_adder_tb.v', showing it is 'Enabled'. The main editor window on the right shows the 'half_adder_tb.v' file with the following code:

```
4 // Engineer:
5 //
6 // Create Date: 08/08/2023 02:58:46 PM
7 // Design Name:
8 // Module Name: half_adder_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 half_adder_tb(
24
25 );
26 endmodule
27
```





DESIGN EXAMPLE

- ❖ Vivado project
 - Creating a Testbench

```
25 module half_adder_tb;  
26     // Inputs  
27     reg x;  
28     reg y;  
29     // Outputs  
30     wire s;  
31     wire c;  
32     // Instantiate the Unit Under Test (UUT)  
33     half_adder uut (.x(x), .y(y), .s(s), .c(c));  
34     reg [1:0] i;  
35     initial begin  
36         // Add stimulus here  
37         for(i = 0; i <= 3; i = i + 1) begin  
38             x = i[1]; y = i[0]; #100;  
39         end  
40     end  
41     initial begin  
42         $monitor($realtime, "ns x=%h y=%h {c,s}=%h", x, y, {c, s});  
43     end  
44 endmodule
```

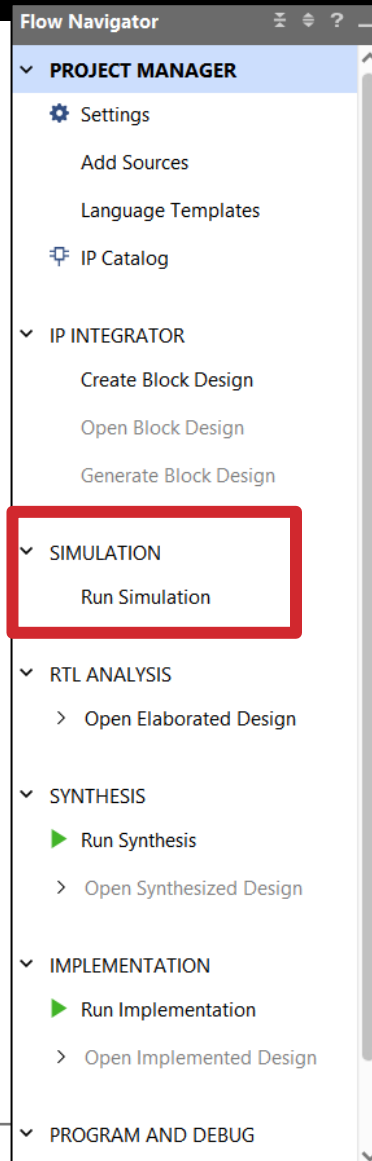
x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1





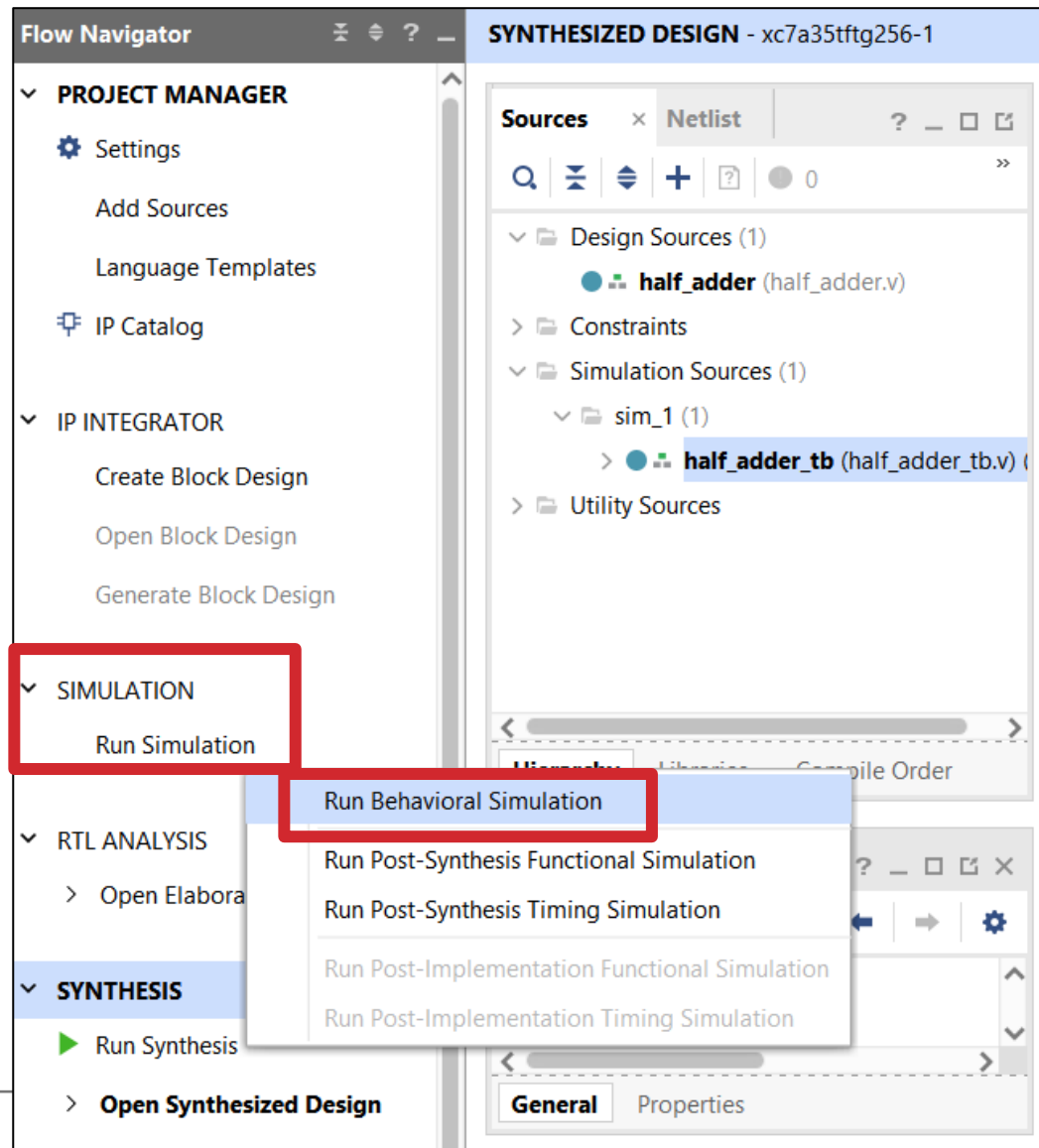
DESIGN EXAMPLE

- ❖ Vivado project
 - Running Simulation



DESIGN EXAMPLE

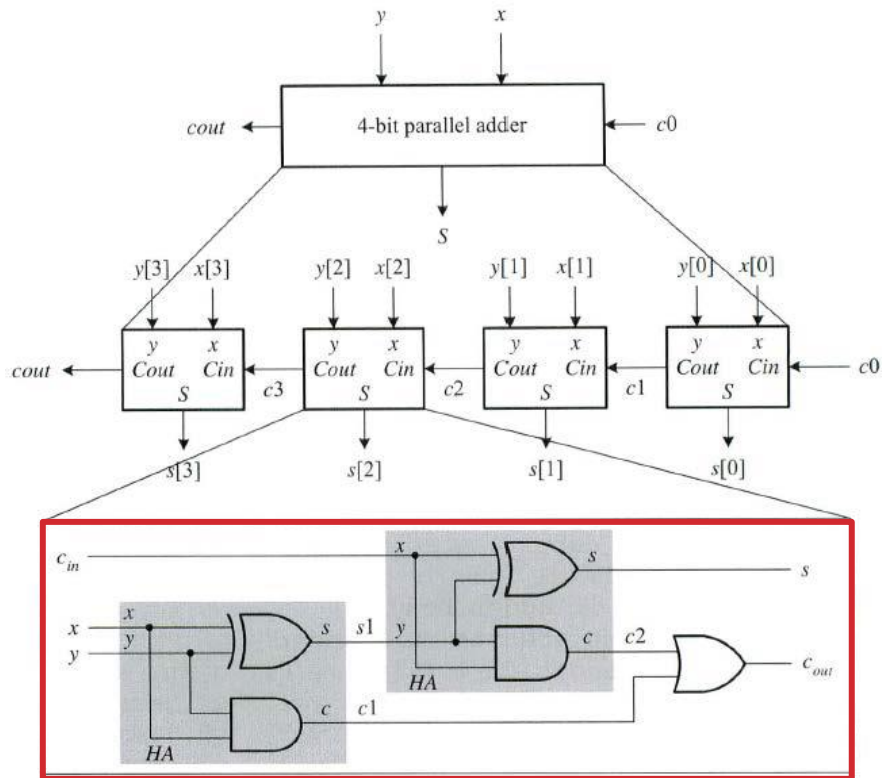
- ❖ Vivado project
 - Running Simulation





DESIGN EXAMPLE

- ❖ Verilog code and Testbench
 - Full Adder Design and Truth Table



full_adder

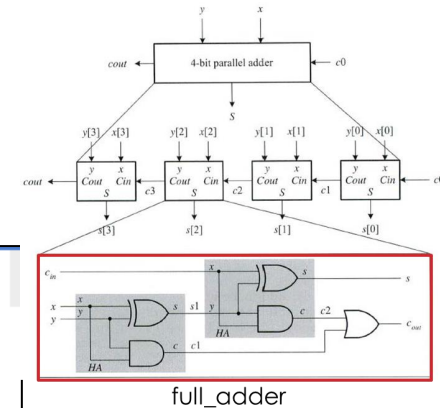
Truth Table

x	y	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



DESIGN EXAMPLE

- ❖ Verilog code and Testbench
 - Verilog Code for Full Adder module



Sources

Design Sources (1)

- full_adder (full_adder.v) (2)
 - ha_1 : half_adder (half_adder.v)
 - ha_2 : half_adder (half_adder.v)

Constraints

Simulation Sources (2)

- sim_1 (2)

