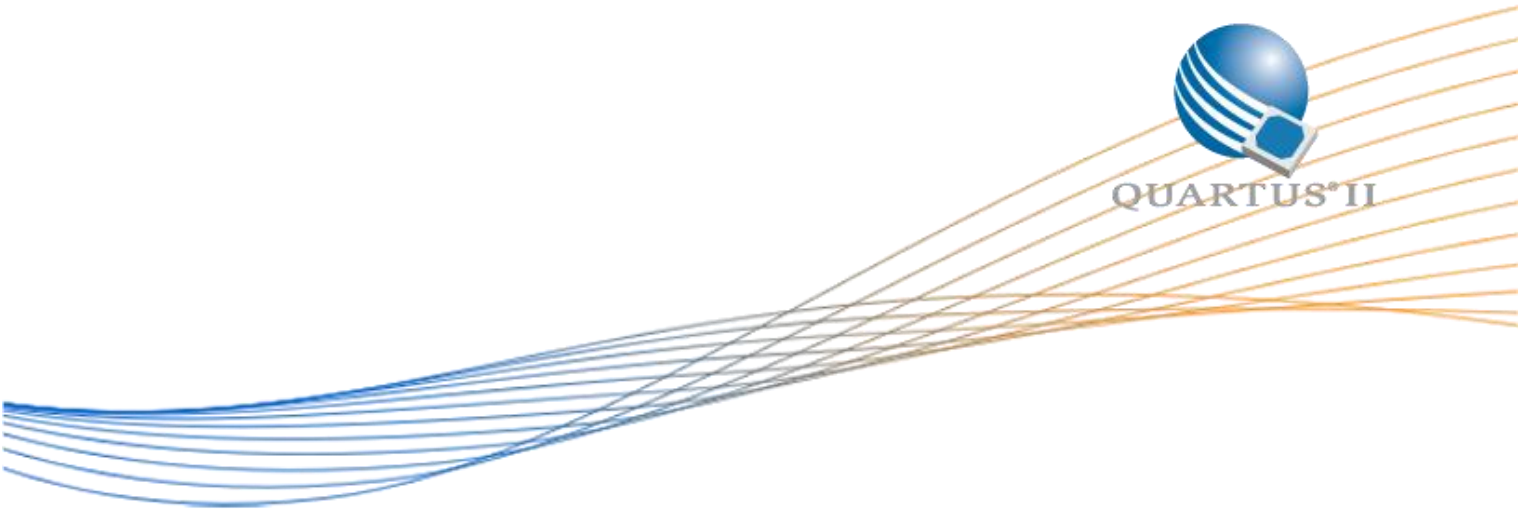


MAX 10 DDR3 x24 with EMIF Debug Feature

Date: 1 September 2015

Revision: 1.0

Table of Contents

List of Figure

List of Table

System Requirement

This example design targets the Max 10 FPGA Development Kit. This design is a 24-bit wide, 300-MHz DDR3 SDRAM interface working with a Max 10 FPGA with its own debug feature.

This design will be using the UniPHY IP for Max10 and an example traffic generator. All these will be used to demonstrate the DDR3 SDRAM functionality.

Design Specifications

The design will utilize the external memory port on the Max 10 Development Kit. We will be utilizing the DDR3 module that is on the development kit. The figure highlights those modules which we will be using on this design.

