

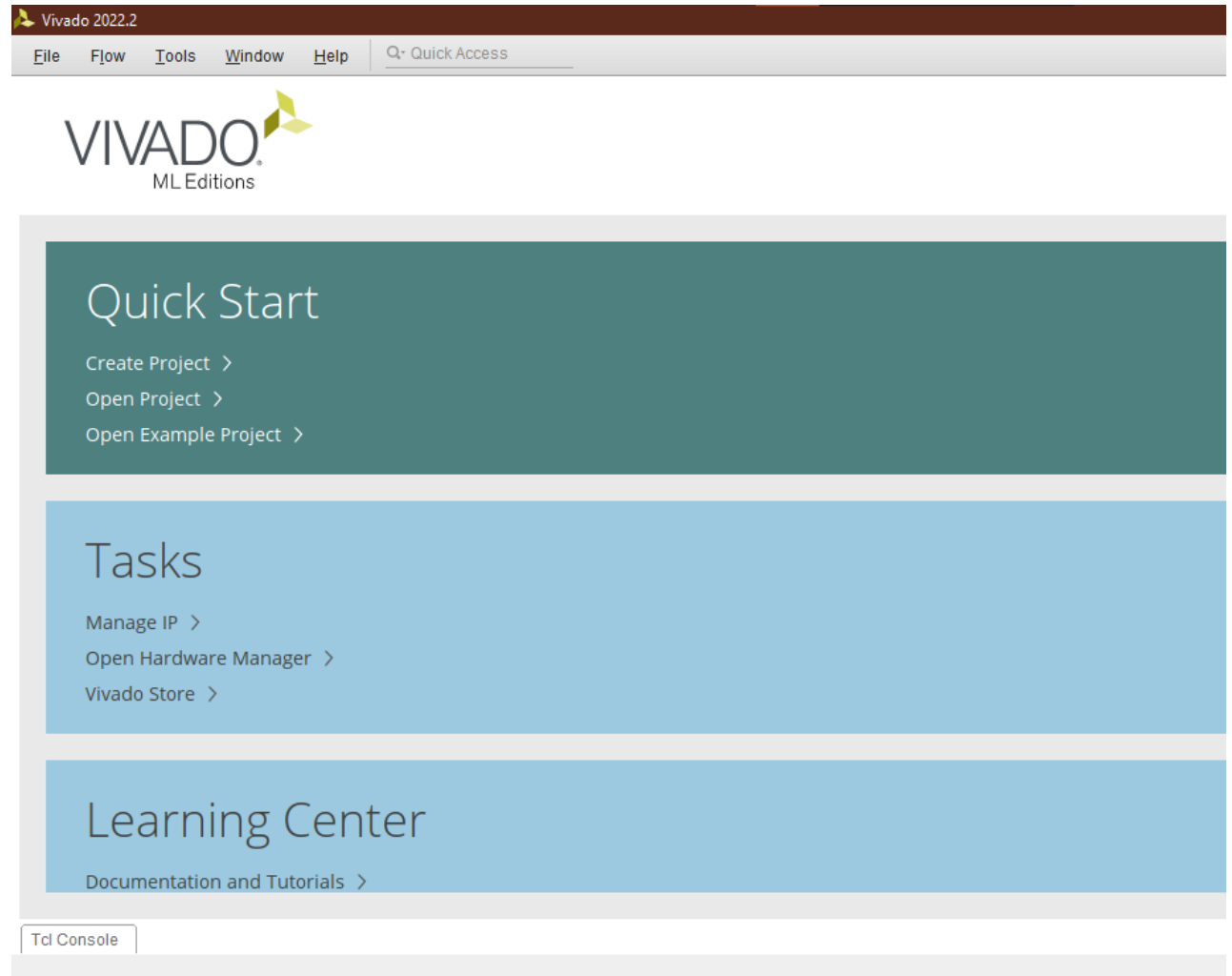
PYNQ GS GPIO
WITH
KRIA KV260

Table of Contents

- Hardware design
- Using the GPIO PYNQ class

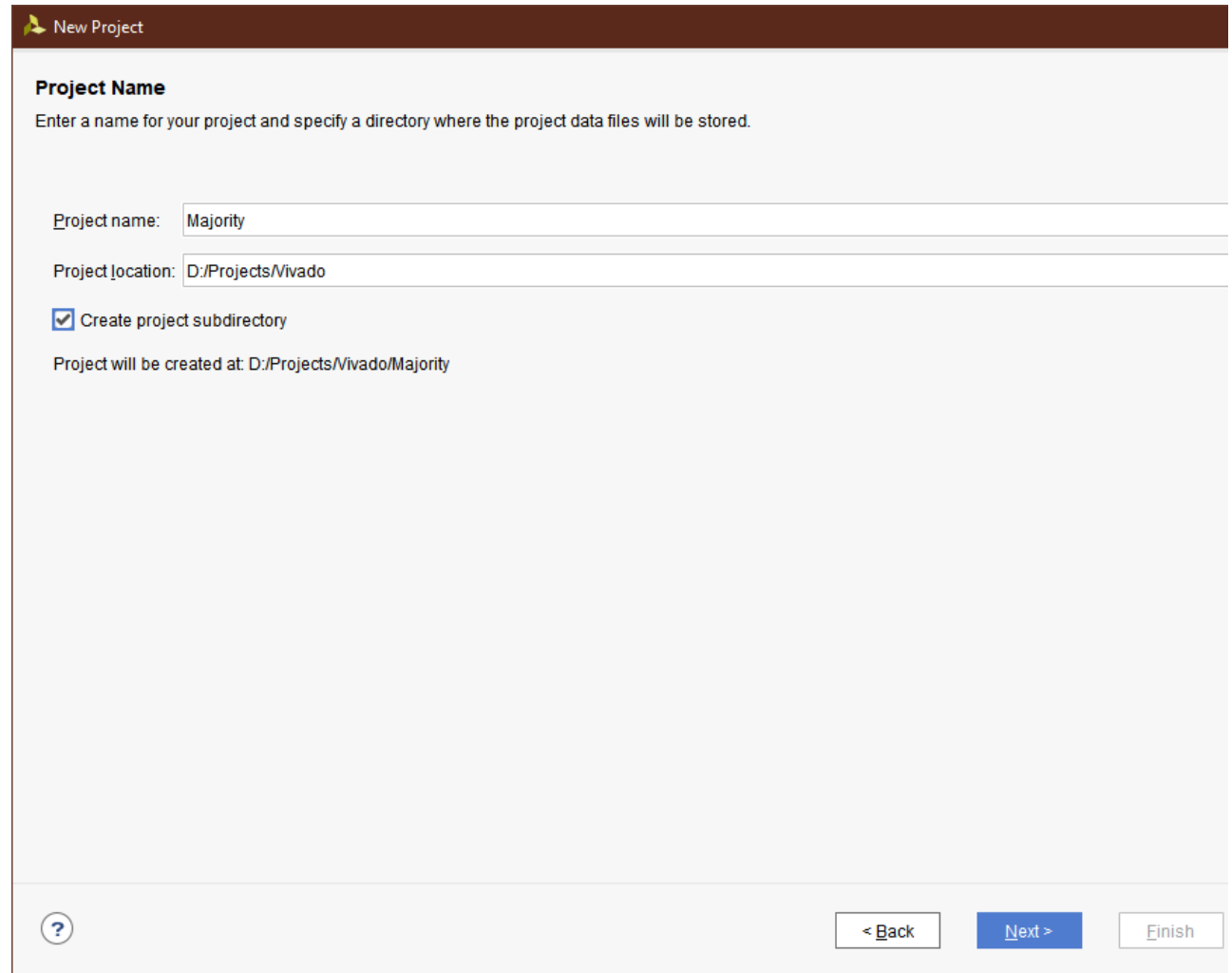
Hardware design Project creation

- Open Vivado
- Create Project



Hardware design Project creation

- Open Vivado
- Create Project
- Choose your project name and directory



New Project

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

Project name: Majority

Project location: D:/Projects/Vivado

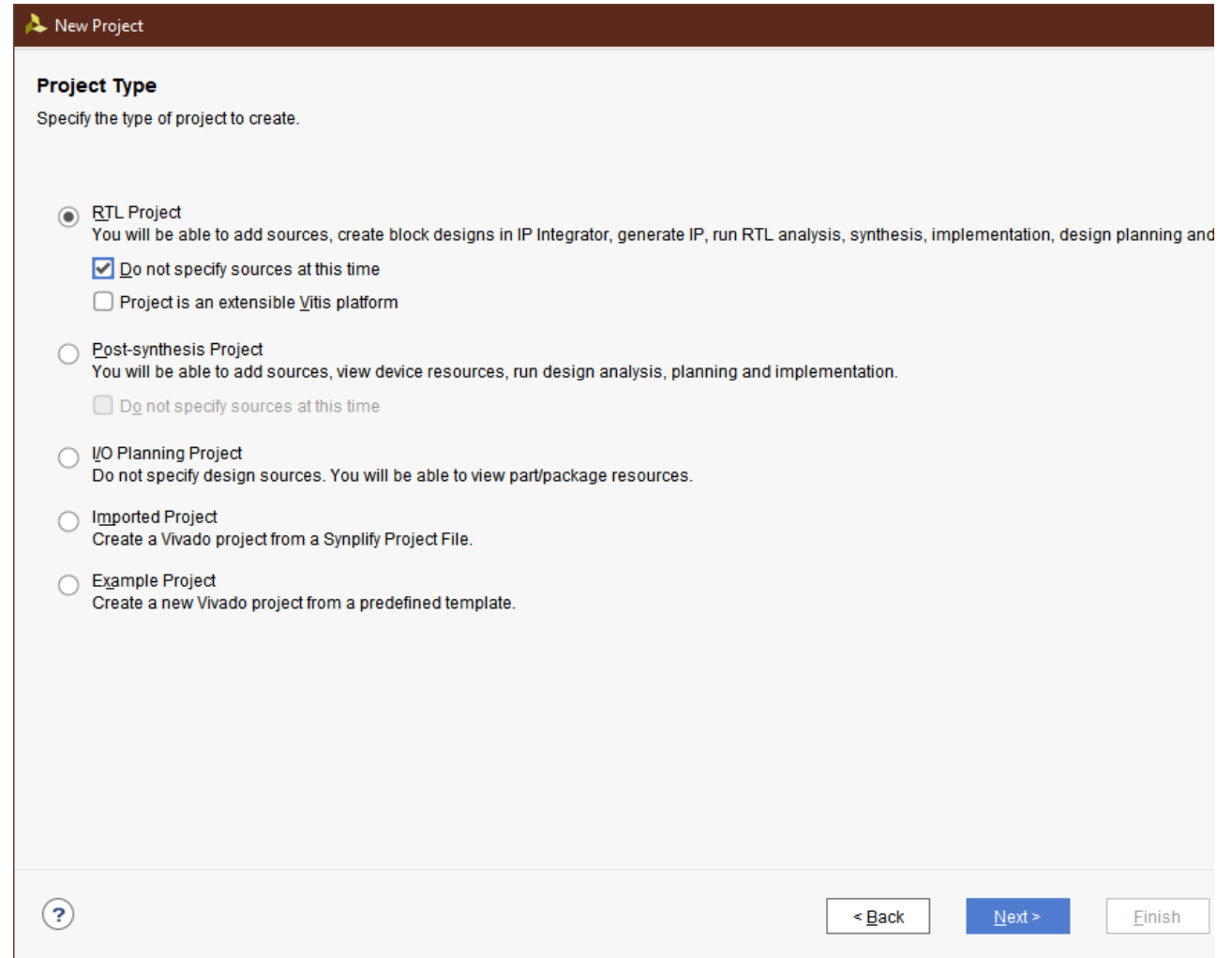
☒ Create project subdirectory

Project will be created at: D:/Projects/Vivado/Majority

? < Back Next > Finish

Hardware design Project creation

- Open Vivado
- Create Project
- Choose your project name and directory
- Choose RTL Project and not specify sources for now



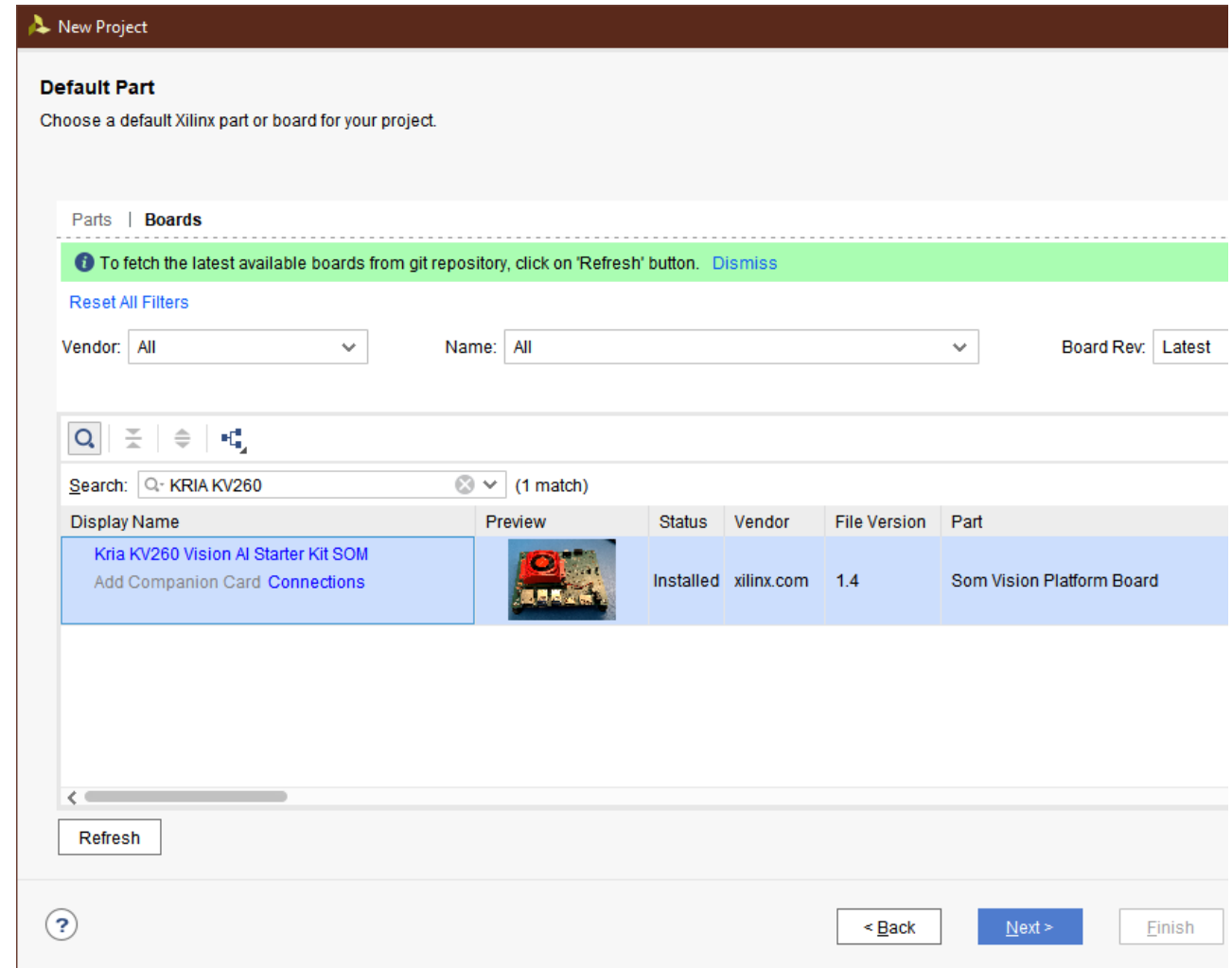
The screenshot shows the 'New Project' dialog box in Vivado. The title bar is dark red with a yellow Vivado logo and the text 'New Project'. The main area is light gray and titled 'Project Type' with the instruction 'Specify the type of project to create.' Below this, there are five radio button options:

- ☒ **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
☒ Do not specify sources at this time
☐ Project is an extensible Vitis platform
- ☐ **Post-synthesis Project**
You will be able to add sources, view device resources, run design analysis, planning and implementation.
☐ Do not specify sources at this time
- ☐ **I/O Planning Project**
Do not specify design sources. You will be able to view part/package resources.
- ☐ **Imported Project**
Create a Vivado project from a Synplify Project File.
- ☐ **Example Project**
Create a new Vivado project from a predefined template.

