EECS 207002
Logic Design Laboratory
邏輯設計實驗

Vivado Basys3 Implementation

黃元豪
Yuan-Hao Huang

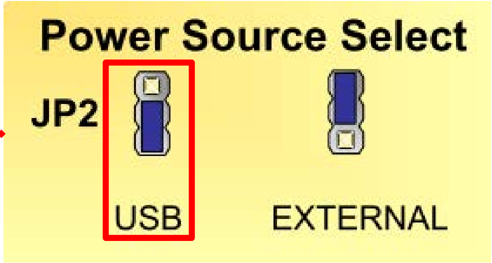
國立清華大學電機工程學系
Department of Electrical Engineering
National Tsing-Hua University

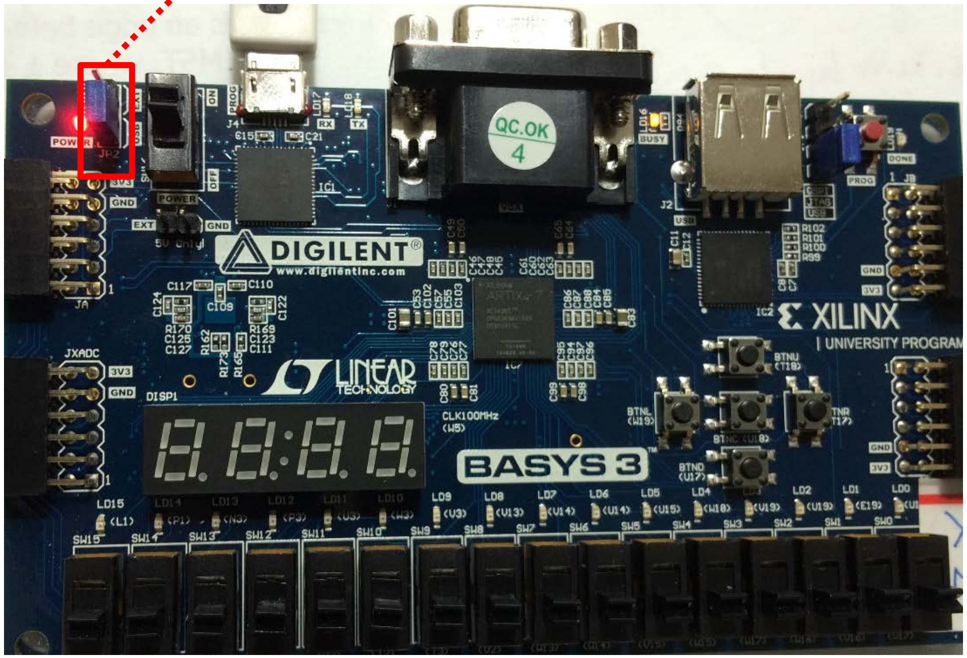
Outline

- Introduction to Basys3
- Preparation
- Synthesis
- Implementation
- Bitstream Generation
- Open Target
- Program Device
- Lab1_1
- Lab1_4

Introduction to Basys3 (1/3)

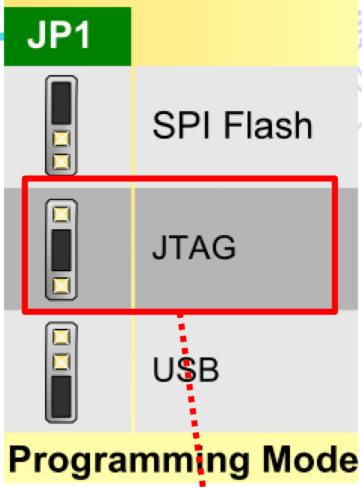
- Receive "Power" from:
  - the microUSB port (recommend)
  - a 5V external power supply

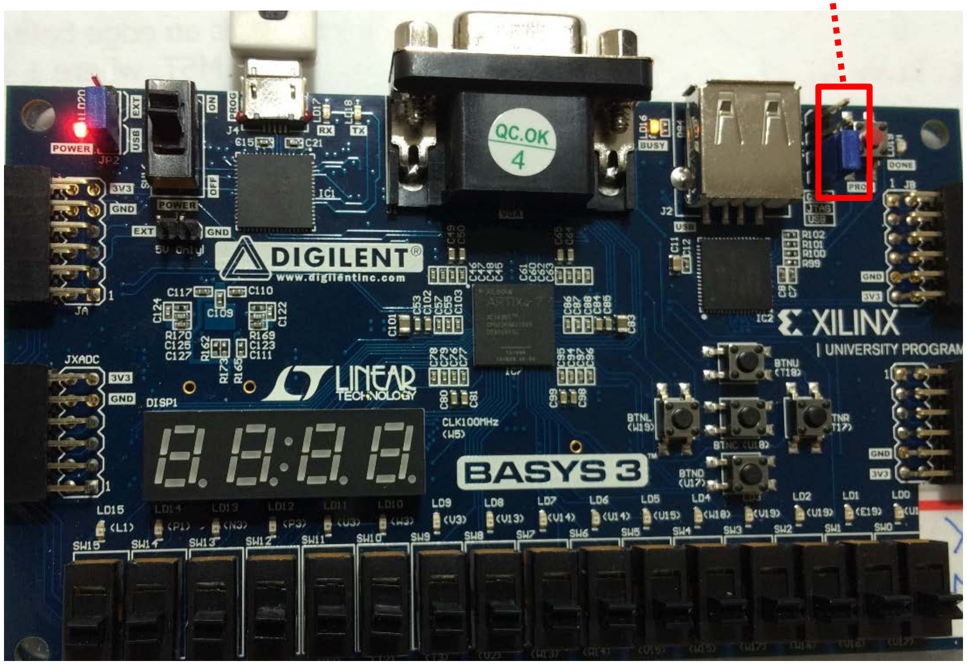




Introduction to Basys3 (2/3)

- Receive "Bitstream" from:
  - **Flash**
  - **JTAG** (recommend)
  - **USB Drive**

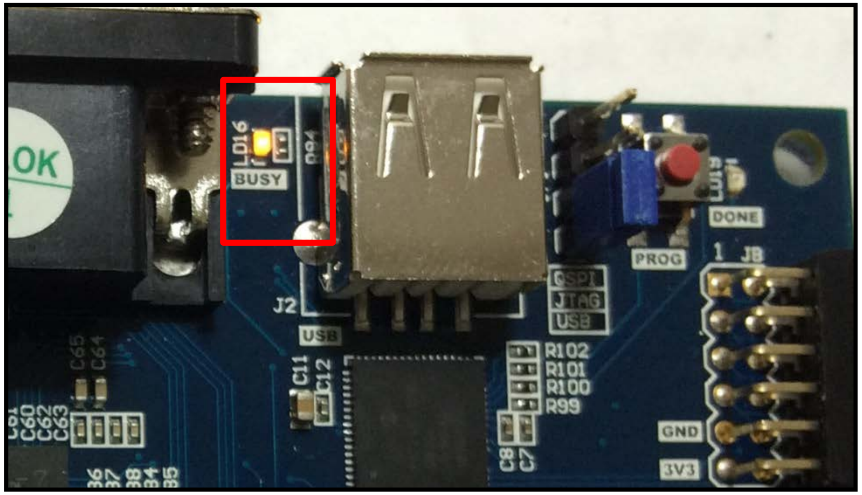


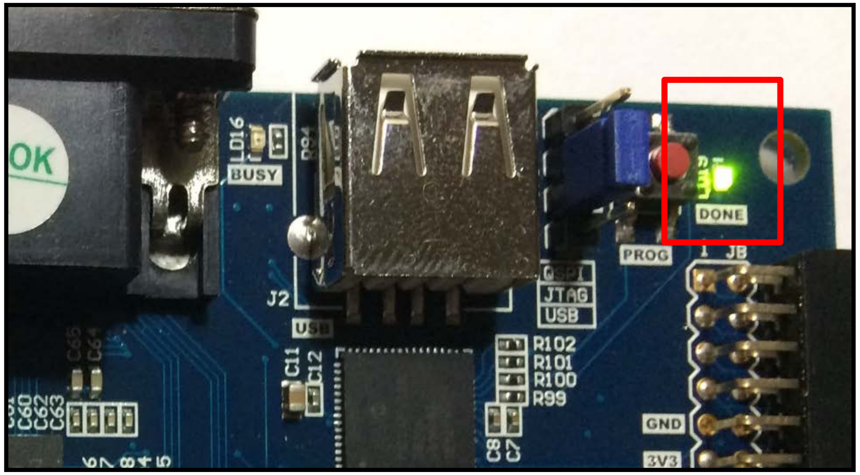


Introduction to Basys3 (3/3)

- Make sure that the programming is done by LED color

BUSY (yellow) ⟶ **DONE (green)**





Design Procedure

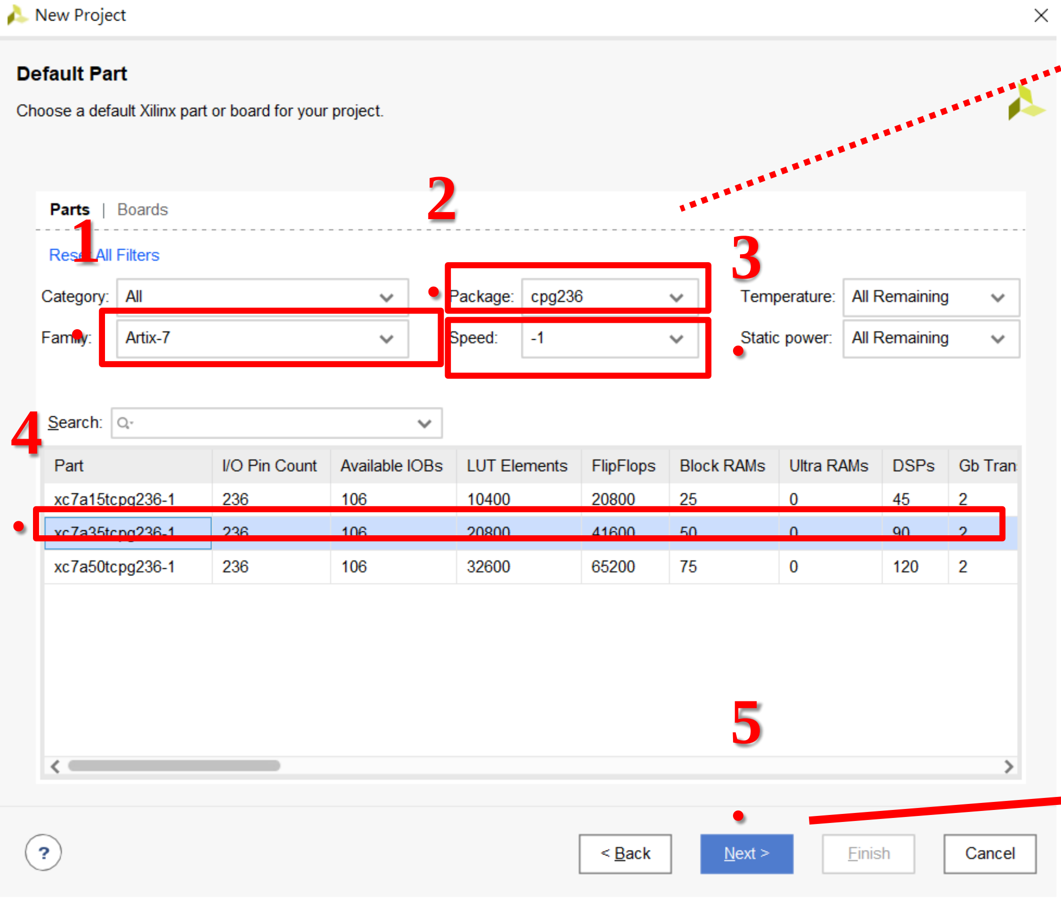
- Vivado and Basys3 Design procedure
  1) Create a Vivado project
  2) Simulate the design
  3) Synthesis the design
  4) Implement the design
  5) Perform timing simulation
  6) Verfy functionality in the hardware

Preparation_1: Demo Board (1/2)

- Basys3 Demo Board
- USB2microUSB power line

Preparation_1: Demo Board (2/2)

- Select Suitable Chip Type
- Find the information on your box



New Project

Default Part

Choose a default Xilinx part or board for your project.

