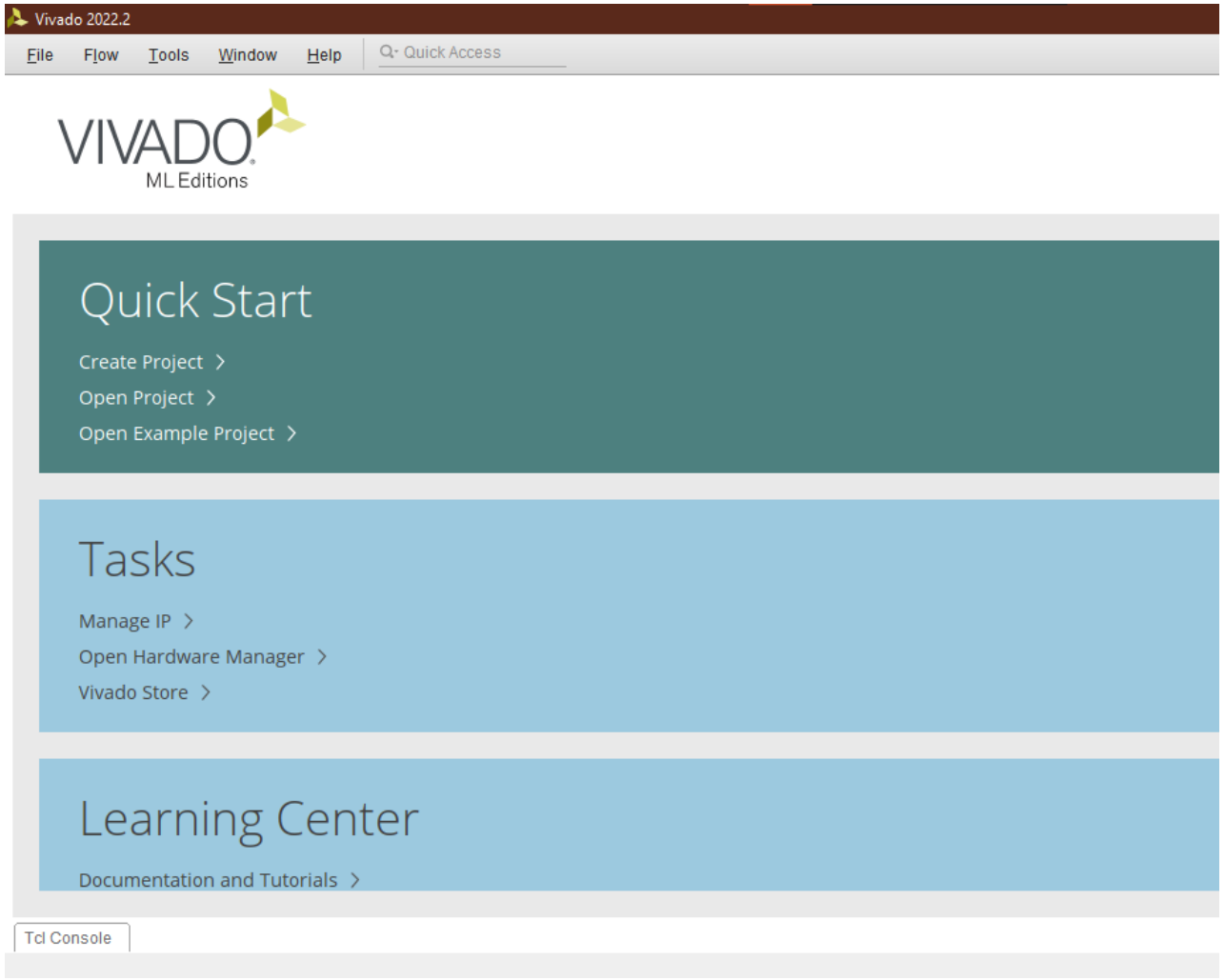
PYNQ GS GPIO
WITH
KRIA KV260

Table of Contents

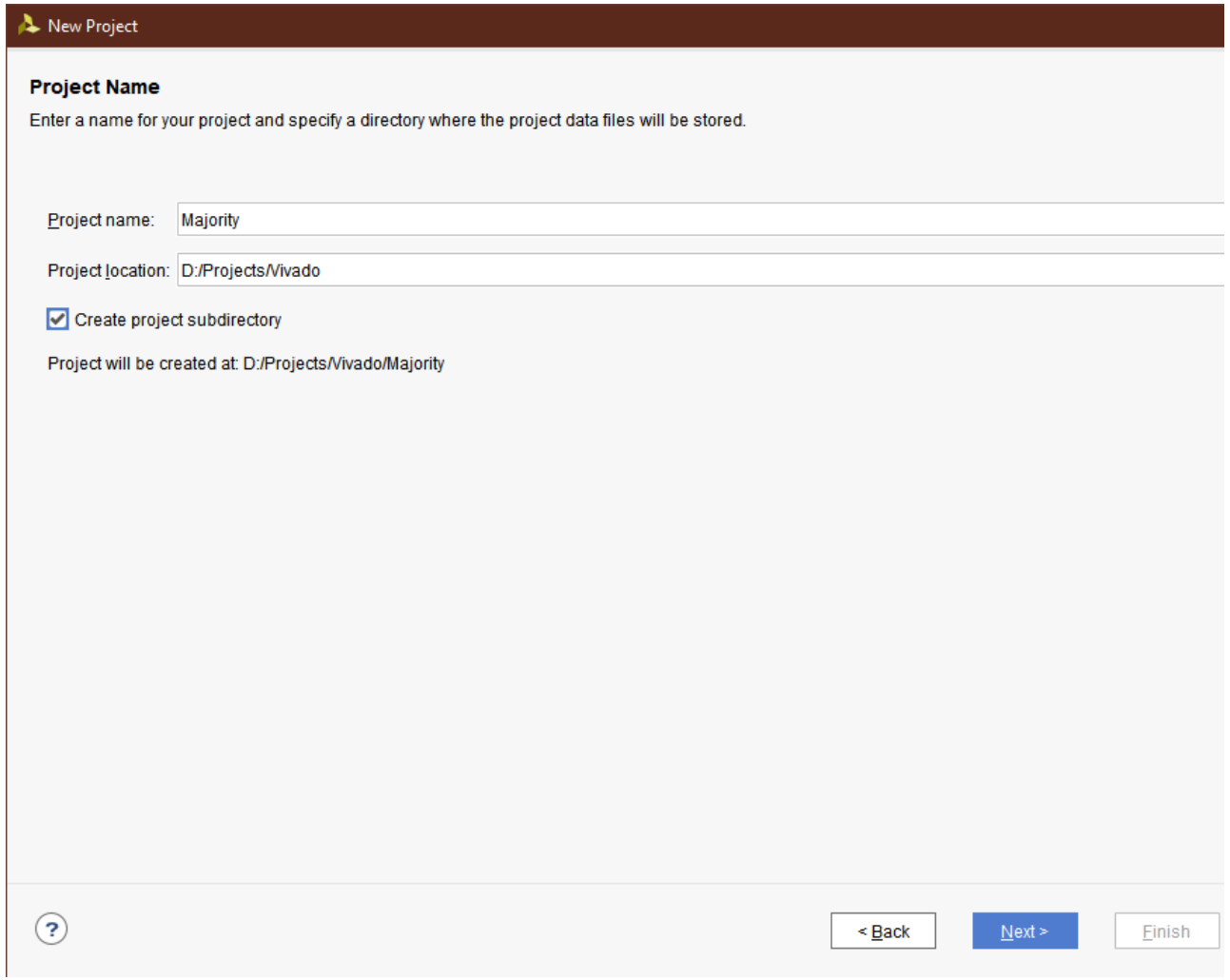
Hardware design
Project creation

- Open Vivado
- Create Project



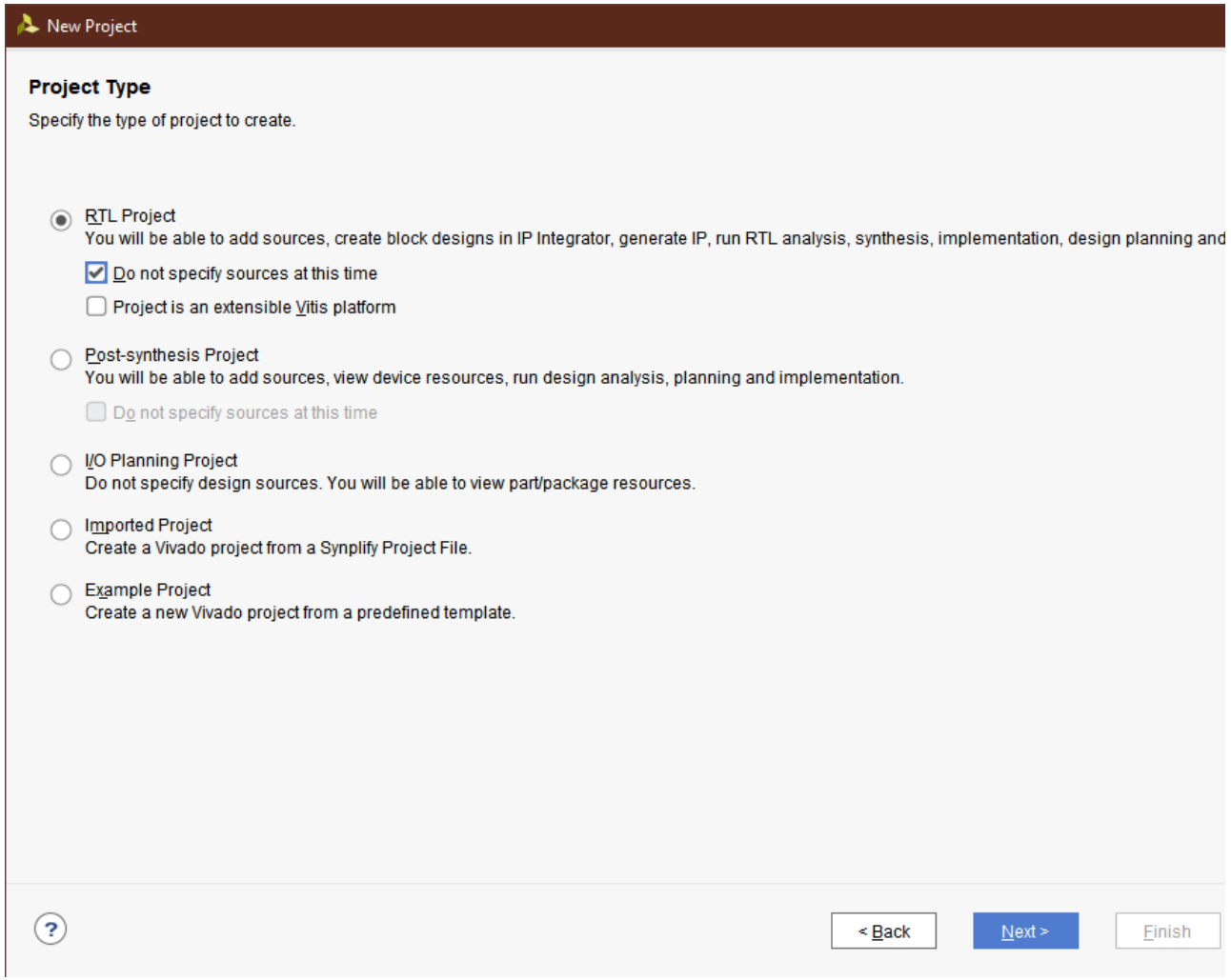
Hardware design
Project creation

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



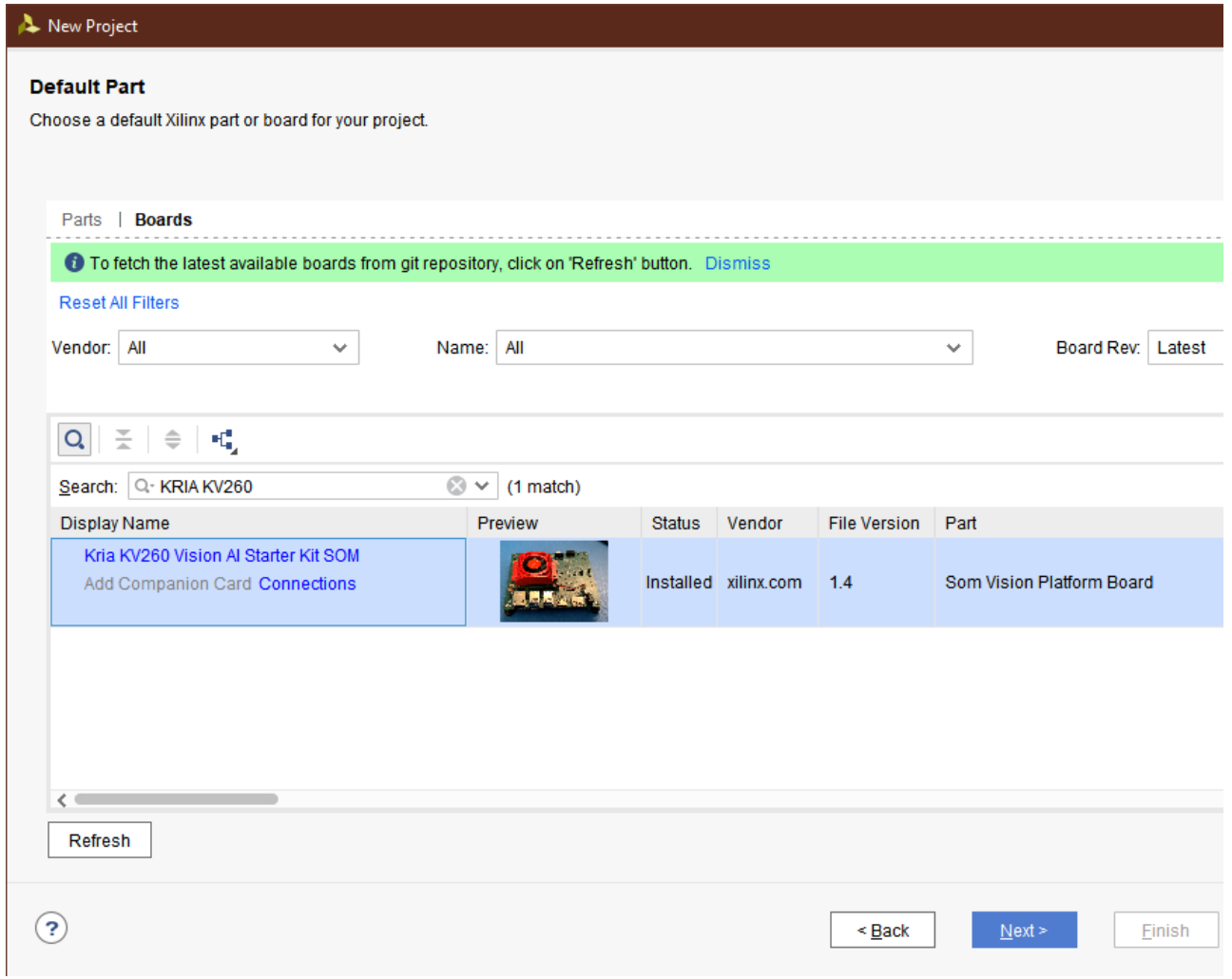
Hardware design
Project creation

