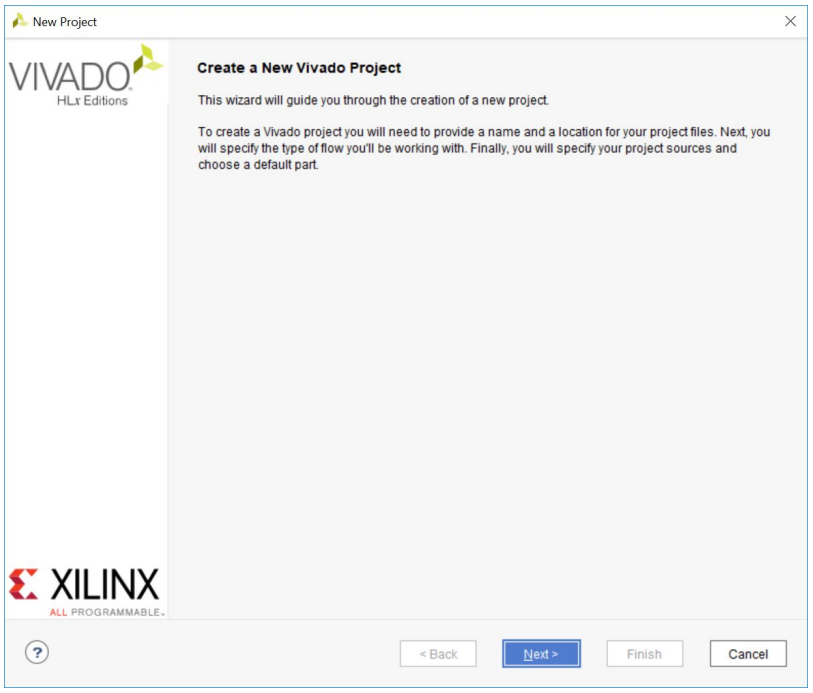
Instruction

This Instruction will help to use the HAW_RISC_V CPU architecture and run a assembler program.

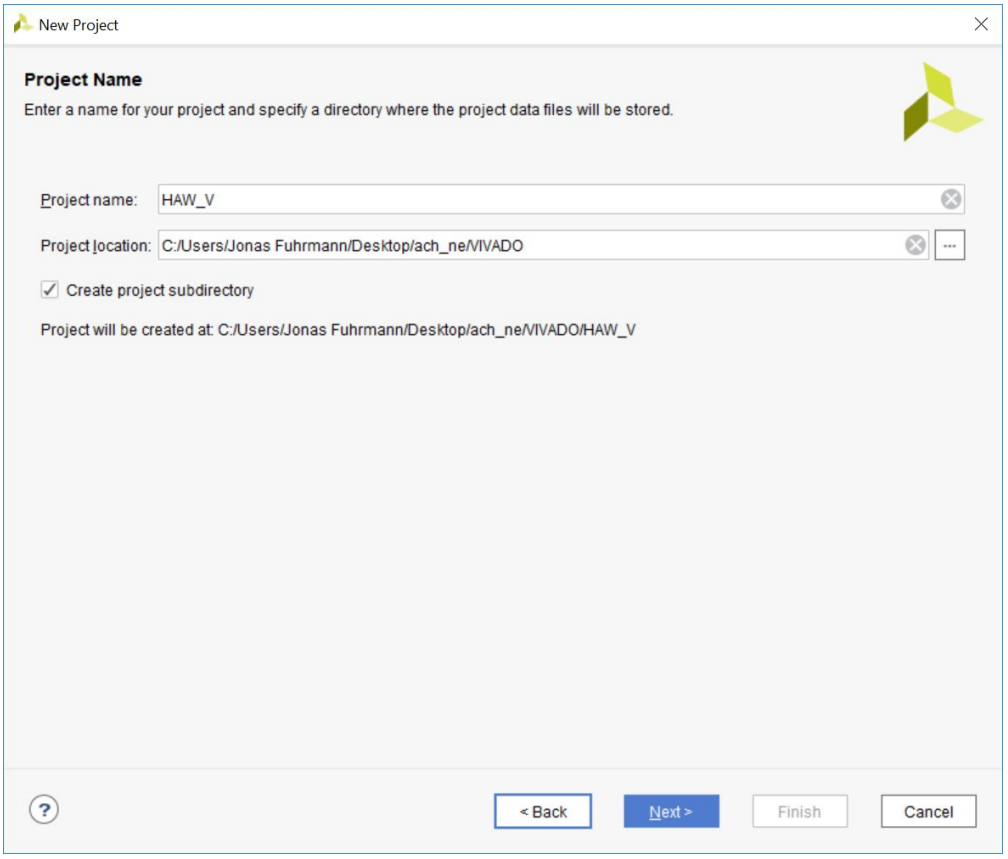
Create a Project

- Open Vivado

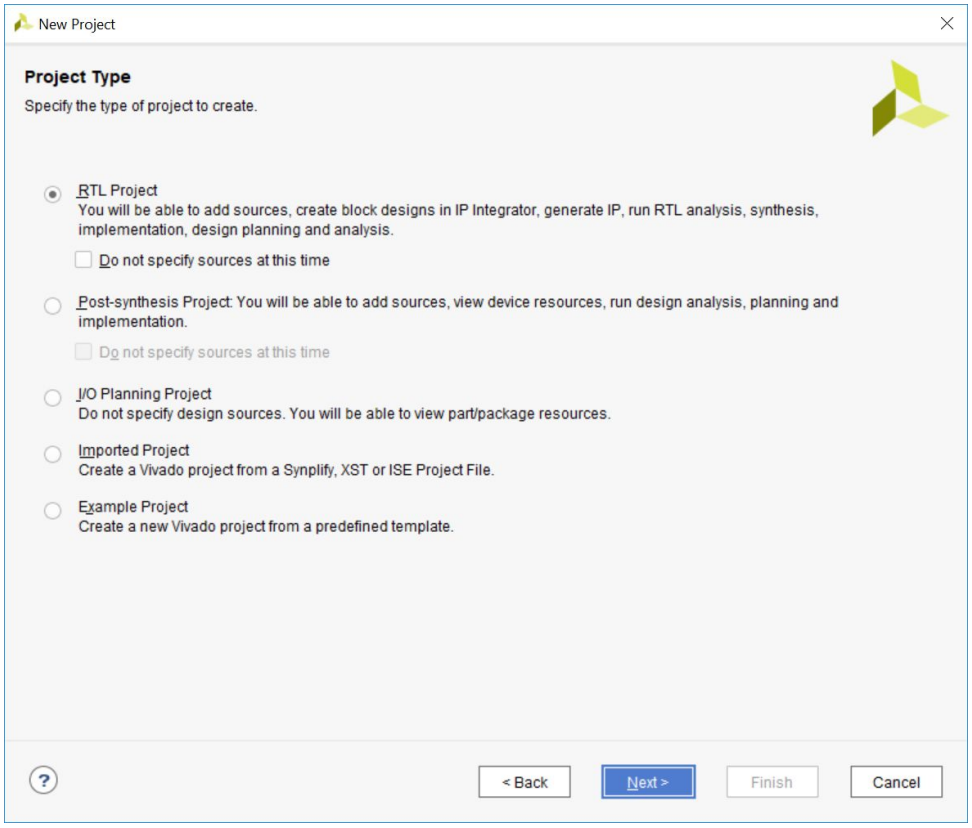
Create a new project and follow the wizard from Vivado as shown below:



