TESTBENCHES: like function calls with parameters. U TEST ur code’s functionality by giving some ip vals. These are for testing the behaviour of code BEFORE SYNTHESIS, THIS IS FOR SIMULATION PURPOSE.

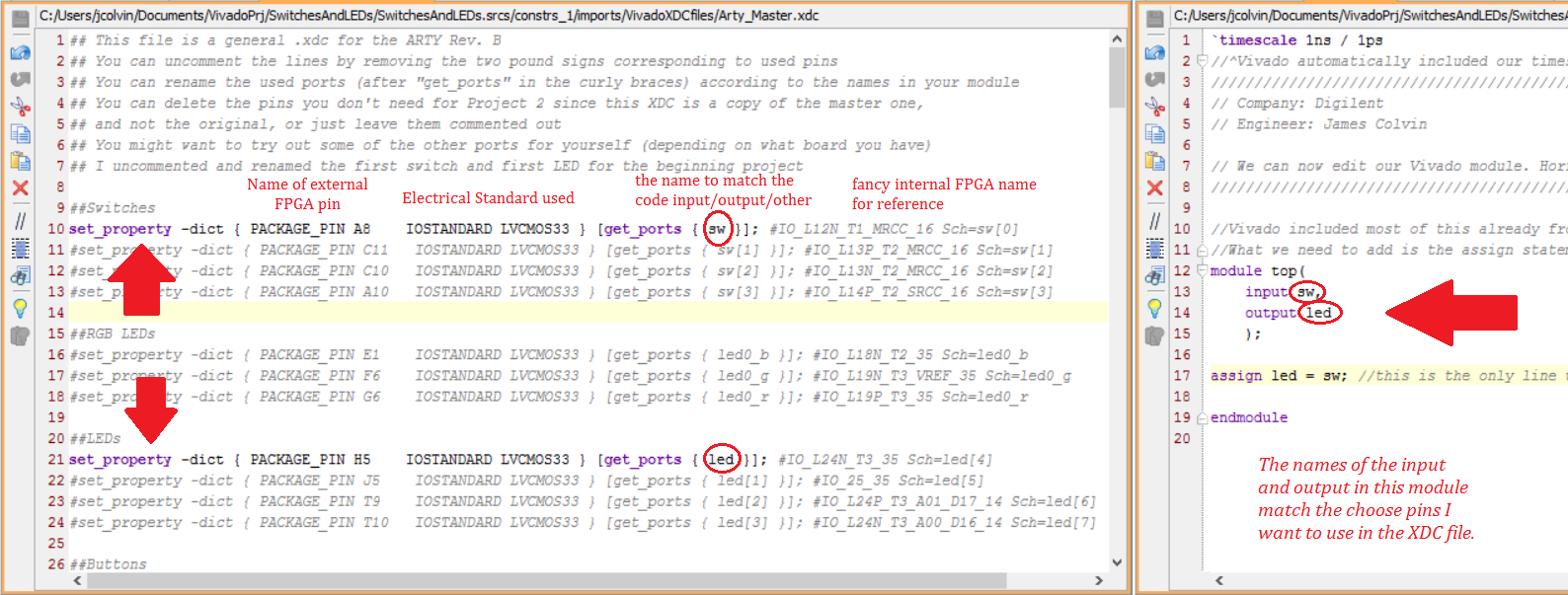
Actual code files: Function body definition

AFTER SYNTHESIS:

When programming the board with the design, u don’t need testbench…because these ip values for fun calls are going to be provided by the ip pins on the boolean board.