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



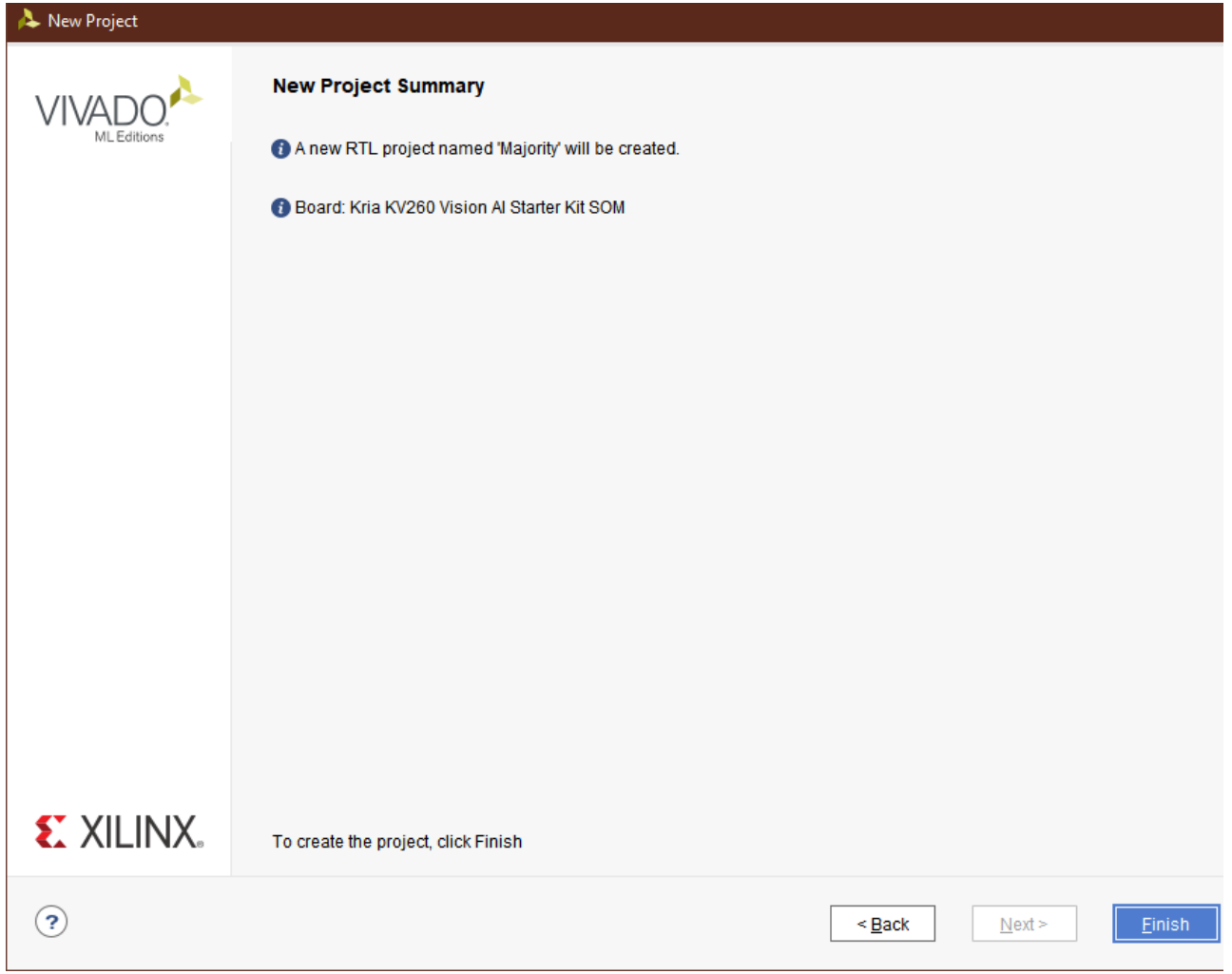
Hardware design
Project creation

- 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”



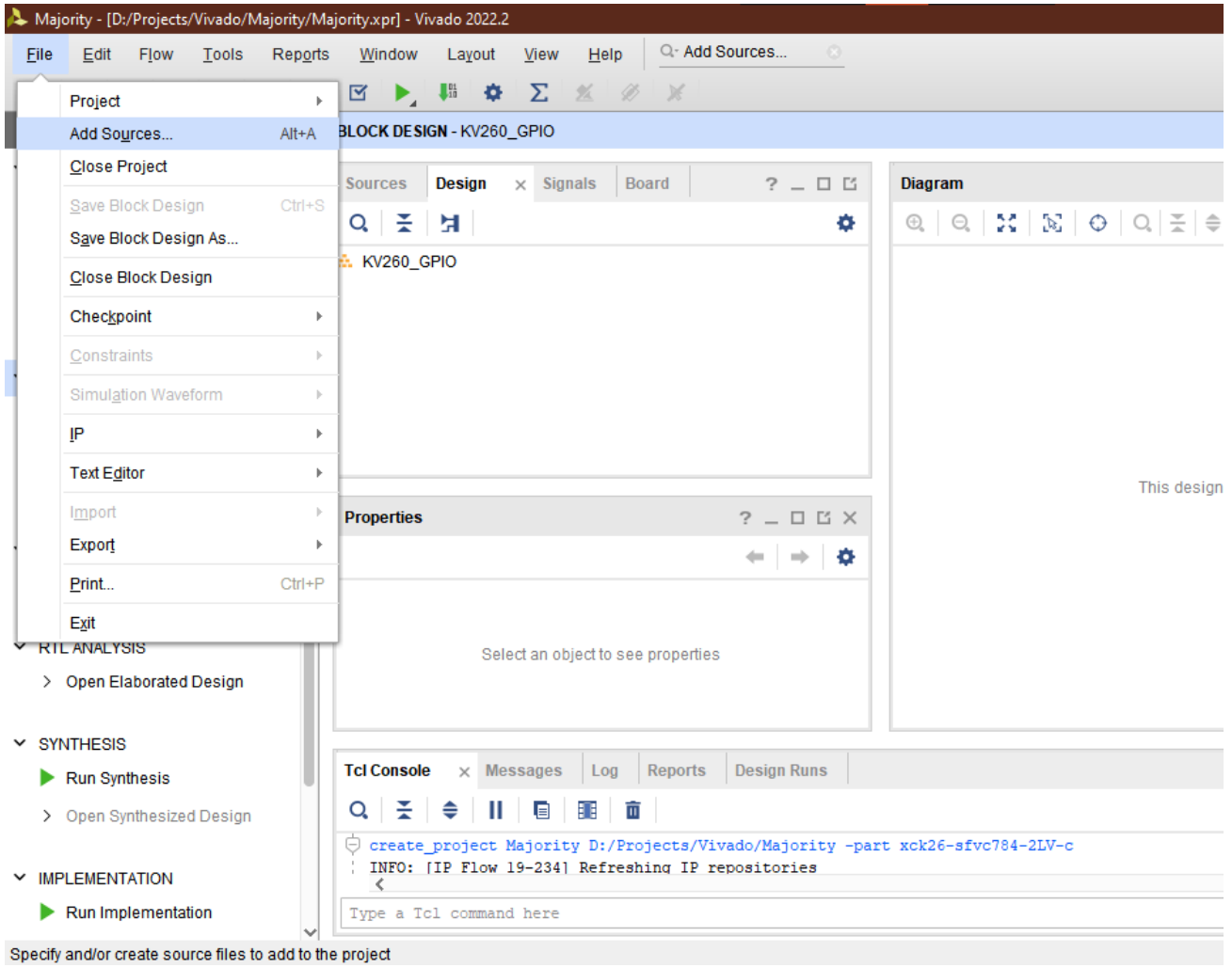
Hardware design
Project creation

- 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



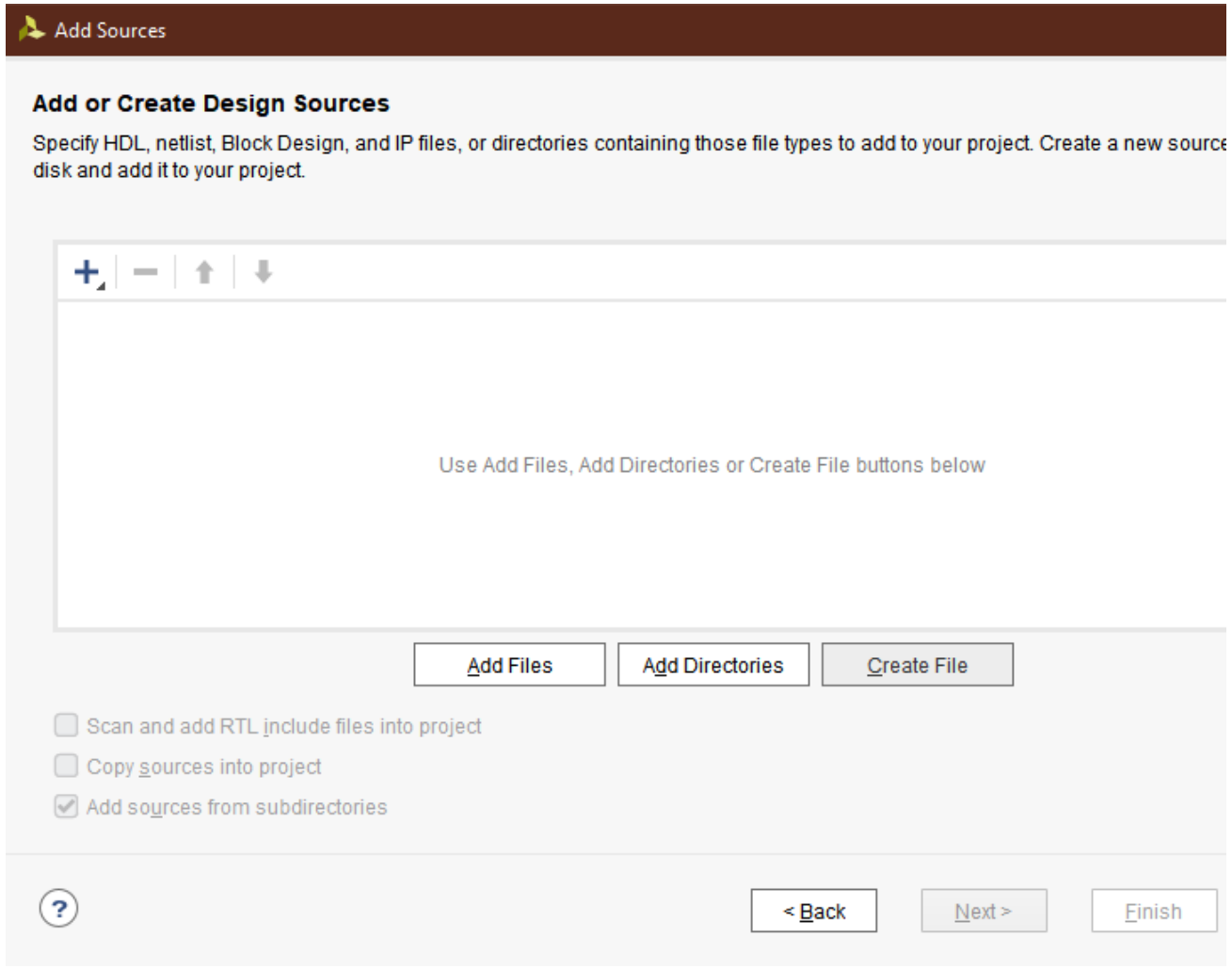
Hardware design
Add design source

- File → Add Sources or (ALT + A)