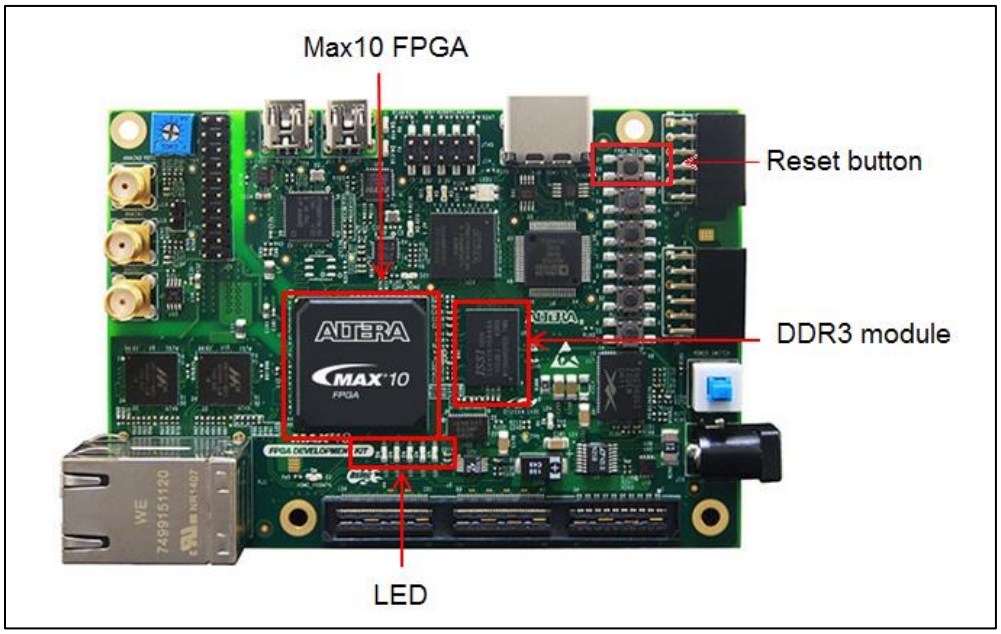


Figure 1: Max 10 development kit connection used for DDR3 example design

The table below lists the specifications for this design:

Attribute	Specification
Quartus version	QuartusII v15.0
FPGA	10M50DAF484C6GES
Kit	Development Kit
Memory speed	300MHz
Memory topology	X24-bit, 4 DDR3 SDRAM components
IP used	DDR3 SDRAM Controller II with UniPHY IP and generated top Quartus project

Table 1: Specification of design

Pin Assignment and Description

Pin	Location	I/O Standard	Description
a[12]	J14	1.5V	Address & Command pin
a[11]	E20	1.5V	Address & Command pin
a[10]	Y20	1.5V	Address & Command pin
a[9]	E22	1.5V	Address & Command pin
a[8]	D22	1.5V	Address & Command pin
a[6]	E21	1.5V	Address & Command pin
a[5]	F19	1.5V	Address & Command pin
a[3]	U20	1.5V	Address & Command pin
a[0]	V20	1.5V	Address & Command pin
a[7]	B20	1.5V	Address & Command pin
a[4]	C20	1.5V	Address & Command pin
a[2]	A21	1.5V	Address & Command pin
a[1]	D19	1.5V	Address & Command pin
ba[2]	W22	1.5V	Address & Command pin
ba[1]	N18	1.5V	Address & Command pin
ba[0]	V22	1.5V	Address & Command pin
cas_n[0]	U19	1.5V	Address & Command pin
ck[0]	D18	1.5V	pll reference clock pin
ck_n[0]	E18	1.5V	pll reference clock pin
cke[0]	W20	1.5V	Address & Command pin
cs_n[0]	Y22	1.5V	Address & Command pin
dm[2]	T18	1.5V	DM pin
dm[1]	N19	1.5V	DM pin
dm[0]	J15	1.5V	DM pin
dq[23]	P20	1.5V	DQ pin
dq[22]	P15	1.5V	DQ pin
dq[21]	T19	1.5V	DQ pin
dq[20]	R15	1.5V	DQ pin
dq[19]	R20	1.5V	DQ pin
dq[18]	P14	1.5V	DQ pin
dq[17]	P19	1.5V	DQ pin
dq[16]	R14	1.5V	DQ pin
dq[15]	N20	1.5V	DQ pin

Table 2: Pin Assignment for Example Design

Pin	Location	I/O Standard	Description
dq[14]	L19	1.5V	DQ pin
dq[13]	M15	1.5V	DQ pin
dq[12]	L18	1.5V	DQ pin
dq[11]	M14	1.5V	DQ pin
dq[10]	M20	1.5V	DQ pin
dq[9]	M18	1.5V	DQ pin
dq[8]	L20	1.5V	DQ pin
dq[7]	K19	1.5V	DQ pin
dq[6]	H20	1.5V	DQ pin
dq[5]	J20	1.5V	DQ pin
dq[4]	H19	1.5V	DQ pin
dq[3]	K18	1.5V	DQ pin
dq[2]	H18	1.5V	DQ pin
dq[1]	K20	1.5V	DQ pin
dq[0]	J18	1.5V	DQ pin
dqs[2]	R18	1.5V	DQS pin
dqs[1]	L14	1.5V	DQS pin
dqs[0]	K14	1.5V	DQS pin
odt[0]	W19	1.5V	ODT pin
ras_n[0]	V18	1.5V	Address & Command pin
we_n[0]	Y21	1.5V	Address & Command pin
reset_n	B22	1.5V	global reset pin
generator_0_status_fail	U22	1.5V	status pin
generator_0_status_pass	AA2	1.5V	status pin
generator_0_status_test_complete	AA2	1.5V	status pin