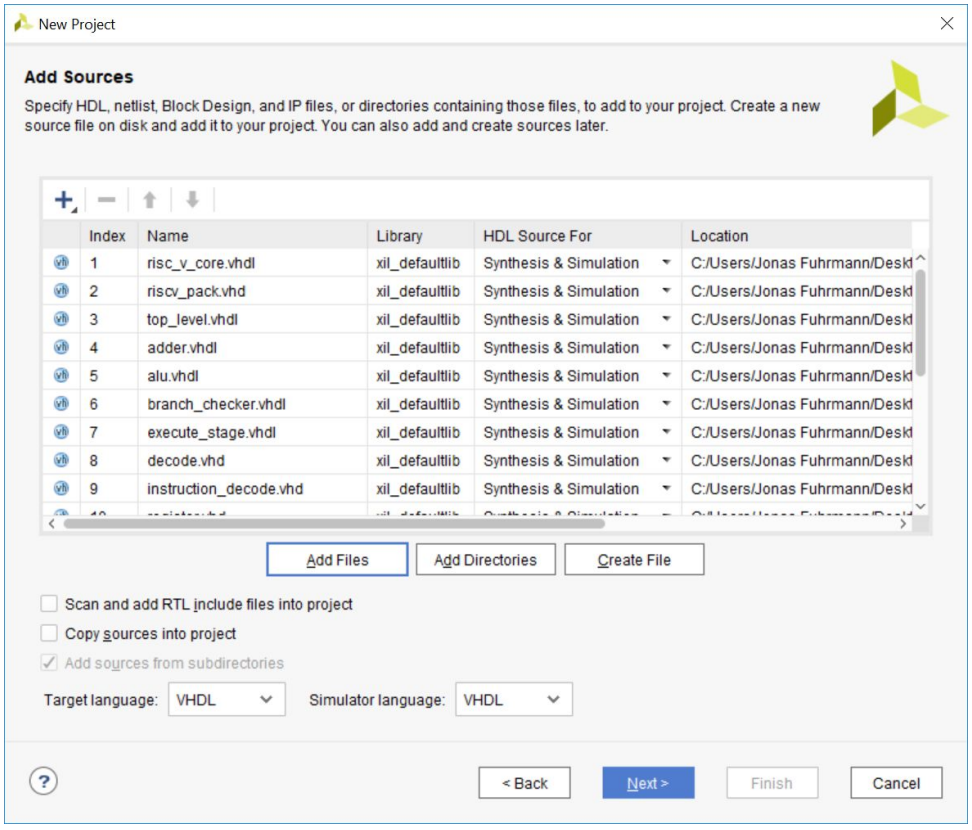
Click on **Next>** to continue.



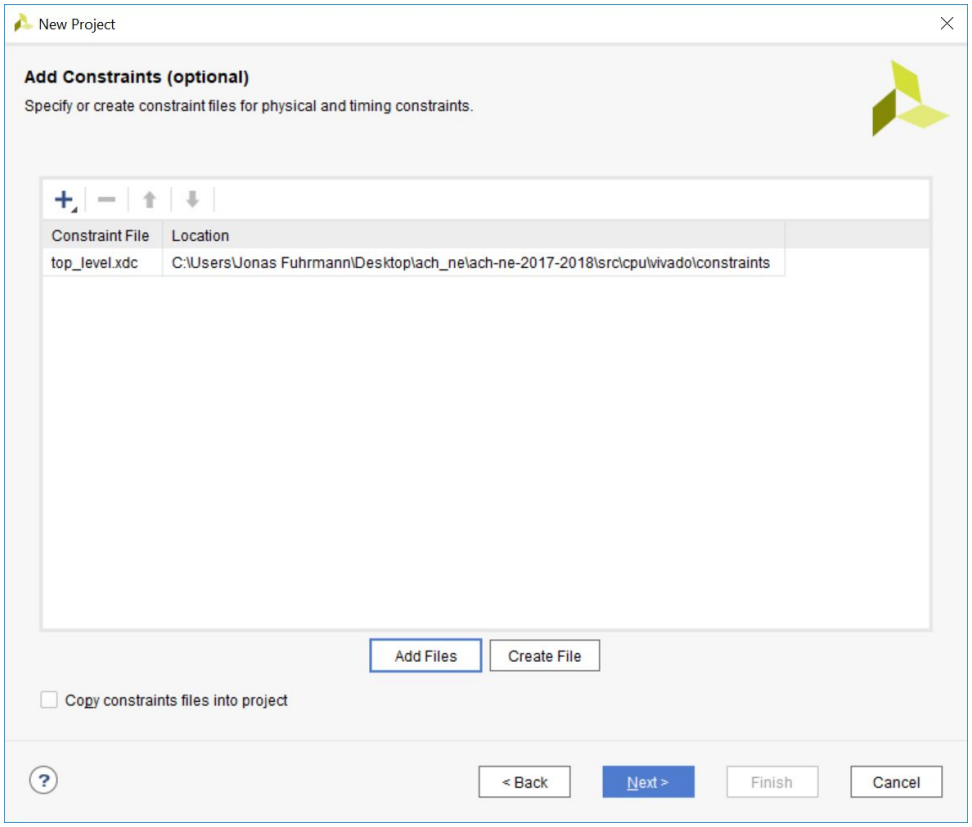
Enter a project name and a project location and select the checkbox to create project subdirectories. Click on **Next>** to continue.



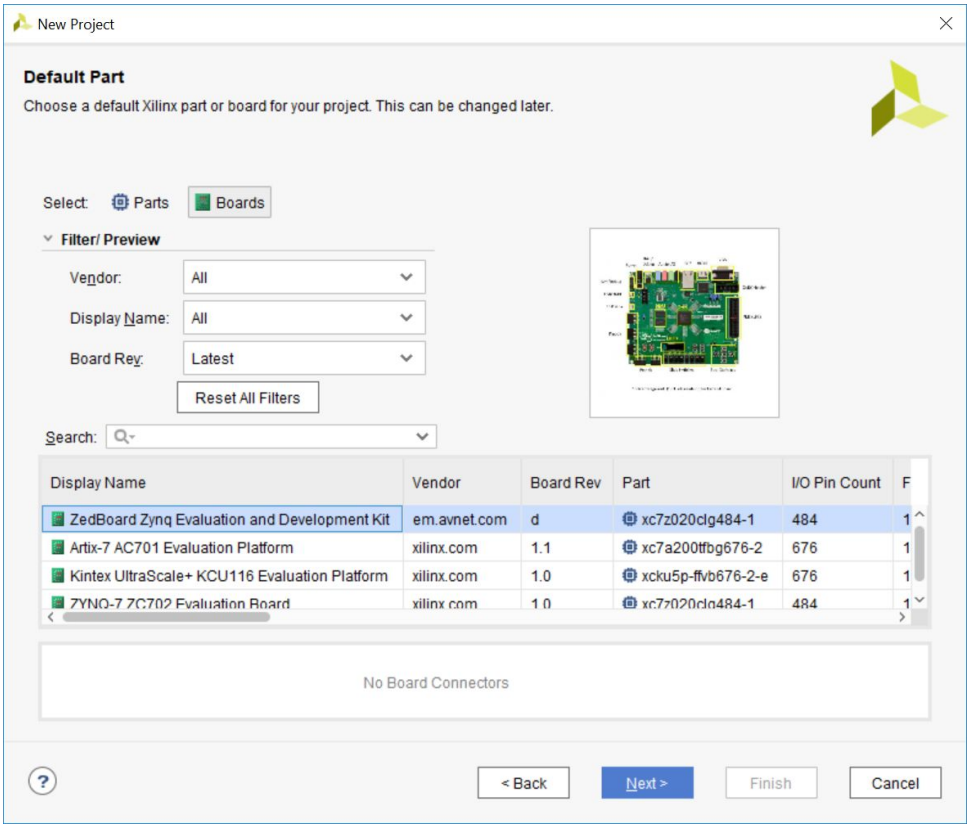
Choose *RTL Project* and click on **Next>** to continue.



Add all *.vhd/.vhdl* source from all subdirectories to the Vivado project. All *.vhdl files* with **_tb** are not needed. Click on **Add Files**.
Select VHDL as Target and Simulator language.
Click on **Next>** to continue.



