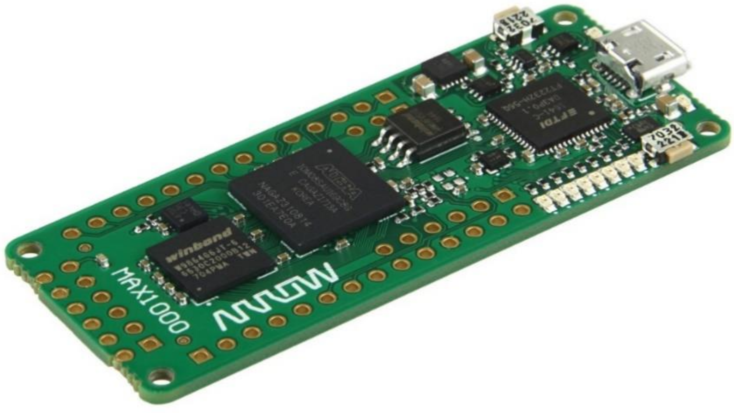
MAX1000

SignalTap 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

This tutorial provides comprehensive information to help you understand how to analyze and verify your Intel FPGA design within Quartus Prime software. This lab explains how to use Embedded Logic Analyzer megafunction, and gives detailed, step-by-step procedures on how to set up and run Embedded Logic Analyzer.

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. There are also other similar dependencies within the project that require you to enter the correct names.

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
- Lab files: SignalTap_lab_template: Template files are required to complete the lab. Includes: uart_tx.vhd, uart_tx.sdc
- 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. SignalTap overview

The SignalTap Embedded Logic Analyzer megafunction, which is provided with Quartus Prime software allows you to analyze and verify System-on-a-Programmable Chip (SOPC) designs. It is a powerful tool that lets you capture signals from internal FPGA nodes while the device is running in system, which gives you non-instructive access to signals from internal device nodes.

The Arrow USB Blaster, which is the onboard programming interface of MAX1000, provide analysis support, allowing you to transfer signal data to the Quartus Prime software. Using the intuitive Quartus interface, you can select signals, set up triggering events, configure memory, and display waveforms. You can also use data retrieved by Embedded Logic Analyzer to debug and verify your design.

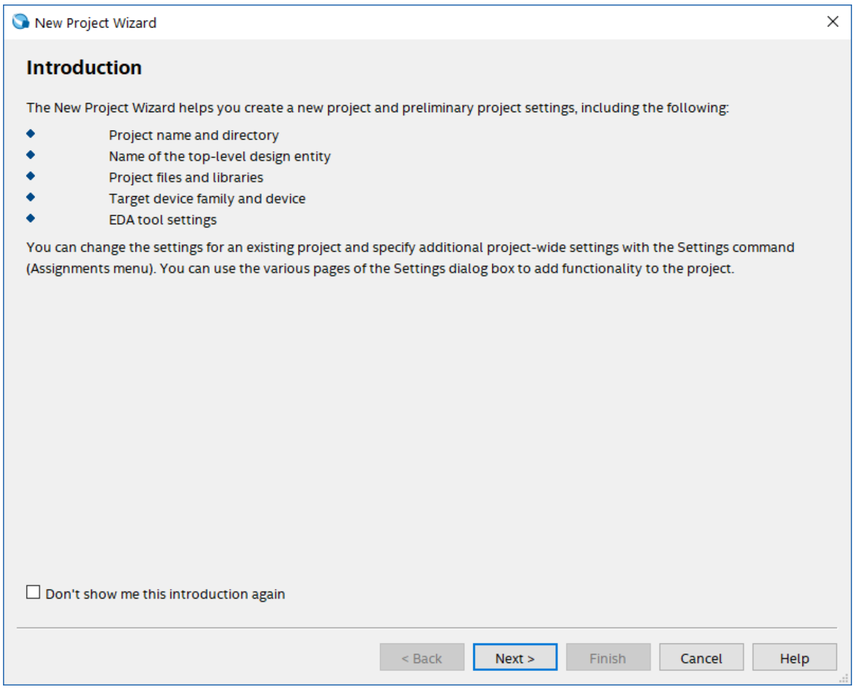
4. Project with MAX1000

4.1 Quartus Prime project

4.1.1 Create a new Quartus Prime project

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

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



New Project Wizard

Introduction

The New Project Wizard helps you create a new project and preliminary project settings, including the following:

- Project name and directory
- Name of the top-level design entity
- Project files and libraries
- Target device family and device
- EDA tool settings

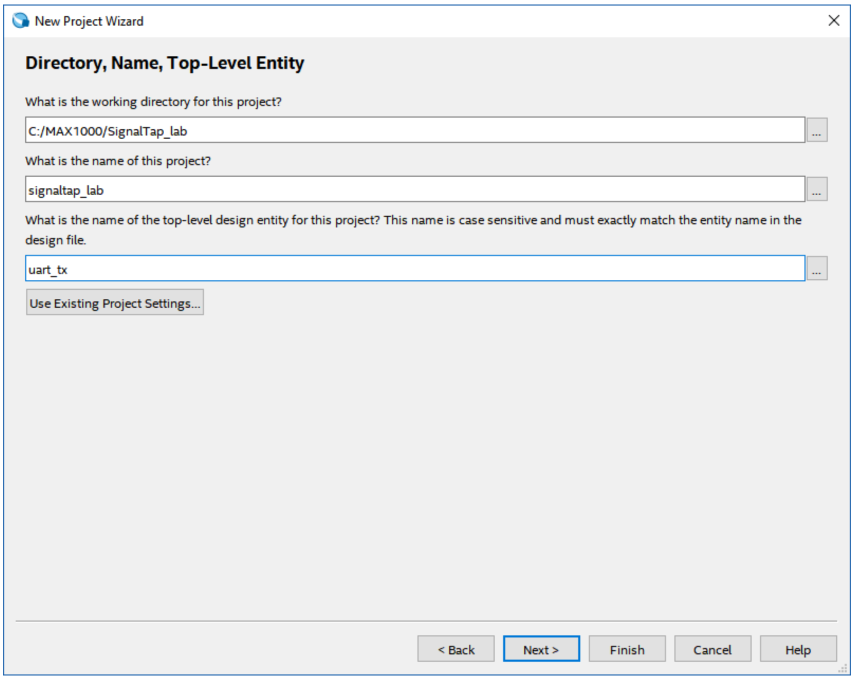
You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.

☐ Don't show me this introduction again

4.1.1.3 Click **Next**.

4.1.1.4 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/SignalTap_lab**
- Specify the name of the project: **signaltap_lab**
- Specify the name of the top-level entity: **uart_tx**



New Project Wizard

Directory, Name, Top-Level Entity

What is the working directory for this project?

C:/MAX1000/SignalTap_lab

What is the name of this project?

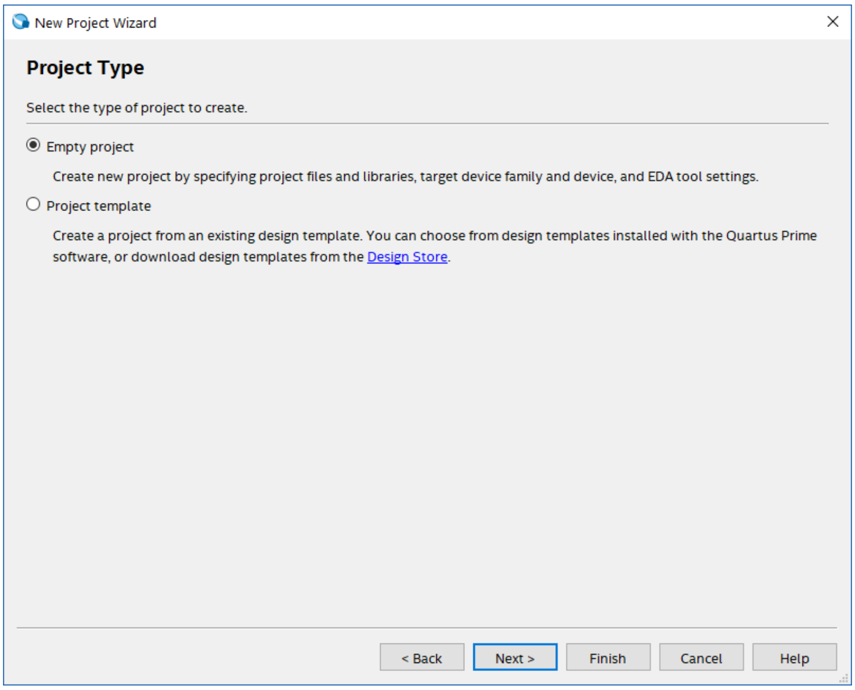
signaltap_lab

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

uart_tx

4.1.1.5 Click **Next**.

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



New Project Wizard

Project Type

Select the type of project to create.

● Empty project

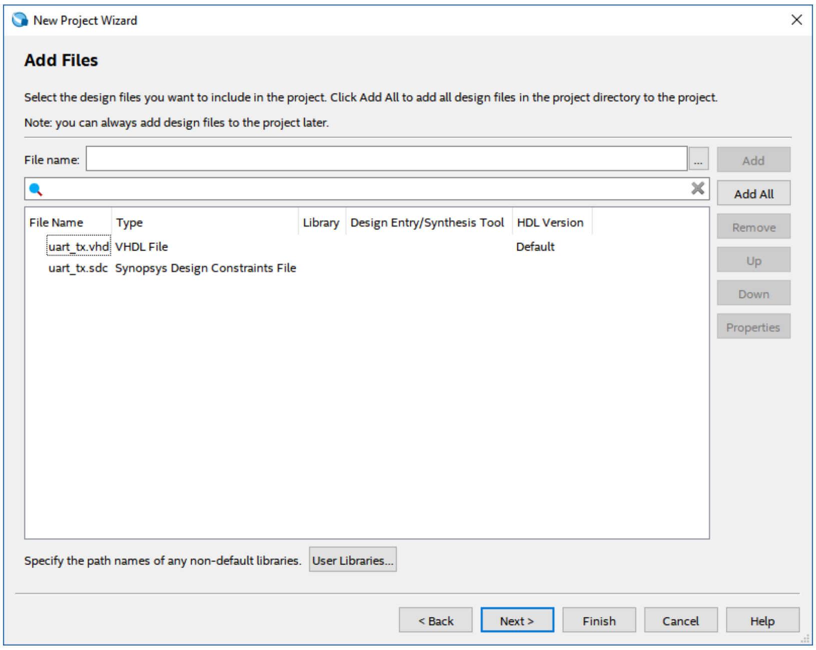
Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.

○ Project template

Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the Design Store.

4.1.1.7 On the Add Files page, add source files to the project by clicking on the ... button and browse into the lab files folder where you will locate the provided design files and add:

- **uart_tx.vhd**
- **uart_tx.sdc**



New Project Wizard

Add Files

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.

Note: you can always add design files to the project later.

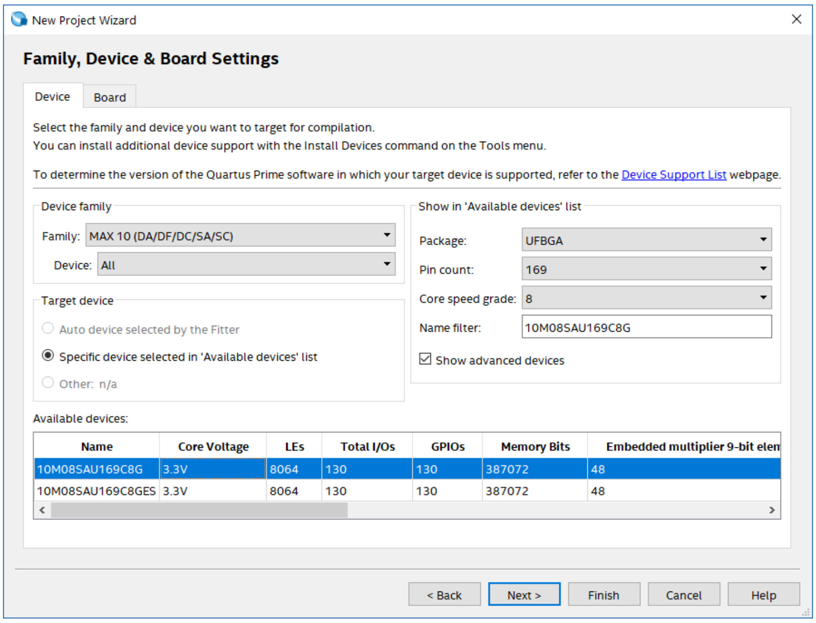
File name:

File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version
uart_tx.vhd	VHDL File			Default
uart_tx.sdc	Synopsys Design Constraints File			

Specify the path names of any non-default libraries.

4.1.1.8 Click **Next**.

4.1.1.9 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**.



New Project Wizard

Family, Device & Board Settings

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the Device Support List webpage.