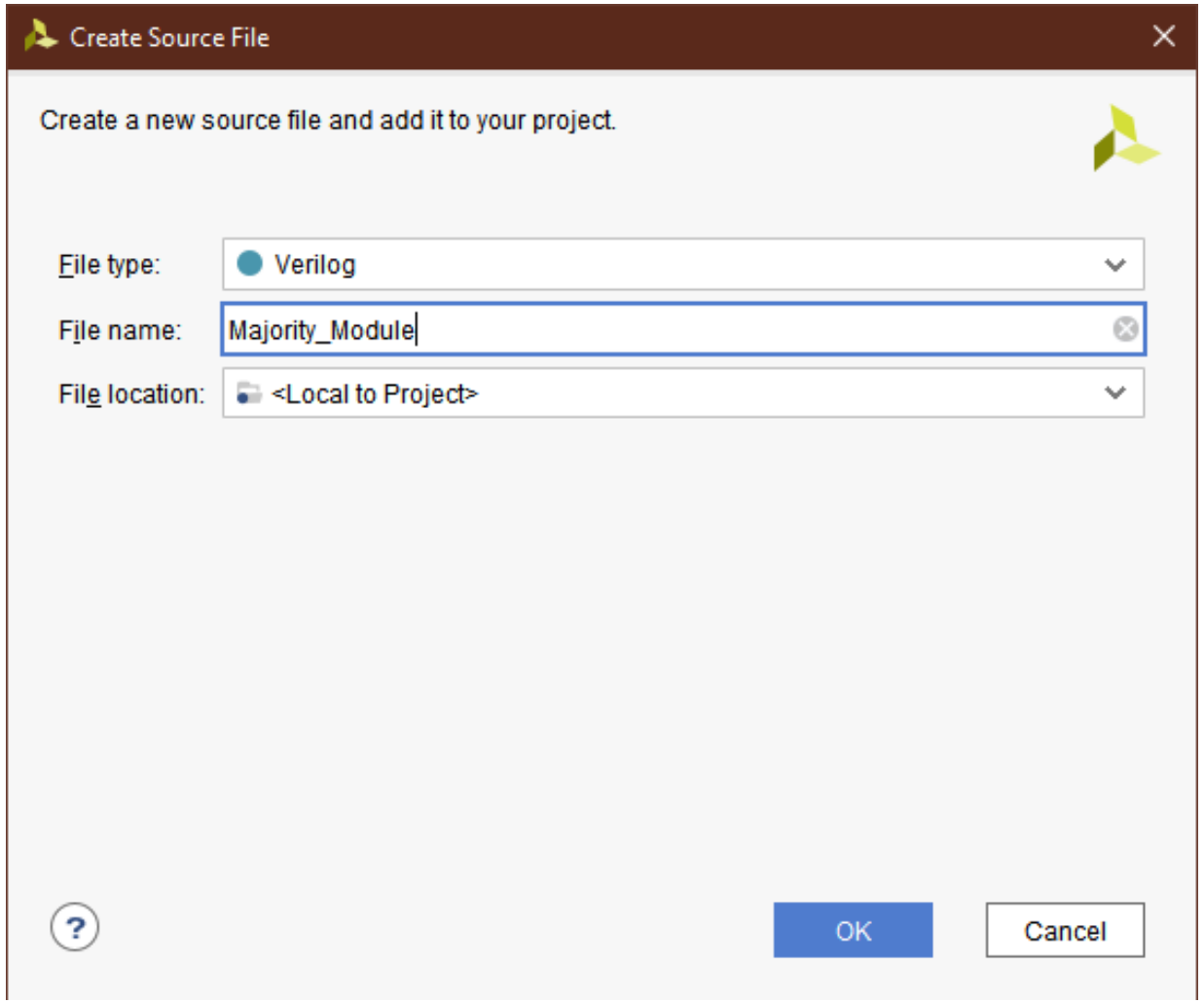
Hardware design
Add design source

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



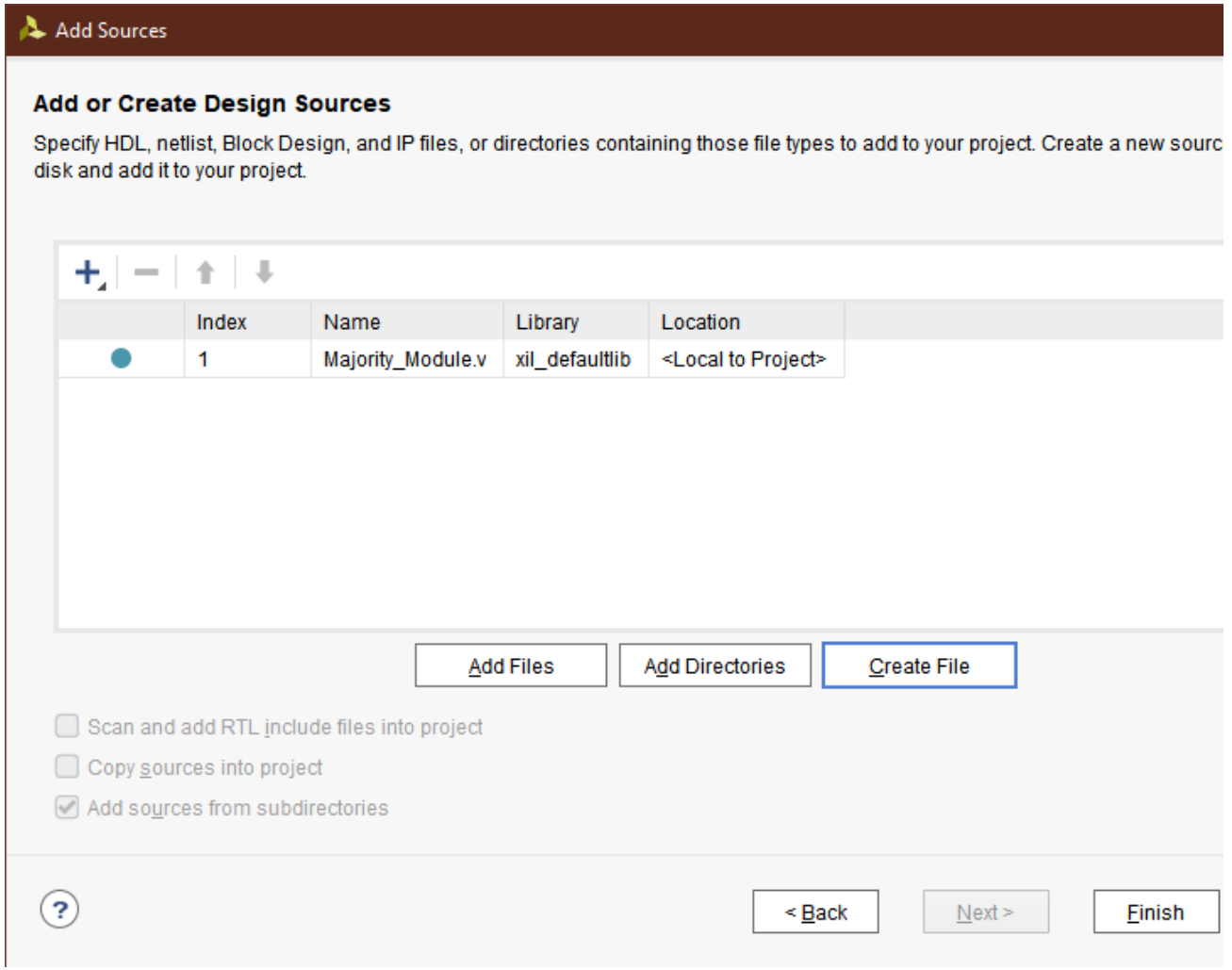
Hardware design
Add design source

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



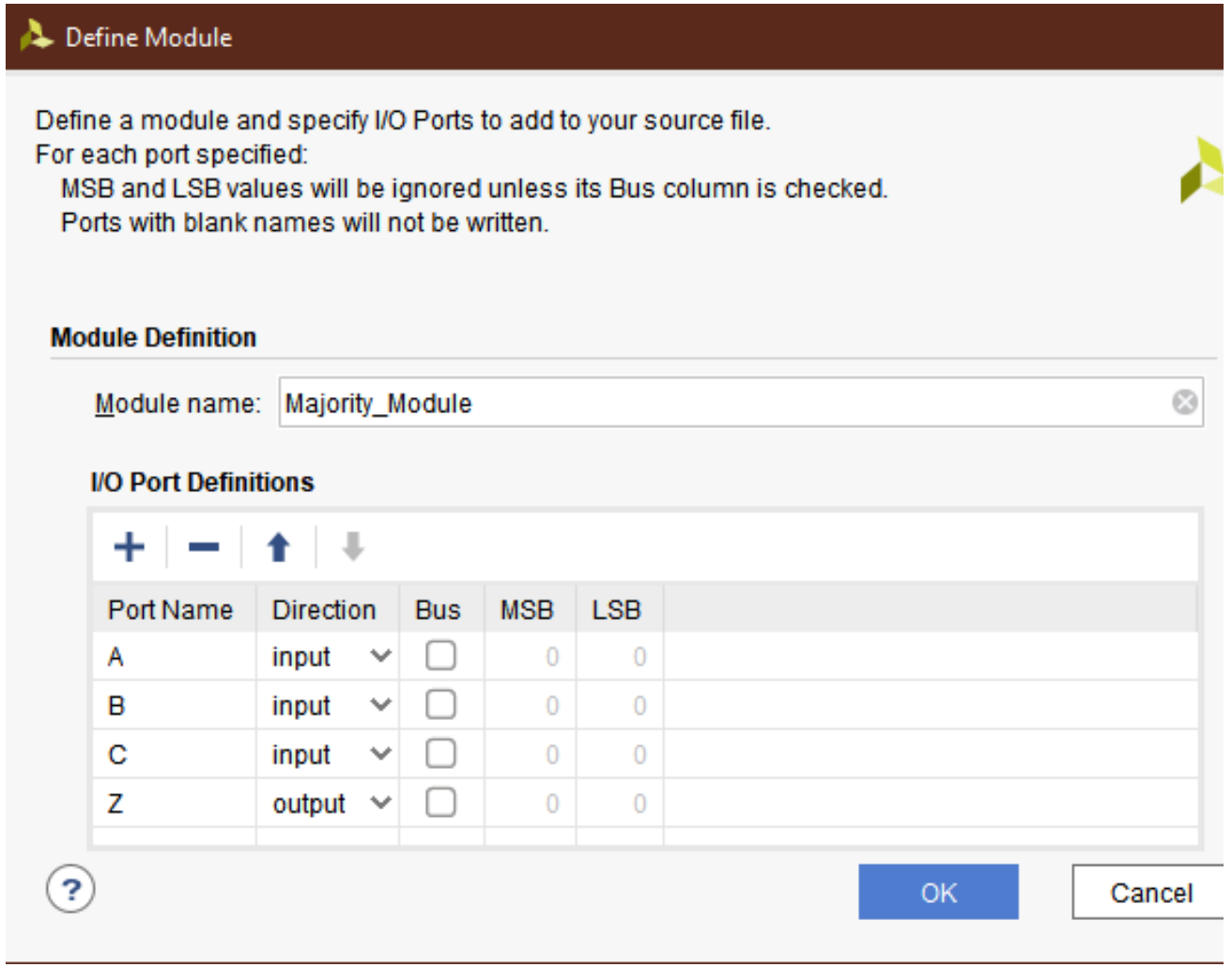
Hardware design
Add design source

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



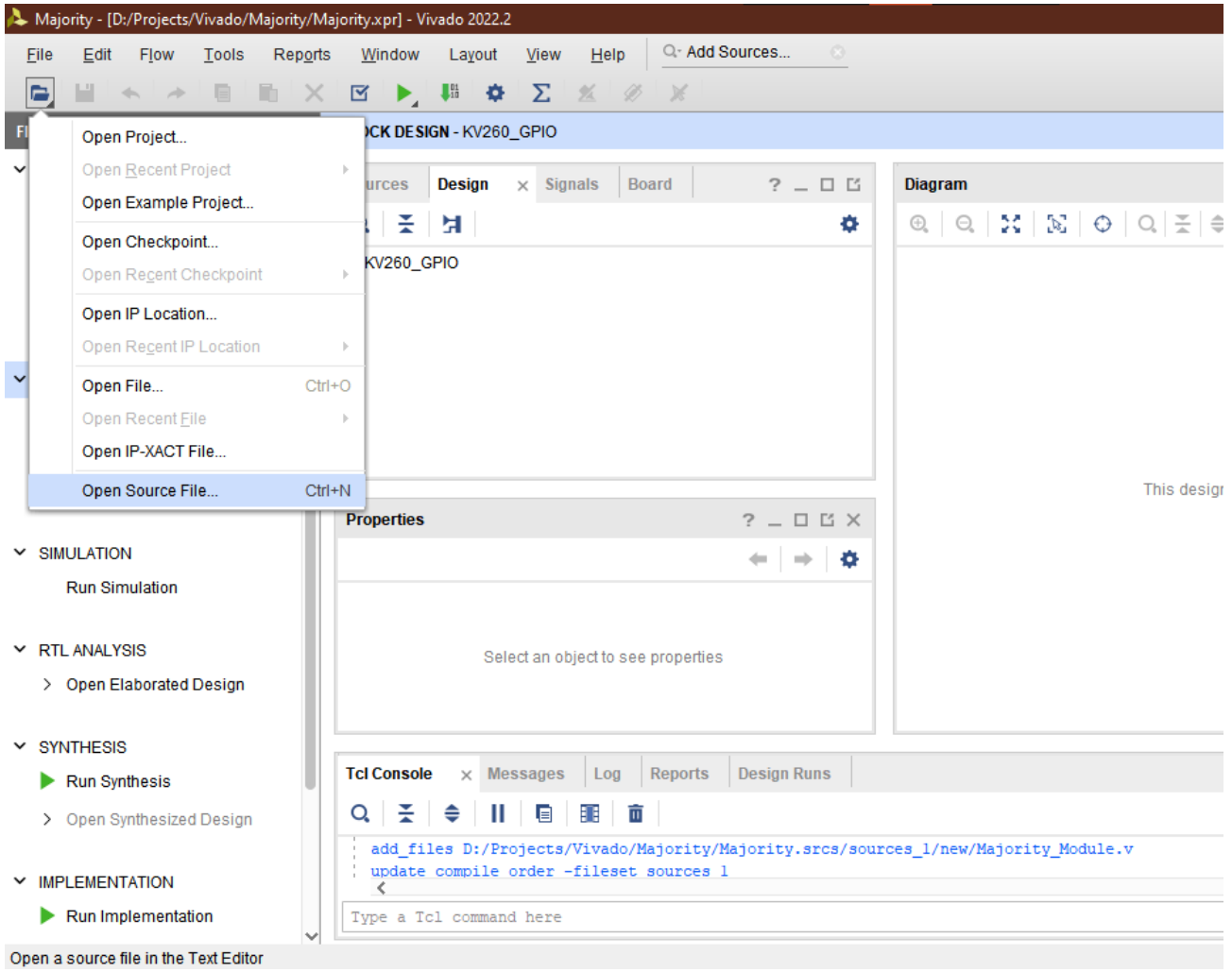
Hardware design
Add design source

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



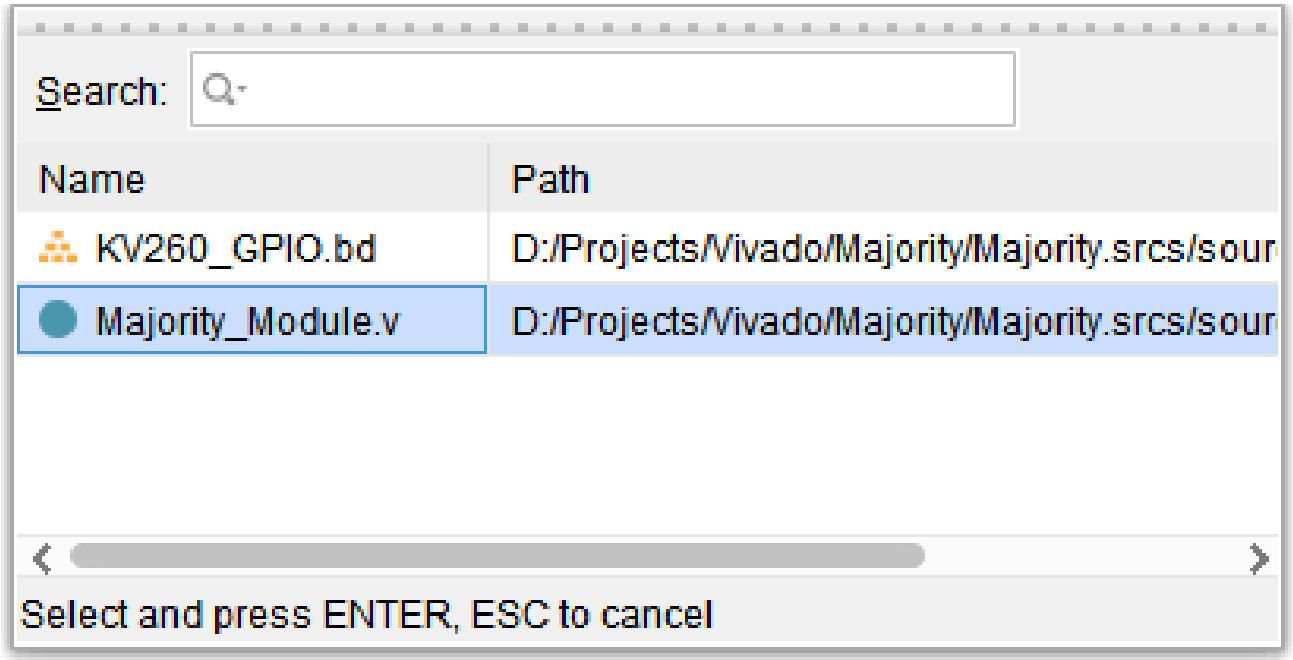
Hardware design
Add design source

- 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.