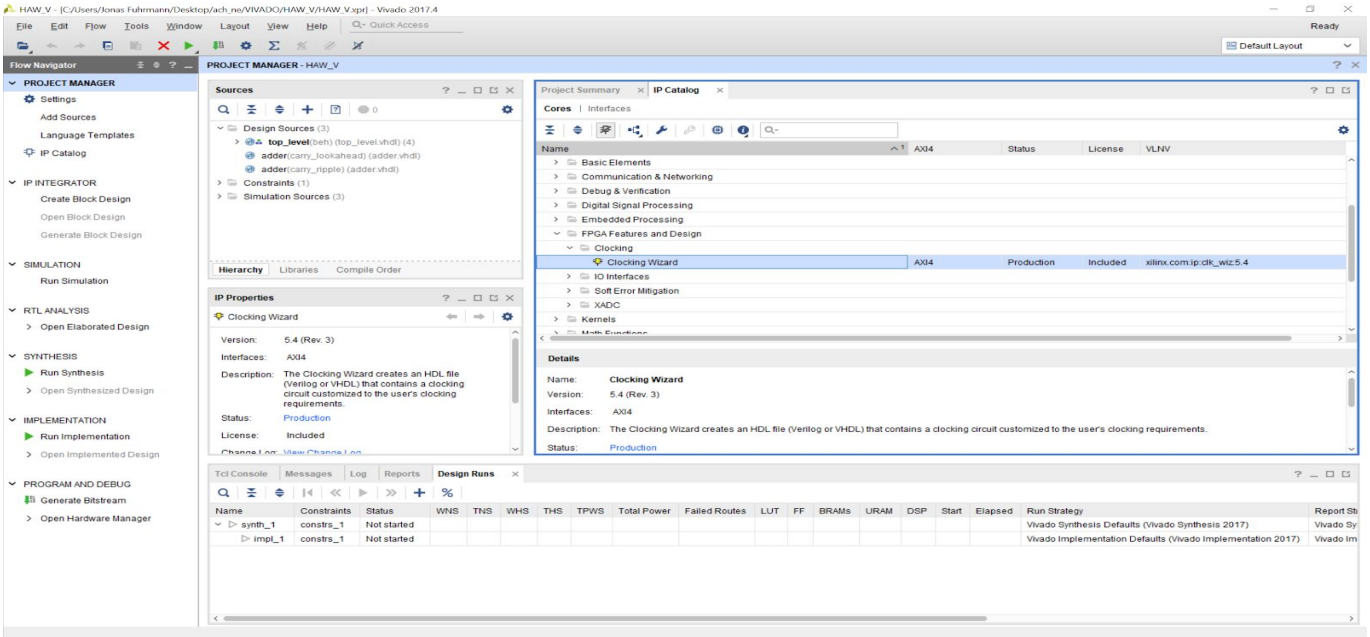
Add the top_level.xdc file as an constraints file. Click on **Add Files** and then on **Next>** to continue.

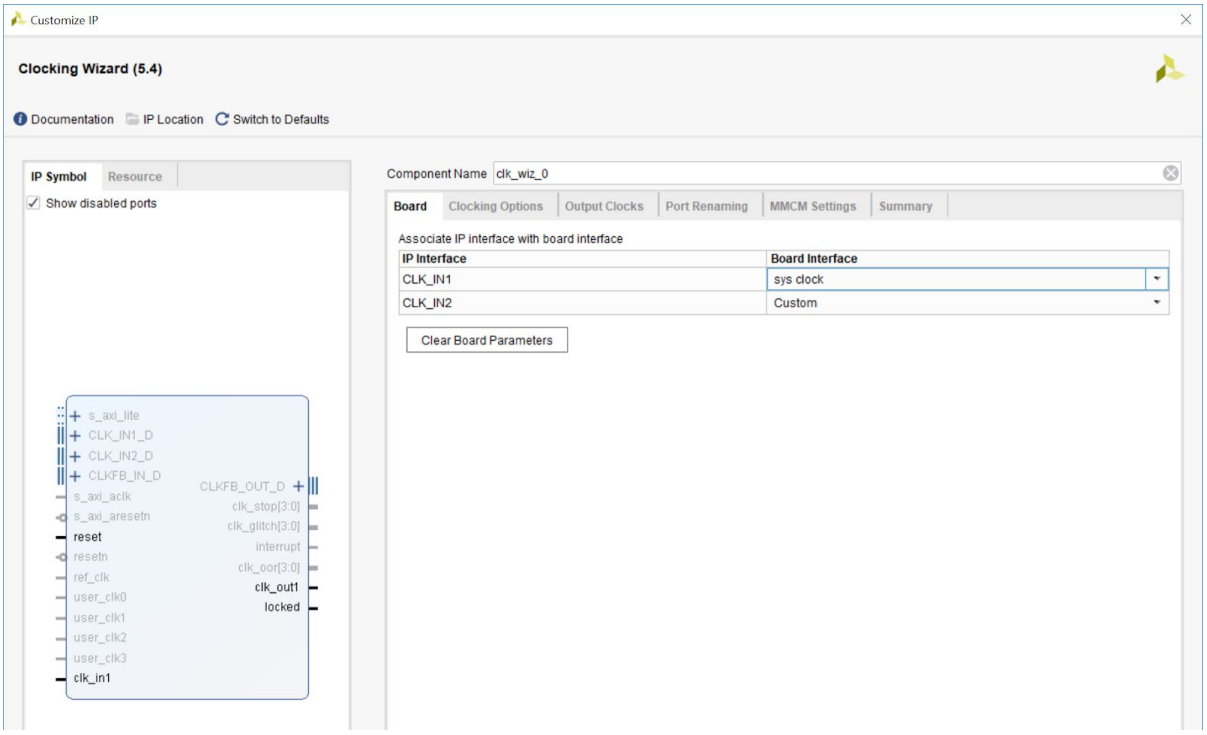


Select *ZedBoard Zynq Evaluation and Development Kit* from the given Boards and click on **Next>** to finish the wizard.

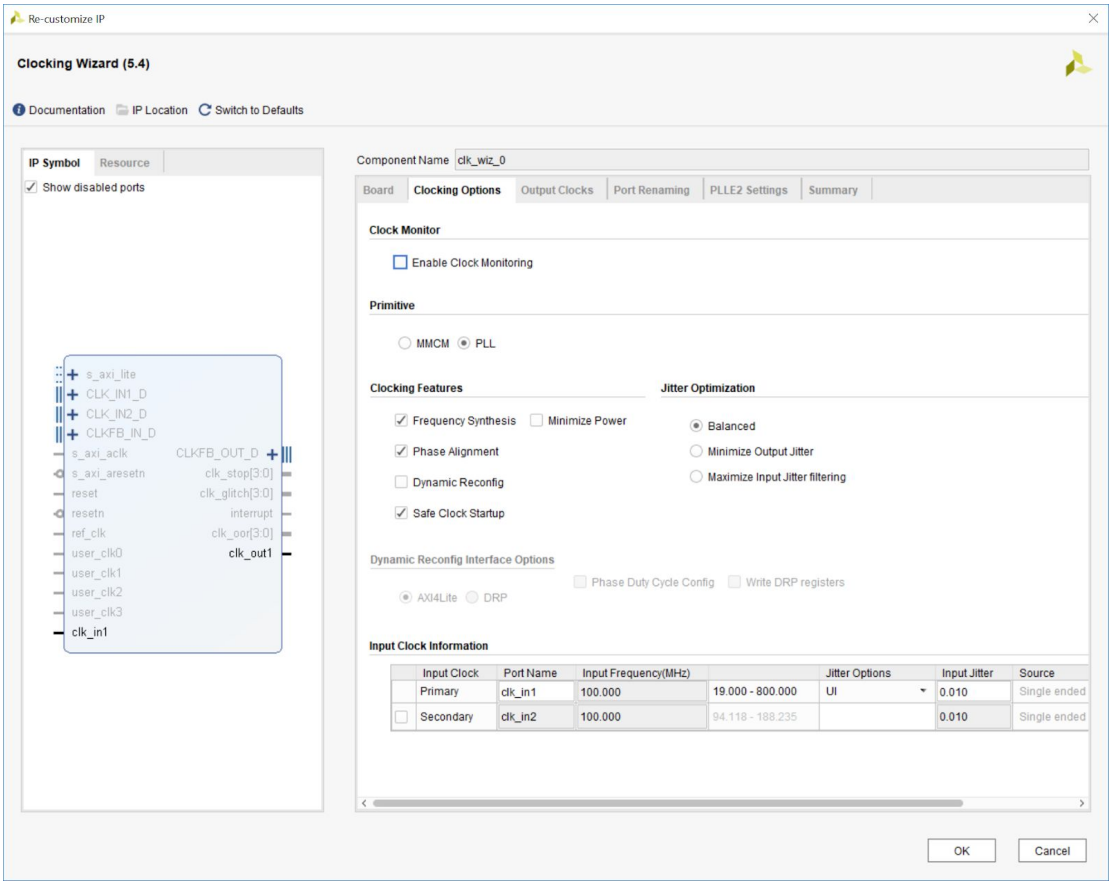
Generate a Clock

The CPU needs a clock with 50 MHz, but the Board runs with 100 MHz System Clock. Therefore we use the Ip Catalog from Vivado to GEnerate a PLL with 50 MHz.
Click on **Ip Catalog** on the Project Manager and open the **Clocking Wizard**.

