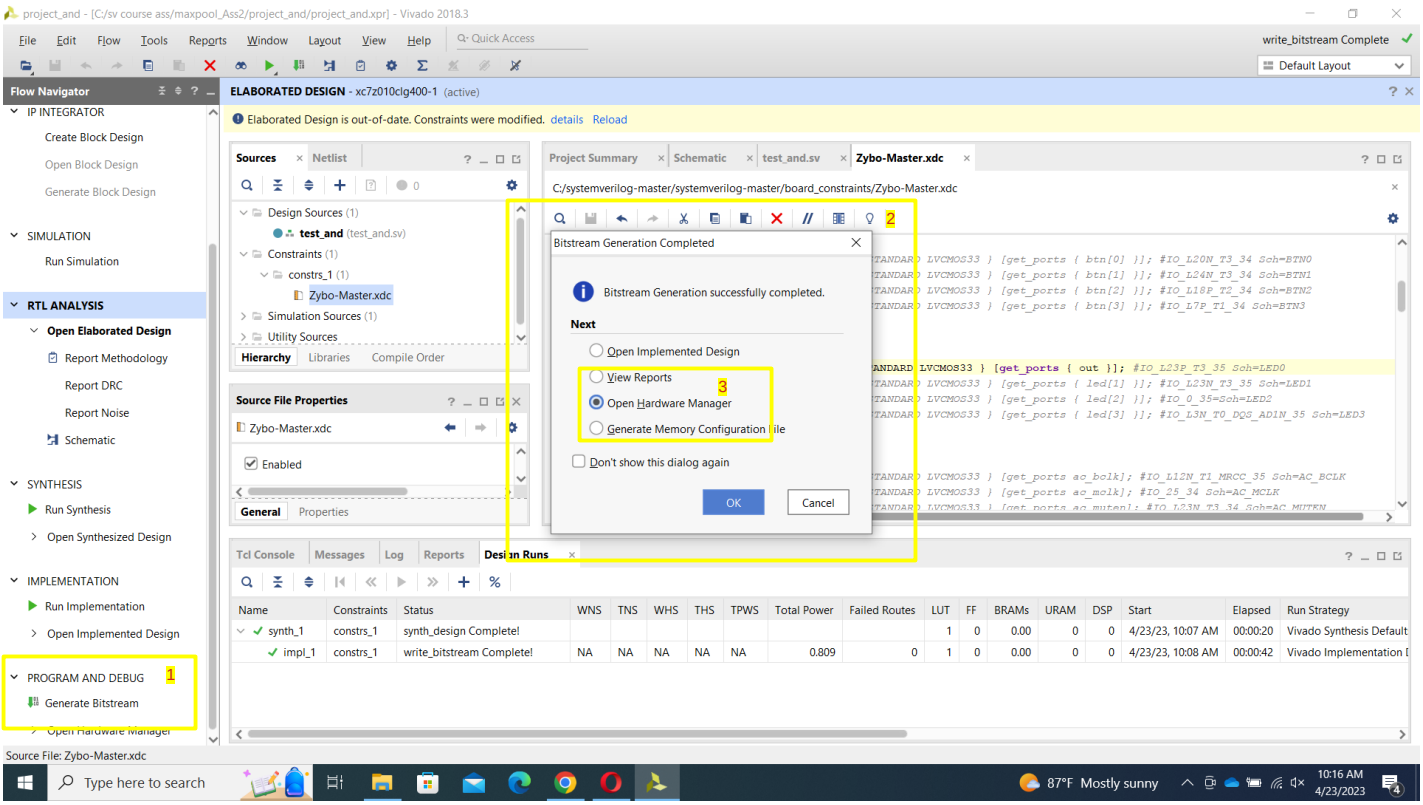
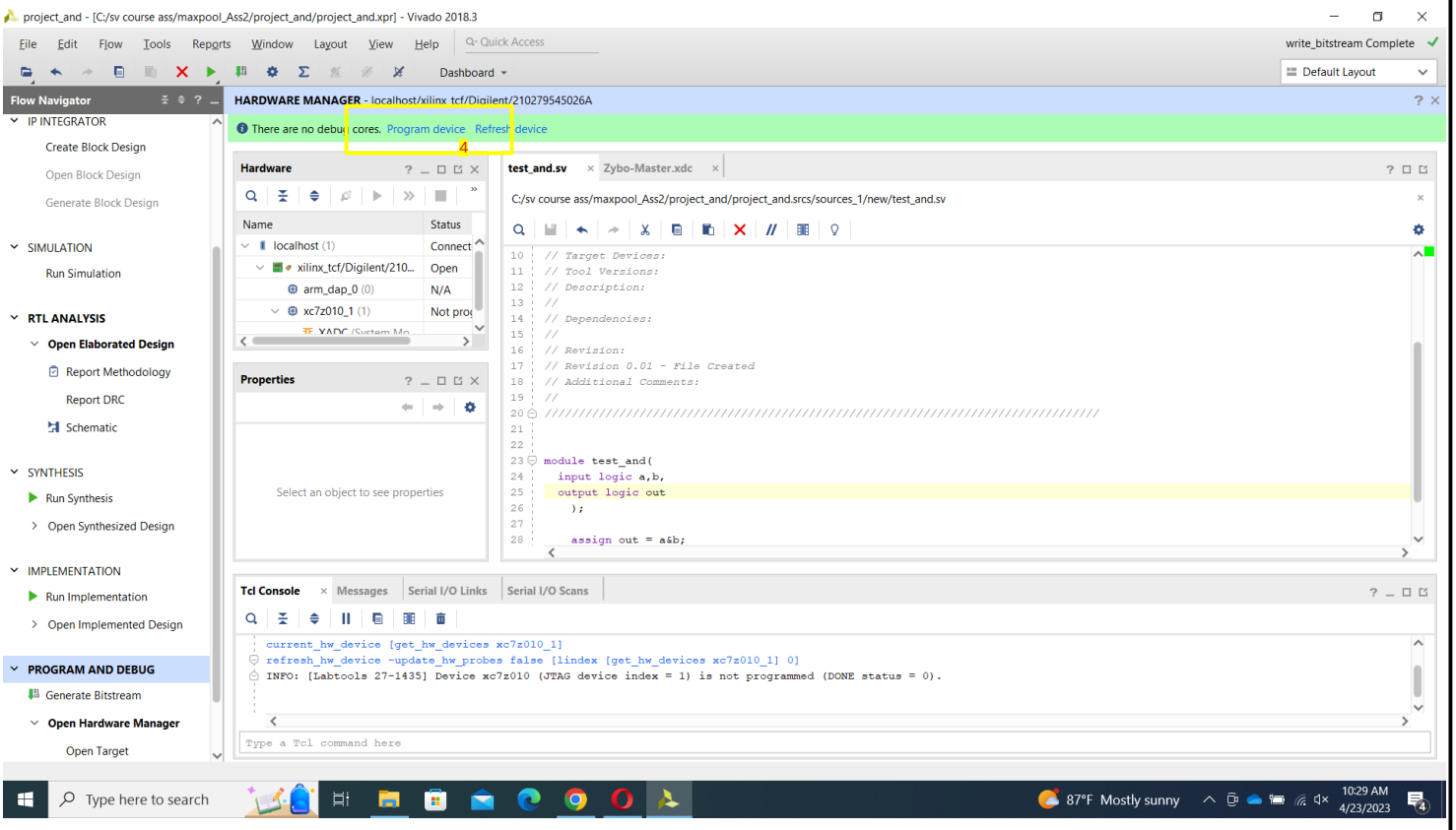
