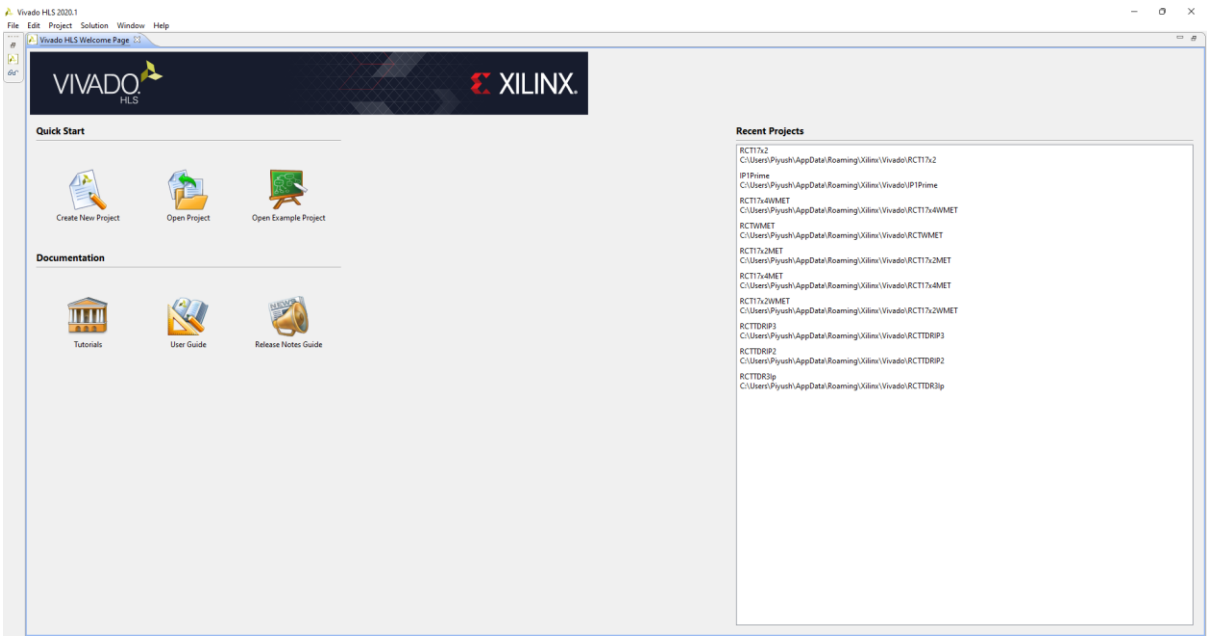
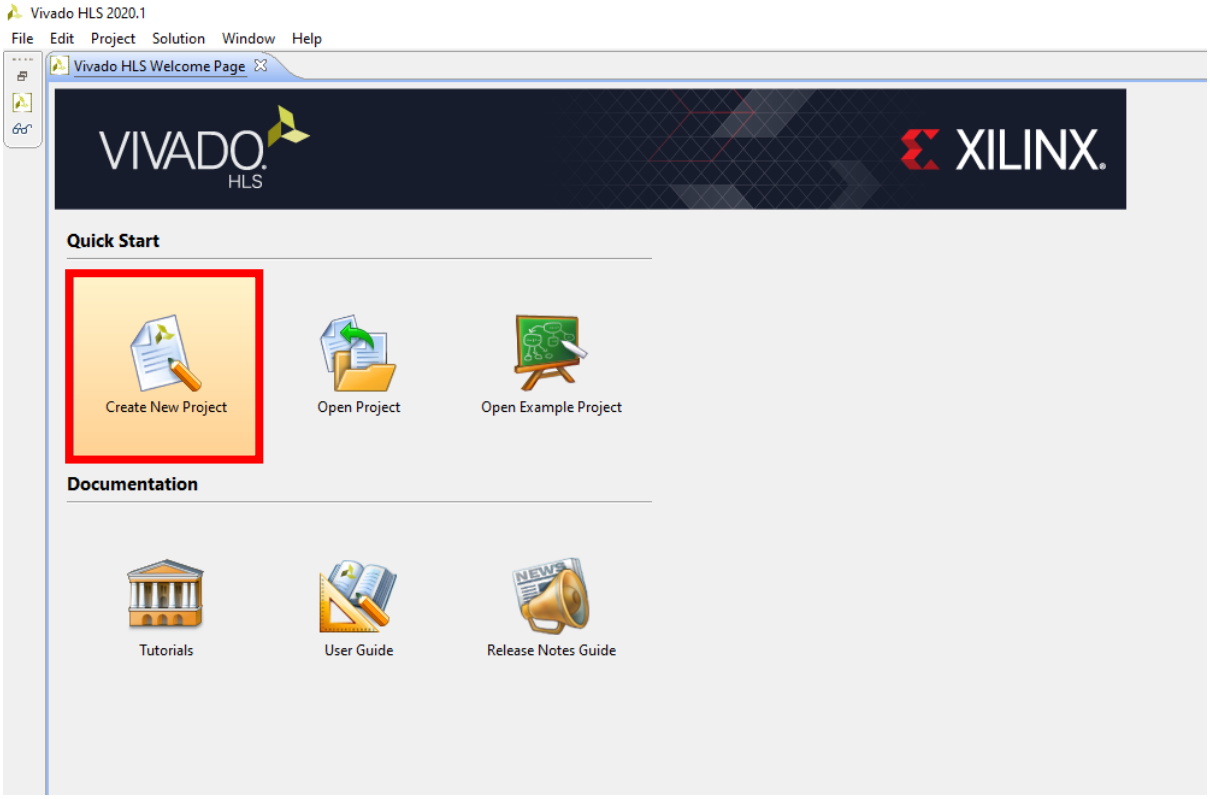
VIVADO-HLS Instructions