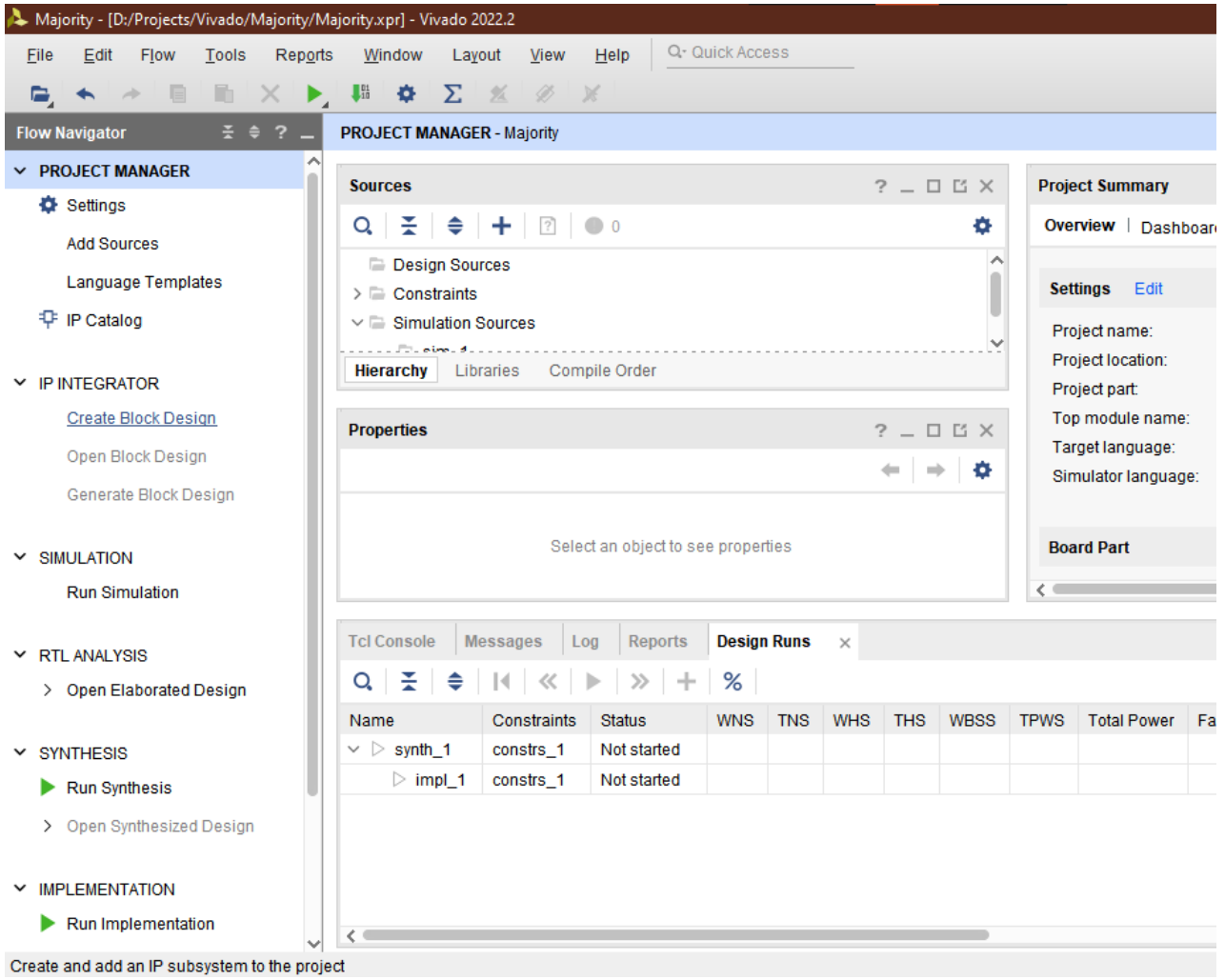
```
module Majority_Module(
    input A, B, C
    output Z
    );

    and(and0, A, B);
    and(and1, B, C);
    and(and2, A, C);
    or(OUTOUT, and0, and1, and2);

endmodule
```

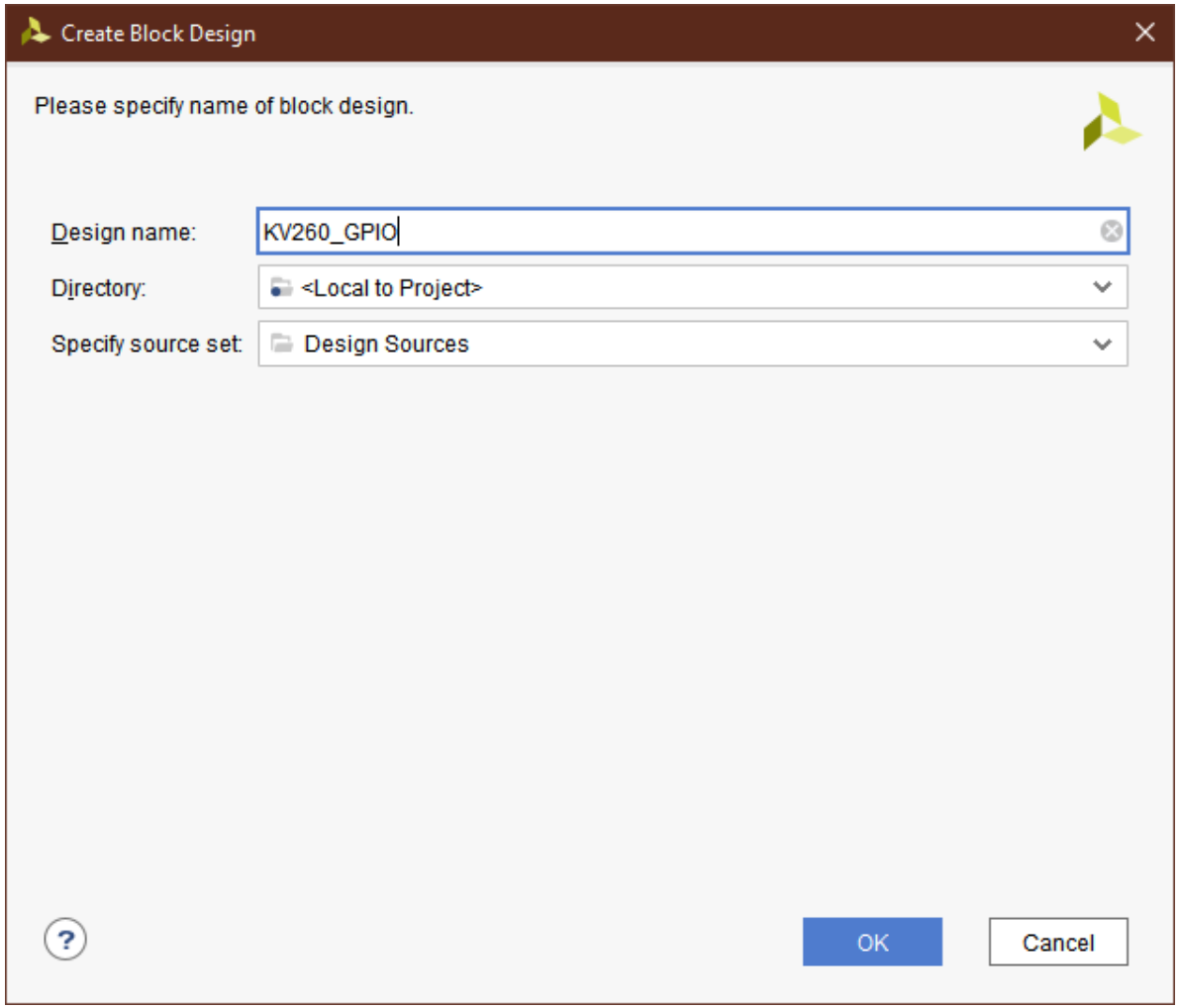
Hardware design
Block design

- Click “Create Block Design”