Table 2: Pin Assignment for Example Design

Design Hardware Test

1. Download the MAX 10 design on the design store and install the design templates.

- In the Quartus II software, create a Quartus II project using the **New Project Wizard** available from the File menu.
- Select **Project template** on the **Project Type** page

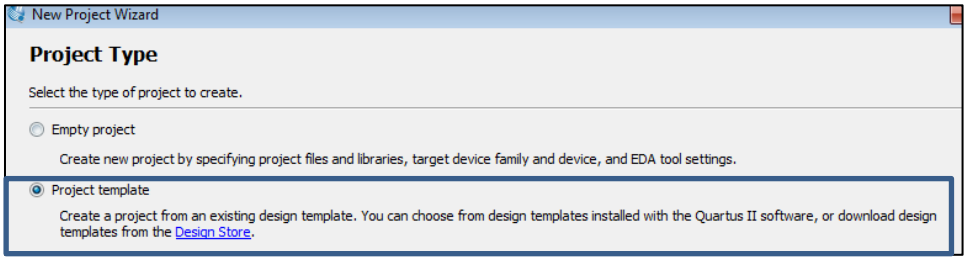


Figure 2: Project Type Selection

- Select **Install the design templates** on the **Design Templates** page

Figure 3: Install Templates

- On the pop-up window, select the **myplatform.par** file in the directory where you store the MAX 10 design from the design store and click **OK**. This will include the design template into your Quartus II.

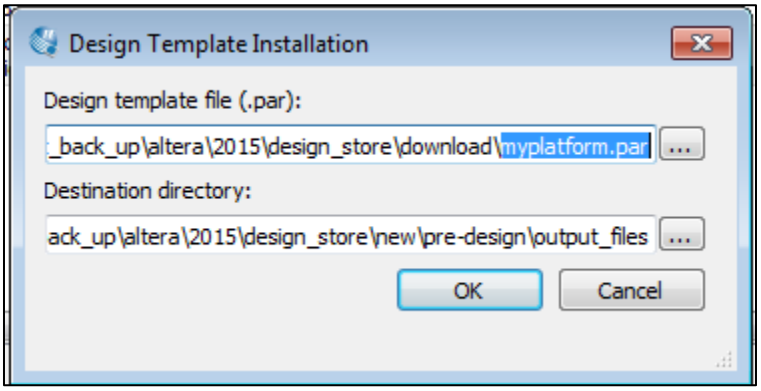


Figure 4: Locating design template

- Select the right design on the Available design templates window, and the latest design template is located at the lowest row.

