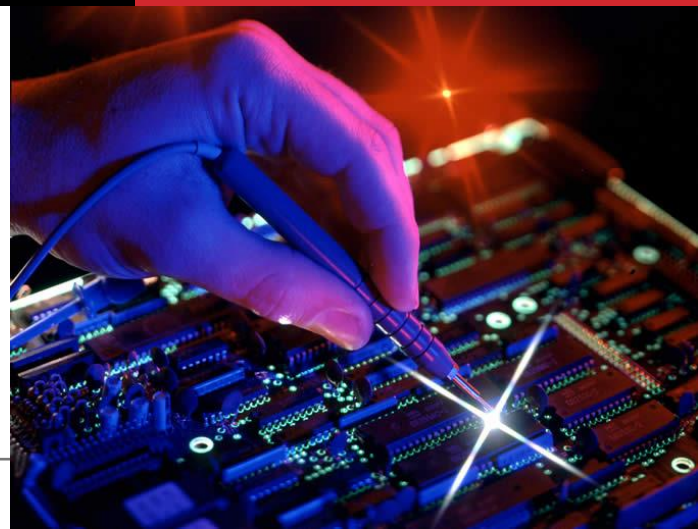
Advanced Computer Architecture & Advanced Microprocessor System

VIVADO DESIGN EXAMPLE

Dennis A. N. Gookyi



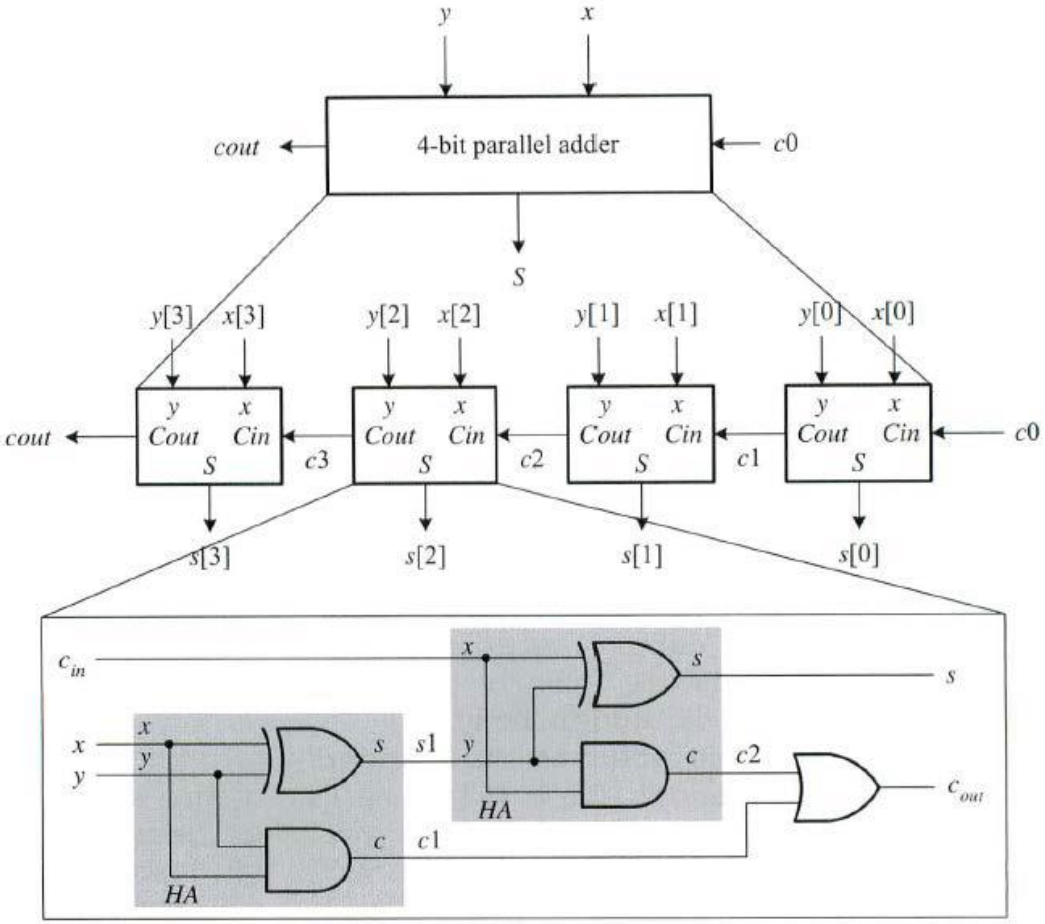
CONTENTS

- **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

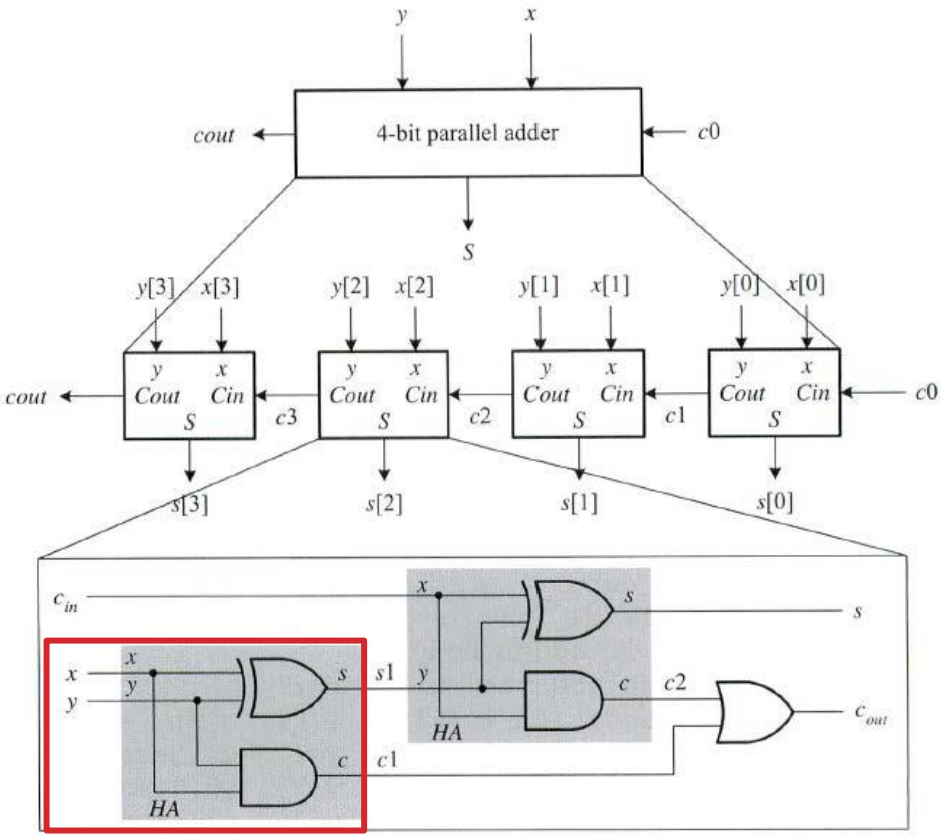
Design Example

- 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



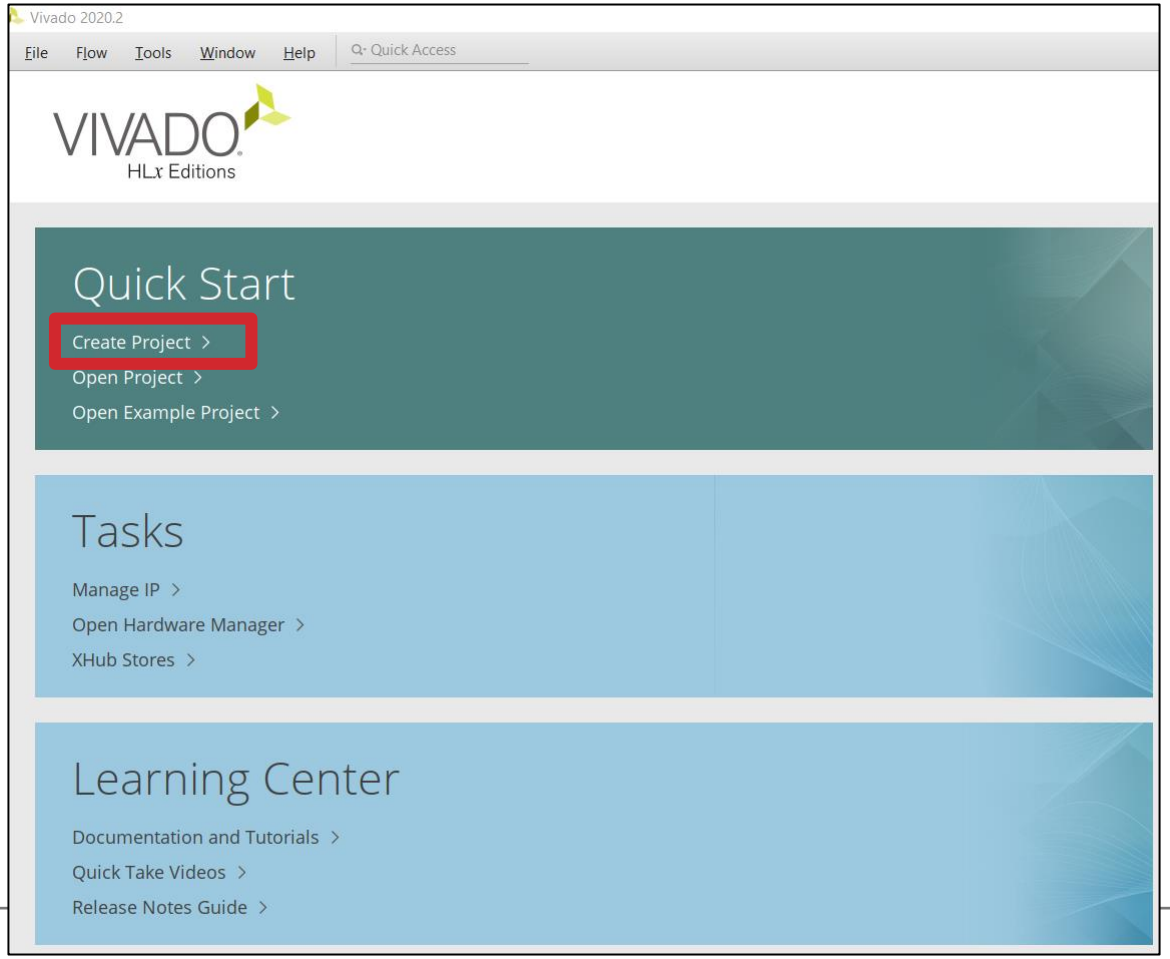
half_adder

Truth Table

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

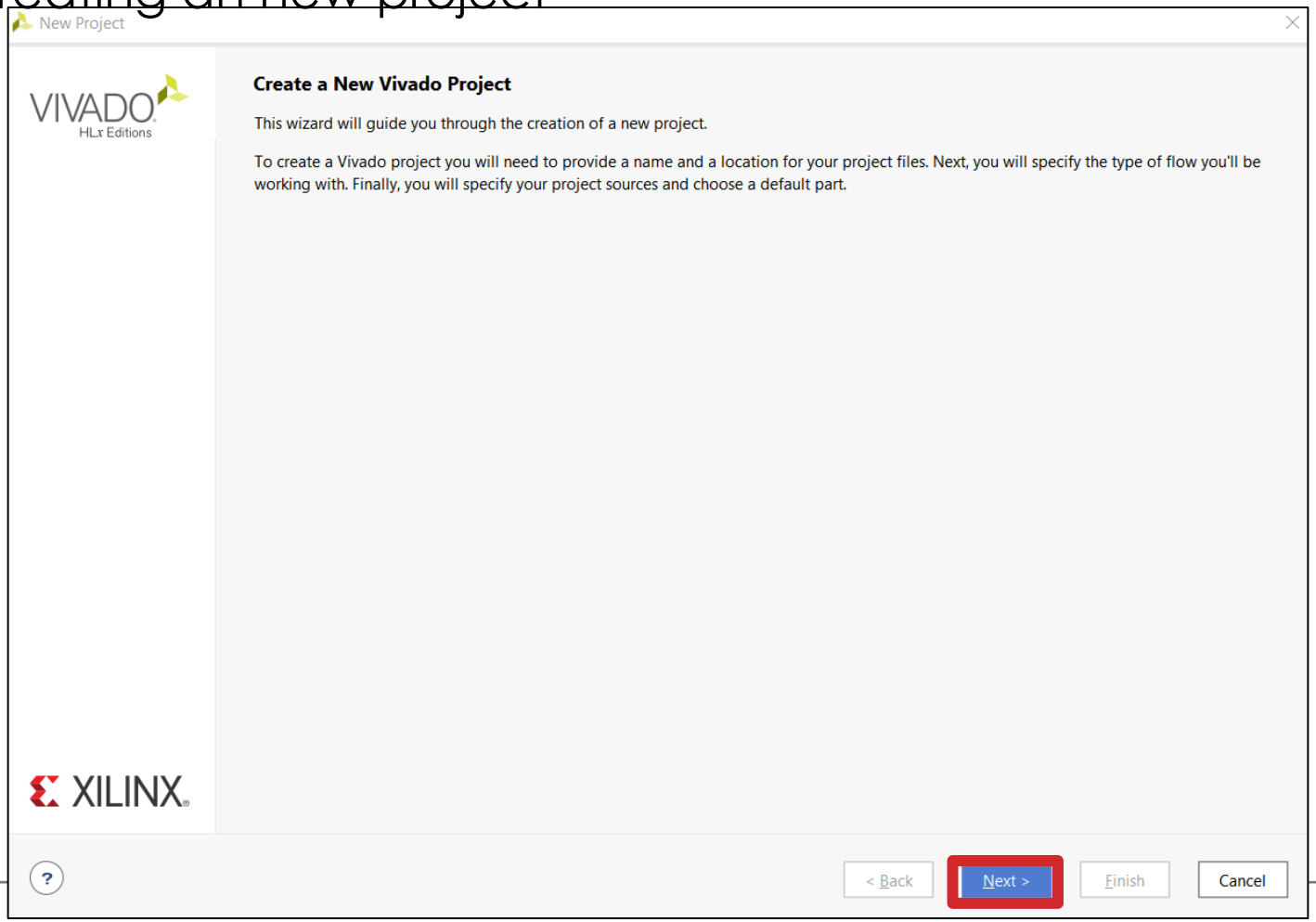
DESIGN EXAMPLE

- Vivado project
  - Creating a new project



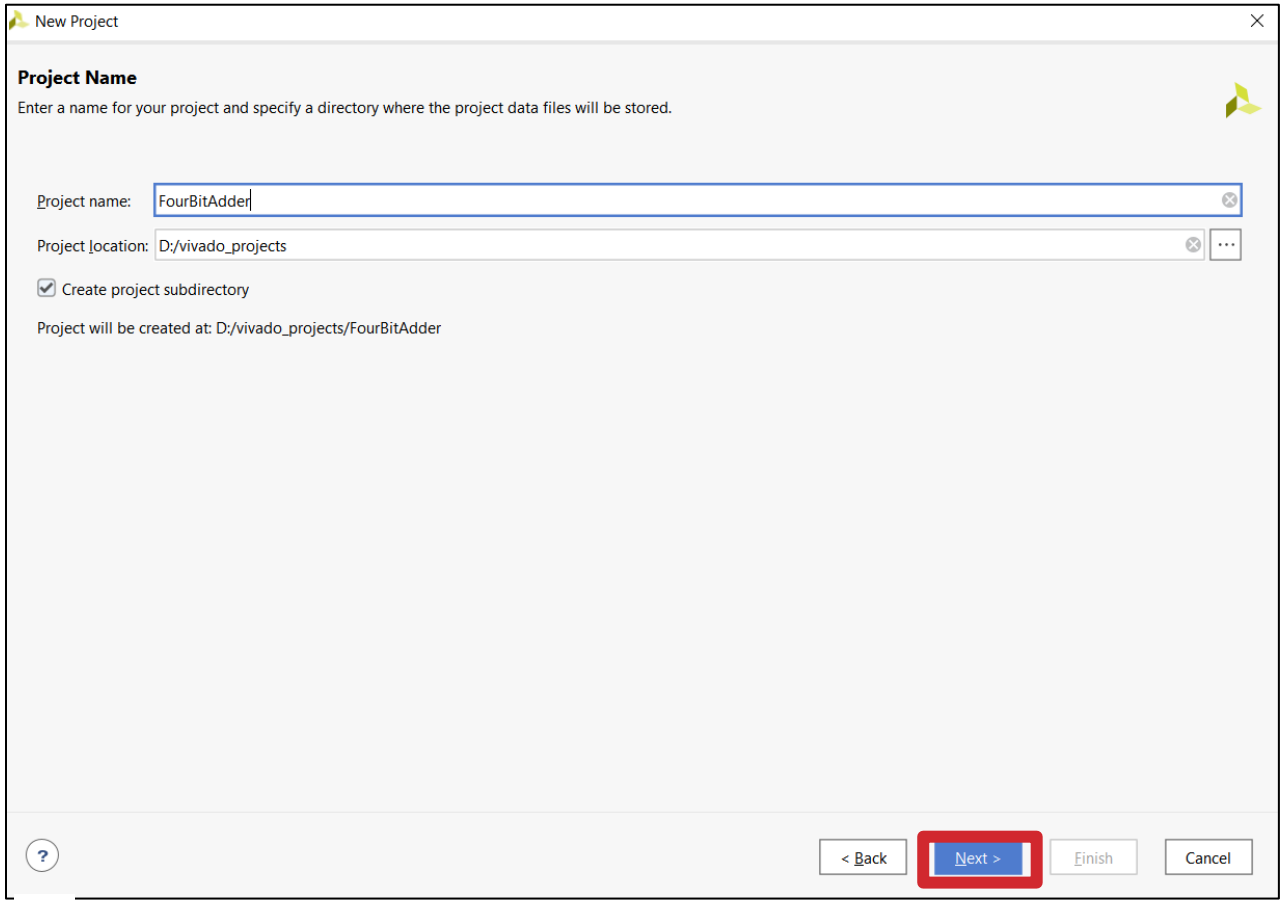
DESIGN EXAMPLE

- Vivado project
  - Creating an new project



DESIGN EXAMPLE

- Vivado project
  - Creating an new project

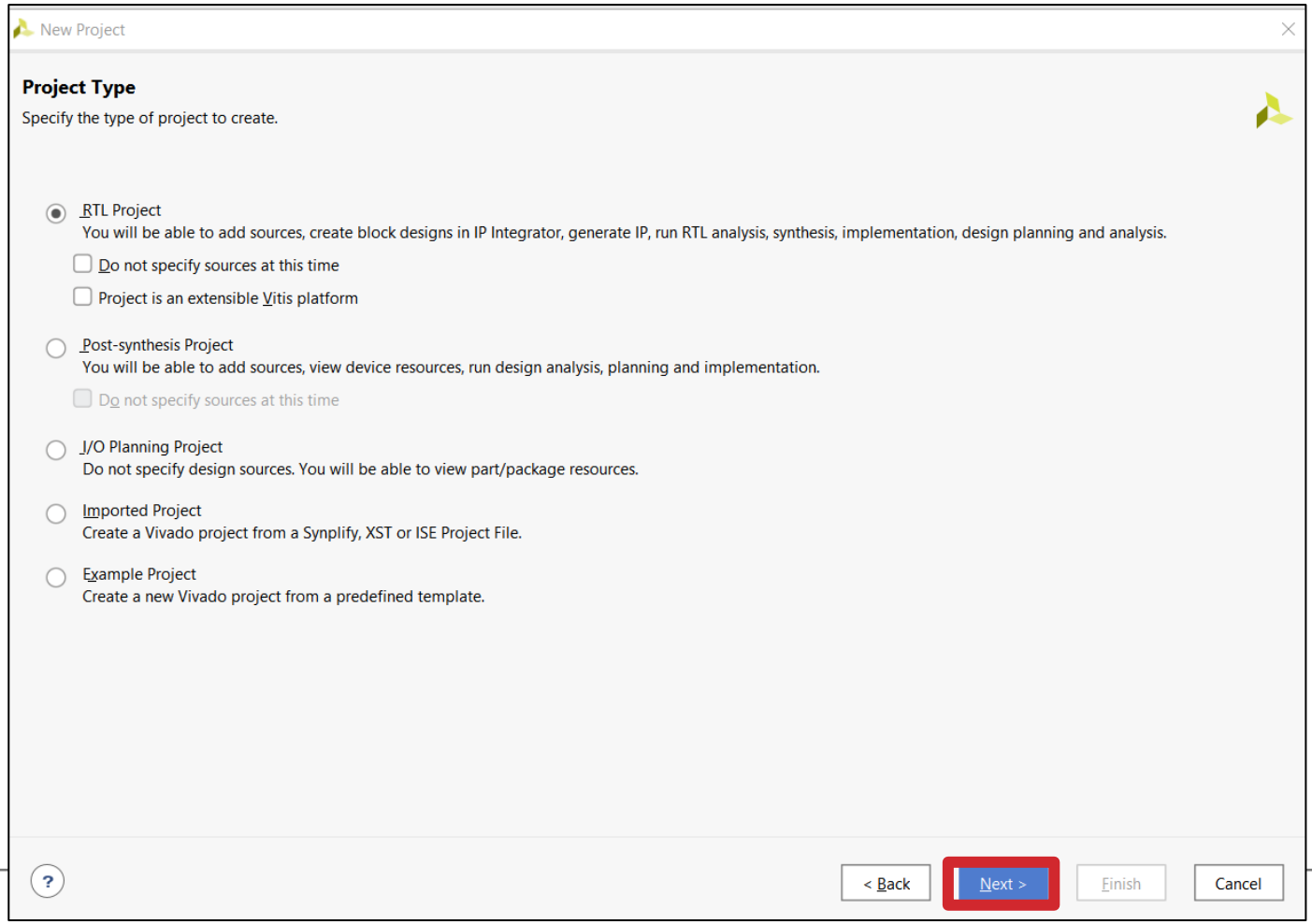


Project name
→ must start with an alphabet
→ can include "_" and numbers

Location
→location of the project
→subfolders can be created

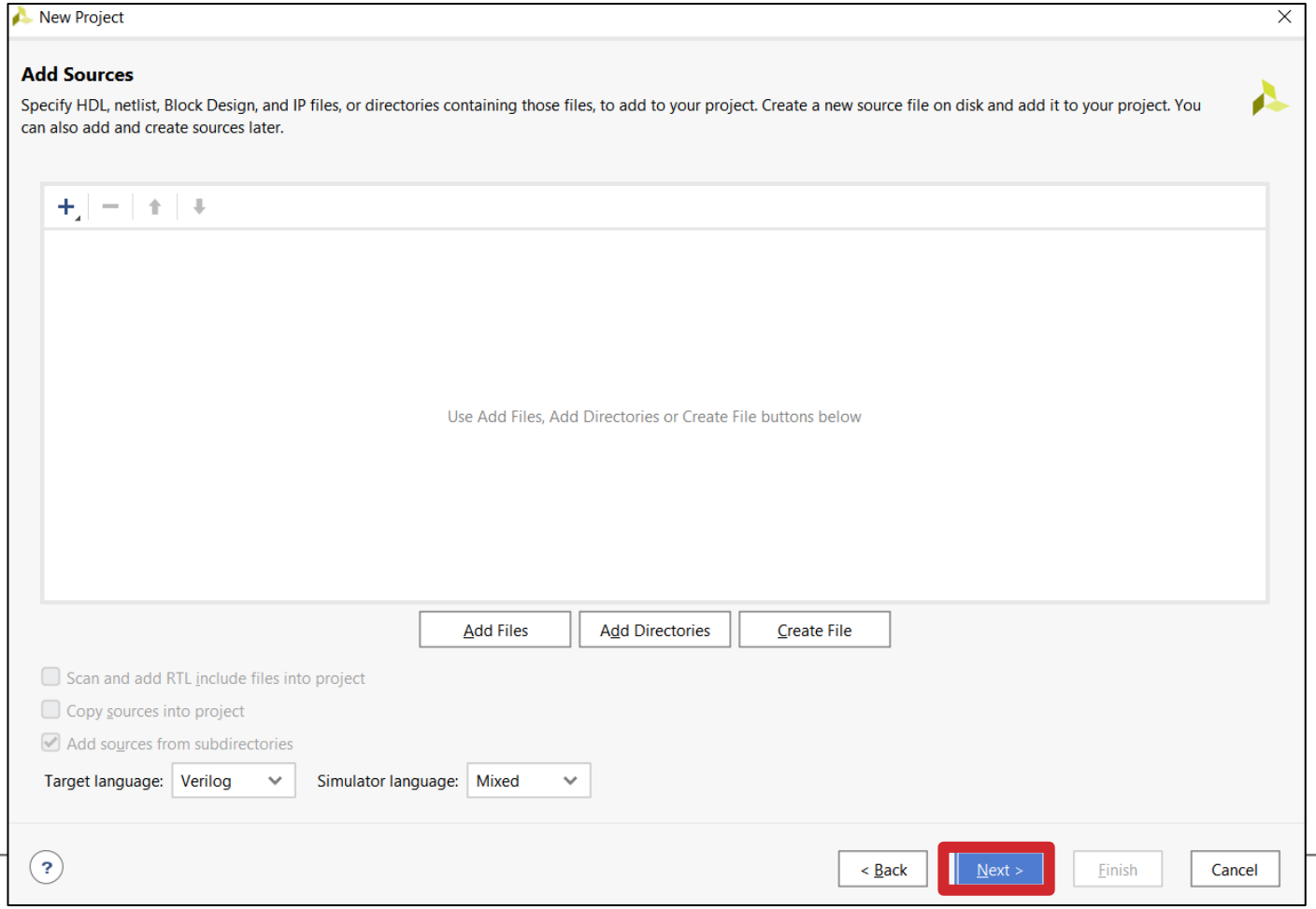
DESIGN EXAMPLE

- Vivado project
  - Creating an new project



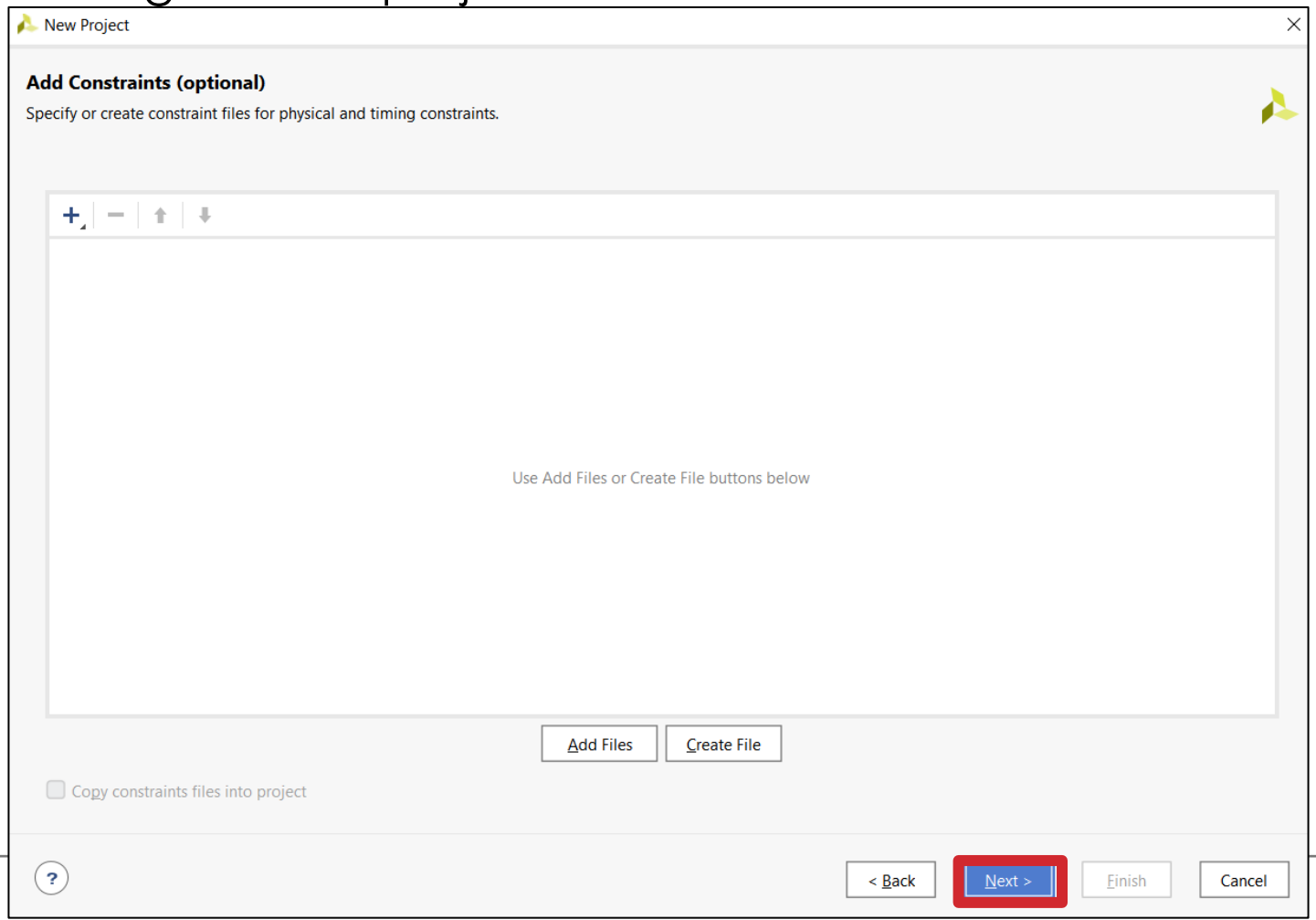
DESIGN EXAMPLE

- Vivado project
  - Creating an new project



DESIGN EXAMPLE

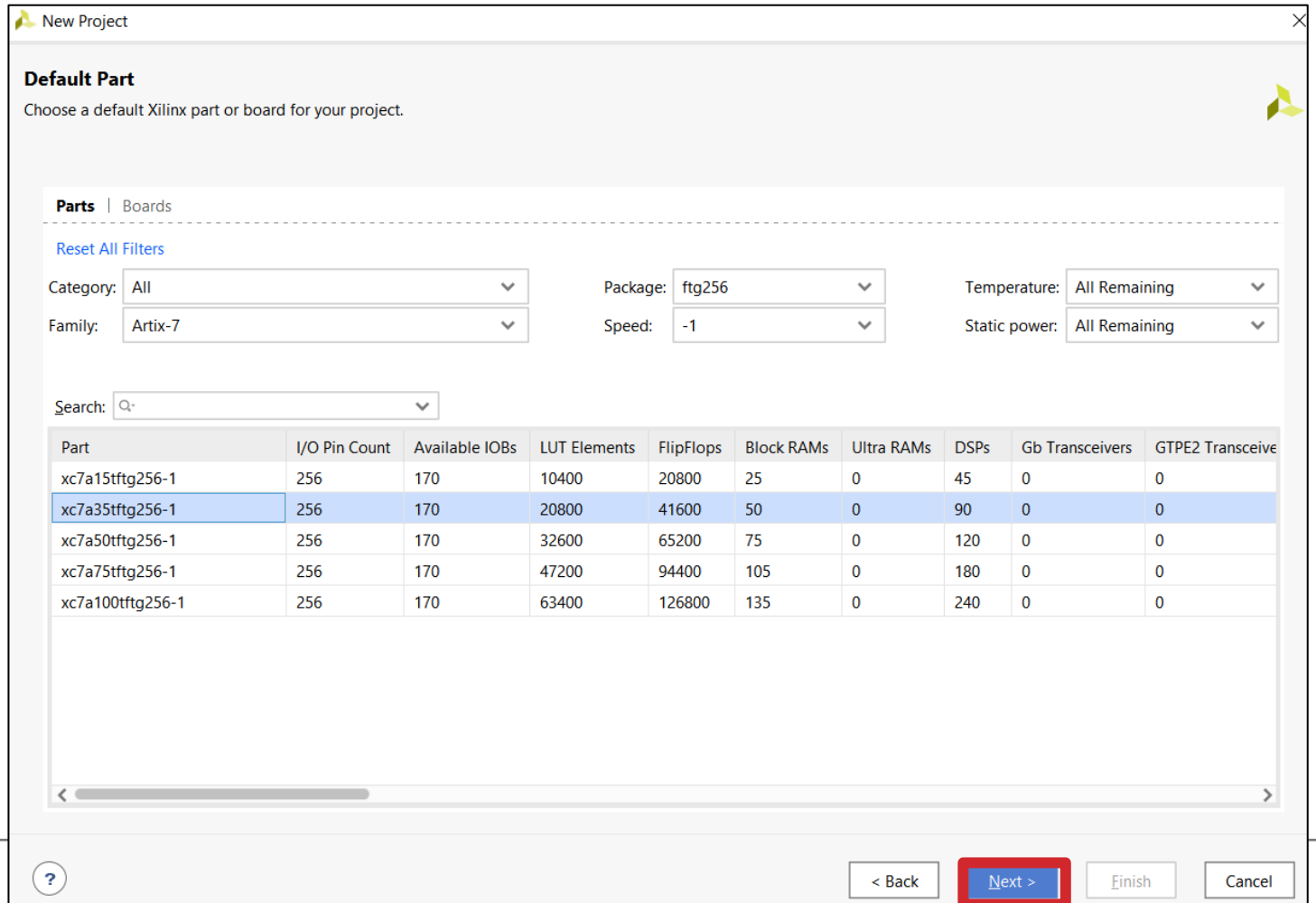
- Vivado project
  - Creating an new project



DESIGN EXAMPLE

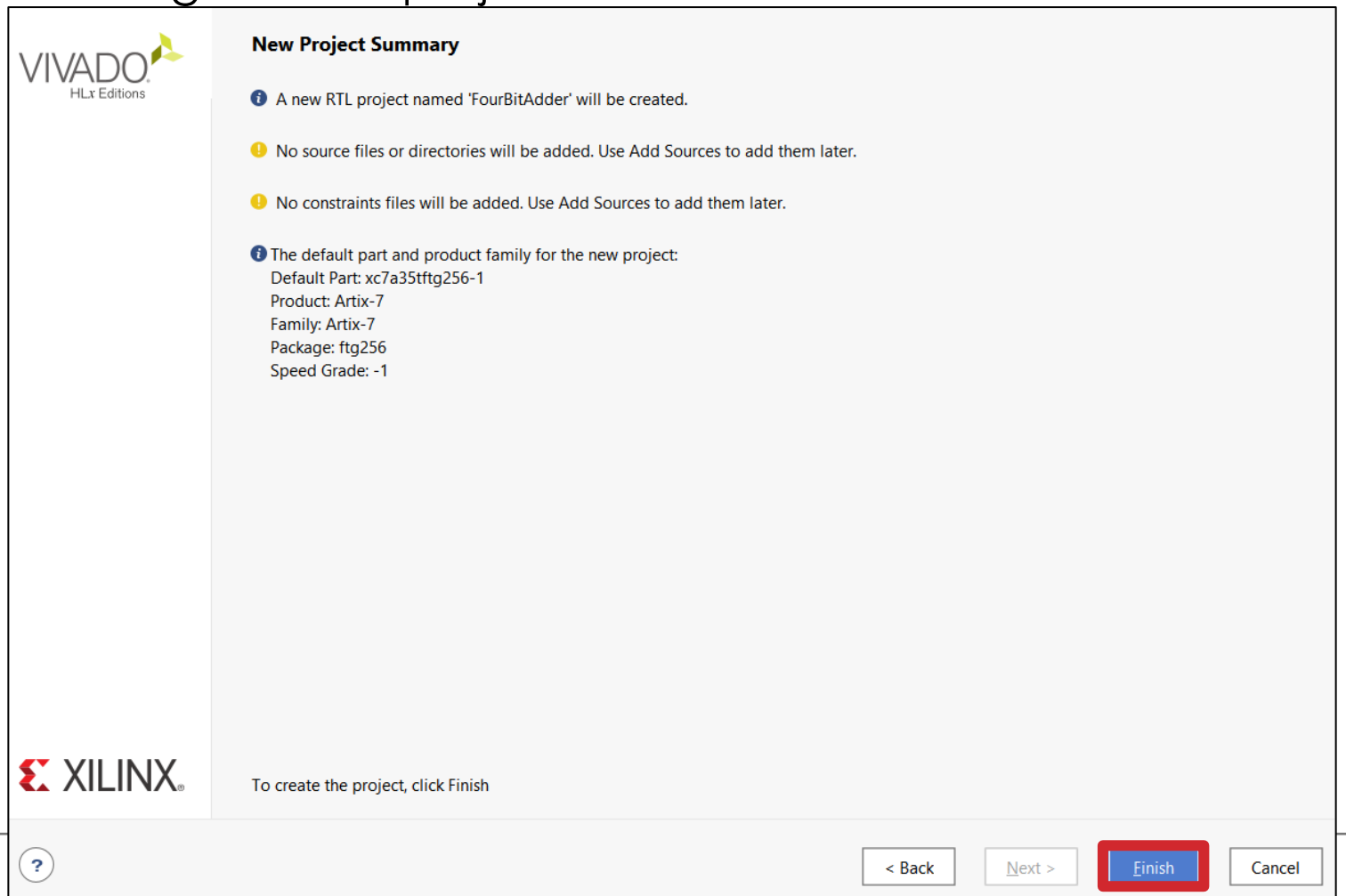
- Vivado project
  - Creating an new project