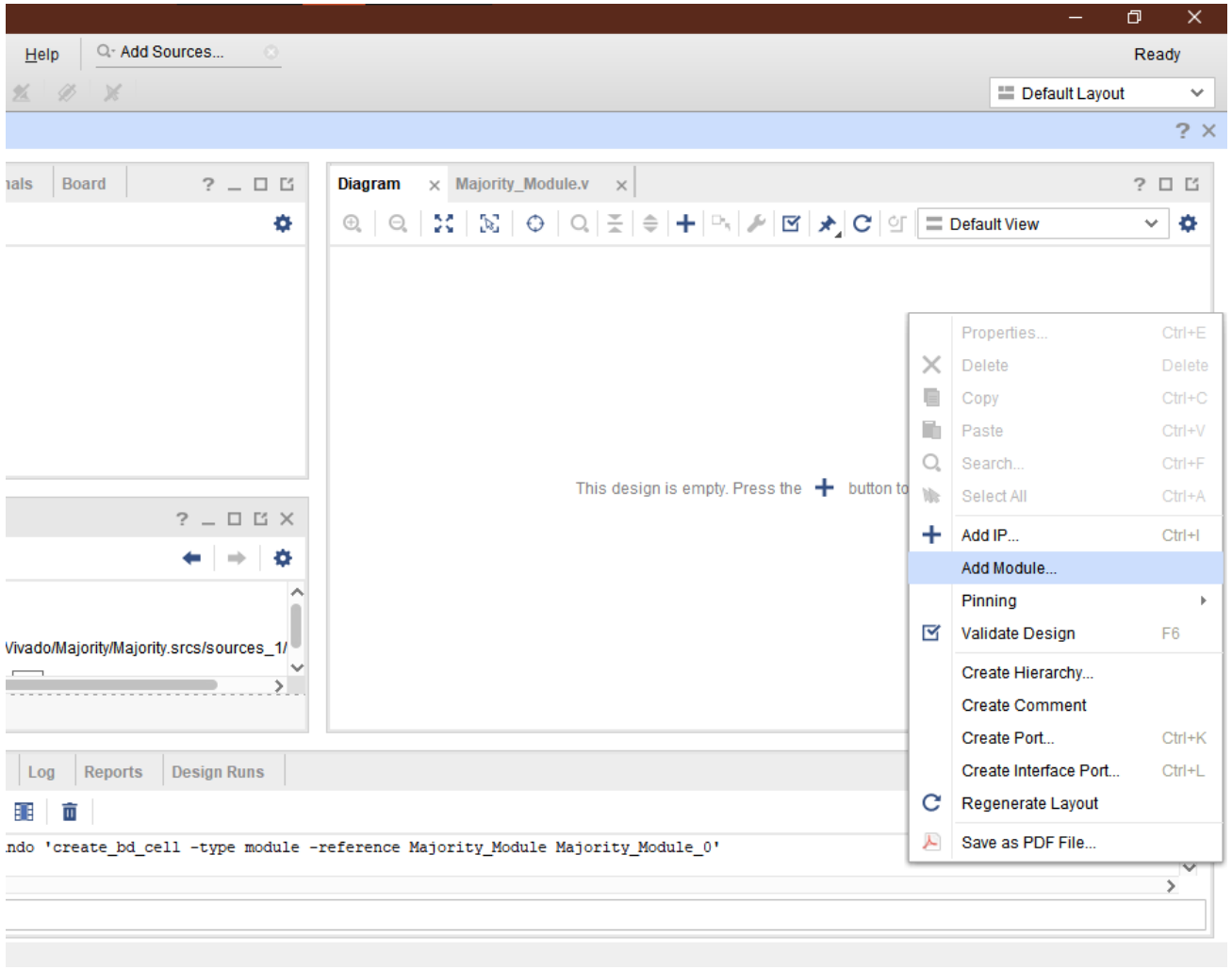
Hardware design
Block design

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



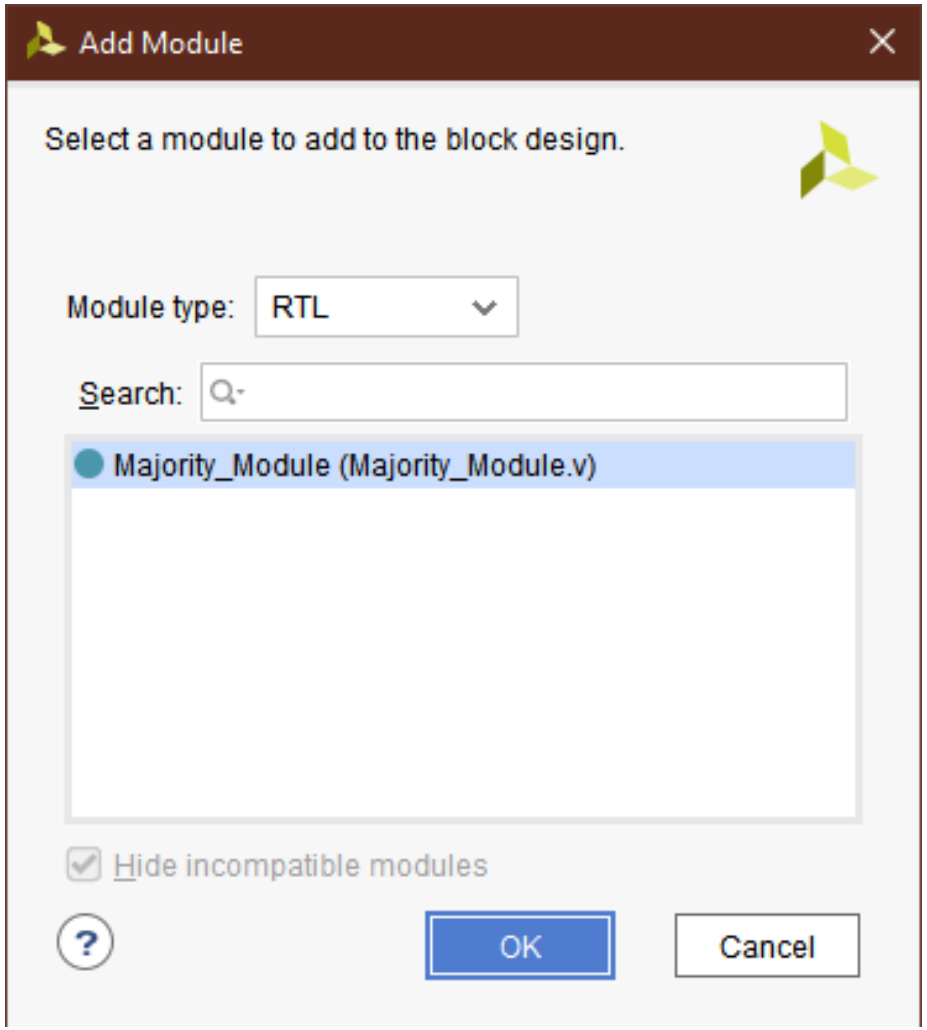
Hardware design
Block design

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



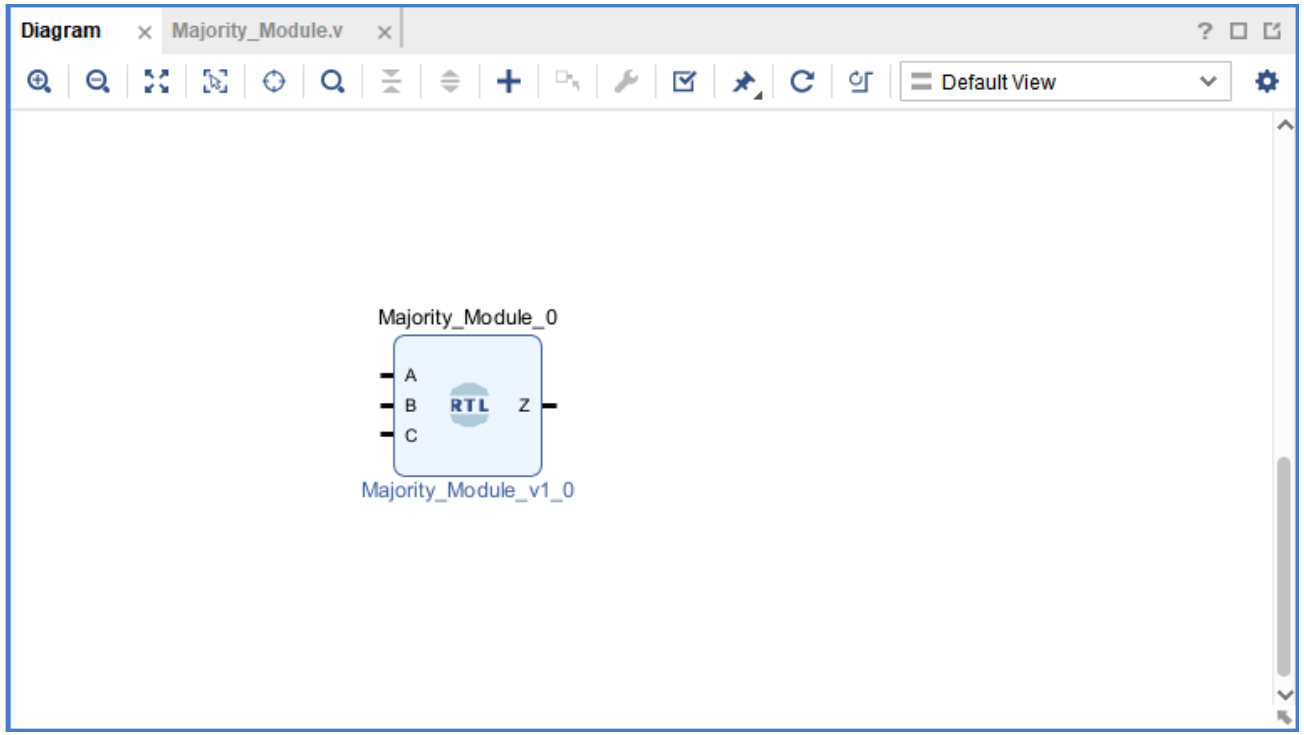
Hardware design
Block design

- 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”



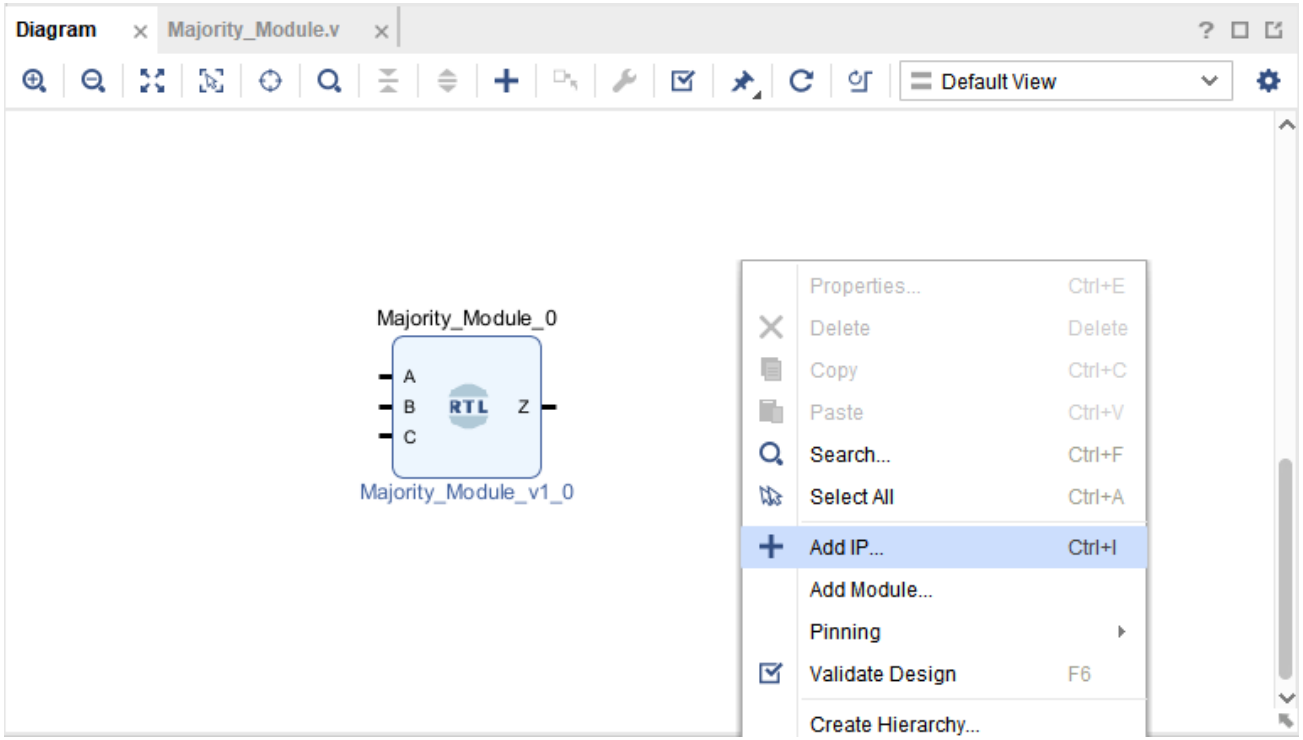
Hardware design
Block design

- 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”



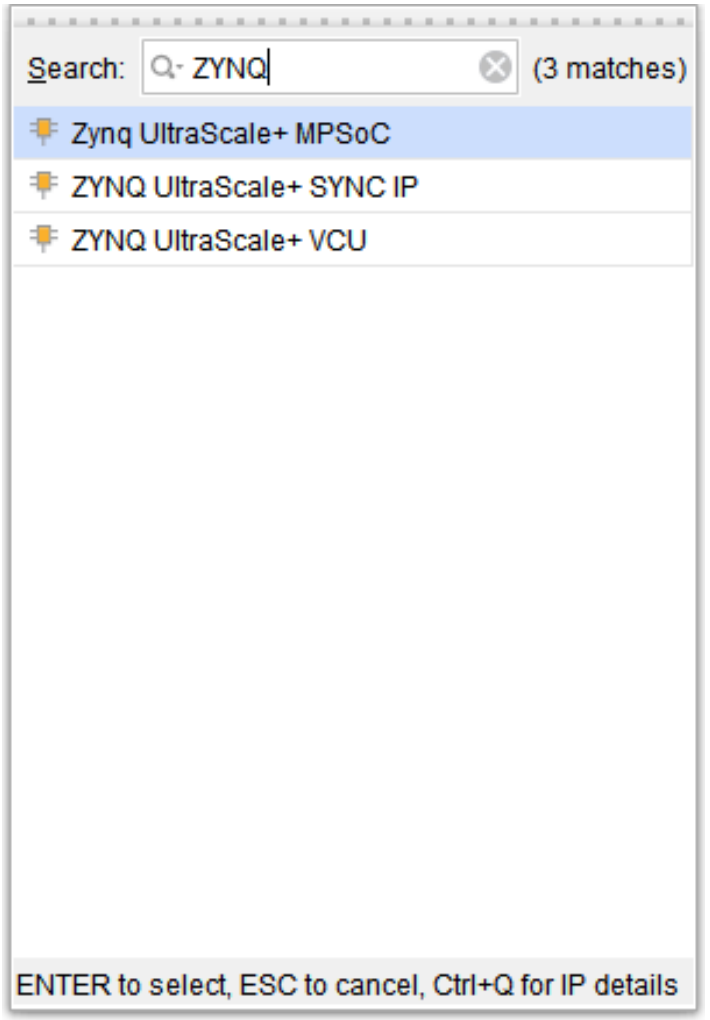
Hardware design
Block design

- 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”