At the bottom of the dialog, there is a help icon (a circle with a question mark) on the left, and three buttons on the right: '< Back' (disabled), 'Next >' (active/highlighted in blue), and 'Finish' (disabled).

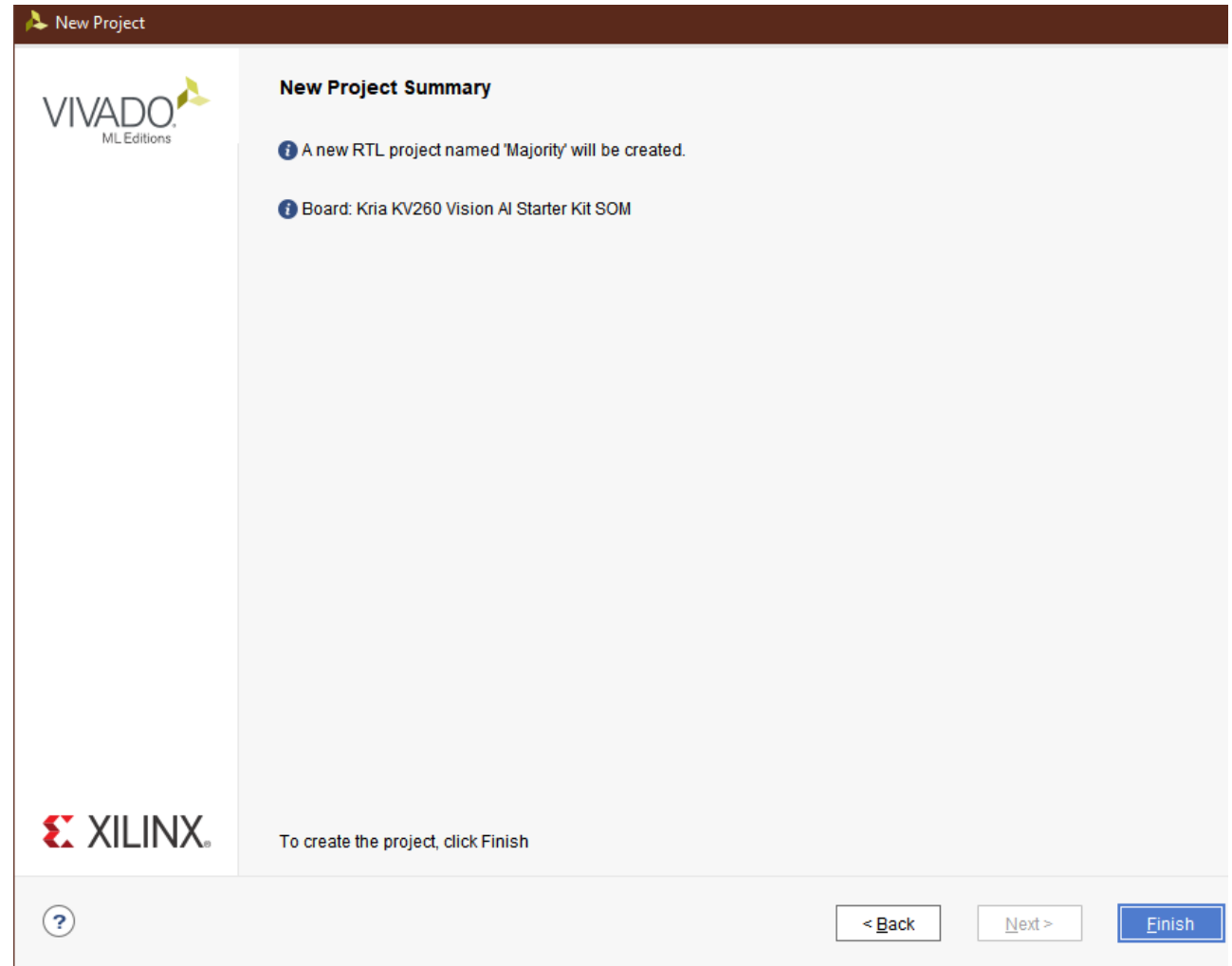
Hardware design Project creation

- Open Vivado
- Create Project
- Choose your project name and directory
- Choose RTL Project and not specify sources for now
- Under the “Boards” tab, search and choose the board “KRIA KV260”



Hardware design Project creation

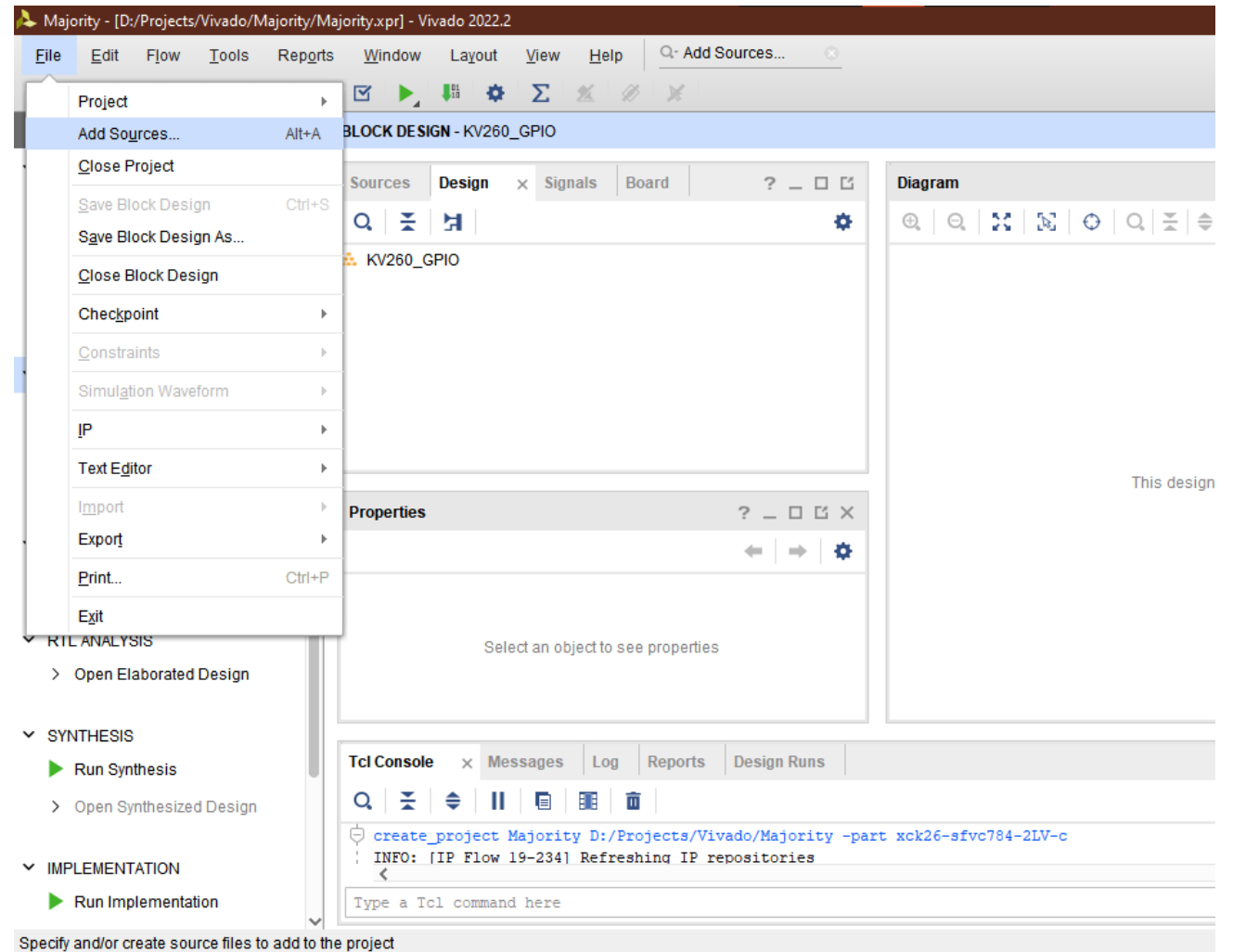
- Open Vivado
- Create Project
- Choose your project name and directory
- Choose RTL Project and not specify sources for now
- Under the “Boards” tab, search and choose the board “KRIA KV260”
- Click Next and Finish



Hardware design

Add design source


- File → Add Sources or (ALT + A)







Hardware design

Add design source

- File → Add Sources or (ALT + A)
- Click Next and choose “Create File”

 Add Sources


Add or Create Design Sources
Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source disk and add it to your project.



Use Add Files, Add Directories or Create File buttons below

Add FilesAdd DirectoriesCreate File

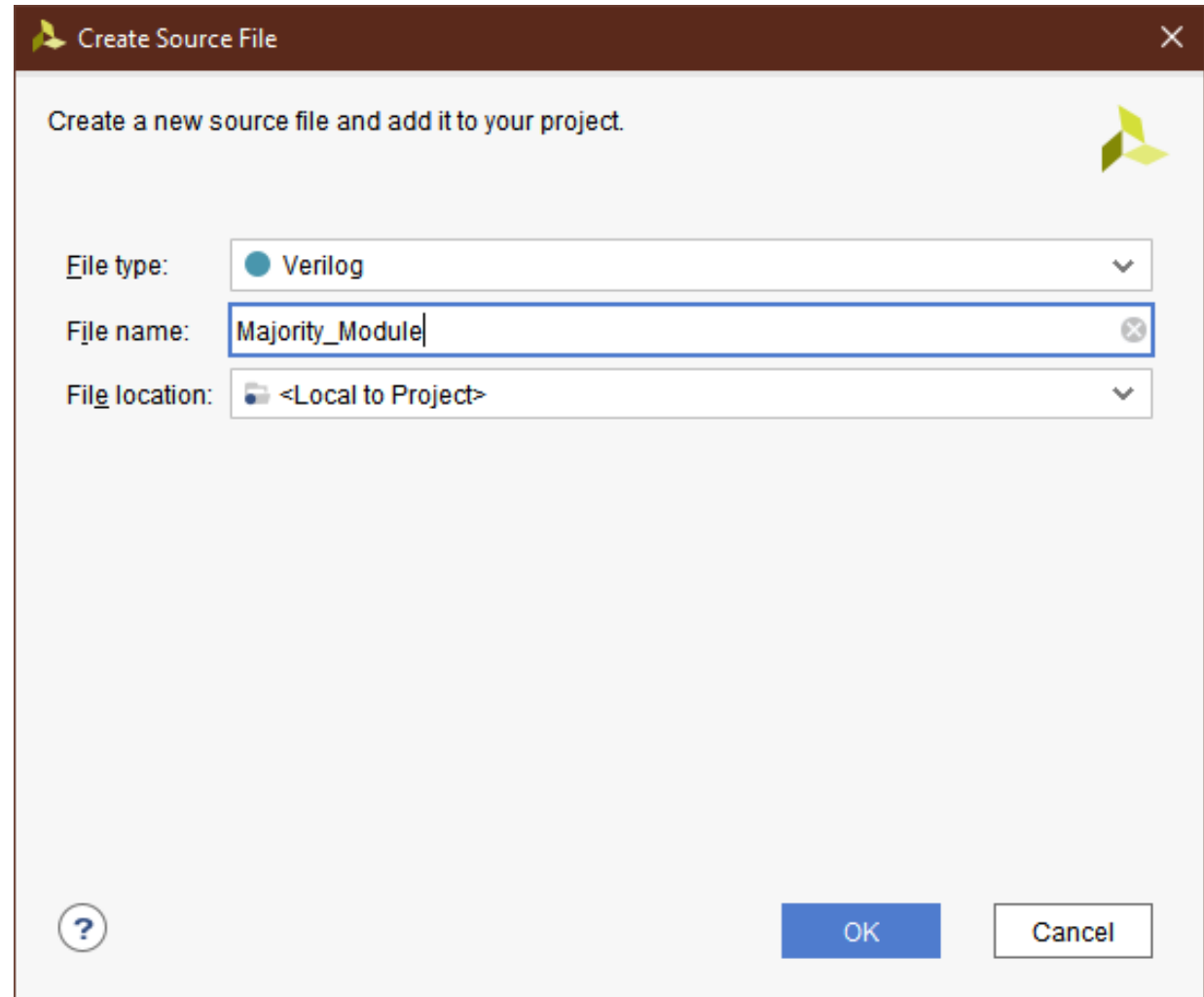
☐ Scan and add RTL include files into project
☐ Copy sources into project
☒ Add sources from subdirectories