Device family

Family: MAX 10 (DA/DF/DC/SA/SC)

Device: All

Target device

○ Auto device selected by the Fitter

● Specific device selected in 'Available devices' list

○ Other: n/a

Show in 'Available devices' list

Package: UFBGA

Pin count: 169

Core speed grade: 8

Name filter: 10M08SAU169C8G

☑ Show advanced devices

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elem
10M08SAU169C8G	3.3V	8064	130	130	387072	48
10M08SAU169C8GES	3.3V	8064	130	130	387072	48

4.1.1.10 Click **Finish**.

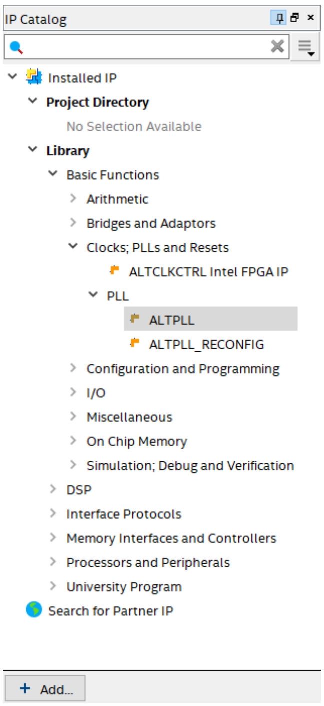
4.2 Design entry

Overview: In this module you will add missing components to your design.

4.2.1 Add PLL to the Quartus project

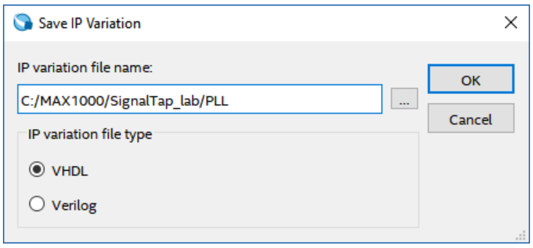
4.2.1.1 From the IP Catalog panel on the left side, expand the menus for the **Basic Functions → Clocks; PLLs and Resets → PLL** and double click on **ALTPLL**.

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



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

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


Save IP Variation

IP variation file name: C:/MAX1000/SignalTap_lab/PLL

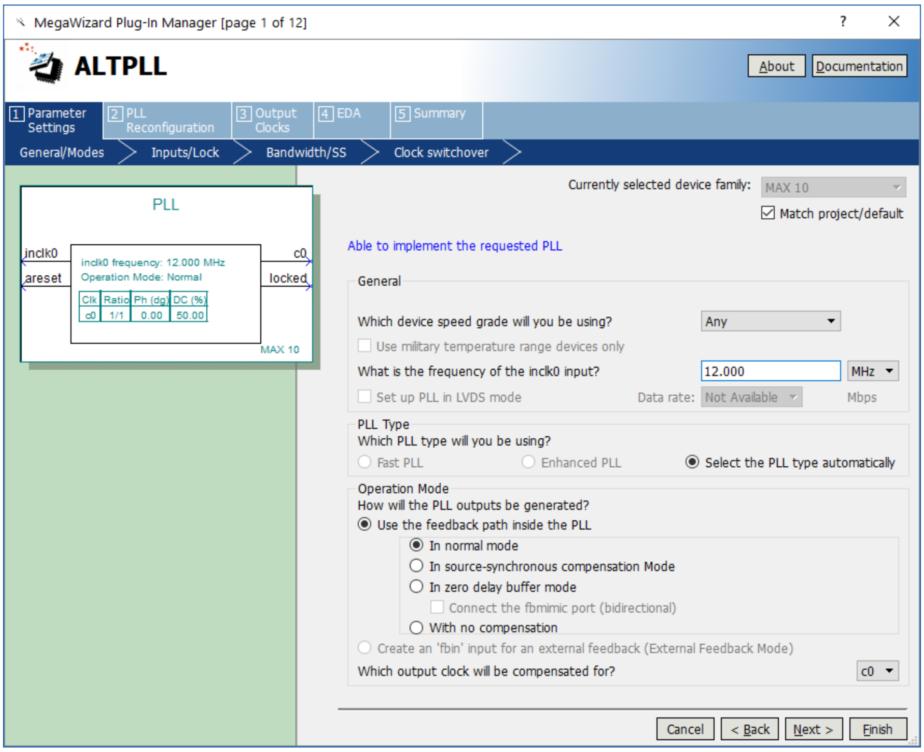
IP variation file type

● VHDL

○ Verilog

4.2.1.3 Click **OK**.

4.2.1.4 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.



MegaWizard Plug-In Manager [page 1 of 12]

ALTPLL

inclk0 frequency: 12.000 MHz
Operation Mode: Normal

Clk	Ratio	Ph (dg)	DC (%)
c0	1/1	0.00	50.00

Currently selected device family: MAX 10

☑ Match project/default

Able to implement the requested PLL

General

Which device speed grade will you be using? Any

□ Use military temperature range devices only

What is the frequency of the inclk0 input? 12.000 MHz

□ Set up PLL in LVDS mode Data rate: Not Available Mbps

PLL Type

Which PLL type will you be using?

○ Fast PLL ○ Enhanced PLL ● Select the PLL type automatically

Operation Mode

How will the PLL outputs be generated?

● Use the feedback path inside the PLL

● In normal mode

○ In source-synchronous compensation Mode

○ In zero delay buffer mode

□ Connect the fbmimic port (bidirectional)

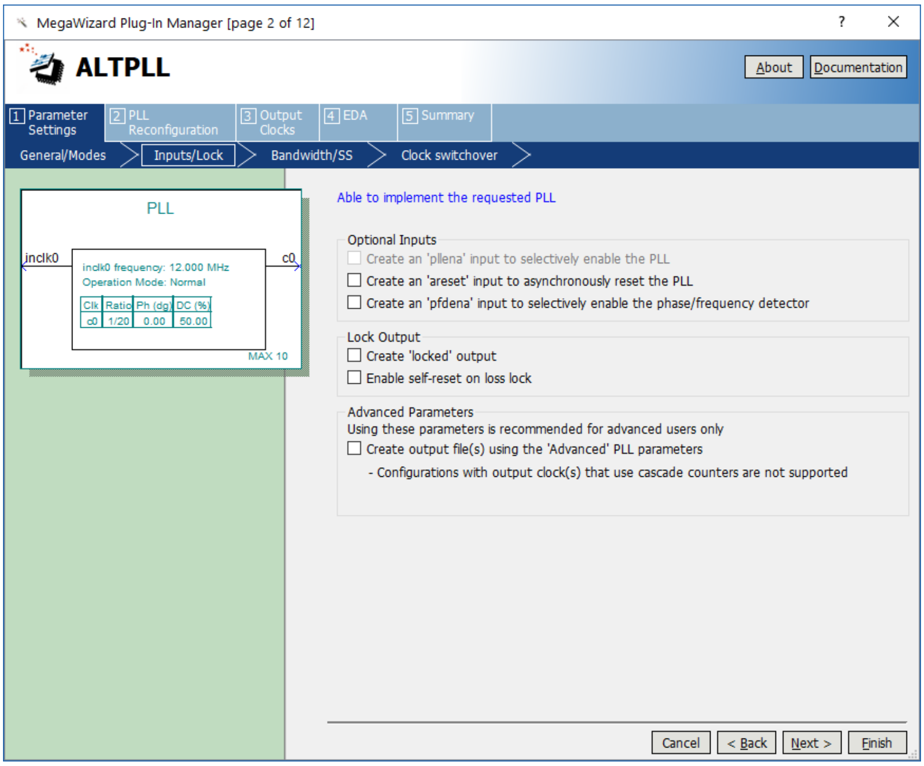
○ With no compensation

○ Create an 'fbin' input for an external feedback (External Feedback Mode)

Which output clock will be compensated for? c0

4.2.1.5 Click **Next**.

4.2.1.6 Under Input/Lock tab (page 2 of 12) uncheck 'areset' input and locked output option.



MegaWizard Plug-In Manager [page 2 of 12]

ALTPLL

inclk0 frequency: 12.000 MHz
Operation Mode: Normal

Clk	Ratio	Ph (dg)	DC (%)
c0	1/20	0.00	50.00

Able to implement the requested PLL

Optional Inputs

□ Create an 'pllena' input to selectively enable the PLL

□ Create an 'areset' input to asynchronously reset the PLL

□ Create an 'pfdena' input to selectively enable the phase/frequency detector

Lock Output

□ Create 'locked' output

□ Enable self-reset on loss lock

Advanced Parameters

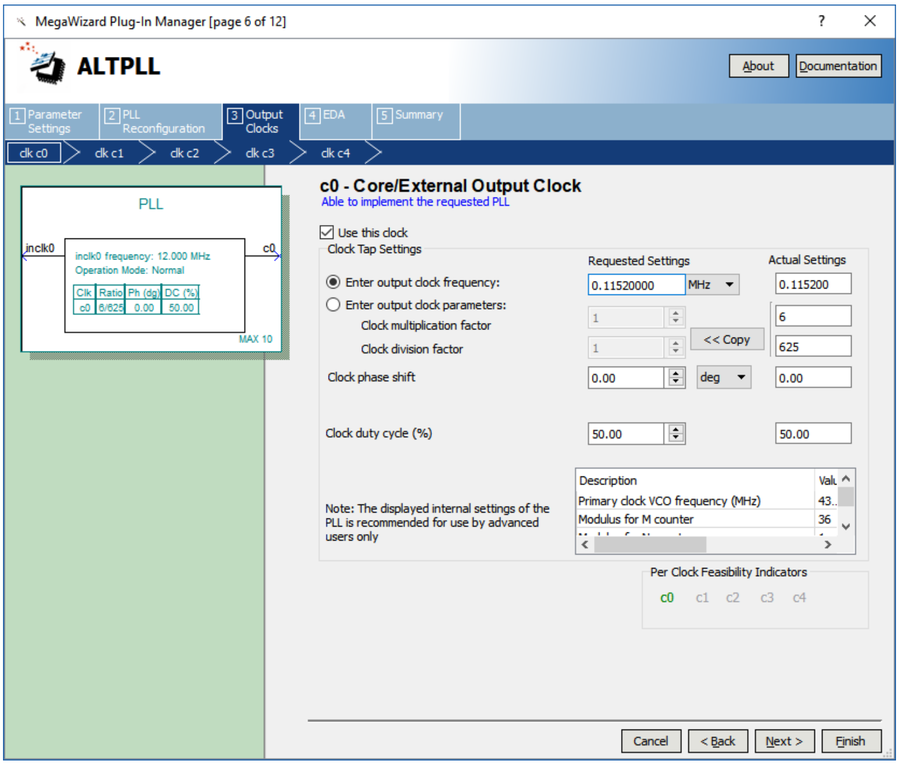
Using these parameters is recommended for advanced users only

□ Create output file(s) using the 'Advanced' PLL parameters

- Configurations with output clock(s) that use cascade counters are not supported

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

4.2.1.8 Under the clk c0 tab (page 6 of 12) select “Enter output clock frequency” and set **0.1152 MHz**.


MegaWizard Plug-In Manager [page 6 of 12]

ALTPLL

About | Documentation

1 Parameter Settings | 2 PLL Reconfiguration | 3 Output Clocks | 4 EDA | 5 Summary

clk c0 | clk c1 | clk c2 | clk c3 | clk c4

PLL
inclk0
c0
inclk0 frequency: 12.000 MHz
Operation Mode: Normal
Clk | Ratio | Ph (dg) | DC (%)
c0 | 6/625 | 0.00 | 50.00
MAX 10

c0 - Core/External Output Clock

Able to implement the requested PLL

☑ Use this clock

Clock Tap Settings

	Requested Settings		Actual Settings
● Enter output clock frequency:	0.11520000	MHz	0.115200
○ Enter output clock parameters:			
Clock multiplication factor	1	<< Copy	6
Clock division factor	1		625
Clock phase shift	0.00	deg	0.00
Clock duty cycle (%)	50.00		50.00

Note: The displayed internal settings of the PLL is recommended for use by advanced users only

Description	Valu
Primary clock VCO frequency (MHz)	43..
Modulus for M counter	36

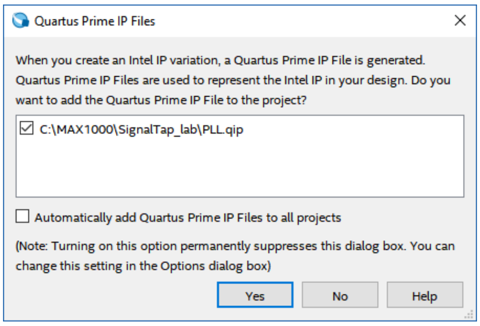
Per Clock Feasibility Indicators

c0 c1 c2 c3 c4

Cancel | < Back | Next > | Finish

4.2.1.9 Click **Finish.** This will take you to the Summary tab (page 12 of 12). Click **Finish** again to close ALTPLL MegaWizard Manager.