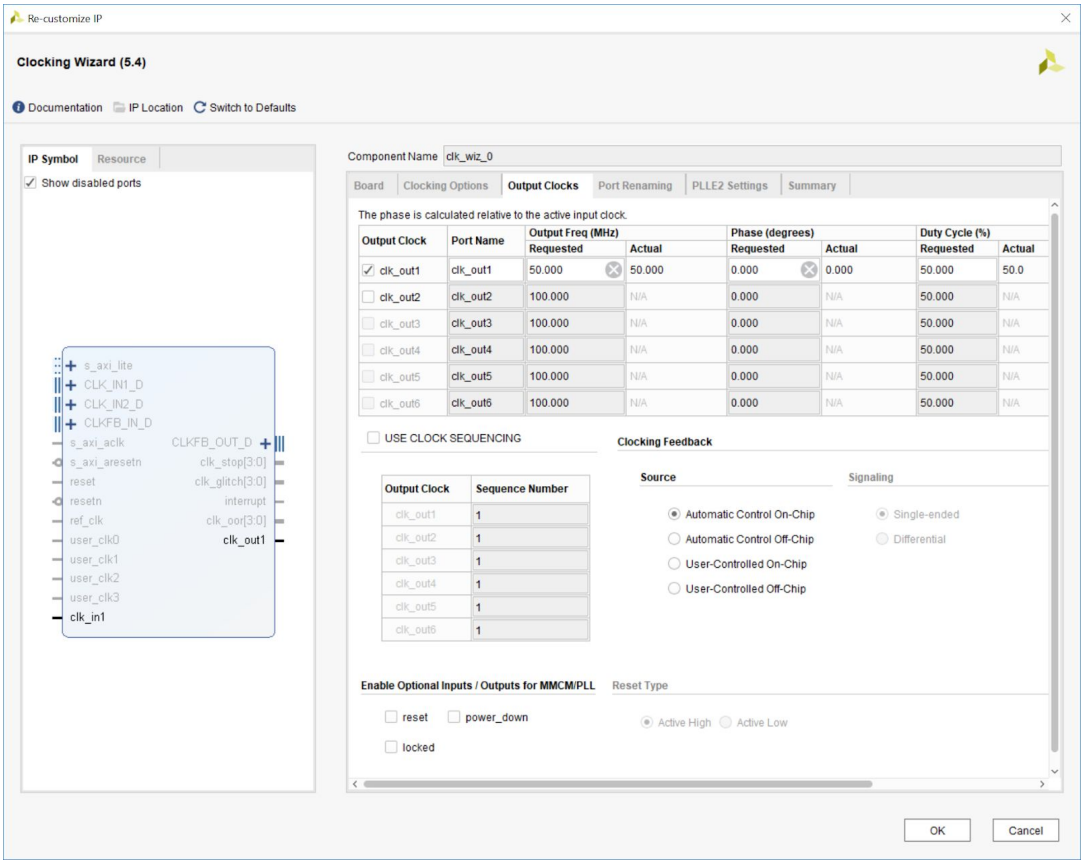




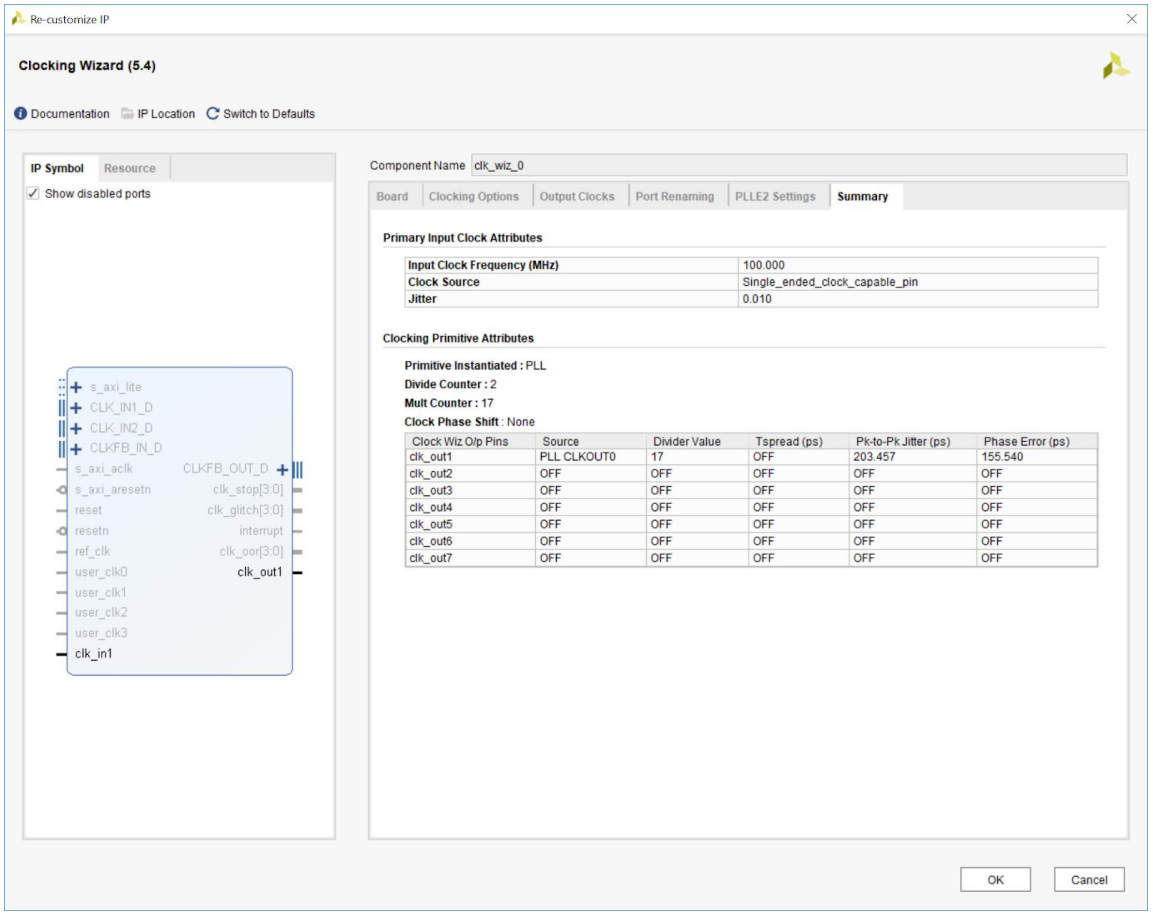
Select the **sys clock** as *CLK_IN1*. The Component Name should be clk_wiz_0!
Click on the index tab **Clocking Options** to continue.



Disable *Clock Monitoring*. Set everything as seen above!
Set the Port Name as *clk_in1* and set the Jitter Options to *UI* and Input Jitter to *0.010*.
Click on the index tab **Output Clocks.**



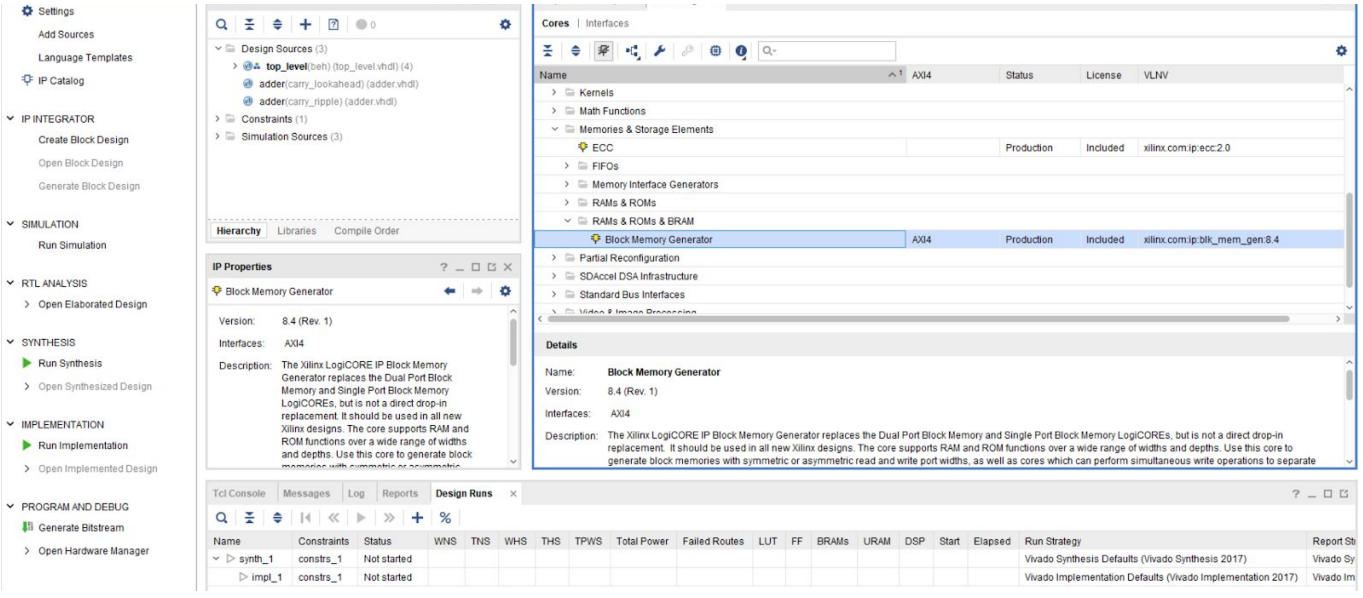
Select *clk_out1* as Output Clock and as Port Name. Set *50.000* as Output Freq (MHz) Requested and set Phase (degrees) Requested to *0.000* and Duty Cycle (%) Requested toi *50.000*.
Click on the index tab **Summery** to continue.

