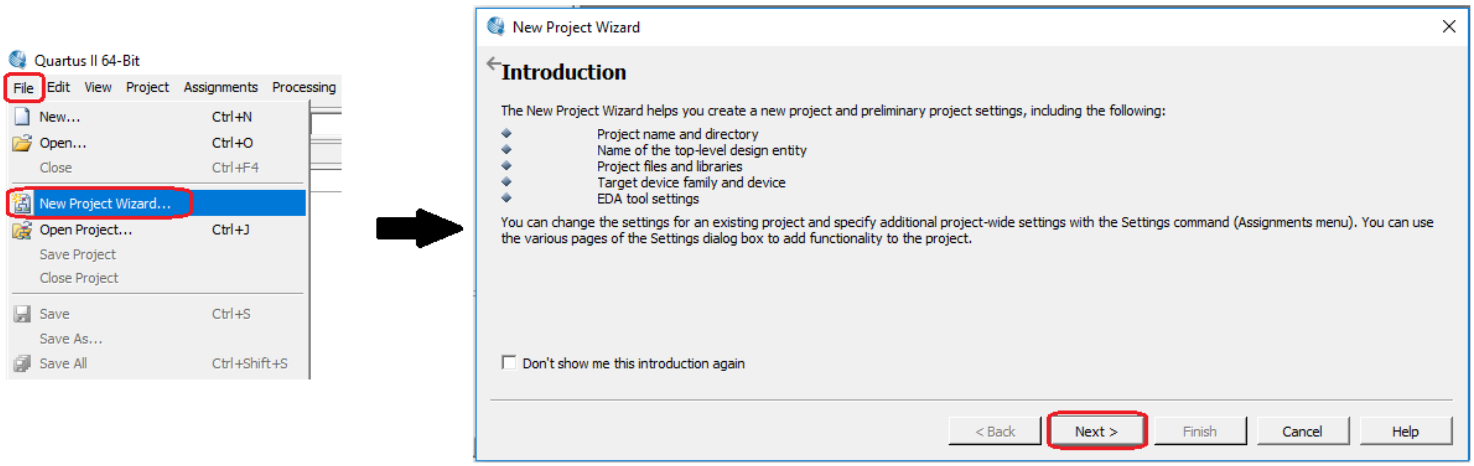
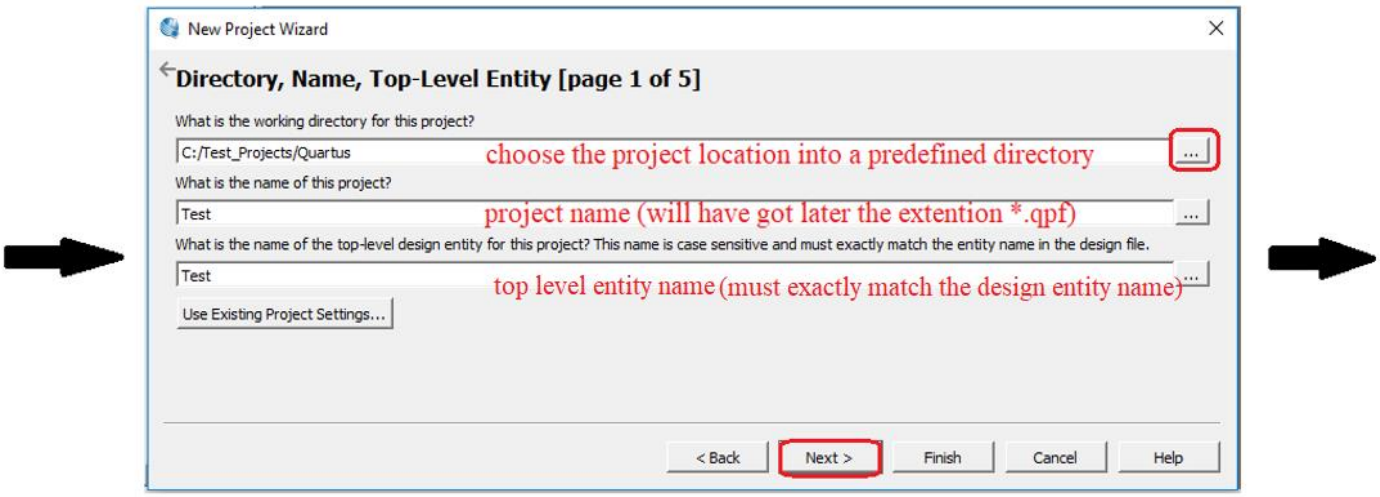
Table of Contents

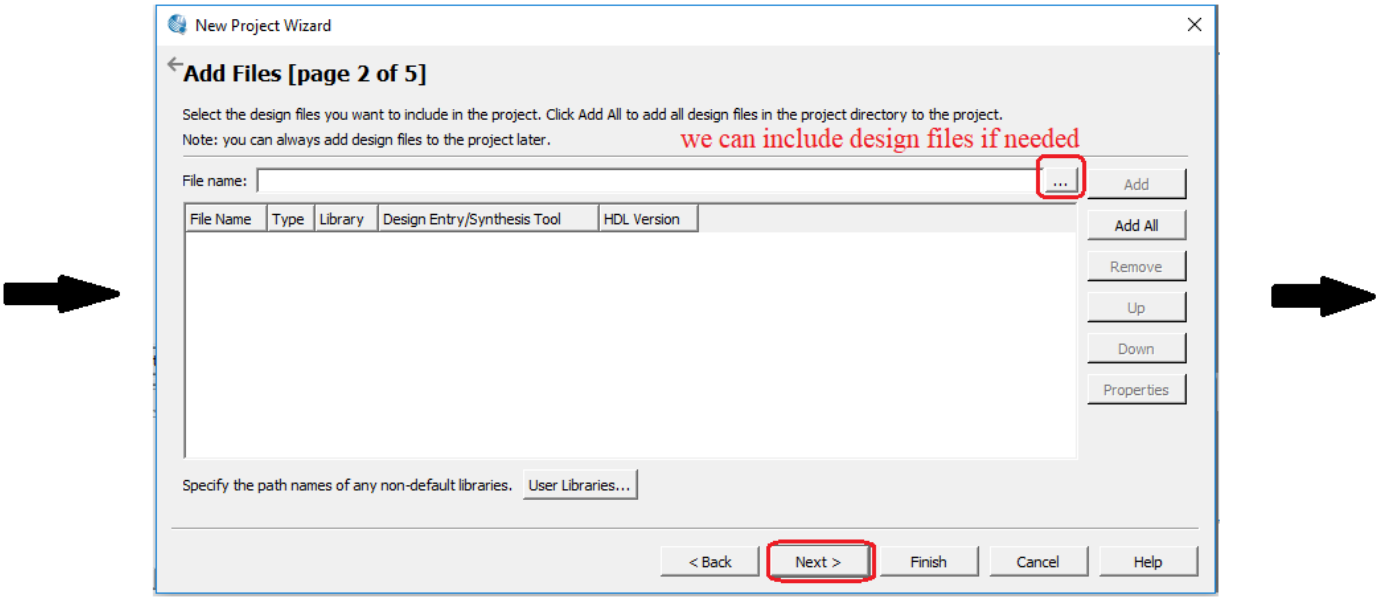
Quartus - Create or Open a Project

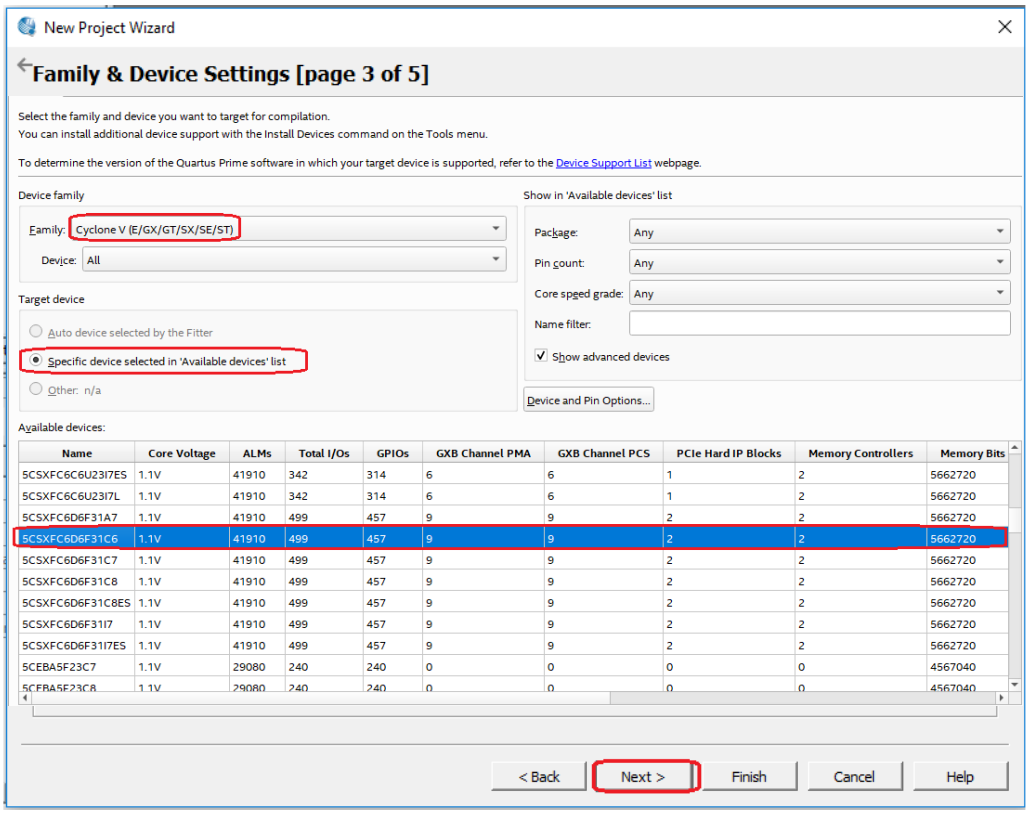
A. Create a New Project:

1. Step 1 – create a new project and add VHDL files:



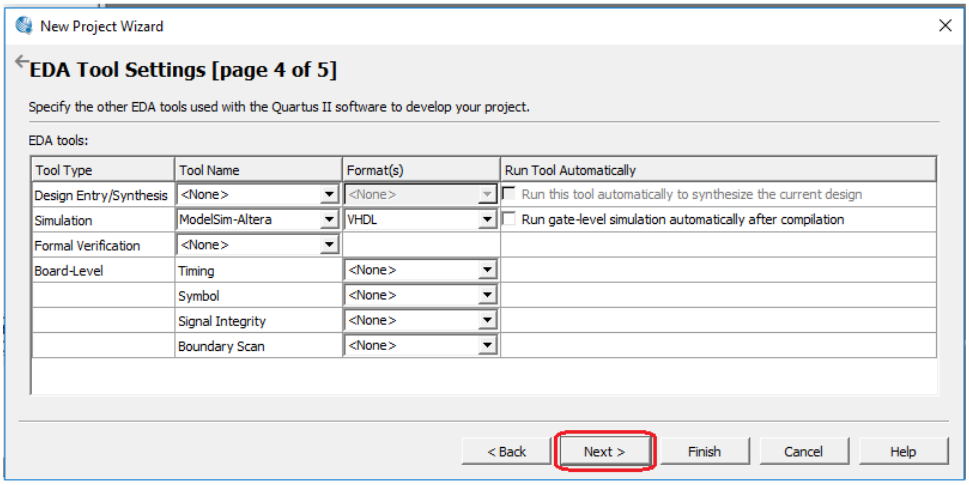


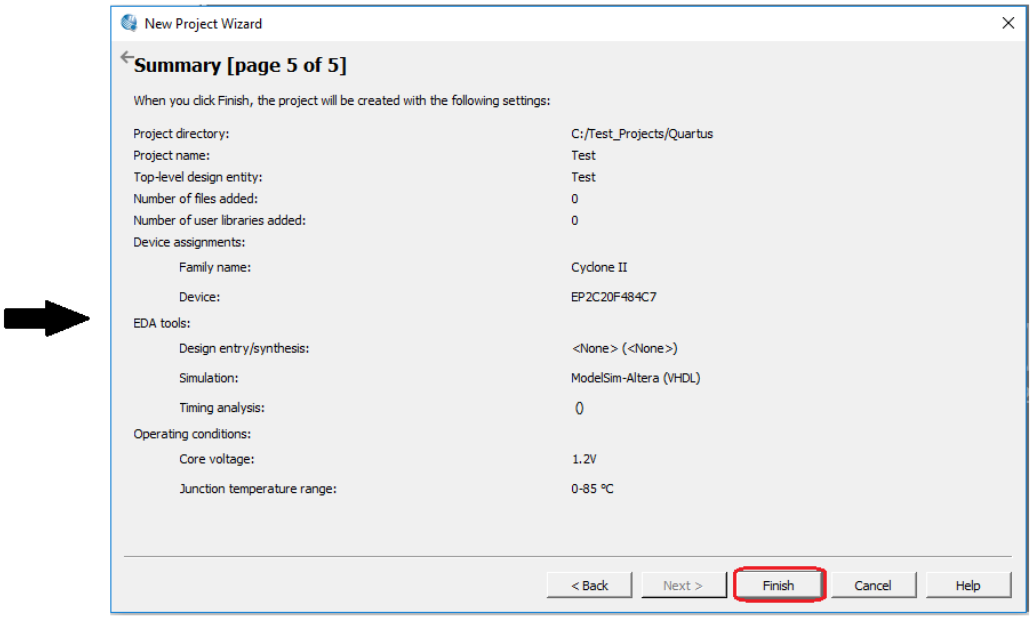




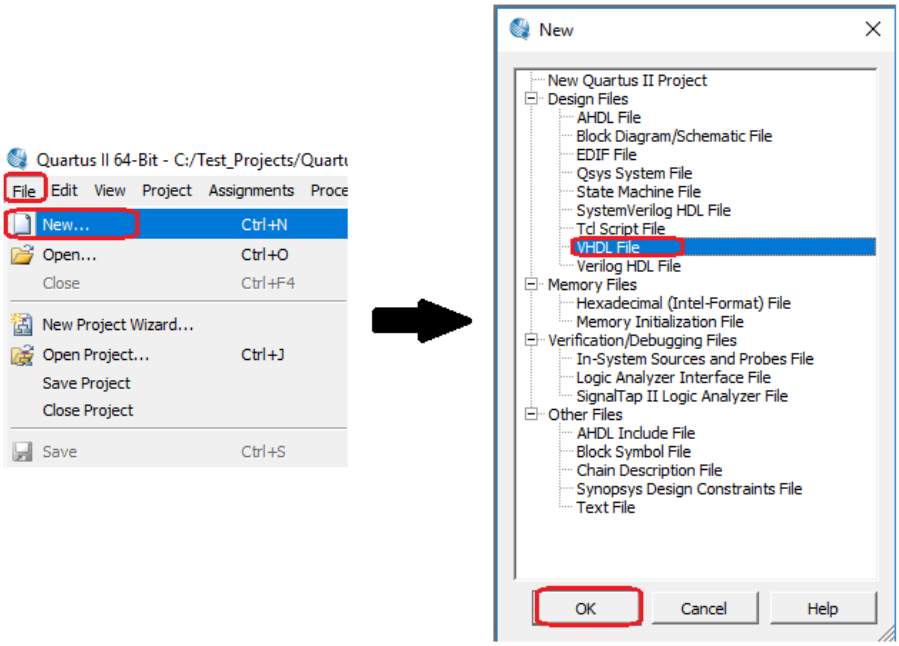
Zoom in

5CSXFC6D6F31C6

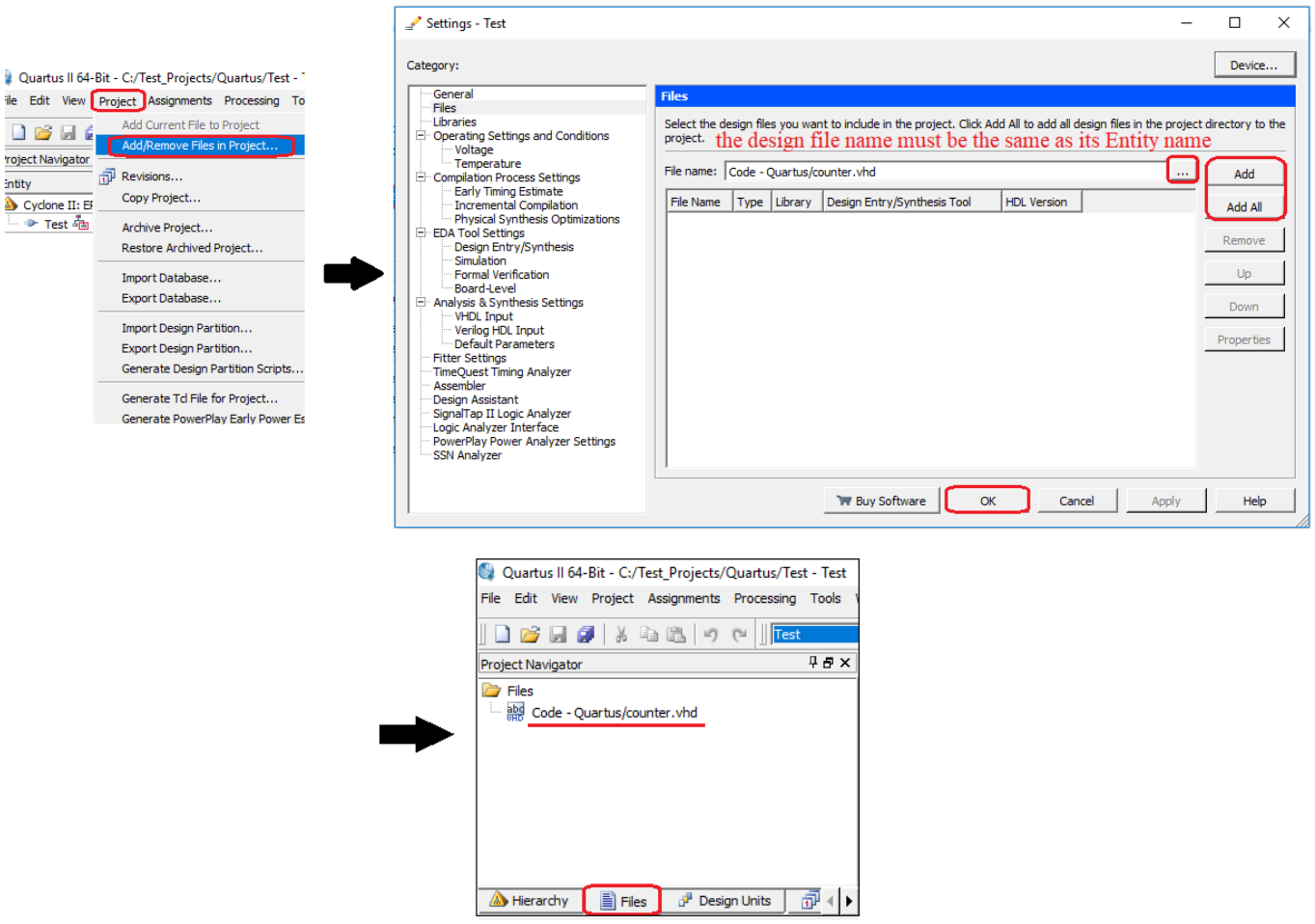




2. Step 2 – Add project files:
   a. In order to open VHDL blank file use the next step (if you're using VHDL existing files, copy these files into project folder and skip to clause b):

