AMD VITIS + VIVADO WORFLOW TUTORIAL

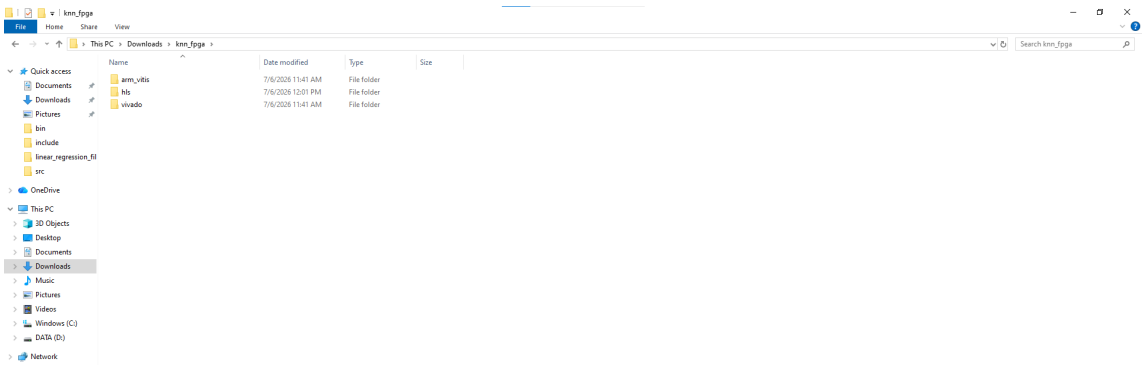
Created by: Parth Sharma (UH Junior)

This document provides a step-by-step guide for implementing a hardware accelerator using the AMD Vitis HLS, Vivado, and Vitis software development workflow. The process begins with designing and verifying a hardware module in Vitis HLS, where C/C++ code is synthesized into a hardware IP core. The generated IP is then integrated into a hardware design using Vivado, where the complete system is configured, connected, and used to generate a bitstream and hardware platform (XSA). Finally, the exported hardware platform is imported into Vitis to develop, build, and execute the embedded software application that interfaces with the hardware accelerator.

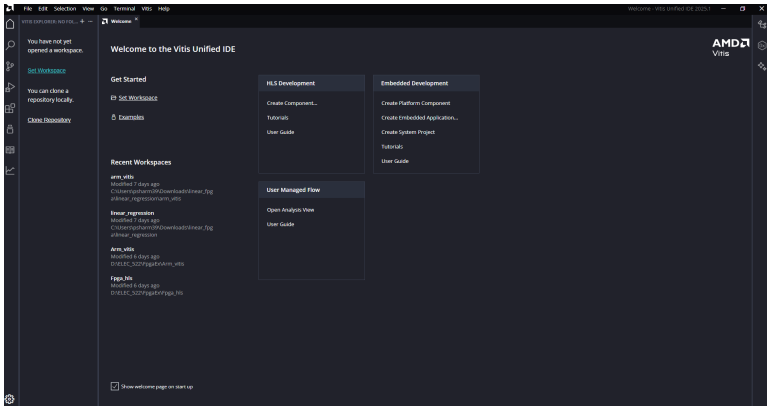
Following this workflow ensures a structured approach to hardware/software co-design, enabling efficient development, verification, and deployment of FPGA-based applications. This guide is intended for users who are developing custom IP cores and integrating them into AMD Xilinx FPGA platforms using the Vitis Unified IDE.

HLS Component

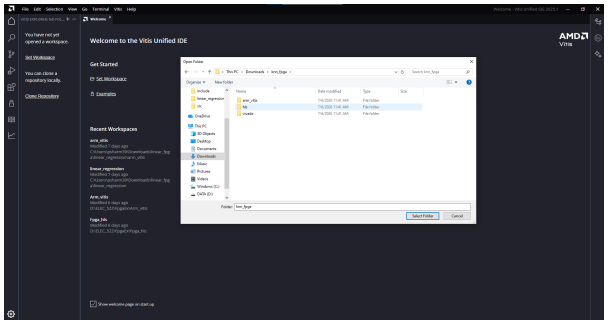
Step 1: Create 3 sub-folders in a folder each for hls component, Vivado and for Vitis.



Step 2: Open Vitis and set workspace

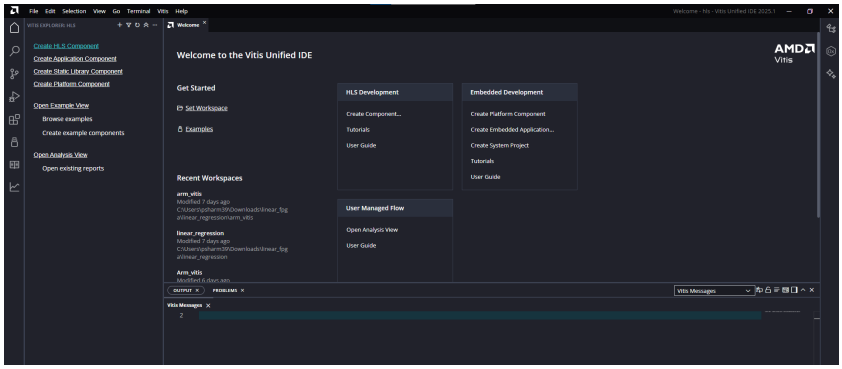


Step 3: Choose your hls folder

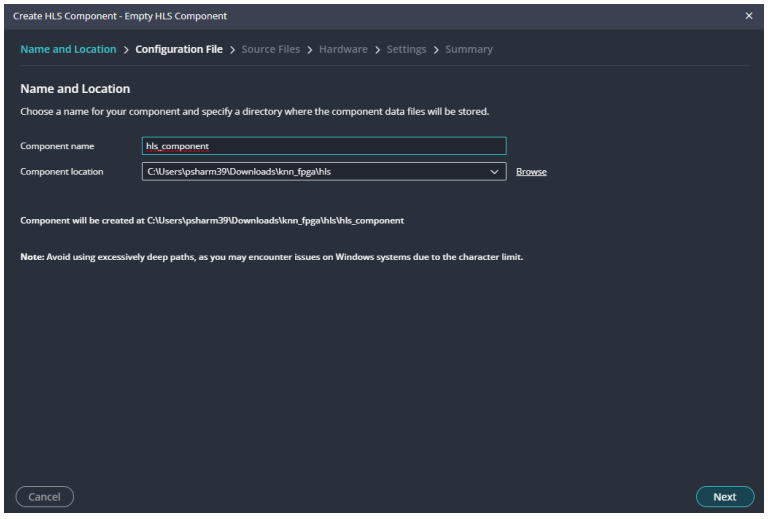


It will create a workspace for you !!!!