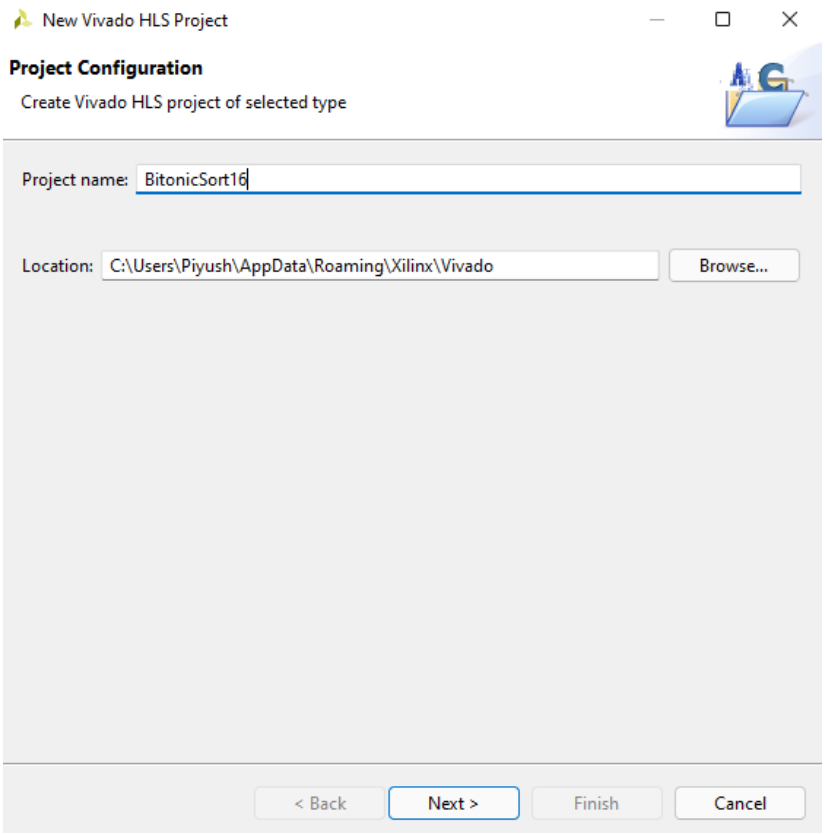
- Download the source code and instructions PDF using the following link
  - https://github.com/piyushkumarhcu/BitonicSort-HLS/tree/master/bitonic16hls
- Source VIVADO setting64.sh (please check the path of your Vivado installation directory first)
  - source /Xilinx/Vivado/2020.1/settings64.sh
- Create the project folder with a relevant name
  - mkdir HLS19PHPE04
  - cd HLS19PHPE04
- Create a "src" folder and store the source code (*.h and *.cpp files) in the same
- Open the Vivado-HLS using the following command in terminal (after sourcing the setting64.sh) (in HLS19PHPE04 directory)
  - vivado_hls
- You will see the following screen



