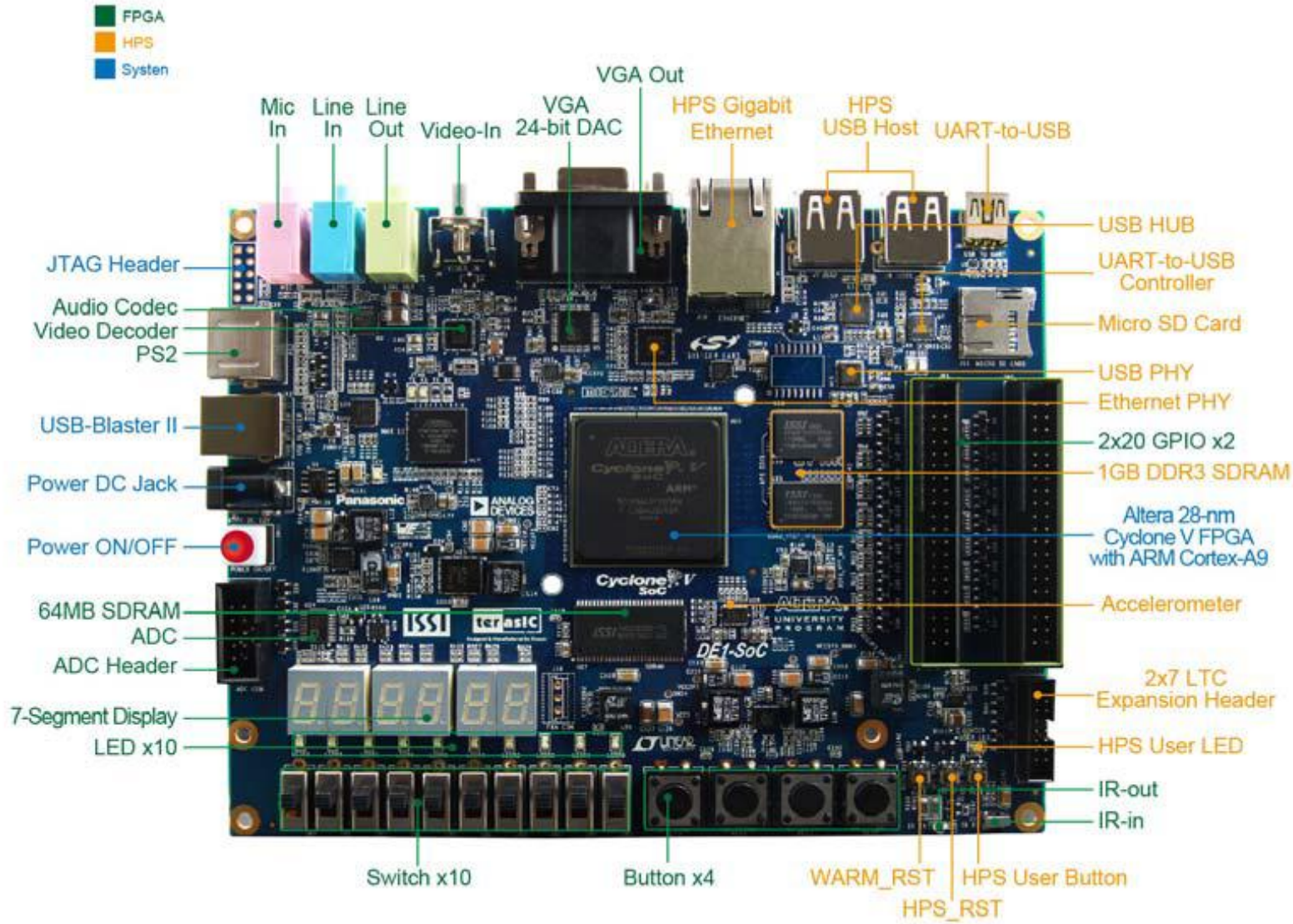
Programmable Logic Devices
First Laboratory Session

Prof. Claudio Passerone

(Electronics for Embedded Systems)

Altera DE1-SoC Board

- Altera DE1-SoC Board
  - FPGA + HPS
  - Memories
  - ADCs
  - Switches
  - Buttons
  - LEDs
  - Displays
  - Codecs
    - Audio
    - Video
  - Connectors
  - Expansion Headers



Altera DE1-SoC Board

- Pin assignments
  - Switches
    - SW0 → PIN_AB12
    - SW1 → PIN_AC12
    - …
  - Buttons
    - KEY0 → PIN_AA14
    - …
  - LEDs
    - LEDR0 → PIN_V16
    - …
  - Clock
    - CLOCK_50 → PIN_AF14

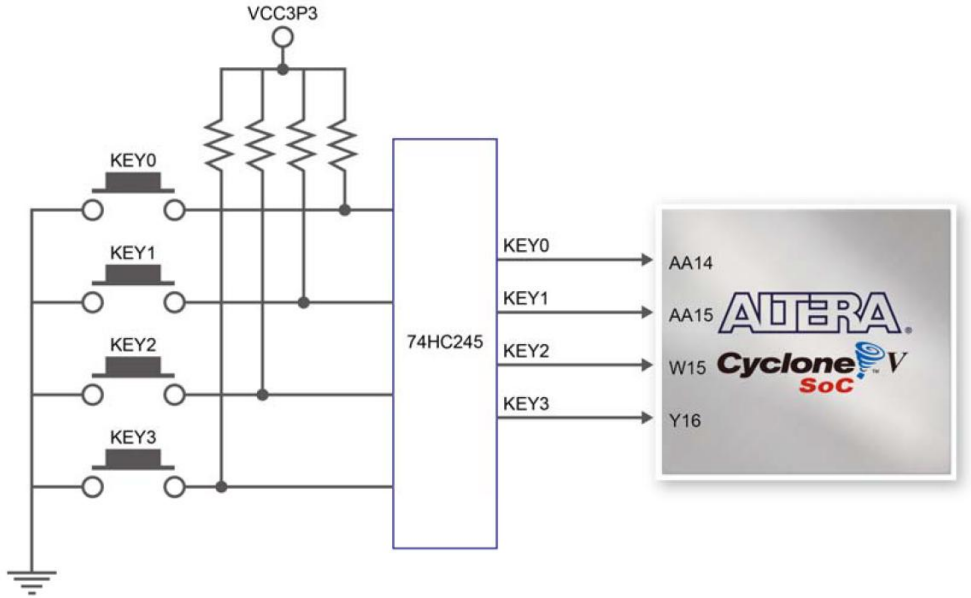
Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SW[0]	PIN_AB12	Slide Switch[0]	3.3V
SW[1]	PIN_AC12	Slide Switch[1]	3.3V
SW[2]	PIN_AF9	Slide Switch[2]	3.3V
SW[3]	PIN_AF10	Slide Switch[3]	3.3V
SW[4]	PIN_AD11	Slide Switch[4]	3.3V
SW[5]	PIN_AD12	Slide Switch[5]	3.3V
SW[6]	PIN_AE11	Slide Switch[6]	3.3V
SW[7]	PIN_AC9	Slide Switch[7]	3.3V
SW[8]	PIN_AD10	Slide Switch[8]	3.3V

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
KEY[0]	PIN_AA14	Push-button[0]	3.3V
KEY[1]	PIN_AA15	Push-button[1]	3.3V
KEY[2]	PIN_W15	Push-button[2]	3.3V
KEY[3]	PIN_Y16	Push-button[3]	3.3V

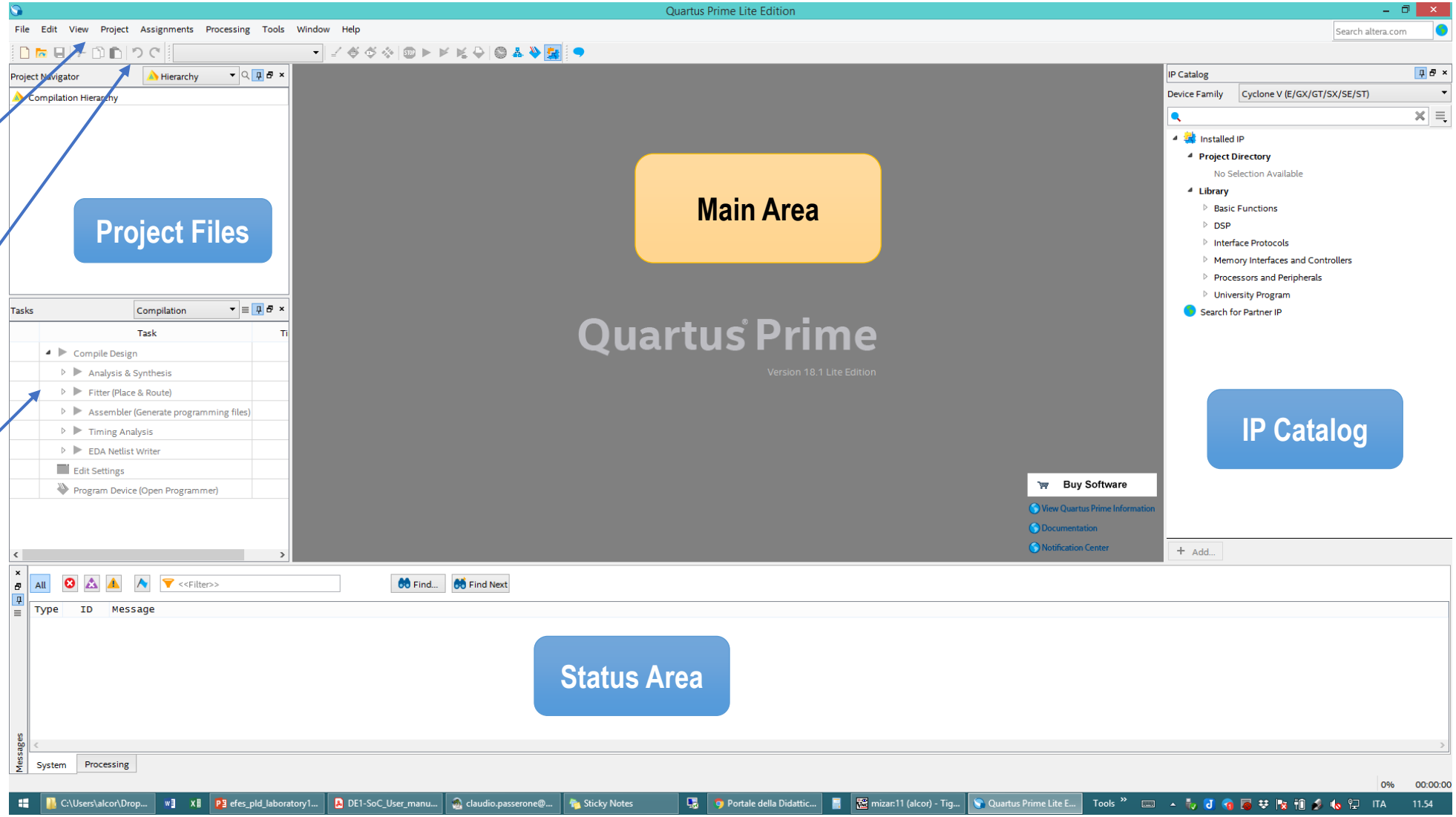
Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
LEDR[0]	PIN_V16	LED [0]	3.3V
LEDR[1]	PIN_W16	LED [1]	3.3V
LEDR[2]	PIN_V17	LED [2]	3.3V
LEDR[3]	PIN_V18	LED [3]	3.3V
LEDR[4]	PIN_W17	LED [4]	3.3V
LEDR[5]	PIN_W19	LED [5]	3.3V
LEDR[6]	PIN_Y19	LED [6]	3.3V
LEDR[7]	PIN_W20	LED [7]	3.3V
LEDR[8]	PIN_W21	LED [8]	3.3V
LEDR[9]	PIN_Y21	LED [9]	3.3V

Altera DE1-SoC Board

- Schematic diagrams
  - Useful to understand how things work
- See also
  - UART
  - Expansion Headers
  - VGA (Triple DAC)
  - 7-Seg Displays

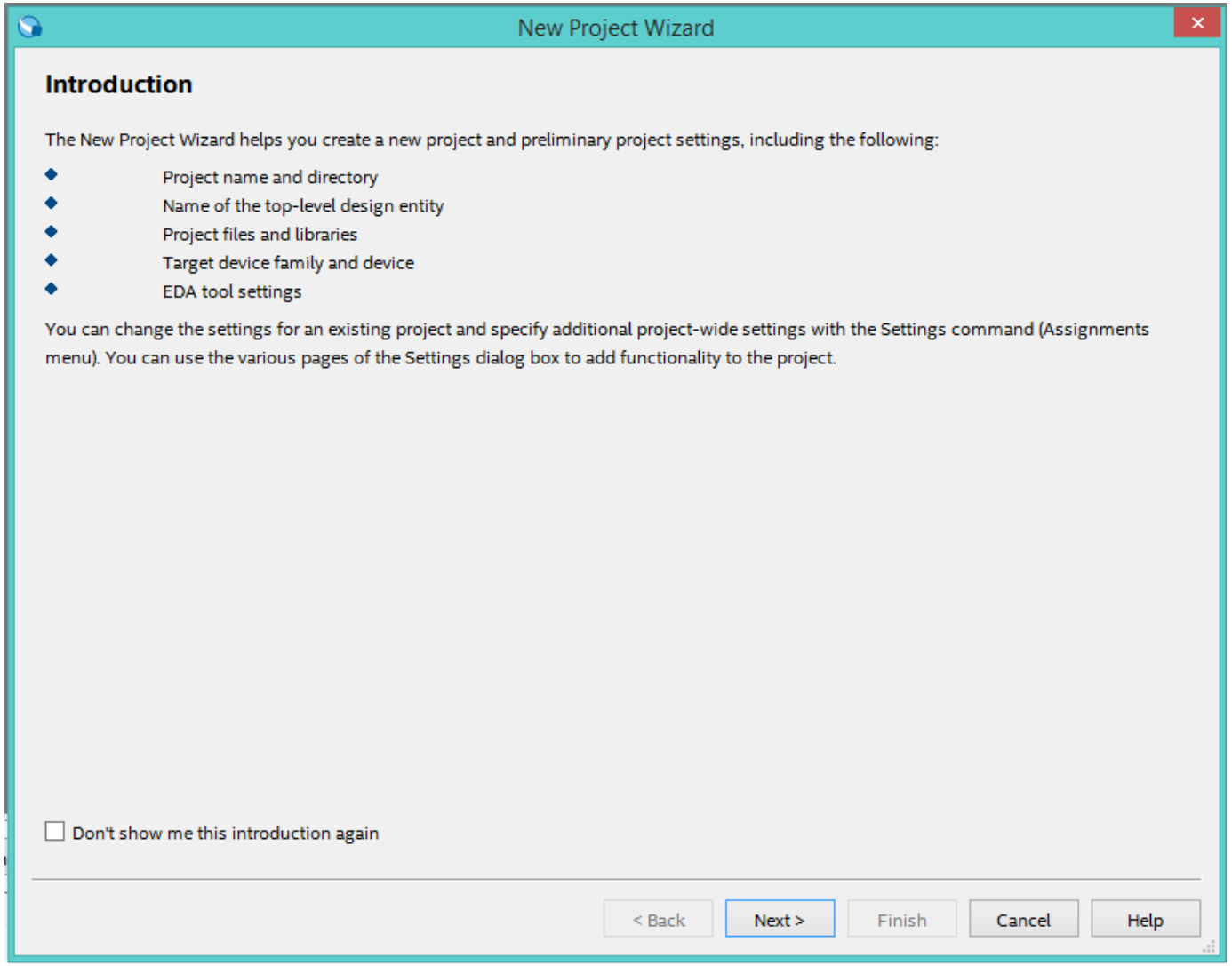


Quartus Prime



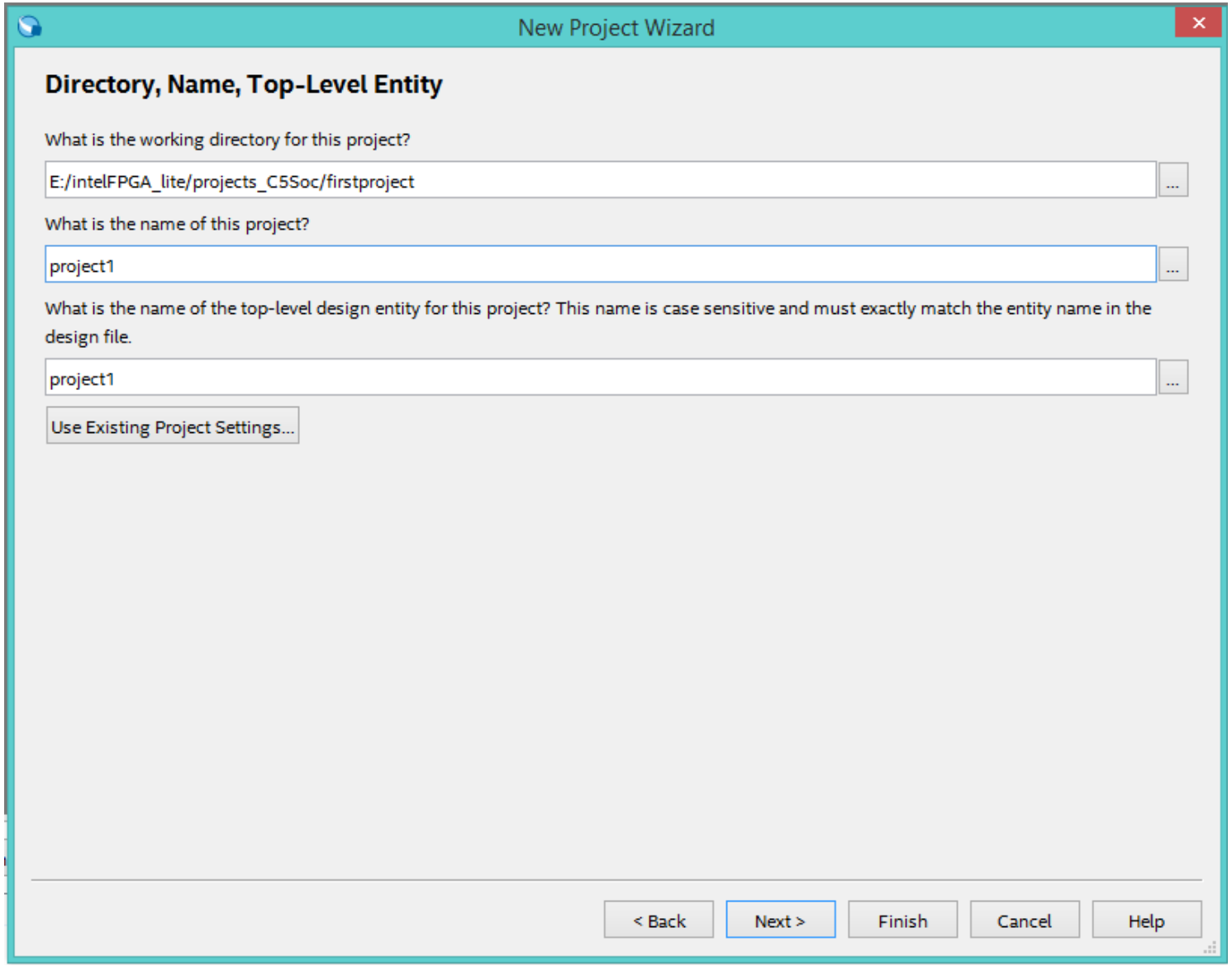
Quartus Prime

- Projects
  - Create a project
    - Under File menu
    - Choose New Project Wizard…
      - Click Next in the window that appears