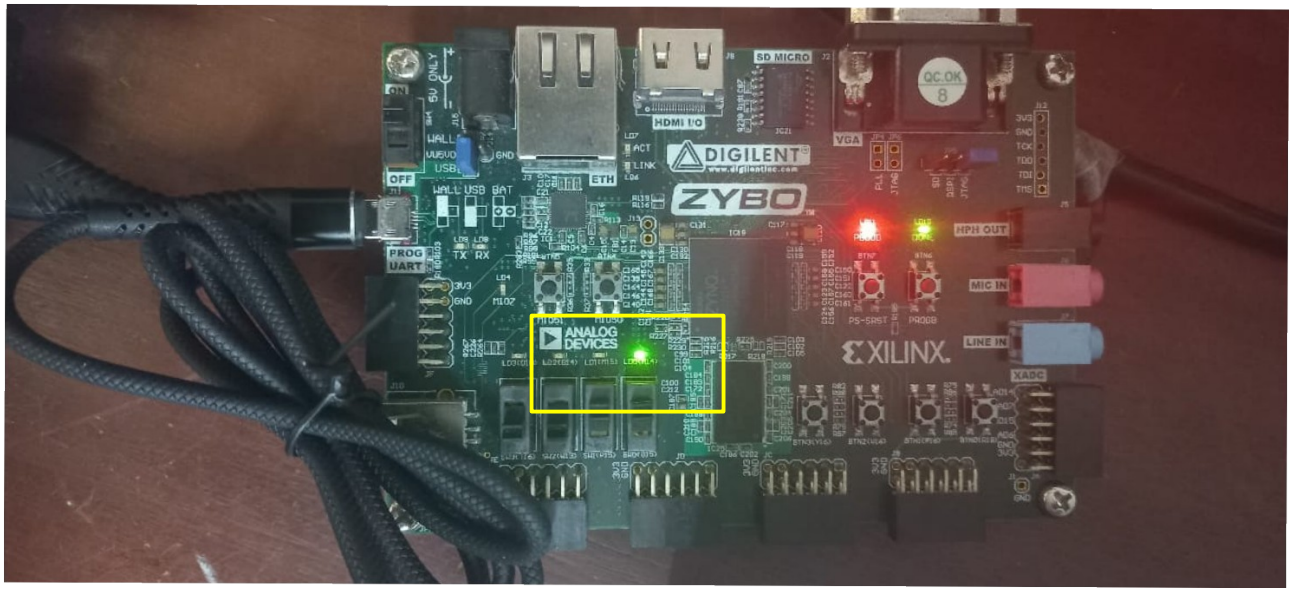
And gate