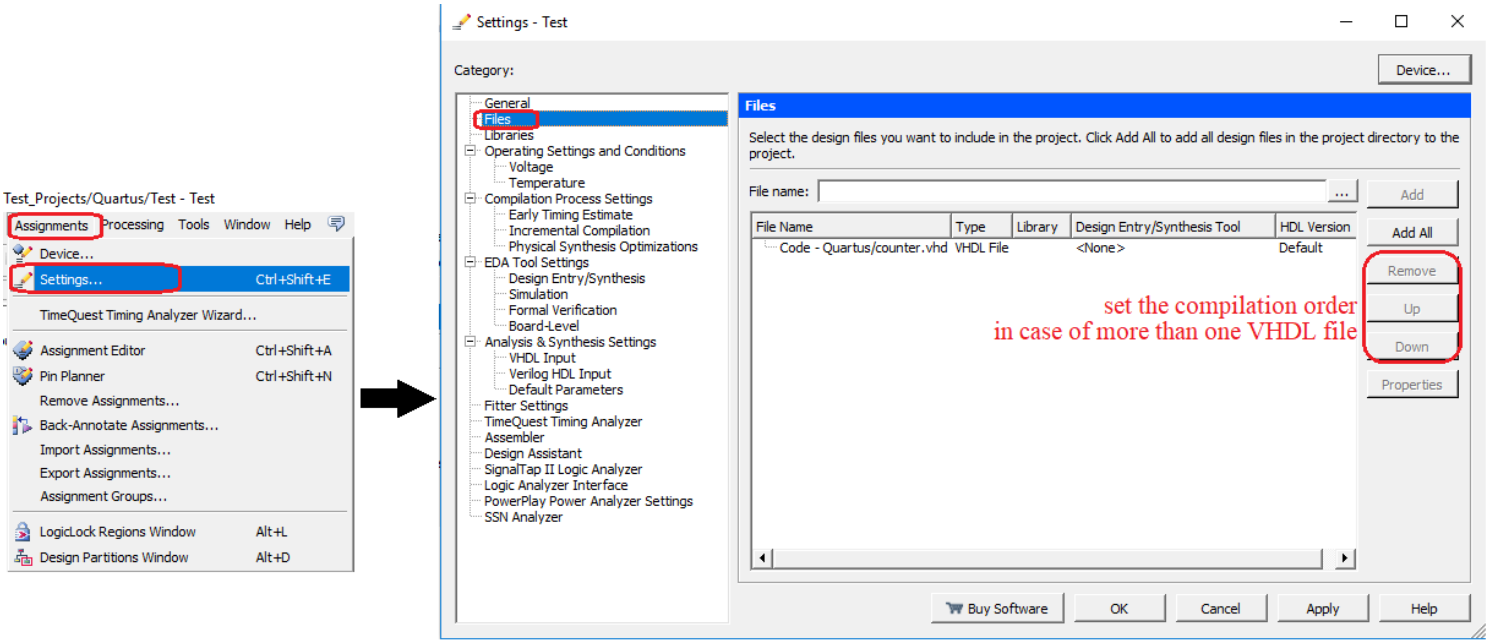


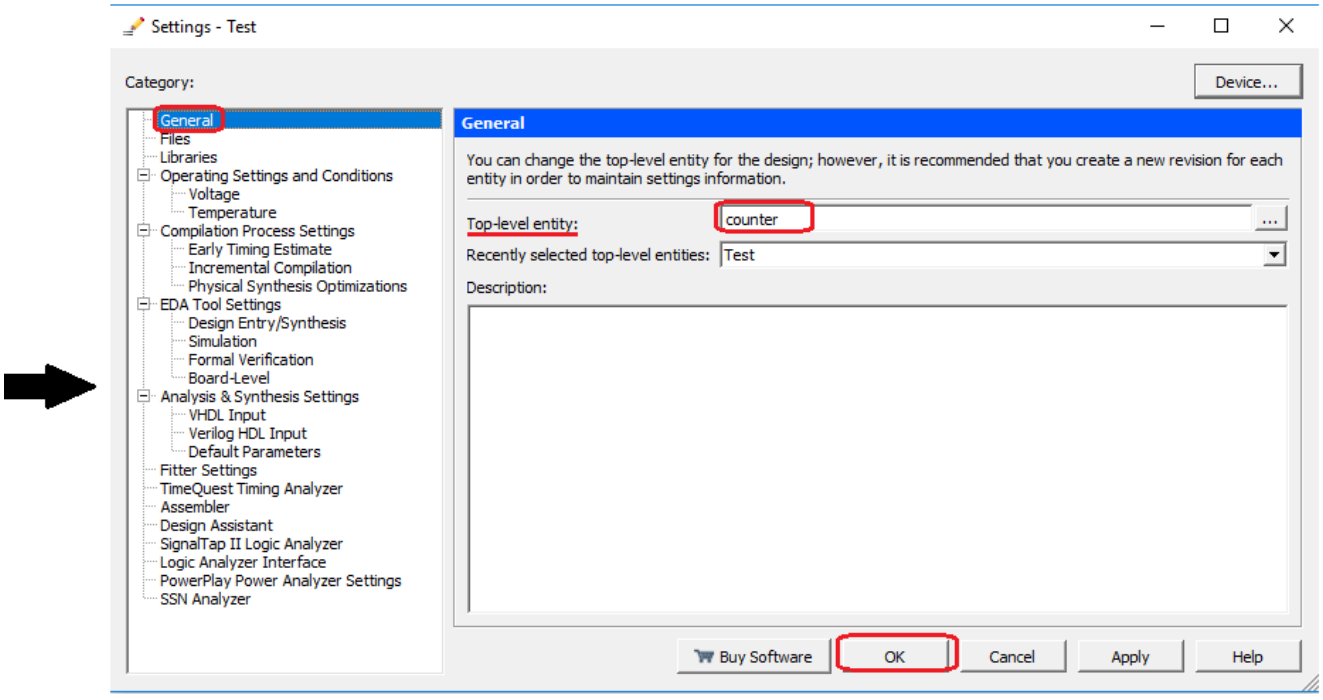
   b. Add the project VHDL existing files:



3. Step 3 – Code compilation:

   a. Set the compilation order and top level entity:





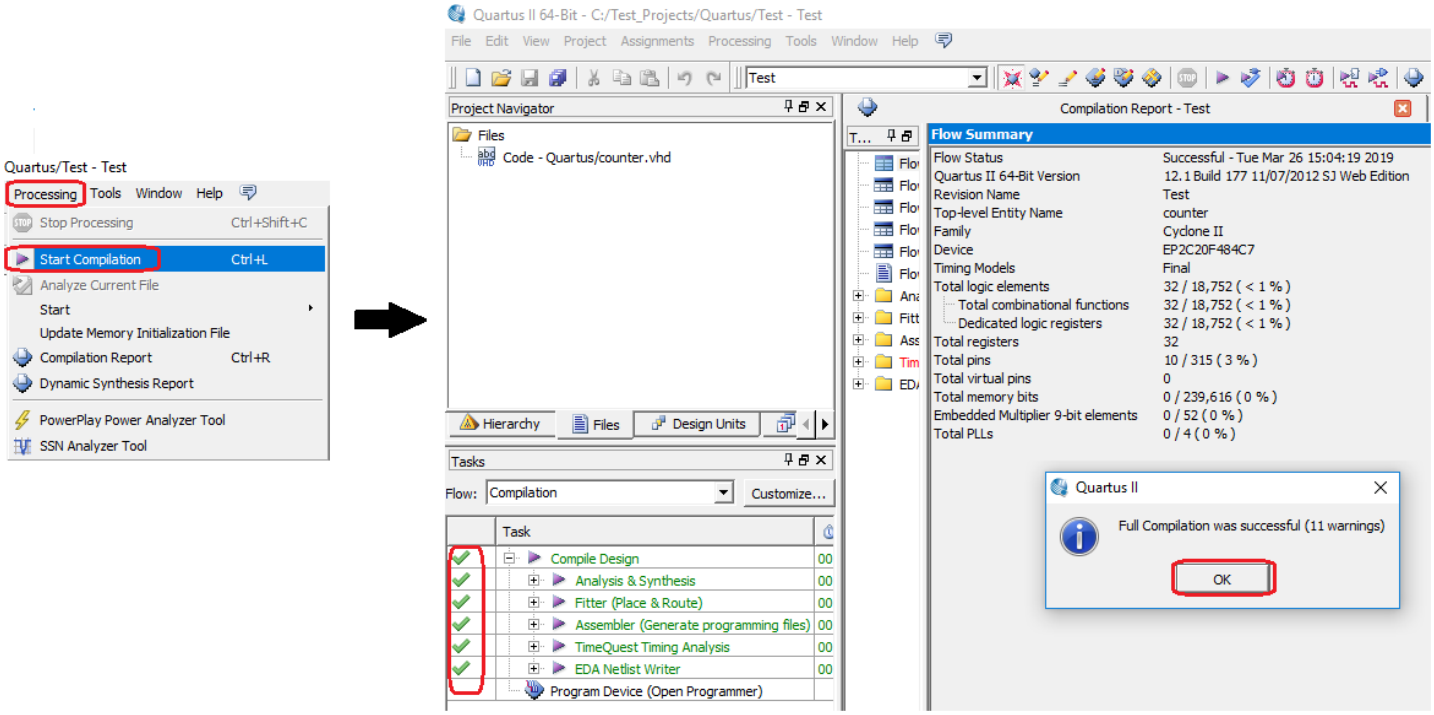
counter.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity counter is port (
    clk,enable : in std_logic;
    q          : out std_logic_vector (7 downto 0));
end counter;