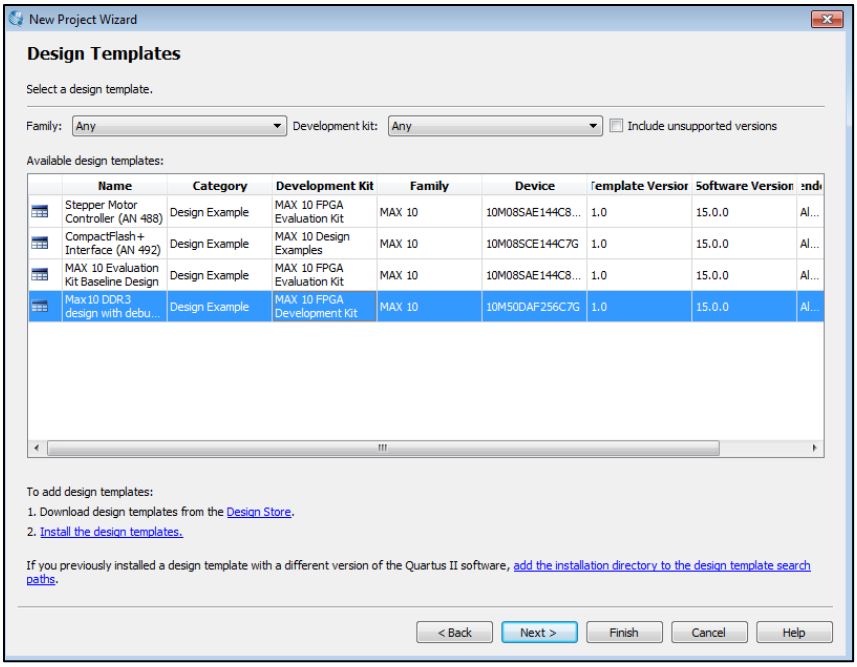


Figure 5: Selecting Design Template

- Select Next on the Design Template page and then Finish on the Summary page

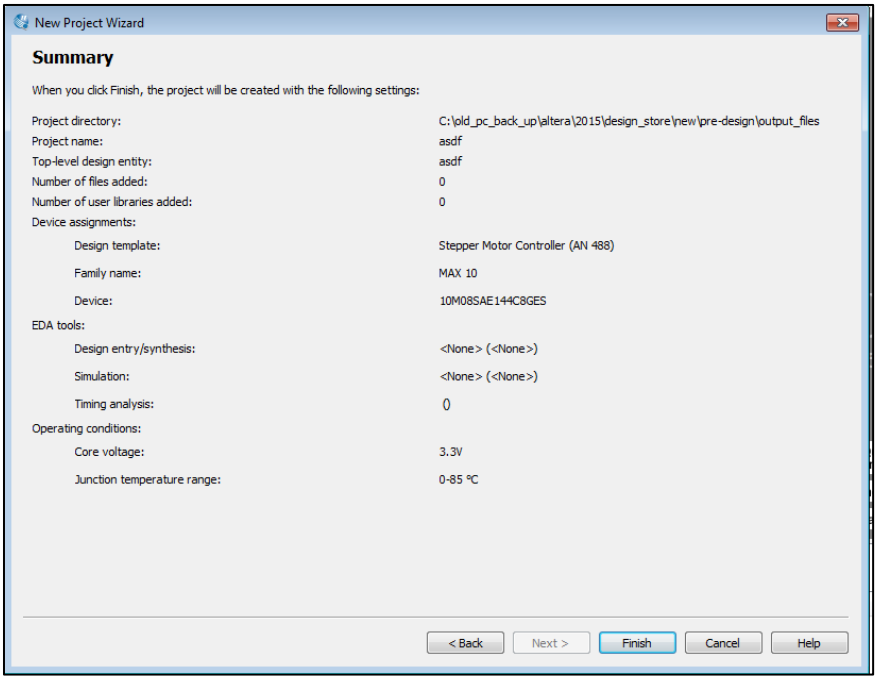


Figure 6: Summary of New Design

2. Open the design templatesIn the Quartus II software, launch **SignalTap II Logic** from **Tools** menu.
3. In **Jtag Chain Configuration** window under **SignalTap Logic** GUI, configure the **Hardware** and **Device** based on the targeted board. For **File**, browse for the **m10.sof** file (located in <design_directory>/platform/output_files folder) and click **Open**.

4. Click **Program Device** button  to configure the FPGA.

5. Select the SignalTap instance and click the **Autorun Analysis** button  next to **Instance Manager** label. The SignalTap II Logic Analyzer Pane will shows the acquired data from each signal in the **Data** tab.

6. On the development kit, push once and release the reset button on the development kit to reset the design.

7. Now observe the data for **local_cal_success** and **traffic_gen_pass** and **traffice_gen_timeout** instances change from 0 to 1 on the SignalTap which indicate the calibration is success and pass the example driver tests. Also the LED1 on the board will light up and turn to green colour.

