VLSI for DSP

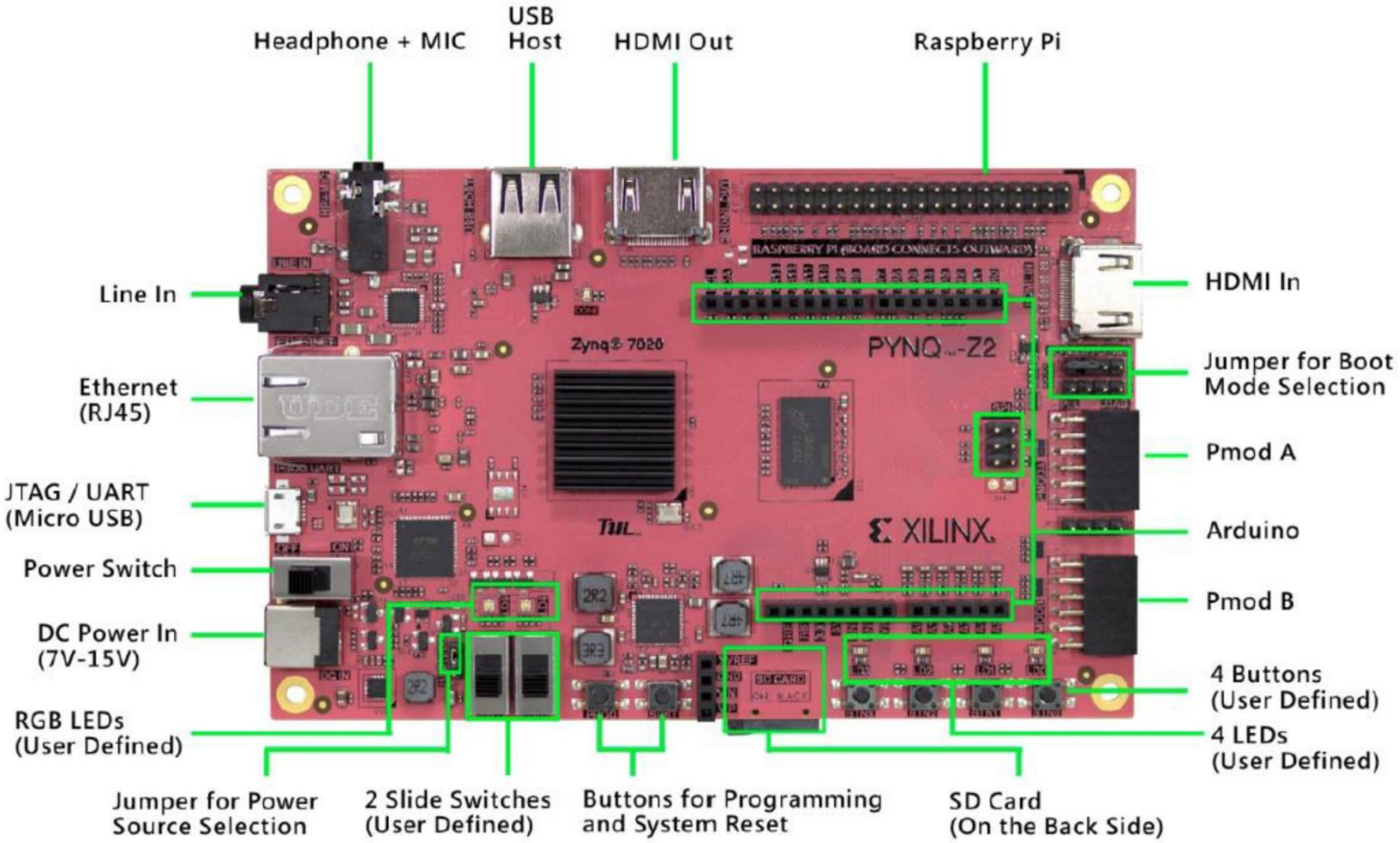
Yuan-Ho Chen

Department of Electronic Engineering

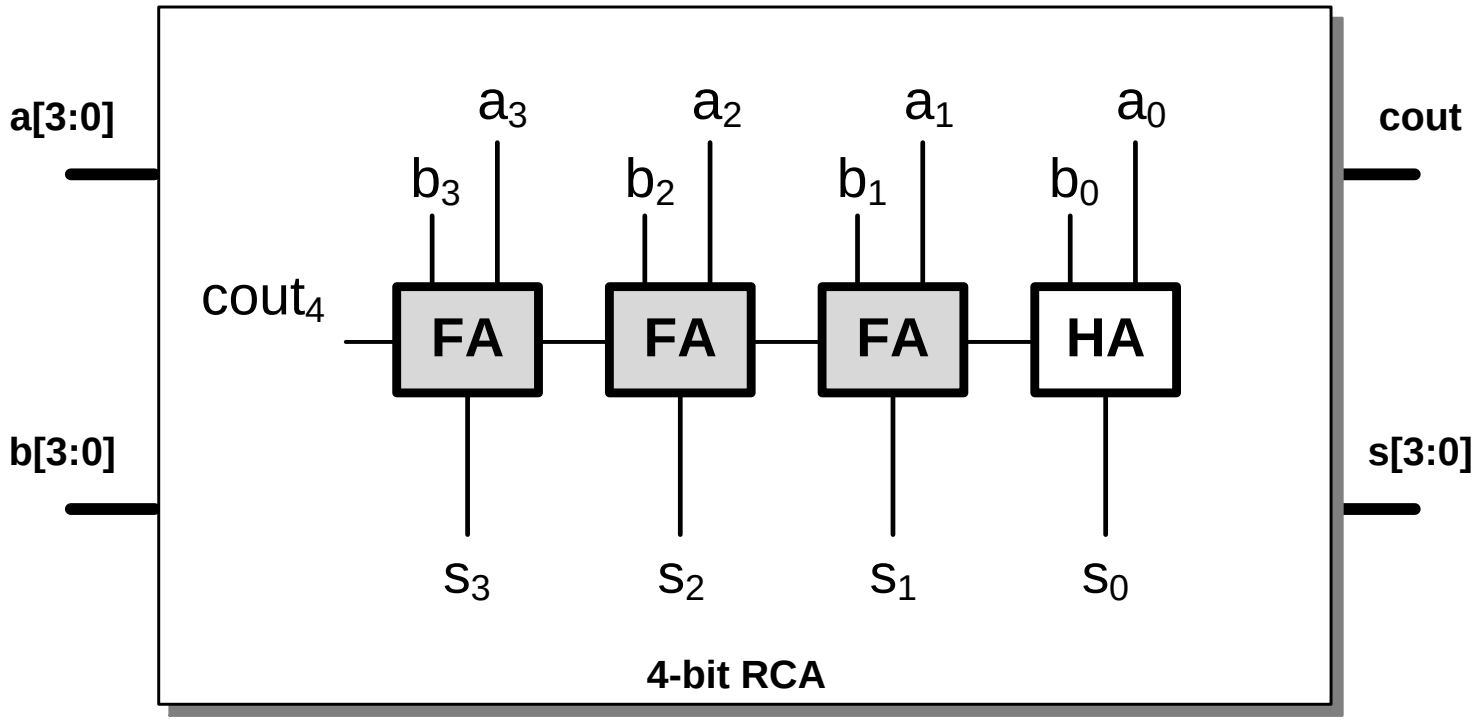
Chang Guan University

chenyh@mail.cgu.edu.tw

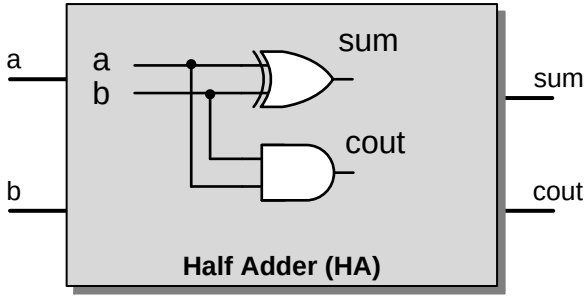
Xilinx PYNQ-Z2 FPGA

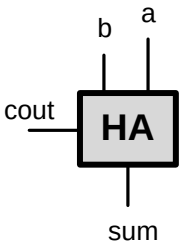


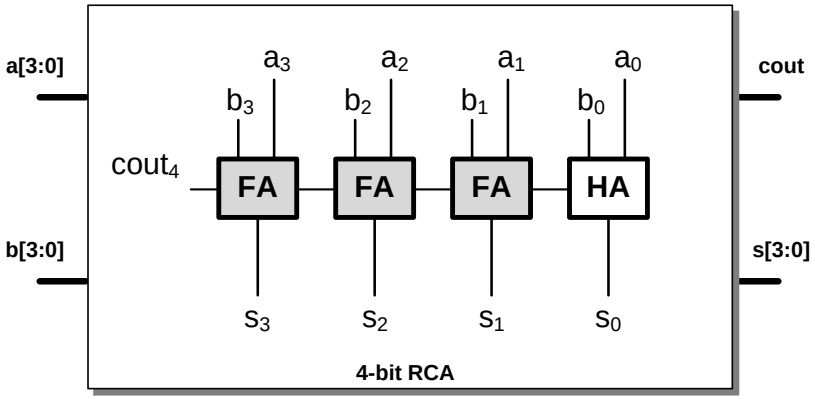
Lab2

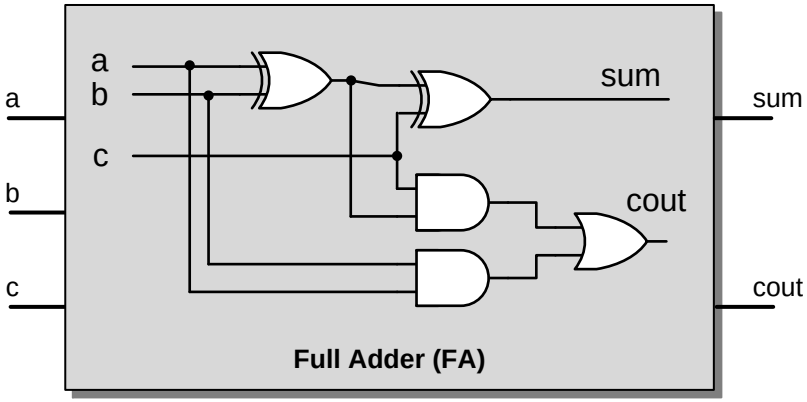

a[3:0]
b[3:0]
cout
s[3:0]
a3
a2
a1
a0
b3
b2
b1
b0
cout4
FA
FA
FA
HA
s3
s2
s1
s0
4-bit RCA