Project Summary | half_adder.v | half_adder_tb.v | **full_adder.v**

D:/vivado_projects/FourBitAdder/FourBitAdder.srscs/sources_1/new/full_adder.v

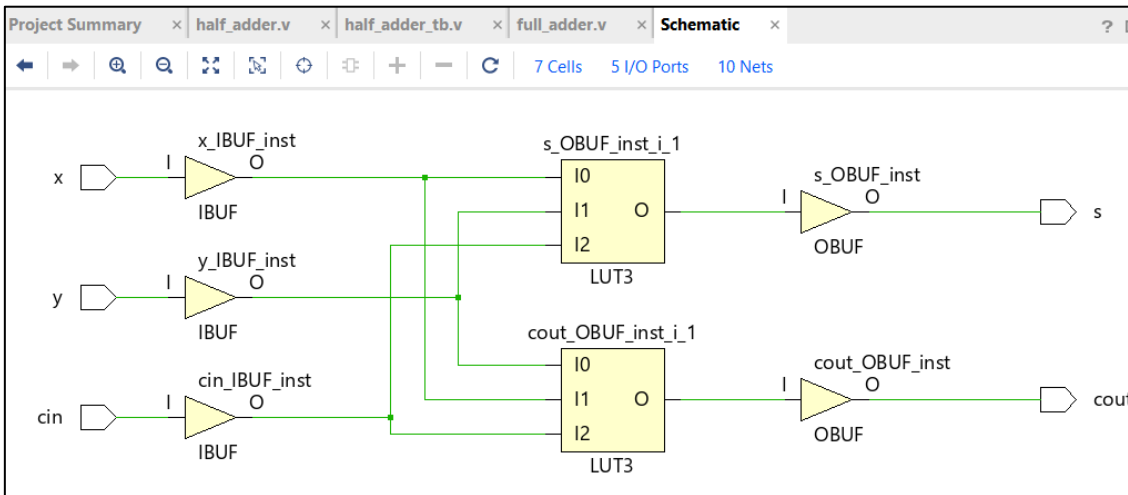
```
21
22
23 module full_adder(x, y, cin, s, cout);
24     //input output declaration
25     input x, y, cin;
26     output s, cout;
27
28     //wire interconnection declaration
29     wire s1, c1, c2;
30
31     //Instantiate full adder using two half adders
32     half_adder ha_1(x, y, s1, c1);
33     half_adder ha_2(cin, s1, s, c2);
34     or(cout, c1, c2);
35
36 endmodule
37
```



DESIGN EXAMPLE

❖ Verilog code and Testbench

□ Full Adder Schematic



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

- Open Elaborated Design

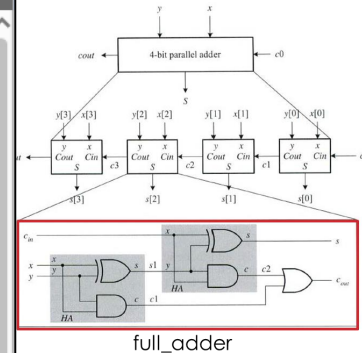
SYNTHESIS

- Run Synthesis
- Open Synthesized Design

IMPLEMENTATION

- Run Implementation
- Open Implemented Design

PROGRAM AND DEBUG

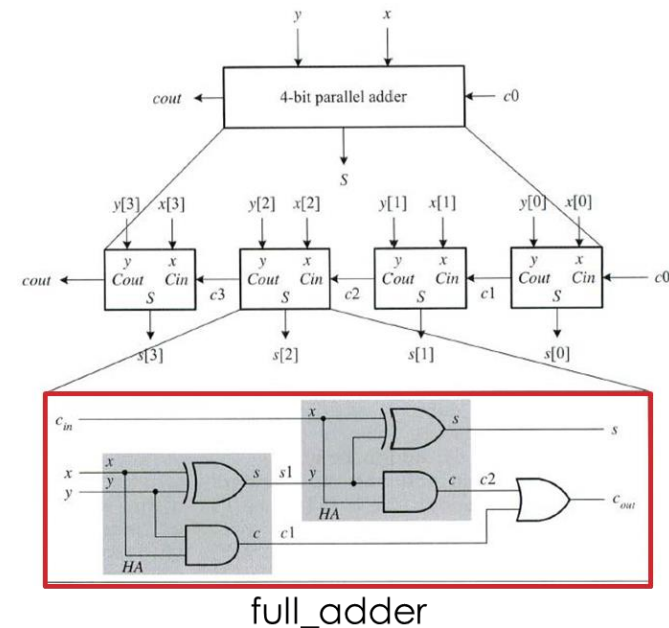




DESIGN EXAMPLE

- ❖ Verilog code and Testbench
 - Full Adder Testbench (Test Vectors)

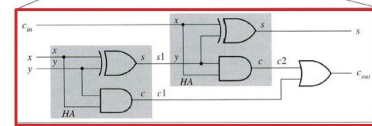
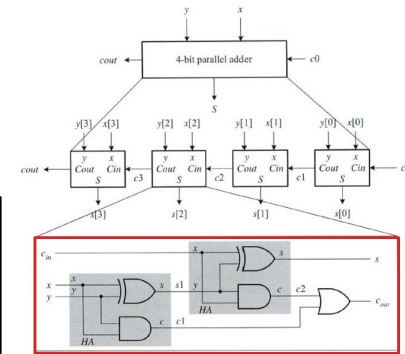
x	y	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



DESIGN EXAMPLE

❖ Verilog code and Testbench

□ Full Adder Testbench



full adder

x	y	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

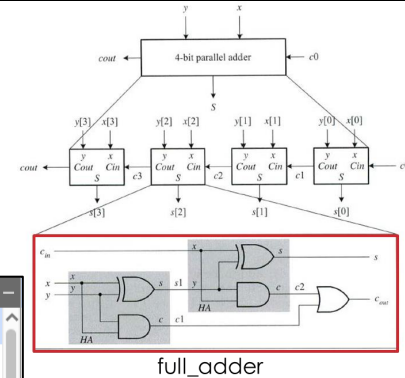
```
23 module full_adder_tb;  
24 //inputs  
25 reg x;  
26 reg y;  
27 reg cin;  
28 //outputs  
29 wire s;  
30 wire cout;  
31 //instantiate the Unit Under Test (UUT)  
32 full_adder uut(.x(x), .y(y), .cin(cin), .s(s), .cout(cout));  
33  
34 reg [2:0] i;  
35 initial begin  
36     for(i = 0; i <= 7; i = i + 1) begin  
37         x = i[2]; y = i[1]; cin = i[0]; #100;  
38     end  
39 end  
40 initial begin  
41     $monitor($realtime, "ns x=%h y=%h cin=%h {cout,s}=%h", x, y, cin, {cout, s});  
42 end  
43 endmodule
```