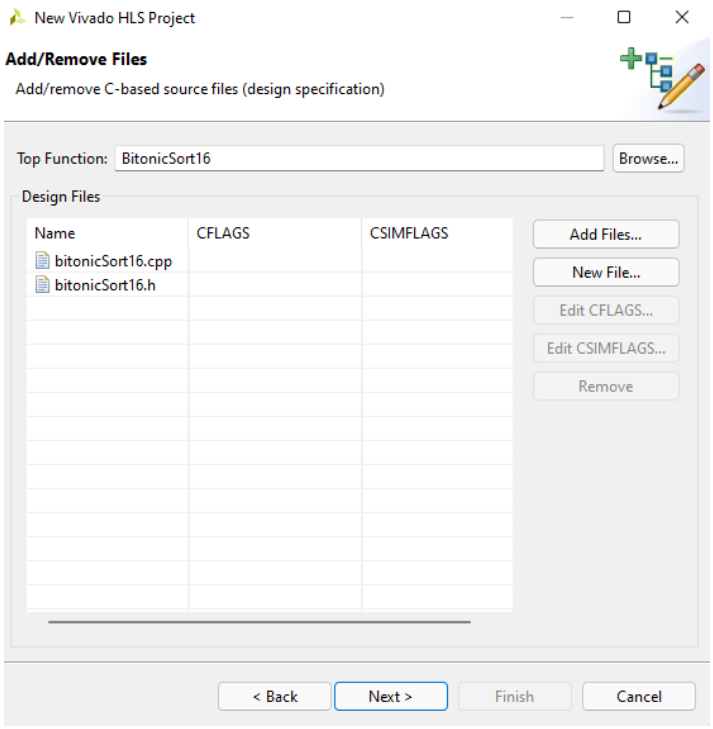
- Please clink on "Create New Project" as shown in the following diagram (red box)



- Provide a project name
  - For example BitonicSort16

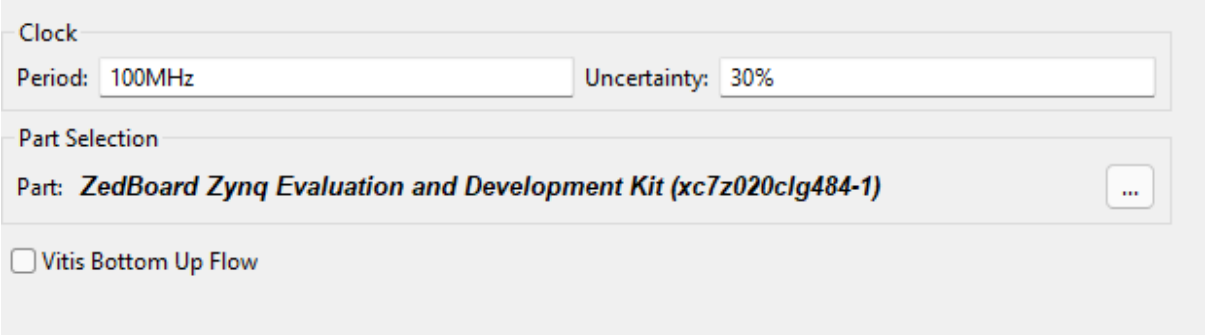


- Click on “Add Files…” and add your main CPP code and header files (please do not add the test bench file here)
- Define and top function for your design by clicking on “Browse…”
- You will observe the following screen



- Click on next and similarly add your test bench *.cpp file
- Click next and you will see the following “Solution Configuration” screen
  - Please specify the following parameters
    - Period (in ns or MHz): 100MHz
    - Uncertainty: 30%
    - Select the device “ZedBoard” by click “…” and then “Device”

- You will observe the following screen

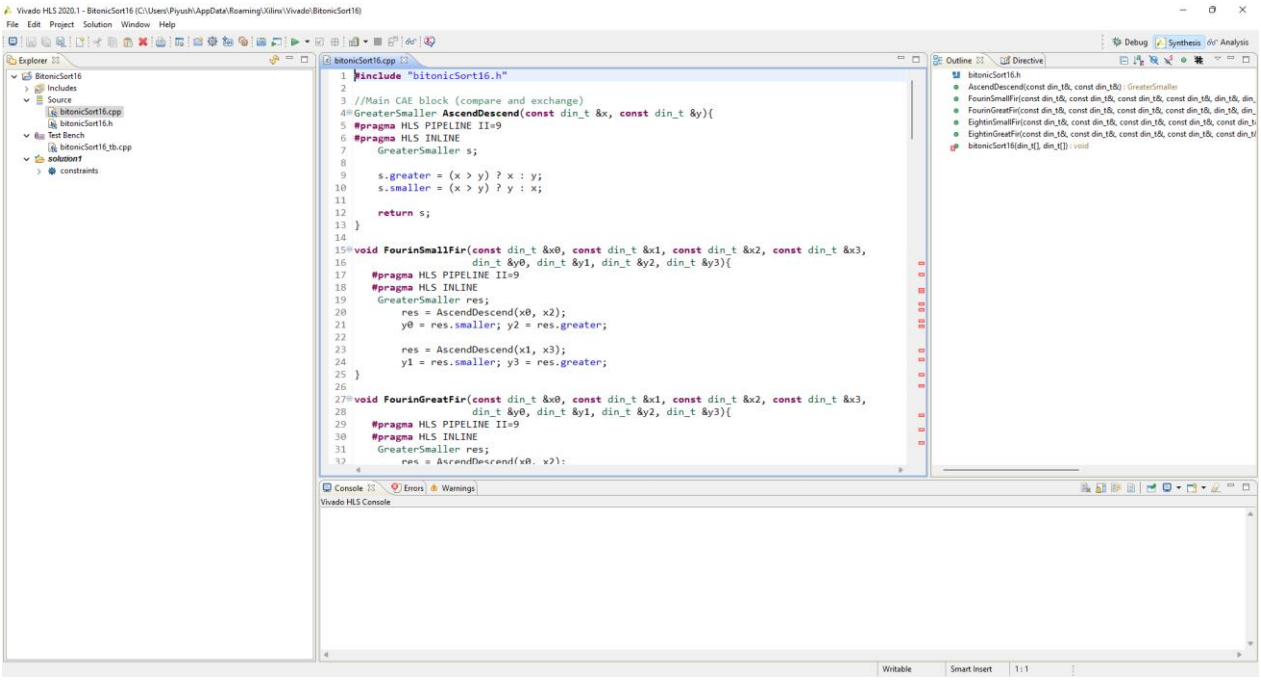


- Click the finish button and you will see the Vivado-HLS main window
  - In the “Explorer” section open the following ribbons

- Source
- Test Bench
- Solution1

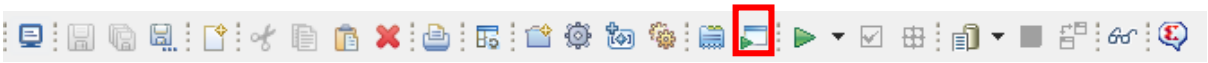
  - Open the bitonicSort16.cpp code

- You will observe the following screen (Please zoom this PDF and confirm)