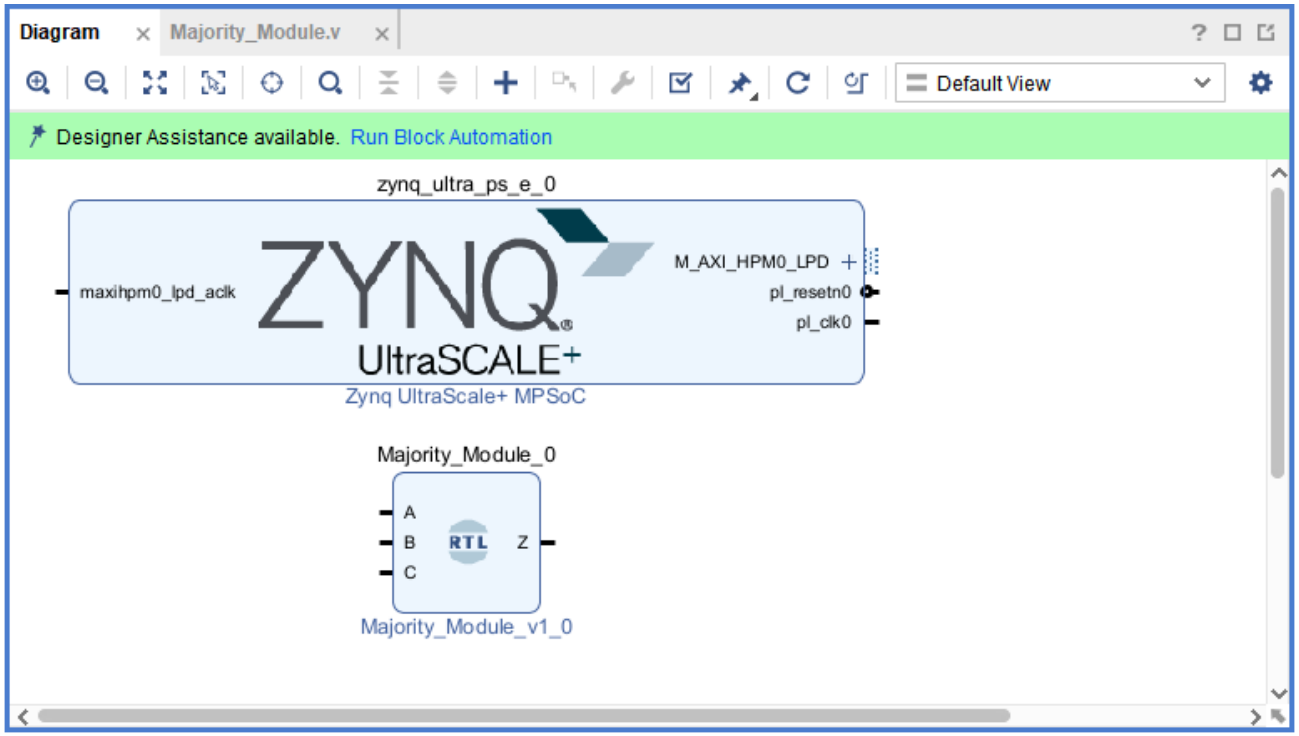
Hardware design
Block design

- 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



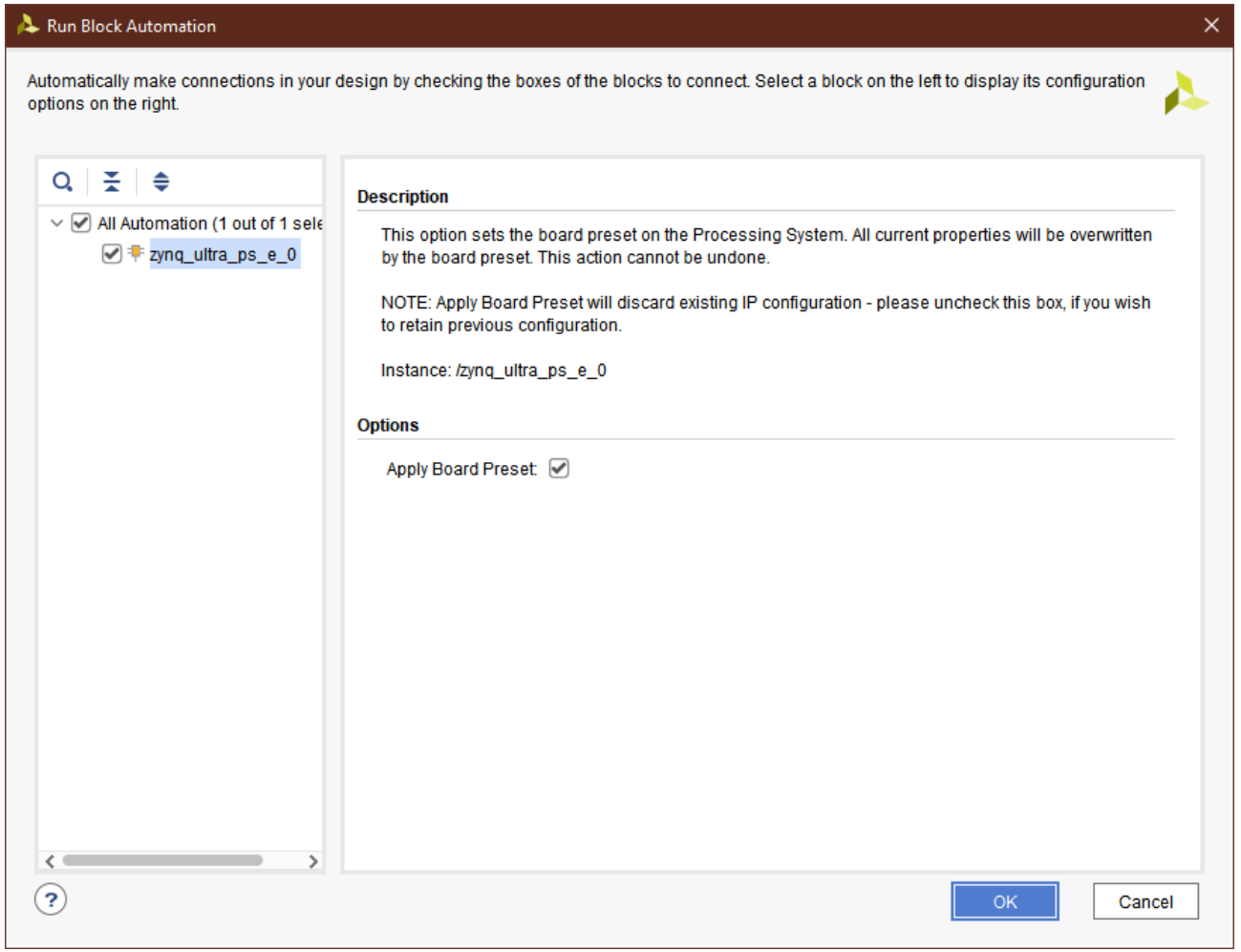
Hardware design
Block design

- 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



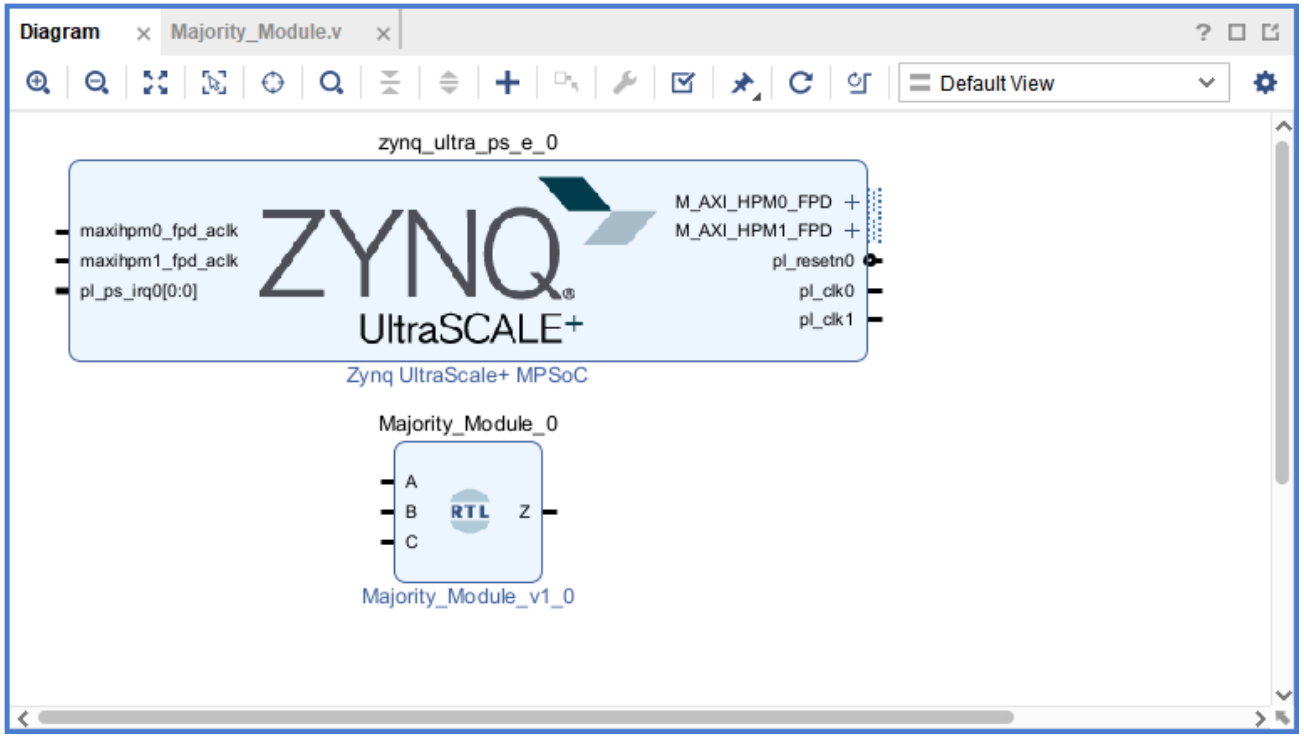
Hardware design
Block design

- 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



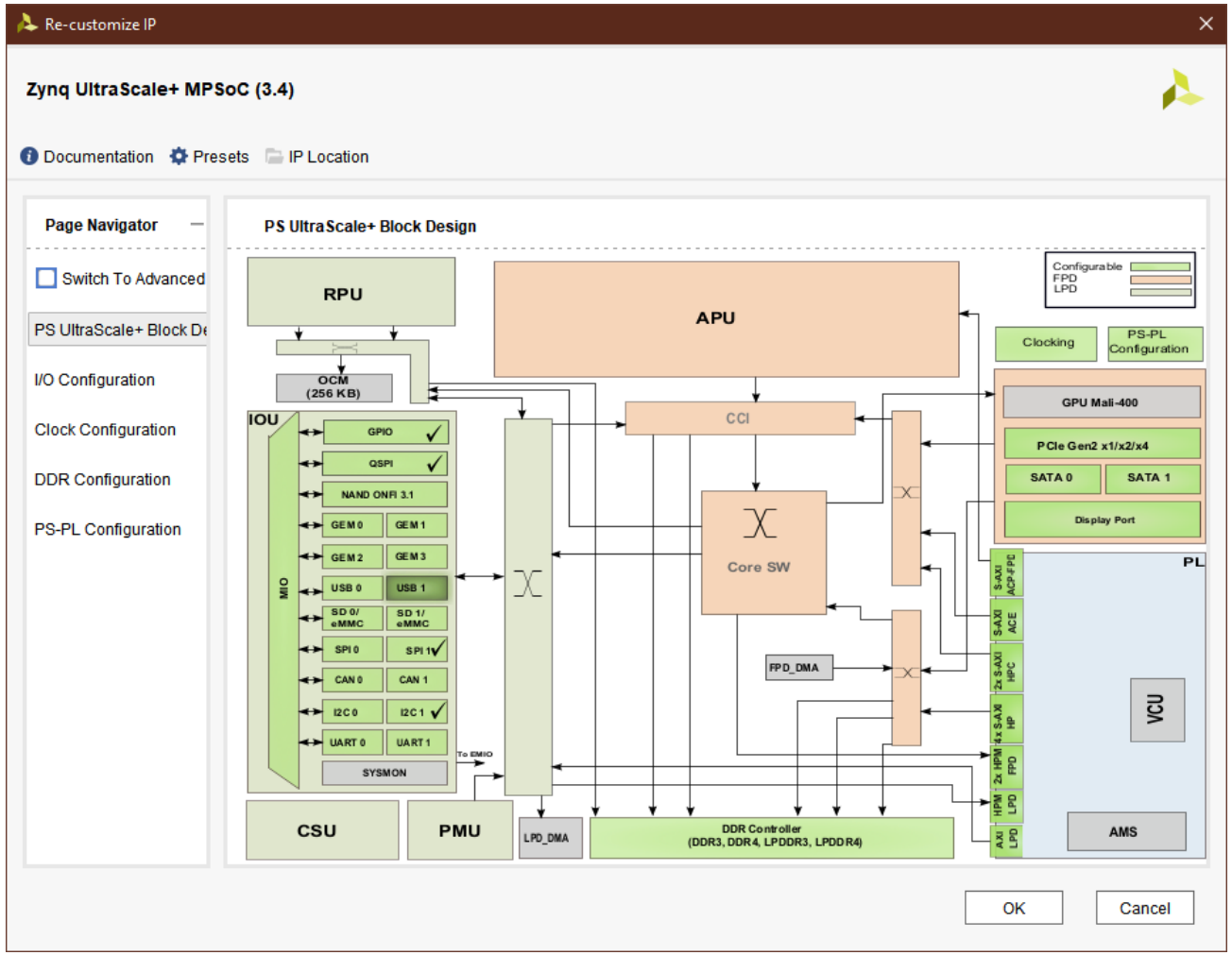
Hardware design
Block design

- 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



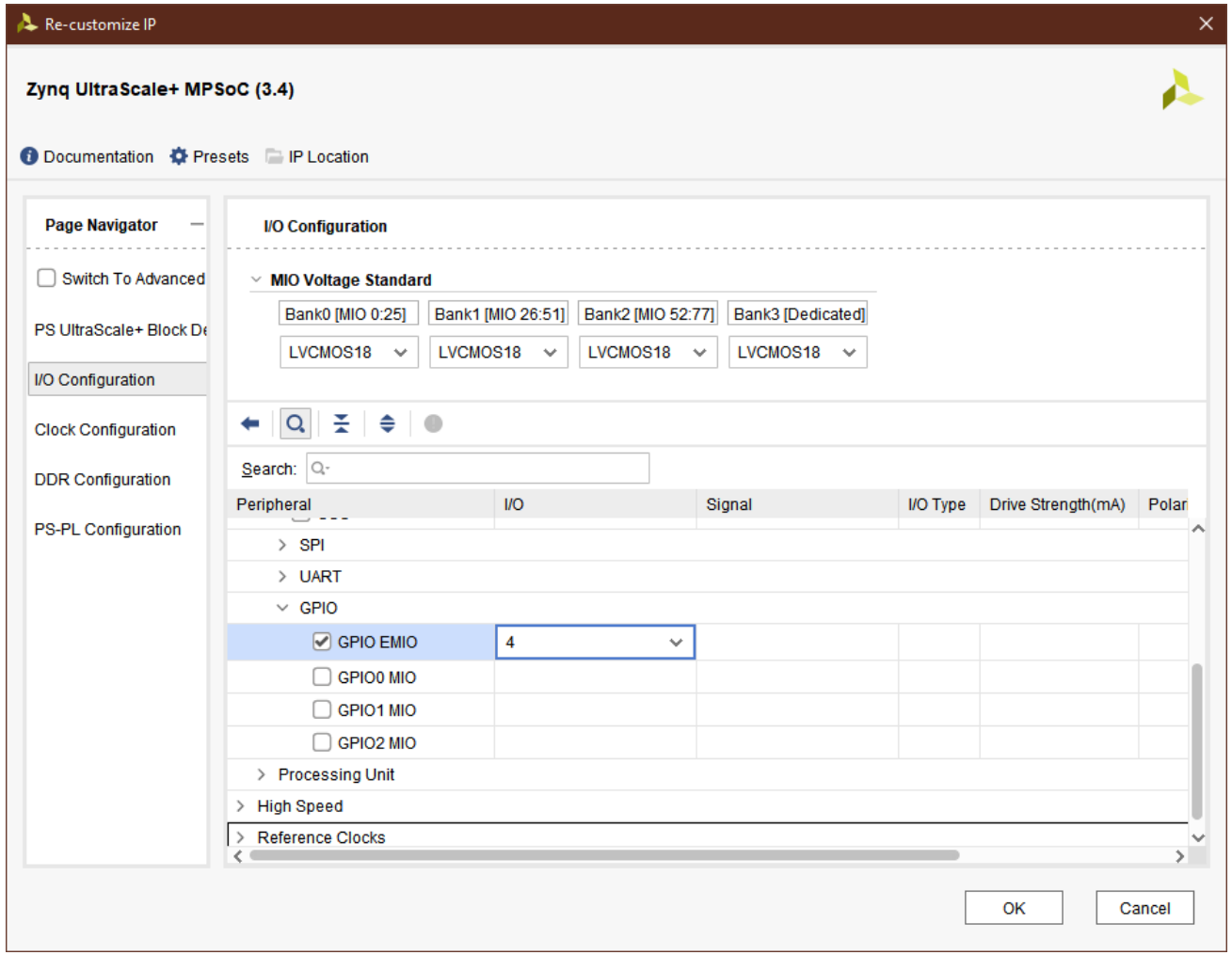
Hardware design
Block design

- 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



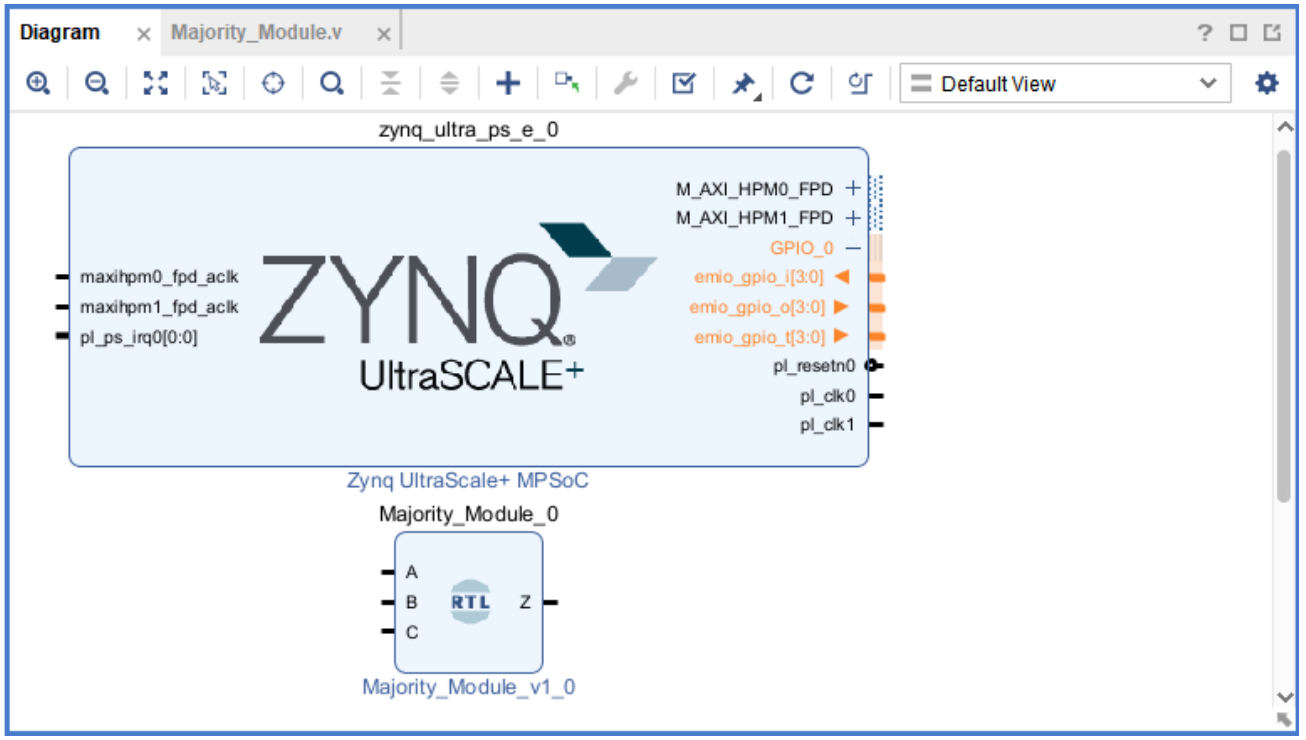
Hardware design
Block design (Cont.)