architecture rtl of counter is
    signal q_int : std_logic_vector (31 downto 0):=x"00000000";
begin
    process (clk)
    begin
        if (rising_edge(clk)) then
           if enable = '1' then
                q_int <= q_int + 1;
           end if;
         end if;
    end process;
    q <= q_int(31 downto 24); -- Output only 8MSB
end rtl;
```

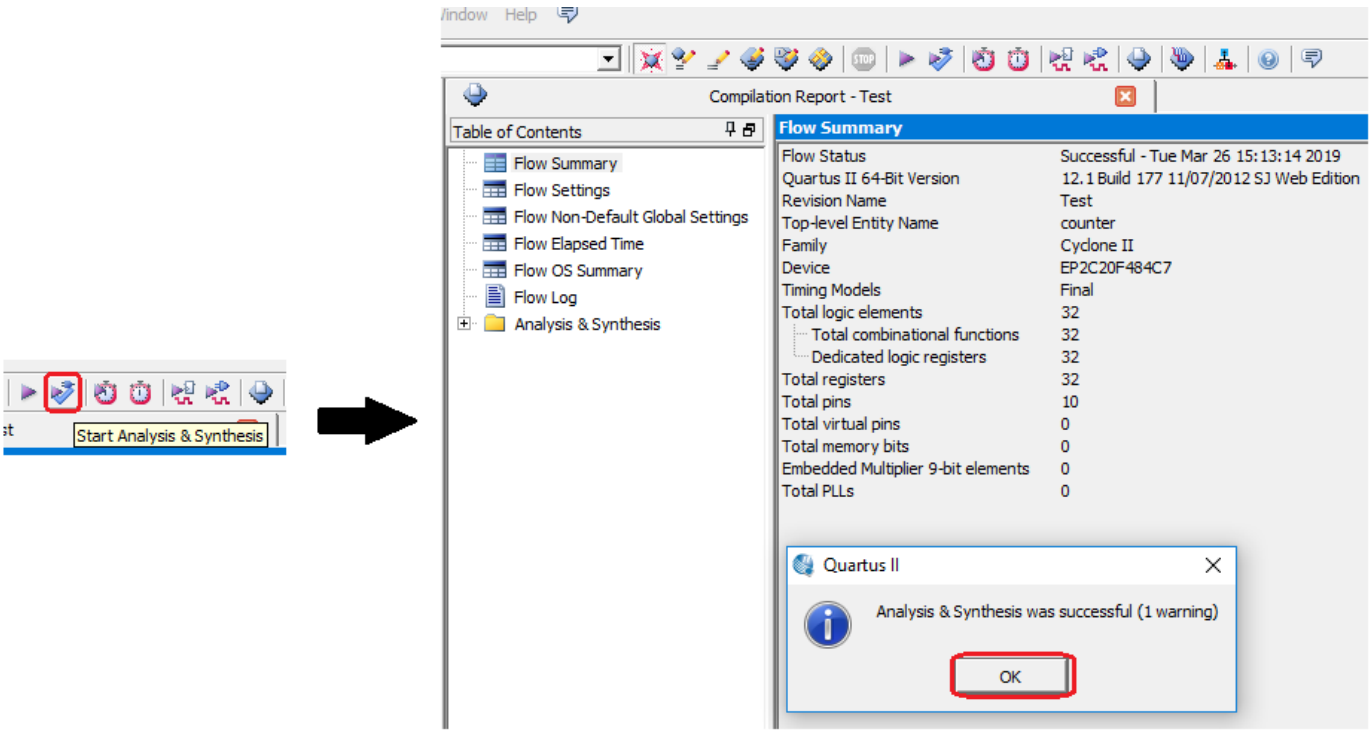
Example application:

- 32bit behavioral counter with enable
- 8-bit MSB of counter output are connected to red LEDs (LEDR7-LEDR0)
- Enable connected to switch SW0
- 50MHz on board clock named CLOCK_50

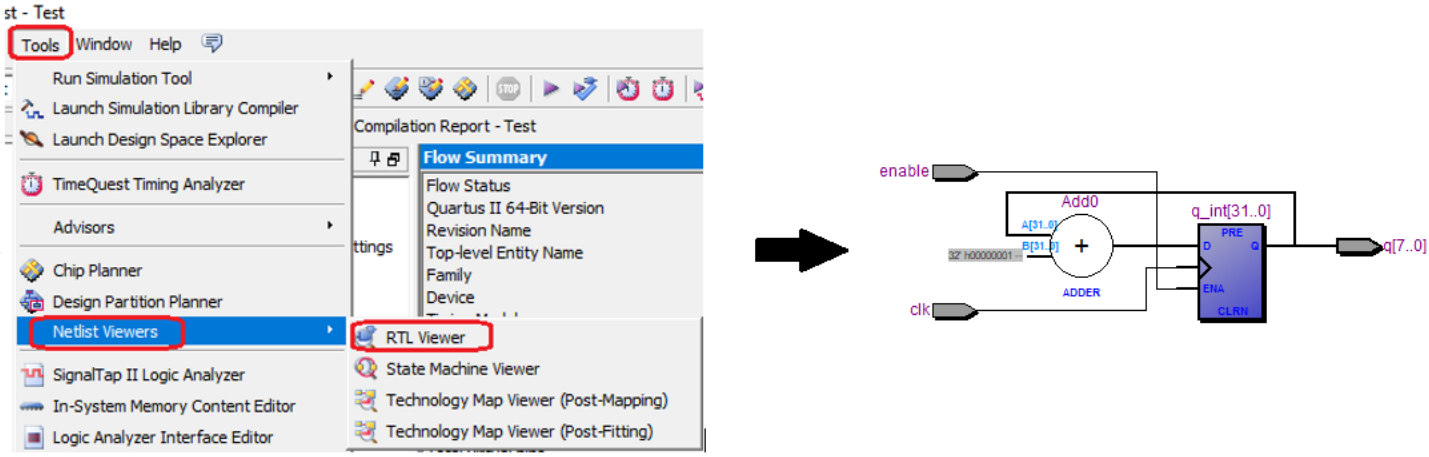
b. Code compilation:



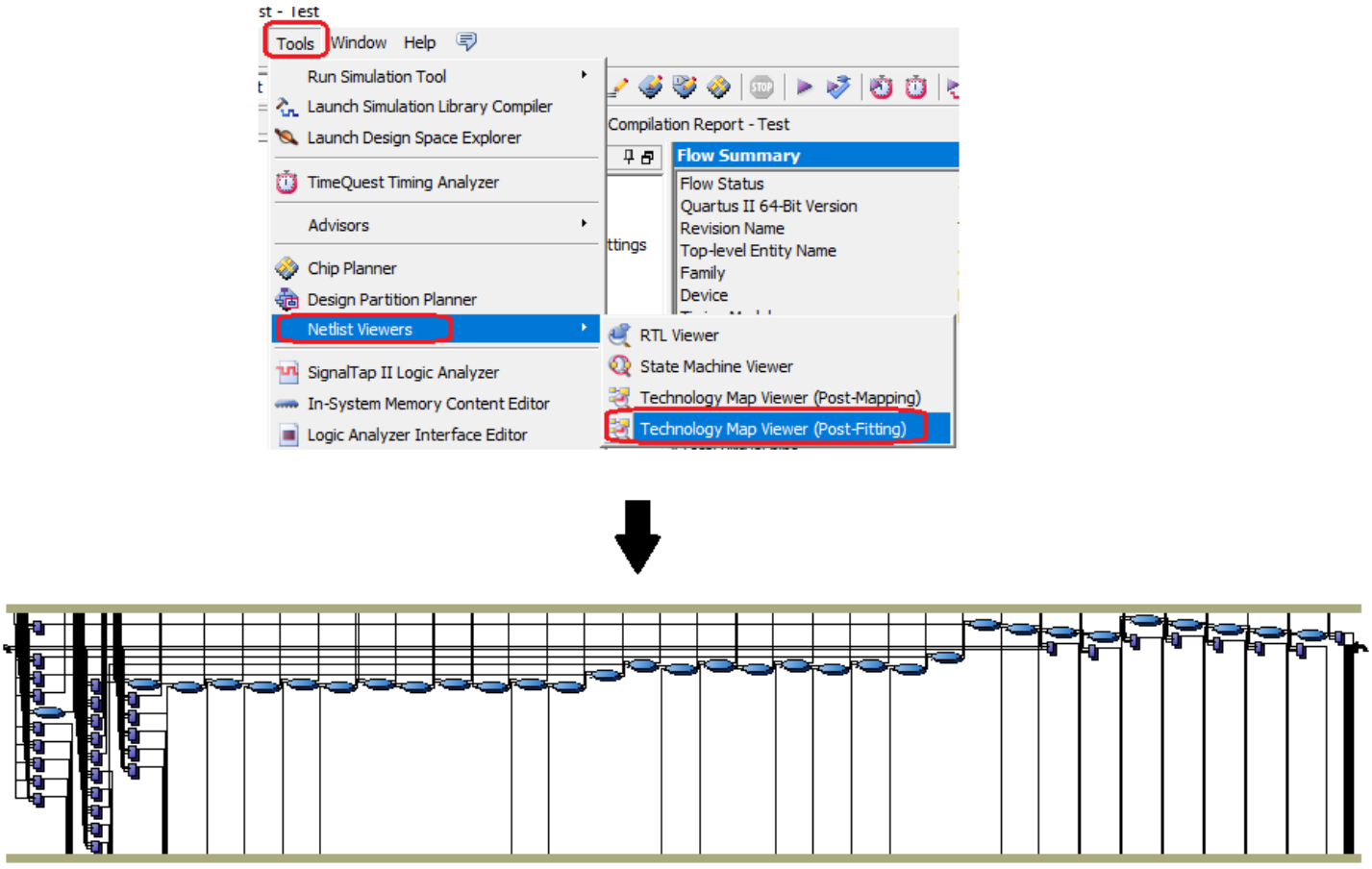
c. Start analysis and synthesis:



4. Step 4 – Synthesis results:
   a. Synthesis **RTL** viewer:

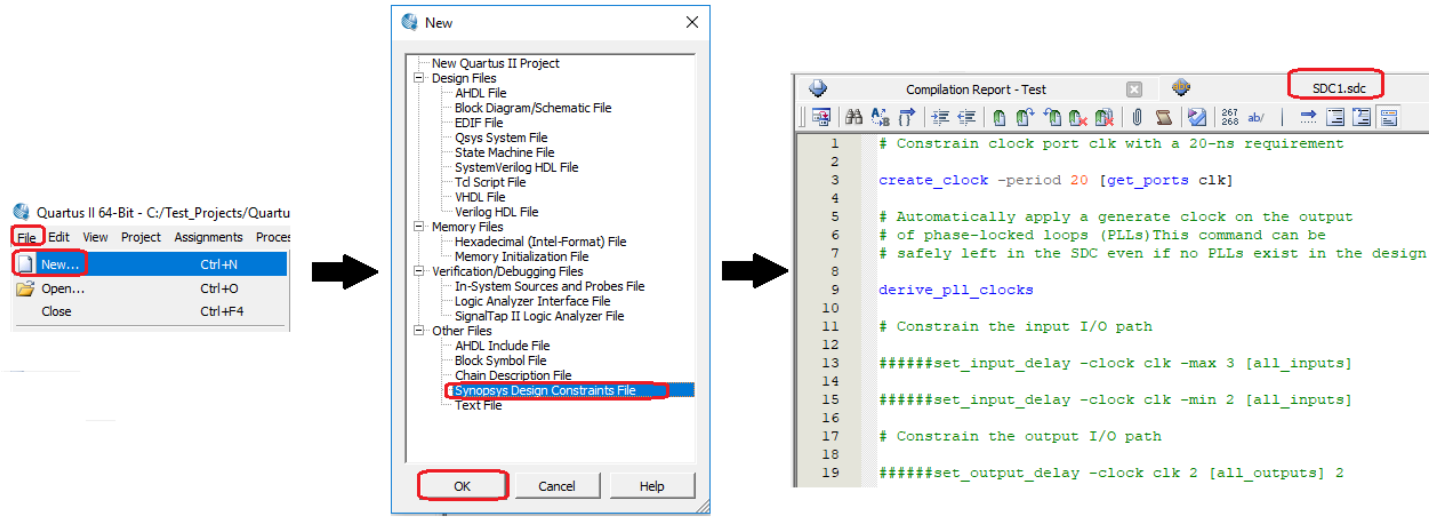


   b. Synthesis **Map** (Post-Fitting) viewer (LEs and FFs combination):

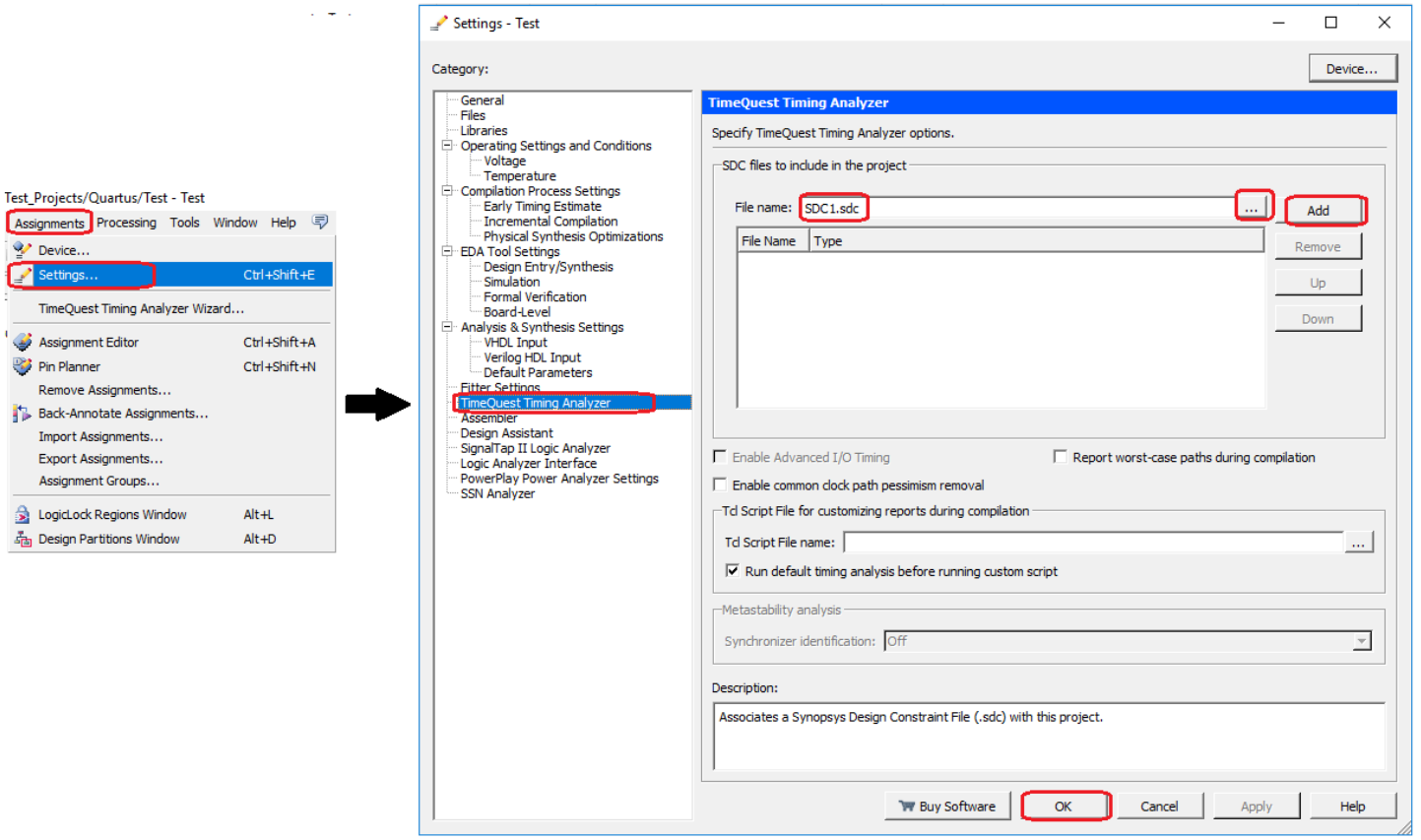


5. Step 5 – Setting system constrains:

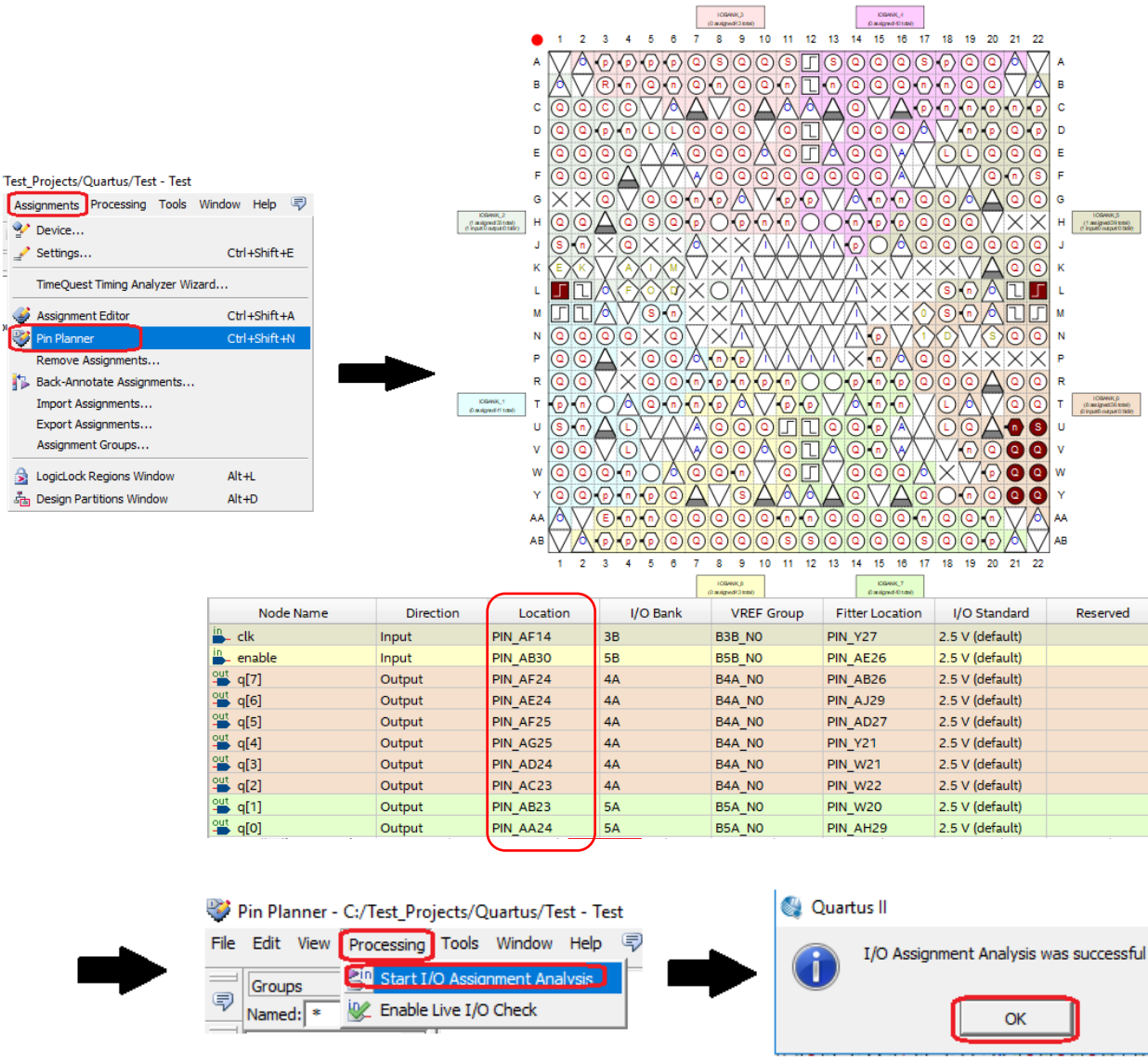
a. Create the system constrains *.sdc file:



b. Add the *.sdc file to the project:



6. Step 6 – Pin Planner Layout (See DE1-Standard user manual on pages 24-26):



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate
clk	Input	PIN_AF14	3B	B3B_N0	PIN_Y27	2.5 V (default)		12mA (default)	
enable	Input	PIN_AB30	5B	B5B_N0	PIN_AE26	2.5 V (default)		12mA (default)	
q[7]	Output	PIN_AF24	4A	B4A_N0	PIN_AB26	2.5 V (default)		12mA (default)	1 (default)
q[6]	Output	PIN_AE24	4A	B4A_N0	PIN_AJ29	2.5 V (default)		12mA (default)	1 (default)
q[5]	Output	PIN_AF25	4A	B4A_N0	PIN_AD27	2.5 V (default)		12mA (default)	1 (default)
q[4]	Output	PIN_AG25	4A	B4A_N0	PIN_Y21	2.5 V (default)		12mA (default)	1 (default)