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


Quartus Prime IP Files

When you create an Intel IP variation, a Quartus Prime IP File is generated. Quartus Prime IP Files are used to represent the Intel IP in your design. Do you want to add the Quartus Prime IP File to the project?

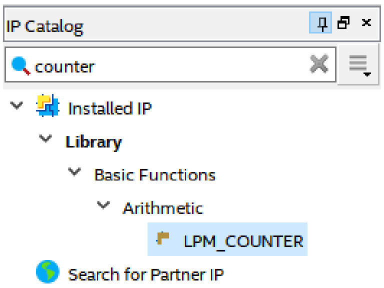
☑ C:\MAX1000\SignalTap_lab\PLL.qip

☐ Automatically add Quartus Prime IP Files to all projects

(Note: Turning on this option permanently suppresses this dialog box. You can change this setting in the Options dialog box)

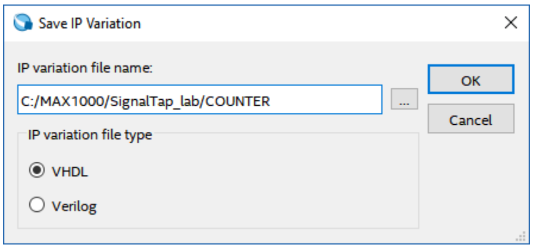
Yes | No | Help

4.2.1.11 In the search bar of IP Catalog, type “counter”, and add **LPM_COUNTER**.



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

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



IP variation file name: C:/MAX1000/SignalTap_lab/COUNTER

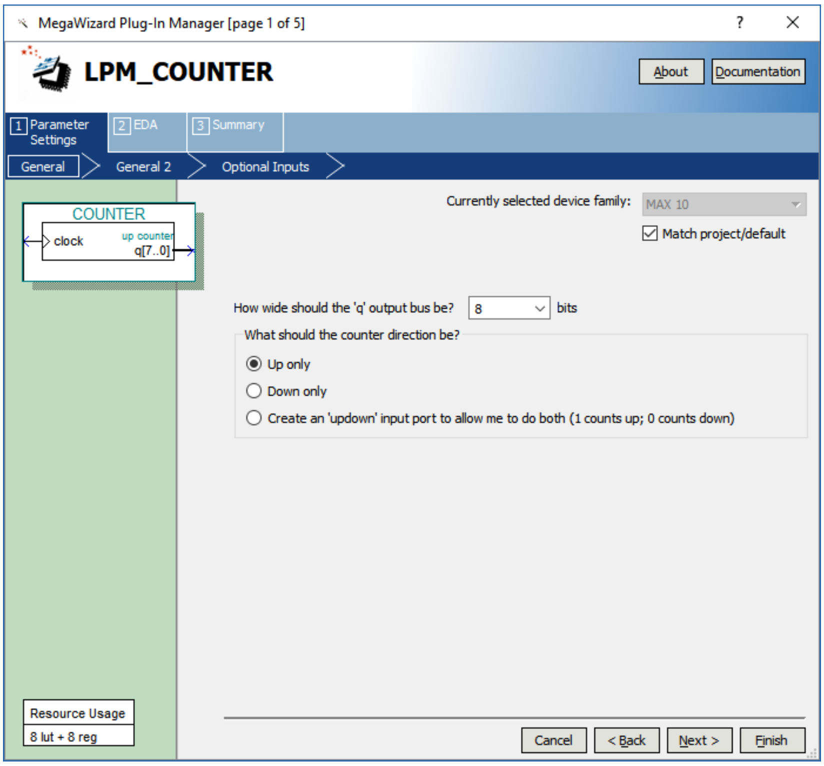
IP variation file type

● VHDL

○ Verilog

4.2.1.13 Click **OK**.

4.2.1.14 On the General Parameter Settings tab (page 1 of 5) accept all defaults and click **Next**.



Currently selected device family: MAX 10

☑ Match project/default

How wide should the 'q' output bus be? 8 bits

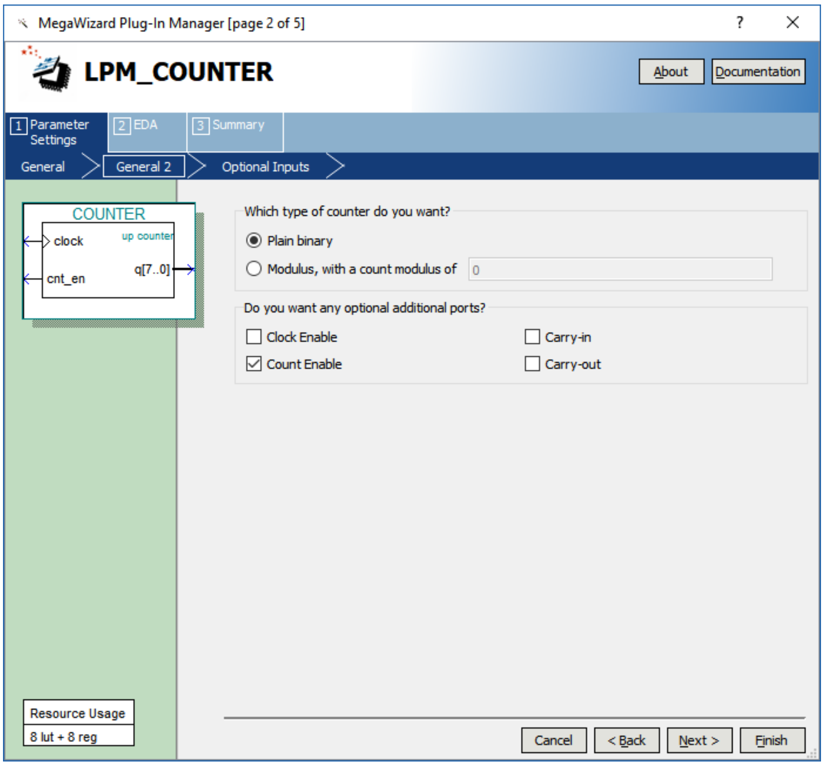
What should the counter direction be?

● Up only

○ Down only

○ Create an 'updown' input port to allow me to do both (1 counts up; 0 counts down)

4.2.1.15 On the General 2 Parameter Settings tab (page 2 of 5) select **Count Enable** and click **Next**.



MegaWizard Plug-In Manager [page 2 of 5]

LPM_COUNTER

Which type of counter do you want?

● Plain binary

○ Modulus, with a count modulus of 0

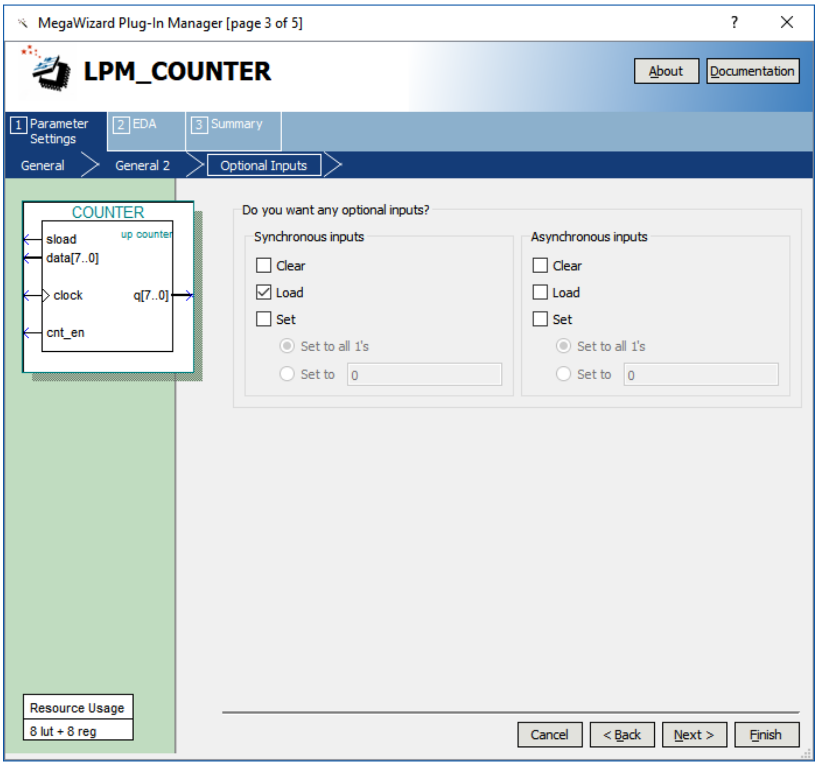
Do you want any optional additional ports?

□ Clock Enable □ Carry-in

☑ Count Enable □ Carry-out

Resource Usage
8 lut + 8 reg

4.2.1.16 On the Optional Inputs tab (page 3 of 5) select **Load**.



MegaWizard Plug-In Manager [page 3 of 5]

LPM_COUNTER

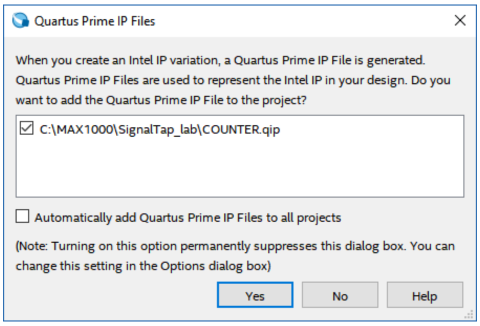
Do you want any optional inputs?

Synchronous inputs	Asynchronous inputs
□ Clear	□ Clear
☑ Load	□ Load
□ Set	□ Set
○ Set to all 1's	○ Set to all 1's
○ Set to 0	○ Set to 0

Resource Usage
8 lut + 8 reg

4.2.1.17 Click **Finish**. This will take you to the Summary tab (page 5 of 5). Click **Finish** again to close LPM_COUNTER MegaWizard Manager.

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



Quartus Prime IP Files

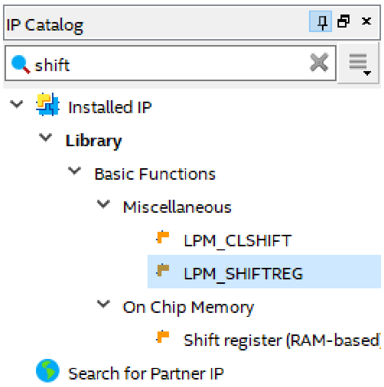
When you create an Intel IP variation, a Quartus Prime IP File is generated. Quartus Prime IP Files are used to represent the Intel IP in your design. Do you want to add the Quartus Prime IP File to the project?

☑ C:\MAX1000\SignalTap_lab\COUNTER.qip

□ Automatically add Quartus Prime IP Files to all projects

(Note: Turning on this option permanently suppresses this dialog box. You can change this setting in the Options dialog box)

4.2.1.19 In the search bar of IP Catalog, type "shift", and add **LPM_SHIFTREG**.



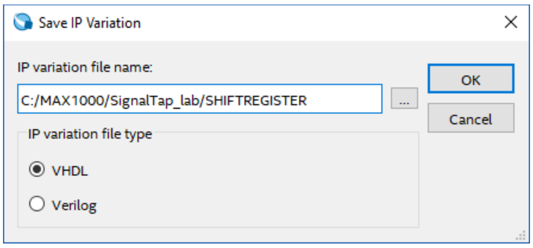
IP Catalog

shift

- Installed IP
  - Library
    - Basic Functions
      - Miscellaneous
        - LPM_CLSHIFT
        - LPM_SHIFTREG
      - On Chip Memory
        - Shift register (RAM-based
- Search for Partner IP

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

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



Save IP Variation

IP variation file name: C:/MAX1000/SignalTap_lab/SHIFTREGISTER

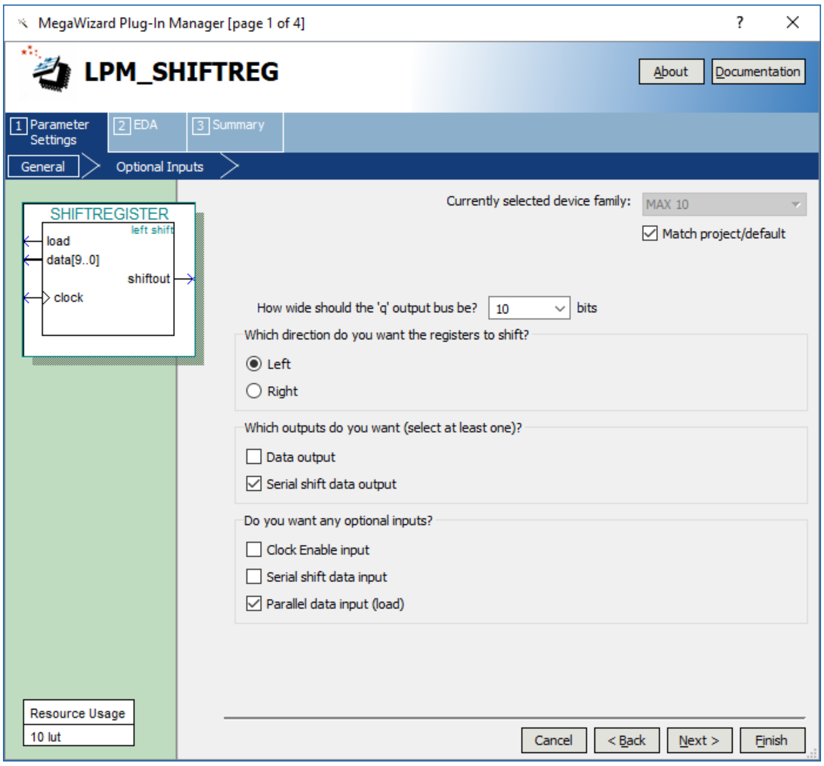
IP variation file type

● VHDL

○ Verilog

4.2.1.21 Click **OK**.

4.2.1.22 On the General tab (page 1 of 4) set the output pus to **10 bits**. For the outputs only select **Serial shift data output** and for the input only select **Parallel data input**.