FPGA Device: Xc7a35tftg256-1



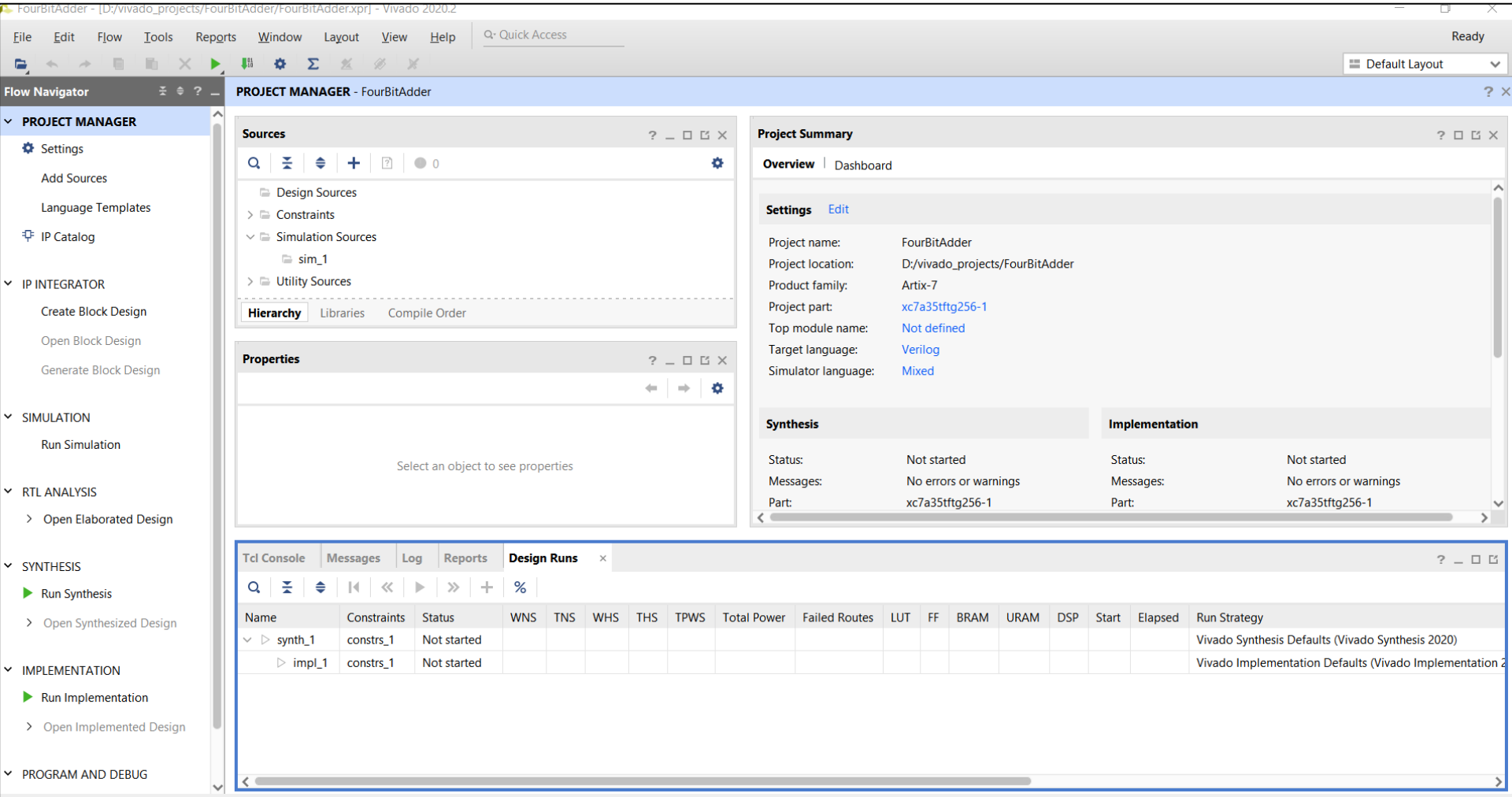
DESIGN EXAMPLE

- Vivado project
  - Creating an new project



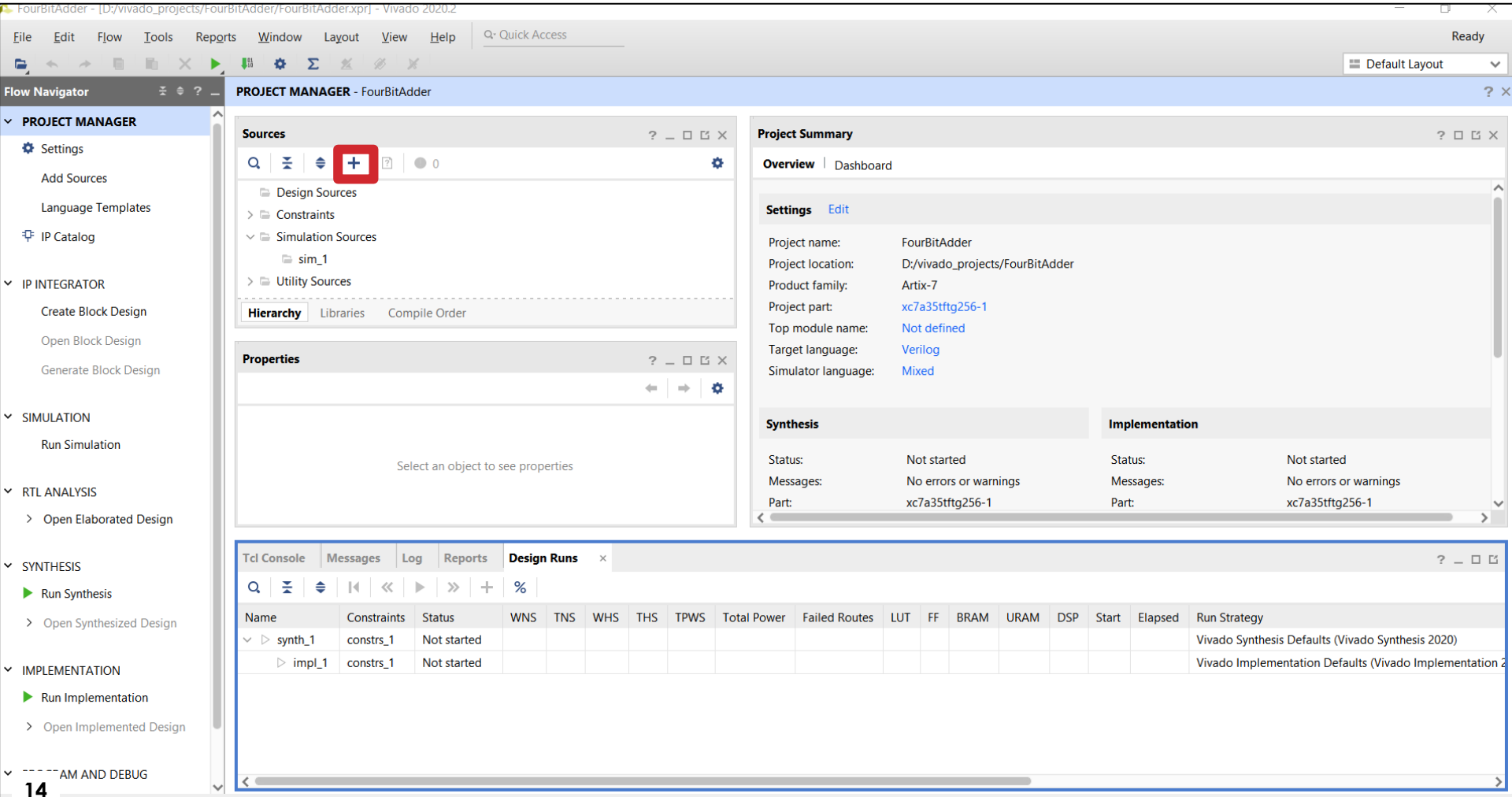
DESIGN EXAMPLE

- Vivado project



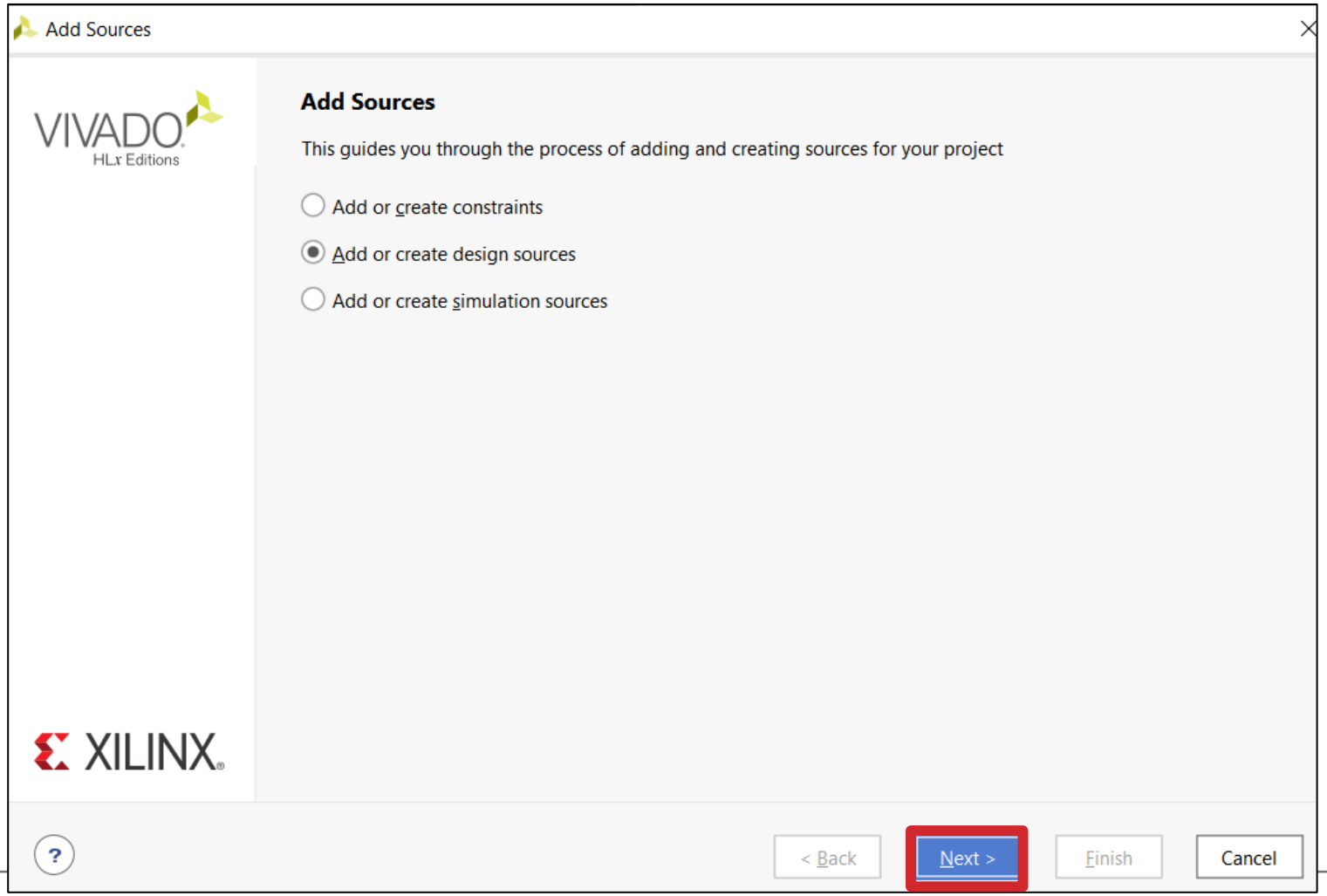
DESIGN EXAMPLE

- Vivado project



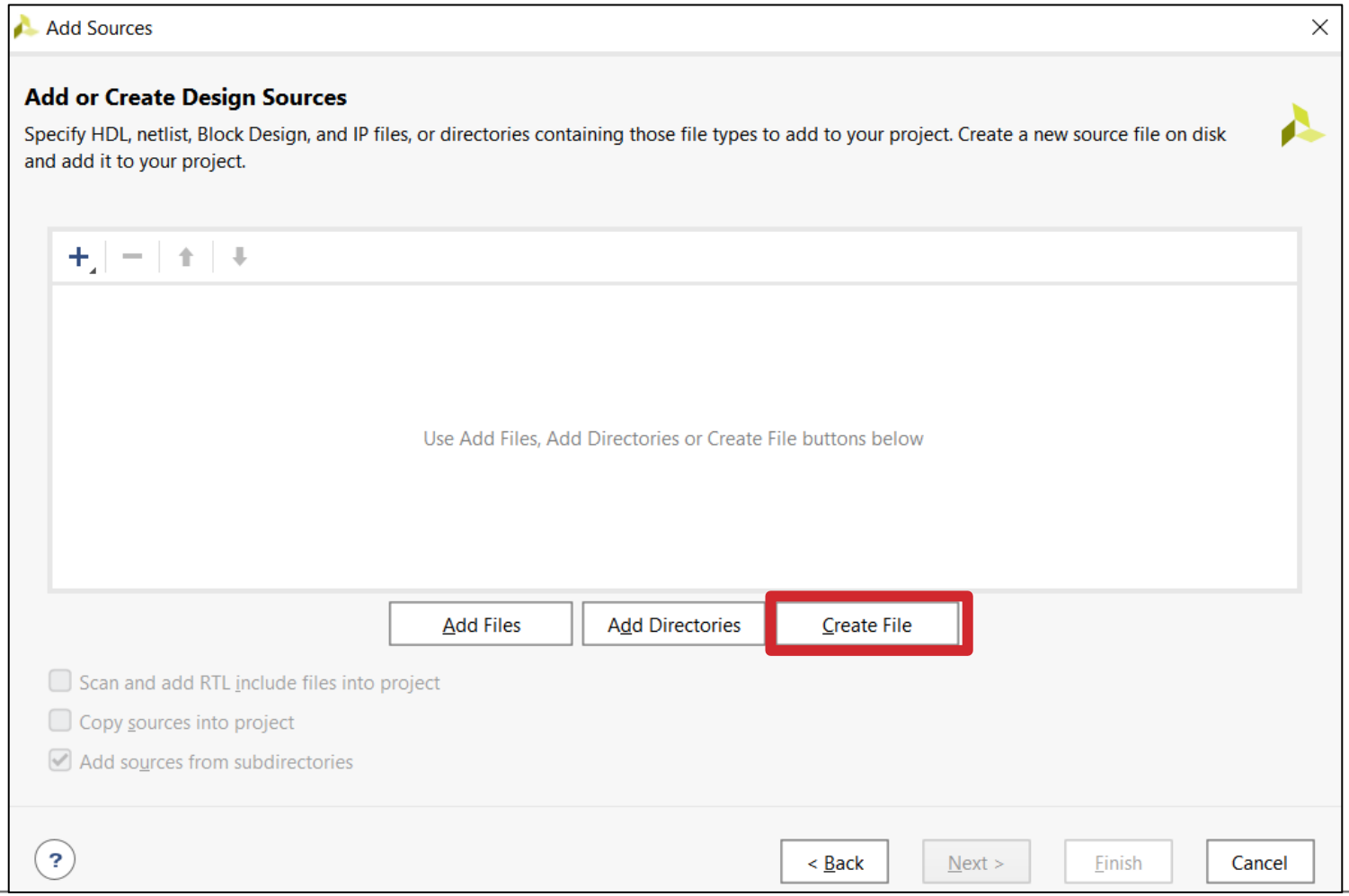
DESIGN EXAMPLE

- Vivado project



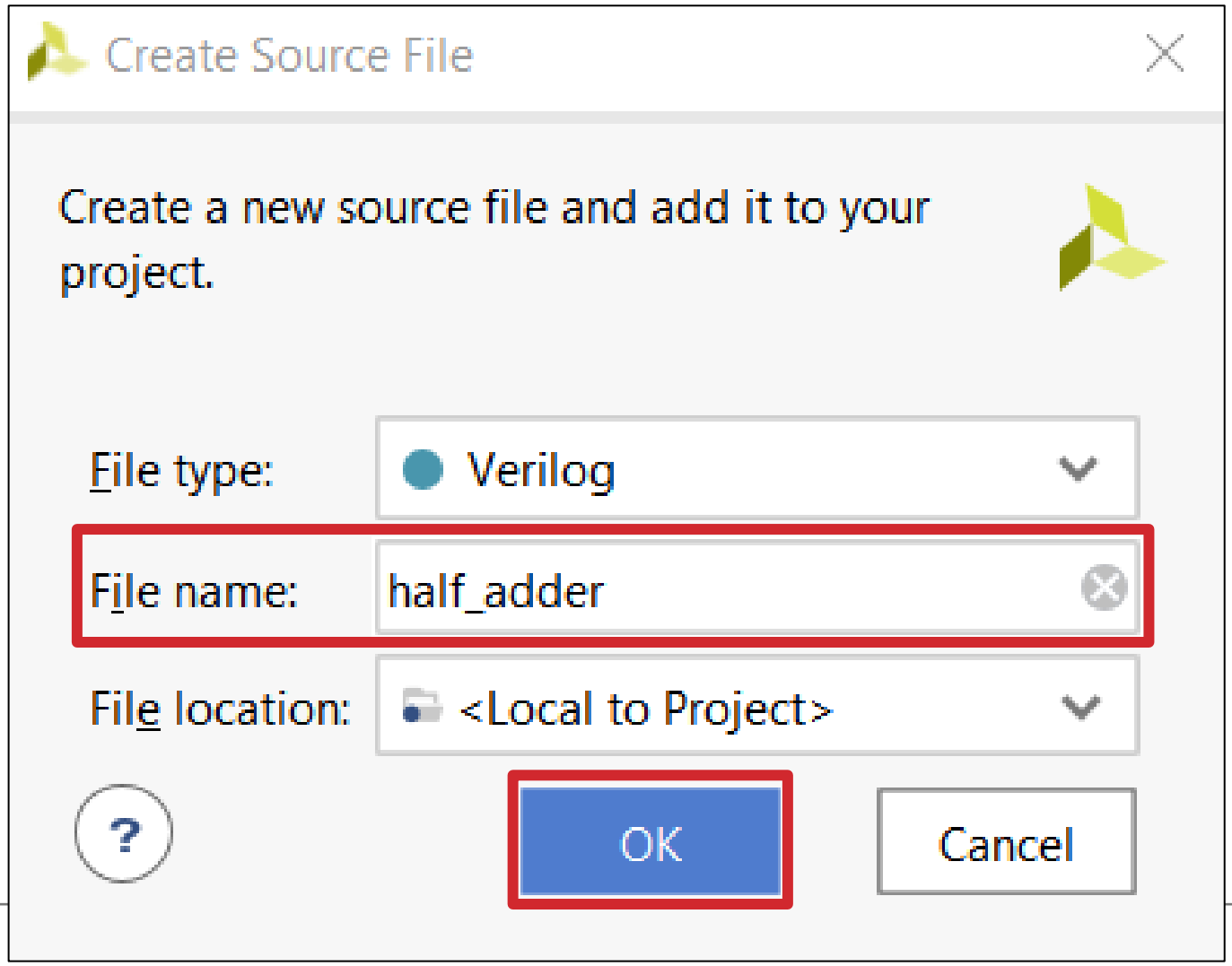
DESIGN EXAMPLE

- Vivado project



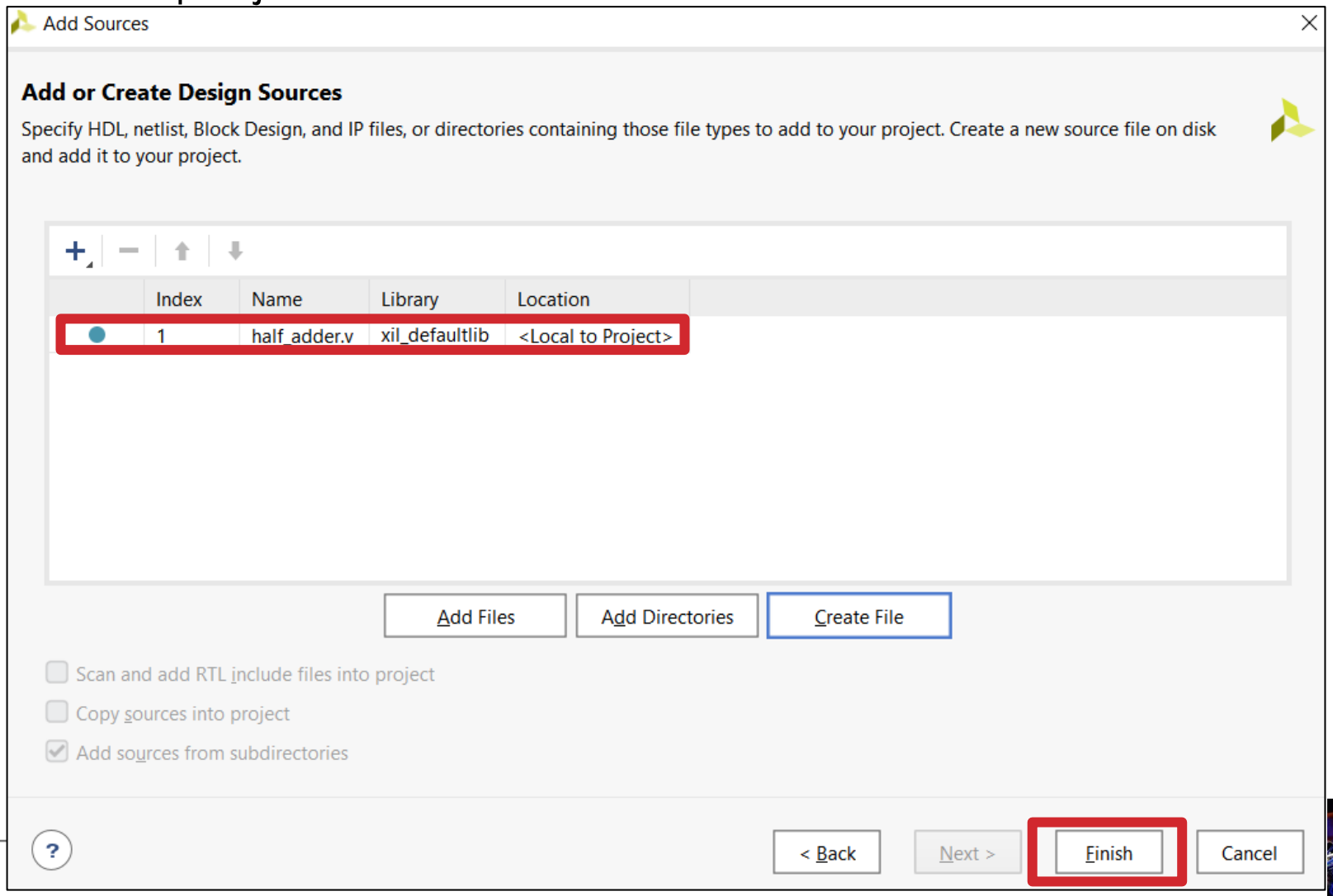
DESIGN EXAMPLE

- Vivado project



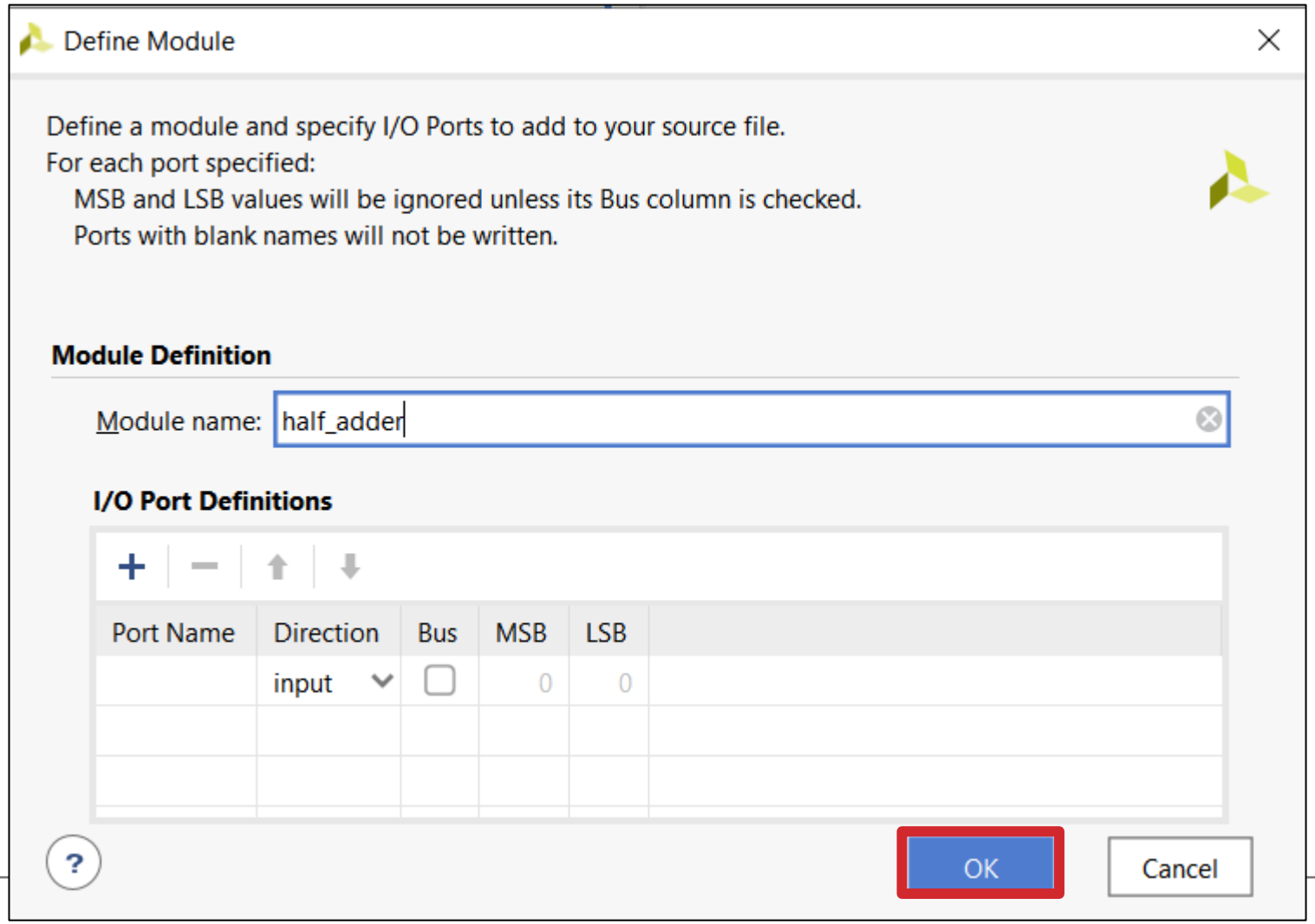
DESIGN EXAMPLE

- Vivado project



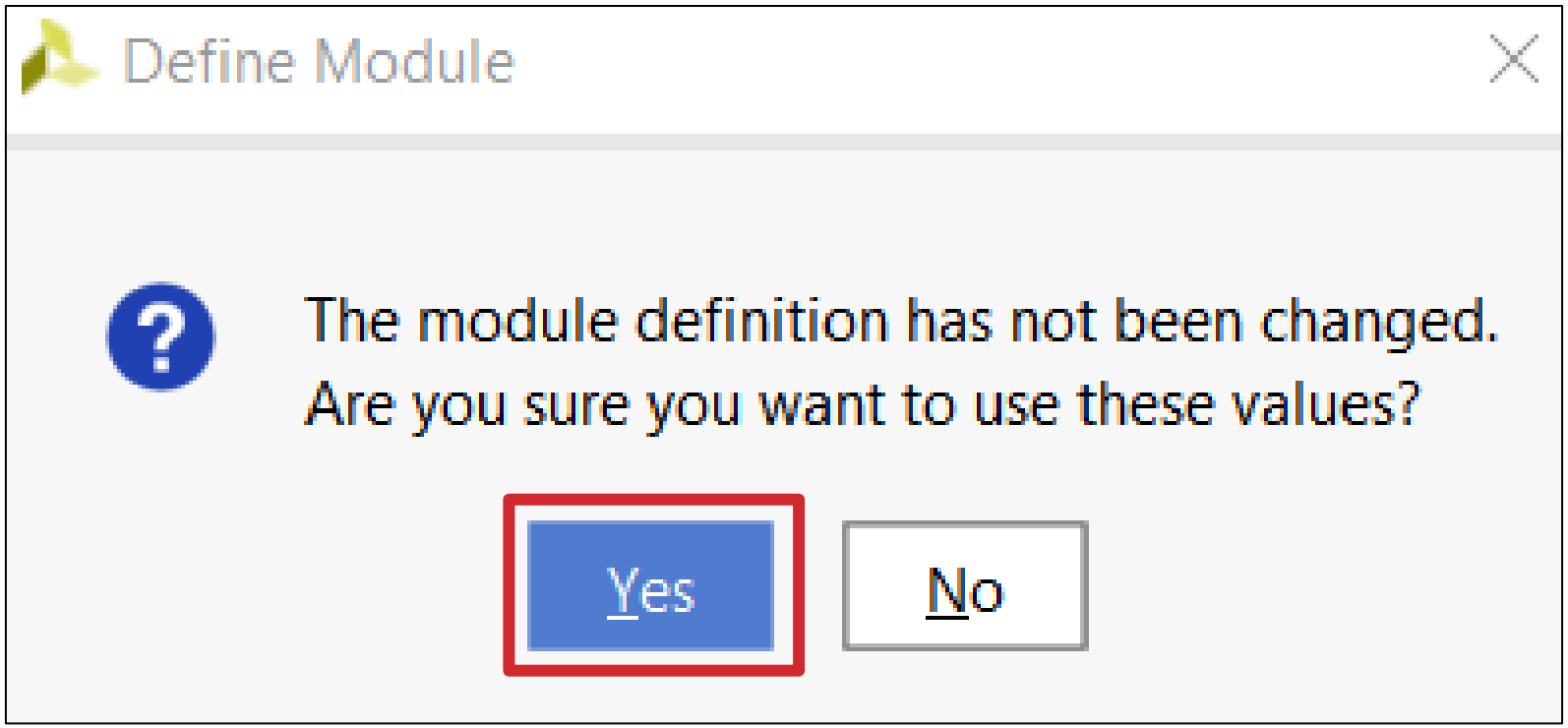
DESIGN EXAMPLE

- Vivado project



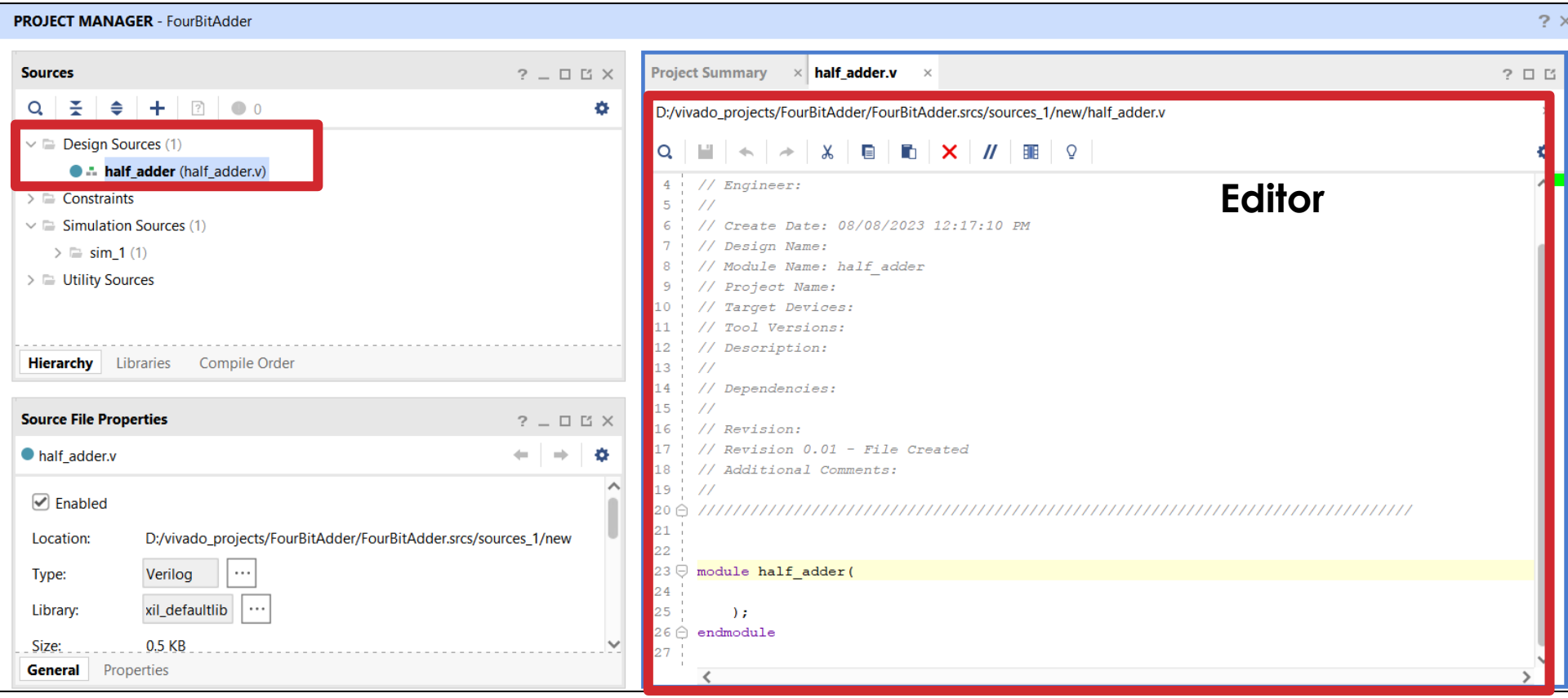
DESIGN EXAMPLE

- Vivado project



DESIGN EXAMPLE

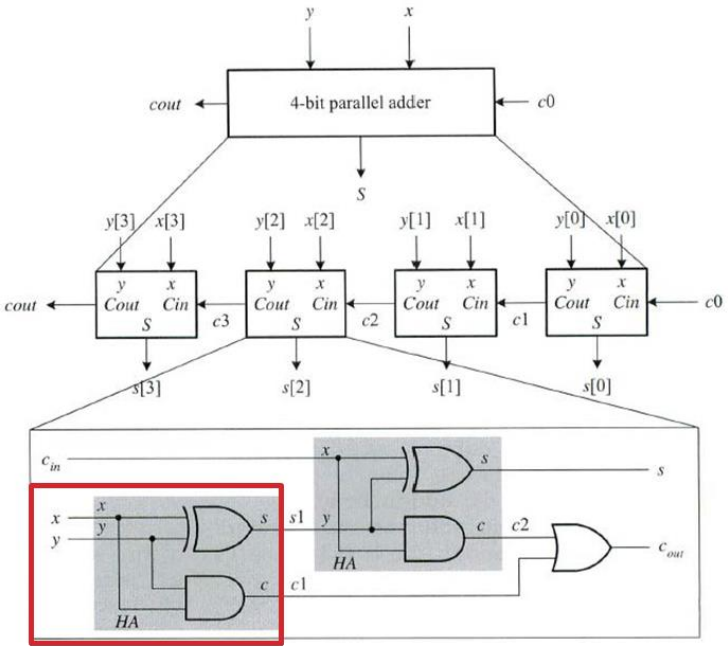
- Vivado project



DESIGN EXAMPLE

- Vivado project
  - Verilog code for Half Adder Module

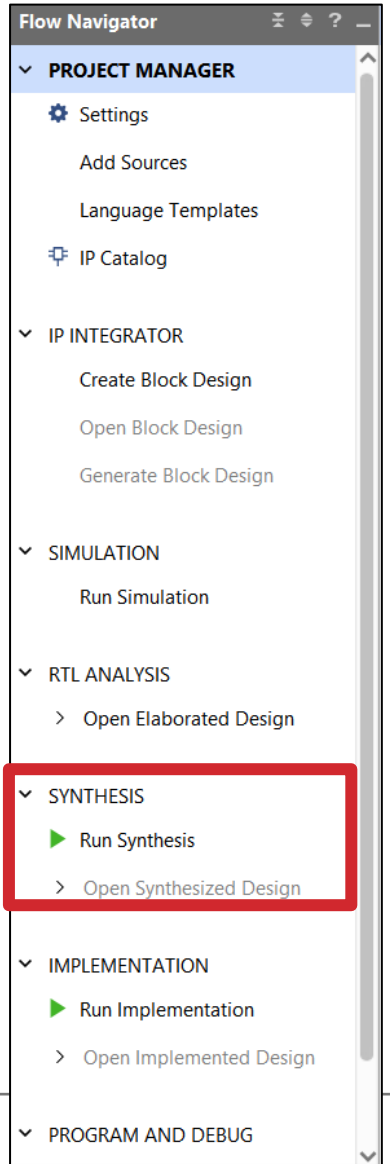
```
23  module half_adder(x, y, s, c);
24
25  //input/output declaration
26  input x, y;
27  output s, c;
28
29  //primitive gate instantiation
30  xor(s, x, y);
31  and(c, x, y);
32
33  endmodule
```

