INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

Department
of
Electronics & Communication Engineering

Embedded Logic Design(ECE270)

Dr. Sumit J Darak

Lab_3: Design and Implement 8-bit Counter operating at a frequency of 1 Hz

Mohammad Shariq
2020220

12-10-2021

OBJECTIVE:

1. Design and implement an 8-bit counter operating at a frequency of 1 Hz and verify its functionality.

2. Get familiar with Clocking Wizard IP and design a frequency divider capable of dividing the given frequency into the desired frequency.

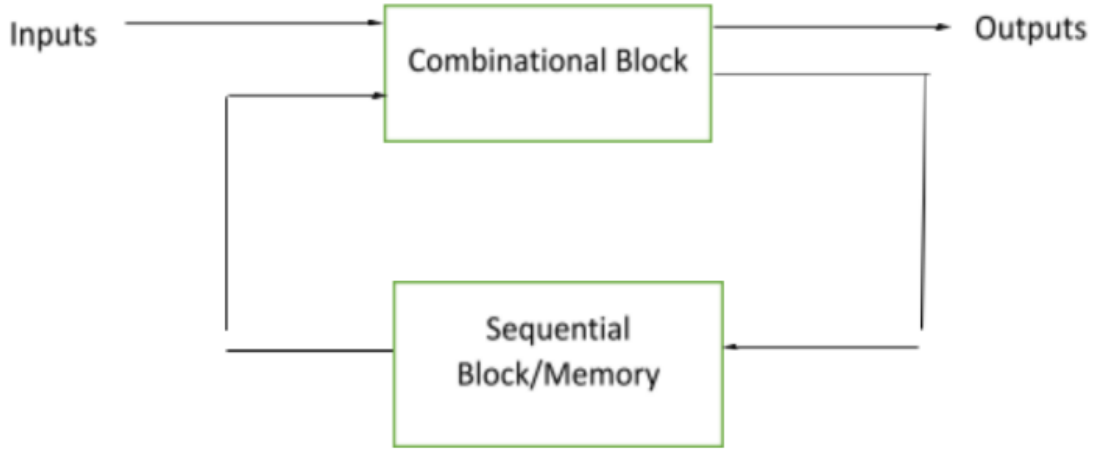
3. Test the functionality of Counter on the Basys3 Board (remote access) using VIO and ILA IP.

Theory:

- ***8-bit counter***

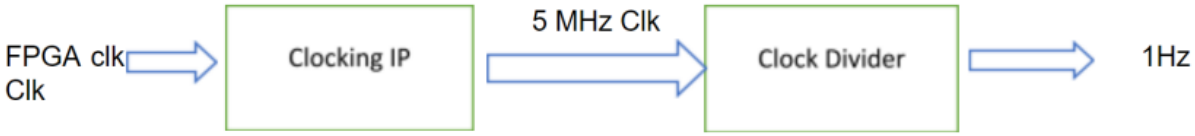
A counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relation to a clock. The most common type is a sequential digital logic circuit with an input line called the clock and multiple output lines. Each pulse applied to the clock input increments or decrements the number in the counter.
A counter circuit is usually constructed of a number of flip-flops connected in cascade. Counters are a very widely used component in digital circuits, and are manufactured as separate integrated circuits and also incorporated as parts of larger integrated circuits.(src.wiki)

2. The Counter is a sequential circuit. The block diagram of a sequential circuit is shown below:



To synthesize a clock of lower frequencies from the input clock frequency of 100 Mhz, we can make use of Clocking IP.

Note: The minimum clock frequency that can be synthesized from Clock IP is 4.7 MHz. So we will first synthesize a clock of 5MHz frequency from Clock IP and then feed it to the clock divider to make a clock of 1Hz frequency.



1. Open the IP Catalog and search for Clocking Wizard IP. Double click on it.

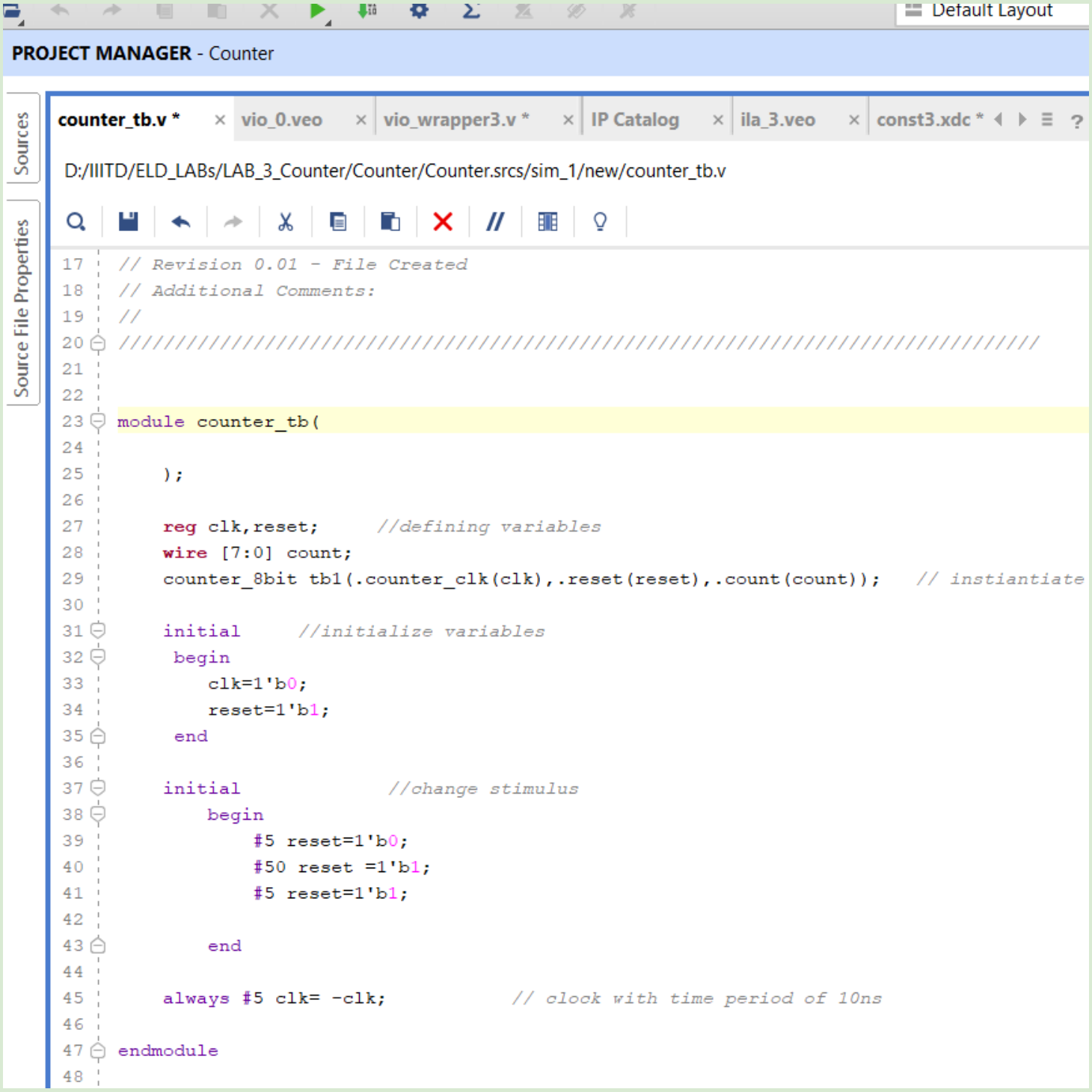
Observations:

programme for functionality of 8-bit counter

D:/IIITD/ELD_LABs/LAB_3_Counter/Counter/Counter.srcs/sources_1/new/top_counter.v

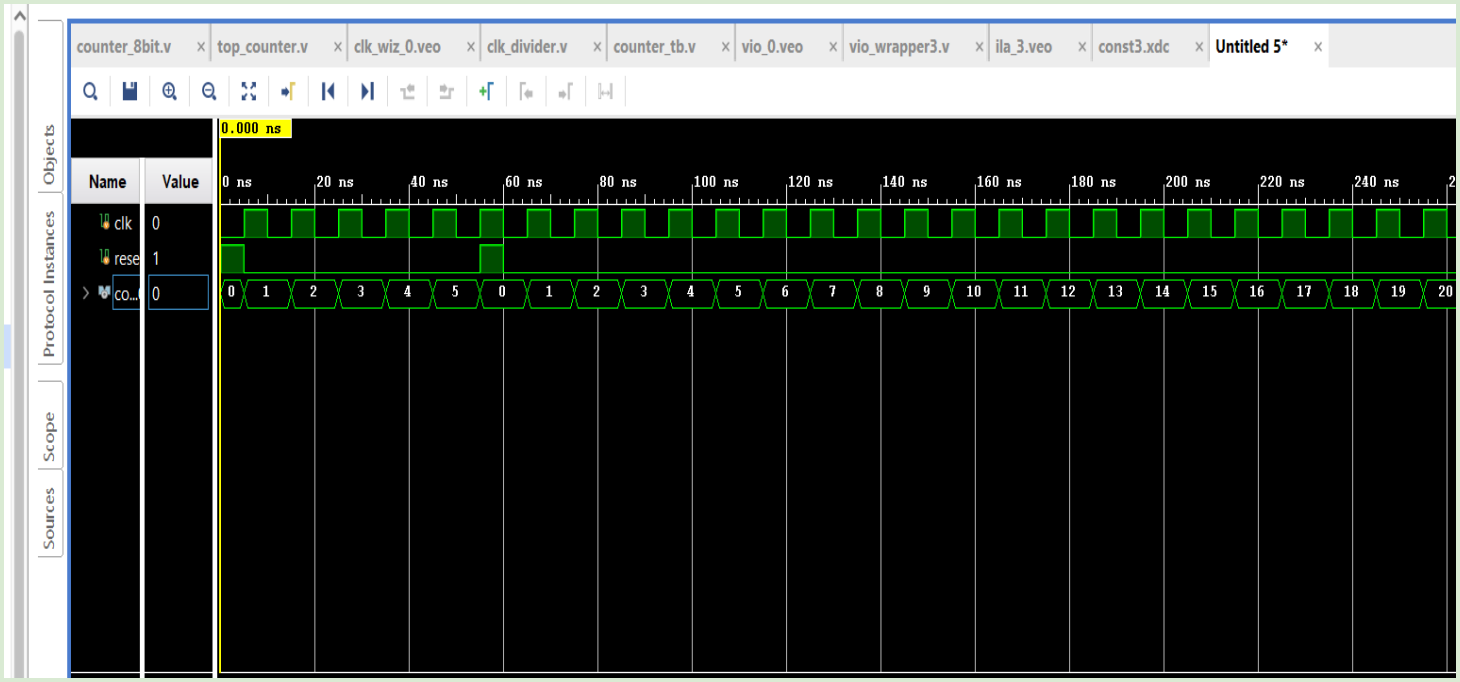
```
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module top_counter(
24      input clk_100M,       // Clock from FPGA itself
25      input reset,
26      output [7:0] count
27
28      );
29
30      wire clk_5M ,clk_1H;
31
32    clk_wiz_0 clk_IN0(
33      // Clock out ports
34      .clk_out1(clk_5M),      // output clk_out1
35     // Clock in ports
36      .clk_in1(clk_100M)      // input clk_in1
37      );
38
39      clk_divider #(.div_value(2499999)) clk_div_IN0(.clk_in(clk_5M),.divided_clk(clk_1H));
40
41      counter_8bit count_IN0(.counter_clk(clk_1H),.reset(reset),.count(count));
42
43      ila_3 ila_IN0(
44      .clk(clk_100M), // input wire clk
45
46
47      .probe0(clk_1H), // input wire [0:0]  probe0
48      .probe1(reset), // input wire [0:0]  probe1
49      .probe2(count) // input wire [7:0]  probe2
50  );
51  endmodule
```