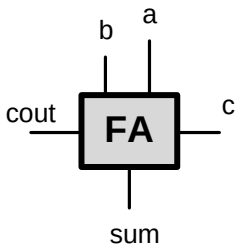
Lab2


a
b
sum
cout
sum
cout
Half Adder (HA)


b
a
cout
HA
sum


a[3:0]
b[3:0]
cout
s[3:0]
a3
a2
a1
a0
b3
b2
b1
b0
cout4
FA
FA
FA
HA
s3
s2
s1
s0
4-bit RCA


a
b
c
sum
cout
a
b
c
sum
cout
Full Adder (FA)


b
a
cout
FA
c
sum

VLSI for DSP

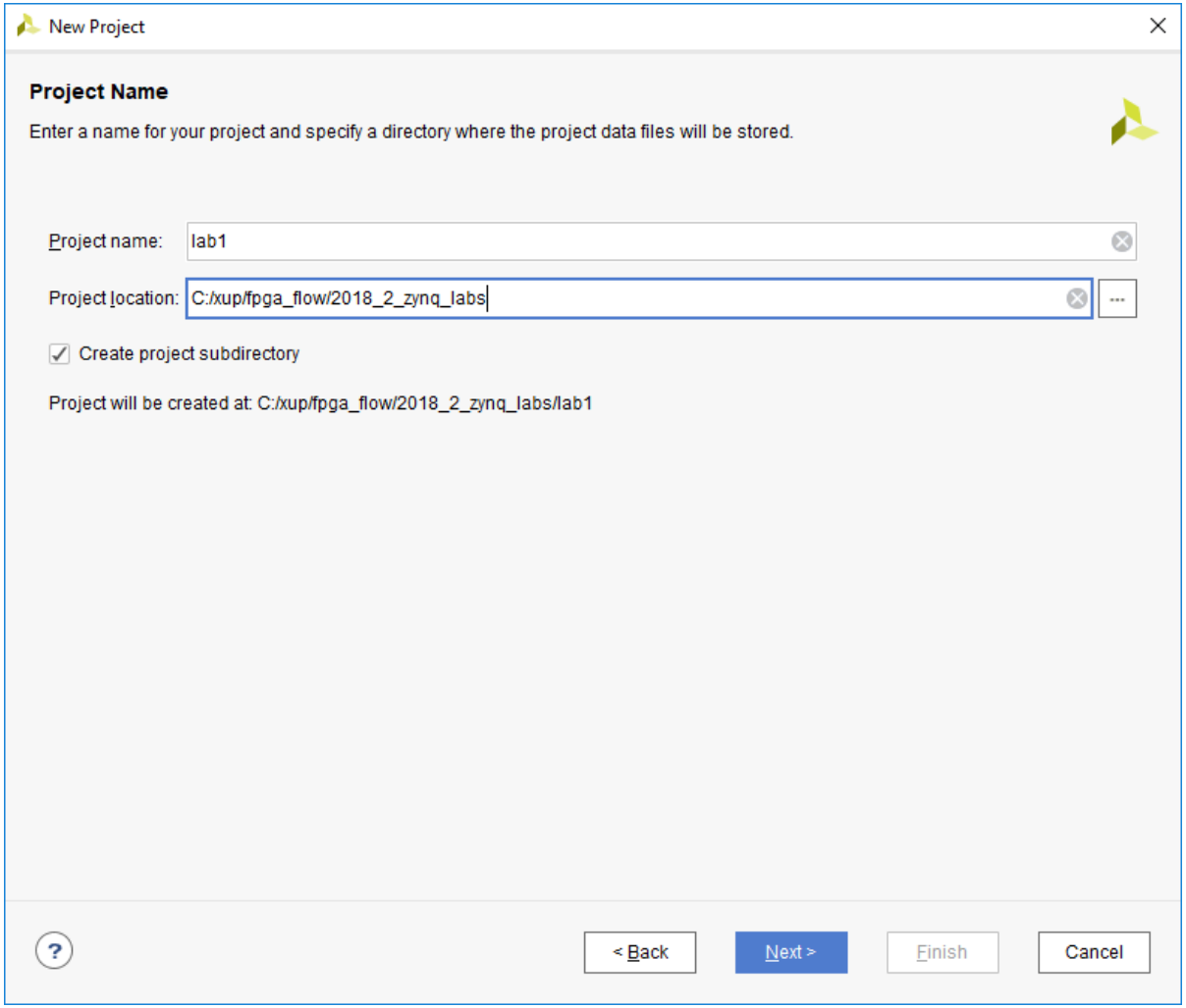
Vivado Design Flow
Getting Started

Department of Electronic Engineering
Chang Guan University
chenyh@mail.cgu.edu.tw

Create a Vivado Project using IDE

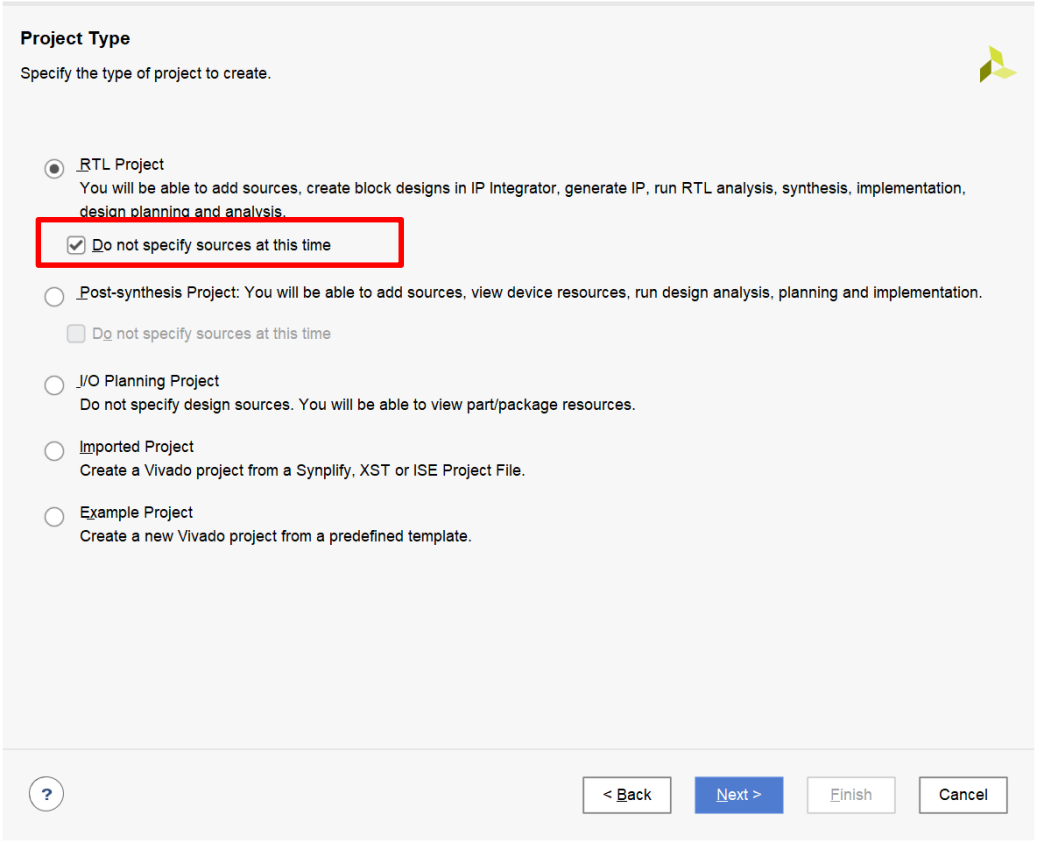
- Launch Vivado and create an empty project.
  1. Open Vivado by selecting **Start > Xilinx Design Tools > Vivado 2018.2**
  2. Click **Create New Project** to start the wizard. You will see *Create A New Vivado Project* dialog box. Click **Next**.
  3. Click the Browse button of the *Project location* 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