Bitstream Generation Completed
Bitstream Generation successfully completed.
Next
Open Implemented Design
View Reports
Open Hardware Manager
Generate Memory Configuration File
Don't show this dialog again
OK
Cancel
1
2
3


HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210279545026A
Program device
Refresh device
4
// Target Devices:
// Tool Versions:
// Description:
// Dependencies:
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
module test_and(
input logic a,b,
output logic out
);
assign out = a&b;
current_hw_device [get_hw_devices xc7z010_1]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7z010_1] 0]
INFO: [Labtools 27-1435] Device xc7z010 (JTAG device index = 1) is not programmed (DONE status = 0).

Showing the input high on the FPGA only when the both a and b pins are high in



step-by-step process for running RTL code on a Zybo board using Vivado:

1.Open Vivado and create a new project by clicking on "File" -> "New Project". Follow the prompts to select the project name, location, and target device. In this case, select the Zybo board as the target device.

2.Add your RTL code to the project by clicking on "File" -> "Add Sources". Select your RTL code files and add them to the project.

3.Create a new block design by clicking on "Create Block Design" under "Flow Navigator". Name the block design and add the Zybo board as a part.

4.Add your RTL code to the block design by clicking on "Add IP" in the block design diagram. Select "Add or create design sources" and add your RTL code files to the block design.

5.Generate the HDL wrapper by clicking on "Generate Output Products" in the block design diagram. Select "Generate RTL Sources" and "Create HDL Wrapper".

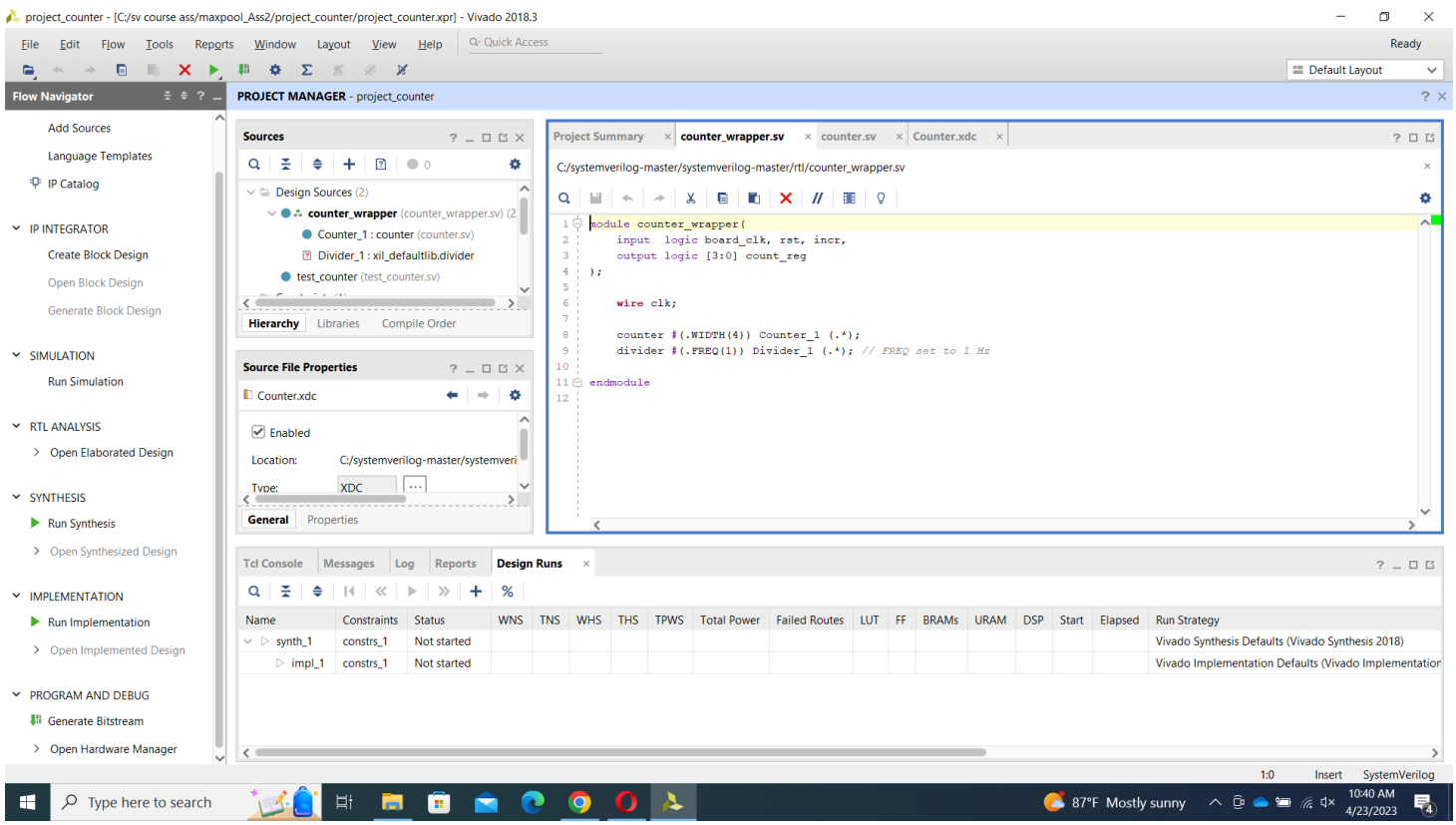
6.Generate the bitstream by clicking on "Generate Bitstream" under "Flow Navigator". Wait for the bitstream generation to complete.