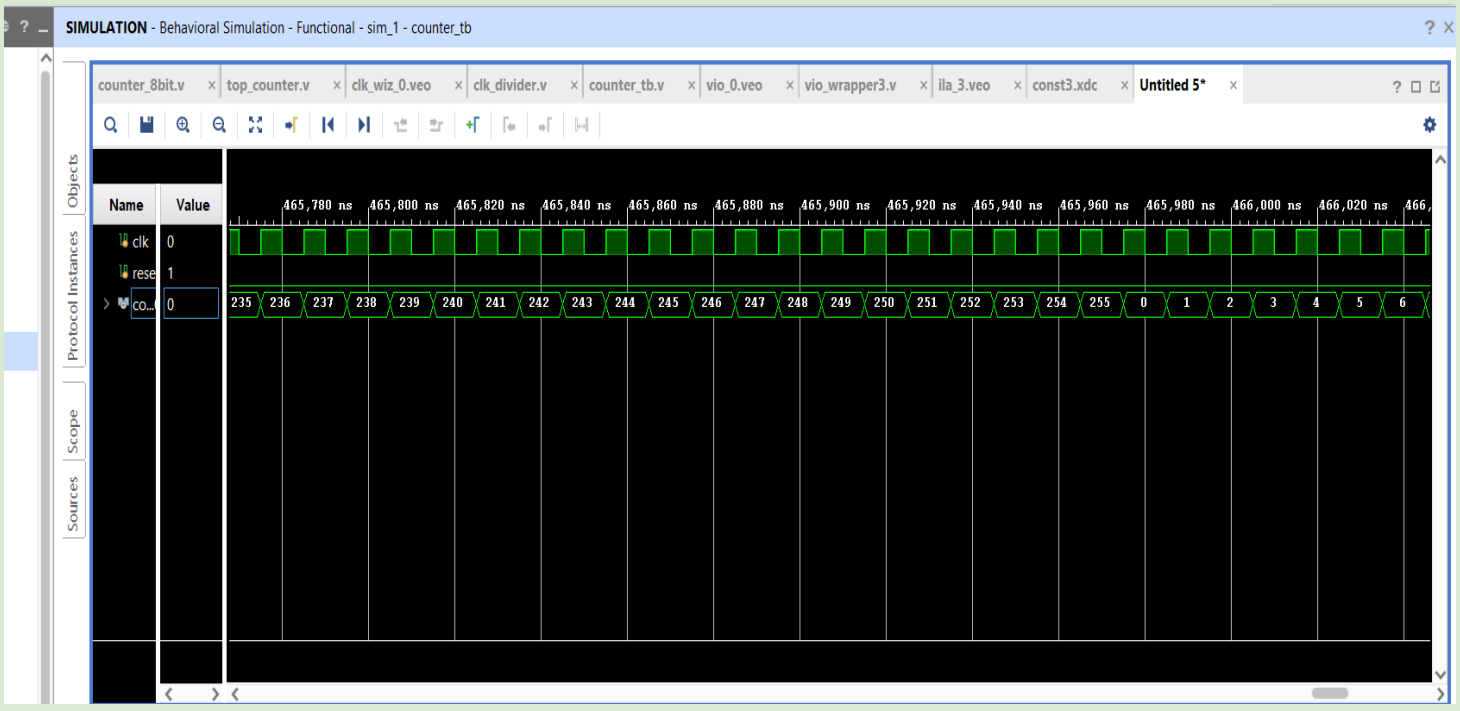
Testbench for 8-bit Counter



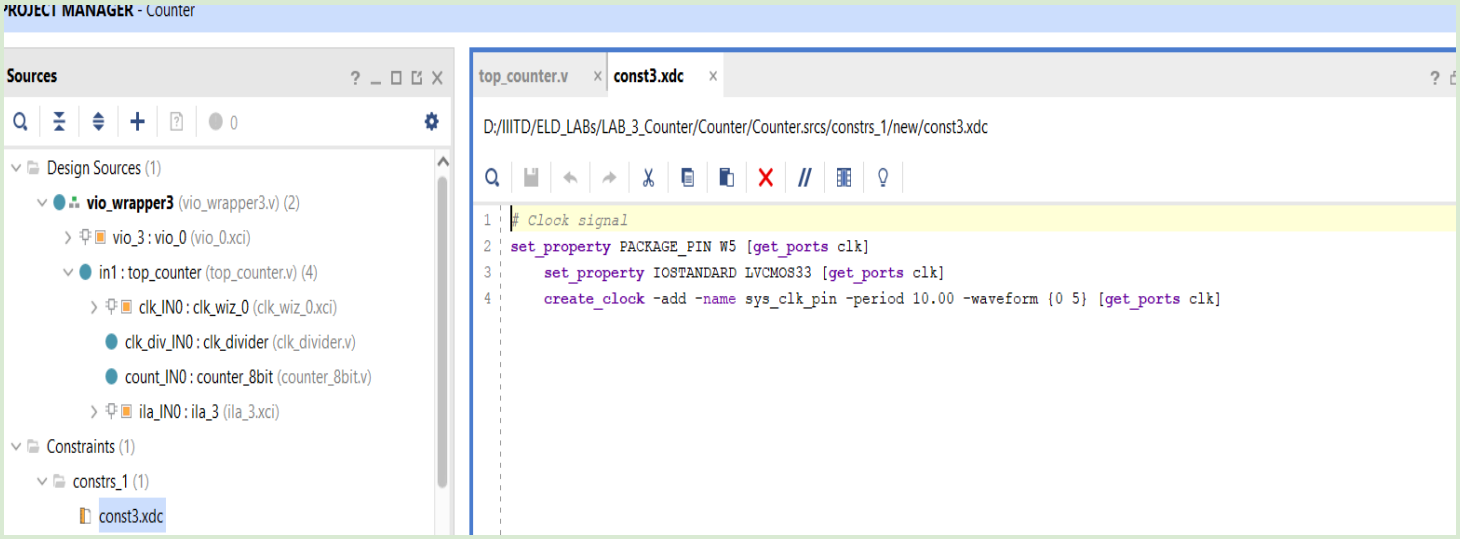

```
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////


23  module counter_tb(

25      );

27      reg clk,reset;      //defining variables
28      wire [7:0] count;
29      counter_8bit tb1(.counter_clk(clk),.reset(reset),.count(count));    // instiantiate

31      initial     //initialize variables
32       begin
33          clk=1'b0;
34          reset=1'b1;
35       end

37      initial             //change stimulus
38          begin
39              #5 reset=1'b0;
40              #50 reset =1'b1;
41              #5 reset=1'b1;

43          end

45      always #5 clk= ~clk;            // clock with time period of 10ns

47  endmodule
```

Results for automated testbench:

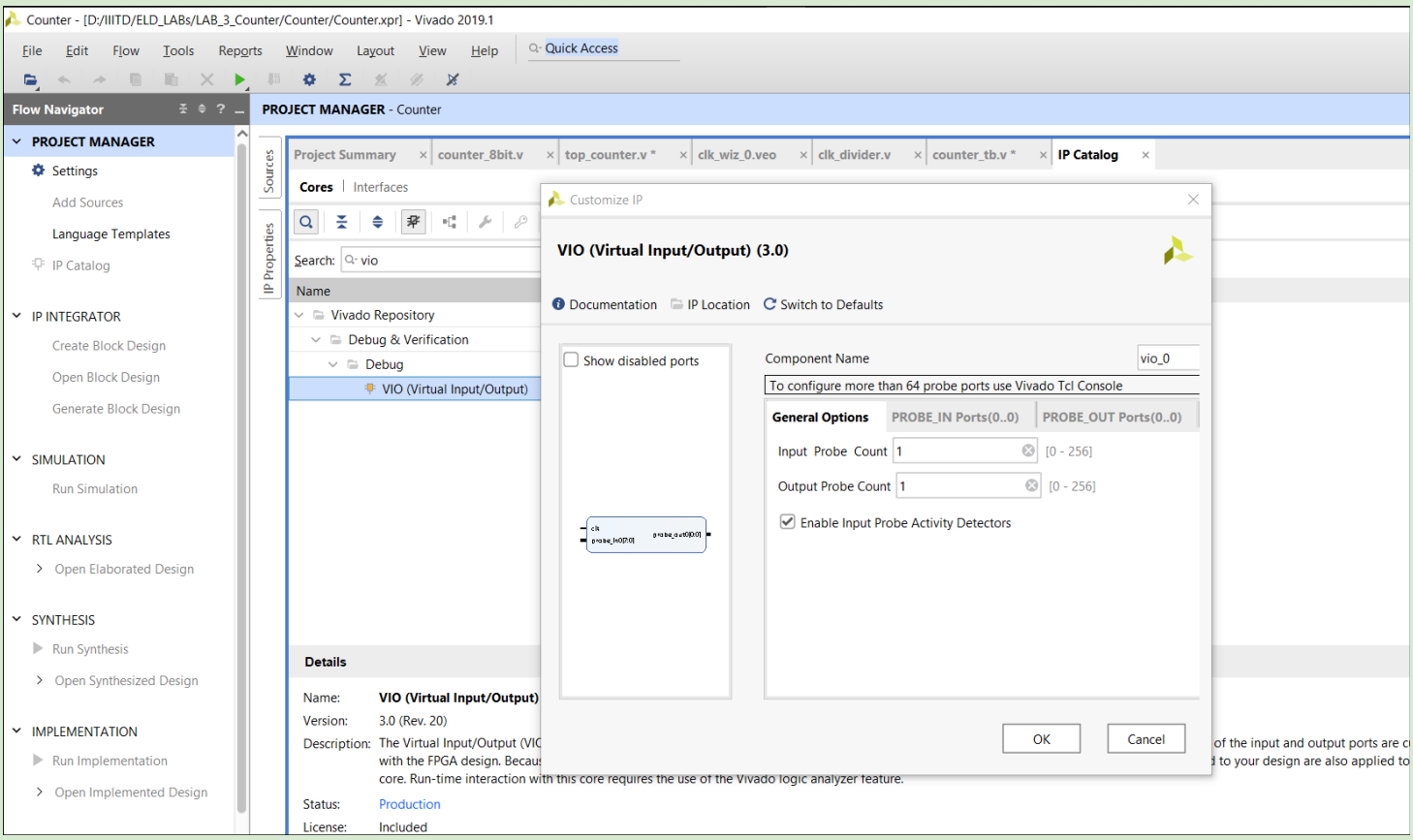


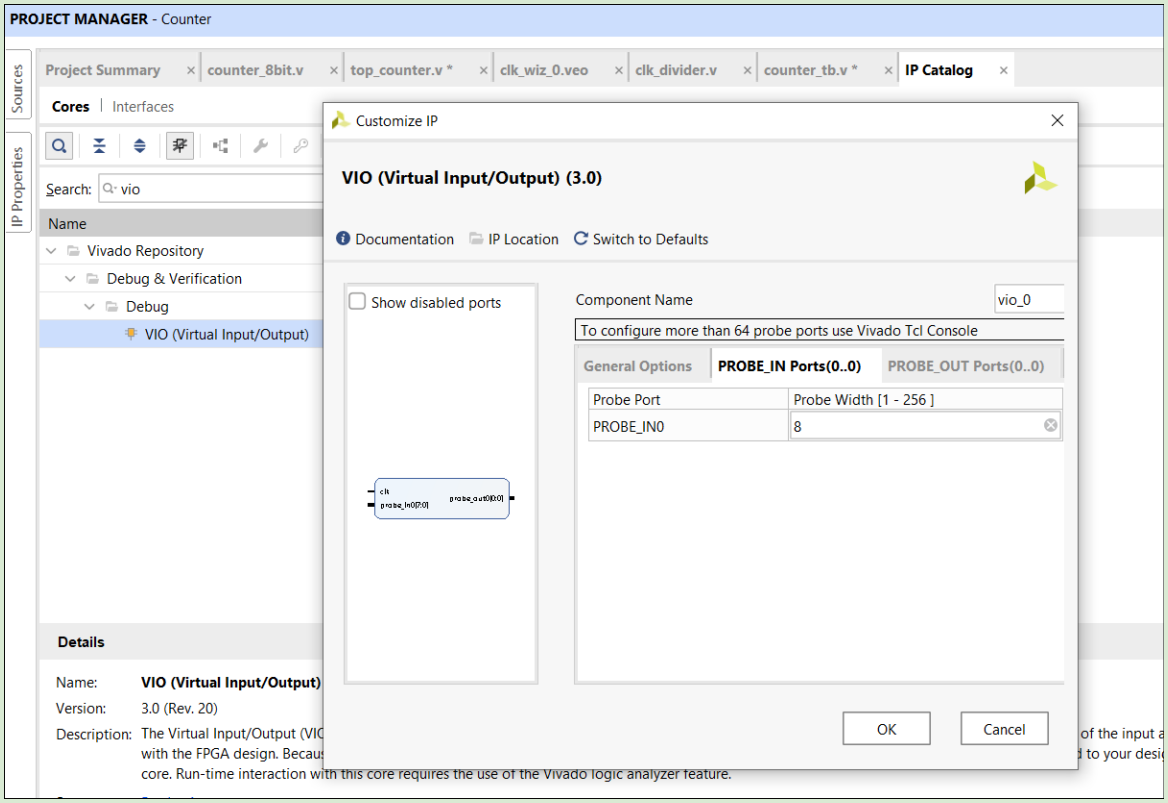


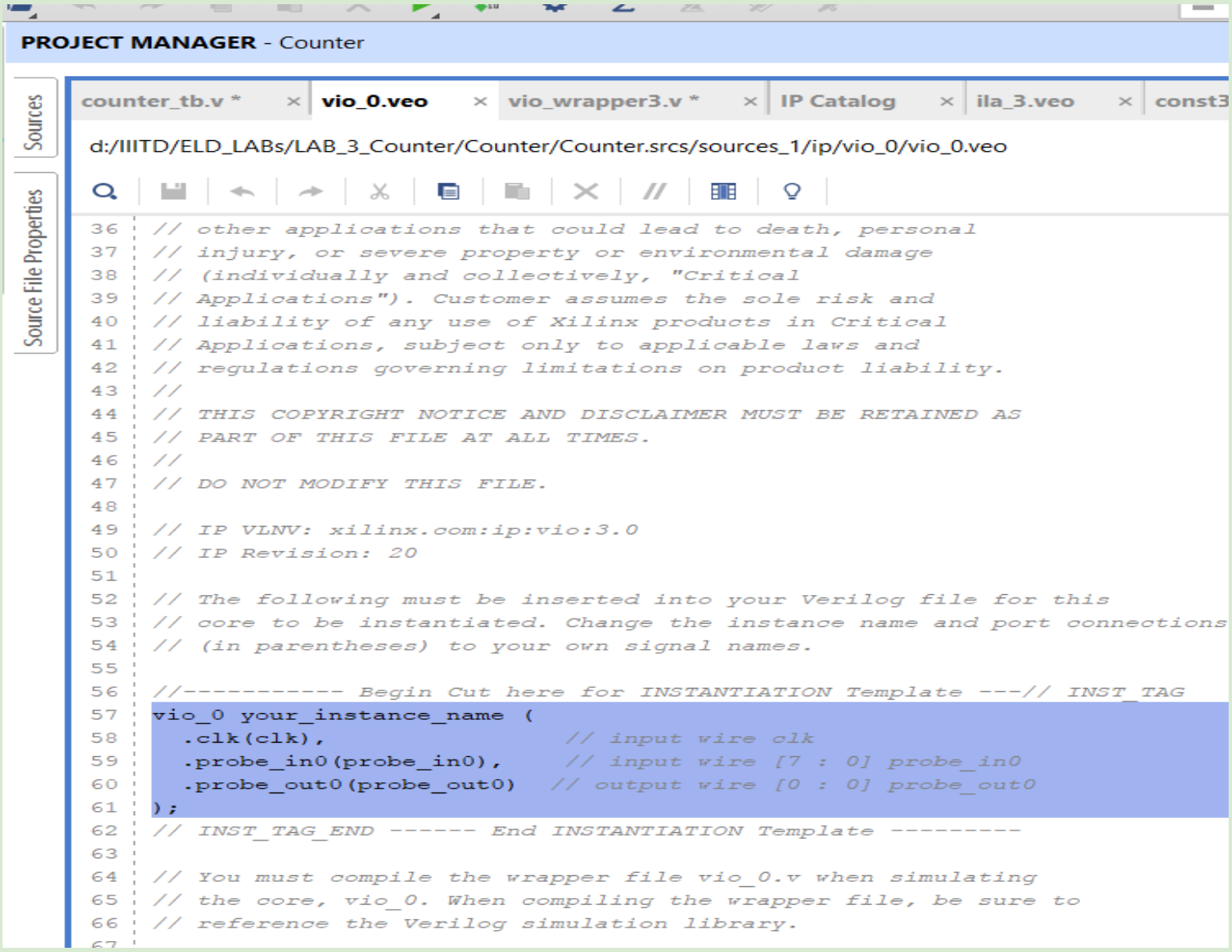
Constraints for the implementation and VIO module programme

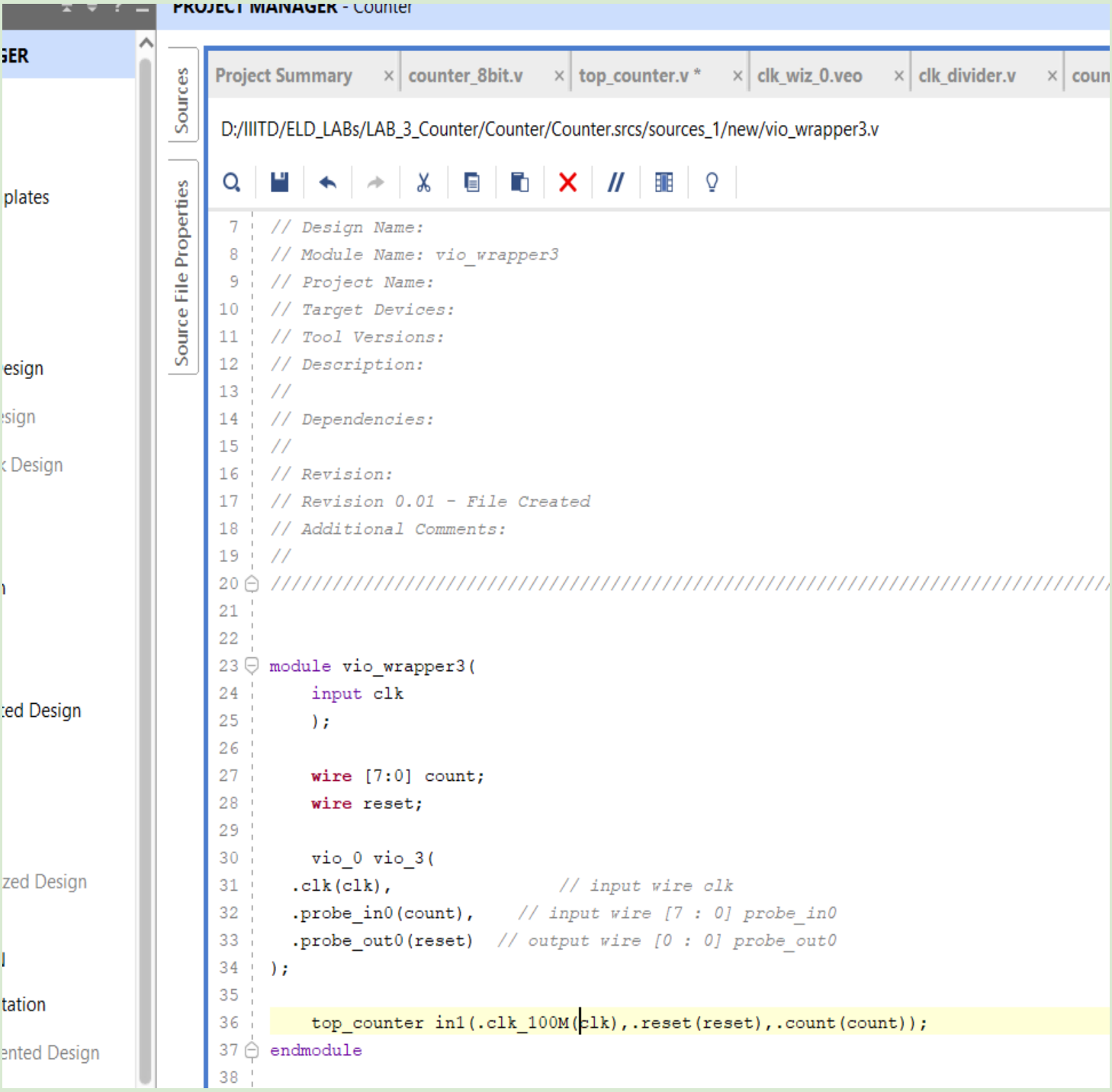


VIO Wrapper Settings:

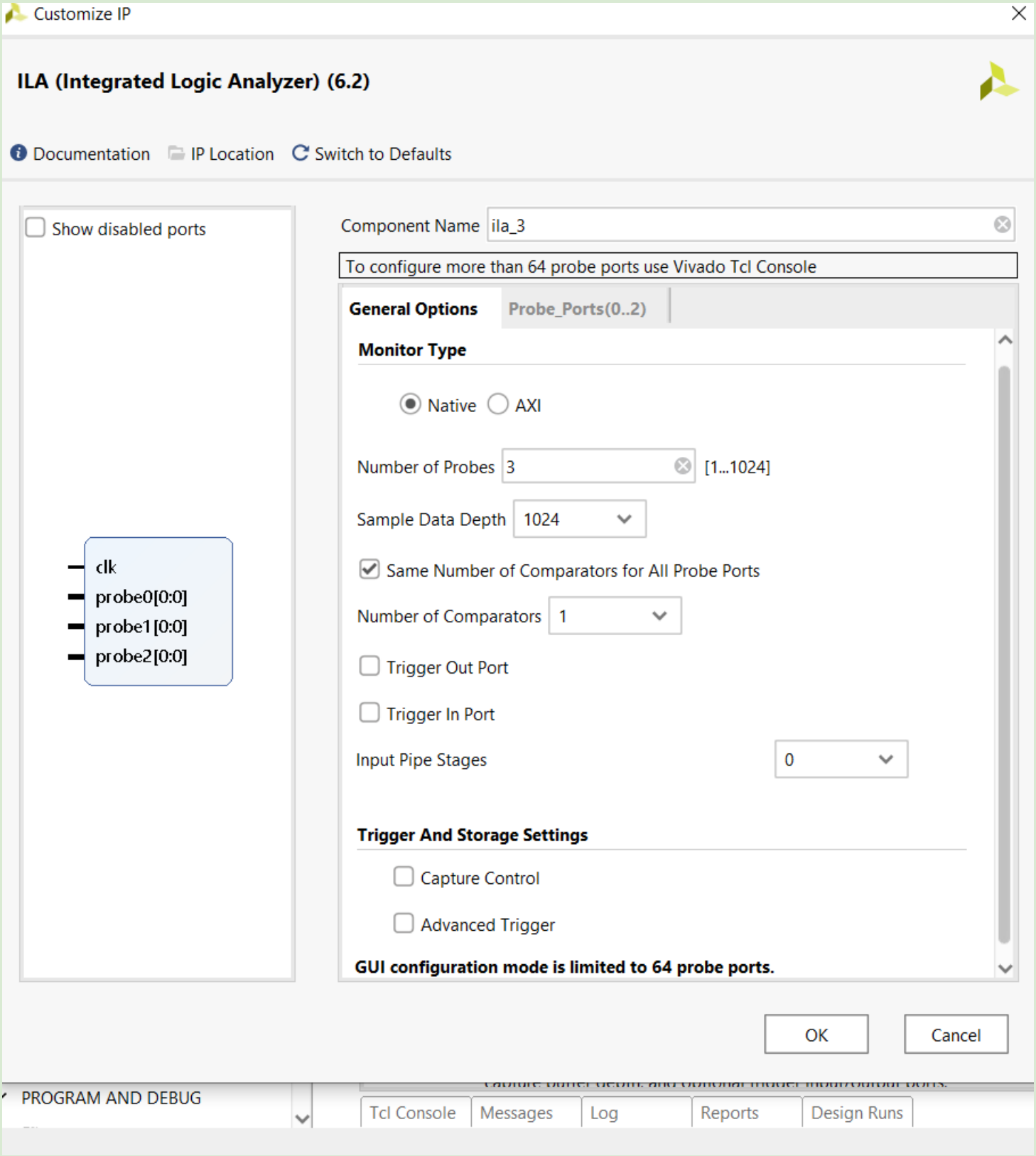



PROJECT MANAGER - Counter
Project Summary
counter_8bit.v
top_counter.v *
clk_wiz_0.veo
clk_divider.v
counter_tb.v *
IP Catalog
Cores
Interfaces
Search: vio
Name
Vivado Repository
Debug & Verification
Debug
VIO (Virtual Input/Output)
Customize IP
VIO (Virtual Input/Output) (3.0)
Documentation
IP Location
Switch to Defaults
Show disabled ports
Component Name
vio_0
To configure more than 64 probe ports use Vivado Tcl Console
General Options
PROBE_IN Ports(0..0)
PROBE_OUT Ports(0..0)
Probe Port
Probe Width [1 - 256]
PROBE_IN0
8
OK
Cancel
Details
Name: VIO (Virtual Input/Output)
Version: 3.0 (Rev. 20)
Description: The Virtual Input/Output (VIO
with the FPGA design. Becau
core. Run-time interaction with this core requires the use of the Vivado logic analyzer feature.

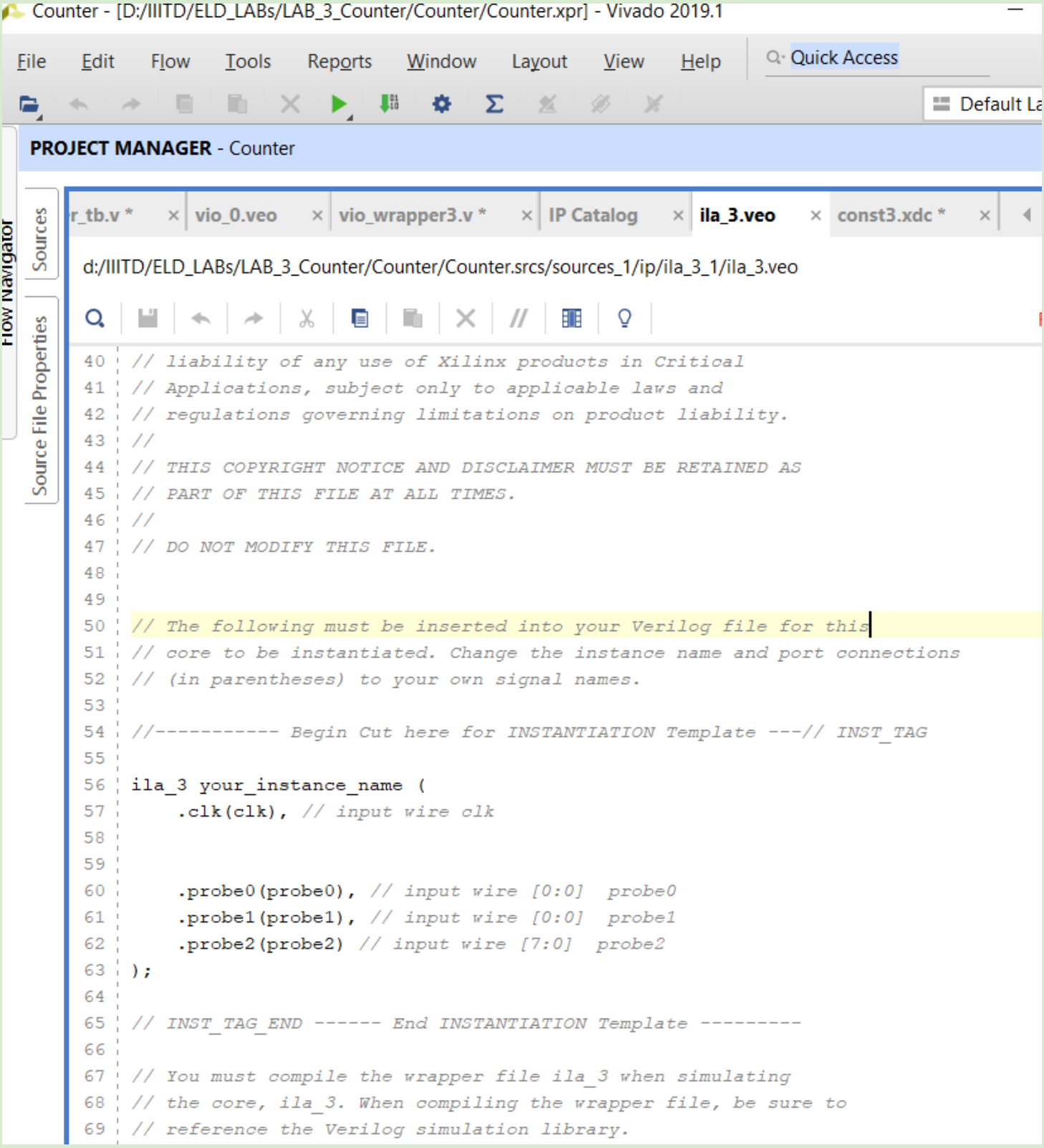

PROJECT MANAGER - Counter
Sources
Source File Properties
counter_tb.v *
vio_0.veo
vio_wrapper3.v *
IP Catalog
ila_3.veo
d:/IIITD/ELD_LABs/LAB_3_Counter/Counter/Counter.srcs/sources_1/ip/vio_0/vio_0.veo
// other applications that could lead to death, personal
// injury, or severe property or environmental damage
// (individually and collectively, "Critical
// Applications"). Customer assumes the sole risk and
// liability of any use of Xilinx products in Critical
// Applications, subject only to applicable laws and
// regulations governing limitations on product liability.
//
// THIS COPYRIGHT NOTICE AND DISCLAIMER MUST BE RETAINED AS
// PART OF THIS FILE AT ALL TIMES.
//
// DO NOT MODIFY THIS FILE.
// IP VLNV: xilinx.com:ip:vio:3.0
// IP Revision: 20
// The following must be inserted into your Verilog file for this
// core to be instantiated. Change the instance name and port connections
// (in parentheses) to your own signal names.
//----------- Begin Cut here for INSTANTIATION Template ---// INST_TAG
vio_0 your_instance_name (
.clk(clk), // input wire clk
.probe_in0(probe_in0), // input wire [7 : 0] probe_in0
.probe_out0(probe_out0) // output wire [0 : 0] probe_out0
);
// INST_TAG_END ------ End INSTANTIATION Template ---------
// You must compile the wrapper file vio_0.v when simulating
// the core, vio_0. When compiling the wrapper file, be sure to
// reference the Verilog simulation library.


PROJECT MANAGER - Counter
Sources
Source File Properties
Project Summary
counter_8bit.v
top_counter.v *
clk_wiz_0.veo
clk_divider.v
D:/IIITD/ELD_LABs/LAB_3_Counter/Counter/Counter.srcs/sources_1/new/vio_wrapper3.v
// Design Name:
// Module Name: vio_wrapper3
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
module vio_wrapper3(
input clk
);
wire [7:0] count;
wire reset;
vio_0 vio_3(
.clk(clk), // input wire clk
.probe_in0(count), // input wire [7 : 0] probe_in0
.probe_out0(reset) // output wire [0 : 0] probe_out0
);
top_counter in1(.clk_100M(clk),.reset(reset),.count(count));
endmodule

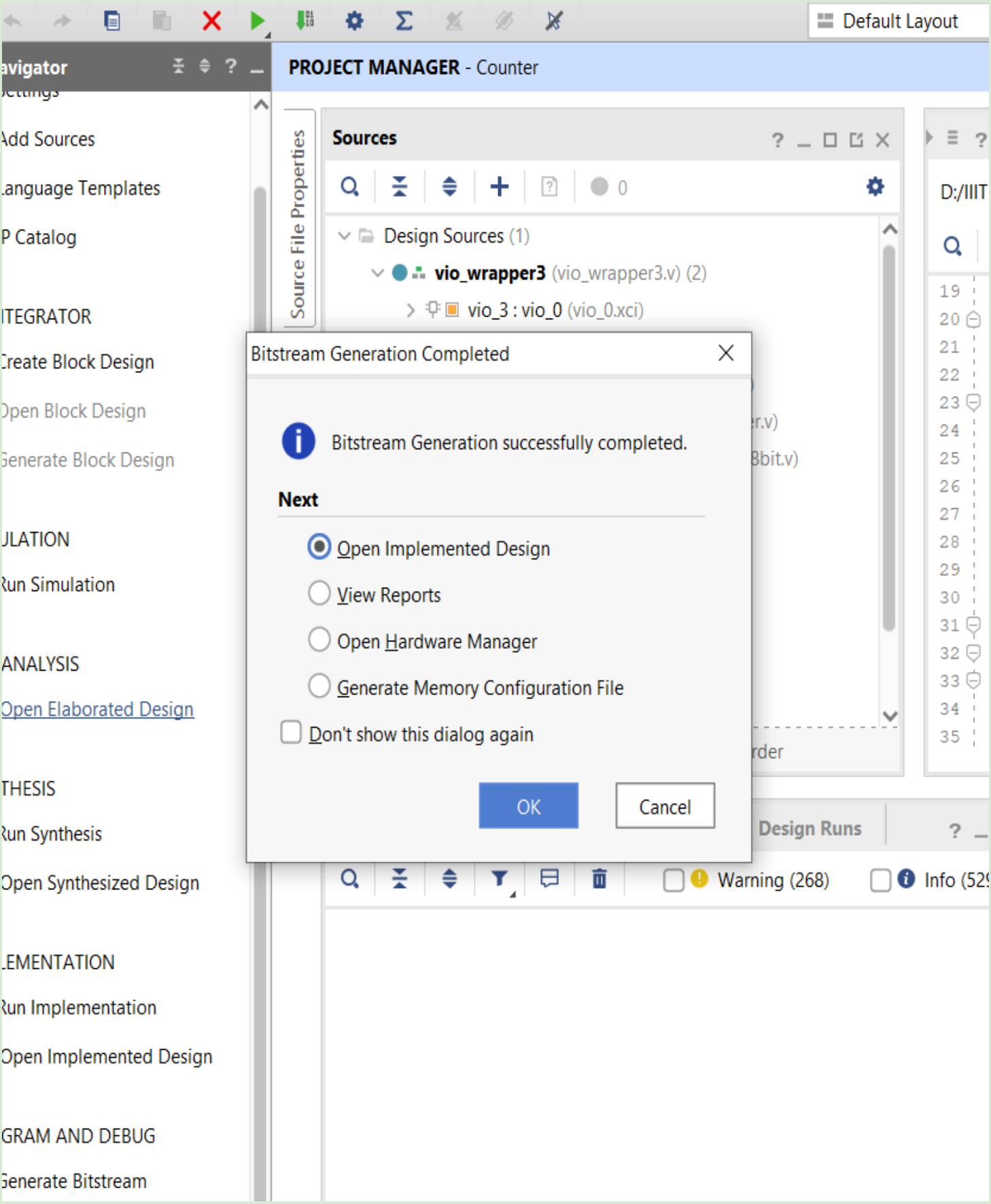
ILA Settings:

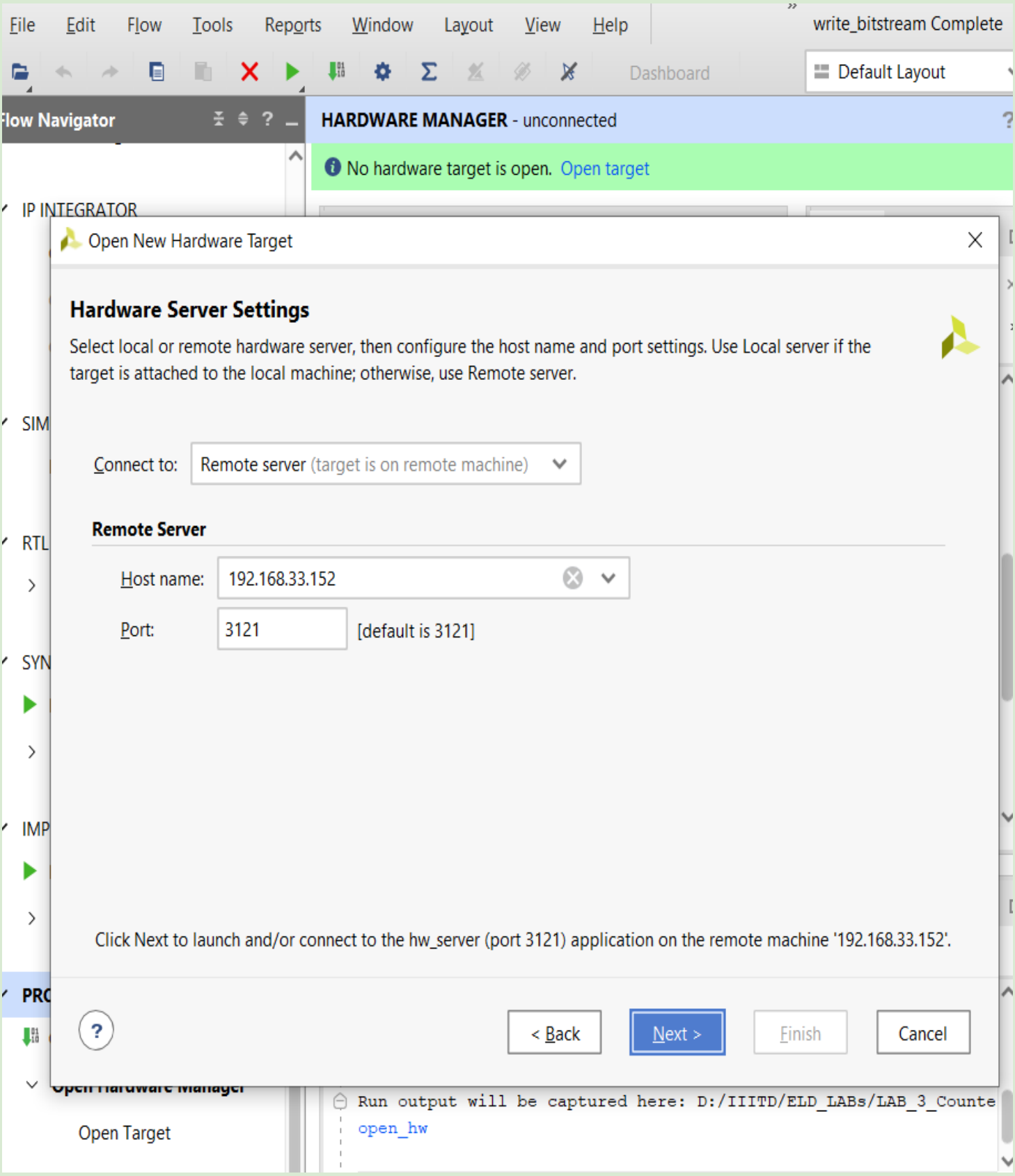


ILA instantiation template:

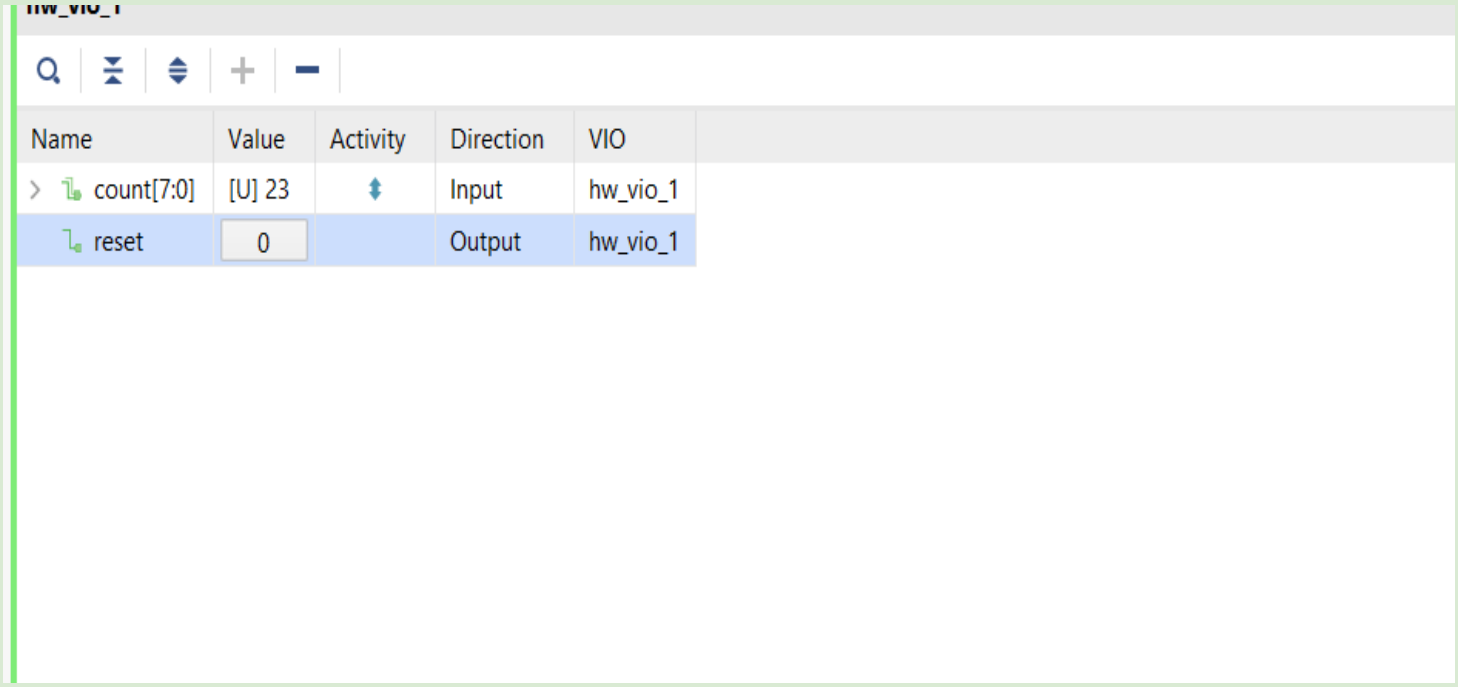


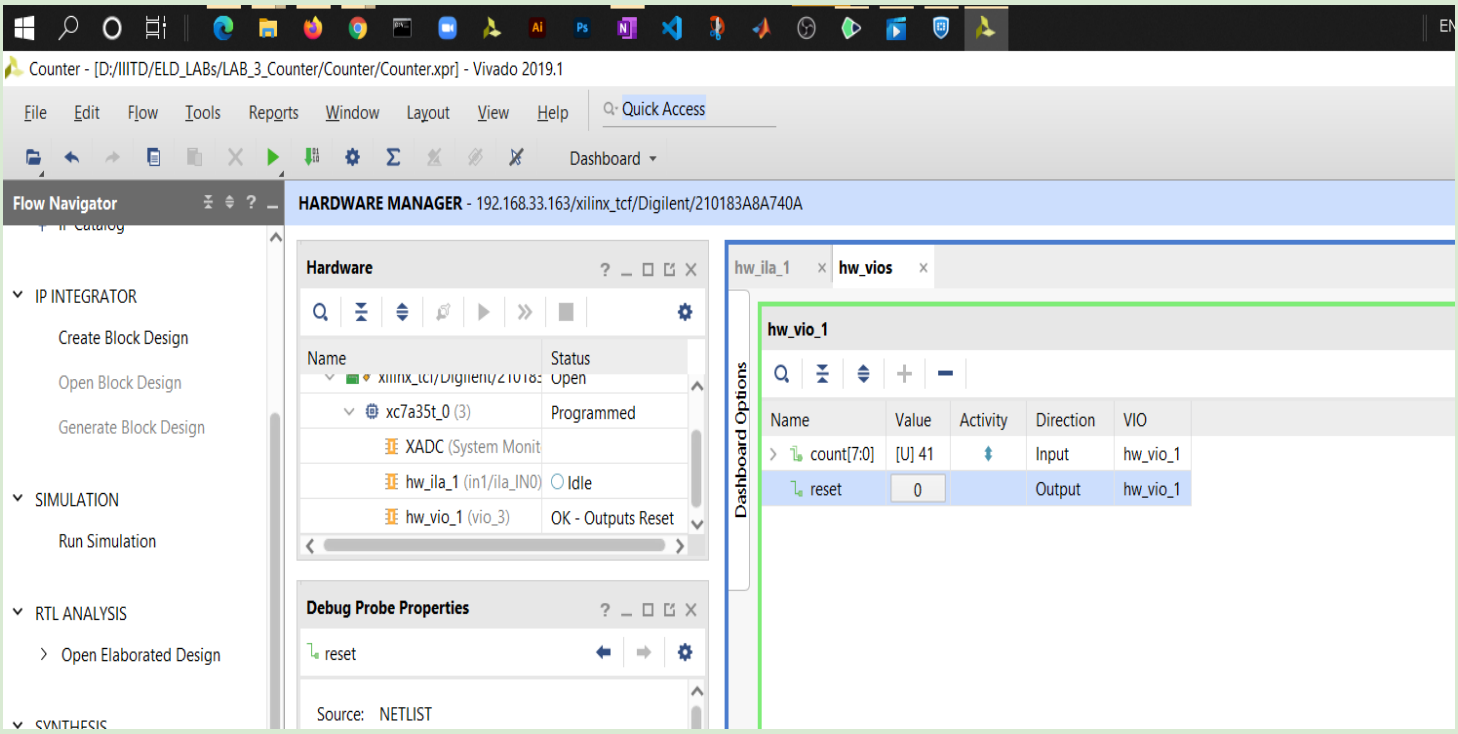
BitStream Generated Successfully:




File Edit Flow Tools Reports Window Layout View Help
write_bitstream Complete
Dashboard
Default Layout
Flow Navigator
HARDWARE MANAGER - unconnected
No hardware target is open. Open target
IP INTEGRATOR
SIM
RTL
SYN
IMP
PRO
Open Hardware Manager
Open Target
Open New Hardware Target
Hardware Server Settings
Select local or remote hardware server, then configure the host name and port settings. Use Local server if the target is attached to the local machine; otherwise, use Remote server.
Connect to: Remote server (target is on remote machine)
Remote Server
Host name: 192.168.33.152
Port: 3121 [default is 3121]
Click Next to launch and/or connect to the hw_server (port 3121) application on the remote machine '192.168.33.152'.
< Back
Next >
Finish
Cancel
Run output will be captured here: D:/IIITD/ELD_LABs/LAB_3_Counte
open_hw

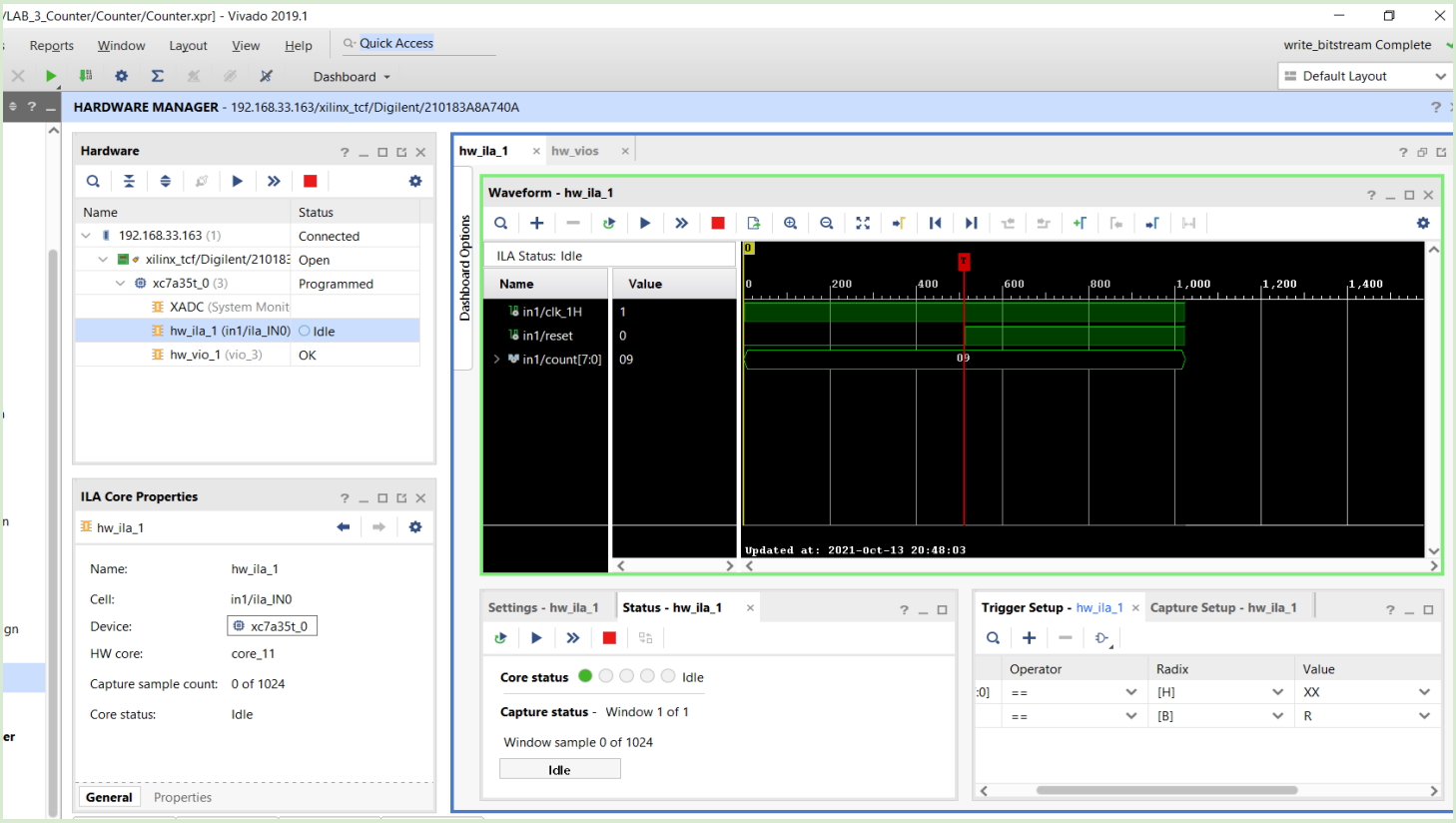
Testing on BASYS3 Board using VIO IP



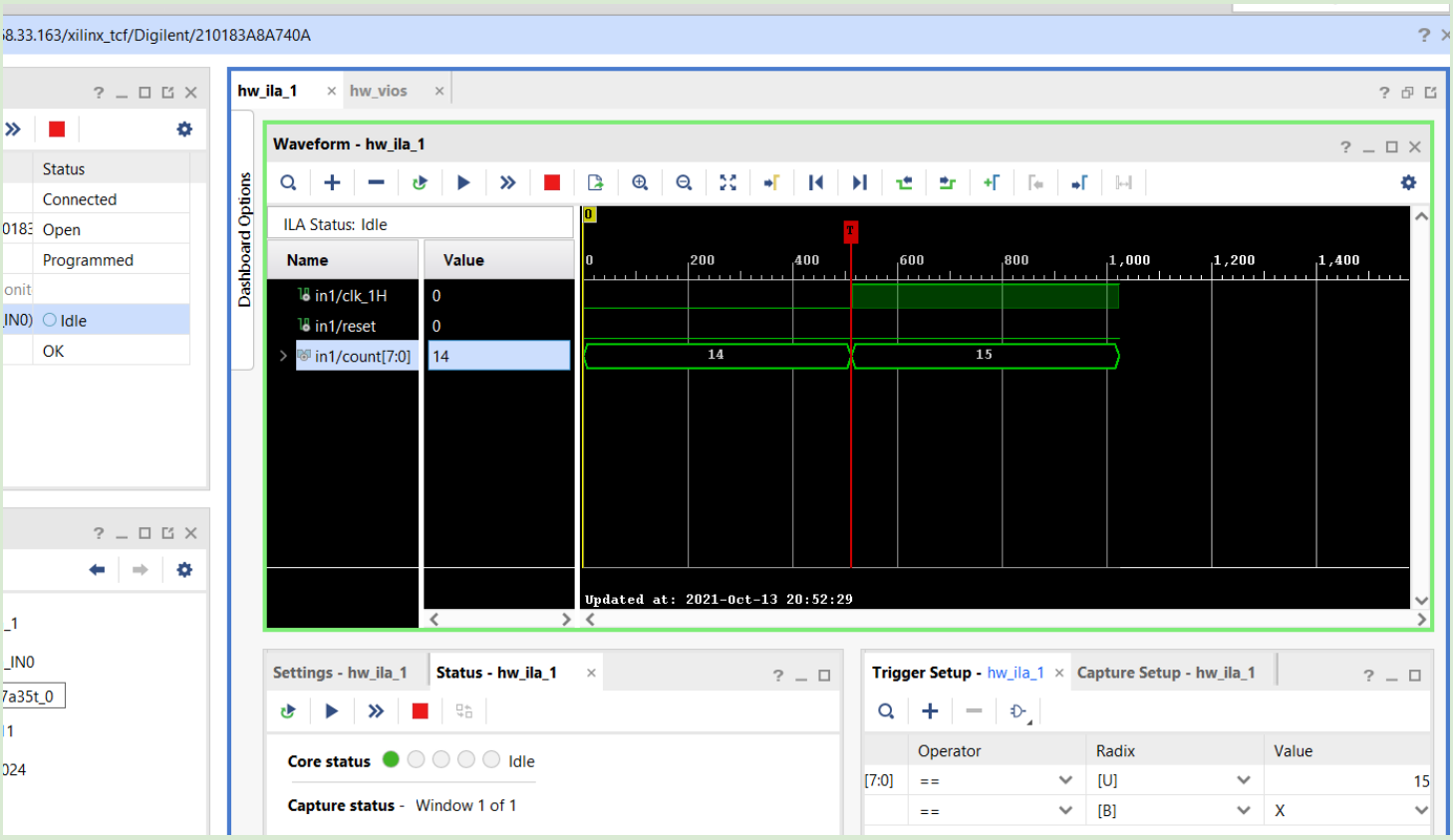


De-assert the reset, and you can see the Counter incrementing from 0 to 255 and back to 0.

Testing On ILA:



showing waveform by ILA after 9 counts.



ILA shows a waveform of clk after 15 counts which are automated with a don't care trigger and ILA was set up to stop at 15 counts.

Conclusion:

Successfully tested functionality of 8-bit Counter on BASYS3 board using VIO IP and ILA (Remote Access).