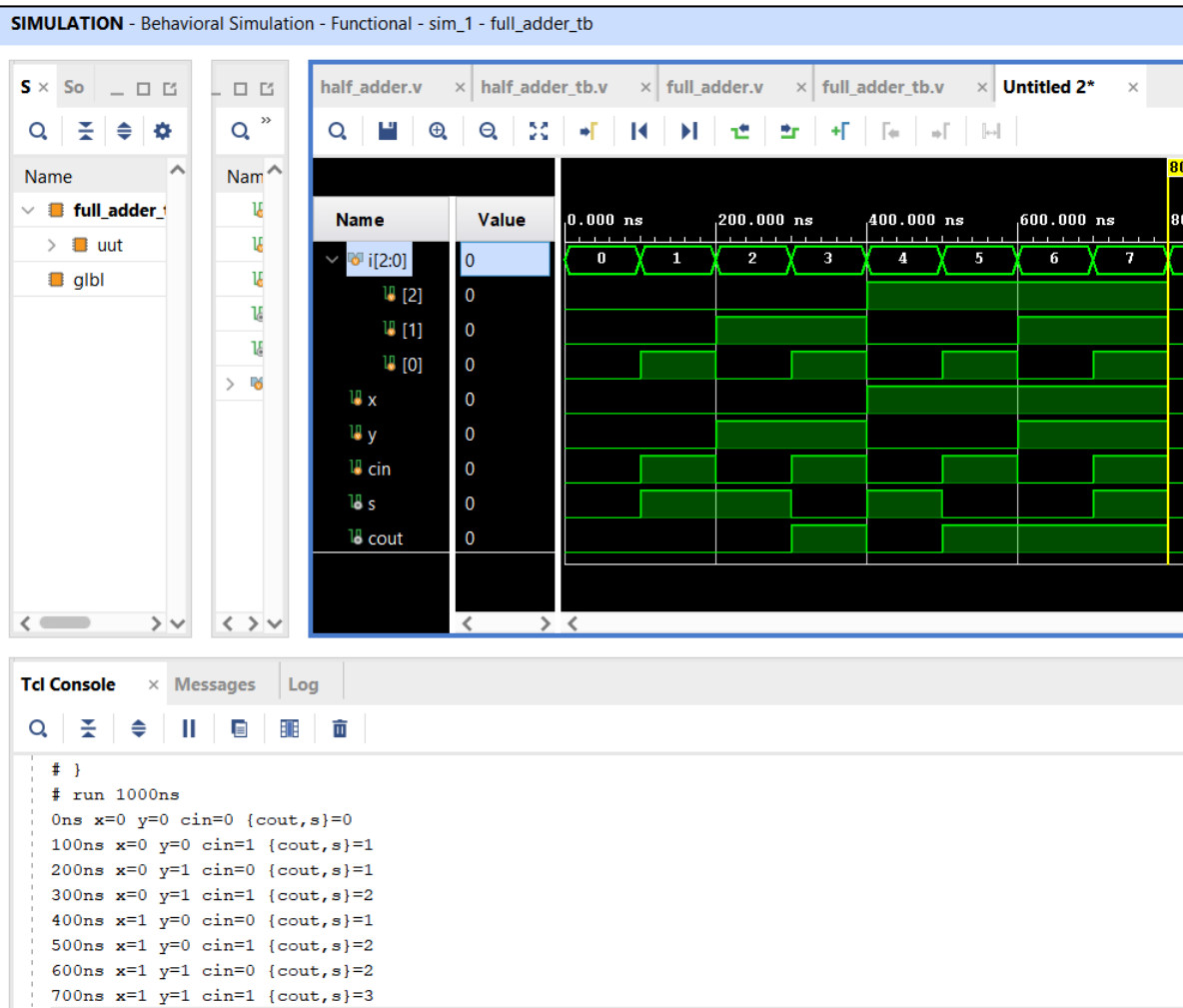
DESIGN EXAMPLE

❖ Verilog code and Testbench

□ Full Adder Simulation Waveform and Console



x	y	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Flow Navigator

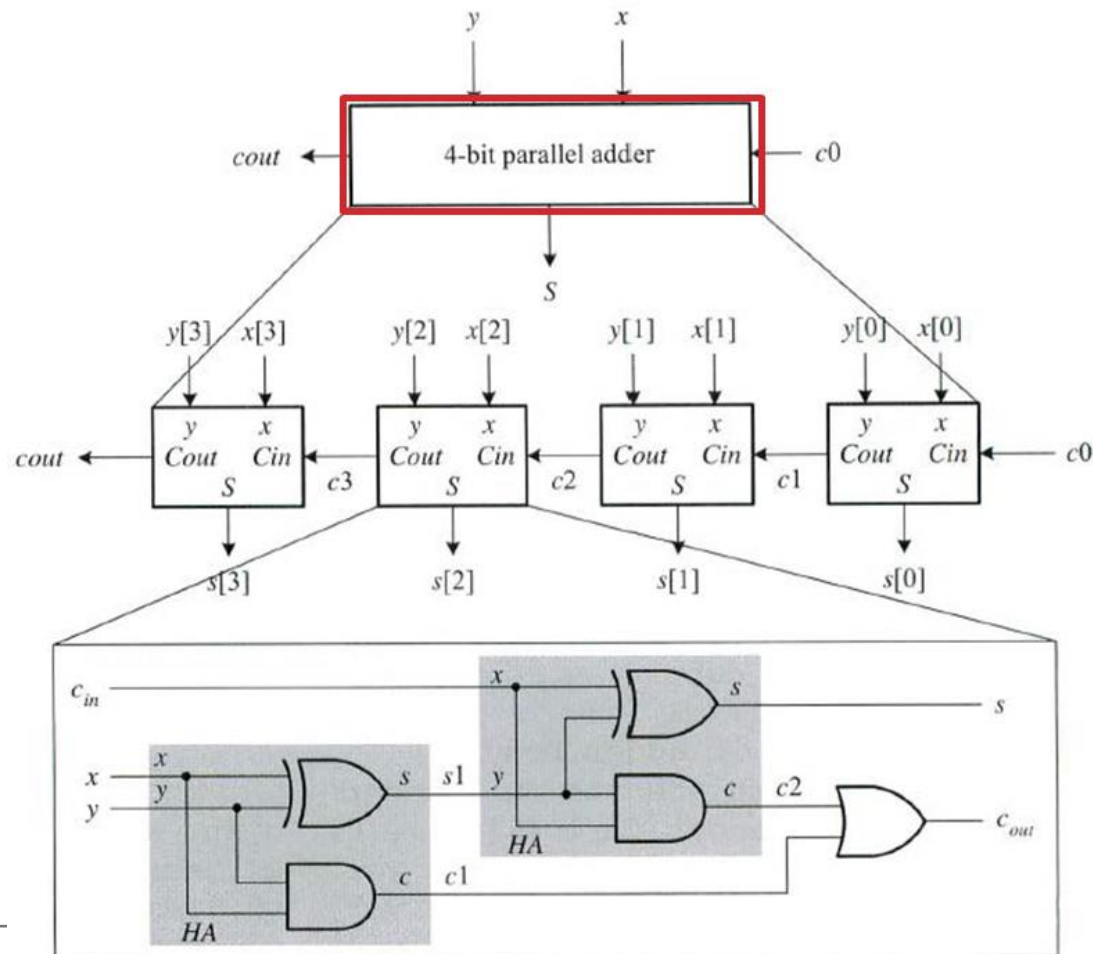
- PROJECT MANAGER
 - Settings
 - Add Sources
 - Language Templates
 - IP Catalog
- IP INTEGRATOR
 - Create Block Design
 - Open Block Design
 - Generate Block Design
- SIMULATION**
 - Run Simulation
- RTL ANALYSIS
 - Open Elaborated Design
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
- PROGRAM AND DEBUG



DESIGN EXAMPLE

❖ Verilog code and Testbench

□ 4-bit Adder Design

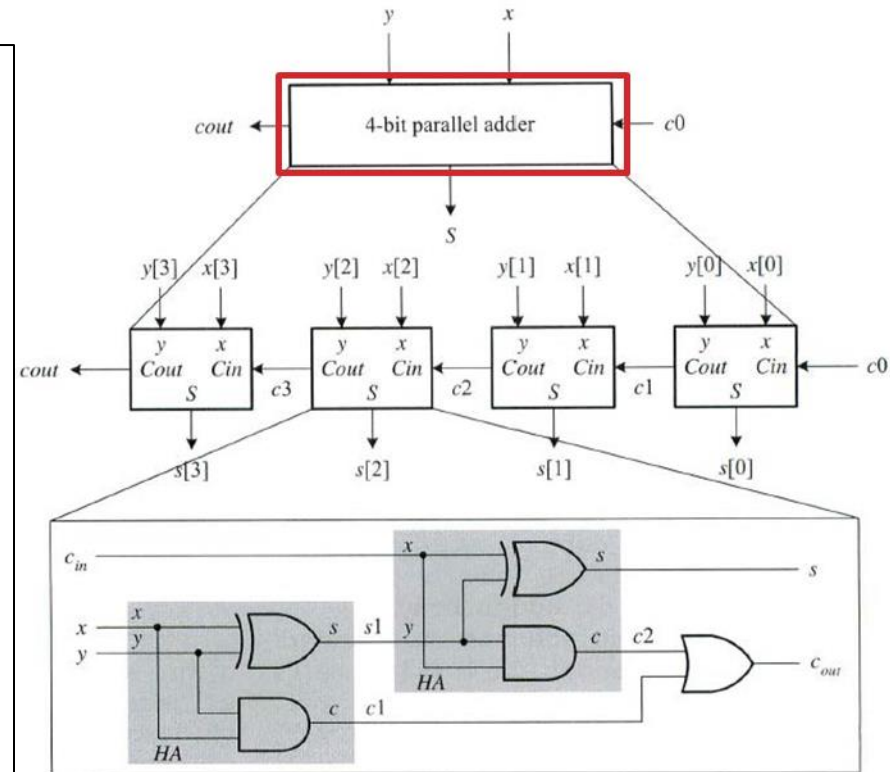


DESIGN EXAMPLE

❖ Verilog code and Testbench

□ Verilog Code for 4-bit Adder module

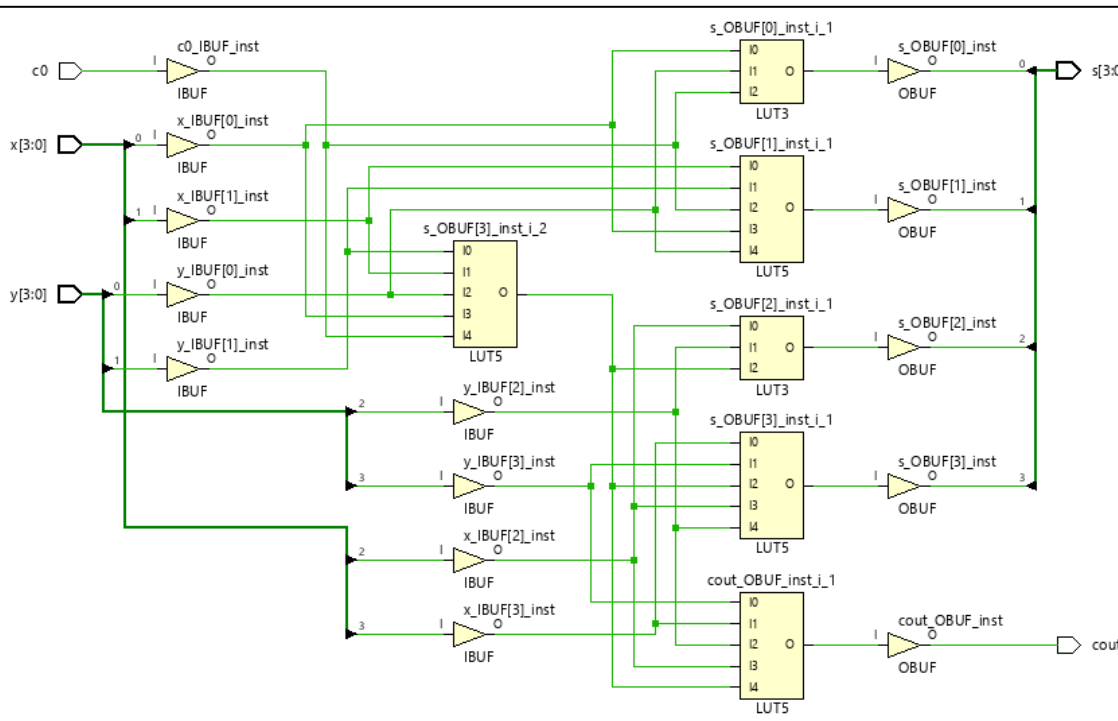
```
23 module four_bit_adder(x, y, c0, s, cout);  
24 //input/output declaration  
25 input [3:0] x, y;  
26 input c0;  
27 output [3:0] s;  
28 output cout;  
29  
30 //wire interconnection declaration  
31 wire c1, c2, c3;  
32  
33 //instantiate 4-bit adder using 4 full adders  
34 full_adder fa_1(x[0], y[0], c0, s[0], c1);  
35 full_adder fa_2(x[1], y[1], c1, s[1], c2);  
36 full_adder fa_3(x[2], y[2], c2, s[2], c3);  
37 full_adder fa_4(x[3], y[3], c3, s[3], cout);  
38  
39 endmodule
```



DESIGN EXAMPLE

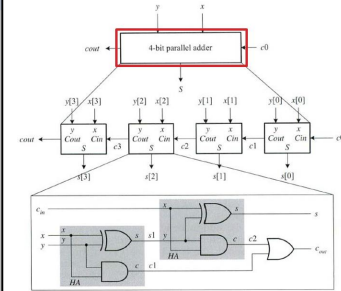
❖ Verilog code and Testbench

□ 4-bit Adder Schematic



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
 - Open Elaborated Design
- SYNTHESIS**
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
- PROGRAM AND DEBUG

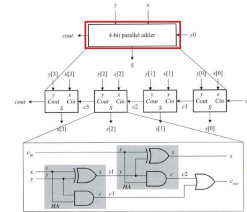


DESIGN EXAMPLE

❖ Verilog code and Testbench

□ 4-bit Adder Testbench

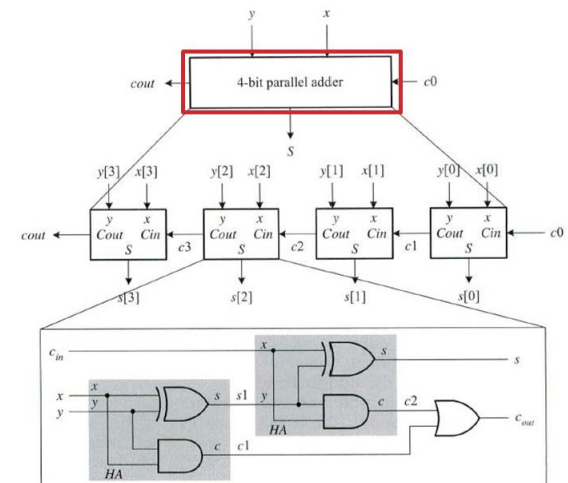
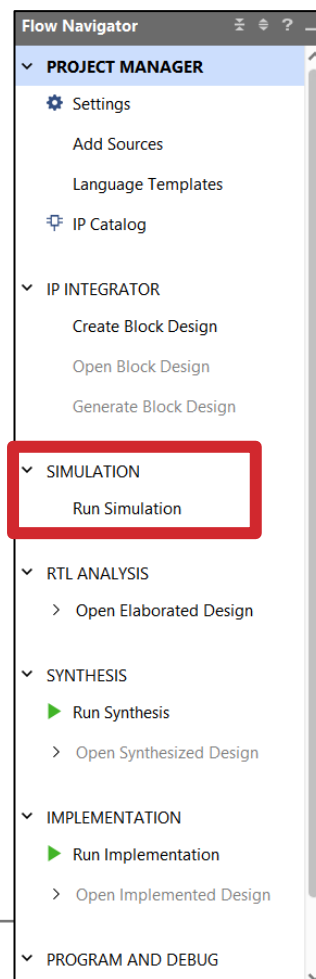
```
23 module four_bit_adder_tb();
24     //inputs
25     reg [3:0] x;
26     reg [3:0] y;
27     reg c0;
28     //outputs
29     wire [3:0] s;
30     wire cout;
31     //instantiate the Unit Under Test (UUT)
32     four_bit_adder uut(.x(x), .y(y), .c0(c0), .s(s), .cout(cout));
33
34     reg [7:0] i;
35     initial begin
36         //add stimulus here
37         for(i = 0; i <= 255; i = i + 1) begin
38             x[3:0] = i[7:0]; y[3:0] = i[3:0]; c0 = 1'b0;
39         end
40     end
41     initial begin
42         $monitor($realtime, "ns x=%h y=%h c0=%h {cout,s}=%h", x, y, c0, {cout, sum});
43     end
44
45 endmodule
```