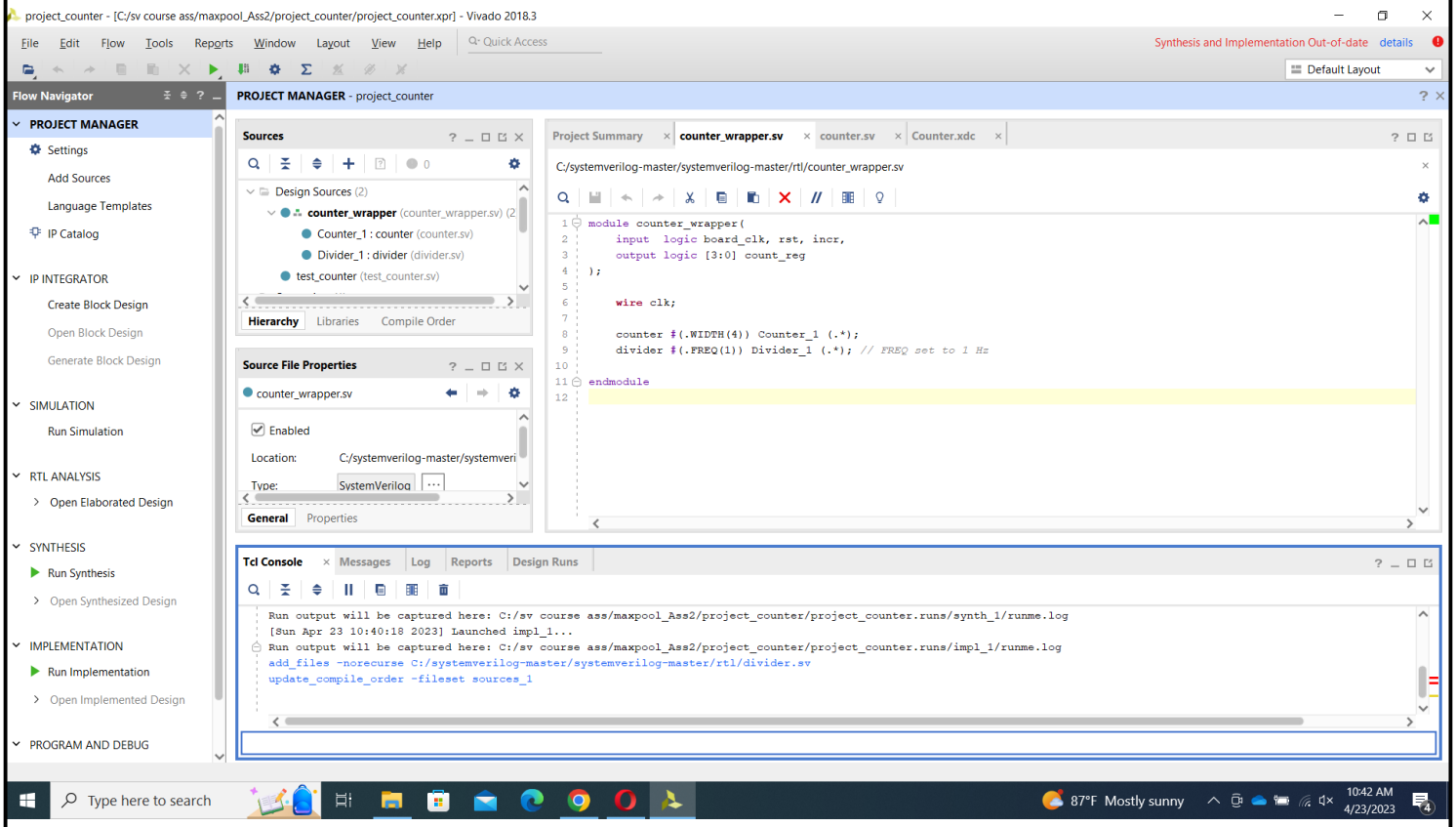
7.Launch the hardware manager by clicking on "Open Hardware Manager" under "Flow Navigator". Connect the Zybo board to your computer using a USB cable.

8.In the hardware manager, click on "Open Target" and select the Zybo board. Then, click on "Program Device" and select the bitstream file you generated in step 6.

9.Click "Program" to program the bitstream onto the Zybo board. Wait for the programming to complete