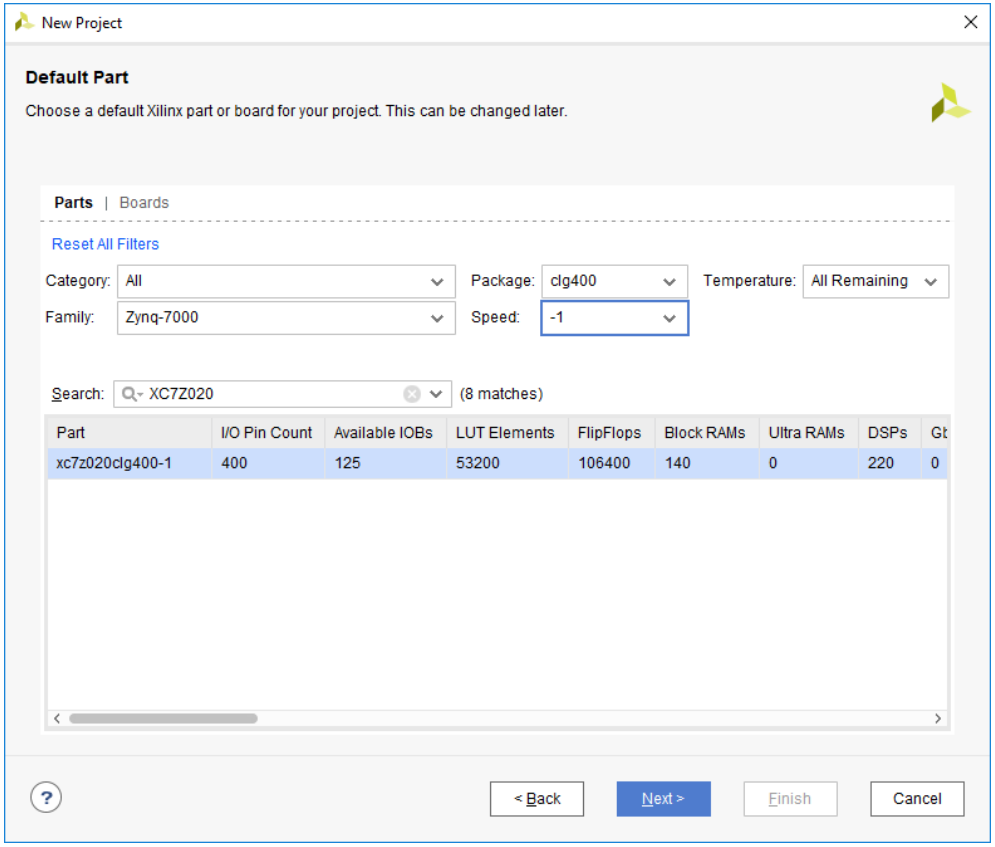
Create a Vivado Project using IDE

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

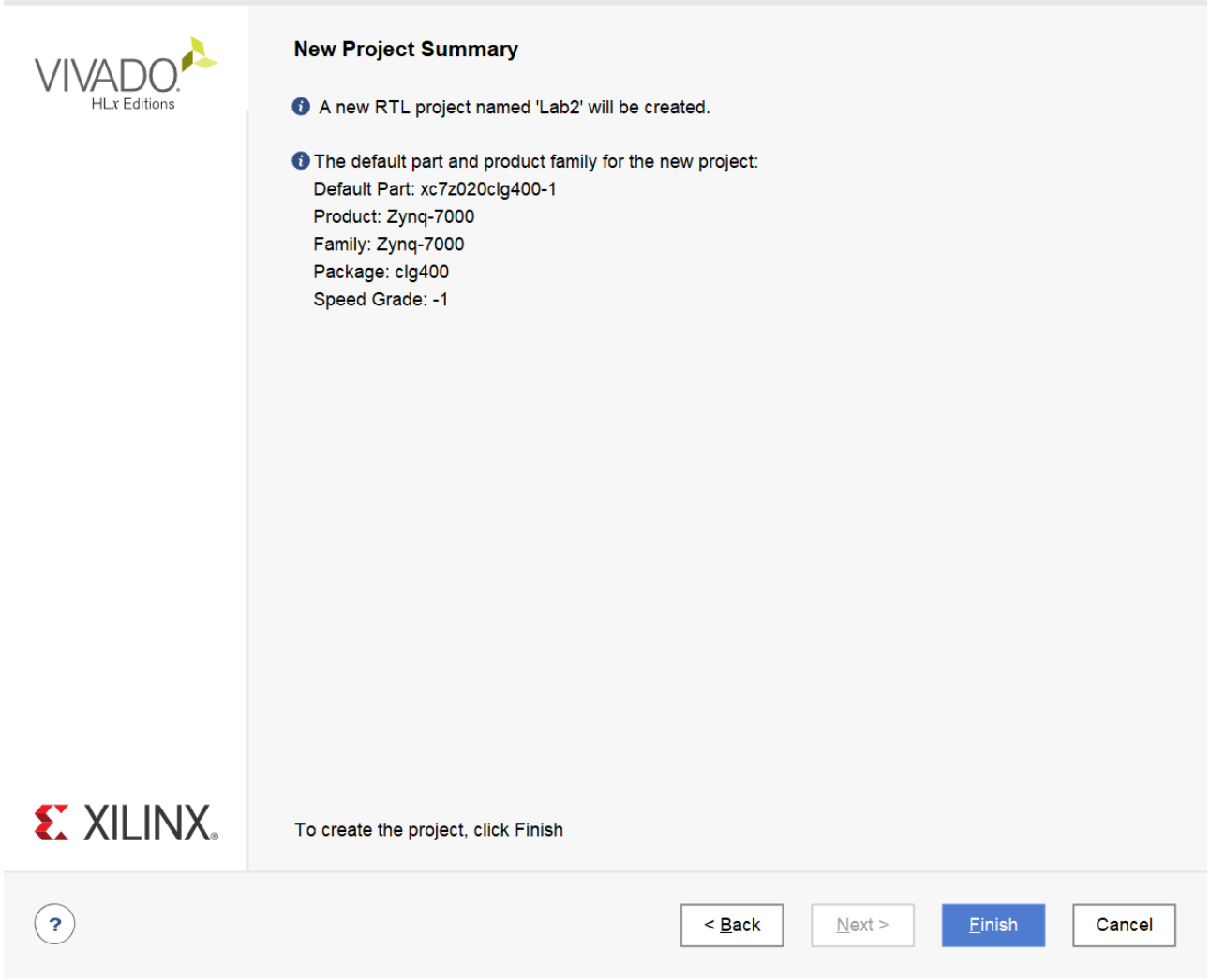


Create a Vivado Project

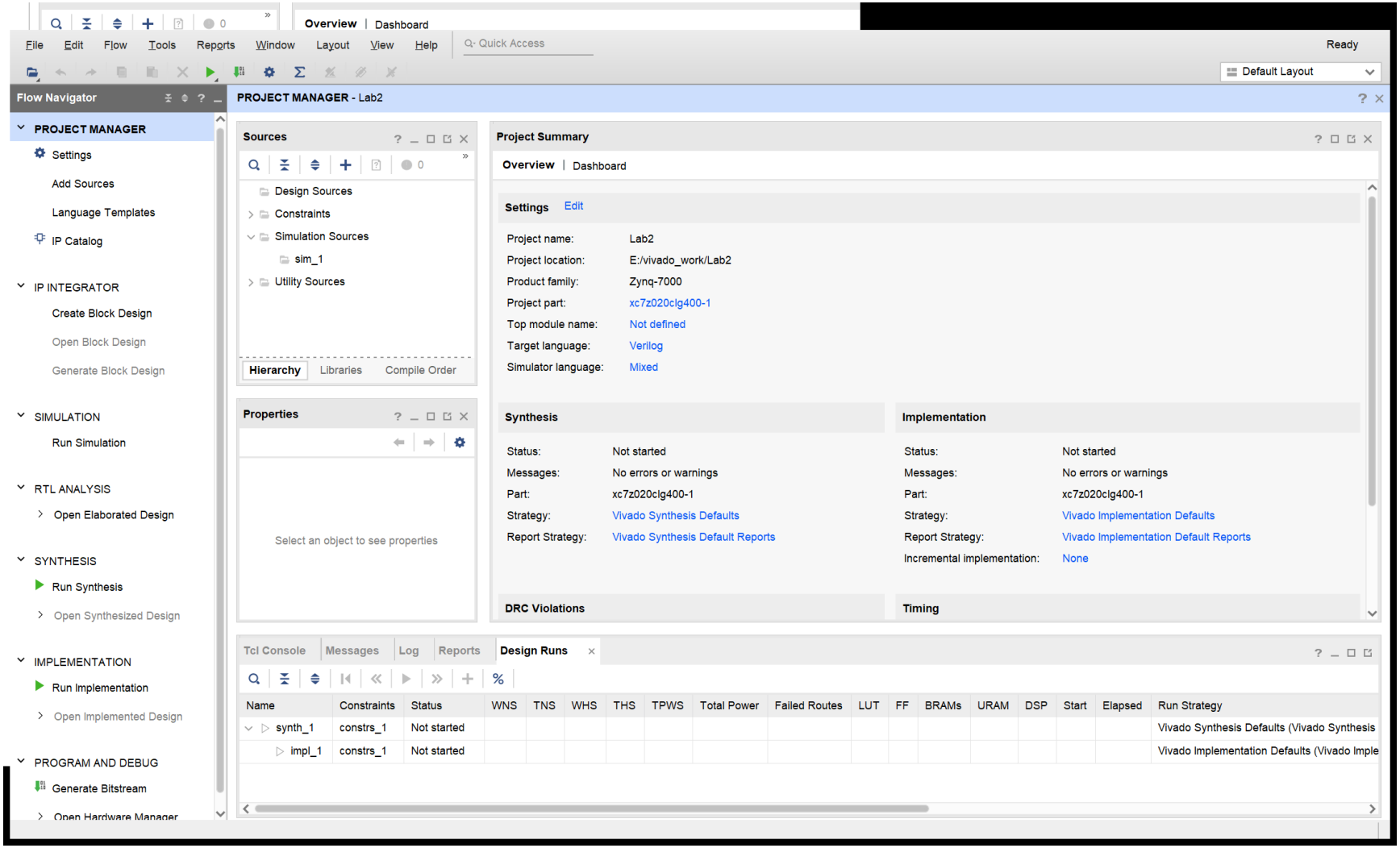
7. In the *Default Part* form, use the **Parts** option and various drop-down fields of the **Filter** section. Select the **XC7Z020clg400-1**.
8. Click **Finish** to create the Vivado project.