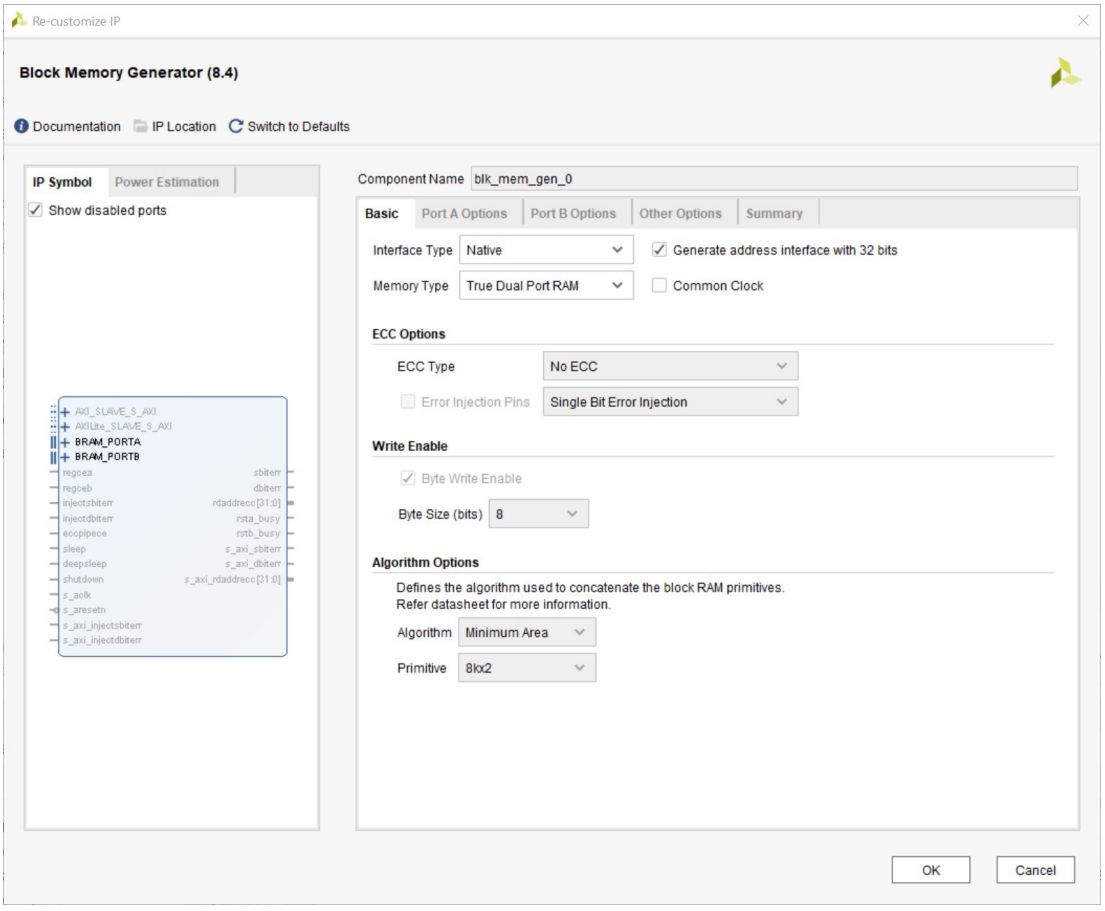


If Everything is set correctly, click on **Ok** to start the generation of the 50 MHz PLL. This will take some time!

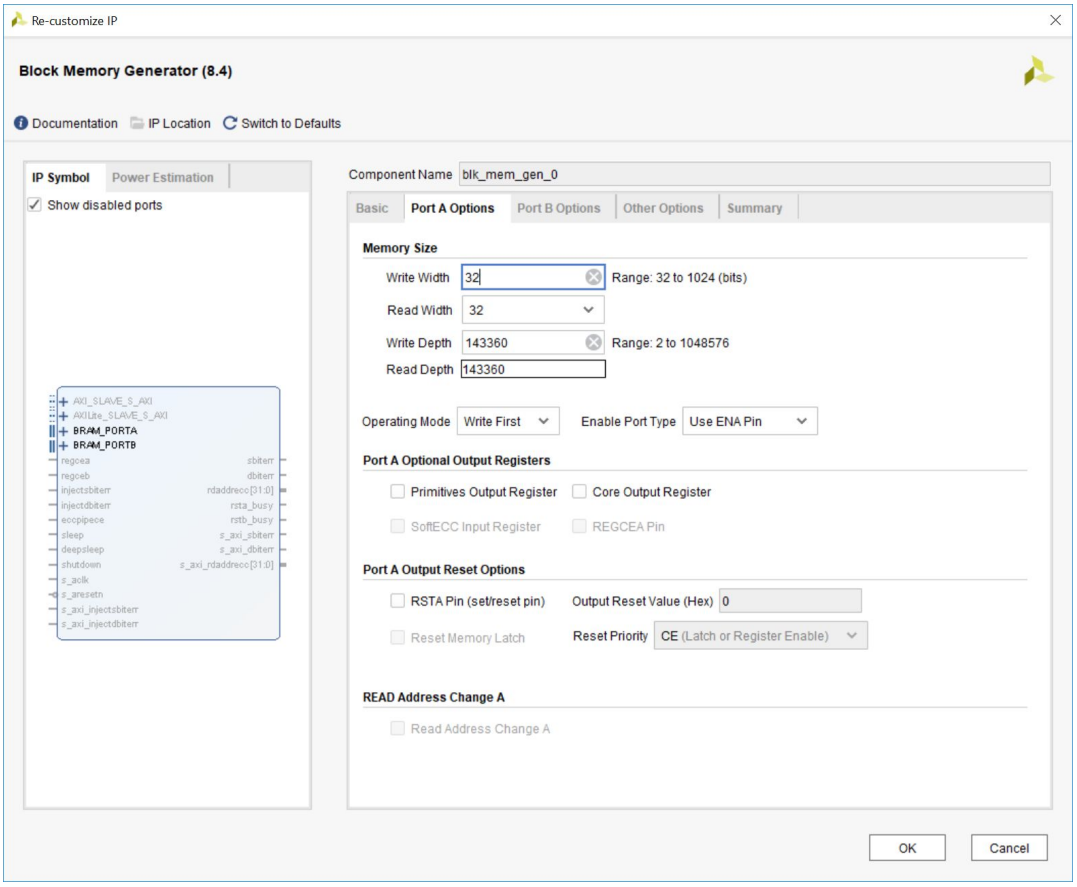
Generate a Block RAM with the Assembler program

The CPU needs BRAM with the Assembler program, to run it. This will be generated by Vivado as well. Therefore we use the Ip catalog again. Open the *Block Memory Generator* from the **IP Catalog**.