DESIGN EXAMPLE

❖ Verilog code and Testbench

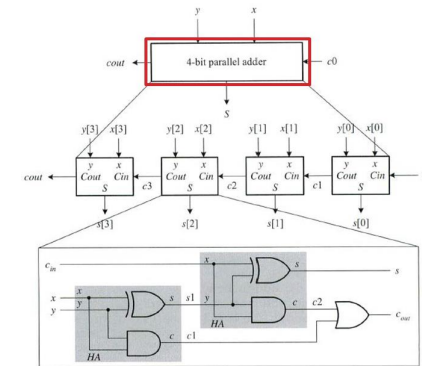
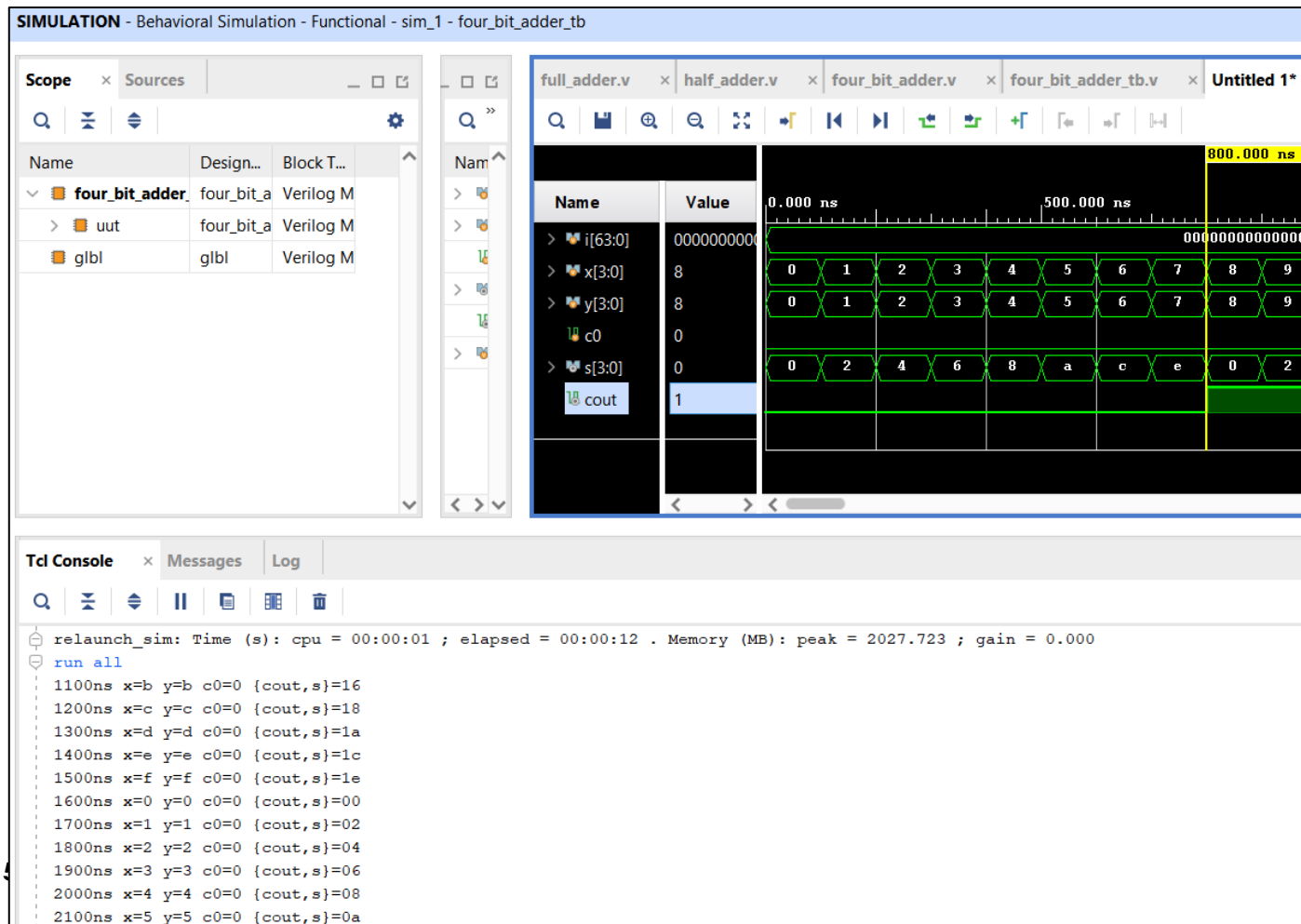
□ 4-bit Adder Simulation Waveform and Console



DESIGN EXAMPLE

❖ Verilog code and Testbench

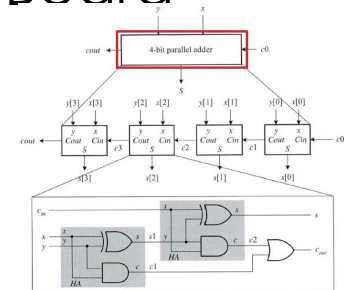
□ 4-bit Adder Simulation Waveform and Console



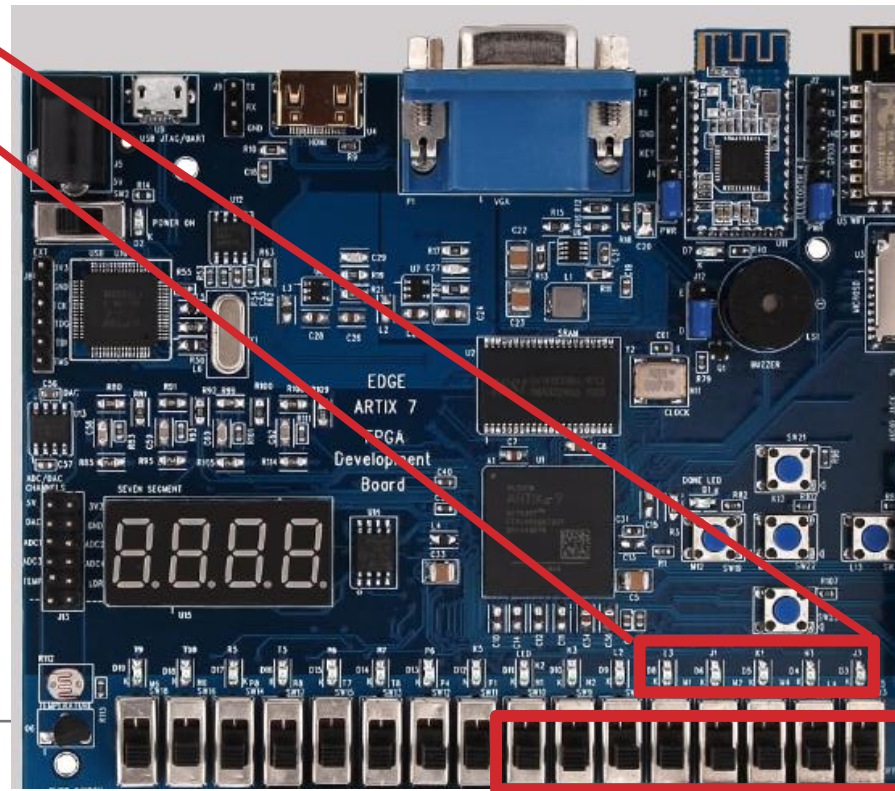
DESIGN EXAMPLE

❖ Implement 4-bit adder on Edge Artix 7 FPGA Board

- Input [3:0] x: {SW7, SW6, SW5, SW4}
- Input [3:0] y: {SW3, SW2, SW1, SW0}
- Output [3:0] s: {LED3, LED2, LED1, LED0}
- Output cout: LED4



Net Name	PIN MAP
LED0	J3
LED1	H3
LED2	J1
LED3	K1
LED4	L3



Net Name	PIN MAP
SW0	L5
SW1	L4
SW2	M4
SW3	M2
SW4	M1
SW5	N3
SW6	N2
SW7	N1

A

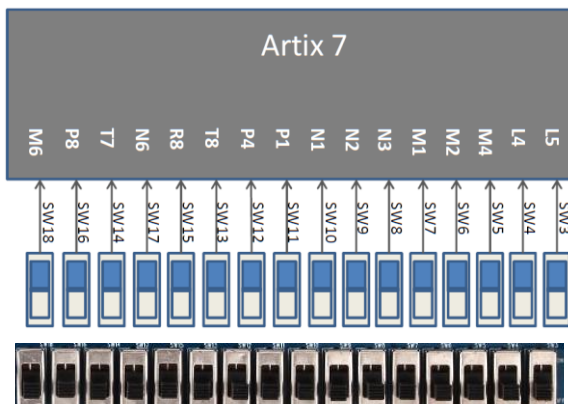




DESIGN EXAMPLE

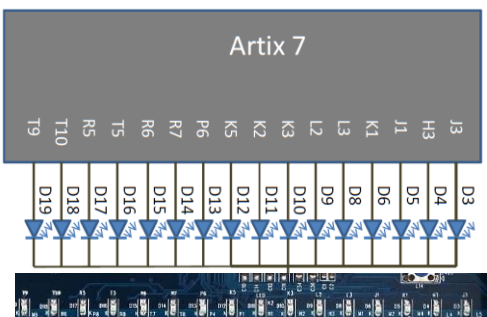
❖ Vivado project

□ 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
```

Net Name	PIN MAP
SW0	L5
SW1	L4
SW2	M4
SW3	M2
SW4	M1
SW5	N3
SW6	N2
SW7	N1