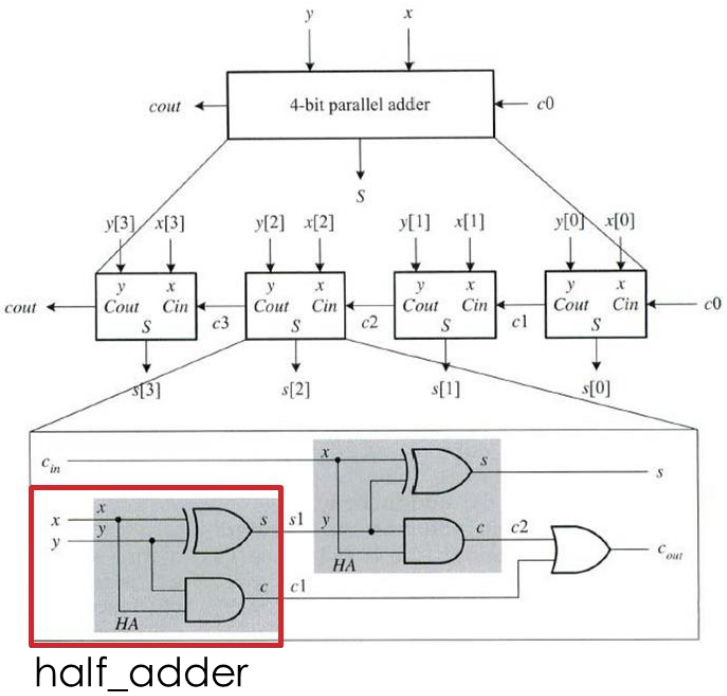


half_adder

DESIGN EXAMPLE

- Vivado project
  - Logic synthesis

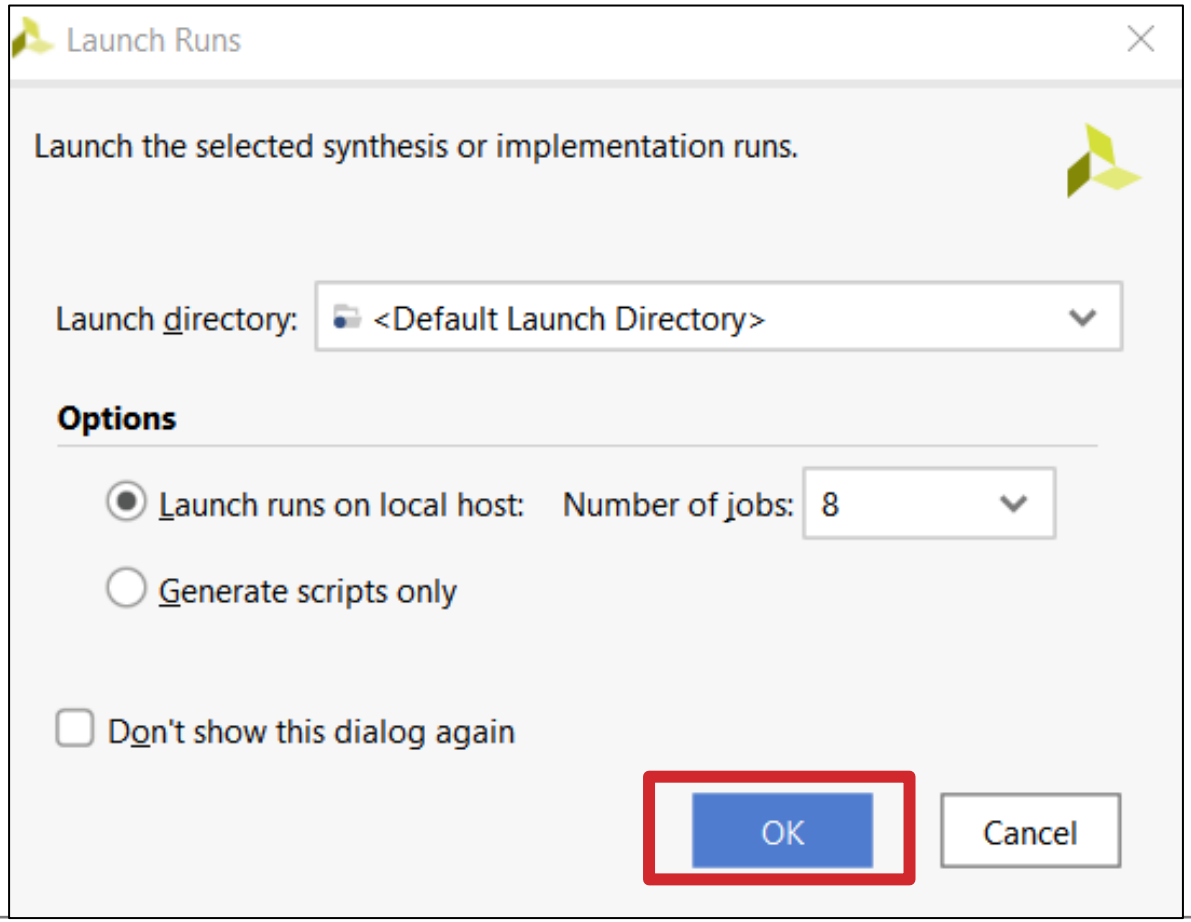




half_adder

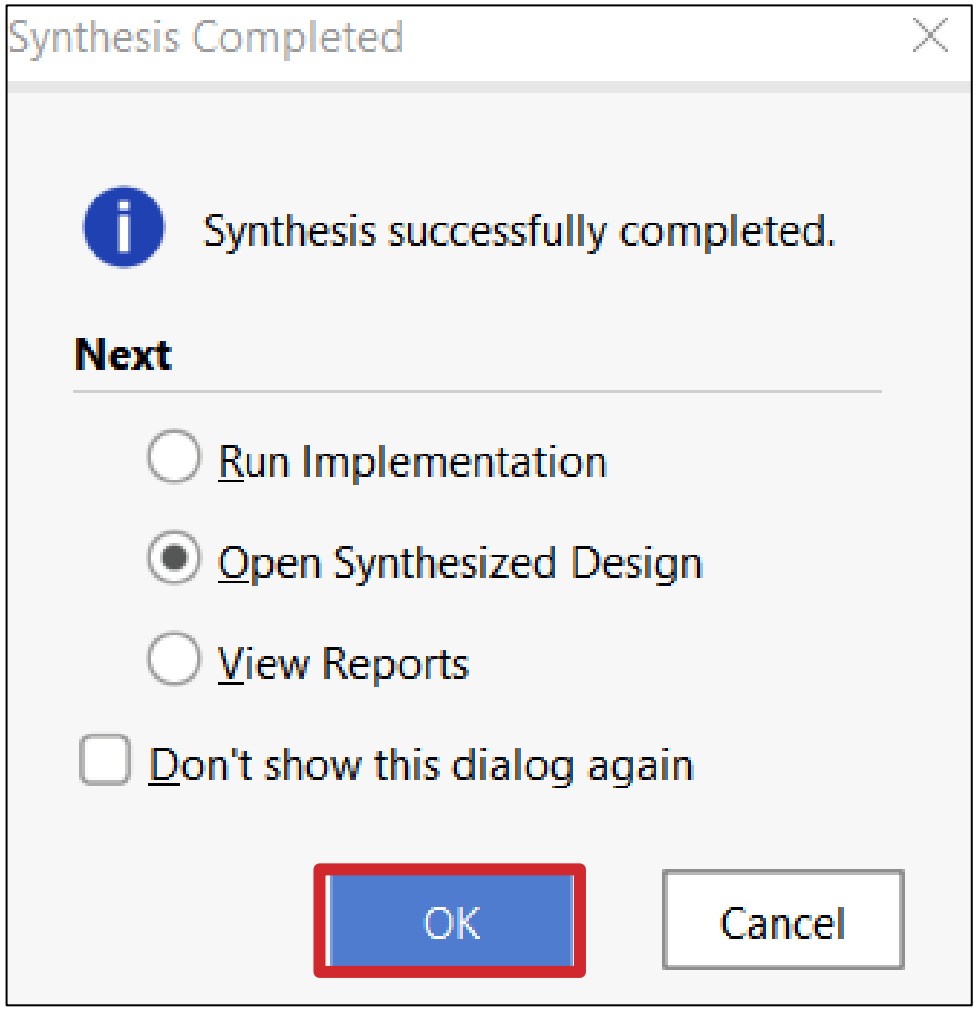
DESIGN EXAMPLE

- Vivado project
  - Logic synthesis



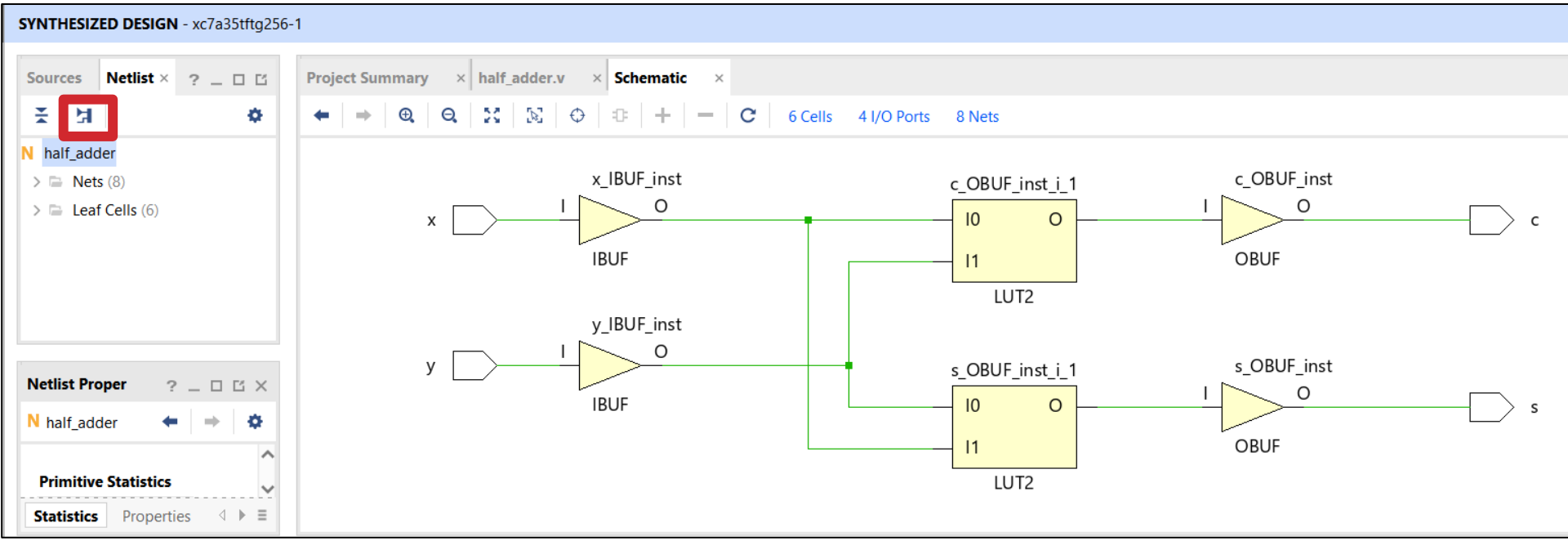
DESIGN EXAMPLE

- Vivado project
  - Logic synthesis



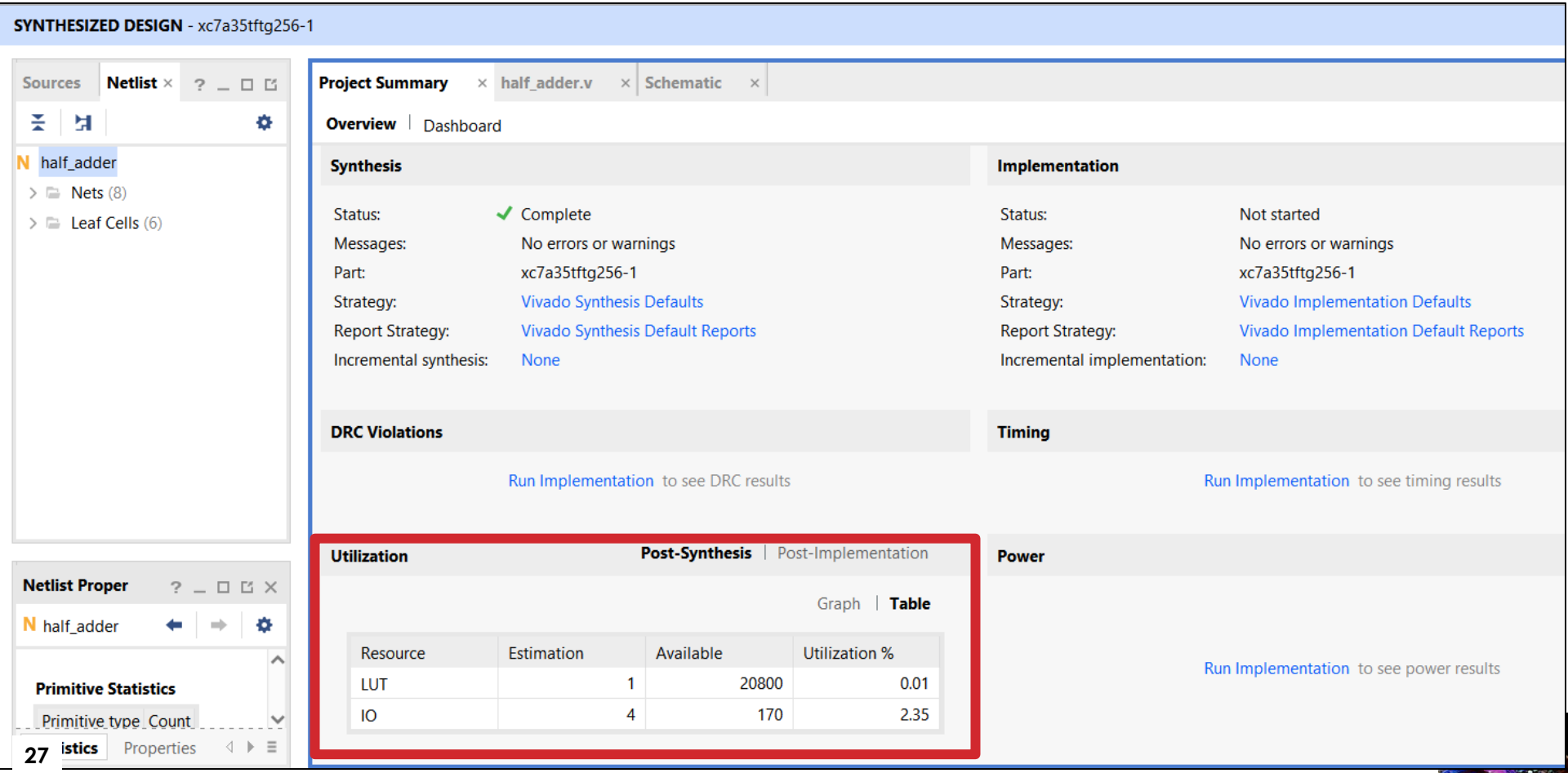
DESIGN EXAMPLE

- Vivado project
  - Logic synthesis (Schematic)



DESIGN EXAMPLE

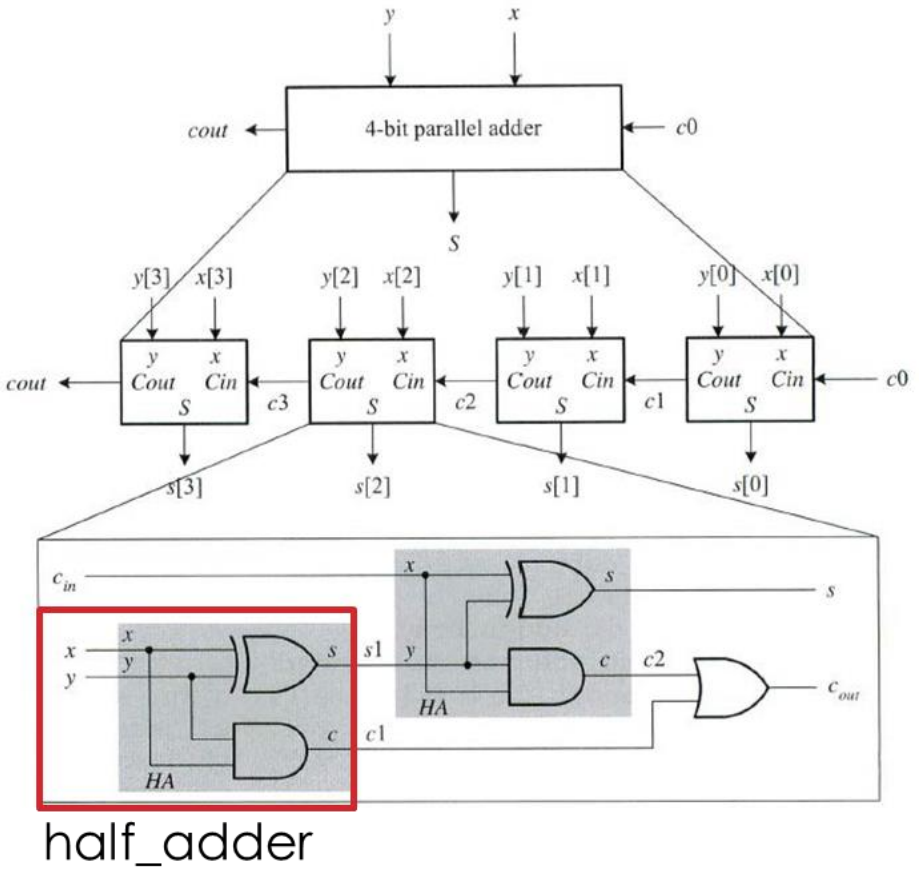
- Vivado project
  - Logic synthesis (Results)



DESIGN EXAMPLE

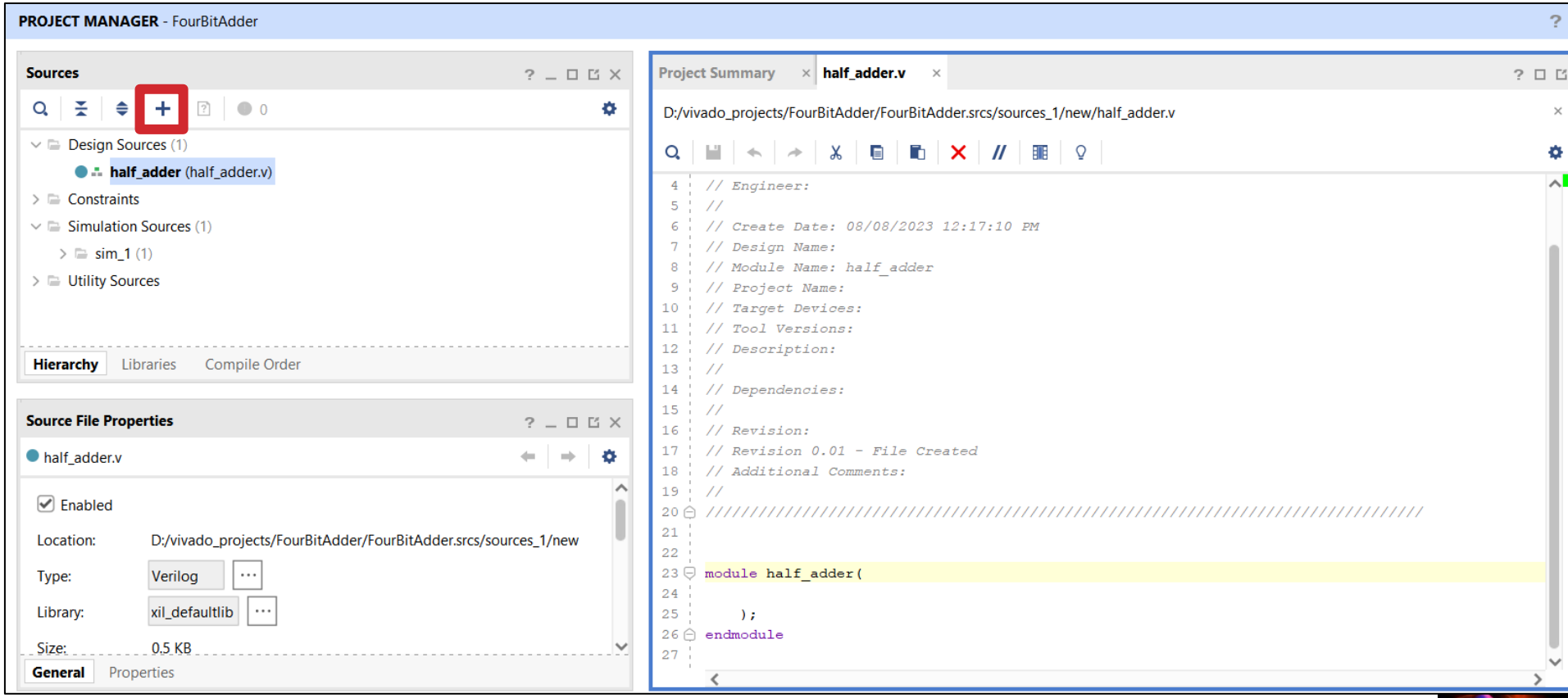
- Vivado project
  - Half Adder Testbench (Test Vectors)

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



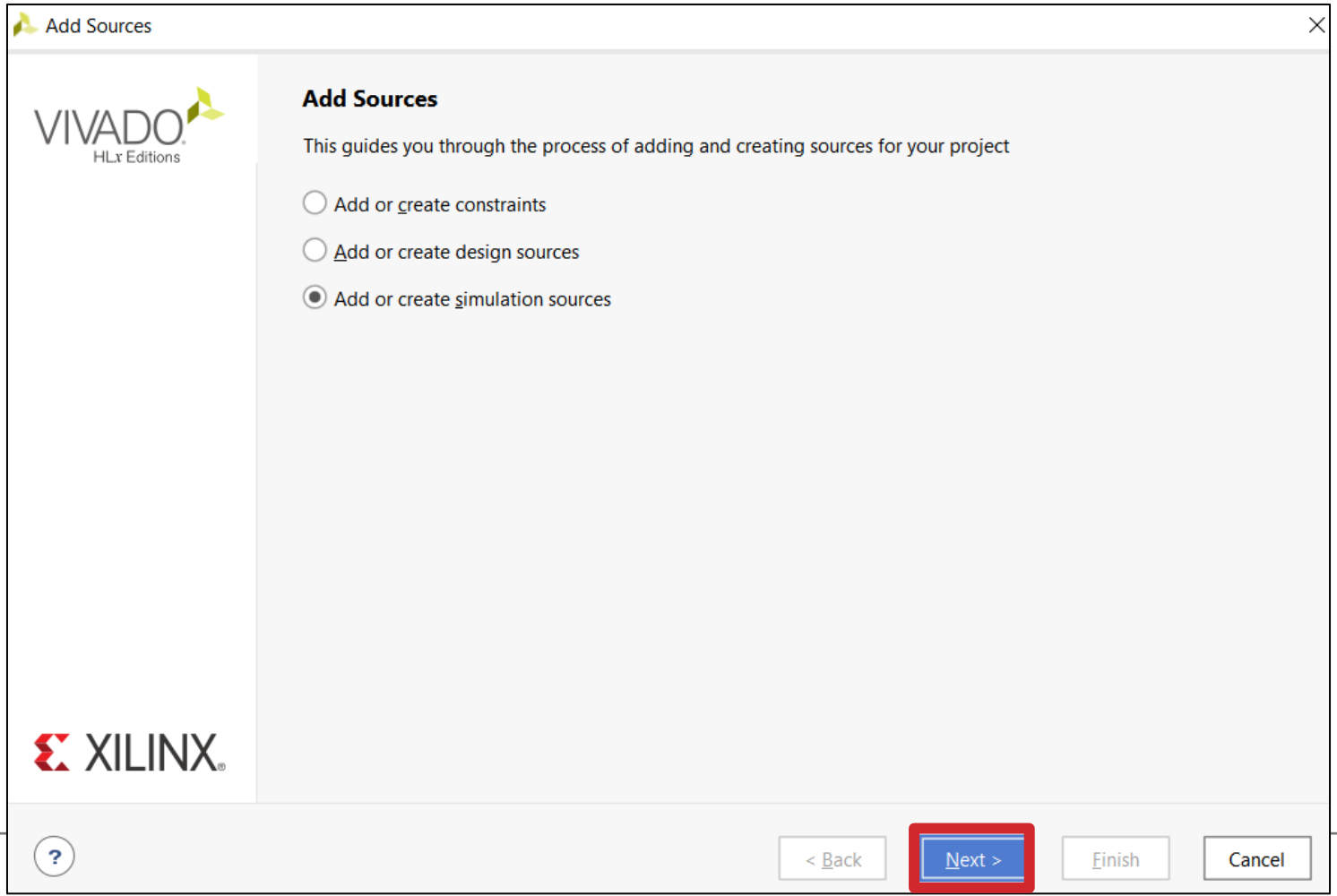
DESIGN EXAMPLE

- Vivado project
  - Creating a Testbench



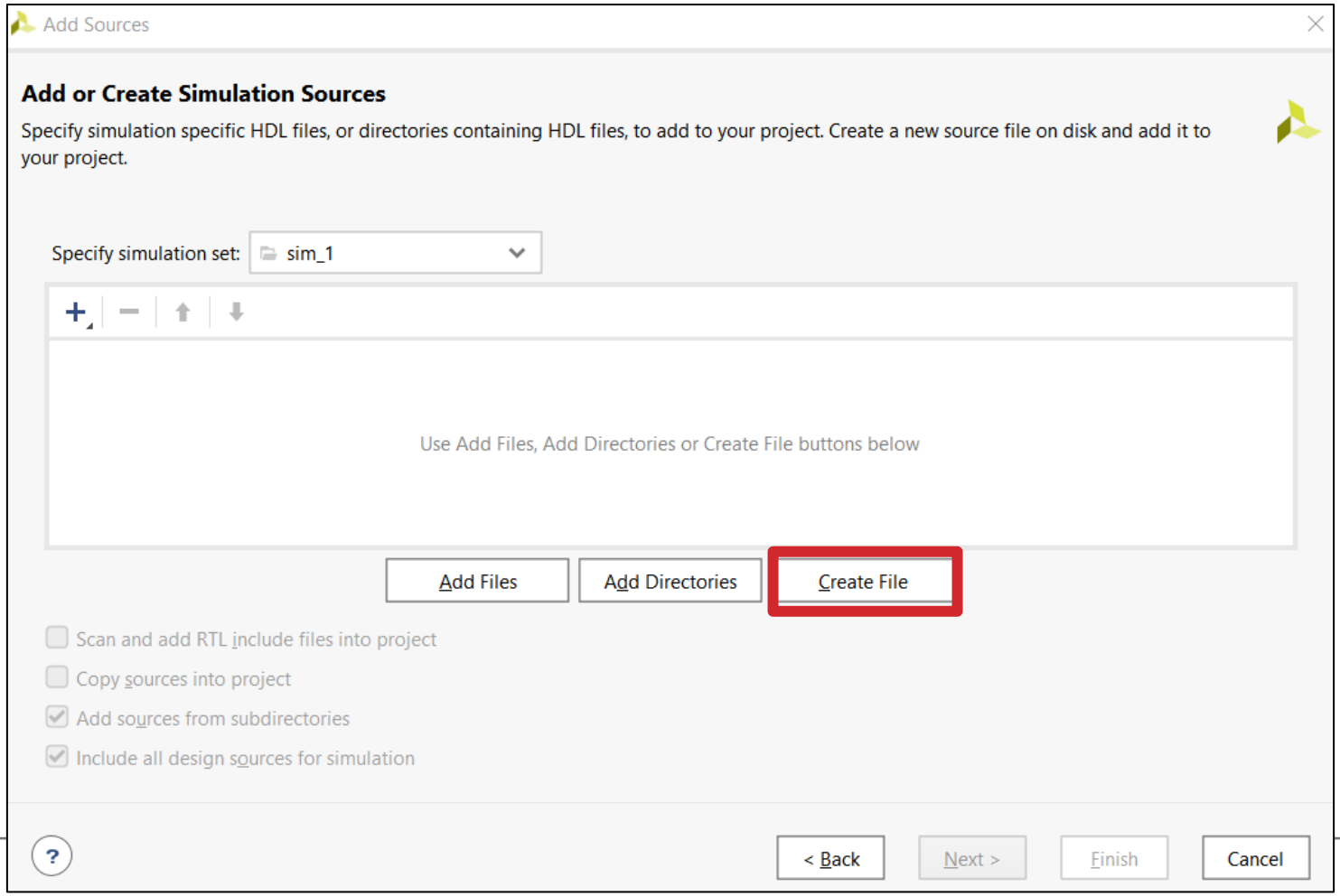
DESIGN EXAMPLE

- Vivado project
  - Creating a Testbench



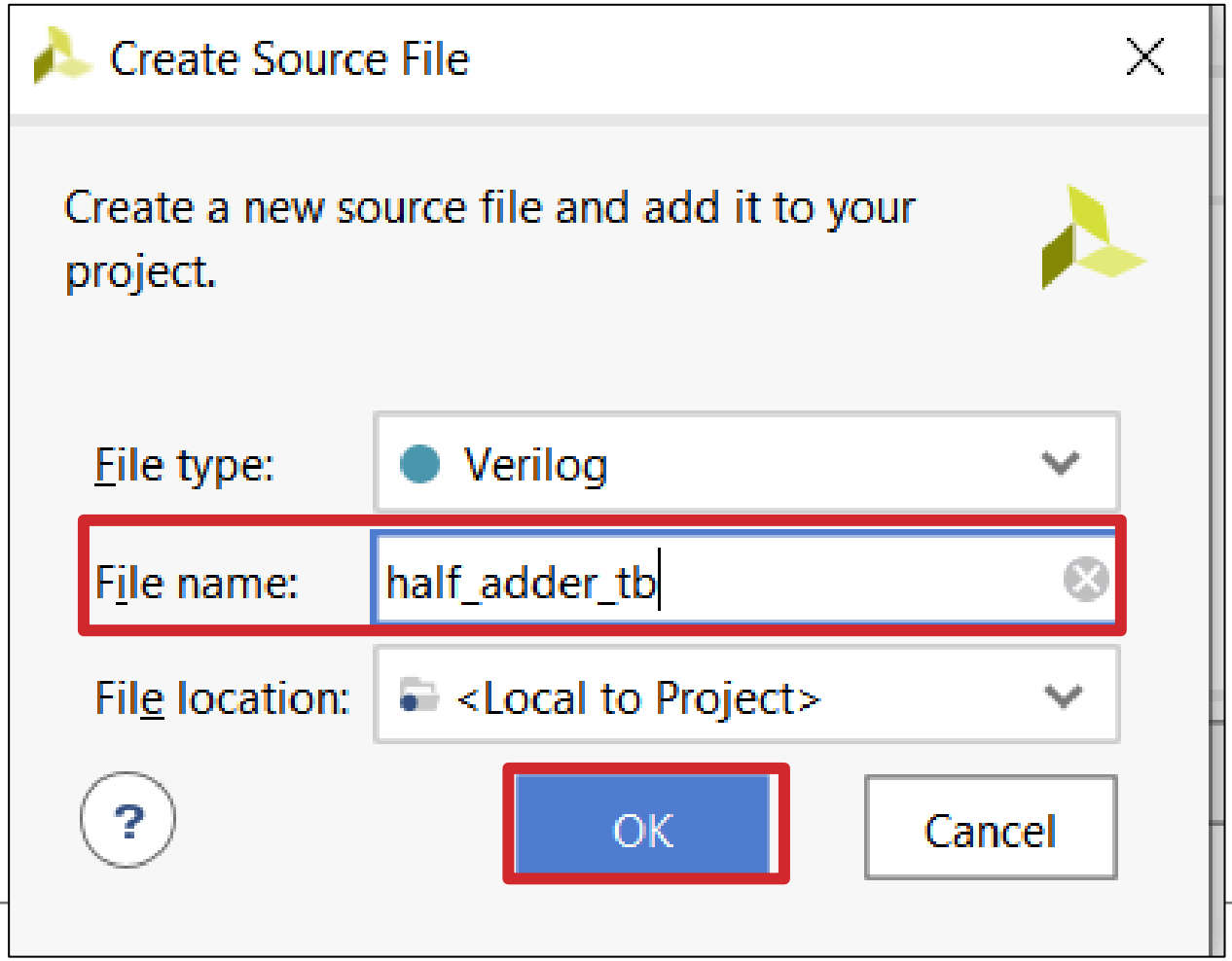
DESIGN EXAMPLE

- Vivado project
  - Creating a Testbench



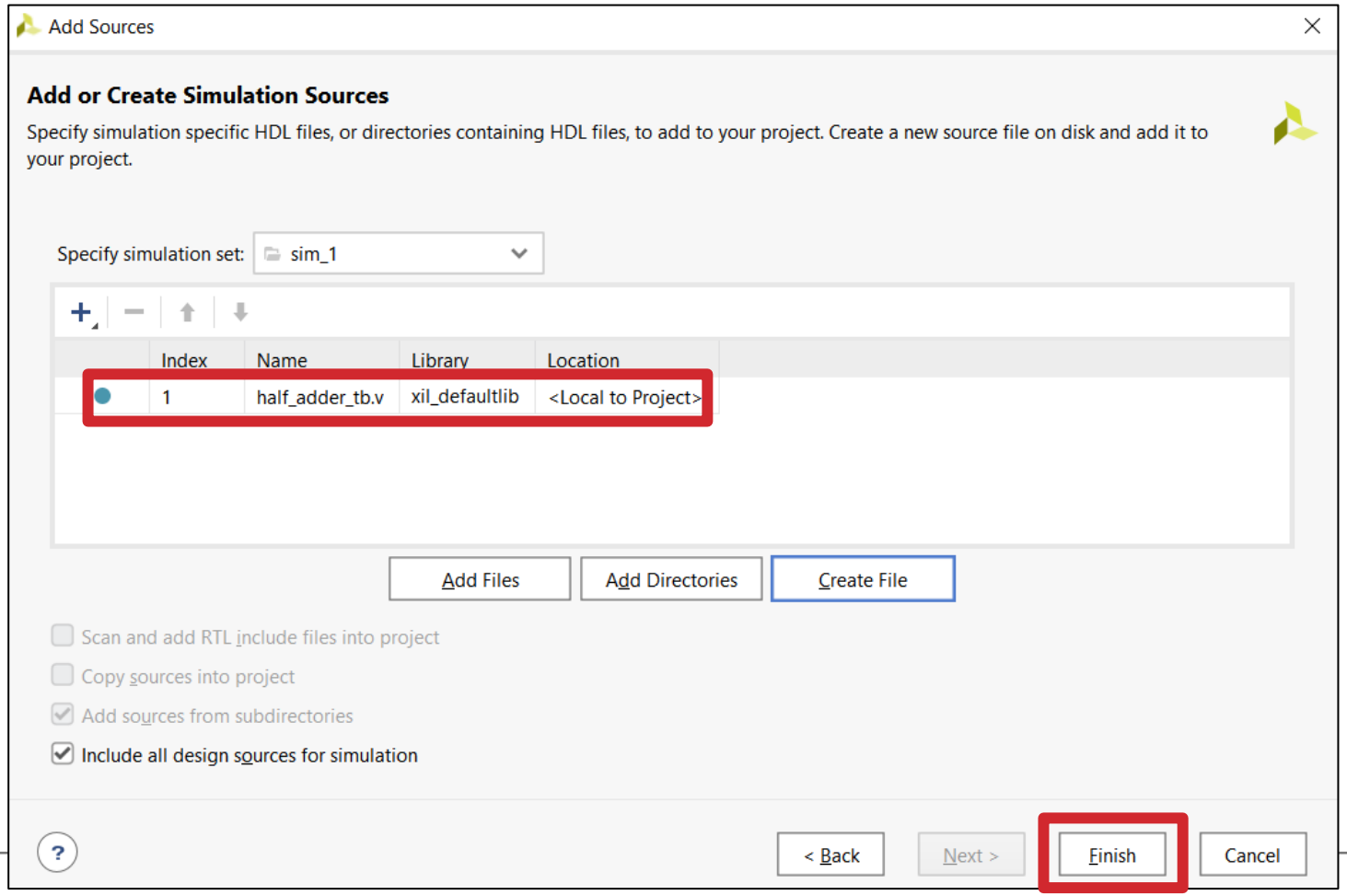
DESIGN EXAMPLE

- Vivado project
  - Creating a Testbench



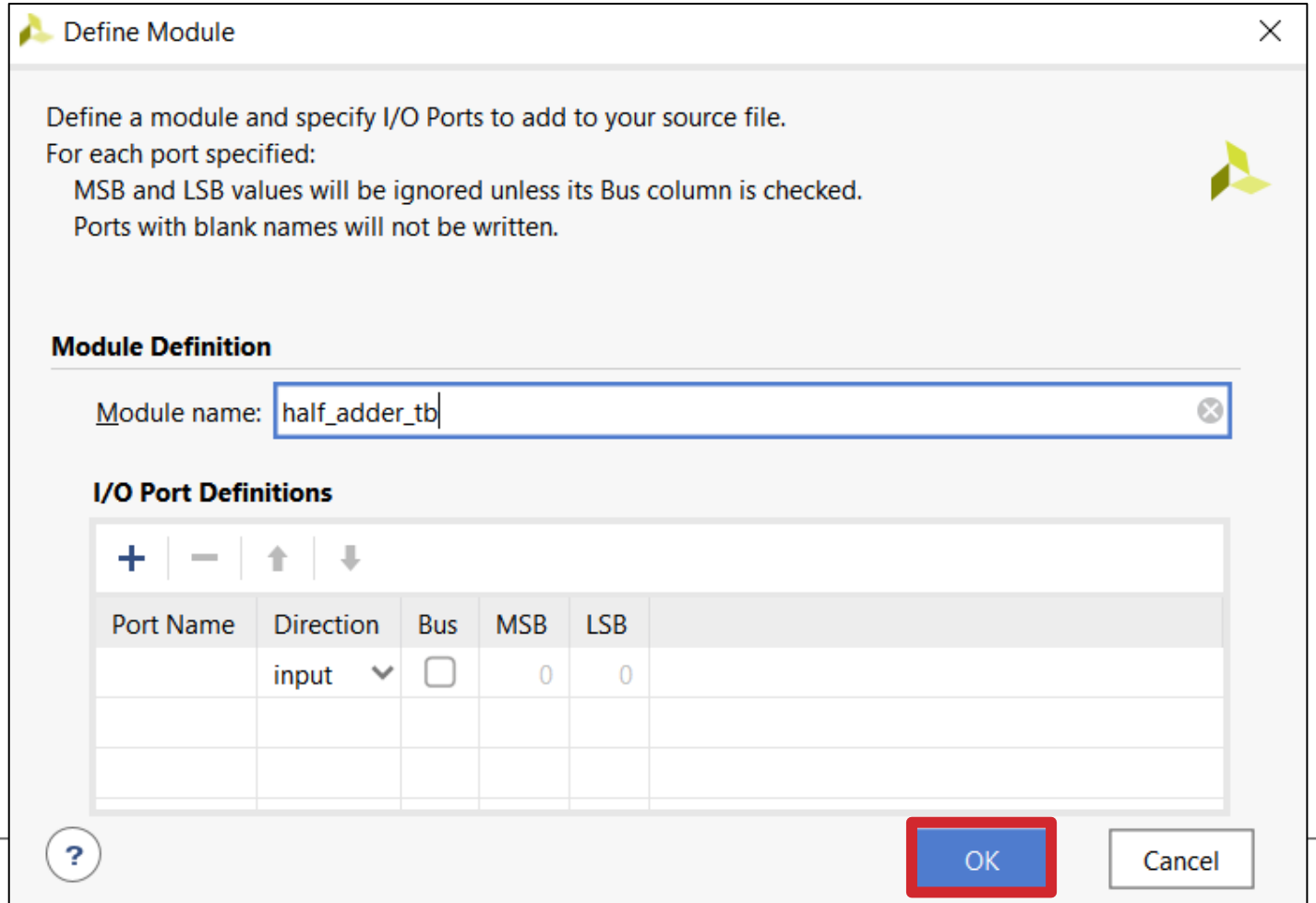
DESIGN EXAMPLE

- Vivado project
  - Creating a Testbench



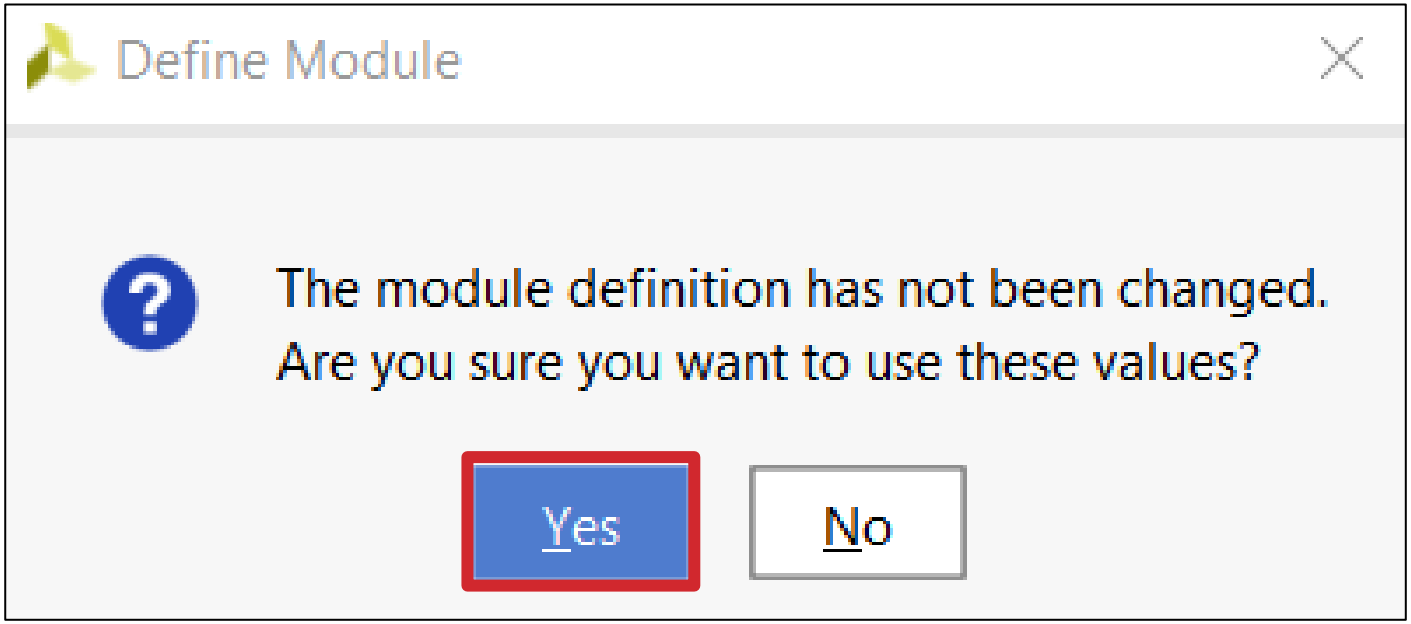
DESIGN EXAMPLE

- Vivado project
  - Creating a Testbench



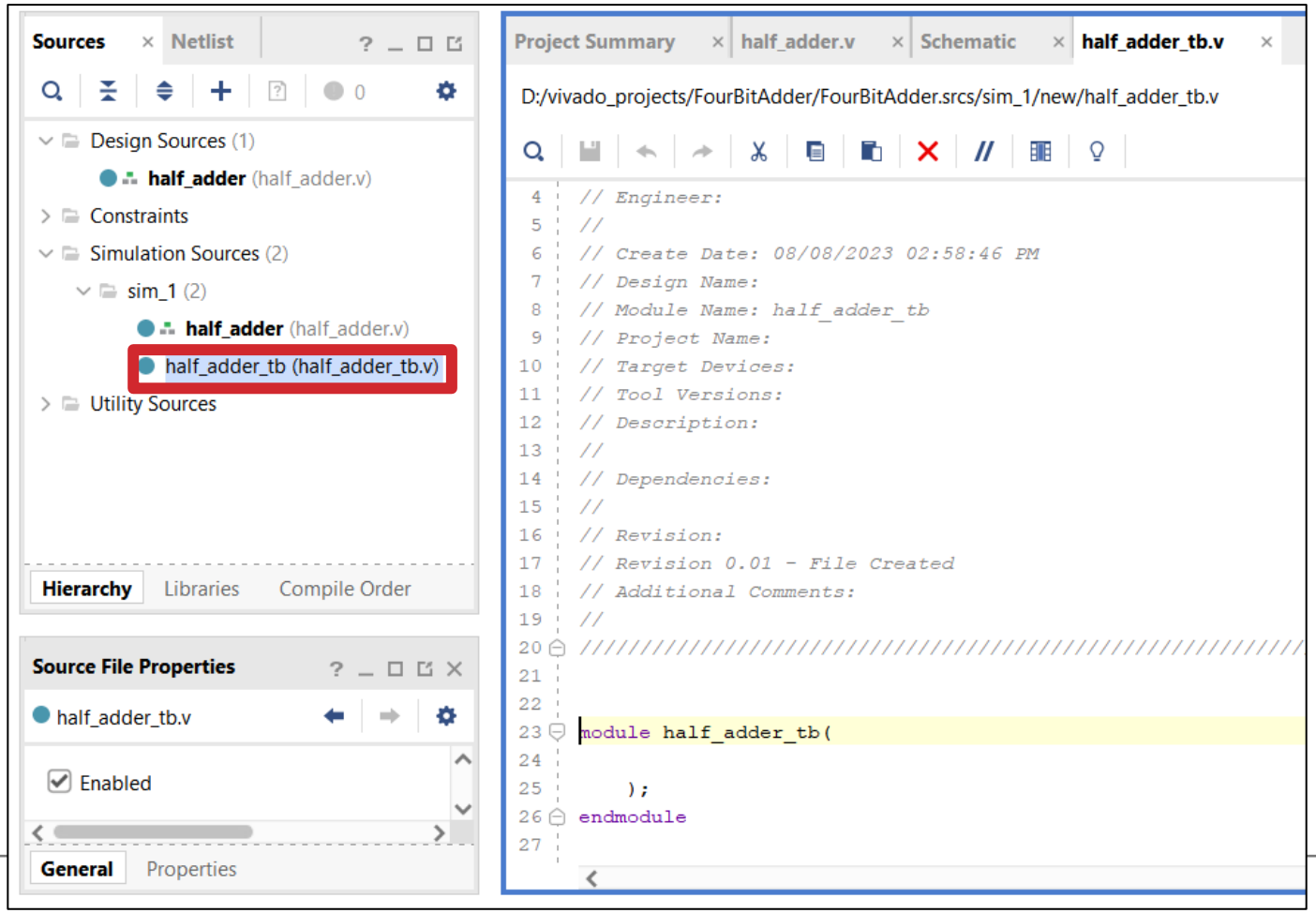
DESIGN EXAMPLE

- Vivado project
  - Creating a Testbench



DESIGN EXAMPLE

- Vivado project
  - Creating a Testbench



DESIGN EXAMPLE

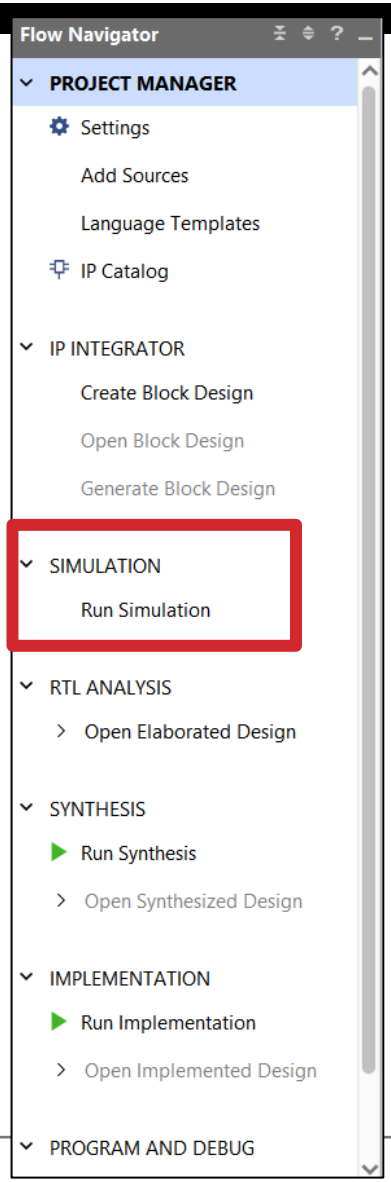
- Vivado project
  - Creating a Testbench

```
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

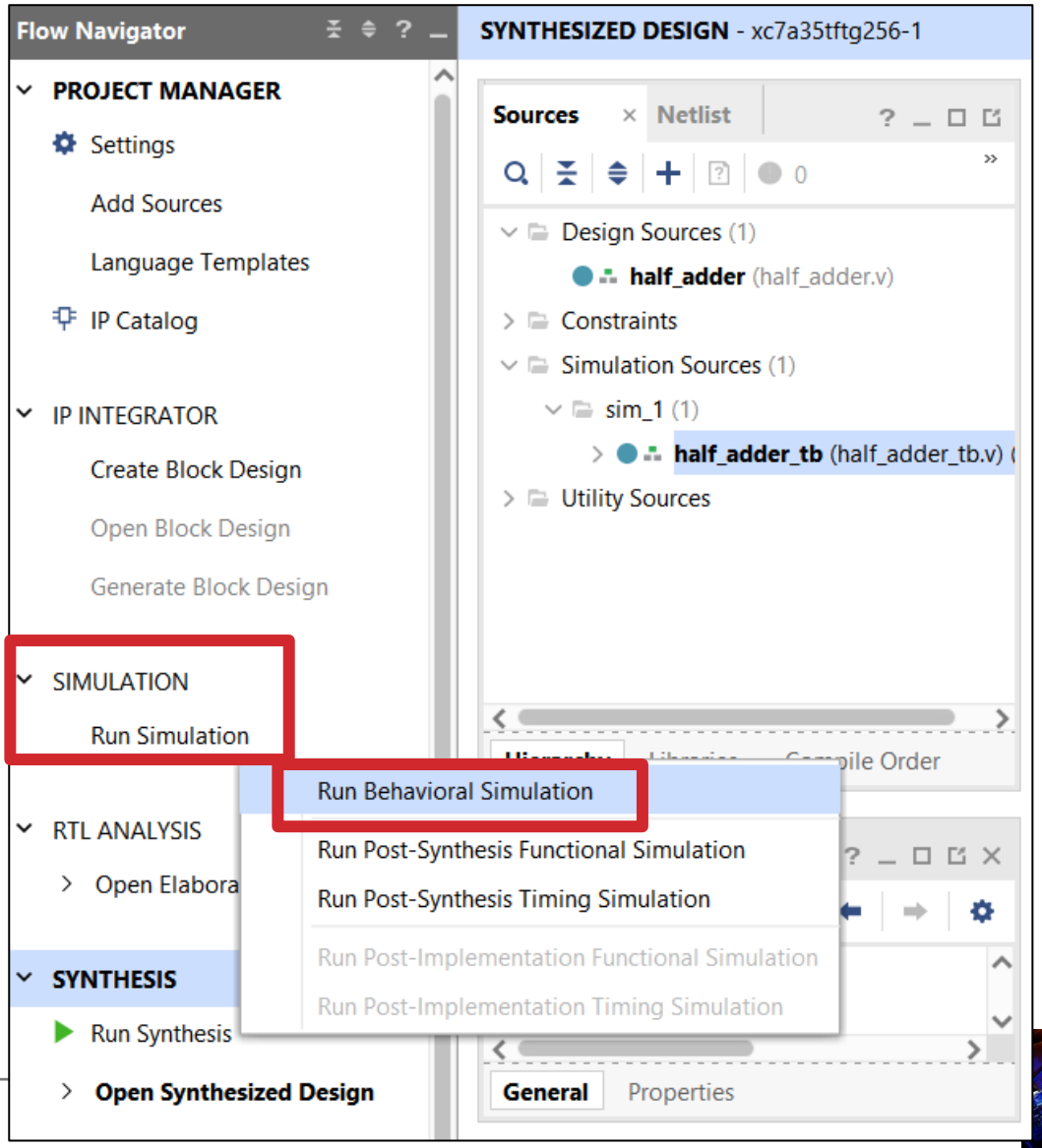
DESIGN EXAMPLE

- Vivado project
  - Running Simulation



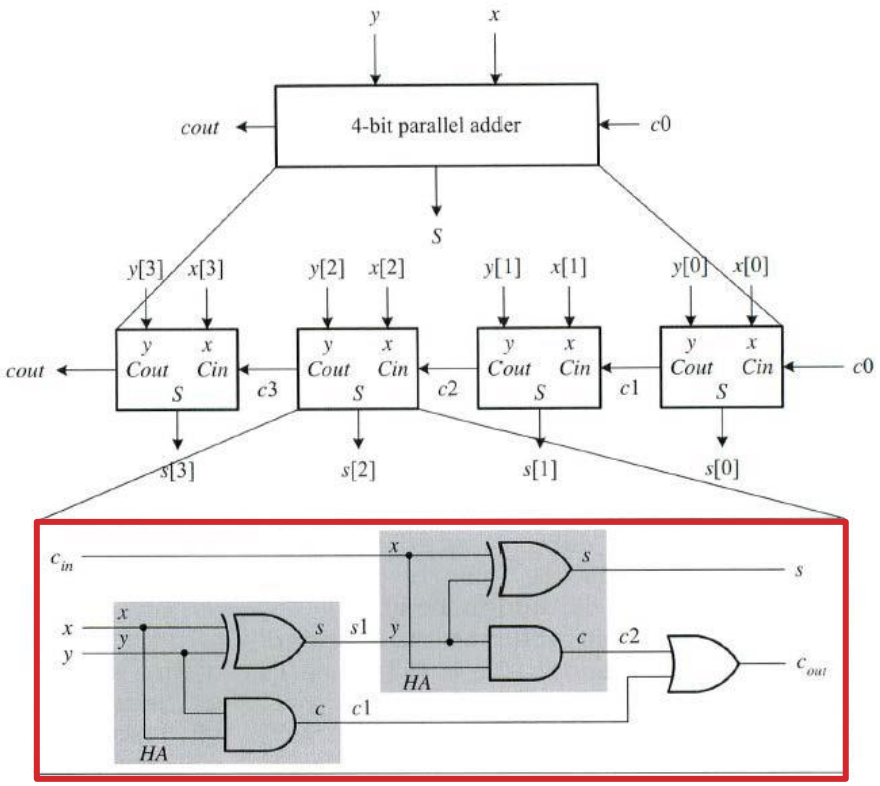
DESIGN EXAMPLE

- Vivado project
  - Running Simulation



DESIGN EXAMPLE

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



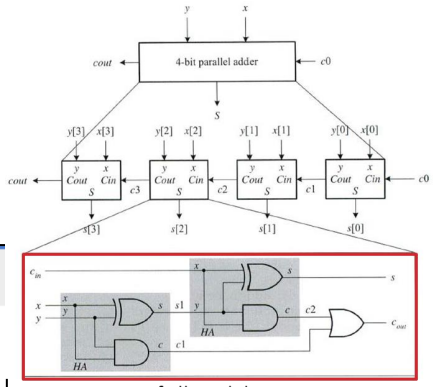
full_adder

Truth Table

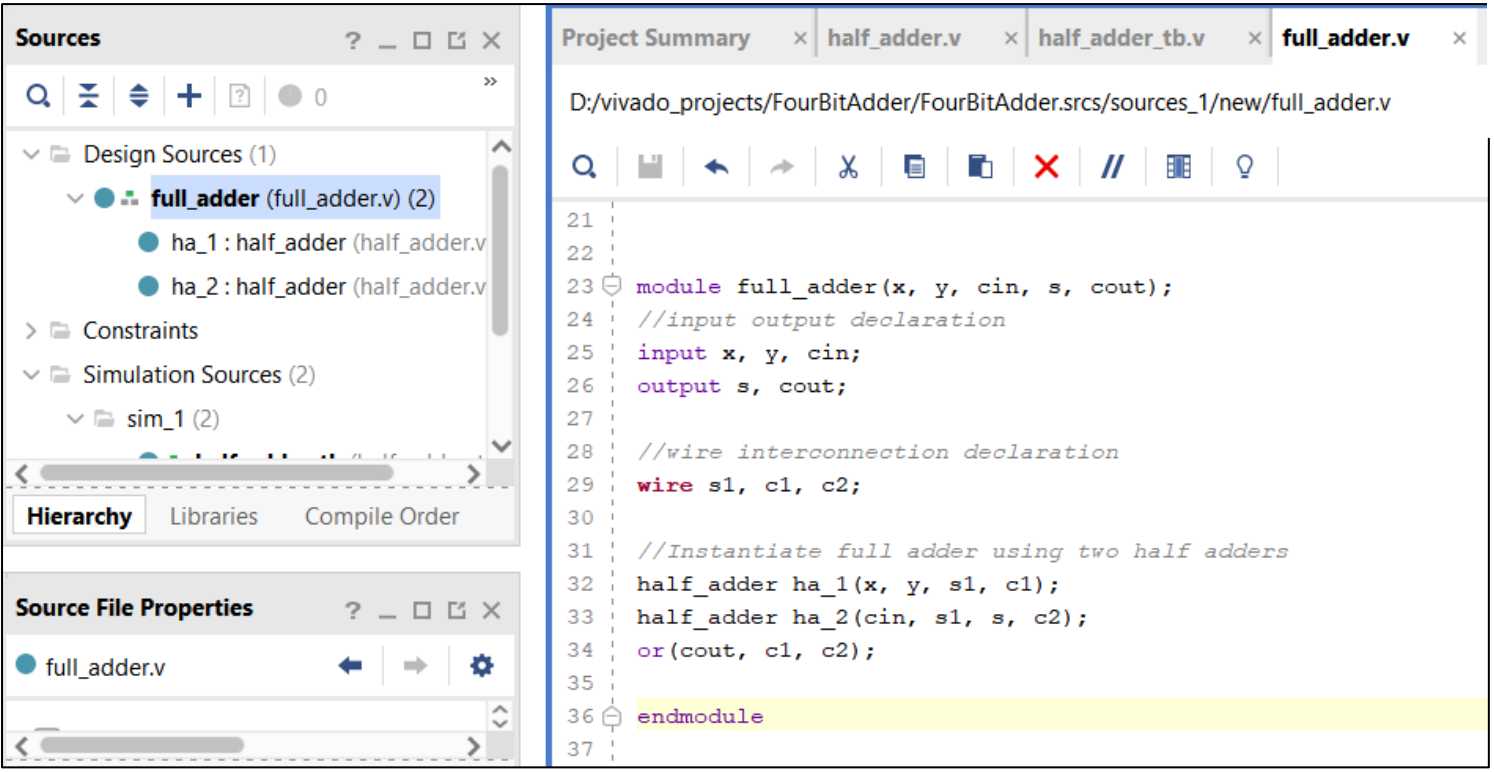
x	y	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

DESIGN EXAMPLE

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

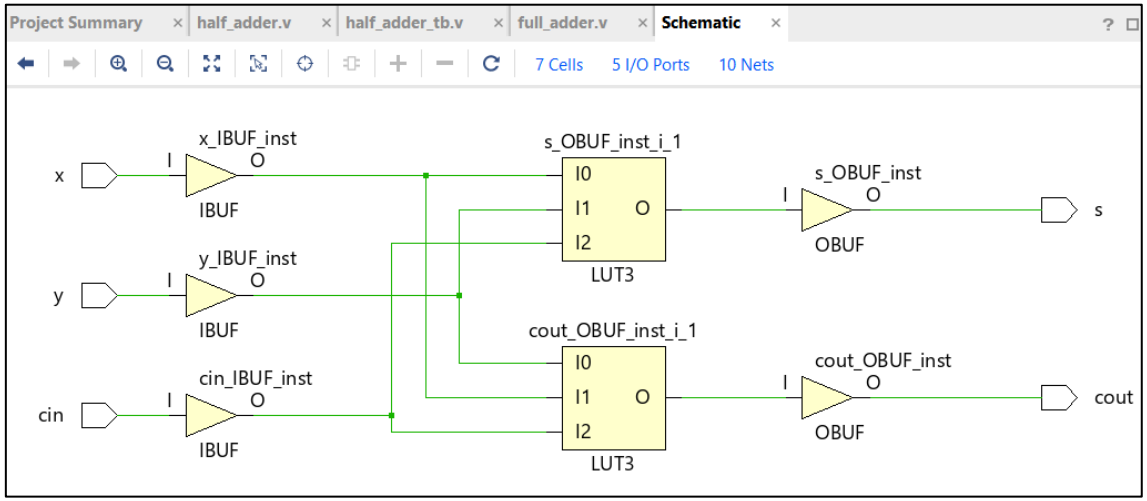


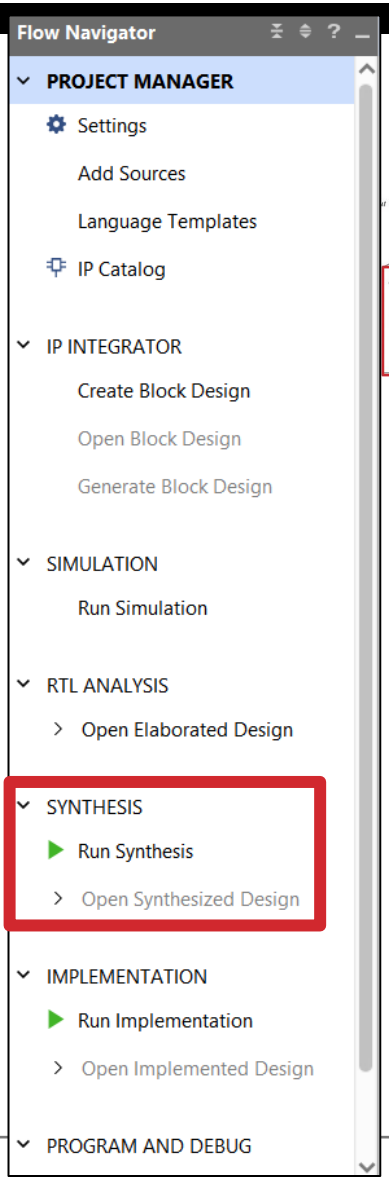
full_adder

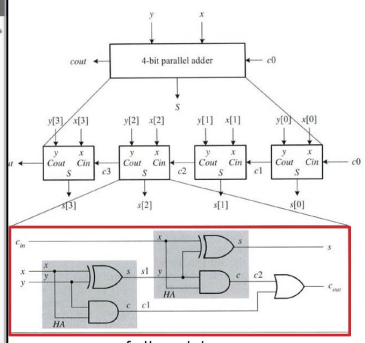


DESIGN EXAMPLE

- Verilog code and Testbench
  - Full Adder Schematic





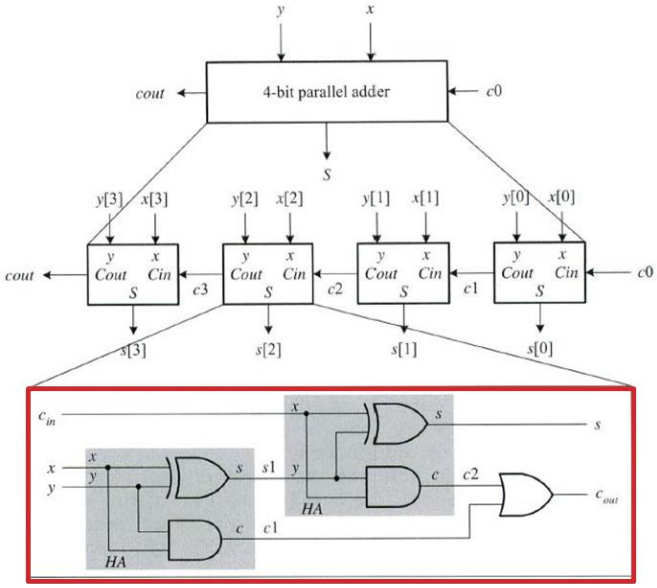


full_adder

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

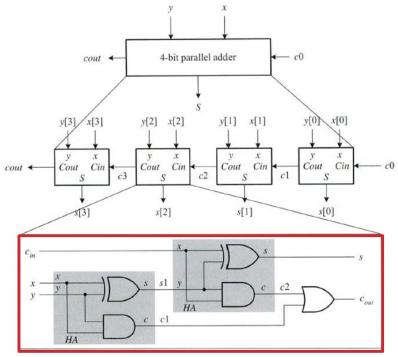


full_adder

DESIGN EXAMPLE

- Verilog code and Testbench
  - Full Adder Testbench

```
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
```

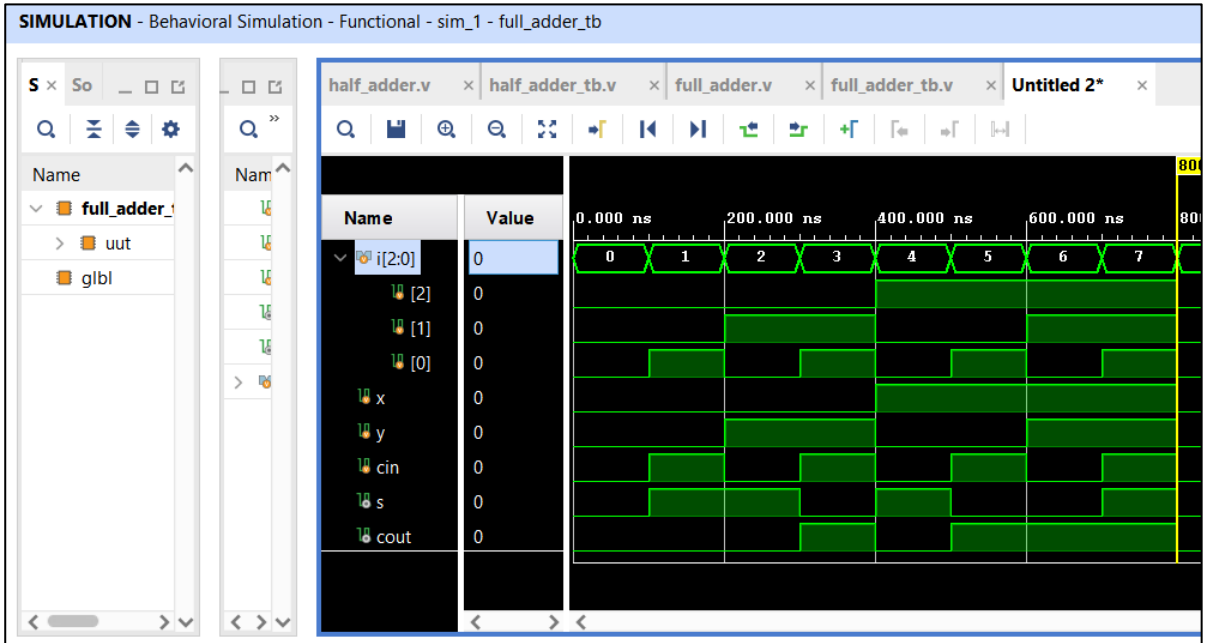


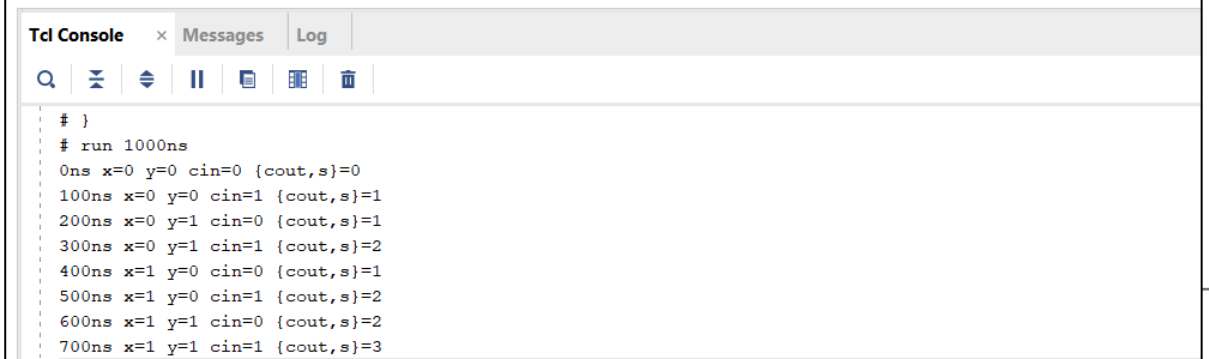
full_adder

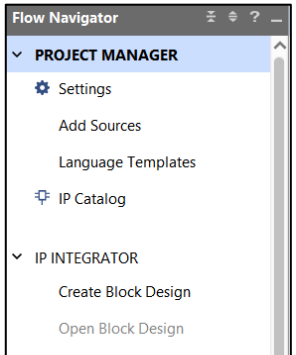
x	y	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

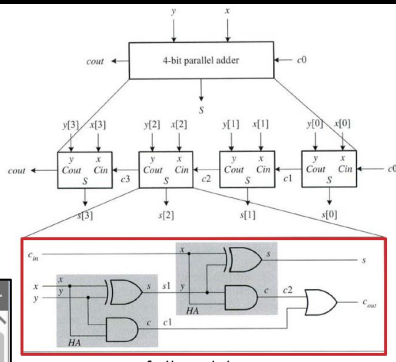
DESIGN EXAMPLE

- Verilog code and Testbench
  - Full Adder Simulation Waveform and Console







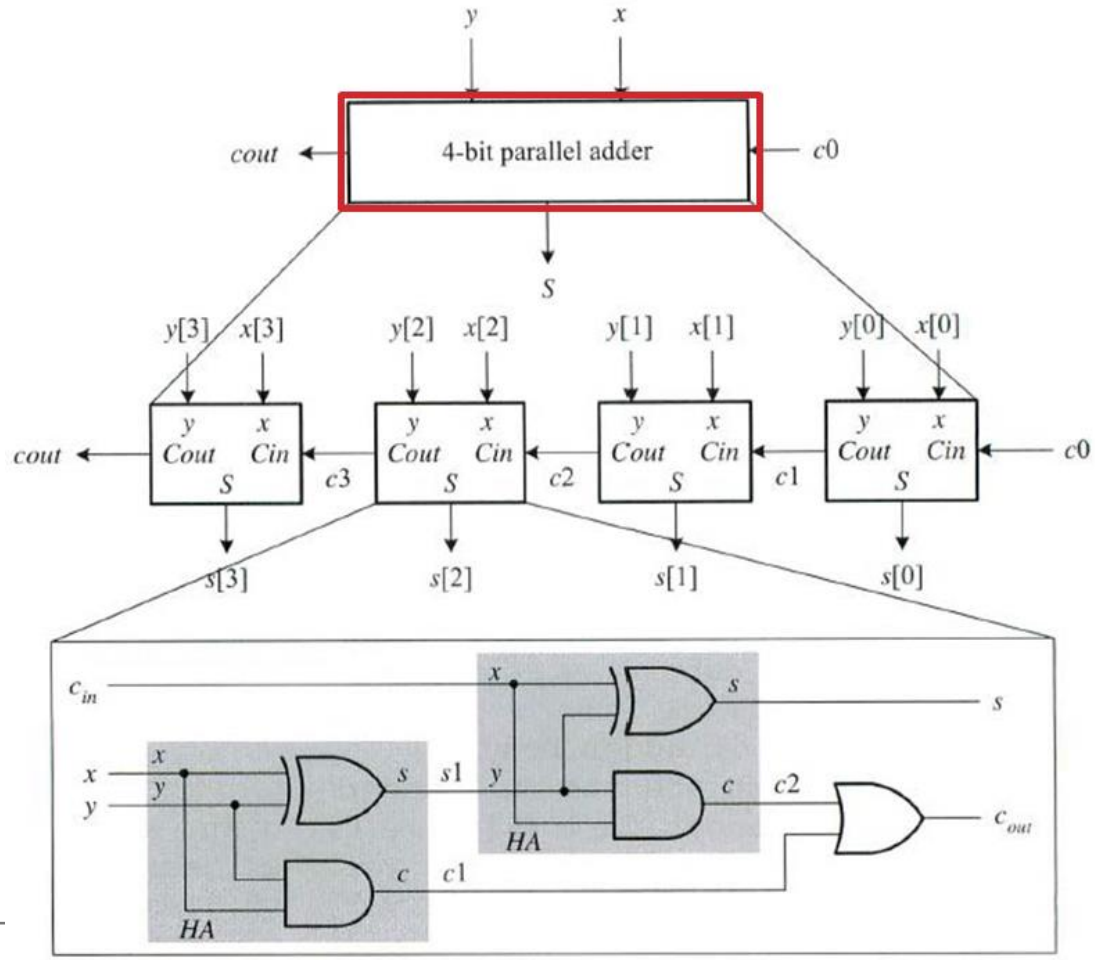


full_adder

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

DESIGN EXAMPLE

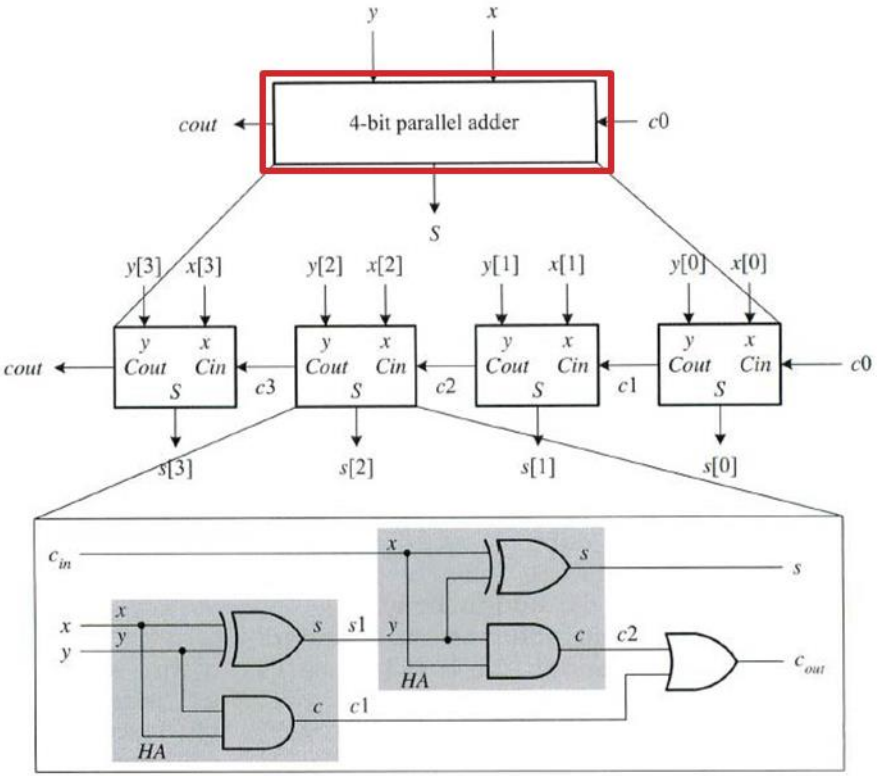
- Verilog code and Testbench
  - 4-bit Adder Design



DESIGN EXAMPLE

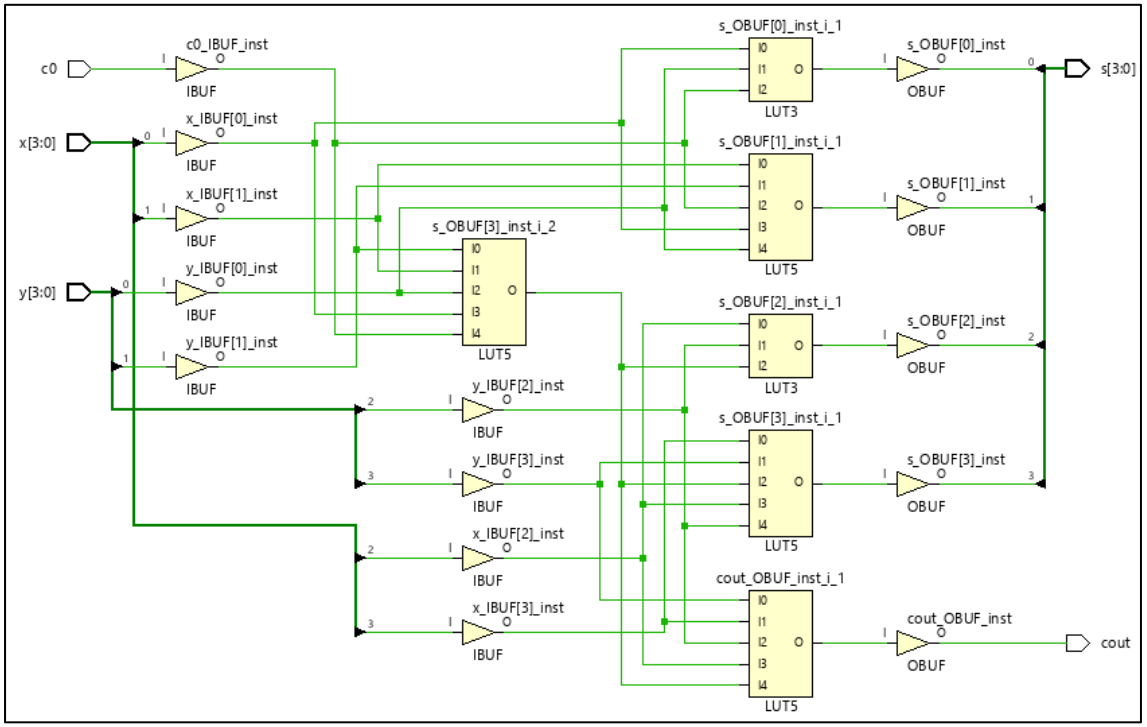
- Verilog code and Testbench
  - Verilog Code for 4-bit Adder module

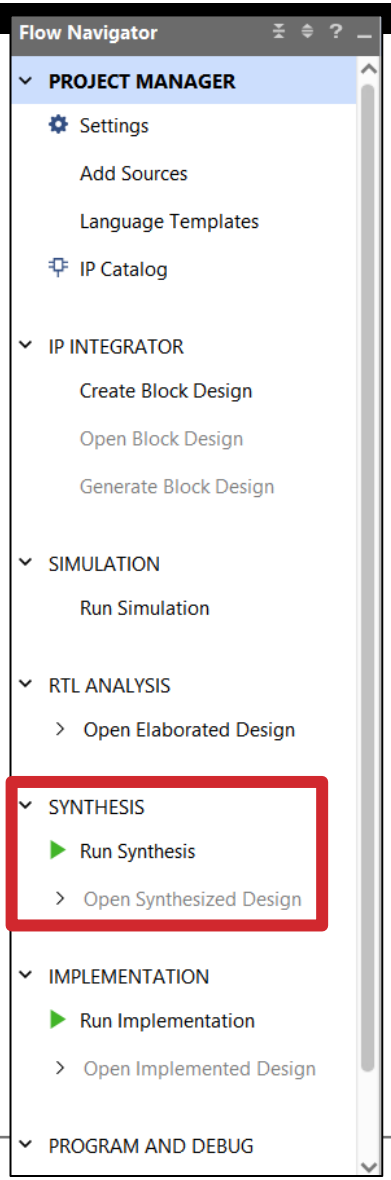
```
23  module four_bit_adder(x, y, c0, s, cout);
24  //input/output declaration
25  input [3:0] x, y;
26  input c0;
27  output [3:0] s;
28  output cout;
29
30  //wire interconnection declaration
31  wire c1, c2, c3;
32
33  //instantiate 4-bit adder using 4 full adders
34  full_adder fa_1(x[0], y[0],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

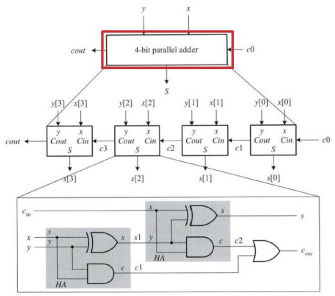


DESIGN EXAMPLE

- Verilog code and Testbench
  - 4-bit Adder Schematic

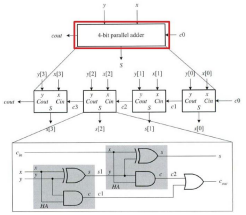






DESIGN EXAMPLE

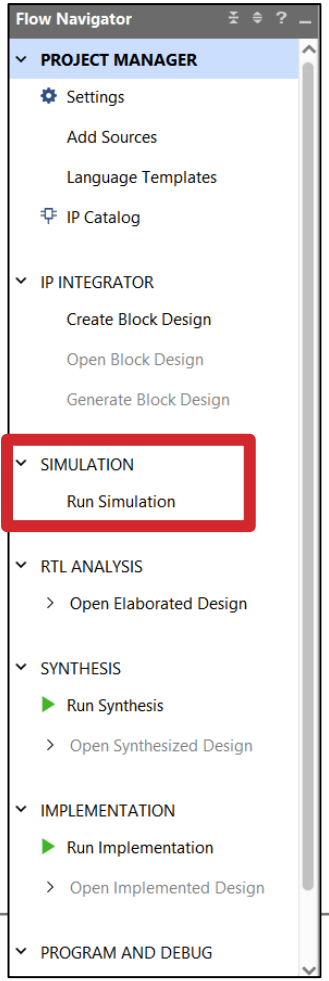
- Verilog code and Testbench
  - 4-bit Adder Testbench

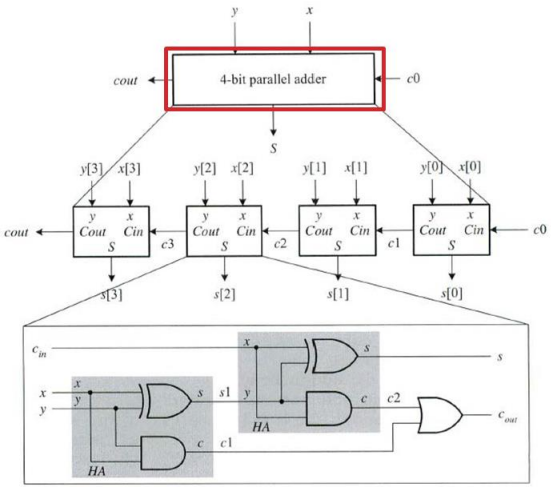


```
23  module four_bit_adder_tb();
24  //inputs
25  reg [3:0] x;
26  reg [3:0] y;
27  reg c0;
28  //outputs
29  wire [3:0] s;
30  wire cout;
31  //instantiate the Unit Under Test (UUT)
32  four_bit_adder uut(.x(x), .y(y), .c0(c0), .s(s), .cout(cout));
33
34  reg [7:0] i;
35  initial begin
36      //add 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

SIMULATION - Behavioral Simulation - Functional - sim_1 - four_bit_adder_tb

Scope | Sources

Name	Design...	Block T...
four_bit_adder_	four_bit_a	Verilog M
uut	four_bit_a	Verilog M
glbl	glbl	Verilog M

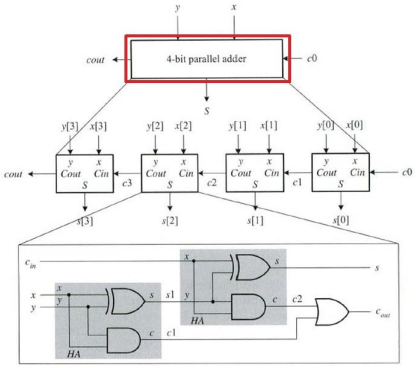
full_adder.v | half_adder.v | four_bit_adder.v | four_bit_adder_tb.v | Untitled 1*

Name	Value
i[63:0]	0000000000
x[3:0]	8
y[3:0]	8
c0	0
s[3:0]	0
cout	1

800.000 ns | 0.000 ns | 500.000 ns

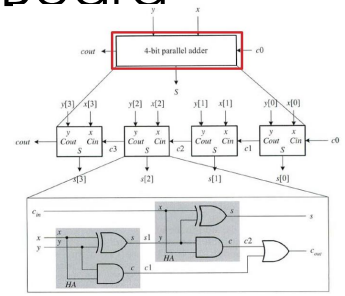
Tcl Console | Messages | Log

```
relaunch_sim: Time (s): cpu = 00:00:01 ; elapsed = 00:00:12 . Memory (MB): peak = 2027.723 ; gain = 0.000
run all
1100ns x=b y=b c0=0 {cout,s}=16
1200ns x=c y=c c0=0 {cout,s}=18
1300ns x=d y=d c0=0 {cout,s}=1a
1400ns x=e y=e c0=0 {cout,s}=1c
1500ns x=f y=f c0=0 {cout,s}=1e
1600ns x=0 y=0 c0=0 {cout,s}=00
1700ns x=1 y=1 c0=0 {cout,s}=02
1800ns x=2 y=2 c0=0 {cout,s}=04
1900ns x=3 y=3 c0=0 {cout,s}=06
2000ns x=4 y=4 c0=0 {cout,s}=08
2100ns x=5 y=5 c0=0 {cout,s}=0a
```

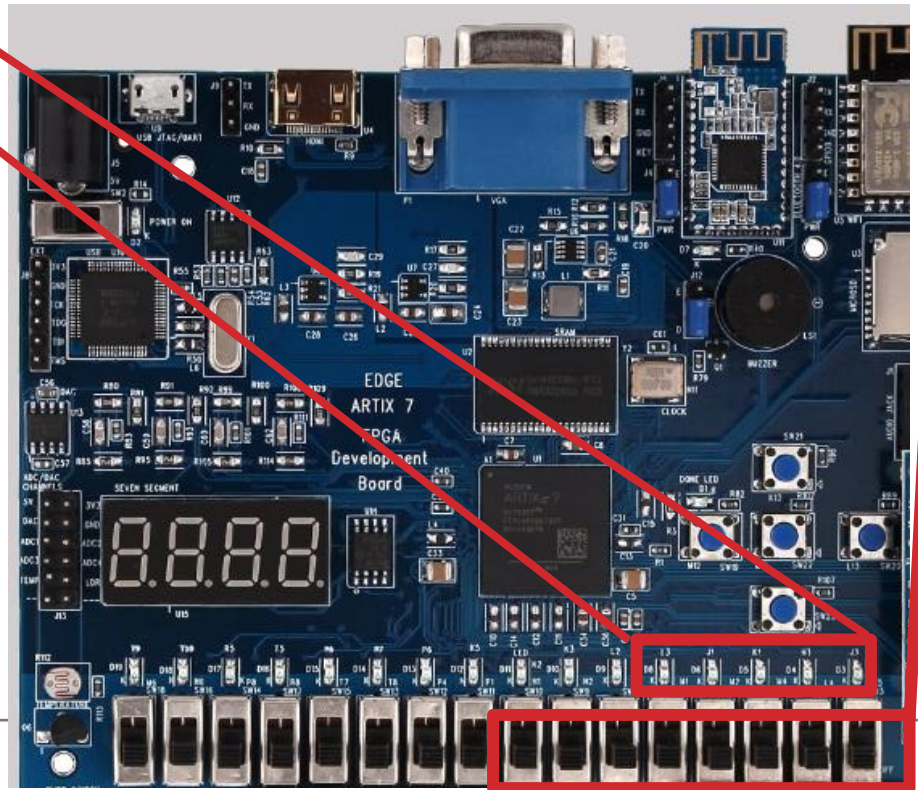


DESIGN EXAMPLE

- 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

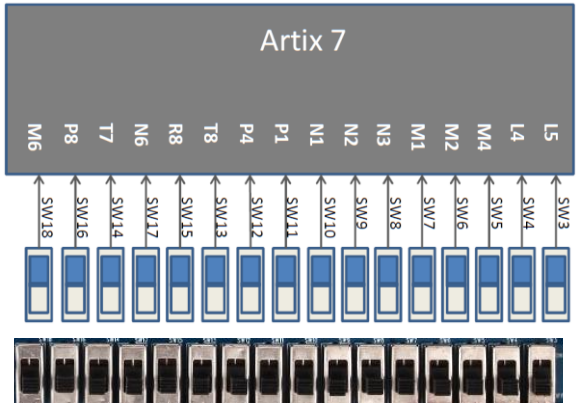


Net Name	PIN MAP
SW0	L5
SW1	L4
SW2	M4
SW3	M2
SW4	M1
SW5	N3
SW6	N2
SW7	N1

A

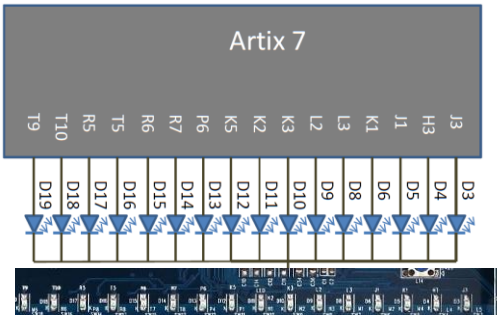
DESIGN EXAMPLE

- 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

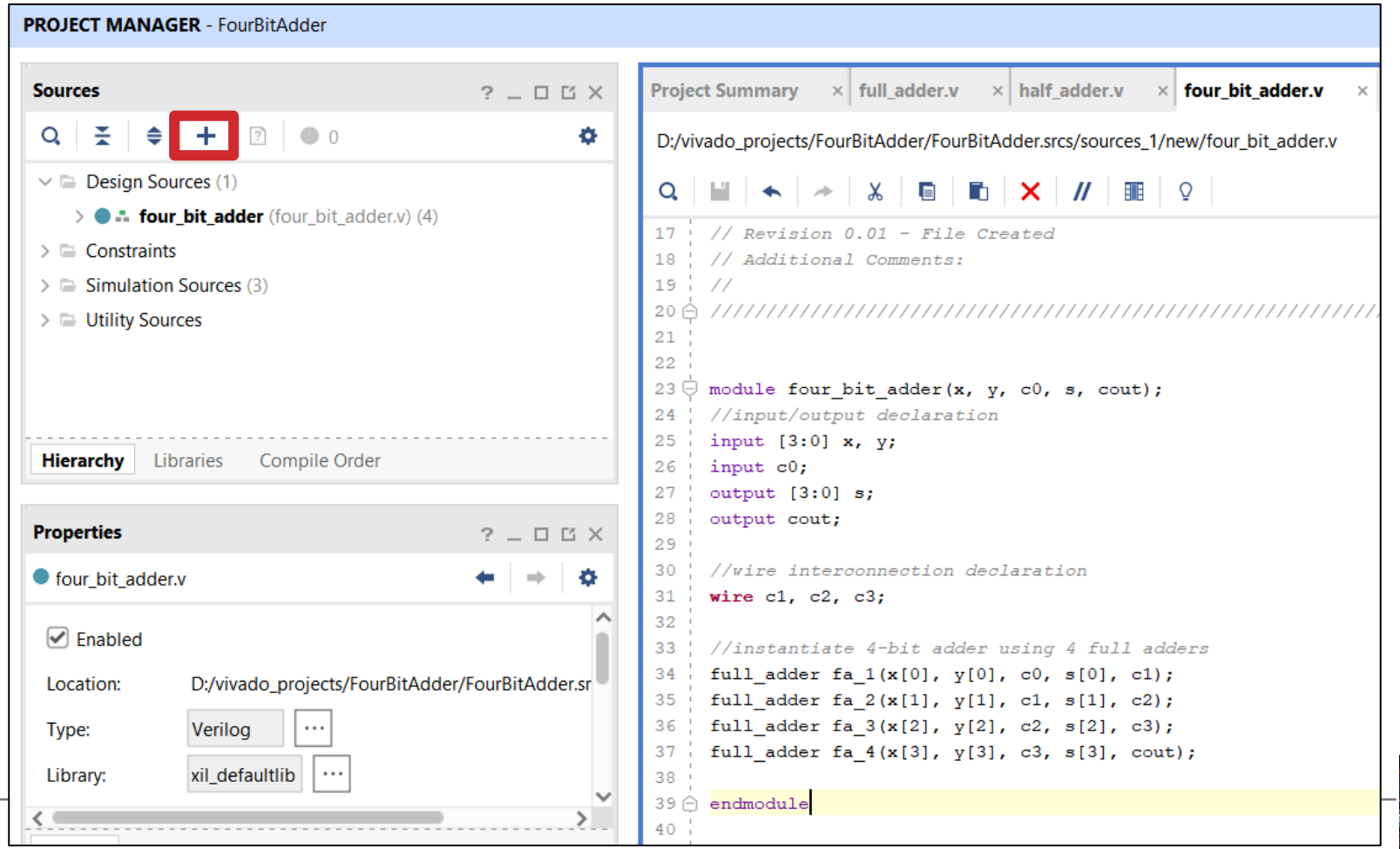



```
# LEDs
set_property -dict { PACKAGE_PIN J3    IOSTANDARD LVCMOS33 } [get_ports { led[0] }];#LSB
set_property -dict { PACKAGE_PIN H3    IOSTANDARD LVCMOS33 } [get_ports { led[1] }];
set_property -dict { PACKAGE_PIN J1    IOSTANDARD LVCMOS33 } [get_ports { led[2] }];
set_property -dict { PACKAGE_PIN 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

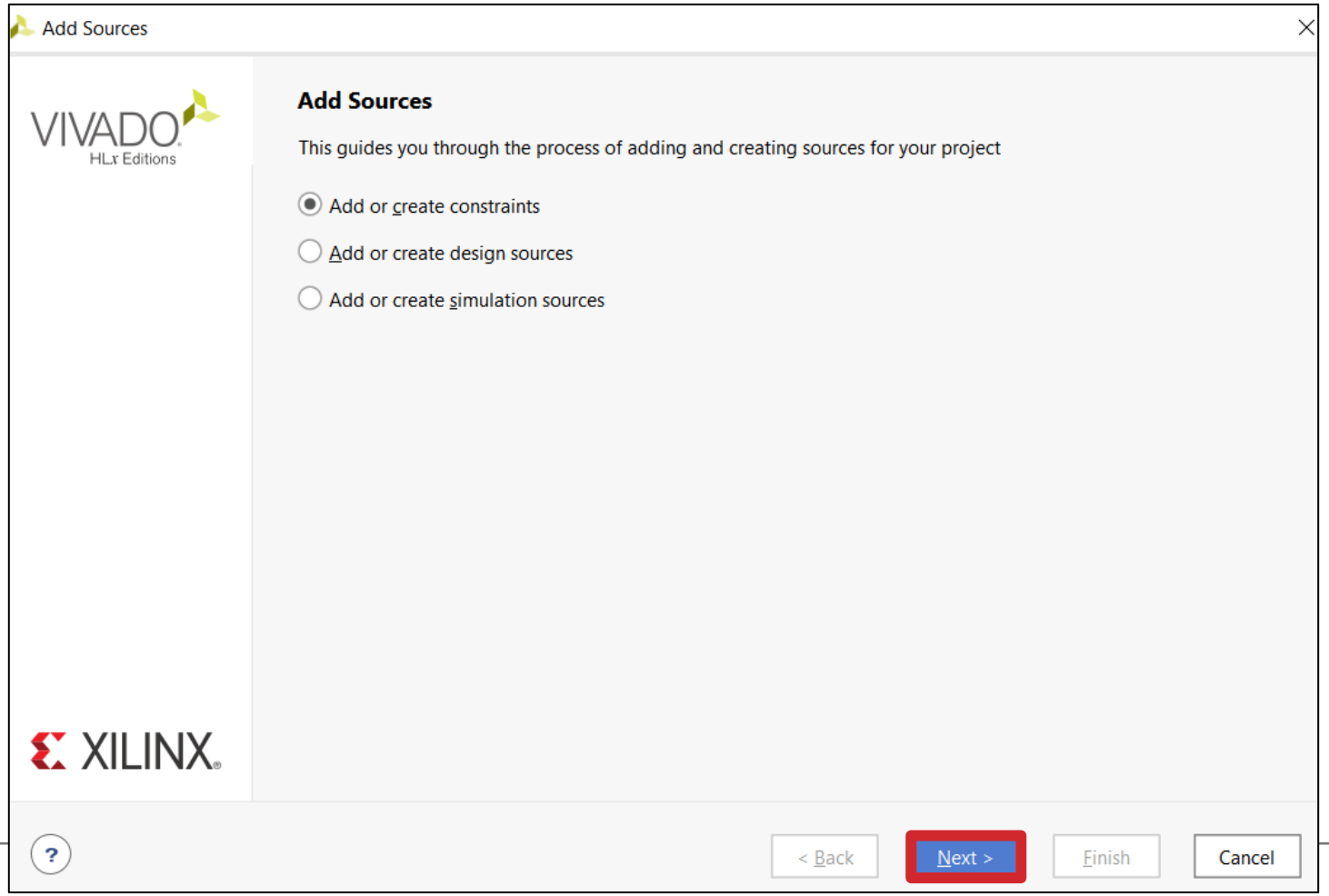
DESIGN EXAMPLE

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



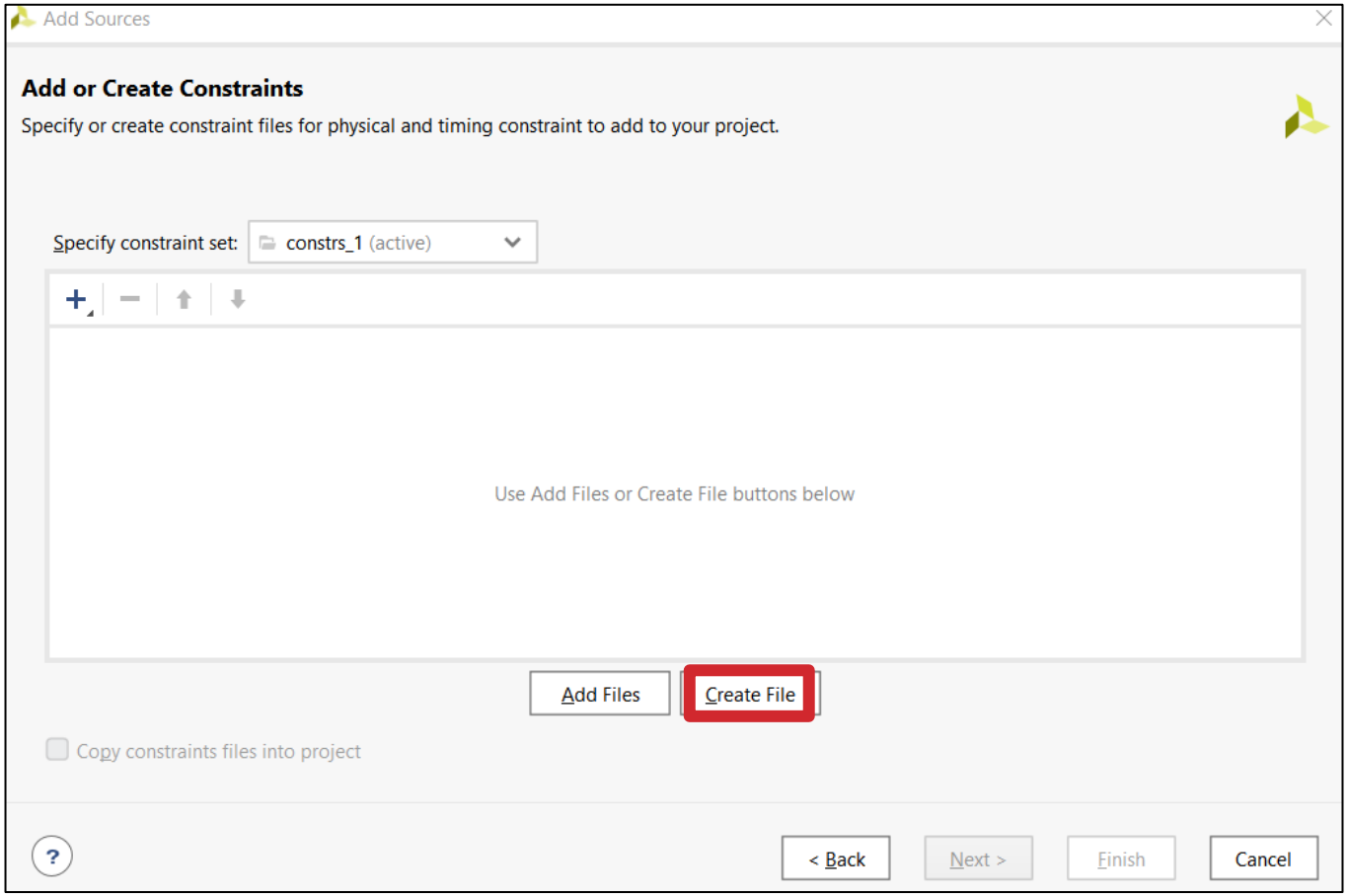
DESIGN EXAMPLE

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



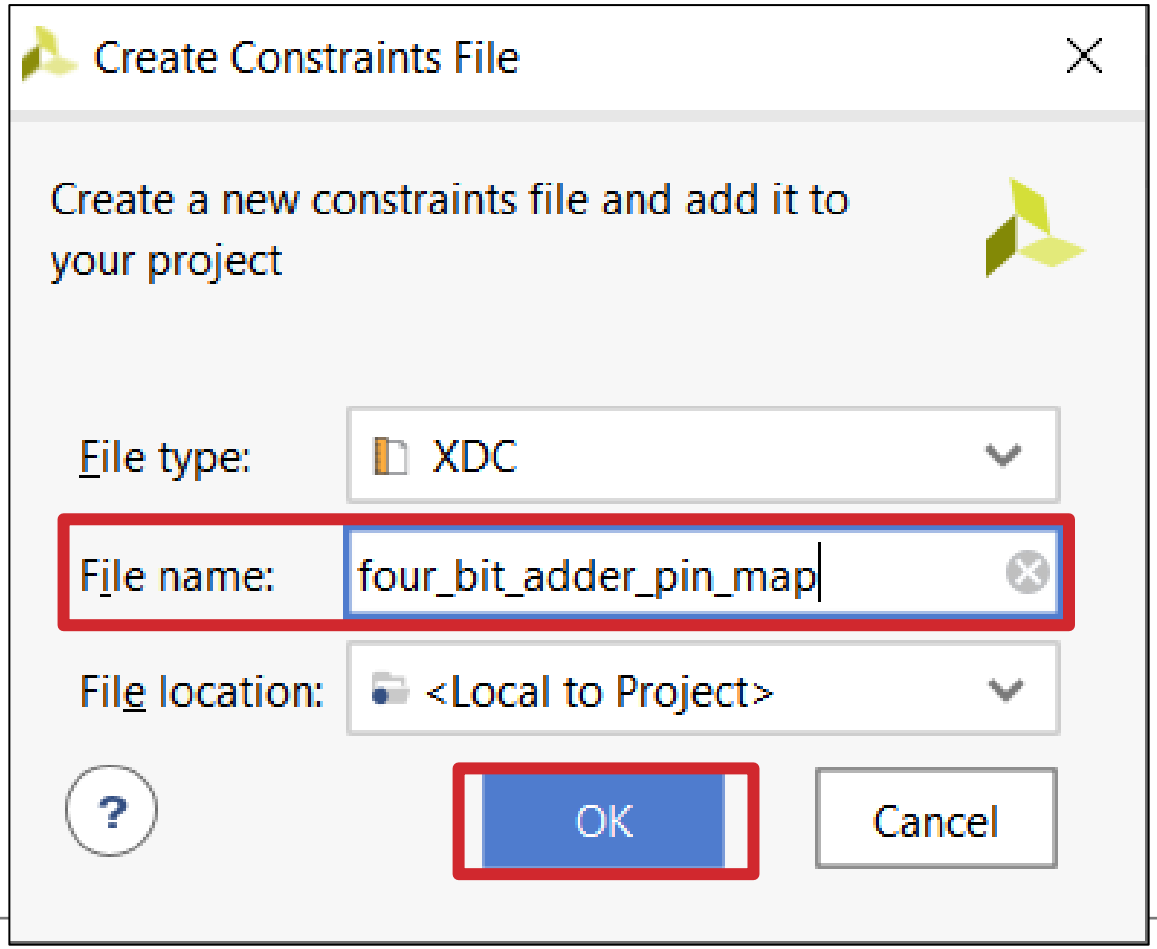
DESIGN EXAMPLE

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



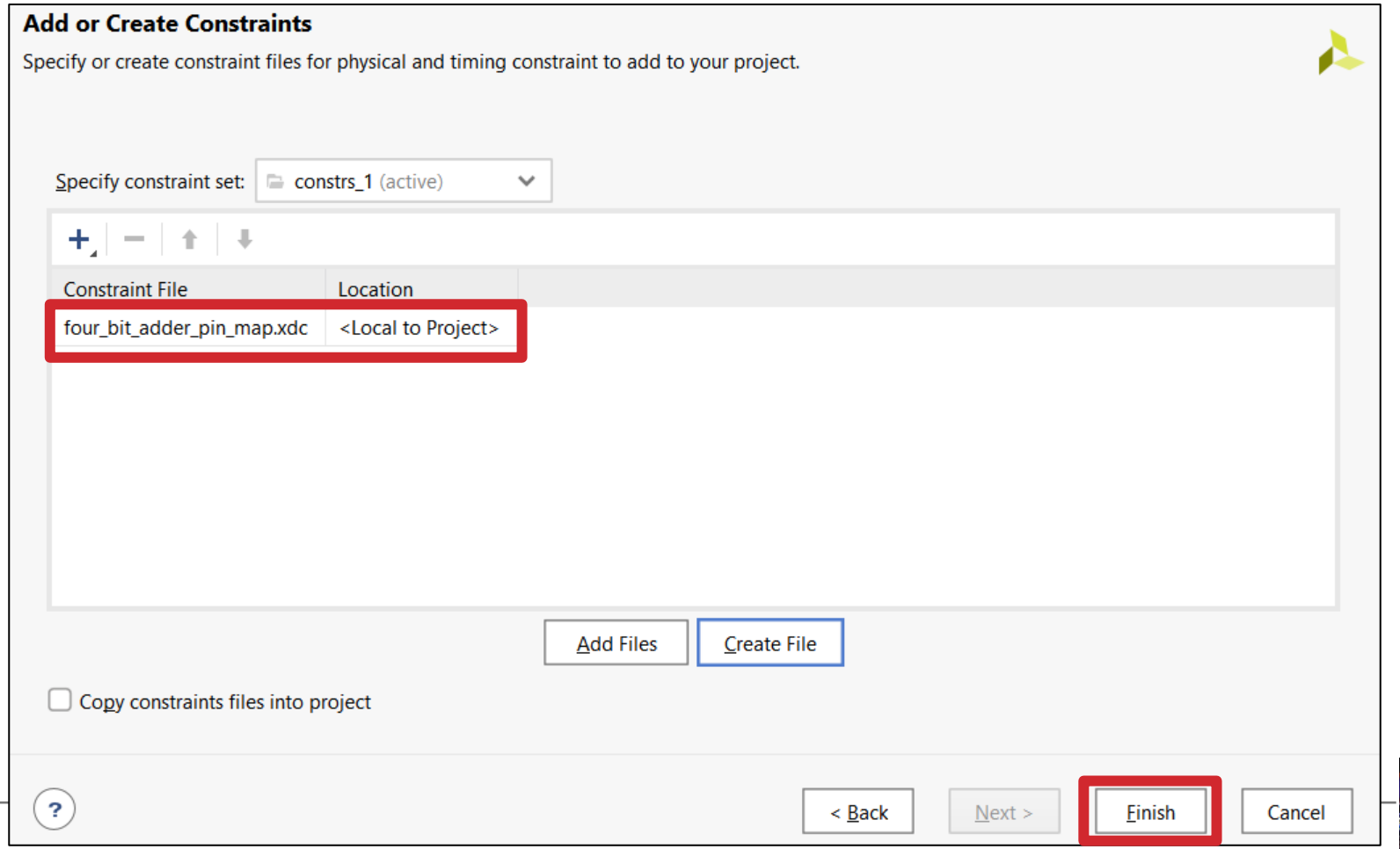
DESIGN EXAMPLE

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



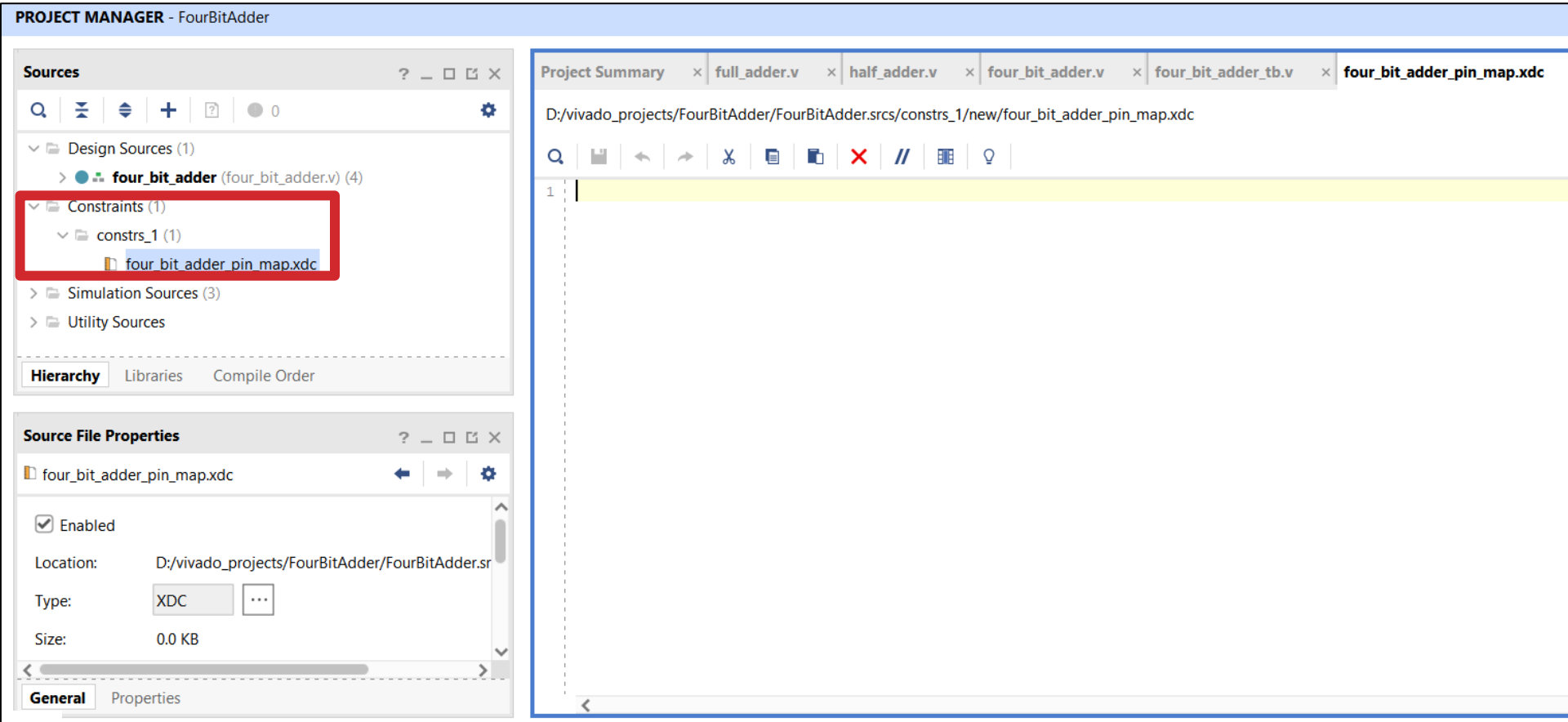
DESIGN EXAMPLE

- 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

Project Summary | full_adder.v | half_adder.v | four_bit_adder.v | four_bit_adder_tb.v | four_bit_ad