Counter module

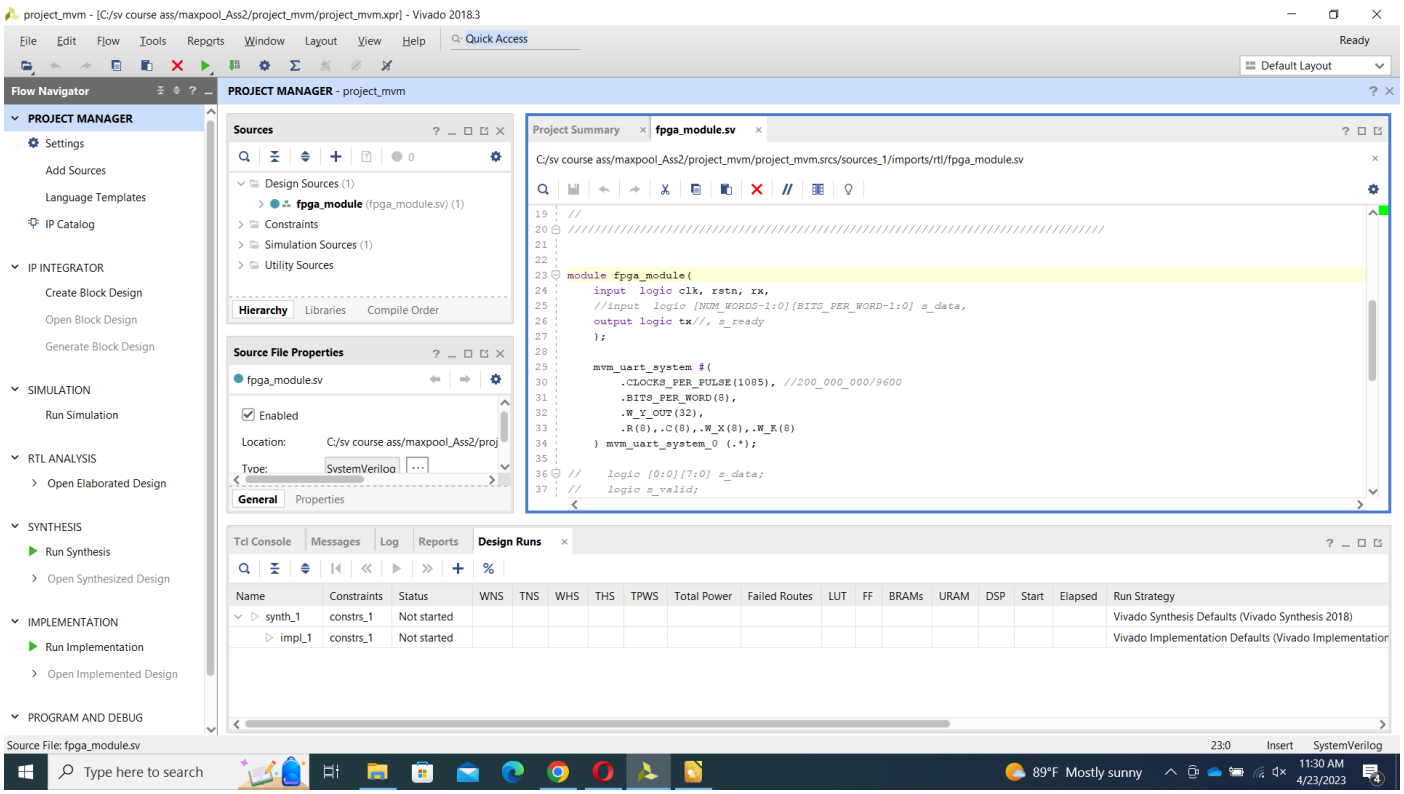




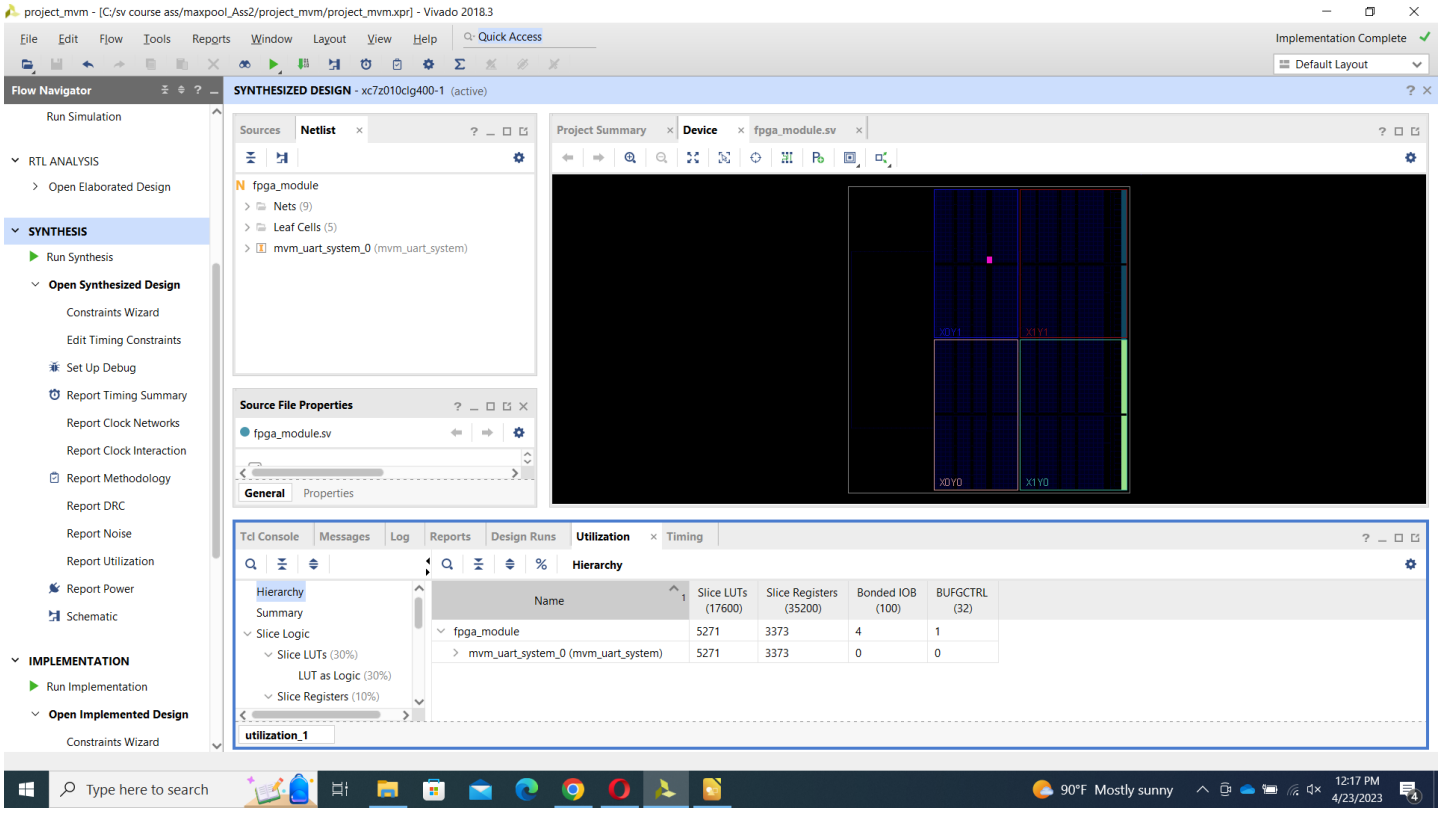
mvm_uart_system

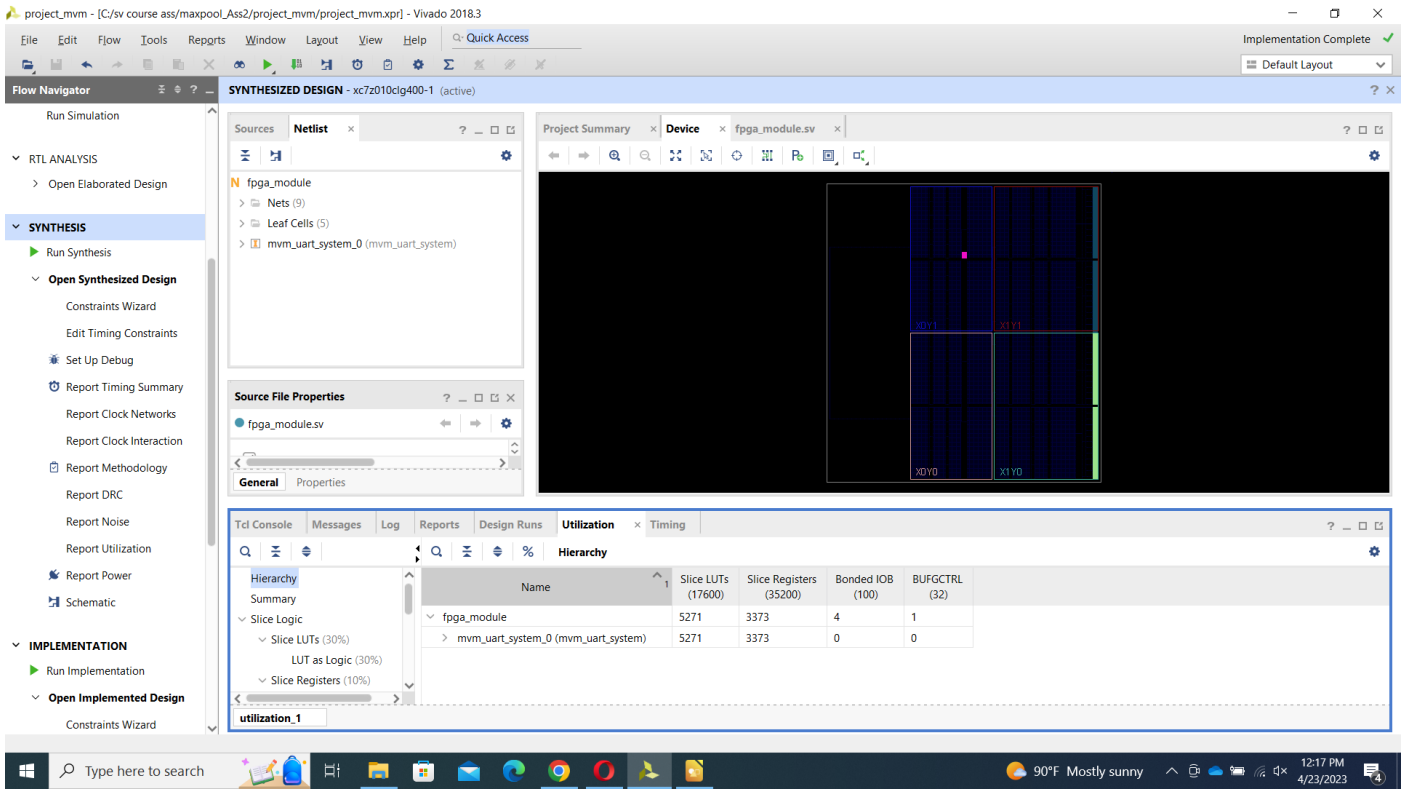
A matrix-vector multiplier is an operation that takes a matrix and a vector as inputs, and produces a vector as output. The output vector is obtained by multiplying each row of the matrix by the corresponding element of the input vector, and then summing the results.