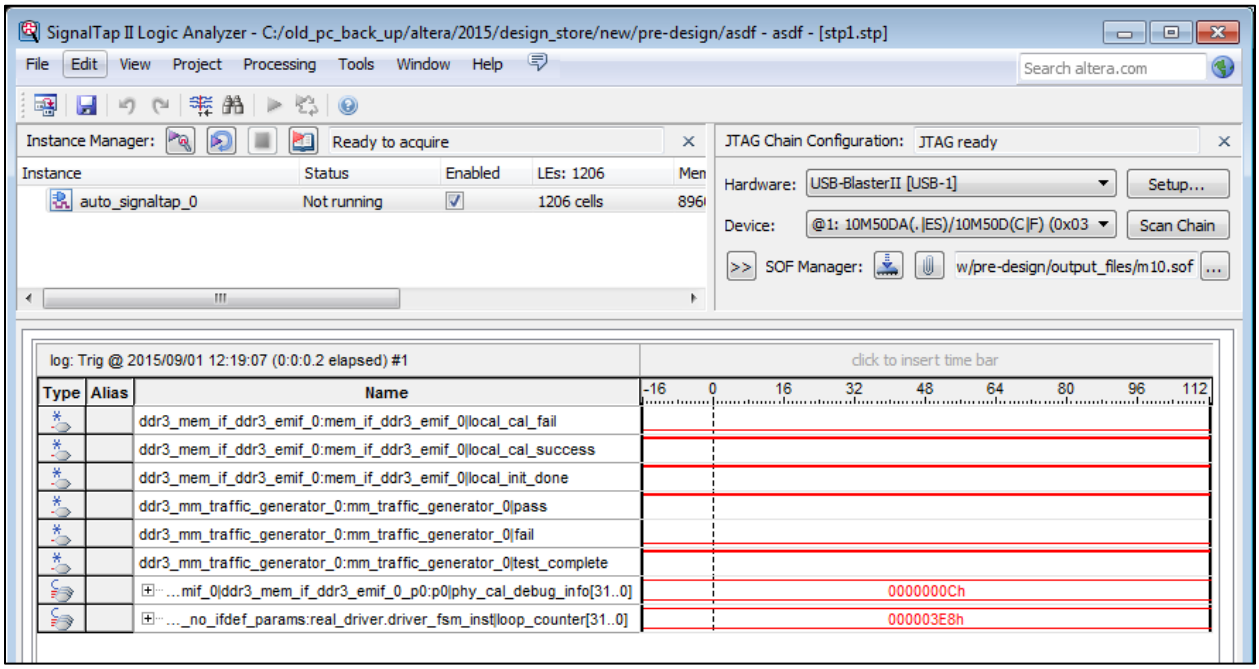


Figure 7: SignalTap Capture

External Memory Interface Debug Features

1. To understand the calibration margin of the design you can use the **SignalTap Logic** from **Tools** menu to monitor the ***phy_cal_debug_info*** signal.

2. The signal contain 32bit of data, with the first 7 bit is the margin. The higher the value, the bigger the margin on your design. The next 24bit is the result of the read and write data comparison. On a 24bit design, each bit will represent the DQ pin read and write data comparison. The accurate data will be shown as zero, while wrong data is shown as one.

3. The phy_cal_debug_info signal is translated into the best comparison result and margin with the following:

- phy_cal_debug_info[31:24] >> best_comp_result[23:16]
- phy_cal_debug_info[23:16] >> best_comp_result[15:8]
- phy_cal_debug_info[15:8] >> best_comp_result[7:0]
- phy_cal_debug_info[7:0] >> margin[7:0]

4. For **passing** interface the **best comparison** result is **zero** and the **margin** is **non-zero** while for **failing** interface the **best comparison** result is **non zero**, **margin** is **zero**. Examples for the *phy_cal_debug_info* signal are shown below:

Debug	best_comp_result[23:16]	best_comp_result[15:8]	best_comp_result[7:0]	margin[7:0]
PASS	0000 0000	0000 0000	0000 0000	0001 0000
Info	Pass	Pass	Pass	16 steps margin

Debug	best_comp_result[23:16]	best_comp_result[15:8]	best_comp_result[7:0]	margin[7:0]
FAIL	0001 0000	1111 1111	0000 0010	0000 0000
Info	Fail	Fail	Fail	No window found

Table 3: Max10 debug feature

Tutorial Steps

In this Tutorial, you will learn on how to recreate this full design using Quartus II v15.0. The lab assumes the reader is a competent user of these tools and many of their features.

Two files have been pre-designed for this lab to save time:
1. A pin location assignments TCL script (m10_ddr3_pin_locations.tcl)
2. A top level QSYS file (ddr3.qsys)
3. A sdc constraint file to ensure signals in SignalTap II is not considered for timing analysis (ddr3_sdc.sdc)

Create a new folder for the project and place these files in it.

Design Generation

1. In the Quartus II software, create a Quartus II project using the **New Project Wizard** available from the File menu.

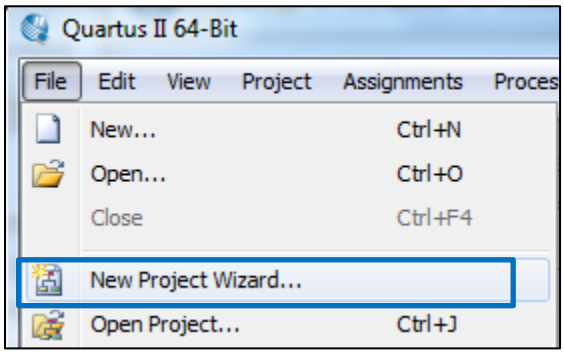


Figure 8: New Project Wizard

For this lab, use following information to setup the project accordingly:

- Working directory : < your project folder >
- Project name : <variation_name>
- Device name : 10M50DAF484C6GES
- Leave other settings to default

