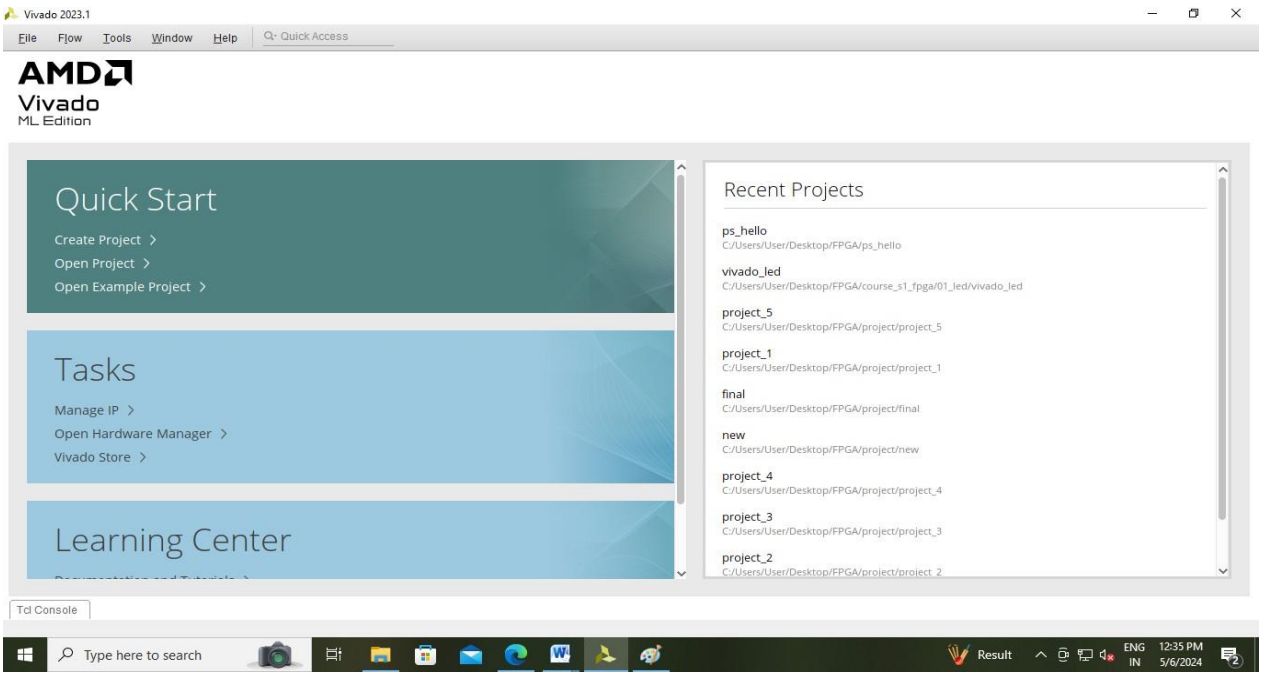
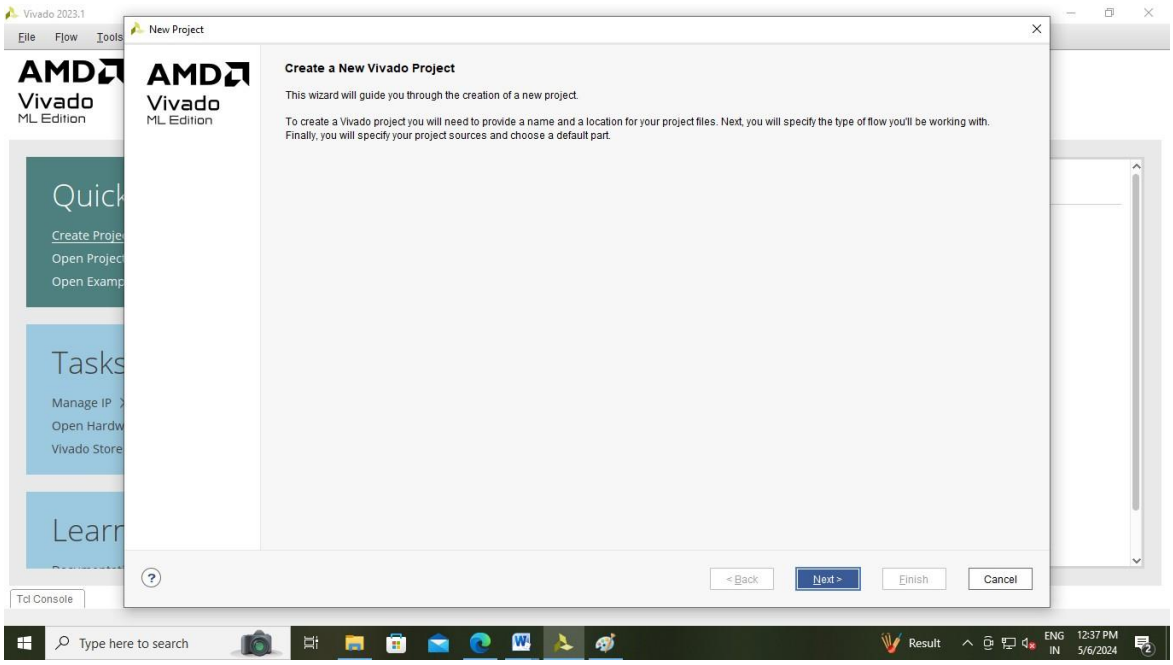
Create Your First Project

- **Open Vivado**: Launch the Vivado Design Suite.

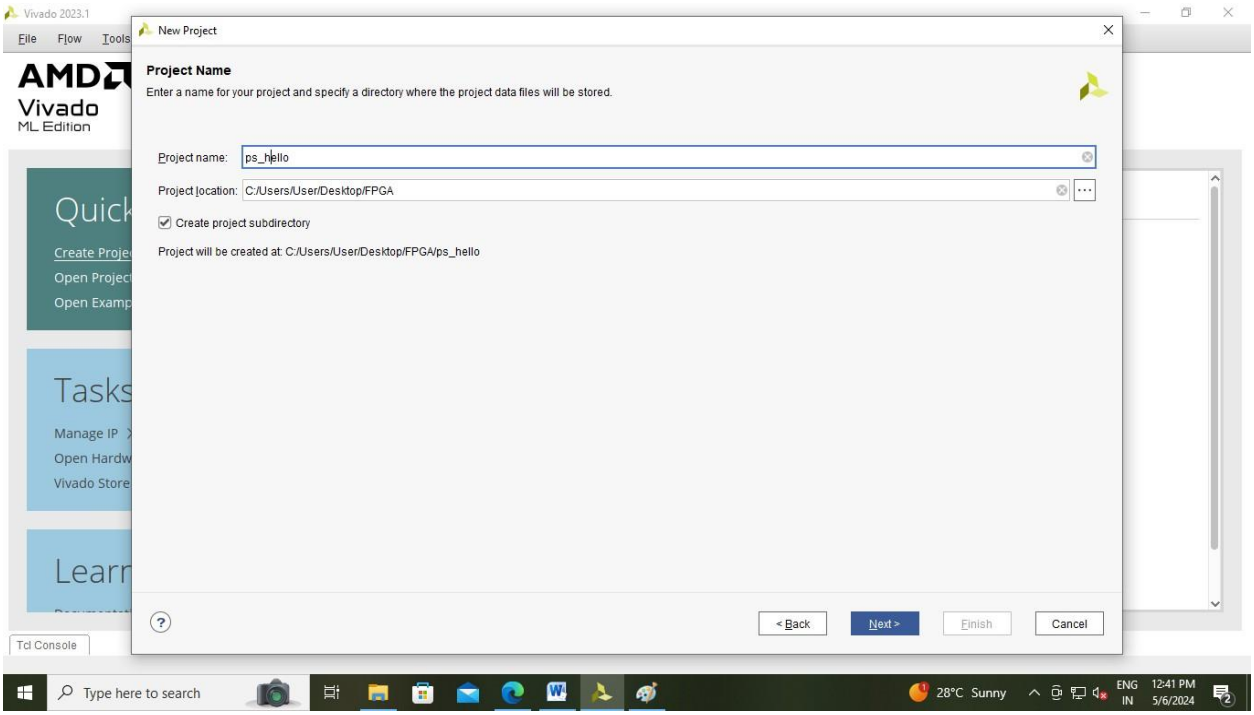
Step 1: click "Create Project", to create a new project.



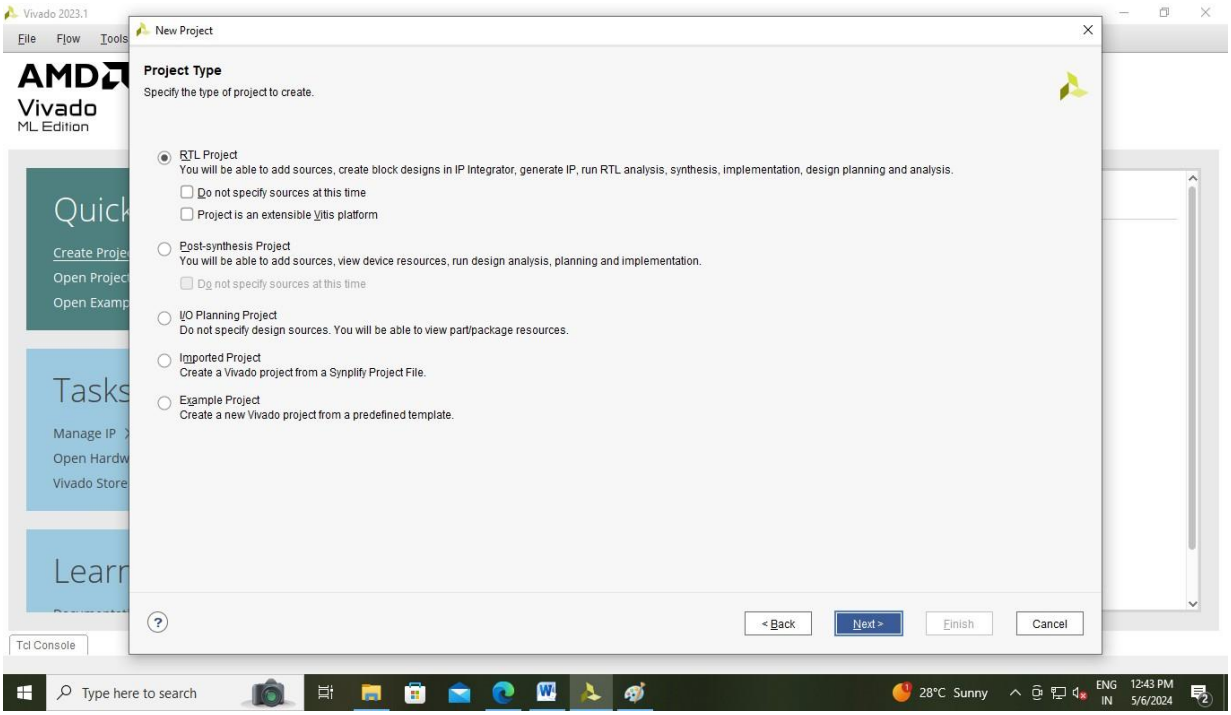
Step 2: A wizard for creating a new project will pop up, click "Next"



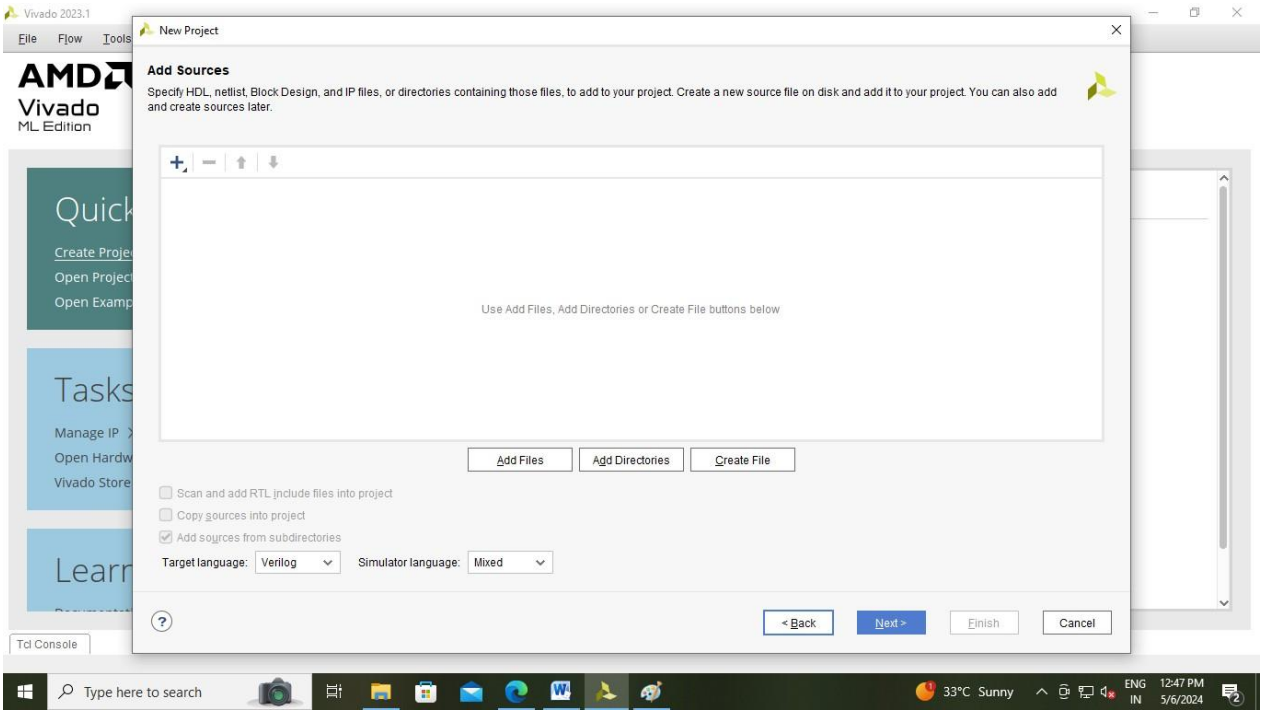
Step 3: In the pop-up dialog box, enter the project name and the directory where the project to be stored. Let's take example of "ps_hello" as a project name. Need to pay attention to the project path "Project location". And click on Next.



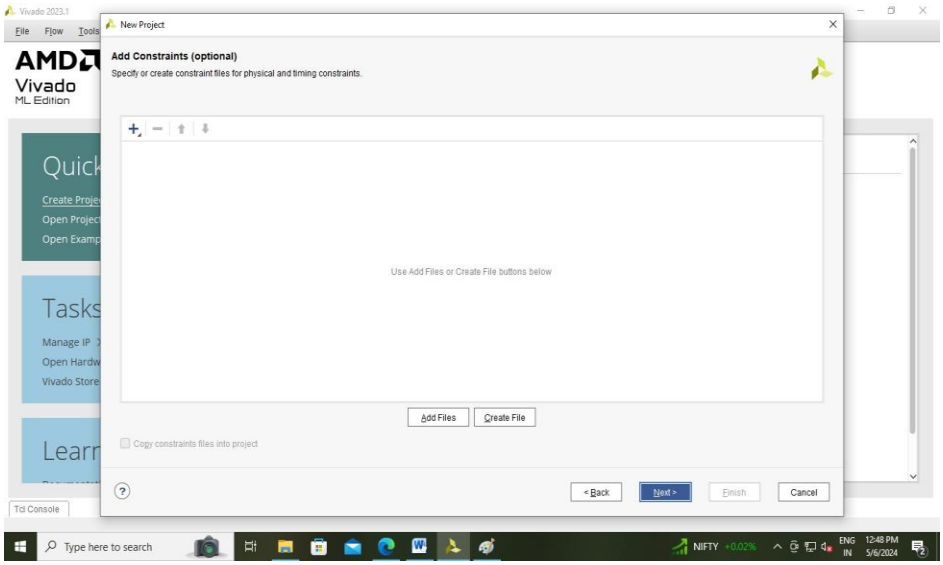
Step 4: Here we need to select the Project Type. Select "RTL Project" as project type. Then click on Next.



Step 5: In this window we need to select Target Language and Simulator language, choose "Verilog" as Target Language and “Mixed” as Simulator language then click on “Next”



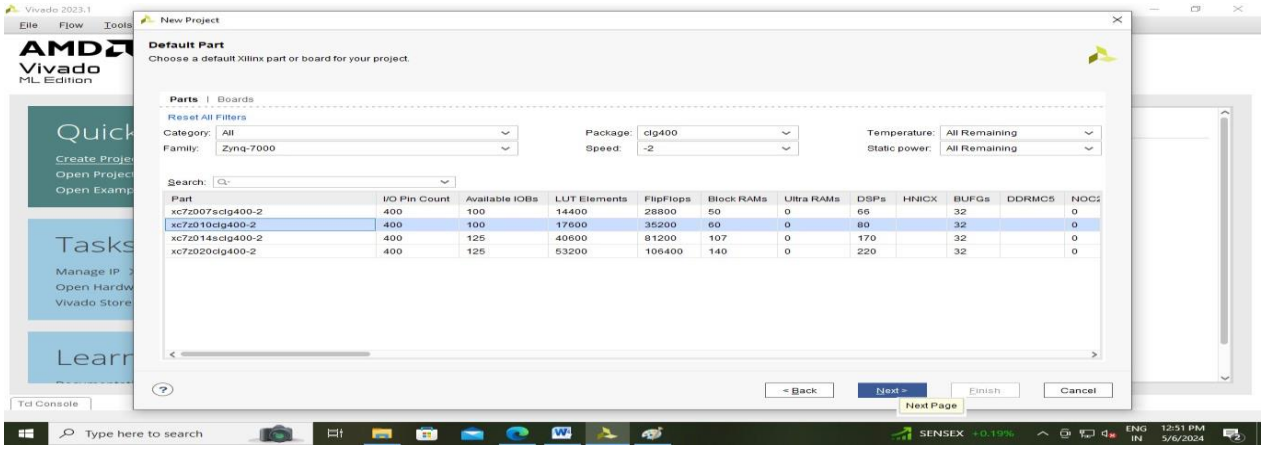
Step 6: Click "Next”, do not add any files



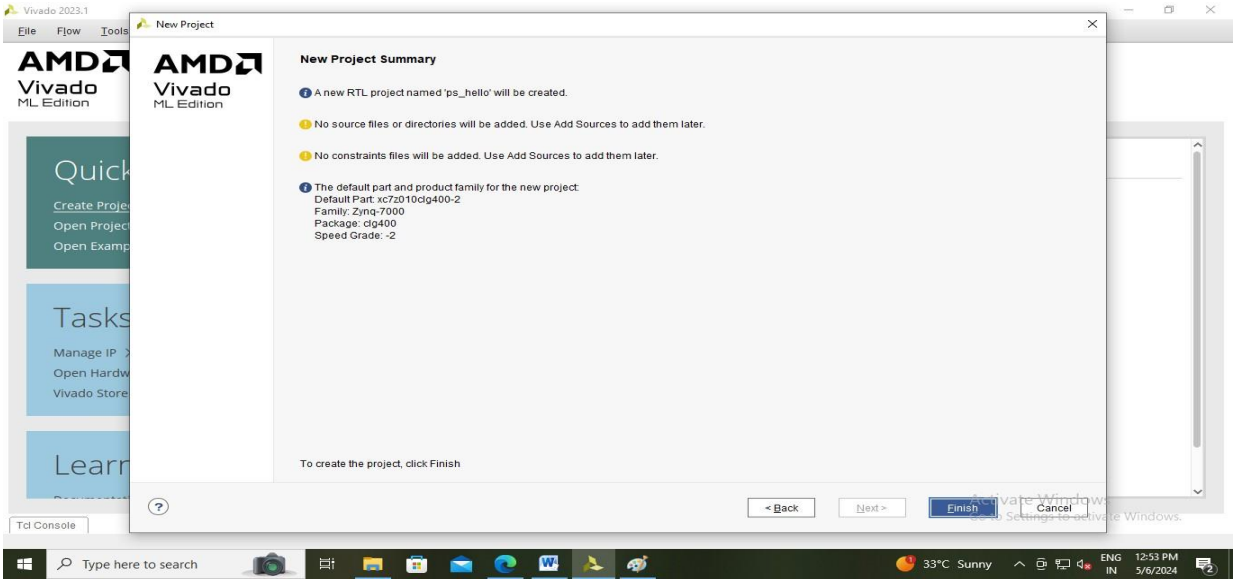
Step 7: In this wizard we need select device Family, Package and speed.

- Category: All
- Family: Zynq-7000
- Package: clg400
- Speed: -2
- Temperature: All Remaining
- Static Power: All Remaining

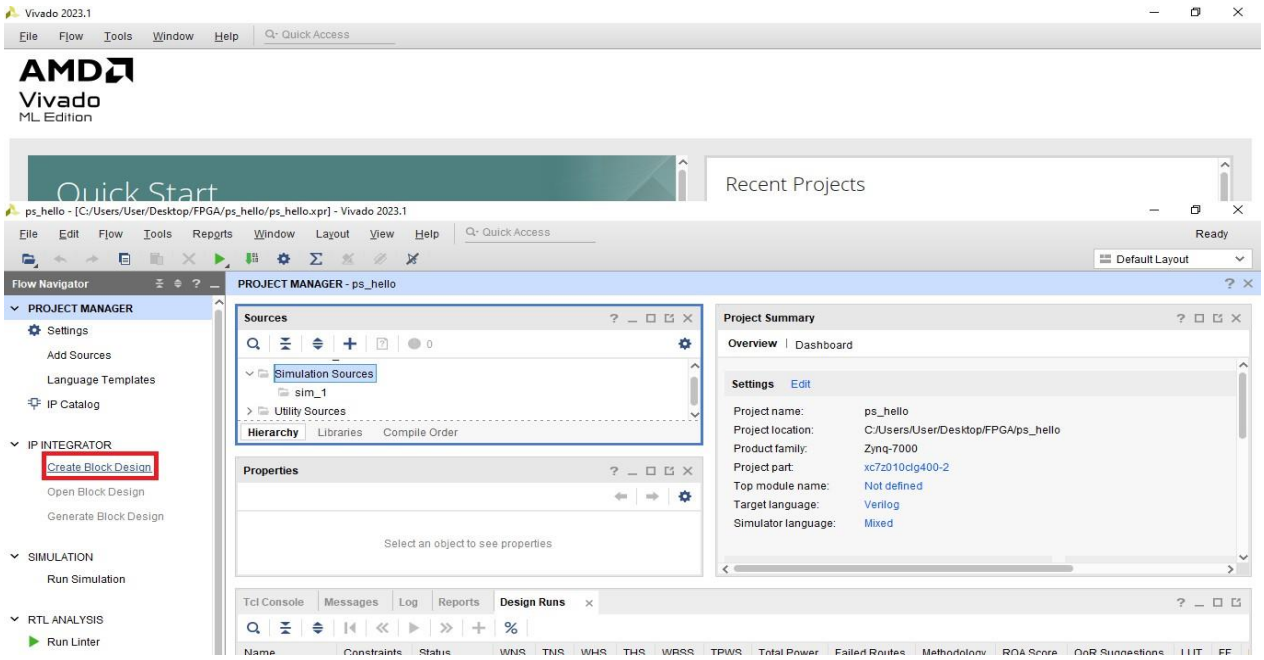
Then click next.



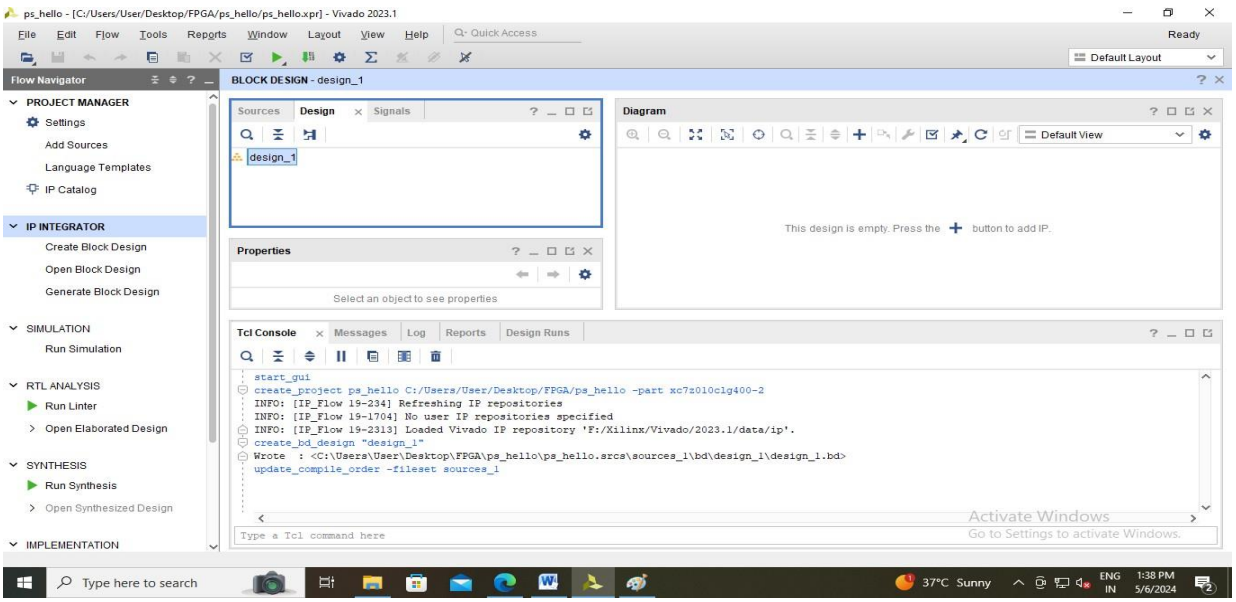
Step 8: Click "Finish" creating the project



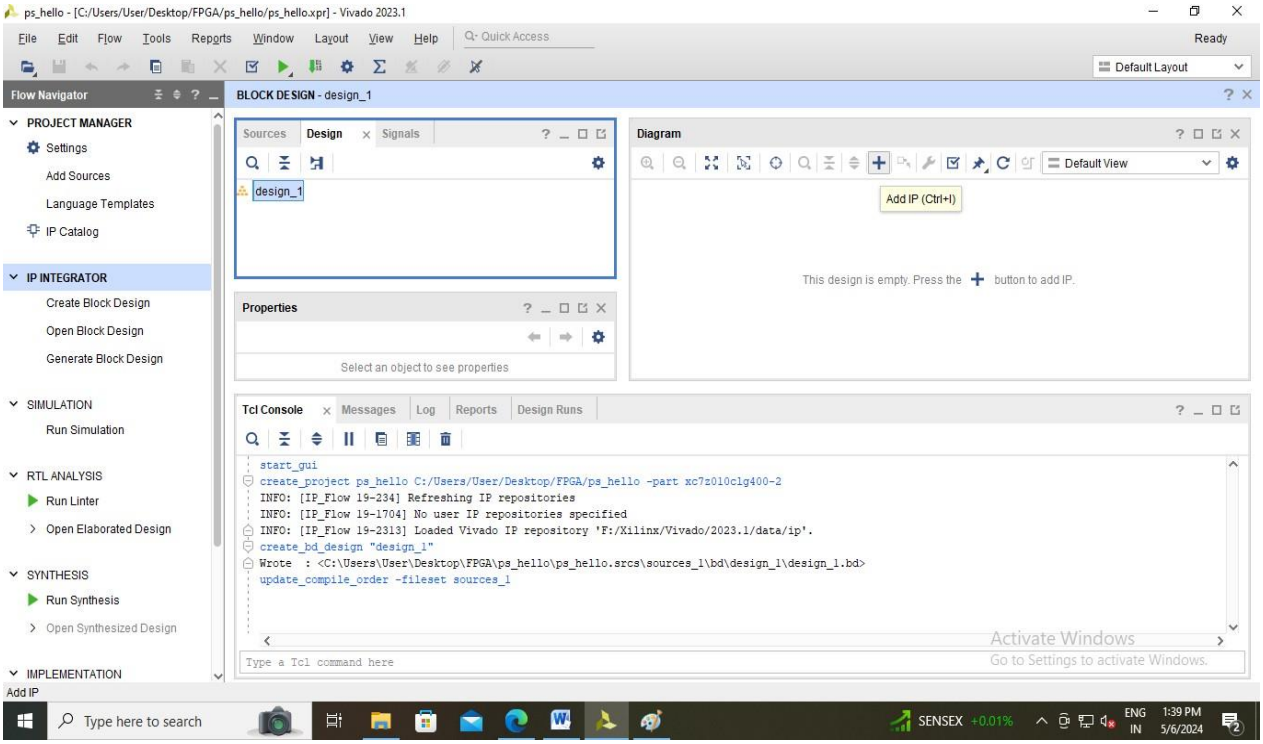
Step 9: Once the Project is created we need to create Block Design. In order to create Block Design please expand Project manager, here expand IP Generator then click on Create Block Design as shown in the image below.



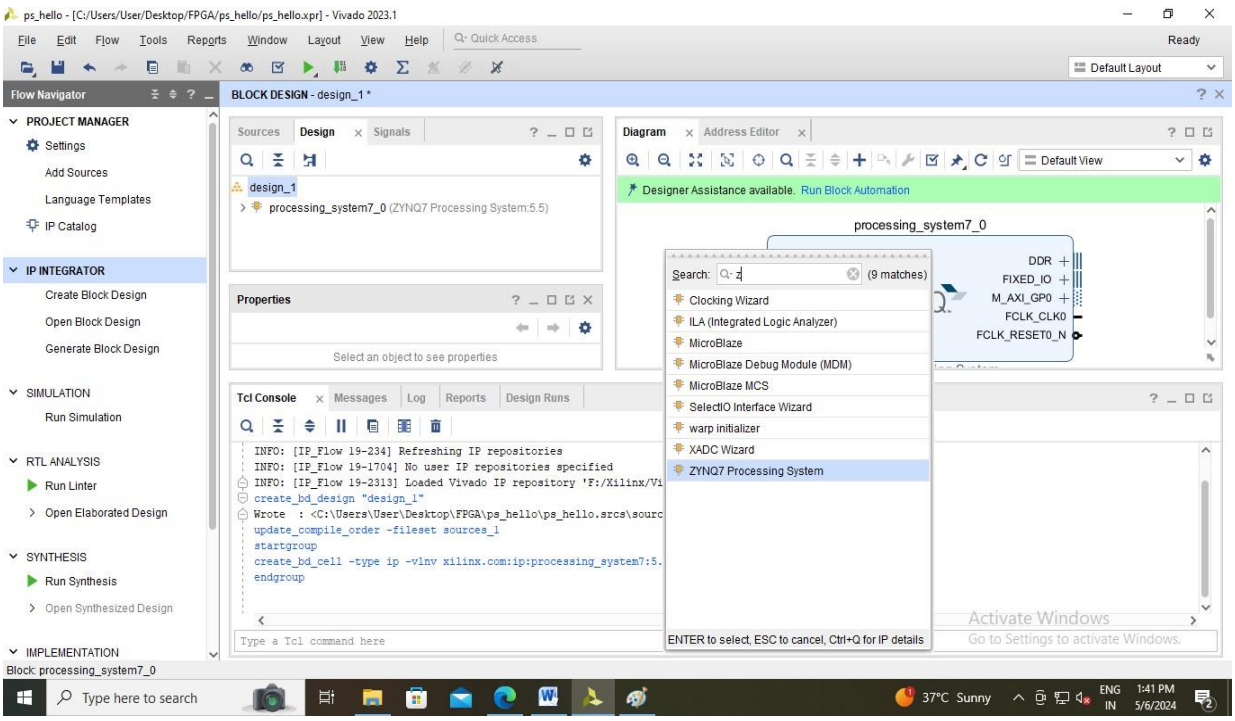
Step 10: Here no modifications can be done, keep the default "design_1"



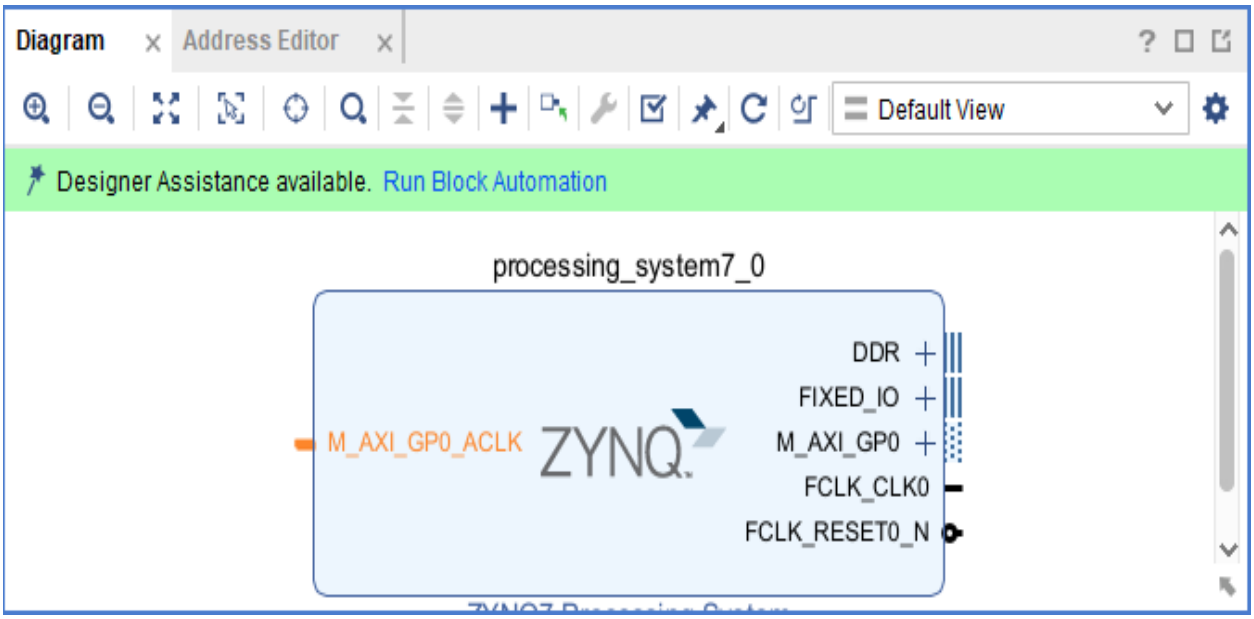
Step 11: Click "Add IP" as shown in the below image.



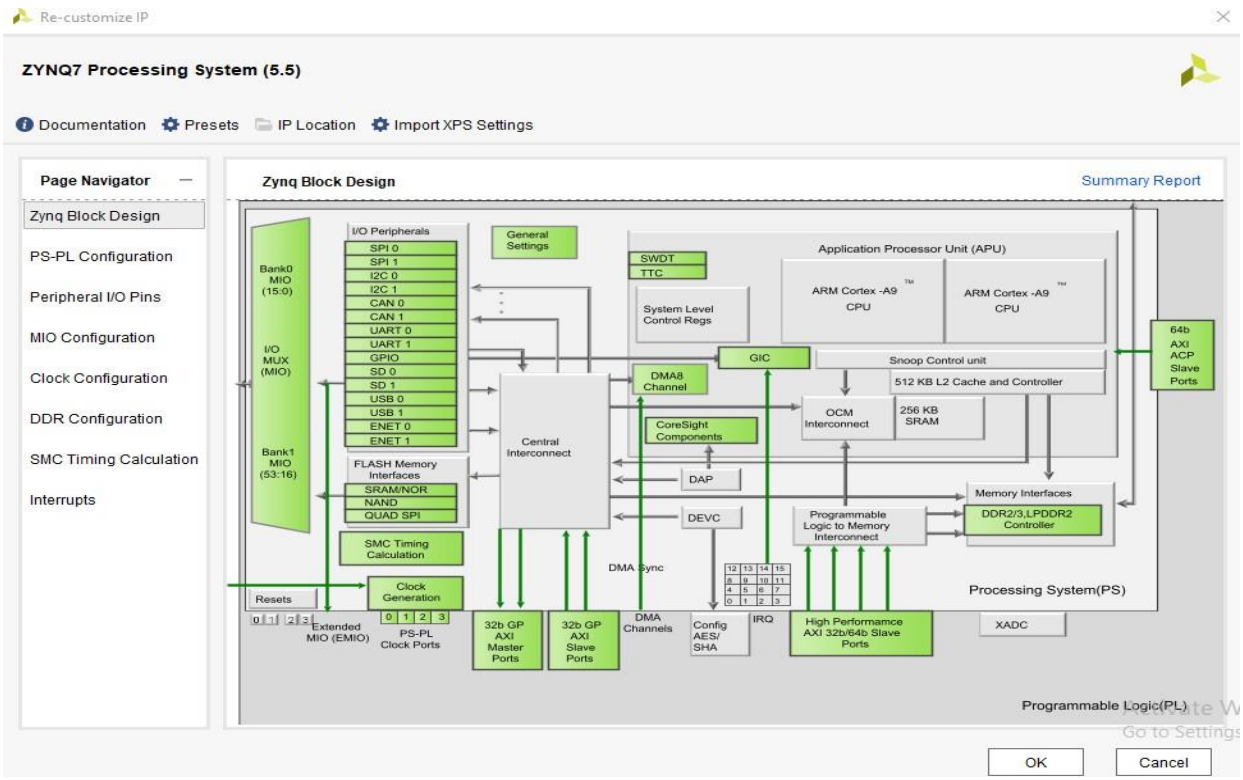
Step 12: Here in search icon search for "zynq", double-click "ZYNQ7 Processing System" as shown in the below image.


ps_hello - [C:/Users/User/Desktop/FPGA/ps_hello/ps_hello.xpr] - Vivado 2023.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
BLOCK DESIGN - design_1 *
Sources
Design
Signals
design_1
processing_system7_0 (ZYNQ7 Processing System:5.5)
Properties
Select an object to see properties
Diagram
Address Editor
Default View
Designer Assistance available. Run Block Automation
processing_system7_0
DDR
FIXED_IO
M_AXI_GP0
FCLK_CLK0
FCLK_RESET0_N
Search: z (9 matches)
Clocking Wizard
ILA (Integrated Logic Analyzer)
MicroBlaze
MicroBlaze Debug Module (MDM)
MicroBlaze MCS
SelectIO Interface Wizard
warp initializer
XADC Wizard
ZYNQ7 Processing System
ENTER to select, ESC to cancel, Ctrl+Q for IP details
Tcl Console
Messages
Log
Reports
Design Runs
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository 'F:/Xilinx/Vi
create_bd_design "design_1"
Wrote : <C:\Users\User\Desktop\FPGA\ps_hello\ps_hello.srcs\sourc
update_compile_order -fileset sources_1
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:processing_system7:5.
endgroup
Type a Tcl command here
Activate Windows
Go to Settings to activate Windows.
Block: processing_system7_0
Type here to search
37°C Sunny
ENG IN
1:41 PM 5/6/2024

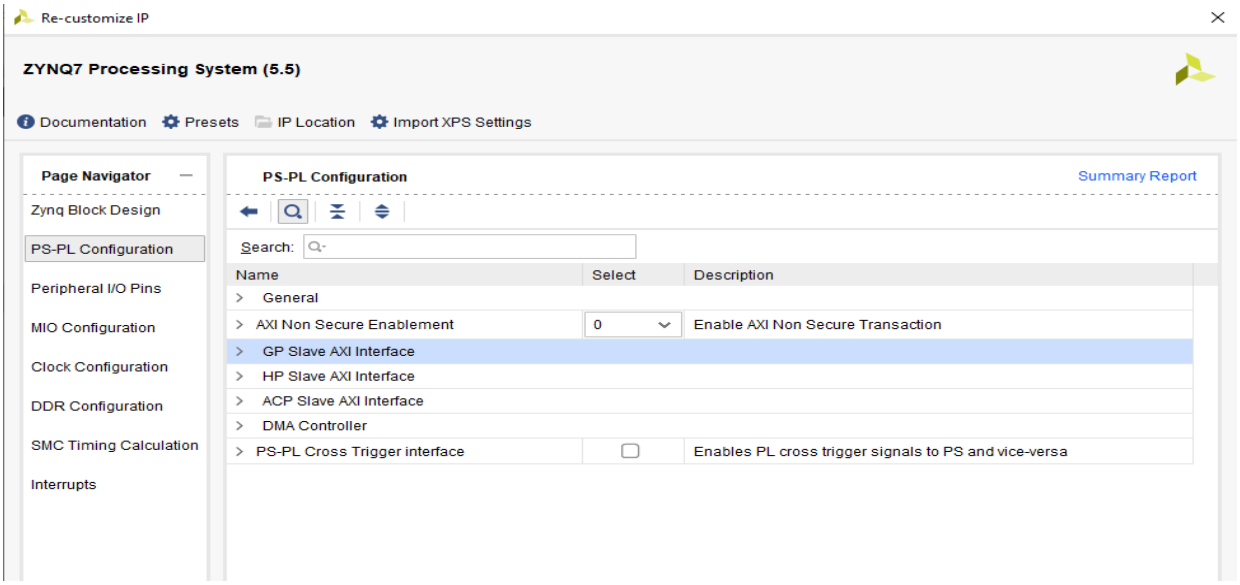
Step 13: Double click on the Block "processing_system7_0", to configure related parameters



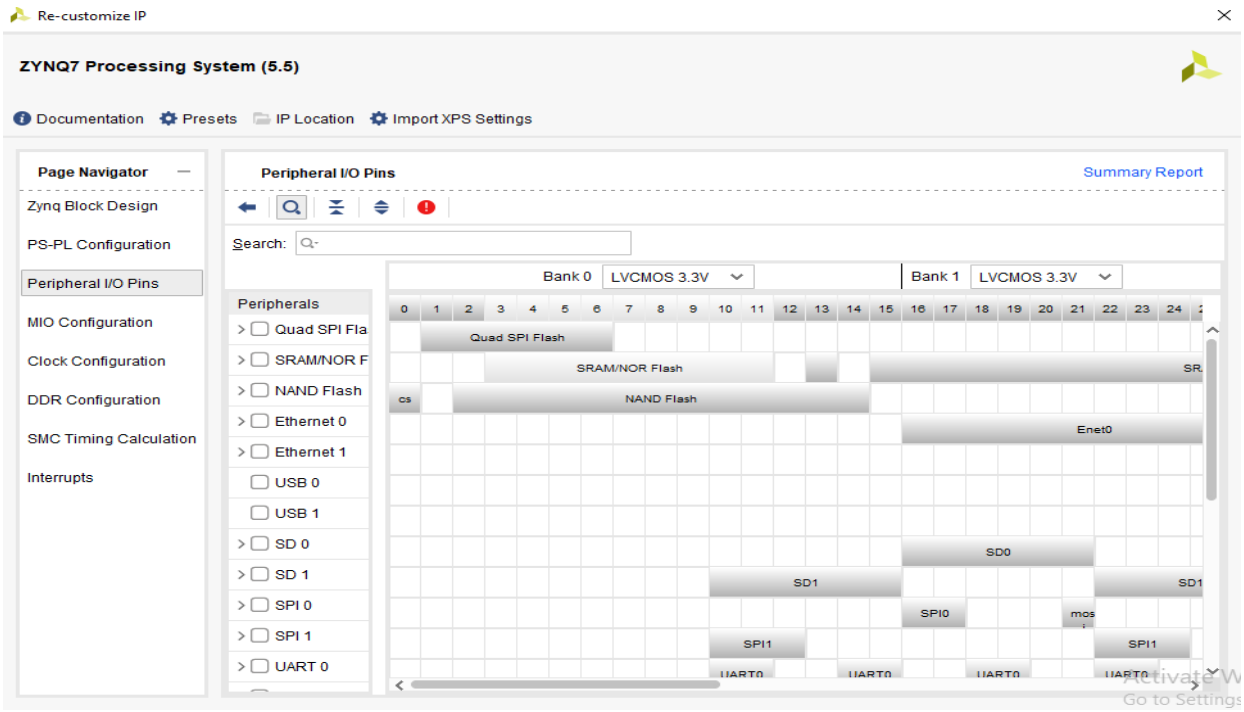
Step 14: On the image mentioned below green color are editable**.**



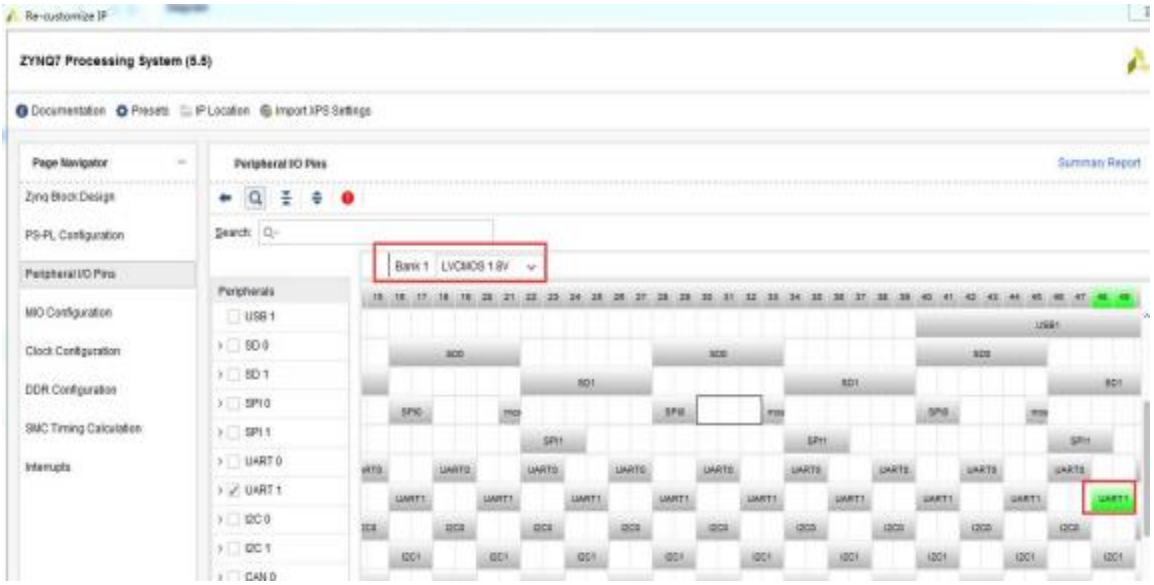
Step 15: The PS-PL interface uses AXI for data interaction between PS and PL. We'll keep the default settings for now and configure it in later chapters.



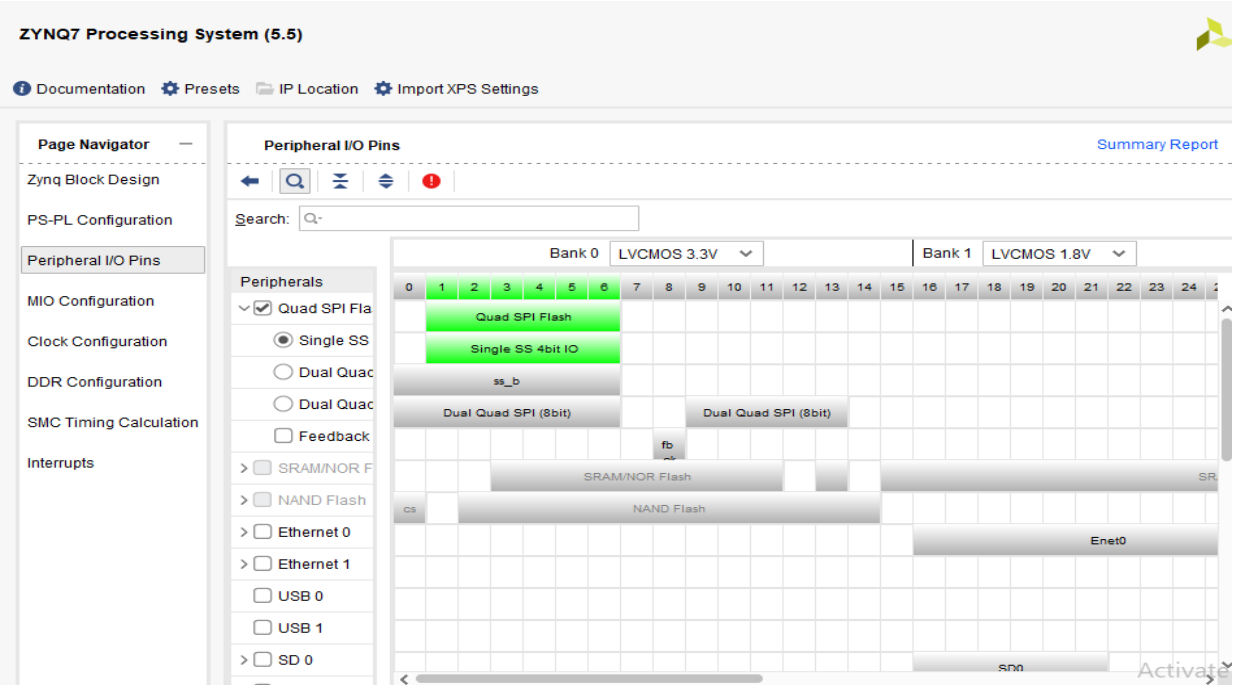
Step 16: Next, click on 'Peripheral I/O Pin'. When setting up PS peripherals in ZYNQ, the options can be overwhelming. Certain pins, like 16-27, can switch between Enet0 or SD0/SD1 functions, but you can only choose one. Check the schematic or user manual to decide which configuration fits best



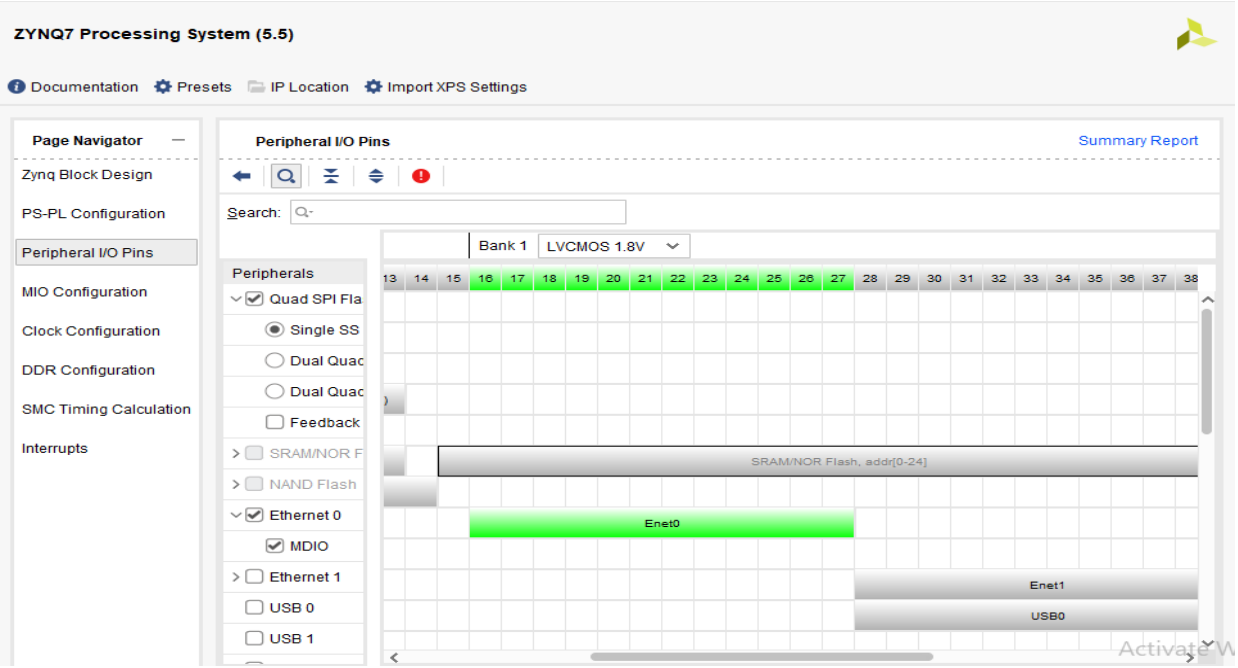
Step 17: The serial port connects to MIO48-MIO49 on the PS. Enable UART1 (MIO48 MIO49) in the 'Peripheral I/O Pins' options. PS MIO is split into two banks: Bank 0 (BANK500) uses 3.3V LVCMOS, and Bank 1 (BANK501) uses 1.8V LVCMOS, as shown in the schematic diagram



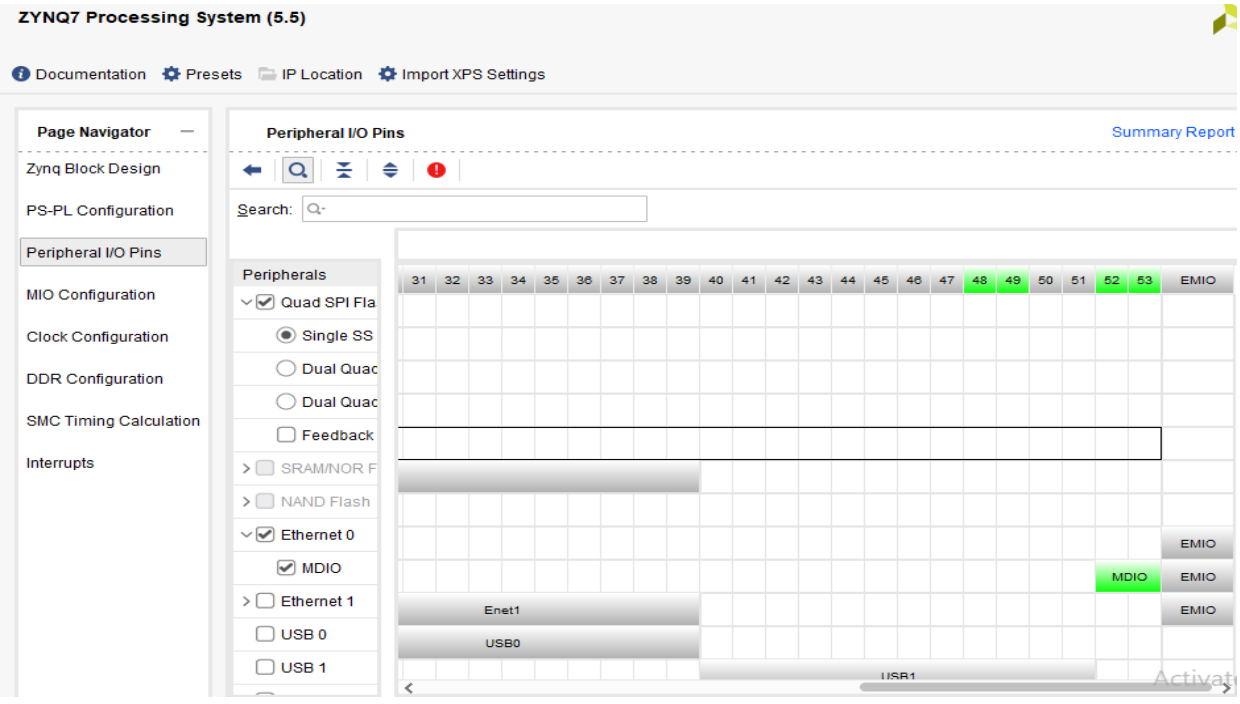
Step 18: Next configuration QSPI. select Quad SPI Flash in that click on Single SS 4-bit IO.



Step 19: Configure Ethernet. The terminal is designed with an Ethernet interface, which can be selected according to the schematic diagram.



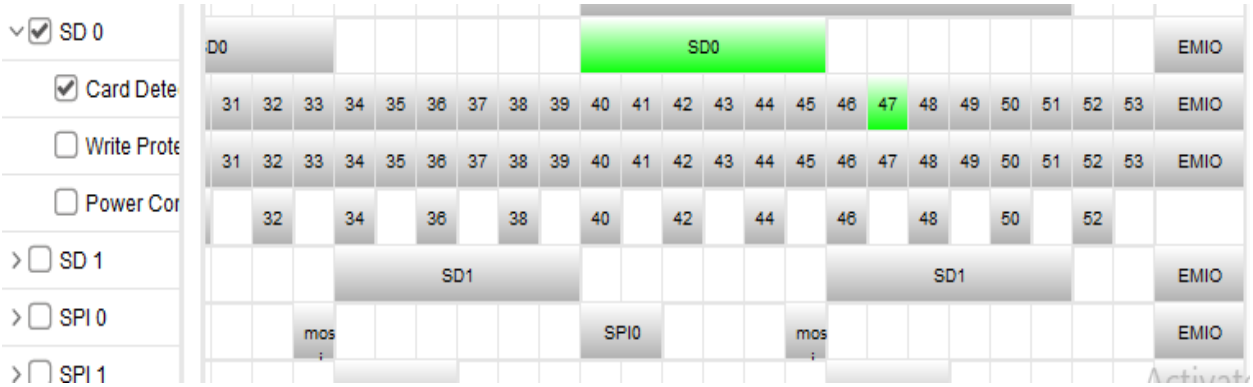
Step 20: Expand the Ethernet 0 and Select MDIO. After selecting you can MDIO in green color as shown in the below image.



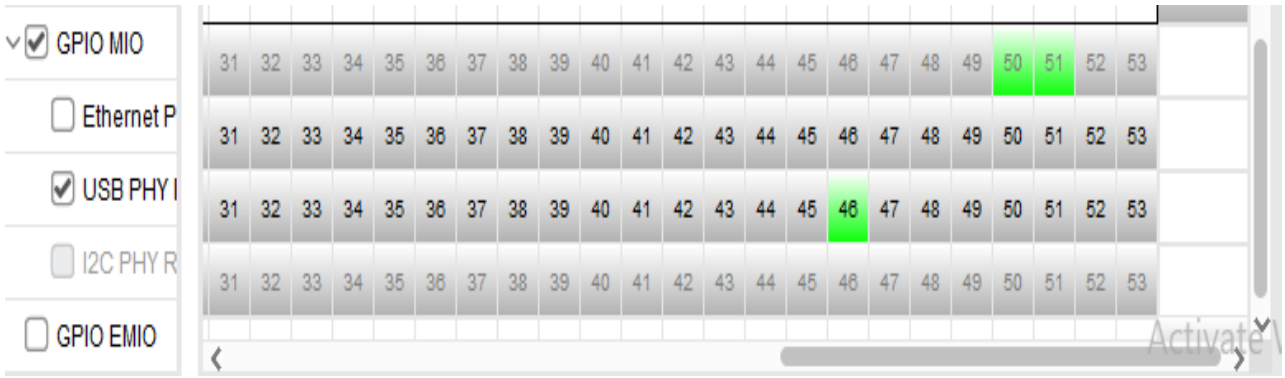
Step 21: To Configuration USB0 by selecting the USB0 as show in the below image.

Step 22: In addition to QSPI, ZYNQ can also boot from an SD card. Choose SD 0 and configure it for

MIO40-MIO45. Use MIO47 for card detection to detect SD card insertion.

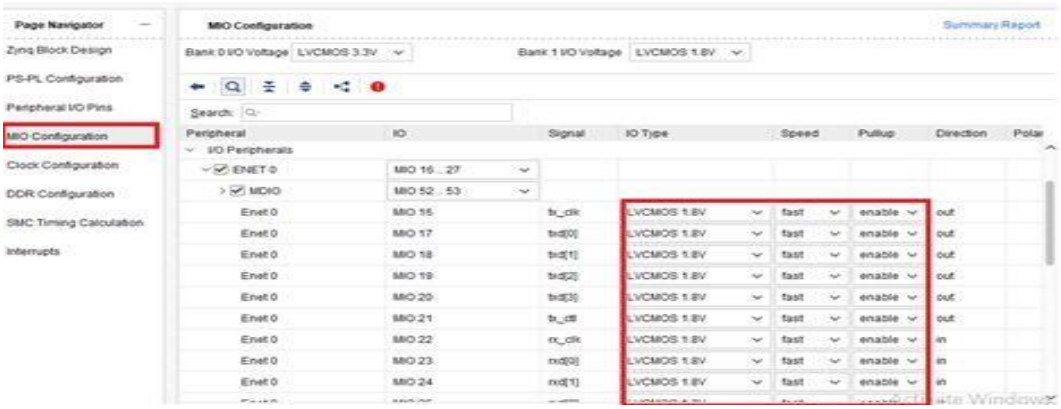


Step 23: In GPIO mode on the PS, we can utilize the remaining unallocated MIO pins as general-purpose I/O. Select MIO46 for USB PHY reset in the GPIO MIO configuration. At this point, the peripheral configuration is completed

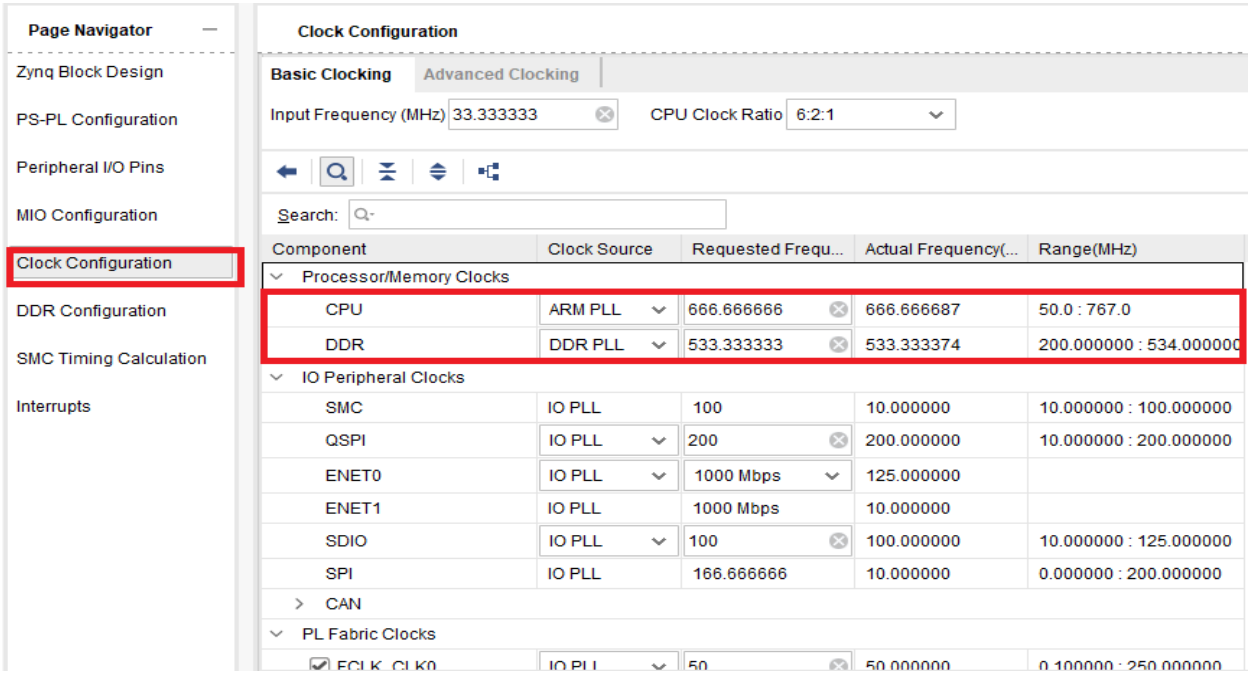


Step 24: Next MIO Configuration.

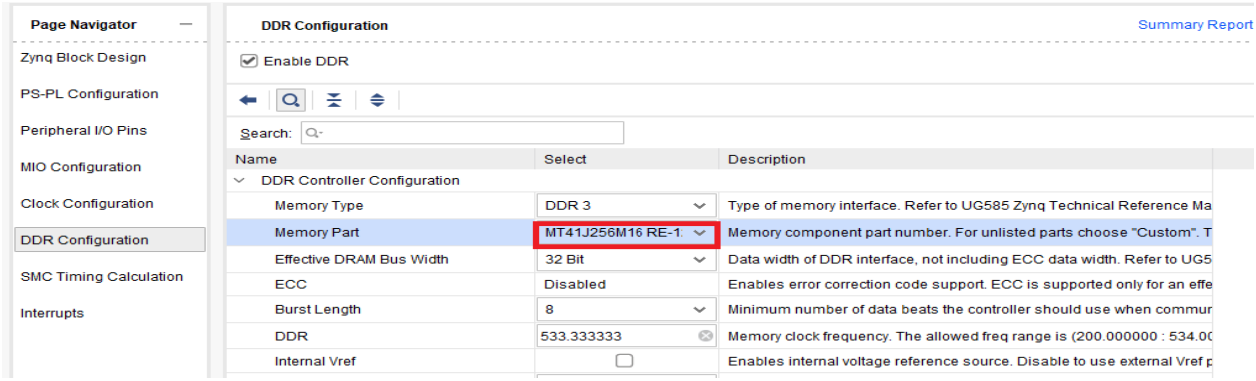
In MIO Configuration. Ensure that Enet0 operates at LVCMOS 1.8V with a fast speed setting and enable Pullup. These parameters are crucial; failure to adjust them may lead to network connectivity issues. Keep other settings as default.



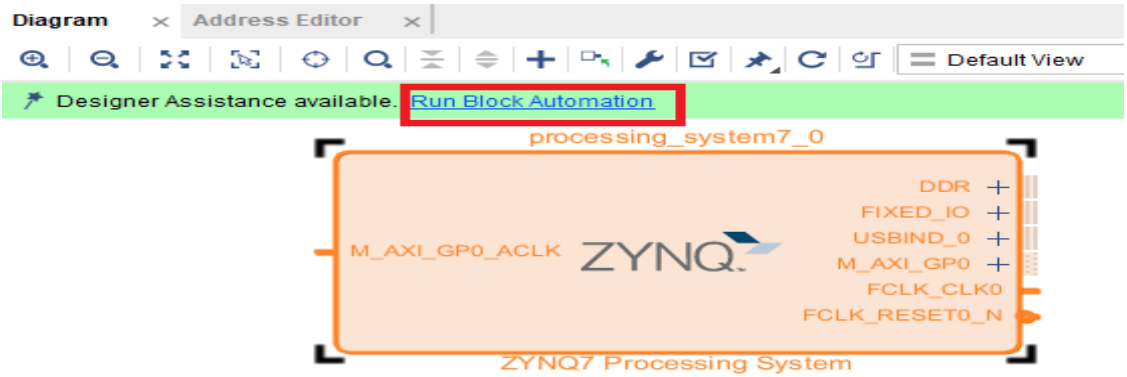
Step 25: Next, proceed to the clock configuration. Ensure that the CPU and DDR values match those shown in the image below.



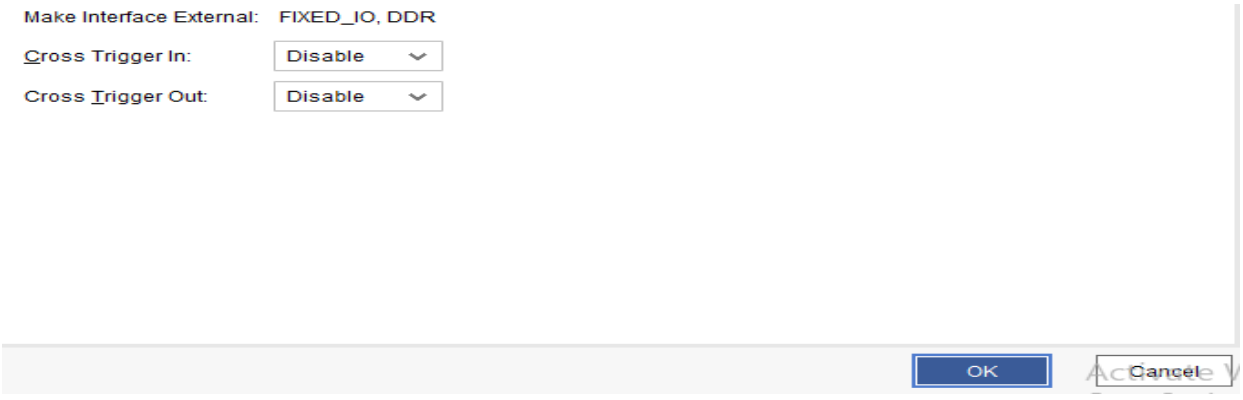
Step 26: Change the model number to "MT41J256M16 RE-125", leave other settings as default, and click OK. This completes the core configuration of ZYNQ so far.



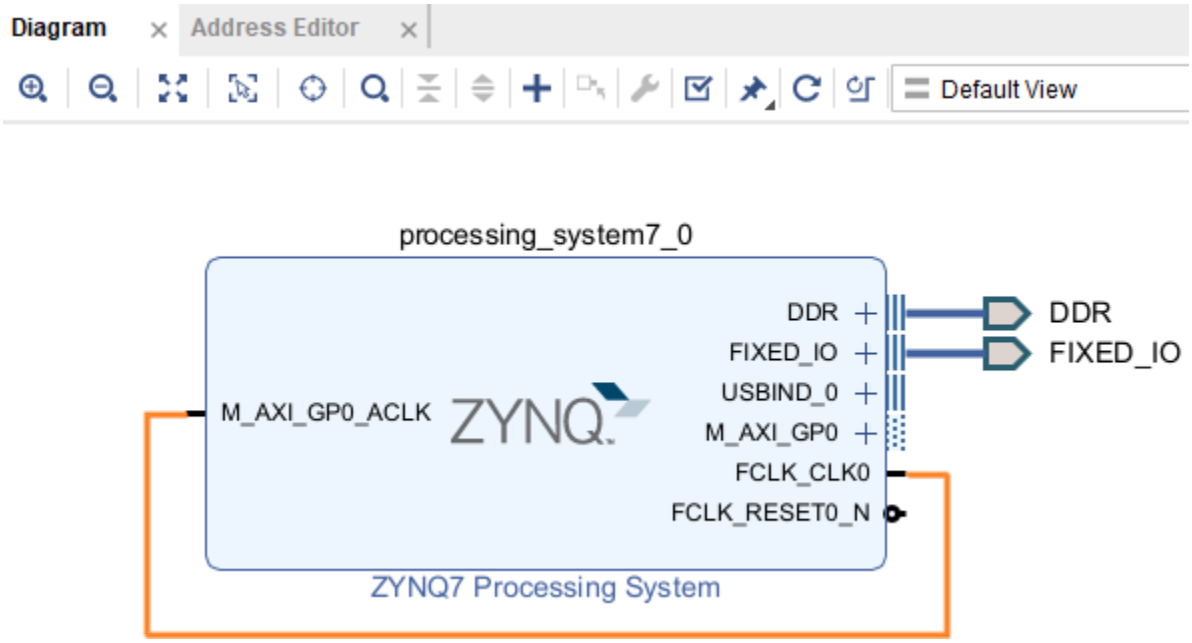
Step 27: Click "Run Block Automation" in Vivado. The software will automatically complete some tasks related to exporting ports.



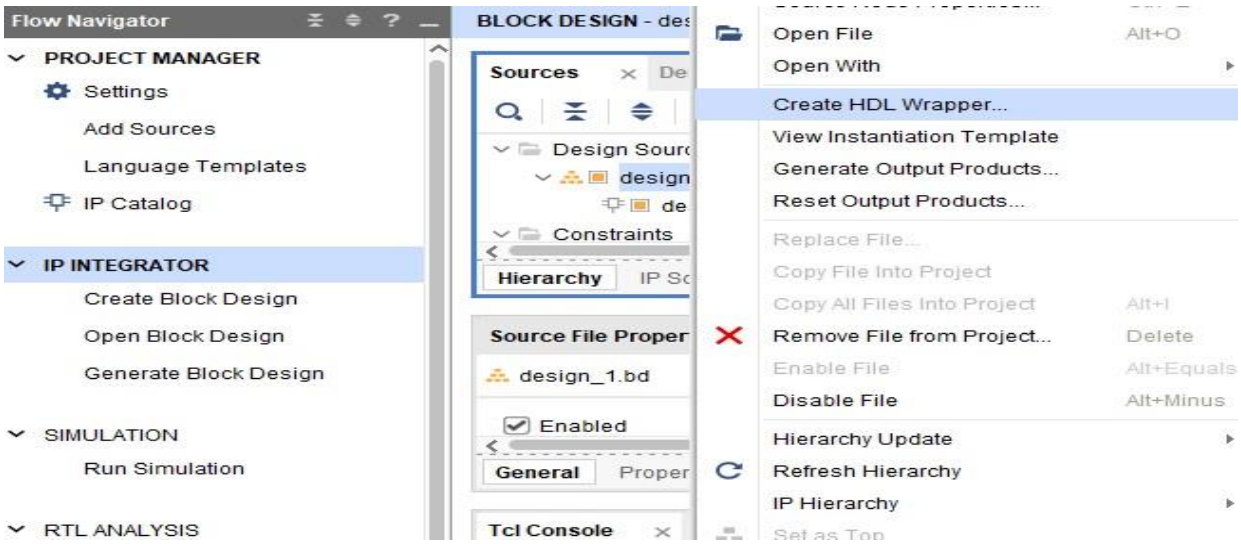
Step 26: Click "by default "OK"



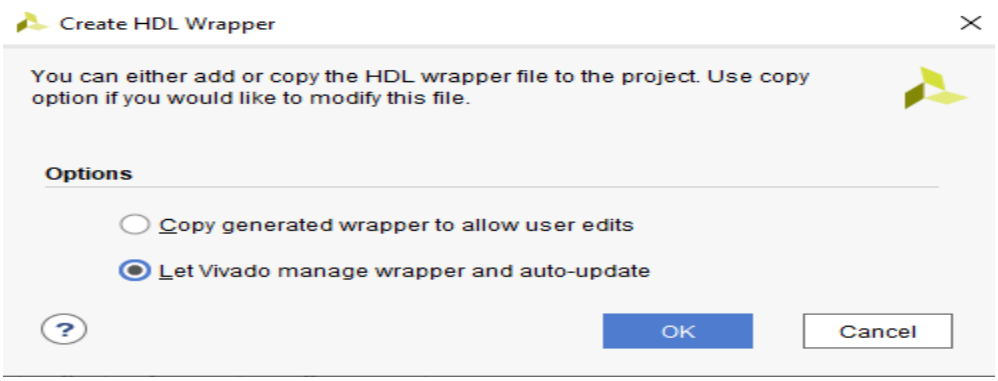
Step28: Connect FCLK_CLK0 and M_AXI_GP0_ACLK, according to save design.



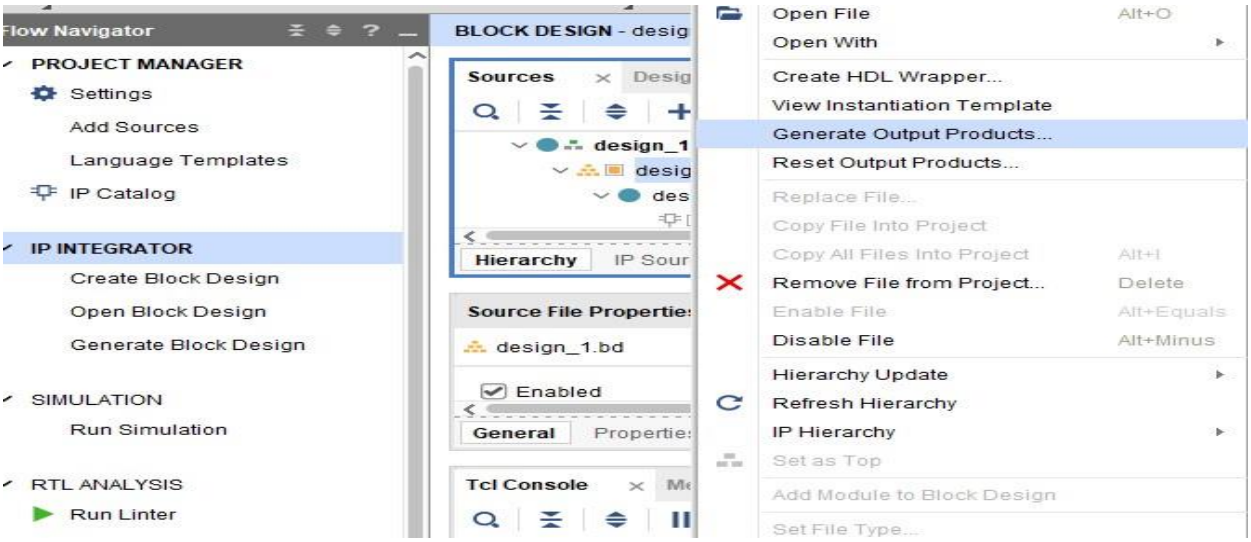
Step 29: Choose "Block Design," then right-click and select "Create HDL Wrapper..." to generate a Verilog or VHDL file for the block design, creating the HDL top-level file.



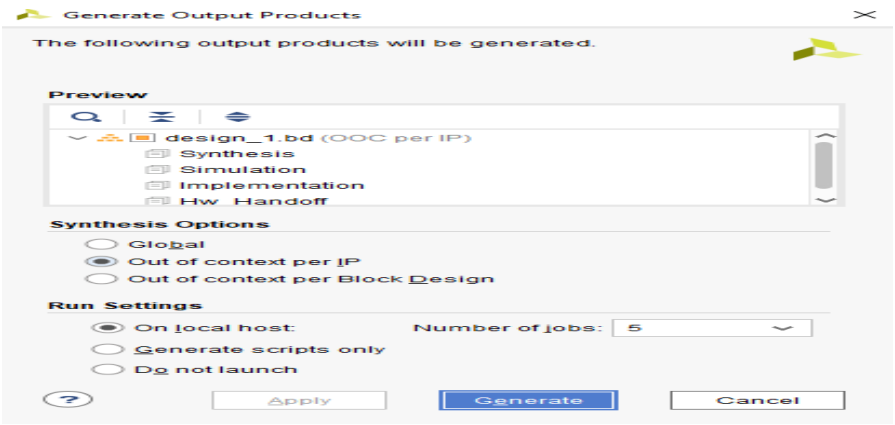
Keep the default options and click OK.



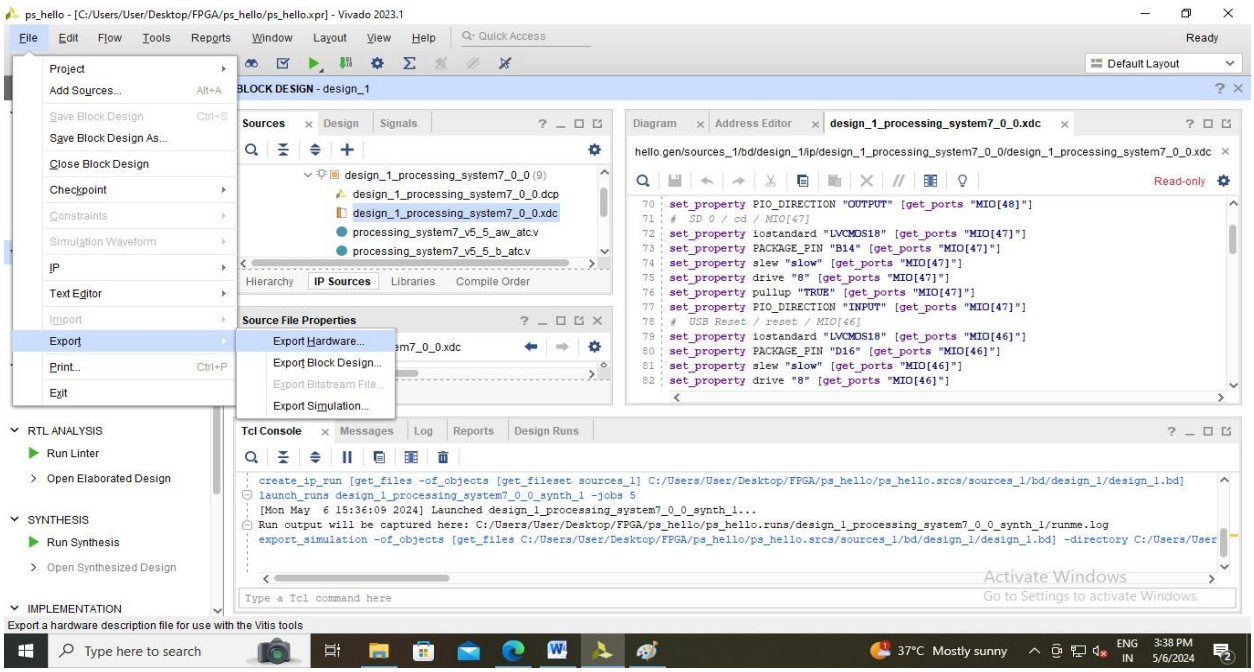
Step 30: In the block design view, right-click and choose "Generate Output Products". This step creates essential files like IP, RTL source, and constraints for further work.



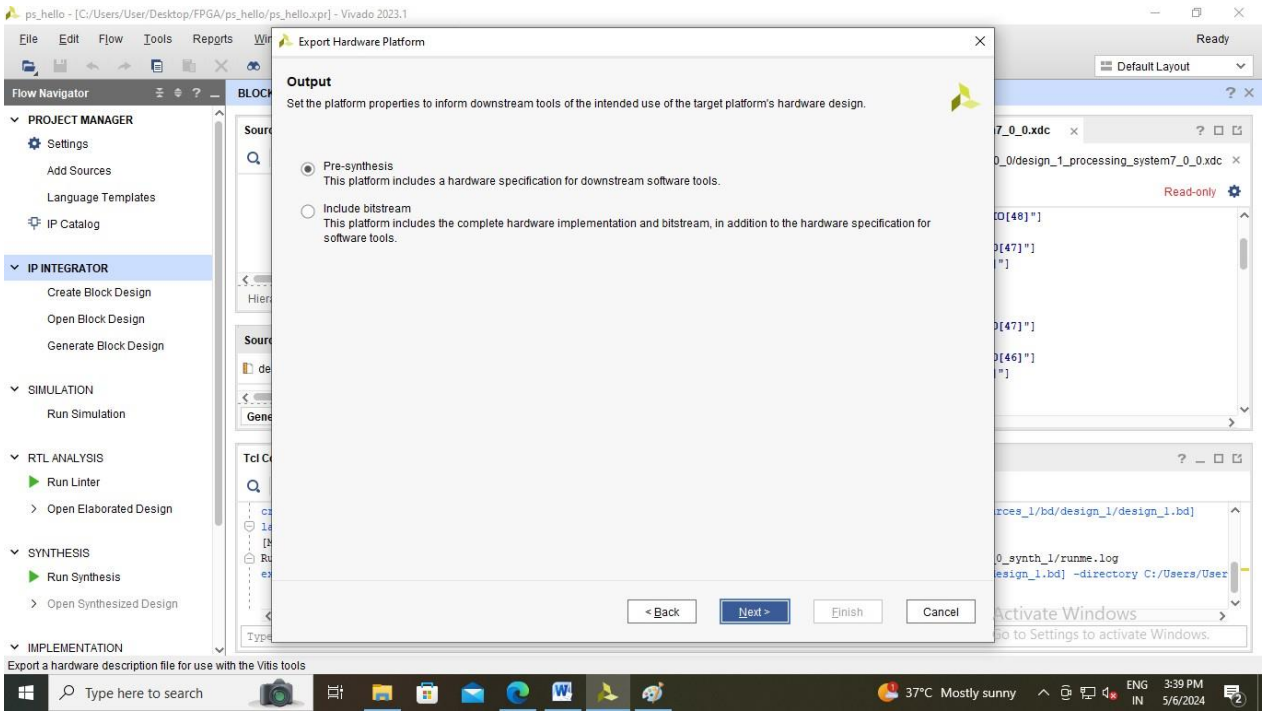
Step 31: Click on "Generate".



Step 32: In the menu bar, go to "File -> Export -> Export Hardware..." This option exports hardware information, including PS terminal configuration details.



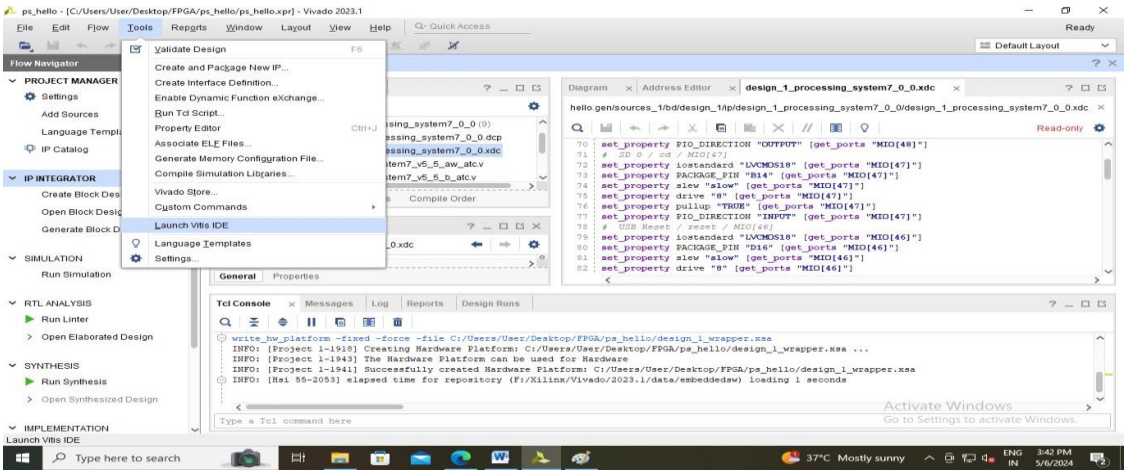
Step 33: Click Next and Finished



Hardware design part is completed. Now we need to launch Vitis.

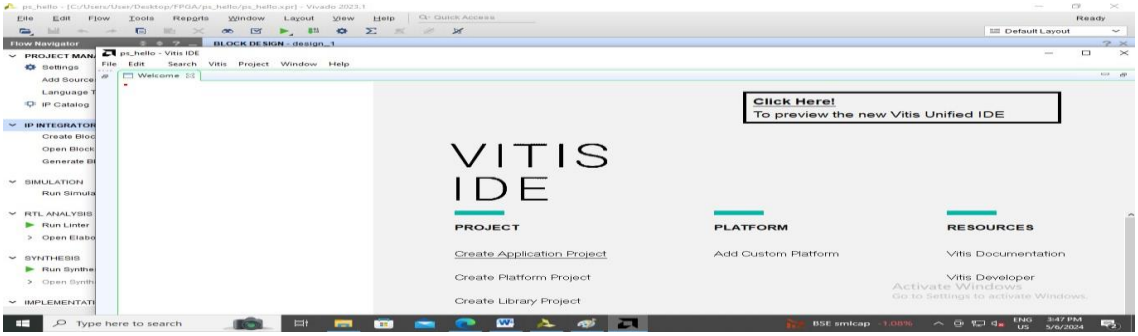
Create Application project

Step 1: Vitis is an independent software tool. To open Vitis, go to "Tools -> Launch Vitis" and start the Vitis software.

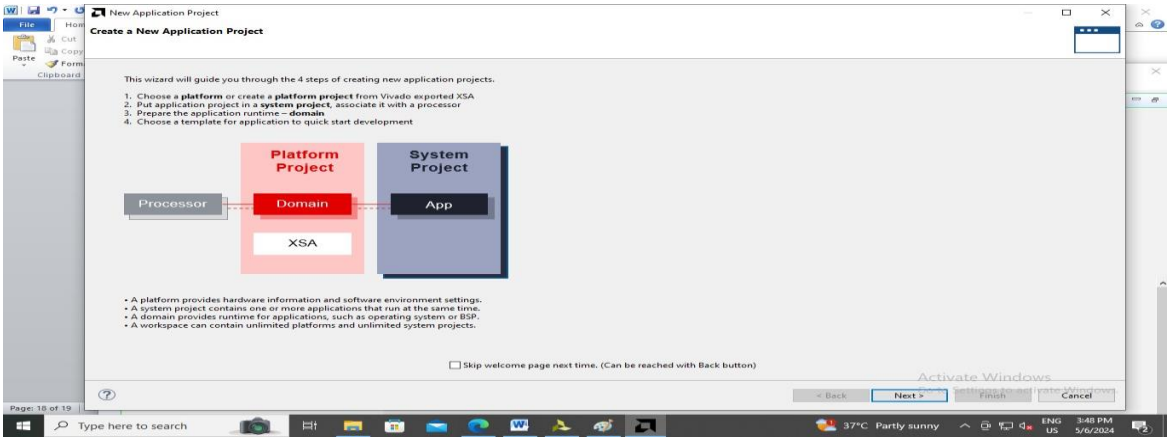


Step 2: Select the previously created folder and click "Launch".

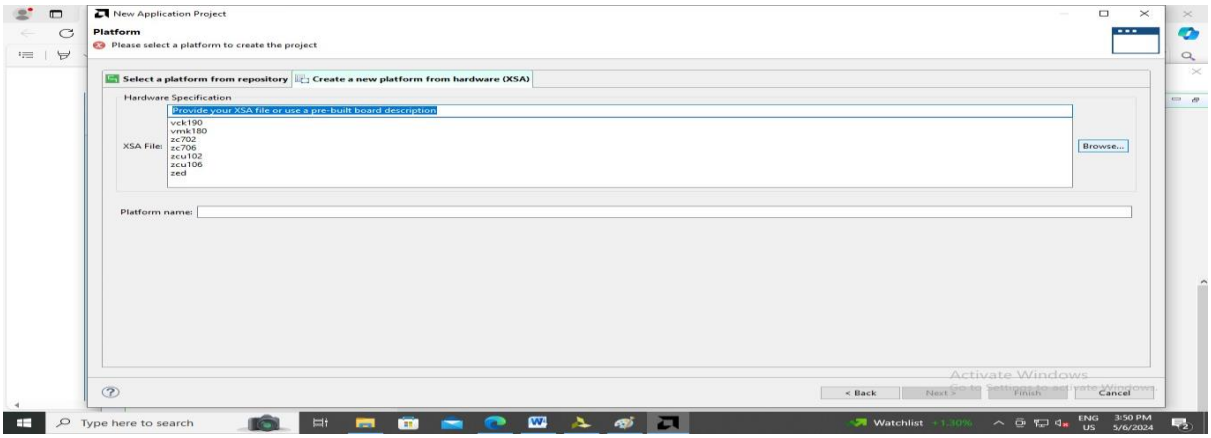
Step 3: Click on "Create Application Project."



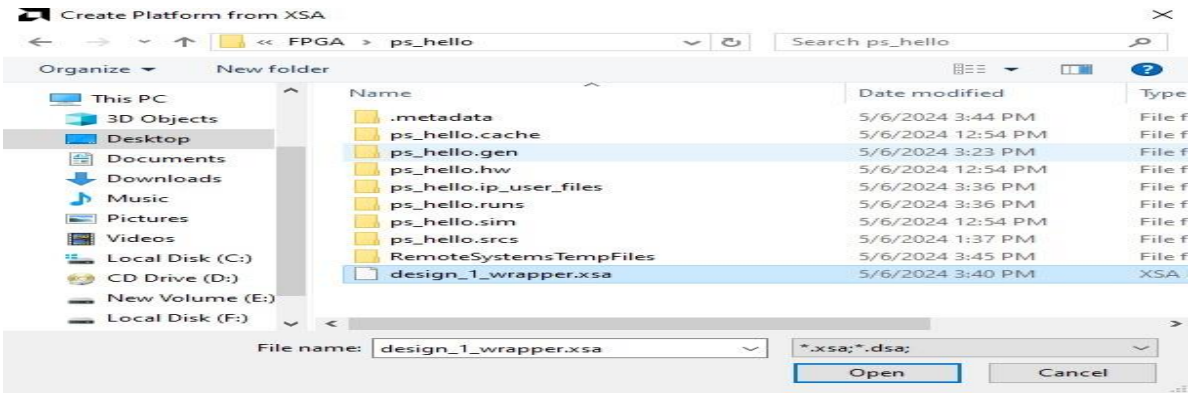
Step4: Click on Next.



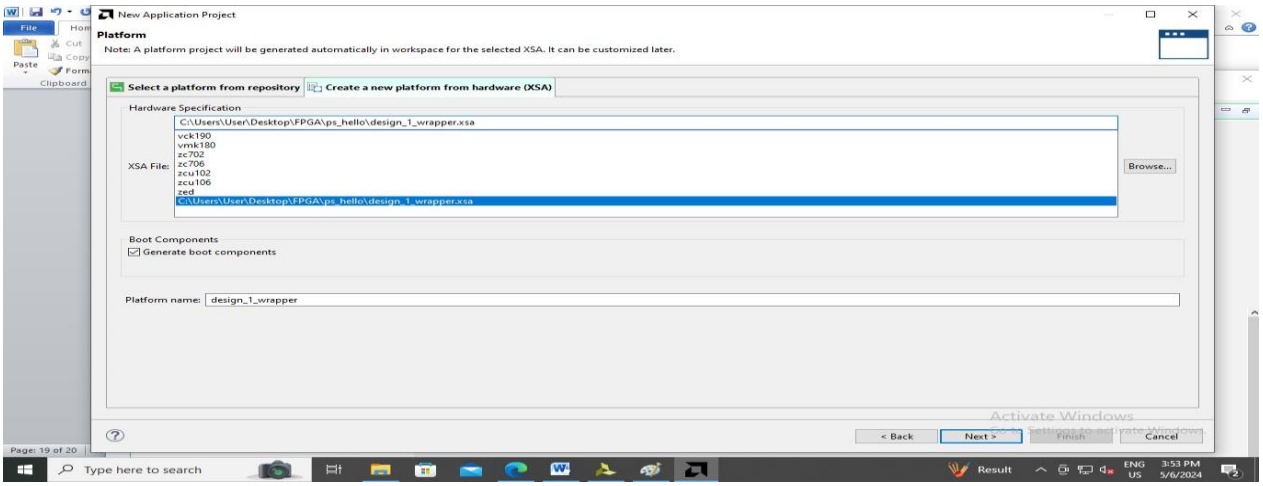
Step 5: Click on "Create a new platform hardware (XSA)". The software provides some hardware platforms for boards, but for your own hardware platform, you can choose "+".



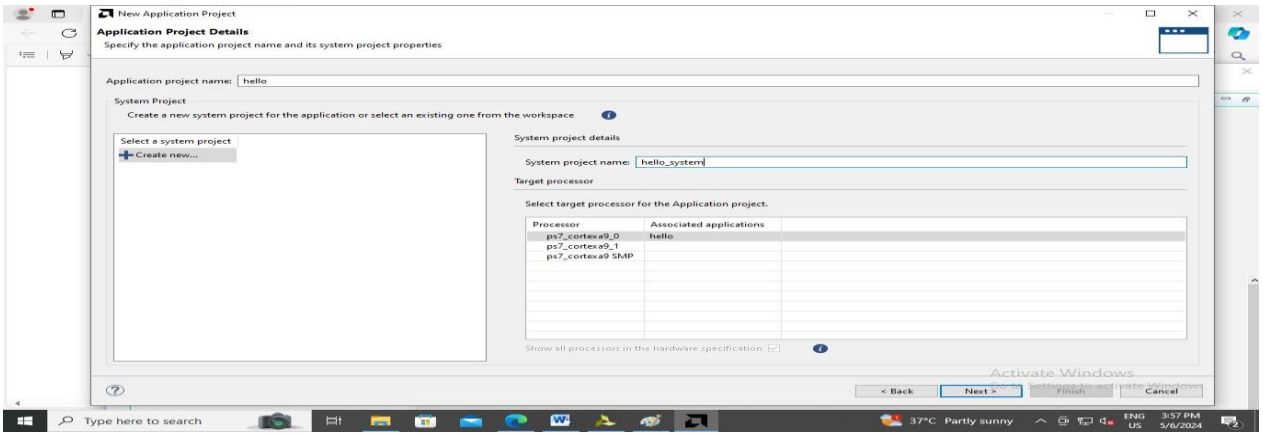
Step 6: Select the previously generated XSA file in project folder and click "Open".



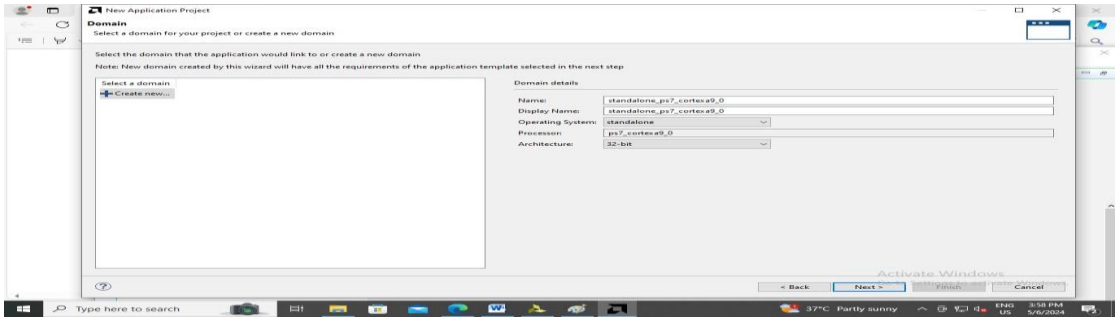
Step 7: Under "Generate boot components," check the option to have the software automatically generate boot components for the project. This is usually selected by default. Click "Next."



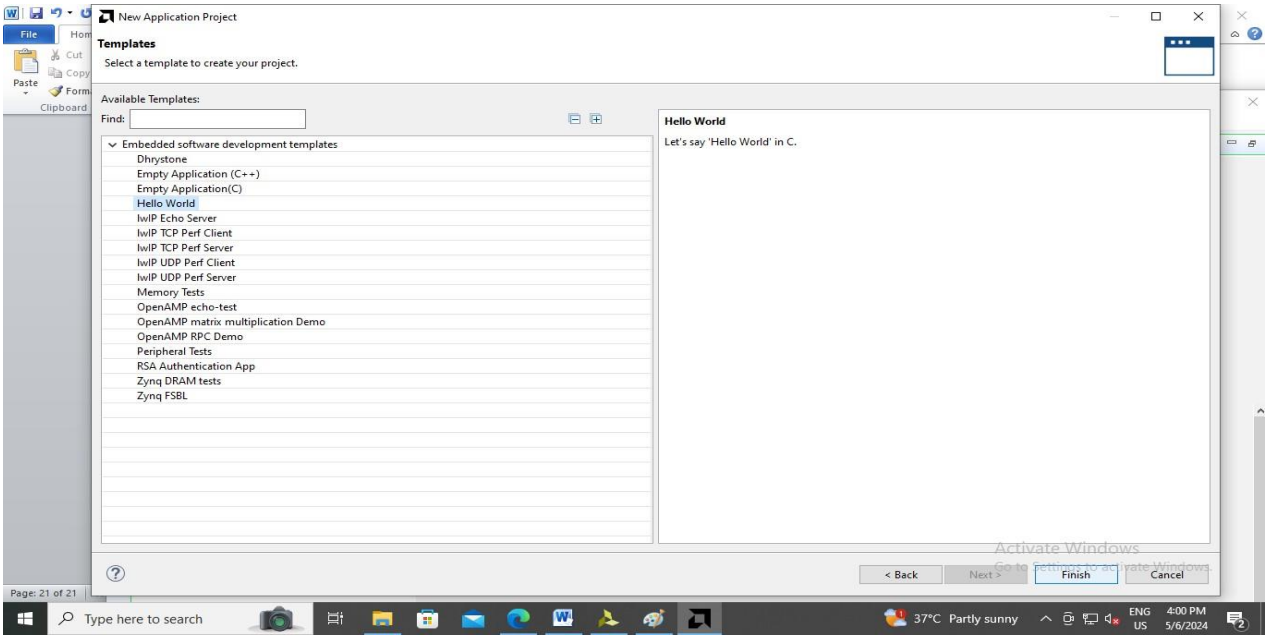
Step 8: Fill in the project name as "hello" and provide a project description if needed. Click "Next."



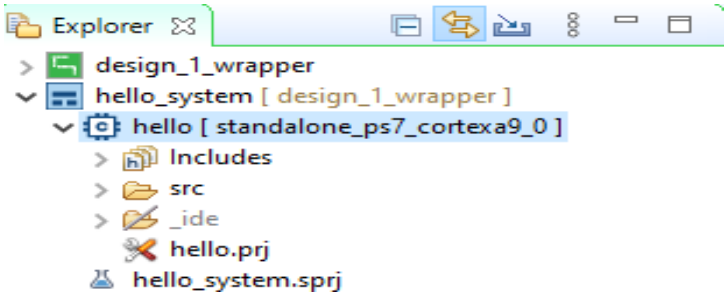
Step 9: Keep everything as it is and click "Next."



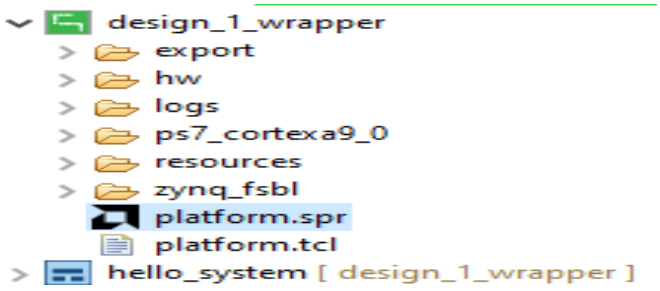
Step 10: In the template selection, choose "Hello World" and click "Finish."



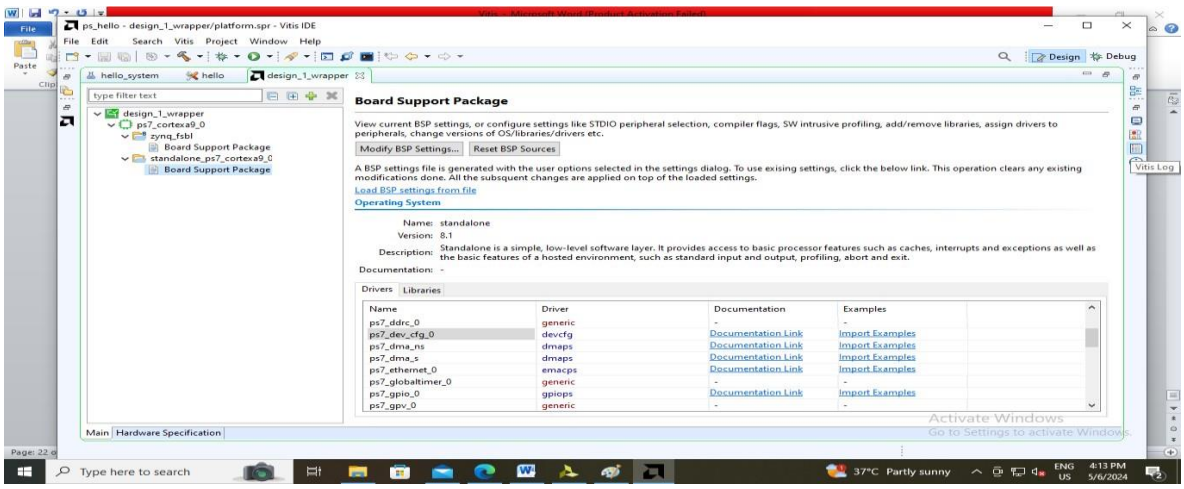
Step 11: After completion, you will see that the projects have been generated.



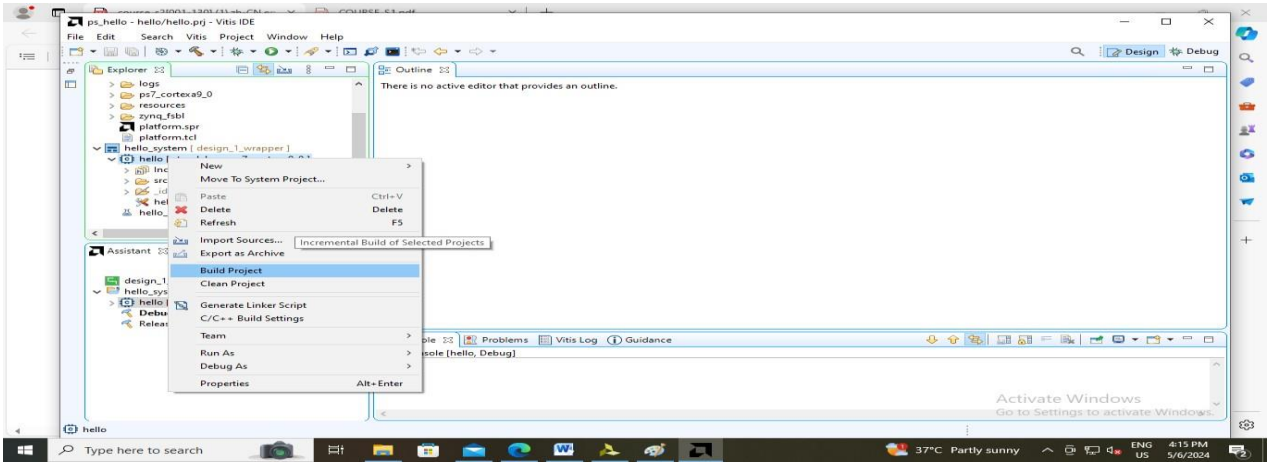
Step 12: Double-click on the .spr file in design_1_wrapper.



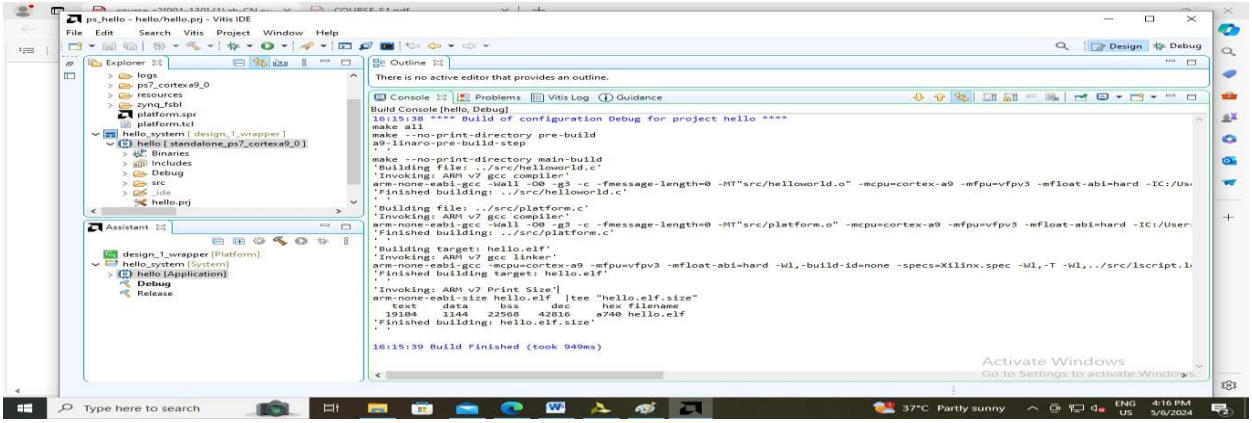
Step13: Click to open the Board Support Package (BSP). Here, you can find the peripheral drivers included in the project. Xilinx provides driver documentation and example imports within the BSP.



Step 14: In the selected Application Project, right-click and choose "Build Project," or click the "Hammer" button on the menu bar to compile the project.



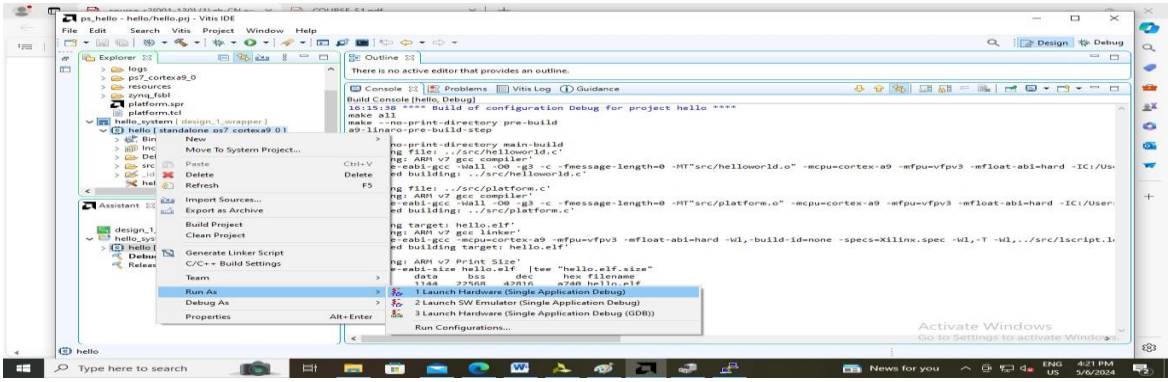
- Console See the compilation process and generate self document



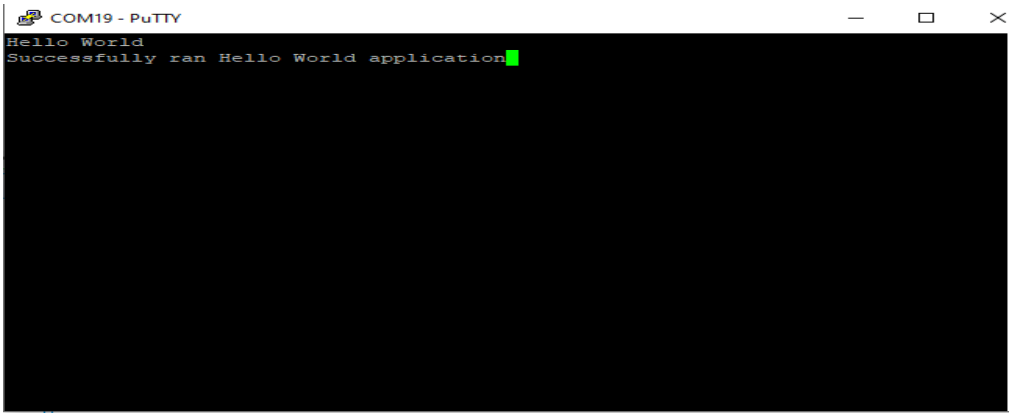
Step 15: Connect the JTAG line to the development board and the UART or USB line to the PC.

Step 16: Use PuTTY, a software that serves as a serial port terminal debugging tool.

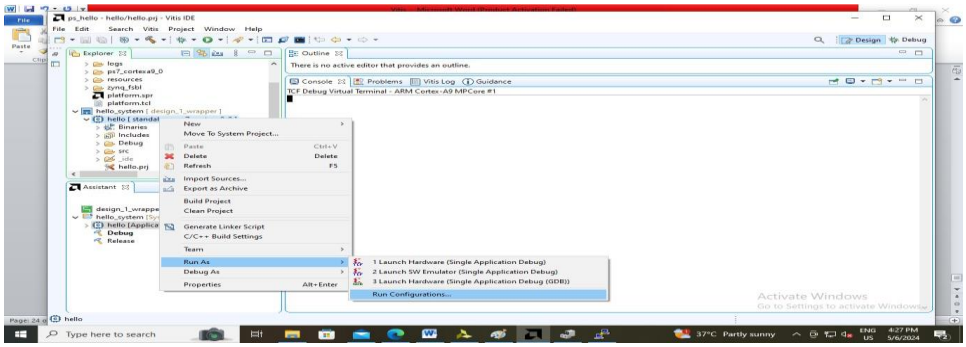
Step 17: Click on "Run As" and select "Launch Hardware (Single Application Debug)."



Step 18: Observe PuTTY software at this time. You should see the output "Hello World."

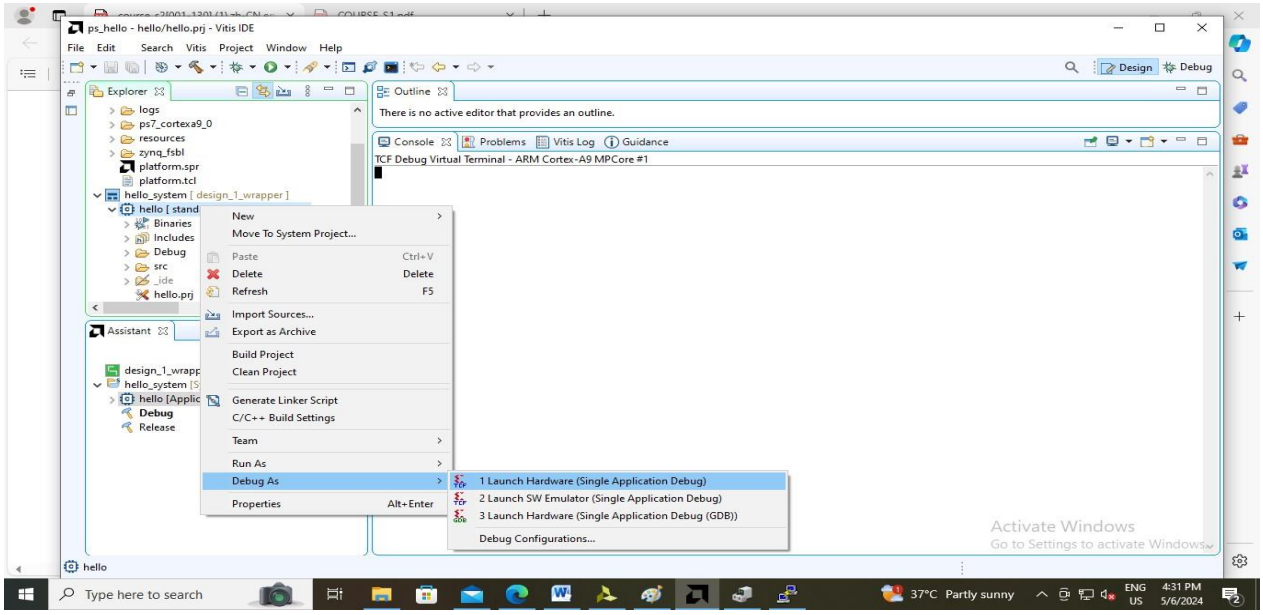


Step 19: For reliable system debugging, it's recommended to right-click on "Run As -> Run Configuration."

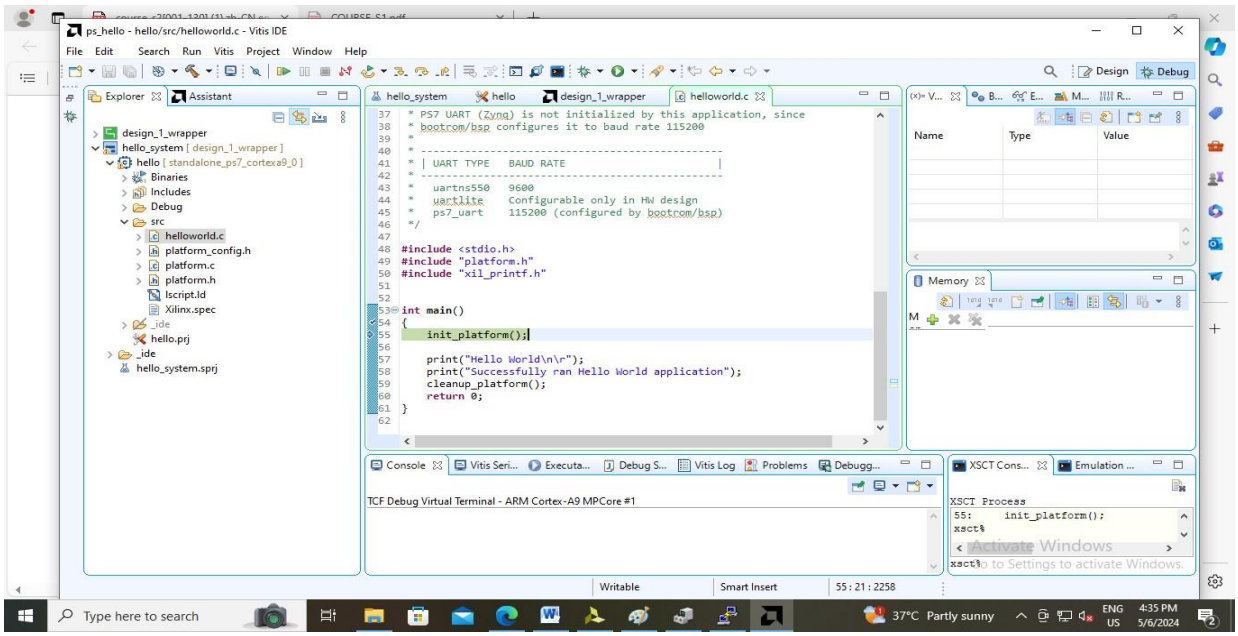


Inside the configuration, you will find "Reset entire system" selected by default, similar to Vitis. If your project includes PL design, ensure to also select "Program FPGA" for complete functionality.

Step 20: Apart from "Run As," there's also "Debug As" available. This allows you to set breakpoints and execute single-step debugging. Click on debug and select "Launch Hardware (Single Application Debug."



Step 21: After debugging, you can view the breakpoints you have set.



2514/U, 7th 'A' Main Road, Opp. to BBMP Swimming Pool, Hampinagar, Vijayanagar 2nd Stage.Bangalore-560104, Karnataka , 560104
Ph: 080-2340-4924 Email:info@tenettech.com