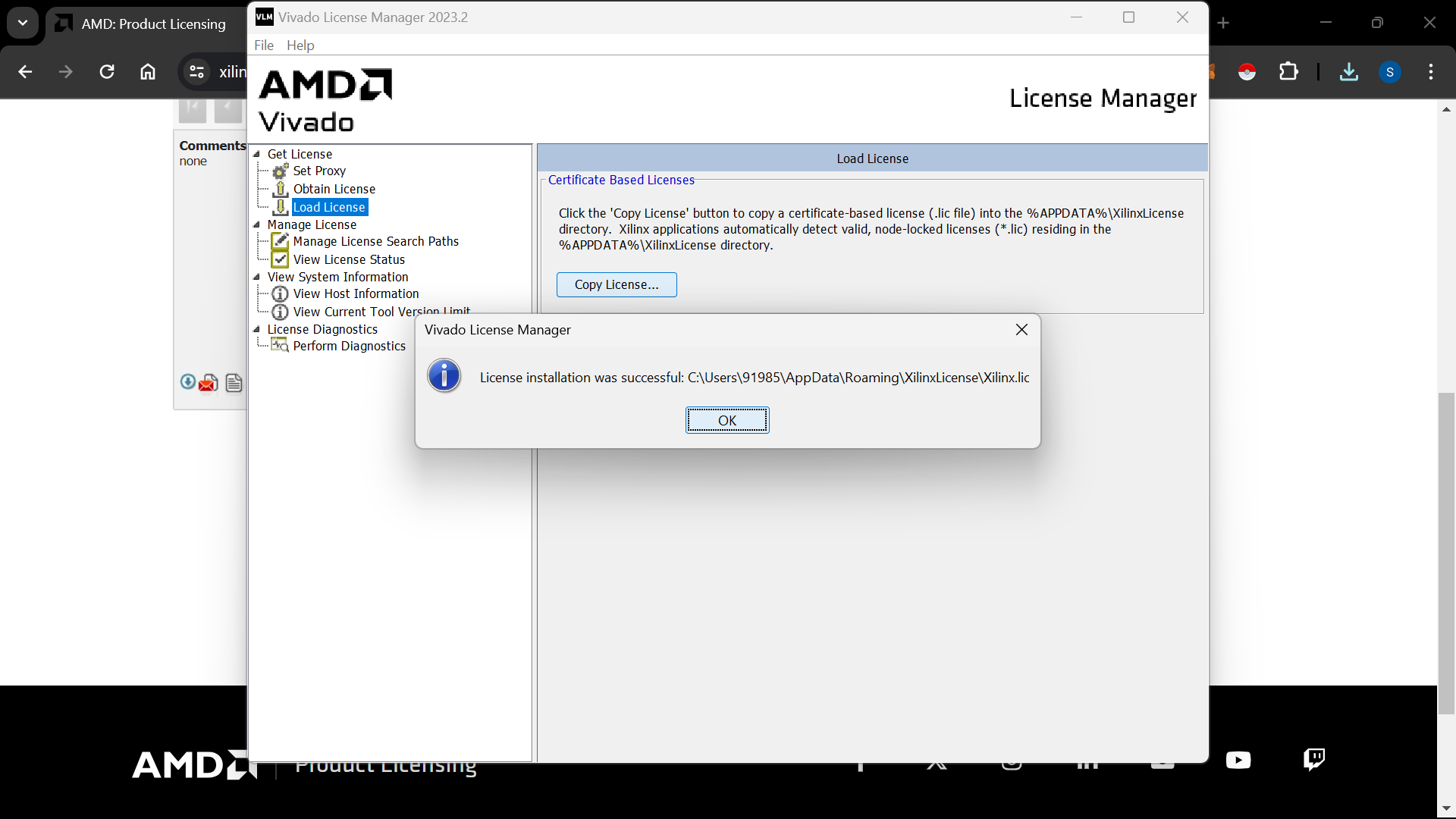
CONSTRAINTS file helps to define, what pins are going to be used. Eg. Led no. So and so, etc.

the only portion of the XDC file you will change is the name of the external pin after get\_ports that is surrounded by curly braces {} to match the input/output that you created in your main module code



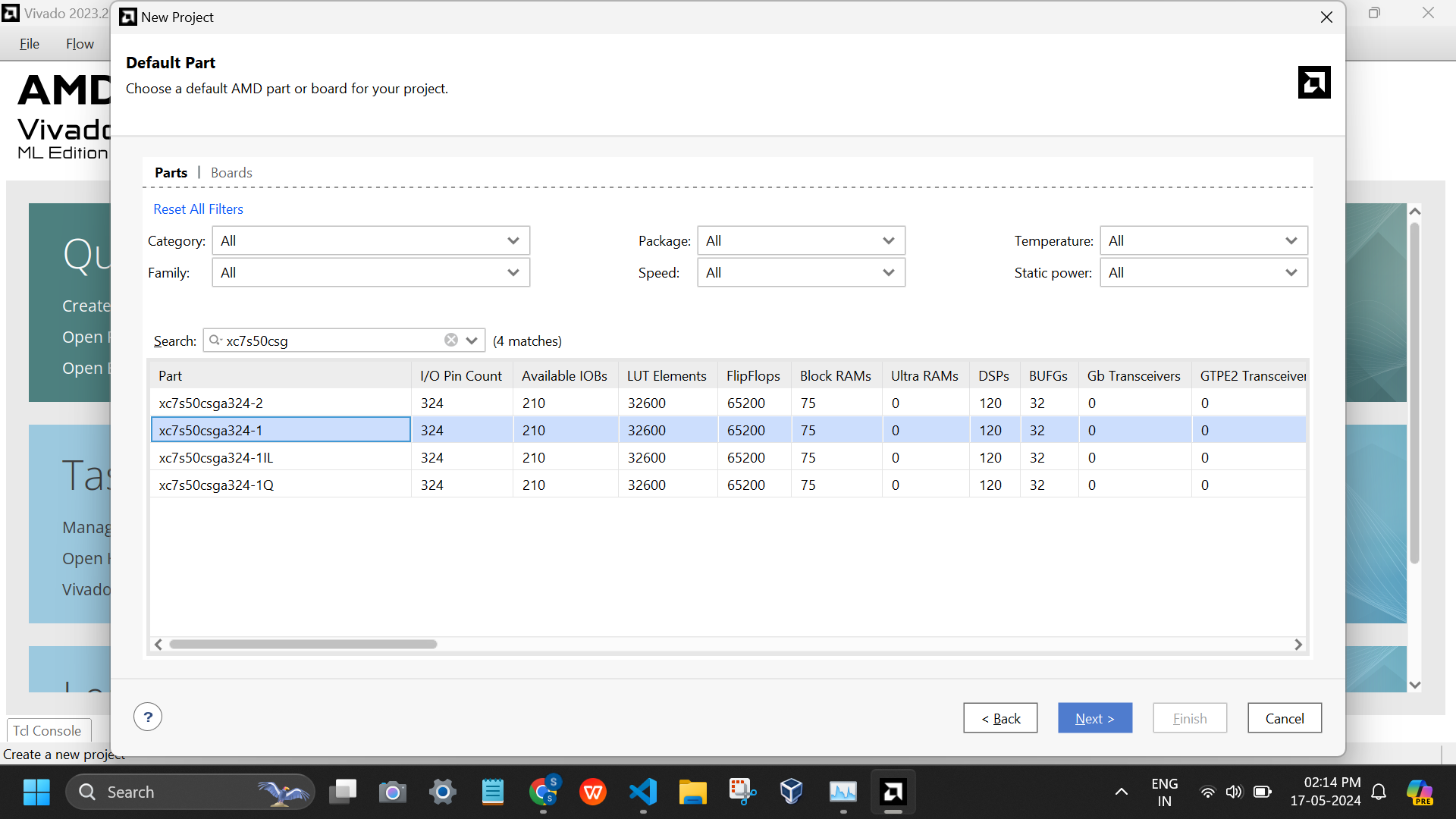
Install vivado from: <https://www.xilinx.com/support/download.html>