```
# 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
```

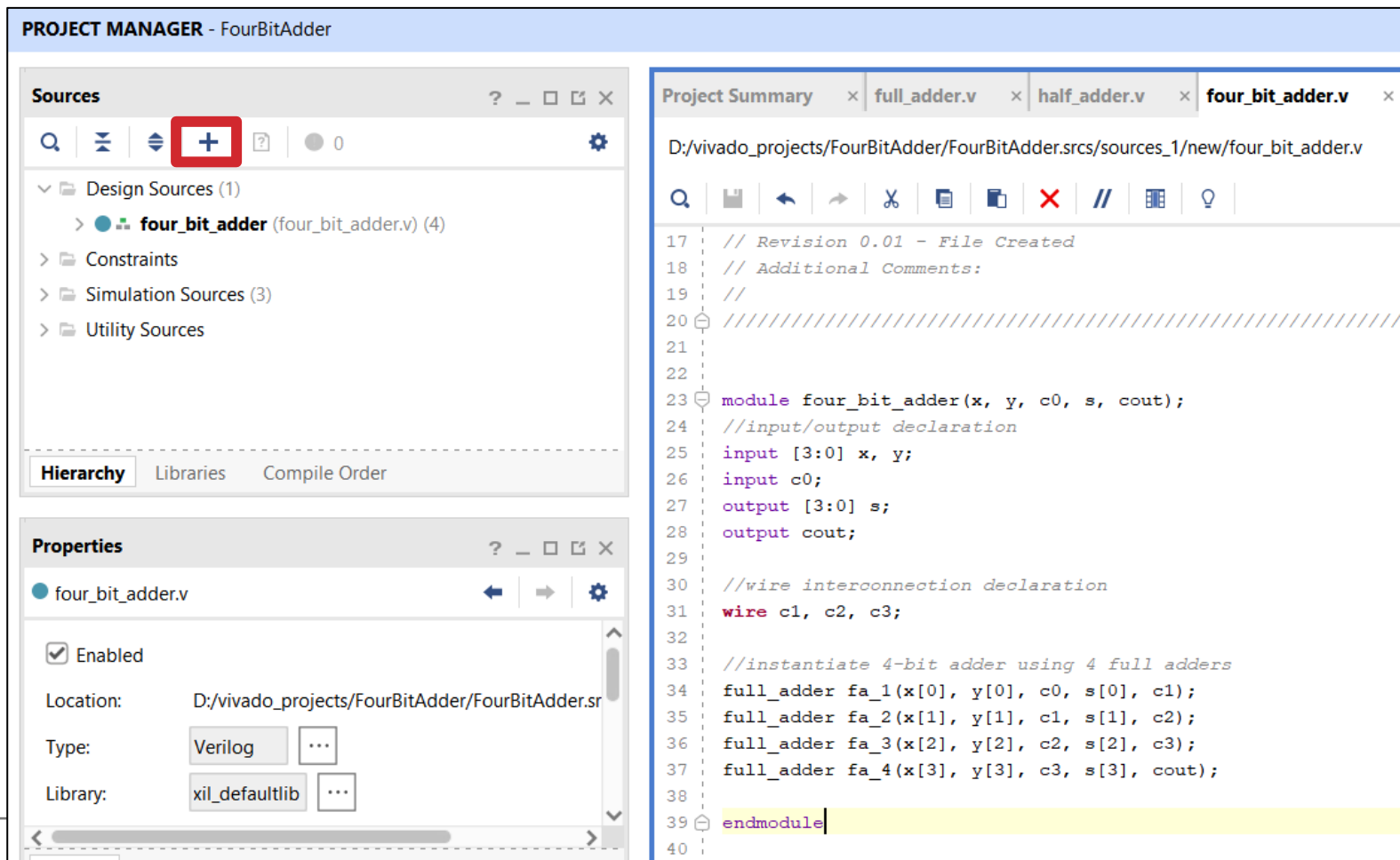
Net Name	PIN MAP
LED0	J3
LED1	H3
LED2	J1
LED3	K1
LED4	L3



DESIGN EXAMPLE

❖ Vivado project

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



The screenshot displays the Vivado Project Manager and Code Editor. The Project Manager on the left shows the 'Sources' tab with a red box highlighting the '+' icon for adding a new source. Below it, the 'Properties' tab for 'four_bit_adder.v' is visible, showing it is enabled and located at 'D:/vivado_projects/FourBitAdder/FourBitAdder.sr'. The Code Editor on the right shows the Verilog code for the 'four_bit_adder' module, which instantiates four 'full_adder' modules.

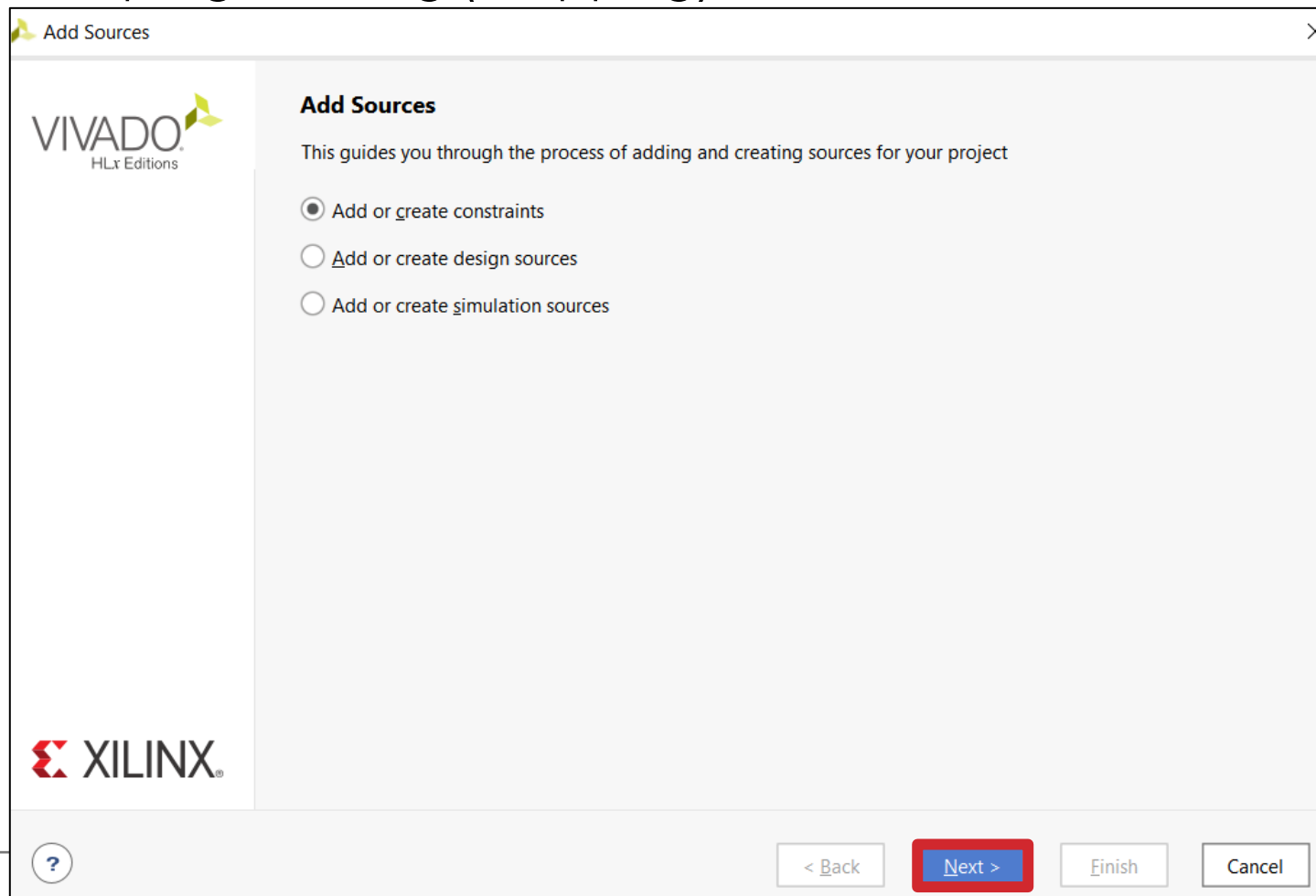
```
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module four_bit_adder(x, y, c0, s, cout);
24 //input/output declaration
25 input [3:0] x, y;
26 input c0;
27 output [3:0] s;
28 output cout;
29
30 //wire interconnection declaration
31 wire c1, c2, c3;
32
33 //instantiate 4-bit adder using 4 full adders
34 full_adder fa_1(x[0], y[0], c0, s[0], c1);
35 full_adder fa_2(x[1], y[1], c1, s[1], c2);
36 full_adder fa_3(x[2], y[2], c2, s[2], c3);
37 full_adder fa_4(x[3], y[3], c3, s[3], cout);
38
39 endmodule
40
```



DESIGN EXAMPLE

❖ Vivado project

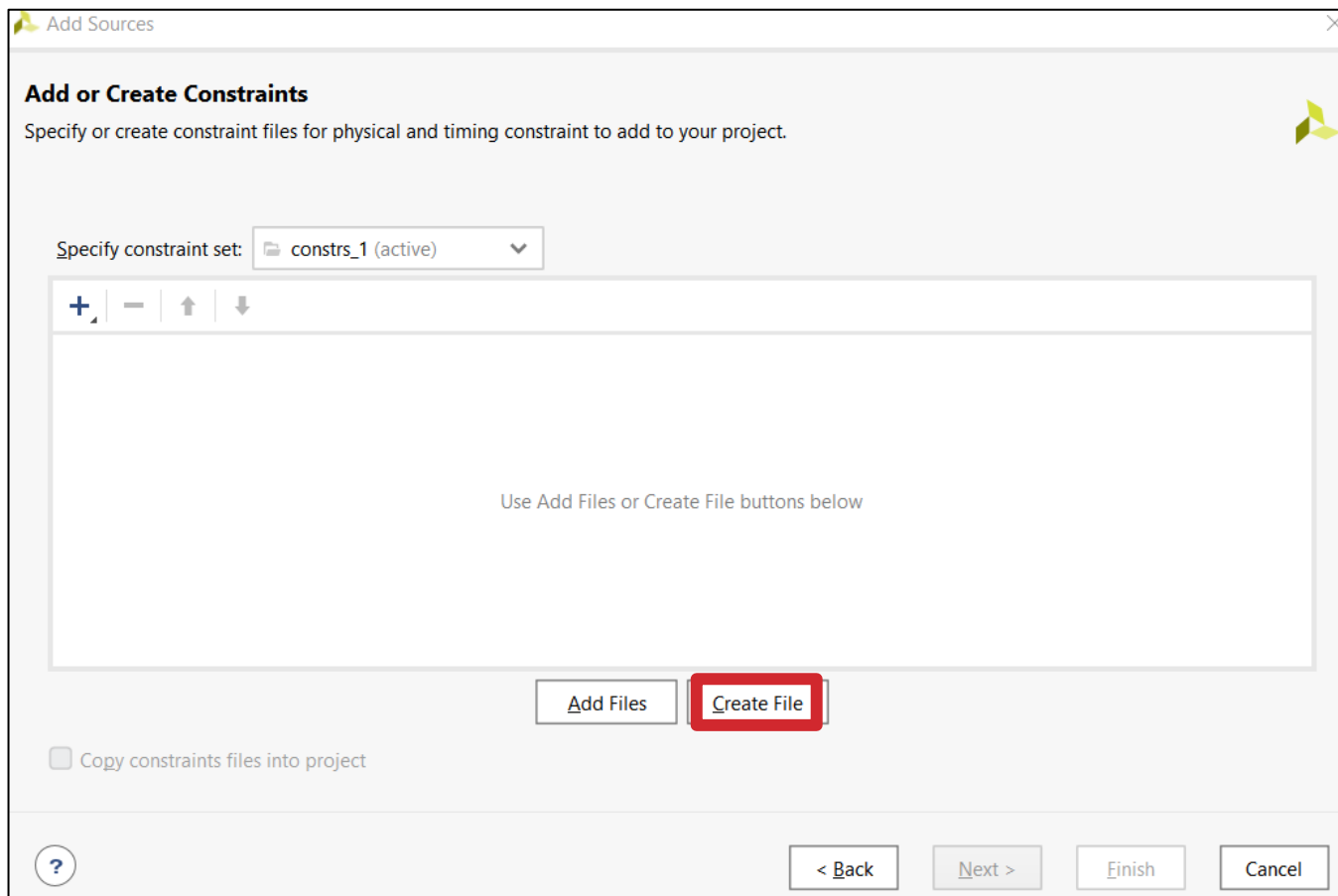
- ❑ FPGA programming (mapping) – Create a constraint file



DESIGN EXAMPLE

❖ Vivado project

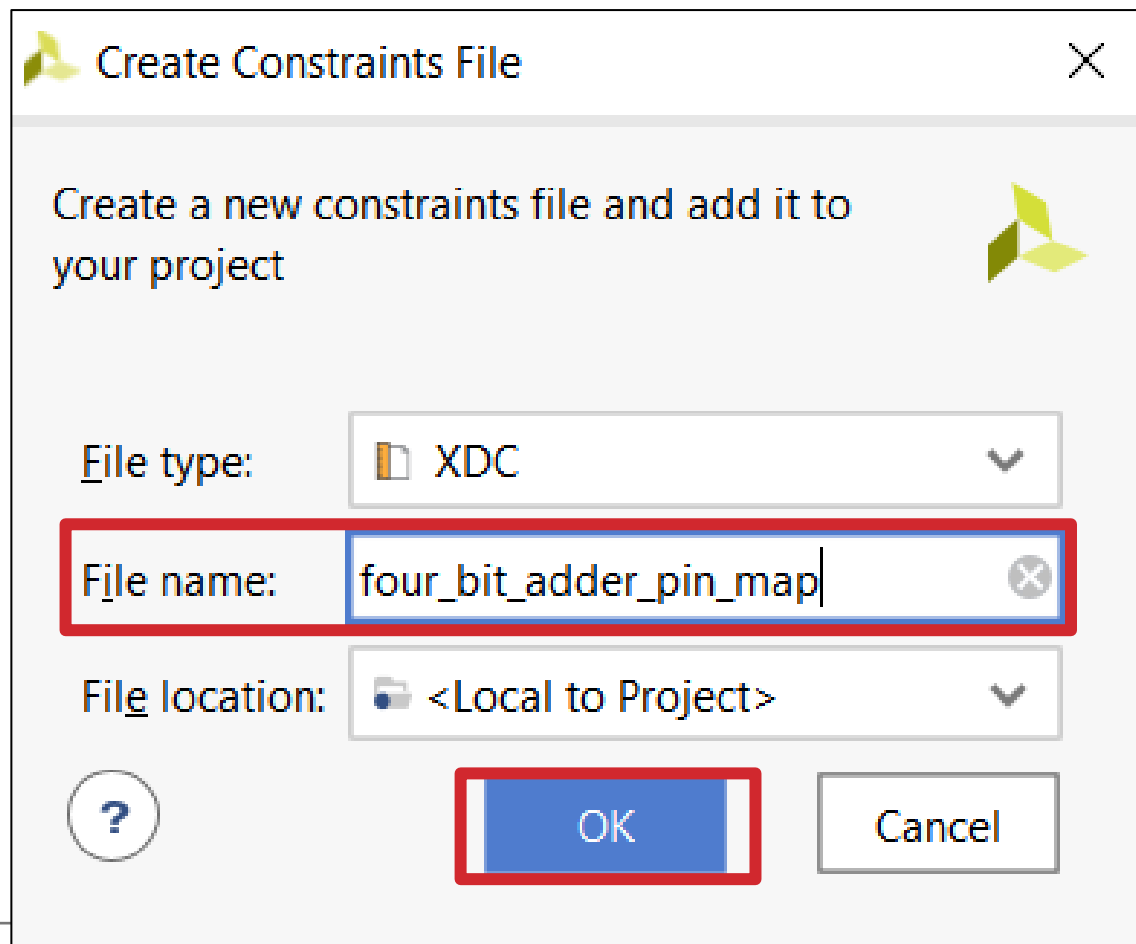
- ❑ FPGA programming (mapping) – Create a constraint file





DESIGN EXAMPLE

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



DESIGN EXAMPLE

❖ Vivado project

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

Add or Create Constraints

Specify or create constraint files for physical and timing constraint to add to your project.

Specify constraint set: constrs_1 (active)

Constraint File	Location
four_bit_adder_pin_map.xdc	<Local to Project>

☐ Copy constraints files into project

Add Files Create File

< Back Next > Finish Cancel

DESIGN EXAMPLE

❖ Vivado project

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

PROJECT MANAGER - FourBitAdder

Sources

- Design Sources (1)
 - four_bit_adder (four_bit_adder.v) (4)
- Constraints (1)
 - constrs_1 (1)
 - four bit adder pin map.xdc
- Simulation Sources (3)
- Utility Sources

Source File Properties

four_bit_adder_pin_map.xdc

- ☒ Enabled
- Location: D:/vivado_projects/FourBitAdder/FourBitAdder.sr
- Type: XDC
- Size: 0.0 KB

Project Summary | full_adder.v | half_adder.v | four_bit_adder.v | four_bit_adder_tb.v | four_bit_adder_pin_map.xdc

D:/vivado_projects/FourBitAdder/FourBitAdder.srcs/constrs_1/new/four_bit_adder_pin_map.xdc

1

DESIGN EXAMPLE

❖ Vivado project

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