q[3]	Output	PIN_AD24	4A	B4A_N0	PIN_W21	2.5 V (default)		12mA (default)	1 (default)
q[2]	Output	PIN_AC23	4A	B4A_N0	PIN_W22	2.5 V (default)		12mA (default)	1 (default)
q[1]	Output	PIN_AB23	5A	B5A_N0	PIN_W20	2.5 V (default)		12mA (default)	1 (default)
q[0]	Output	PIN_AA24	5A	B5A_N0	PIN_AH29	2.5 V (default)		12mA (default)	1 (default)

Table 3-5 Pin Assignment of Clock Inputs

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
CLOCK_50	PIN_AF14	50 MHz clock input	3.3V

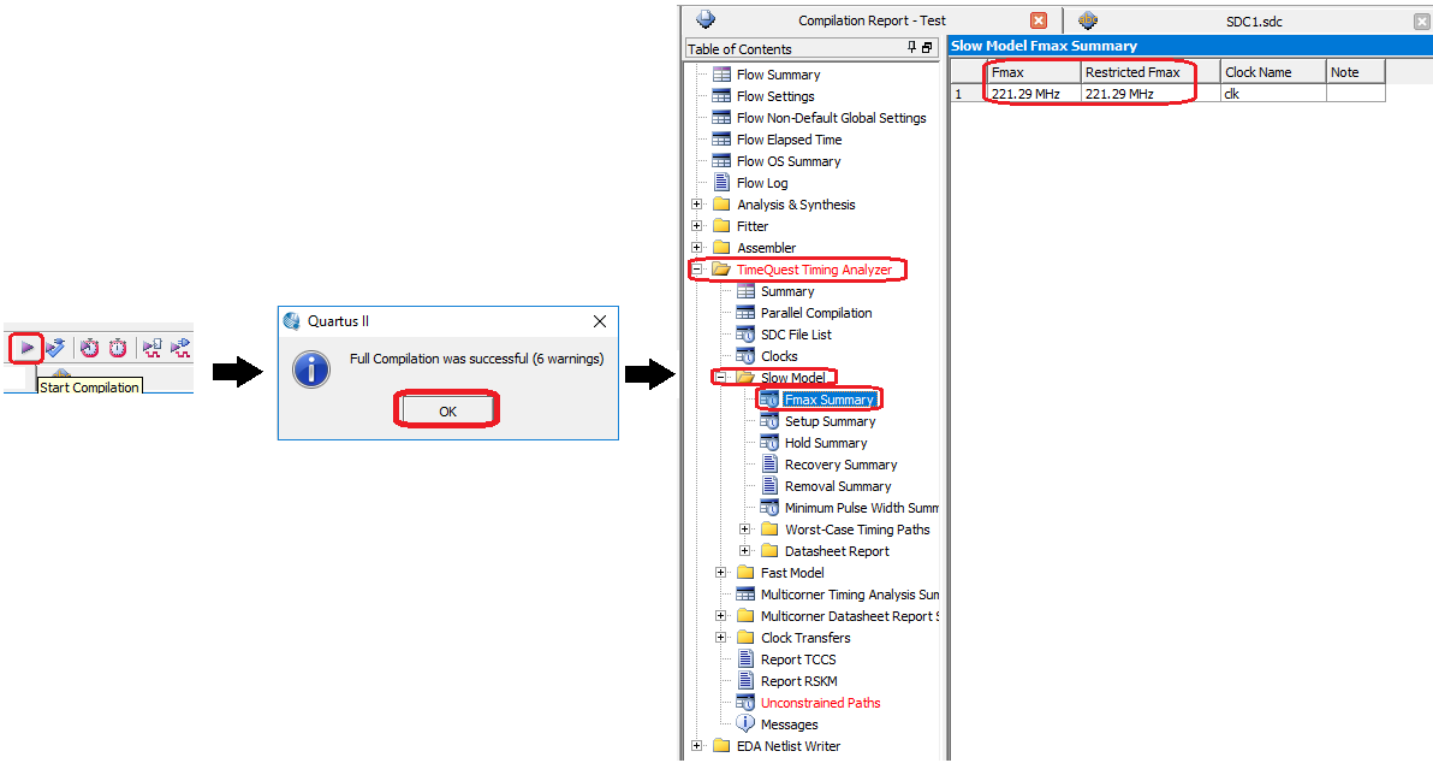
Table 3-6 Pin Assignment of Slide Switches

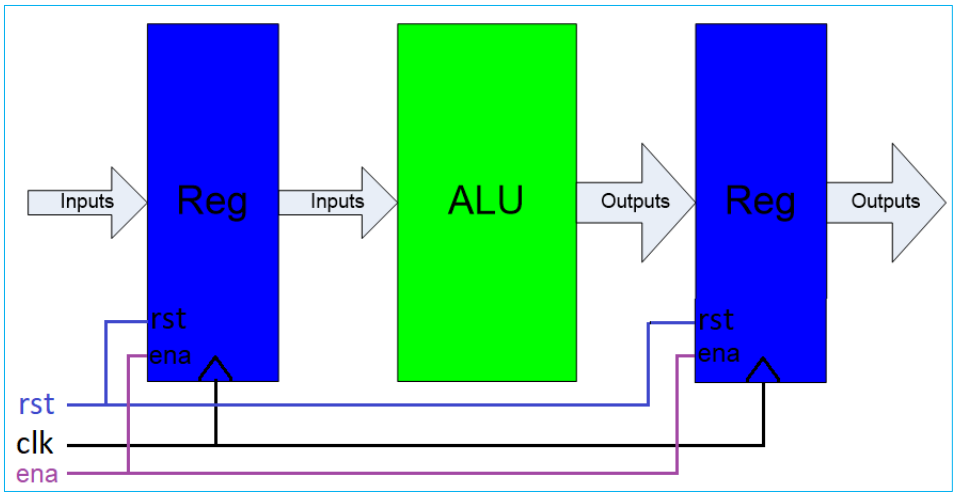
Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SW[0]	PIN_AB30	Slide Switch[0]	Depend on JP3

Table 3-8 Pin Assignment of LEDs

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
LEDR[0]	PIN_AA24	LED [0]	3.3V
LEDR[1]	PIN_AB23	LED [1]	3.3V
LEDR[2]	PIN_AC23	LED [2]	3.3V
LEDR[3]	PIN_AD24	LED [3]	3.3V
LEDR[4]	PIN_AG25	LED [4]	3.3V
LEDR[5]	PIN_AF25	LED [5]	3.3V
LEDR[6]	PIN_AE24	LED [6]	3.3V
LEDR[7]	PIN_AF24	LED [7]	3.3V

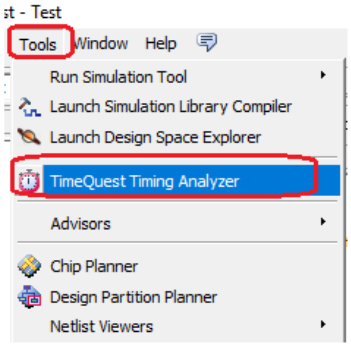
7. Step 7 – Full compilation (finding of f_{max}):

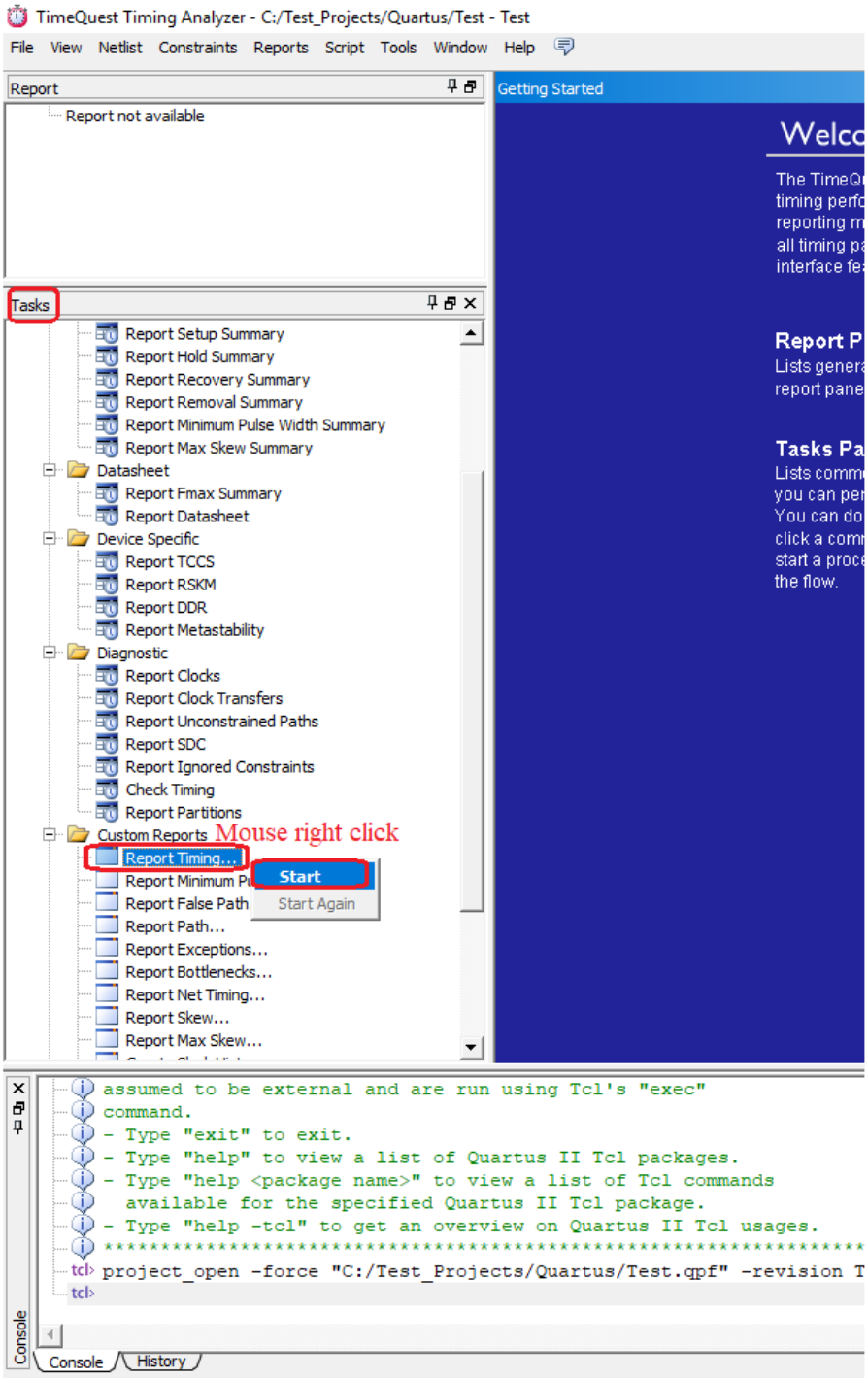


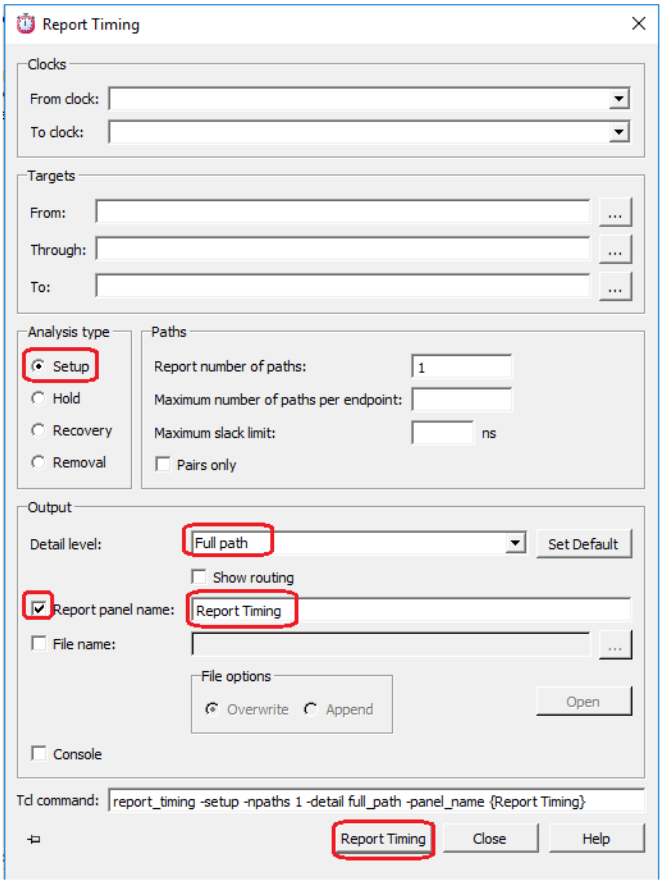


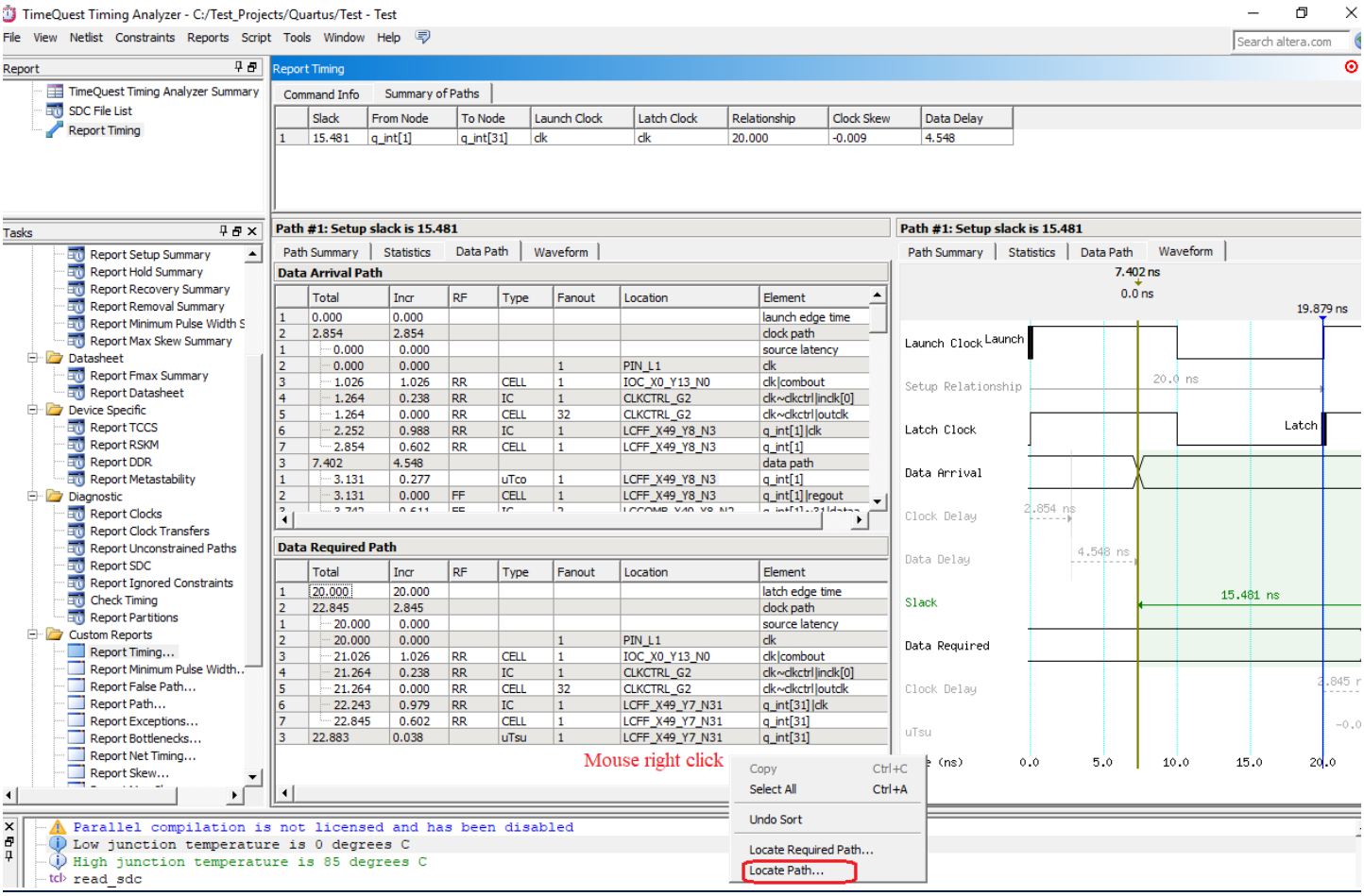
Explanation: In order Quartus IDE can calculate f_{max} the logic design parts must be wrapped by registers

8. Step 8 – Finding Critical Path Location:

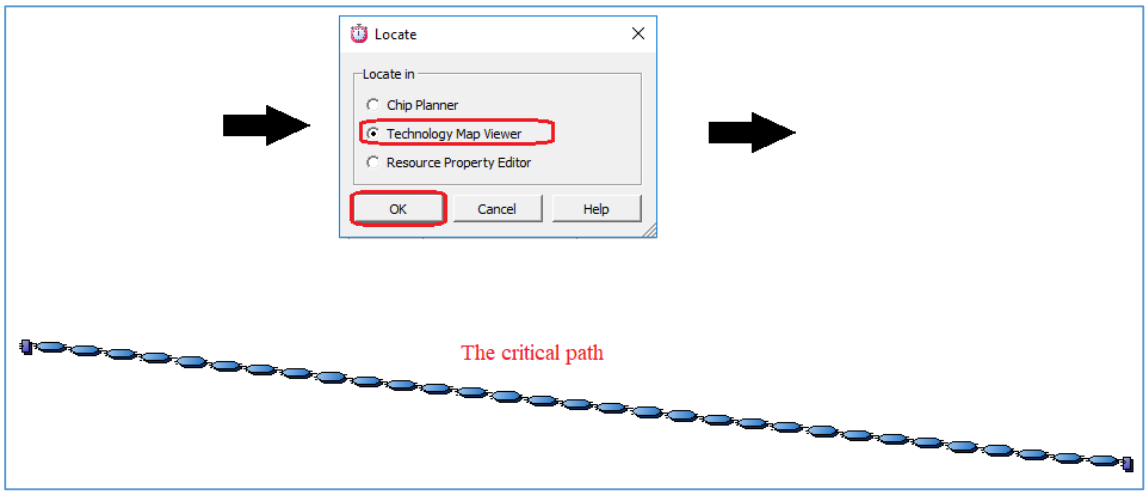




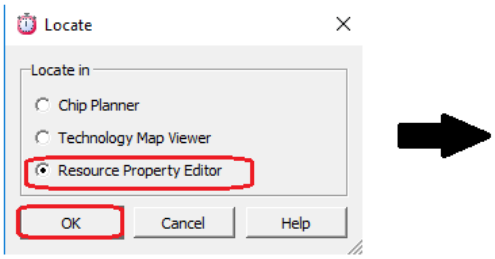

Report Timing
Clocks
From clock:
To clock:
Targets
From:
Through:
To:
Analysis type
Setup
Hold
Recovery
Removal
Paths
Report number of paths: 1
Maximum number of paths per endpoint:
Maximum slack limit: ns
Pairs only
Output