Refer to <https://youtu.be/fBFn32Al0yw>



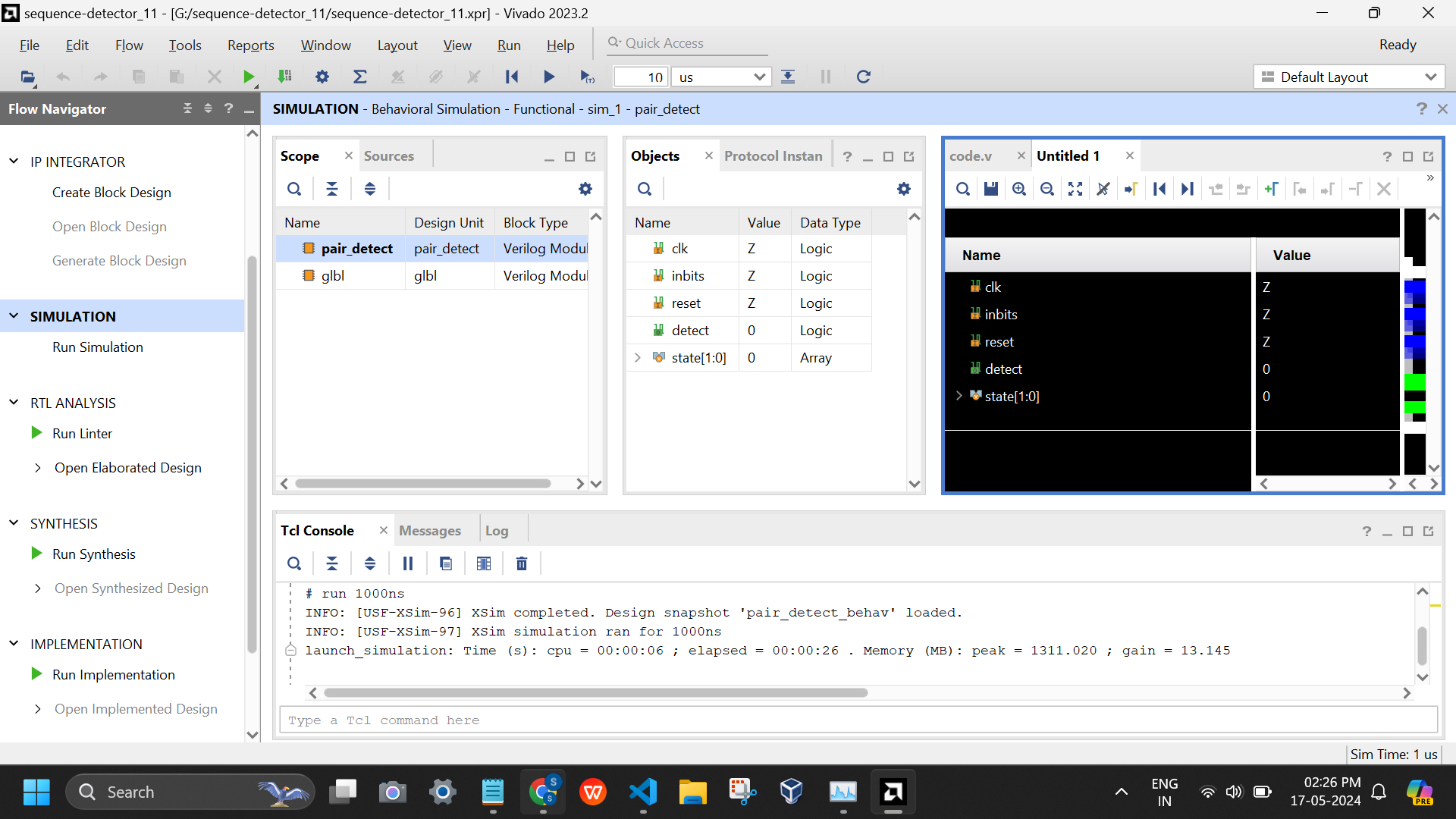
Open vivado,

Make a new prj

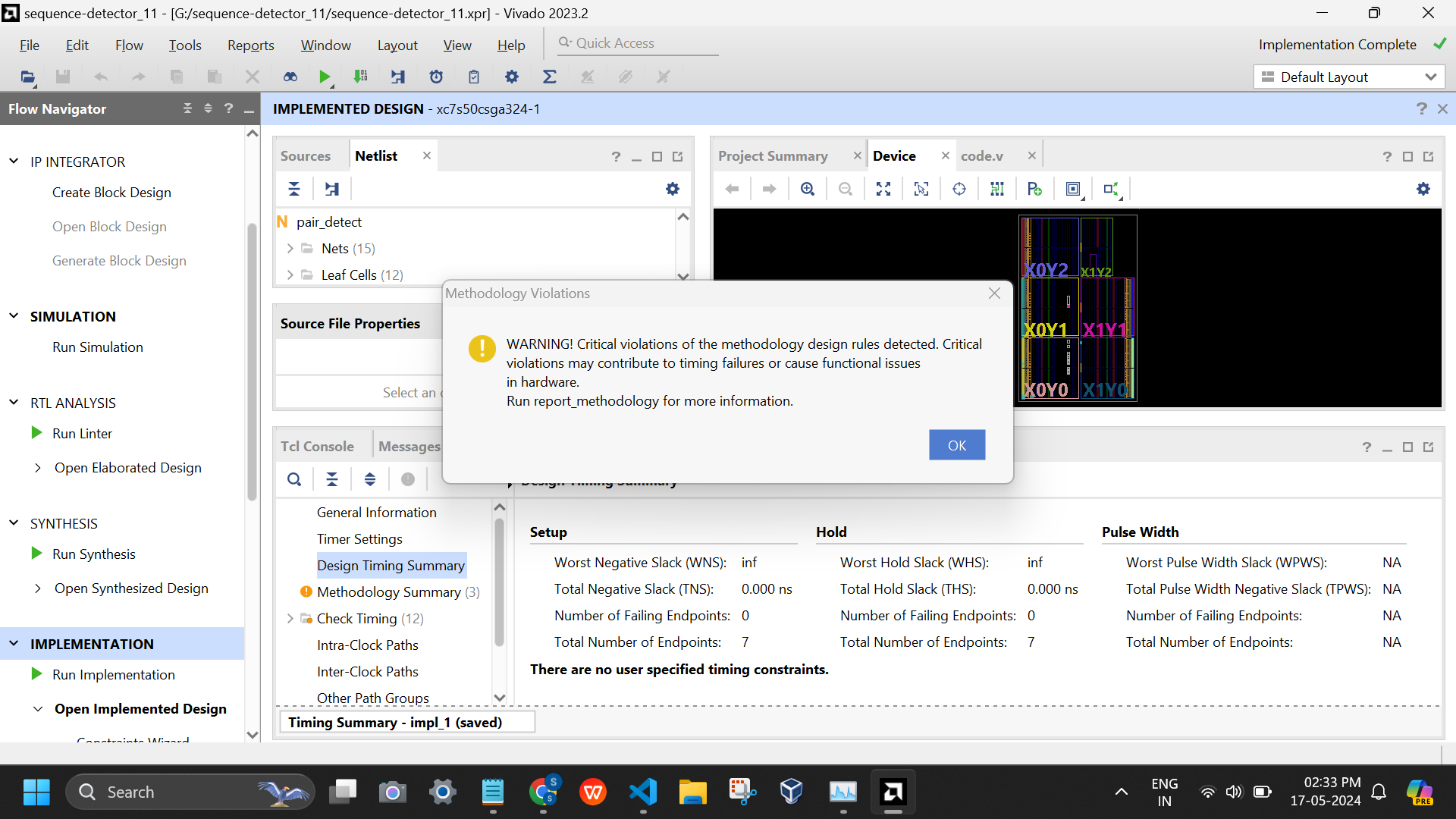


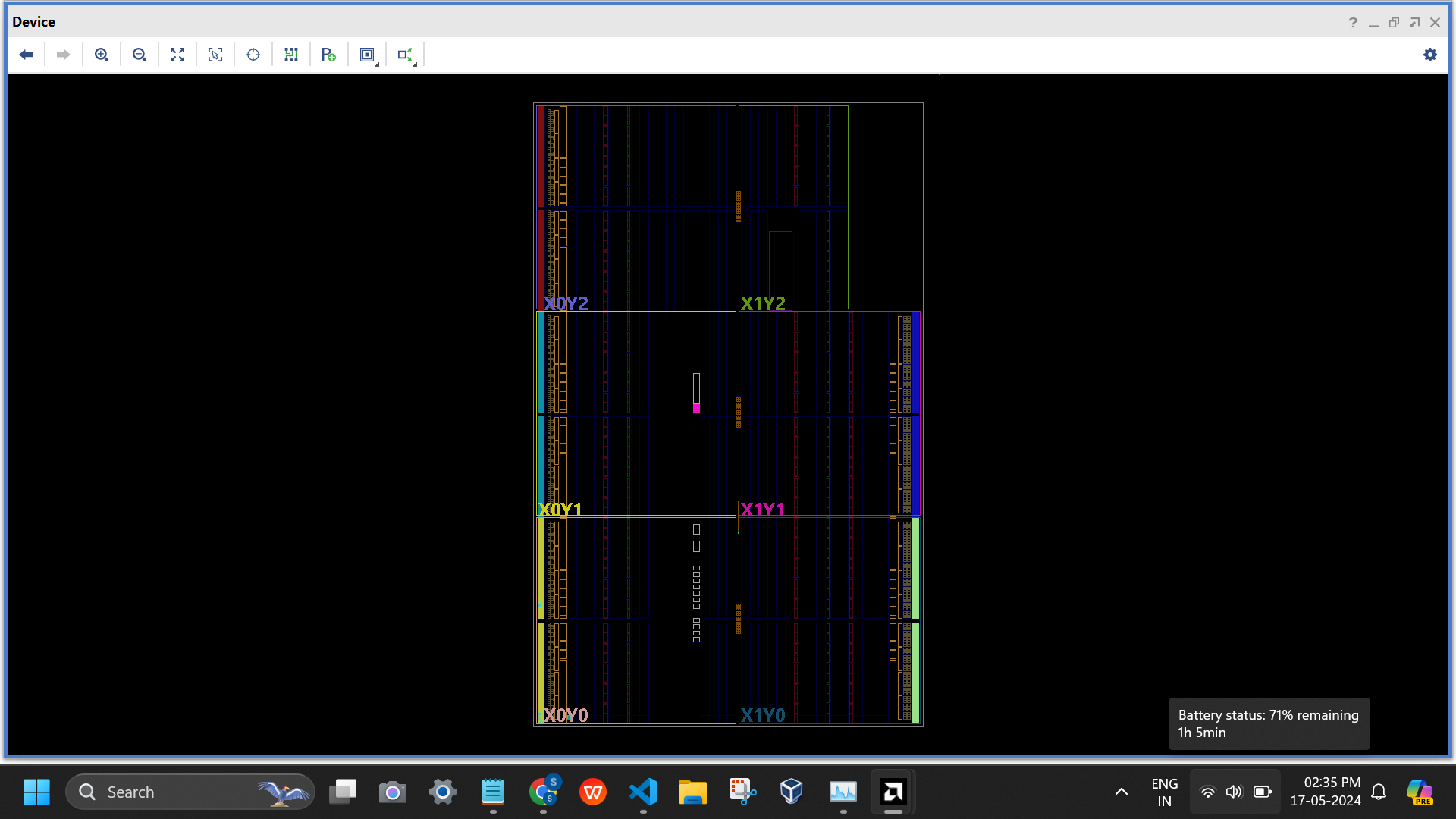
Paste the code and tb

Run simulation