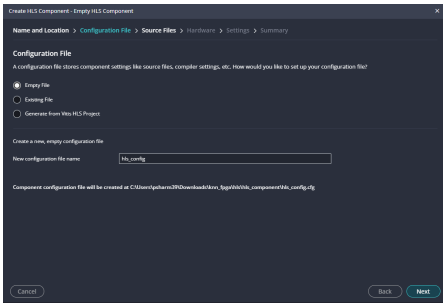
Step 4: Click on create hls component on left taskbar



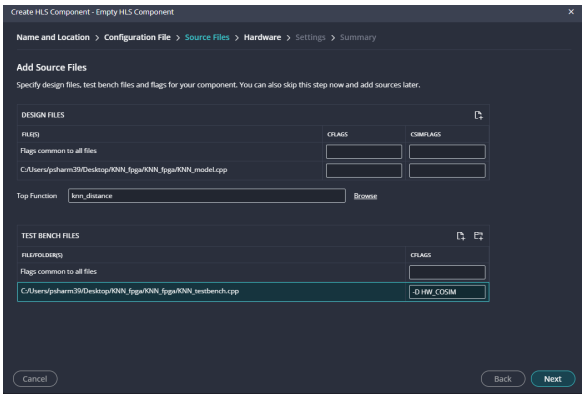
Step 5: A window will appear. Give your component name and component location. Click on next.



Step 6: Click on empty file and give a new configuration file name. Click on next.

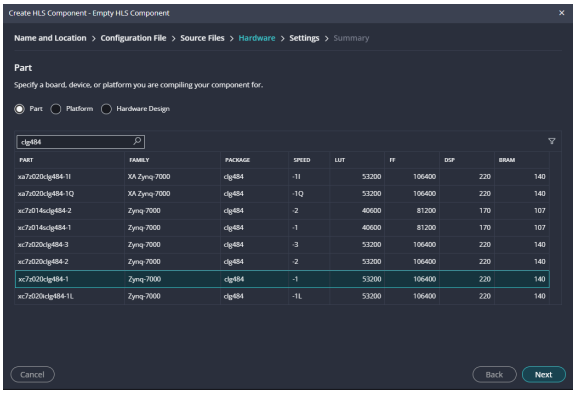


Step 7: Add your source file and test bench file. Write '-D HW_COSIM' under CFLAGS for testbench file only. Also, click on browse under top function section to get the top function. Click on next.

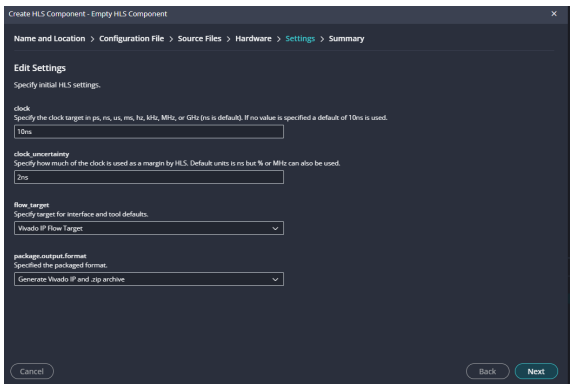


NOTE: In AMD Vitis HLS, -D HW_COSIM is a C/C++ preprocessor macro used in testbenches to switch between pure software simulation and hardware co-simulation. It allows you to conditionally compile test code that compares C-simulation outputs against RTL-simulation outputs (C/RTL Co-Simulation) to ensure hardware accuracy.

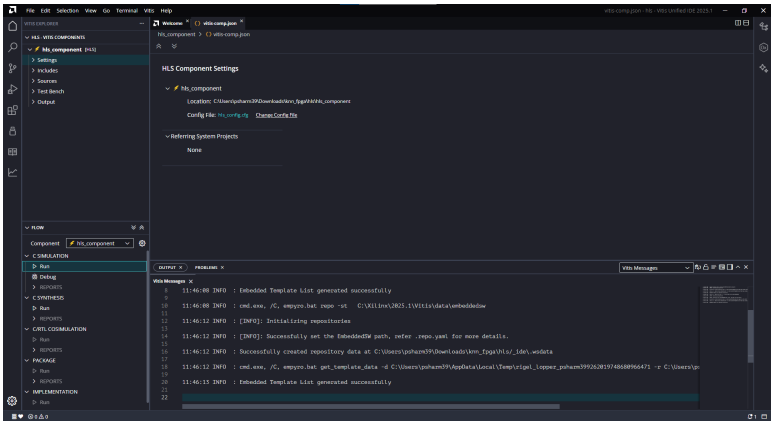
Step 8: Select the hardware (part/platform/hardware design) you are working on and click on next



Step 9: Edit settings by entering the clock, clock_uncertainty, flow_target and package.output.format. Click on next.

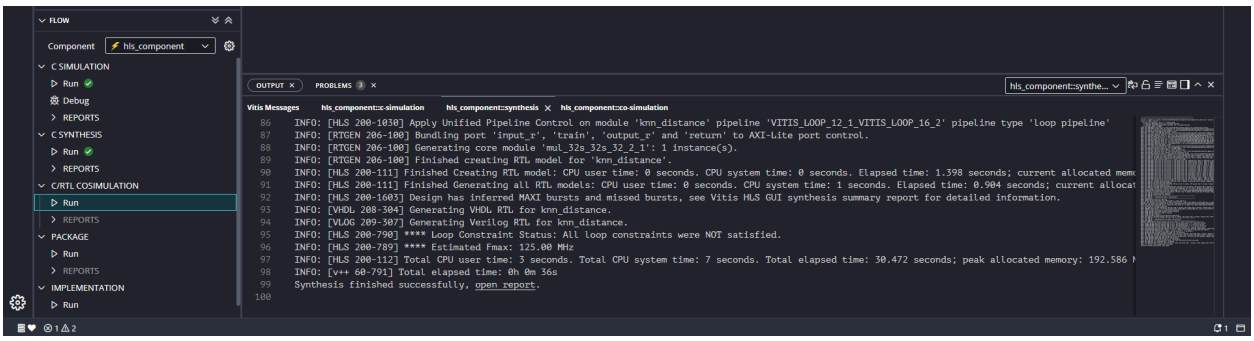


Step 10: HLS component will be created. Now, under the flow section in bottom-left, click on run under C simulation.