```
Project Summary x full_adder.v x half_adder.v x four_bit_adder.v x four_bit_adder_tb.v x four_bit_ac
D:/vivado_projects/FourBitAdder/FourBitAdder.srscs/constrs_1/new/four_bit_adder_pin_map.xdc

# Switches
1 set_property -dict { PACKAGE_PIN L5 IOSTANDARD LVCMOS33 } [get_ports { y[0] }]; #LSB
2 set_property -dict { PACKAGE_PIN L4 IOSTANDARD LVCMOS33 } [get_ports { y[1] }];
3 set_property -dict { PACKAGE_PIN M4 IOSTANDARD LVCMOS33 } [get_ports { y[2] }];
4 set_property -dict { PACKAGE_PIN M2 IOSTANDARD LVCMOS33 } [get_ports { y[3] }];
5 set_property -dict { PACKAGE_PIN M1 IOSTANDARD LVCMOS33 } [get_ports { x[0] }];
6 set_property -dict { PACKAGE_PIN N3 IOSTANDARD LVCMOS33 } [get_ports { x[1] }];
7 set_property -dict { PACKAGE_PIN N2 IOSTANDARD LVCMOS33 } [get_ports { x[2] }];
8 set_property -dict { PACKAGE_PIN N1 IOSTANDARD LVCMOS33 } [get_ports { x[3] }]; #MSB
9
10
11 # LEDs
12 set_property -dict { PACKAGE_PIN J3 IOSTANDARD LVCMOS33 } [get_ports { s[0] }]; #LSB
13 set_property -dict { PACKAGE_PIN H3 IOSTANDARD LVCMOS33 } [get_ports { s[1] }];
14 set_property -dict { PACKAGE_PIN J1 IOSTANDARD LVCMOS33 } [get_ports { s[2] }];
15 set_property -dict { PACKAGE_PIN K1 IOSTANDARD LVCMOS33 } [get_ports { s[3] }];
16 set_property -dict { PACKAGE_PIN L3 IOSTANDARD LVCMOS33 } [get_ports { cout }]; #MSB
```

Net Name	PIN MAP
SW0	L5
SW1	L4
SW2	M4
SW3	M2
SW4	M1
SW5	N3
SW6	N2
SW7	N1

Net Name	PIN MAP
LED0	J3
LED1	H3
LED2	J1
LED3	K1
LED4	L3

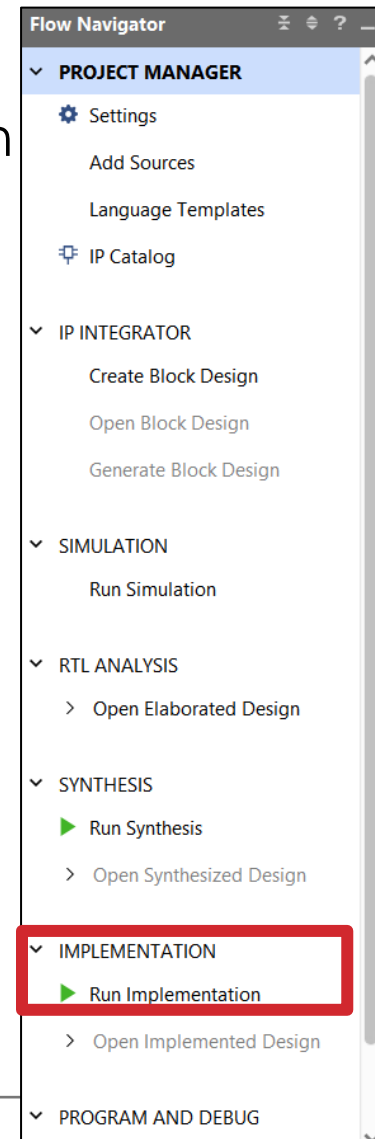




DESIGN EXAMPLE

❖ Vivado project

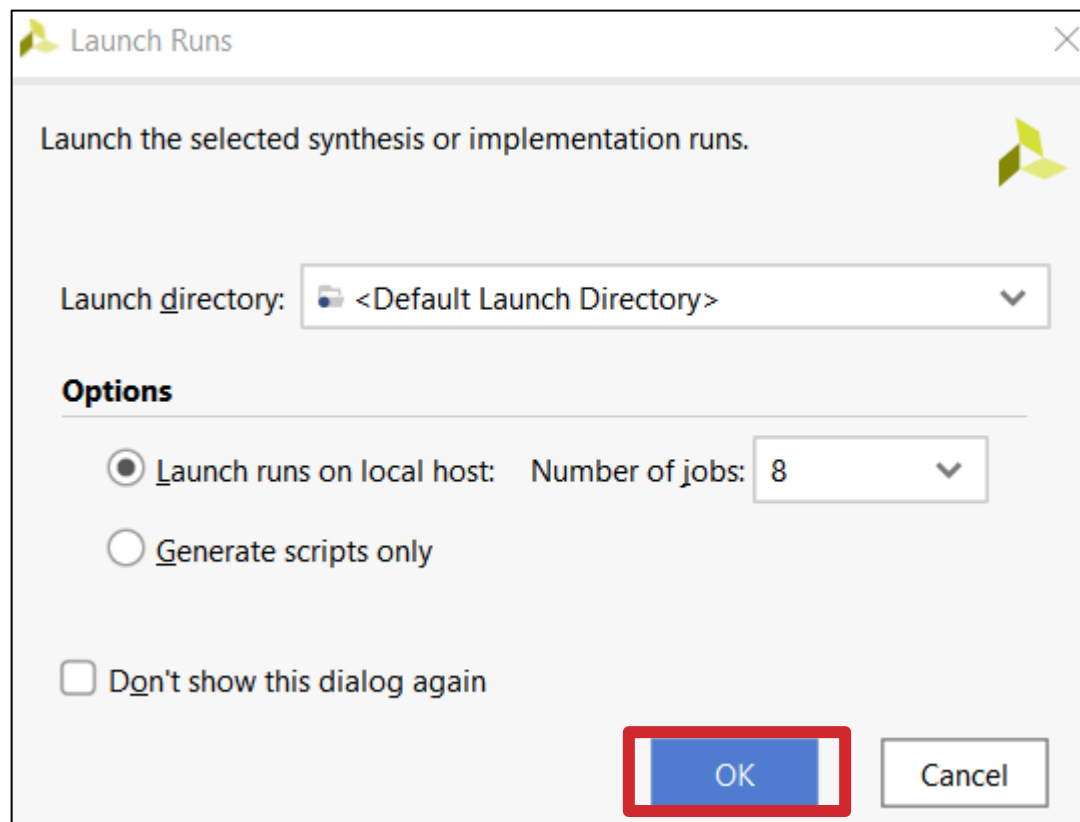
- FPGA programming (mapping) – Implement Design





DESIGN EXAMPLE

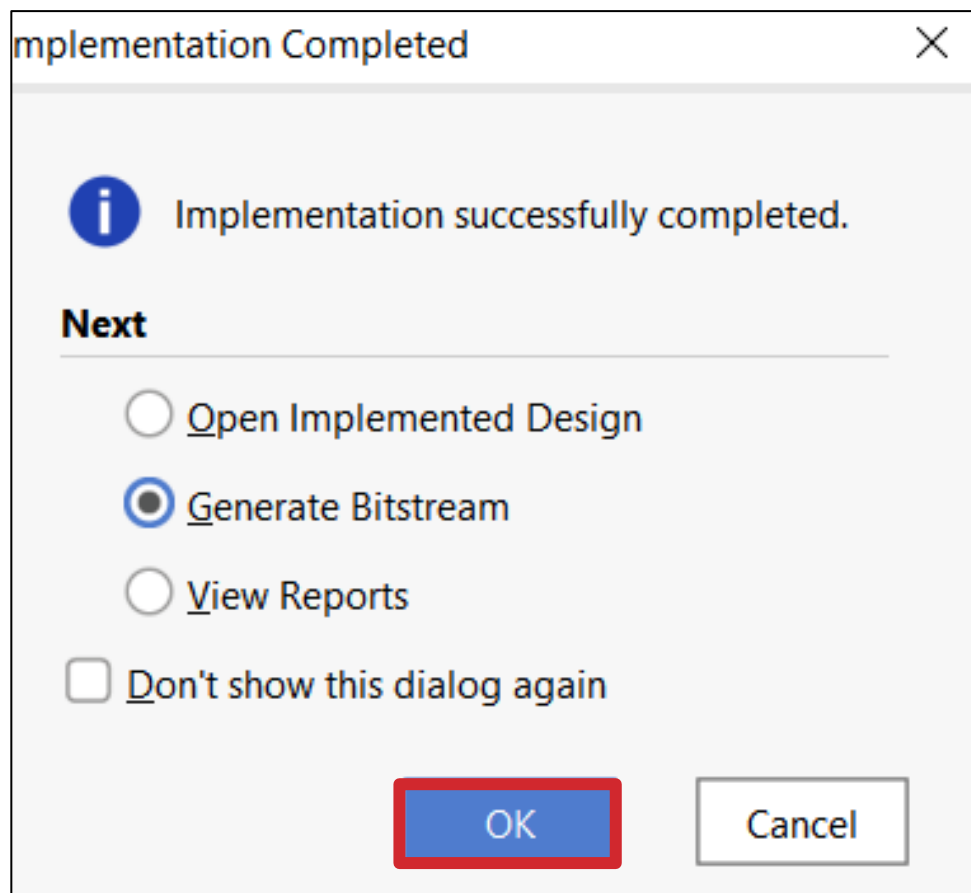
- ❖ Vivado project
 - FPGA programming (mapping) – Implement Design