MegaWizard Plug-In Manager [page 1 of 4]

LPM_SHIFTREG

Currently selected device family: MAX 10

☑ Match project/default

How wide should the 'q' output bus be? 10 bits

Which direction do you want the registers to shift?

● Left

○ Right

Which outputs do you want (select at least one)?

☐ Data output

☑ Serial shift data output

Do you want any optional inputs?

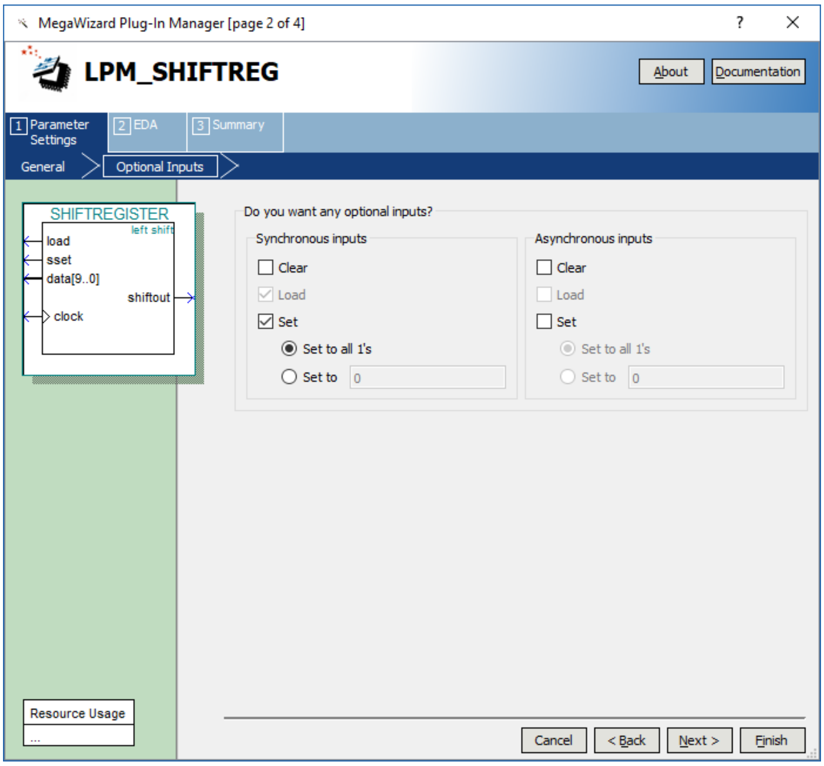
☐ Clock Enable input

☐ Serial shift data input

☑ Parallel data input (load)

4.2.1.23 Click **Next**.

4.2.1.24 On the Optional Inputs tab (page 2 of 4) only select **Set** for Synchronous inputs.


MegaWizard Plug-In Manager [page 2 of 4]

LPM_SHIFTREG

Do you want any optional inputs?

Synchronous inputs

☐ Clear

☑ Load

☑ Set

● Set to all 1's

○ Set to 0

Asynchronous inputs

☐ Clear

☐ Load

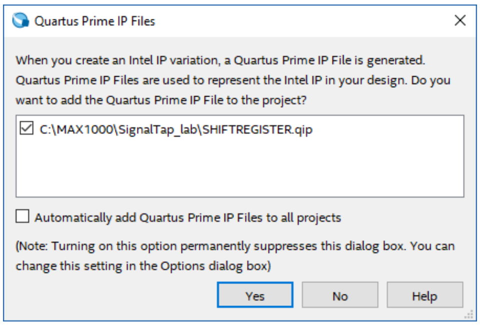
☐ Set

● Set to all 1's

○ Set to 0

4.2.1.25 Click **Finish**. This will take you to the Summary tab (page 4 of 4). Click **Finish** again to close LPM_SHIFTREG MegaWizard manager.

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



Quartus Prime IP Files

When you create an Intel IP variation, a Quartus Prime IP File is generated. Quartus Prime IP Files are used to represent the Intel IP in your design. Do you want to add the Quartus Prime IP File to the project?

☑ C:\MAX1000\SignalTap_lab\SHIFTREGISTER.qip

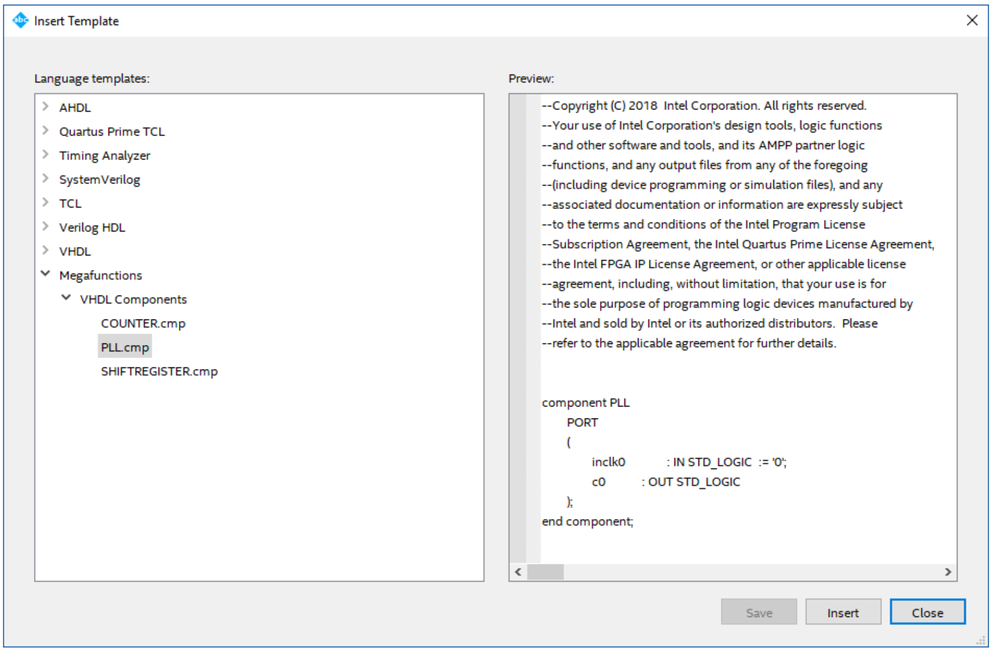
□ Automatically add Quartus Prime IP Files to all projects

(Note: Turning on this option permanently suppresses this dialog box. You can change this setting in the Options dialog box)

4.2.1.27 Double click on uart_tx to open top-level entity.

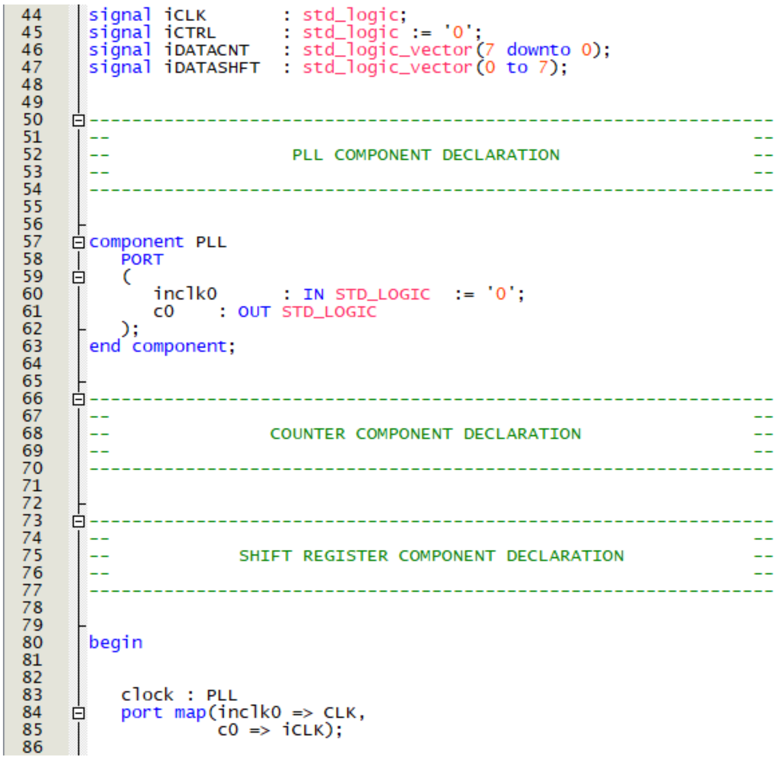
4.2.1.28 Right click in the text editor and select **Insert Template** form the pop-up menu.

4.2.1.29 In the Insert Template window, expand the “Megafunction” and after it the “VHDL Components”. The three components that were created can now be seen. Click on **PLL.cmp** to see the template code.



```
--Copyright (C) 2018  Intel Corporation. All rights reserved.
--Your use of Intel Corporation's design tools, logic functions
--and other software and tools, and its AMPP partner logic
--functions, and any output files from any of the foregoing
--(including device programming or simulation files), and any
--associated documentation or information are expressly subject
--to the terms and conditions of the Intel Program License
--Subscription Agreement, the Intel Quartus Prime License Agreement,
--the Intel FPGA IP License Agreement, or other applicable license
--agreement, including, without limitation, that your use is for
--the sole purpose of programming logic devices manufactured by
--Intel and sold by Intel or its authorized distributors.  Please
--refer to the applicable agreement for further details.


component PLL
	PORT
	(
		inclk0		: IN STD_LOGIC  := '0';
		c0		: OUT STD_LOGIC
	);
end component;
```

4.2.1.30 Move the cursor where the “PLL component declaration” place is marked in the architecture section of uart_tx.vhd and click **Insert**.
If you want, you can delete the comment lines.



```
signal iCLK       : std_logic;
signal iCTRL      : std_logic := '0';
signal iDATACNT   : std_logic_vector(7 downto 0);
signal iDATASHFT  : std_logic_vector(0 to 7);


----------------------------------------------------------------
--                                                            --
--                 PLL COMPONENT DECLARATION                  --
--                                                            --
----------------------------------------------------------------


component PLL
    PORT
    (
        inclk0      : IN STD_LOGIC  := '0';
        c0     : OUT STD_LOGIC
    );
end component;


----------------------------------------------------------------
--                                                            --
--               COUNTER COMPONENT DECLARATION                --
--                                                            --
----------------------------------------------------------------


----------------------------------------------------------------
--                                                            --
--           SHIFT REGISTER COMPONENT DECLARATION             --
--                                                            --
----------------------------------------------------------------


begin


    clock : PLL
    port map(inclk0 => CLK,
                c0 => iCLK);

```

4.2.1.31 Repeat these steps with COUNTER.cmp and SHIFTREGISTER.cmp.

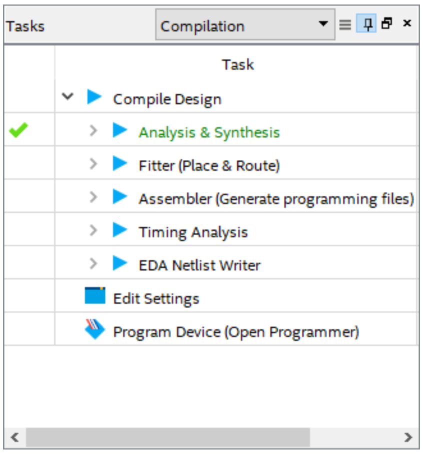
4.2.1.32 If you inserted all three components declaration into the source code, **close** Insert Template window.