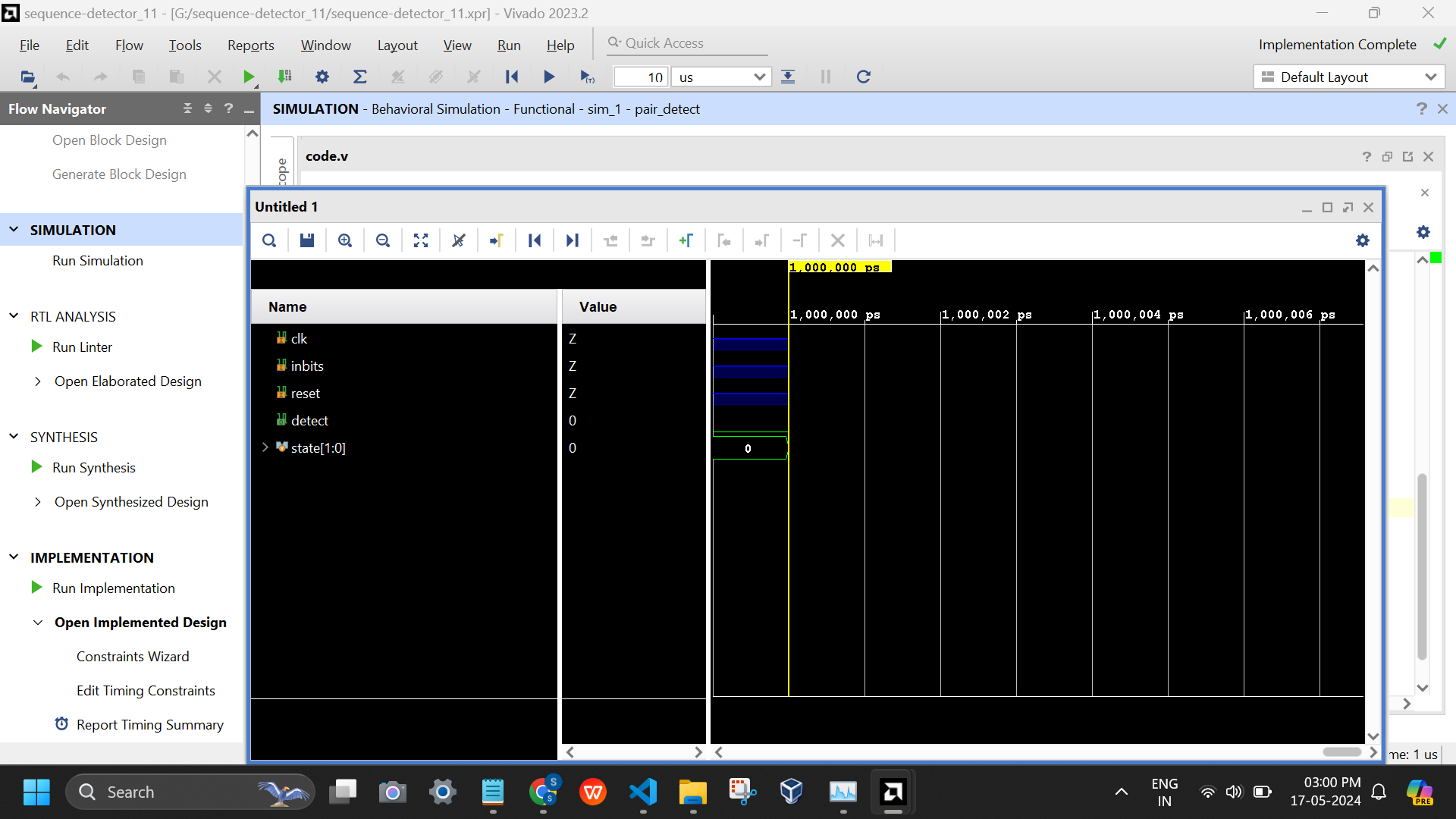


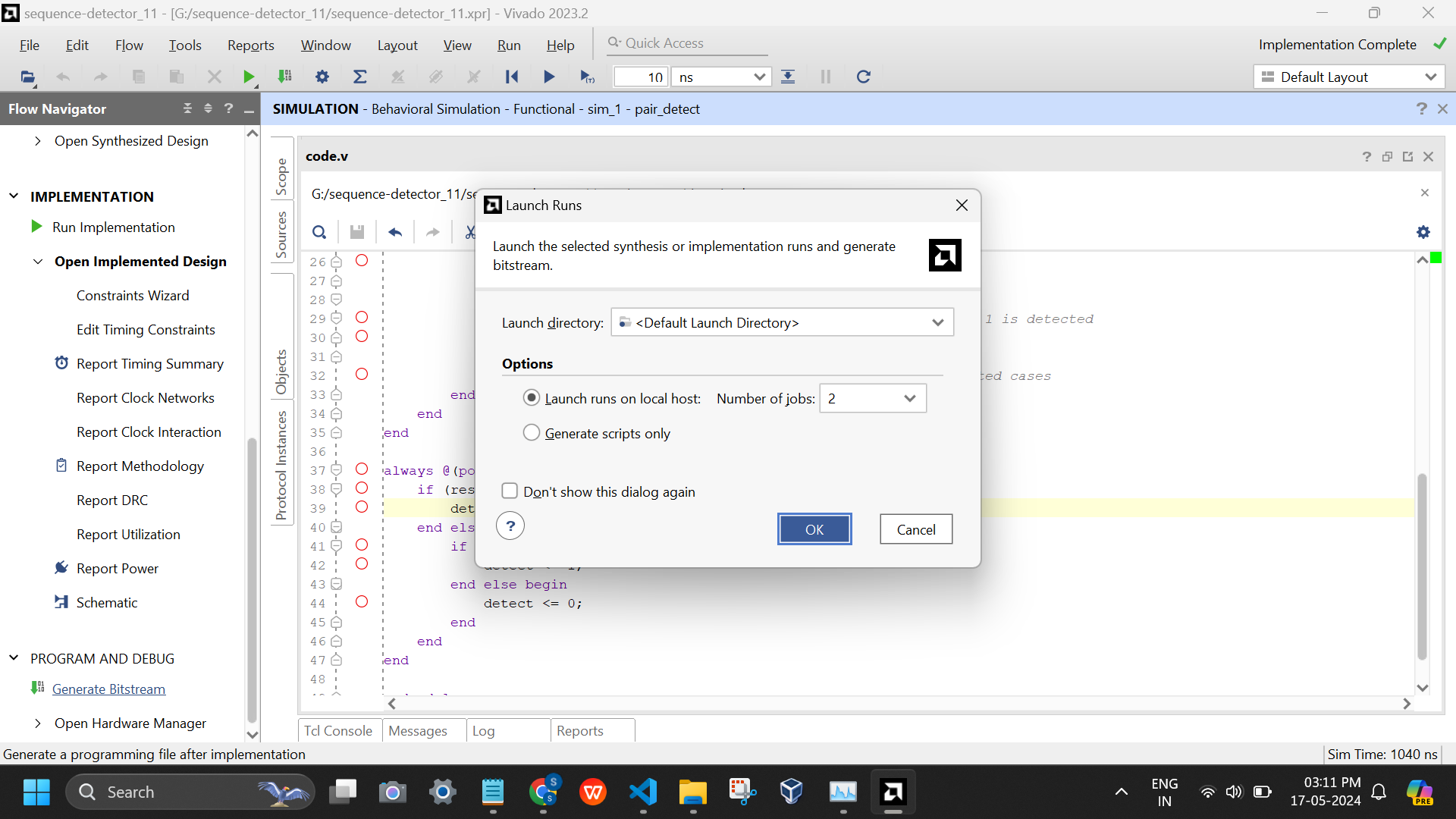
Implemented design





See timing dgms



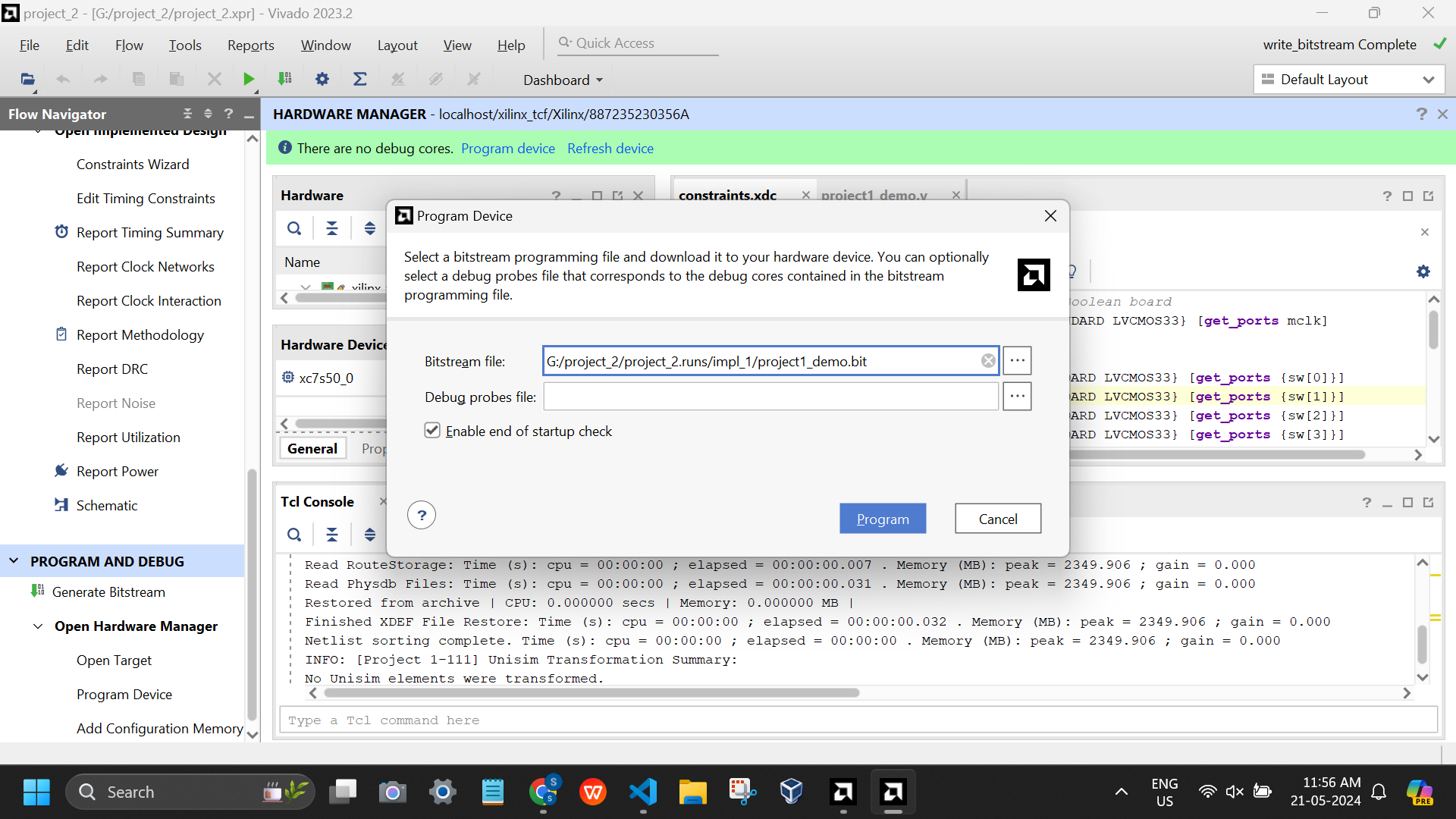


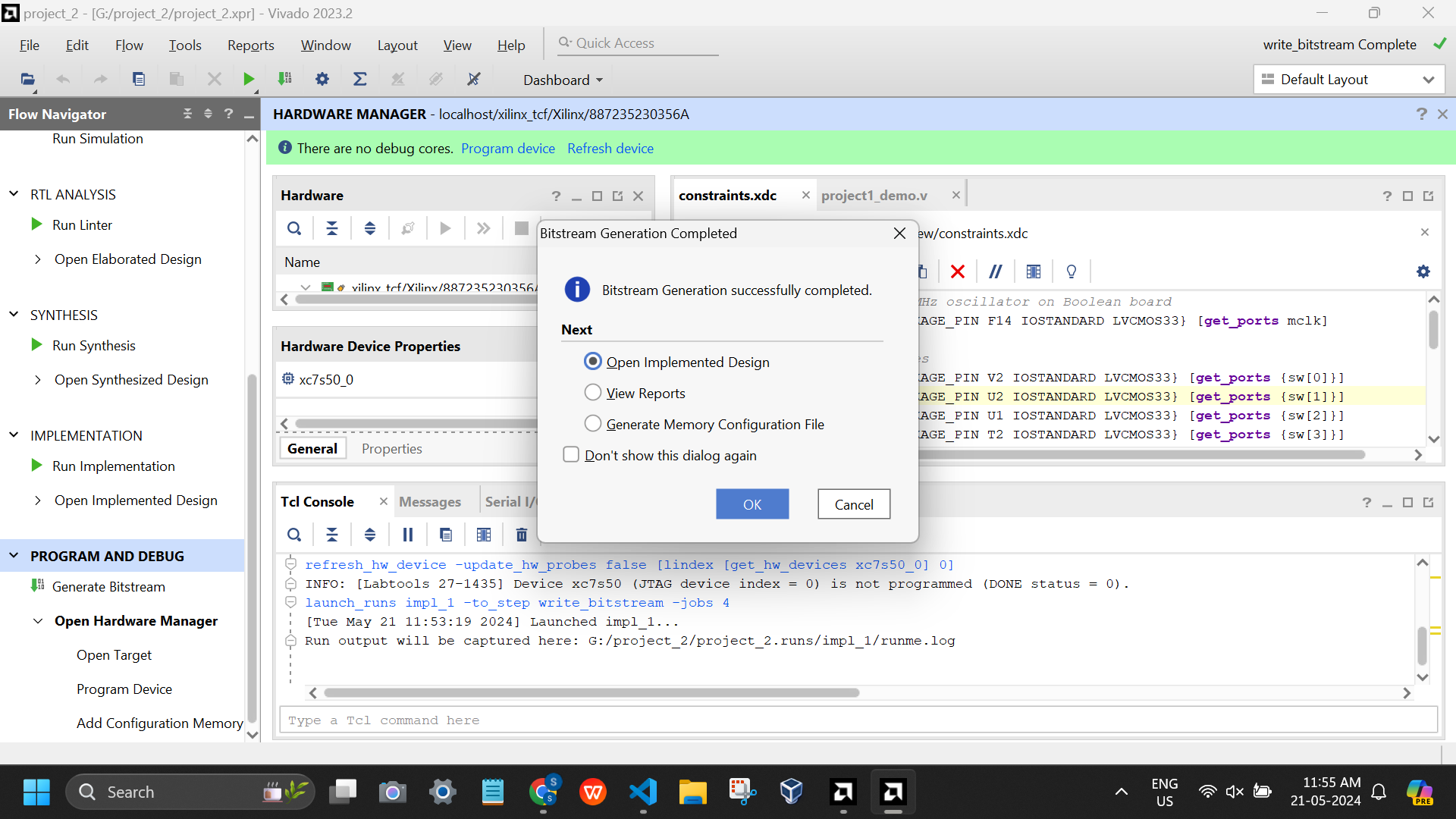
AFTER CODE IS VERIFIED,

Making constraints file .xdc