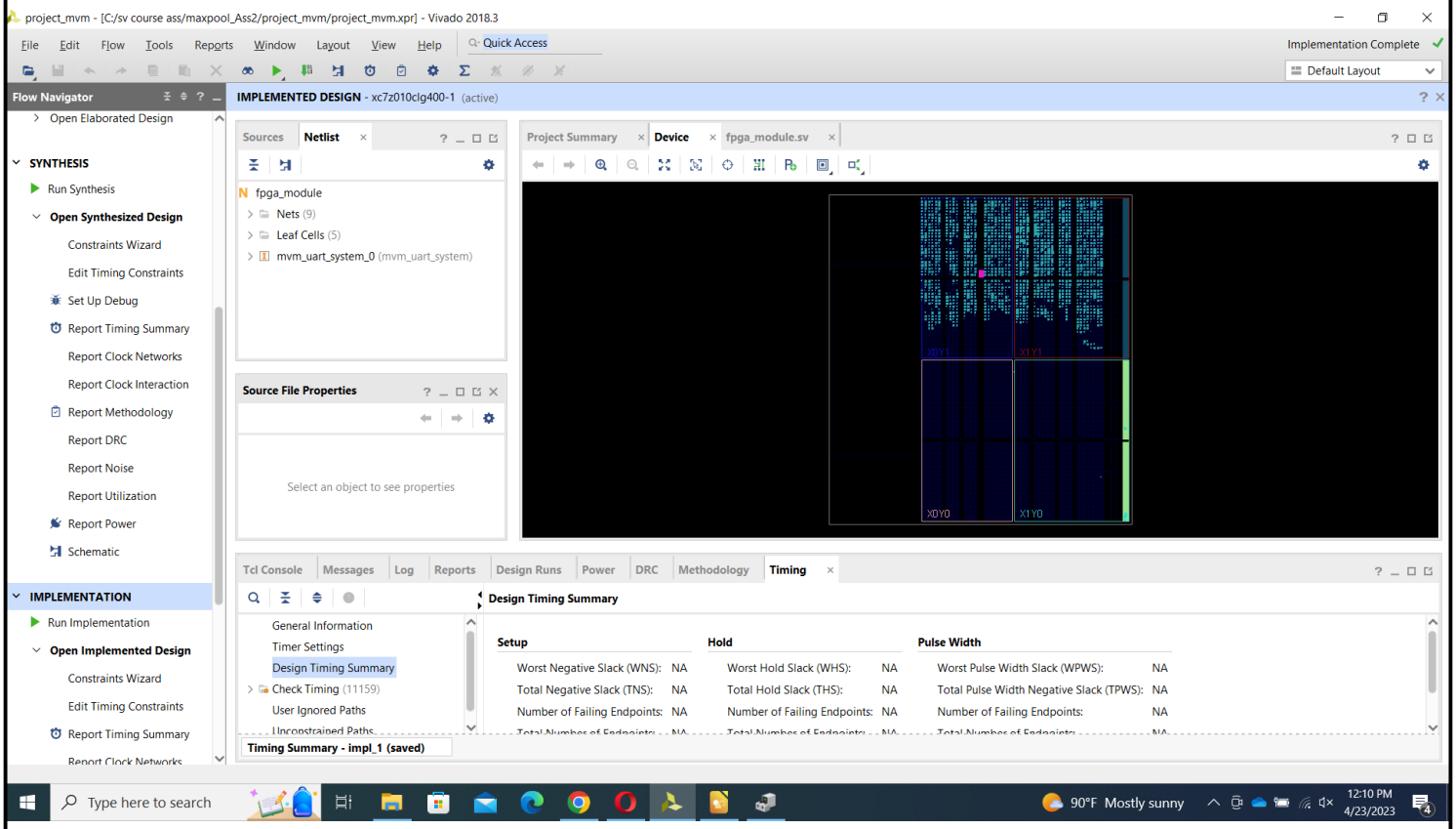
UART+AXIS streams are also used on the final system of mvm_uart_system.