4.2.1.33 Save your design by clicking on 💾 button or select **File → Save** from the menu.

4.2.2 Analysis and Synthesis

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

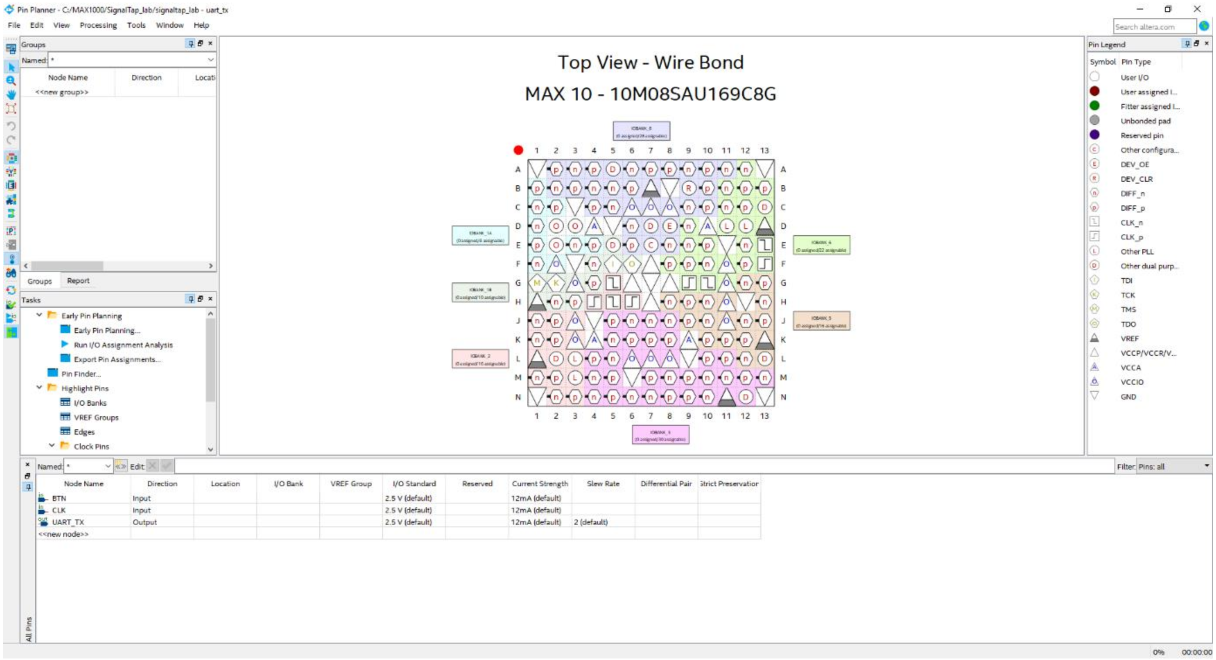
There should be no errors. If there are errors, they should be fixed before continuing. If there are no errors the compilation task windows should look like this:



	Task
	Compile Design
✓	Analysis & Synthesis
	Fitter (Place & Route)
	Assembler (Generate programming files)
	Timing Analysis
	EDA Netlist Writer
	Edit Settings
	Program Device (Open Programmer)

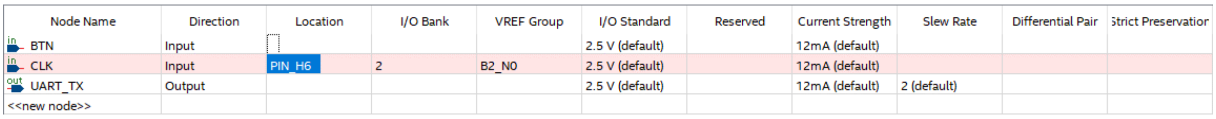
4.2.3 Pin Assignments

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



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)			
CLK	Input				2.5 V (default)		12mA (default)			
UART_TX	Output				2.5 V (default)		12mA (default)	2 (default)		
<<new node>>										

4.2.3.2 In the bottom table, type **PIN_H6** in Location column of the CLK.



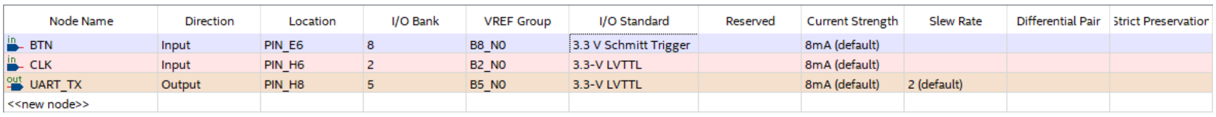
Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservatior
BTN	Input				2.5 V (default)		12mA (default)			
CLK	Input	PIN_H6	2	B2_N0	2.5 V (default)		12mA (default)			
UART_TX	Output				2.5 V (default)		12mA (default)	2 (default)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

4.2.3.3 Repeat the previous step with the following assignments:

Node Name	Pin Location
BTN	PIN_E6
UART_TX	PIN_H8

4.2.3.4 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**.

4.2.3.5 Change the I/O Standard of CLK and UART_TX 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 Preservatior
BTN	Input	PIN_E6	8	B8_N0	3.3 V Schmitt Trigger		8mA (default)			
CLK	Input	PIN_H6	2	B2_N0	3.3-V LVTTL		8mA (default)			
UART_TX	Output	PIN_H8	5	B5_N0	3.3-V LVTTL		8mA (default)	2 (default)		
<<new node>>										

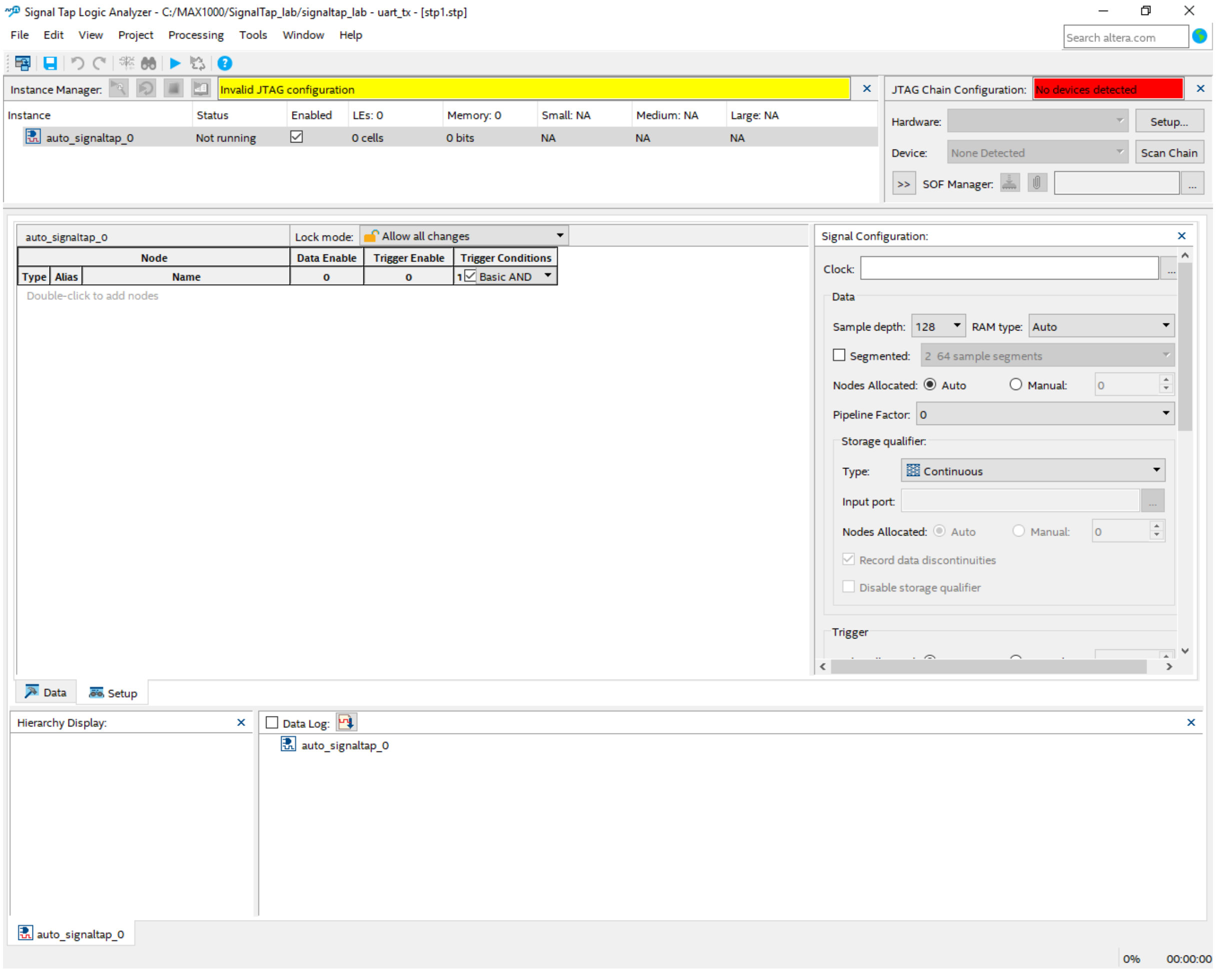
4.2.3.6 **Close** the Pin Planner, the settings are automatically saved.

4.3 Analysis

Overview: In this module you will setup the SignalTap II environment and run it on your MAX1000 board.

4.3.1 SignalTap setup

4.3.1.1 In the Quartus Prime, select in the menu **Tools → Signal Tap Logic Analyzer**.



Signal Tap Logic Analyzer - C:/MAX1000/SignalTap_lab/signaltap_lab - uart_tx - [stp1.stp]

Instance Manager: Invalid JTAG configuration

JTAG Chain Configuration: No devices detected

Instance	Status	Enabled	LEs: 0	Memory: 0	Small: NA	Medium: NA	Large: NA
auto_signaltap_0	Not running	☑	0 cells	0 bits	NA	NA	NA

Hardware:

Device: None Detected

auto_signaltap_0 Lock mode: Allow all changes

Type	Alias	Name	Data Enable	Trigger Enable	Trigger Conditions
			0	0	1☑ Basic AND

Double-click to add nodes

Signal Configuration:

Clock:

Data

Sample depth: 128 RAM type: Auto

☐ Segmented: 2 64 sample segments

Nodes Allocated: ● Auto ○ Manual: 0

Pipeline Factor: 0

Storage qualifier:

Type: Continuous

Input port:

Nodes Allocated: ● Auto ○ Manual: 0

☑ Record data discontinuities

☐ Disable storage qualifier

Trigger

Hierarchy Display:

☐ Data Log: auto_signaltap_0

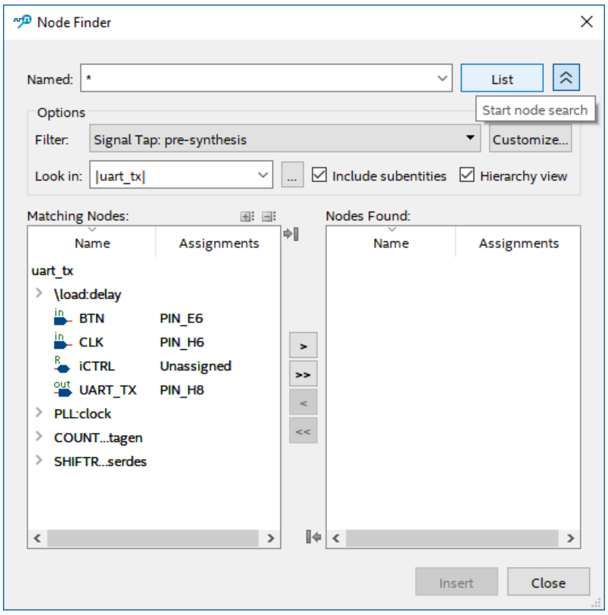
auto_signaltap_0

0% 00:00:00

4.3.1.2 Right click on the auto_signaltap_0 instance and select **Rename Instance** from the pop-up menu. Rename it to **uart_interface**.

4.3.1.3 Double click in the Setup tab to add nodes.

4.3.1.4 In the Node Finder select **Signal Tap: pre-synthesis** for the Filter and click on **List**.



Node Finder

Named: * List Start node search

Options

Filter: Signal Tap: pre-synthesis Customize...

Look in: |uart_tx| ... ☑ Include subentities ☑ Hierarchy view

Matching Nodes:

Name	Assignments
uart_tx	
\load:delay	
BTN	PIN_E6
CLK	PIN_H6
iCTRL	Unassigned
UART_TX	PIN_H8
PLL:clock	
COUNT...tagen	
SHIFTR...serdes	

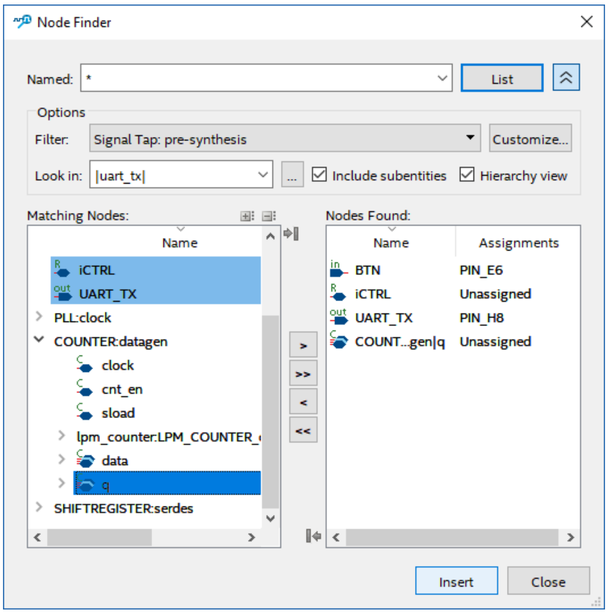
Nodes Found:

Name	Assignments

Insert Close

4.3.1.5 Select **BTN, iCTRL** and **UART_TX** and click on > button.

4.3.1.6 Expand “COUNTER:datagen” and select **q** node group. Click on > button.



Node Finder

Named: * List

Options

Filter: Signal Tap: pre-synthesis Customize...