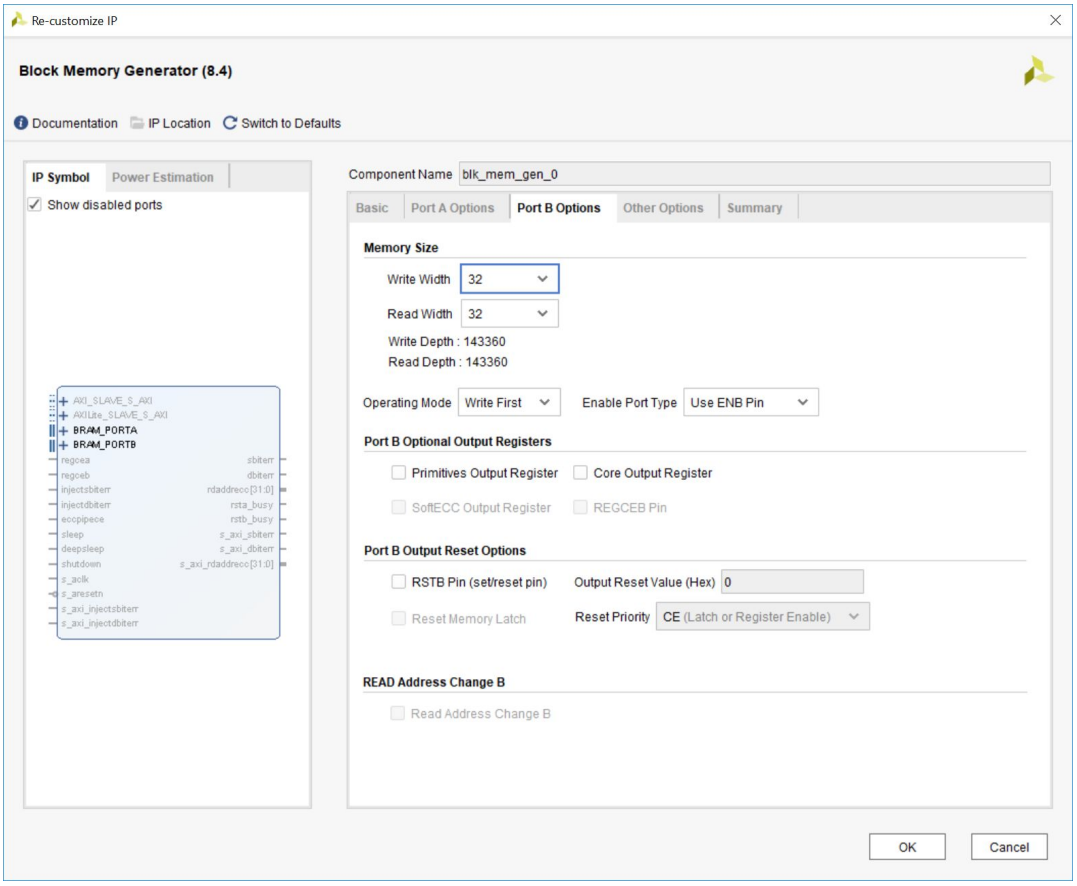




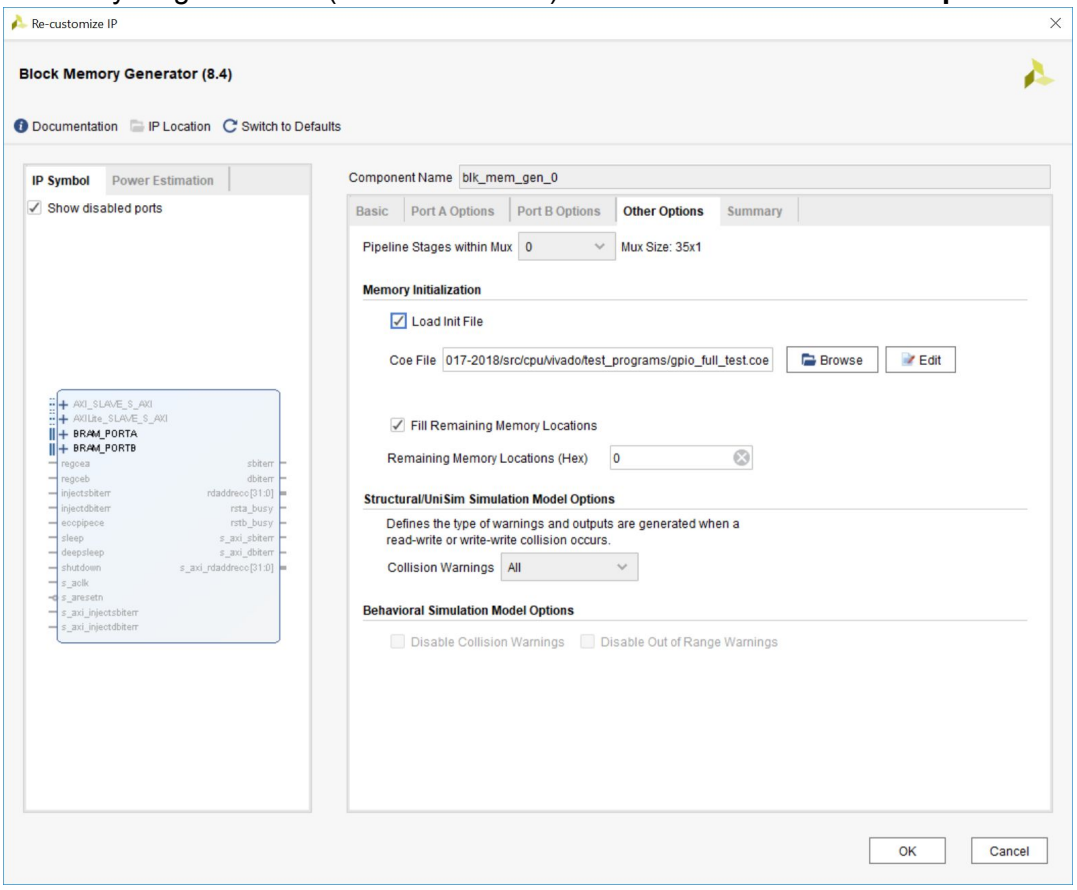
Set *Native* as Interface Type and *True Dual Port Ram* as Memory Type. Select *Generate address interface with 32 bits!* Ecc is not needed. Set everything else as shown above!
Click on the index tab **Port A Options** to continue.



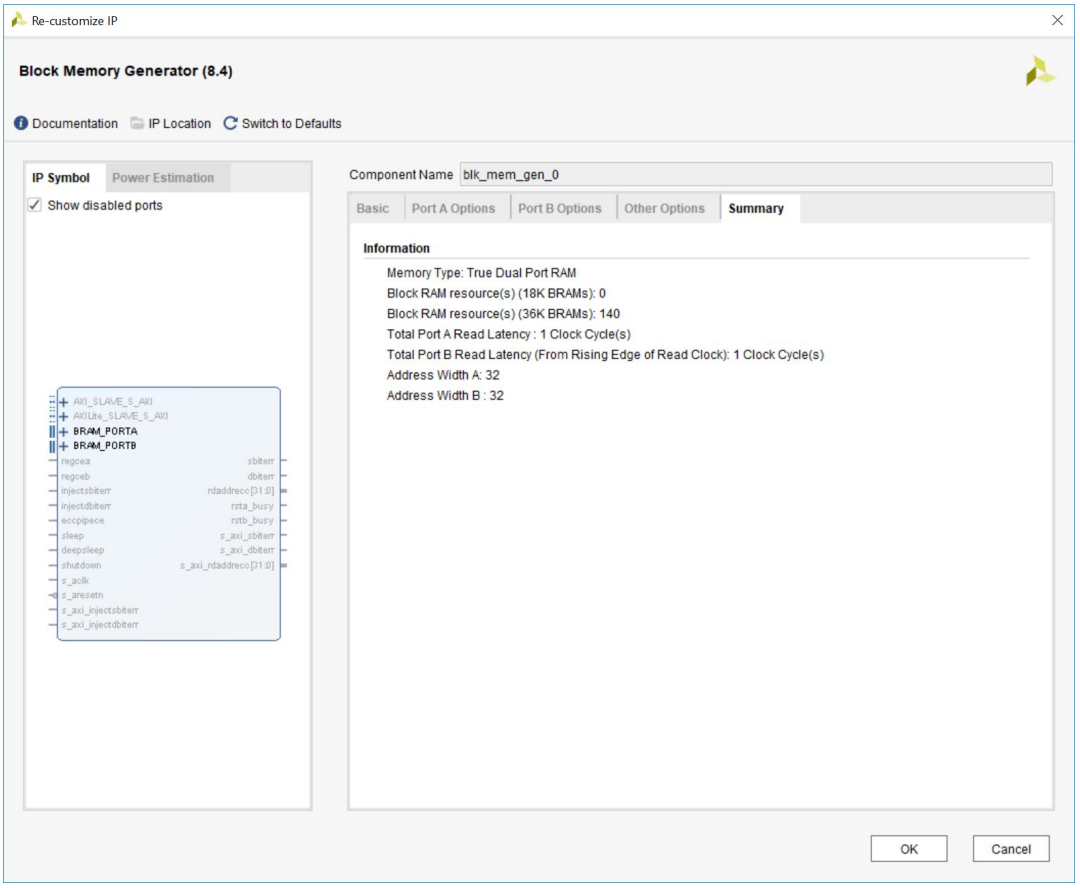
Set write and read Width to *32* and the depth to *143360!* Set Operating Mode to *Write First* and Enable Port Type to *Use ENA Pin.* Disable everything else and continue on tab **Port B Options.**