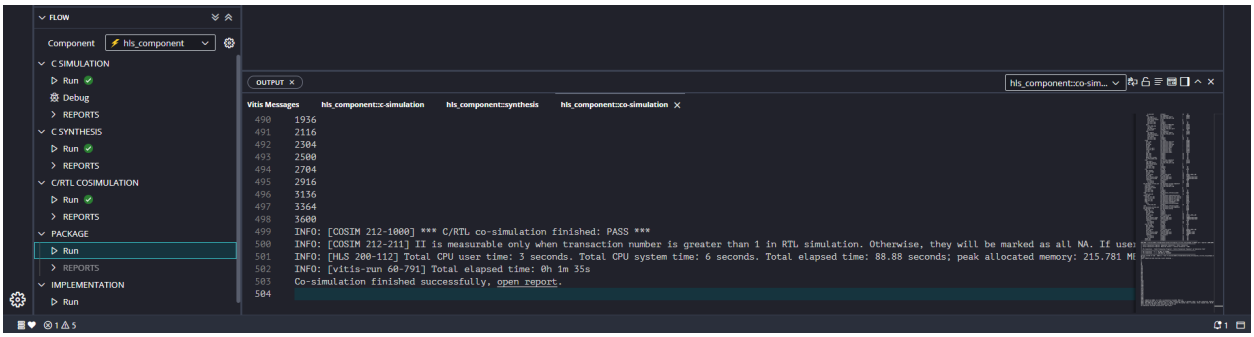


Step 11: Once your simulation is completed, a run option will appear under C synthesis. Click on run.

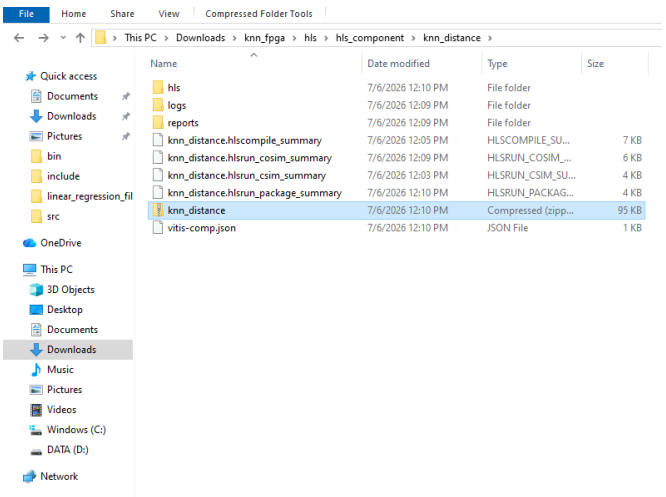
Step 12: Once your synthesis is completed, a run option will appear under C/RTL Cosimulation. Click on run.



Step 13: Once your cosimulation is completed, a run option will appear under Package. Click on run.



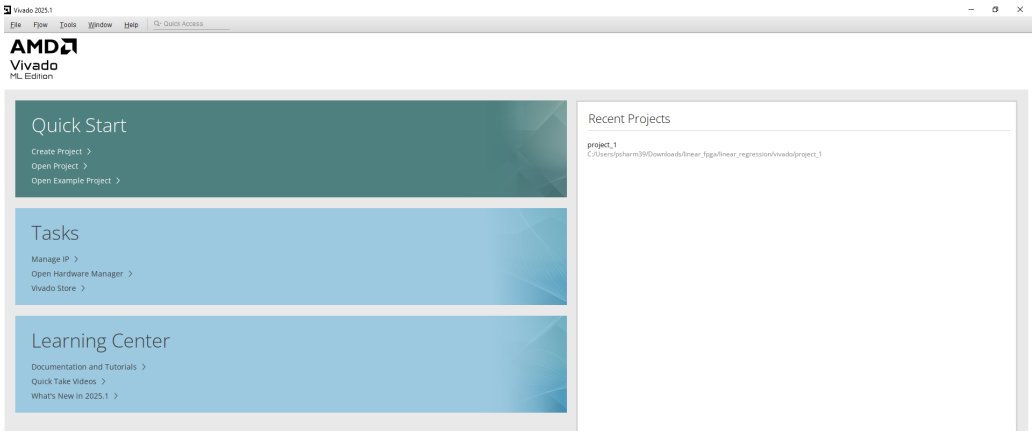
Step 14: Once your package is done, a zip folder will appear in your hls folder



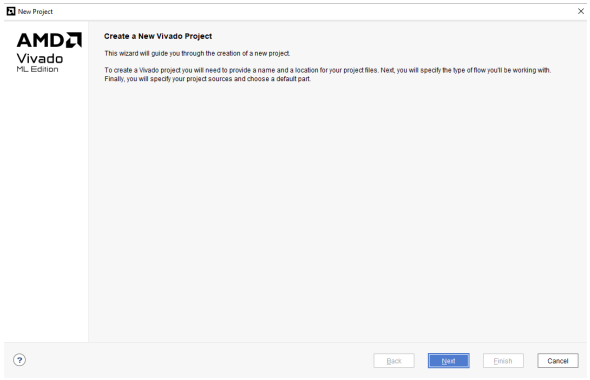
NOTE: This folder is an important aspect as this will be used to get the design in vivado.

VIVADO

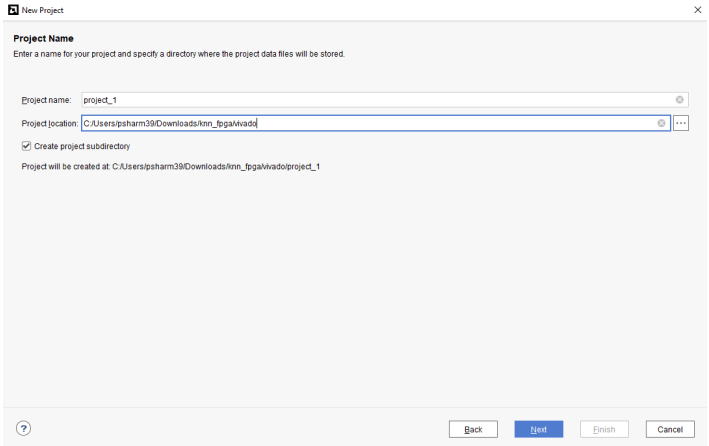
Step 1: Open Vivado and click on create project under Quick start.