Look in: |uart_tx| ... ☑ Include subentities ☑ Hierarchy view

Matching Nodes:

Name
iCTRL
UART_TX
PLL:clock
COUNTER:datagen
clock
cnt_en
sload
lpm_counter:LPM_COUNTER_
data
q
SHIFTREGISTER:serdes

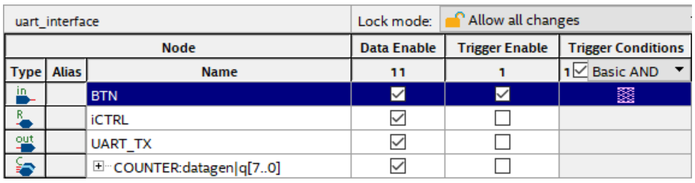
Nodes Found:

Name	Assignments
BTN	PIN_E6
iCTRL	Unassigned
UART_TX	PIN_H8
COUNT...gen|q	Unassigned

Insert Close

4.3.1.7 Click **Insert** and **close** the Node Finder window.

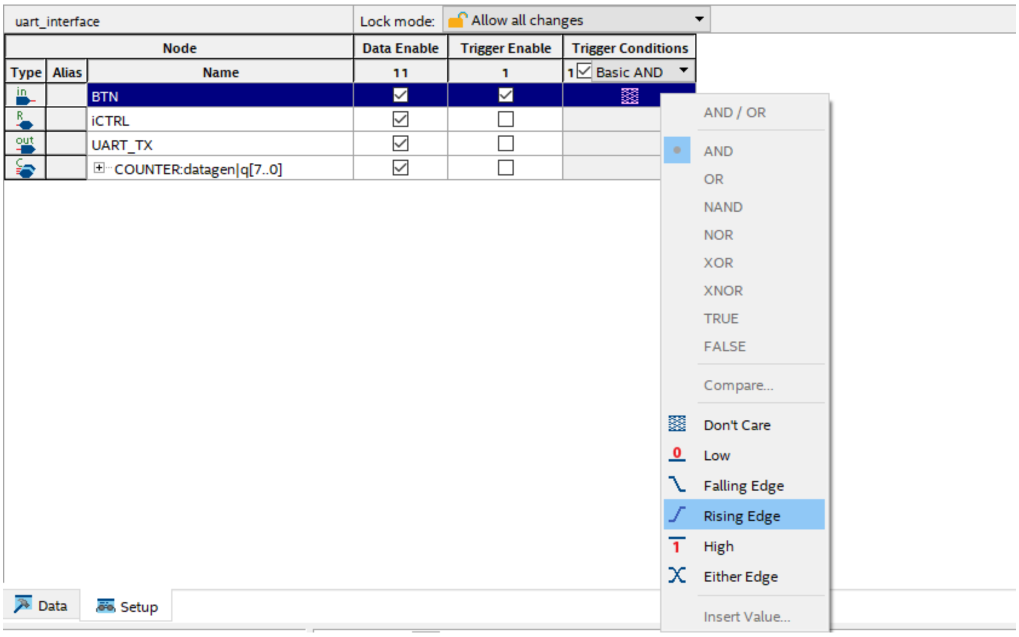
4.3.1.8 Only leave Trigger Enable for BTN and uncheck it for the other nodes.



uart_interface — Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	11	1	1 ☑ Basic AND
in		BTN	☑	☑	▨
R		iCTRL	☑	☐	
out		UART_TX	☑	☐	
		⊞ COUNTER:datagen|q[7..0]	☑	☐	

4.3.1.9 Right click in the Trigger Conditions cell of BTN and select **Rising Edge**.



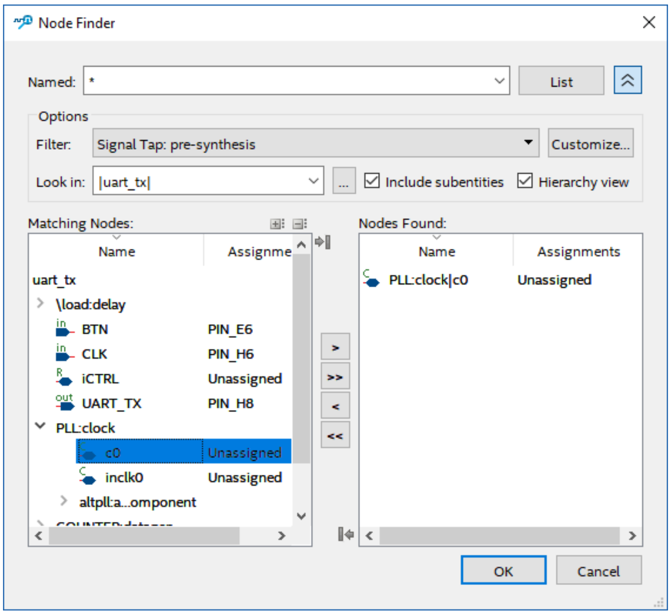
uart_interface — Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	11	1	1 ☑ Basic AND
in		BTN	☑	☑	▨
R		iCTRL	☑	☐	
out		UART_TX	☑	☐	
		⊞ COUNTER:datagen|q[7..0]	☑	☐	

4.3.1.10 In the Signal Configuration window on the right side, click on the ... button to browse the clock signal.

4.3.1.11 In the Node Finder window, make sure, that the Signal Tap: pre-synthesis is selected for the Filter, and click on the **List** button.

4.3.1.12 In the Matching Nodes window expand “PLL:clock” and select **c0**. Click on > button.



Named: *

Options

Filter: Signal Tap: pre-synthesis

Look in: |uart_tx| ☑ Include subentities ☑ Hierarchy view

Matching Nodes:

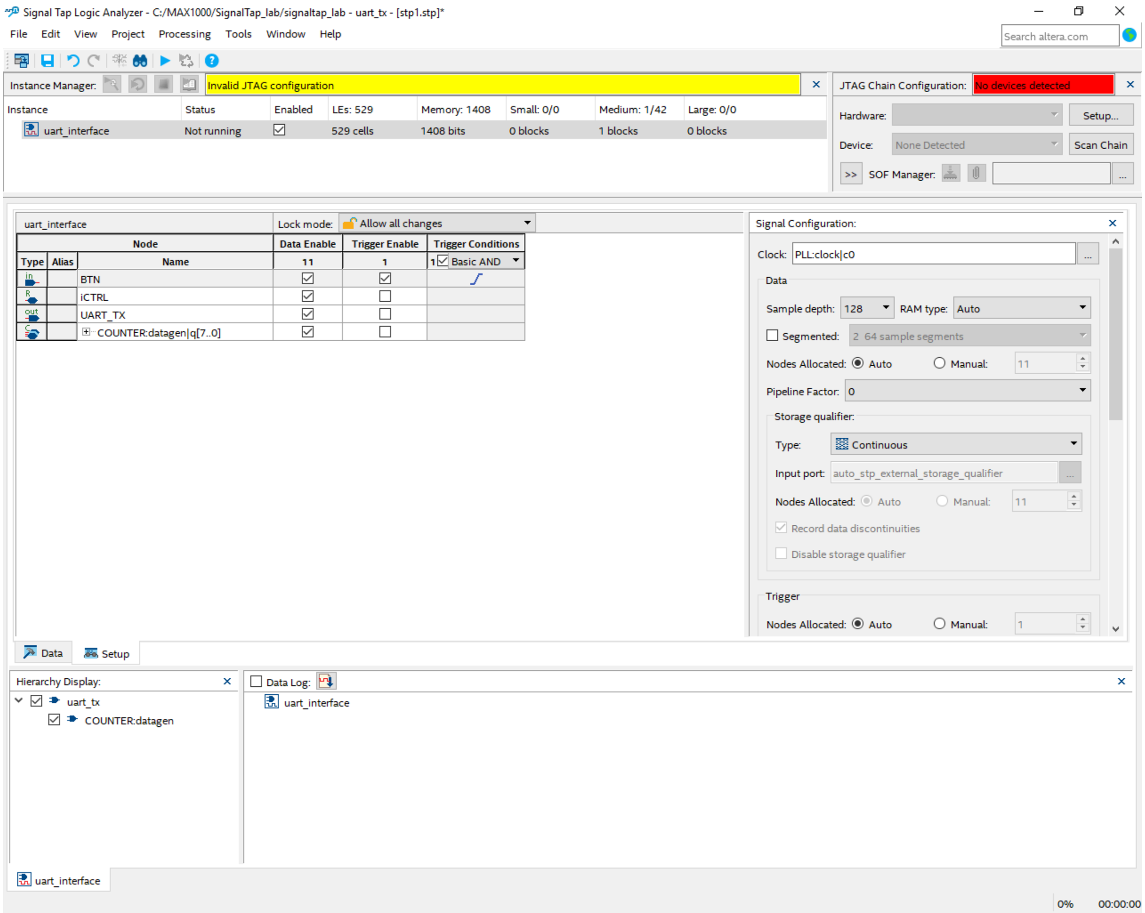
Name	Assignme
uart_tx	
\load:delay	
BTN	PIN_E6
CLK	PIN_H6
iCTRL	Unassigned
UART_TX	PIN_H8
PLL:clock	
c0	Unassigned
inclk0	Unassigned
altpll:a...omponent	
COUNTER:datagen	

Nodes Found:

Name	Assignments
PLL:clock|c0	Unassigned

4.3.1.13 Click **OK** to close window.

4.3.1.14 Leave the other parameters of Signal Configuration by default.
After you have set the nodes, you should look the following settings:



Instance Manager: Invalid JTAG configuration

JTAG Chain Configuration: No devices detected

Hardware:

Device: None Detected

Instance	Status	Enabled	LEs: 529	Memory: 1408	Small: 0/0	Medium: 1/42	Large: 0/0
uart_interface	Not running	☑	529 cells	1408 bits	0 blocks	1 blocks	0 blocks

uart_interface Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	11	1	1 ☑ Basic AND
in		BTN	☑	☑	⎽╱
R		iCTRL	☑	☐	
out		UART_TX	☑	☐	
C		⊞ COUNTER:datagen|q[7..0]	☑	☐	

Signal Configuration:

Clock: PLL:clock|c0

Data

Sample depth: 128 RAM type: Auto

☐ Segmented: 2 64 sample segments

Nodes Allocated: ● Auto ○ Manual: 11

Pipeline Factor: 0

Storage qualifier:

Type: Continuous

Input port: auto_stp_external_storage_qualifier

Nodes Allocated: ● Auto ○ Manual: 11

☑ Record data discontinuities

☐ Disable storage qualifier

Trigger

Nodes Allocated: ● Auto ○ Manual: 1

Hierarchy Display:

☑ uart_tx

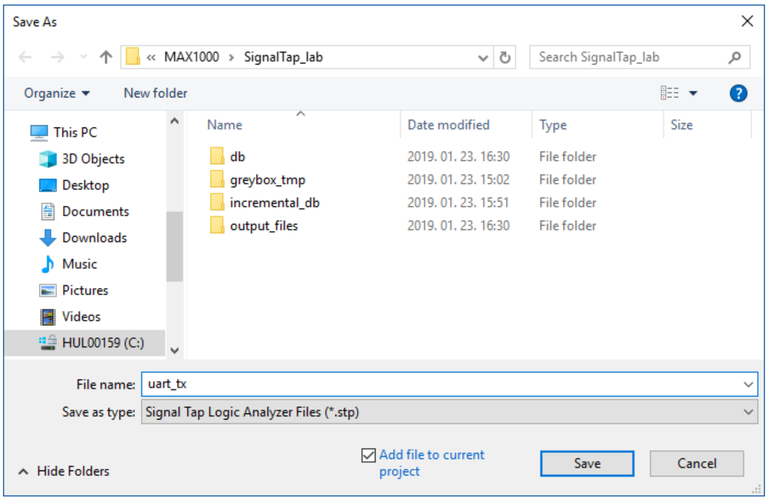
☑ COUNTER:datagen

☐ Data Log:

uart_interface

4.3.1.15 Save the analyzer setting by clicking on 💾 button or **File → Save** and enter the following information:

- File name: **uart_tx**
- Save as type: **Signal Tap Logic Analyzer Files (*.stp)**
- Make sure that “**Add file to current project**” option is checked.