Detail level: Full path
Set Default
Show routing
Report panel name: Report Timing
File name:
File options
Overwrite
Append
Open
Console
Tcl command: report_timing -setup -npaths 1 -detail full_path -panel_name {Report Timing}
Report Timing
Close
Help

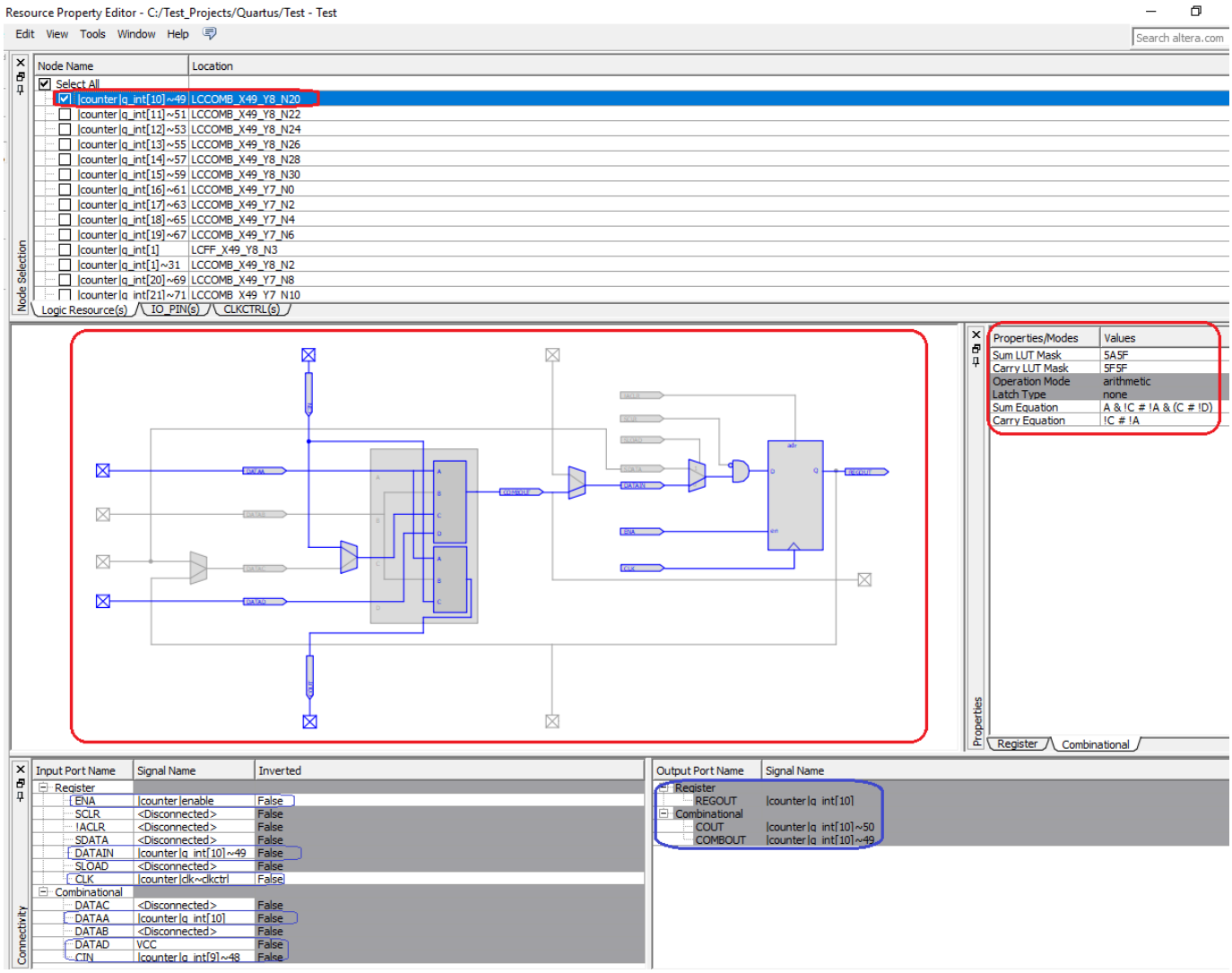

TimeQuest Timing Analyzer - C:/Test_Projects/Quartus/Test - Test
File View Netlist Constraints Reports Script Tools Window Help
Report
TimeQuest Timing Analyzer Summary
SDC File List
Report Timing
Tasks
Report Timing
Command Info
Summary of Paths
Slack From Node To Node Launch Clock Latch Clock Relationship Clock Skew Data Delay
15.481 q_int[1] q_int[31] clk clk 20.000 -0.009 4.548
Path #1: Setup slack is 15.481
Path Summary Statistics Data Path Waveform
Data Arrival Path
Data Required Path
Mouse right click
Copy Ctrl+C
Select All Ctrl+A
Undo Sort
Locate Required Path...
Locate Path...
Parallel compilation is not licensed and has been disabled
Low junction temperature is 0 degrees C
High junction temperature is 85 degrees C
read_sdc

i. Technology Map Viewer:

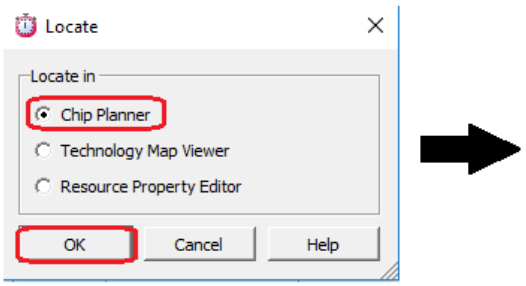


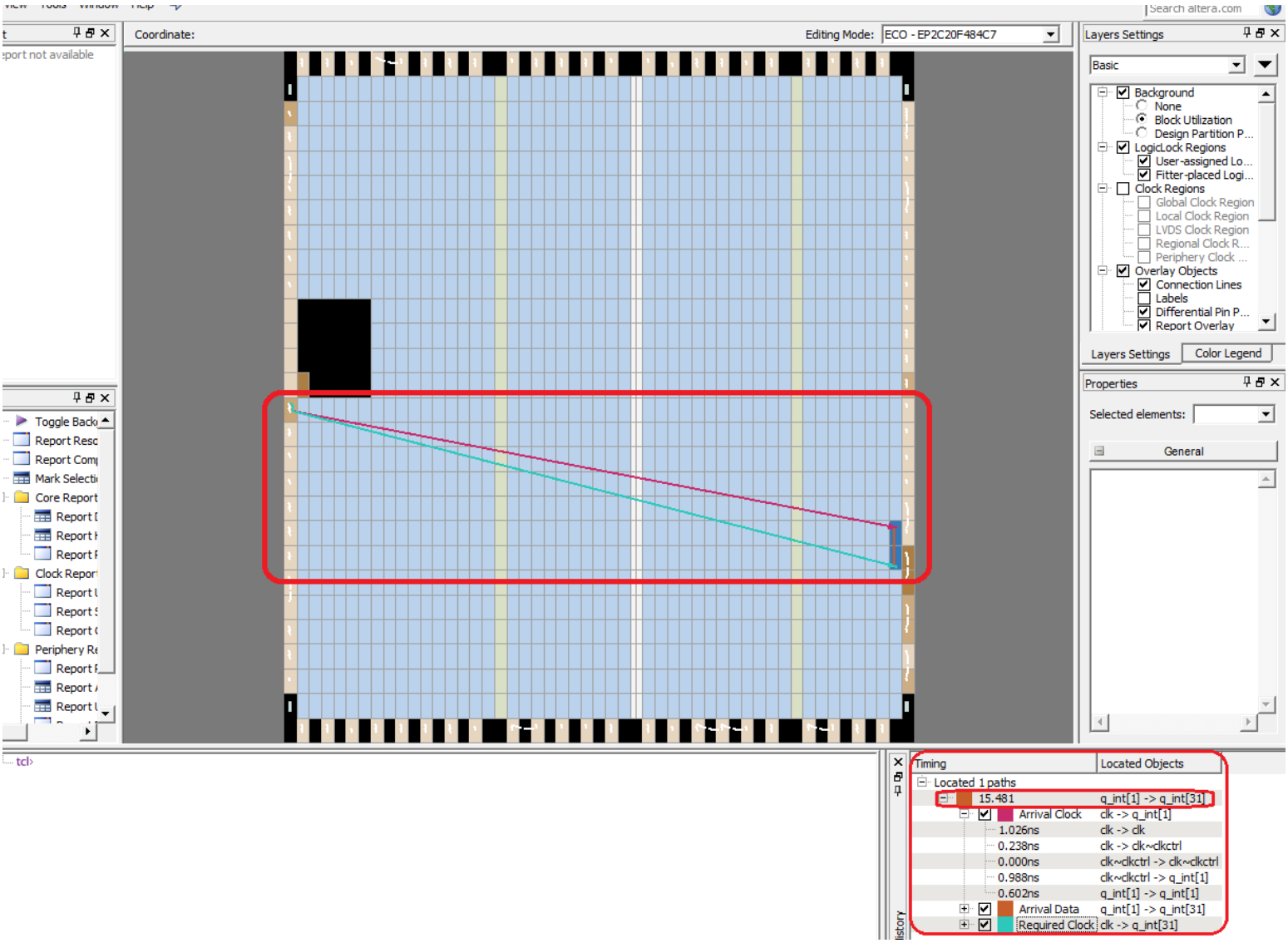
ii. Resource Property Editor:



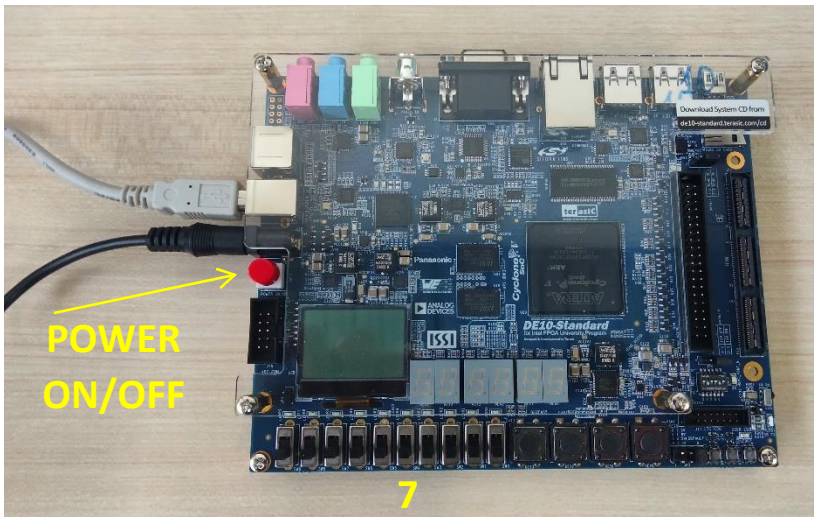


iii. Chip Planner:

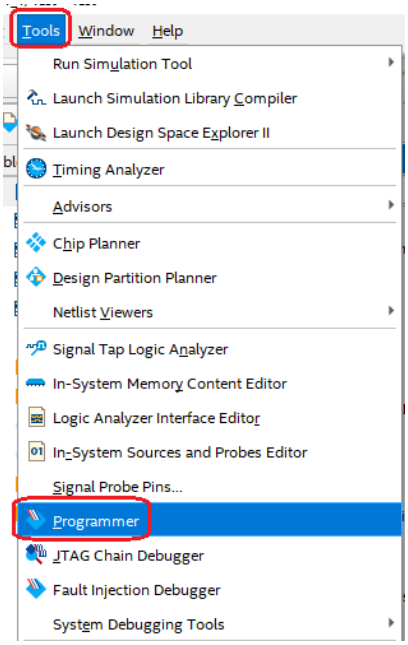


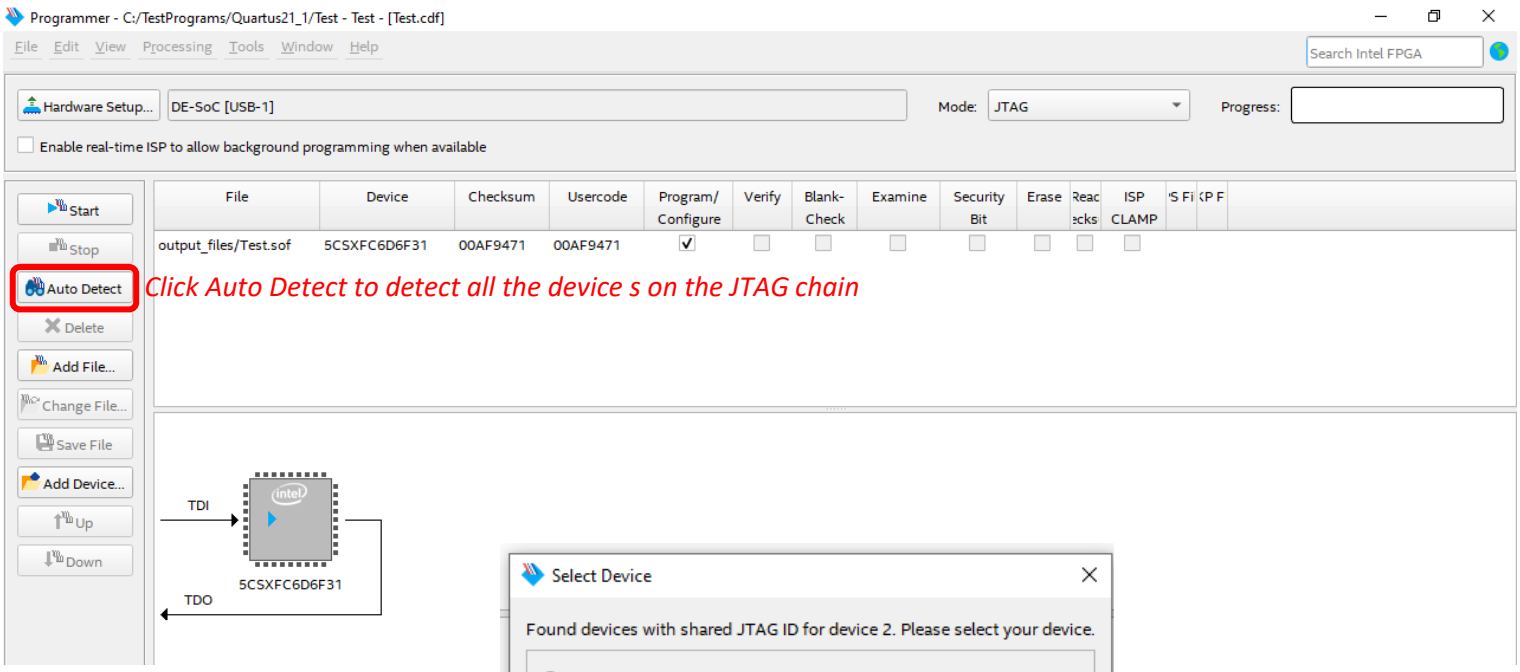


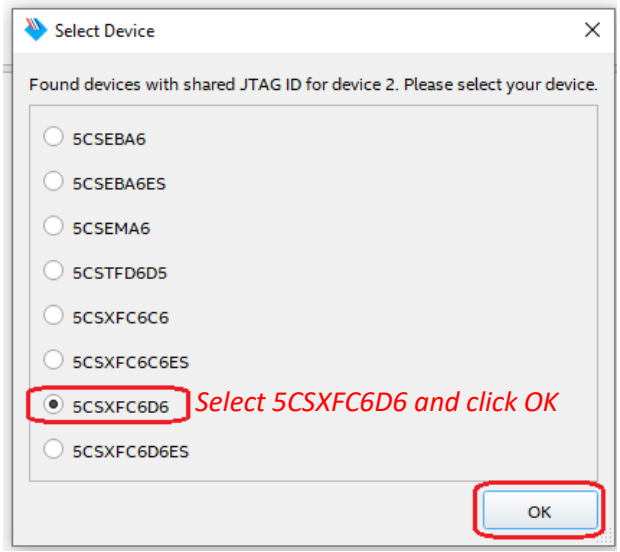
9. Step 9 – Code Programming:

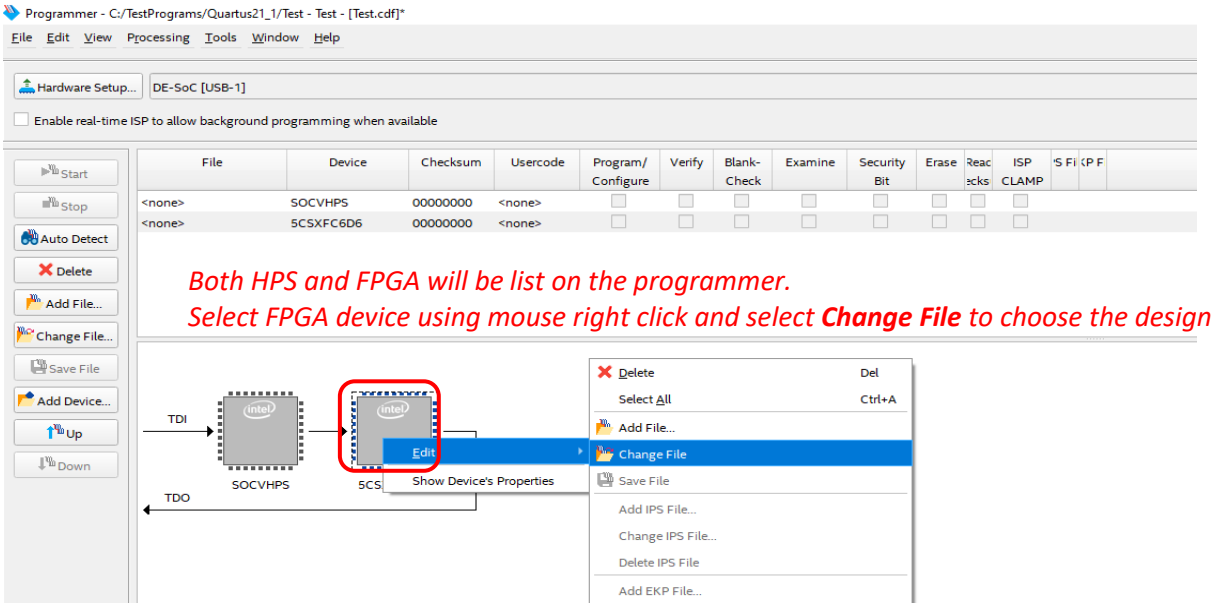


After compiling and verifying your design you are ready to program the FPGA on the development board. You download the SOF you just created into the FPGA using the USB – BlasterII circuitry on the board. Set up your hardware for programming using the following steps:

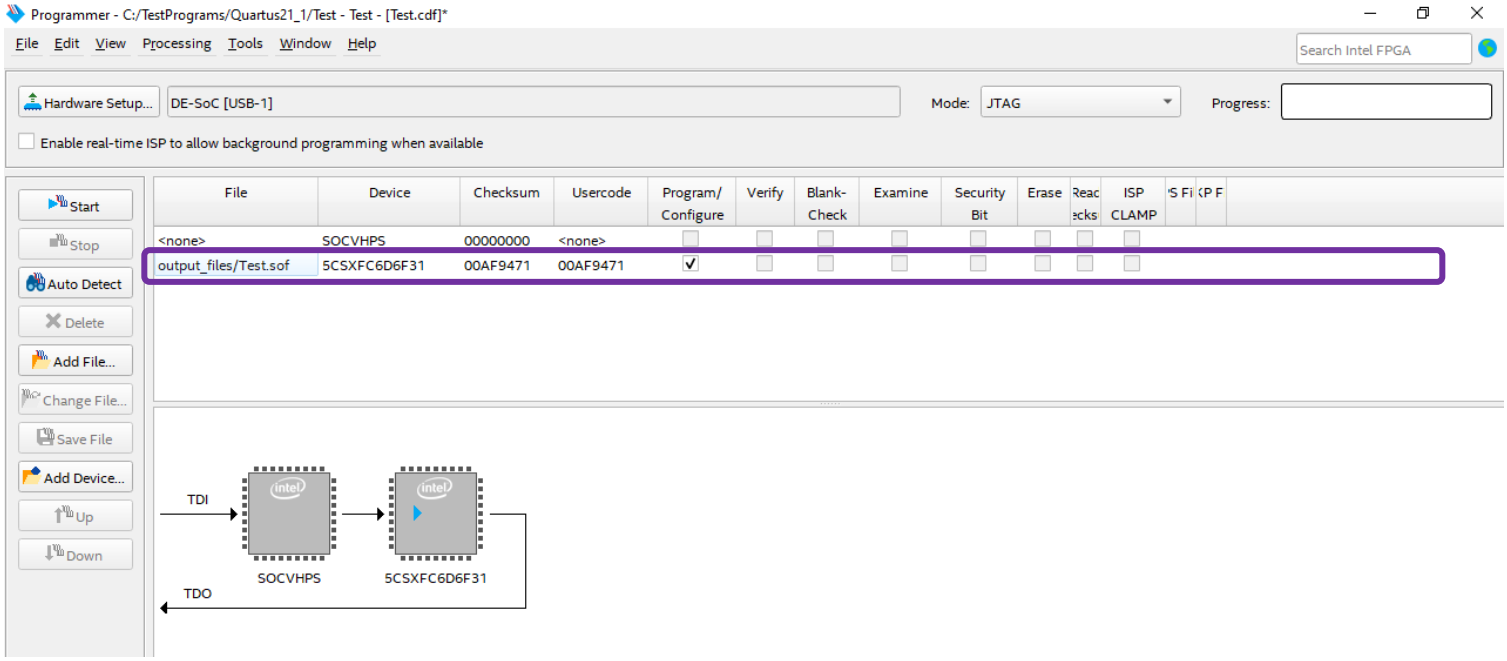


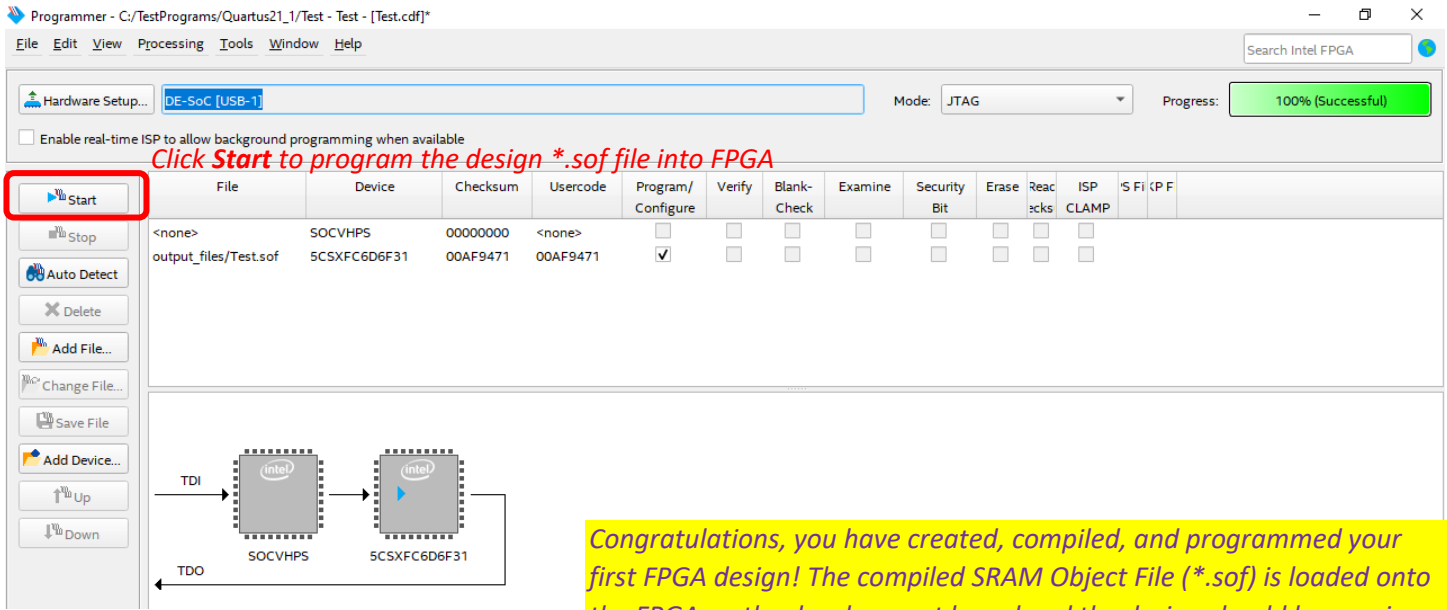






Both HPS and FPGA will be list on the programmer.
*Select FPGA device using mouse right click and select **Change File** to choose the design *.sof file.*



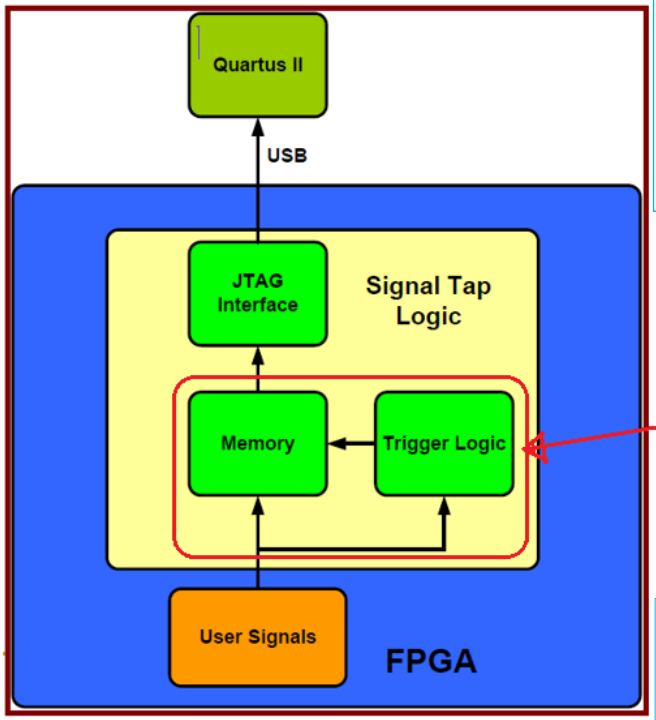


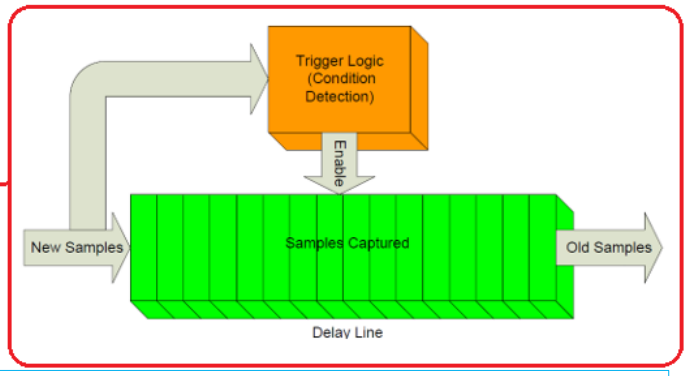
*Click **Start** to program the design *.sof file into FPGA*

Congratulations, you have created, compiled, and programmed your first FPGA design! The compiled SRAM Object File (.sof) is loaded onto the FPGA on the development board and the design should be running.*

B. Verification – Using Signal TAP:

1. Block Diagram:





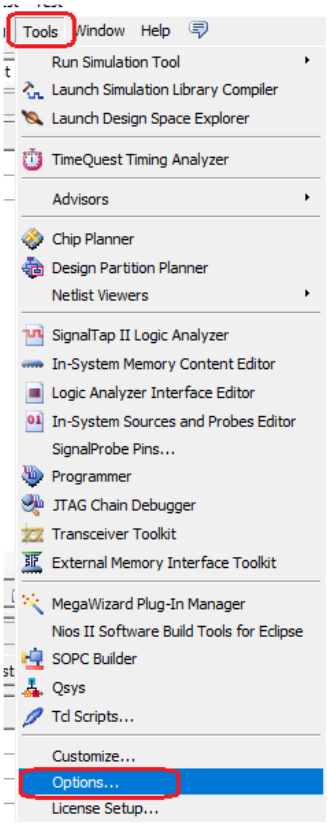
Signal TAP Pros & Cons:

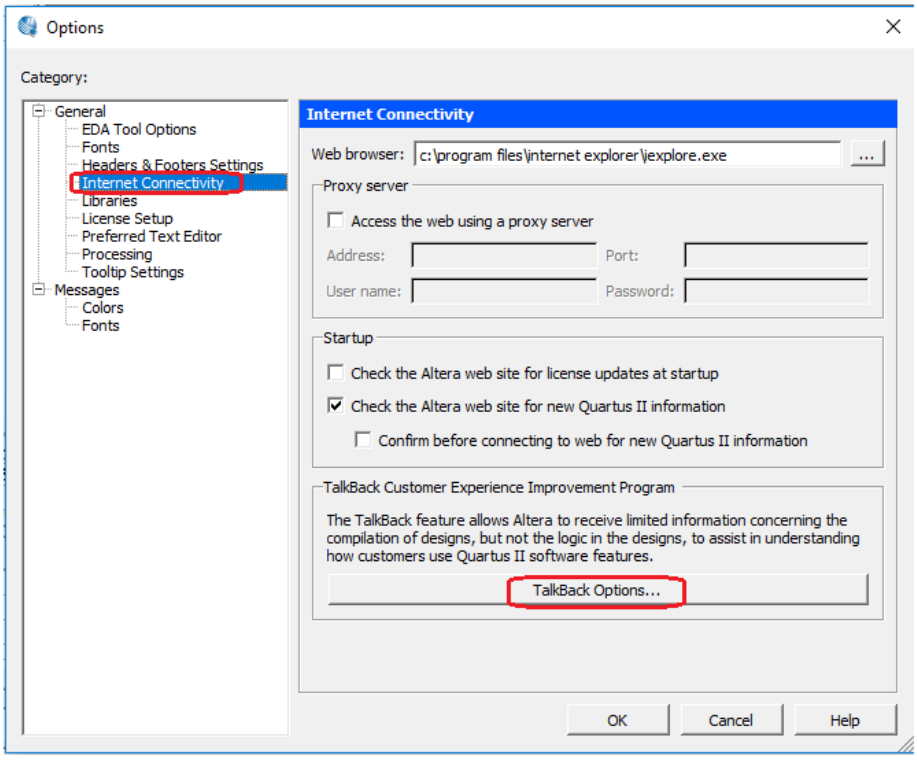
- Captures real time state of FPGA internal signals and pins (up to 200MHz)
- Connects to Quartus II through JTAG
- Do not require huge and expensive equipment
- Uses internal FPGA resources
  - Memory Blocks
  - Logic Elements
- Each time the captured signals list change the design must be recompiled

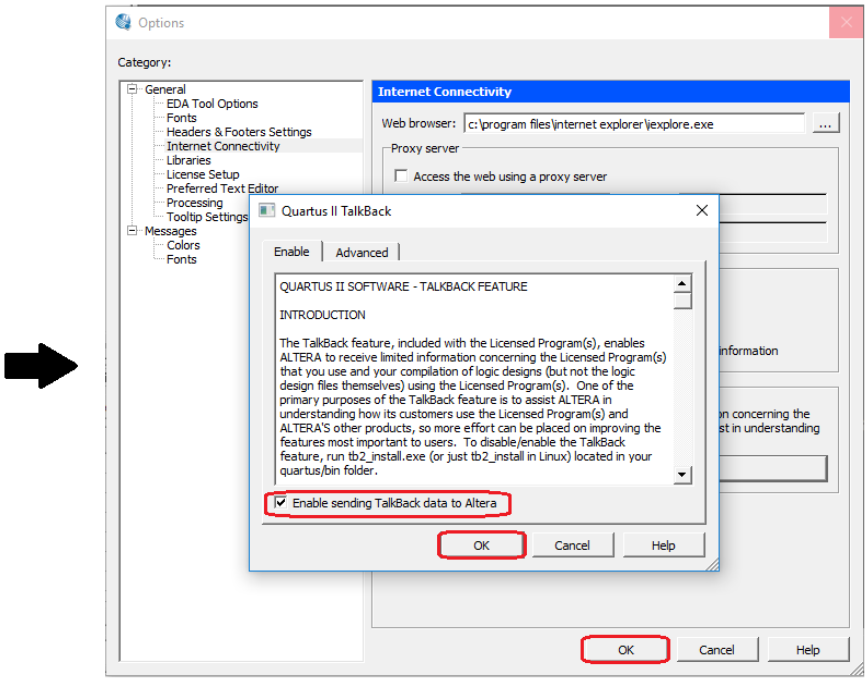
Signal TAP Features:

- Up to 1024 Data Channels
- Multiple Analyzers in One Device
  - Supports Analysis of Multiple Clock Domains
  - Each Analyzer Can Run Simultaneously
- Multiple Analyzers in One Device
- Up to 10 Trigger Levels Per Channel

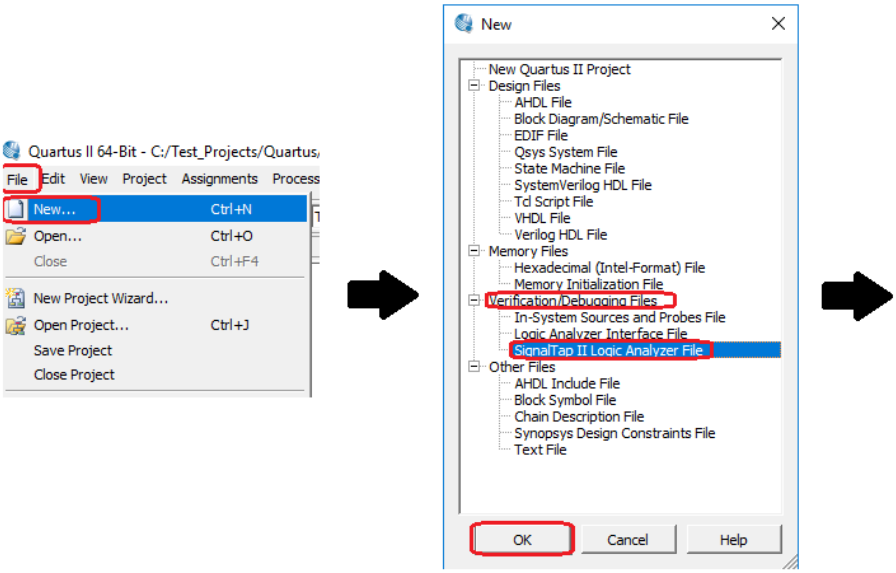
2. Using STP with a student activation license:

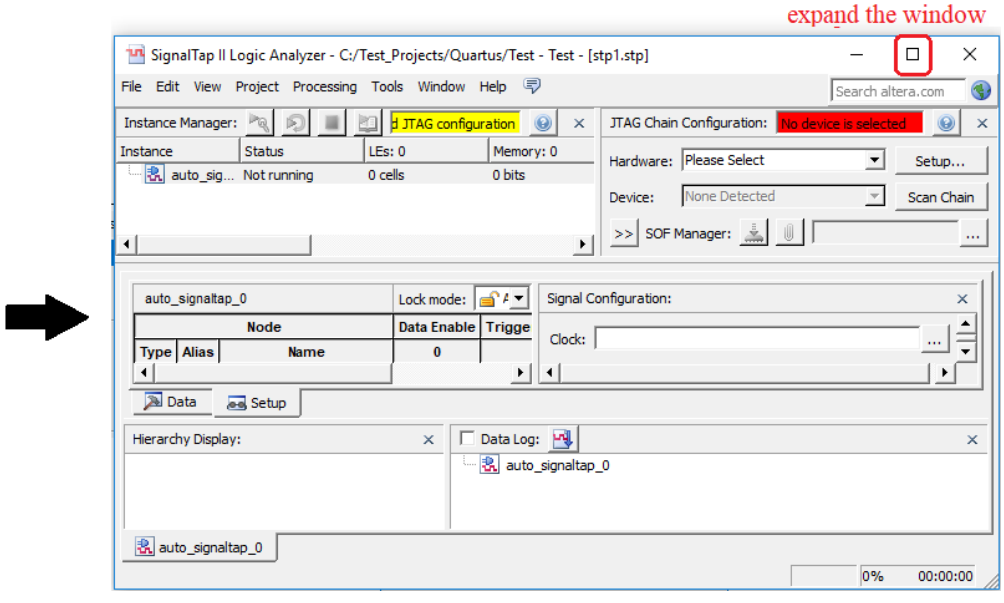




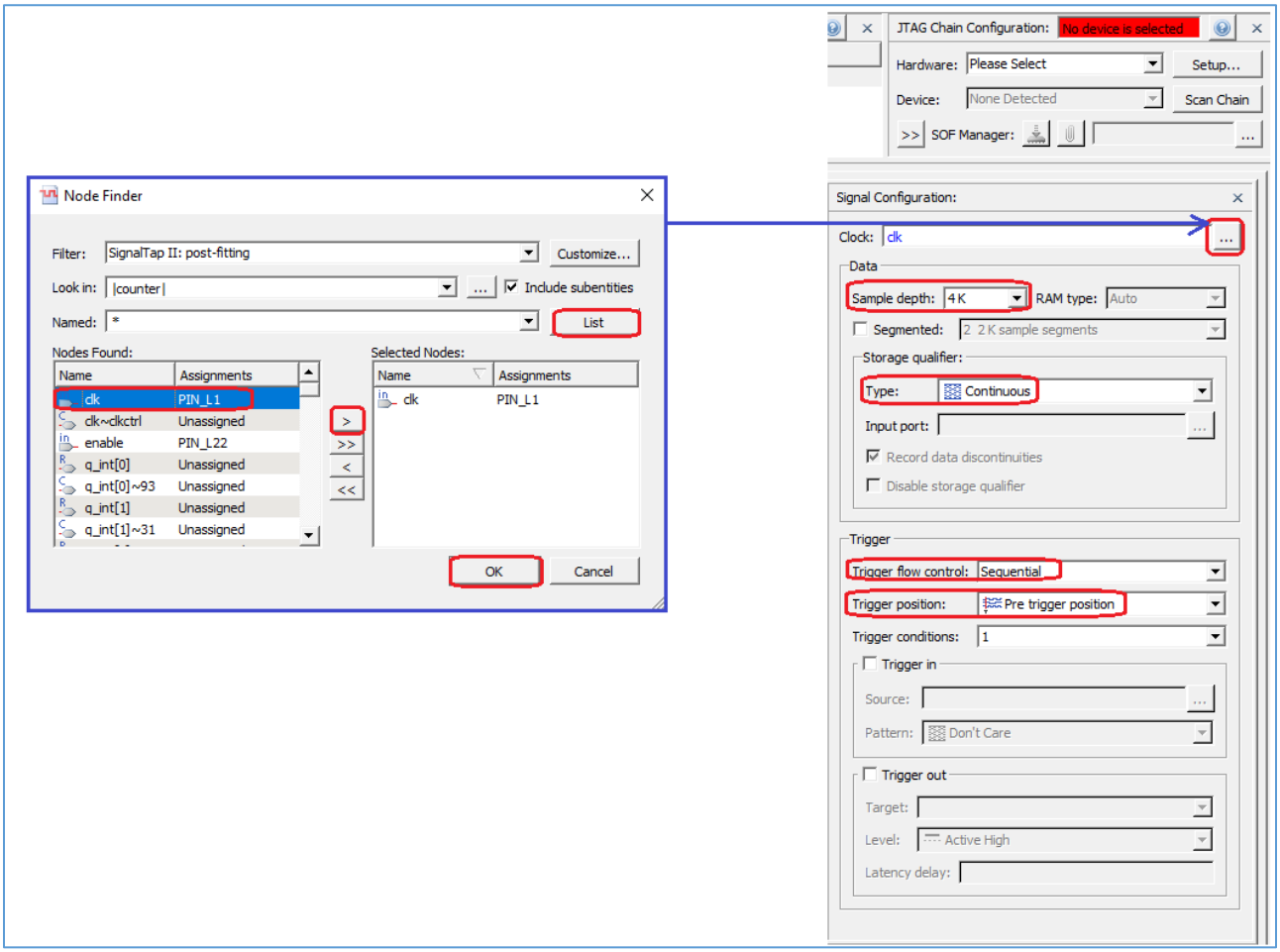


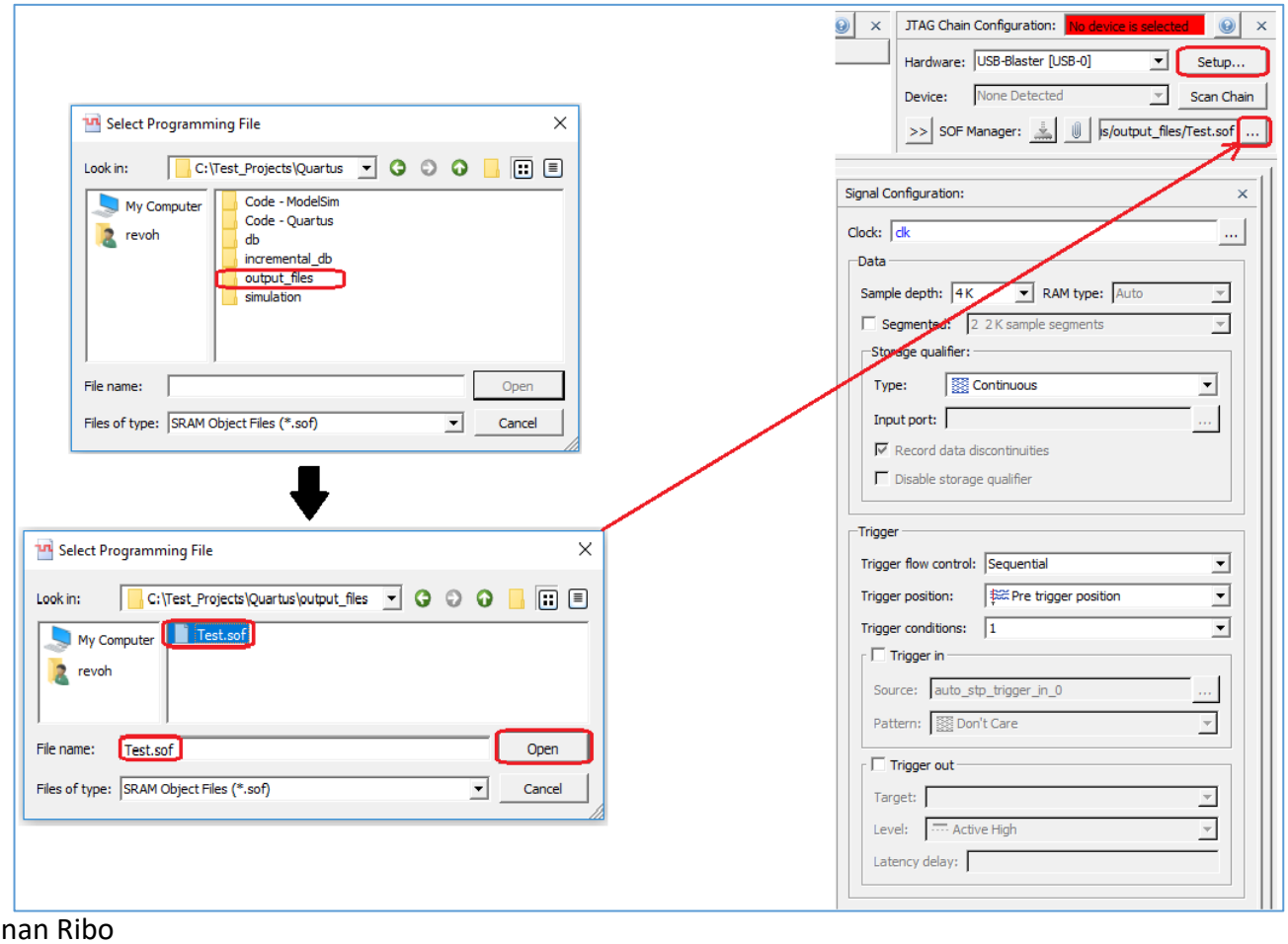
3. Create a new STP file:

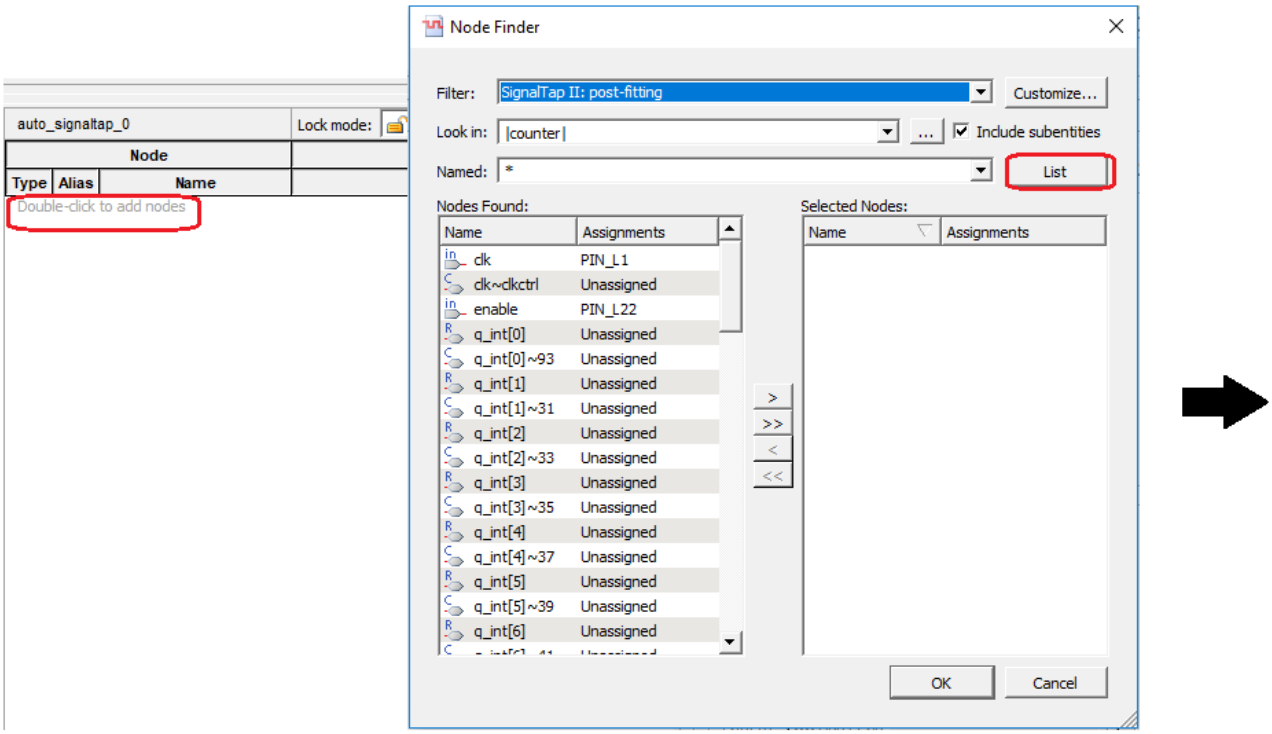


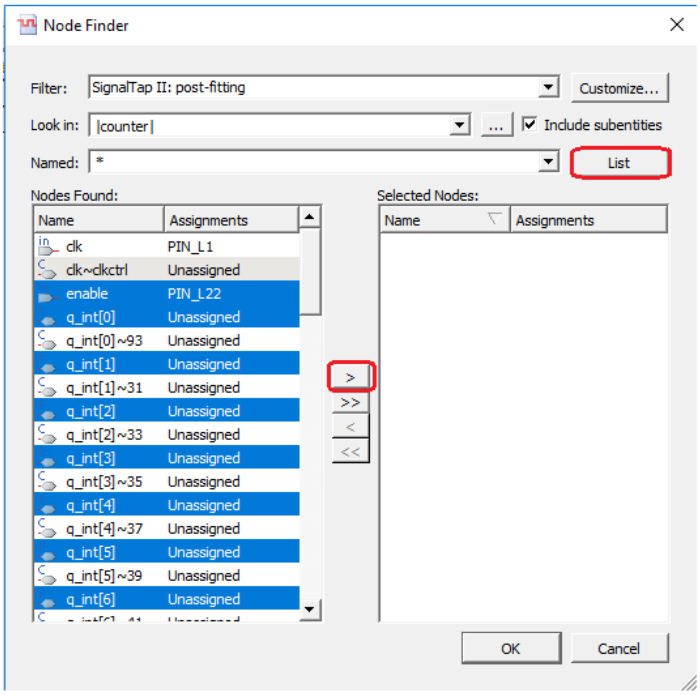


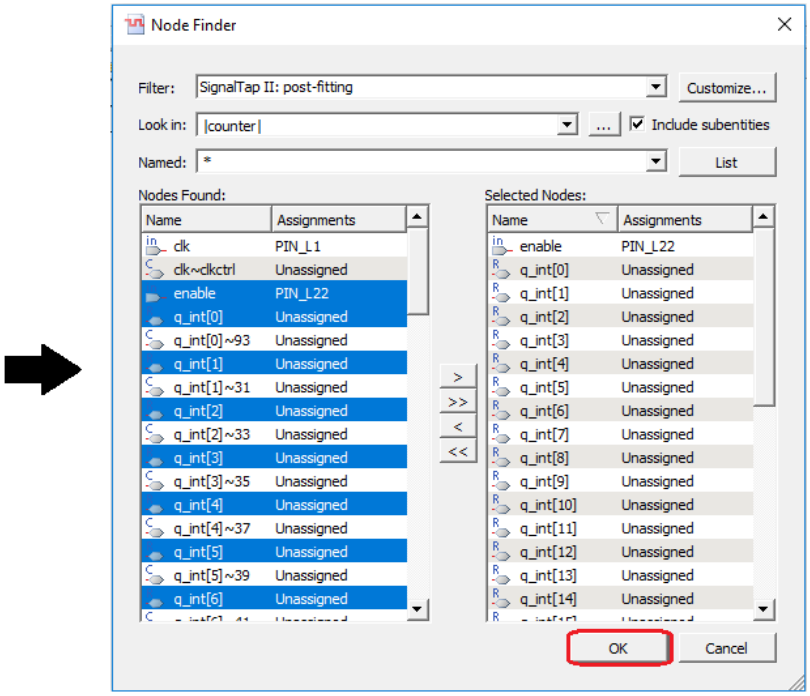