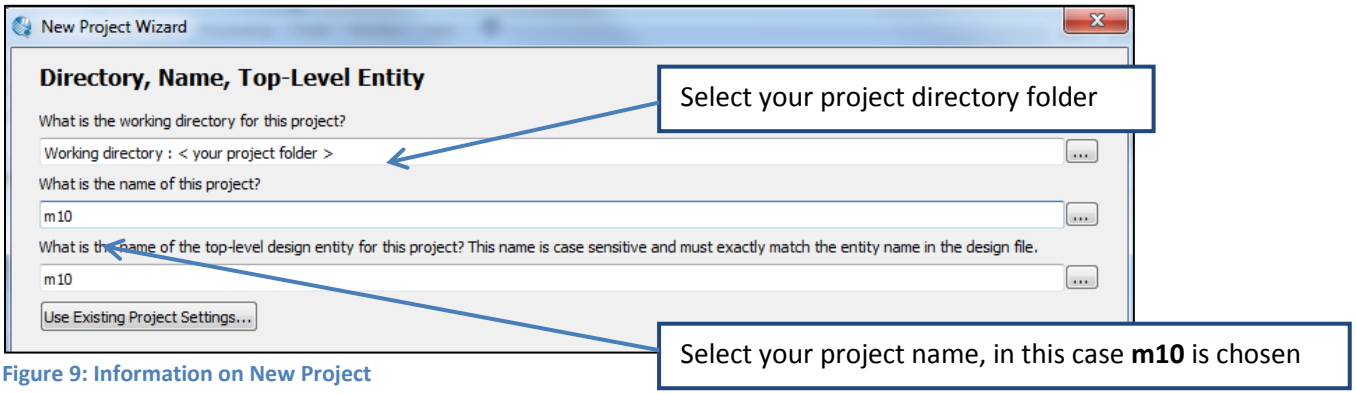


Figure 9: Information on New Project

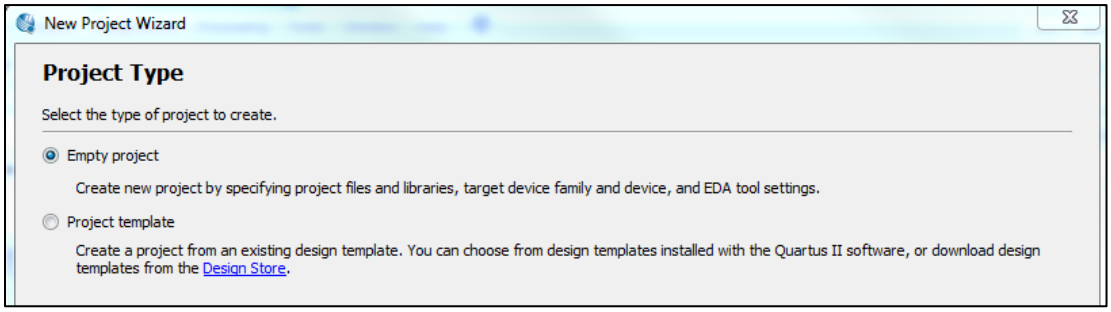


Figure 10: Empty New Project

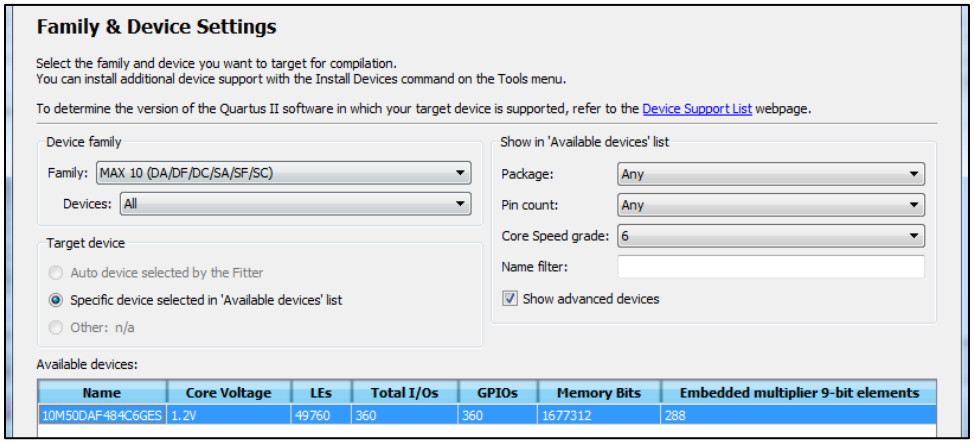


Figure 11: Selecting Device

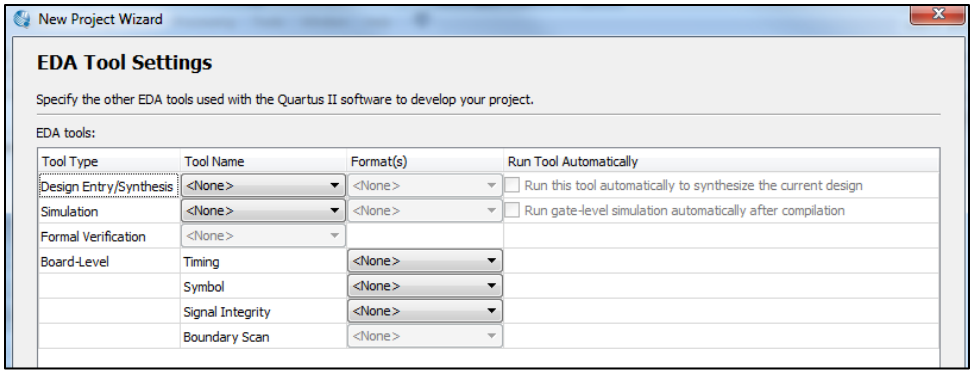


Figure 12: EDA Tool Settings

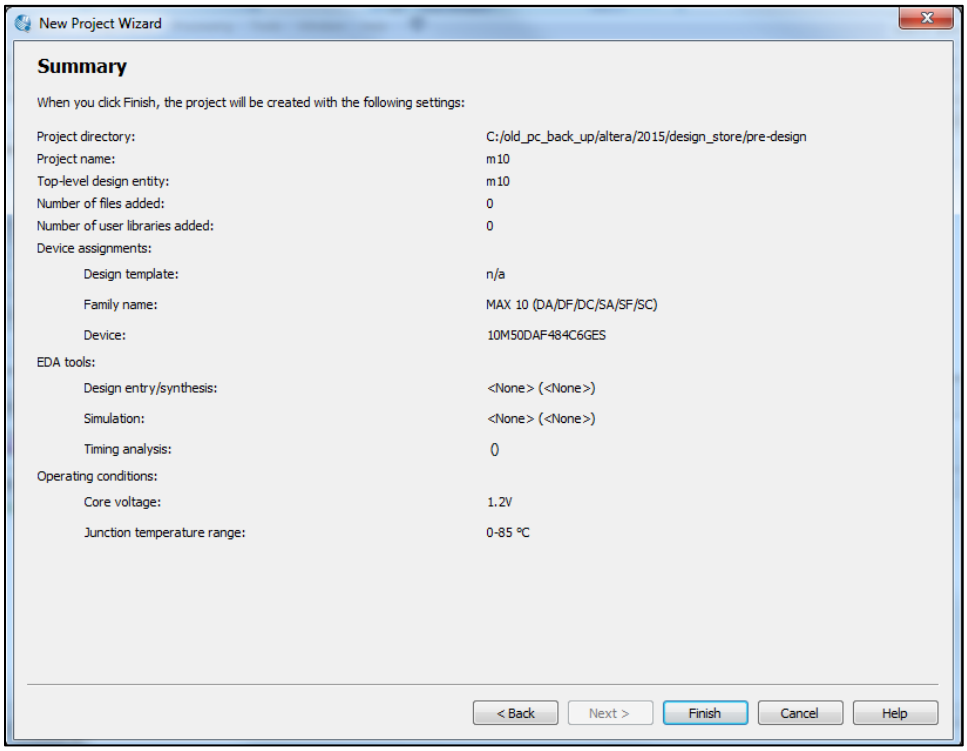


Figure 13: Summary of New Design

2. Launch the **Qsys** from the **Tools** menu and you can select the **ddr3.qsys** file from the pre-design folder and click Open. Or click the Cancel button if you want to generate the design from scratch using the steps below.