Save As

« MAX1000 › SignalTap_lab

Name	Date modified	Type	Size
db	2019. 01. 23. 16:30	File folder	
greybox_tmp	2019. 01. 23. 15:02	File folder	
incremental_db	2019. 01. 23. 15:51	File folder	
output_files	2019. 01. 23. 16:30	File folder	

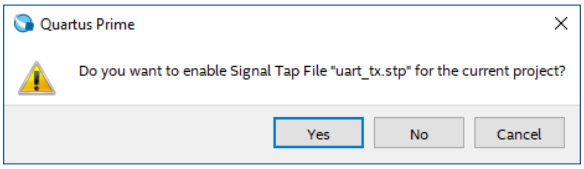
File name: uart_tx

Save as type: Signal Tap Logic Analyzer Files (*.stp)

☑ Add file to current project

4.3.1.16 Click **Save**.

4.3.1.17 In the Quartus Prime pop-up window click **Yes** to add this file to the current project.

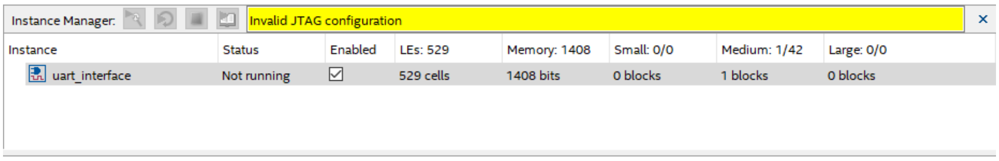


Quartus Prime

Do you want to enable Signal Tap File "uart_tx.stp" for the current project?

4.3.2 Compiling the Design

4.3.2.1 Before the compilation, make sure, that uart_interface instance is enabled in SignalTap.



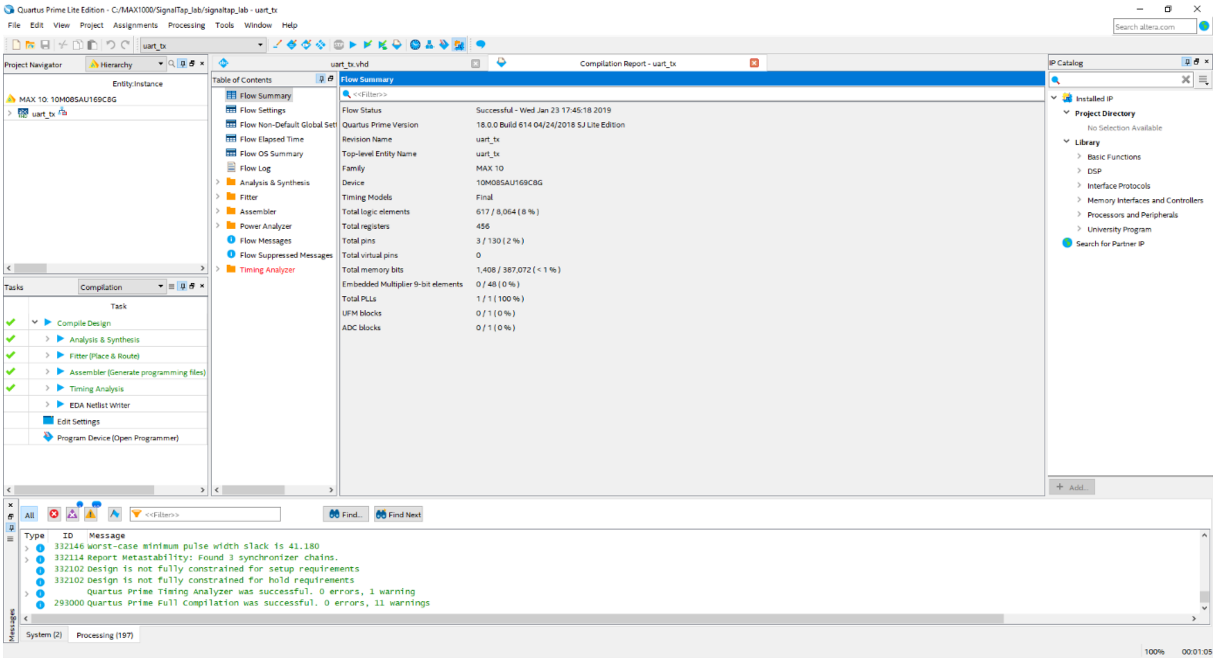
Instance Manager: Invalid JTAG configuration

Instance	Status	Enabled	LEs: 529	Memory: 1408	Small: 0/0	Medium: 1/42	Large: 0/0
uart_interface	Not running	☑	529 cells	1408 bits	0 blocks	1 blocks	0 blocks

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

This process is also available in SignalTap not only in Quartus. Use one that is more comfortable for you.

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.



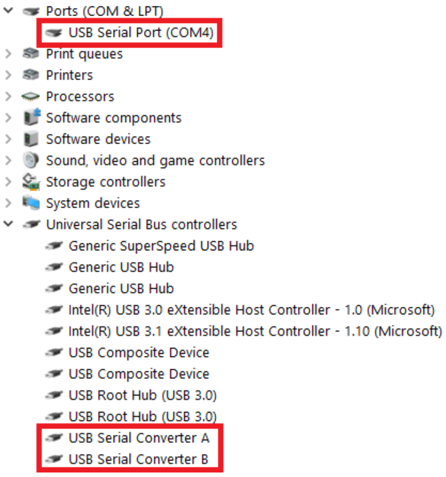
Flow Summary

Flow Status	Successful - Wed Jan 23 17:45:18 2019
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	uart_tx
Top-level Entity Name	uart_tx
Family	MAX 10
Device	10M08SAU169C8G
Timing Models	Final
Total logic elements	617 / 8,064 (8 %)
Total registers	456
Total pins	3 / 130 (2 %)
Total virtual pins	0
Total memory bits	1,408 / 387,072 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	1 / 1 (100 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

```
Type  ID     Message
      332146 Worst-case minimum pulse width slack is 41.180
      332114 Report Metastability: Found 3 synchronizer chains.
      332102 Design is not fully constrained for setup requirements
      332102 Design is not fully constrained for hold requirements
             Quartus Prime Timing Analyzer was successful. 0 errors, 1 warning
      293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings
```

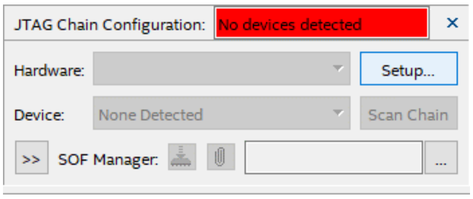
4.3.3 Configuration

4.3.3.1 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):



```
Ports (COM & LPT)
    USB Serial Port (COM4)
Print queues
Printers
Processors
Software components
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers
    Generic SuperSpeed USB Hub
    Generic USB Hub
    Generic USB Hub
    Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
    Intel(R) USB 3.1 eXtensible Host Controller - 1.10 (Microsoft)
    USB Composite Device
    USB Composite Device
    USB Root Hub (USB 3.0)
    USB Root Hub (USB 3.0)
    USB Serial Converter A
    USB Serial Converter B
```

4.3.3.2 Open the SignalTap window and click on the **Setup…** button in the top right corner.

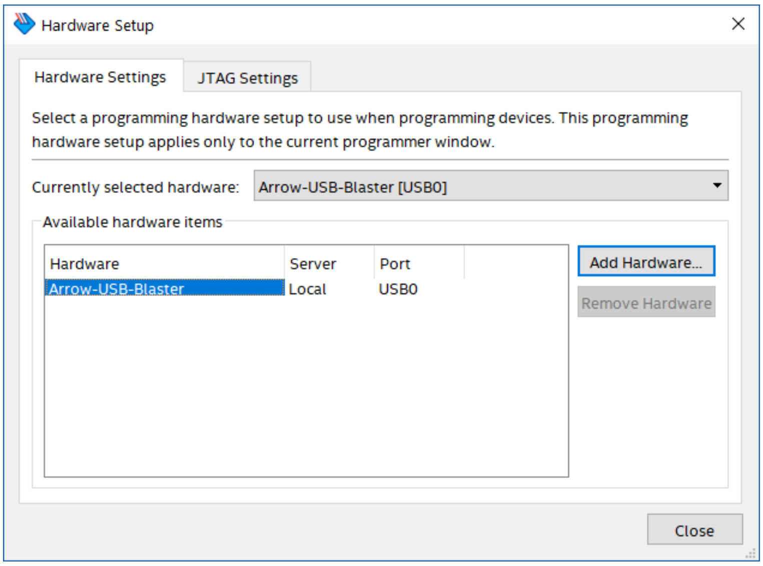


JTAG Chain Configuration: No devices detected

Hardware:

Device: None Detected

4.3.3.3 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).



Hardware Setup

Hardware Settings | JTAG Settings

Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

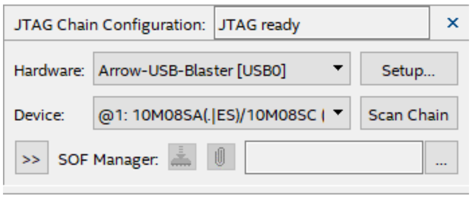
Currently selected hardware: Arrow-USB-Blaster [USB0]

Available hardware items

Hardware	Server	Port
Arrow-USB-Blaster	Local	USB0

4.3.3.4 Click **Close**.

4.3.3.5 The hardware configuration window should be updated as follows.



JTAG Chain Configuration: JTAG ready

Hardware: Arrow-USB-Blaster [USB0]

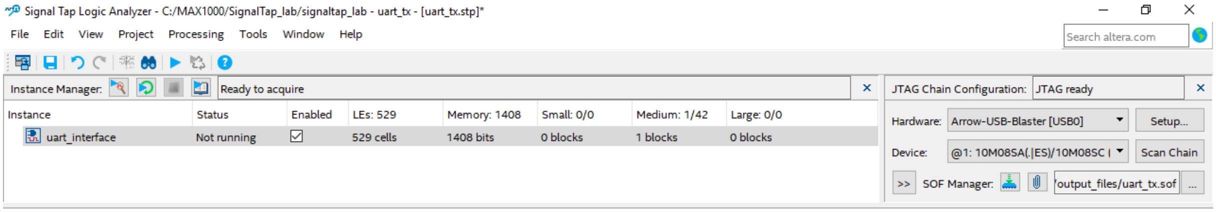
Device: @1: 10M08SA(.|ES)/10M08SC (

4.3.3.6 Click on ... button to choose the programming file.

4.3.3.7 Navigate to **<project_directory>/output_files/** and select the **uart_tx.sof** file.

4.3.3.8 Click **Open**.

4.3.3.9 Click on button to program the board. When the configuration is complete, the message box in the middle should write “Ready to acquire”.



Signal Tap Logic Analyzer - C:/MAX1000/SignalTap_lab/signaltap_lab - uart_tx - [uart_tx.stp]*

Instance Manager: Ready to acquire

Instance	Status	Enabled	LEs: 529	Memory: 1408	Small: 0/0	Medium: 1/42	Large: 0/0
uart_interface	Not running	☑	529 cells	1408 bits	0 blocks	1 blocks	0 blocks

JTAG Chain Configuration: JTAG ready

Hardware: Arrow-USB-Blaster [USB0]

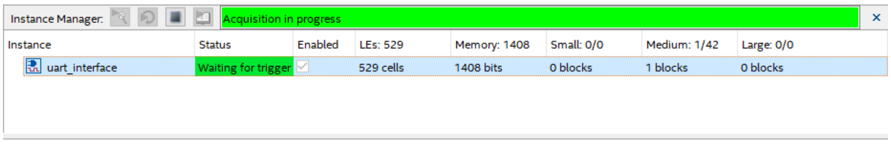
Device: @1: 10M08SA(.|ES)/10M08SC (

SOF Manager: output_files/uart_tx.sof

4.3.4 Analyze the design

4.3.4.1 Click on button to run analysis. When you pressed it, the status of uart_interface instance should change to “Waiting for trigger”.

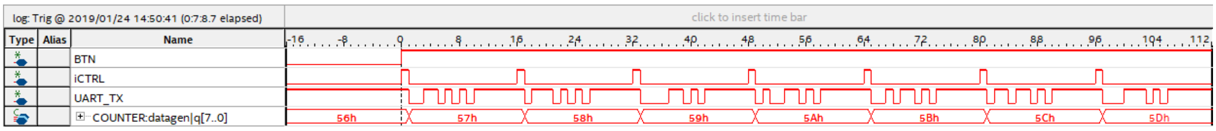
This analyzer type is a single acquisition, it runs only once after triggering.



Instance Manager: Acquisition in progress

Instance	Status	Enabled	LEs: 529	Memory: 1408	Small: 0/0	Medium: 1/42	Large: 0/0
uart_interface	Waiting for trigger	☑	529 cells	1408 bits	0 blocks	1 blocks	0 blocks

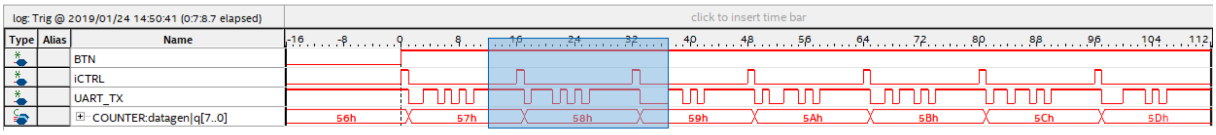
4.3.4.2 Press the user button on MAX1000 board to generate a trigger signal for the analyzer. The trigger is the rising edge of the BTN, so the analyzing will start when you release the button.



log: Trig @ 2019/01/24 14:50:41 (0:7:8.7 elapsed)

Type	Alias	Name
		BTN
		iCTRL
		UART_TX
		⊞ COUNTER:datagen|q[7..0]

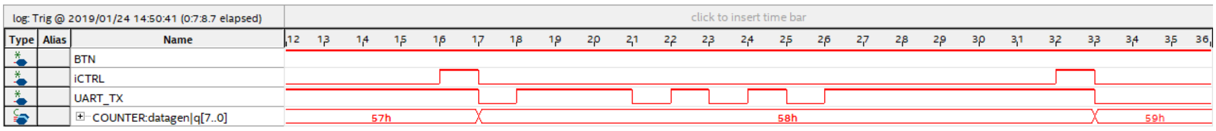
4.3.4.3 You can zoom into the waveform by selecting the section of the examination by mouse.



log: Trig @ 2019/01/24 14:50:41 (0:7:8.7 elapsed)

Type	Alias	Name
		BTN
		iCTRL
		UART_TX
		⊞ COUNTER:datagen|q[7..0]

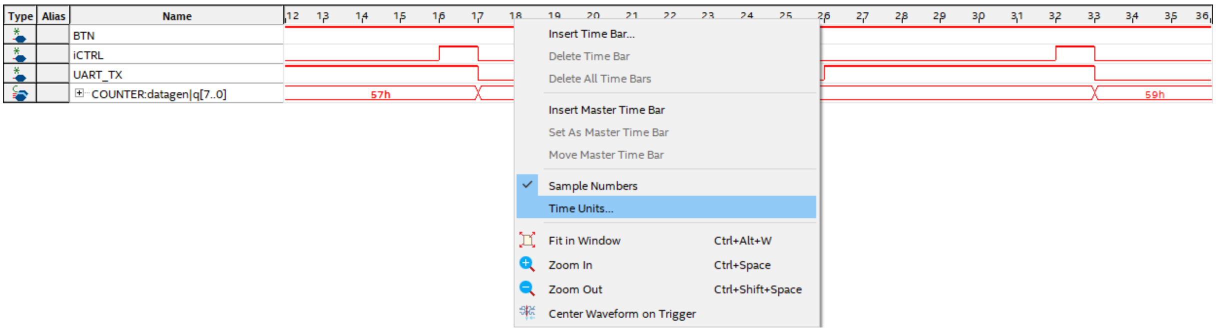
4.3.4.4 On the data screen you can see that the data is enabled on the rising edge of iCLK and it sends the data to the serial UART_TX output. The transmission signal contains a start bit, 8 bits data, starting with LSB, and one stop bit.



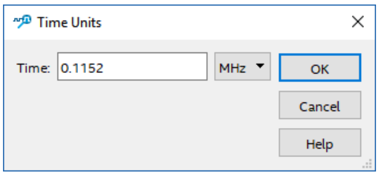
log: Trig @ 2019/01/24 14:50:41 (0:7:8.7 elapsed)

Type	Alias	Name
		BTN
		iCTRL
		UART_TX
		⊞ COUNTER:datagen|q[7..0]

4.3.4.5 On the timescale the default setup is the sample numbers, where each number represents a clock period. You can change it by right click on the timescale and select **Time Units…** from the pop-up menu.



4.3.4.6 In the Time Units window set the time to **0.1152 MHz** and press **OK**. After this process you can check the waveforms with horizontal µs scale.



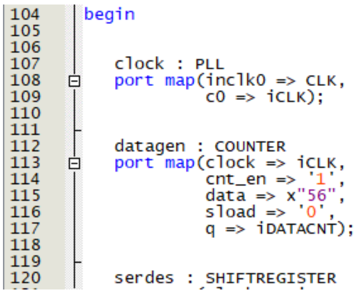
Time Units

Time: 0.1152 MHz

4.3.5 Design modification

4.3.5.1 In Quartus Prime search “**datagen : COUNTER”** component instantiation in the architecture section of uart_tx-vhd after the word ‘begin’. Modify the counter enable and synchronous load in the port map to:

```
cnt_en => ‘1’,
sload => ‘0’,
```



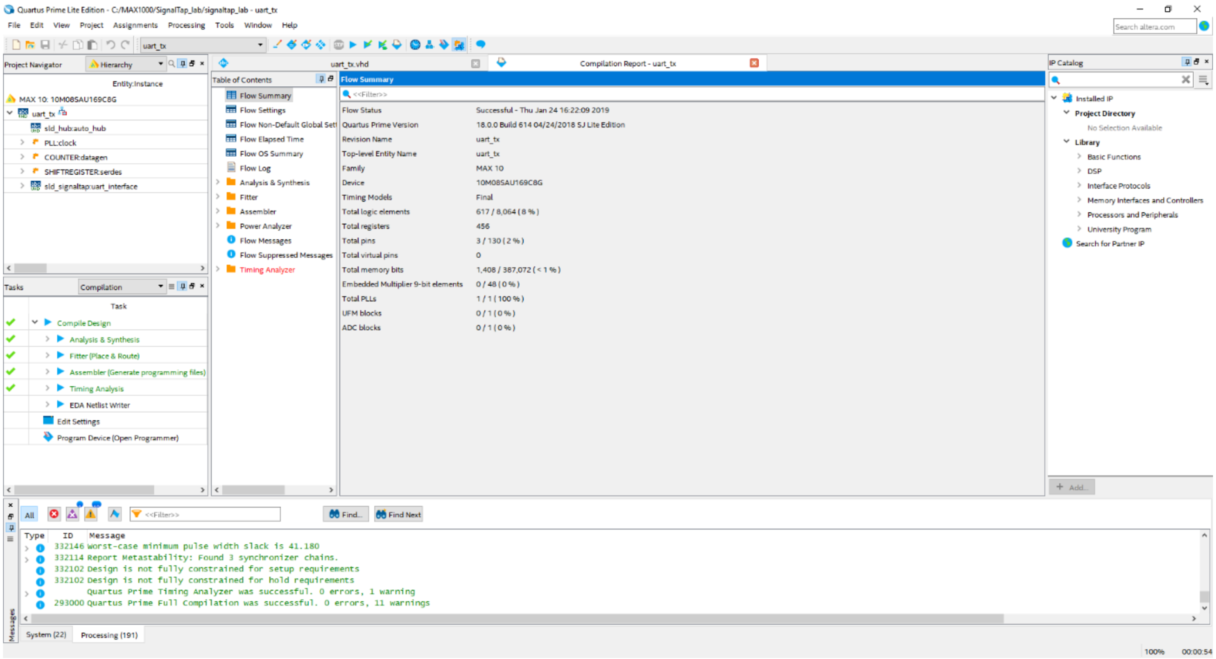
```
begin


   clock : PLL
   port map(inclk0 => CLK,
            c0 => iCLK);


   datagen : COUNTER
   port map(clock => iCLK,
            cnt_en => '1',
            data => x"56",
            sload => '0',
            q => iDATACNT);


   serdes : SHIFTREGISTER
```

4.3.5.2 Save the modification by clicking on 💾 button or select **File → Save** from the menu.

4.3.6 Recompilation and reconfiguration

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

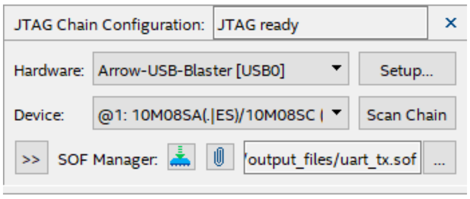
There should be no errors. If there are errors, they should be fixed before re-compiling. The 100% in the lower right corner or a green checkmark next to the Compile Design in the Compilation task window indicates that the compilation was successful.



Flow Summary	
Flow Status	Successful - Thu Jan 24 16:22:09 2019
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	uart_tx
Top-level Entity Name	uart_tx
Family	MAX 10
Device	10M08SAU169C8G
Timing Models	Final
Total logic elements	617 / 8,064 (8 %)
Total registers	456
Total pins	3 / 130 (2 %)
Total virtual pins	0
Total memory bits	1,408 / 387,072 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	1 / 1 (100 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

```
Type  ID     Message
      332146 Worst-case minimum pulse width slack is 41.180
      332114 Report Metastability: Found 3 synchronizer chains.
      332102 Design is not fully constrained for setup requirements
      332102 Design is not fully constrained for hold requirements
             Quartus Prime Timing Analyzer was successful. 0 errors, 1 warning
      293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings
```

4.3.6.2 Open the SignalTap window and make sure, that the hardware, device and the SOF manager have not changed.



JTAG Chain Configuration: JTAG ready

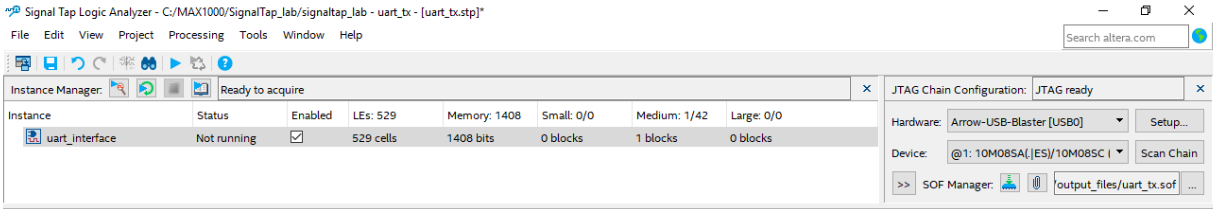
Hardware: Arrow-USB-Blaster [USB0]

Device: @1: 10M08SA(.|ES)/10M08SC (

SOF Manager: output_files/uart_tx.sof

4.3.6.3 Click on button to program the board. You do not need to browse again the configuration file, because after the reconfiguration it updates automatically.

When the configuration is complete, the message box in the middle should write “Ready to acquire”.



Signal Tap Logic Analyzer - C:/MAX1000/SignalTap_lab/signaltap_lab - uart_tx - [uart_tx.stp]*

Instance Manager: Ready to acquire

Instance	Status	Enabled	LEs: 529	Memory: 1408	Small: 0/0	Medium: 1/42	Large: 0/0
uart_interface	Not running	☑	529 cells	1408 bits	0 blocks	1 blocks	0 blocks

JTAG Chain Configuration: JTAG ready

Hardware: Arrow-USB-Blaster [USB0]

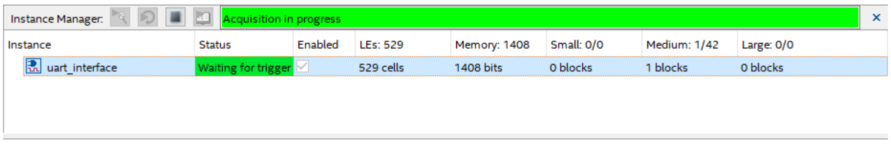
Device: @1: 10M08SA(.|ES)/10M08SC (

SOF Manager: output_files/uart_tx.sof

4.3.7 Analyze the design

4.3.7.1 Click on button to run analysis. When you pressed it, the status of uart_interface instance should change to “Waiting for trigger”.

This is the running mode of the analyzer, it runs until you stop it and refresh the waveform on every triggering.

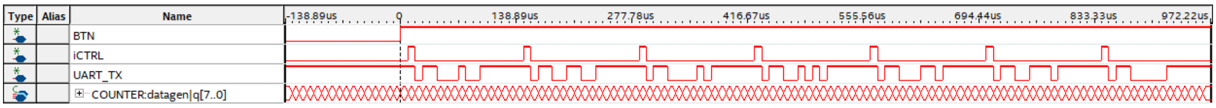


Instance Manager: Acquisition in progress

Instance	Status	Enabled	LEs: 529	Memory: 1408	Small: 0/0	Medium: 1/42	Large: 0/0
uart_interface	Waiting for trigger	☑	529 cells	1408 bits	0 blocks	1 blocks	0 blocks

4.3.7.2 Press the user button on MAX1000 board to generate a trigger signal for the analyzer. The trigger is the rising edge of the BTN, so the analyzing will start when you release the button.

Please note, because of the free running, you could have different data on the counter and accordingly, the UART signal may differ.



4.3.7.3 Zoom into the waveform and press the user button on MAX1000 board several times to see, how the autorun analysis work.

4.3.7.4 Click on button to stop the analysis.

CONGRATULATIONS! YOU HAVE SUCCESSFULLY COMPLETED THE SIGNALTAP LAB!

5 Revision History

Version	Change Log	Date of Change
V1.0	Initial Version	24/01/2019

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.