Parts | Boards

Reset All Filters

Category: All

Family: Artix-7

Package: cpg236

Speed: -1

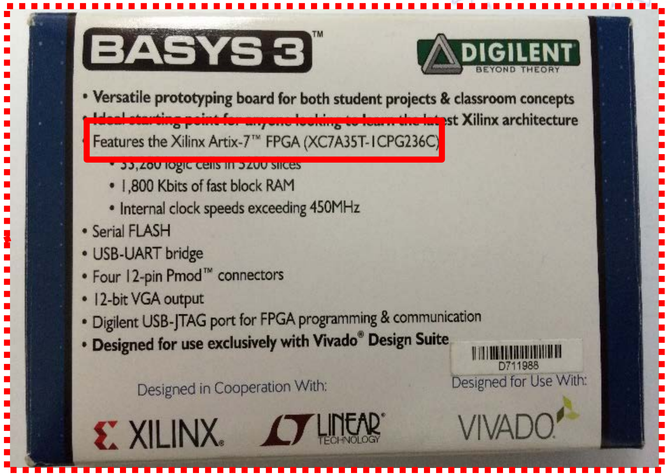
Temperature: All Remaining

Static power: All Remaining

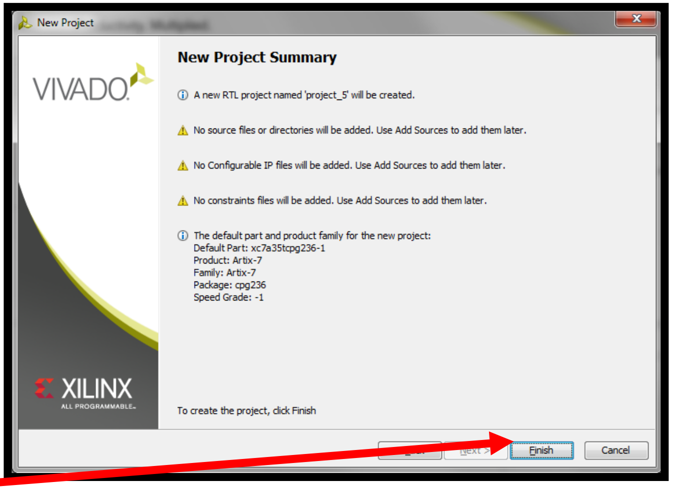
Search:

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb Tran
xc7a15tcpg236-1	236	106	10400	20800	25	0	45	2
xc7a35tcpg236-1	236	106	20800	41600	50	0	90	2
xc7a50tcpg236-1	236	106	32600	65200	75	0	120	2

1
2
3
4
5
?
< Back
Next >
Finish
Cancel



BASYS 3™
DIGILENT
BEYOND THEORY
• Versatile prototyping board for both student projects & classroom concepts
• Ideal starting point for anyone looking to learn the latest Xilinx architecture
• Features the Xilinx Artix-7™ FPGA (XC7A35T-1CPG236C)
• 33,280 logic cells in 5200 slices
• 1,800 Kbits of fast block RAM
• Internal clock speeds exceeding 450MHz
• Serial FLASH
• USB-UART bridge
• Four 12-pin Pmod™ connectors
• 12-bit VGA output
• Digilent USB-JTAG port for FPGA programming & communication
• Designed for use exclusively with Vivado® Design Suite
D711988
Designed in Cooperation With:
Designed for Use With:
XILINX
LINEAR TECHNOLOGY
VIVADO



New Project

New Project Summary

ⓘ A new RTL project named 'project_5' will be created.

⚠ No source files or directories will be added. Use Add Sources to add them later.

⚠ No Configurable IP files will be added. Use Add Sources to add them later.

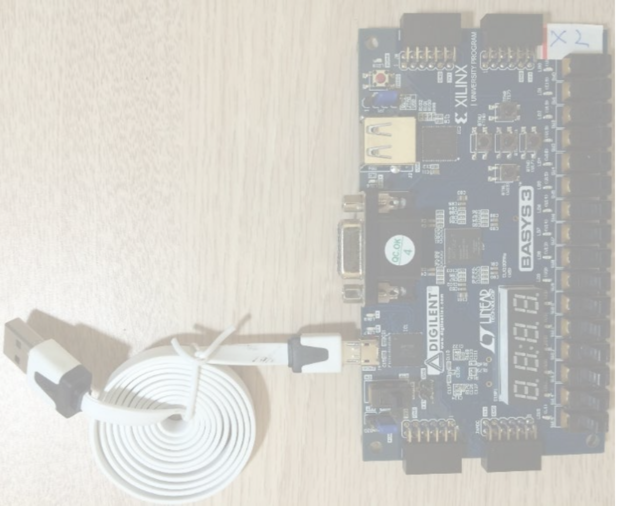
⚠ No constraints files will be added. Use Add Sources to add them later.

ⓘ The default part and product family for the new project:
Default Part: xc7a35tcpg236-1
Product: Artix-7
Family: Artix-7
Package: cpg236
Speed Grade: -1

To create the project, click Finish

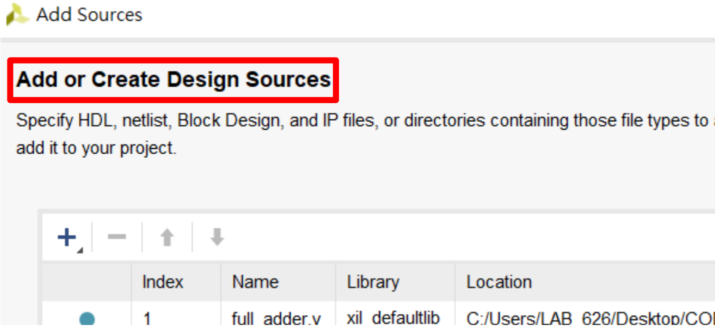
Preparation_2: Sources (1/8)

- Design file(s)
  - e.g. xxx.v
- “Xilinx Design Constraints” file
  - e.g. xxx_constraint.xdc



Preparation_2: Sources (2/8)

- Add Sources:
  - ➢ On "new project" stage

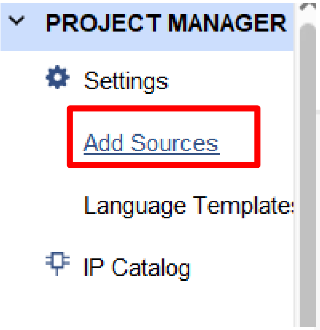


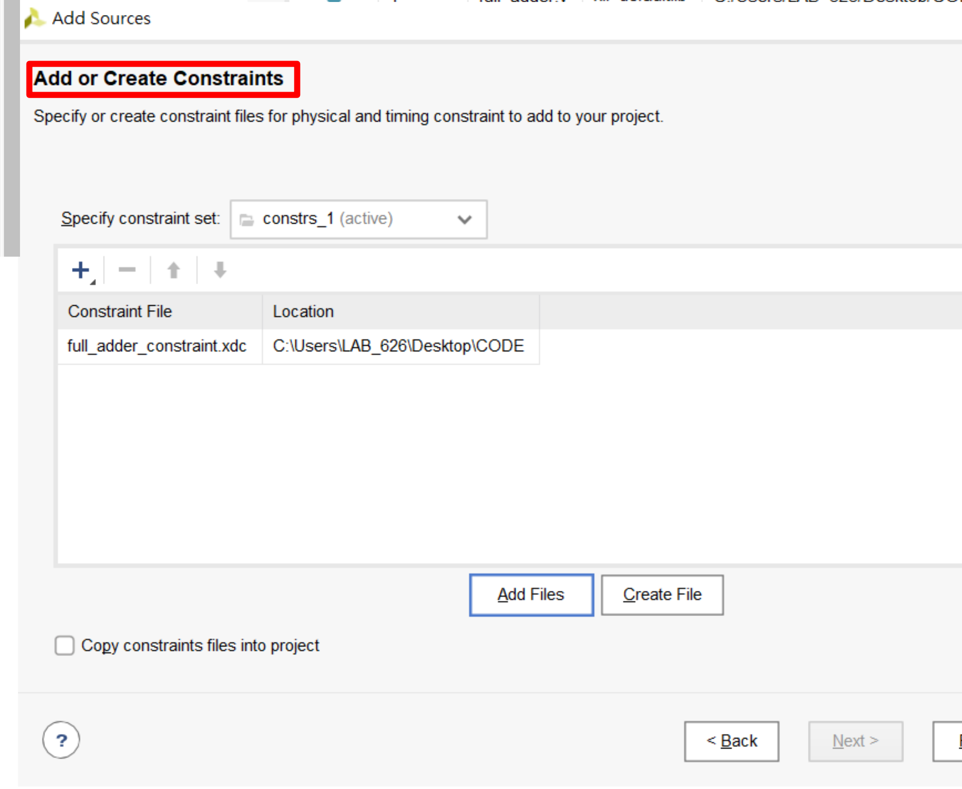
Add Sources

Add or Create Design Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add it to your project.

Index	Name	Library	Location
1	full_adder.v	xil_defaultlib	C:/Users/LAB_626/Desktop/CO





Add Sources

Add or Create Constraints

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

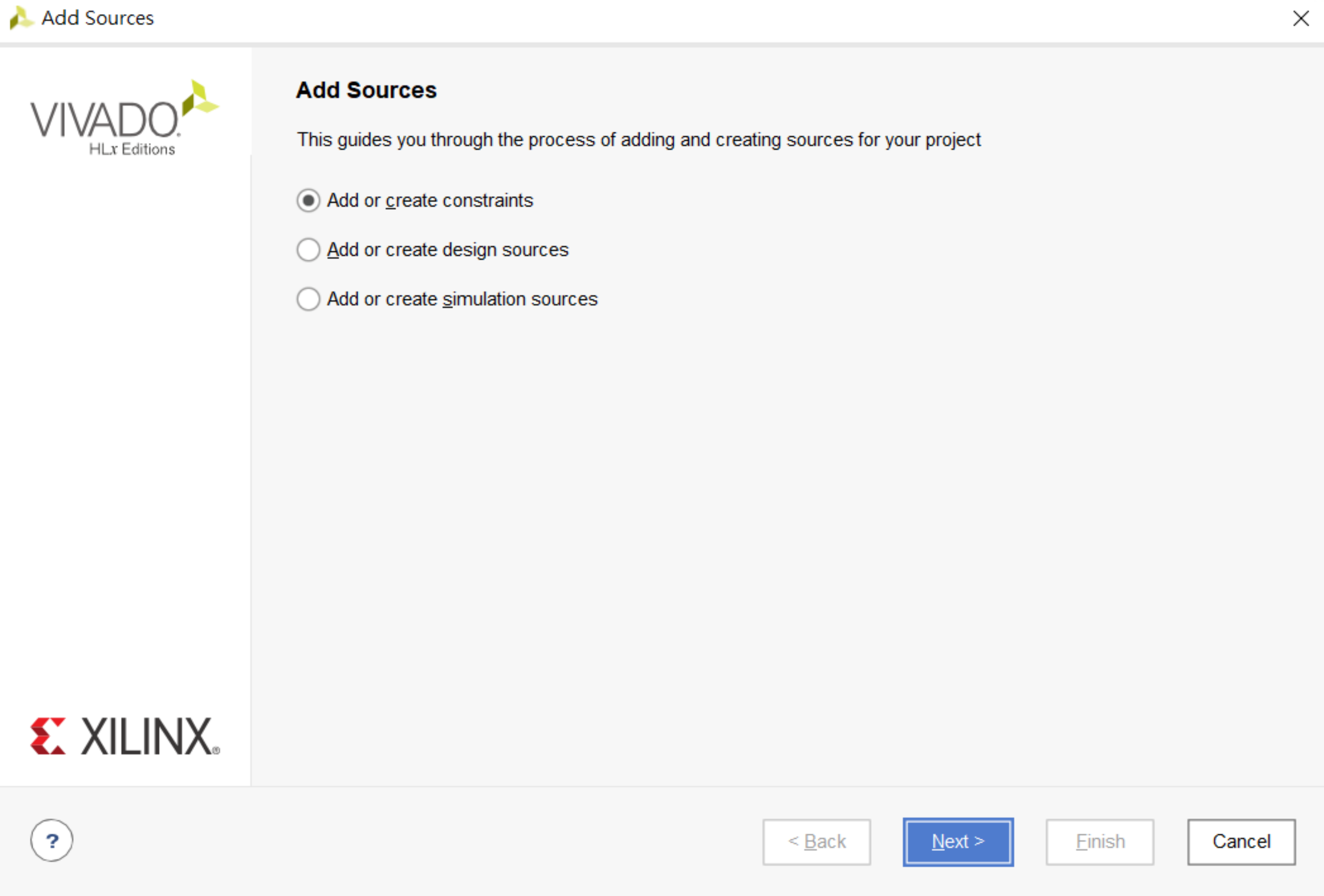
Specify constraint set: constrs_1 (active)

Constraint File	Location
full_adder_constraint.xdc	C:\Users\LAB_626\Desktop\CODE

□ Copy constraints files into project

  - ➢ On an "opened project"

Preparation_2: Sources (3/8)



Add Sources

Add Sources

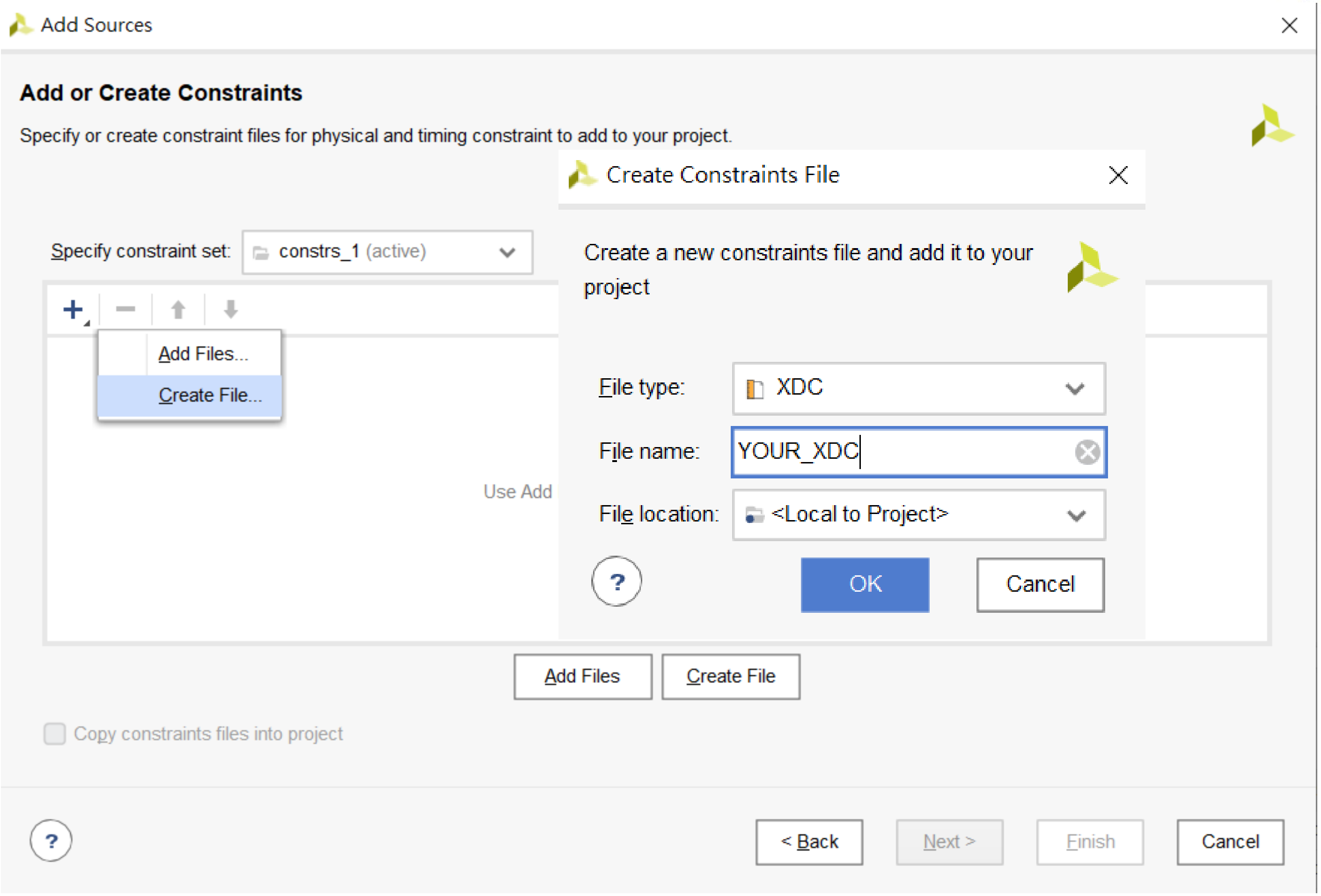
This guides you through the process of adding and creating sources for your project

● Add or create constraints

○ Add or create design sources

○ Add or create simulation sources

Preparation_2: Sources (4/8)



Add Sources

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)

□ Copy constraints files into project

Create Constraints File

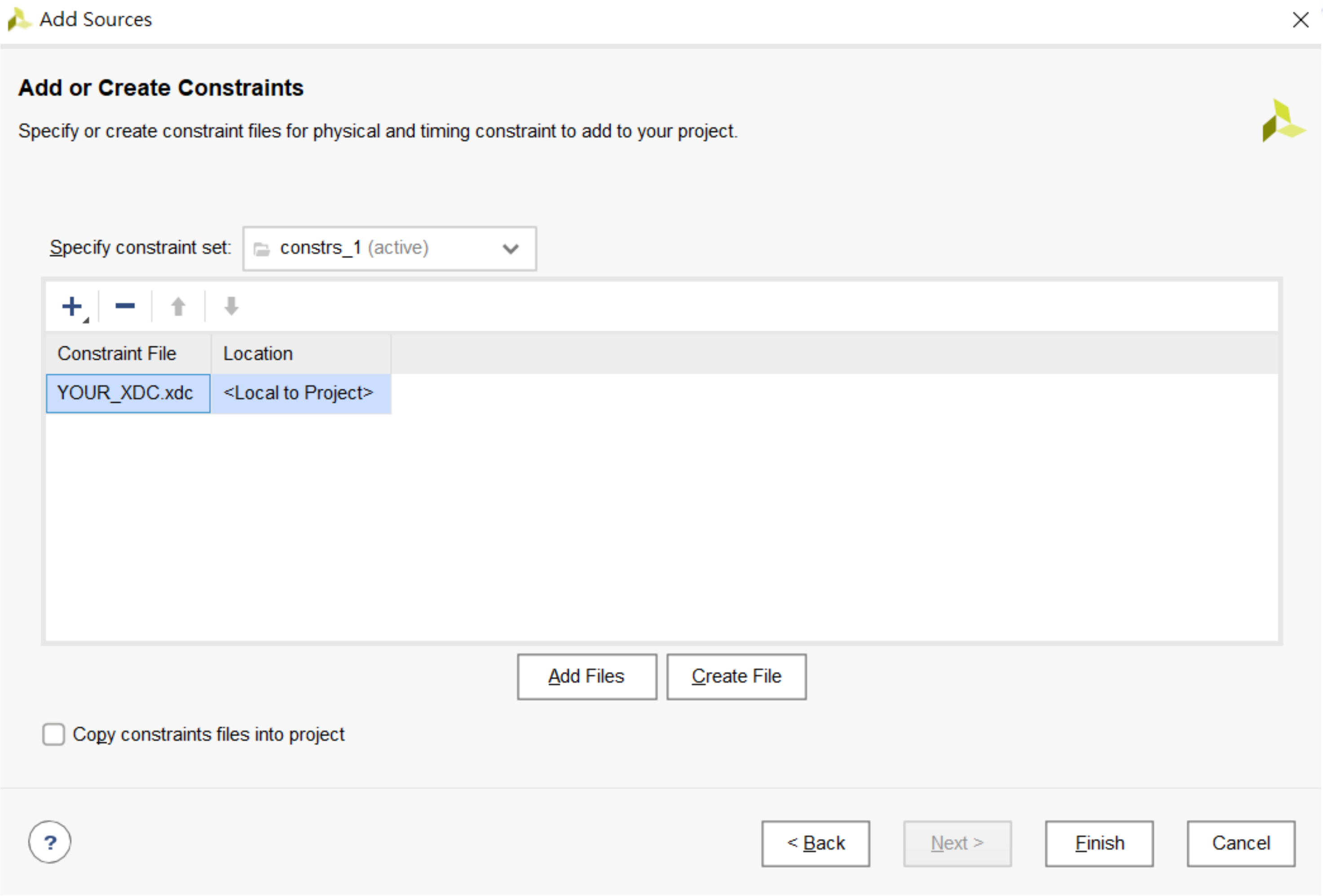
Create a new constraints file and add it to your project

File type: XDC

File name: YOUR_XDC

File location: <Local to Project>

Preparation_2: Sources (5/8)



Add Sources

Add or Create Constraints

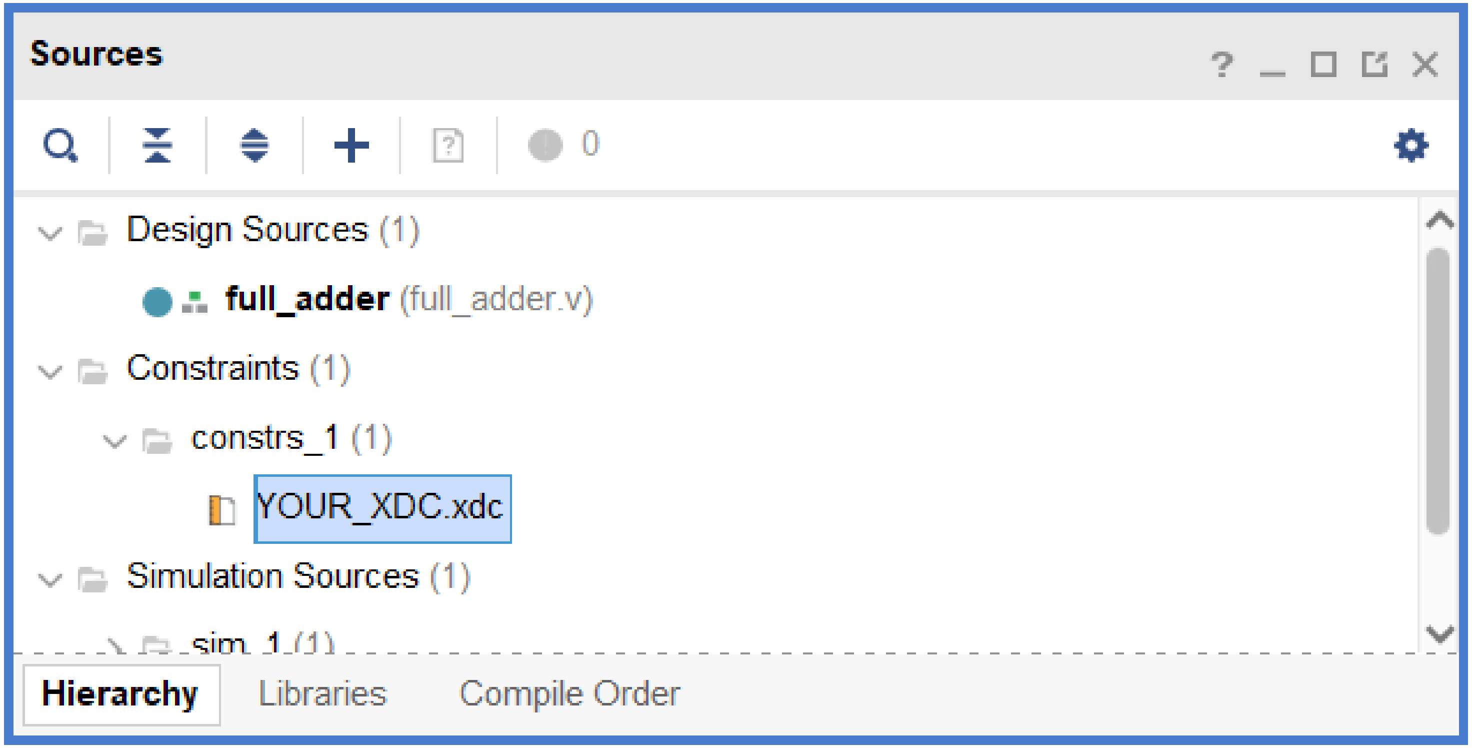
Specify or create constraint files for physical and timing constraint to add to your project.

Specify constraint set: constrs_1 (active)

Constraint File	Location
YOUR_XDC.xdc	<Local to Project>

□ Copy constraints files into project

Preparation_2: Sources (6/8)

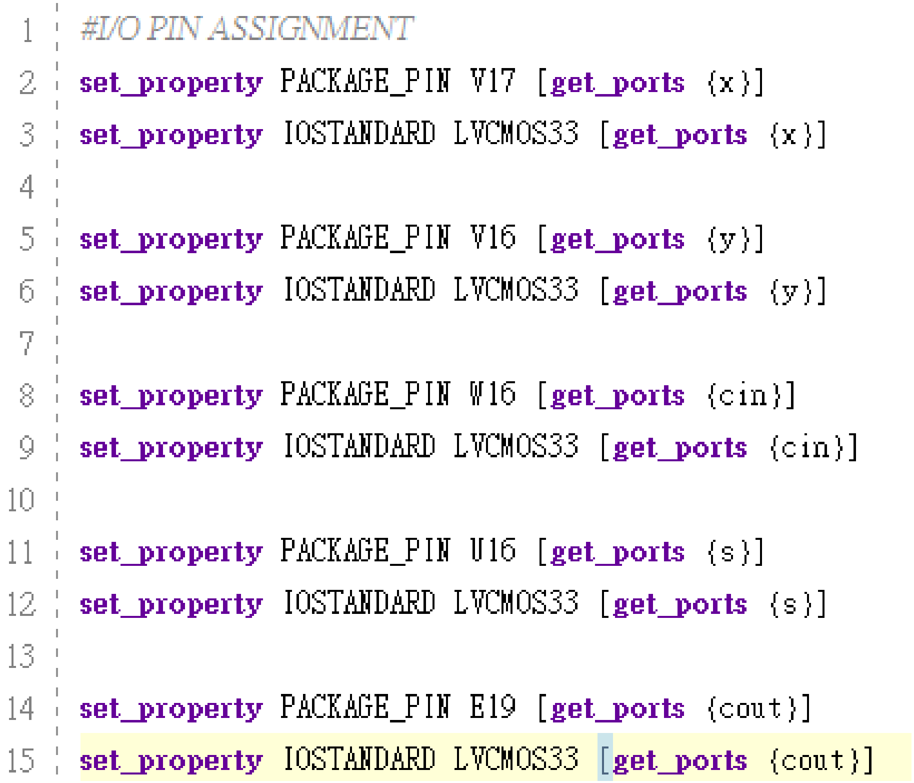


Sources

- Design Sources (1)
  - ● full_adder (full_adder.v)
- Constraints (1)
  - constrs_1 (1)
    - YOUR_XDC.xdc
- Simulation Sources (1)
  - sim_1 (1)

Preparation_2: Sources (7/8)

- Pin assignment for Lab1_1

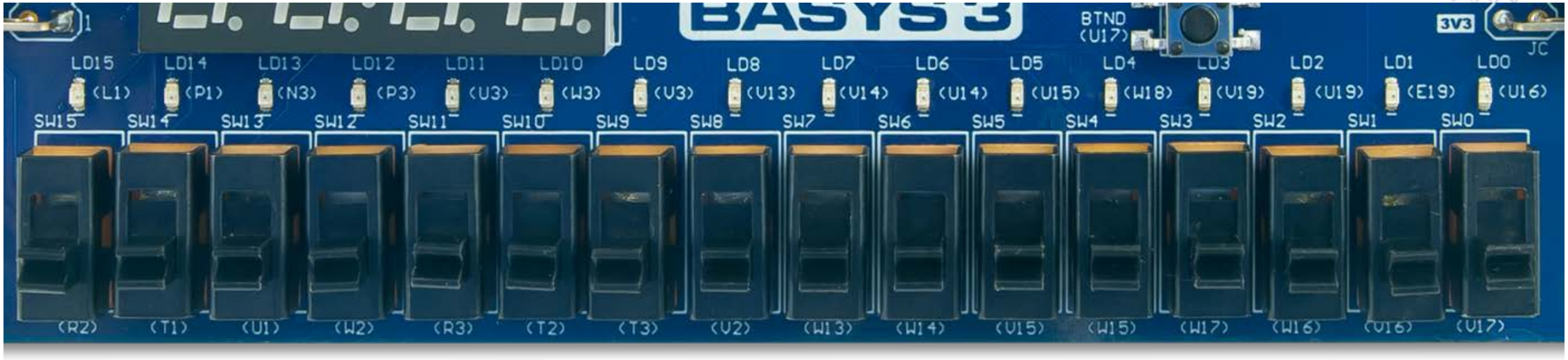


```
#I/O PIN ASSIGNMENT
set_property PACKAGE_PIN V17 [get_ports {x}]
set_property IOSTANDARD LVCMOS33 [get_ports {x}]

set_property PACKAGE_PIN V16 [get_ports {y}]
set_property IOSTANDARD LVCMOS33 [get_ports {y}]

set_property PACKAGE_PIN W16 [get_ports {cin}]
set_property IOSTANDARD LVCMOS33 [get_ports {cin}]

set_property PACKAGE_PIN U16 [get_ports {s}]
set_property IOSTANDARD LVCMOS33 [get_ports {s}]

set_property PACKAGE_PIN E19 [get_ports {cout}]
set_property IOSTANDARD LVCMOS33 [get_ports {cout}]
```

Constraints File (.xdc):

- TCL Language
- I/O pin assignment

Preparation_2: Sources (8/8)

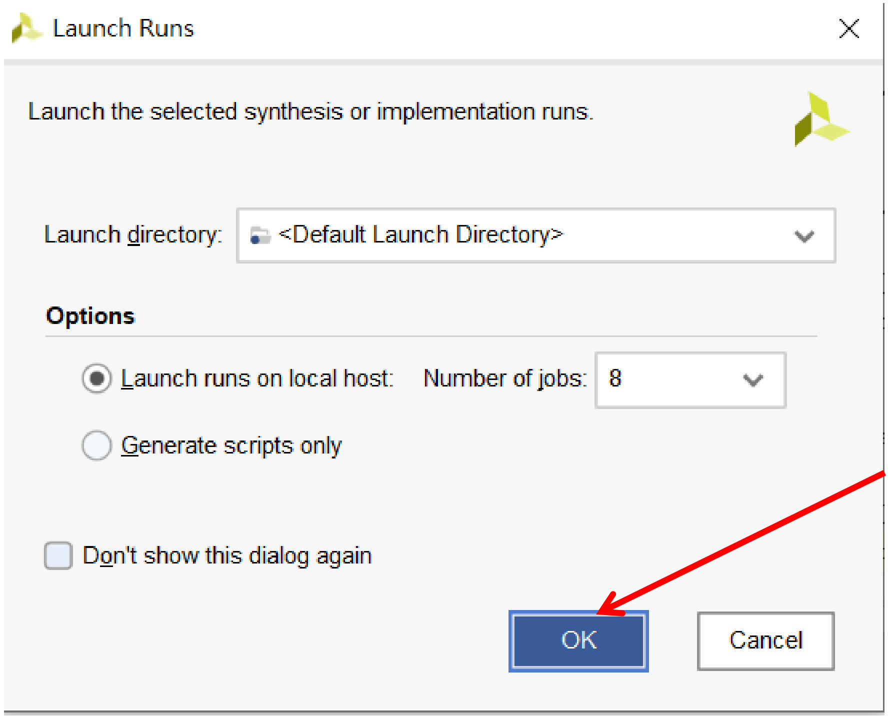


LD15	LD14	LD13	LD12	LD11	LD10	LD9	LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
L1	P1	N3	P3	U3	W3	V3	V13	V14	U14	U15	W18	V19	U19	E19	U16

SW15	SW14	SW13	SW12	SW11	SW10	SW9	SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
R2	T1	U1	W2	R3	T2	T3	V2	W13	W14	V15	W15	W17	W16	V16	V17

Synthesis (1/2)

- Confirm your sources
- Run Synthesis



Launch Runs

Launch the selected synthesis or implementation runs.

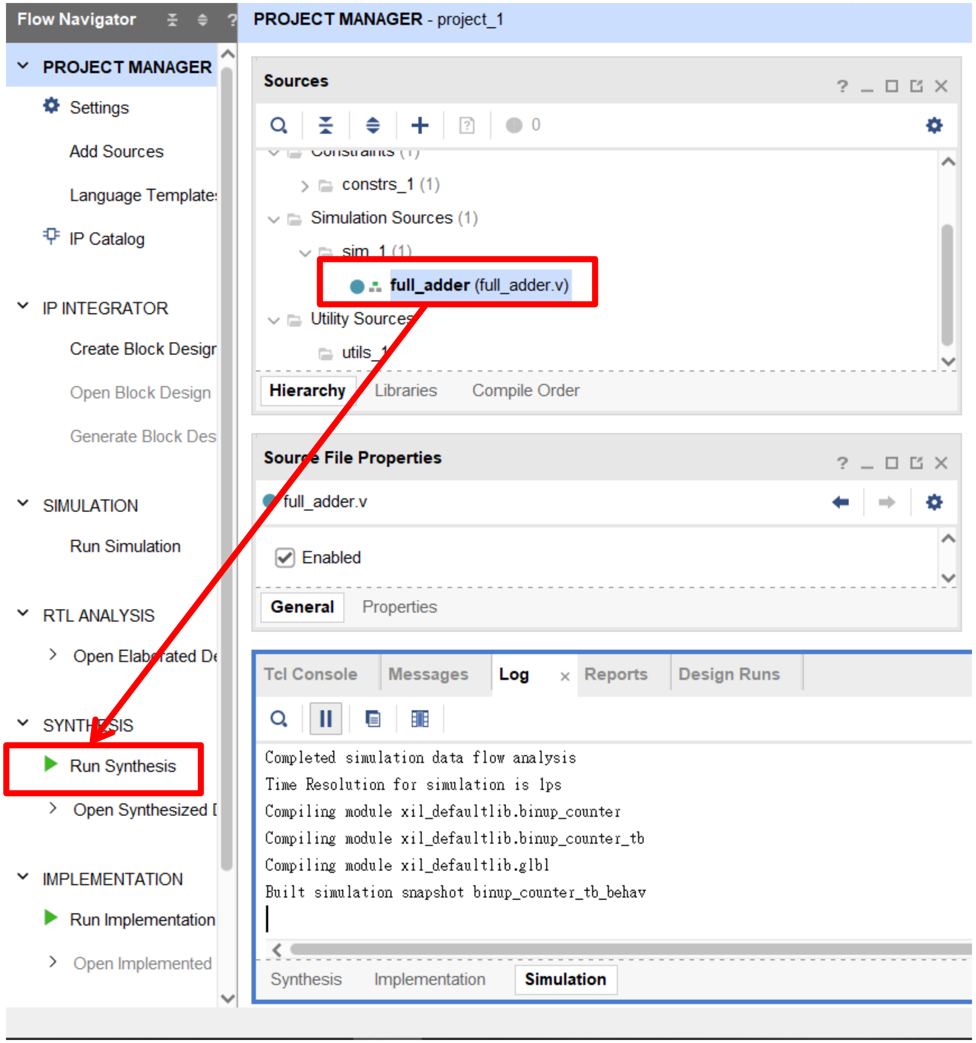
Launch directory: <Default Launch Directory>

Options

● Launch runs on local host: Number of jobs: 8

○ Generate scripts only

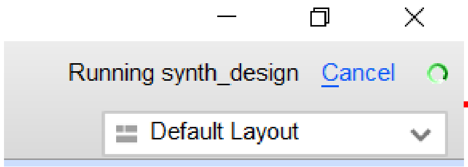
□ Don't show this dialog again



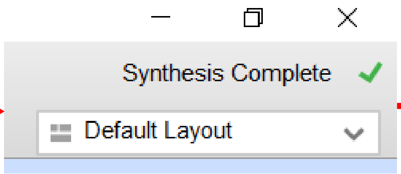
```
Completed simulation data flow analysis
Time Resolution for simulation is 1ps
Compiling module xil_defaultlib.binup_counter
Compiling module xil_defaultlib.binup_counter_tb
Compiling module xil_defaultlib.glbl
Built simulation snapshot binup_counter_tb_behav
```

Synthesis (2/2)

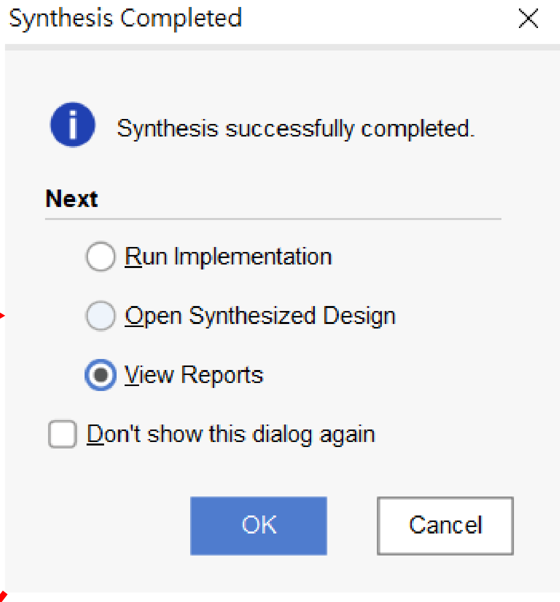
- Make sure that no Critical Warnings or Errors



Running synth_design Cancel



Synthesis Complete ✓



Synthesis Completed

Synthesis successfully completed.

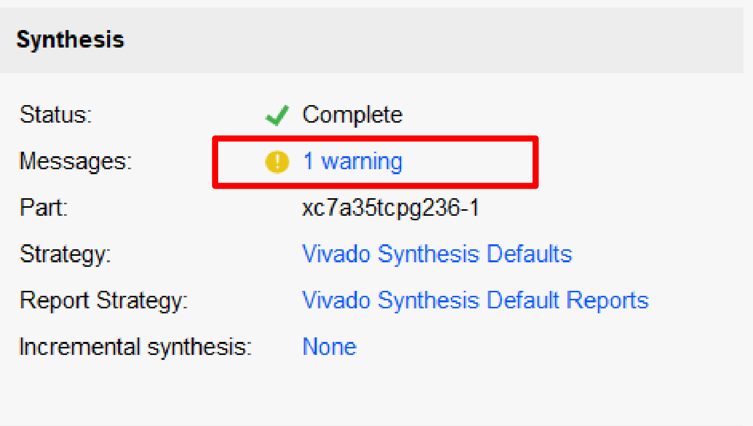
Next

○ Run Implementation

○ Open Synthesized Design

● View Reports

□ Don't show this dialog again

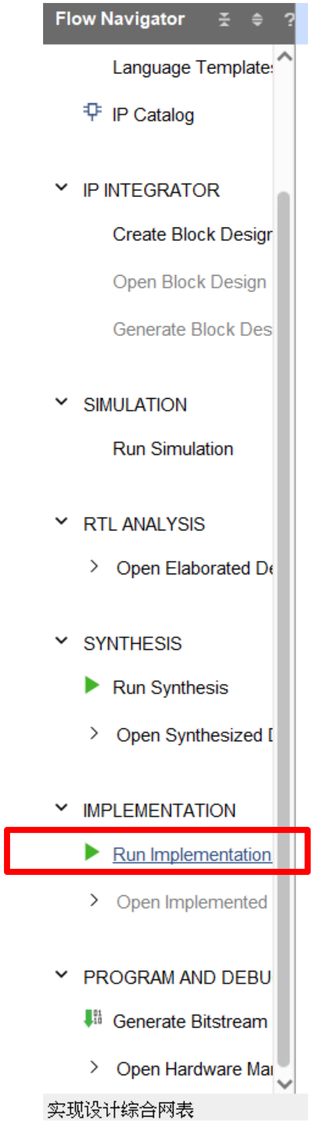


Synthesis

Status:	✓ Complete
Messages:	1 warning
Part:	xc7a35tcpg236-1
Strategy:	Vivado Synthesis Defaults
Report Strategy:	Vivado Synthesis Default Reports
Incremental synthesis:	None

Implementation (1/2)

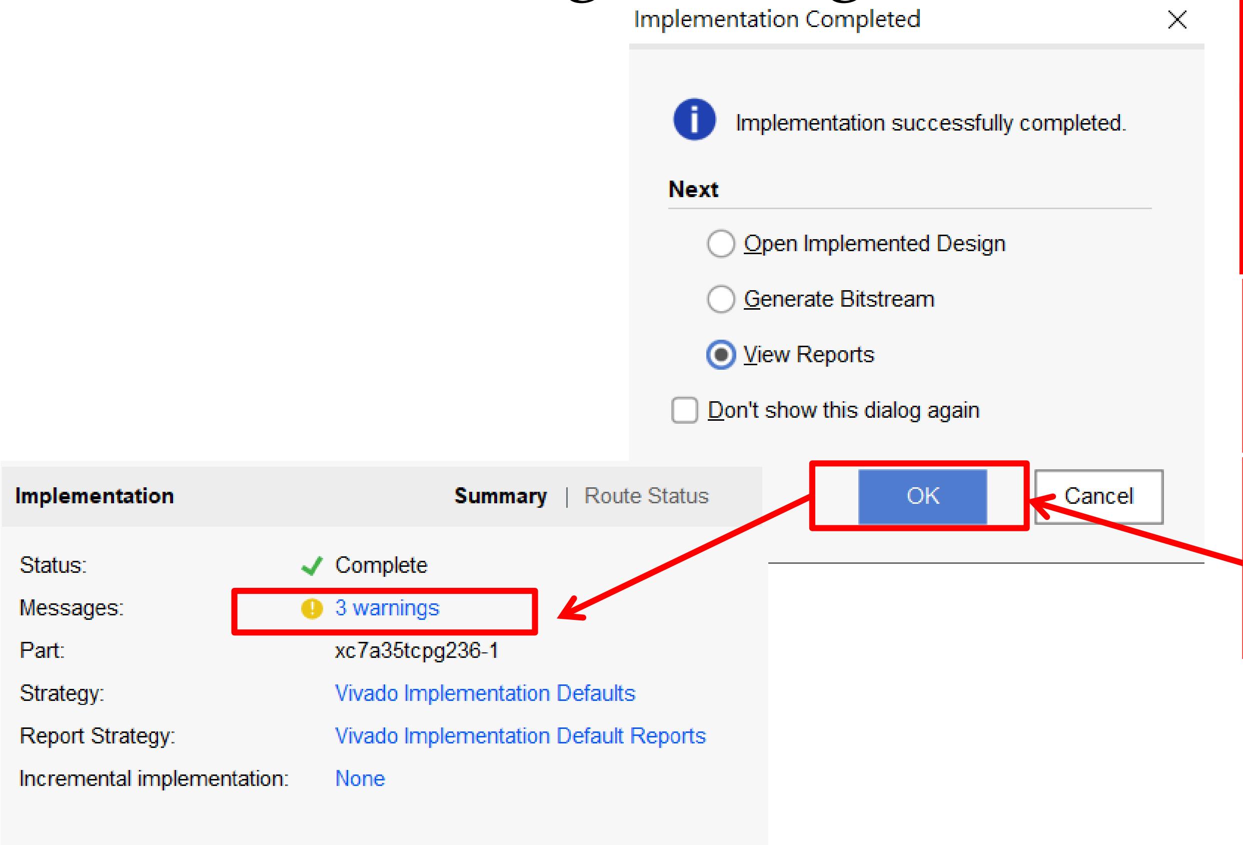
- Run Implementation



实现设计综合网表

Implementation (2/2)

- Try to understand the error/warning messages



Implementation Completed

Implementation successfully completed.

Next

○ Open Implemented Design

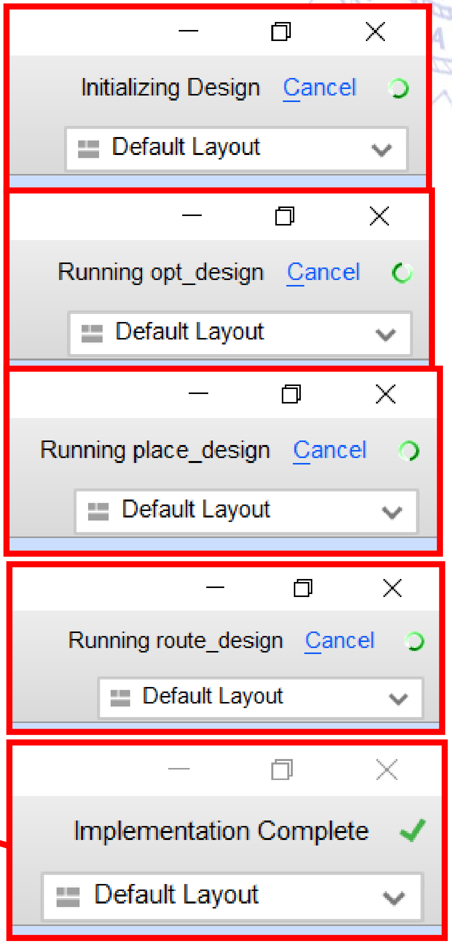
○ Generate Bitstream

● View Reports

□ Don't show this dialog again

Implementation

Status:	✓ Complete
Messages:	! 3 warnings
Part:	xc7a35tcpg236-1
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None



Initializing Design Cancel

Running opt_design Cancel

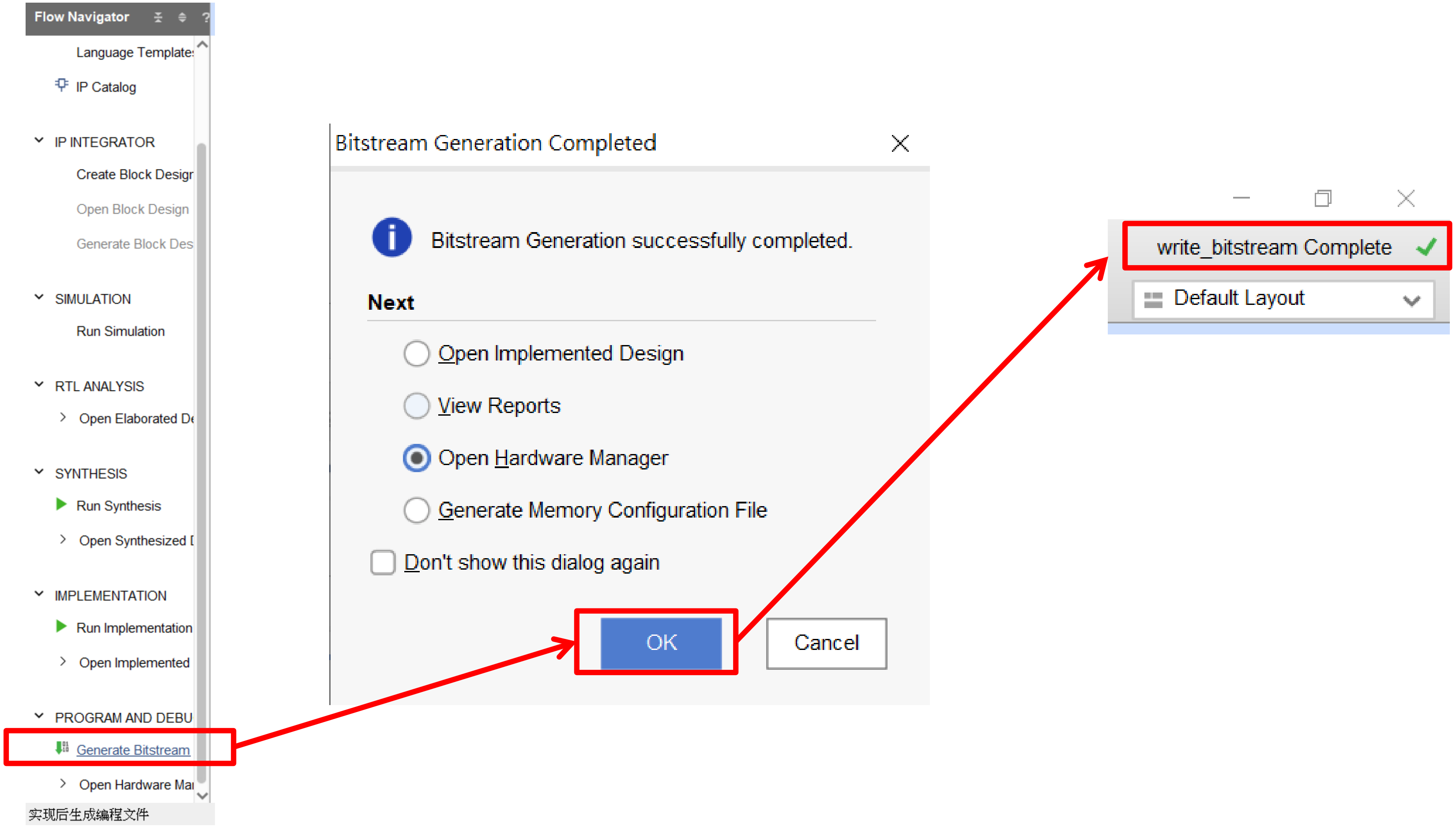
Running place_design Cancel

Running route_design Cancel

Implementation Complete ✓

Generate Bitstream

- Generating bitstream



Bitstream Generation Completed

Bitstream Generation successfully completed.

Next

○ Open Implemented Design

○ View Reports

● Open Hardware Manager

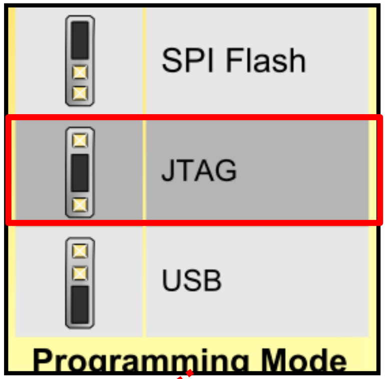
○ Generate Memory Configuration File

□ Don't show this dialog again

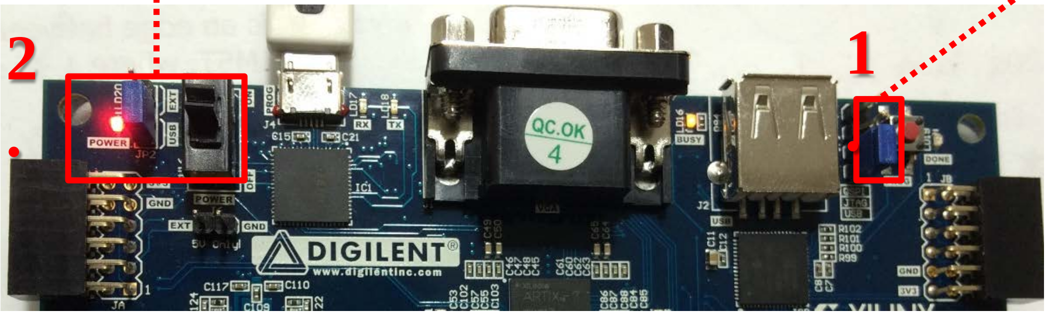
write_bitstream Complete ✓

Open Target (1/2)

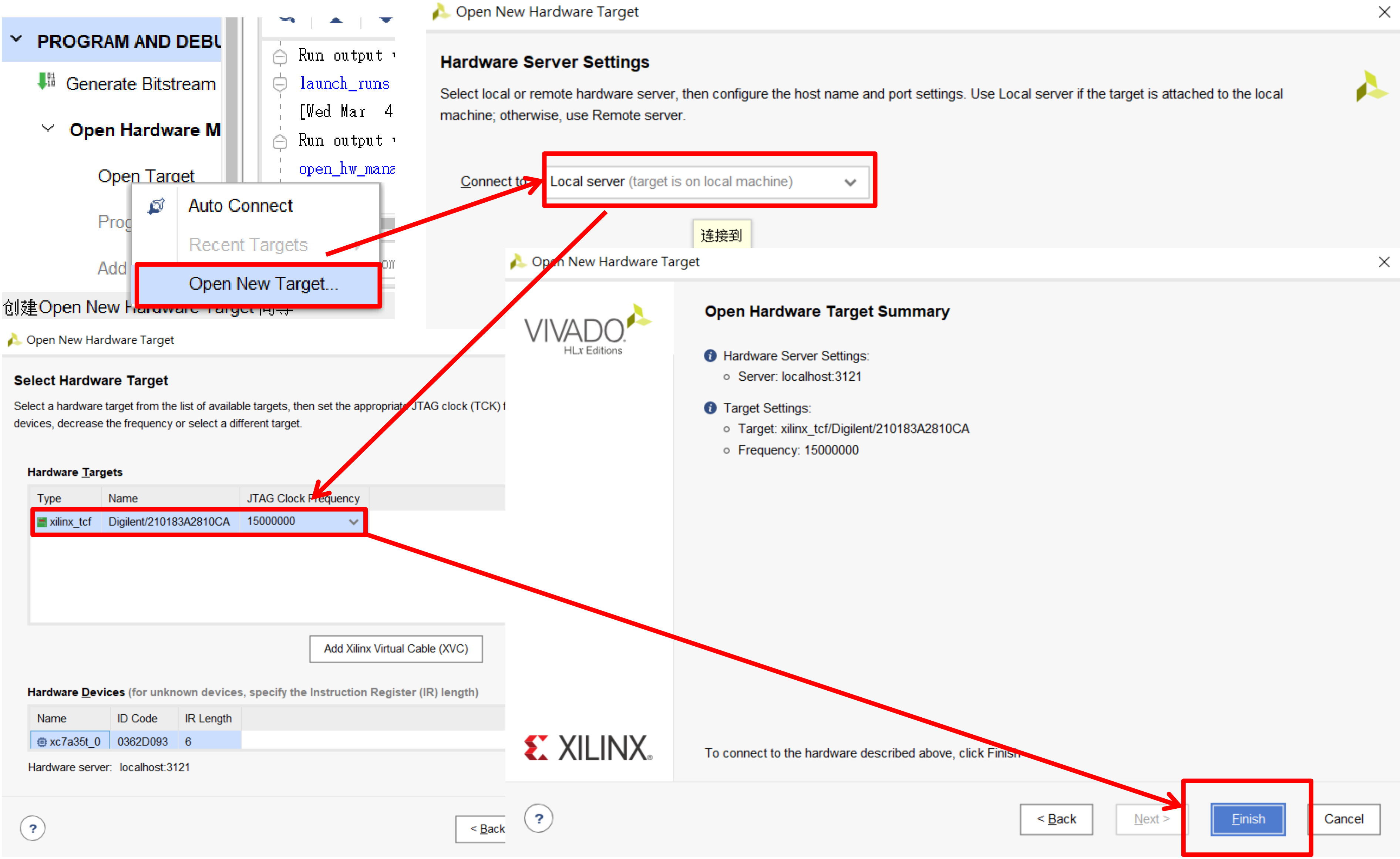
Step 1:
Change blue jumper to [**JTAG**] mode



Step 2:
Turn the power switch to [**ON**]



Open Target (2/2)



Open New Hardware Target

Hardware Server Settings

Select local or remote hardware server, then configure the host name and port settings. Use Local server if the target is attached to the local machine; otherwise, use Remote server.

Connect to: Local server (target is on local machine)

Open New Hardware Target

Select Hardware Target

Select a hardware target from the list of available targets, then set the appropriate JTAG clock (TCK) f devices, decrease the frequency or select a different target.

Hardware Targets

Type	Name	JTAG Clock Frequency
xilinx_tcf	Digilent/210183A2810CA	15000000

Hardware Devices (for unknown devices, specify the Instruction Register (IR) length)

Name	ID Code	IR Length
xc7a35t_0	0362D093	6

Hardware server: localhost:3121

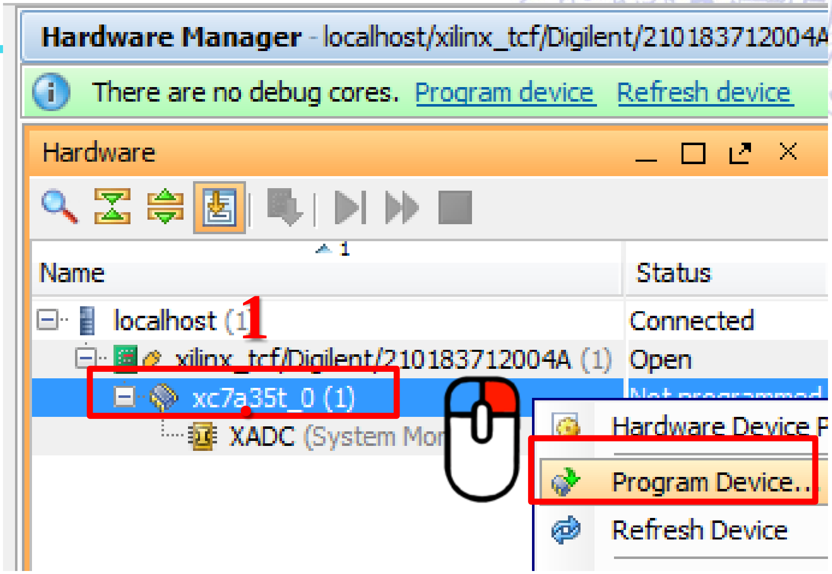
Open New Hardware Target

Open Hardware Target Summary

- Hardware Server Settings:
  - Server: localhost:3121
- Target Settings:
  - Target: xilinx_tcf/Digilent/210183A2810CA
  - Frequency: 15000000

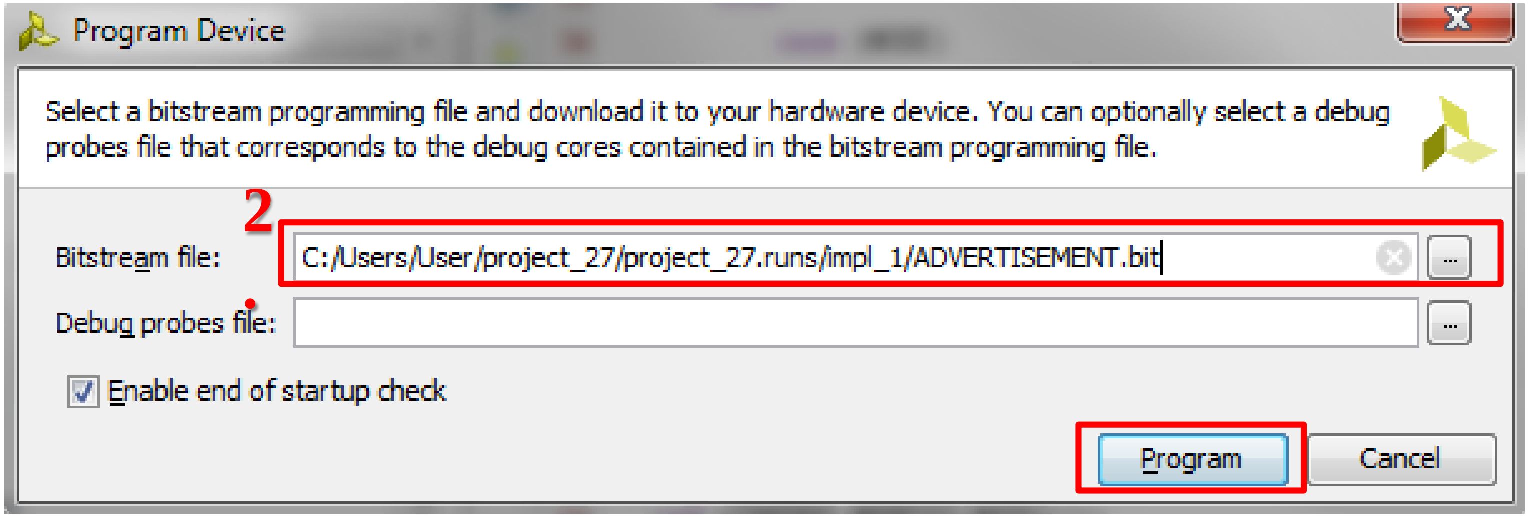
To connect to the hardware described above, click Finish

Program Device



Hardware Manager - localhost/xilinx_tcf/Digilent/210183712004A

There are no debug cores. Program device Refresh device



Program Device

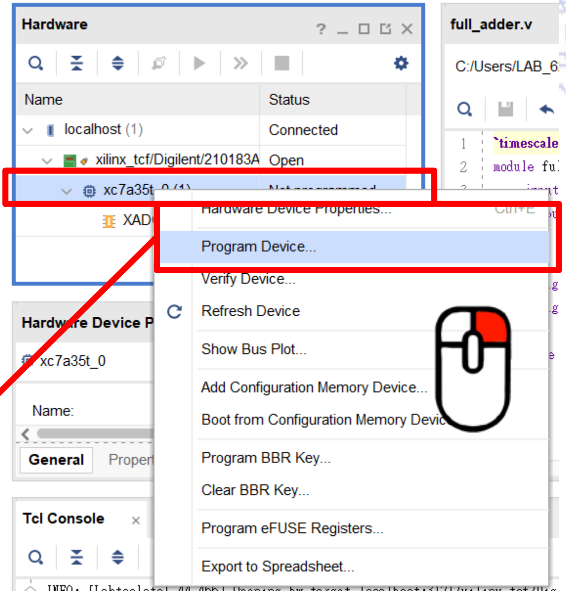
Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: C:/Users/User/project_27/project_27.runs/impl_1/ADVERTISEMENT.bit