- 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”



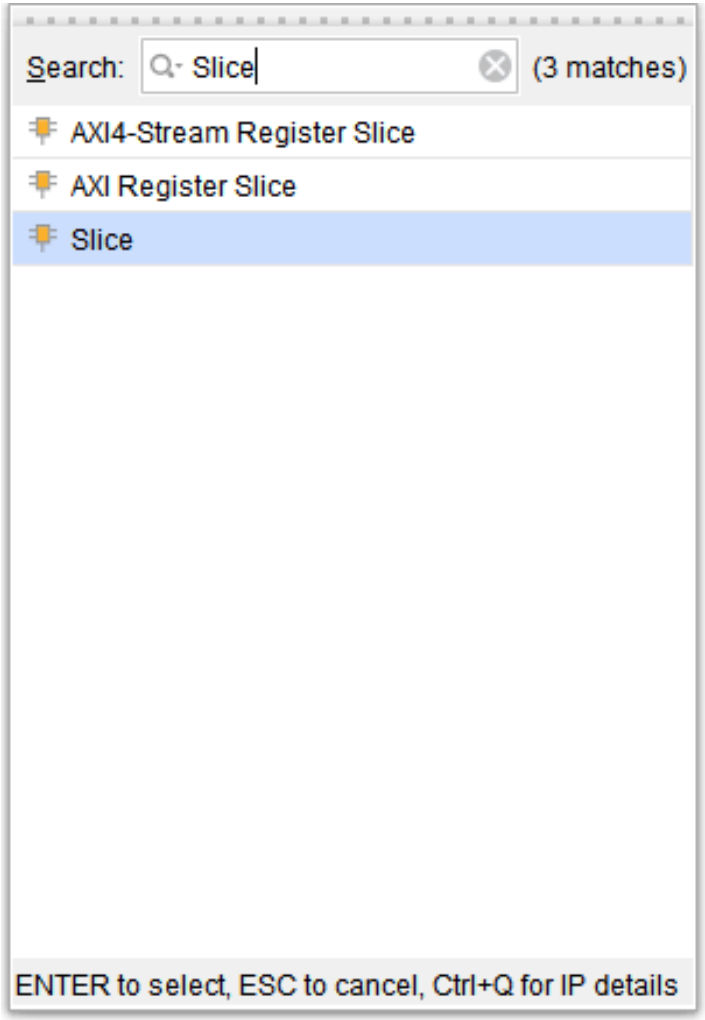
Hardware design
Block design (Cont.)

- 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”



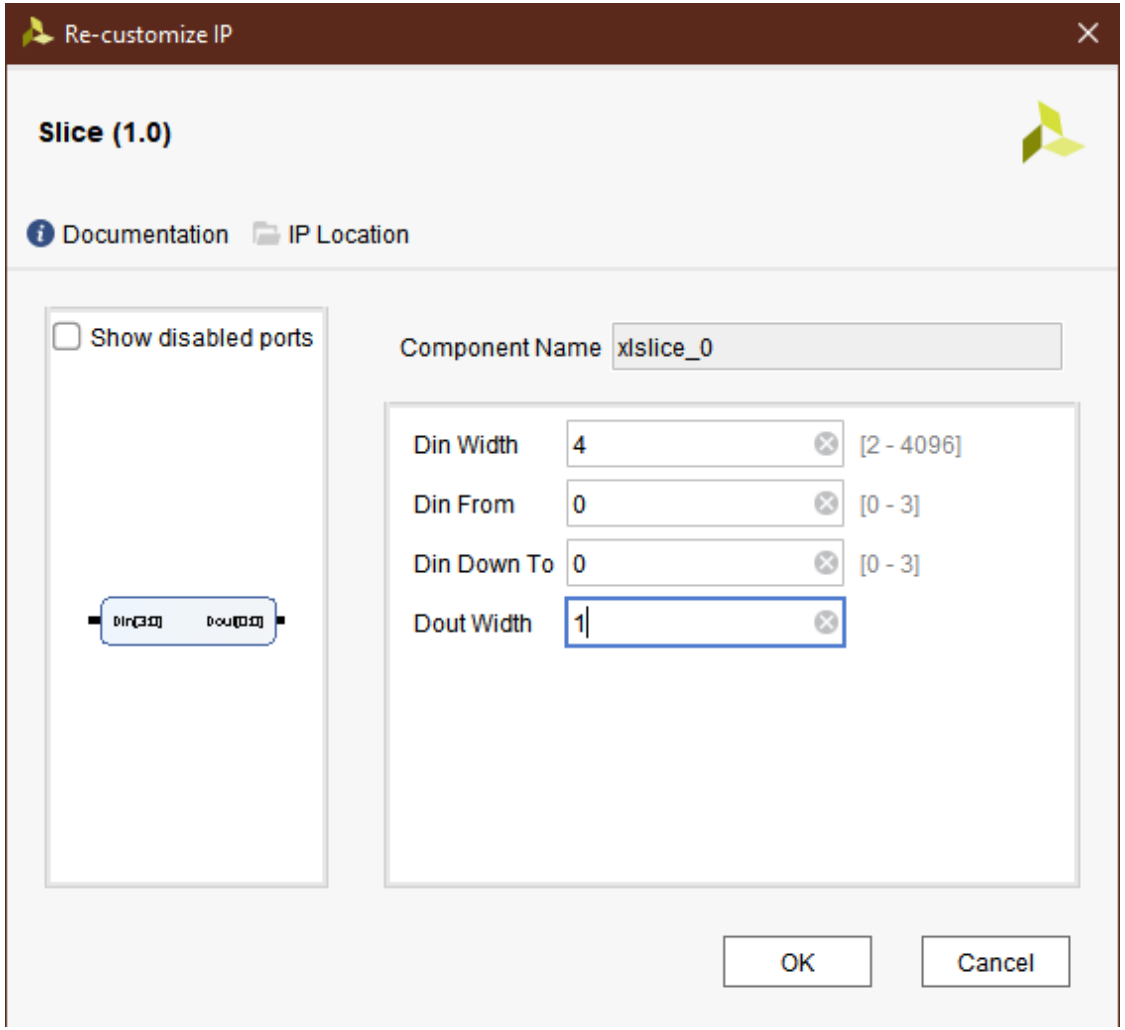
Hardware design
Block design (Cont.)

- 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



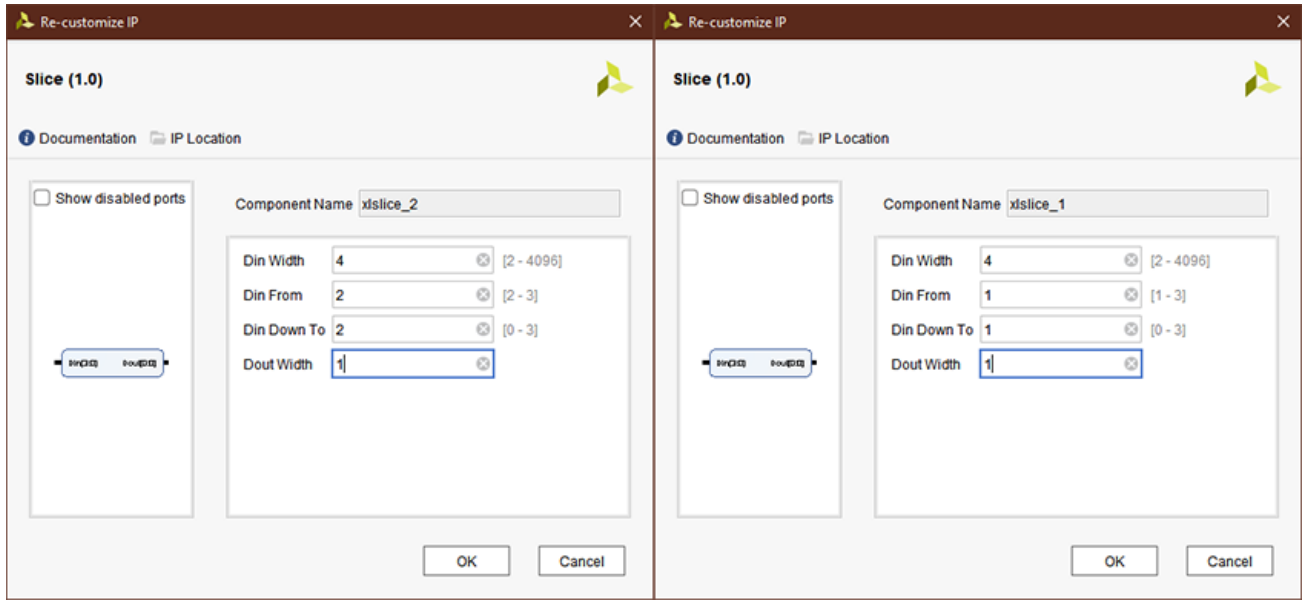
Hardware design
Block design (Cont.)

- 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



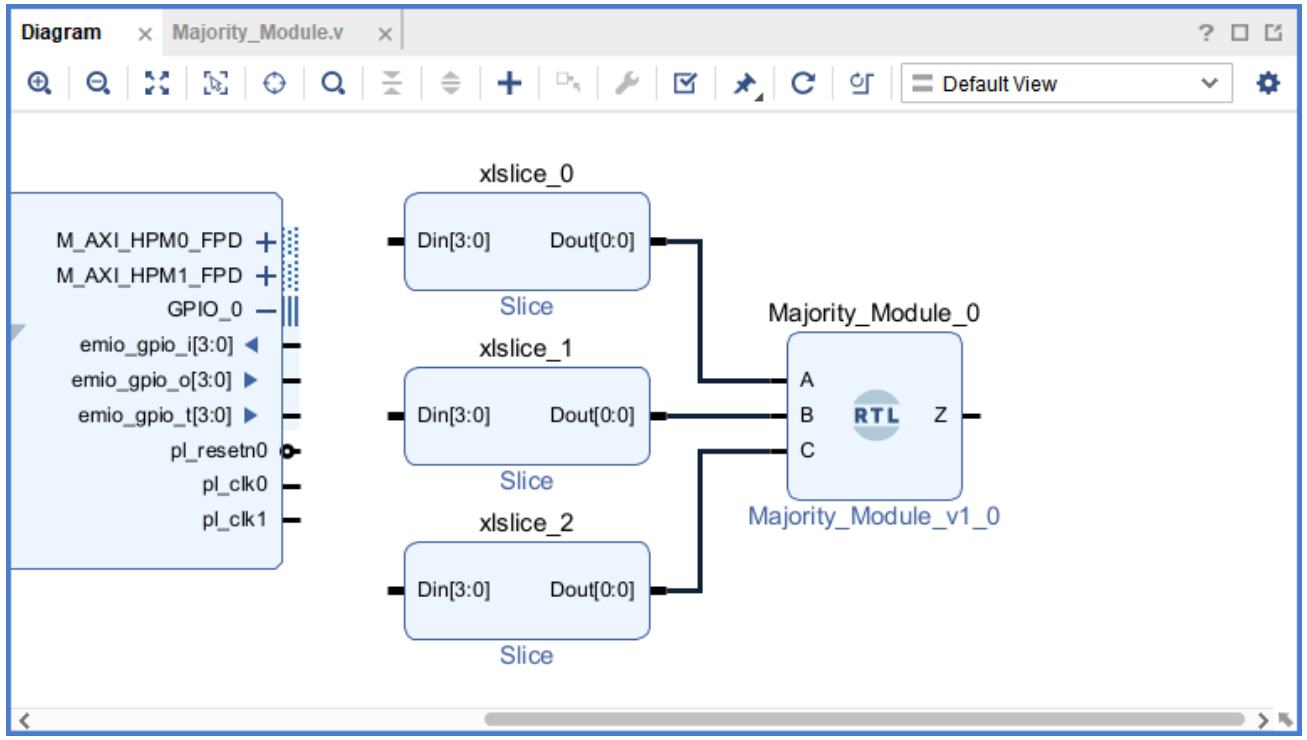
Hardware design
Block design (Cont.)

- 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



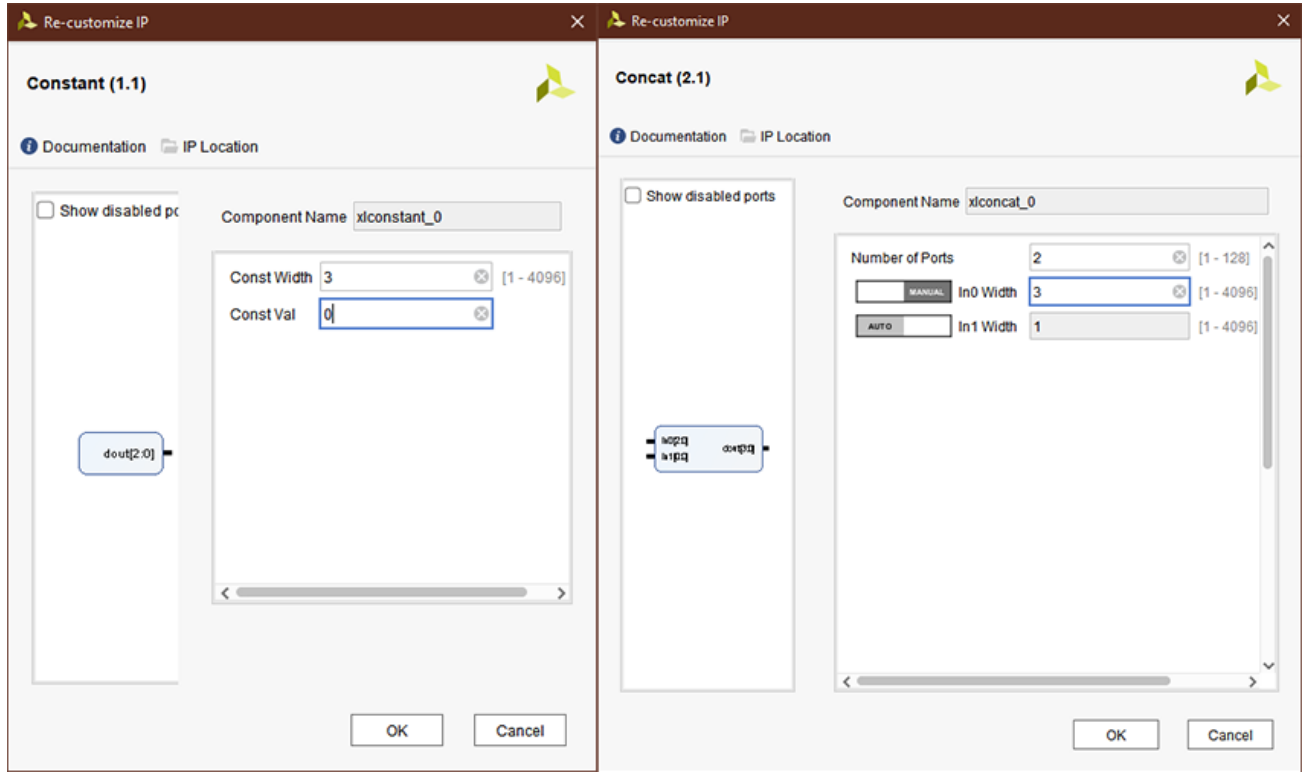
Hardware design
Block design (Cont.)