< BackNext >Finish

Hardware design

Add design source

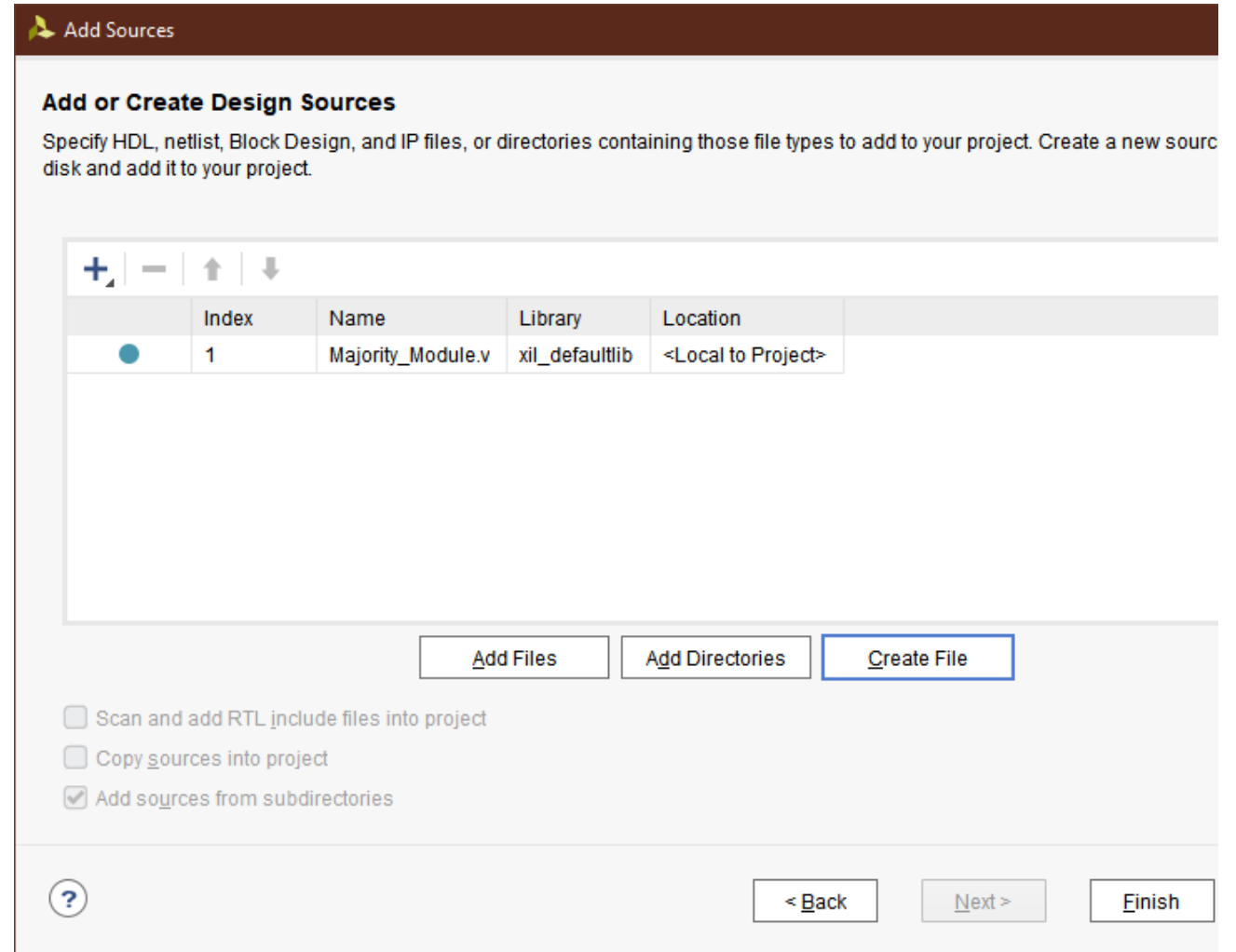
- File → Add Sources or (ALT + A)
- Click Next and choose “Create File”
- Choose your name and click “OK”



Hardware design

Add design source

- File → Add Sources or (ALT + A)
- Click Next and choose “Create File”
- Choose your name and click “OK”
- Click “Finish”



The screenshot shows the 'Add Sources' dialog box with a dark brown header. Below the header, the title 'Add or Create Design Sources' is followed by a descriptive text: 'Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source disk and add it to your project.' A table with four columns (Index, Name, Library, Location) contains one entry: '1', 'Majority_Module.v', 'xil_defaultlib', and '<Local to Project>'. Below the table are three buttons: 'Add Files', 'Add Directories', and 'Create File' (which is highlighted with a blue border). Underneath these buttons are three checkboxes: 'Scan and add RTL include files into project' (unchecked), 'Copy sources into project' (unchecked), and 'Add sources from subdirectories' (checked). At the bottom, there is a help icon (a question mark in a circle) on the left and three buttons: '< Back', 'Next >', and 'Finish'.

Add Sources

Add or Create Design Sources

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

	Index	Name	Library	Location
●	1	Majority_Module.v	xil_defaultlib	<Local to Project>

Add Files **Add Directories** **Create File**

☐ Scan and add RTL include files into project

☐ Copy sources into project


☒ Add sources from subdirectories

? < Back Next > Finish

Hardware design

Add design source

- File → Add Sources or (ALT + A)
- Click Next and choose “Create File”
- Choose your name and click “OK”
- Click “Finish”
- Specify your ports





 Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.


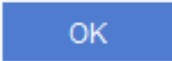
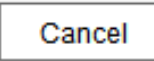
Module Definition

Module name: Majority_Module

I/O Port Definitions



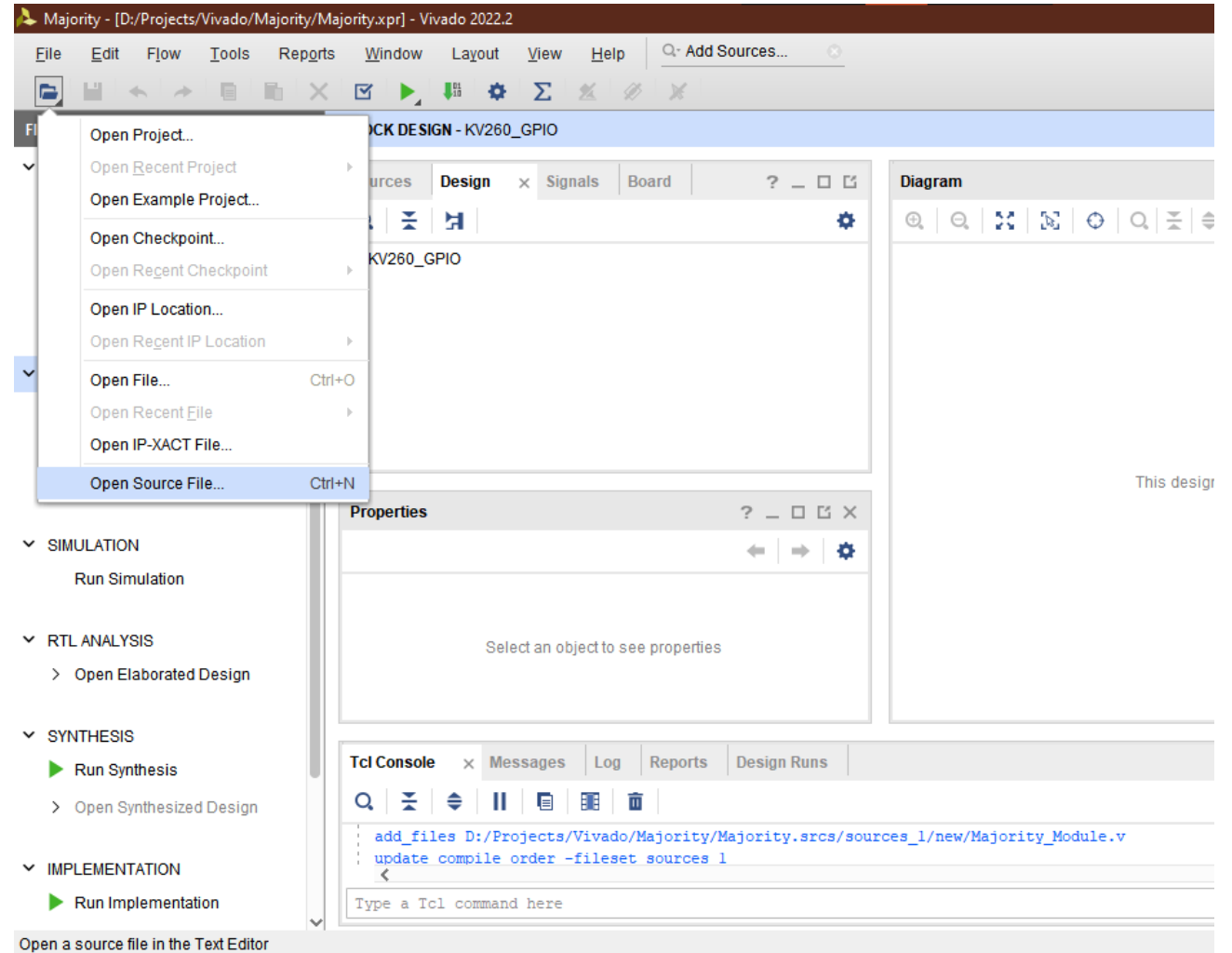
Port Name	Direction	Bus	MSB	LSB
A	input ▾	<input type="checkbox"/>	0	0
B	input ▾	<input type="checkbox"/>	0	0
C	input ▾	<input type="checkbox"/>	0	0
Z	output ▾	<input type="checkbox"/>	0	0

Hardware design

Add design source

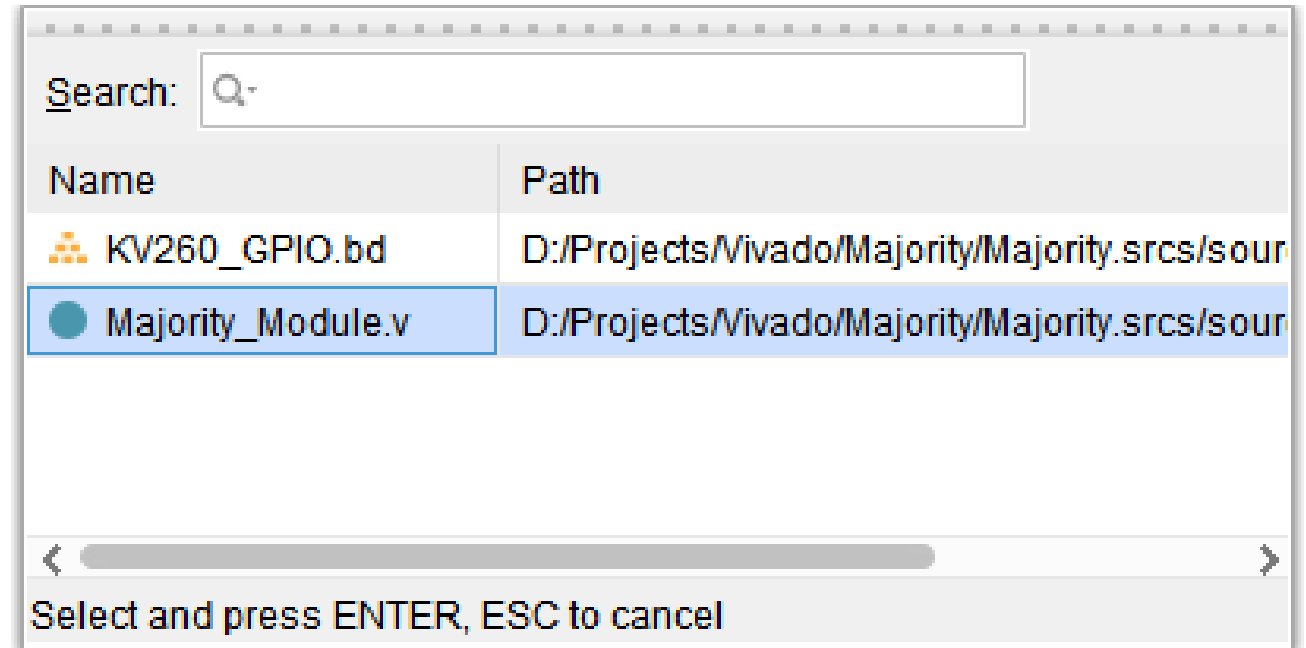
- File → Add Sources or (ALT + A)
- Click Next and choose “Create File”
- Choose your name and click “OK”
- Click “Finish”
- Specify your ports
- Open → Open Source File



Hardware design

Add design source

- File → Add Sources or (ALT + A)
- Click Next and choose “Create File”
- Choose your name and click “OK”
- Click “Finish”
- Specify your ports
- Open → Open Source File
- Choose your file and hit ENTER



Hardware design

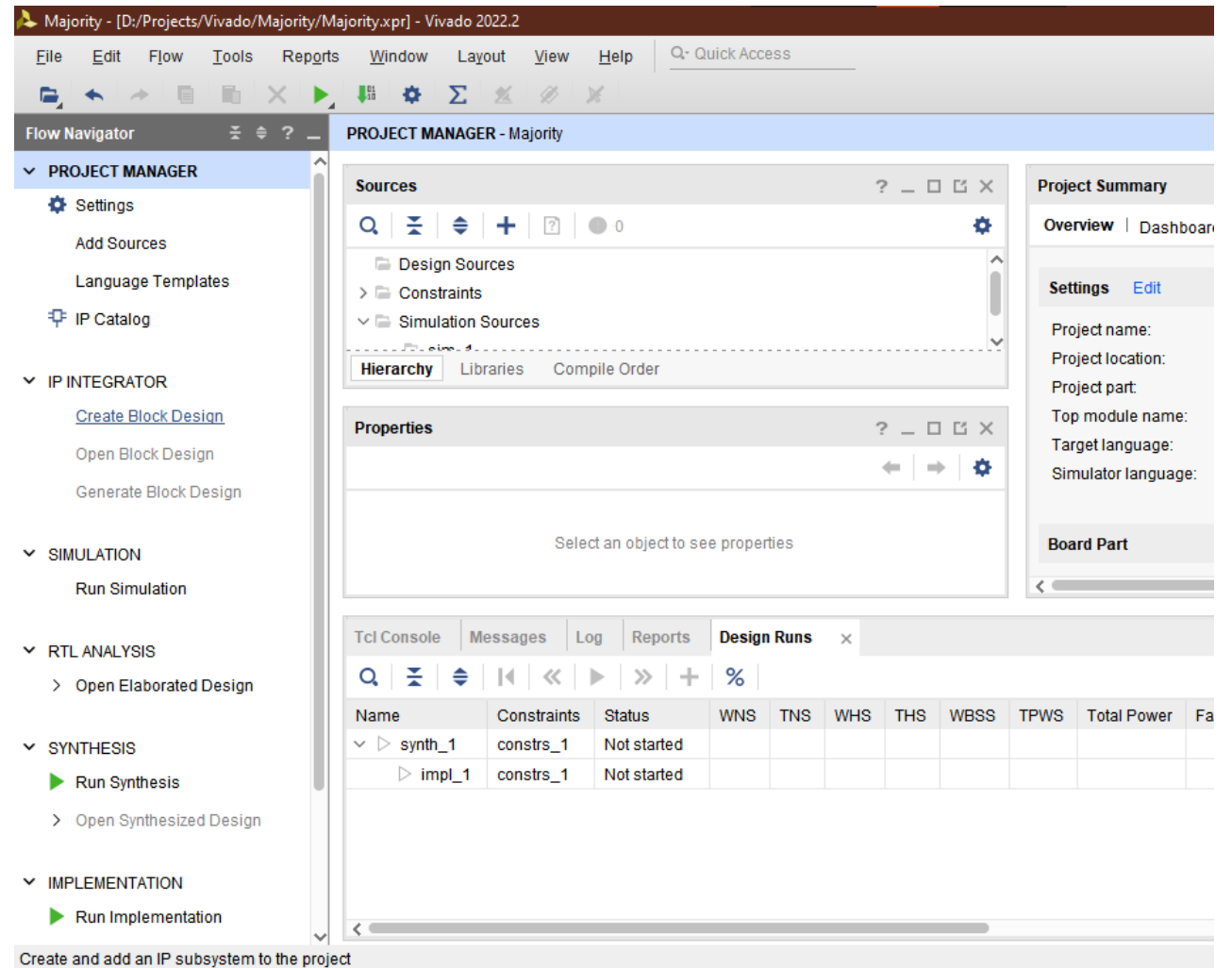
Add design source

- File → Add Sources or (ALT + A)
- Click Next and choose “Create File”
- Choose your name and click “OK”
- Click “Finish”
- Specify your ports
- Open → Open Source File
- Choose your file and hit ENTER
- Type your code for a majority circuit and save it.

```
module Majority_Module(  
    input A, B, C  
    output Z  
);  
  
    and(and0, A, B);  
    and(and1, B, C);  
    and(and2, A, C);  
    or(OUTOUT, and0, and1, and2);  
  
endmodule
```