Set everything as Port A (as shown above) and continue on tab **Other Options.**



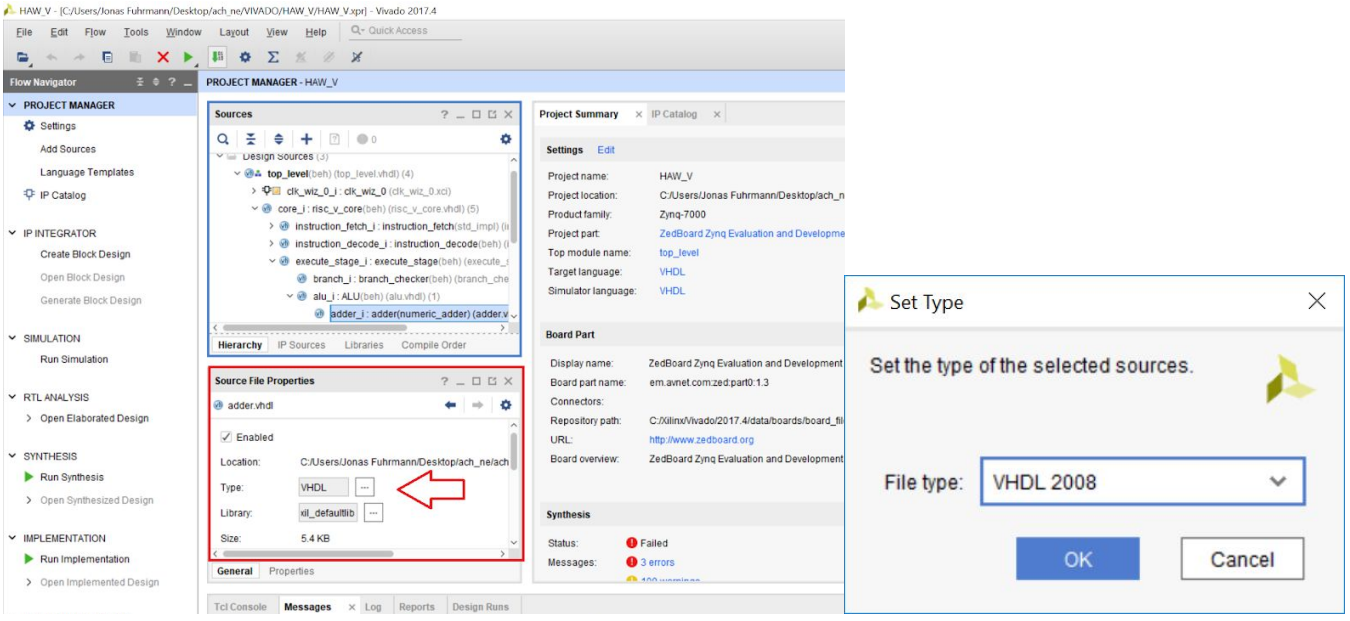
Enable the Option *Load init file* and select from the project folder the file *gpio_full_test.coe* as the Coe file.The Coe file contains the assembler program in the RISC V type. Fill the Remaining Memory with *0.* Click on the the tab **Summary** to check the settings.



Start the generation of the BRAM by clicking on **Ok.** This will take some Time!

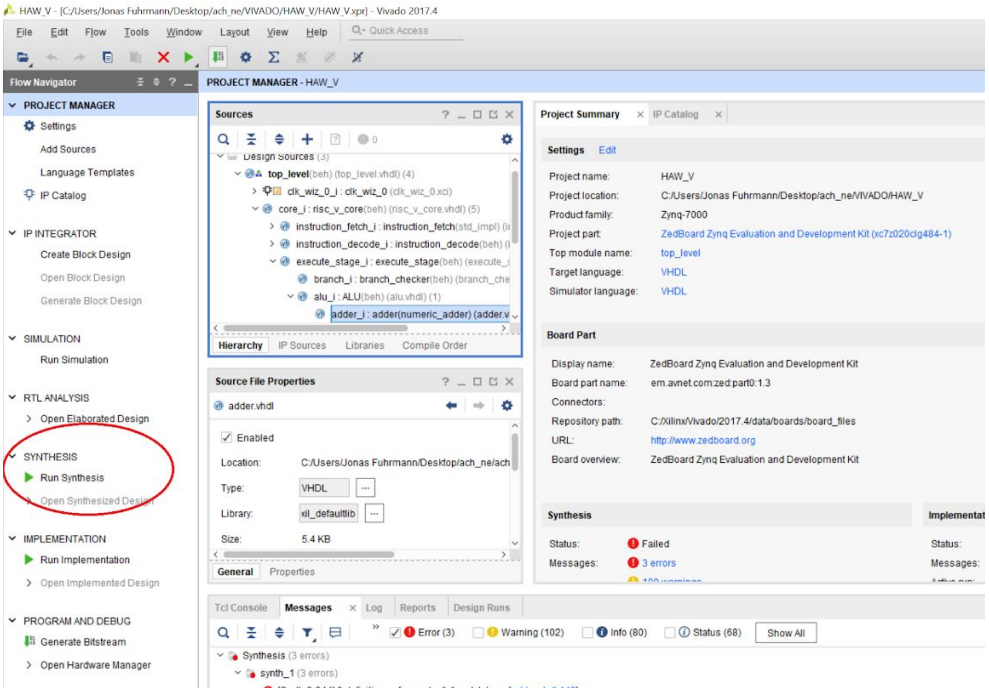
Run the Synthesis

It is important, that every component is generated before the Synthesis is started (Clock and BRAM)!
To run the Synthesis all vhdl files should be **VHDL 2008**.
This can be set here:



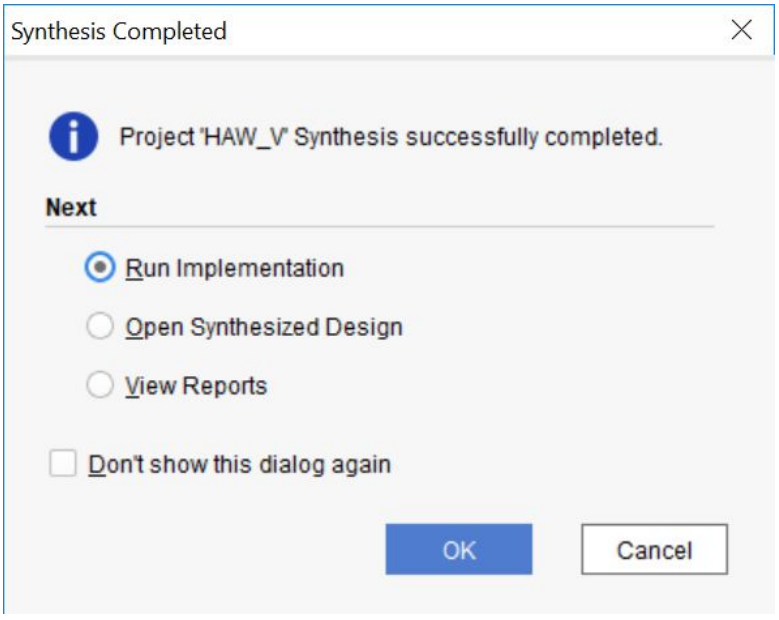
Click on the … and select *VHDL 2008.* Click on **OK** to continue. This must be done for **all vhdl Files**!

If this is done click on *Run Synthesis.*

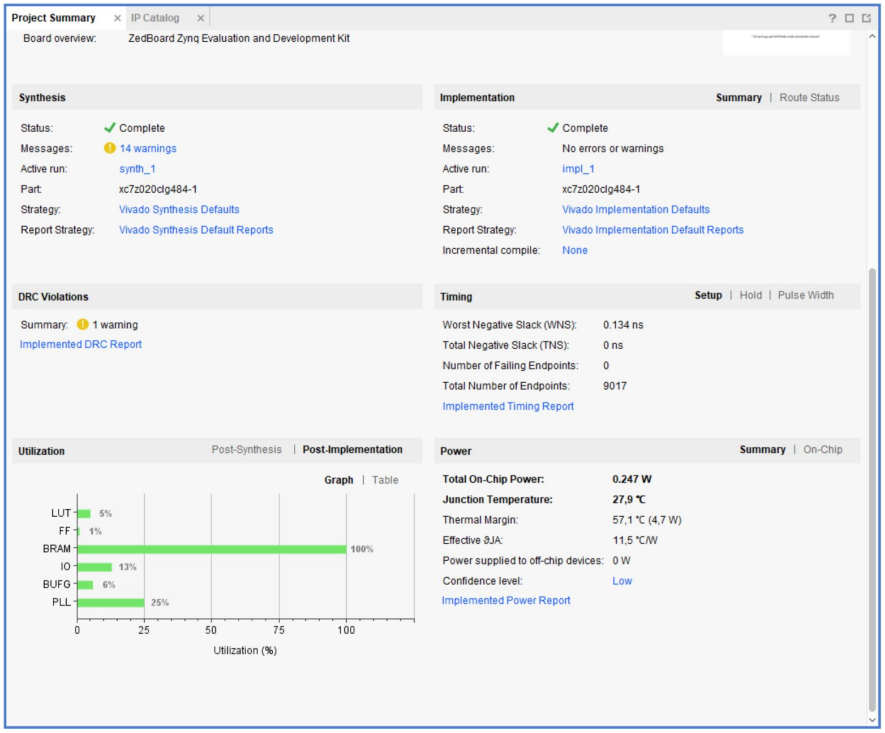


Run the Implementation

After the Synthesis is finished successfully , Vivado opens a window as shown below. Click on *Start Implementation.*

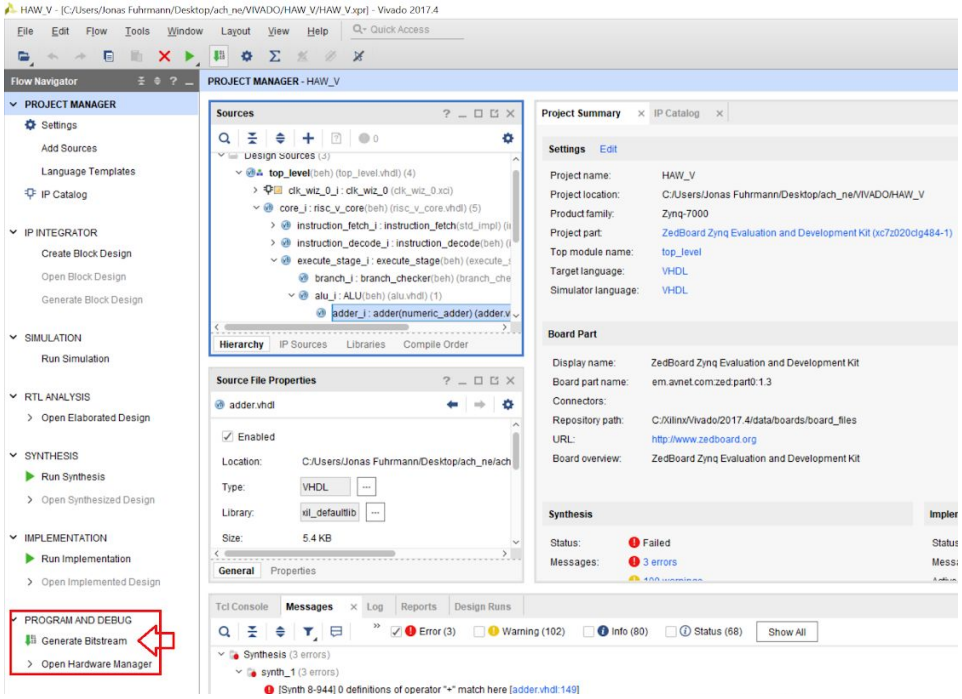


After the Implementation there is an Report (Project Summary) which shows different kind of Information about used hardware, power, critical path etc..



Generate a Bitstream

After the implementation we need a bitstream. Click on **Generate Bitstream**.



Program a Bitstream on the ZedBoard

After the generation of the bitstream is successfully completed, Vivado opens a window. Select *Open Hardware Manager* and click on **OK.**

Before plugging in the Zedboard, there are Jumper settings that need to be checked.

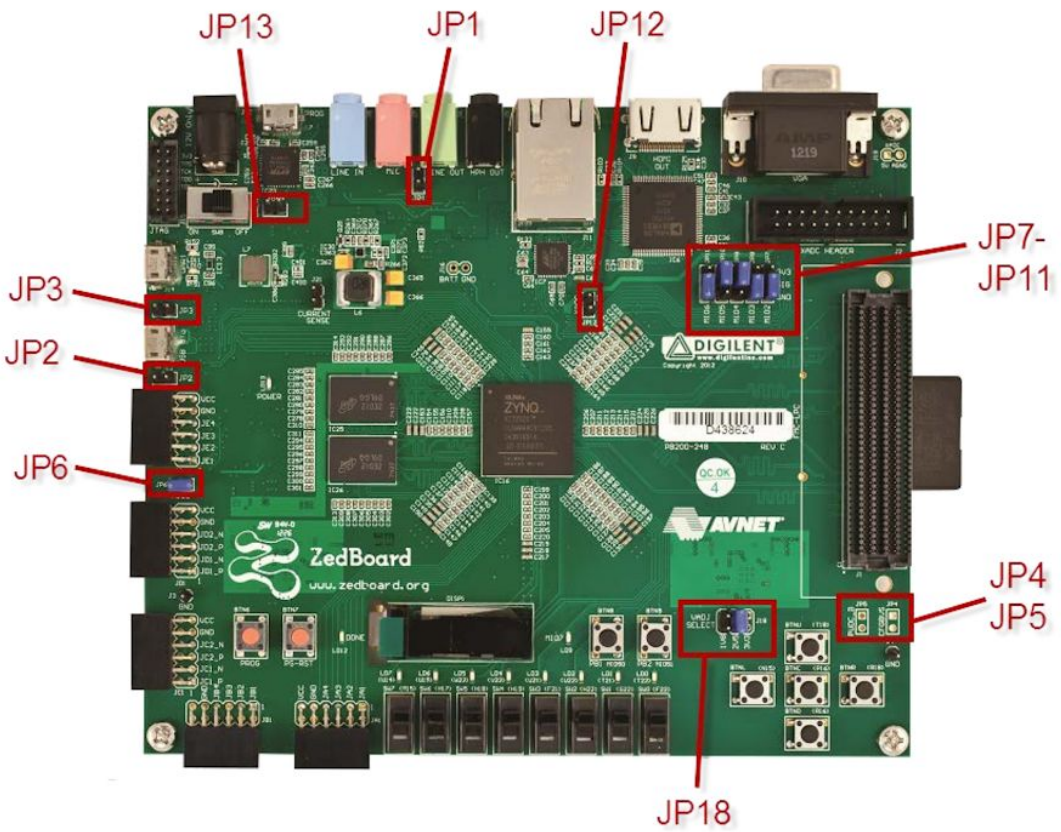


Figure 20 - ZedBoard Jumper Map

Set Jumper **JP7- JP11** all to 0 to set the correct Mode!
After this connect the Zedboard with power and with the USB cable to the Pc with Vivado to upload the program (Use the USB-PORT right next to the Power connection). Turn on the ZedBoard.
The Hardware Manager from Vivado should show the ZedBoard. If not, update the Hardware Manager. Click on it and program the Bitstream.
The program *gpio_full_test.coe* uses the switches and the buttons to trigger the LEDs.

Create own coe files for HAW RISC V CPU

To create own coe files out of assembler programs, the RISC V toolchain is needed. After that there is a script which can be found in *vivado/helper_scrips.* It is important to know, that the compiler doesn't link addresses from the *.data section*. The *.data section* can be put behind the *.text section*. Then the offset has to be calculated and insert into the .coe file.