4. Configurations (after expansion of the previous window):

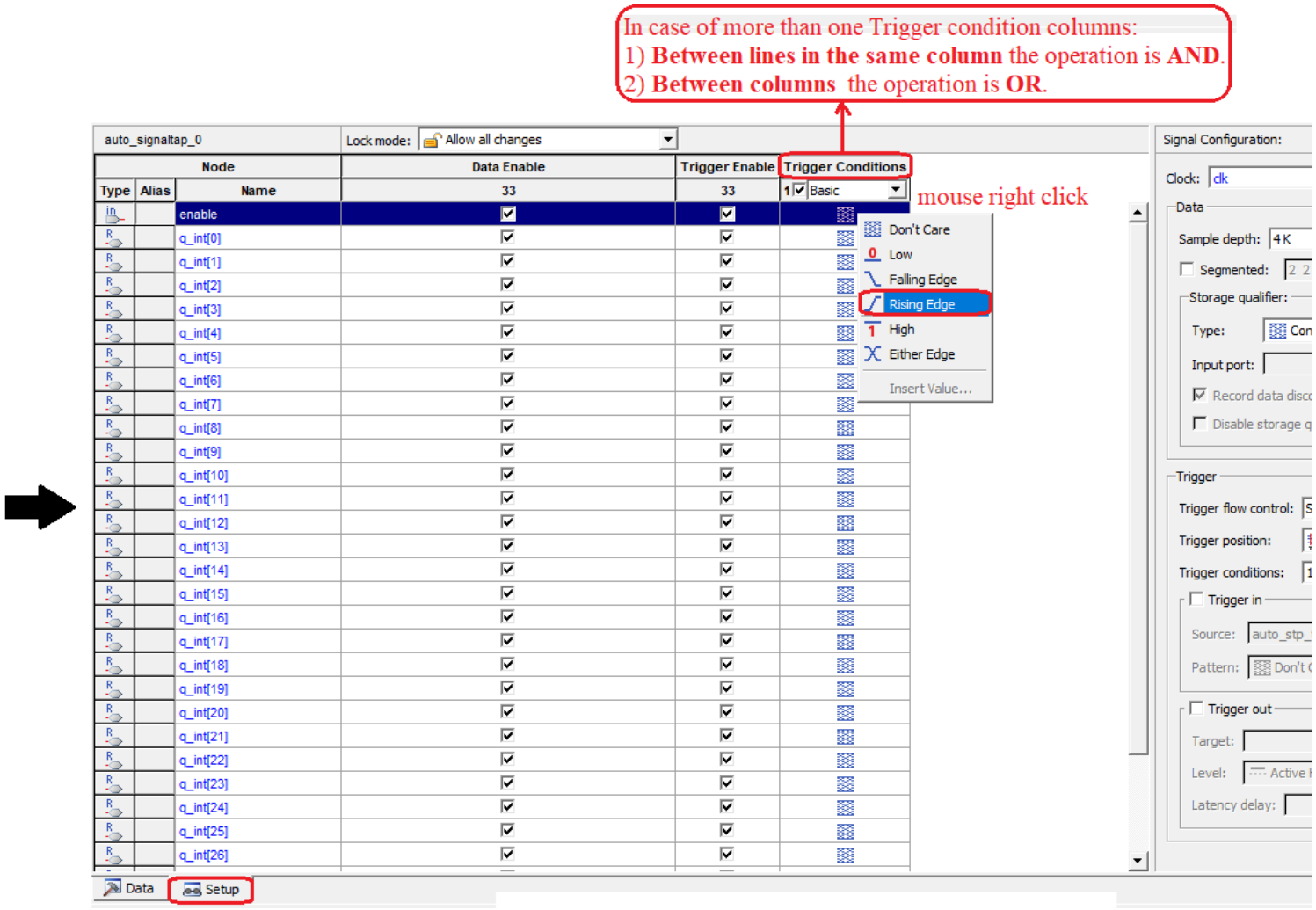


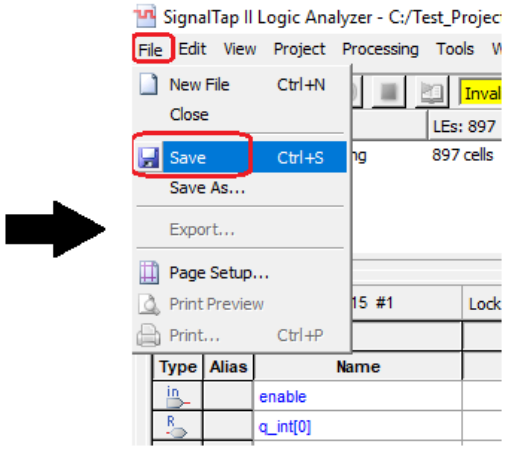



auto_signaltap_0
Lock mode:
Node
Type
Alias
Name
Double-click to add nodes
Node Finder
Filter:
SignalTap II: post-fitting
Customize...
Look in:
|counter|
Include subentities
Named:
*
List
Nodes Found:
Name
Assignments
clk
PIN_L1
clk~clkctrl
Unassigned
enable
PIN_L22
q_int[0]
Unassigned
q_int[0]~93
Unassigned
q_int[1]
Unassigned
q_int[1]~31
Unassigned
q_int[2]
Unassigned
q_int[2]~33
Unassigned
q_int[3]
Unassigned
q_int[3]~35
Unassigned
q_int[4]
Unassigned
q_int[4]~37
Unassigned
q_int[5]
Unassigned
q_int[5]~39
Unassigned
q_int[6]
Unassigned
Selected Nodes:
Name
Assignments
>
>>
<
<<
OK
Cancel

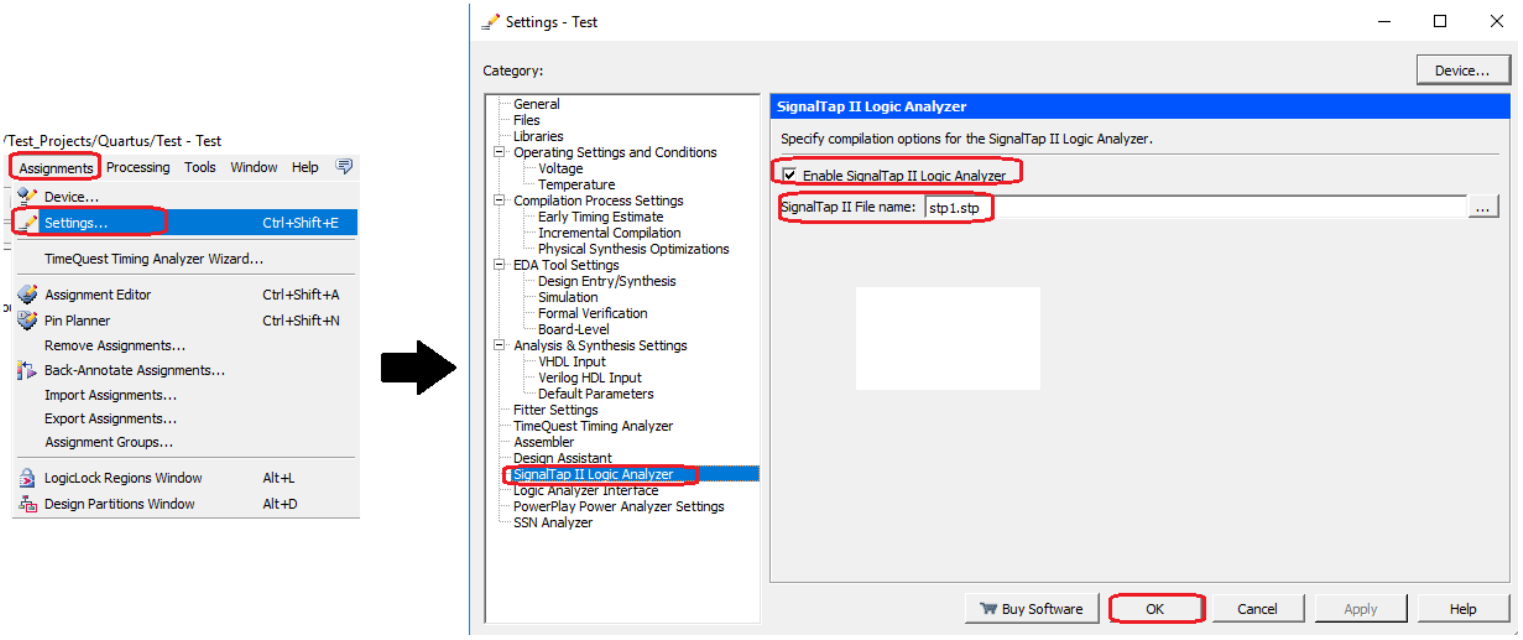

Node Finder
Filter:
SignalTap II: post-fitting
Customize...
Look in:
|counter|
Include subentities
Named:
*
List
Nodes Found:
Name
Assignments
clk
PIN_L1
clk~clkctrl
Unassigned
enable
PIN_L22
q_int[0]
Unassigned
q_int[0]~93
Unassigned
q_int[1]
Unassigned
q_int[1]~31
Unassigned
q_int[2]
Unassigned
q_int[2]~33
Unassigned
q_int[3]
Unassigned
q_int[3]~35
Unassigned
q_int[4]
Unassigned
q_int[4]~37
Unassigned
q_int[5]
Unassigned
q_int[5]~39
Unassigned
q_int[6]
Unassigned
Selected Nodes:
Name
Assignments
>
>>
<
<<
OK
Cancel


Node Finder
Filter:
SignalTap II: post-fitting
Customize...
Look in:
|counter|
Include subentities
Named:
*
List
Nodes Found:
Name
Assignments
clk
PIN_L1
clk~clkctrl
Unassigned
enable
PIN_L22
q_int[0]
Unassigned
q_int[0]~93
Unassigned
q_int[1]
Unassigned
q_int[1]~31
Unassigned
q_int[2]
Unassigned
q_int[2]~33
Unassigned
q_int[3]
Unassigned
q_int[3]~35
Unassigned
q_int[4]
Unassigned
q_int[4]~37
Unassigned
q_int[5]
Unassigned