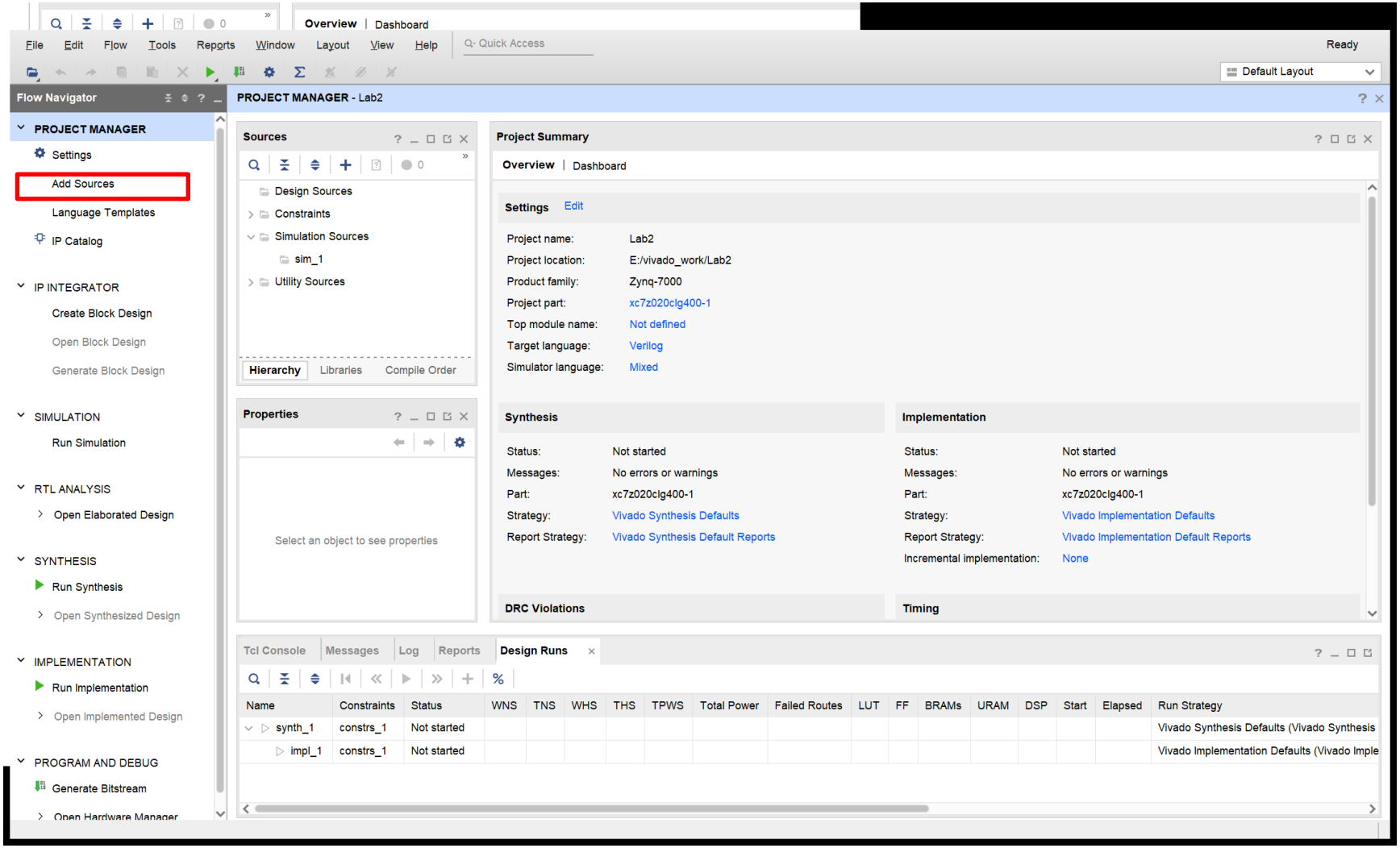
Create a Vivado Project



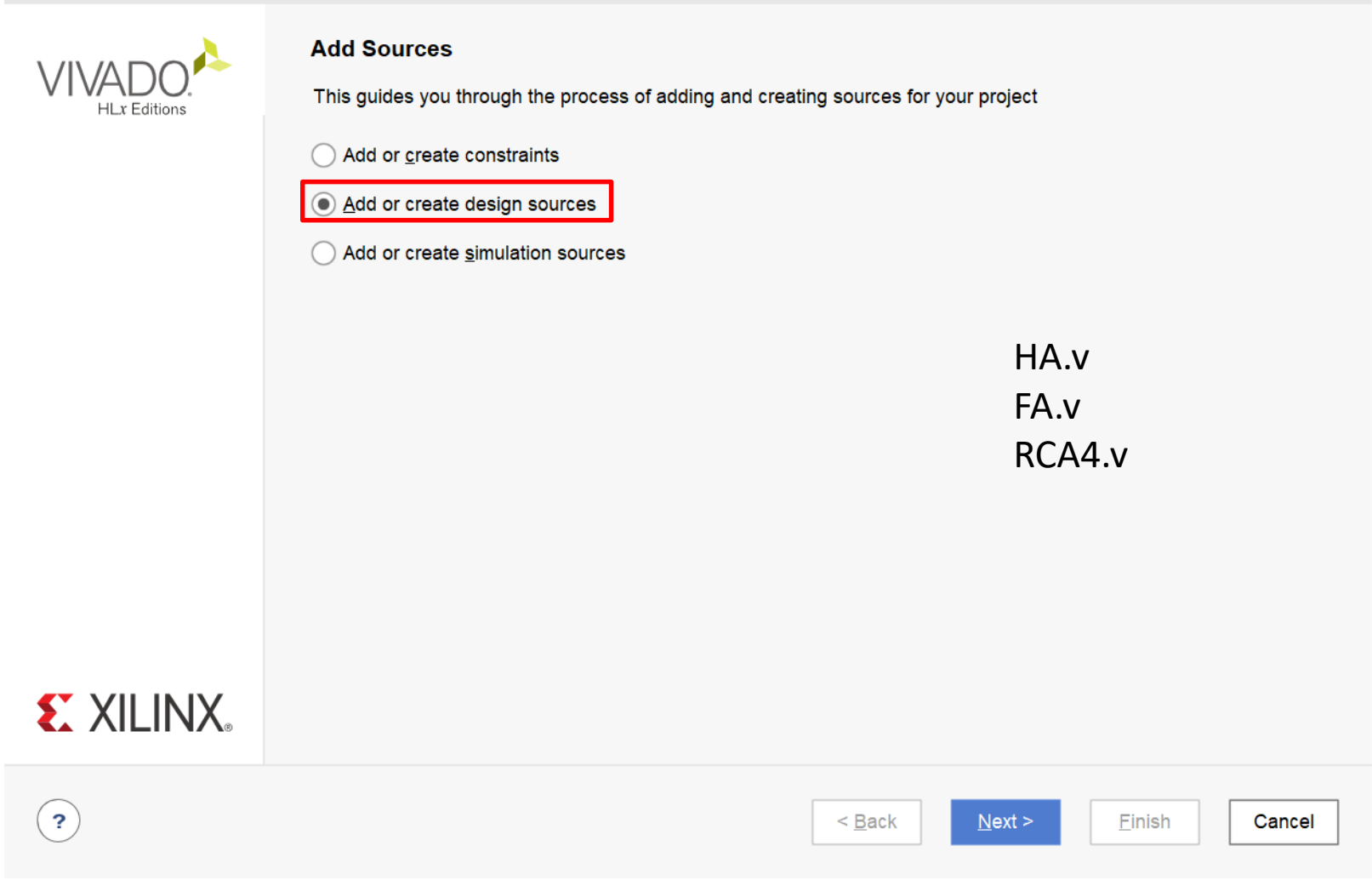
Create a Vivado Project



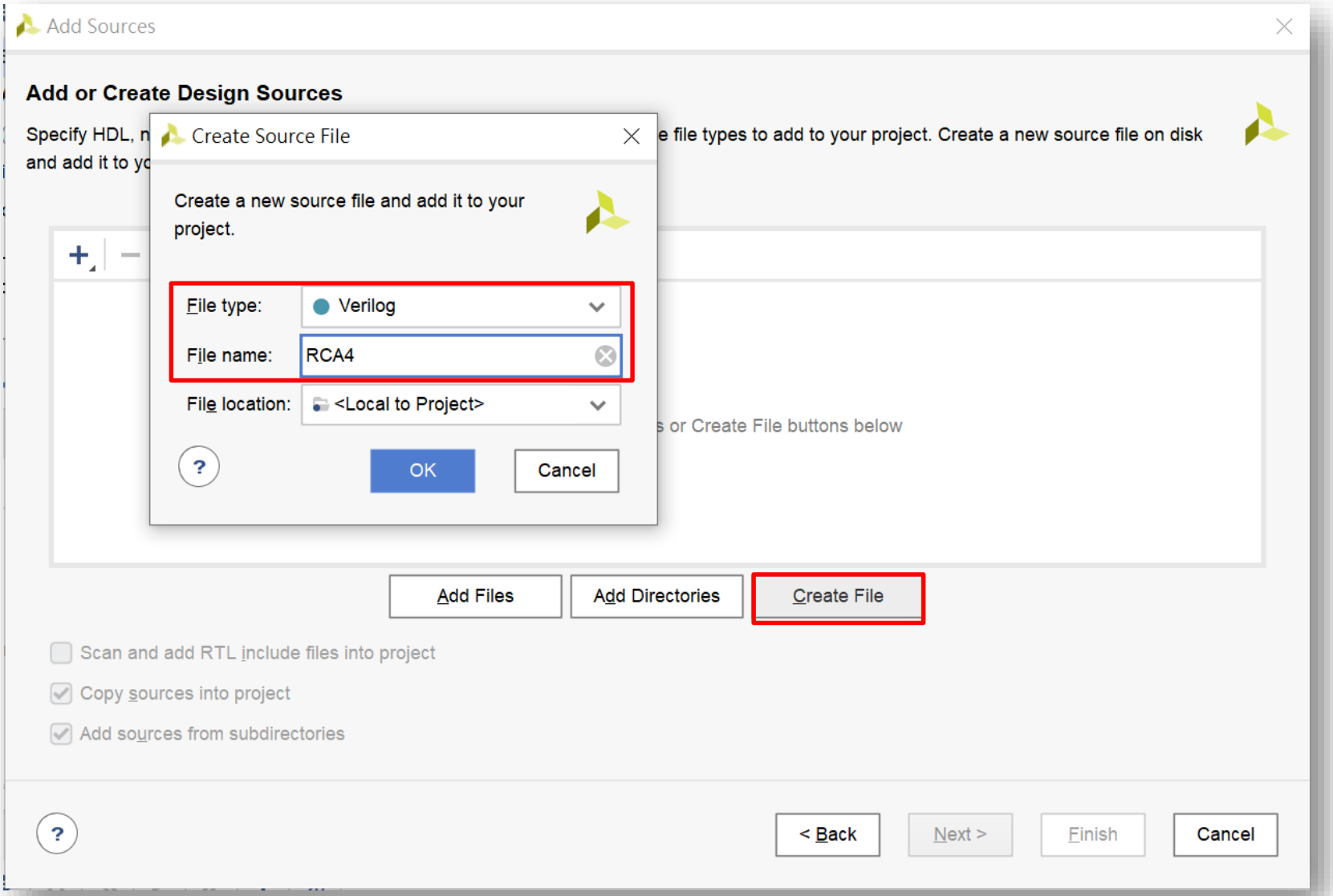
Create Lab2.v



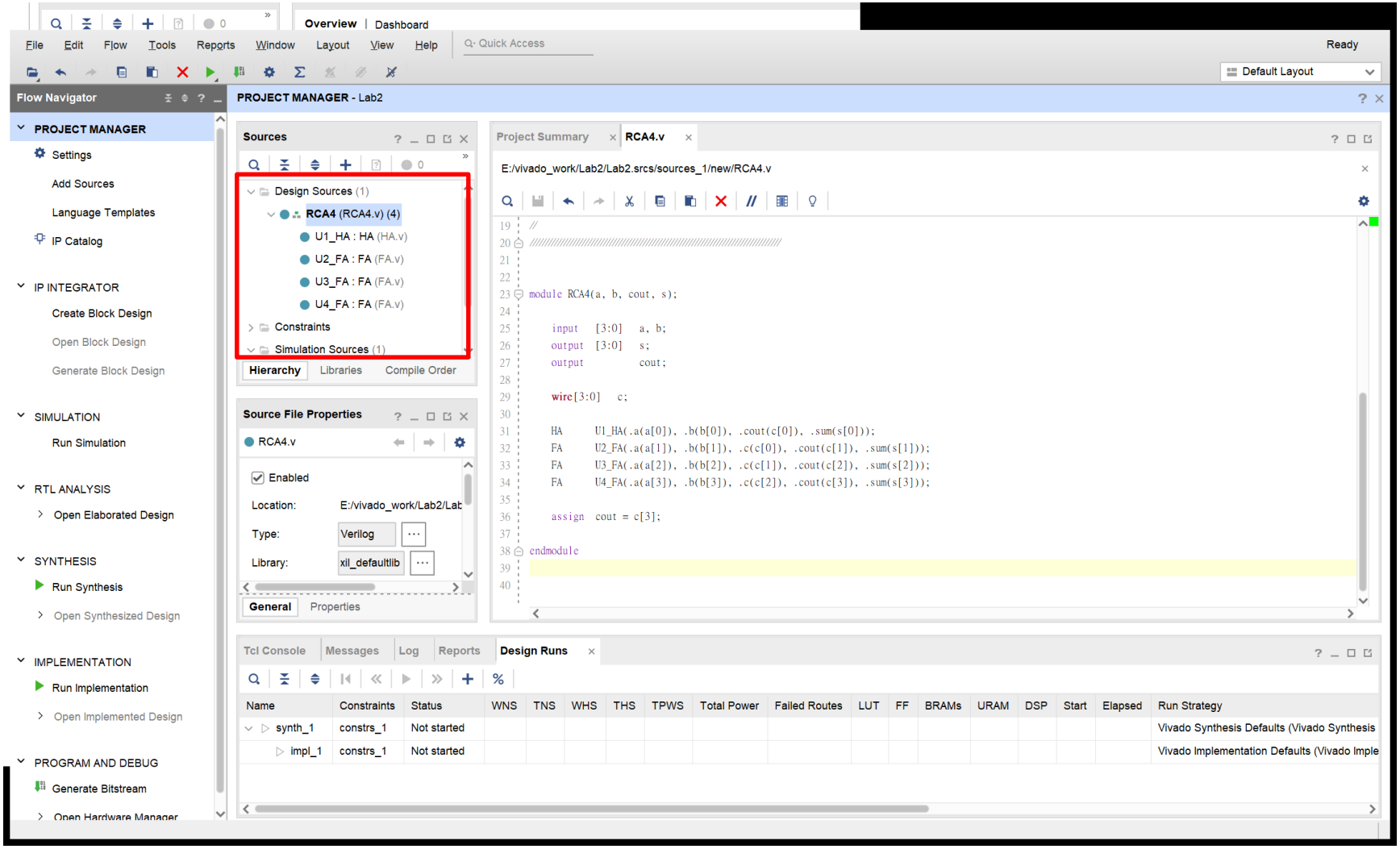
Create RCA4.v



Create RCA4.v



Add HA.v and FA.v



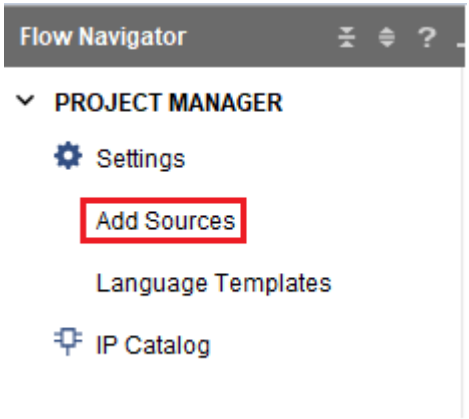
VLSI for DSP

Simulate the Design using the Vivado Simulator

Department of Electronic Engineering
Chang Guan University
chenyh@mail.cgu.edu.tw