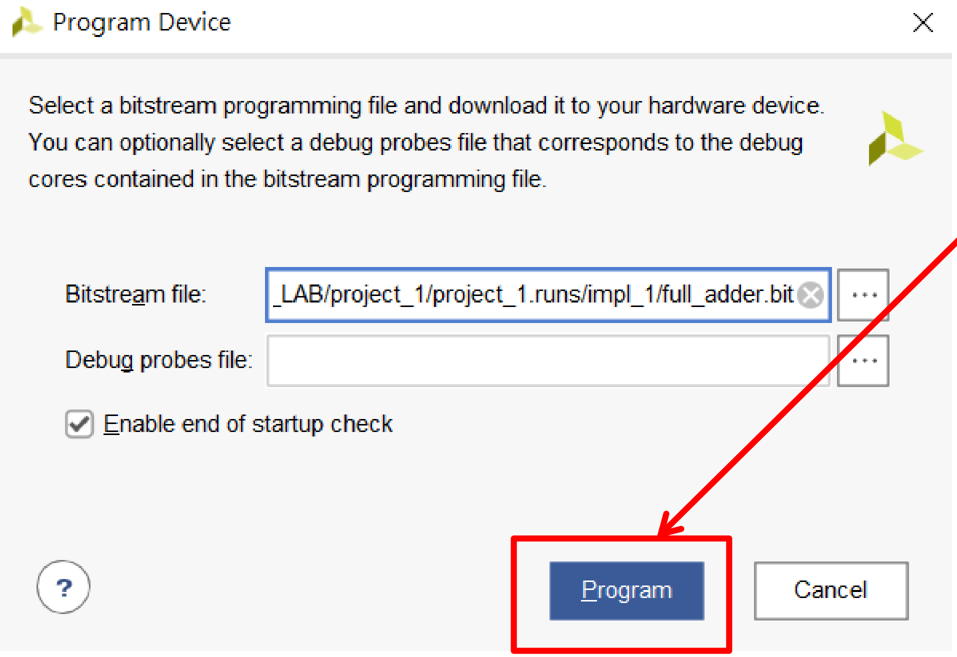
Debug probes file:

☑ Enable end of startup check

Program Device



Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210183A	Open
xc7a35t_0 (1)	Not programmed



Program Device

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

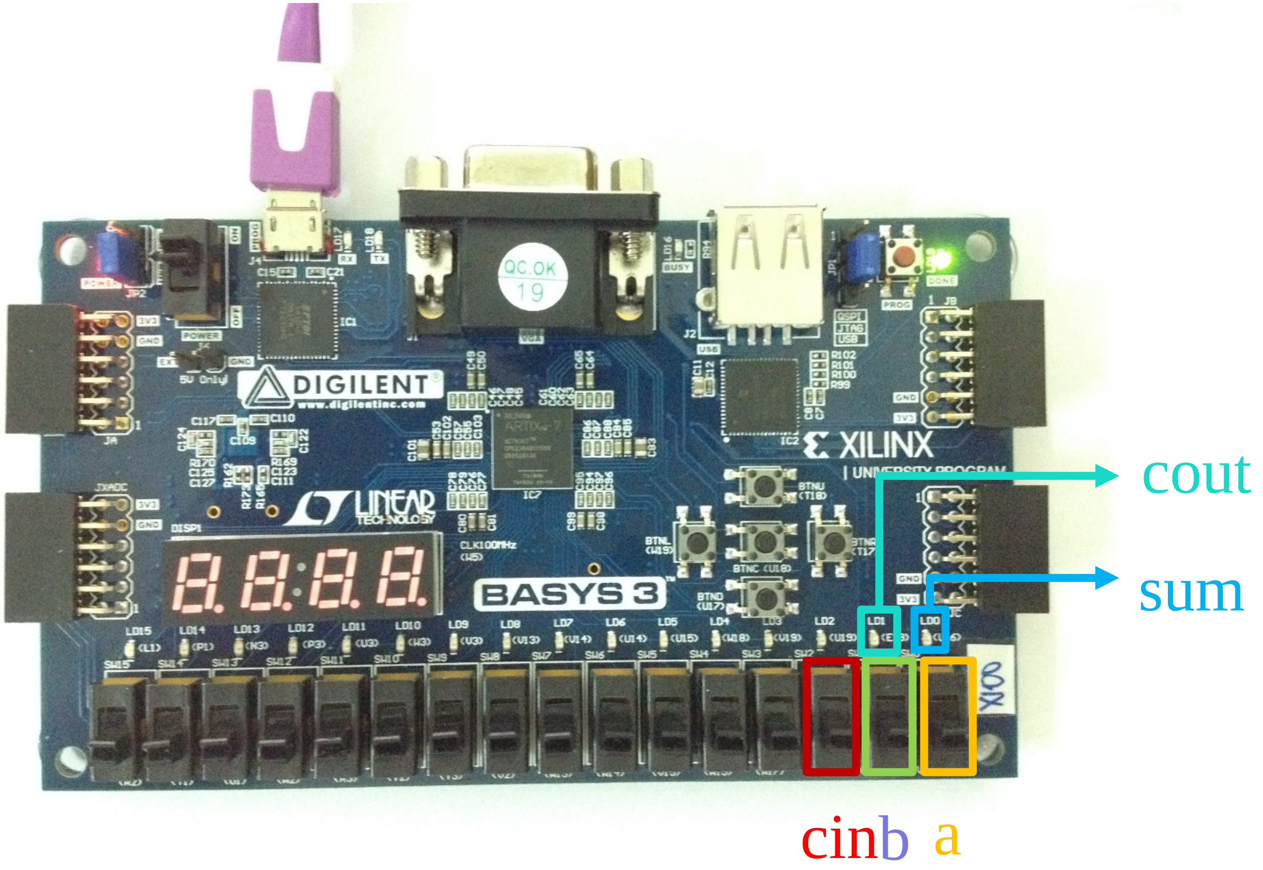
Bitstream file: _LAB/project_1/project_1.runs/impl_1/full_adder.bit

Debug probes file:

☑ Enable end of startup check

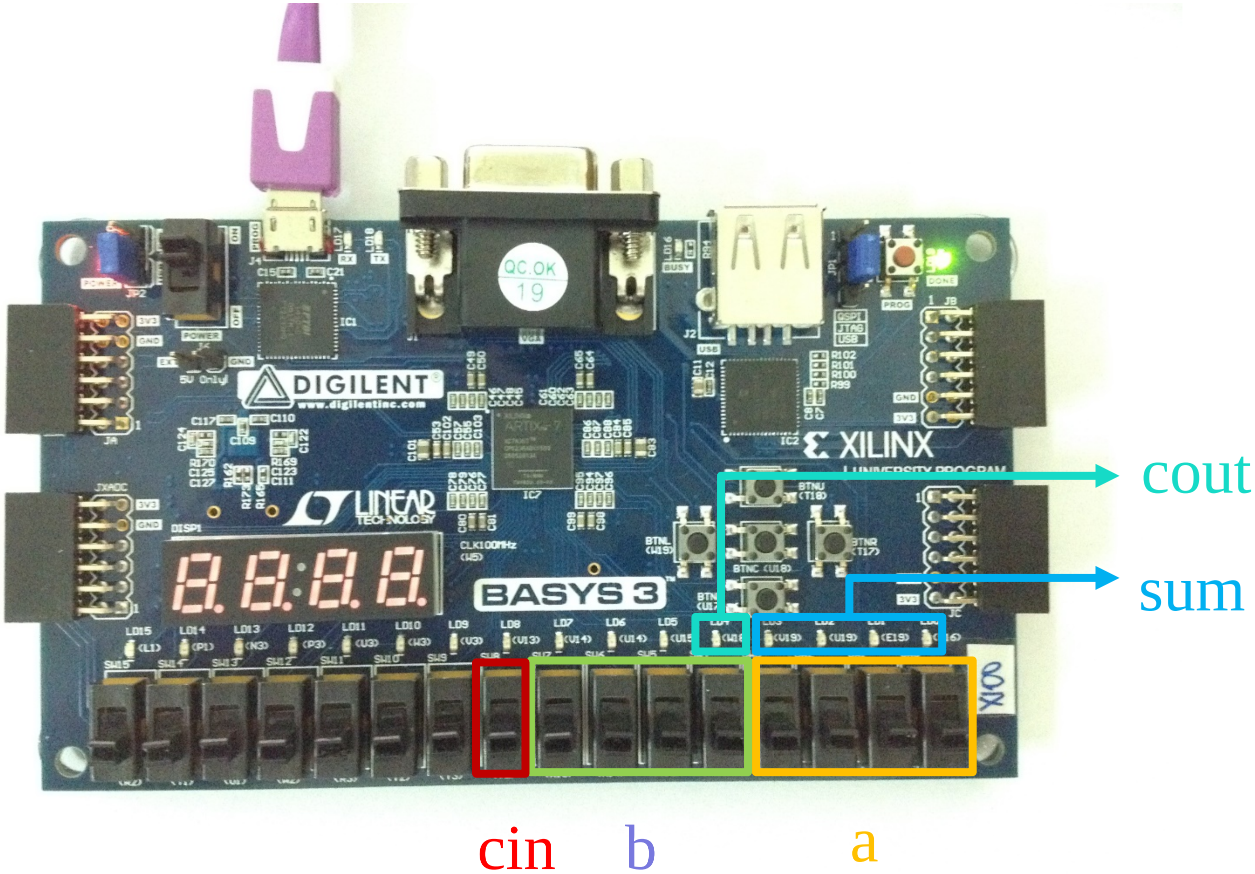
Demo Lab1_1

a = 1’b0
b = 1’b0
cin = 1’b0
sum = 1’b0
cout = 1’b0



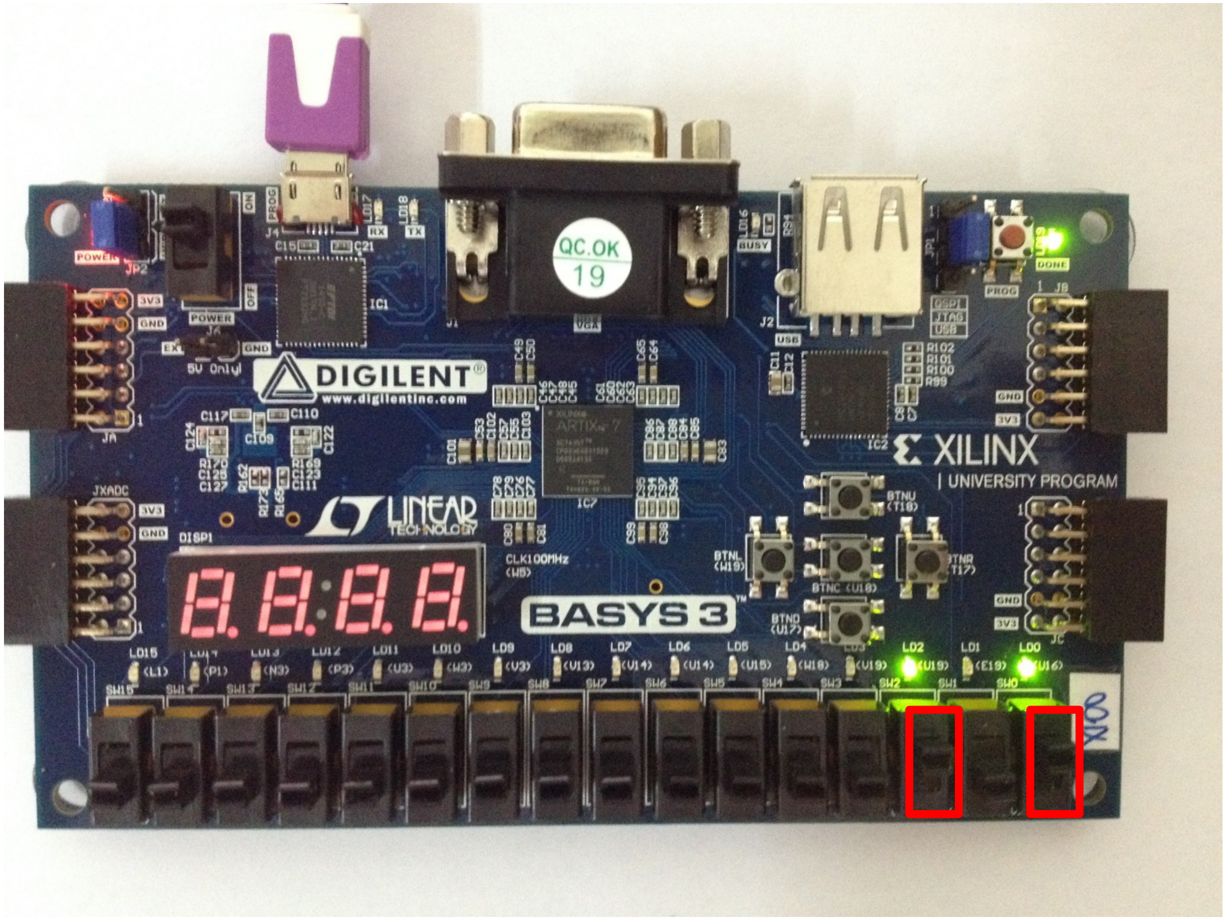
Practice Lab1_4 (1/6)

a = 4’b0000
b = 4’b0000
cin = 1’b0
sum = 4’b0000
cout = 1’b0



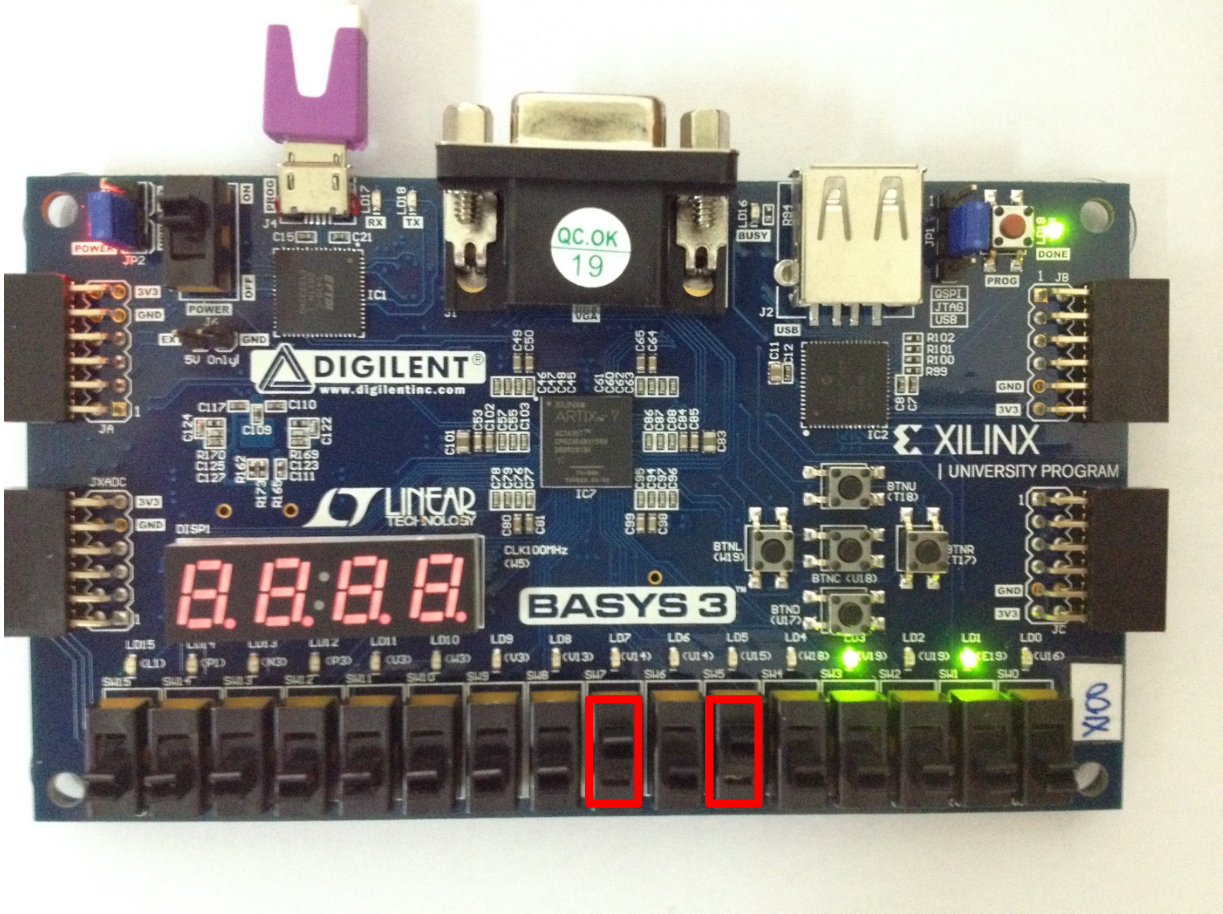
Practice Lab1_4 (2/6)

a   = 4’b0101
b   = 4’b0000
cin   = 1’b0
sum = 4’b0101
cout = 1’b0



Practice Lab1_4 (3/6)

a	= 4’b0000
b	= 4’b1010
cin	= 1’b0
sum	= 4’b1010
cout	= 1’b0



Practice Lab1_4 (4/6)

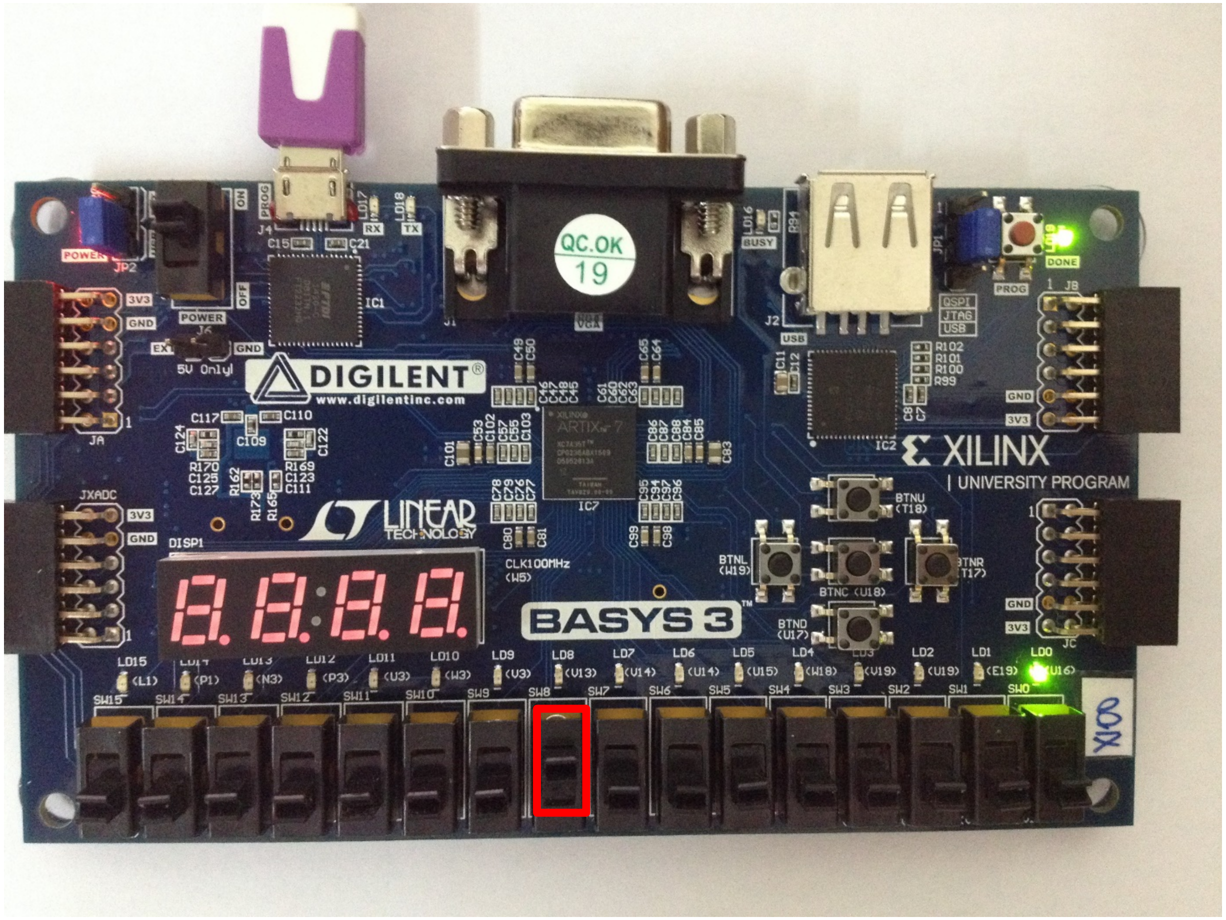
a = 4’b0000

b = 4’b0000

cin = 1’b1

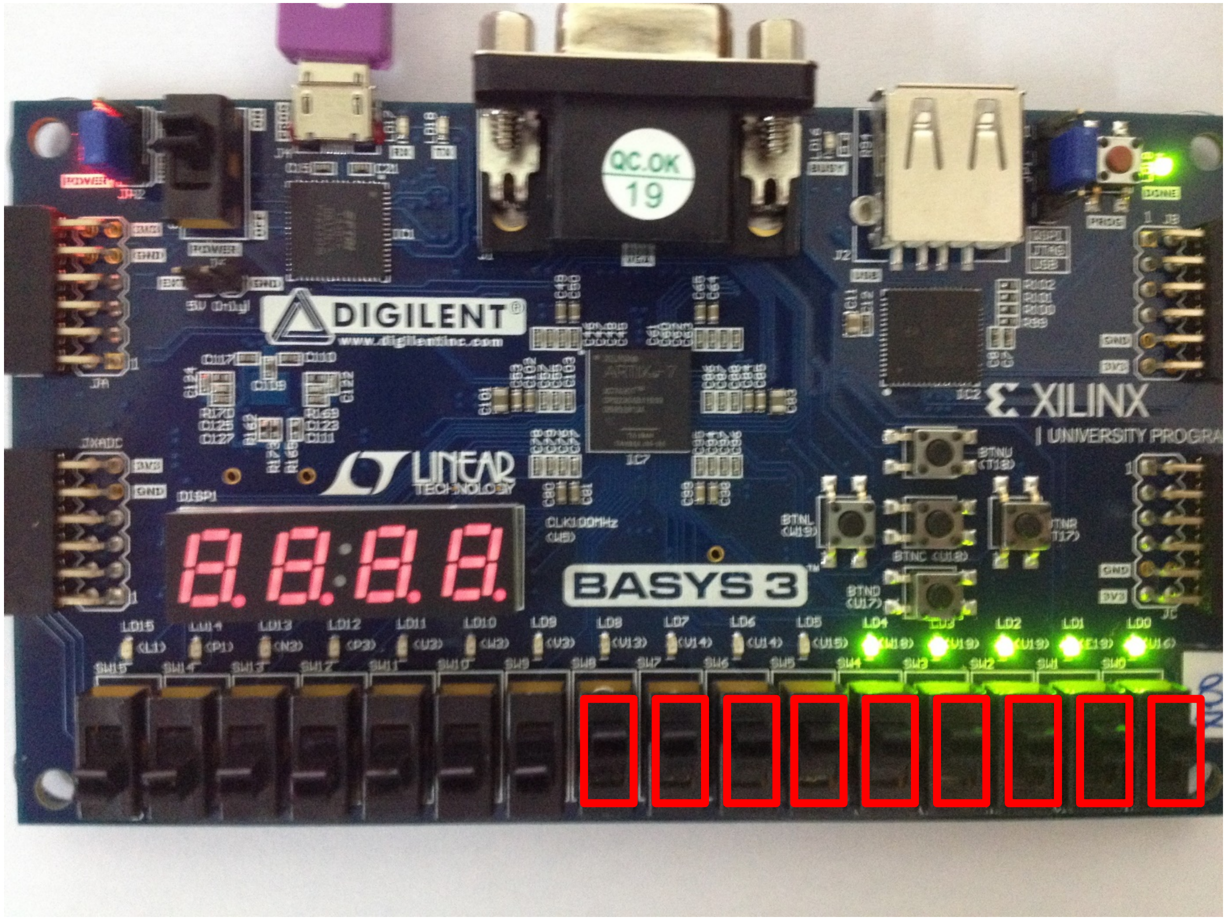
sum = 4’b0000

cout = 1’b1



Practice Lab1_4 (5/6)

a = 4’b1111
b = 4’b1111
cin = 1’b1
sum = 4’b1111
cout = 1’b1



Practice Lab1_4 (6/6)

a = 4’b0011
b = 4’b0011
cin = 1’b1
sum = 4’b0111
cout = 1’b0