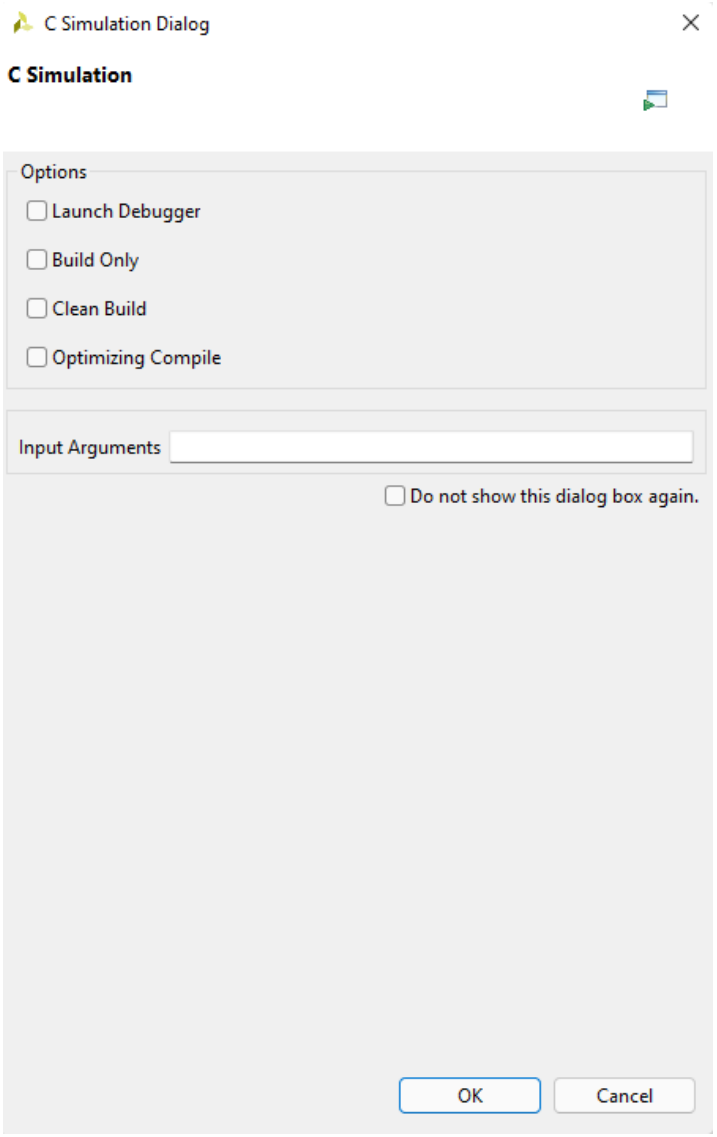


Simulation: Checking the output integrity through C++ simulation

- Please click on the following button to run the C-simulation (red box)

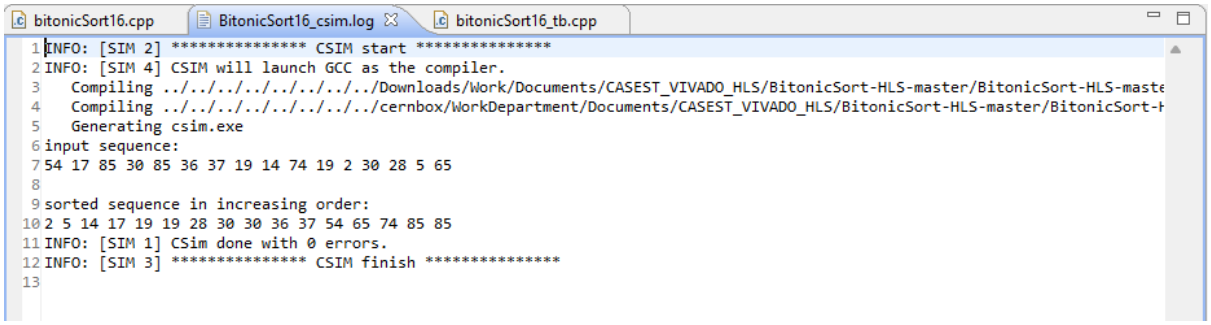


- Click on OK (please check the other option in Xilinx HLS user guide and keep it handy for your convenience) (https://docs.xilinx.com/v/u/en-US/ug902-vivado-high-level-synthesis).



- The output is printed and available in the *csim.log file (opens automatically after the simulation)

- Please check the results before moving to the HLS Synthesis



Synthesis: Conversion of C++ based design into HDL (Verilog and VHDL)

- Please click on the following button to run the Synthesis (red box)

- Synthesis is running in background and you can check the “Console” window for the real-time synthesis log. (Very important to observe the design related issue and its impact on HDL conversion).
- Once the synthesis will finish you can check the results as shown in the following figure

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	6.342 ns	3.00 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
4	4	40.000 ns	40.000 ns	5	5	function

Detail

Instance

Loop

Latency, clock estimated (might not be final or accurate) and pipeline interval achieved

Utilization Estimates

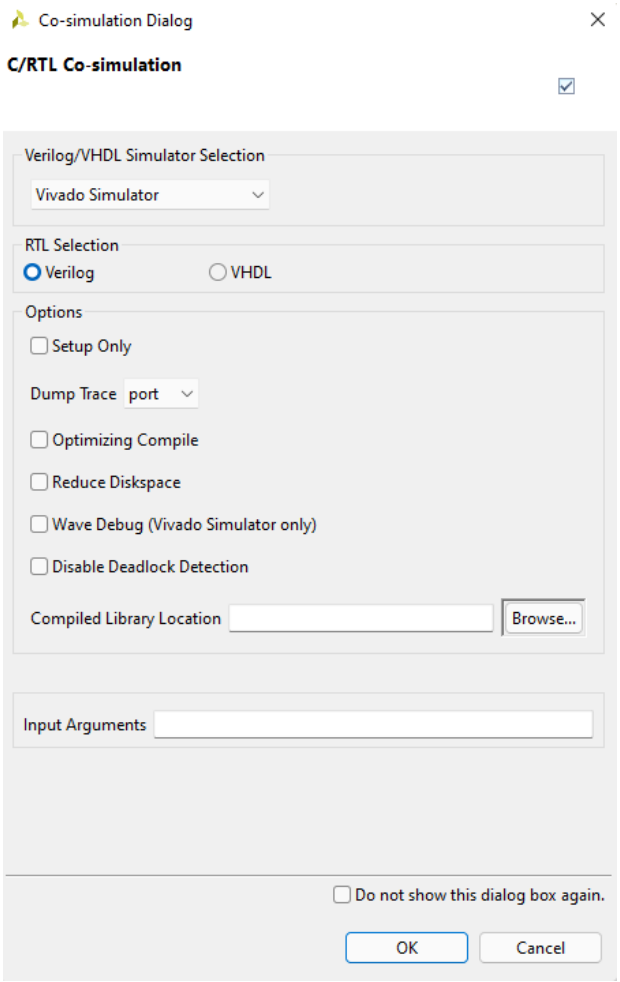
Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	6560	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	33	-
Register	-	-	2053	-	-
Total	0	0	2053	6593	0
Available	280	220	106400	53200	0
Utilization (%)	0	0	1	12	0

Very early estimate of utilization (changes heavily after real synthesis, place and route)

Co-simulation: Verifying the RTL design and checking the waveform in Vivado

- Please click on the following button to run the Co-Simulation (red box) and select the following options



- You will observe the following results and please check and scroll the Console window for detail results.

Cosimulation Report for 'bitonicSort16'

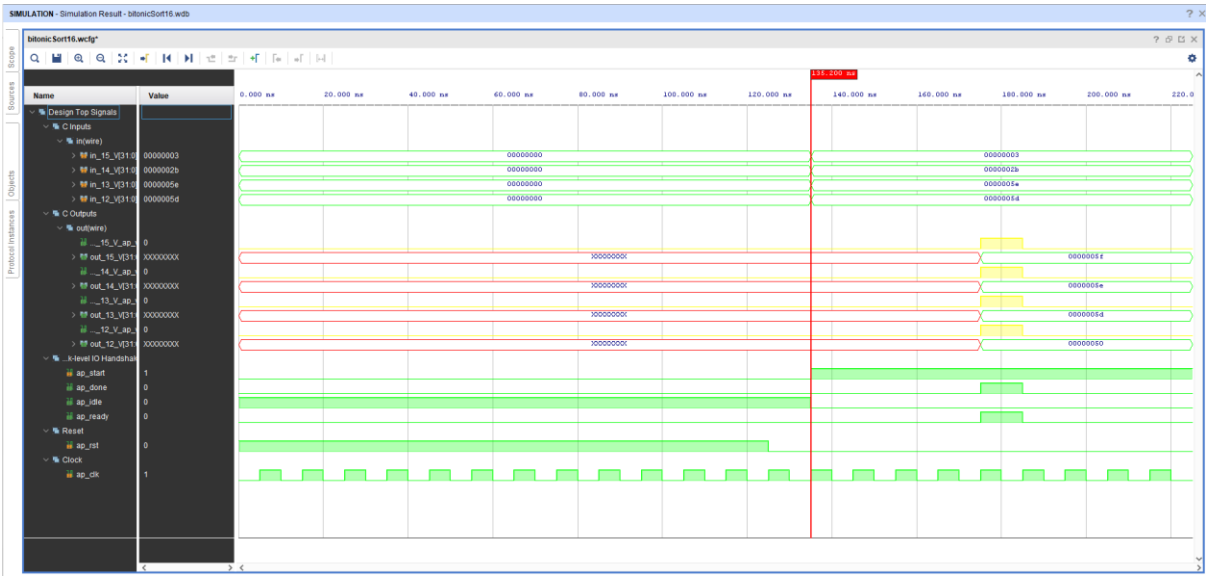
Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	4	4	4	NA	NA	NA

Export the report(.html) using the Export Wizard

- To open the waveform in Vivado please click on the following icon (red box)

- Pop-out the bundled signal (C Inputs, C outputs, Block level IO handshake,..)
- You will see the following screen (for efficient display purpose, few signals are deleted from this waveform, you will observe more input and output signals).

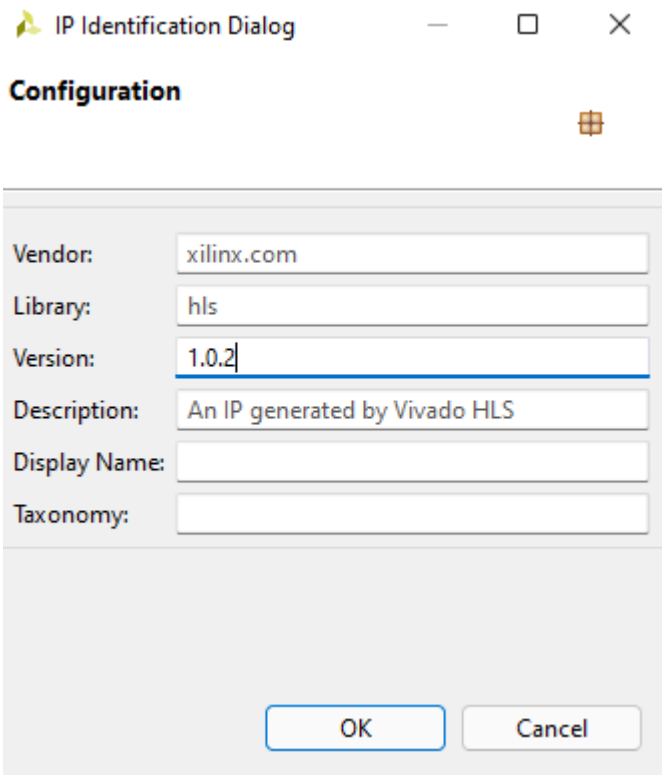


Export IP/RTL: Exporting the final design

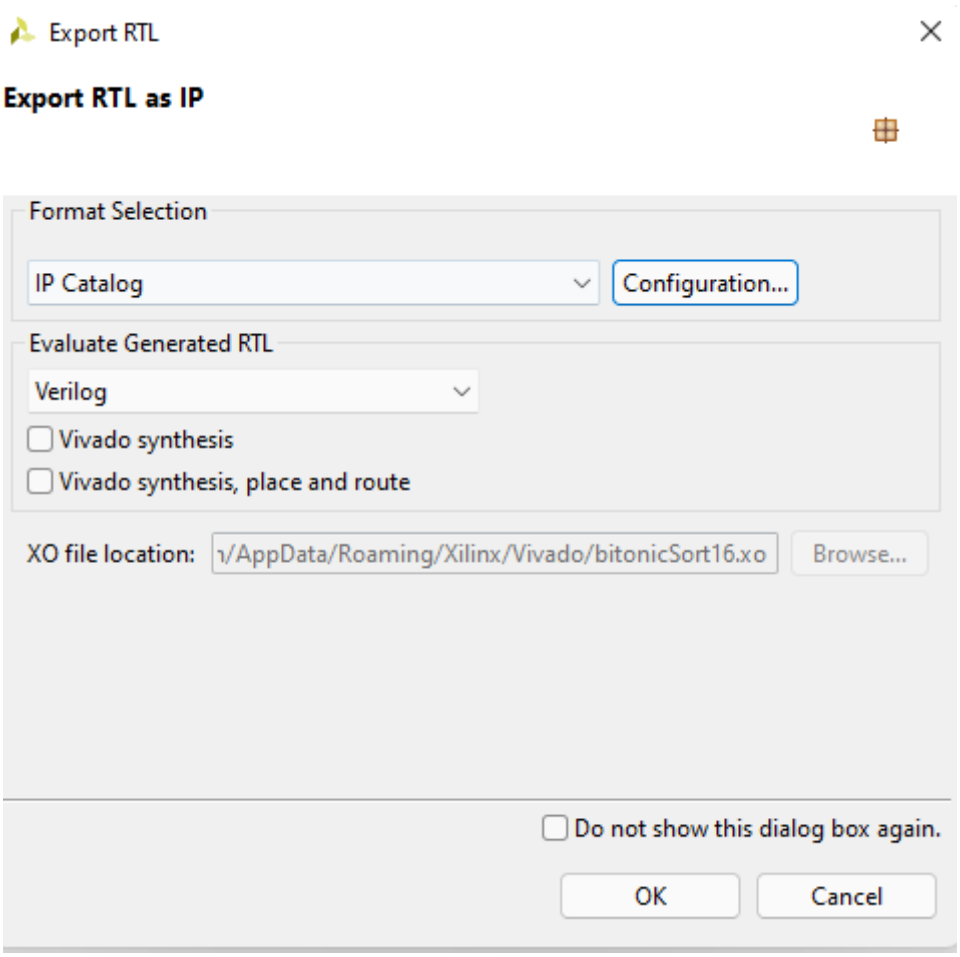
- You can export the design (RTL) in either Verilog or VHDL and in the same time you can evaluate your design in Vivado
  - Running the Vivado Synthesis, Place and Route
  - Optional, but recommended for fairly complex design
  - It will provide the real utilization and the max clock your circuit can afford to run without facing any data integrity issue (negative setup and hold slack)
  - Caveat: Vivado run the synthesis, place and route purely based on the clock information provided during the HLS synthesis and it will neglect the IO PIN constraints.