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

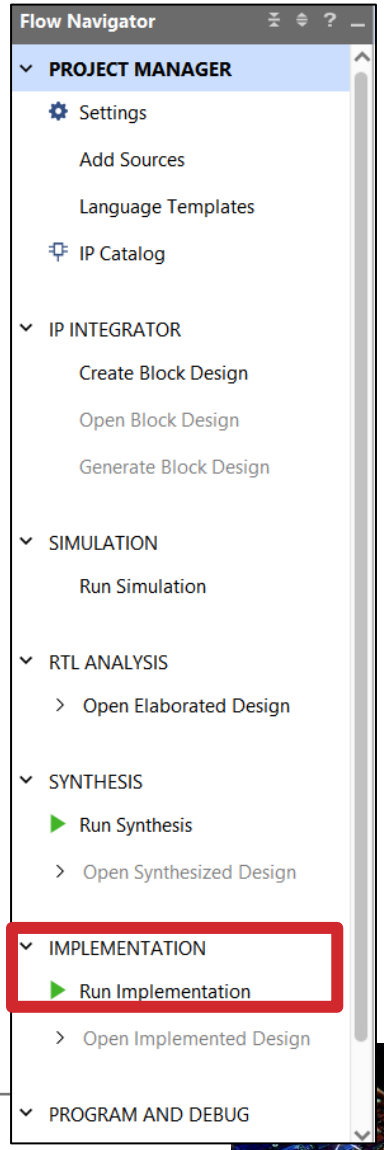
```
# Switches
set_property -dict { PACKAGE_PIN L5    IOSTANDARD LVCMOS33 } [get_ports { y[0] }];#LSB
set_property -dict { PACKAGE_PIN L4    IOSTANDARD LVCMOS33 } [get_ports { y[1] }];
set_property -dict { PACKAGE_PIN M4    IOSTANDARD LVCMOS33 } [get_ports { y[2] }];
set_property -dict { PACKAGE_PIN M2    IOSTANDARD LVCMOS33 } [get_ports { y[3] }];
set_property -dict { PACKAGE_PIN M1    IOSTANDARD LVCMOS33 } [get_ports { x[0] }];
set_property -dict { PACKAGE_PIN N3    IOSTANDARD LVCMOS33 } [get_ports { x[1] }];
set_property -dict { PACKAGE_PIN N2    IOSTANDARD LVCMOS33 } [get_ports { x[2] }];
set_property -dict { PACKAGE_PIN N1    IOSTANDARD LVCMOS33 } [get_ports { x[3] }];#MSB

# LEDs
set_property -dict { PACKAGE_PIN J3    IOSTANDARD LVCMOS33 } [get_ports { s[0] }];#LSB
set_property -dict { PACKAGE_PIN H3    IOSTANDARD LVCMOS33 } [get_ports { s[1] }];
set_property -dict { PACKAGE_PIN J1    IOSTANDARD LVCMOS33 } [get_ports { s[2] }];
set_property -dict { PACKAGE_PIN K1    IOSTANDARD LVCMOS33 } [get_ports { s[3] }];
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

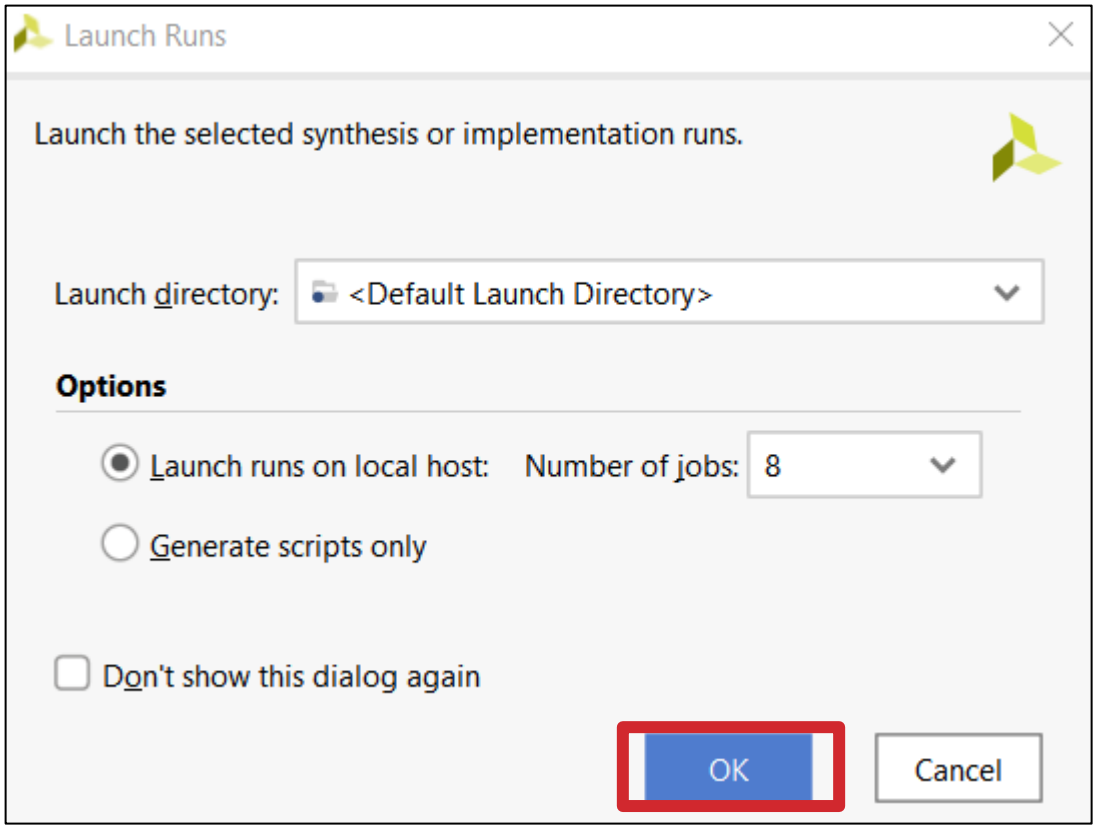
DESIGN EXAMPLE

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



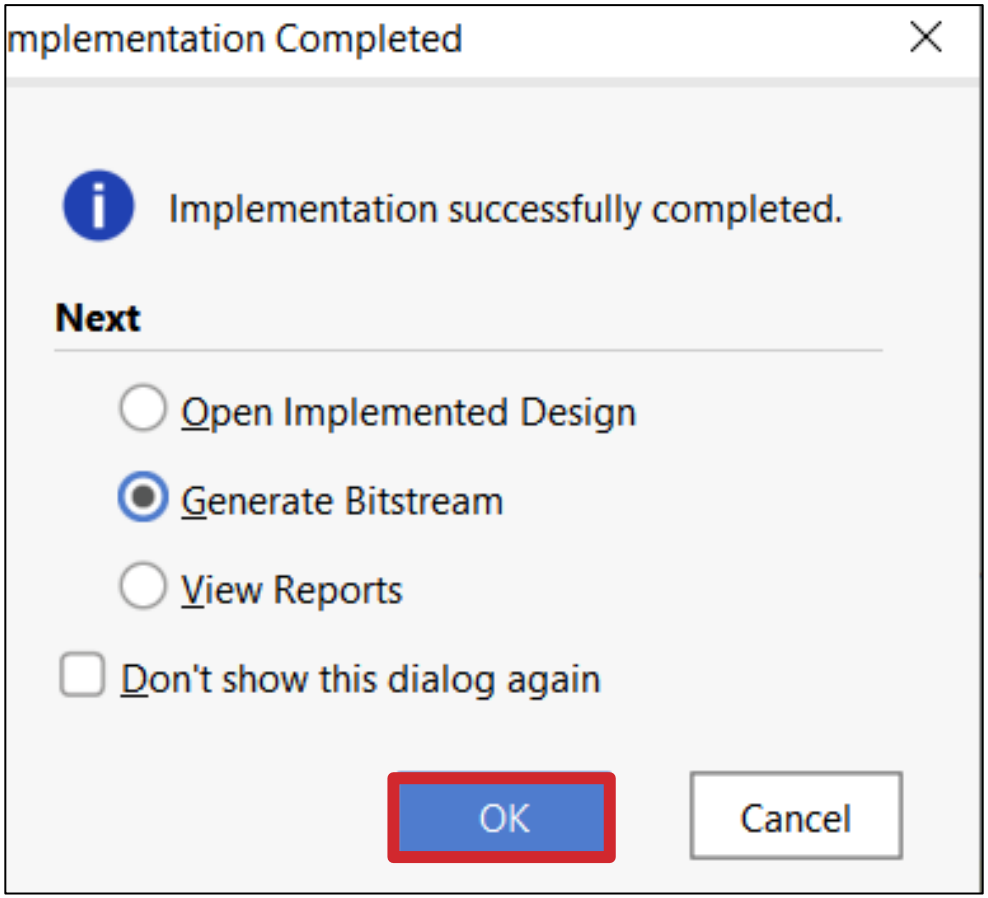
DESIGN EXAMPLE

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



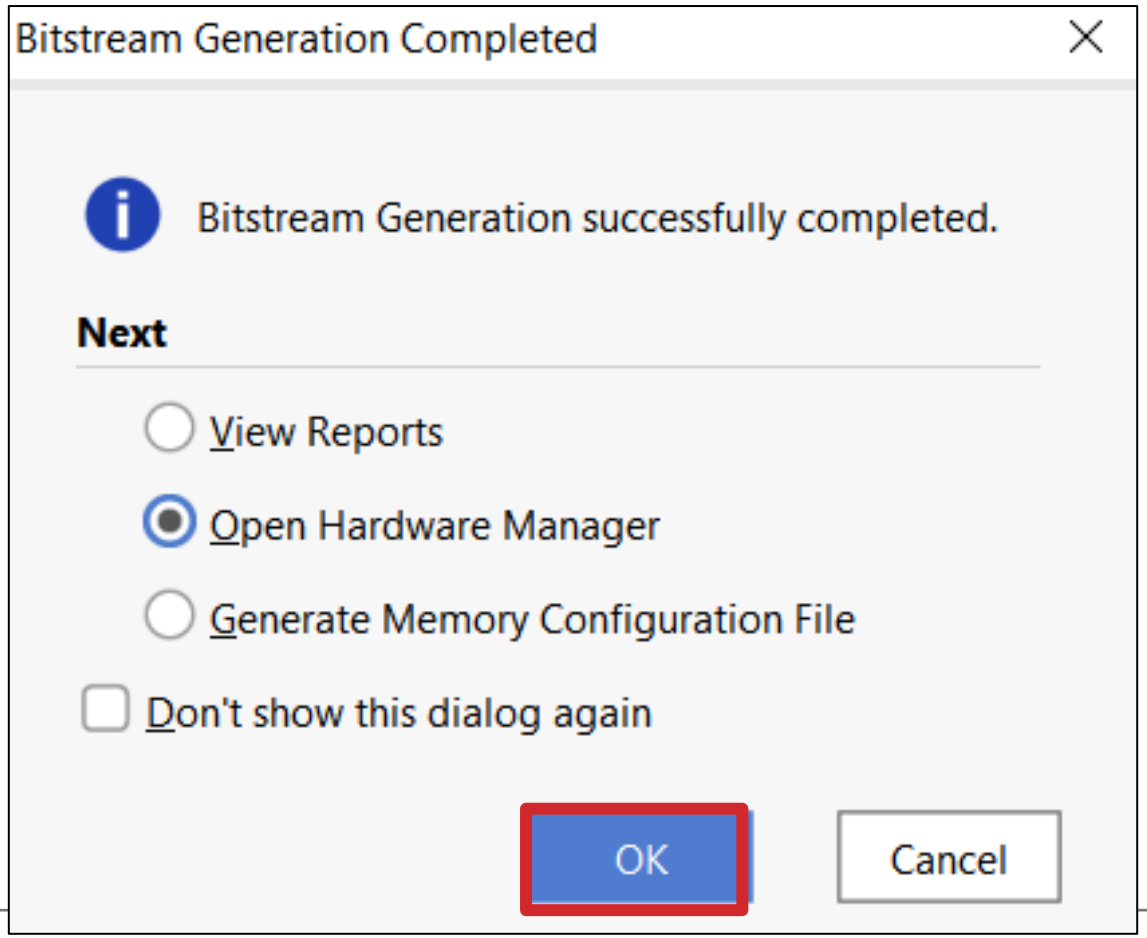
DESIGN EXAMPLE

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



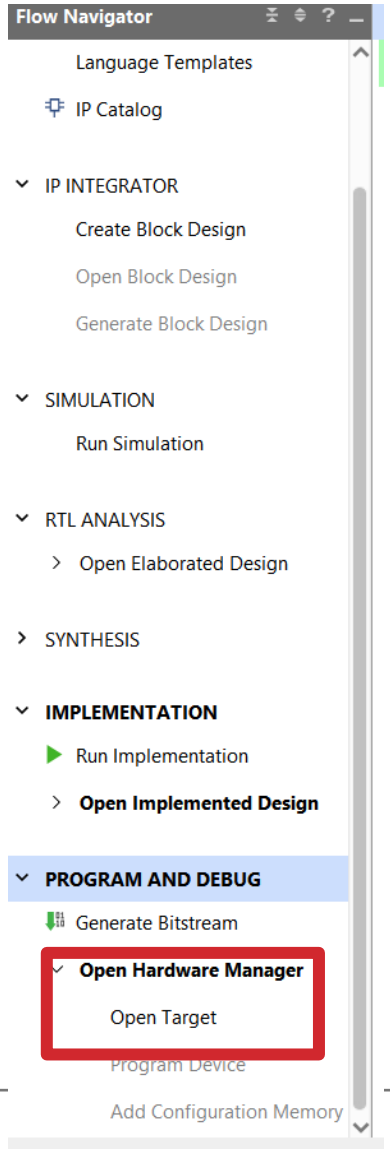
DESIGN EXAMPLE

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



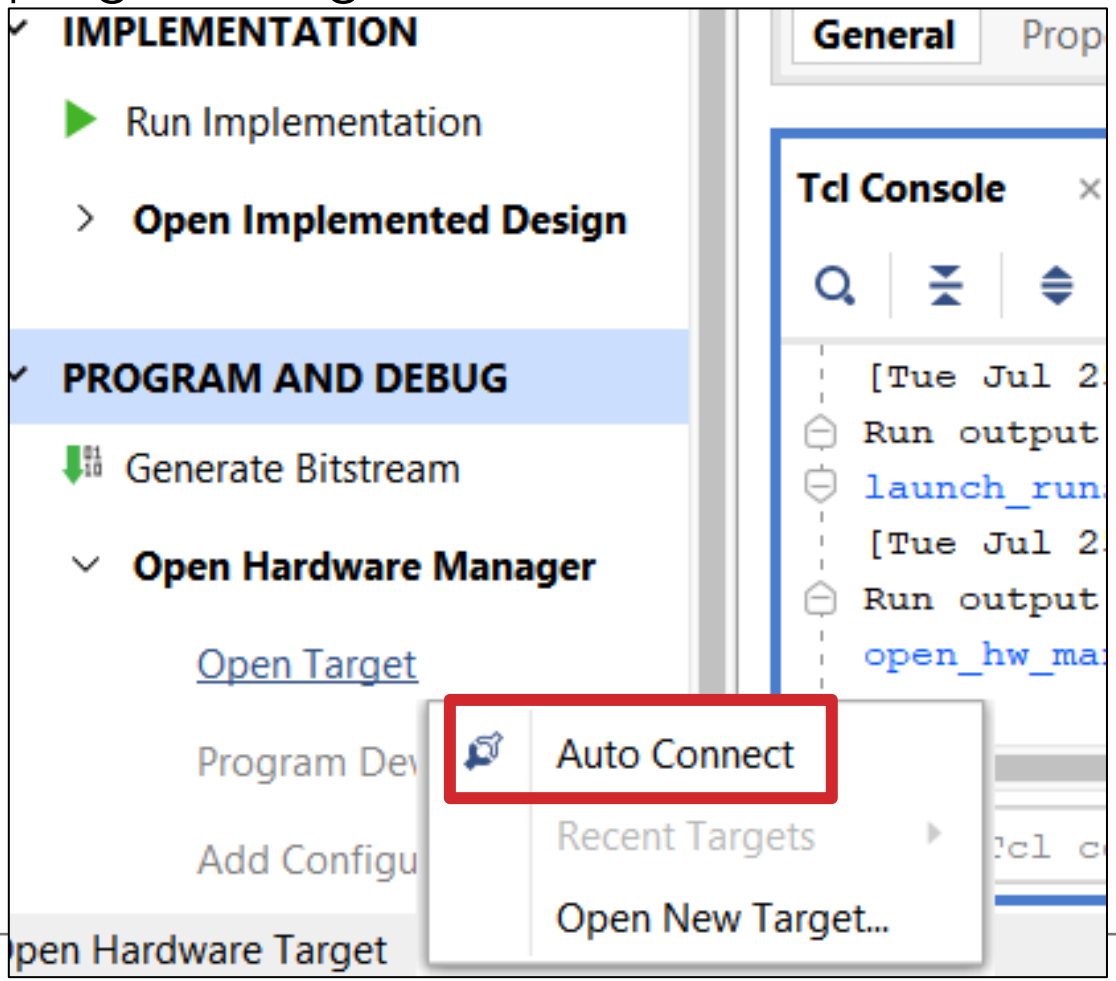
DESIGN EXAMPLE

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



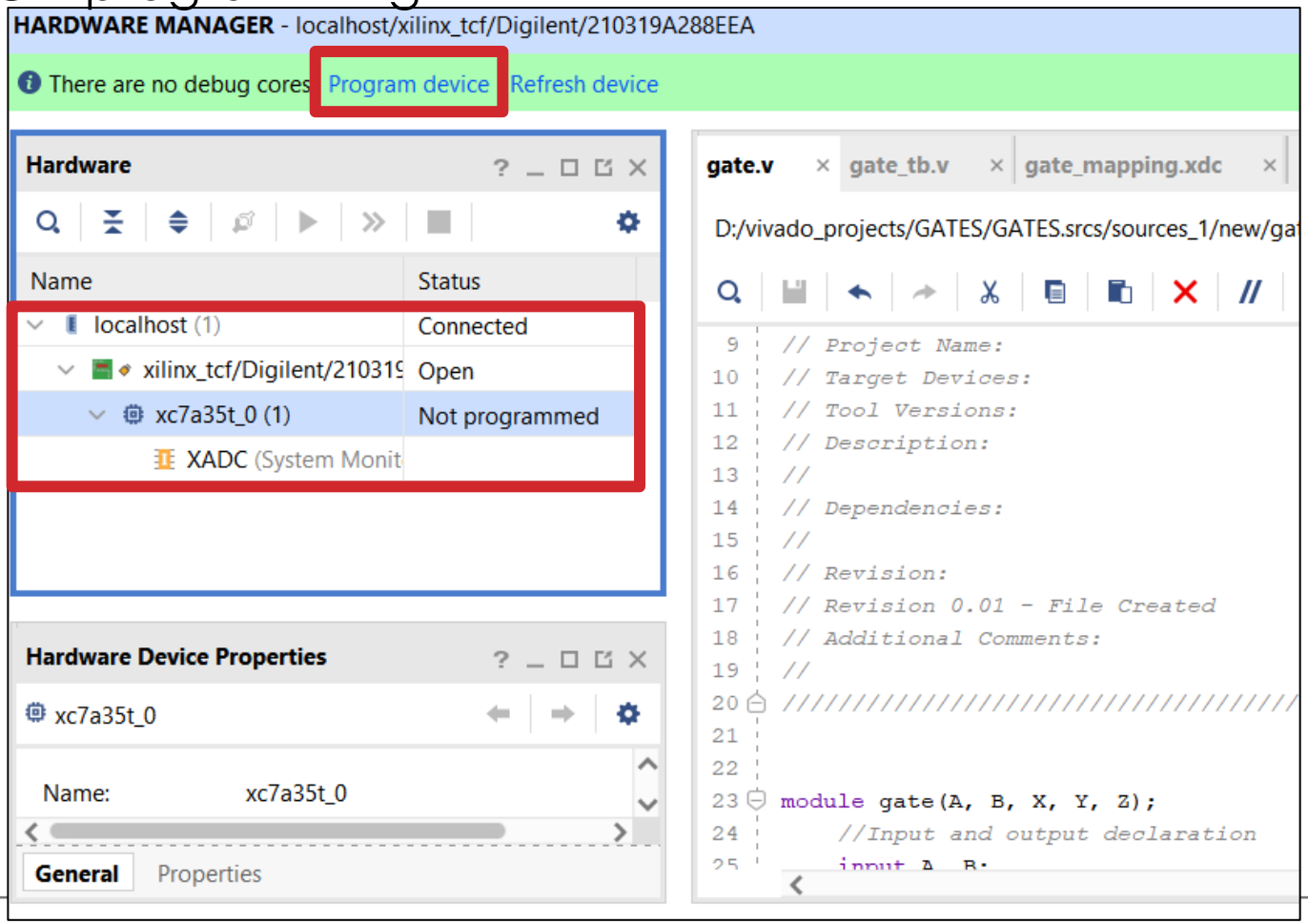
DESIGN EXAMPLE

- Vivado project
  - FPGA programming



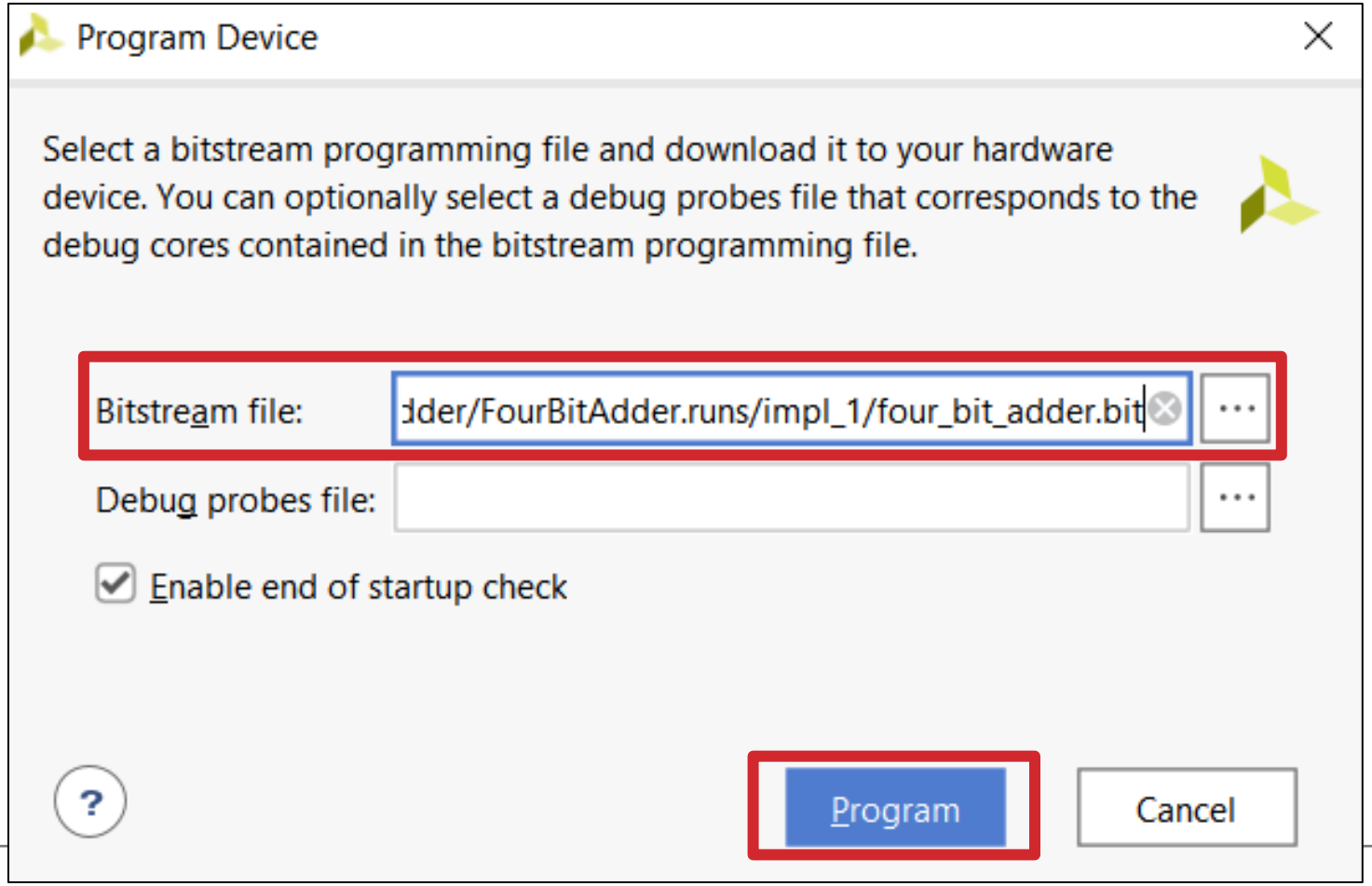
DESIGN EXAMPLE

- Vivado project
  - FPGA programming



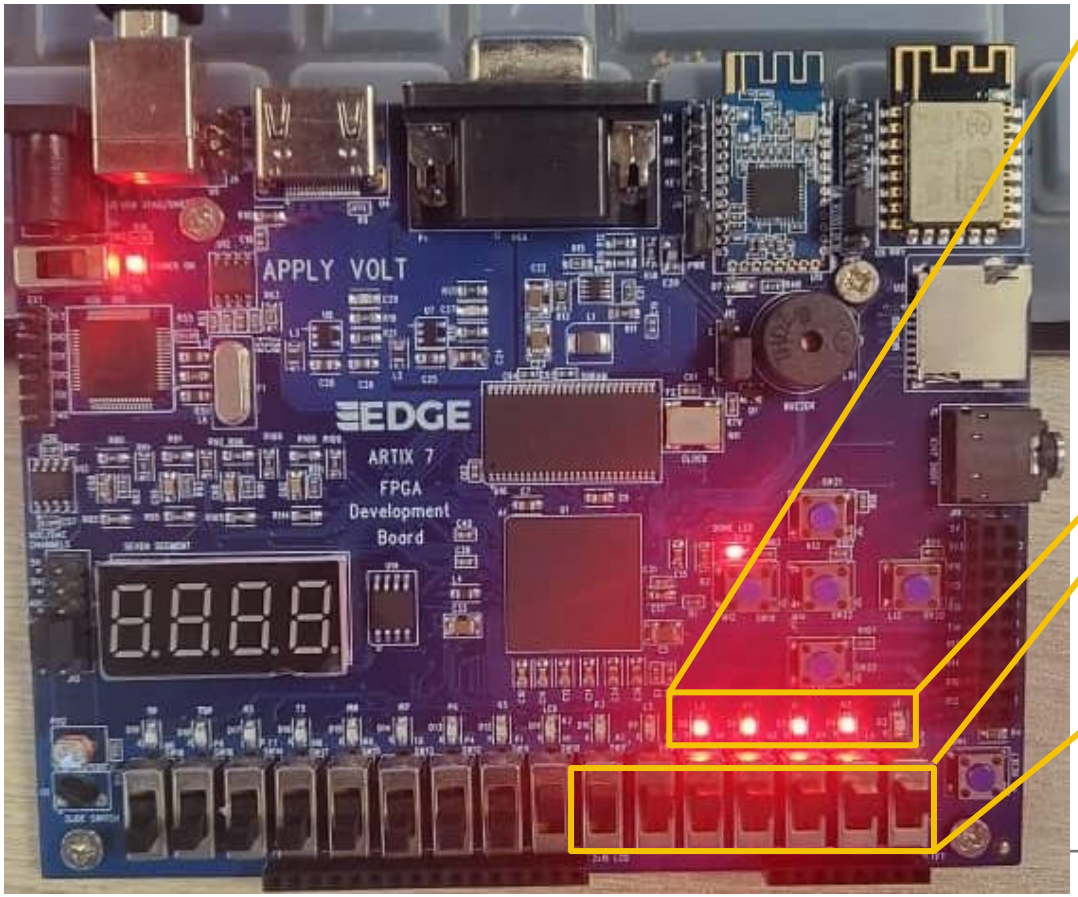
DESIGN EXAMPLE

- Vivado project
  - FPGA programming



DESIGN EXAMPLE

- Programming FPGA
  - Board Demo
    - Example:15 + 15 = 30

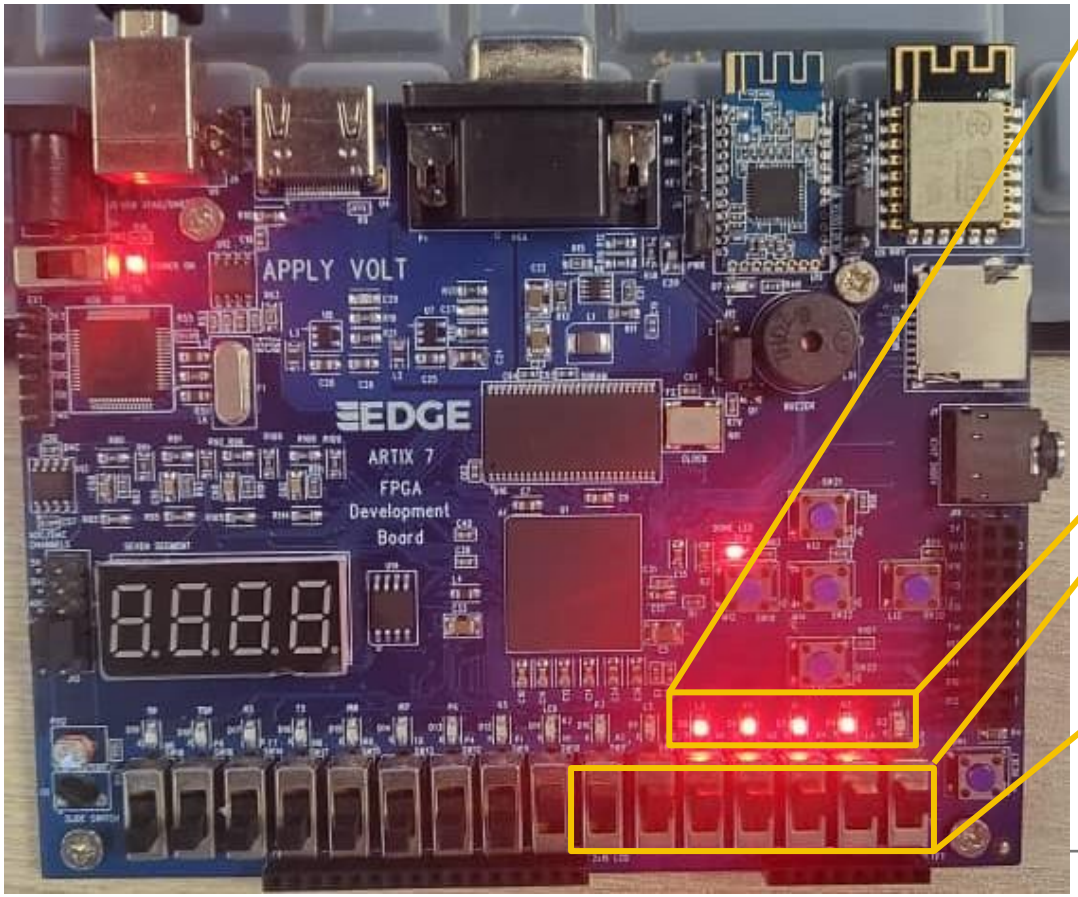


{c_out, sum[3:0]}
Binary: 11110
Decimal: 30

{x[3:0], y[3:0]}
Binary: {11111, 1111}
Decimal: {15, 15}

DESIGN EXAMPLE

- Programming FPGA
  - Board Demo
    - Example:15 + 15 = 30

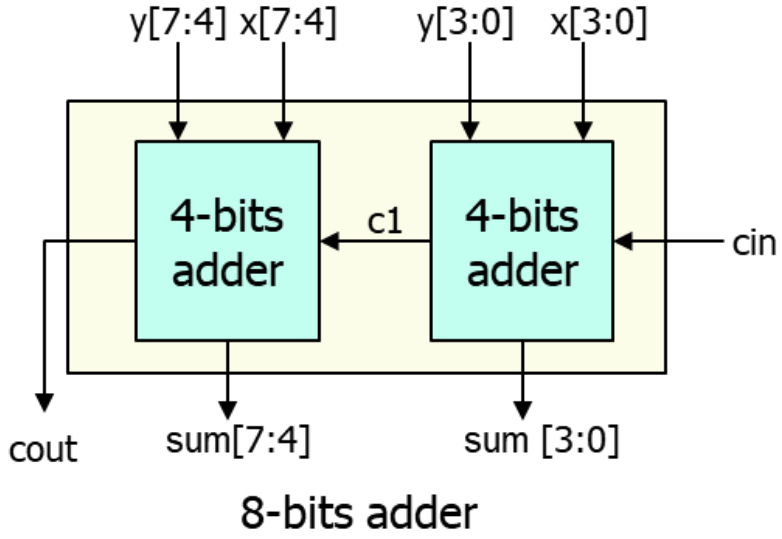


{c_out, sum[3:0]}
Binary: 11110
Decimal: 30

{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