- 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



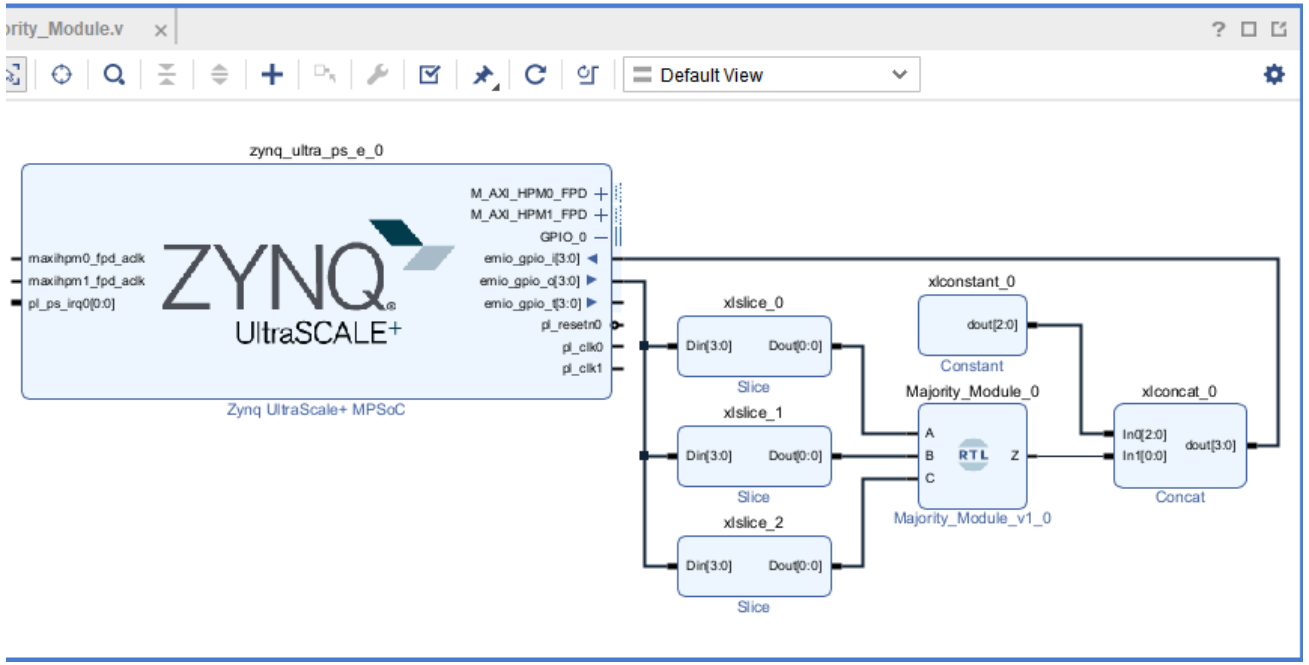
Hardware design
Block design (Cont.)

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



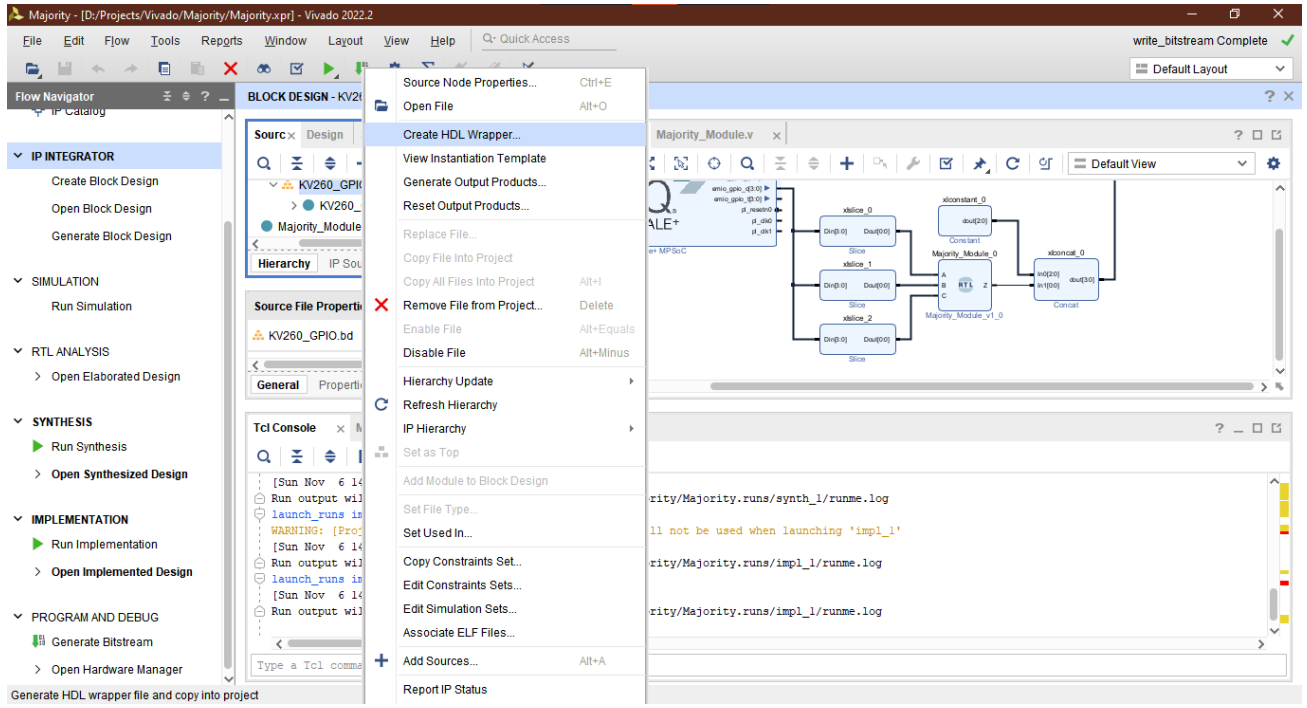
Hardware design
Block design (Cont.)

- 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**



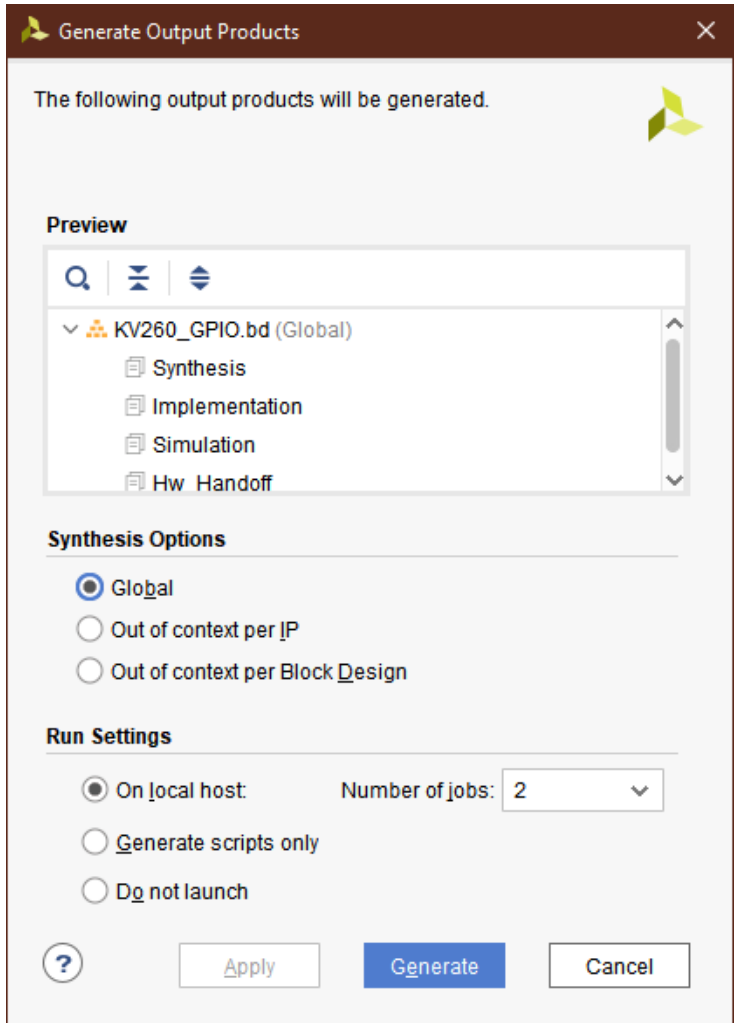
Hardware design
Block design (Cont.)

- 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



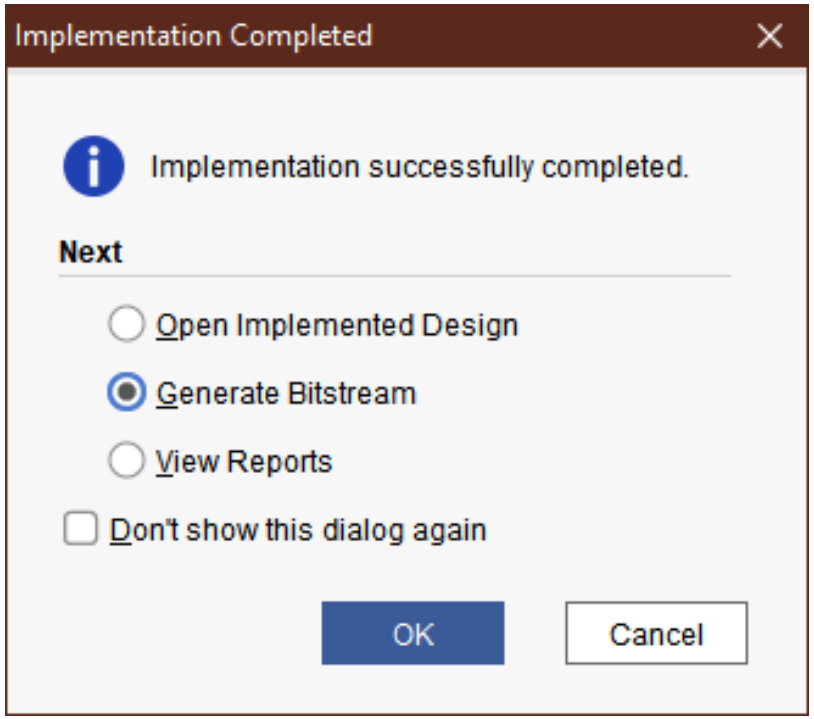
Hardware design
Block design (Cont.)

- 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 clock 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 clock 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

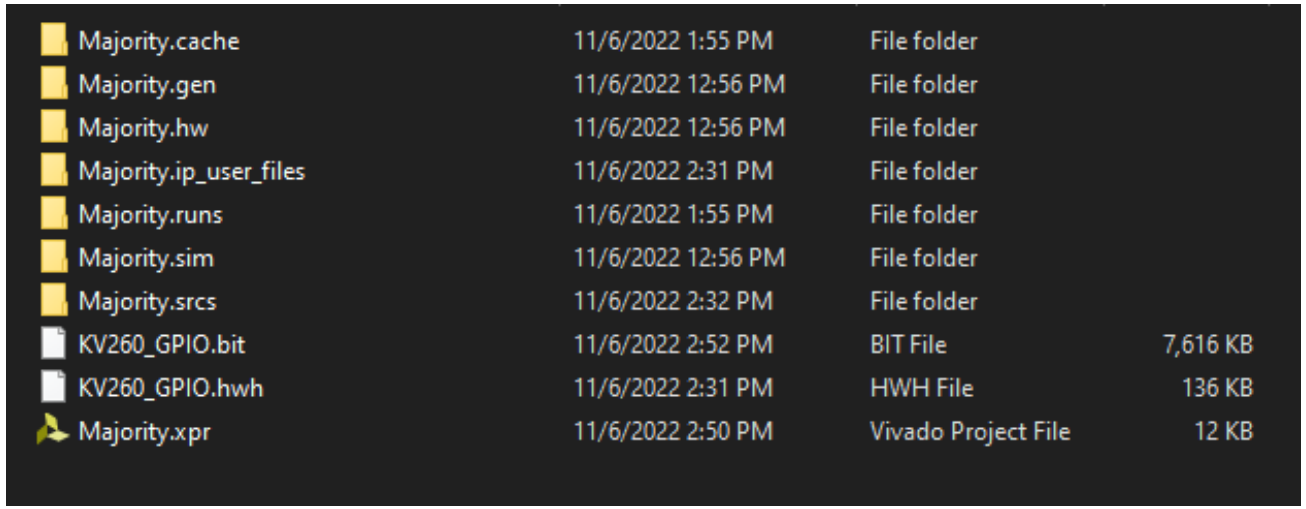
- You'll need 2 files, one will be the hwh file located here

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

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

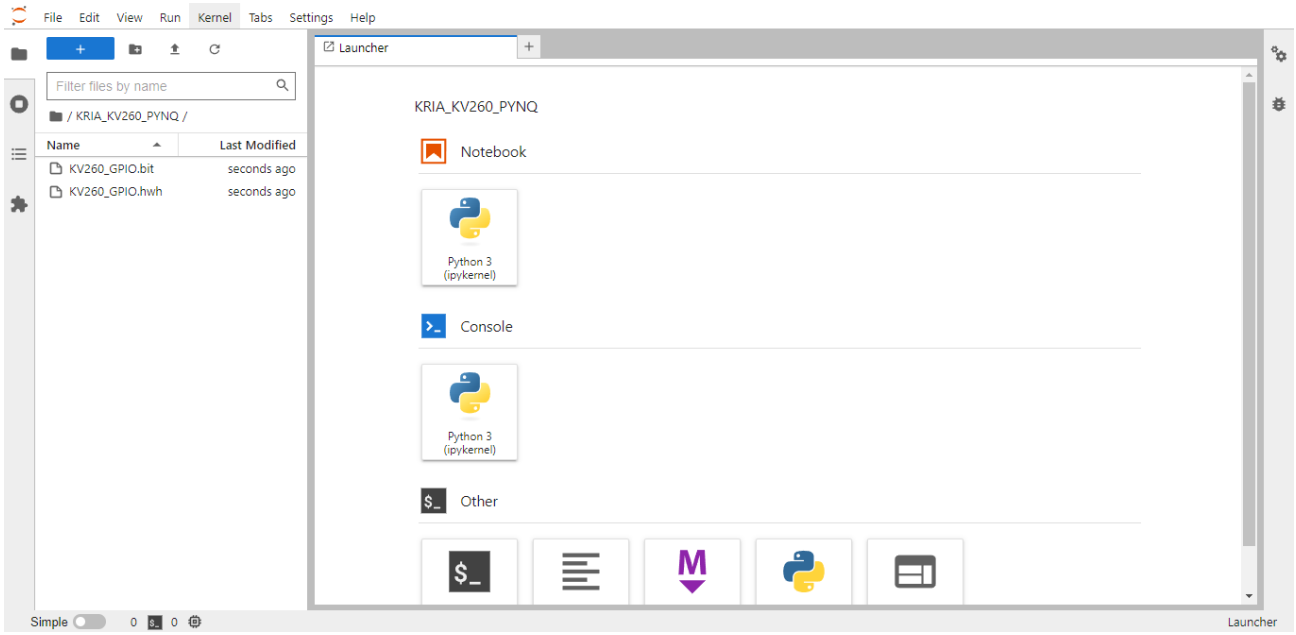
"Majority\Majority.runs\impl_1"

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



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
```