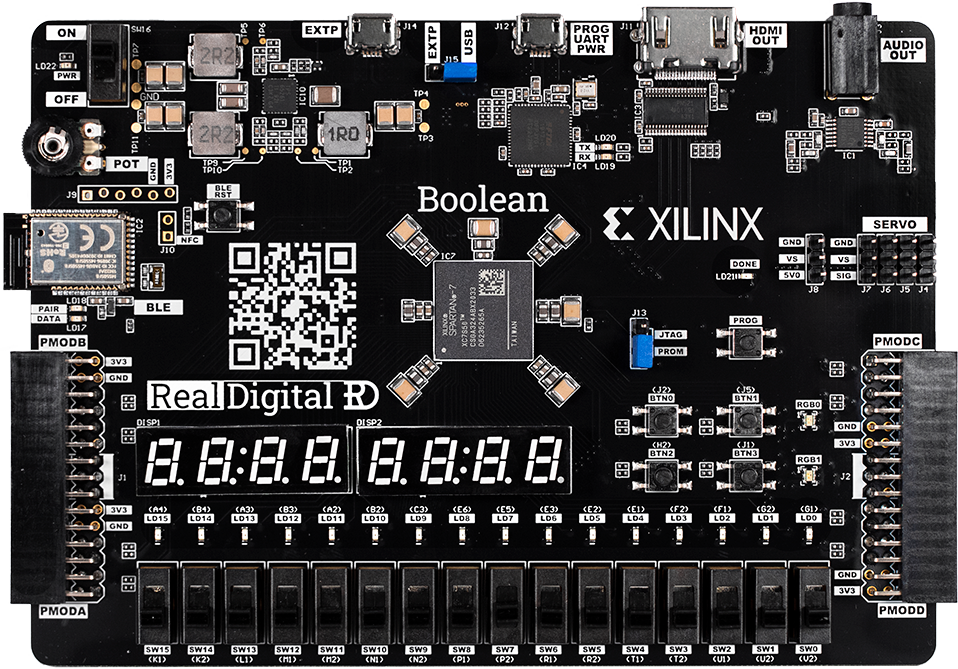
Run synthesis

Connect the board and generate bitstream





Test by giving ips to the boolean board:



If in an ide like vscode: simulate (verilog extension)

Run using cmds:

iverilog -o alu-op alu.vl alu\_tb.vl

vvp alu-op

(alu.vl is code file, alu\_tb.vl being testbench file, alu-op is the name of op file that gets generated after running this cmd)

PROJECT:

1. Making an ALSU (Arithmetic logic and shift unit)

( project\_1 has alu.v and alu.xdc)

Menu driven opcode to operation mapping operations for alsu is done in code. Similar to assembly level language where theres an opcode for every opn (indirect direct instructions assignments etc)

1. Basics project

(project\_2 has a project1\_demo.v file with a constraints.xdc file)

Ref: <https://www.realdigital.org/doc/c4ceeb20d229e5f3d4e32f3a74e343e9>

Follow this and copy paste all the files as they are.

(or download the prj folder as .xpr file is provided in my github to directly run it in vivado.)