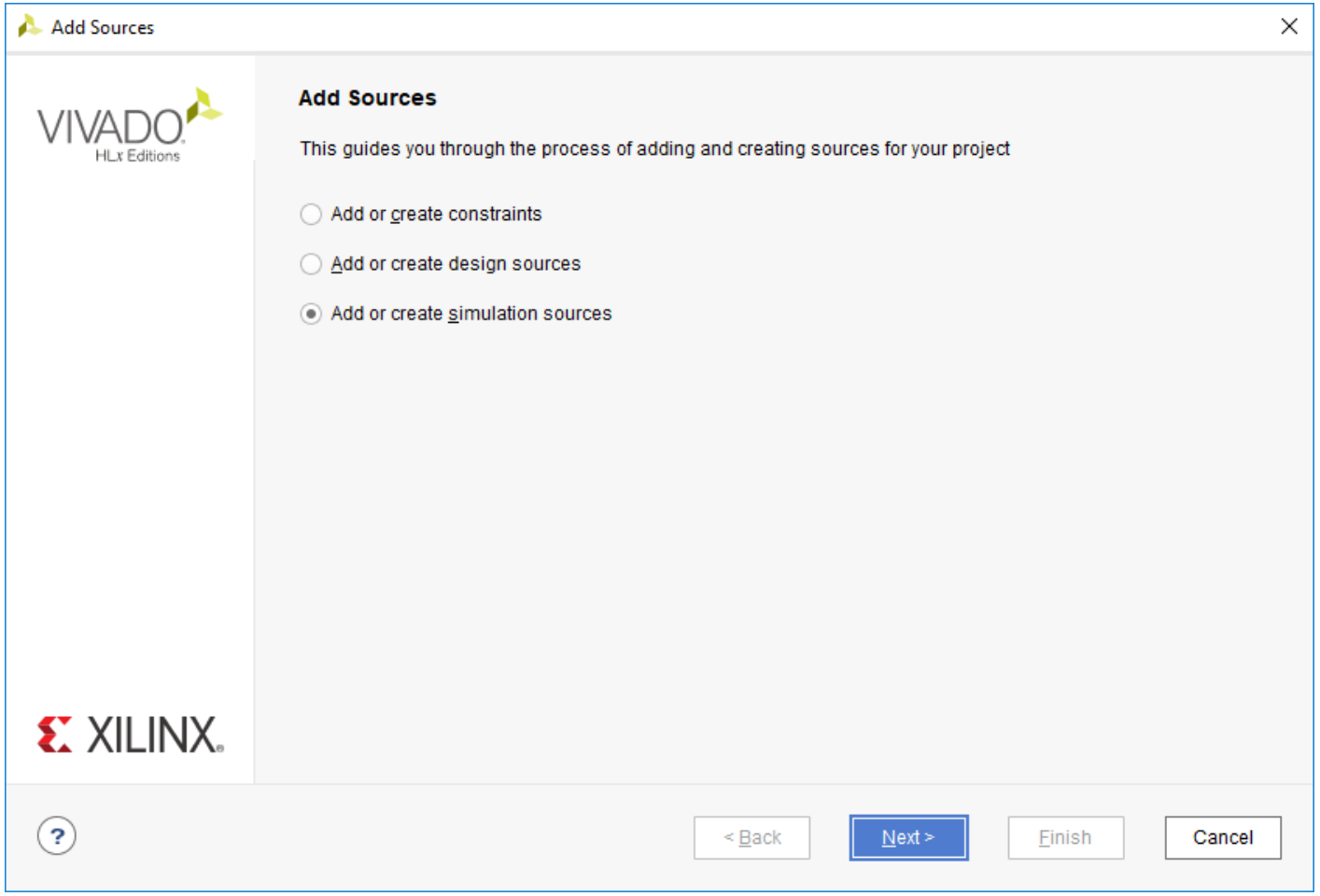
Add the TM_RCA4.v testbench file

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



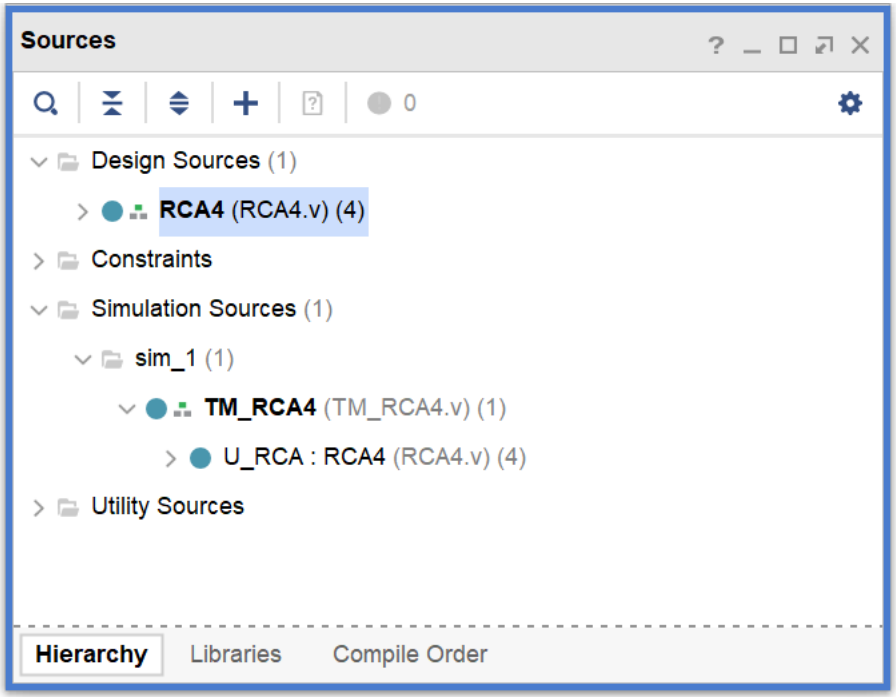
Add the testbench file

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



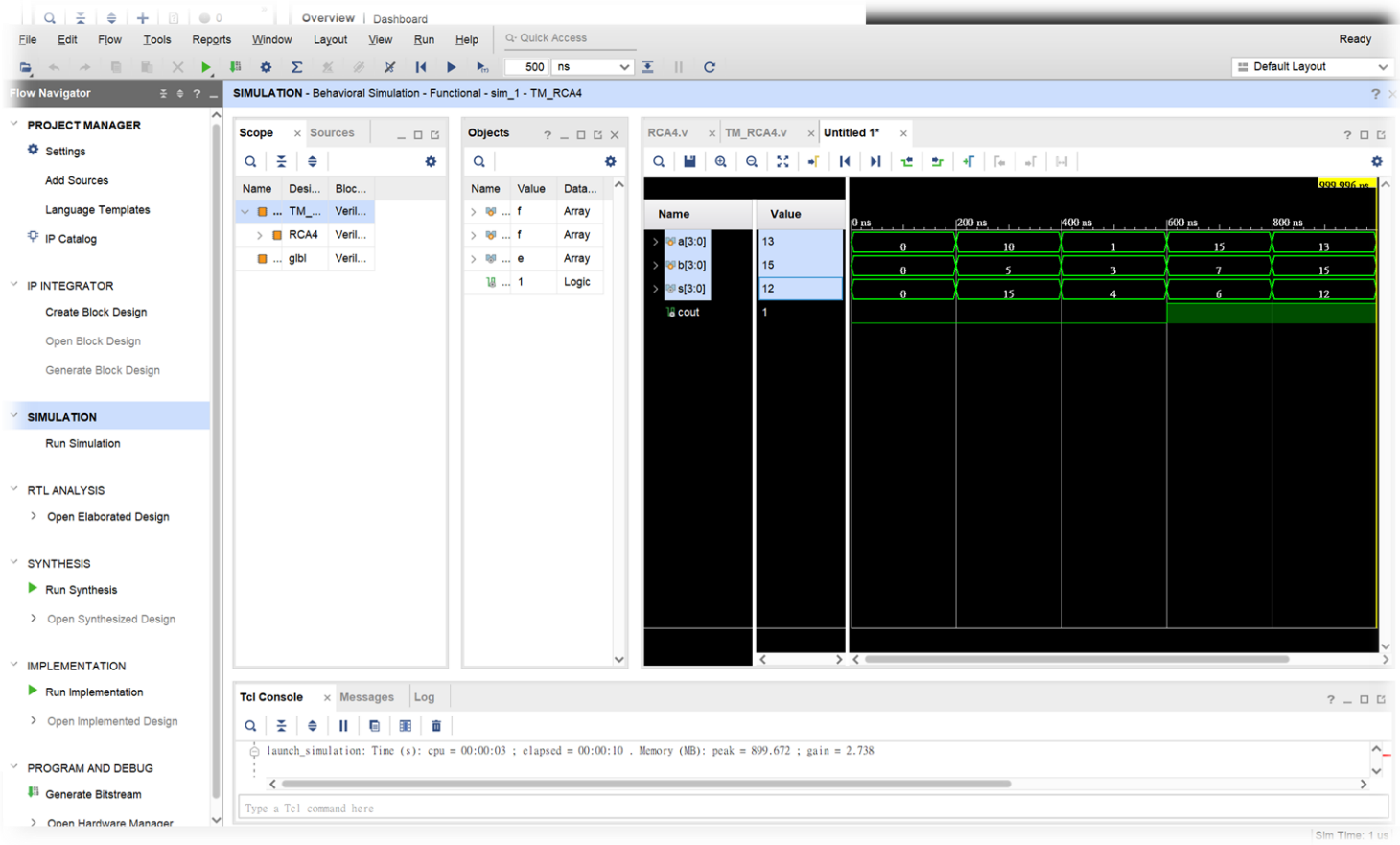
Add the testbench file

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.

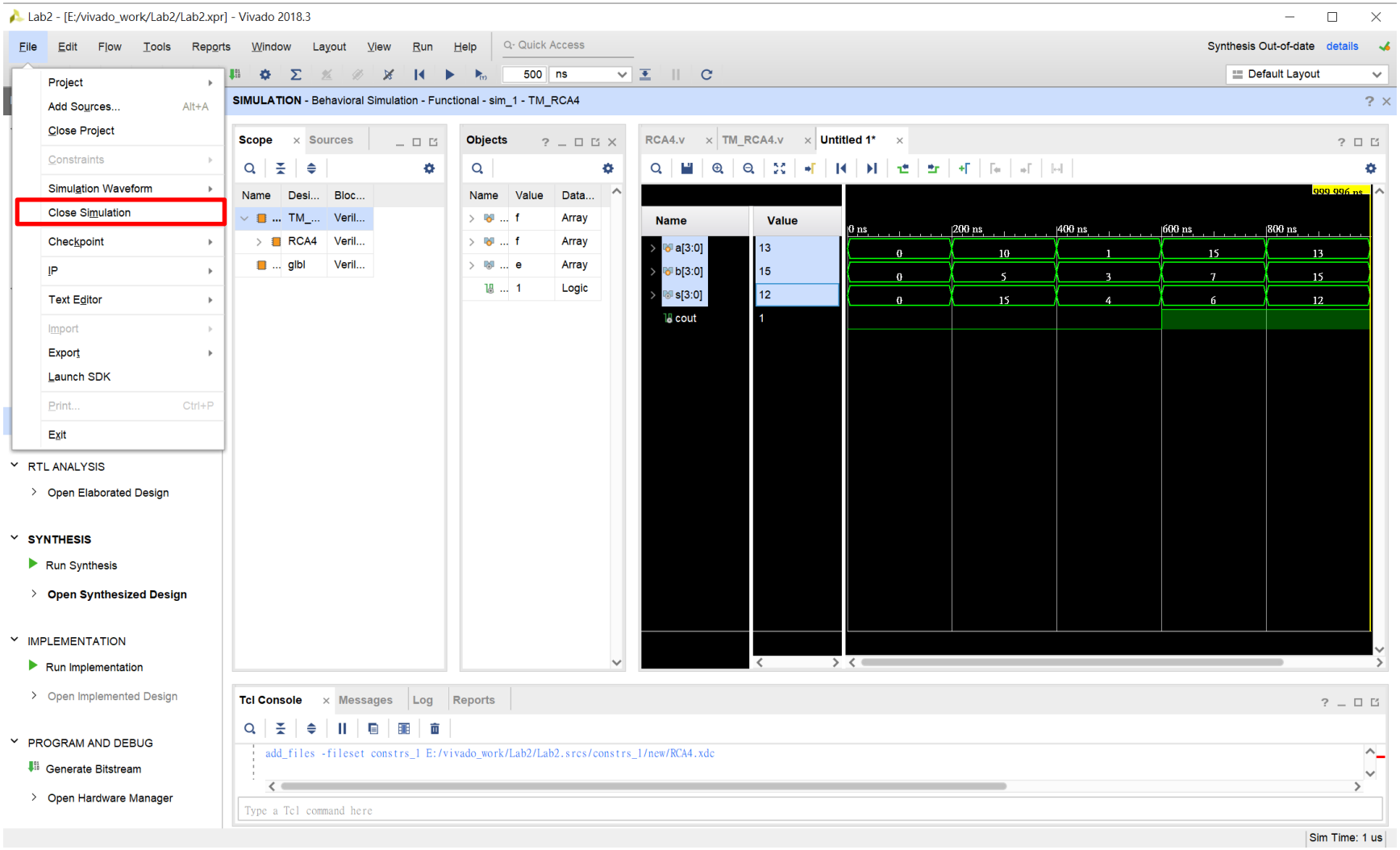


Simulate the design

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



Close Simulation



VLSI for DSP

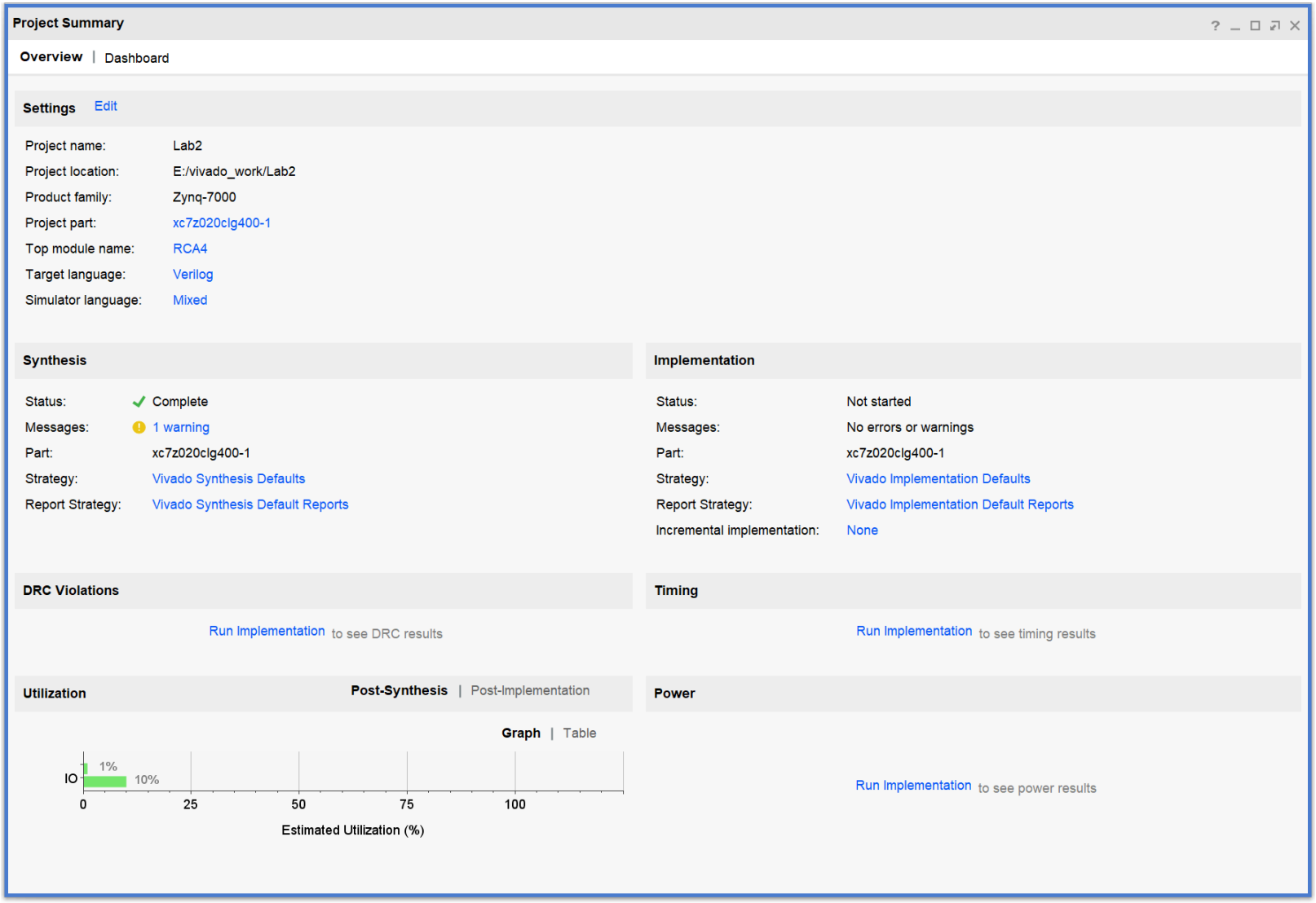
Synthesize the Design

Department of Electronic Engineering
Chang Guan University
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



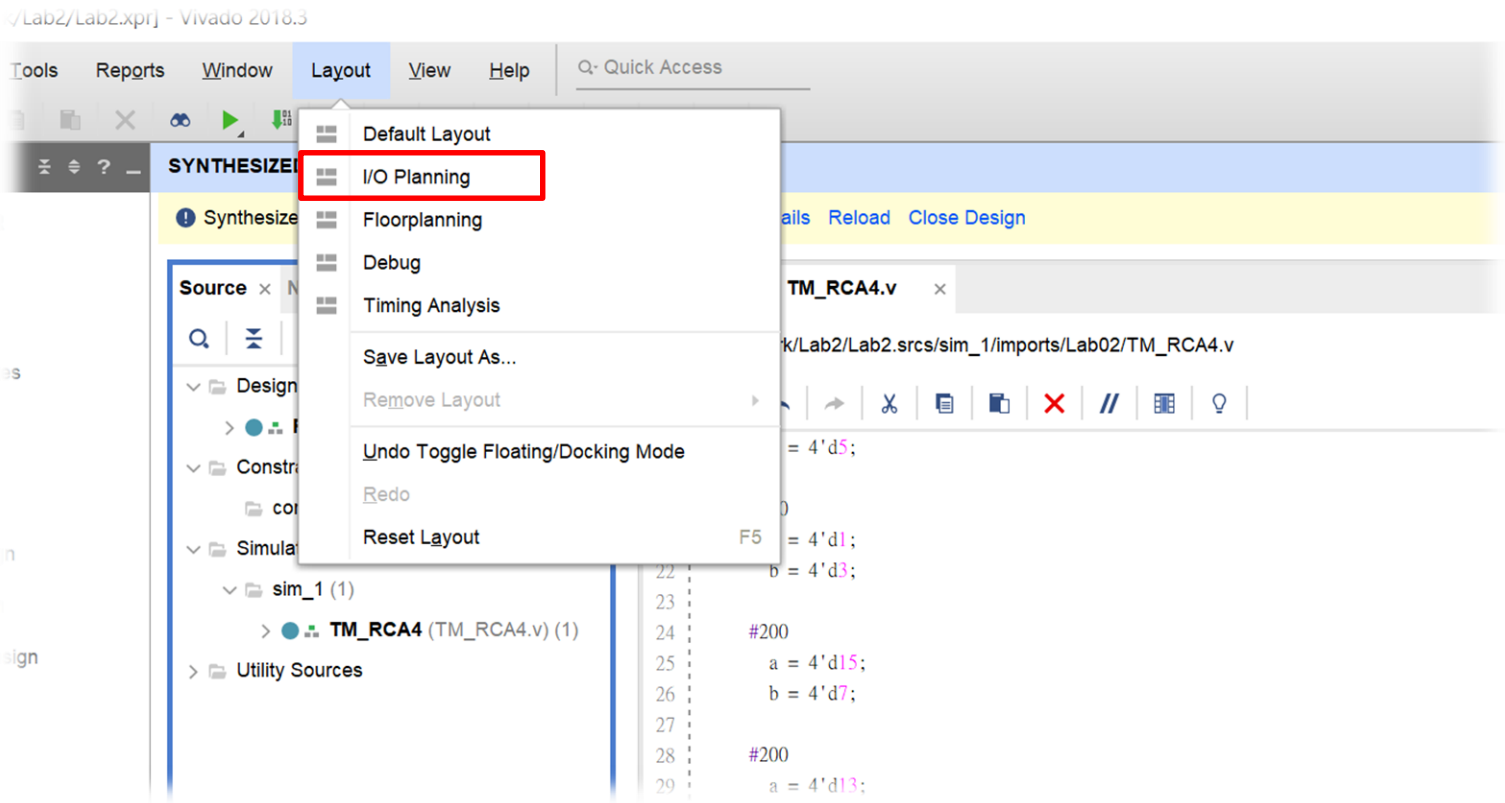
VLSI for DSP

Implement the Design

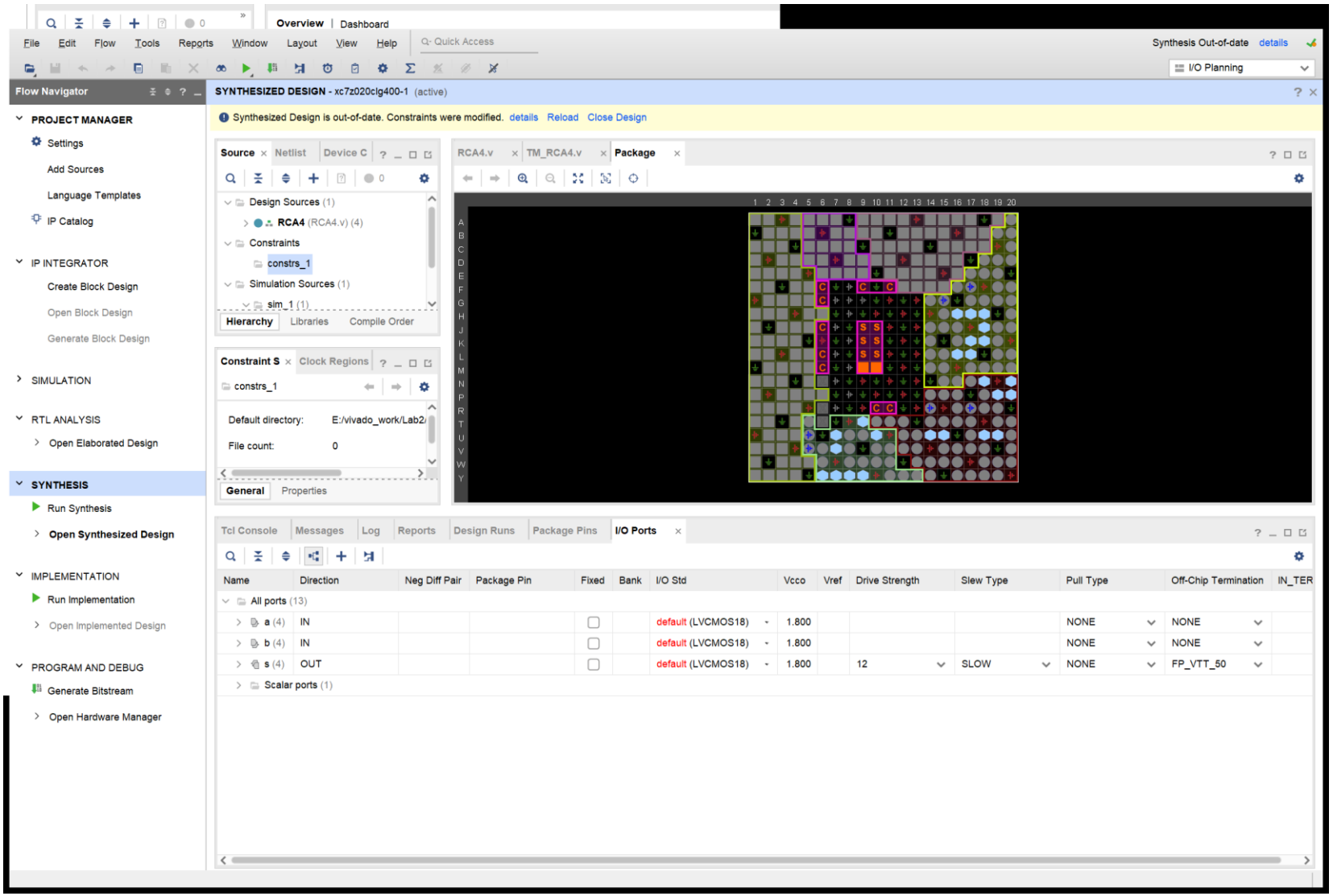
Department of Electronic Engineering
Chang Guan University
chenyh@mail.cgu.edu.tw

I/O Planning

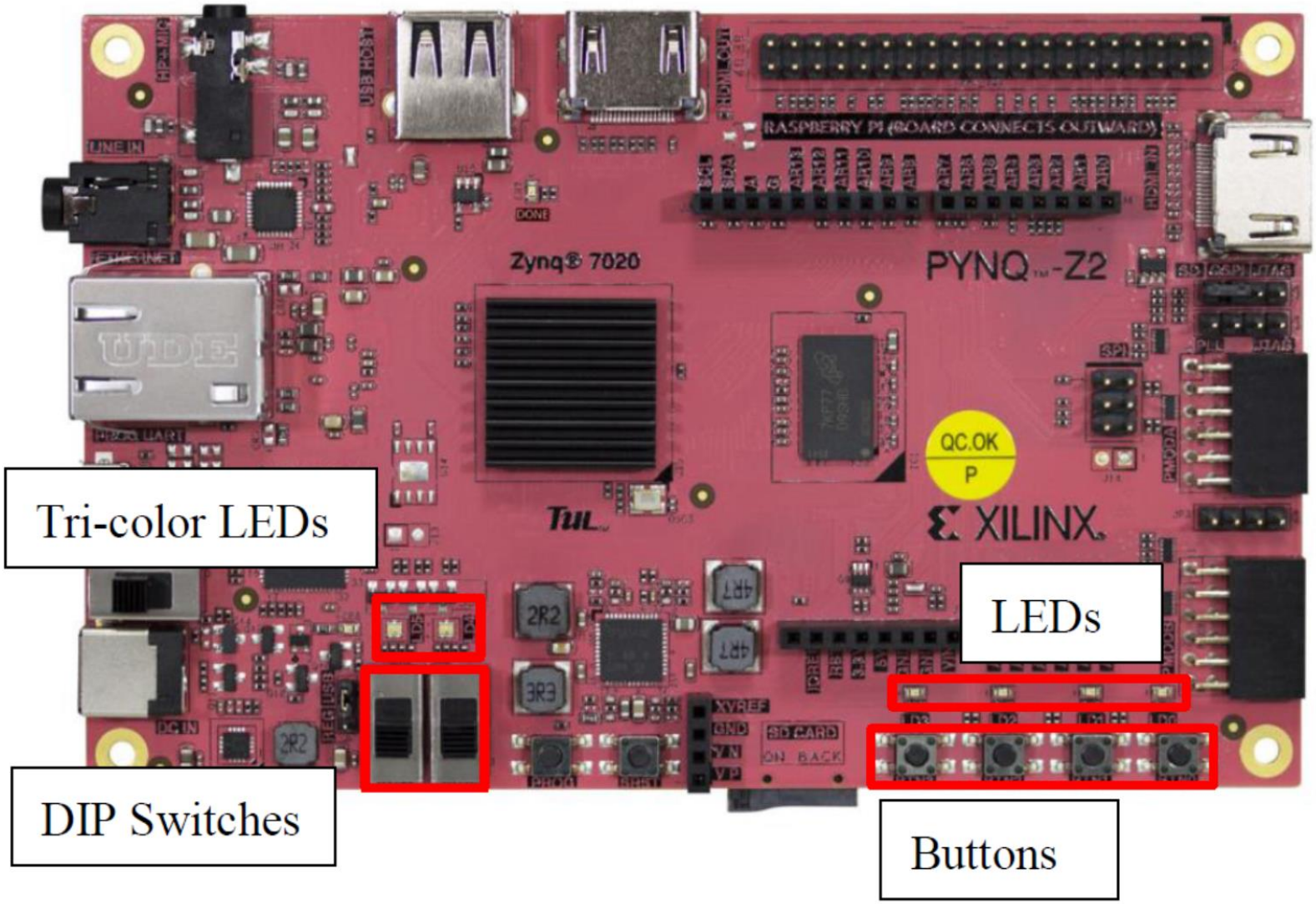
- Layout → I/O Planning



I/O Planning


Overview | Dashboard
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Synthesis Out-of-date details
I/O Planning
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 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 Clock Regions
constrs_1
Default directory: E:/vivado_work/Lab2/
File count: 0
General Properties
RCA4.v TM_RCA4.v Package
Tcl Console Messages Log Reports Design Runs Package Pins 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_TER
All ports (13)
a (4) IN default (LVCMOS18) 1.800 NONE NONE
b (4) IN default (LVCMOS18) 1.800 NONE NONE
s (4) OUT default (LVCMOS18) 1.800 12 SLOW NONE FP_VTT_50
Scalar ports (1)

I/O Planning


Tri-color LEDs
DIP Switches
LEDs
Buttons
Zynq® 7020
PYNQ™-Z2
XILINX
RASPBERRY PI (BOARD CONNECTS OUTWARD)

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.srcs/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

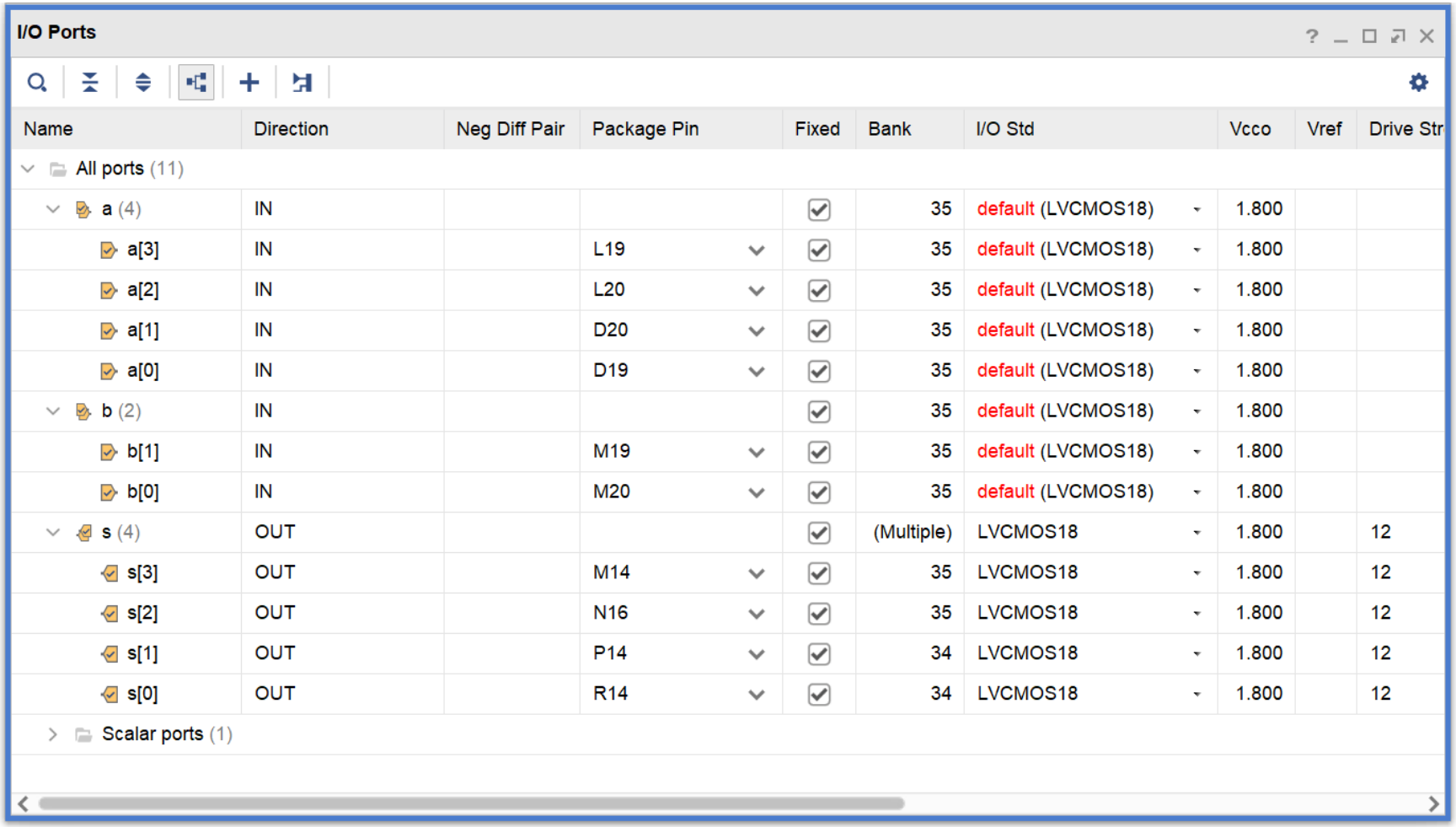
Table 10 Dip switch PL pin mapping

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			☑	35	default (LVCMOS18)	1.800		
a[3]	IN		L19	☑	35	default (LVCMOS18)	1.800		
a[2]	IN		L20	☑	35	default (LVCMOS18)	1.800		
a[1]	IN		D20	☑	35	default (LVCMOS18)	1.800		
a[0]	IN		D19	☑	35	default (LVCMOS18)	1.800		
b (2)	IN			☑	35	default (LVCMOS18)	1.800		
b[1]	IN		M19	☑	35	default (LVCMOS18)	1.800		
b[0]	IN		M20	☑	35	default (LVCMOS18)	1.800		
s (4)	OUT			☑	(Multiple)	LVCMOS18	1.800		12
s[3]	OUT		M14	☑	35	LVCMOS18	1.800		12
s[2]	OUT		N16	☑	35	LVCMOS18	1.800		12
s[1]	OUT		P14	☑	34	LVCMOS18	1.800		12
s[0]	OUT		R14	☑	34	LVCMOS18	1.800		12
Scalar ports (1)									