- Please click on the following button to run the Export-RTL (red box)

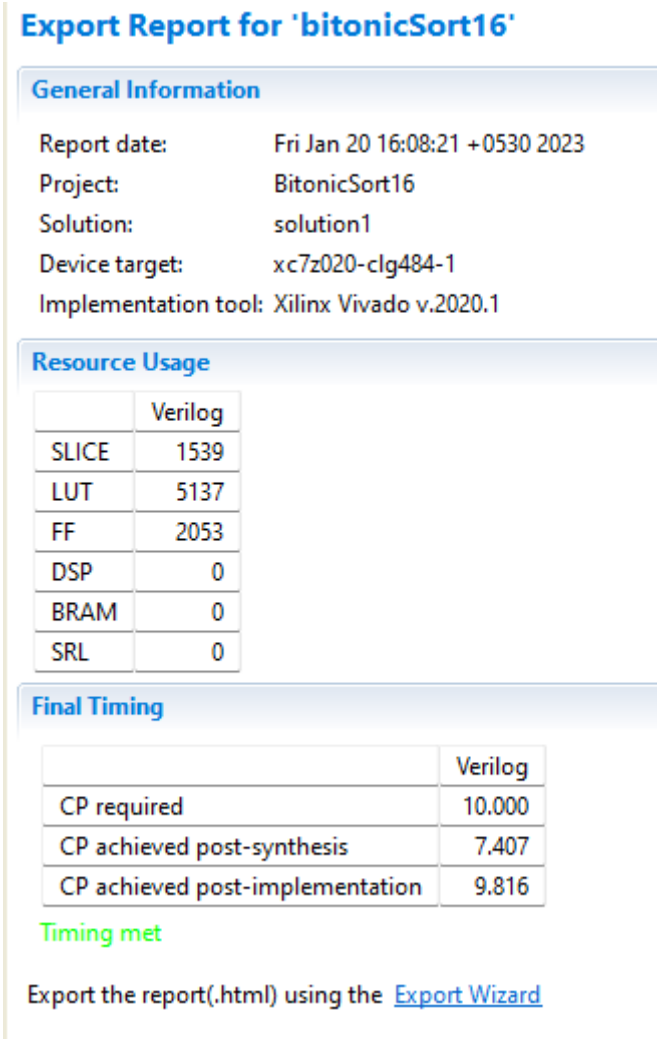
- In the Export-RTL dialogue box click on the “Configuration…” and fill following parameters in Version row (as shown in the figure below):
  - Version: 1.0.2
  - Click OK
- Please click on the tickbox of
  - Vivado synthesis, place and route.



- You will see the following screen and click on OK to proceed for Export RTL



- Based on the complexity of the design Vivado will generate the results and you will see the following window after it completes the export and evaluation.



Export Report for 'bitonicSort16'

General Information

Report date:	Fri Jan 20 16:08:21 +0530 2023
Project:	BitonicSort16
Solution:	solution1
Device target:	xc7z020-clg484-1
Implementation tool:	Xilinx Vivado v.2020.1

Resource Usage

	Verilog
SLICE	1539
LUT	5137
FF	2053
DSP	0
BRAM	0
SRL	0

Final Timing

	Verilog
CP required	10.000
CP achieved post-synthesis	7.407
CP achieved post-implementation	9.816

Timing met

Export the report(.html) using the Export Wizard

- For any other Vivado-HLS tool and designing related queries please refer to the UG902 user guide.
- https://docs.xilinx.com/v/u/en-US/ug902-vivado-high-level-synthesis

Thanks