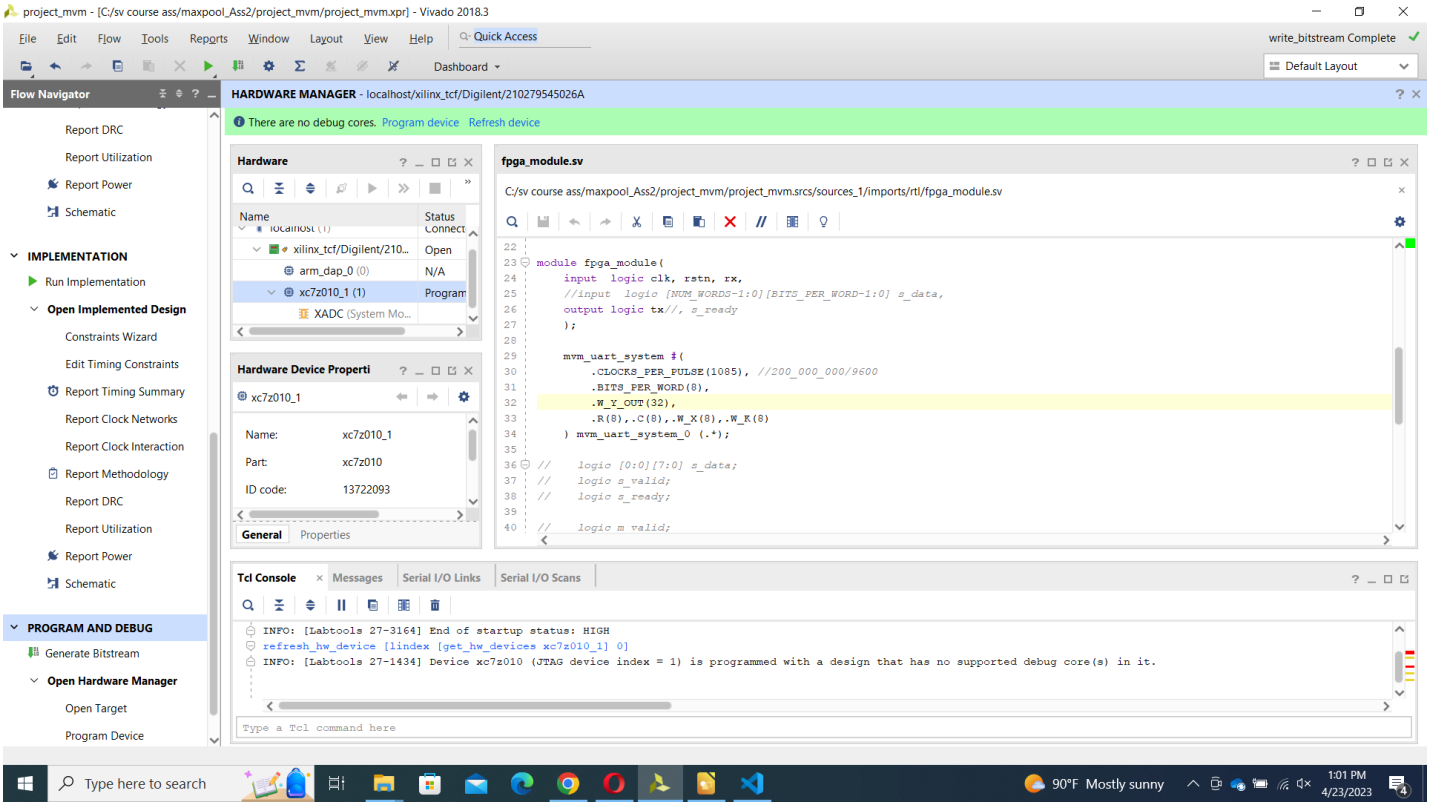
utilization reports of synthesis




project_mvm - [C:/sv course ass/maxpool_Ass2/project_mvm/project_mvm.xpr] - Vivado 2018.3
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
Constraints Wizard
SYNTHESIZED DESIGN - xc7z010clg400-1 (active)
Sources
Netlist
fpga_module
Nets (9)
Leaf Cells (5)
mvm_uart_system_0 (mvm_uart_system)
Project Summary
Device
fpga_module.sv
Source File Properties
fpga_module.sv
General
Properties
Tcl Console
Messages
Log
Reports
Design Runs
Utilization
Timing
Hierarchy
Summary
Slice Logic
Slice LUTs (30%)
LUT as Logic (30%)
Slice Registers (10%)
Name
Slice LUTs (17600)
Slice Registers (35200)
Bonded IOB (100)
BUFGCTRL (32)
fpga_module 5271 3373 4 1
mvm_uart_system_0 (mvm_uart_system) 5271 3373 0 0
utilization_1
Type here to search
90°F Mostly sunny
12:17 PM
4/23/2023


project_mvm - [C:/sv course ass/maxpool_Ass2/project_mvm/project_mvm.xpr] - Vivado 2018.3
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
Constraints Wizard
Edit Timing Constraints
Report Timing Summary
IMPLEMENTED DESIGN - xc7z010clg400-1 (active)
Sources
Netlist
fpga_module
Nets (9)
Leaf Cells (5)
mvm_uart_system_0 (mvm_uart_system)
Project Summary
Device
fpga_module.sv
Source File Properties
Select an object to see properties
Tcl Console
Messages
Log
Reports
Design Runs
Power
DRC
Methodology
Timing
Design Timing Summary
General Information
Timer Settings
Design Timing Summary
Check Timing (11159)
User Ignored Paths
Setup
Worst Negative Slack (WNS): NA
Total Negative Slack (TNS): NA
Number of Failing Endpoints: NA
Hold
Worst Hold Slack (WHS): NA
Total Hold Slack (THS): NA
Number of Failing Endpoints: NA
Pulse Width
Worst Pulse Width Slack (WPWS): NA
Total Pulse Width Negative Slack (TPWS): NA
Number of Failing Endpoints: NA
Timing Summary - impl_1 (saved)
Type here to search
90°F Mostly sunny
12:10 PM
4/23/2023

Successfully upload the code to the zybo fpga given in the lab



Some of the Final resultss of the mvm_uart system