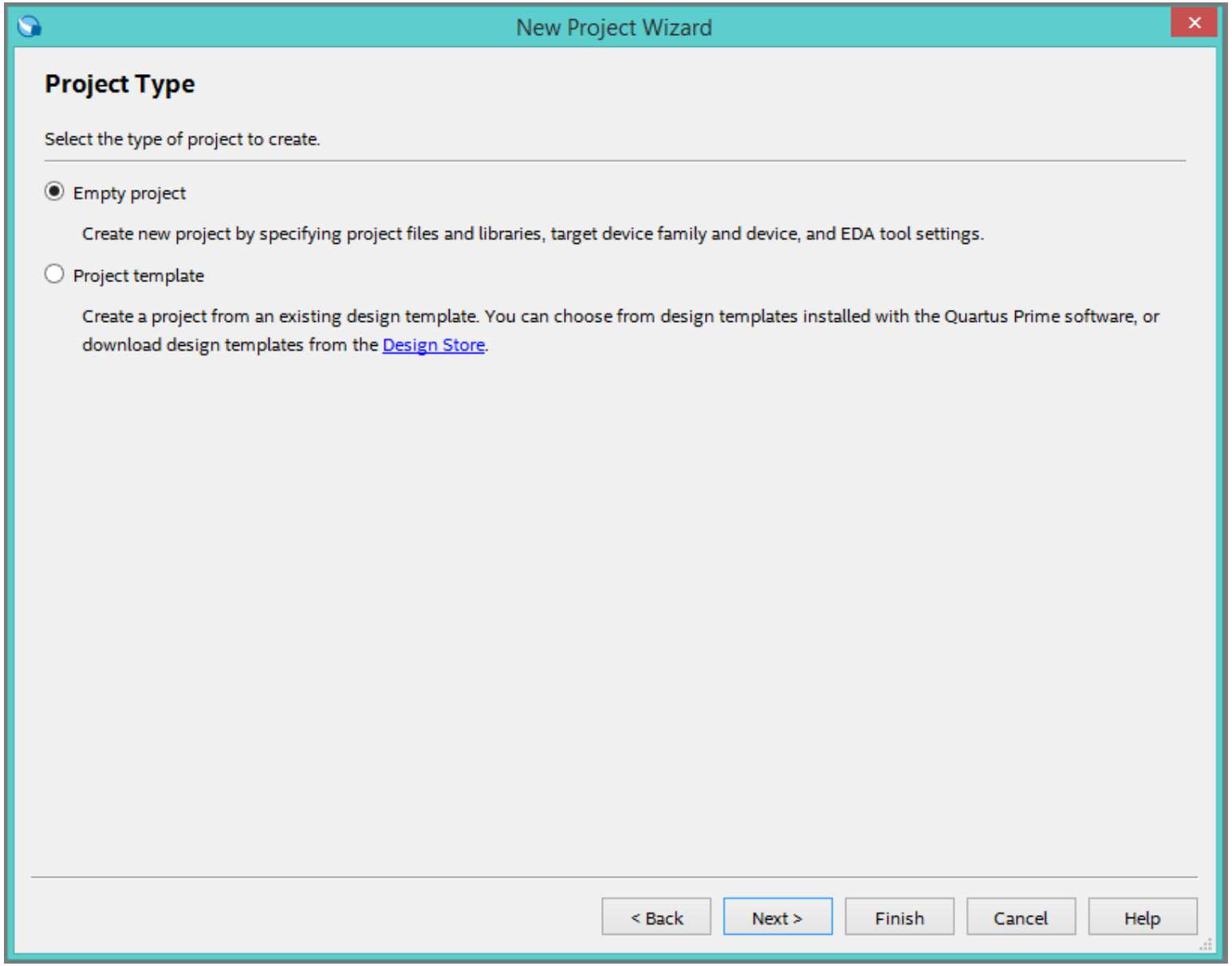
Quartus Prime

- Projects
  - Create a project
    - Specify design directory
      - Directory names without spaces
      - You can use the character / (slash) instead of \ (backslash) to separate folder names
      - Make sure to have write access to the selected directory
      - Click Next



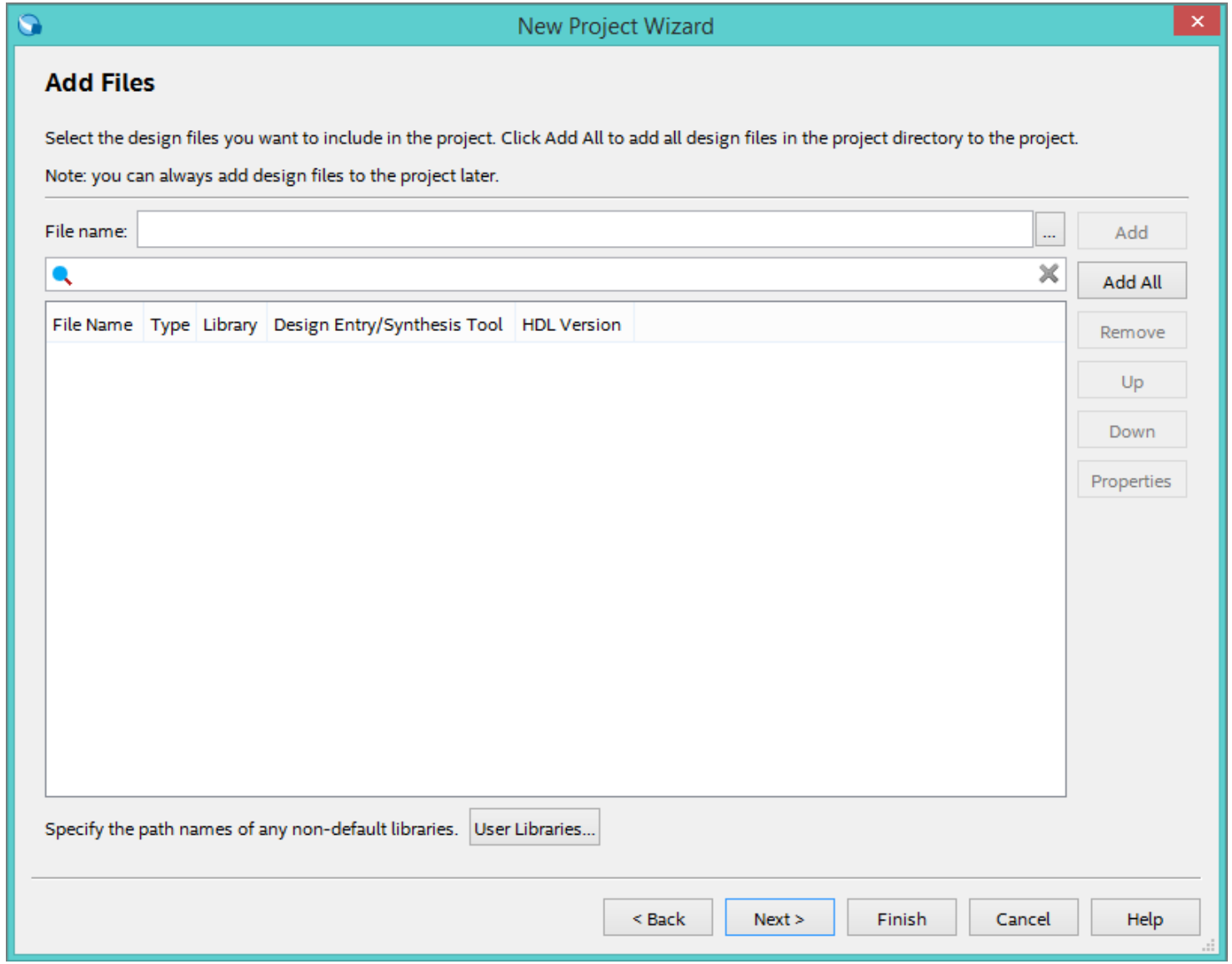
Quartus Prime

- Projects
  - Create a project
    - Specify template
      - Use Empty project if there is no template
      - Click Next



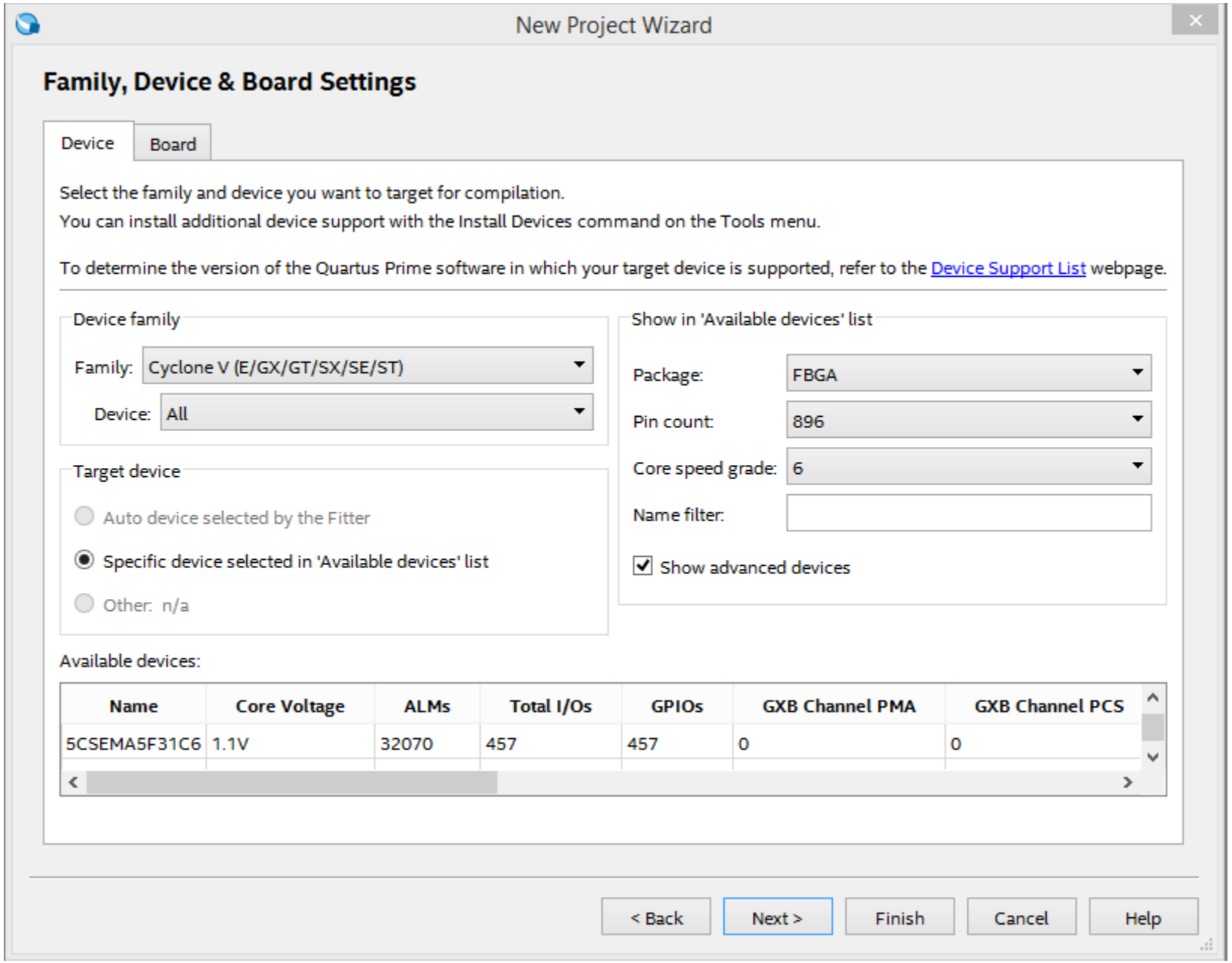
Quartus Prime

- Projects
  - Create a project
    - Specify design files
      - Add design files here if any
      - Otherwise simply click Next
        - You can always add design files later
        - You can write design files within the tool



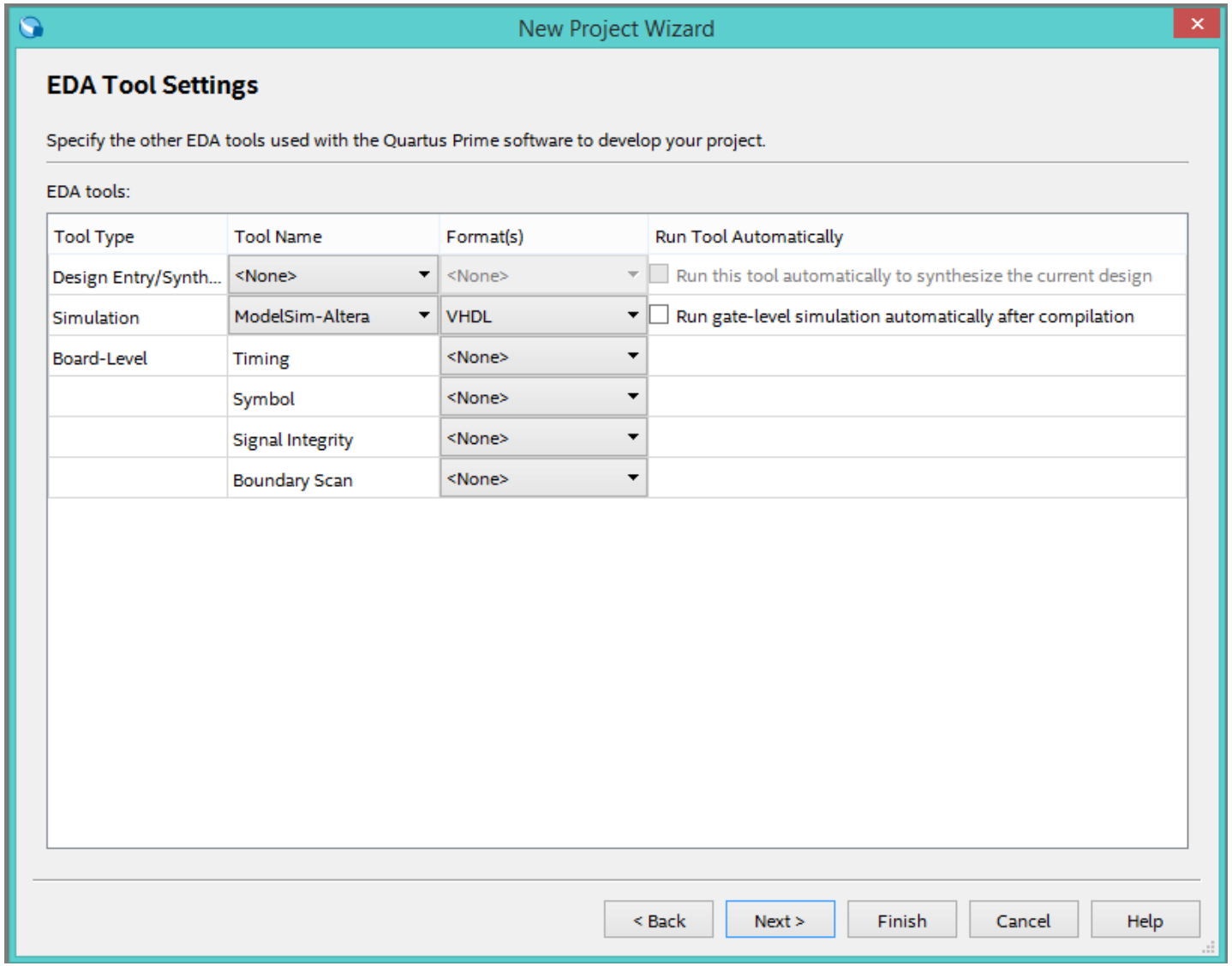
Quartus Prime

- Projects
  - Create a project
    - Select the device
      - 5CSEMA5F31C6
      - Cyclone V family
      - FBGA Package
      - 896 pins
      - Speed grade 6
      - Click Next
    - Different board
      - Cyclone 10 LP family
      - 10CL025YE144C8G
      - 10CL006YE144C8G



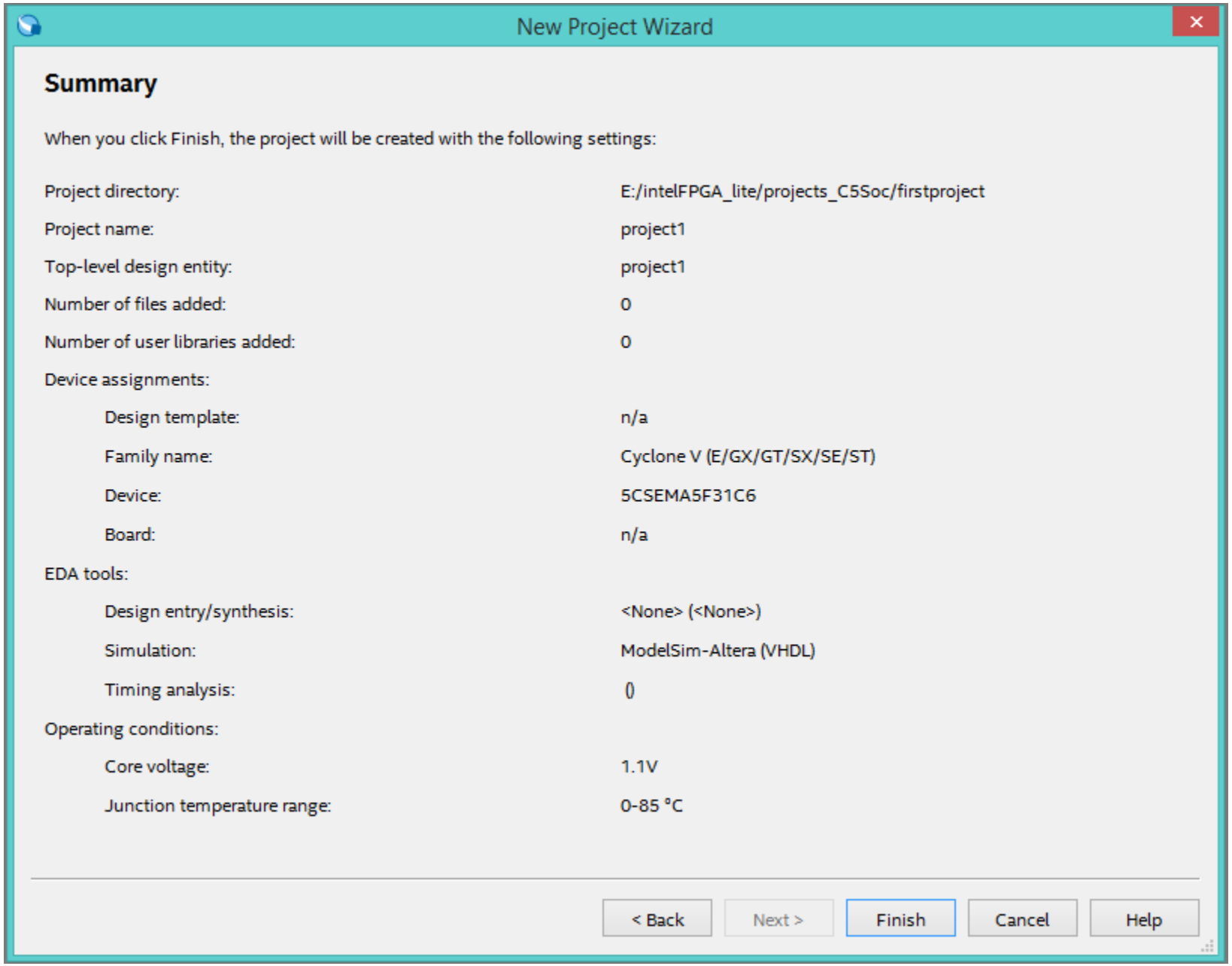
Quartus Prime

- Projects
  - Create a project
    - Select the simulator
      - ModelSim-Altera
        - Specify VHDL as the Format
      - All others tools are already included
        - Or not needed in this class
      - When designing a NiosII system or an HPS system you may want to leave this blank
      - Click Next



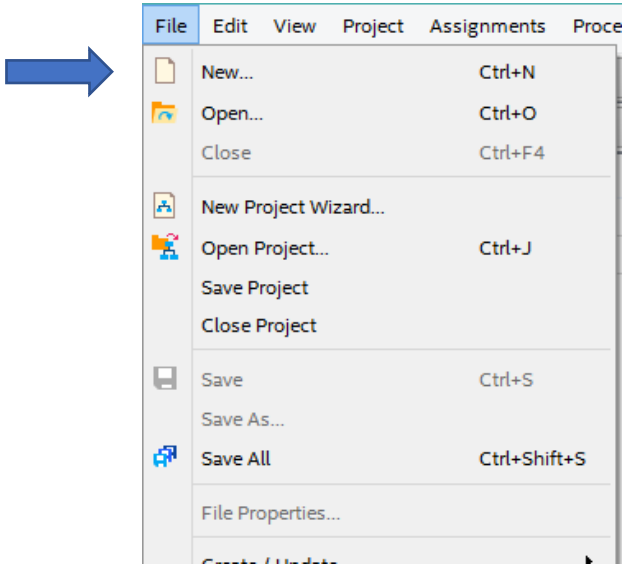
Quartus Prime

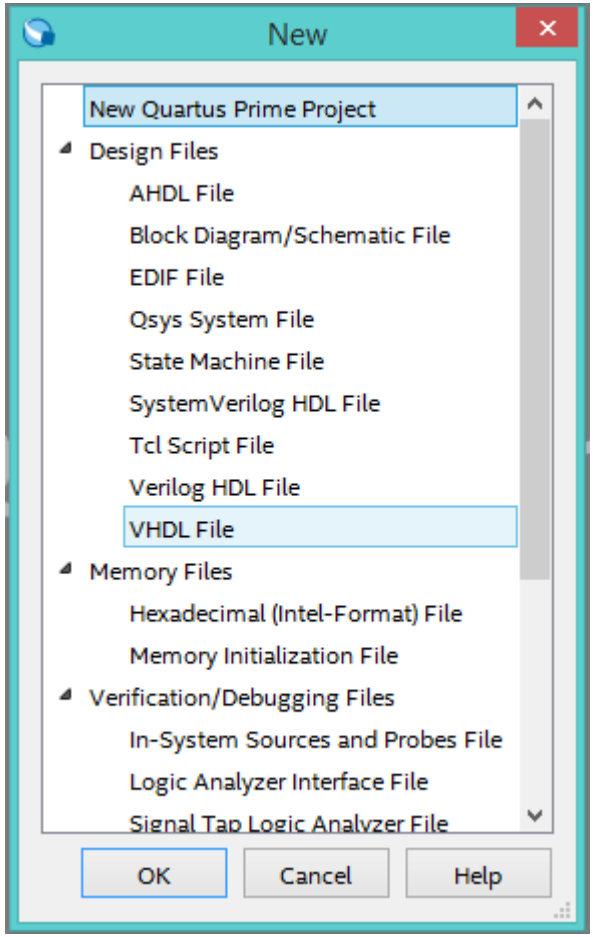
- Projects
  - Create a project
    - Summary
      - Check all data
      - Click Finish



Quartus Prime

- Editor
  - Create new VHDL file
    - Under File menu
    - Click New
    - Choose your desired language
      - Editor provides syntax highlighting
      - You can use your own editor, if you prefer
    - One entity in each file
    - Save the file in the project directory
    - Give to the file the same name of the entity
      - Makes debugging easier



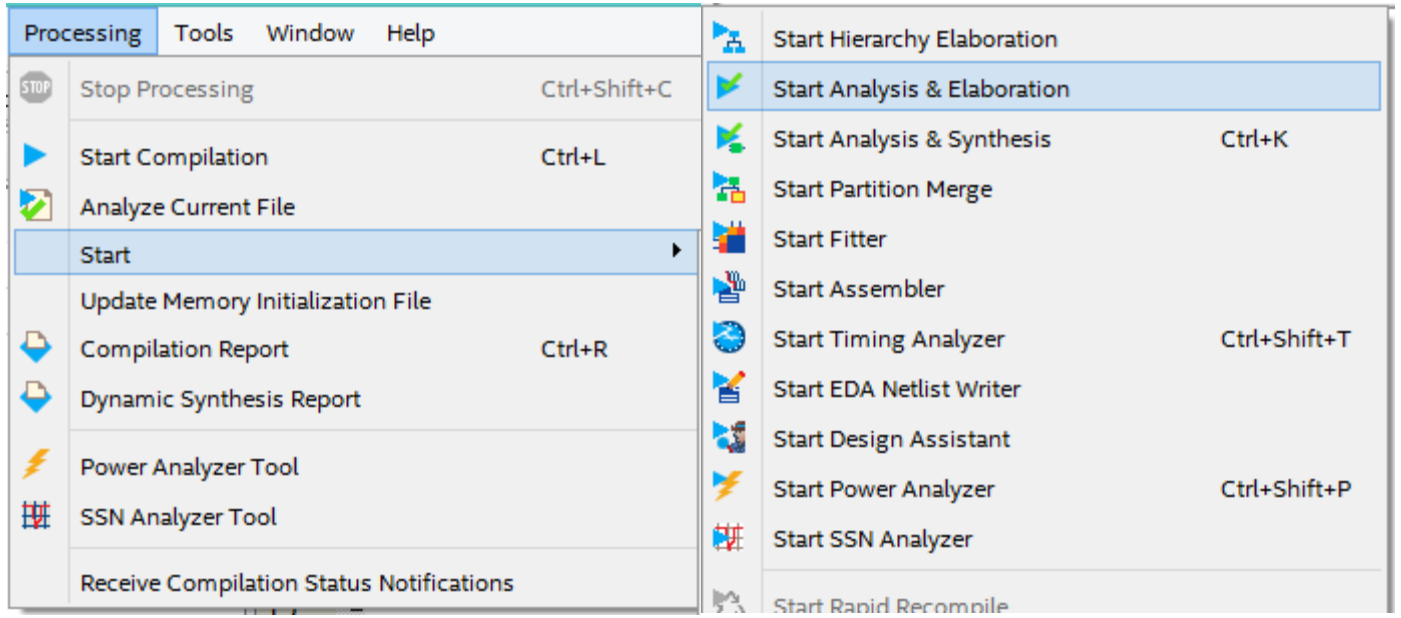


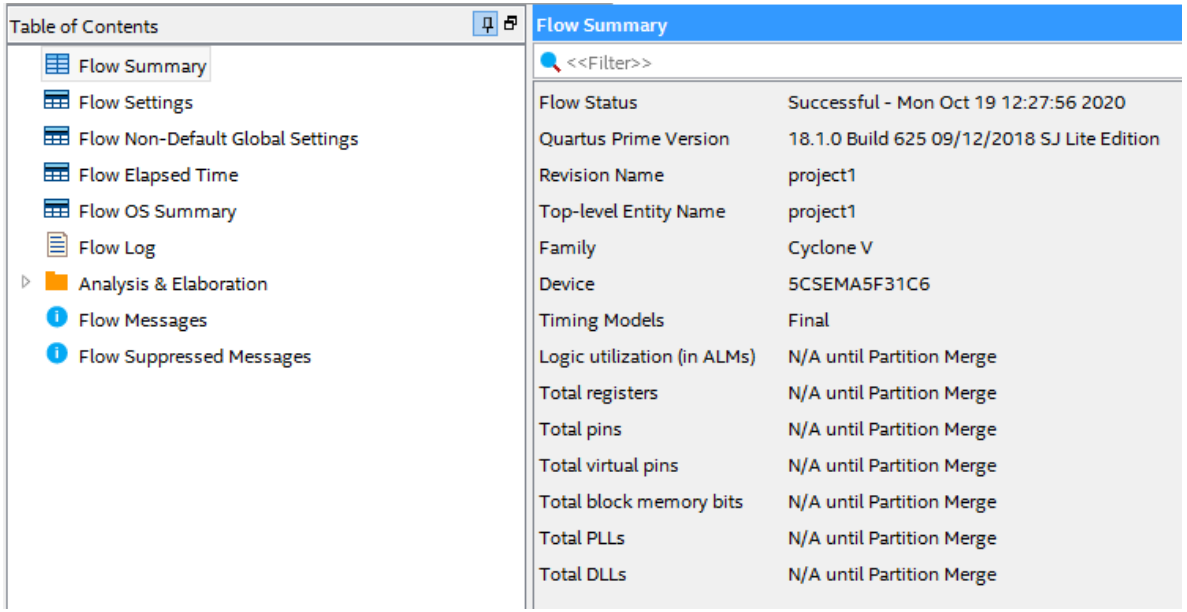

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity project1 is
5     port (
6        a, b: in std_logic;
7        y: out std_logic
8     );
9  end entity project1;
10
11 architecture rtl of project1 is
12 begin
13
14    y <= a and b;
15
16 end architecture rtl;
```

File name: project1.vhd

Quartus Prime

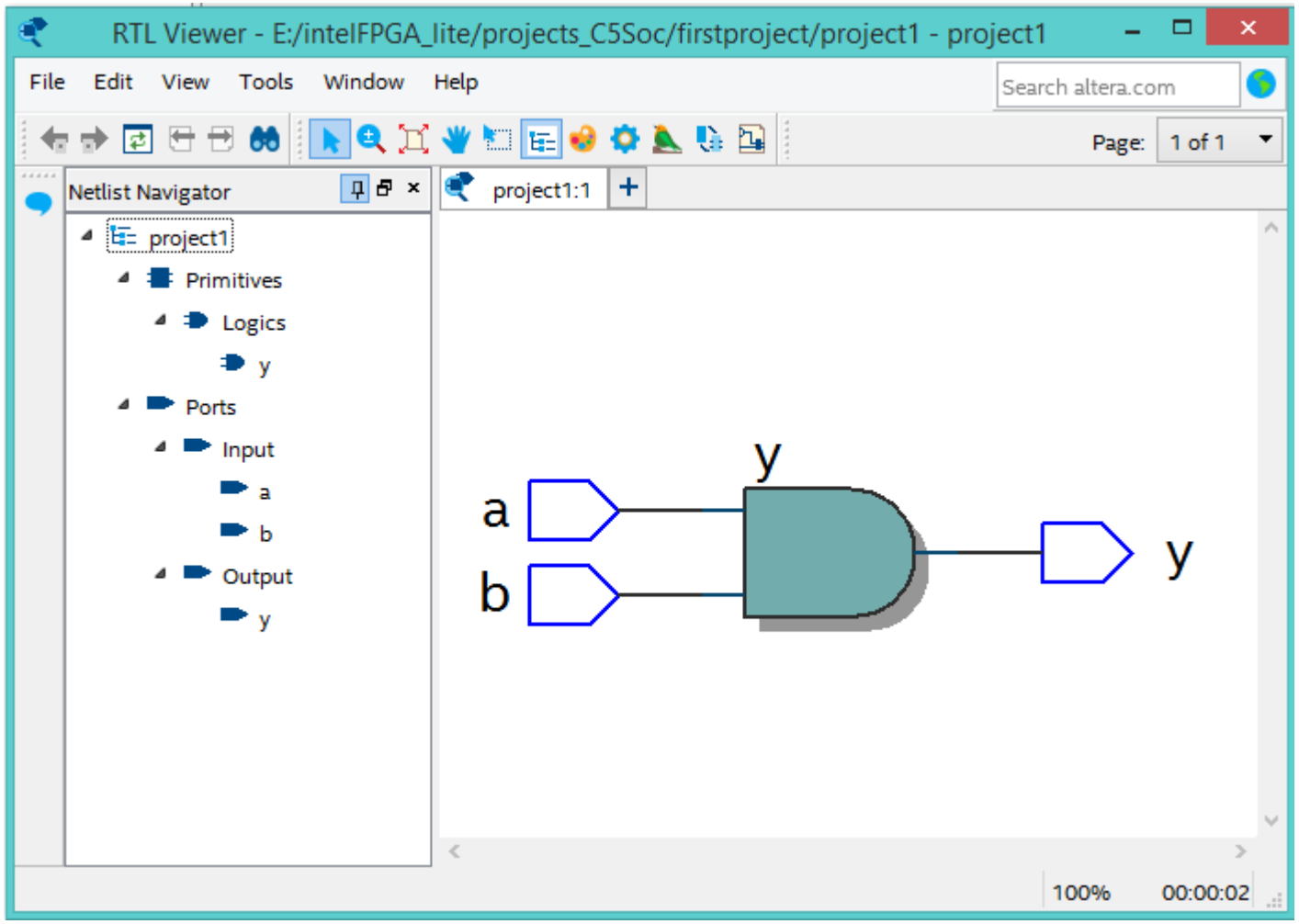
- Initial synthesis
  - Under Processing menu
    - Analyze Current File
      - Check syntax errors
      - Try inserting syntax errors to see what happens
      - Check also warnings
        - In particular Critical Warnings
    - Start → Start Analysis & Elaboration
      - Initial synthesis of the entire design
      - Generates an initial netlist
      - Generates an initial report





Quartus Prime

- Initial synthesis
  - Under Tools menu
    - Netlist Viewers → RTL Viewer



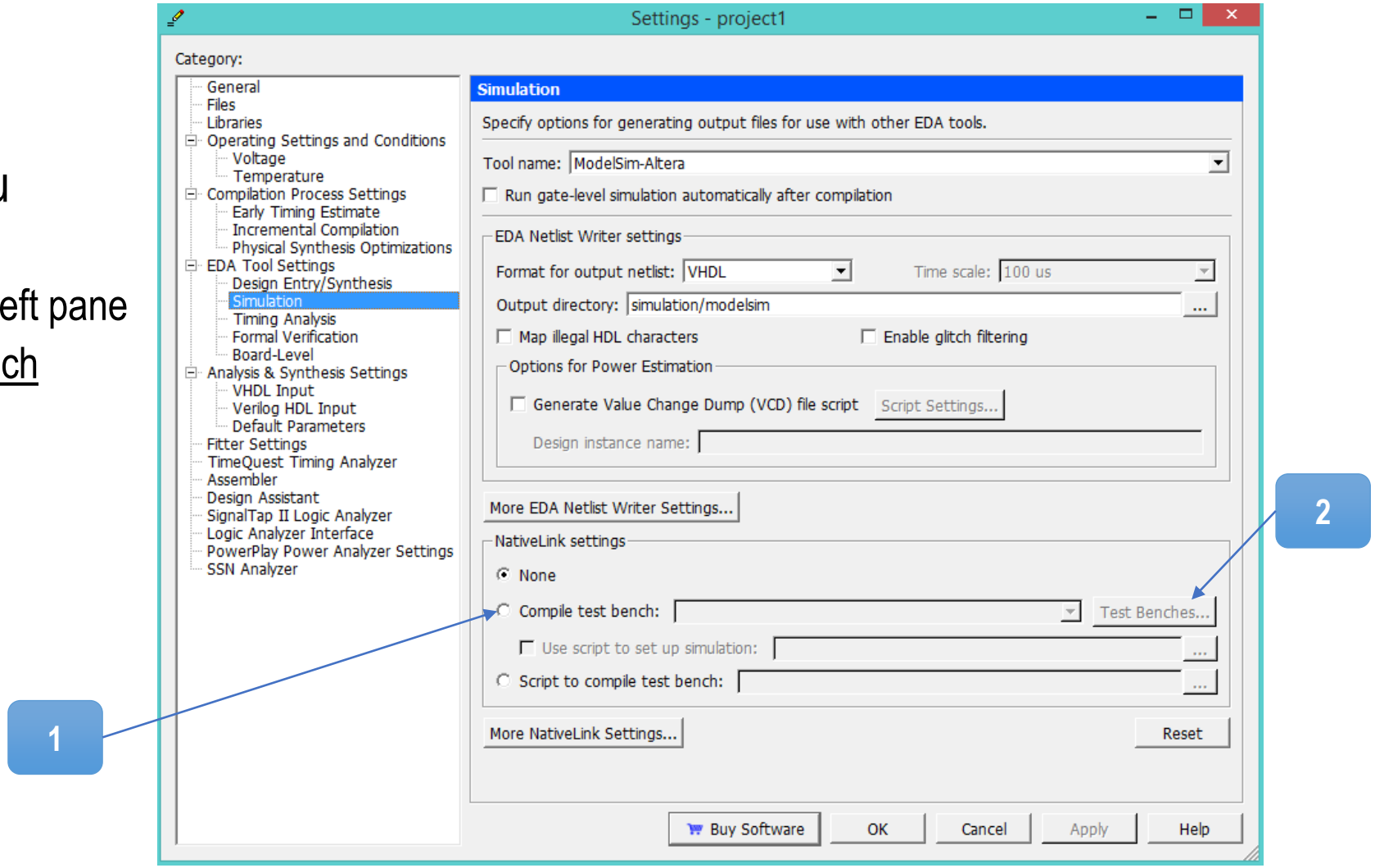
Quartus Prime

- Writing a testbench
  - Entity with no ports
  - Component is the Device Under Test (DUT)
    - Specify a label for the port map
    - Used later in simulation
  - Specify signals and initialize them
  - Specify a clock
    - Dedicated process
  - Specify stimuli
    - Another Process
    - Use wait to let time pass
      - It is not synthesizable
  - Save it in the project directory

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  use ieee.std_logic_arith.all;
5
6  entity testbench is
7  end entity;
8
9  architecture struct of testbench is
10
11    component project1 is
12       port (
13          a, b: in  std_logic;
14          y:    out std_logic
15       );
16    end component project1;
17
18    signal a: std_logic := '0';
19    signal b: std_logic := '0';
20    signal y: std_logic;
21
22    signal clk: std_logic := '0';
23    constant clk_period: time := 5 ns;
24
25 begin
26
27    dut: project1 port map (a, b, y);
28
29    myclock: process is
30       begin
31          clk <= '0';
32          wait for clk_period/2;
33          clk <= '1';
34          wait for clk_period/2;
35       end process myclock;
36
37    stim: process is
38       begin
39          a <= '0';
40          b <= '0';
41          wait for 20 ns;
42          a <= '1';
43          wait for 20 ns;
44          a <= '0';
45          b <= '1';
46          wait for 20 ns;
47          a <= '1';
48          wait;
49       end process stim;
50
51 end architecture struct;
```

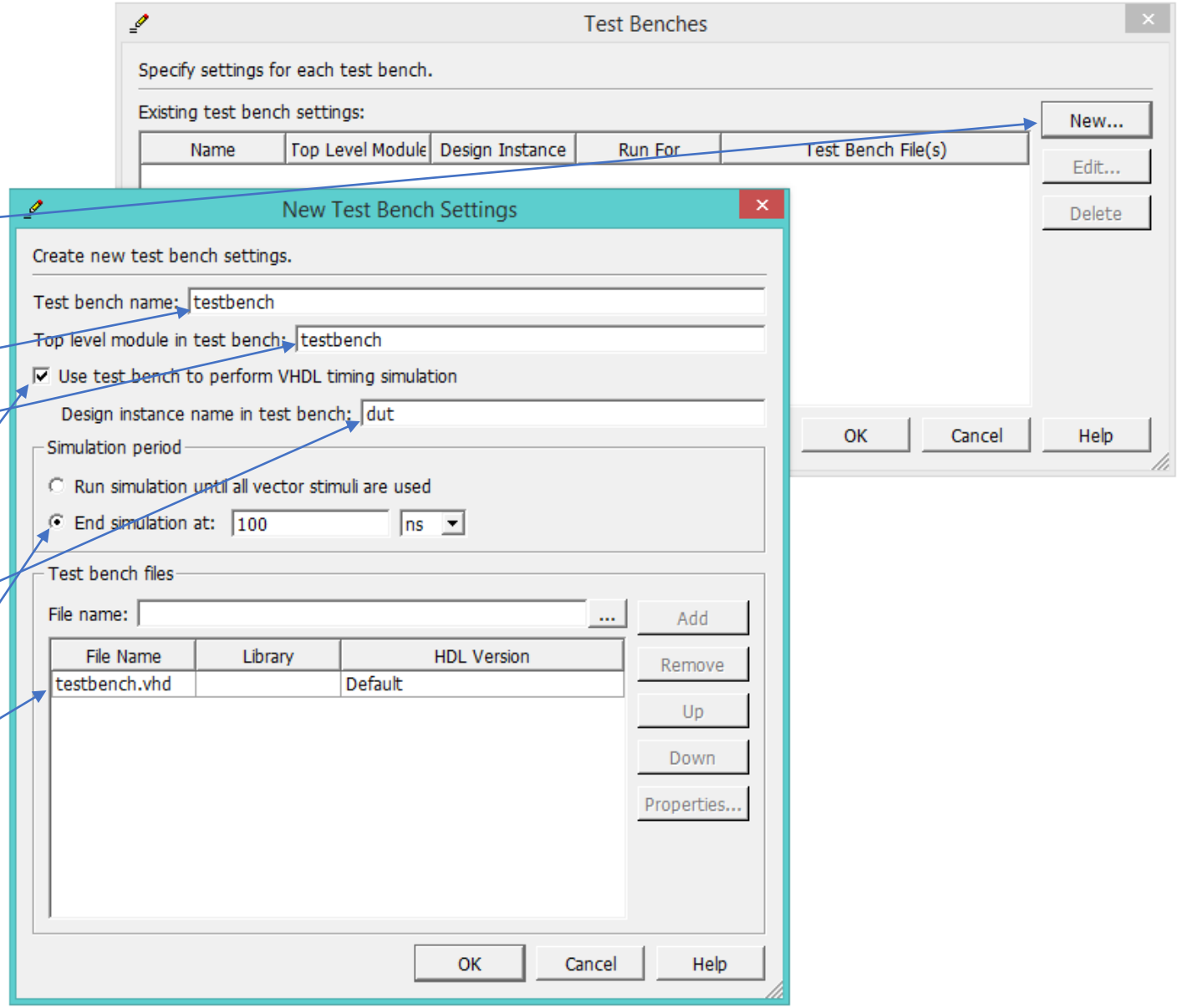
Quartus Prime

- Configure Simulator
  - Under Assignments menu
    - Choose Settings…
    - Select Simulation on the left pane
    - Click on Compile test bench
    - Click on Test Benches…



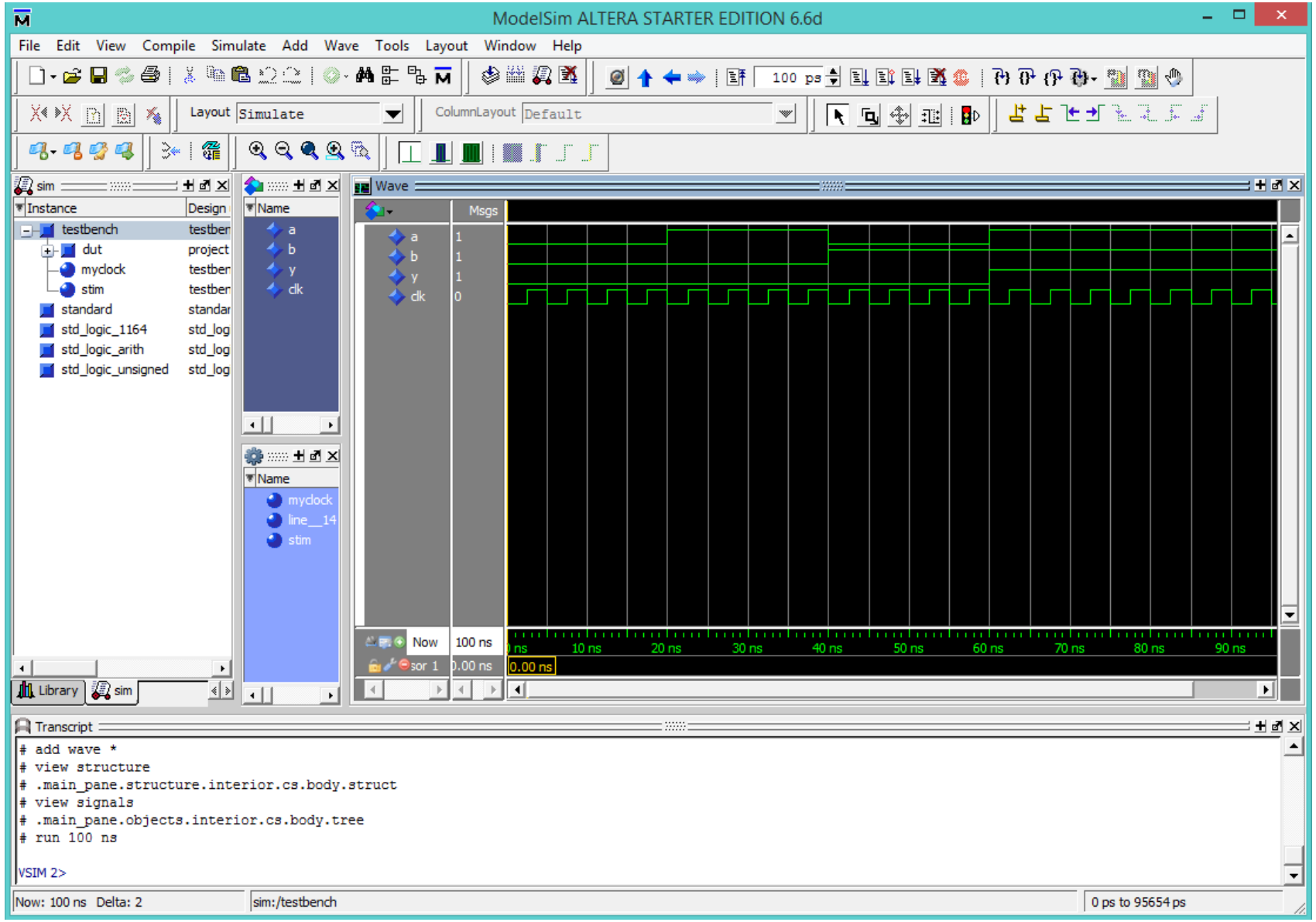
Quartus Prime

- Configure Simulator
  - Click on New
  - Fill in the New Test Bench Settings
    - Test bench name
    - Top level module in test bench
    - Activate Use test bench to perform VHDL timing simulation
    - Give the Design instance name in test bench
      - It is the label of the component
    - Choose the Simulation period
    - Add the testbench file
    - Click OK
      - Three times (three windows)



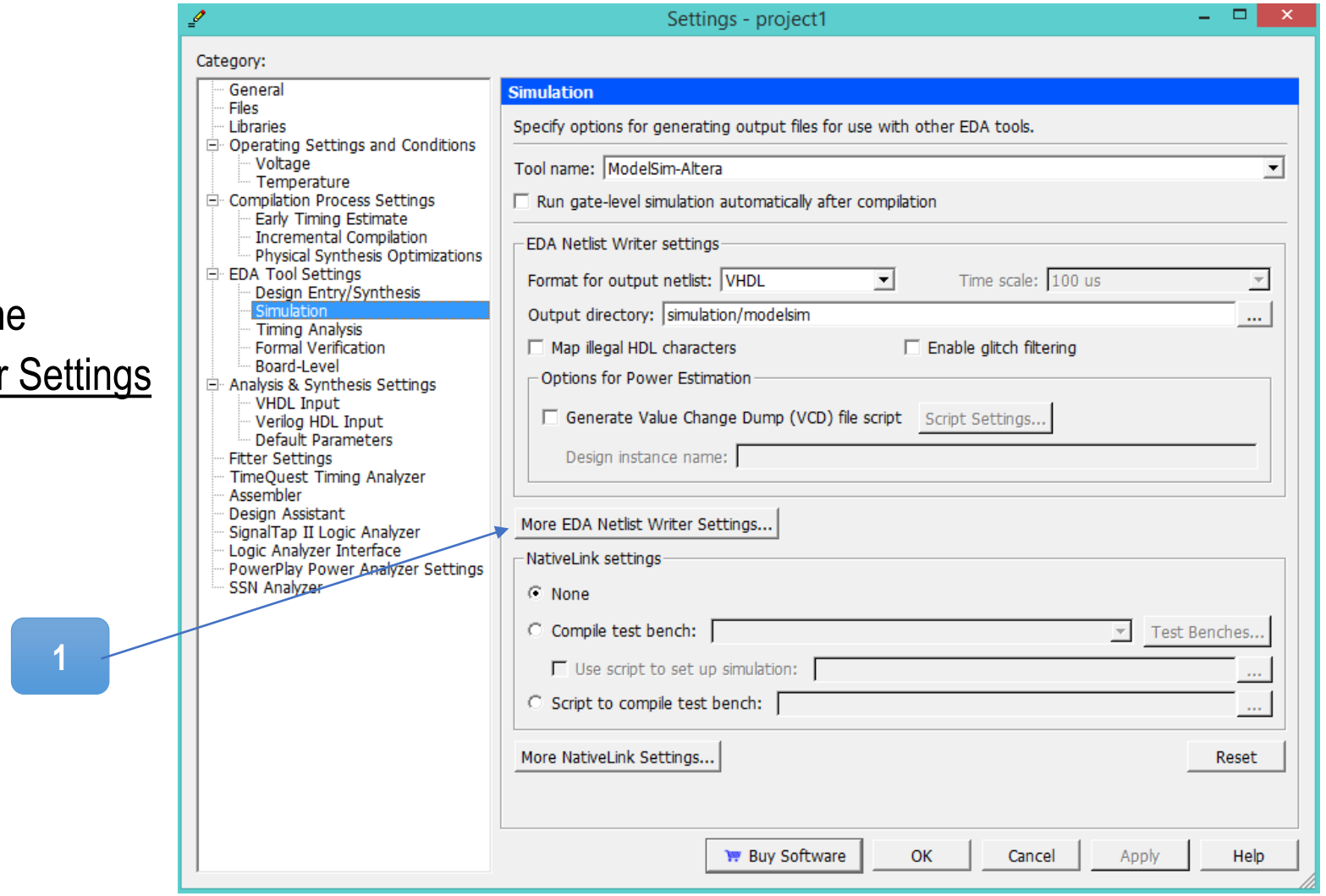
Quartus Prime

- Run Simulator
  - Under Tools menu
    - Choose Run Simulation Tool
    - Select RTL Simulation
    - If there are no errors, the simulation starts
    - You may need to reduce the zoom level in the wave window



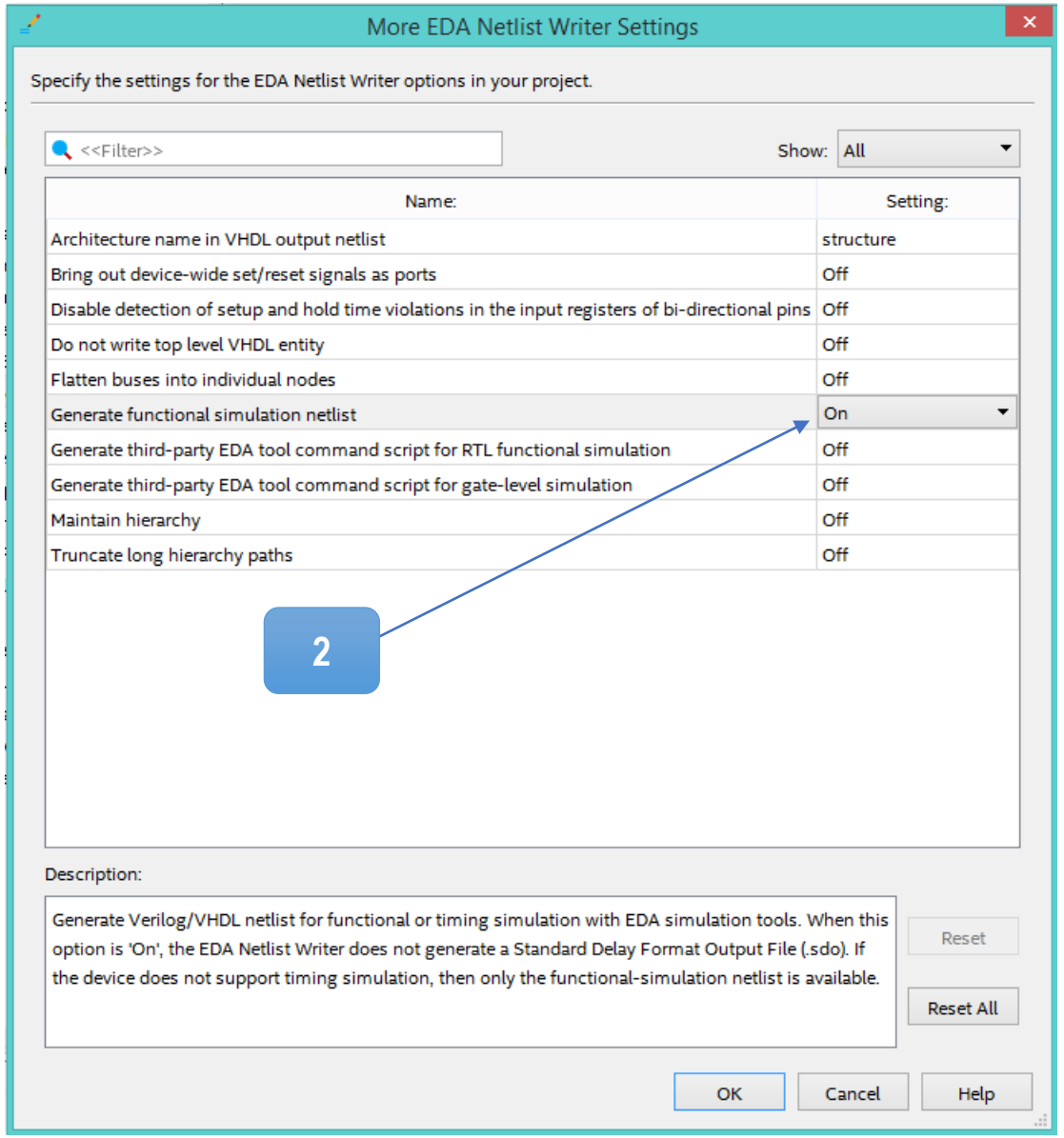
Quartus Prime

- Configure Simulator
  - Under Assignments menu
    - Choose Settings…
    - Select Simulation on the left pane
    - Click on More EDA Netlist Writer Settings



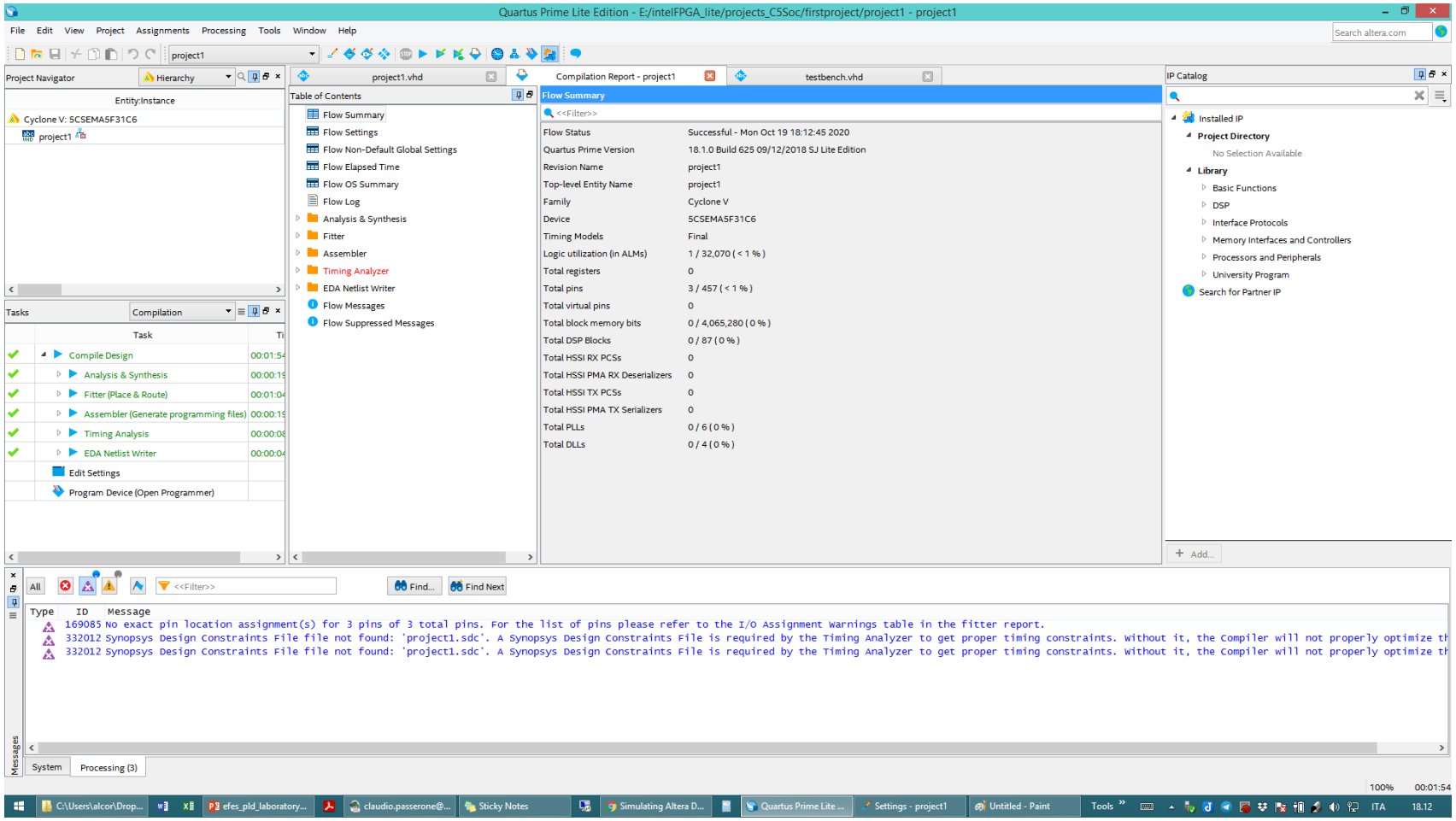
Quartus Prime

- Configure Simulator
  - Under Assignments menu
    - Choose Settings…
    - Select Simulation on the left pane
    - Click on More EDA Netlist Writer Settings
    - Enable Generate functional simulation netlist
      - Needed for gate level simulation
      - But Intel does not provide timing simulation anymore
    - Click OK twice (in two windows)



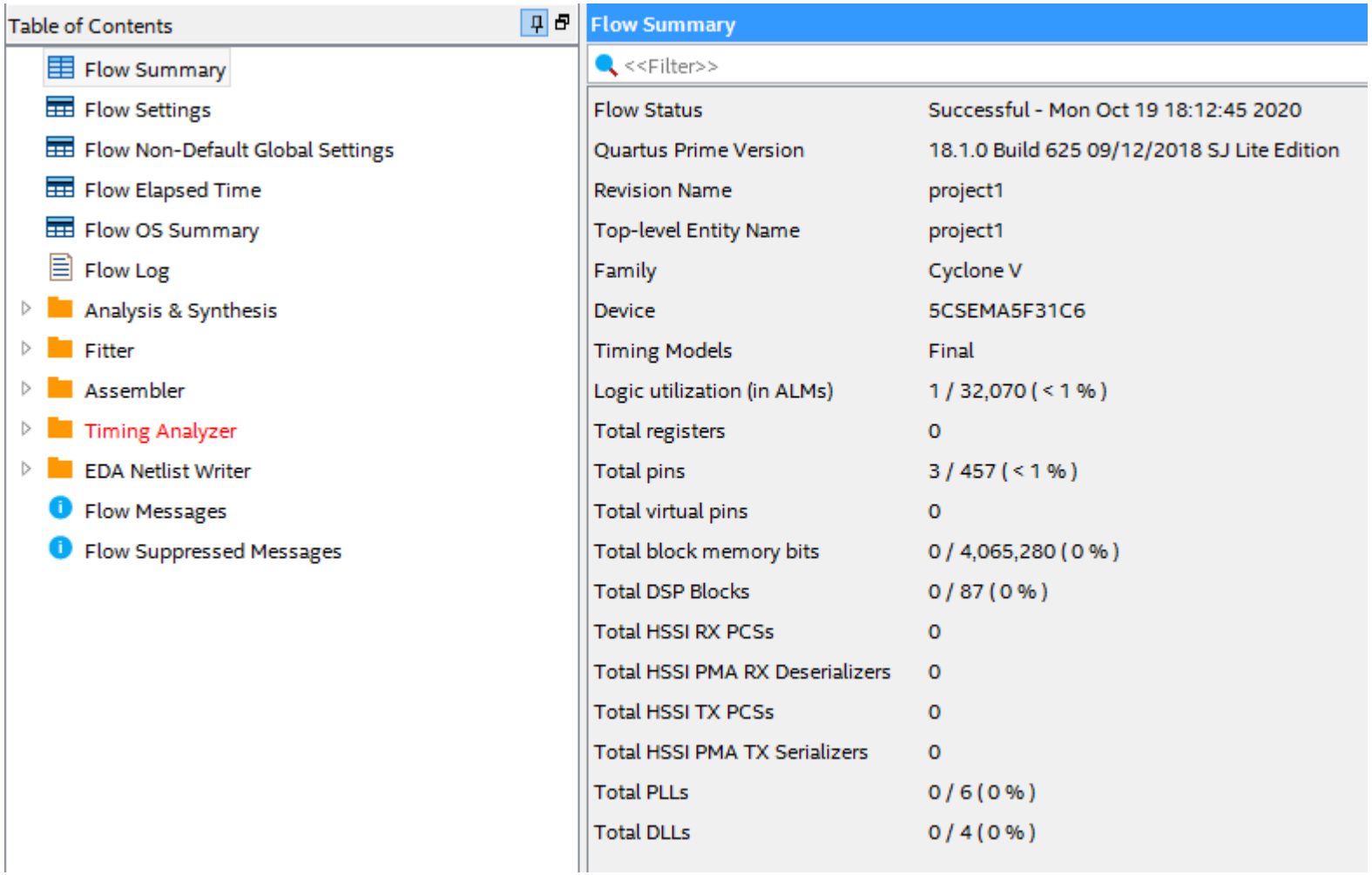
Quartus Prime

- Place & Route
  - Under Processing menu
    - Select Start Compilation
    - Several steps are performed
      - Analysis
      - Fitter
      - Assembler
      - Timing Analysis
      - EDA Netlist Writer
    - There are some Critical Warnings
      - Pins
      - Timing constraints



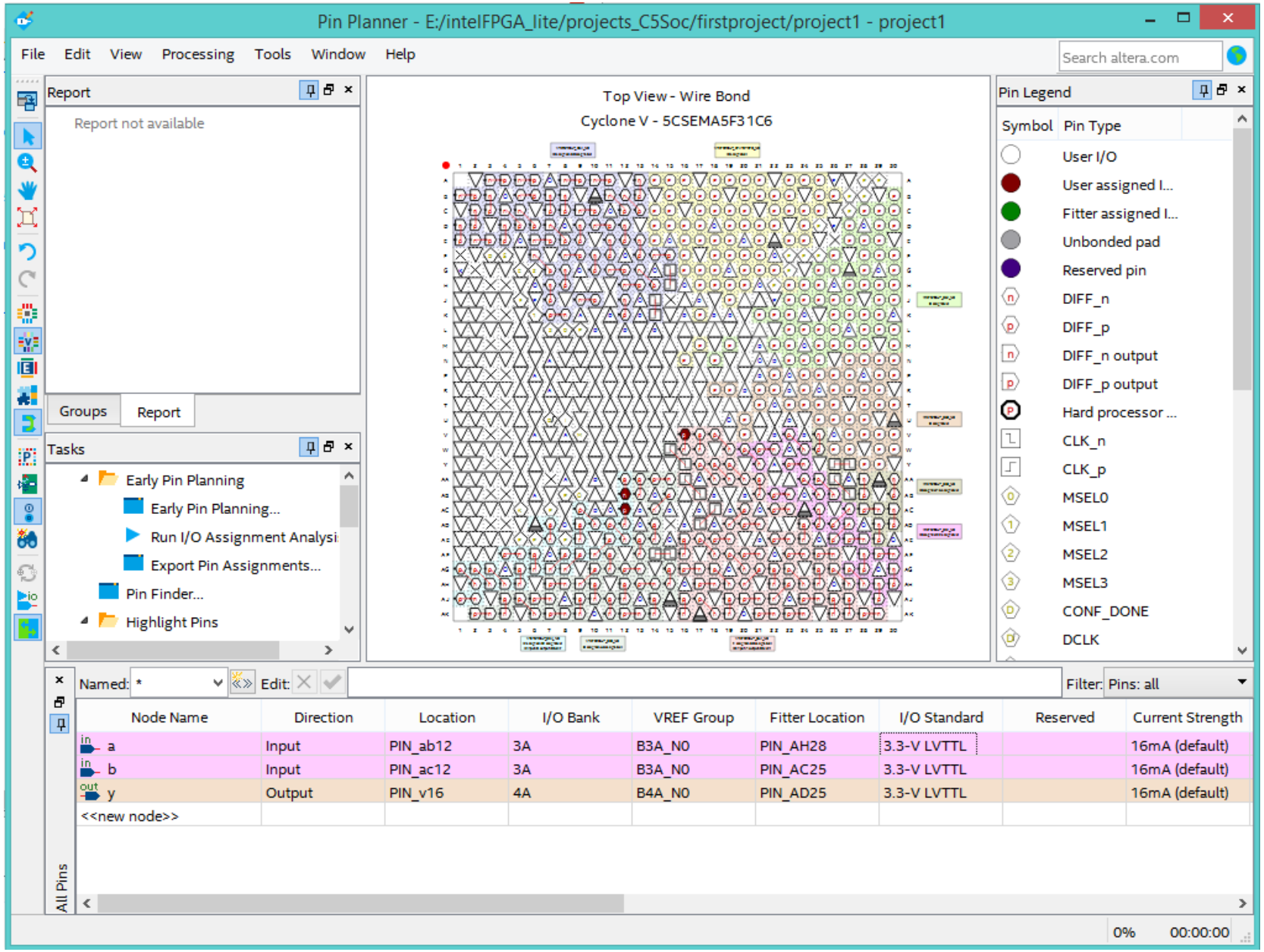
Quartus Prime

- Place & Route
  - Compilation report
    - It gives more information
      - Logic Elements
      - Registers
      - Pins
      - Memory
    - Expand the various tool reports



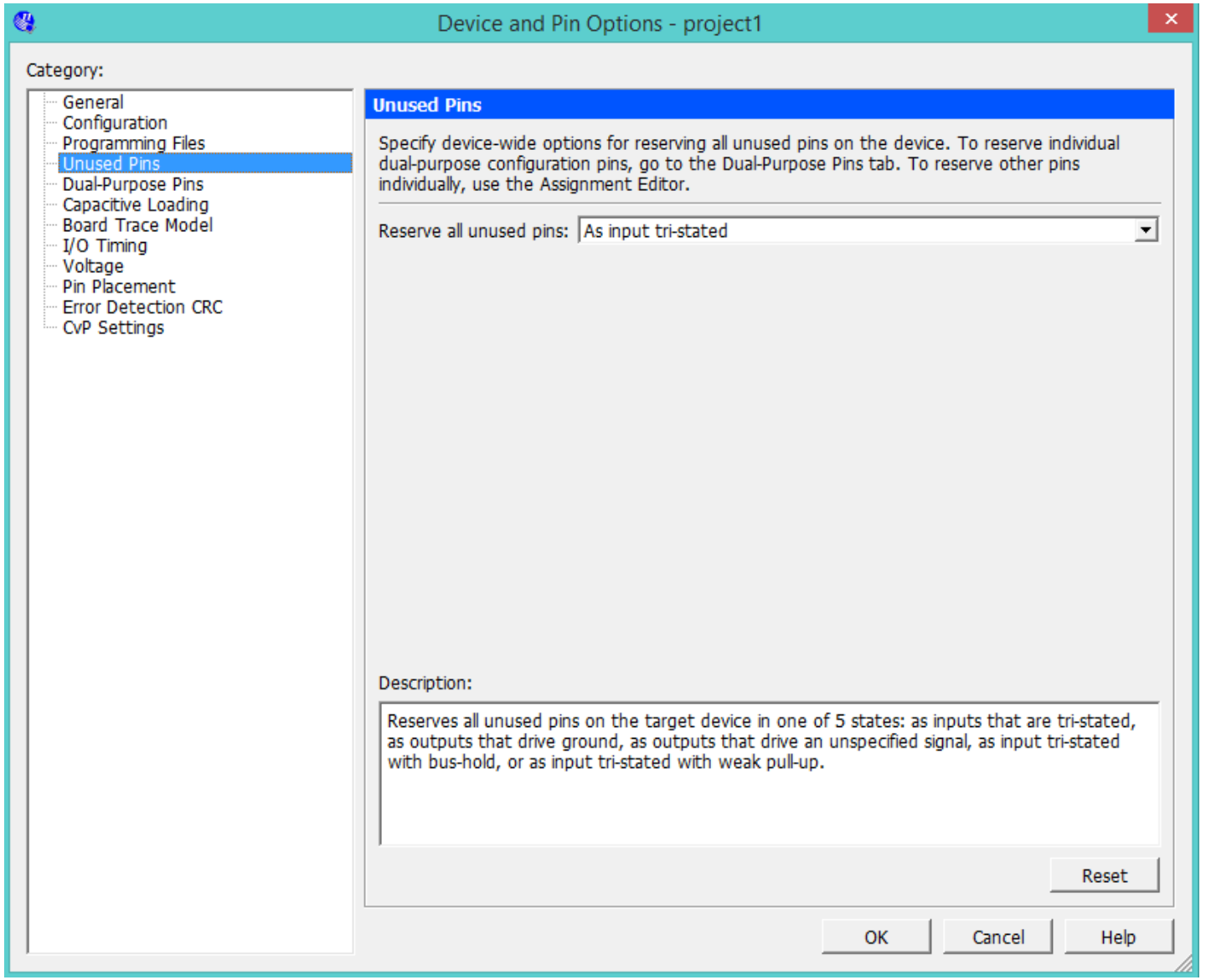
Quartus Prime

- Assign pins
  - Under Assignments menu
    - Choose Pin Planner
      - Assign pin locations
      - Check
        - I/O standard
        - Current Strength
      - If a pin does not show
        - Run compilation again
      - You don't need to save
      - But you need to recompile the entire design if any pin is changed



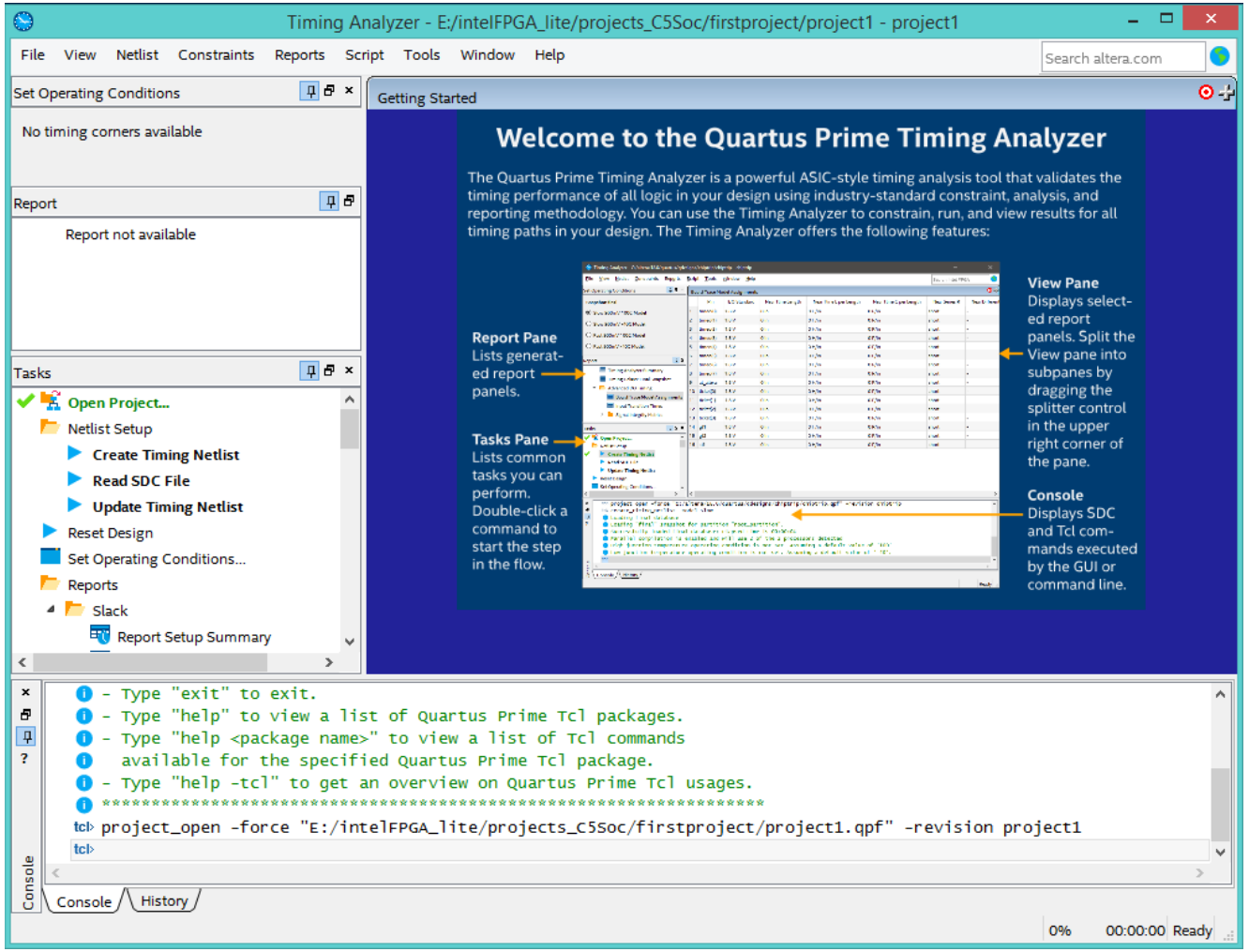
Quartus Prime

- Assign unused pins
  - Under Assignments menu
    - Choose Device…
      - Click Device and Pin Options…
      - Select Unused Pins in the left pane
      - Select As input tri-stated in the right pane



Quartus Prime

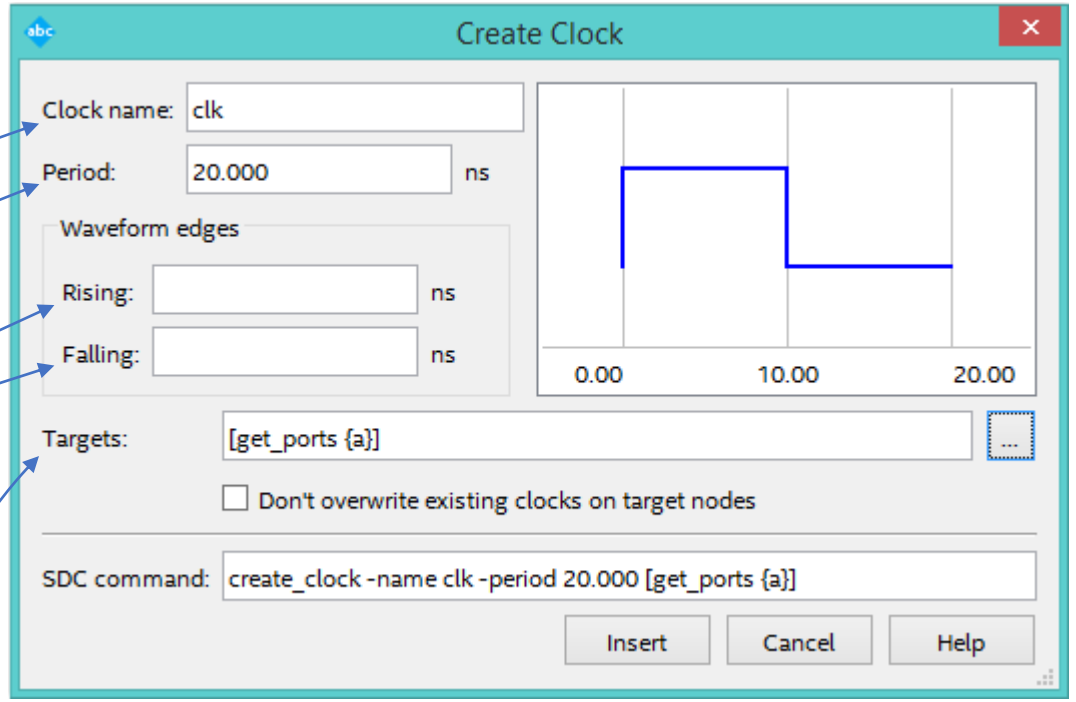
- Specify Timing Constraints
  - Under Tools menu
    - Choose Timing Analyzer
    - In the new window
    - Choose Netlist Setup → Create Timing Netlist
    - Create a new SDC file
      - Choose File → New SDC File
      - Or open an existing SDC File
    - The SDC file opens in the Quartus Prime window
    - Right click on the SDC file area and choose Insert Constraint
      - Or use the Constraints menu on the Timing Analyzer



Quartus Prime

- Specify Timing Constraints
  - For a clock
    - Choose Create Clock…
      - Specify a name
      - Give the clock period
      - Specify rising and falling instants
        - Leave them blank for 50% duty cycle clock
      - Identify the target pin
    - Click Insert

    - If using the Constraints menu on the Timing Analyzer
      - Click Run
      - Then write the SDC file with Constraints → Write SDC File…

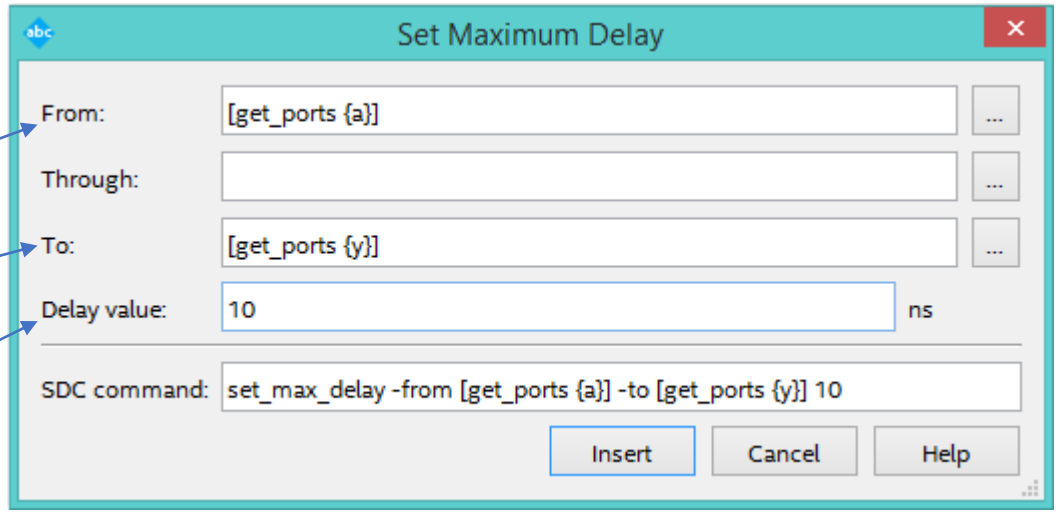


For combinational logic circuits, like the simple AND gate, you don't need to specify a constraint for the clock
- A clock does not exist!

Quartus Prime

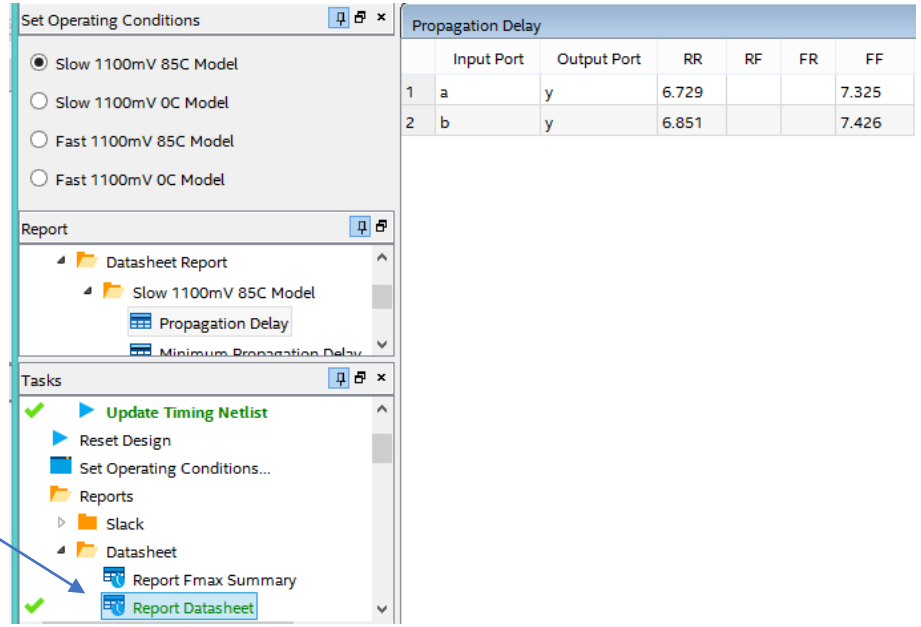
- Specify Timing Constraints
  - For combinational logic
    - Choose Set Maximum Delay…
      - Specify the input port
      - Specify the output port
      - Specify the maximum propagation delay
    - Click Insert

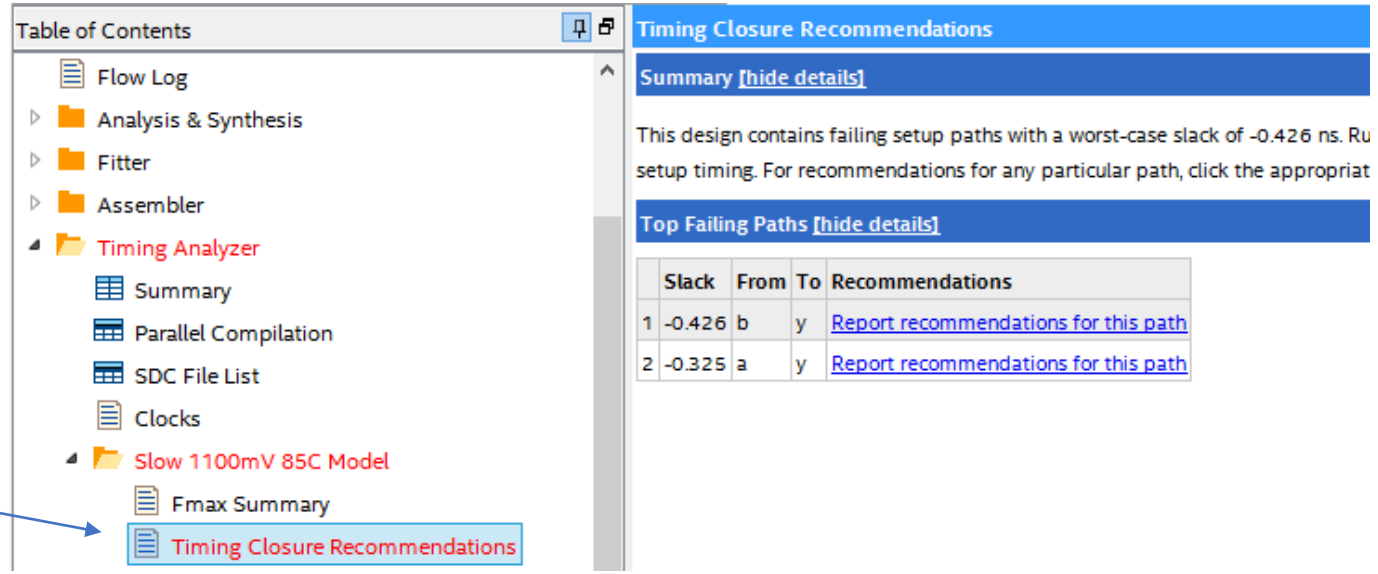
    - If using the Constraints menu on the Timing Analyzer
      - Click Run
      - Then write the SDC file with Constraints → Write SDC File…



Quartus Prime

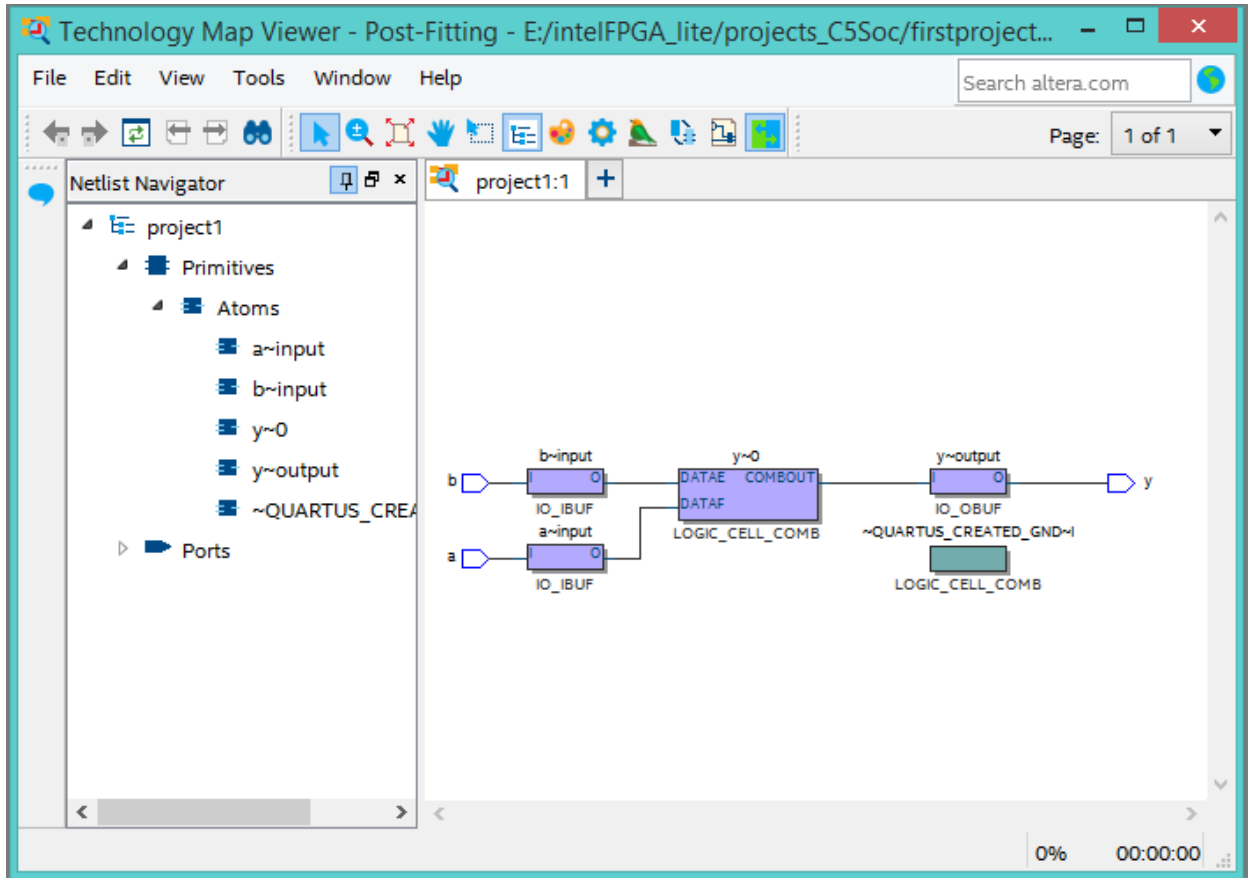
- Check Timing Analysis
  - In the Timing Analyzer
    - Check the Report Datasheet under the Reports
    - If timing constraints are not met, check also the Timing Analyzer in the Compilation Report in Quartus
      - For a sequential circuit, check the Fmax Summary
    - Changing the pin assignments affects performance
    - Changing the timing constraints affects performance
  - When timing constraints are not met check the Timing Closure Recommendations
    - Slack is the error





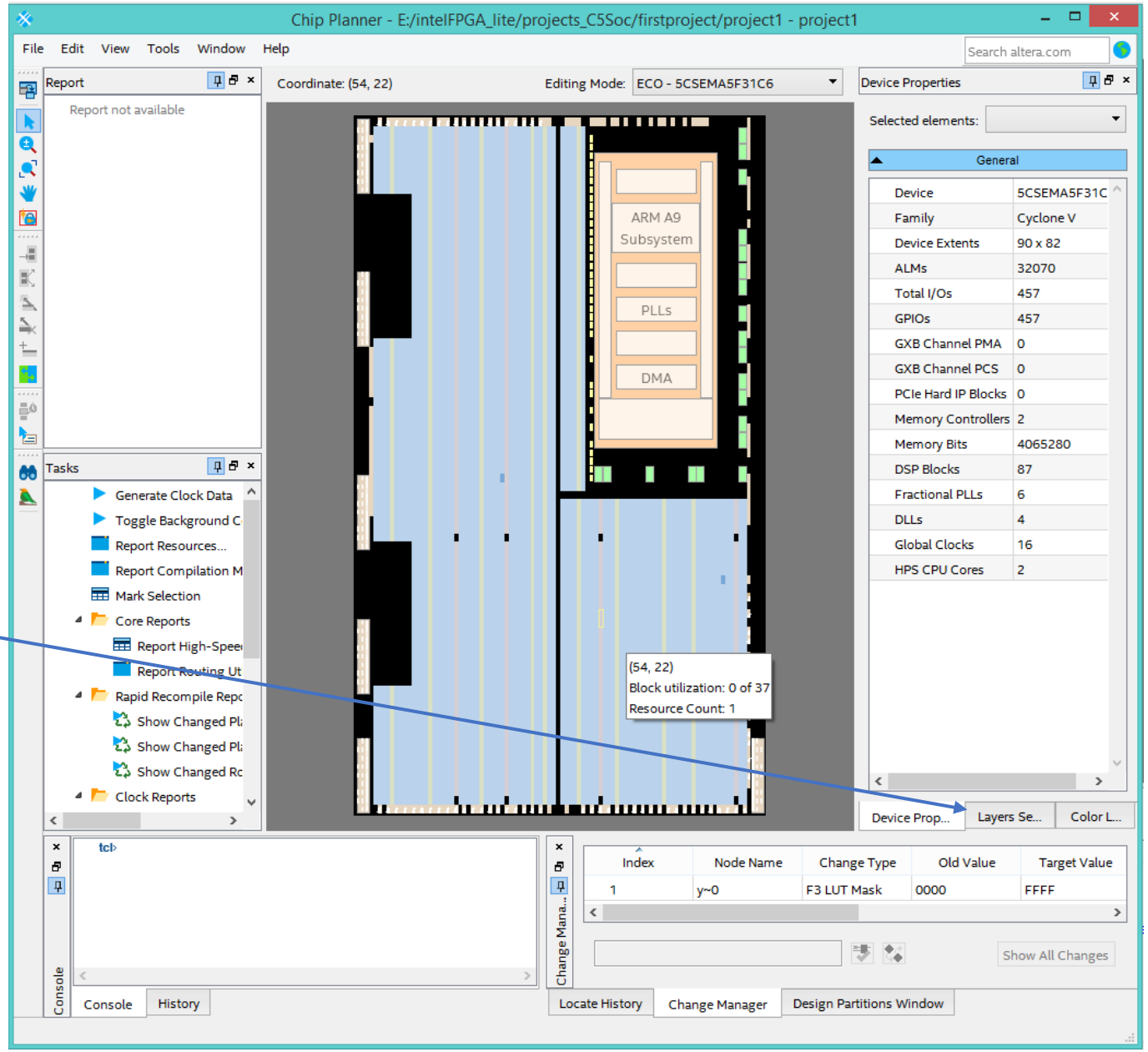
Quartus Prime

- Place & Route
  - Under Tools menu
    - Under Netlist Viewers
      - Technology Map Viewer (Post-Mapping)
      - Technology Map Viewer (Post-Fitting)
    - Right click on an object
      - Choose Locate Node
        - Can locate the node in various tools
          - Pin Planner
          - Chip Planner
          - Design File
          - …



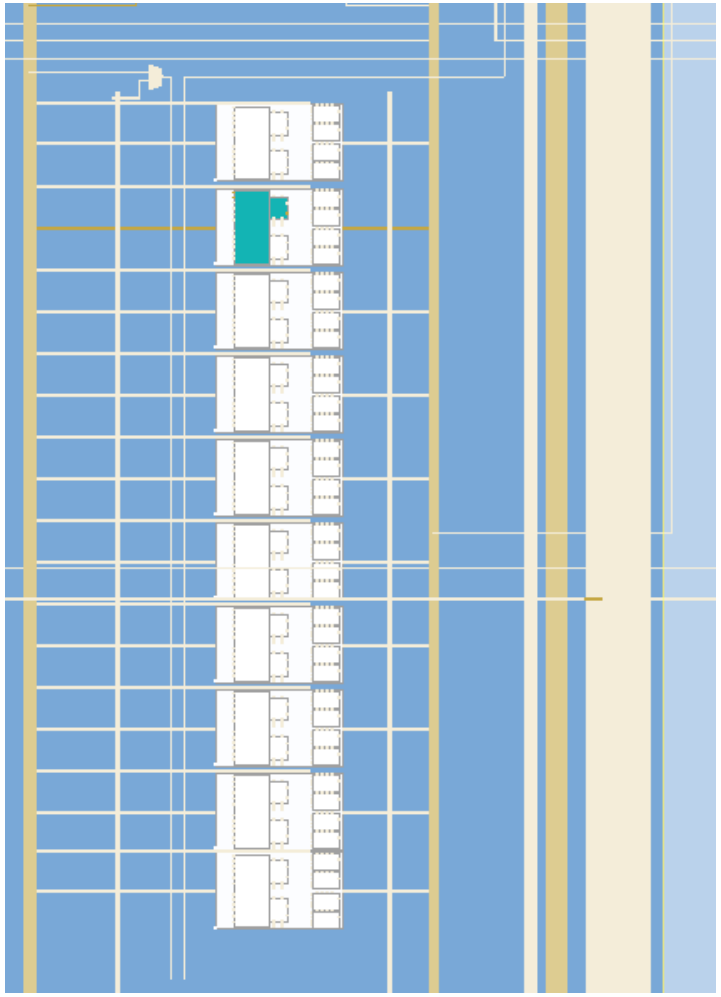
Quartus Prime

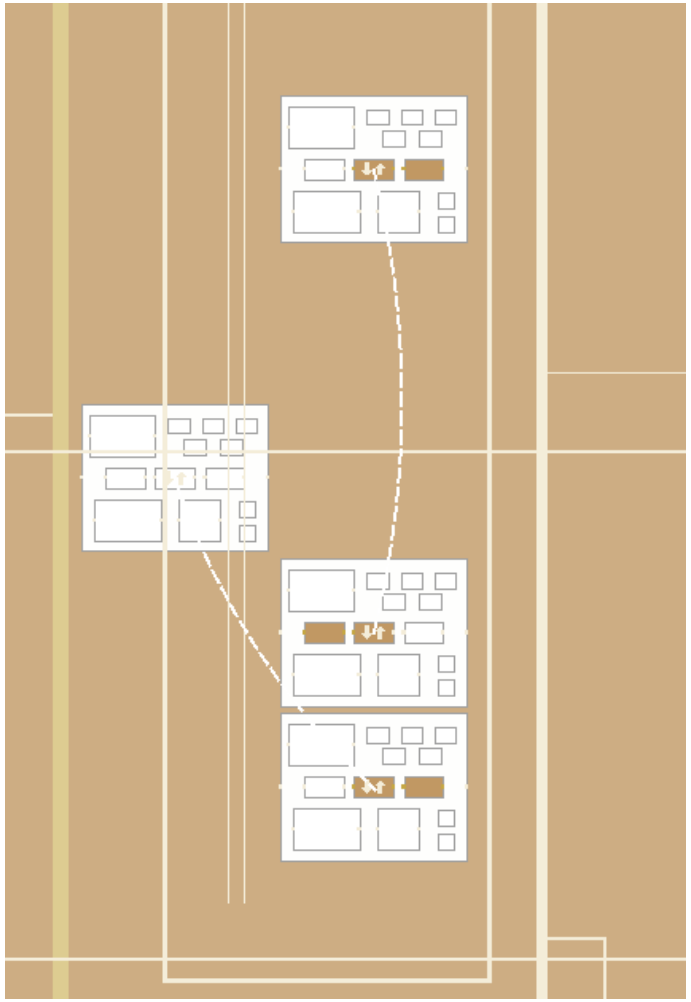
- Floorplan
  - Under Tools menu
    - Choose Chip Planner
      - You can zoom in different areas of the chip
      - You can select different resources to show
        - Under Layers Settings
        - Logic Element utilization
        - Routing utilization
        - Select additional details



Quartus Prime

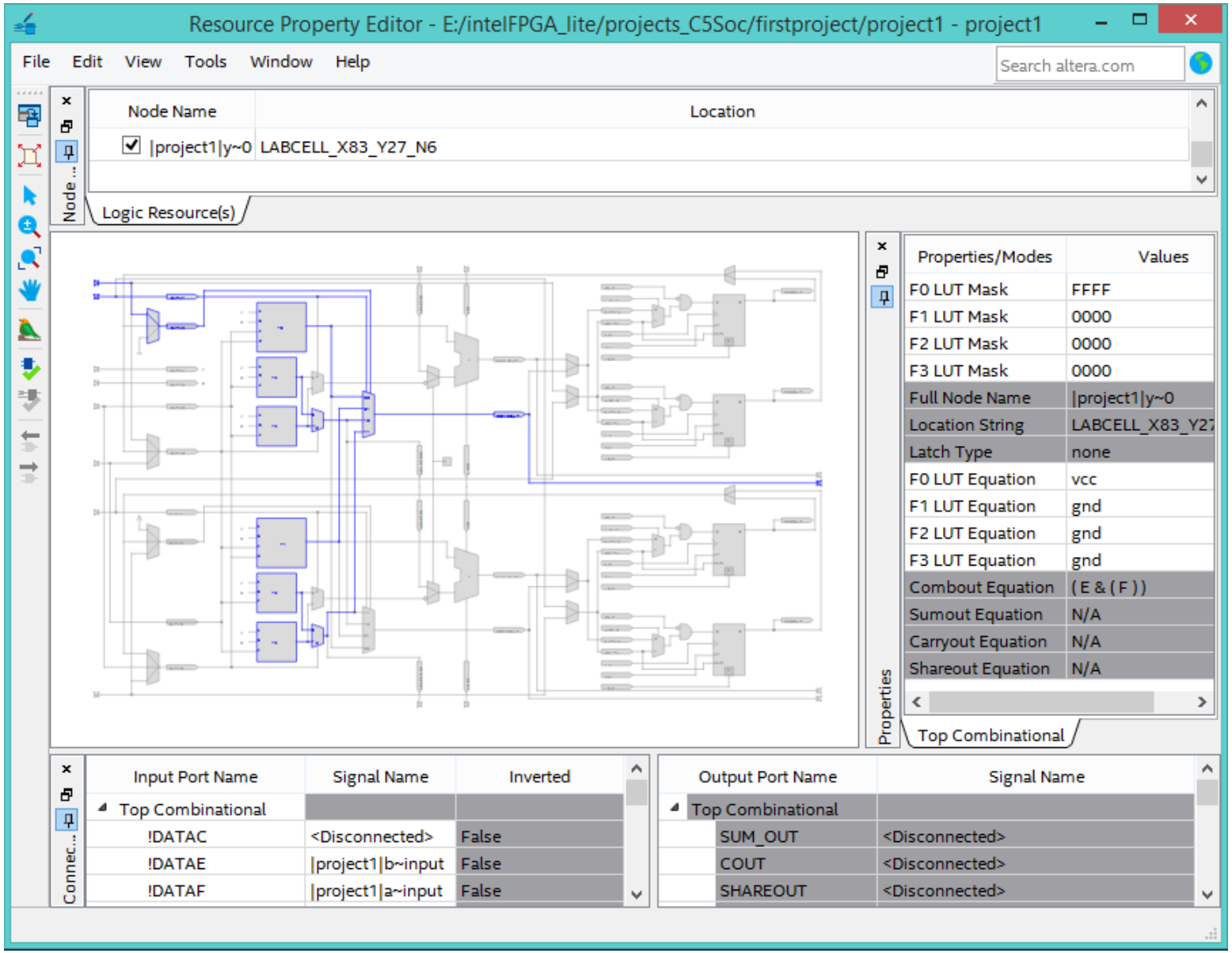
- Floorplan
  - Zoom on a Logic Array Block and Input/Output Block
    - The second Logic Element is used
    - Three I/O Elements are used
    - Additional routing details enabled





Quartus Prime

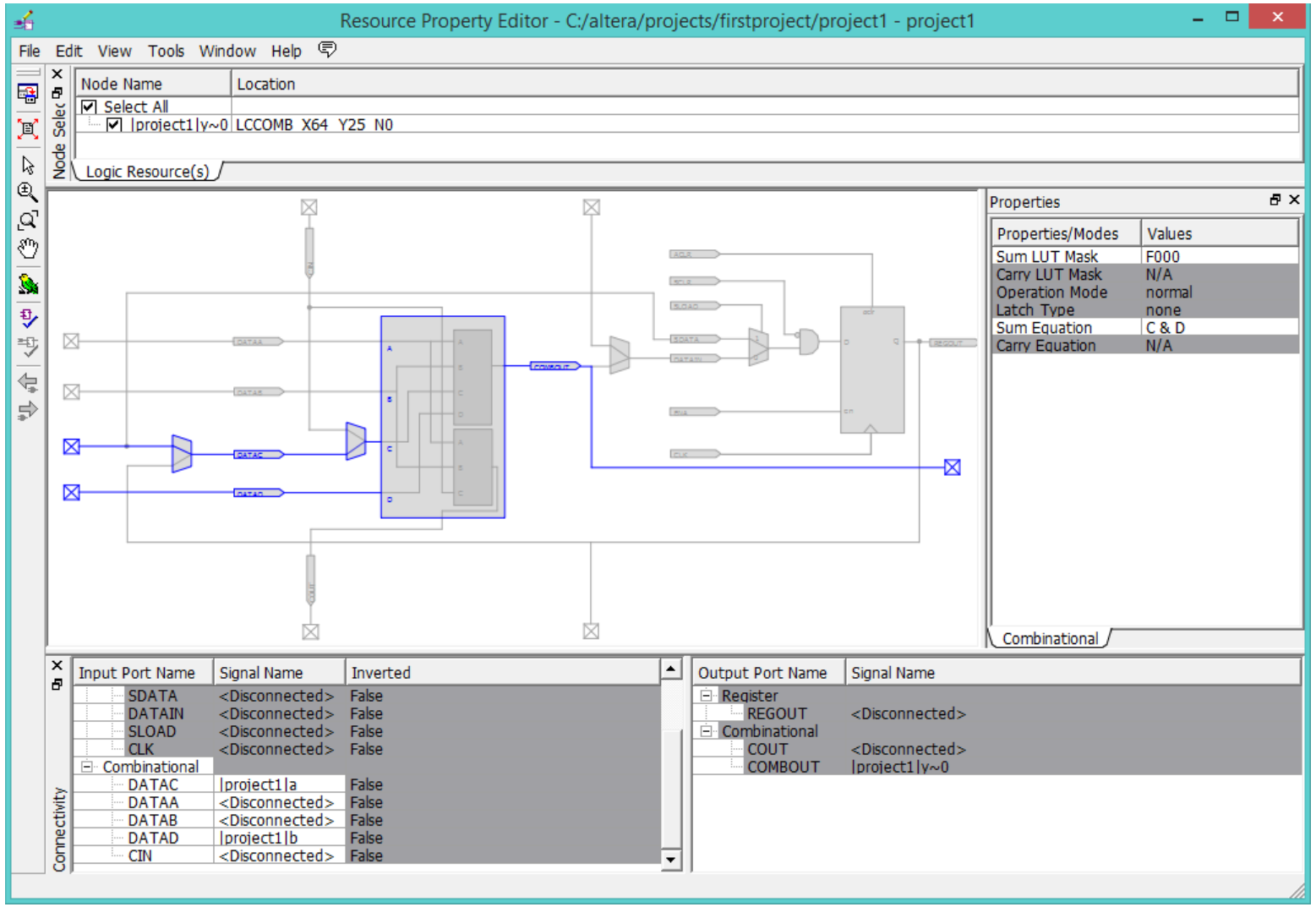
- Floorplan
  - Double click on a Logic Element
    - See how it is used
    - The LUT content is in the column on the right
    - Input and Output ports are listed at the bottom
    - Blocks used are highlighted



Quartus II

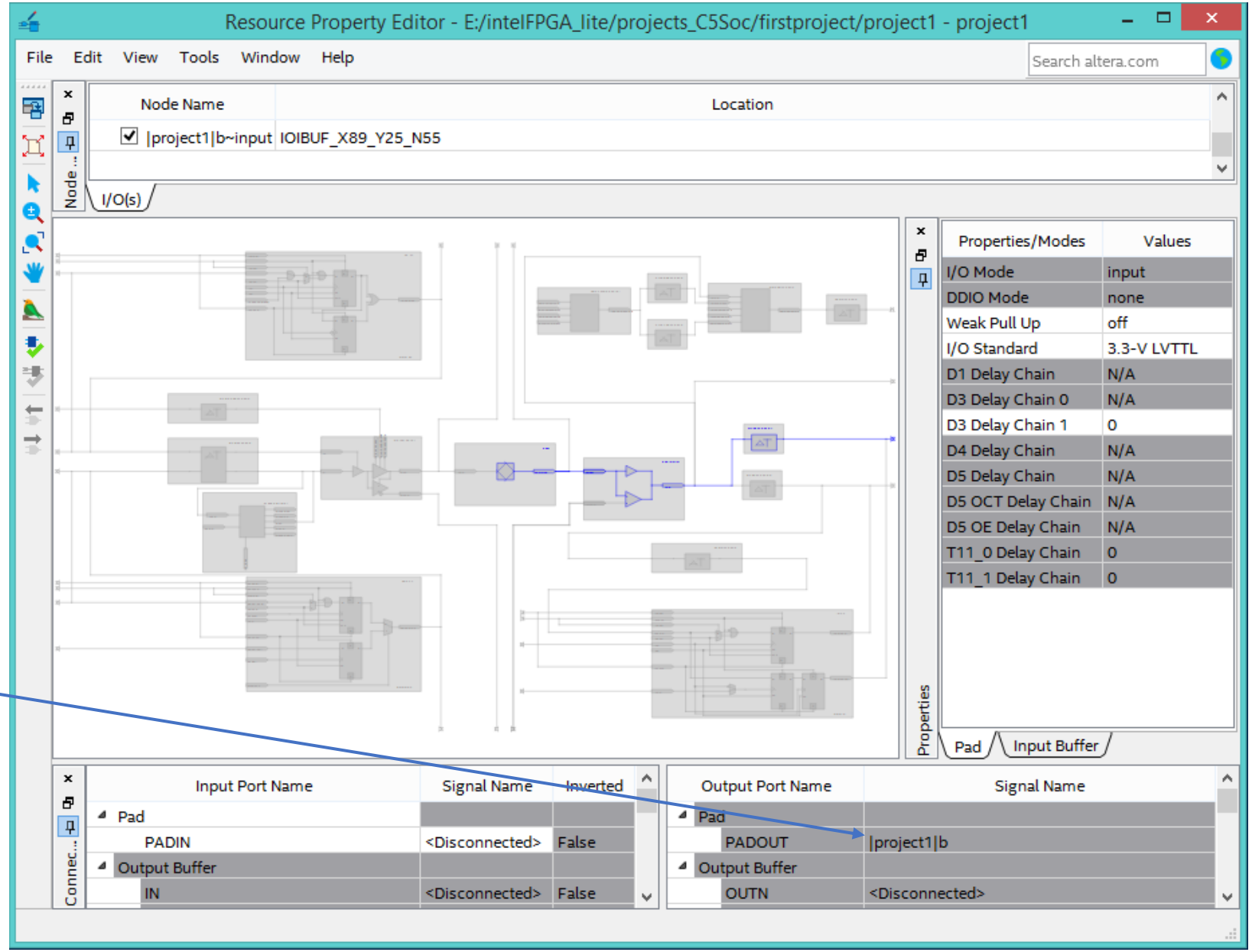
- Floorplan
  - Double click on a Logic Element
    - With a Cyclone II FPGA
    - LE similar to a Cyclone 10 LP
    - See how it is used
    - The picture on the right is from Quartus 11.1sp1
      - With Quartus 18 it looks slightly different

This is an older version of Quartus using a *vintage* Cyclone II FPGA, which has a different logic element compared to the Cyclone V



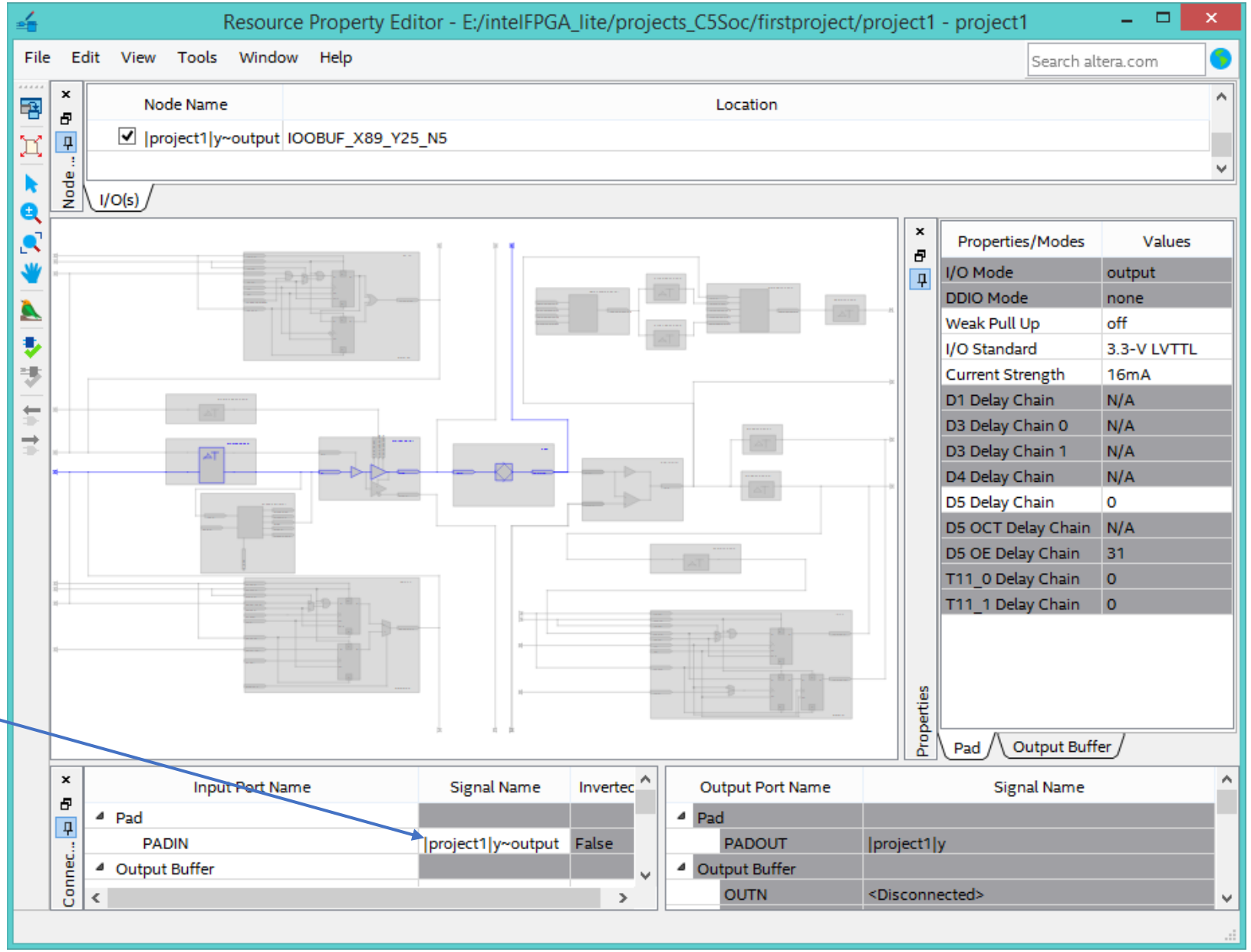
Quartus Prime

- Floorplan
  - Double click on an Input/Output Element
    - See how it is used
      - Input
      - Output
    - The picture on the right is for input *b*
    - It is an output of the I/O Element



Quartus Prime

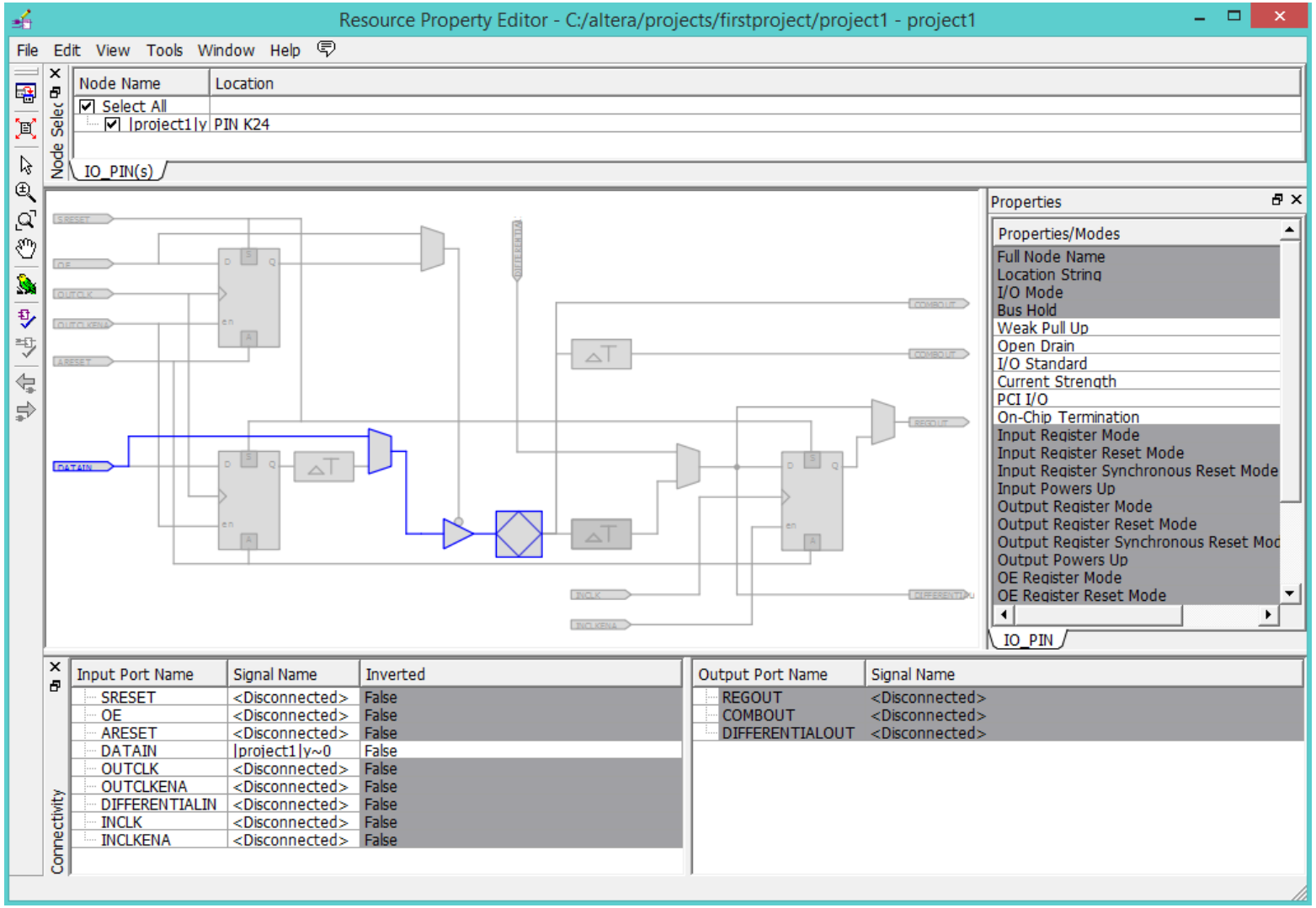
- Floorplan
  - Double click on an Input/Output Element
    - See how it is used
      - Input
      - Output
    - The picture on the right is for output *y*
    - It is an input of the I/O Element



Quartus II

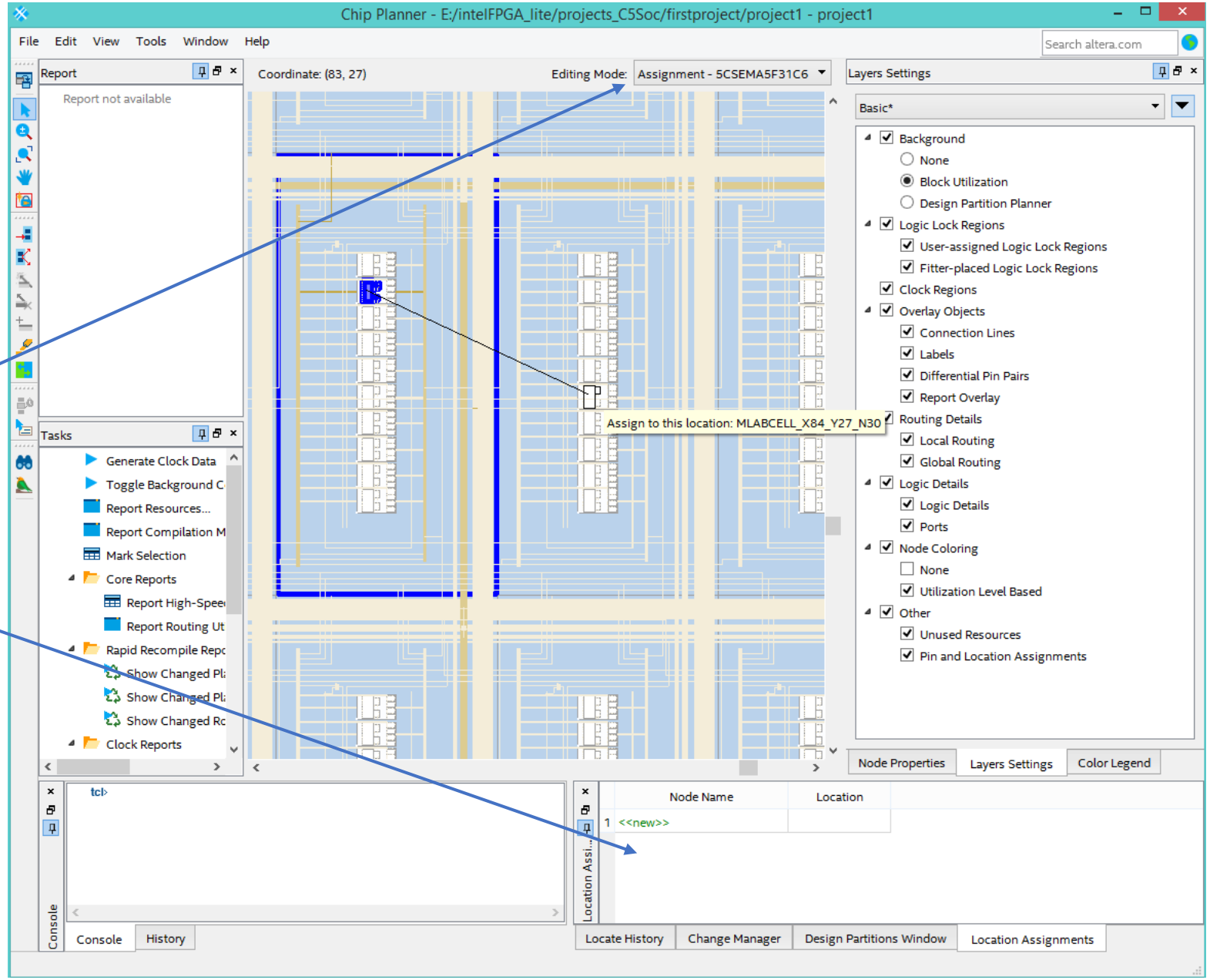
- Floorplan
  - Double click on an Input/Output Element
    - With a Cyclone II FPGA
    - IOE similar to a Cyclone 10 LP
    - See how it is used
    - The picture on the right is from Quartus 11.1sp1
      - With Quartus 18 it looks slightly different

This is an older version of Quartus using a *vintage* Cyclone II FPGA, which has a different logic element compared to the Cyclone V



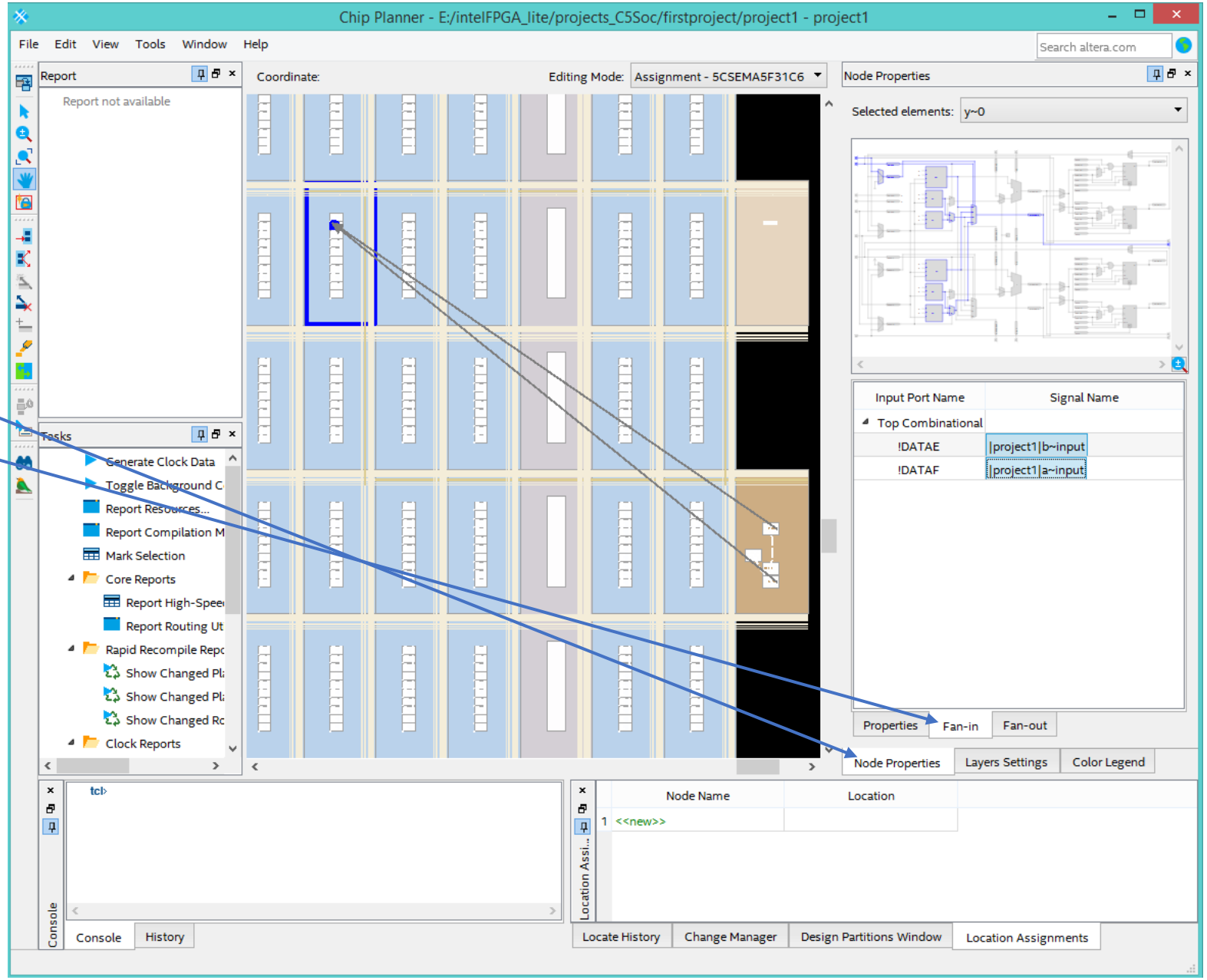
Quartus Prime

- Floorplan
  - You can drag objects to new locations
    - Set the Editing Mode to Assignment
    - Drag the object
    - The new assignment will appear at the bottom
    - If you open the Assignment Editor in Quartus under the Assignment menu you see the new location
    - Recompile to get the new floorplan



Quartus Prime

- Floorplan
  - You can highlight connections
    - Select a Logic Element
    - Click on Node Properties
    - Click on Fan-in or Fan-out
    - Click on signal names

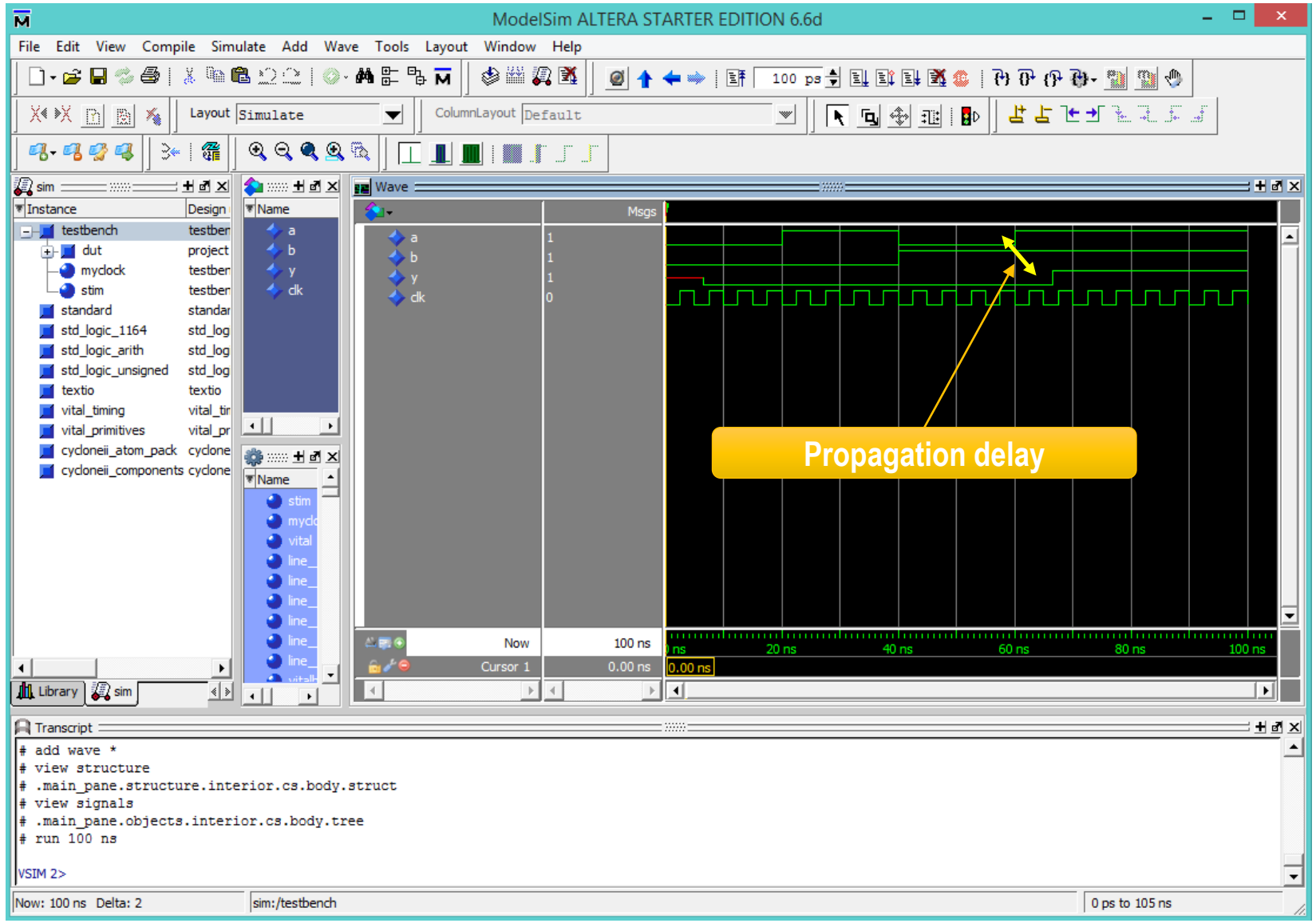


Quartus Prime

- Floorplan
  - Every time you make a change
    - Recompile your project
    - Check timing analysis and datasheet reports
    - Timing strongly depends on
      - Technology Mapping and Placing
        - How to map a function on the available resources
      - Routing
        - Length of the interconnections
  - If you use ECO (Engineering Change Orders) mode instead of Assignment mode
    - The full compilation is not necessary
    - But changes are not reflected in many other steps of the design flow
    - If you recompile, chances are that your changes will be reverted

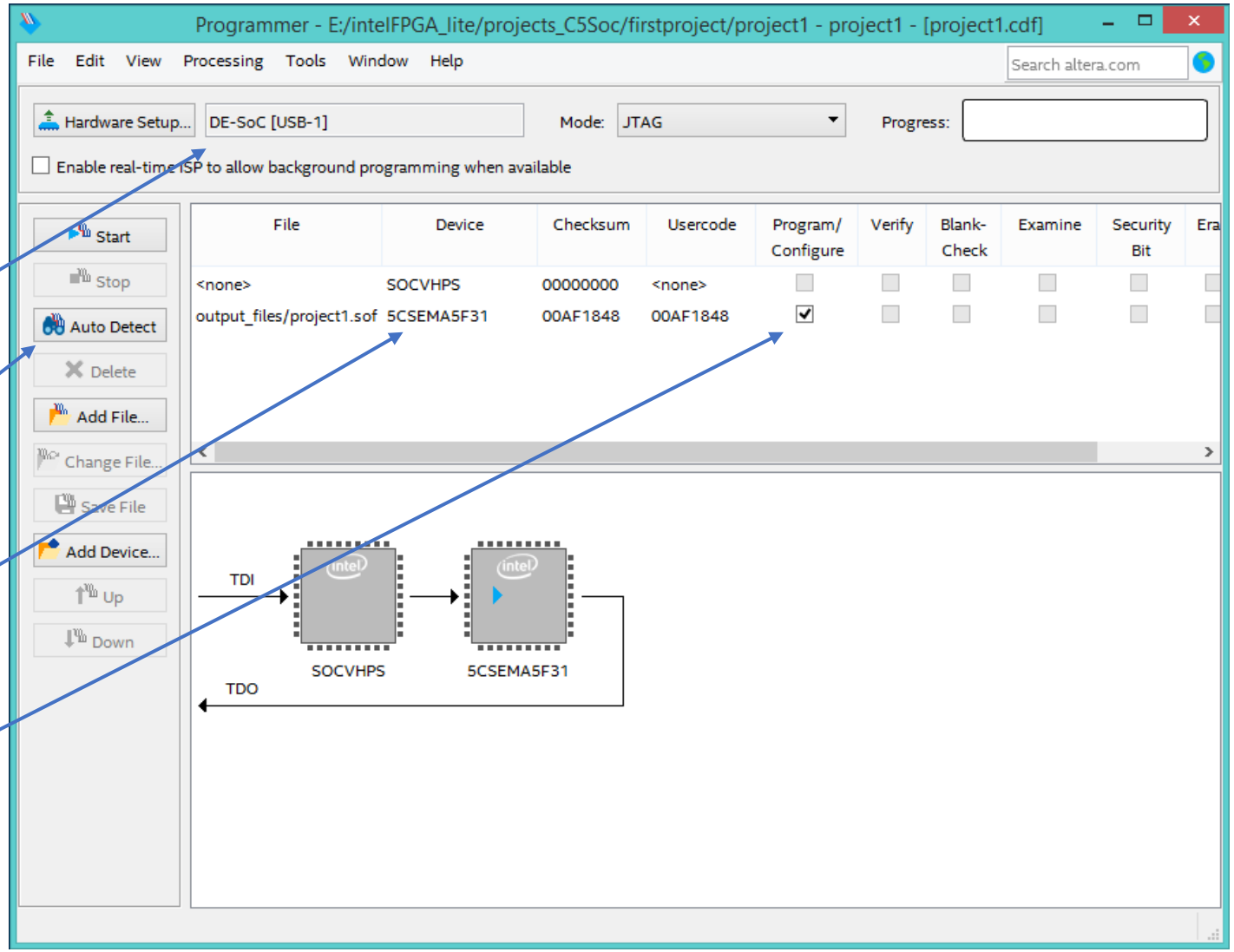
Quartus II

- Run Simulator
  - Under Tools menu
    - Choose Run EDA Simulation Tool
    - Select EDA Gate Level Simulation…
    - Choose the Slow Model and click Run
    - If there are no errors, the simulation starts
    - You now see all estimated delays
    - This is on Quartus 11 with the Cyclone II FPGA
      - No timing model for Cyclone V and 10 LP



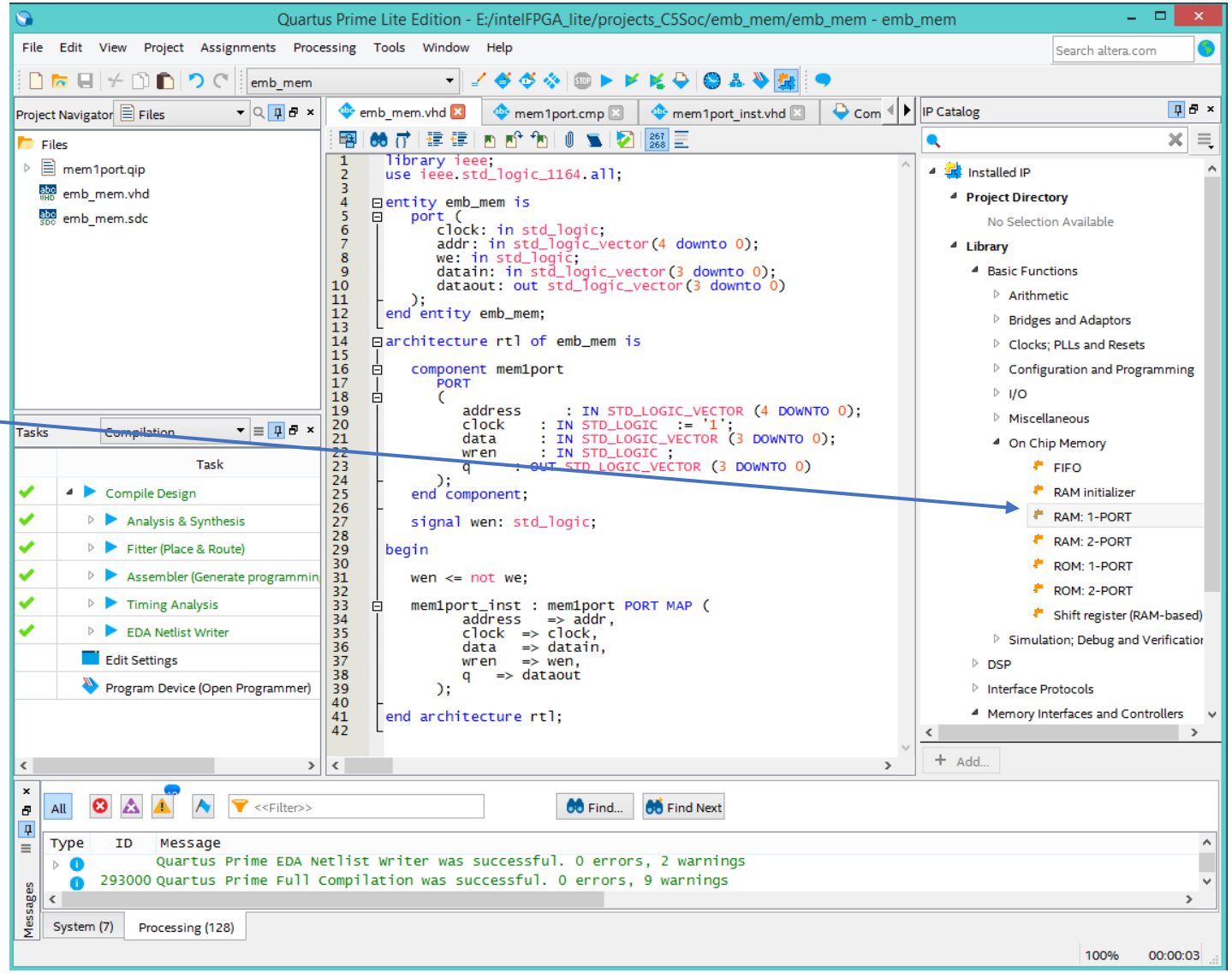
Quartus Prime

- Run Programmer
  - Under Tools menu
    - Choose Programmer
      - Check the Hardware Setup
        - It should be DE-SoC
      - Click Auto Detect and select the 5CSEMA5 device and then overwrite any existing settings
      - Right click on the row for 5CSEMA5 and choose Change File
      - Look for the .sof programmer file under output_files in the project directory
      - Select Program/Configure
      - Click Start
    - ENJOY!



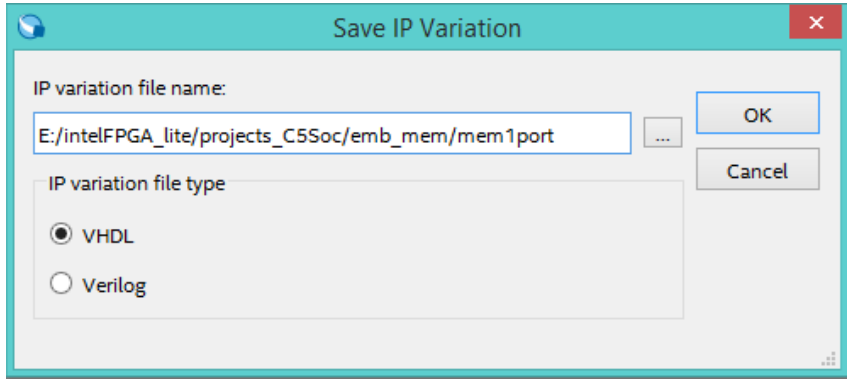
Quartus Prime

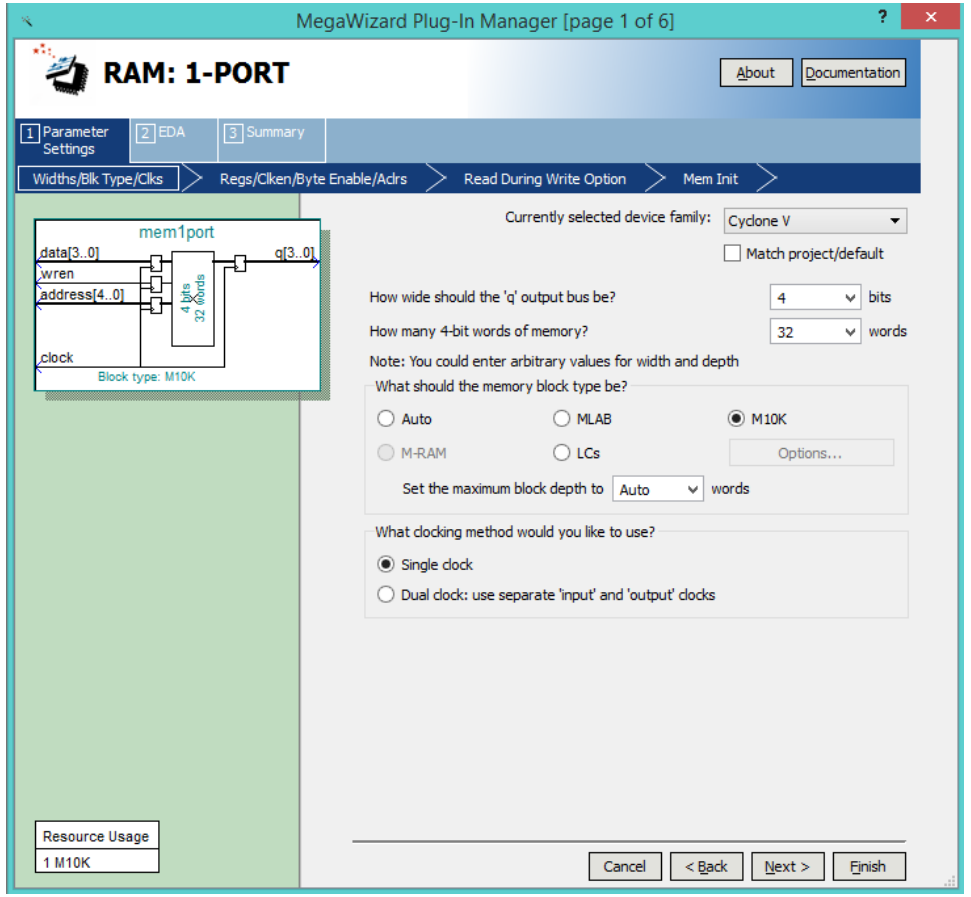
- Adding IP components
  - Look for the component in the IP Catalog
    - RAM: 1-PORT
    - PLL Intel FPGA IP
    - LPM_MULT
  - Right click on it to see Details
  - Double click to add the component



Quartus Prime

- Adding IP components
  - Create an IP Variation
    - You can create many of them with different names
    - Each variation may have different options
  - Set options and parameters in the MegaWizard Plug-In Manager
    - There are often many options
    - Some are not easy to understand
      - Usually leave default values
      - Unless you know what you are doing
    - Select the Instantiation template file
      - Useful when creating the system
  - Click Finish
  - Add Quartus Prime IP file to the project





Quartus Prime

- Adding IP components
  - Create a VHDL file with your top-level entity
    - The top-level is not the IP component
  - Declare the component at the beginning of the architecture
    - The file with extension .cmp already contains the component declaration
      - Just copy from it
  - Instantiate the component in your architecture
    - If you selected the Instantiation template file, then open the inst.vhd file
      - Copy the instantiation template
      - Assign the corresponding signals to the ports
    - You can even instantiate more than one component
  - Write a testbench and simulate
  - Program the FPGA and try it out on hardware