DESIGN EXAMPLE

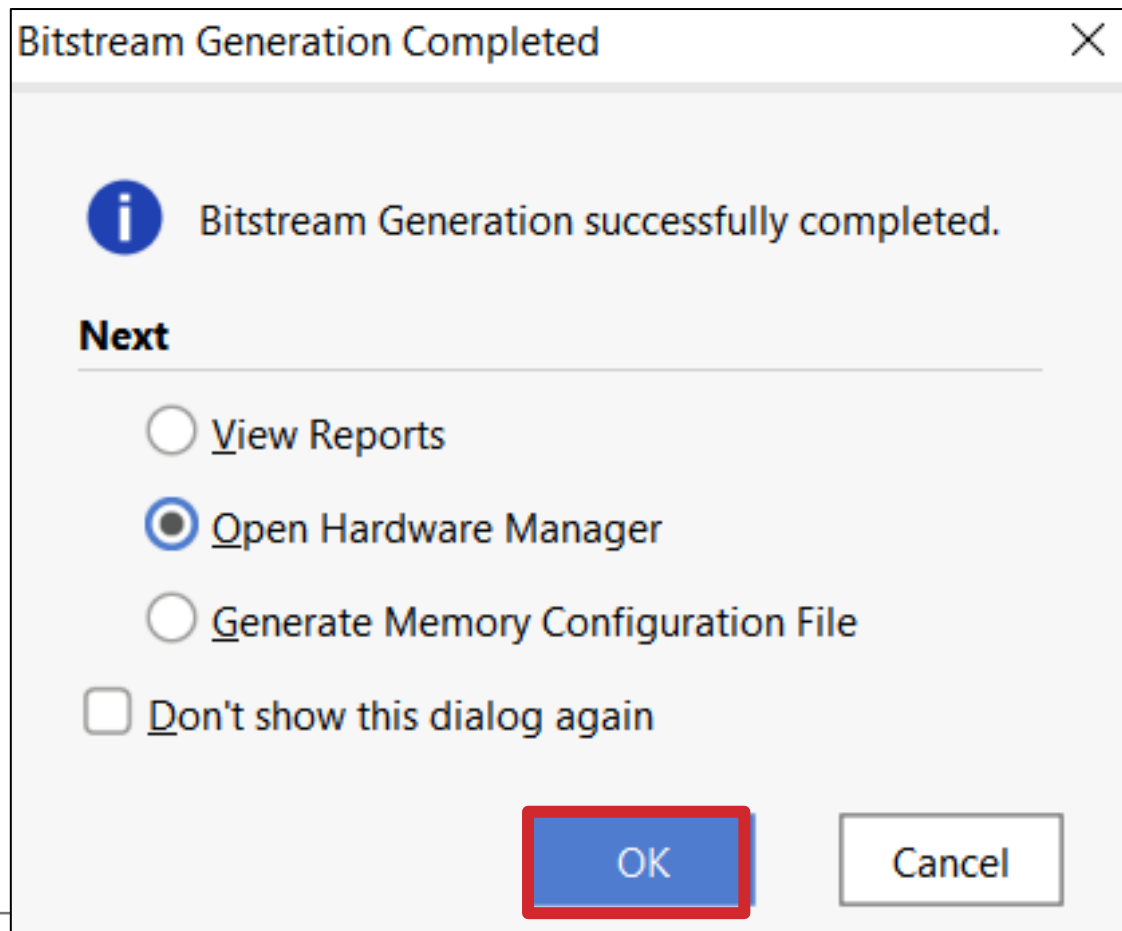
- ❖ Vivado project
 - FPGA programming (mapping) – Generate Programming File





DESIGN EXAMPLE

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



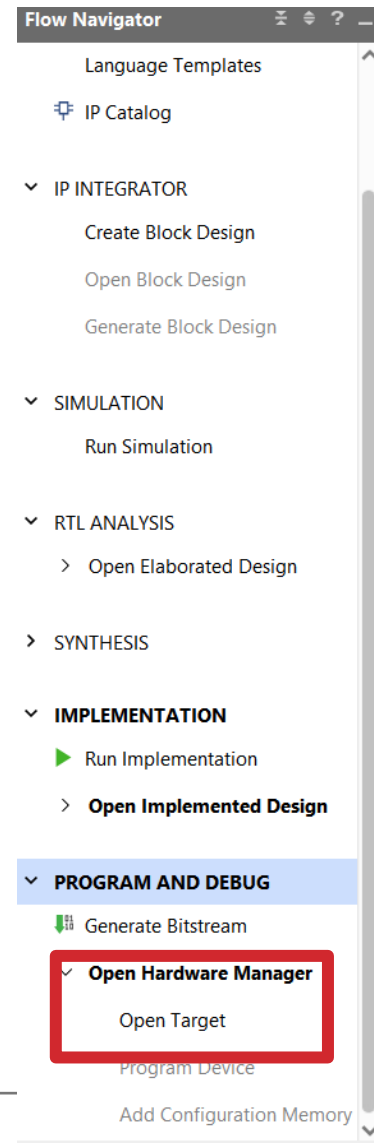


DESIGN EXAMPLE

❖ Vivado project

□ FPGA programming

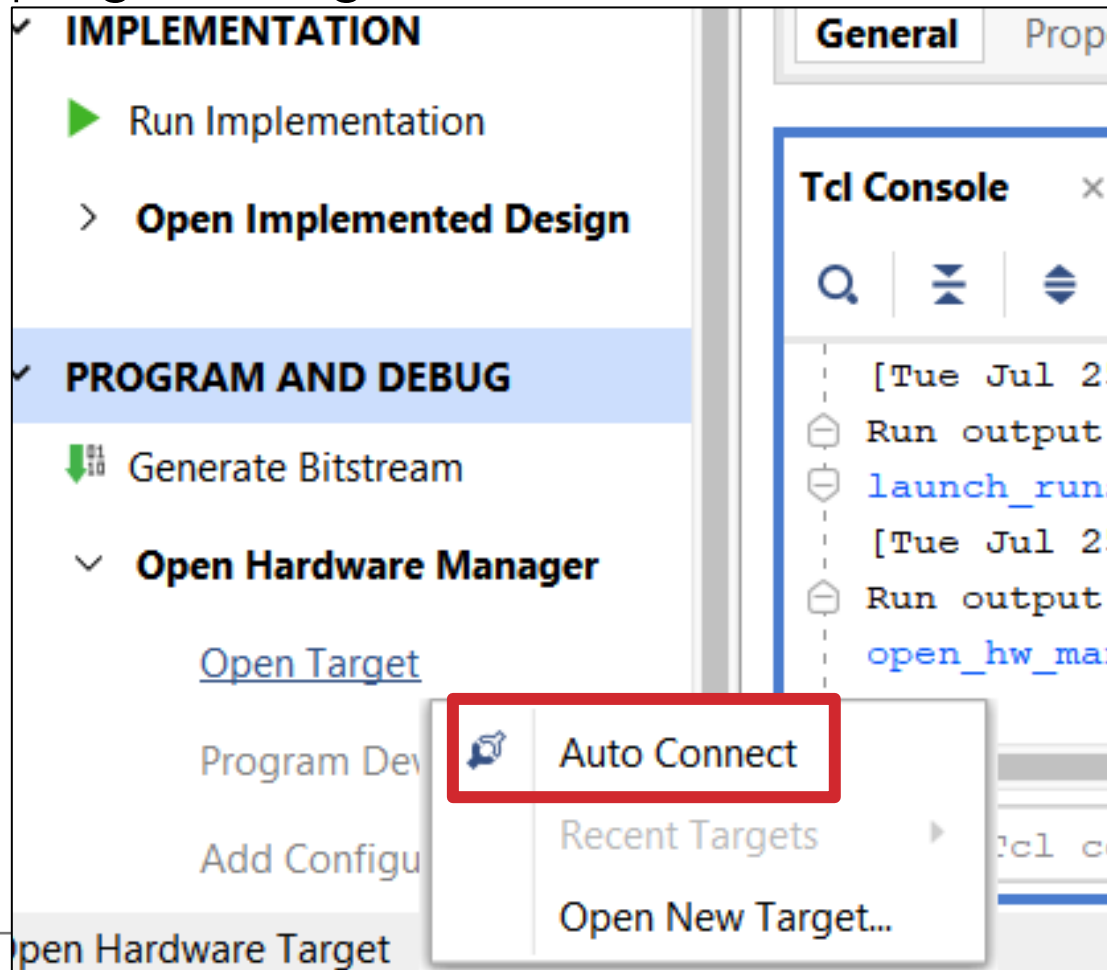
- Connect the FPGA board to the computer and click on Open Target





DESIGN EXAMPLE

- ❖ Vivado project
 - FPGA programming



DESIGN EXAMPLE

- ❖ Vivado project
- FPGA programming

The screenshot displays the Vivado Hardware Manager interface. At the top, the title bar reads "HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210319A288EEA". Below this, a green status bar contains the message "There are no debug cores" and two buttons: "Program device" (highlighted with a red box) and "Refresh device".

The main area is divided into two panes. The left pane, titled "Hardware", shows a tree view of the hardware components. A red box highlights the following structure:

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

The right pane, titled "Hardware Device Properties", shows the properties for the selected device, "xc7a35t_0". The "General" tab is active, displaying the device name and a scrollable list of properties.

On the far right, a code editor shows the contents of "gate.v". The code includes comments for project name, target devices, tool versions, description, dependencies, and revision, followed by a Verilog module definition:

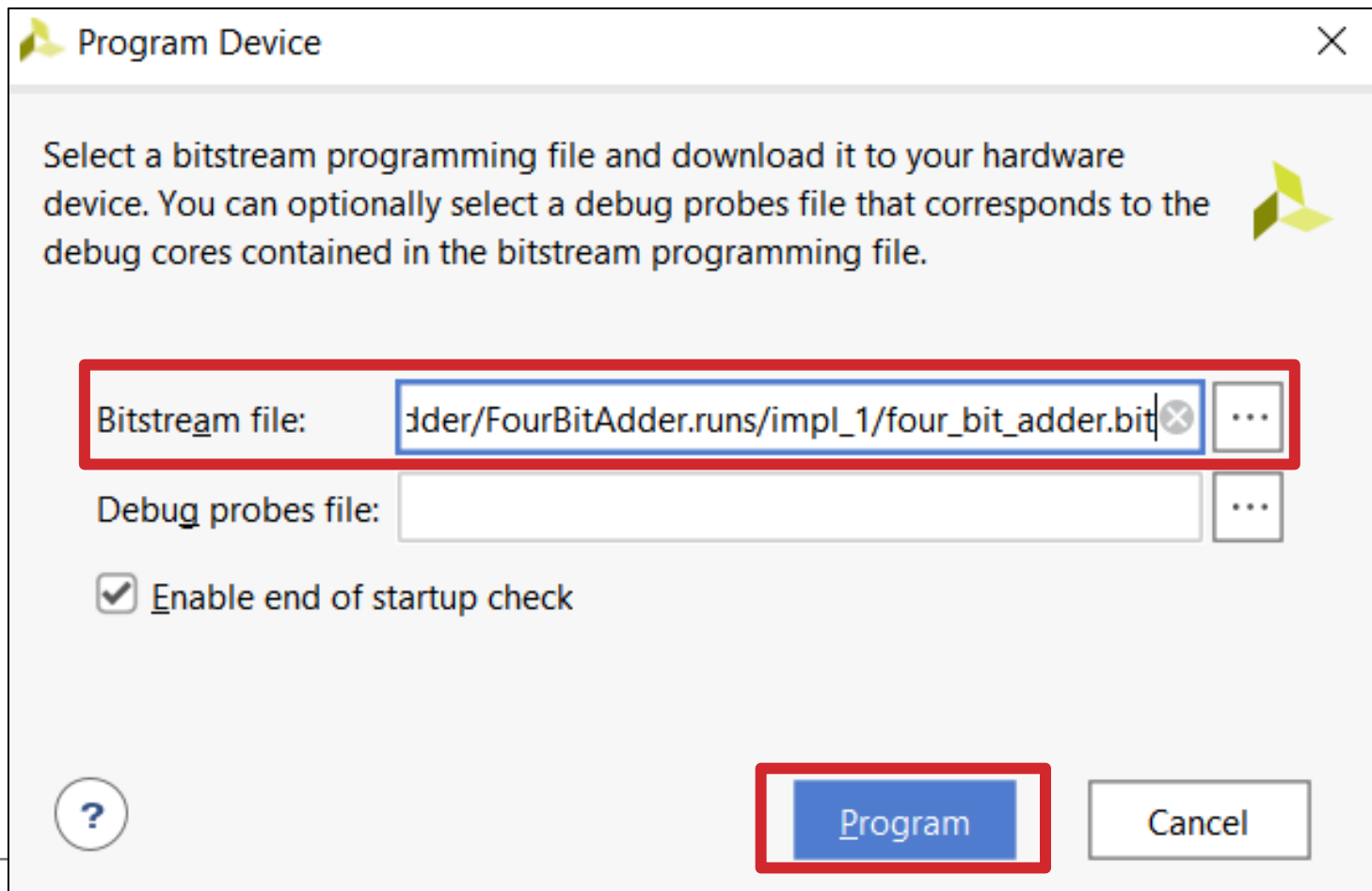
```
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////
21
22
23 module gate(A, B, X, Y, Z);
24     //Input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