q_int[5]~39
Unassigned
q_int[6]
Unassigned
Selected Nodes:
Name
Assignments
enable
PIN_L22
q_int[0]
Unassigned
q_int[1]
Unassigned
q_int[2]
Unassigned
q_int[3]
Unassigned
q_int[4]
Unassigned
q_int[5]
Unassigned
q_int[6]
Unassigned
q_int[7]
Unassigned
q_int[8]
Unassigned
q_int[9]
Unassigned
q_int[10]
Unassigned
q_int[11]
Unassigned
q_int[12]
Unassigned
q_int[13]
Unassigned
q_int[14]
Unassigned
>
>>
<
<<
OK
Cancel


In case of more than one Trigger condition columns:
1) Between lines in the same column the operation is AND.
2) Between columns the operation is OR.
auto_signaltap_0
Lock mode:
Allow all changes
Node
Data Enable
Trigger Enable
Trigger Conditions
Type
Alias
Name
33
33
1
Basic
mouse right click
Don't Care
Low
Falling Edge
Rising Edge
High
Either Edge
Insert Value...
Signal Configuration:
Clock:
clk
Data
Sample depth:
4K
Segmented:
Storage qualifier:
Type:
Input port:
Trigger
Trigger flow control:
Trigger position:
Trigger conditions:
Trigger in
Source:
Pattern:
Trigger out
Target:
Level:
Latency delay:
Data
Setup

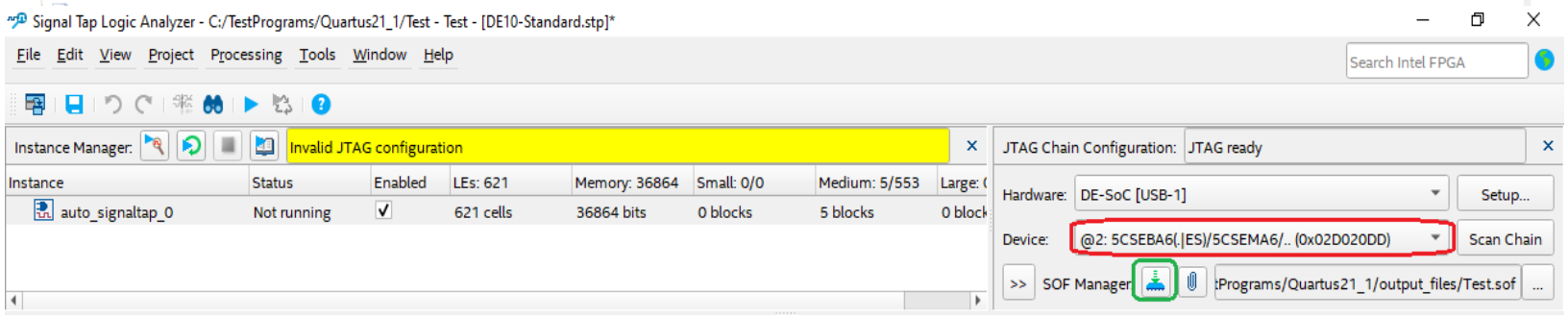

SignalTap II Logic Analyzer - C:/Test_Projec
File
Edit
View
Project
Processing
Tools
New File
Ctrl+N
Close
Save
Ctrl+S
Save As...
Export...
Page Setup...
Print Preview
Print...
Ctrl+P
LEs: 897
897 cells
Type
Alias
Name
enable
q_int[0]

5. Project Compilation and Programming (an eventually the Signals results):

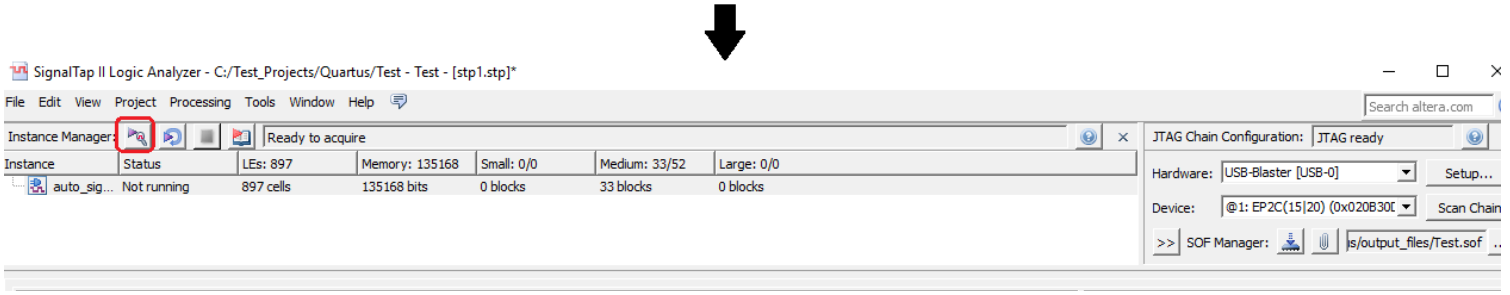


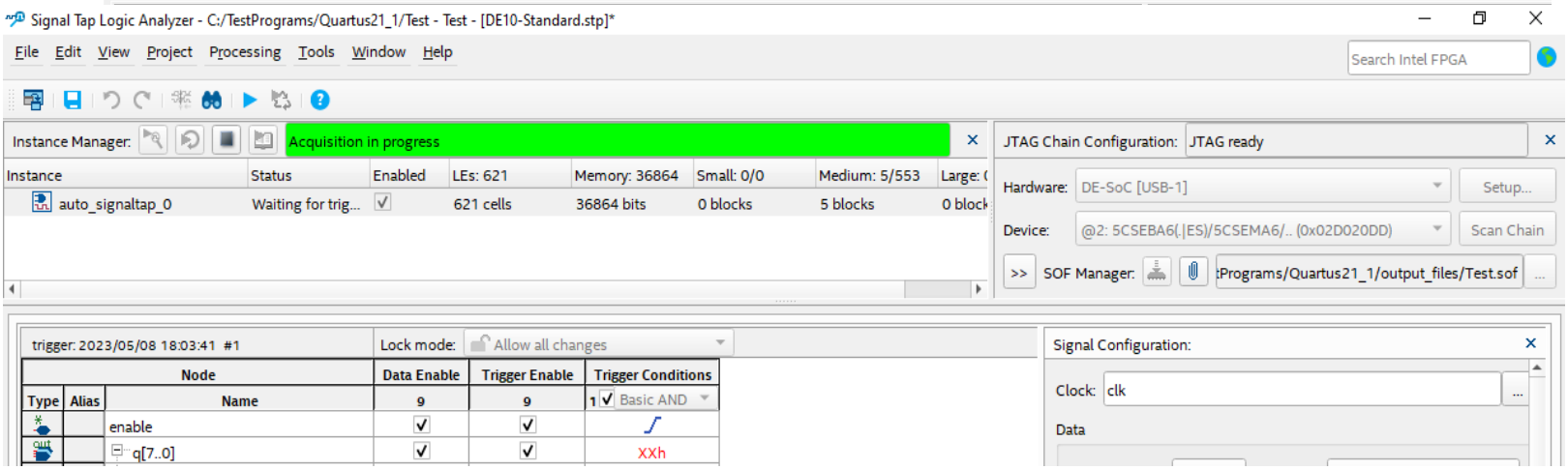
Turn ON the DE1 Board power

Open the STP file

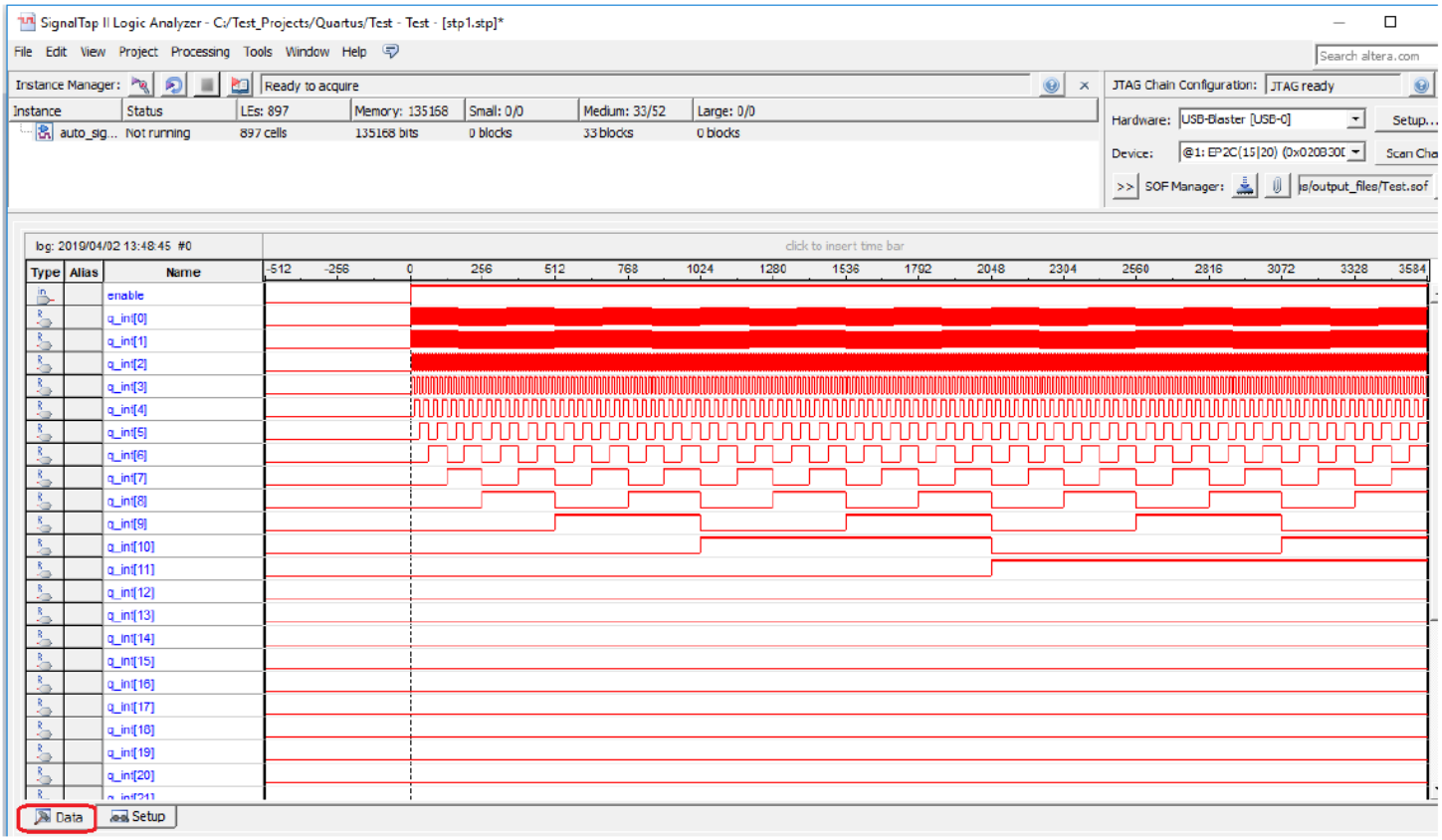


Device programming

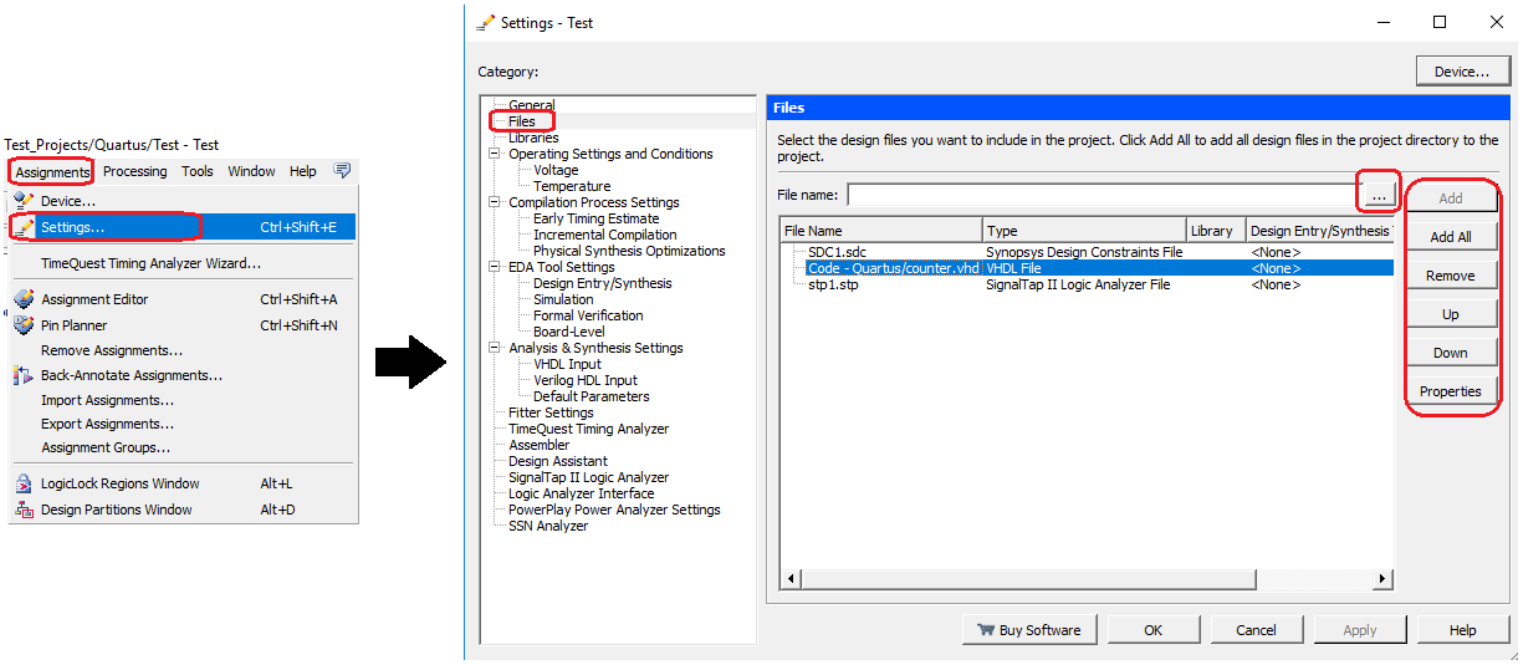




Assert the Trigger



C. Changing the VHDL source files of the project:



D. Open Existing Project: