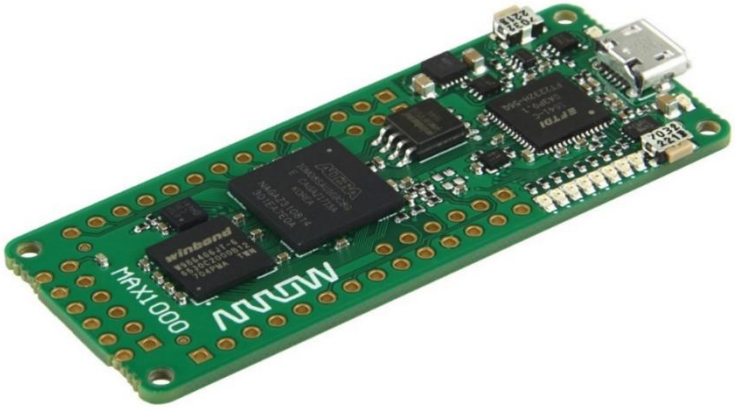
MAX1000

Signal Probe Debugging Lab



Software and hardware requirements to complete all exercises

Software Requirements: Quartus® Prime Lite or Standard Edition version 18.0 or 18.1

Hardware Requirements: ARROW MAX1000 Board

Revision 1.0

1. Introduction

The Signal Probe feature allows you to route a user specified internal node to a top-level I/O without affecting the existing fitting in a design. Using the Signal Probe allows you to investigate internal device signals without performing a full compilation.

Lab Notes: Many of the names that the lab asks you to choose for files, components, and other objects in this exercise must be spelled exactly as directed. This nomenclature is necessary because the pre-written software application includes variables that use the names of the hardware peripherals. Naming the components differently can cause errors.

2. Getting Started

The first objective is to ensure that you have all the necessary hardware items and software installed so that the lab can be completed successfully. Below is a list of items required to complete this lab:

- MAX1000 Board **(**10M08SAU169C8G)
- USB Cable
- Quartus Prime 18.0 Lite was used for this lab. Previous/newer versions should work (If no Quartus Prime is installed, refer to MAX1000 User Guide for instructions)
- Installed Arrow USB Drivers (If not, refer to MAX1000 User Guide for instructions)
- Personal computer or laptop running 64-bit Linux / Windows 7 or later with at least an Intel i3 core (or equivalent), 4GB RAM and 12 GB of free hard disk space
- A desire to learn!

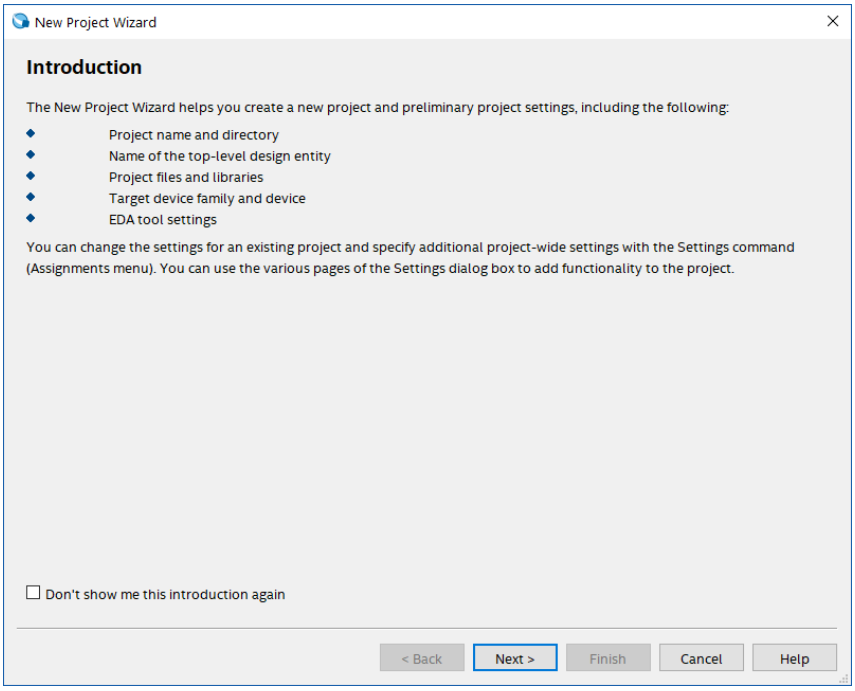
3. Project with MAX1000

3.1 New Quartus Prime project

3.1.1 New project creation

3.1.1.1 If not already open, from the Start menu or the Desktop, open the Quartus Prime 18.0 Lite software.

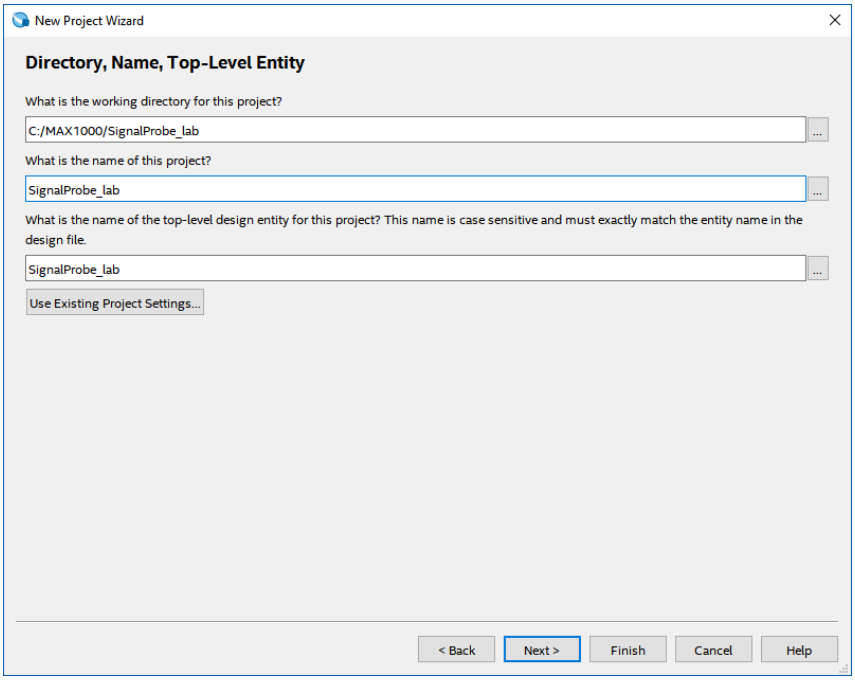
3.1.1.2 Create a new project using the New Project Wizard: **File → New Project Wizard**.



3.1.1.3 Click **Next**.

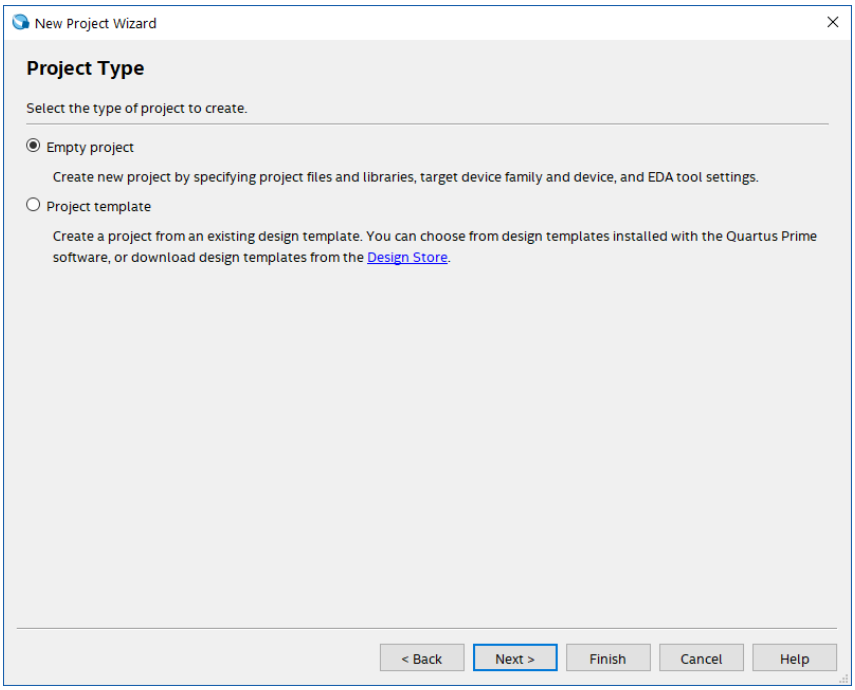
3.1.1.4 Configure the New Project Wizard directory, name and top-level entity information:

- Enter a directory in which you will store your Quartus project files for this design, for example, **C:/MAX1000/SignalProbe_lab**
- Specify the name of the project: **SignalProbe_lab**
- Specify the name of the top-level entity: **SignalProbe_lab**

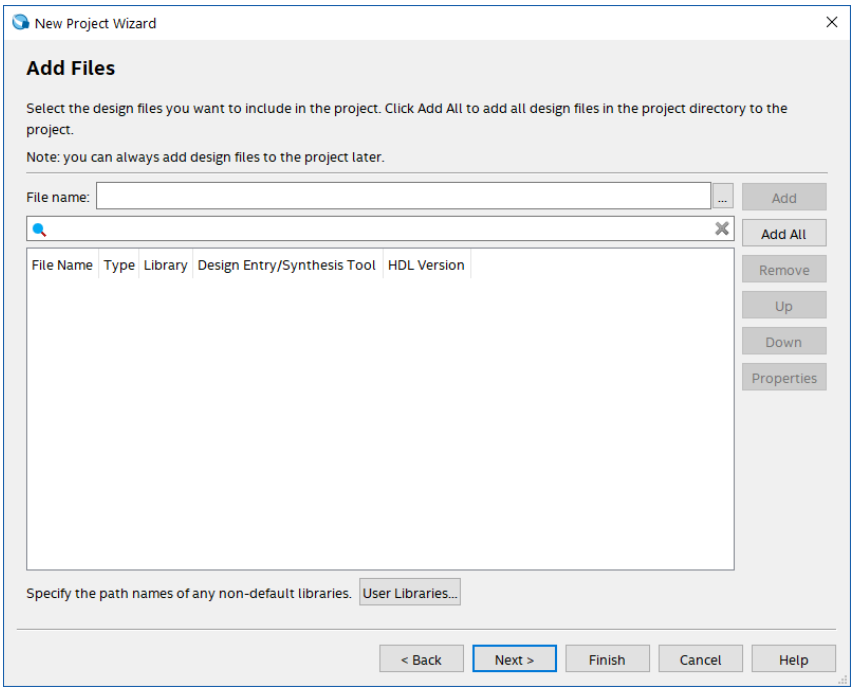


3.1.1.5 Click **Next**.

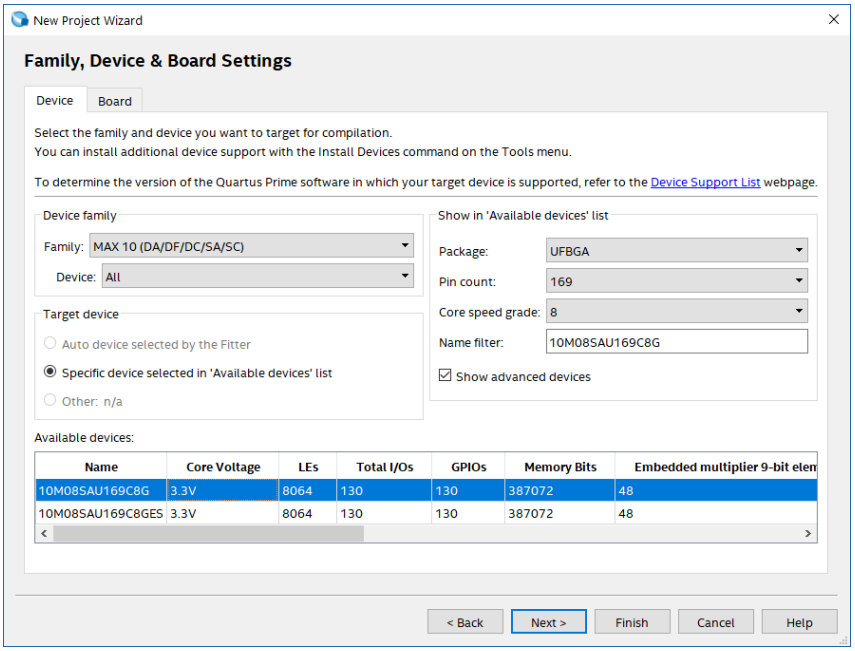
3.1.1.6 On the Project Type page, select **"Empty project"** and click **Next**.



3.1.1.7 On the Add Files page, click **Next**.



3.1.1.8 Specify Family and Device Settings. Use pull-down menus to select MAX10 family or enter the part number in the Name Filter text box. The part number is **10M08SAU169C8G**.



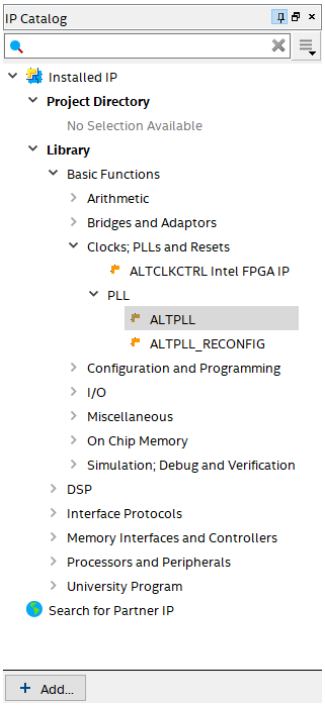
3.1.1.9 Click **Finish**.

3.2 Design entry

3.2.1 Add PLL to the Quartus Project

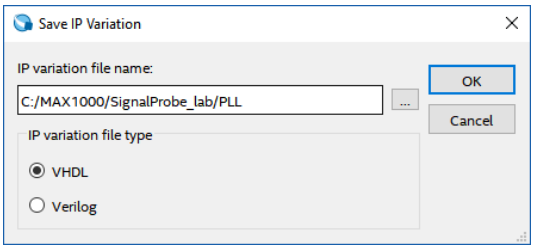
3.2.1.1 From the IP Catalog panel on the left side, expand the menus for the **Basic Functions → Clocks; PLLs and Resets → PLL** and double click on **ALTPLL**.

If the IP catalog is not visible, then right click on the toolbar and select IP catalog.



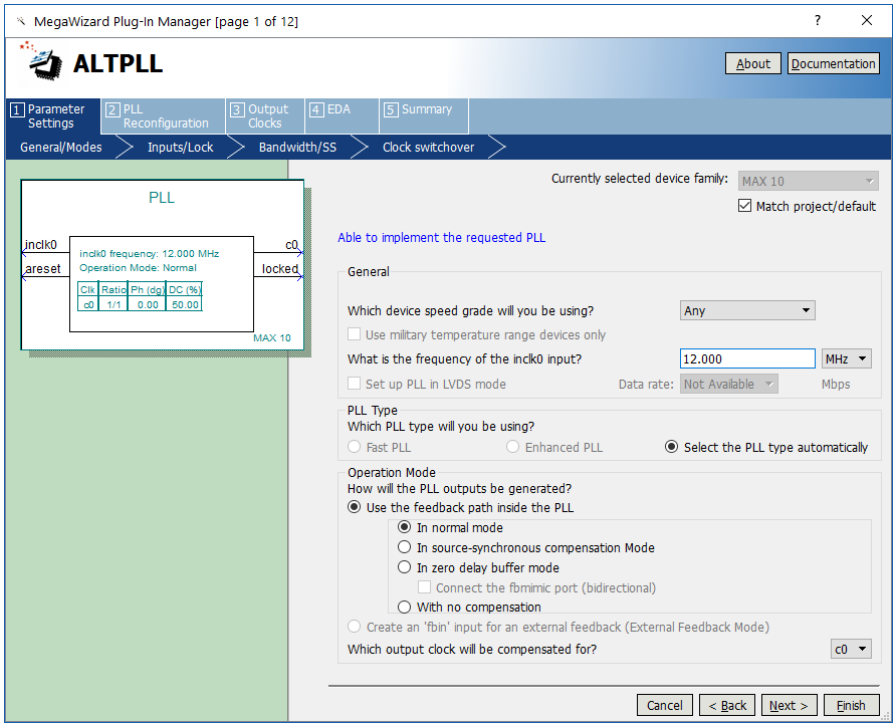
3.2.1.2 On the Save IP Variation window, enter the following information.

- IP variation file name: **<project_directory>/PLL**
- IP variation file type: **VHDL**



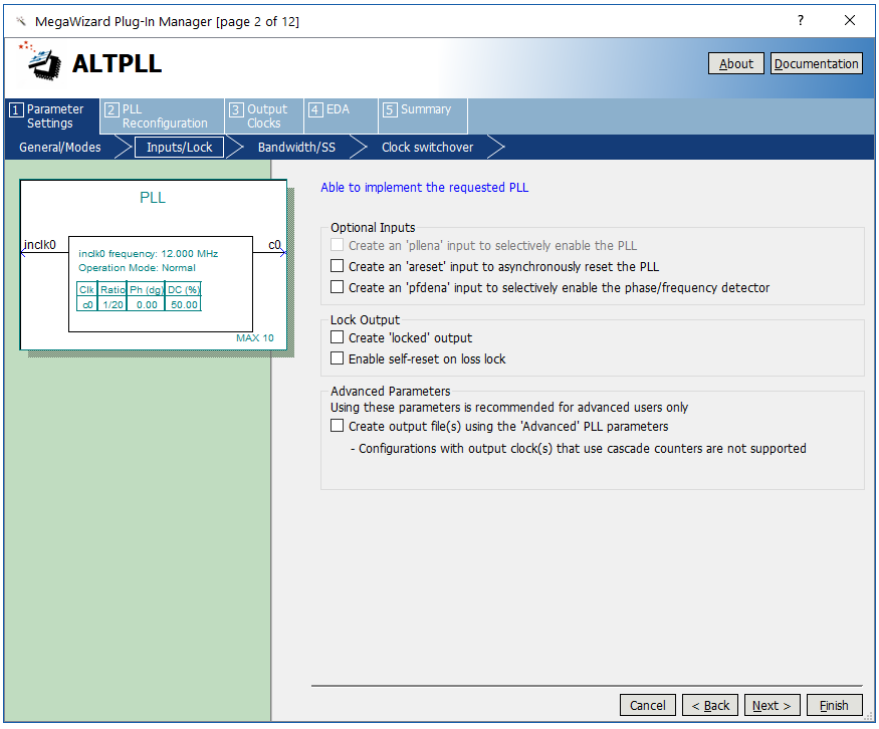
3.2.1.3 Click **OK**.

3.2.1.4 Under General/Modes tab (page 1 of 12) of PLL MegaWizard change the frequency of clock input to **12 MHz.** This source is provided by the internal oscillator in the MAX10 FPGA.



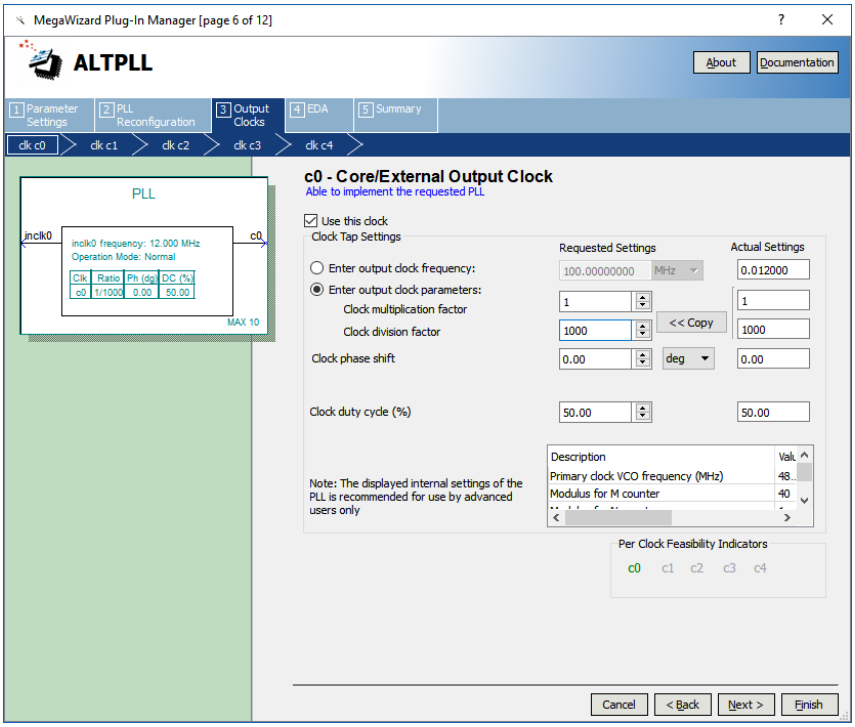
3.2.1.5 Click **Next**.

3.2.1.6 Under Input/Lock tab (page 2 of 12) uncheck ‘areset’ input and locked output option.



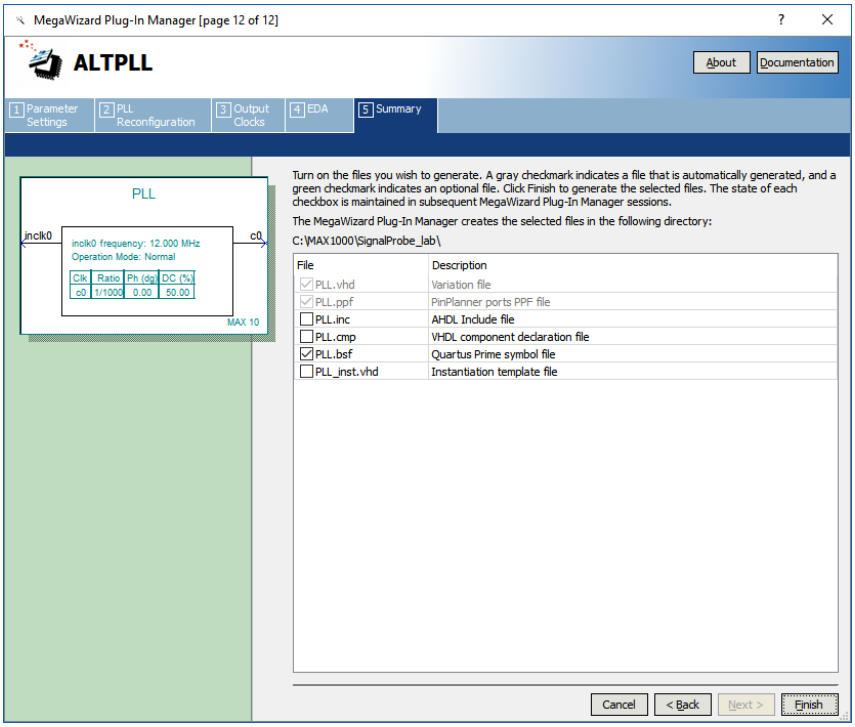
3.2.1.7 Click **Next** until you reach the **Output Clocks** tab (page 6 of 12).

3.2.1.8 Under the clk c0 tab (page 6 of 12) select “Enter output clock parameters” and set Clock division factor to **1000**. Leave the rest as default.

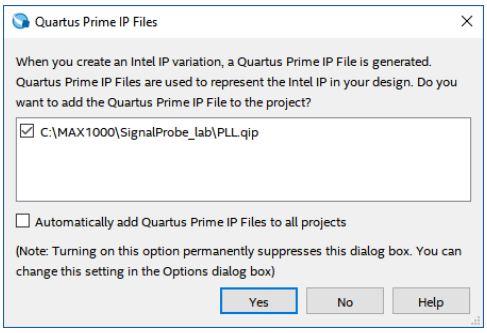


3.2.1.9 Click **Finish.** This will take you to the **Summary** tab (page 12 of 12).

3.2.1.10 Select **PLL.bsf** checkbox and click **Finish**.

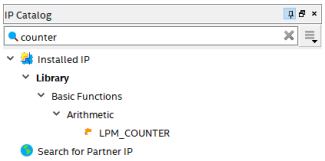


3.2.1.11 In the pop-up Quartus Prime IP Files accept all defaults and click **Yes**.

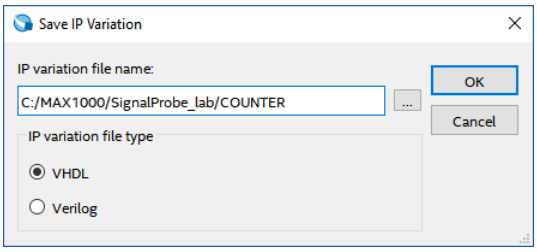


3.2.2 Add counter to the Quartus Project

3.2.2.1 In the search bar of the IP Catalog, type “counter”, and double click on **LPM_COUNTER**.

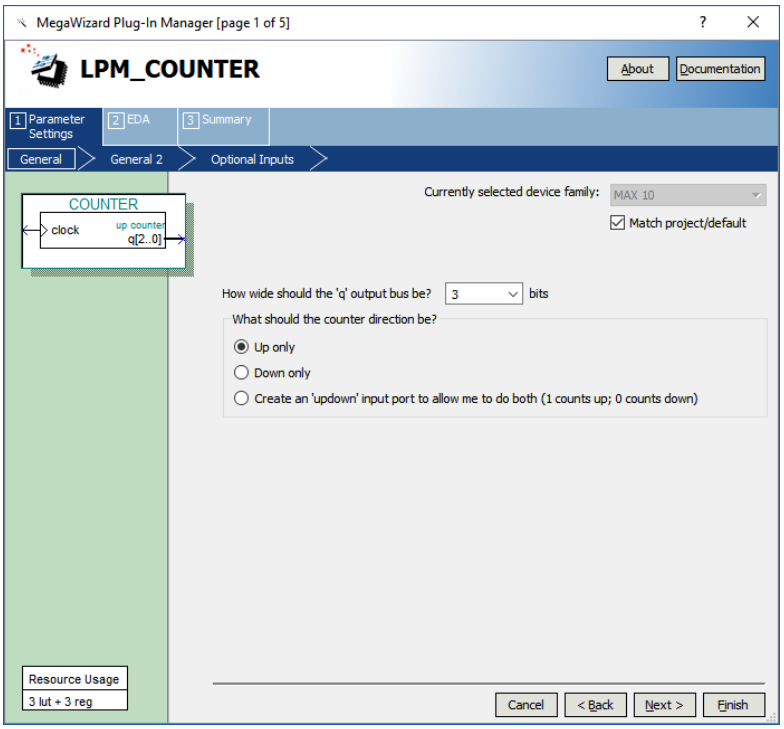


3.2.2.2 In the Save IP Variation window enter **COUNTER** for the IP variation file name and select **VHDL.**



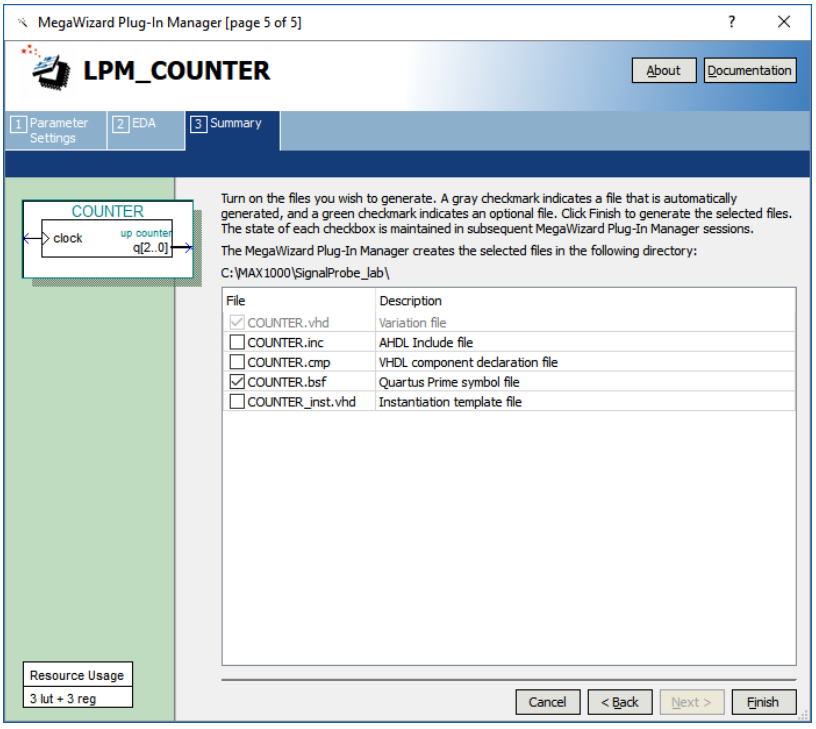
3.2.2.3 Press **OK**.

3.2.2.5 Change the ‘q’ output bus to **3 bits**.

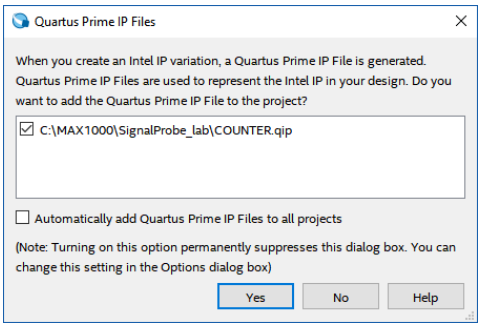


3.2.2.6 Click **Finish.** This will take you to the **Summary** tab (page 5 of 5).

3.2.2.7 Select **COUNTER.bsf** checkbox and click **Finish**.

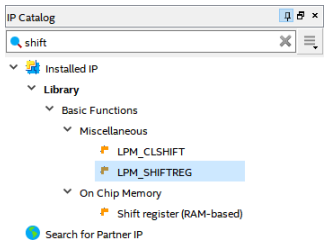


3.2.2.8 In the pop-up Quartus Prime IP Files accept all defaults and click **Yes**.

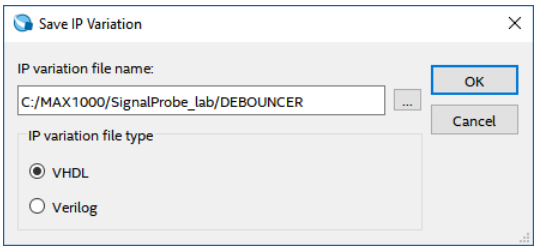


3.2.3 Add shift register to the Quartus Project

3.2.3.1 In the search bar of the IP Catalog, type “shift”, and double click on **LPM_SHIFTREG**.

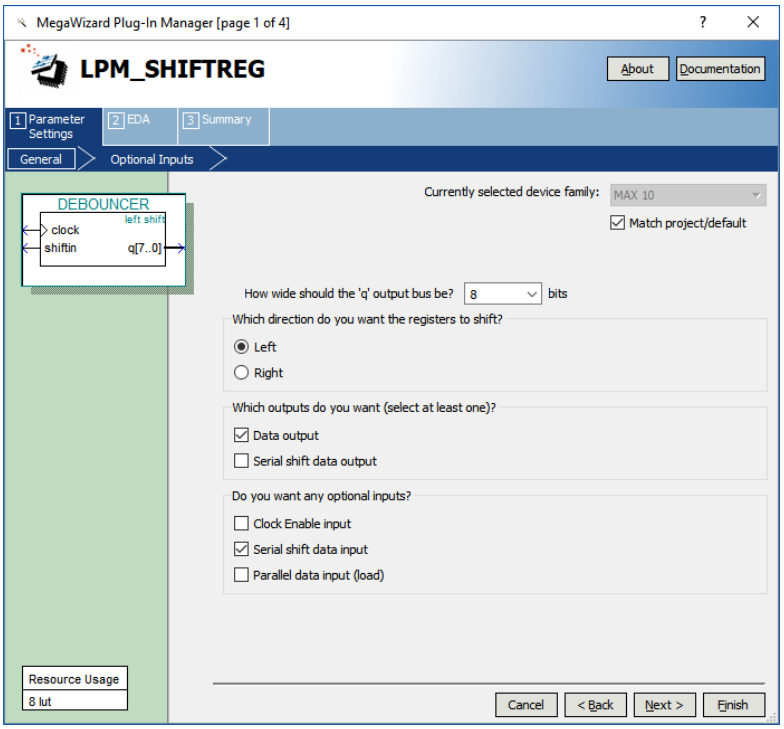


3.2.3.2 In the Save IP Variation window enter **DEBOUNCER** for the IP variation file name and select **VHDL.**

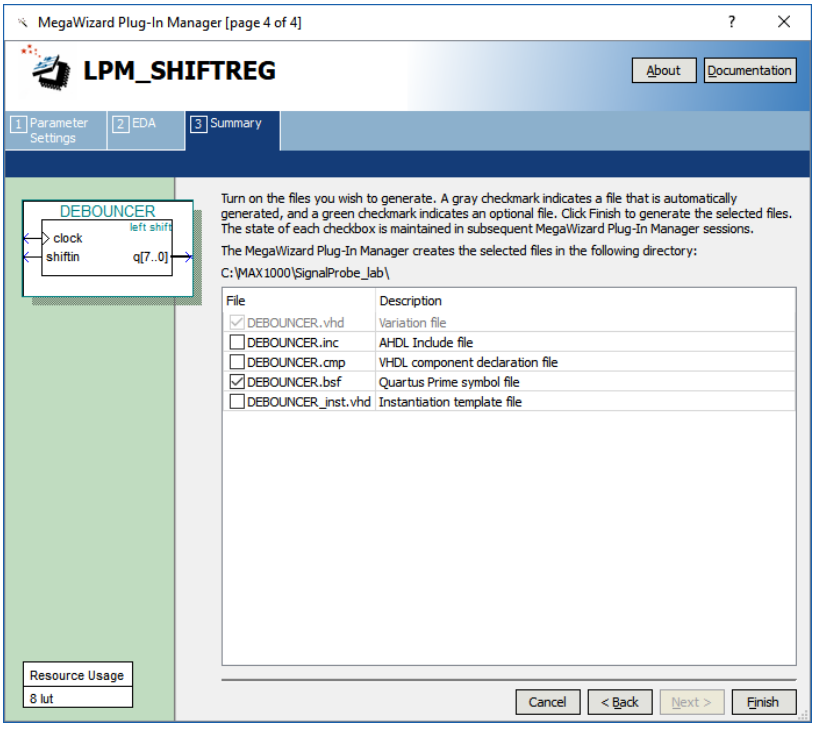


3.2.3.3 Press **OK**.

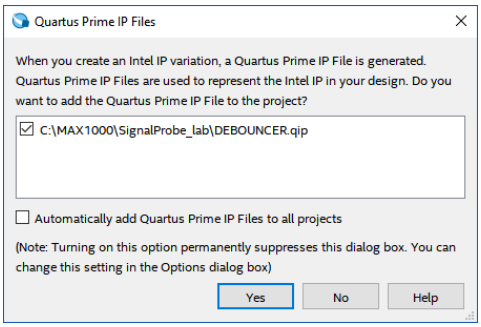
3.2.3.5 Accept all defaults and press **Finish**.



3.2.3.6 On the **Summary** tab (page 4 of 4) select **DEBOUNCER.bsf** and click **Finish**.

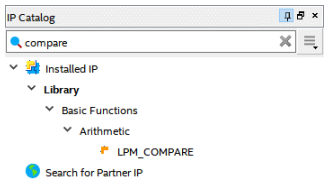


3.2.3.7 In the pop-up Quartus Prime IP Files accept all defaults and click **Yes**.

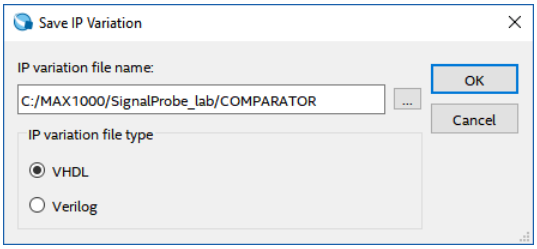


3.2.4 Add comparator to the Quartus Project

3.2.4.1 In the search bar of the IP Catalog, type “compare”, and double click on **LPM_COMPARE**.

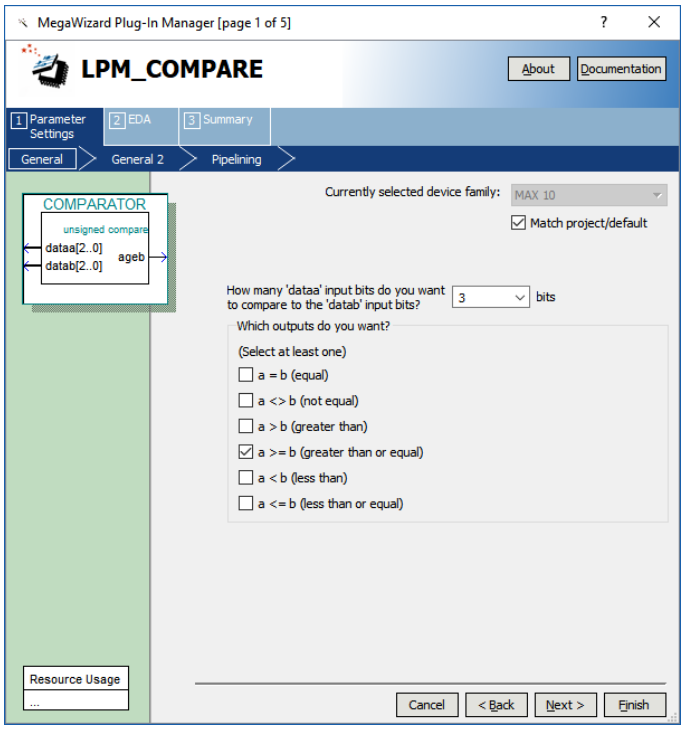


3.2.4.2 In the Save IP Variation window enter **COMPARATOR** for the IP variation file name and select **VHDL.**



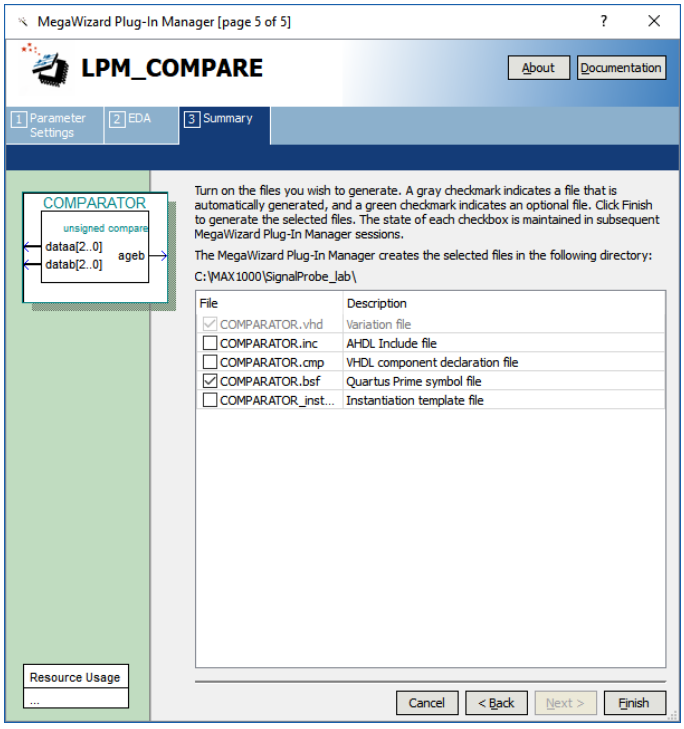
3.2.4.3 Press **OK**.

3.2.4.5 On the General tab set **3 bits** for the inputs and select **a >= b (greater than or equal)**.

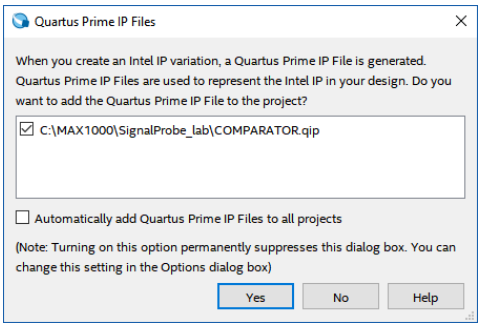


3.2.4.6 Press **Finish**.

3.2.4.7 On the **Summary** tab (page 5 of 5) select **COMPARATOR.bsf** and click **Finish**.

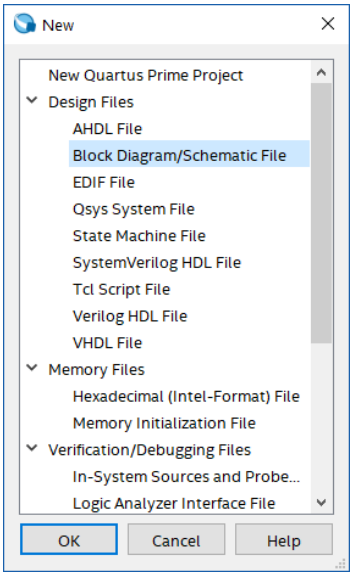


3.2.4.8 In the pop-up Quartus Prime IP Files accept all defaults and click **Yes**.

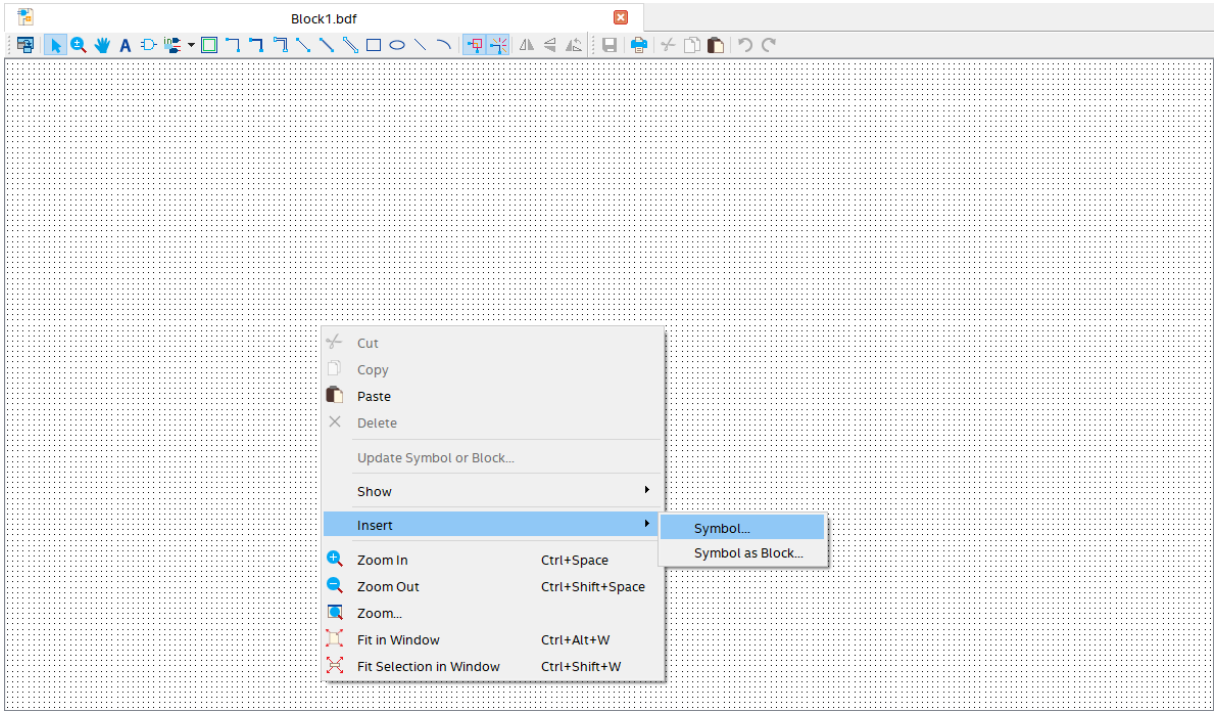


3.2.5 Creating schematic

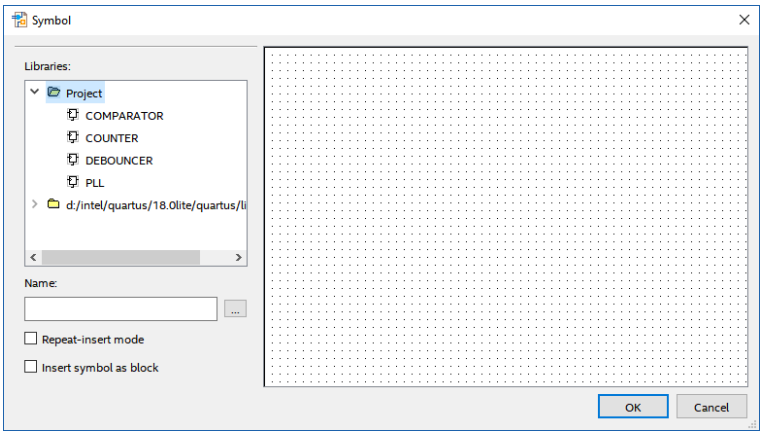
3.2.5.1 Choose **File → New → Block Diagram/Schematic File** and click **OK**. A new schematic will be created, where the components can be added.



3.2.5.2 Right click in the schematic page, and select **Insert → Symbol…**



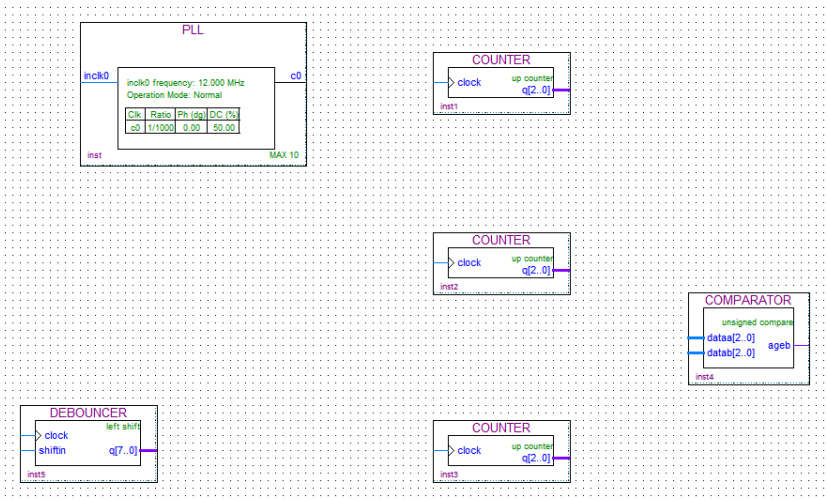
3.2.5.3 In the Symbol window, expand the “Project” folder and the four components that were created can now be seen.



3.2.5.4 Select **PLL** and click **OK**.

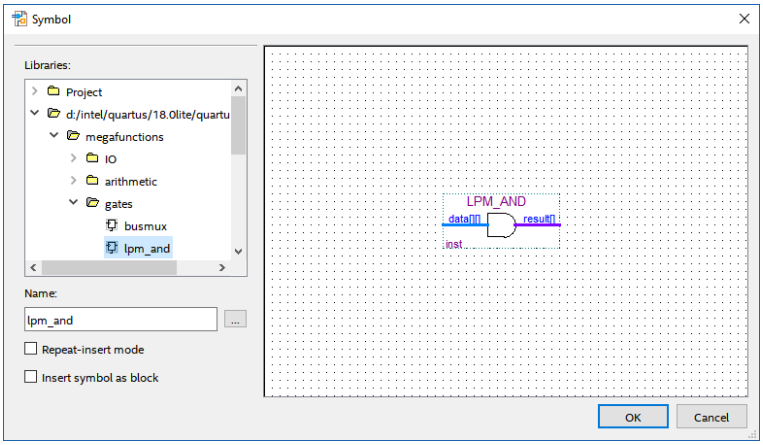
3.2.5.5 The PLL component can be added by left clicking on the schematic page.

3.2.5.6 Just like in the steps from 4.2.3.2 to 4.2.3.5, do the same for 3 counters, comparator and debouncer to add them to the schematic.

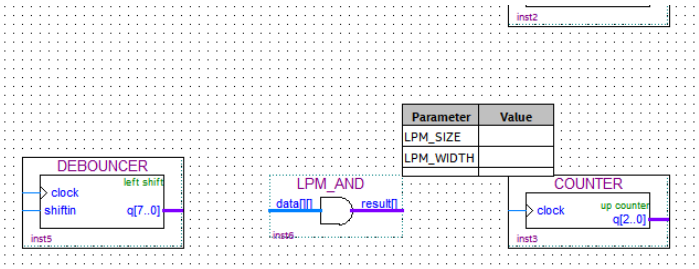


3.2.5.7 Open again the Symbol window and expand the basic libraries.

3.2.5.8 Browse **megafunction → gates** and select **lpm_and**.



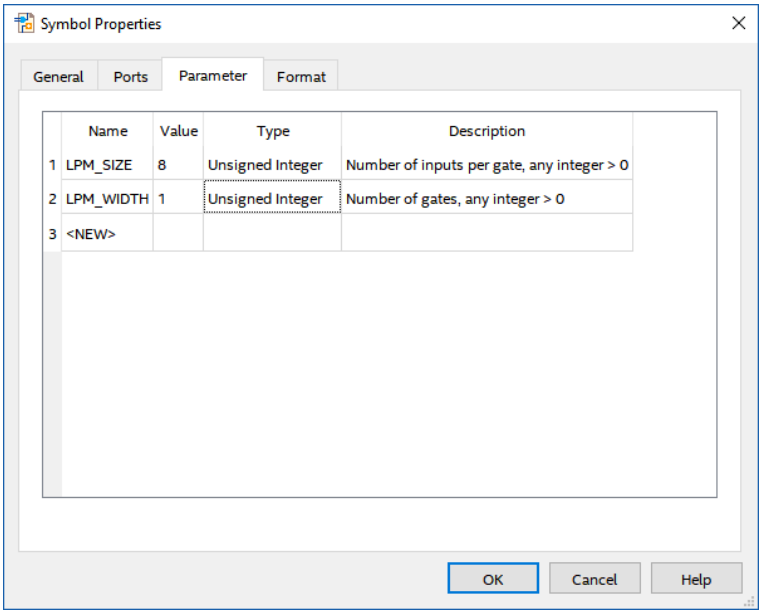
3.2.5.9 Add it to the schematic.



3.2.5.10 Right click on the LPM_AND and select **Properties**.

3.2.5.11 Choose Parameter tab and enter the following information. You can select the parameters from the drop-down menu if you double click on the cell.

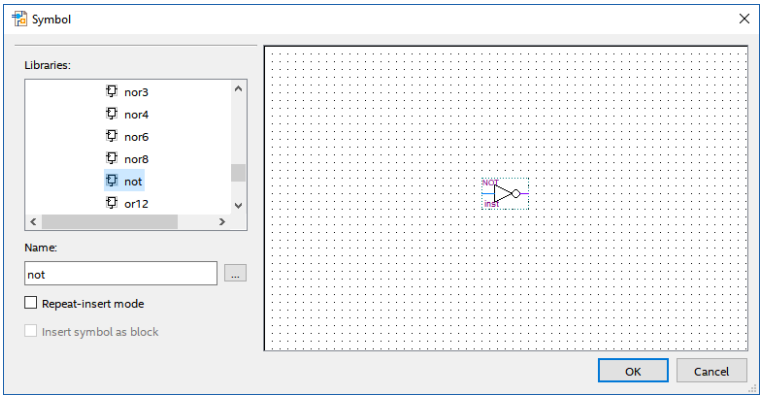
Name	Value	Type
LPM_SIZE	8	Unsigned Integer
LPM_WIDTH	1	Unsigned Integer



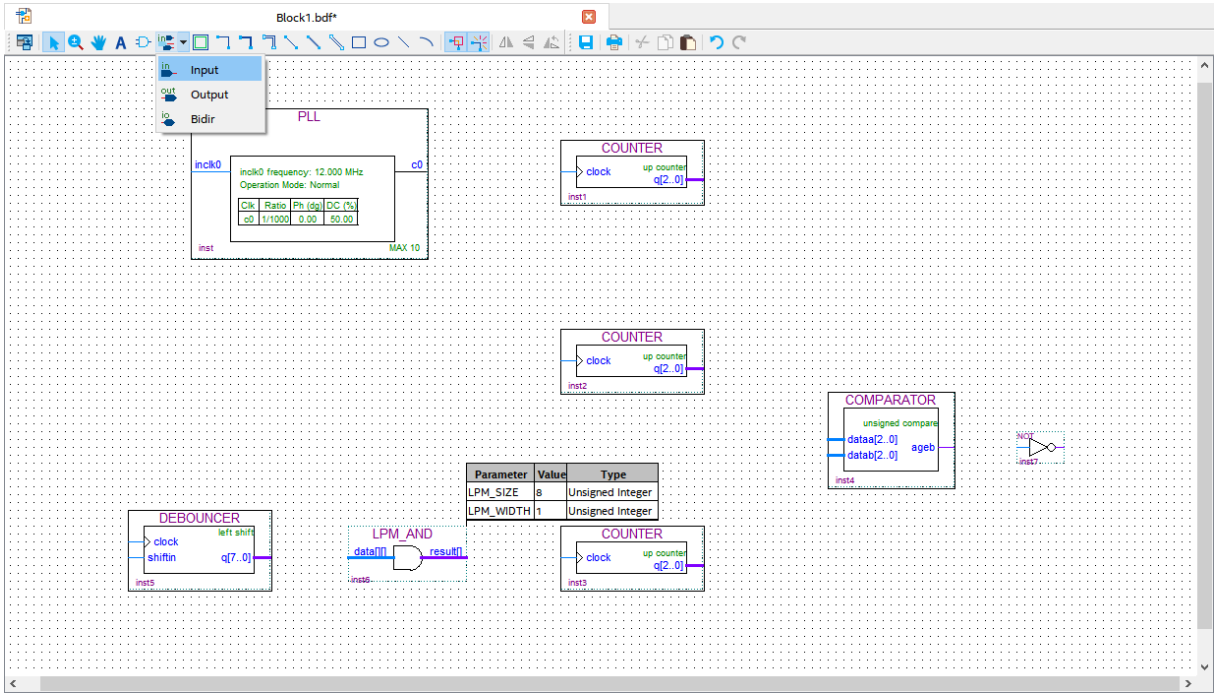
3.2.5.12 Click **OK**.

3.2.5.13 Open again the Symbol window and browse **primitives → logic.**

3.2.5.14 Select **not** and add it to the schematic.



3.2.5.15 Click on the **Pin Tool** on the top button bar and select **Input**.



3.2.5.16 Add one input pin for **shiftin** of the DEBOUNCER, and an additional one input pin for **inclk0** of the PLL.

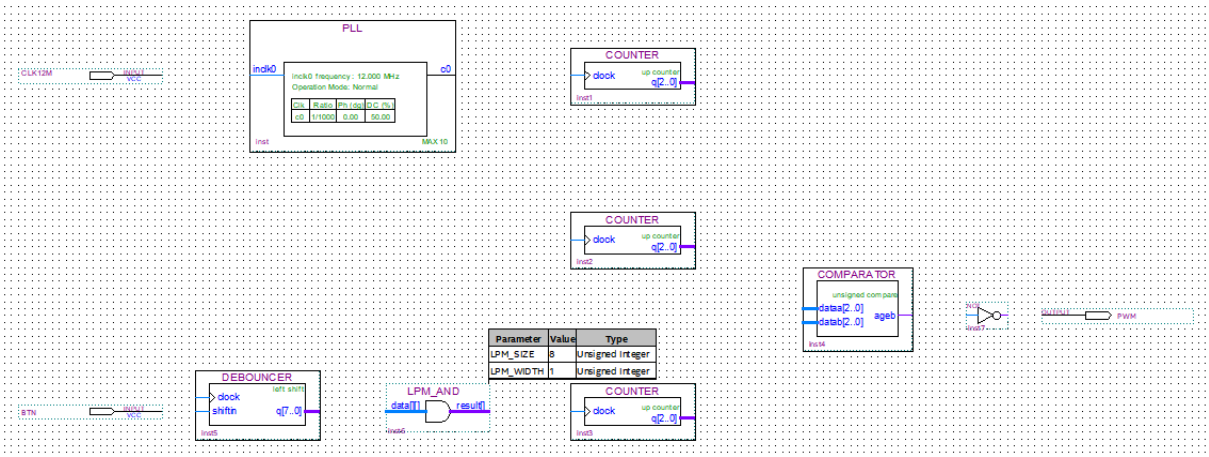
3.2.5.17 Click on the **Pin Tool** as before and select **Output**.

3.2.5.18 Add one output pin for the not gate.

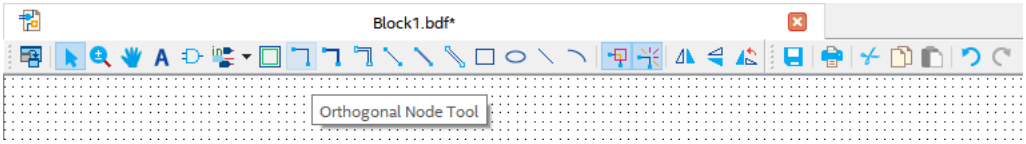
3.2.5.19 Rename the pins by double clicking its current name.

- pin_name1 to **CLK12M**. This is going to be the clock signal coming into the FPGA.
- pin_name2 to **BTN**. This is going to be the signal of the button coming into the FPGA.
- pin_name3 to **PWM**. This is going to be the output signal.

3.2.5.20 At this point all components are added to the schematic and should look as follows:

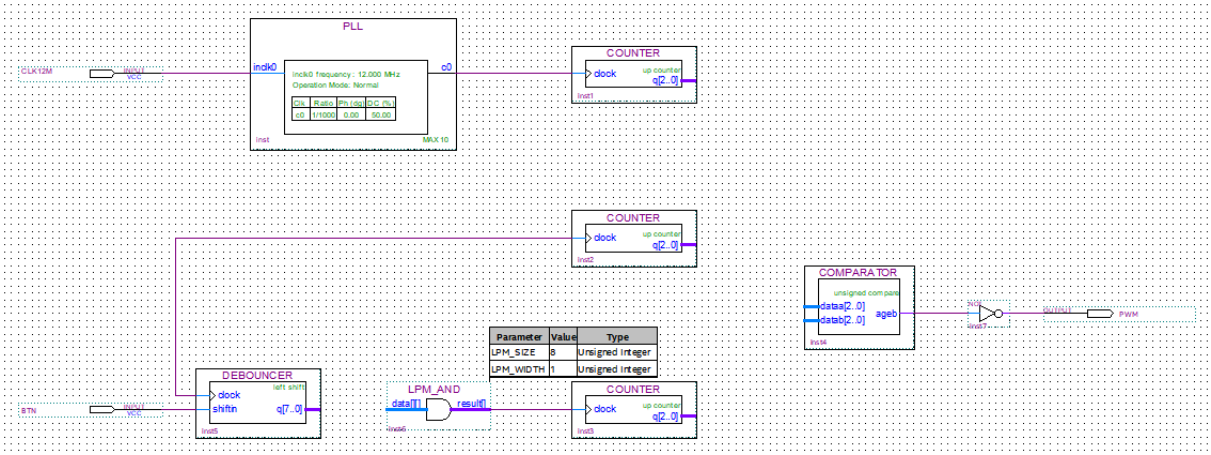


3.2.5.21 Select **Node Tool** on the top button toolbar.

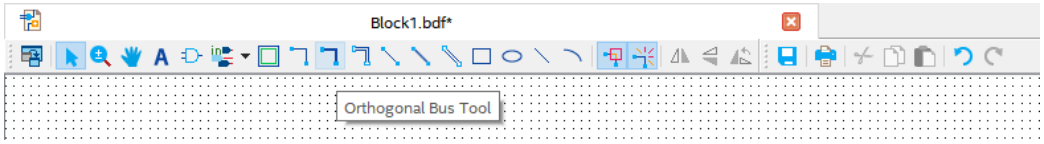


3.2.5.22 Connect the wires:

CLK12M	→	PLL | **inclk0**
PLL | **c0**	→	COUNTER(inst1) | **clock**
BTN	→	DEBOUNCER | **shiftin**
DEBOUNCER | **clock**	→	COUNTER(inst2) | **clock**
LPM_AND | **result[]**	→	COUNTER(inst3) | **clock**
COMPARATOR | **ageb**	→	NOT | **input**
NOT | **output**	→	**PWM**



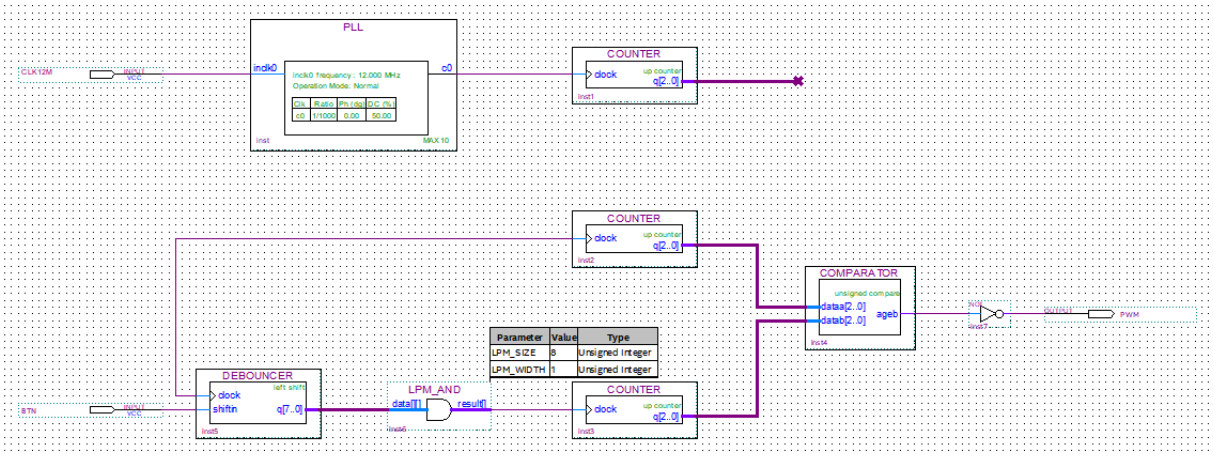
3.2.5.23 Select the **Bus Tool** on the top button toolbar.



3.2.5.24 Create the connections:

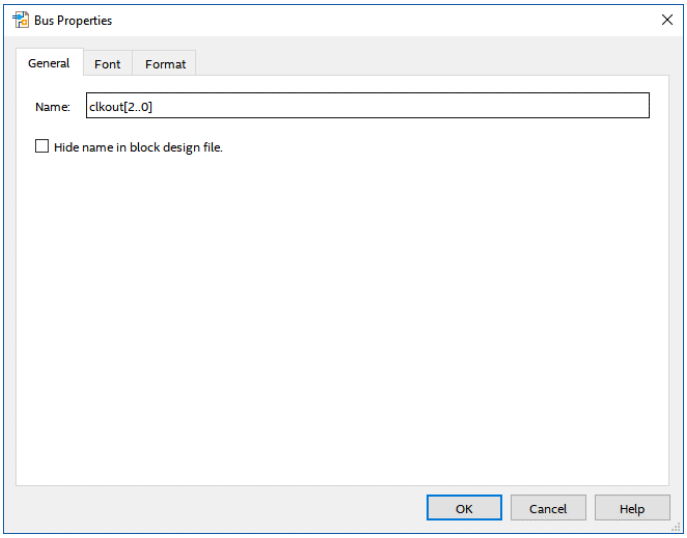
COUNTER(inst2) | **q[2..0]**	→	COMPARATOR | **dataa[2..0]**
COUNTER(inst3) | **q[2..0]**	→	COMPARATOR | **datab[2..0]**
DEBOUNCER | **q[7..0]**	→	LPM_AND | **data[][]**

3.2.5.25 Using the bus tool create a connection coming out of the COUNTER (inst1).



3.2.5.26 Right click on the bus line of the COUNTER(inst1)'s output and select **Properties**.

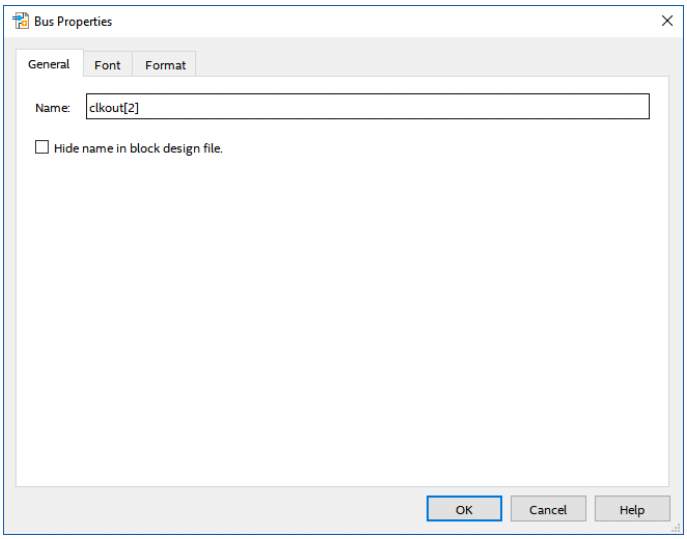
3.2.5.27 Set the name of the bus to **clkout[2..0]**.



3.2.5.28 Click **OK**.

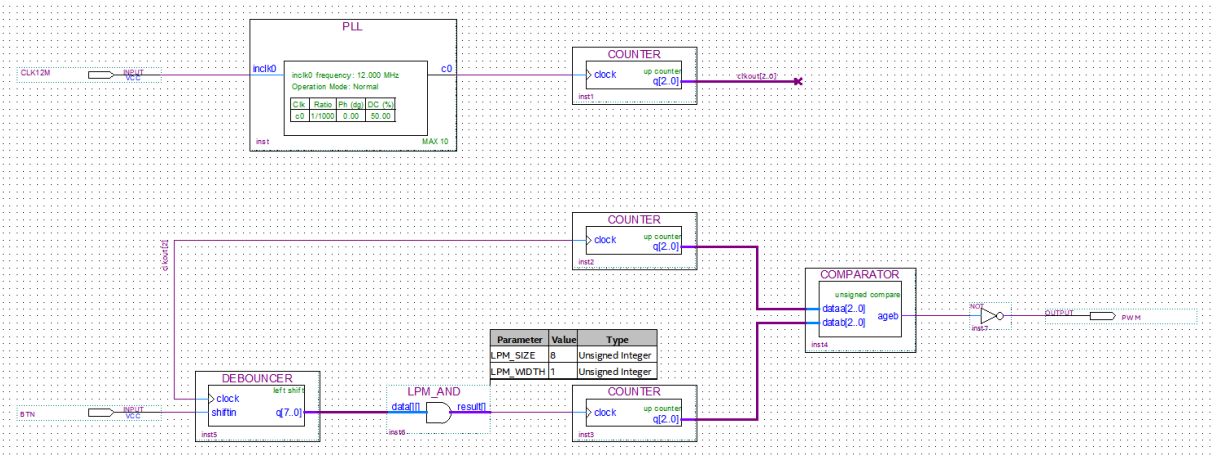
3.2.5.29 Right click on the wire of the DEBOUNCER clock and select **Properties**.

3.2.5.30 Set the name of the bus to **clkout[2]**.



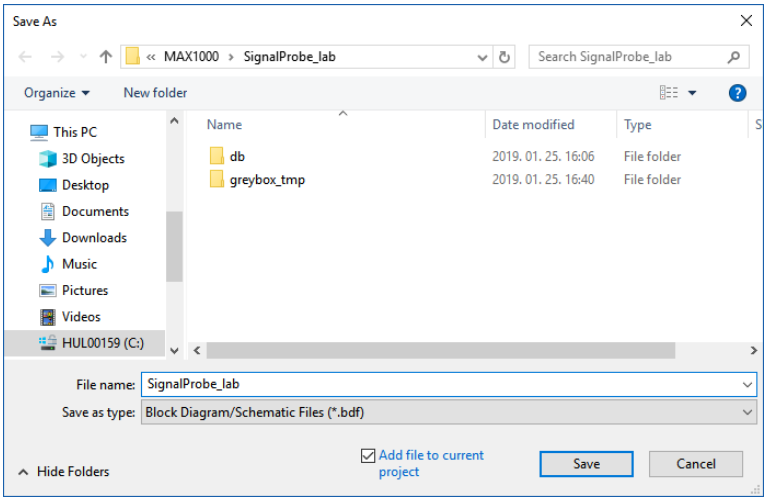
3.2.5.31 Click **OK**.

3.2.5.32 Verify that your schematic is the same as below:



3.2.5.33 Save your design by clicking on button or **File → Save** and enter the following information.

- File name: **SignalProbe_lab**
- Save as type: **Block Diagram/Schematic Files (*.bdf)**
- Make sure that **"Add file to current project"** option is checked.



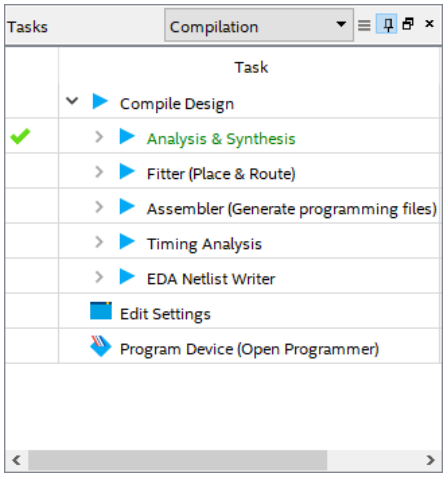
3.2.5.34 Click **Save**.

3.3 Compile design

3.3.1 Analysis and Synthesis

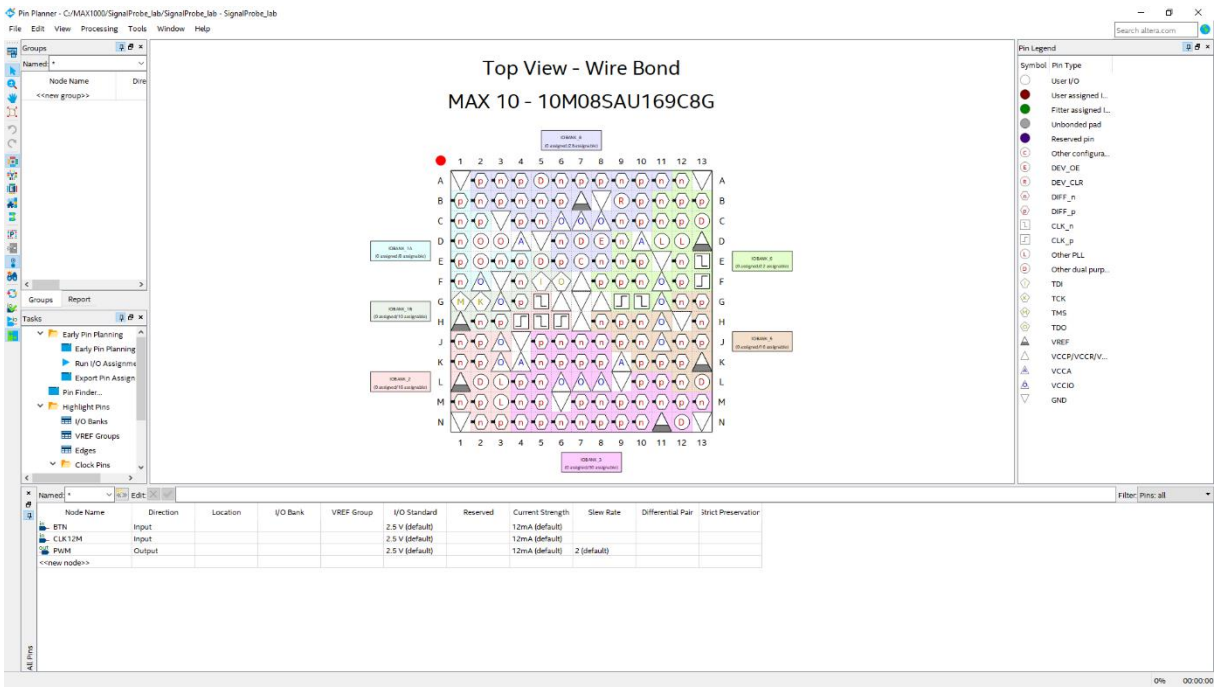
3.3.1.1 Run Analysis and Synthesis by clicking on button on the toolbars, or **Processing → Start → Analysis and Synthesis**.

There should be no errors. If there are errors, they should be fixed before continuing. If there are no errors the compilation task windows should look like this:



3.3.2 Pin Assignments

3.3.2.1 Open **Pin Planner** by clicking on button on the toolbars, or **Assignments → Pin Planner**.



3.3.2.2 In the bottom table, type **PIN_H6** in Location column of the CLK12M.

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
BTN	Input				2.5 V (default)		12mA (default)			
CLK12M	Input	PIN_H6	2	B2_N0	2.5 V (default)		12mA (default)			
PWM	Output				2.5 V (default)		12mA (default)	2 (default)		
<<new node>>										

3.3.2.3 Repeat the previous step with the following assignments:

Node Name	Pin Location
BTN	PIN_E6
PWM	PIN_A8

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
BTN	Input	PIN_E6	8	B8_N0	2.5 V (default)		12mA (default)			
CLK12M	Input	PIN_H6	2	B2_N0	2.5 V (default)		12mA (default)			
PWM	Output	PIN_A8	8	B8_N0	2.5 V (default)		12mA (default)	2 (default)		
<<new node>>										

3.3.2.4 Double click in the I/O Standard column of BTN to open a drop-down list and change the 2.5V (Default) to **3.3 V Schmitt Trigger**.

3.3.2.5 Change the I/O Standard of CLK12M and PWM to **3.3-V LVTTL**.

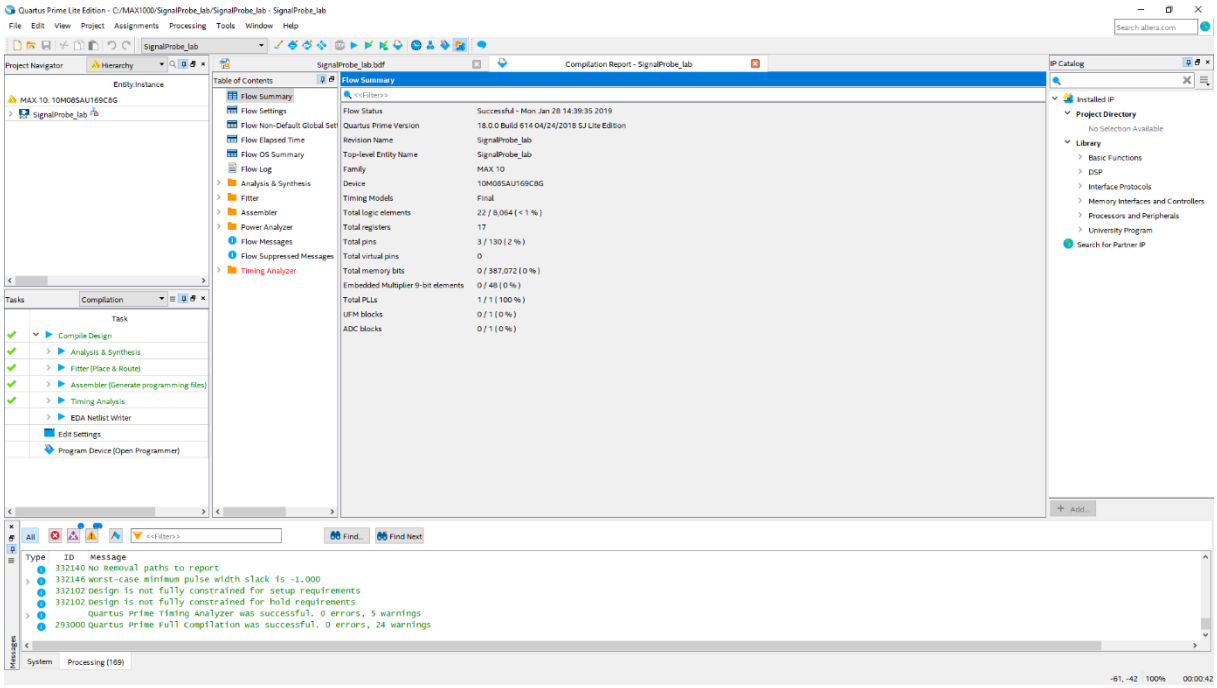
Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
BTN	Input	PIN_E6	8	B8_N0	3.3 V Schmitt Trigger		8mA (default)			
CLK12M	Input	PIN_H6	2	B2_N0	3.3-V LVTTL		8mA (default)			
PWM	Output	PIN_A8	8	B8_N0	3.3-V LVTTL		8mA (default)	2 (default)		
<<new node>>										

3.3.2.6 Close the Pin Planner, the settings are automatically saved.

3.3.3 Compiling the Design

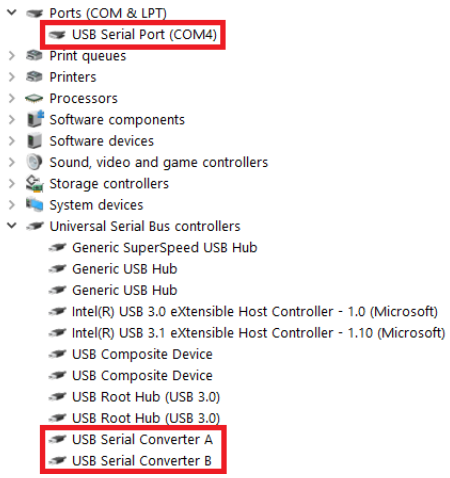
3.3.3.1 Start Compilation by clicking on  button on the toolbars, or **Processing → Start Compilation**.

There should be no errors. If there are errors, they should be fixed before re-compilating. The 100% in the lower right corner or a green checkmark next to the Compile Design in the Compilation task window indicates that the compilation was successful.

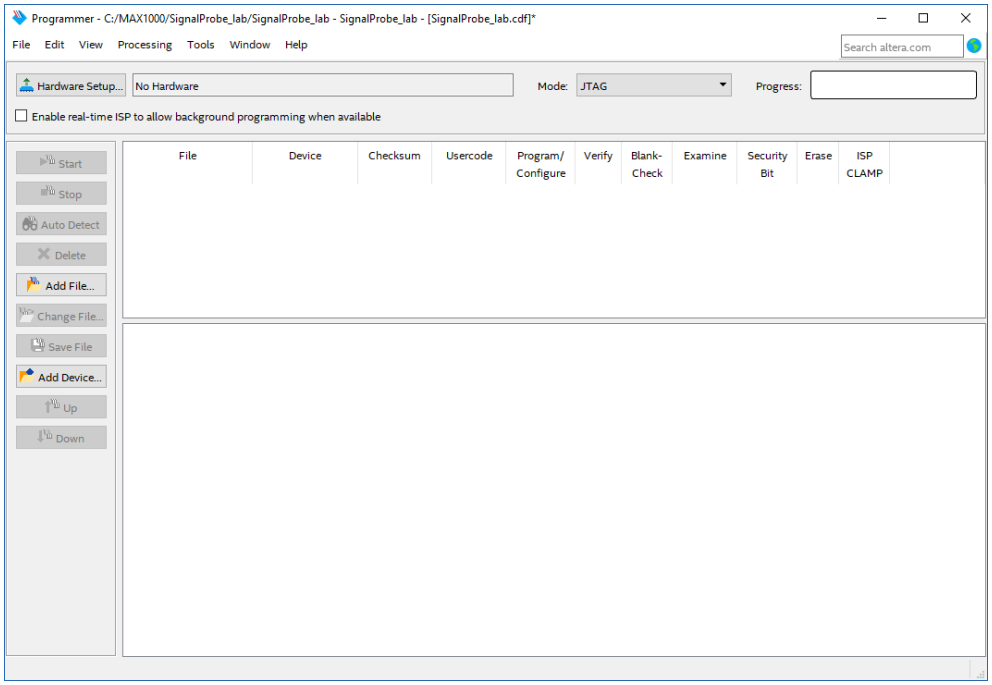


3.3.4 Configuration

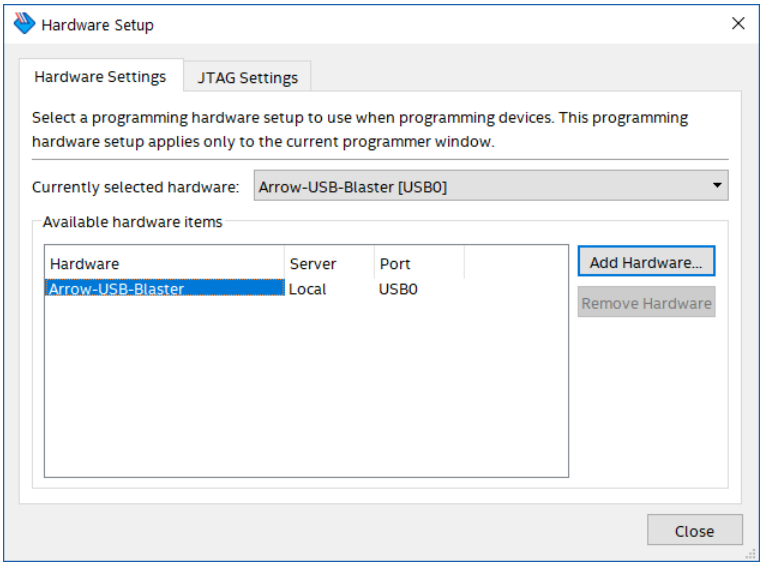
3.3.4.1 Connect your MAX1000 board to your PC using an USB cable. Since the Arrow USB Blaster should be already installed, the Window's Device Manager should display the following entries are highlighted in red (port number may differ depending on your PC):



3.3.4.2 Open the Quartus Prime Programmer from **Tools → Programmer** or double click on Program Device (Open Programmer) from the Task window.

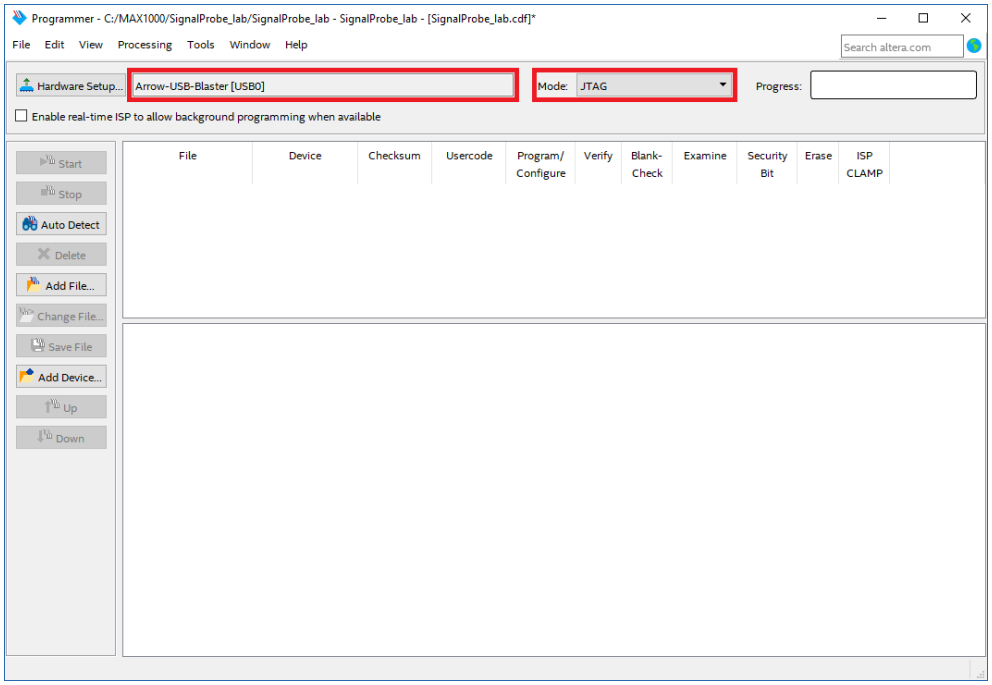


3.3.4.3 Click **Hardware Setup…** and double click **Arrow-USB-Blaster** entry in the Hardware Setup tab. The Currently selected hardware should now show Arrow-USB-Blaster [USB0] (depending on your PC, the USB port number may variant).



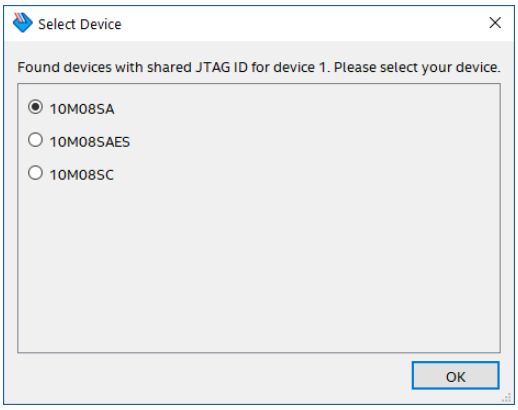
3.3.4.4 Click **Close**.

3.3.4.5 Make sure the hardware setup is Arrow-USB-Blaster [USB0] and the mode is JTAG. Click **Auto Detect**.



3.3.4.6 If the configuration has been added by default, you can skip the following steps and continue with the 3.3.4.11 point.

3.3.4.7 Select **10M08SA** device and click **OK** in the pop-up window.

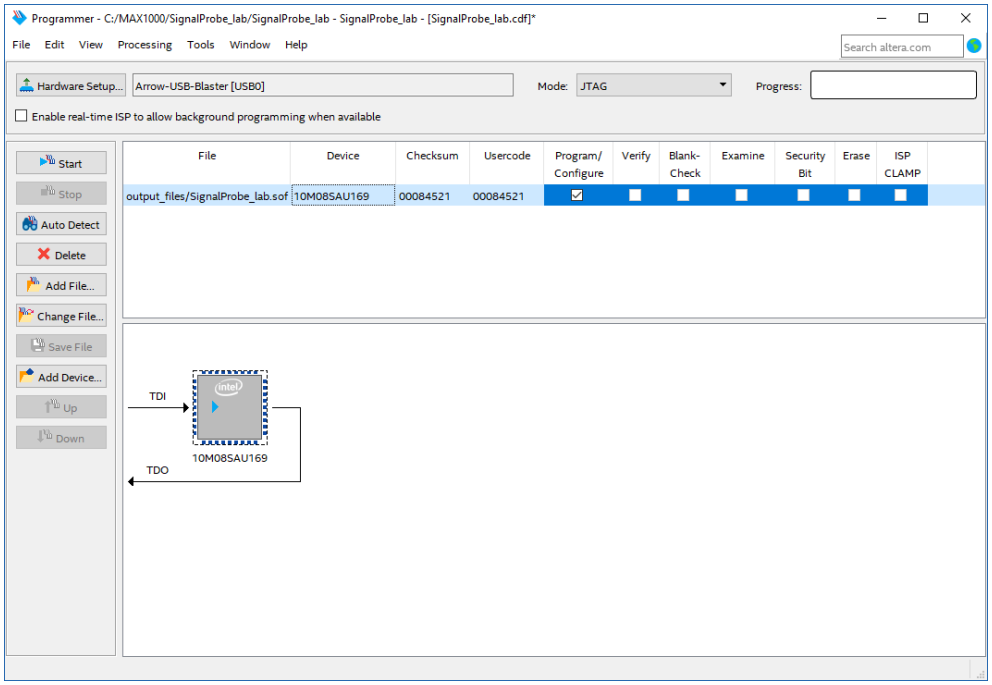


3.3.4.8 Click **Change File...** or double click <none> to choose the programming file.

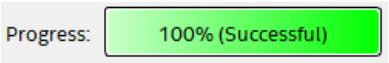
3.3.4.9 Navigate to **<project_directory>/output_files/** and select the **SignalProbe_lab.sof** file.

3.3.4.10 Click **Open**.

3.3.4.11 Make sure the Programmer shows the correct file and the correct part in the JTAG chain and check the Program/Configure checkbox.



3.3.4.12 Click **Start** to program the board. When the configuration is complete, the Progress bar should show 100% (Successful).



3.3.4.13 After the programming, when you press the user button, the duty cycle of PWM will change, and accordingly, you can check it visually on LED[0].

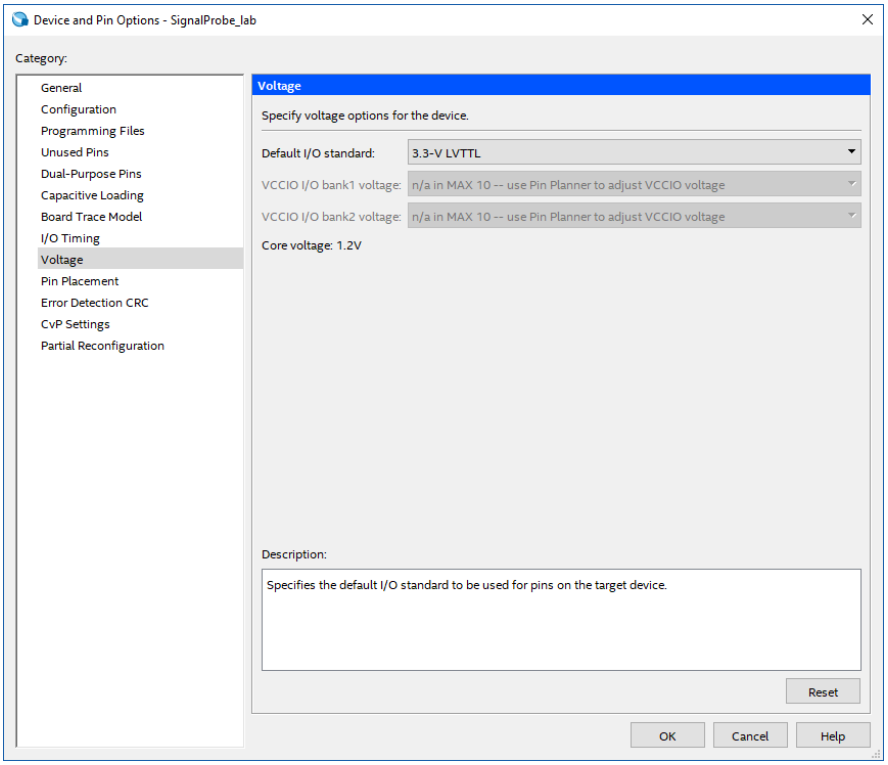
3.4 Signal Probe

3.4.1 Signal Probe setup

3.4.1.1 In the Quartus Prime, select **Assignments → Device…** and click on the **Device and Pin Options…** button.

3.4.1.2 Choose the **Voltage** category.

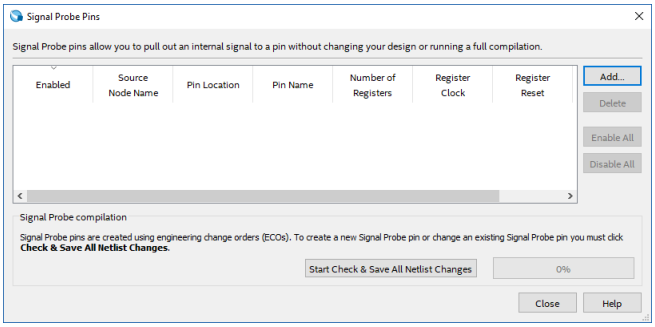
3.4.1.3 Set the Default I/O standard to **3.3-V LVTTL** from the drop-down menu.



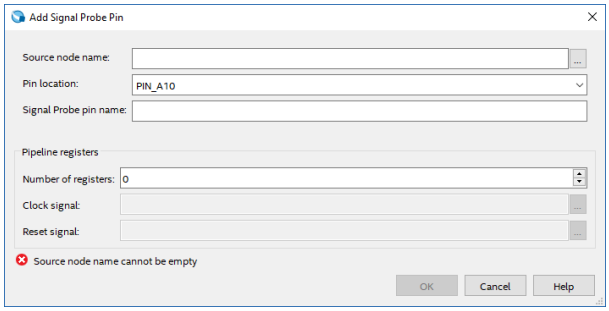
3.4.1.4 Press **OK** to close Device and Pin Options window. Press again **OK** to close Device window.

3.4.1.5 Select **Tools → Signal Probe Pins…** from the menu.

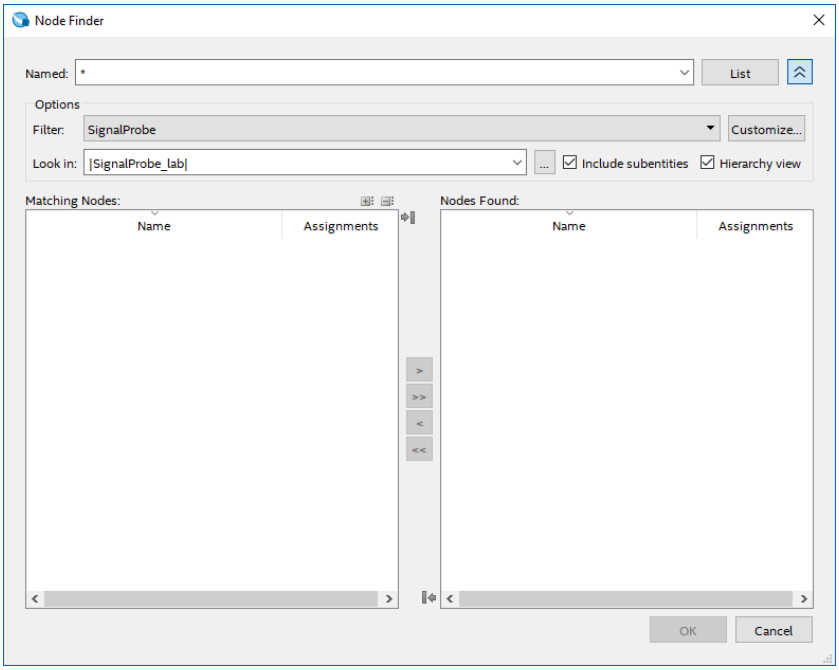
3.4.1.6 Click on **Add…** button.



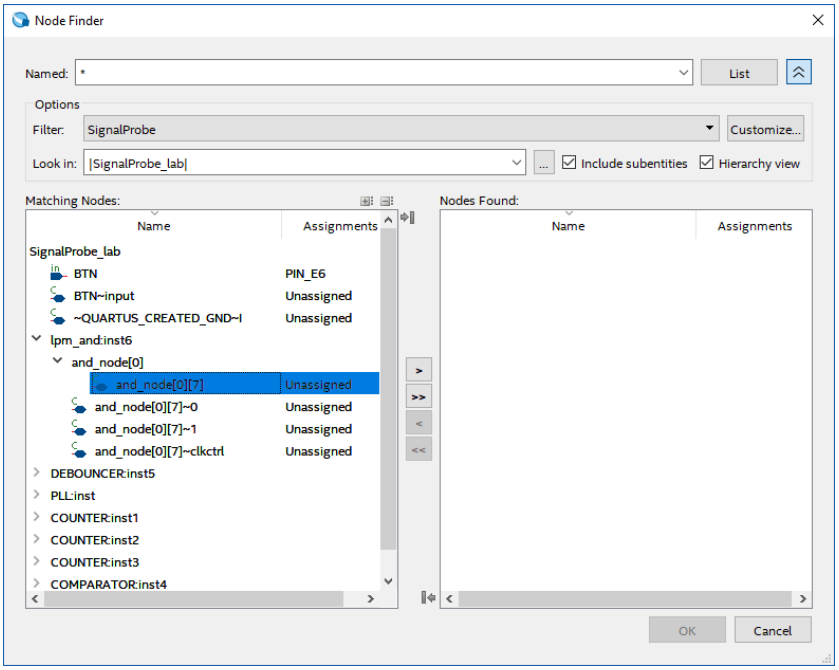
3.4.1.7 Click on the ... button to browse source node.



3.4.1.8 In the Node Finder window set the Filter to **SignalProbe** and click on the **List** button.

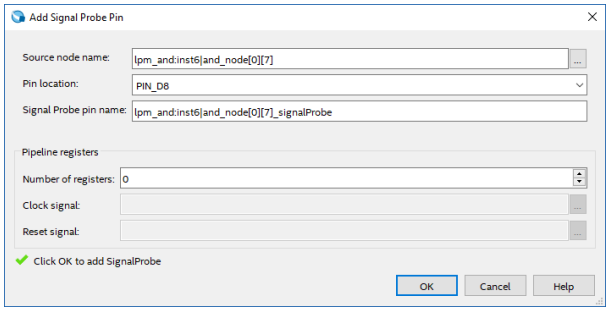


3.4.1.9 From the Matching Nodes window expand **lpm_and:inst6 → and_node[0]** and select **and_node[0][7]**.



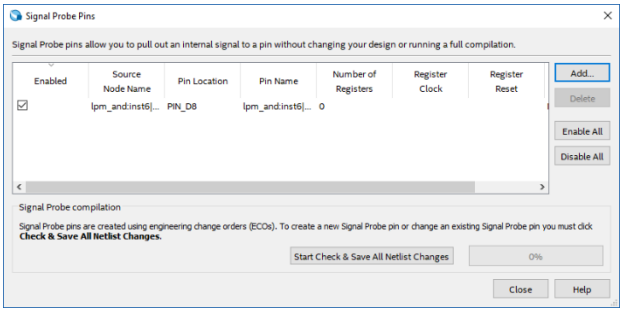
3.4.1.10 Click on the  button and click **OK**.

3.4.1.11 In the Add Signal Probe Pin window set the Pin location to **PIN_D8** and leave the rest as default.



3.4.1.12 Click **OK**.

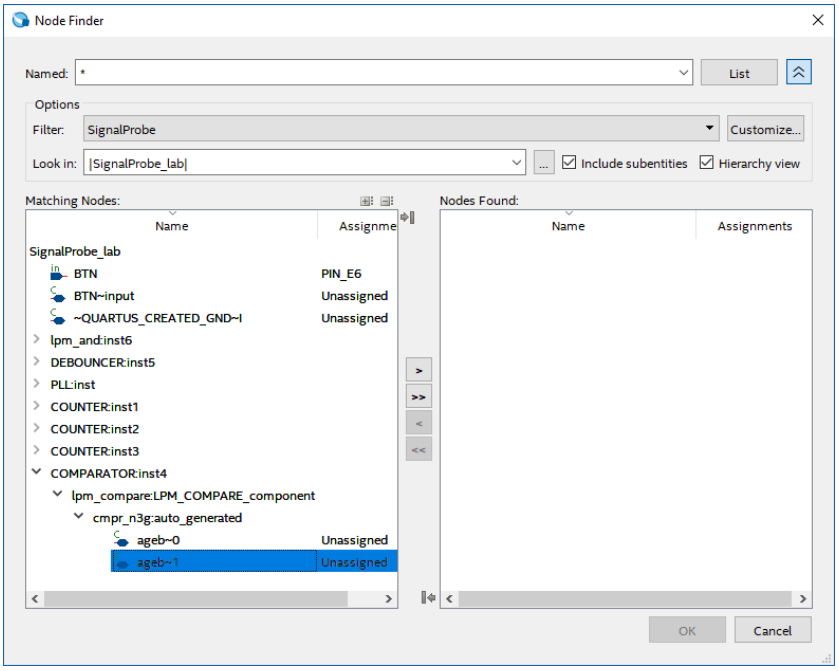
3.4.1.13 Click again on the **Add…** button in the Signal Probe Pins window.



3.4.1.14 Click on the ... button to browse source node in the Add Signal Probe Pin window.

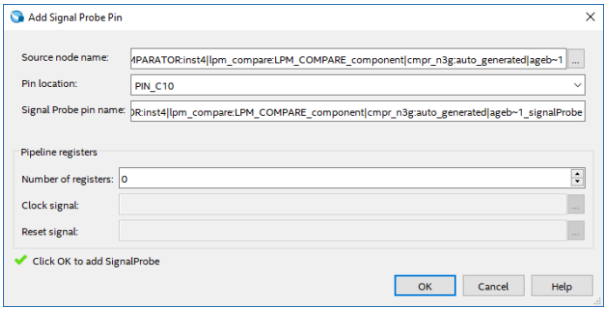
3.4.1.15 In the Node Finder window make sure that the Filter is **SignalProbe** and click on the **List** button.

3.4.1.16 From the Matching Nodes window expand **COMPARATOR:inst4 → lpm_compare:LPM_COMPARE_component → cmpr_n3g:auto_generated** and select **ageb~1**



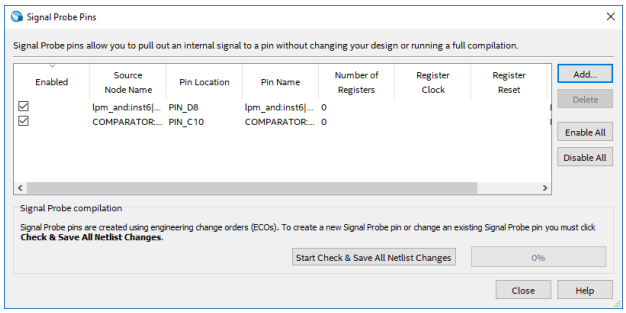
3.4.1.17 Click on the > button and click **OK**.

3.4.1.18 In the Add Signal Probe Pin window set the Pin location to **PIN_C10** and leave the rest as default.



3.4.1.19 Click **OK**.

3.4.1.20 Make sure that both nodes are enabled and click **Start Check & Save all Netlist Changes**.



3.4.1.21 When the ECO fitting is complete, click **close**.

3.4.2 Reconfiguration

3.4.2.1 Open Quartus Programmer window.

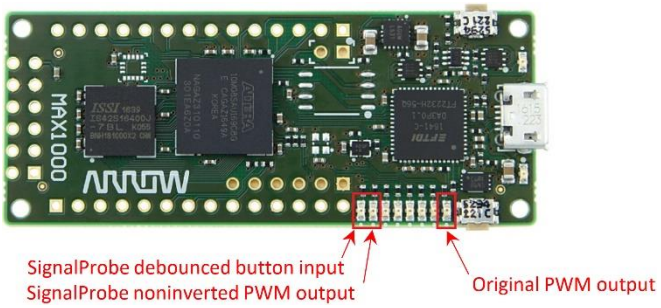
3.4.2.2 Click **Start** to program the board.

The SignalProbe compilation generated a new programming file (.sof) which is automatically updated in the Programmer. You do not need to add again.

3.4.3 Testing

The LED[7] are driven by the debounced user button. When you press it, the LED will blink accordingly to this.

The LED[6] are driven by the noninverted PWM output. The brightness will change inversely as the original, LED[0] output. Please note, because the comparator is set to a >= b, the high value will appear on the output in every counting period. So, in contrast to LED[0], this LED will never turn off completely.



CONGRATULATIONS! YOU HAVE SUCCESSFULLY COMPLETED THE SIGNAL PROBE DEBUGGING LAB!

5 Revision History

Version	Change Log	Date of Change
V1.0	Initial Version	28/01/2019

6 Legal Disclaimer

ARROW ELECTRONICS

EVALUATION BOARD LICENSE AGREEMENT

By using this evaluation board or kit (together with all related software, firmware, components, and documentation provided by Arrow, "Evaluation Board"), You ("You") are agreeing to be bound by the terms and conditions of this Evaluation Board License Agreement ("Agreement"). Do not use the Evaluation Board until You have read and agreed to this Agreement. Your use of the Evaluation Board constitutes Your acceptance of this Agreement.

PURPOSE

The purpose of this evaluation board is solely intended for evaluation purposes. Any use of the Board beyond these purposes is on your own risk. Furthermore, according the applicable law, the offering Arrow entity explicitly does not warrant, guarantee or provide any remedies to you with regard to the board.

LICENSE

Arrow grants You a non-exclusive, limited right to use the enclosed Evaluation Board offering limited features only for Your evaluation and testing purposes in a research and development setting. Usage in a live environment is prohibited. The Evaluation Board shall not be, in any case, directly or indirectly assembled as a part in any production of Yours as it is solely developed to serve evaluation purposes and has no direct function and is not a finished product.

EVALUATION BOARD STATUS

The Evaluation Board offers limited features allowing You only to evaluate and test purposes. The Evaluation Board is not intended for consumer or household use. You are not authorized to use the Evaluation Board in any production system, and it may not be offered for sale or lease, or sold, leased or otherwise distributed for commercial purposes.

OWNERSHIP AND COPYRIGHT

Title to the Evaluation Board remains with Arrow and/or its licensors. This Agreement does not involve any transfer of intellectual property rights ("IPR) for evaluation board. You may not remove any copyright or other proprietary rights notices without prior written authorization from Arrow or it licensors.

RESTRICTIONS AND WARNINGS

Before You handle or use the Evaluation Board, You shall comply with all such warnings and other instructions and employ reasonable safety precautions in using the Evaluation Board. Failure to do so may result in death, personal injury, or property damage.

You shall not use the Evaluation Board in any safety critical or functional safety testing, including but not limited to testing of life supporting, military or nuclear applications. Arrow expressly disclaims any responsibility for such usage which shall be made at Your sole risk.

WARRANTY

Arrow warrants that it has the right to provide the evaluation board to you. This warranty is provided by Arrow in lieu of all other warranties, written or oral, statutory, express or implied, including any warranty as to merchantability, non-infringement, fitness for any particular purpose, or uninterrupted or error-free operation, all of which are expressly disclaimed. The evaluation board is provided "as is" without any other rights or warranties, directly or indirectly.

You warrant to Arrow that the evaluation board is used only by electronics experts who understand the dangers of handling and using such items, you assume all responsibility and liability for any improper or unsafe handling or use of the evaluation board by you, your employees, affiliates, contractors, and designees.

LIMITATION OF LIABILITIES

In no event shall Arrow be liable to you, whether in contract, tort (including negligence), strict liability, or any other legal theory, for any direct, indirect, special, consequential, incidental, punitive, or exemplary damages with respect to any matters relating to this agreement. In no event shall arrow's liability arising out of this agreement in the aggregate exceed the amount paid by you under this agreement for the purchase of the evaluation board.

IDENTIFICATION

You shall, at Your expense, defend Arrow and its Affiliates and Licensors against a claim or action brought by a third party for infringement or misappropriation of any patent, copyright, trade secret or other intellectual property right of a third party to the extent resulting from (1) Your combination of the Evaluation Board with any other component, system, software, or firmware, (2) Your modification of the Evaluation Board, or (3) Your use of the Evaluation Board in a manner not permitted under this Agreement. You shall indemnify Arrow and its Affiliates and Licensors against and pay any resulting costs and damages finally awarded against Arrow and its Affiliates and Licensors or agreed to in any settlement, provided that You have sole control of the defense and settlement of the claim or action, and Arrow cooperates in the defense and furnishes all related evidence under its control at Your expense. Arrow will be entitled to participate in the defense of such claim or action and to employ counsel at its own expense.

RECYCLING

The Evaluation Board is not to be disposed as an urban waste. At the end of its life cycle, differentiated waste collection must be followed, as stated in the directive 2002/96/EC. In all the countries belonging to the European Union (EU Dir. 2002/96/EC) and those following differentiated recycling, the Evaluation Board is subject to differentiated recycling at the end of its life cycle, therefore: It is forbidden to dispose the Evaluation Board as an undifferentiated waste or with other domestic wastes. Consult the local authorities for more information on the proper disposal channels. An incorrect Evaluation Board disposal may cause damage to the environment and is punishable by the law.