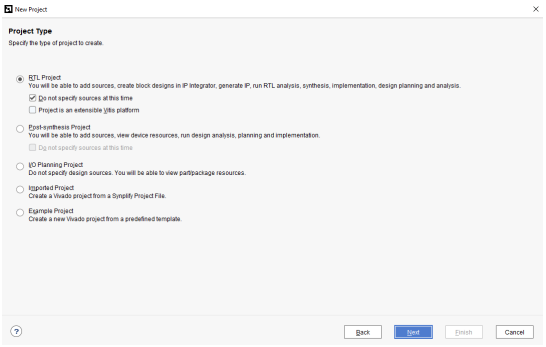
Step 2: A window will appear. Click on next.



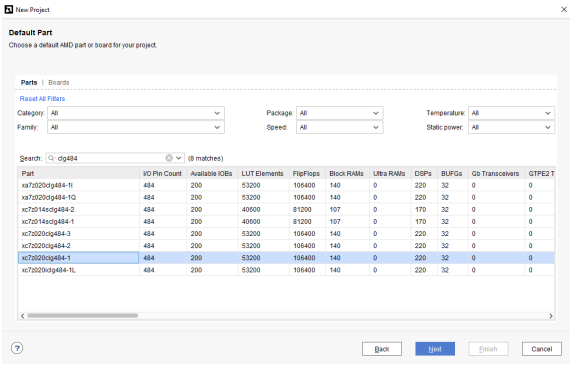
Step 3: Name your project and choose the Vivado subfolder in your main folder as your project location. Click on next.



Step 4: Choose your project type and click on next.

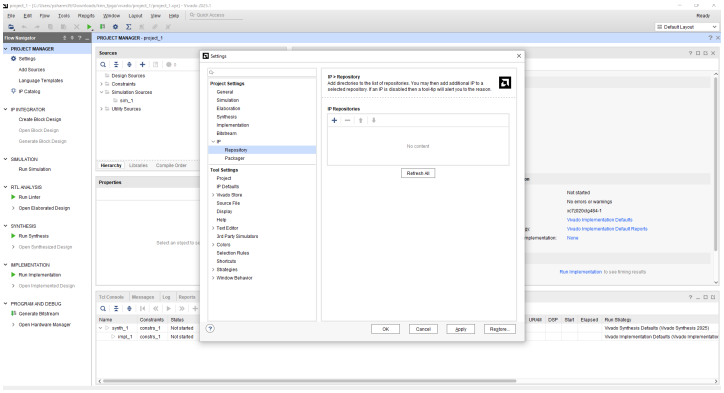


Step 5: Select the part/board you want to work on. Click on next.



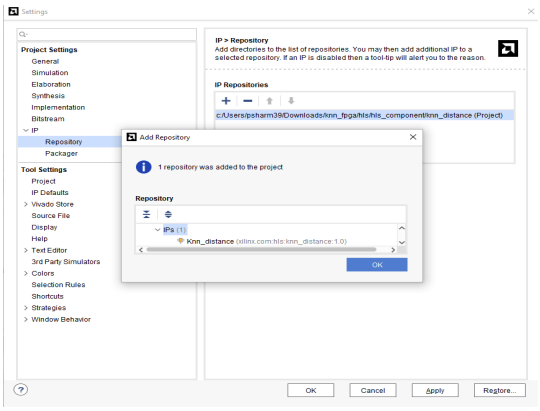
Step 6: A whole workspace will be created. Click on settings in left taskbar.

Step 7: Click on IP and then repository.

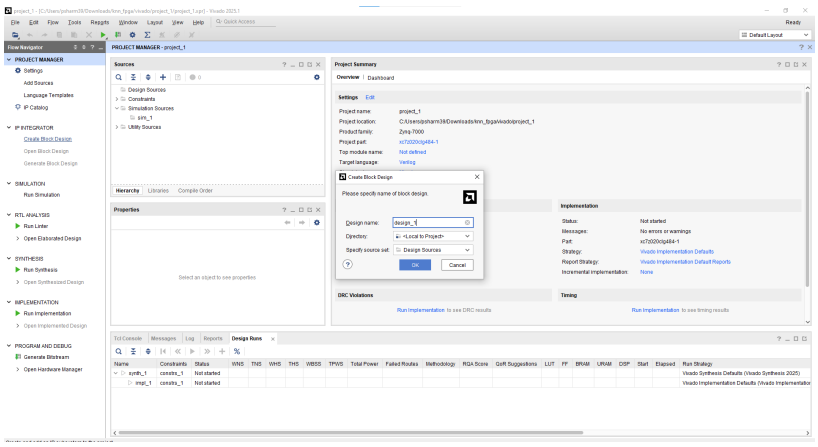


Step 8: Click on the + sign and select the folder which has your zipped folder you got from Vitis hls.

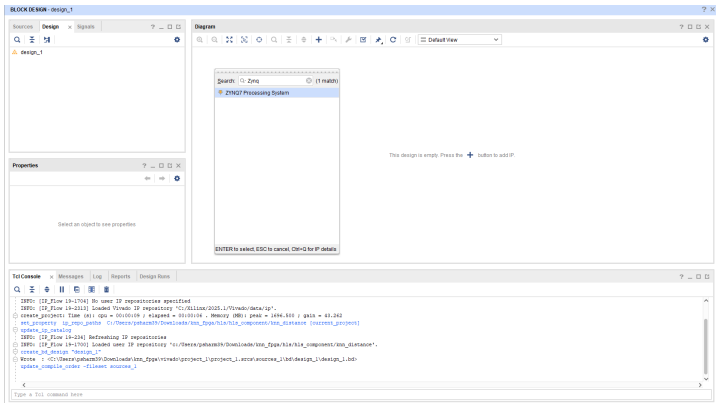
Step 9: Once you add the folder, you should be able to see a small window with your ip. Click on OK. Click on Apply option to apply it and click OK.



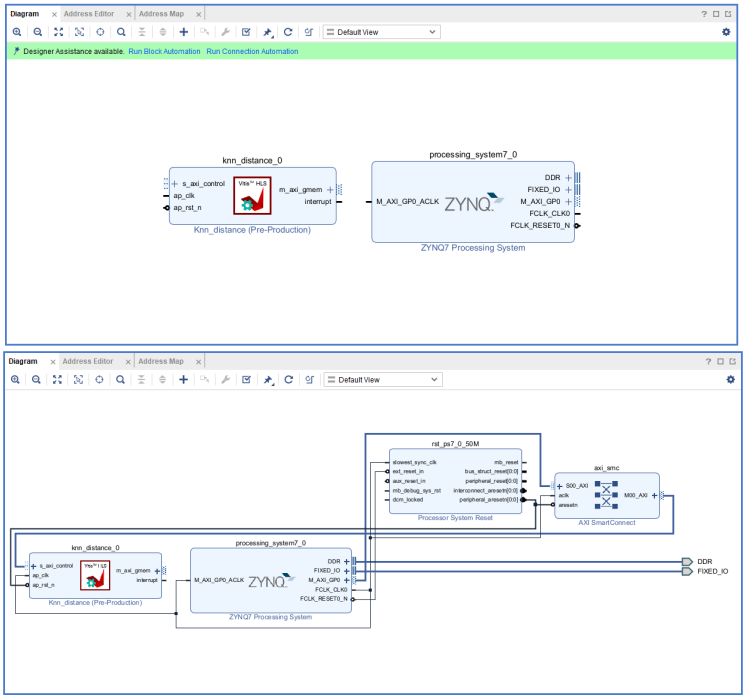
Step 10: Once you add your ip, click on create block design under IP integrator. A small box will appear. Provide a design name, directory and specify source. Click on OK.



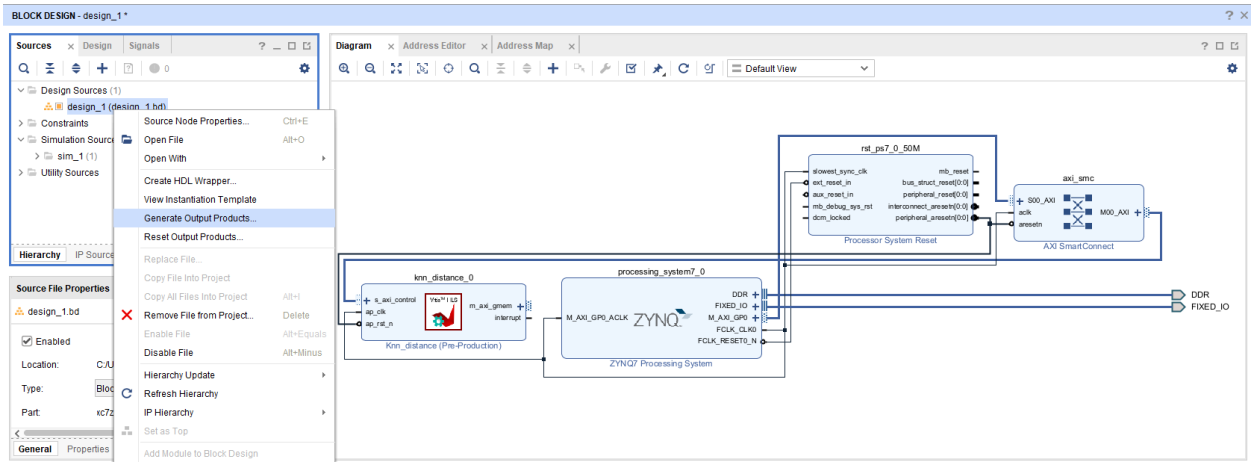
Step 11: An empty box will appear on right. Click on + sign and search for your main processing system/hardware and ip you created. Select those 2 to get both.



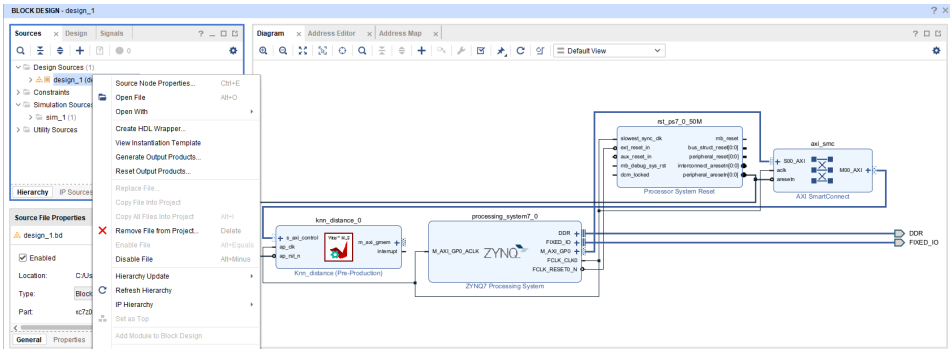
Step 12: Once you add your designs, you can see them in the empty space like below. Click on run block automation and then, click on run connection automation to get the complete design and wiring.



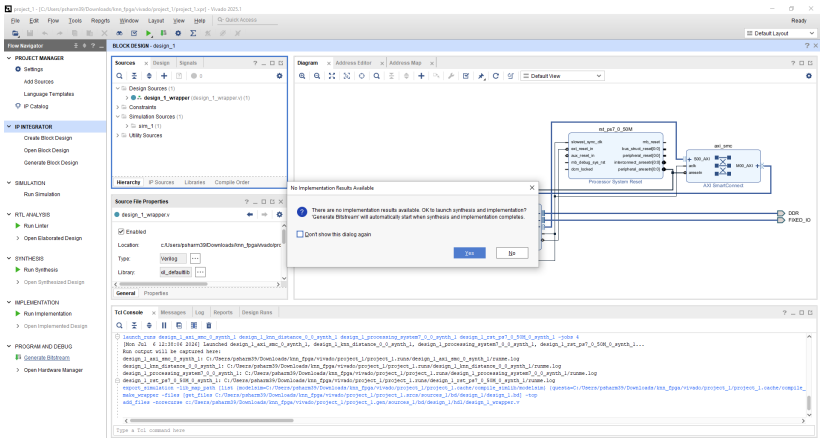
Step 13: Now, under sources in top left box, go to design sources and right click on your design. Select Generate output products.



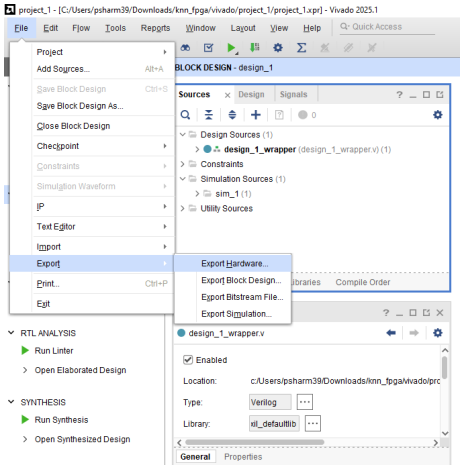
Step 14: Click on create HDL wrapper



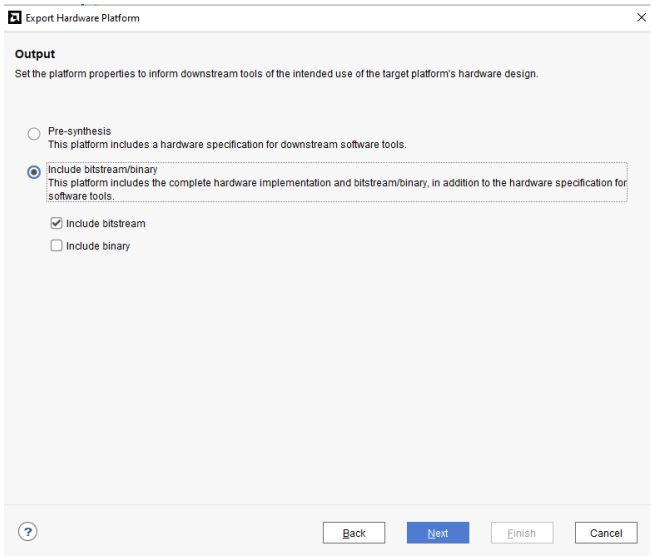
Step 15: Once you get the design wrapper, generate bitstream by clicking on generate bitstream option under program and debug option on left taskbar.



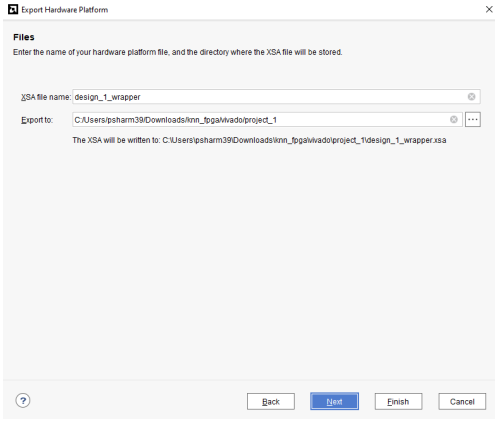
Step 16: After generating bitstream, go to file and export hardware.



Step 17: Select Include bitstream/binary and select Include bitstream option. Click on next.



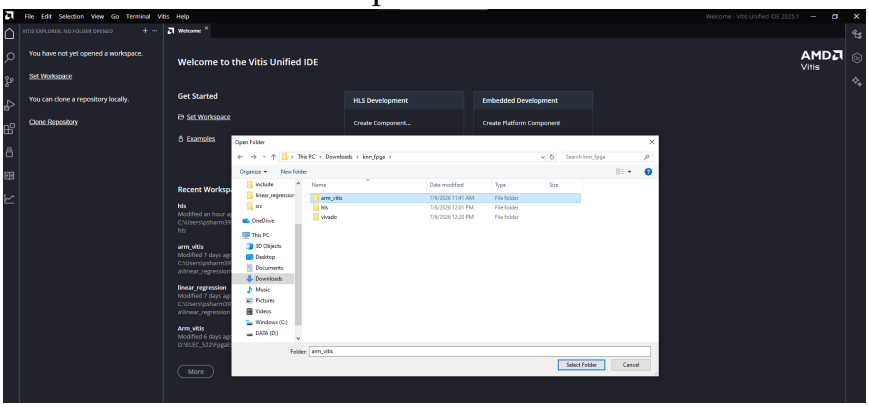
Step 18: Enter a file name and select your Vivado subfolder as the export location. Click on next and finish.



This will give you a XSA file in your Vivado subfolder which will be used in Vitis again.

VITIS

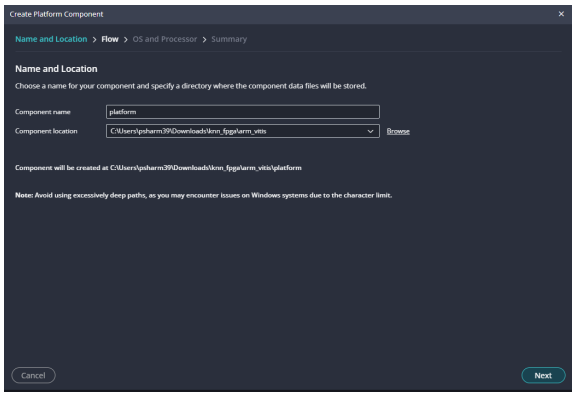
Step 1: Go to Vitis again and click on set workspace. Select your vitis folder as the workspace this time.



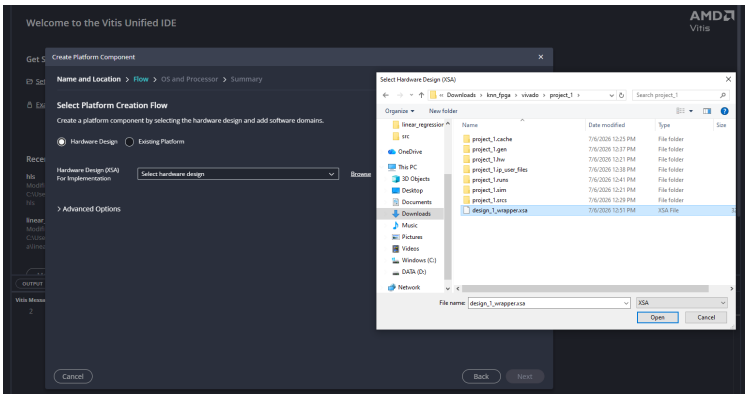
NOTE: This workspace will be different from the hls workspace.

Step 2: Click on create platform component on left taskbar

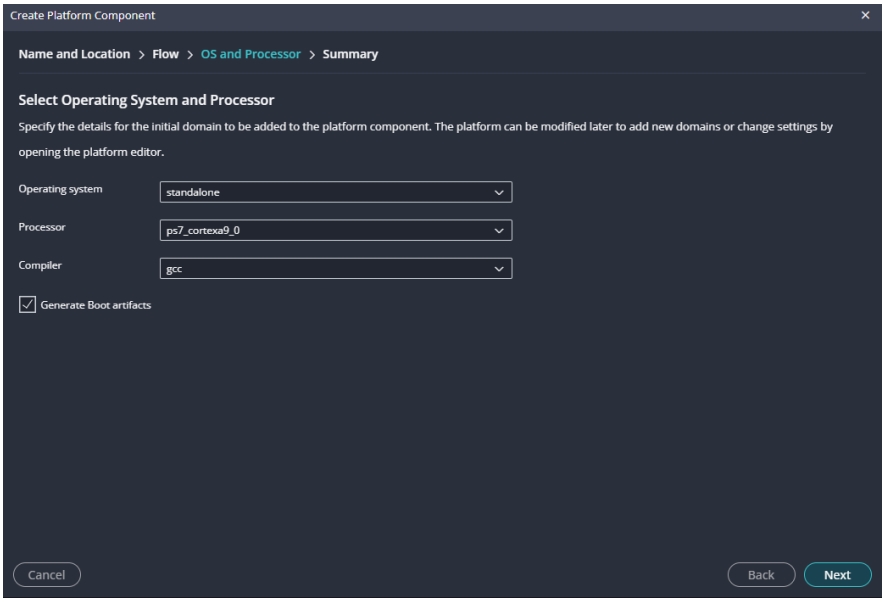
Step 3: A window will appear. Give your platform name and location. Click on next.



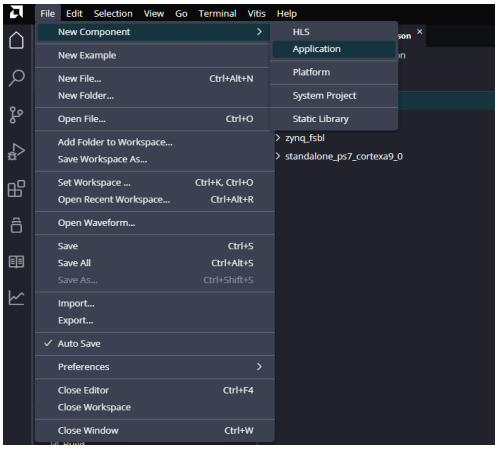
Step 4: Click on hardware design option. Click on browse and select your XSA file under Vivado subfolder. Click on next.



Step 5: Get your operating system and processor. Click on next.

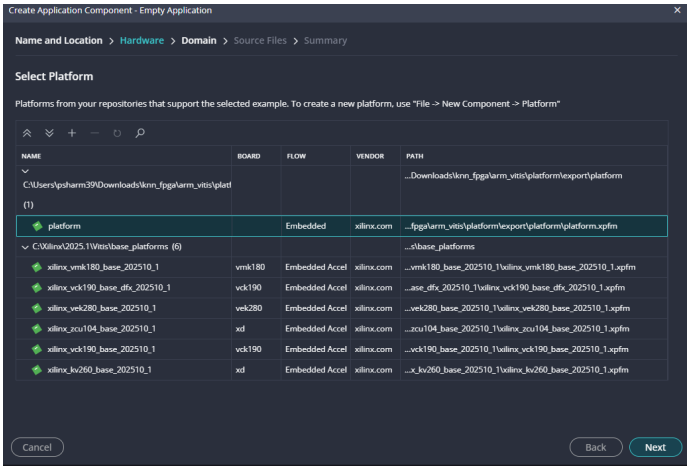


Step 6: Once your platform is created, go to file, new component, application.

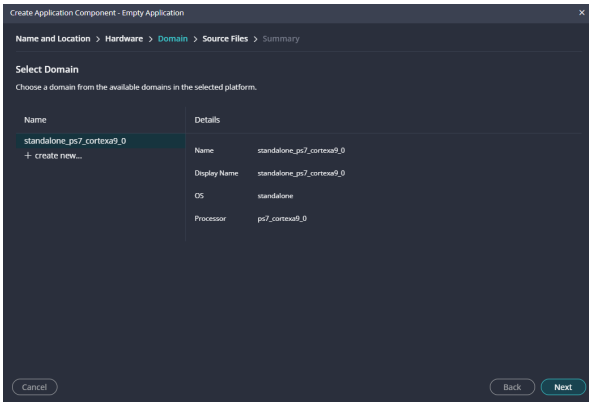


Step 7: A window will appear. Give your app component name and location. Click on next.

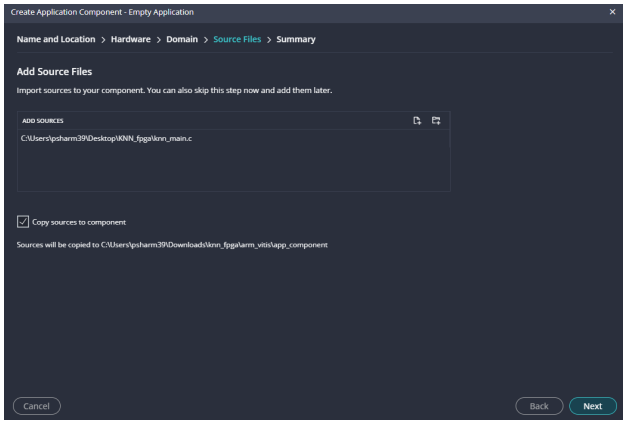
Step 8: Choose the platform you just created. Click on next.



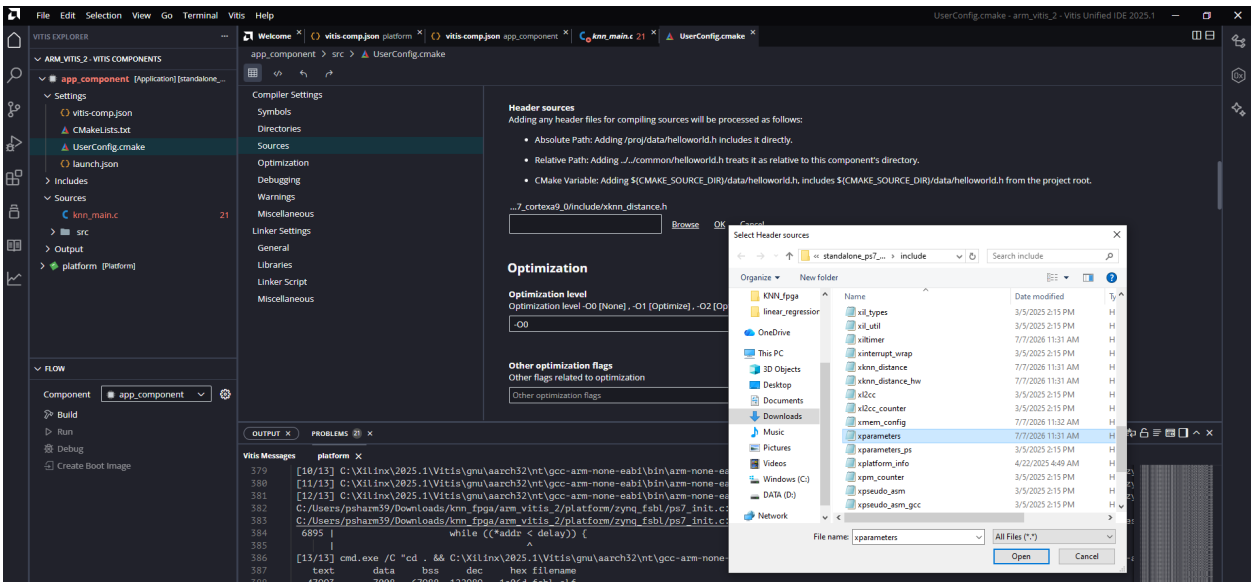
Step 9: Select Domain



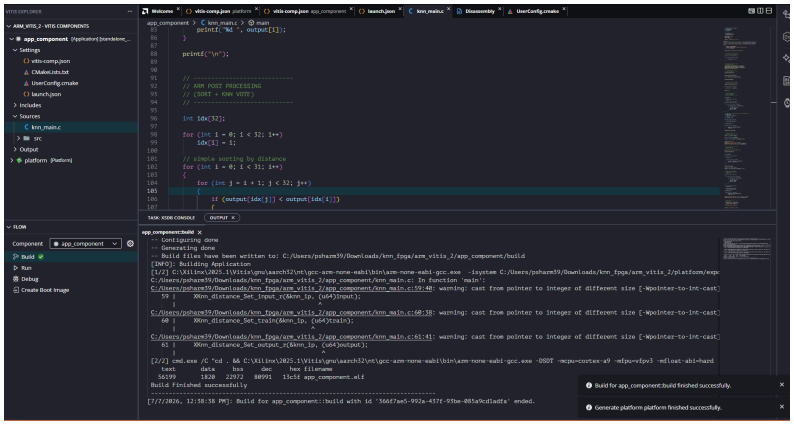
Step 10: Add source file (the file which has your main code)



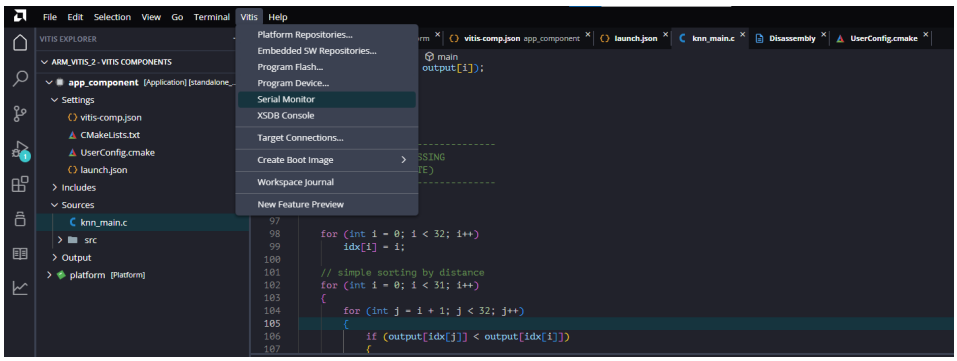
Step 11: Add header file by clicking on UserConfig.cmake and then sources. Click on Add here option under header sources and add your header file.



Step 12: Click on build option under flow section in bottom left section.

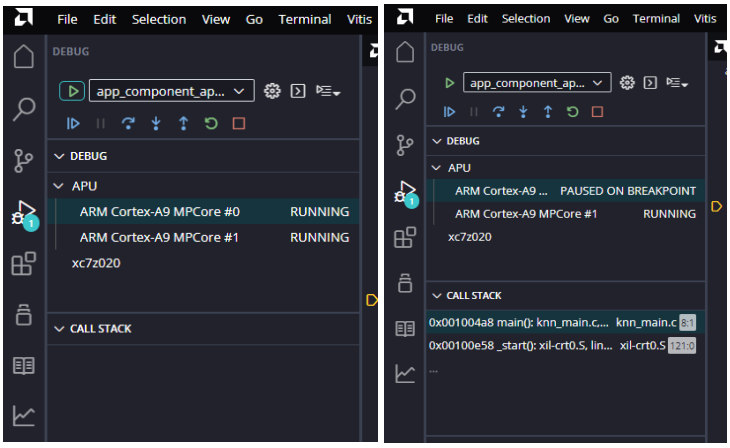


Step 13: After build, go to Vitis option in top taskbar and select serial monitor. Select your com port and baud rate.



Step 14: Click on run option under flow.

Step 15: Go to debug on left taskbar and click on continue (the blue button with play sign and a | before play sign). If you don't see anything under call stack like in first photo then click on stop and run it again.



Step 16: Once you click on continue, you should be able to see your output.