xdc file

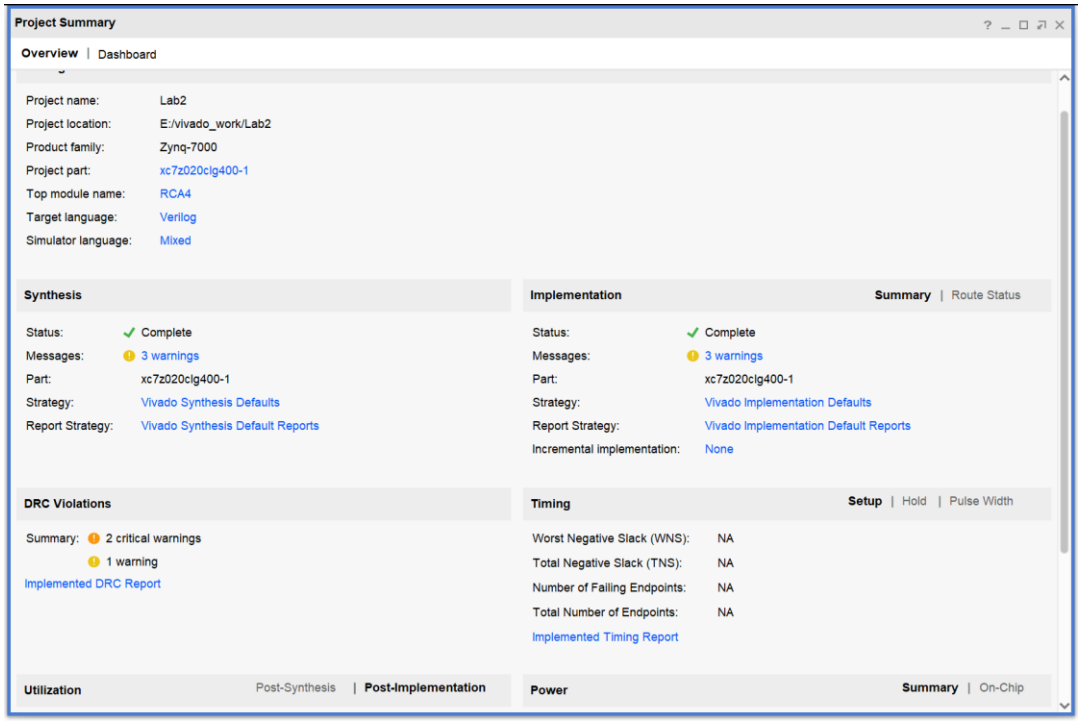
RCA4.xdc *

E:/vivado_work/Lab2/Lab2.srcs/constrs_1/new/RCA4.xdc

```
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

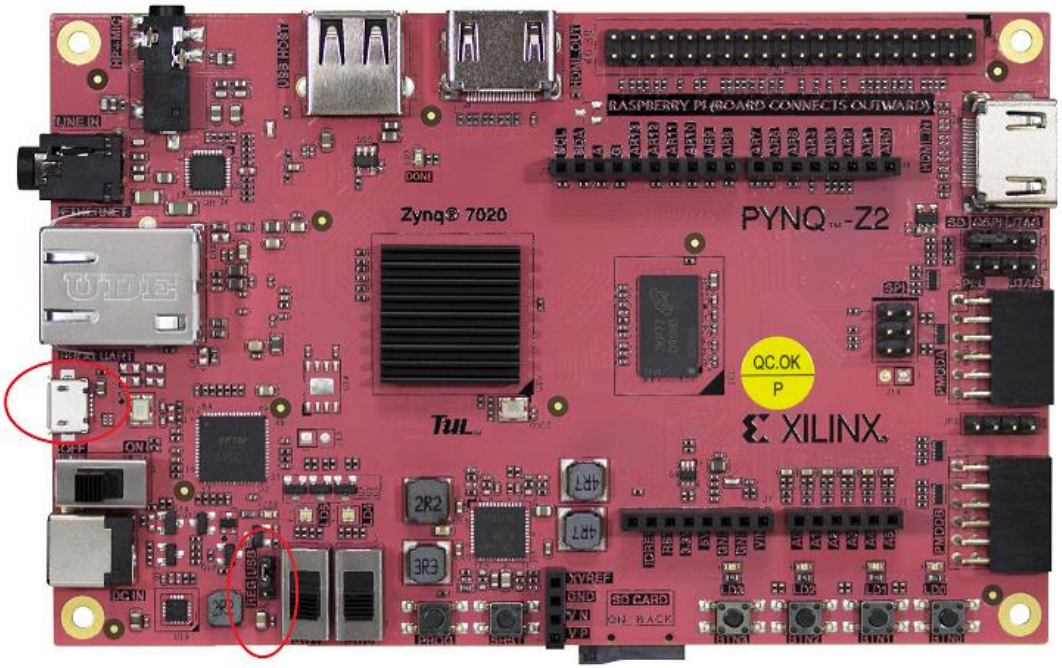
Generate the Bitstream and Verify Functionality

Department of Electronic Engineering
Chang Guan University
chenyh@mail.cgu.edu.tw

Generate the Bitstream

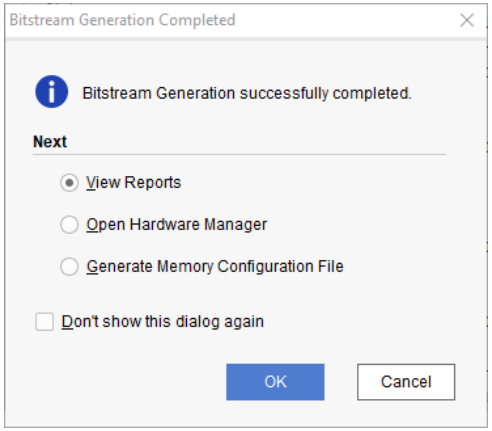
- **Connect the board and power it ON. Generate the bitstream, open a hardware session, and program the FPGA.**

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.



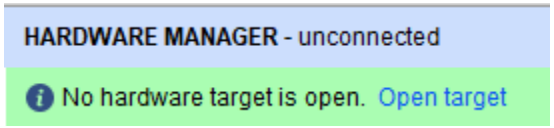
Generate the Bitstream

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.



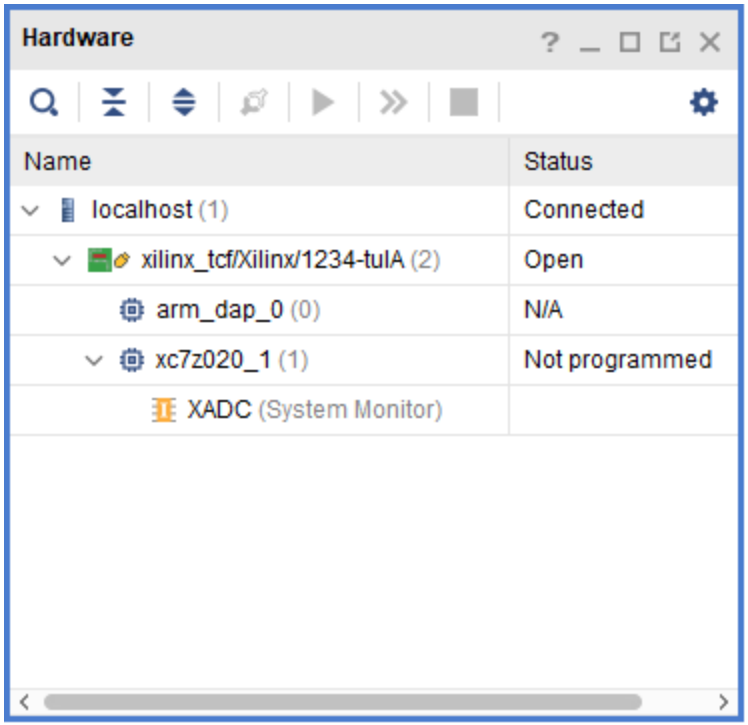
Generate the Bitstream

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.



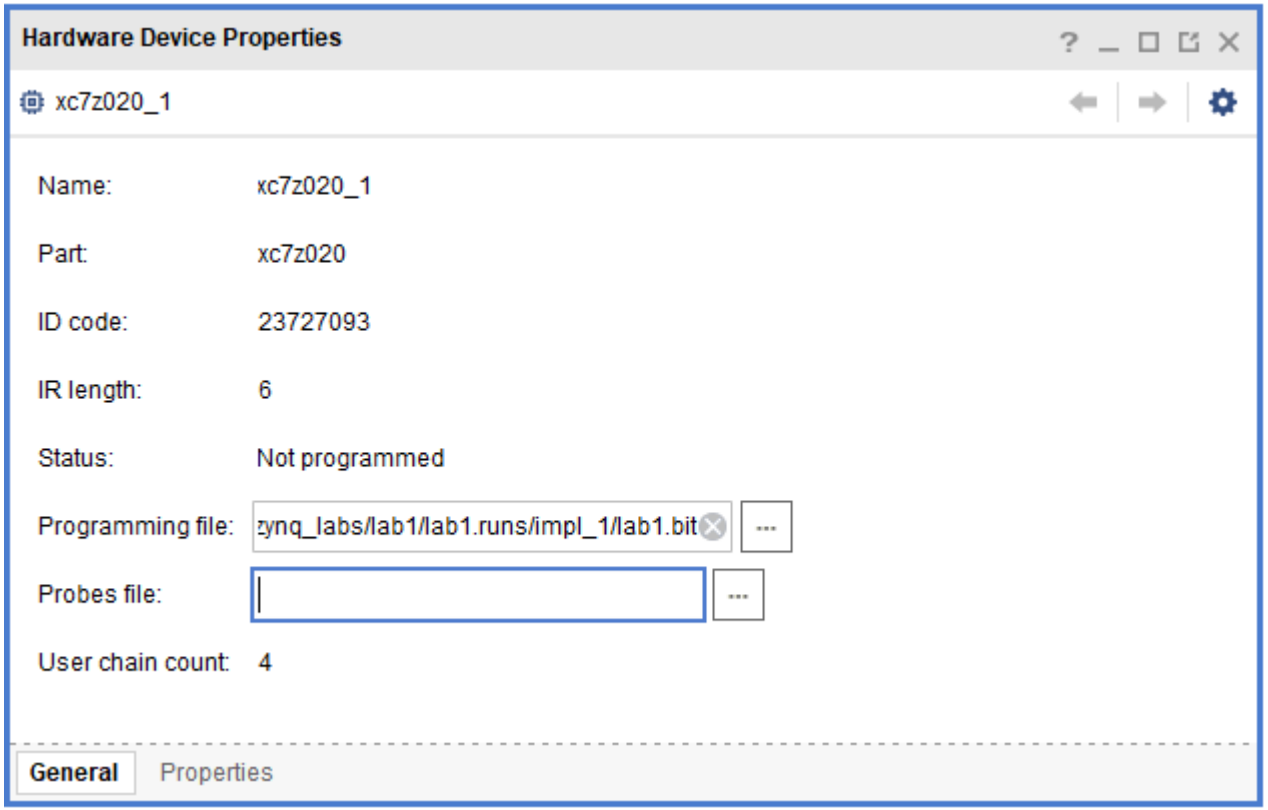
Download

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.



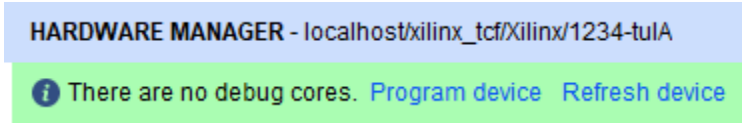
Download

- 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.*



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**.