

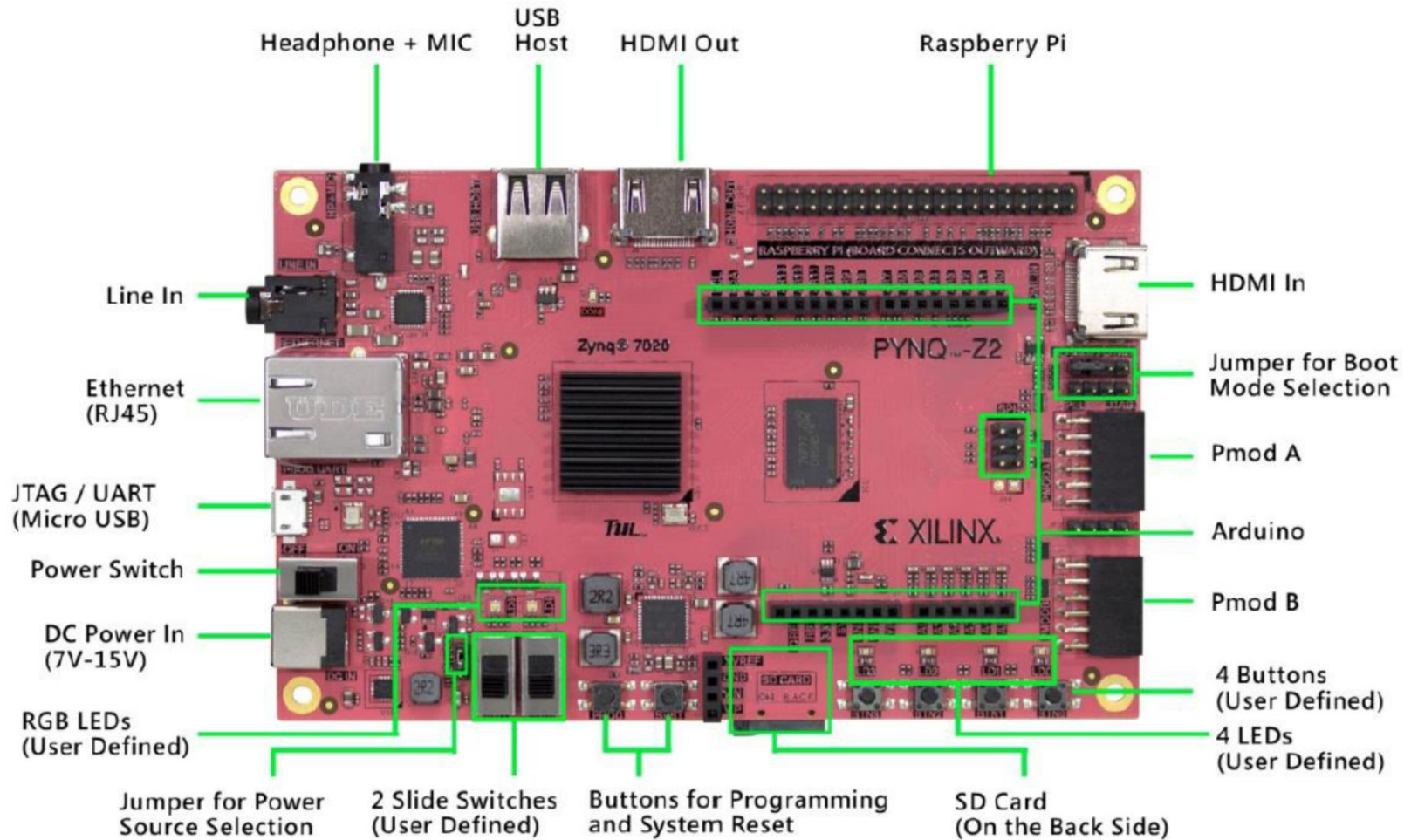
# VLSI for DSP

**Yuan-Ho Chen**

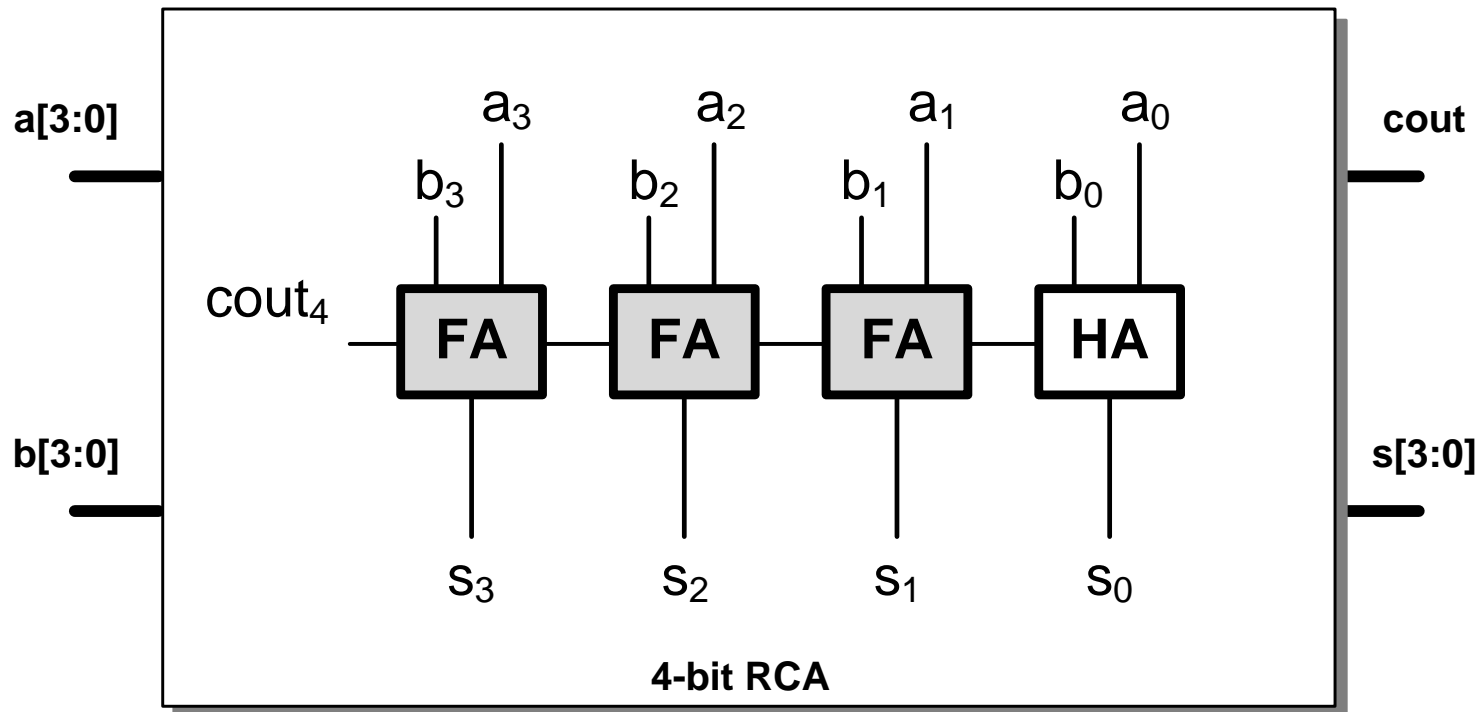
**Department of Electronic Engineering  
Chang Guan University  
[chenyh@mail.cgu.edu.tw](mailto:chenyh@mail.cgu.edu.tw)**



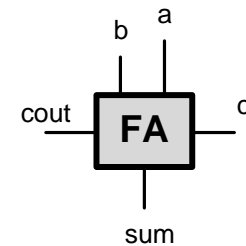
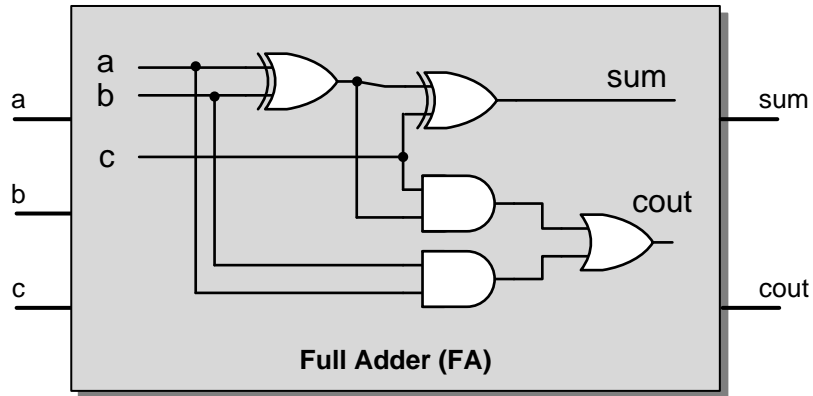
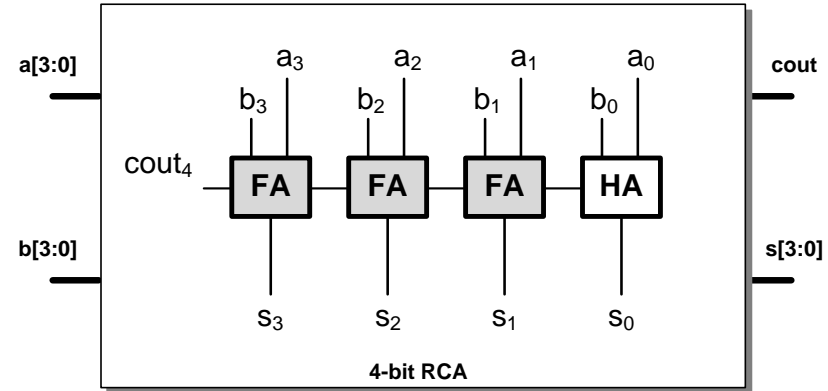
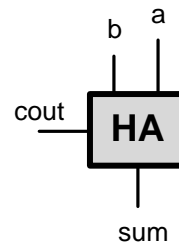
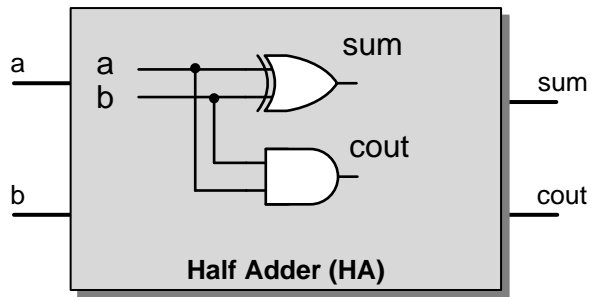
# Xilinx PYNQ-Z2 FPGA



# Lab2



# Lab2



# VLSI for DSP

## Vivado Design Flow Getting Started

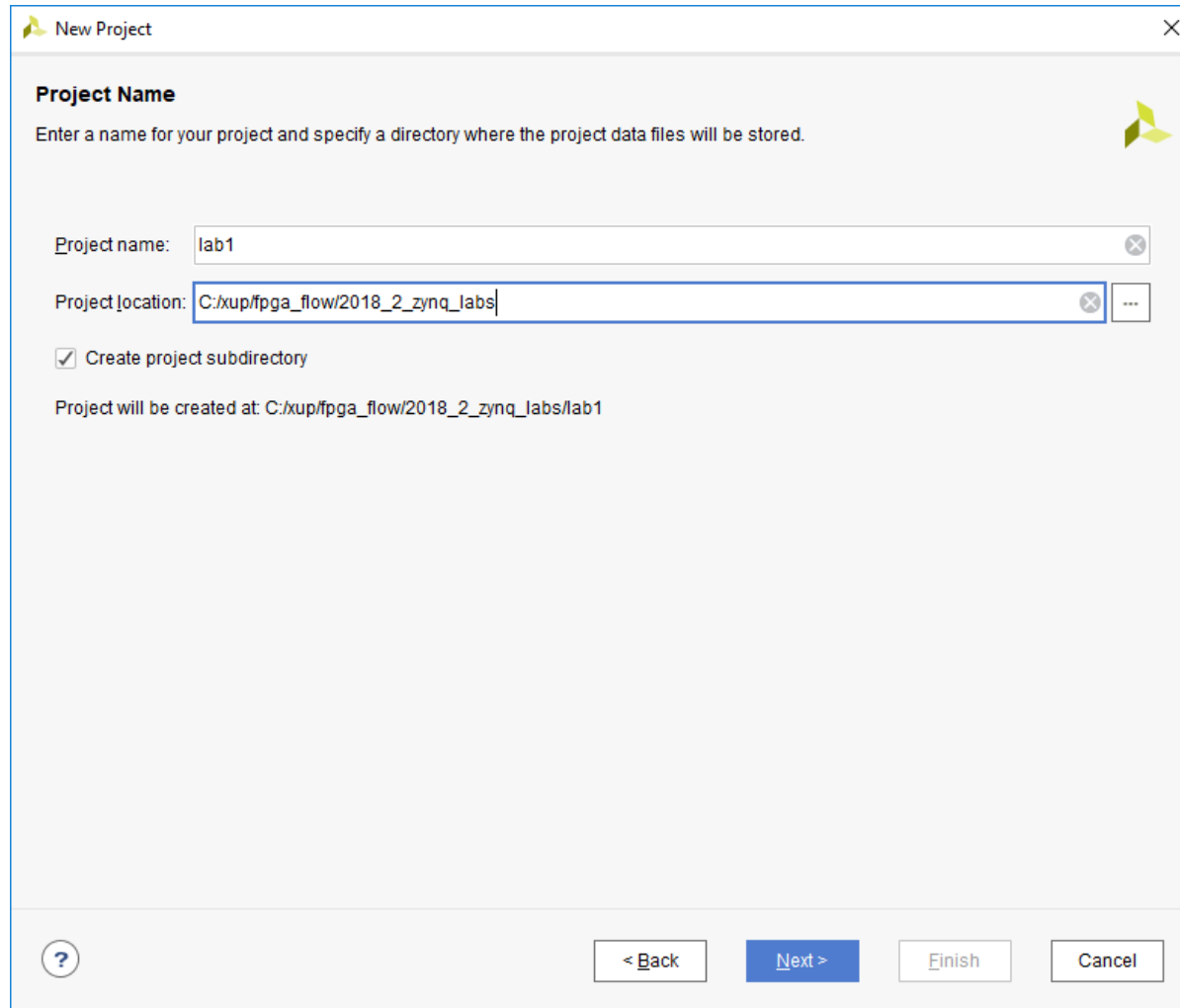
Department of Electronic Engineering  
Chang Guan University  
[chenyh@mail.cgu.edu.tw](mailto:chenyh@mail.cgu.edu.tw)



# Create a Vivado Project using IDE

- Launch Vivado and create an empty project.
  1. Open Vivado by selecting **Start > Xilinx Design Tools > Vivado 2018.2**
  2. Click **Create New Project** to start the wizard. You will see *Create A New Vivado Project* dialog box. Click **Next**.
  3. Click the Browse button of the *Project location* field of the **New Project** form, browse to **C:\xilinx\_works**, and click **Select**.
  4. Enter **lab2** in the *Project name* field. Make sure that the *Create Project Subdirectory* box is checked. Click **Next**.

# Create a Vivado Project using IDE



The image shows the 'New Project' dialog box in the Vivado IDE. The dialog has a title bar with a yellow Vivado logo and a close button. The main area is titled 'Project Name' and contains the instruction: 'Enter a name for your project and specify a directory where the project data files will be stored.' There are two text input fields: 'Project name:' with the value 'lab1' and 'Project location:' with the value 'C:/xup/fpga\_flow/2018\_2\_zynq\_labs'. The 'Project location' field has a blue border and a file explorer icon to its right. Below these fields is a checkbox labeled 'Create project subdirectory' which is checked. At the bottom of the main area, it says 'Project will be created at: C:/xup/fpga\_flow/2018\_2\_zynq\_labs/lab1'. The bottom of the dialog features a help icon (question mark in a circle) on the left and four buttons: '< Back', 'Next >' (highlighted in blue), 'Finish', and 'Cancel'.

**New Project**

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

Project name: lab1

Project location: C:/xup/fpga\_flow/2018\_2\_zynq\_labs

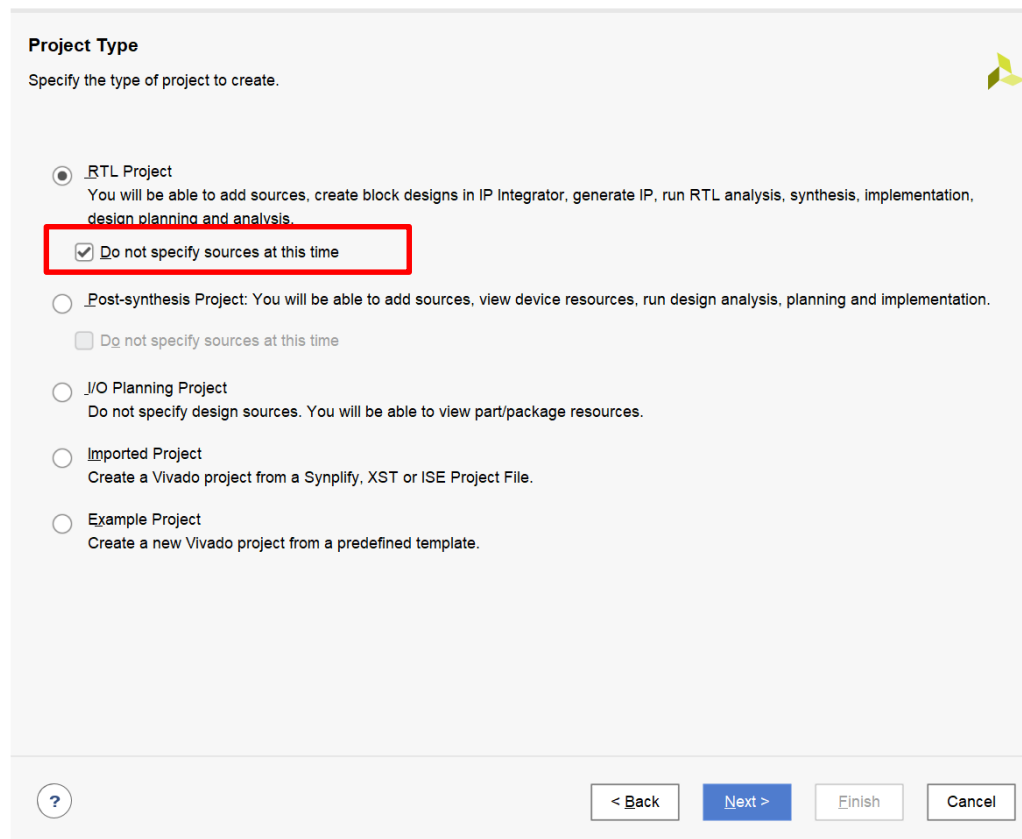
☒ Create project subdirectory

Project will be created at: C:/xup/fpga\_flow/2018\_2\_zynq\_labs/lab1

? < Back Next > Finish Cancel

# Create a Vivado Project using IDE

5. Select **RTL Project** option in the *Project Type* form, and click **Next**.



**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**






☐ **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.

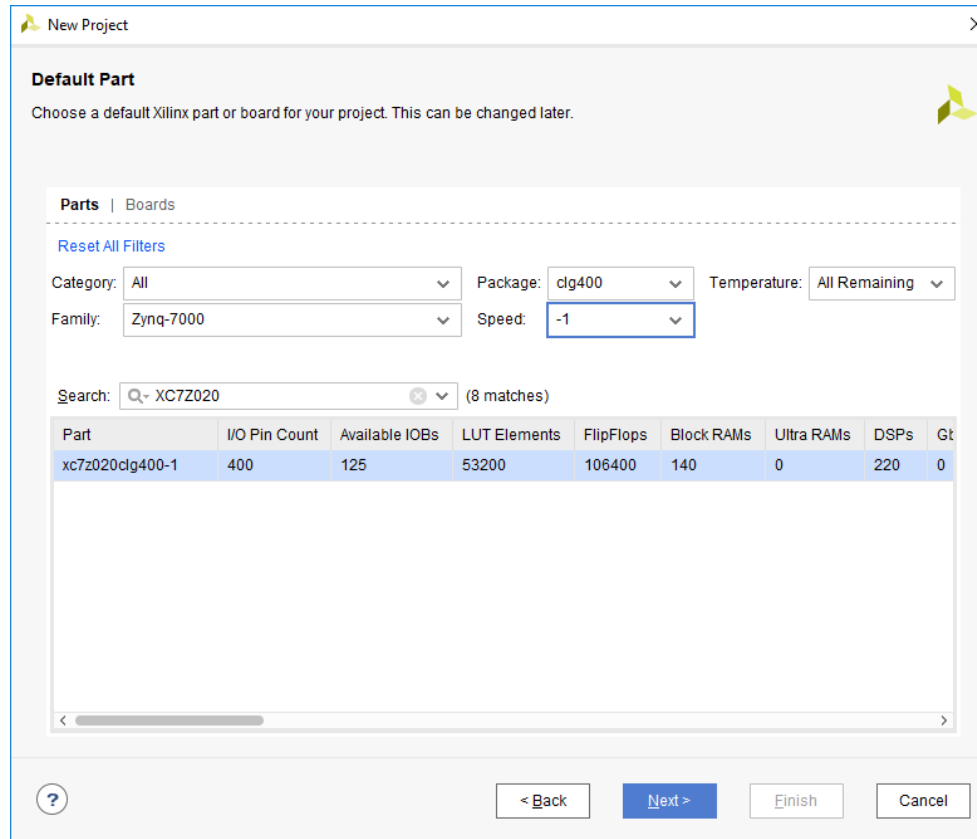
☐ **Example Project**  
Create a new Vivado project from a predefined template.



# Create a Vivado Project

7. In the *Default Part* form, use the **Parts** option and various drop-down fields of the **Filter** section. Select the **XC7Z020clg400-1**.
8. Click **Finish** to create the Vivado project.



**New Project**

**Default Part**  
Choose a default Xilinx part or board for your project. This can be changed later.

**Parts** | Boards

[Reset All Filters](#)



Category: All Package: clg400 Temperature: All Remaining  
Family: Zynq-7000 Speed: -1

Search:  (8 matches)

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gt
xc7z020clg400-1	400	125	53200	106400	140	0	220	0

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

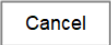
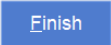
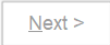
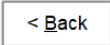

# Create a Vivado Project



## New Project Summary

- i** A new RTL project named 'Lab2' will be created.
- i** The default part and product family for the new project:
  - Default Part: xc7z020clg400-1
  - Product: Zynq-7000
  - Family: Zynq-7000
  - Package: clg400
  - Speed Grade: -1

To create the project, click Finish



# Create a Vivado Project

The screenshot displays the Vivado IDE interface for a new project named "Lab2". The interface is divided into several panes:

- Flow Navigator:** Located on the left, it shows the project hierarchy with sections for PROJECT MANAGER, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, IMPLEMENTATION, and PROGRAM AND DEBUG.
- Project Manager - Lab2:** The central pane, which is further divided into:
  - Sources:** Shows the project's source files, including Design Sources, Constraints, Simulation Sources (sim\_1), and Utility Sources.
  - Properties:** A pane for viewing the properties of selected objects.
  - Project Summary:** A pane providing an overview of the project's settings and status.
- Design Runs:** A table at the bottom showing the status of various design runs.

**Project Summary Details:**

- Overview | Dashboard**
- Settings Edit**
- Project name: Lab2
- Project location: E:/vivado\_work/Lab2
- Product family: Zynq-7000
- Project part: xc7z020clg400-1
- Top module name: Not defined
- Target language: Verilog
- Simulator language: Mixed

**Synthesis and Implementation Status:**

- Synthesis:** Status: Not started; Messages: No errors or warnings; Part: xc7z020clg400-1; Strategy: Vivado Synthesis Defaults; Report Strategy: Vivado Synthesis Default Reports.
- Implementation:** Status: Not started; Messages: No errors or warnings; Part: xc7z020clg400-1; Strategy: Vivado Implementation Defaults; Report Strategy: Vivado Implementation Default Reports; Incremental implementation: None.

**Design Runs Table:**

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

# Create Lab2.v



The screenshot displays the Vivado IDE interface for a project named "Lab2". The left sidebar shows the "PROJECT MANAGER" with the "Add Sources" button highlighted in a red box. The main workspace is divided into several panels:

- Sources:** Shows the project hierarchy with "Design Sources", "Constraints", "Simulation Sources" (containing "sim\_1"), and "Utility Sources".
- Project Summary:** Provides an overview of the project settings, including project name, location, product family, and target language.
- Synthesis:** Displays the synthesis status, messages, and strategy.
- Implementation:** Displays the implementation status, messages, and strategy.
- Design Runs:** A table showing the progress of various design runs.

The "Design Runs" table is as follows:

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

# Create RCA4.v



## 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

HA.v  
FA.v  
RCA4.v



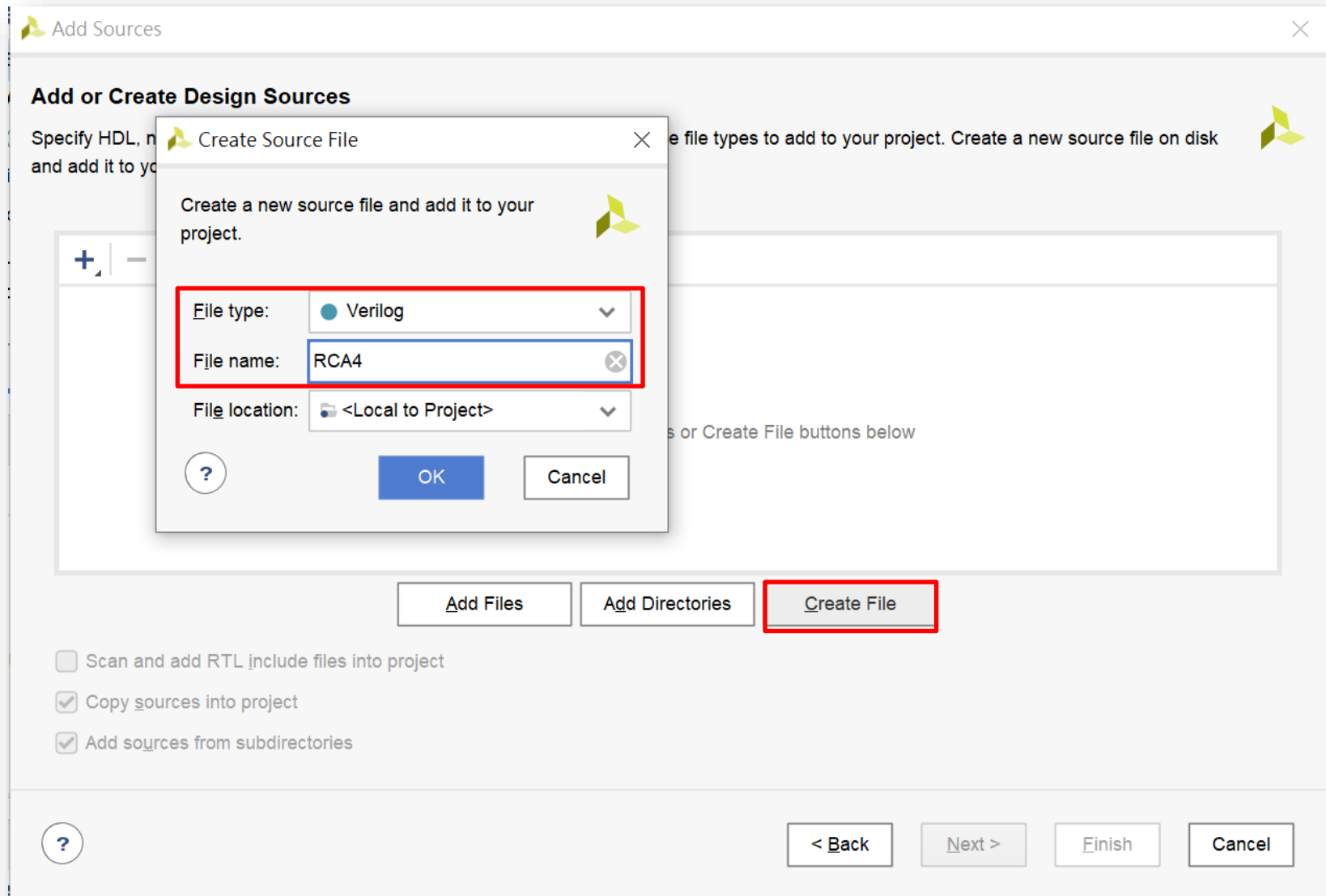
< Back

Next >

Finish

Cancel

# Create RCA4.v



# Add HA.v and FA.v

The screenshot displays the Xilinx Vivado IDE interface. The **PROJECT MANAGER - Lab2** window is open, showing the **Sources** tab. A red box highlights the **Design Sources (1)** section, which contains the following items:

- RCA4 (RCA4.v) (4)
  - U1\_HA : HA (HA.v)
  - U2\_FA : FA (FA.v)
  - U3\_FA : FA (FA.v)
  - U4\_FA : FA (FA.v)

The **Source File Properties** window is also open, showing the properties for **RCA4.v**. The **General** tab is selected, and the **Enabled** checkbox is checked. The **Location** is **E:/vivado\_work/Lab2/Lab2**, the **Type** is **Verilog**, and the **Library** is **xil\_defaultlib**.

The **Project Summary** window is open, showing the Verilog code for **RCA4.v**:

```
19 :  
20 :  
21 :  
22 :  
23 module RCA4(a, b, cout, s);  
24 :  
25 input [3:0] a, b;  
26 output [3:0] s;  
27 output cout;  
28 :  
29 wire[3:0] c;  
30 :  
31 HA U1_HA(.a(a[0]), .b(b[0]), .cout(c[0]), .sum(s[0]));  
32 FA U2_FA(.a(a[1]), .b(b[1]), .c(c[0]), .cout(c[1]), .sum(s[1]));  
33 FA U3_FA(.a(a[2]), .b(b[2]), .c(c[1]), .cout(c[2]), .sum(s[2]));  
34 FA U4_FA(.a(a[3]), .b(b[3]), .c(c[2]), .cout(c[3]), .sum(s[3]));  
35 :  
36 assign cout = c[3];  
37 :  
38 endmodule  
39 :  
40 :
```

The **Design Runs** window is open, showing the status of the synthesis and implementation runs:

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

# VLSI for DSP

## Simulate the Design using the Vivado Simulator



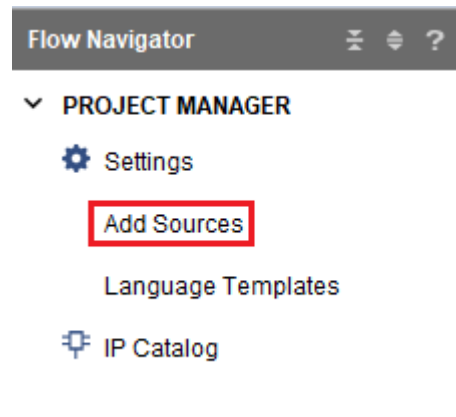
Department of Electronic Engineering  
Chang Guan University  
[chenyh@mail.cgu.edu.tw](mailto:chenyh@mail.cgu.edu.tw)





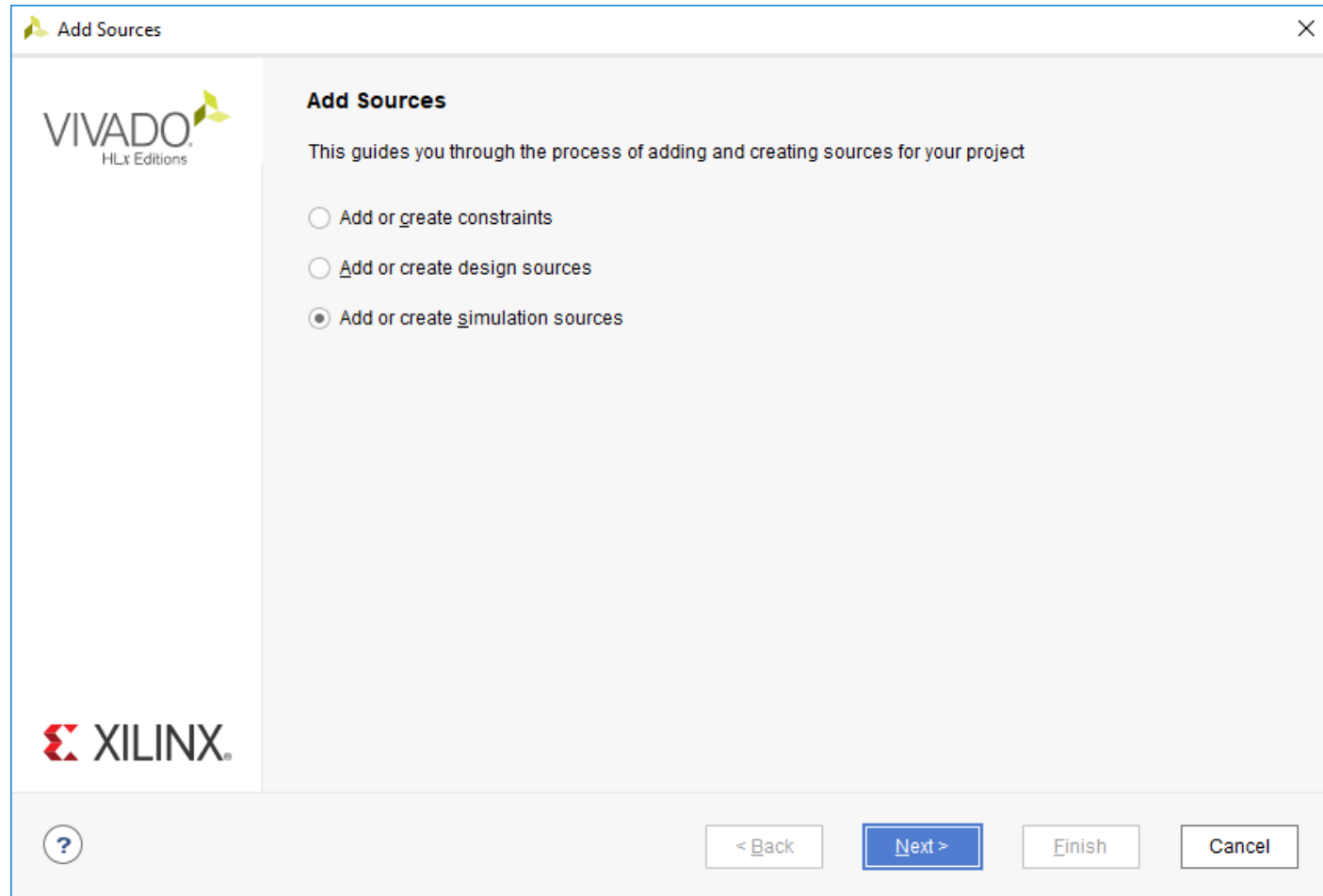
# Add the TM\_RCA4.v testbench file

1. Click **Add Sources** under the *Project Manager* tasks of the *Flow Navigator* pane.



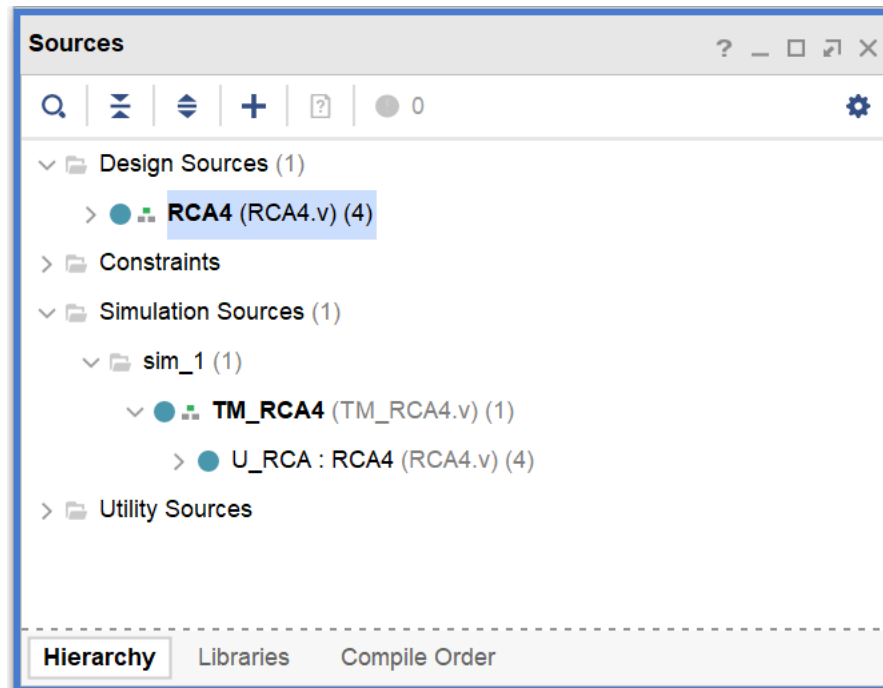
# Add the testbench file

2. Select the *Add or Create Simulation Sources* option and click **Next**.



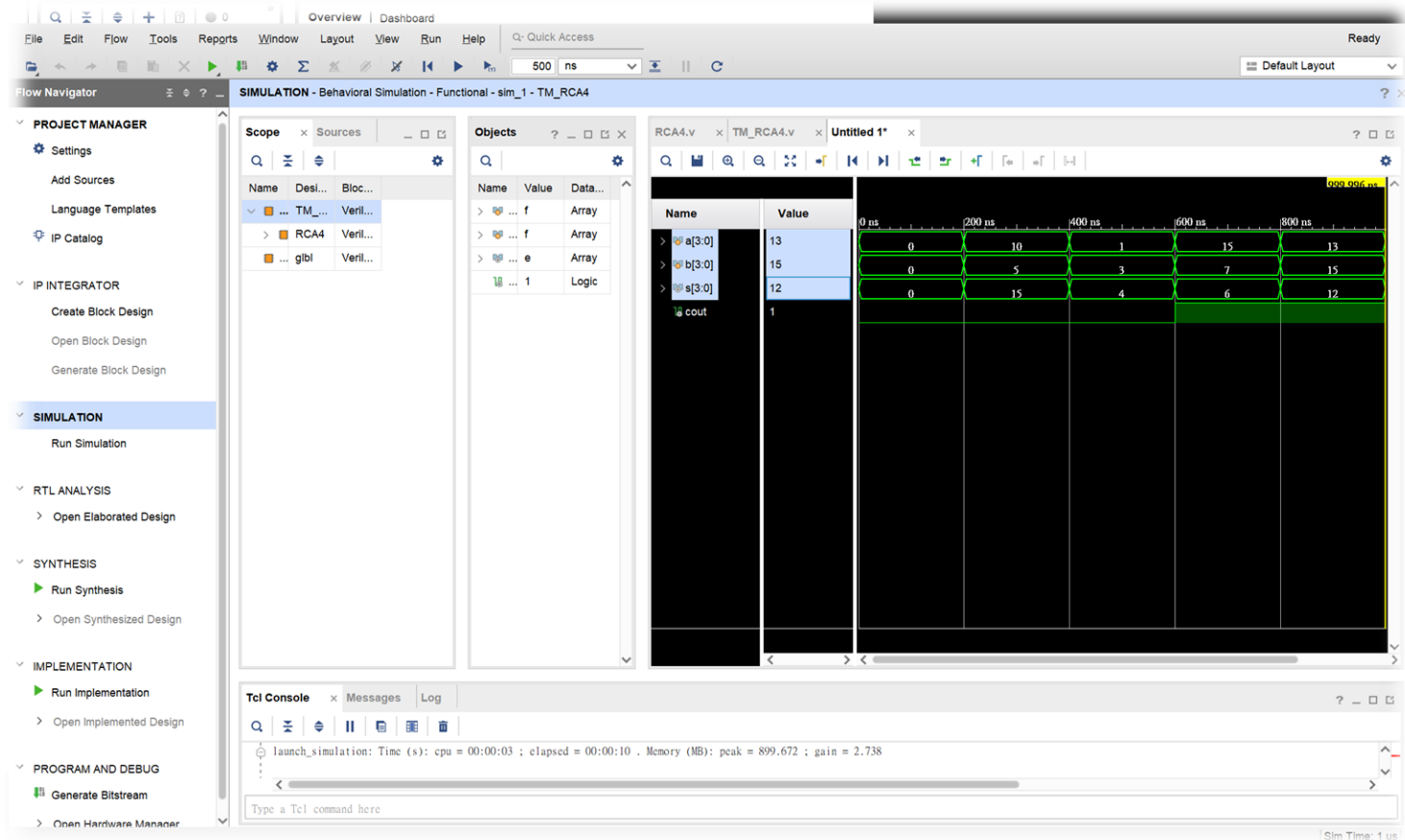
# Add the testbench file

3. In the *Add Sources Files* form, click the **Blue Plus** button and then **Add Files...**
4. Select *TM\_RCA4.v* and click **OK**.
5. Click **Finish**.
6. Select the *Sources* tab and expand the *Simulation Sources* group.



# Simulate the design

7. Click on **Simulation > Run Simulation > Run Behavioral Simulation** under the *Project Manager* tasks of the *Flow Navigator* pane.



# Close Simulation

Lab2 - [E:/vivado\_work/Lab2/Lab2.xpr] - Vivado 2018.3

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

Synthesis Out-of-date details

Default Layout

500 ns

**File**

- Project
  - Add Sources... Alt+A
  - Close Project
  - Constraints
  - Simulation Waveform
  - Close Simulation**
  - Checkpoint
  - JP
  - Text Editor
  - Import
  - Export
  - Launch SDK
  - Print... Ctrl+P
  - Exit
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

**SIMULATION - Behavioral Simulation - Functional - sim\_1 - TM\_RCA4**

Scope Sources

Name	Desi...	Bloc...
TM...	Veril...	
RCA4	Veril...	
gbl	Veril...	

Objects

Name	Value	Data...
f	Array	
f	Array	
e	Array	
1	Logic	

RCA4.v x TM\_RCA4.v x Untitled 1\*

Name	Value
a[3:0]	13
b[3:0]	15
s[3:0]	12
cout	1

0 ns 200 ns 400 ns 600 ns 800 ns

0 10 1 15 13

0 5 3 7 15

0 15 4 6 12

600.000 ns

Tcl Console Messages Log Reports

```
add_files -fileset constrs_1 E:/vivado_work/Lab2/Lab2.srcs/constrs_1/new/RCA4.xdc
```

Type a Tcl command here

Sim Time: 1 us

# VLSI for DSP


## Synthesize the Design



Department of Electronic Engineering  
Chang Guan University  
[chenyh@mail.cgu.edu.tw](mailto:chenyh@mail.cgu.edu.tw)



# Synthesize the design

1. Click on **Run Synthesis** under the *SYNTHESIS* tasks of the *Flow Navigator* pane.
2. Select the *Open Synthesized Design* option and click **OK** as we want to look at the synthesis output before progressing to the implementation stage.
  - Click **Yes** to close the elaborated design if the dialog box is displayed.
3. Select the **Project Summary** tab and understand the various windows.
  - If you don't see the Project Summary tab then select **Window > Project Summary** or click the **Project Summary** icon 

# Synthesize the design

**Project Summary**?—□↶✕

**Overview** | Dashboard

**Settings** [Edit](#)

Project name: Lab2

Project location: E:/vivado\_work/Lab2

Product family: Zynq-7000

Project part: xc7z020clg400-1

Top module name: RCA4

Target language: Verilog

Simulator language: Mixed

**Synthesis**

Status: ✓ Complete

Messages: ⚠ 1 warning

Part: xc7z020clg400-1

Strategy: [Vivado Synthesis Defaults](#)

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

**Implementation**

Status: Not started

Messages: No errors or warnings

Part: xc7z020clg400-1

Strategy: [Vivado Implementation Defaults](#)

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

Incremental implementation: [None](#)

**DRC Violations**

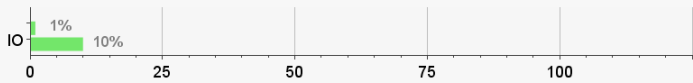
[Run Implementation](#) to see DRC results

**Timing**

[Run Implementation](#) to see timing results

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

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



Estimated Utilization (%)

Category	Utilization (%)
IO	10%

**Power**

[Run Implementation](#) to see power results



# VLSI for DSP

## Implement the Design

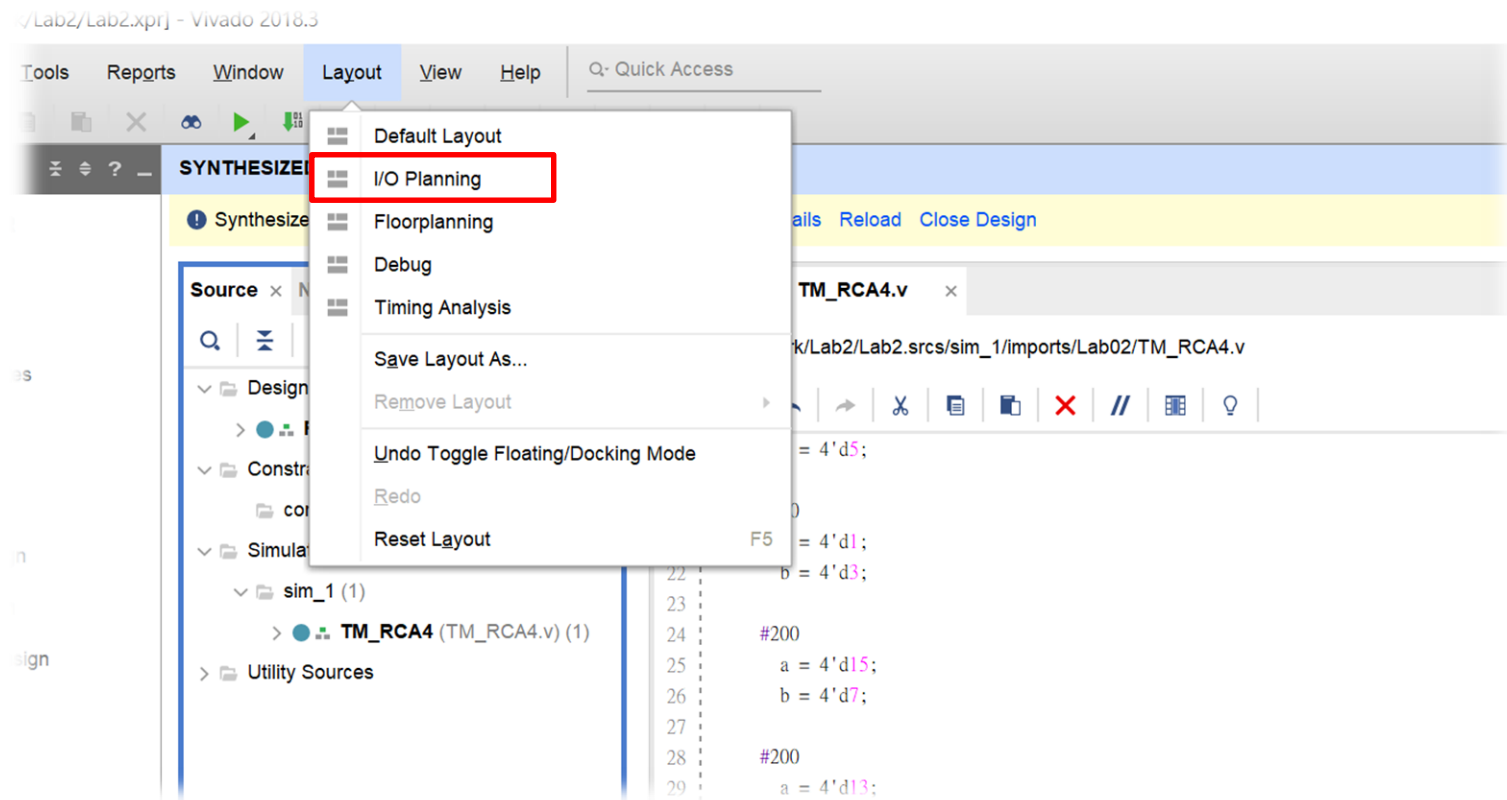


Department of Electronic Engineering  
Chang Guan University  
[chenyh@mail.cgu.edu.tw](mailto:chenyh@mail.cgu.edu.tw)

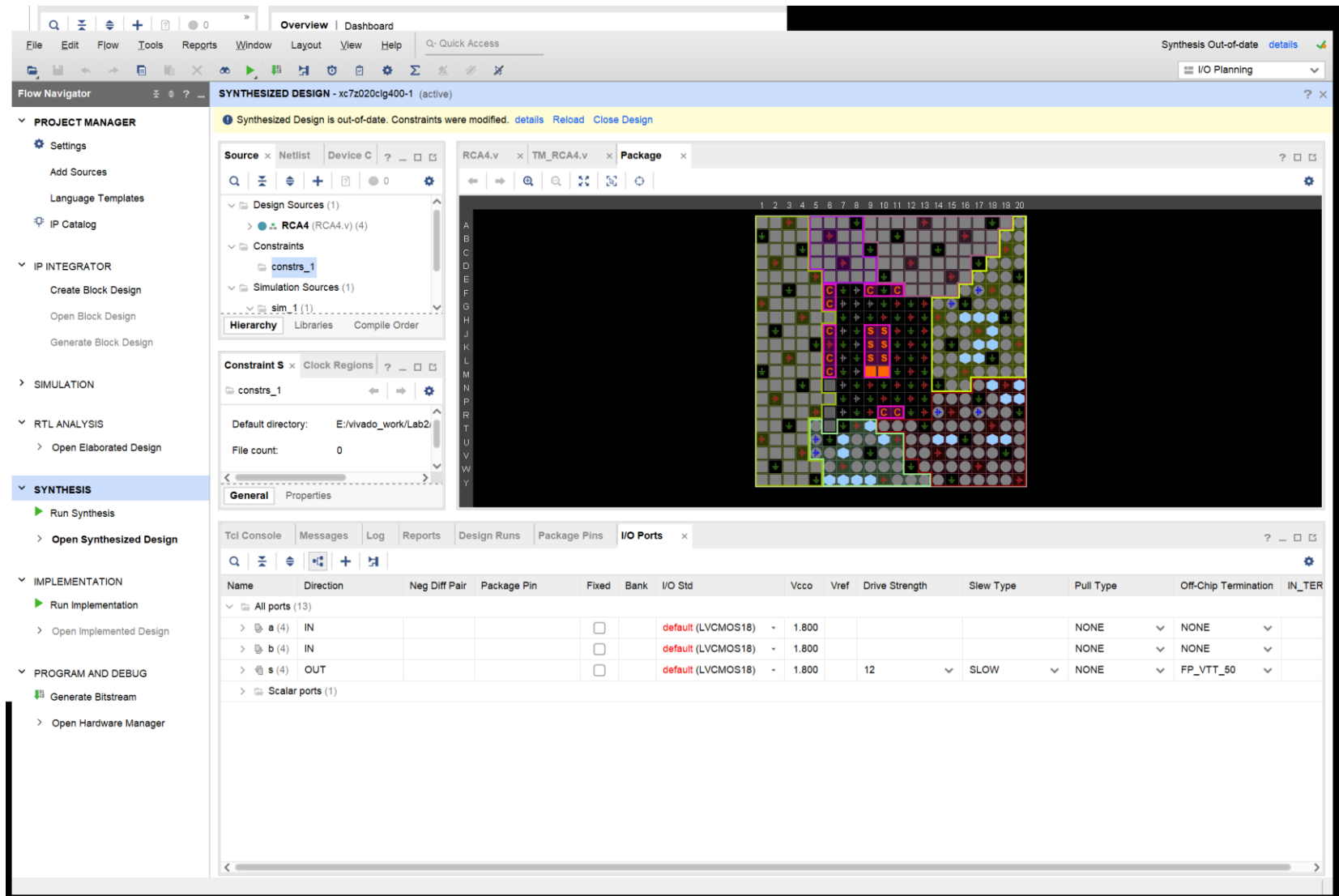


# I/O Planning

- Layout → I/O Planning



# I/O Planning



The screenshot shows the Xilinx Vivado IDE with the I/O Planning window open. The window displays a grid of pins with various constraints and a table of I/O ports.

**Flow Navigator:**

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

**SYNTHESIZED DESIGN - xc7z020clg400-1 (active)**

Synthesized Design is out-of-date. Constraints were modified. [details](#) [Reload](#) [Close Design](#)

**Source** x Netlist Device C ? - □ □

Design Sources (1)

- RCA4 (RCA4.v) (4)

Constraints

- constrs\_1

Simulation Sources (1)

- sim\_1 (1)

Hierarchy Libraries Compile Order

**Constraint S** x Clock Regions ? - □ □

constrs\_1

Default directory: E:/vivado\_work/Lab2

File count: 0

General Properties

**I/O Ports**

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
All ports (13)													
a (4)	IN			<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE	
b (4)	IN			<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE	
s (4)	OUT			<input type="checkbox"/>		default (LVCMOS18)	1.800		12	SLOW	NONE	FP_VTT_50	
Scalar ports (1)													



# FPGA pins

Signal Name	PL PIN
BTN0	D19
BTN1	D20
BTN2	L20
BTN3	L19

*Table 9 Push Button PL pin mapping*

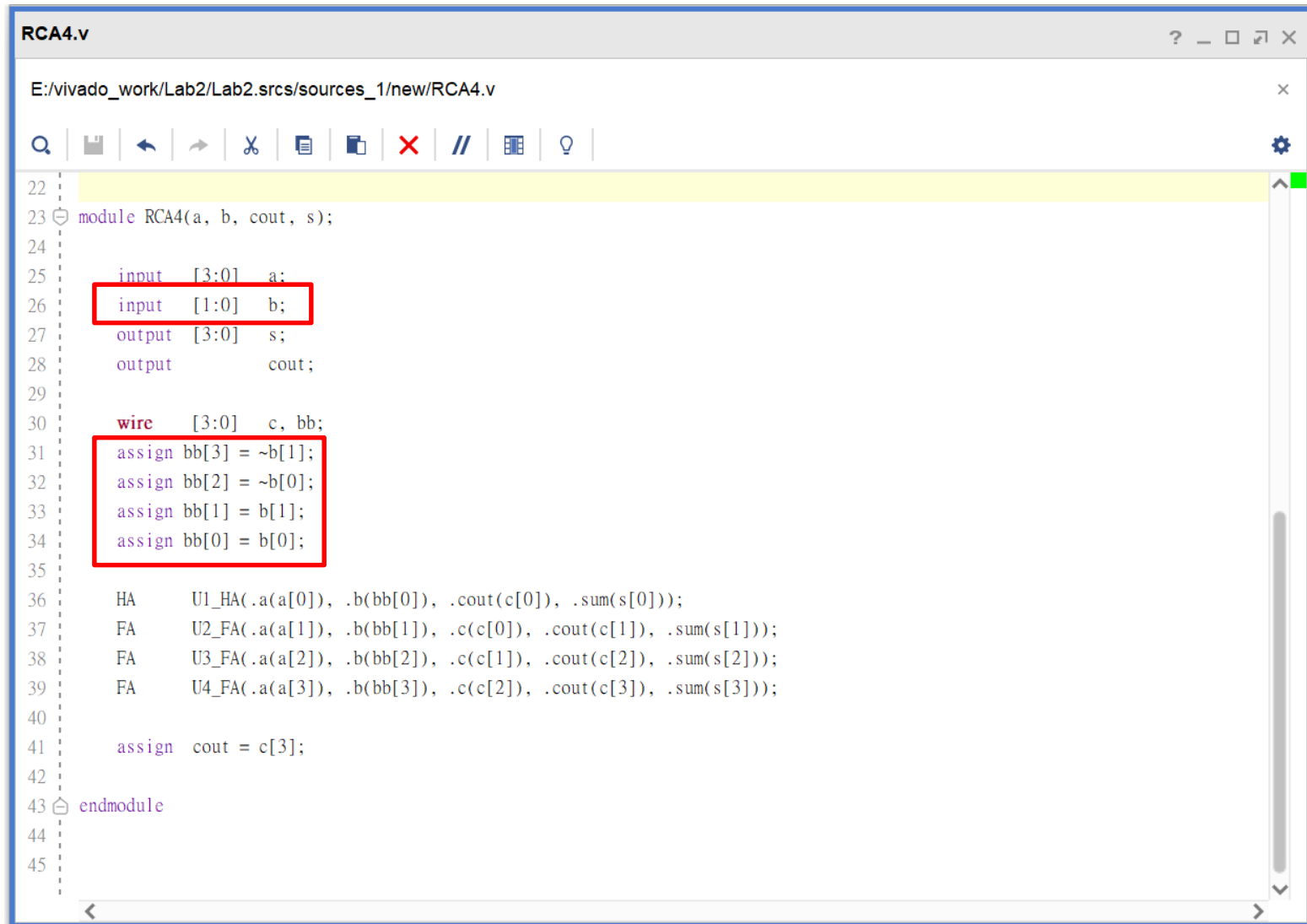
Signal Name	PL PIN
SW0	M20
SW1	M19

*Table 10 Dip switch PL pin mapping*

Signal Name	PL PIN
LED0	R14
LED1	P14
LED2	N16
LED3	M14

*Table 11 LED PL pin mapping*

# Modify



```
RCA4.v
E:/vivado_work/Lab2/Lab2.srscs/sources_1/new/RCA4.v

22
23 module RCA4(a, b, cout, s);
24
25     input  [3:0]  a;
26     input  [1:0]  b;
27     output [3:0]  s;
28     output                cout;
29
30     wire  [3:0]  c, bb;
31     assign bb[3] = ~b[1];
32     assign bb[2] = ~b[0];
33     assign bb[1] = b[1];
34     assign bb[0] = b[0];
35
36     HA    U1_HA(.a(a[0]), .b(bb[0]), .cout(c[0]), .sum(s[0]));
37     FA    U2_FA(.a(a[1]), .b(bb[1]), .c(c[0]), .cout(c[1]), .sum(s[1]));
38     FA    U3_FA(.a(a[2]), .b(bb[2]), .c(c[1]), .cout(c[2]), .sum(s[2]));
39     FA    U4_FA(.a(a[3]), .b(bb[3]), .c(c[2]), .cout(c[3]), .sum(s[3]));
40
41     assign cout = c[3];
42
43 endmodule
44
45
```

# FPGA pins

a[3:0]

Signal Name	PL PIN
BTN0	D19
BTN1	D20
BTN2	L20
BTN3	L19

*Table 9 Push Button PL pin mapping*

b[1:0]

Signal Name	PL PIN
SW0	M20
SW1	M19

*Table 10 Dip switch PL pin mapping*

s[3:0]

Signal Name	PL PIN
LED0	R14
LED1	P14
LED2	N16
LED3	M14

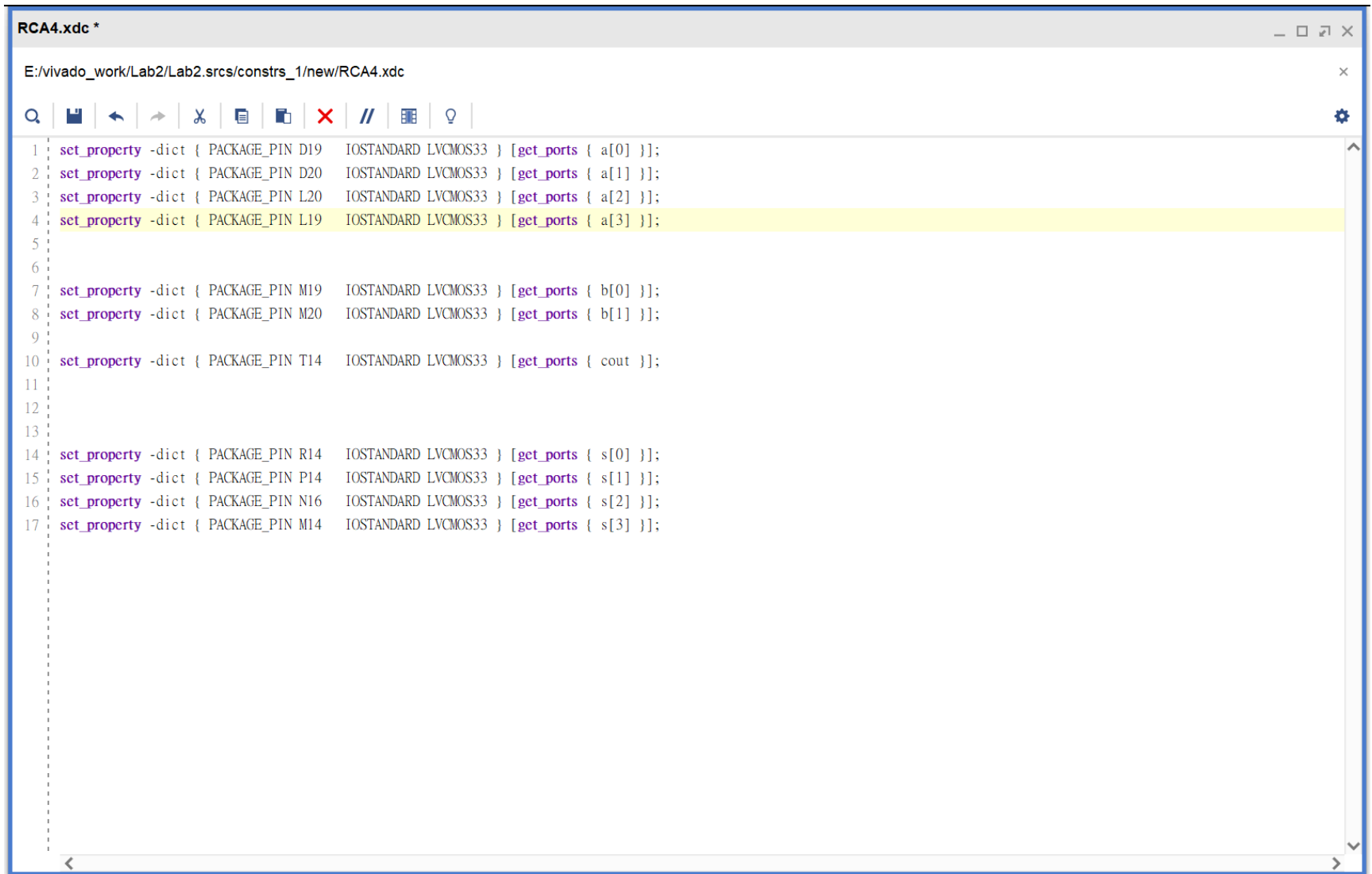
*Table 11 LED PL pin mapping*

# FPGA pins

I/O Ports										
Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Str	
▼ All ports (11)										
▼ a (4)	IN			<input checked="" type="checkbox"/>	35	default (LVCMOS18)	1.800			
a[3]	IN		L19	<input checked="" type="checkbox"/>	35	default (LVCMOS18)	1.800			
a[2]	IN		L20	<input checked="" type="checkbox"/>	35	default (LVCMOS18)	1.800			
a[1]	IN		D20	<input checked="" type="checkbox"/>	35	default (LVCMOS18)	1.800			
a[0]	IN		D19	<input checked="" type="checkbox"/>	35	default (LVCMOS18)	1.800			
▼ b (2)	IN			<input checked="" type="checkbox"/>	35	default (LVCMOS18)	1.800			
b[1]	IN		M19	<input checked="" type="checkbox"/>	35	default (LVCMOS18)	1.800			
b[0]	IN		M20	<input checked="" type="checkbox"/>	35	default (LVCMOS18)	1.800			
▼ s (4)	OUT			<input checked="" type="checkbox"/>	(Multiple)	LVCMOS18	1.800		12	
s[3]	OUT		M14	<input checked="" type="checkbox"/>	35	LVCMOS18	1.800		12	
s[2]	OUT		N16	<input checked="" type="checkbox"/>	35	LVCMOS18	1.800		12	
s[1]	OUT		P14	<input checked="" type="checkbox"/>	34	LVCMOS18	1.800		12	
s[0]	OUT		R14	<input checked="" type="checkbox"/>	34	LVCMOS18	1.800		12	
> Scalar ports (1)										



# xdc file

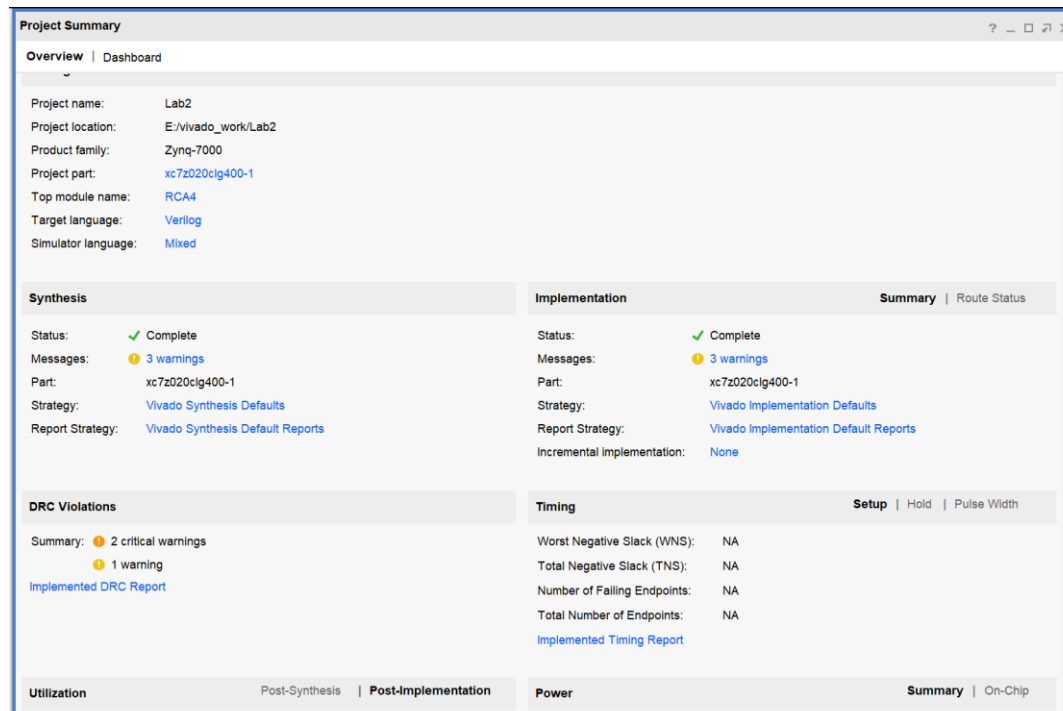


The screenshot shows a text editor window titled "RCA4.xdc \*". The address bar indicates the file path: "E:/vivado\_work/Lab2/Lab2.srcs/constrs\_1/new/RCA4.xdc". The editor contains 17 lines of XDC code. Line 4 is highlighted in yellow. The code consists of multiple "set\_property" commands for various pins, each with a dictionary of properties including PACKAGE\_PIN, IOSTANDARD, LVCMOS33, and a get\_ports list.

```
1 set_property -dict { PACKAGE_PIN D19 IOSTANDARD LVCMOS33 } [get_ports { a[0] }];
2 set_property -dict { PACKAGE_PIN D20 IOSTANDARD LVCMOS33 } [get_ports { a[1] }];
3 set_property -dict { PACKAGE_PIN L20 IOSTANDARD LVCMOS33 } [get_ports { a[2] }];
4 set_property -dict { PACKAGE_PIN L19 IOSTANDARD LVCMOS33 } [get_ports { a[3] }];
5
6
7 set_property -dict { PACKAGE_PIN M19 IOSTANDARD LVCMOS33 } [get_ports { b[0] }];
8 set_property -dict { PACKAGE_PIN M20 IOSTANDARD LVCMOS33 } [get_ports { b[1] }];
9
10 set_property -dict { PACKAGE_PIN T14 IOSTANDARD LVCMOS33 } [get_ports { cout }];
11
12
13
14 set_property -dict { PACKAGE_PIN R14 IOSTANDARD LVCMOS33 } [get_ports { s[0] }];
15 set_property -dict { PACKAGE_PIN P14 IOSTANDARD LVCMOS33 } [get_ports { s[1] }];
16 set_property -dict { PACKAGE_PIN N16 IOSTANDARD LVCMOS33 } [get_ports { s[2] }];
17 set_property -dict { PACKAGE_PIN M14 IOSTANDARD LVCMOS33 } [get_ports { s[3] }];
```

# Implement the design

1. Click on **Run Implementation** under the *Implementation* tasks of the *Flow Navigator* pane.
  - The implementation process will be run on the synthesized design. When the process is completed an *Implementation Completed* dialog box with three options will be displayed.
2. Select **Generate Bitstream** and click **OK**.



# VLSI for DSP

## Generate the Bitstream and Verify Functionality

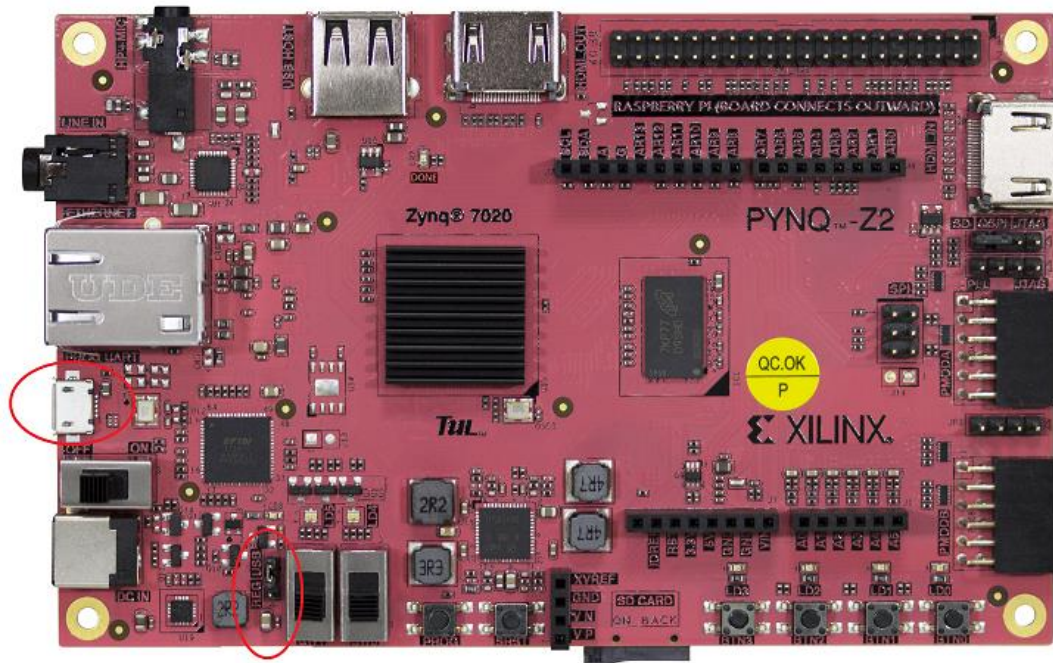


Department of Electronic Engineering  
Chang Guan University  
[chenyh@mail.cgu.edu.tw](mailto:chenyh@mail.cgu.edu.tw)



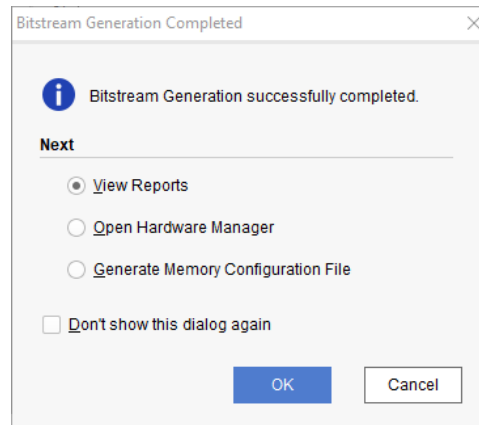
# Generate the Bitstream

- **Connect the board and power it ON. Generate the bitstream, open a hardware session, and program the FPGA.**
  1. Make sure that the Micro-USB cable is connected to the JTAG PROG connector.
  2. The PYNQ-Z2 can be powered through USB power via the JTAG PROG.
    - Make sure that the board is set to use USB power.



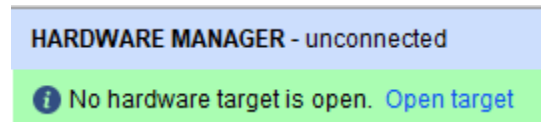
# Generate the Bitstream

3. Power **ON** the board.
4. Click on the **Generate Bitstream** entry under the *PROGRAM AND DEBUG* tasks of the *Flow Navigator* pane.
  - The bitstream generation process will be run on the implemented design. When the process is completed a *Bitstream Generation Completed* dialog box with three options will be displayed.



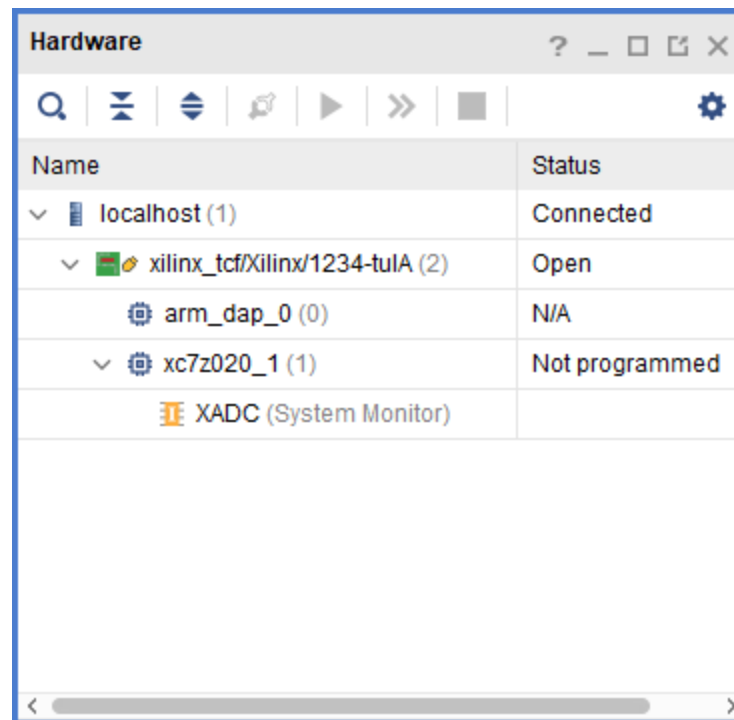
# Generate the Bitstream

5. Select the *Open Hardware Manager* option and click **OK**.
  - The Hardware Manager window will open indicating “unconnected” status.
6. Click on the **Open target** link.



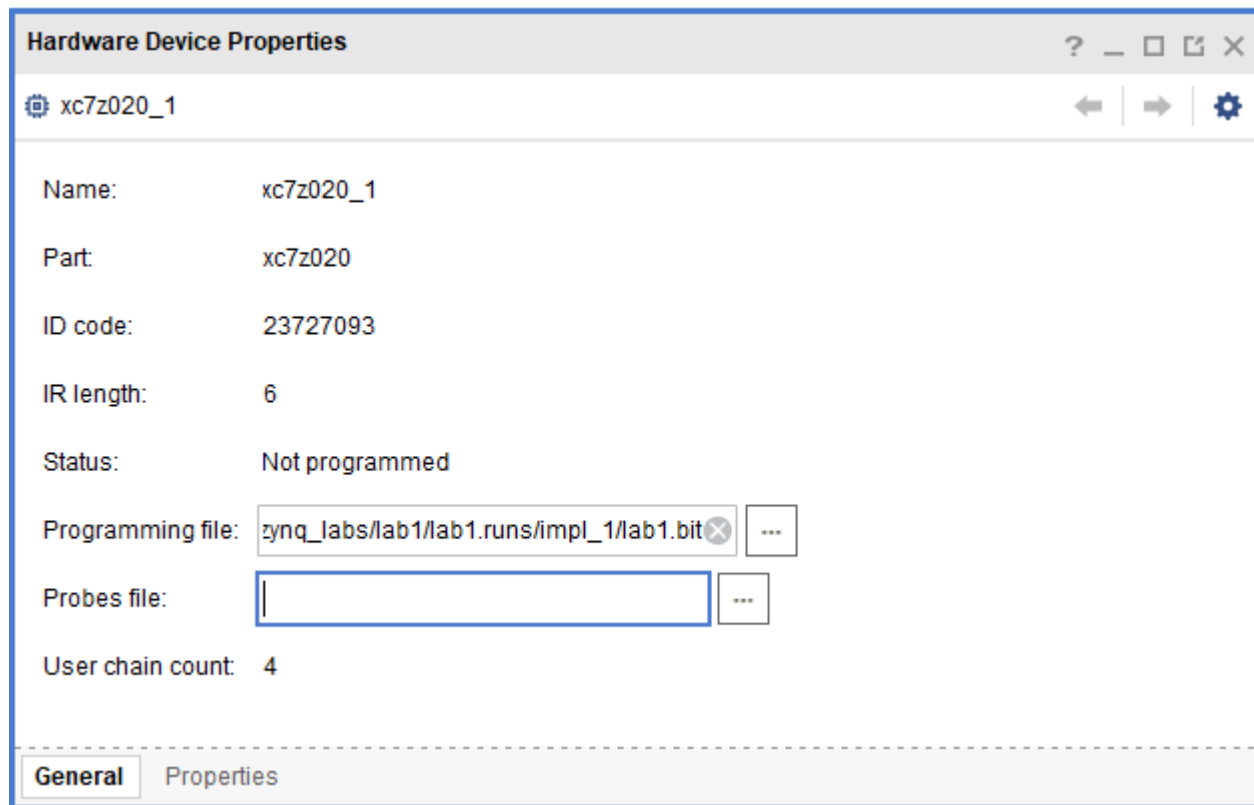
# Download

7. From the dropdown menu, click **Auto Connect**.
  - The Hardware Session status changes from Unconnected to the server name and the device is highlighted. Also notice that the Status indicates that it is not programmed.



# Download

- Select the device and verify that the lab2.bit is selected as the programming file in the General tab.





# Download

8. Click on the *Program device* link in the green information bar to program the target FPGA device. Another way is to right click on the device and select *Program Device*.

HARDWARE MANAGER - localhost/xilinx\_tcf/Xilinx/1234-tula

 There are no debug cores. [Program device](#) [Refresh device](#)

*Selecting to program the FPGA*

# Download

9. Click **Program** to program the FPGA.
  - The DONE LED will lit when the device is programmed. You may see some other LEDs lit depending on switch positions.
10. Verify the functionality by flipping the switches and observing the output on the LEDs (Refer to the earlier logic diagram).
11. When satisfied, power **OFF** the board.
12. Close the hardware session by selecting **File > Close Hardware Manager**.
13. Click **OK** to close the session.
14. Close the **Vivado** program by selecting **File > Exit** and click **OK**.