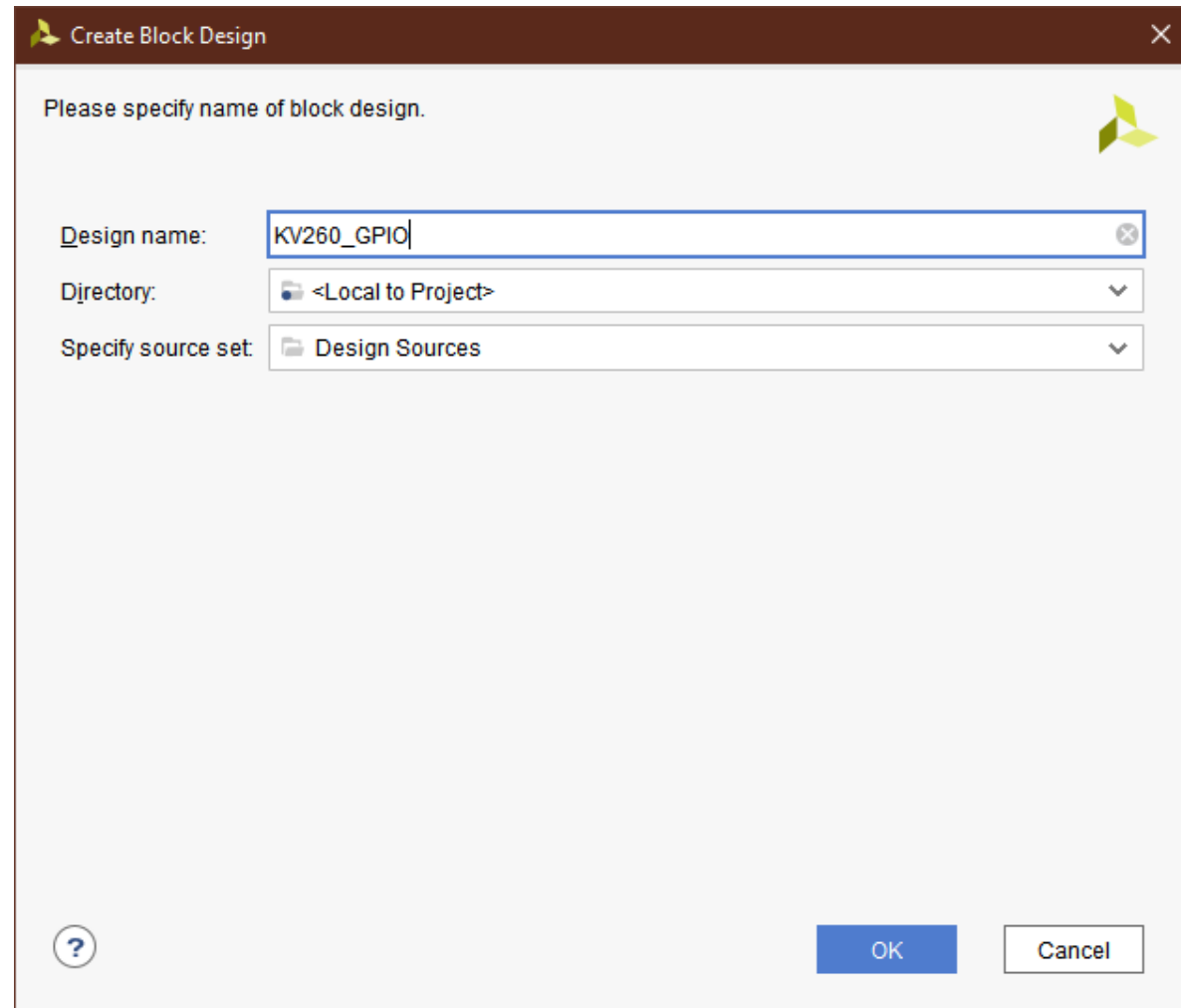
Hardware design Block design

- Click “Create Block Design”



Hardware design Block design

- Click “Create Block Design”
- Choose your name and click “OK”



The image shows a 'Create Block Design' dialog box with a dark red title bar. The main area is light gray and contains the instruction 'Please specify name of block design.' followed by a small yellow logo. There are three input fields: 'Design name:' with the text 'KV260_GPIO', 'Directory:' with a dropdown menu showing '<Local to Project>', and 'Specify source set:' with a dropdown menu showing 'Design Sources'. At the bottom, there is a help icon (a circle with a question mark), an 'OK' button, and a 'Cancel' button.

Create Block Design

Please specify name of block design.

Design name: KV260_GPIO

Directory: <Local to Project>

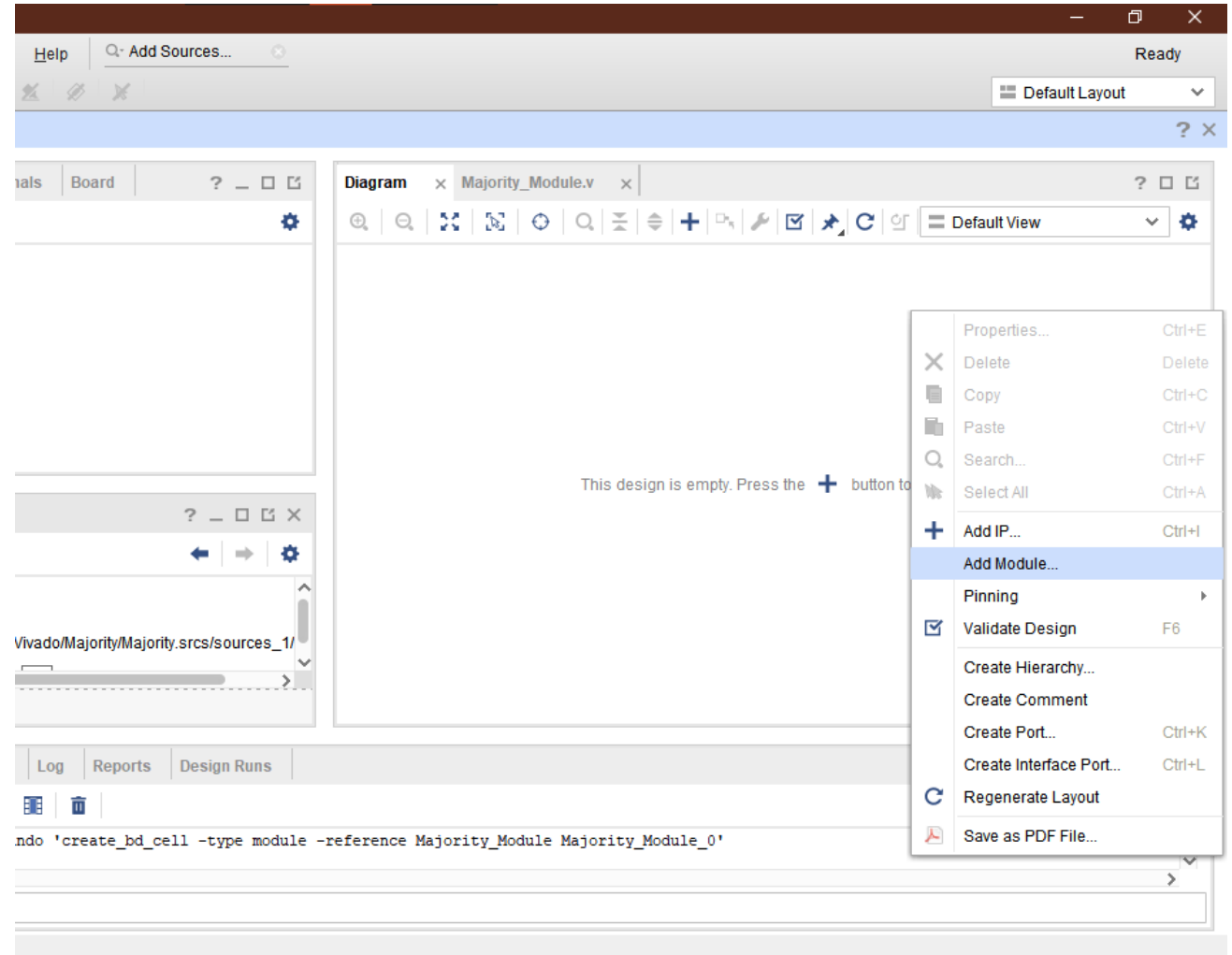
Specify source set: Design Sources

?

OK Cancel

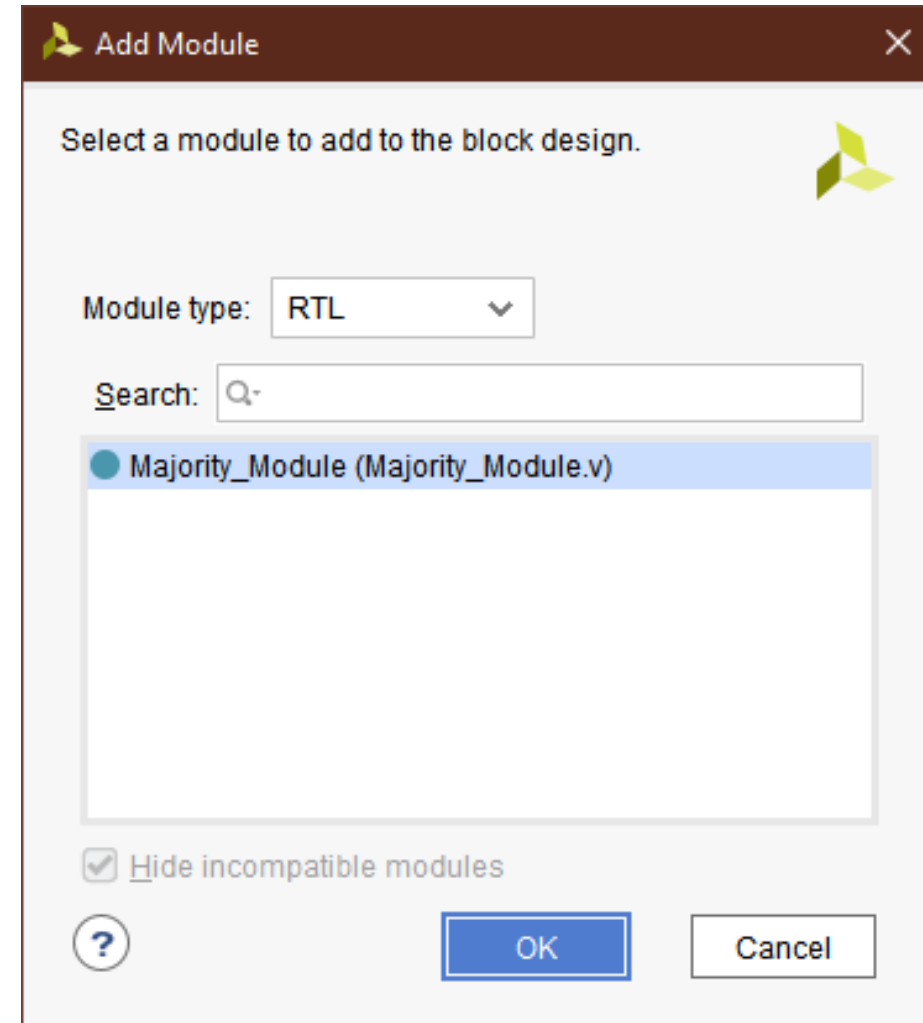
Hardware design Block design

- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”



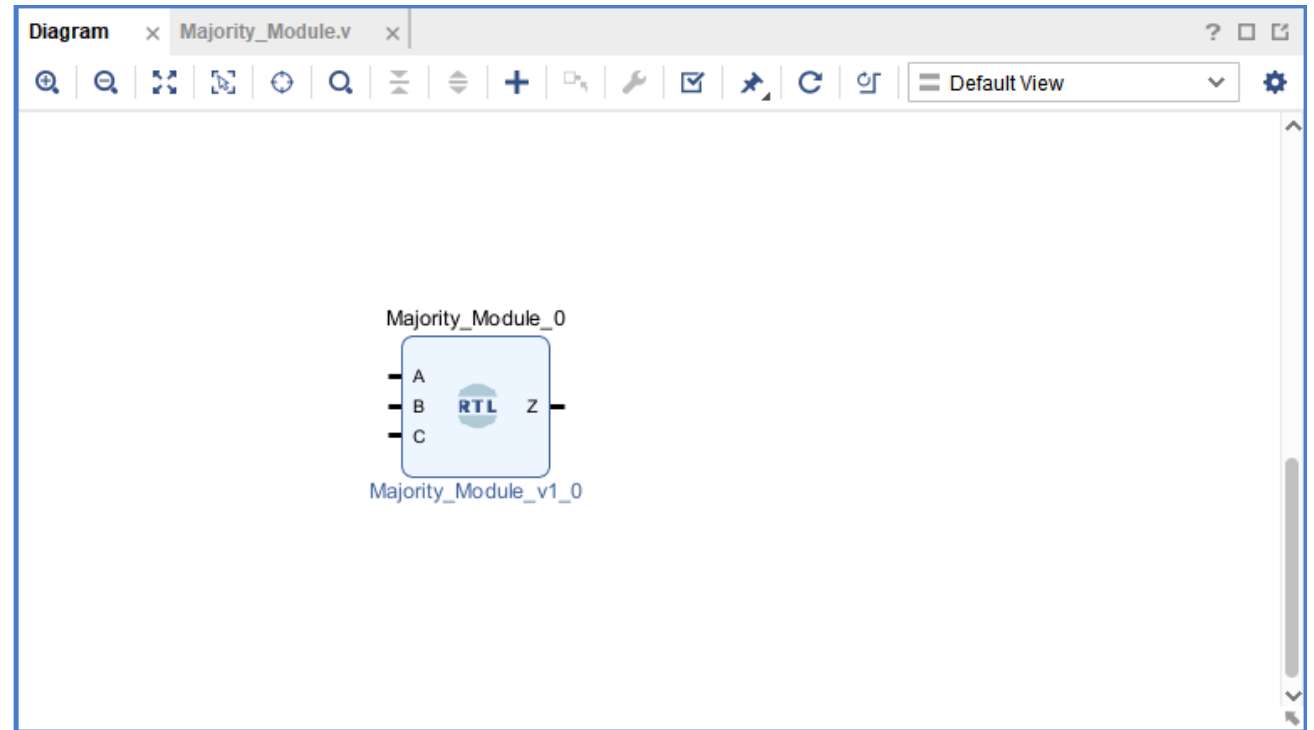
Hardware design Block design

- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”
- Choose your module and click “OK”



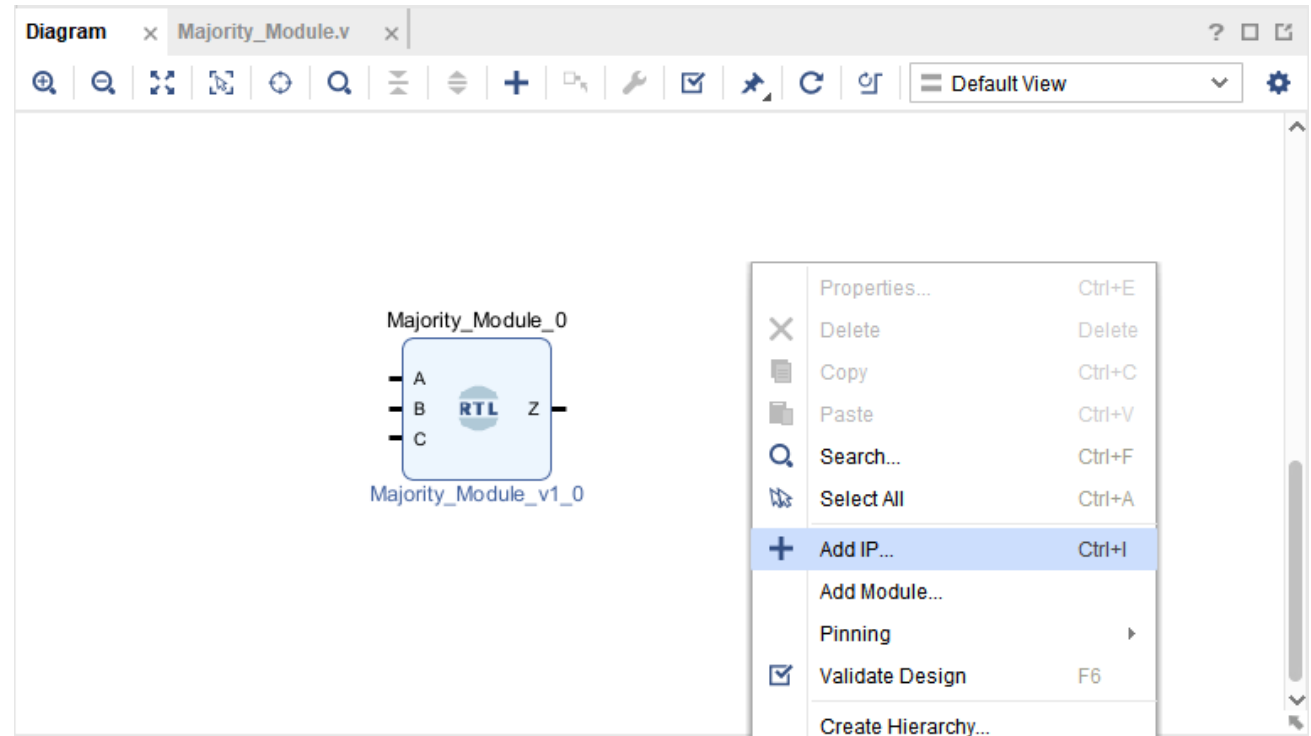
Hardware design Block design

- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”
- Choose your module and click “OK”



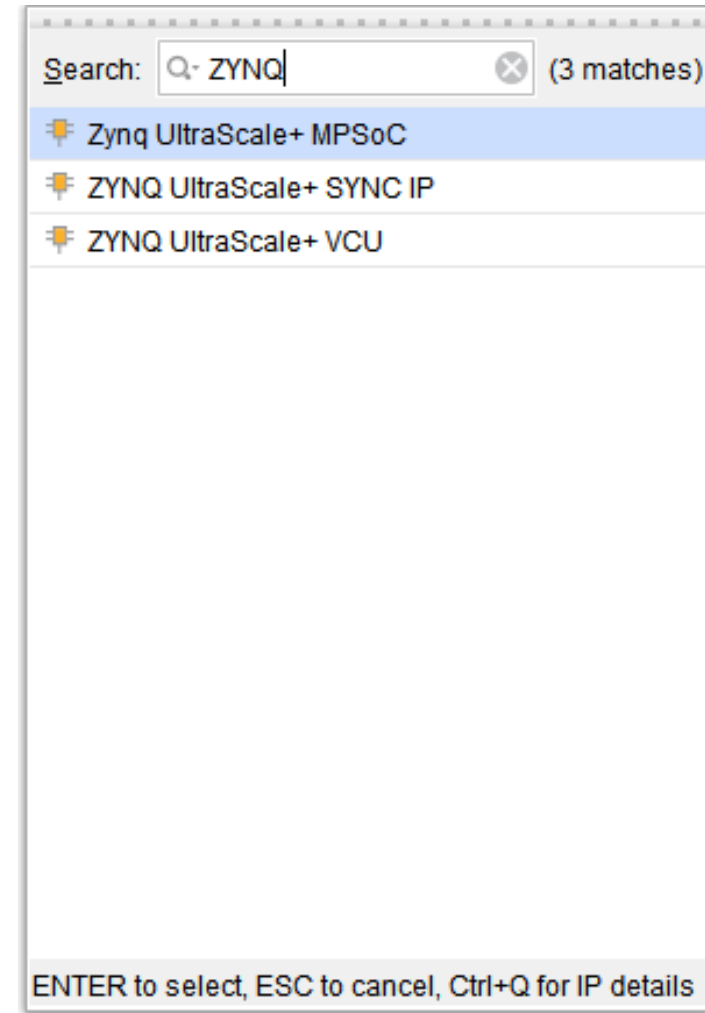
Hardware design Block design

- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”
- Choose your module and click “OK”
- Right click on the diagram and click “Add IP”



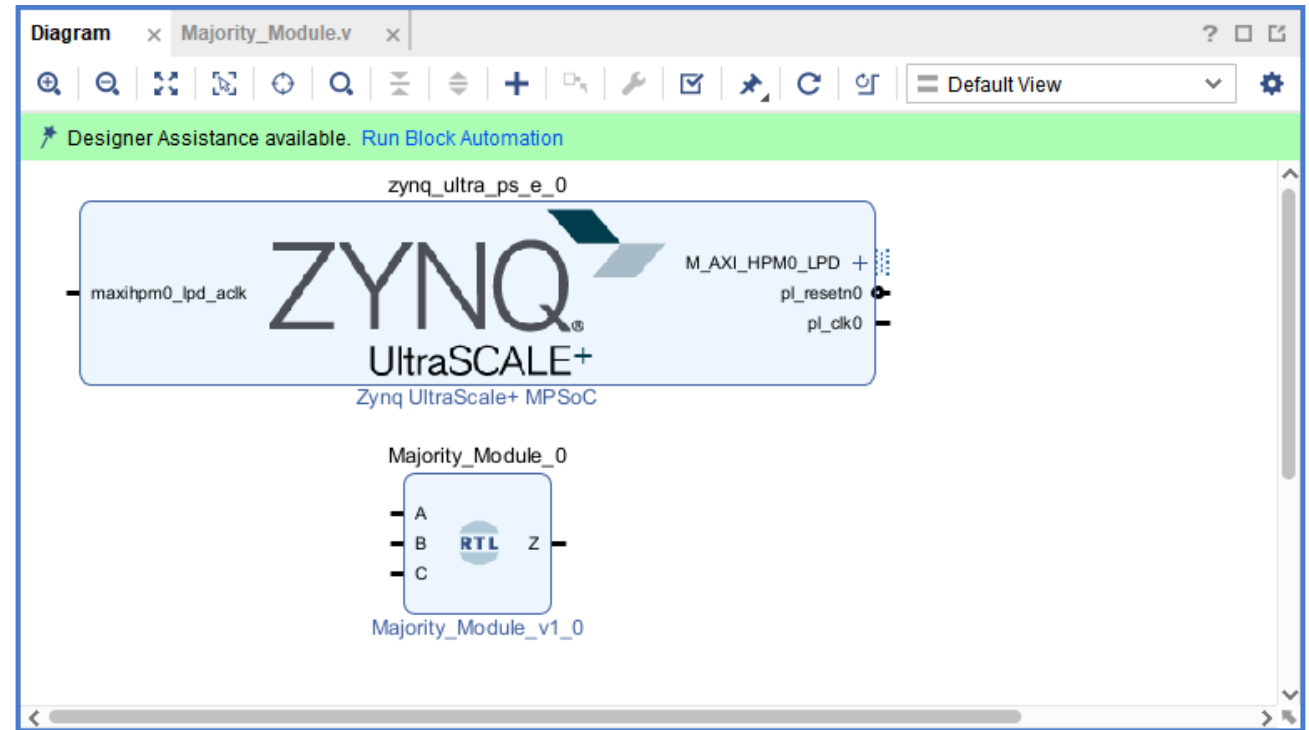
Hardware design Block design

- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”
- Choose your module and click “OK”
- Right click on the diagram and click “Add IP”
- Search and select ZYNQ MPSoC



Hardware design Block design

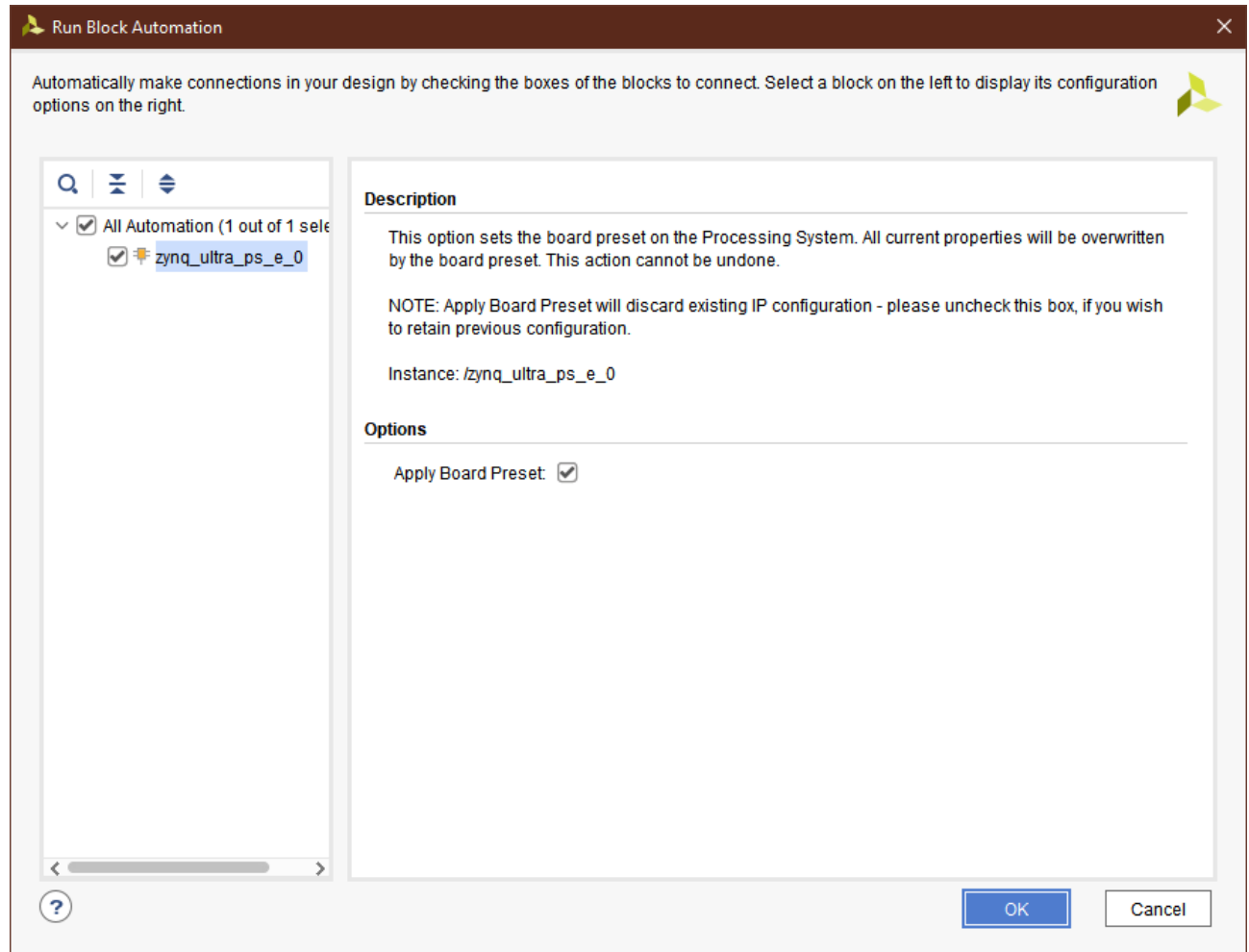
- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”
- Choose your module and click “OK”
- Right click on the diagram and click “Add IP”
- Search and select ZYNQ MPSoC
- Click “Run Block Automation” to apply the specifications and settings of the KV260 Board to the ZYNQ block



Hardware design

Block design

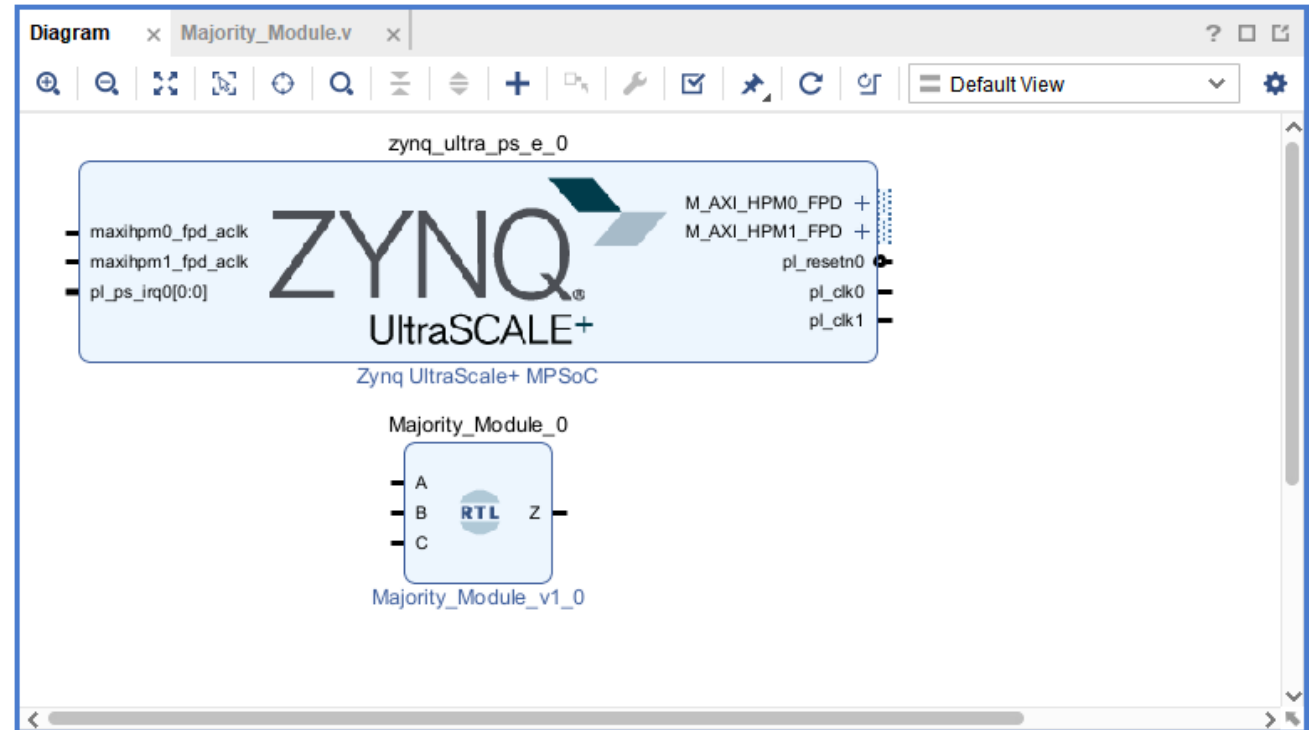
- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”
- Choose your module and click “OK”
- Right click on the diagram and click “Add IP”
- Search and select ZYNQ MPSoC
- Click “Run Block Automation” to apply the specifications and settings of the KV260 Board to the ZYNQ block



Hardware design

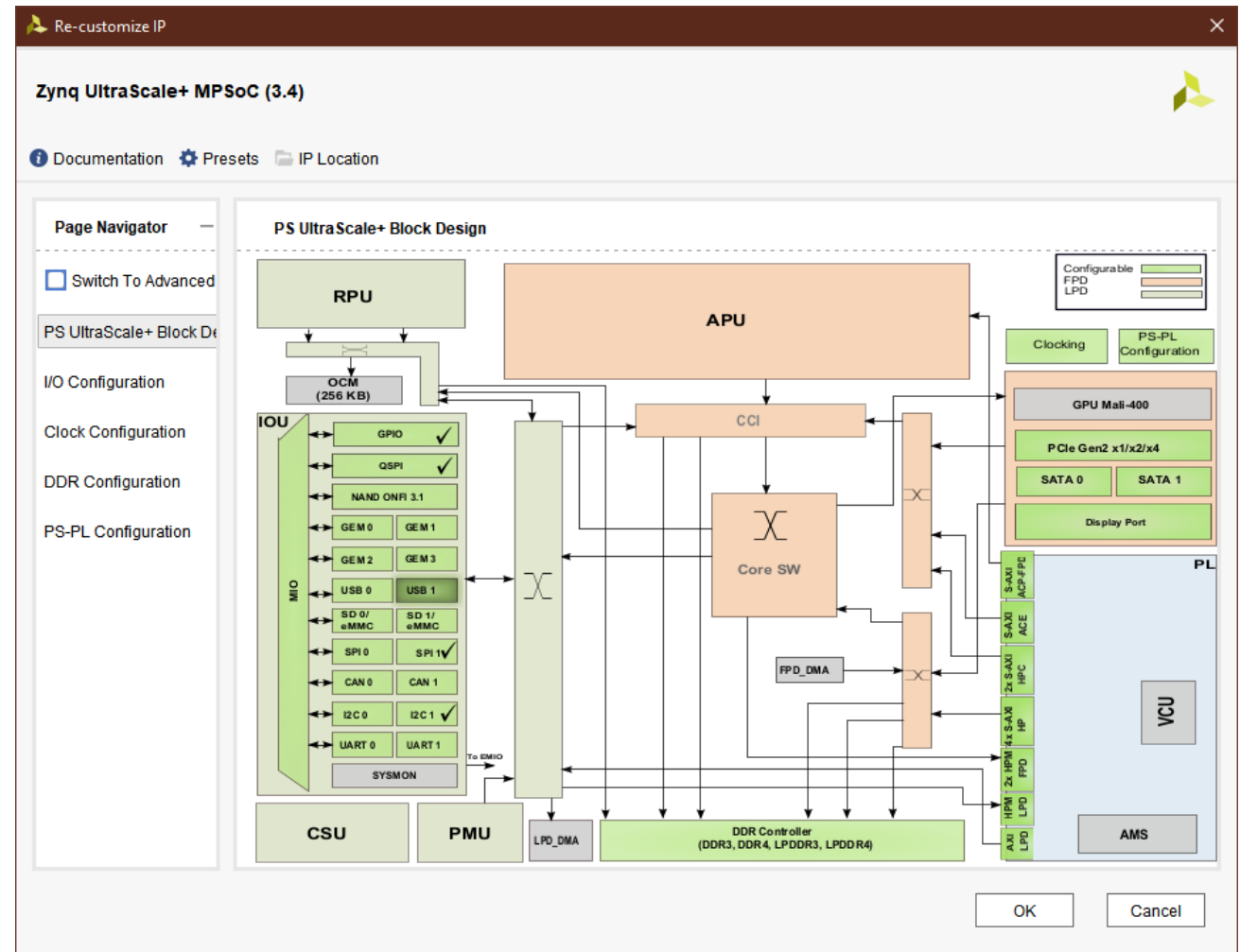
Block design

- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”
- Choose your module and click “OK”
- Right click on the diagram and click “Add IP”
- Search and select ZYNQ MPSoC
- Click “Run Block Automation” to apply the specifications and settings of the KV260 Board to the ZYNQ block



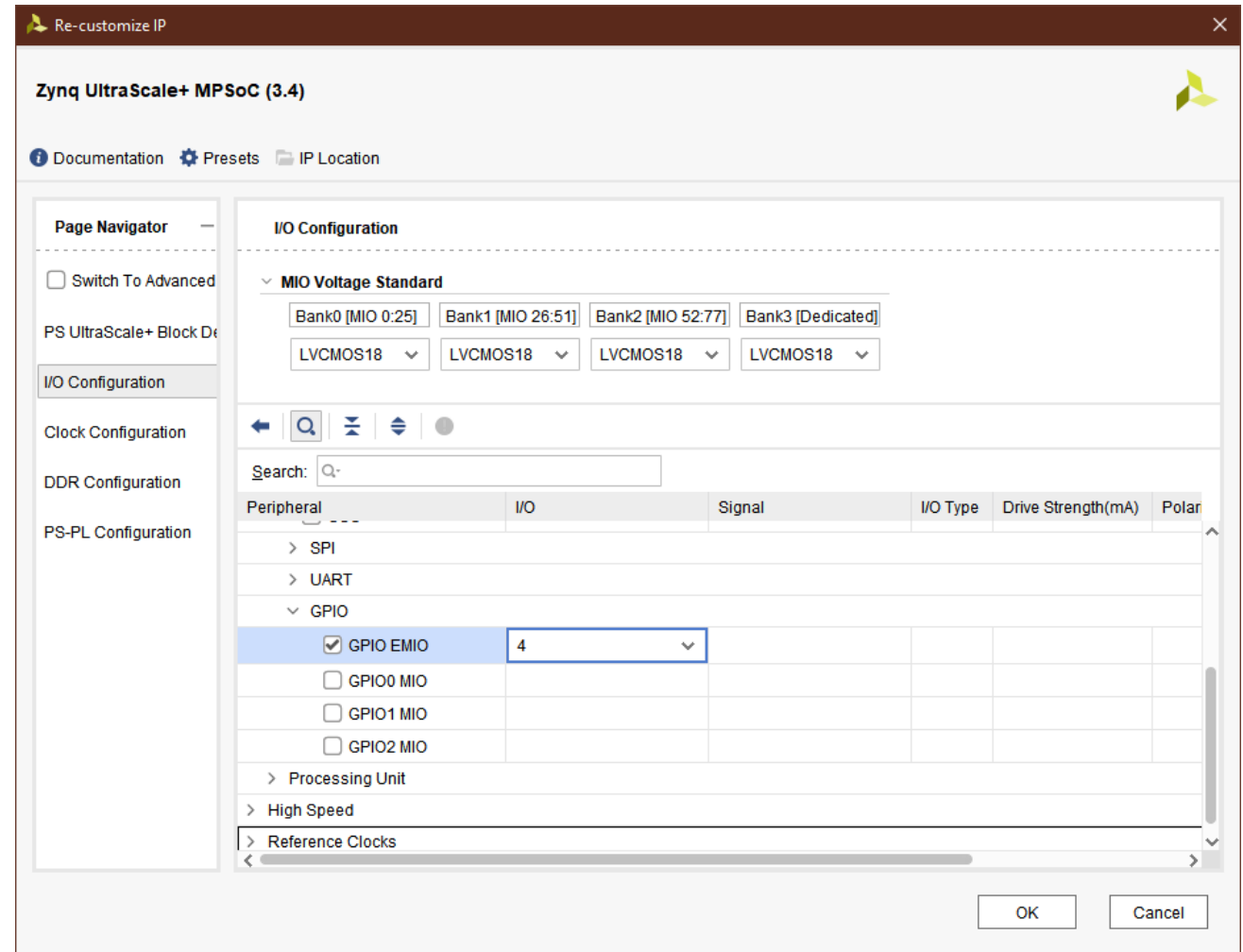
Hardware design Block design

- Click “Create Block Design”
- Choose your name and click “OK”
- Right click on the diagram you just created and click “Add module”
- Choose your module and click “OK”
- Right click on the diagram and click “Add IP”
- Search and select ZYNQ MPSoC
- Click “Run Block Automation” to apply the specifications and settings of the KV260 Board to the ZYNQ block
- Double click on the ZYNQ block to customize it
- First of all, search and disable all AXI HMP clocks in PS-PL Configuration



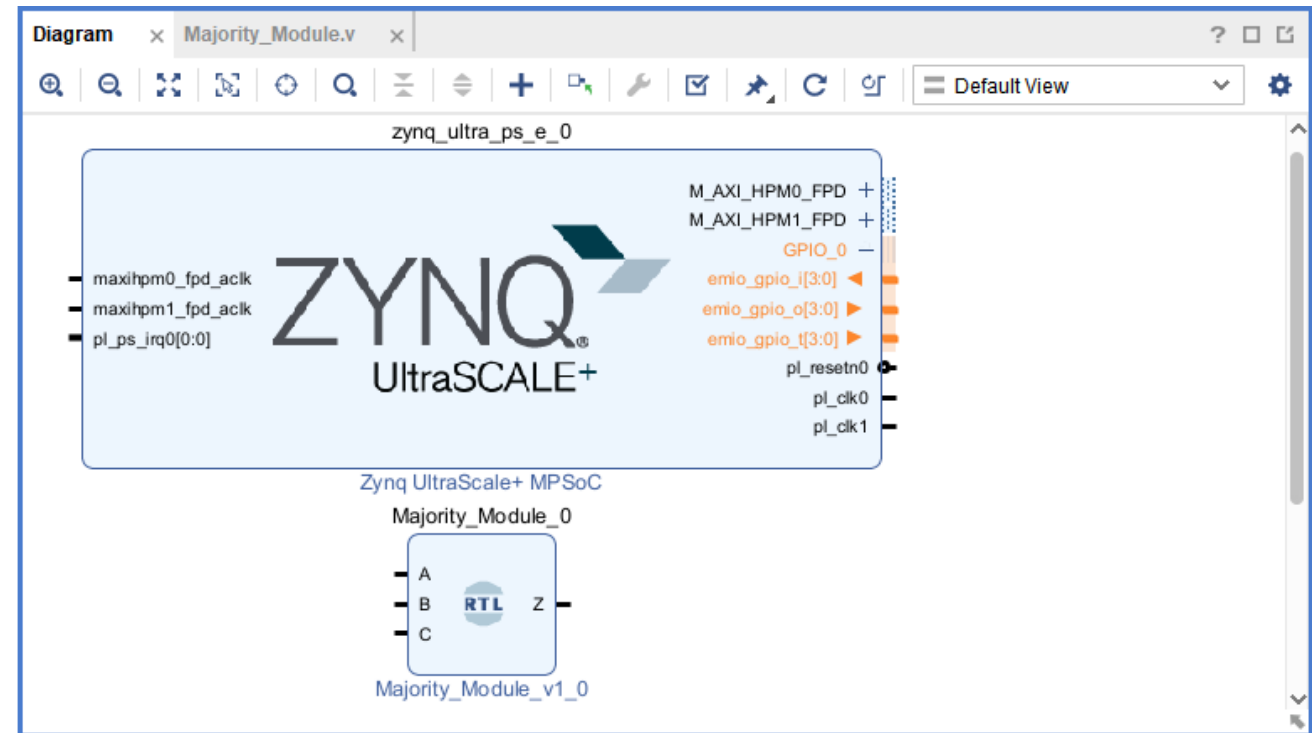
Hardware design Block design (Cont.)

- Under I/O Configuration → Low Speed → I/O Peripherals → GPIO, select “GPIO EMIO” and make it 4, also deselect GPIO1 MIO and GPIO2 MIO, click “OK”



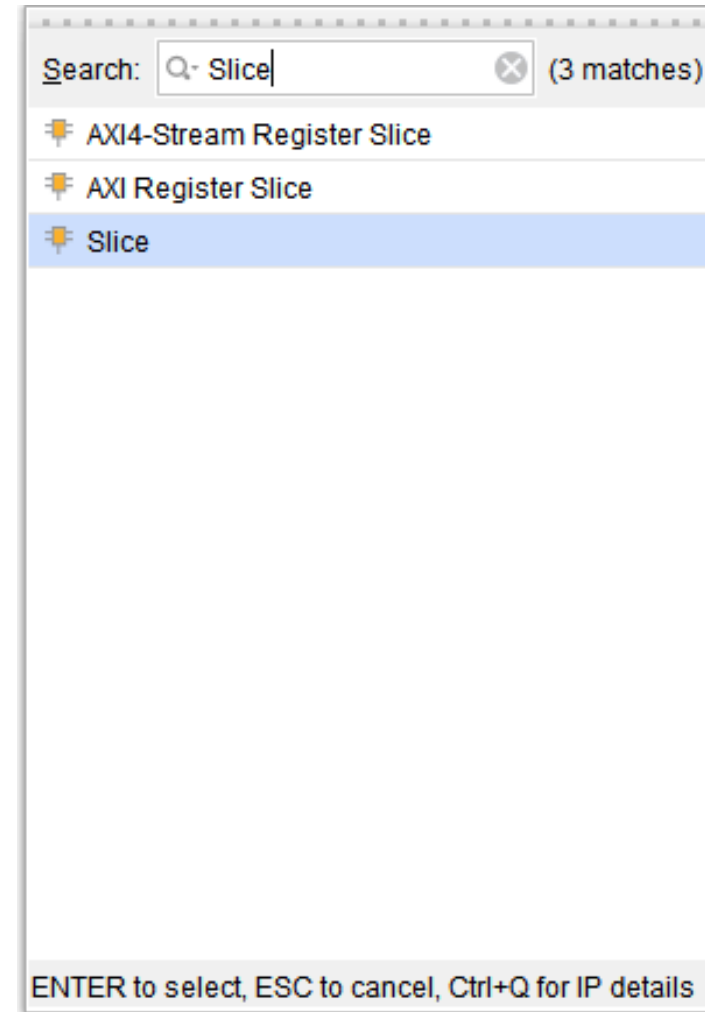
Hardware design Block design (Cont.)

- Under I/O Configuration → Low Speed → I/O Peripherals → GPIO, select “GPIO EMIO” and make it 4, also deselect GPIO1 MIO and GPIO2 MIO, click “OK”



Hardware design Block design (Cont.)

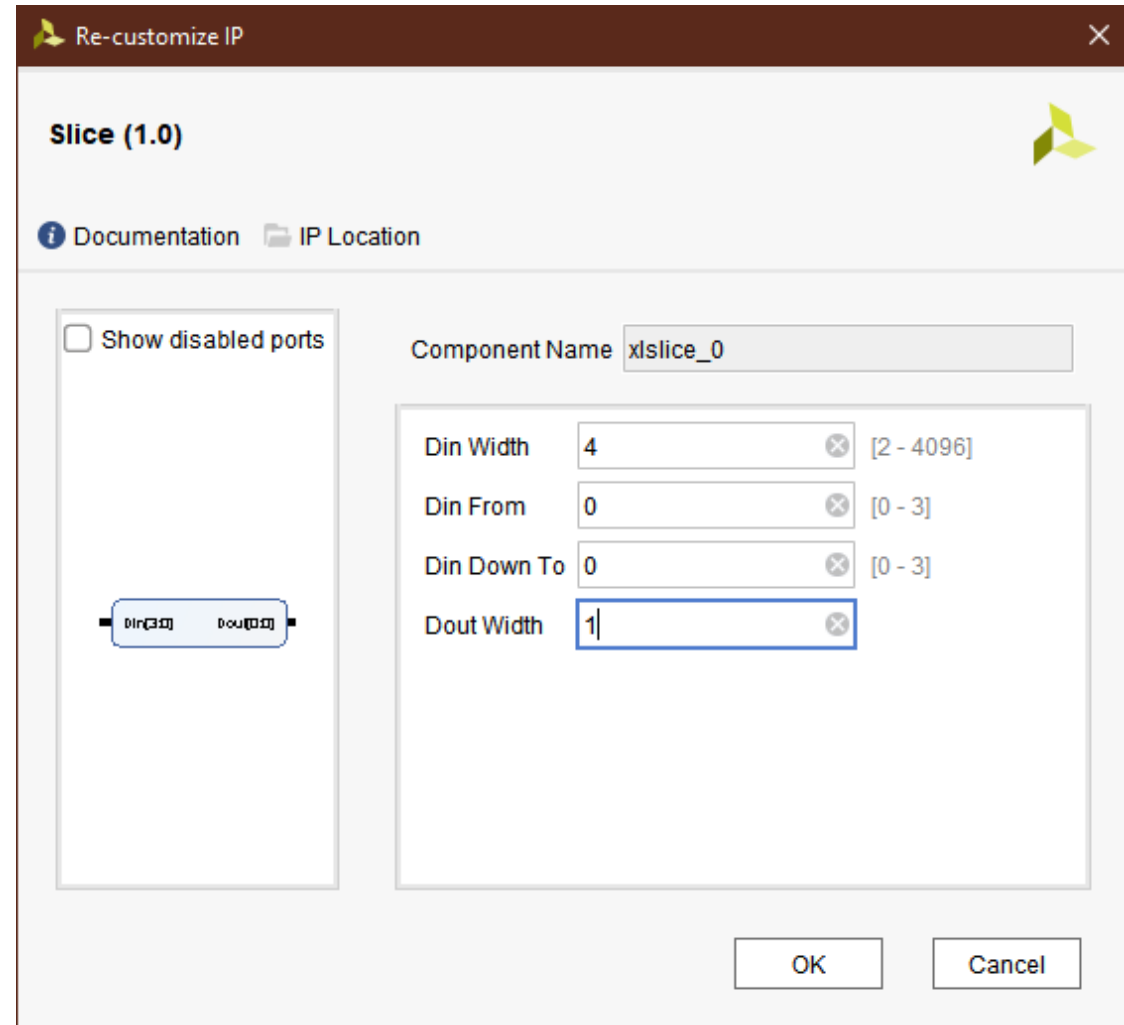
- Under I/O Configuration → Low Speed → I/O Peripherals → GPIO, select “GPIO EMIO” and make it 4, also deselect GPIO1 MIO and GPIO2 MIO, click “OK”
- Add another IP called Slice and customize it as follows



Hardware design

Block design (Cont.)

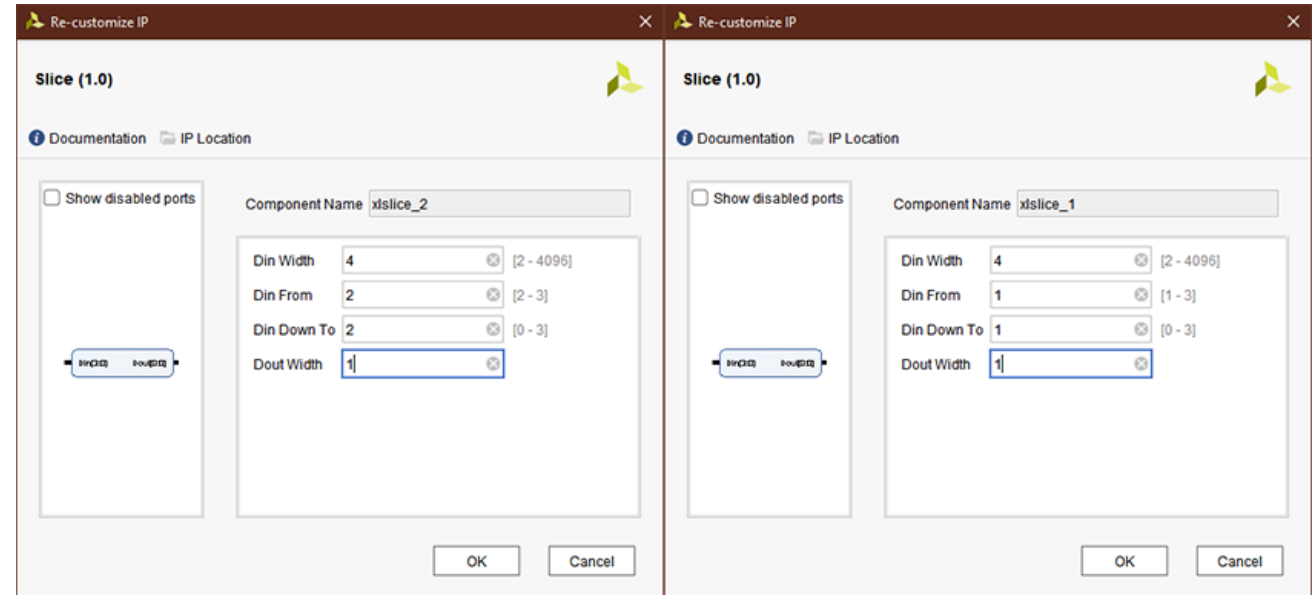
- Under I/O Configuration → Low Speed → I/O Peripherals → GPIO, select “GPIO EMIO” and make it 4, also deselect GPIO1 MIO and GPIO2 MIO, click “OK”
- Add another IP called Slice and customize it as follows



Hardware design

Block design (Cont.)

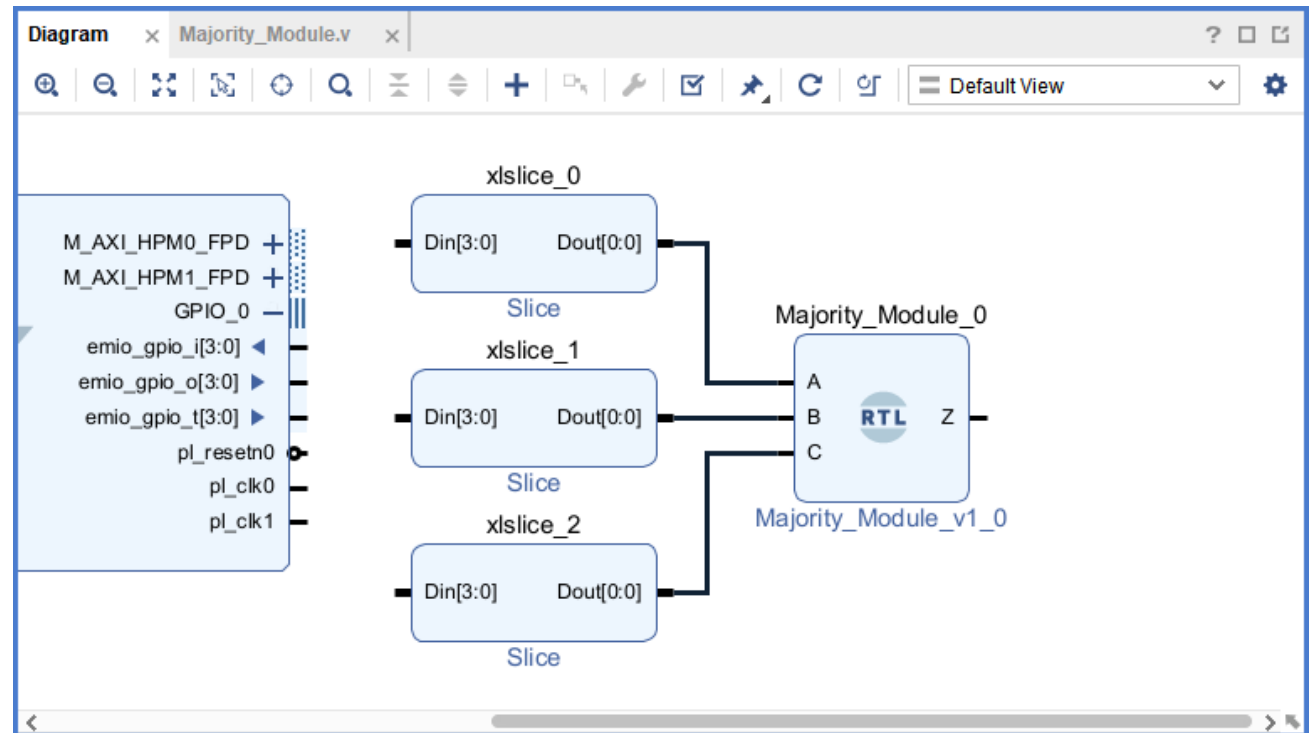
- Under I/O Configuration → Low Speed → I/O Peripherals → GPIO, select “GPIO EMIO” and make it 4, also deselect GPIO1 MIO and GPIO2 MIO, click “OK”
- Add another IP called Slice and customize it as follows
- Add 2 more Slices and customize it



Hardware design

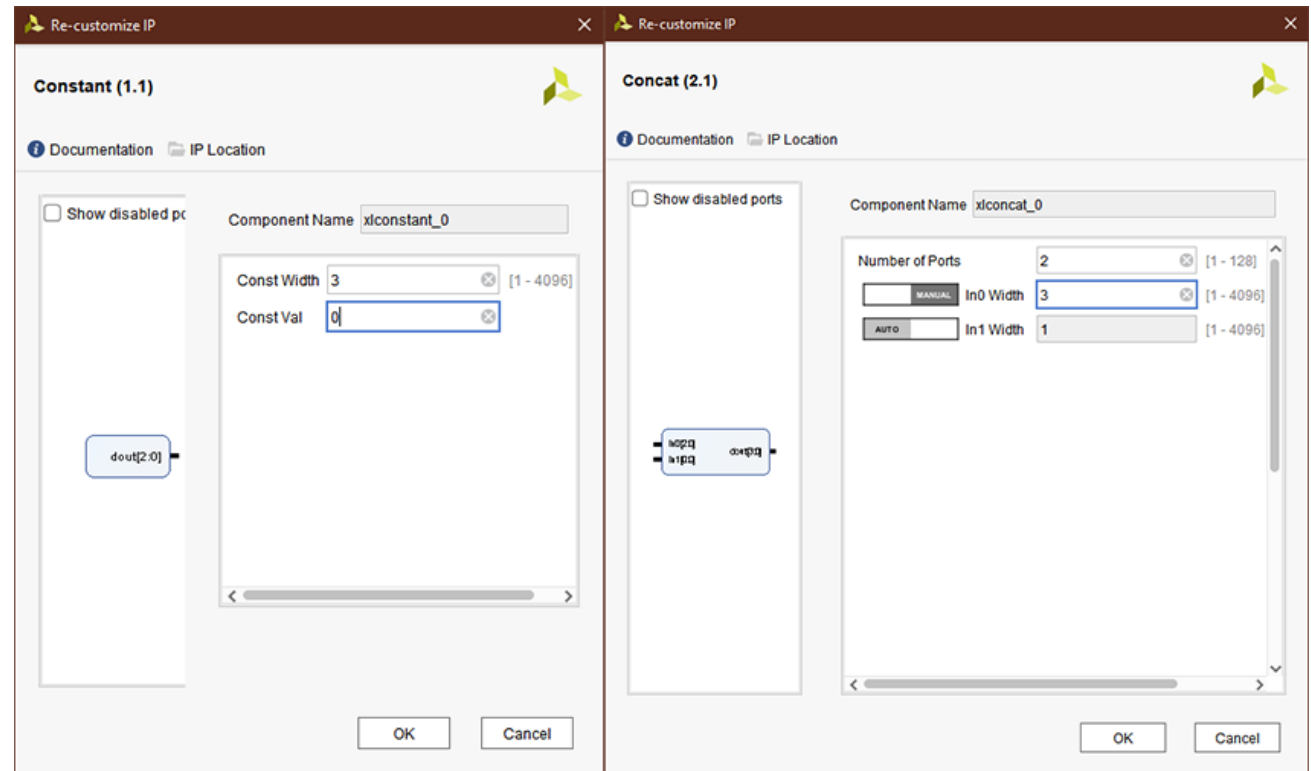
Block design (Cont.)

- Under I/O Configuration → Low Speed → I/O Peripherals → GPIO, select “GPIO EMIO” and make it 4, also deselect GPIO1 MIO and GPIO2 MIO, click “OK”
- Add another IP called Slice and customize it as follows
- Add 2 more Slices and customize it
- Note that each slice will be one of your input port. Therefore you can connect the output of each slices to the input of your majority module
- And for the output of our majority module, we will have to concatenate it with dummy signals to fit the GPIO ports



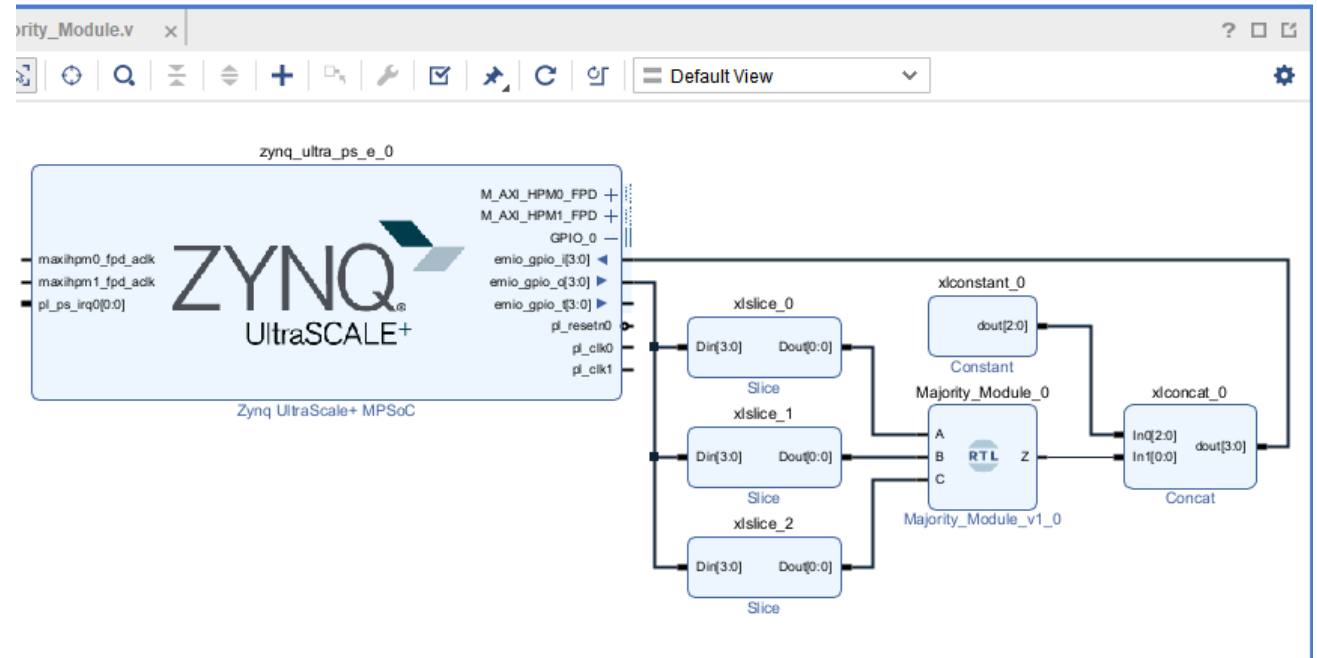
Hardware design Block design (Cont.)

- Add one concat IP and one constant IP. Customize them as follows



Hardware design Block design (Cont.)

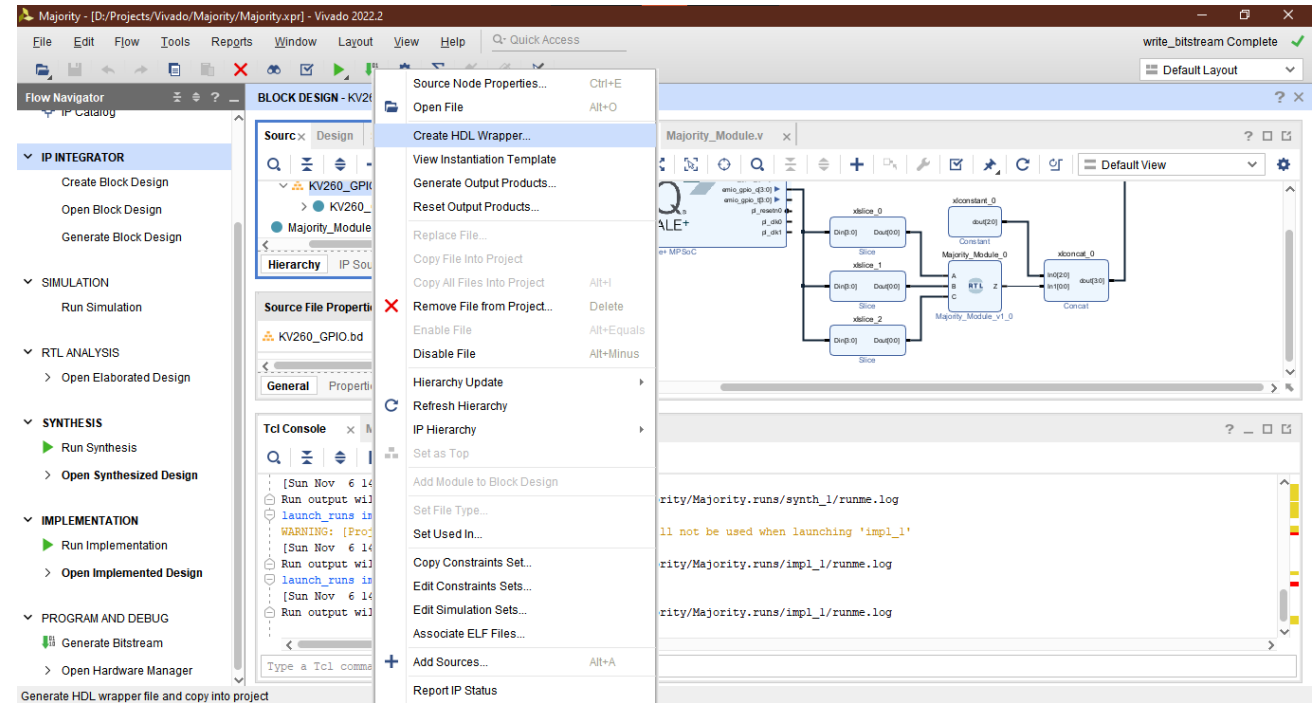
- Add one concat IP and one constant IP. Customize them as follows
- Connect concat, constant and majority block
- And finally, connect each of the slices to the **GPIO Out**, and connect the **dout** of the concat block to the **GPIO In**



Hardware design

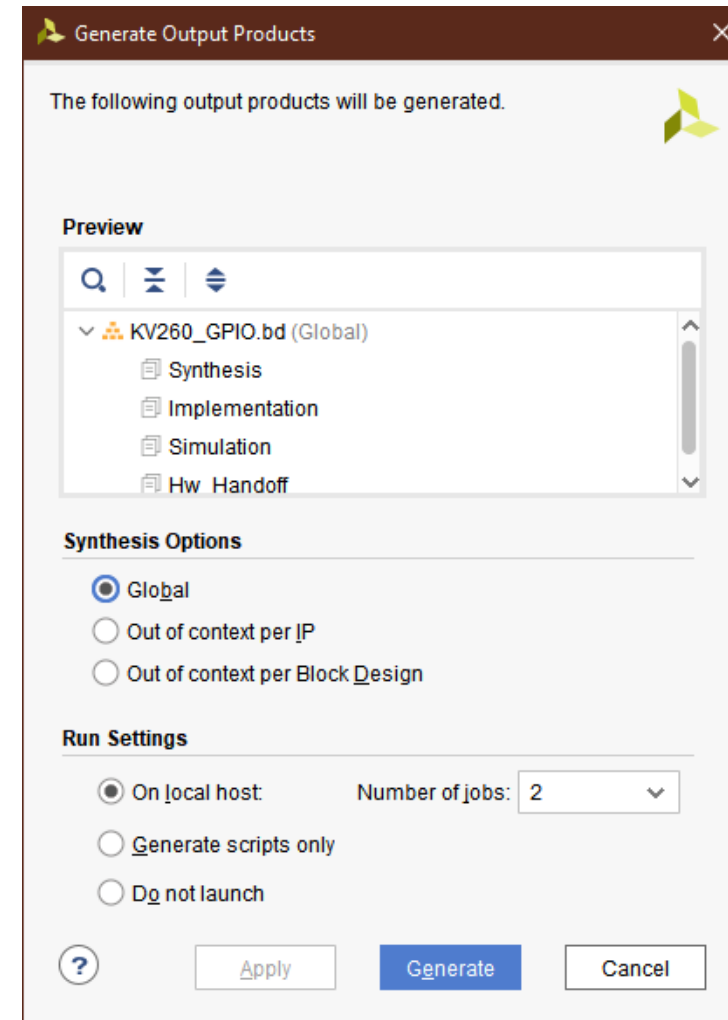
Block design (Cont.)

- Add one concat IP and one constant IP. Customize them as follows
- Connect concat, constant and majority block
- And finally, connect each of the slices to the **GPIO Out**, and connect the **dout** of the concat block to the **GPIO In**
- Right click on your block design file and select create HDL wrapper



Hardware design Block design (Cont.)

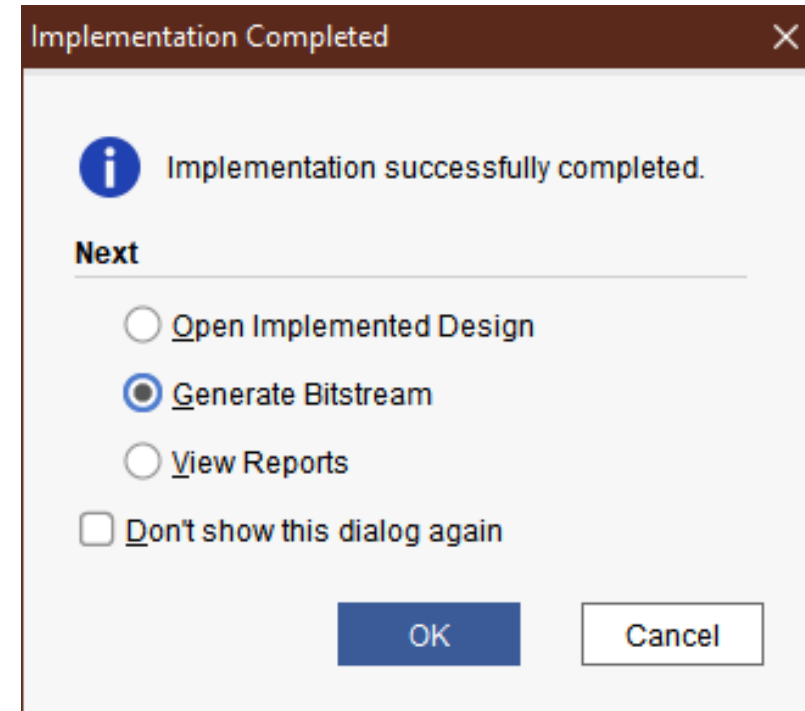
- Add one concat IP and one constant IP. Customize them as follows
- Connect concat, constant and majority block
- And finally, connect each of the slices to the **GPIO Out**, and connect the **dout** of the concat block to the **GPIO In**
- Right click on your block design file and select create HDL wrapper
- Right click on your block design file again and select Generate output products. Select Global in the synthesis option and click generate.



Hardware design

Block design (Cont.)

- Add one concat IP and one constant IP. Customize them as follows
- Connect concat, constant and majority block
- And finally, connect each of the slices to the **GPIO Out**, and connect the **dout** of the concat block to the **GPIO In**
- Right click on your block design file and select create HDL wrapper
- Right click on your block design file again and select Generate output products. Select Global in the synthesis option and click generate.
- You can now generate bitstream



Hardware design

File location











- You'll need 2 files, one will be the hwh file located here

"Majority\Majority.gen\sources_1\bd\KV260_GPIO\hw_handoff"

- And the other is the bit stream file, located here

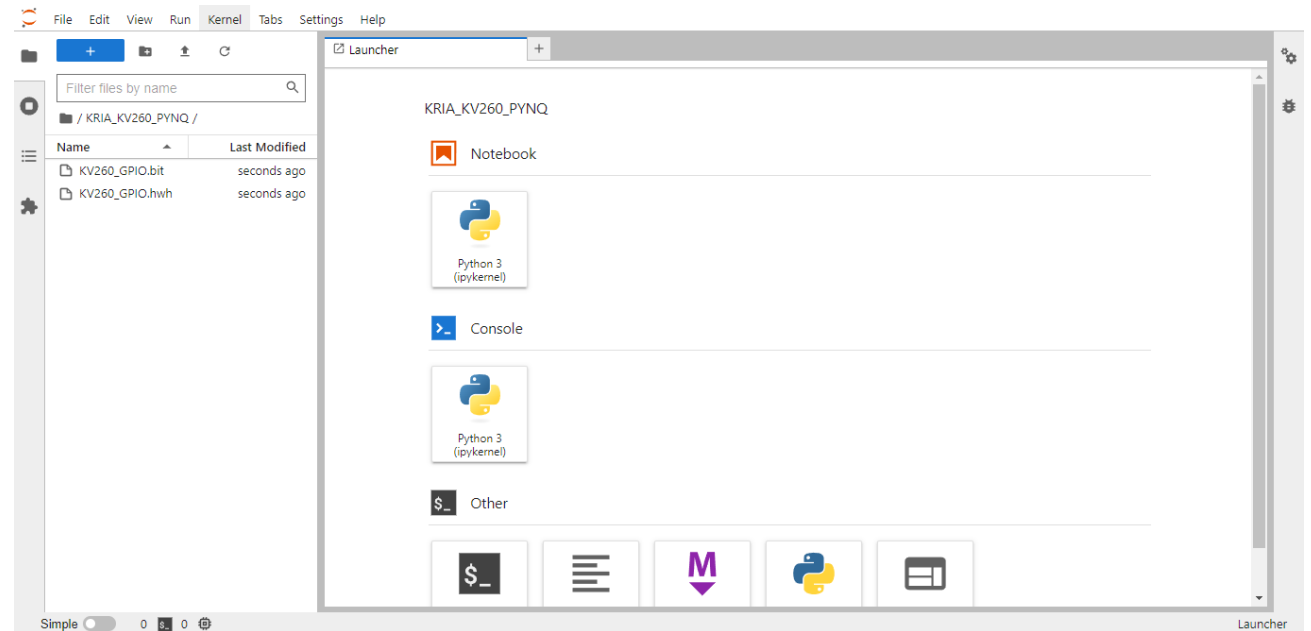
"Majority\Majority.runs\impl_1"

- Move them to the global folder and make sure they have the same name

 Majority.cache	11/6/2022 1:55 PM	File folder	
 Majority.gen	11/6/2022 12:56 PM	File folder	
 Majority.hw	11/6/2022 12:56 PM	File folder	
 Majority.ip_user_files	11/6/2022 2:31 PM	File folder	
 Majority.runs	11/6/2022 1:55 PM	File folder	
 Majority.sim	11/6/2022 12:56 PM	File folder	
 Majority.srcs	11/6/2022 2:32 PM	File folder	
 KV260_GPIO.bit	11/6/2022 2:52 PM	BIT File	7,616 KB
 KV260_GPIO.hwh	11/6/2022 2:31 PM	HWH File	136 KB
 Majority.xpr	11/6/2022 2:50 PM	Vivado Project File	12 KB

Using the GPIO PYNQ Class

- Open Jupyter lab, drag and drop those 2 files in.



Using the GPIO PYNQ Class

- Open Jupyter lab, drag and drop those 2 files in.
- Create a new notebook and write in these lines

```
from pynq import Overlay  
ol = Overlay("./KV260_GPIO.bit")
```

```
from pynq import GPIO
```

```
A = GPIO(GPIO.get_gpio_pin(0), 'out')
```

```
B = GPIO(GPIO.get_gpio_pin(1), 'out')
```

```
C = GPIO(GPIO.get_gpio_pin(2), 'out')
```

```
Z = GPIO(GPIO.get_gpio_pin(3), 'in')
```


Using the GPIO PYNQ Class

- Open Jupyter lab, drag and drop those 2 files in.
- Create a new notebook and write in these lines
- You can then test by using these lines.

```
[ ]: A.write(1)  
     B.write(1)  
     C.write(0)
```

```
[ ]: Z.read()
```

Using the GPIO PYNQ Class

- Open Jupyter lab, drag and drop those 2 files in.
- Create a new notebook and write in these lines
- You can then test by using these lines.
- With the inputs in the figure, you should get the output 1

```
[1]: from pynq import Overlay  
ol = Overlay("./KV260_GPIO.bit")
```

```
[2]: from pynq import GPIO
```

```
[3]: A = GPIO(GPIO.get_gpio_pin(0), 'out')  
B = GPIO(GPIO.get_gpio_pin(1), 'out')  
C = GPIO(GPIO.get_gpio_pin(2), 'out')
```

```
[4]: Z = GPIO(GPIO.get_gpio_pin(3), 'in')
```

```
[5]: A.write(1)  
B.write(1)  
C.write(0)
```

```
[6]: Z.read()
```

```
[6]: 1
```