DESIGN EXAMPLE

- ❖ Vivado project
 - FPGA programming



DESIGN EXAMPLE

❖ Programming FPGA

□ Board Demo

- Example: $15 + 15 = 30$

{c_out, sum[3:0]}

Binary: 11110

Decimal: 30

{x[3:0], y[3:0]}

Binary: {11111, 1111}

Decimal: {15, 15}



DESIGN EXAMPLE

❖ Programming FPGA

□ Board Demo

- Example: $15 + 15 = 30$

{c_out, sum[3:0]}

Binary: 11110

Decimal: 30

{x[3:0], y[3:0]}

Binary: {11111, 1111}

Decimal: {15, 15}

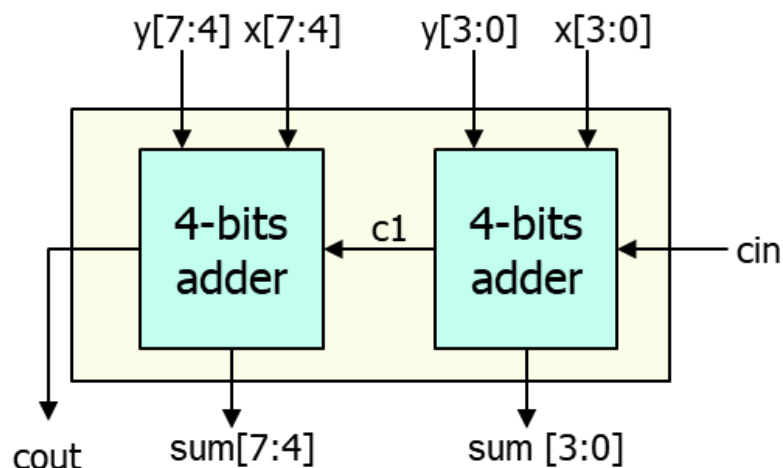




EXERCISE

❖ 8-BIT ADDER

- Submit as a PPT file including the following content
- 8-bit Adder Verilog Source code with comments
- Synthesis netlist schematic
- Synthesis results
- 8-bit Adder Verilog Testbench with comments
- Simulation waveform including analysis
- Board Test



Port	Width	I/O	Description
x	8	Input	Input Data
y	8	Input	Input Data
cin	1	Input	Carry Input
sum	8	Output	Result Output
cout	1	Output	Carry Output

8-bits adder