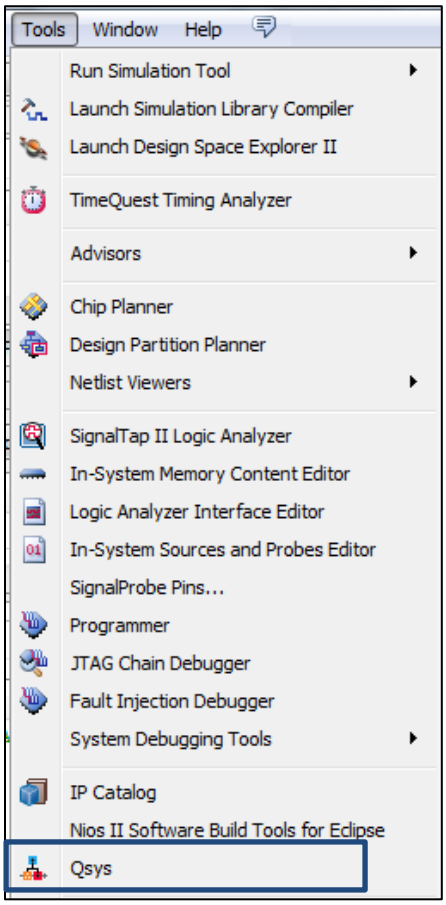


Figure 14: Launching Qsys

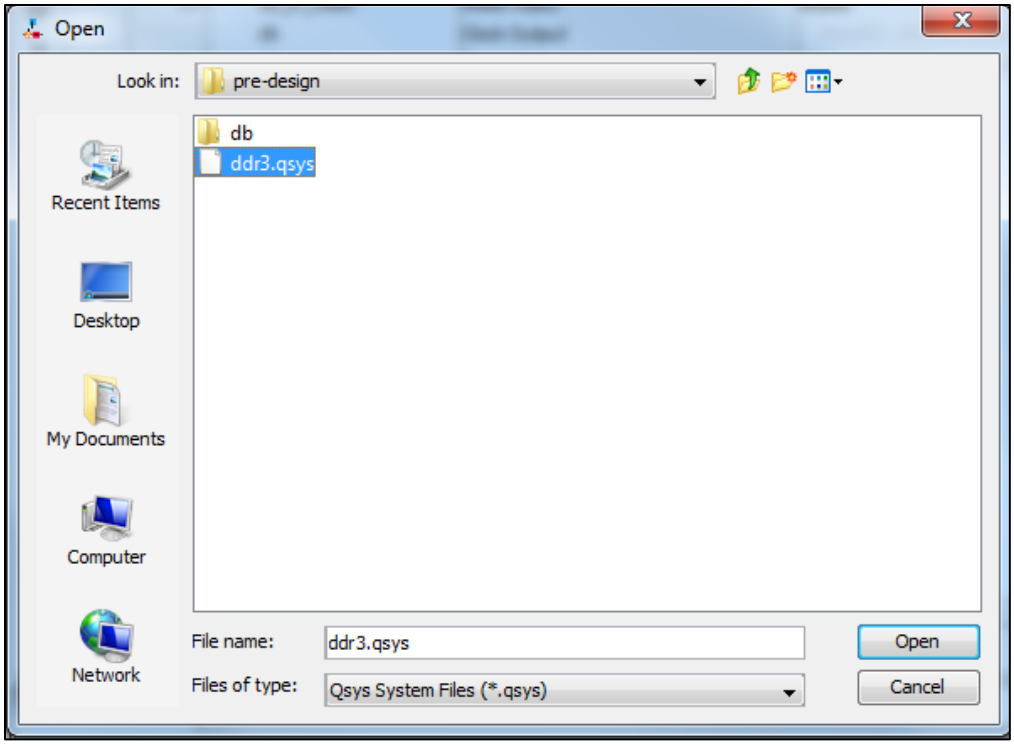


Figure 15: Open pre-designed Qsys file

3. Double click **DDR3 SDRAM Controller with UniPHY** IP from the **Memory Interfaces and Controllers > Memory Interfaces with UniPHY** folder in the **Library** list.

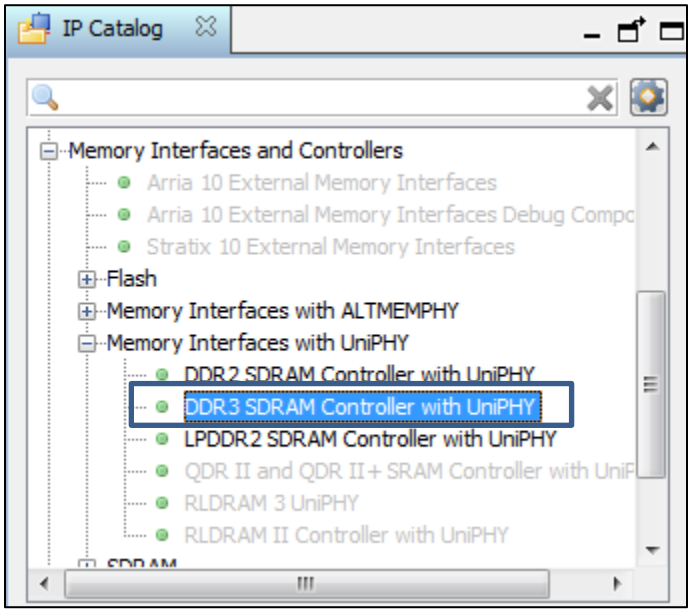


Figure 16: DDR3 Controller

4. Set parameters for Memory Controller with UniHY

- **PHY Settings Tab**

i. Set Speed grade to 6.

ii. Set Memory clock frequency to 300 MHz

iii. Set PLL reference clock frequency to 100 MHz.

iv. Select **Half** for half-rate Avalon-MM interface.

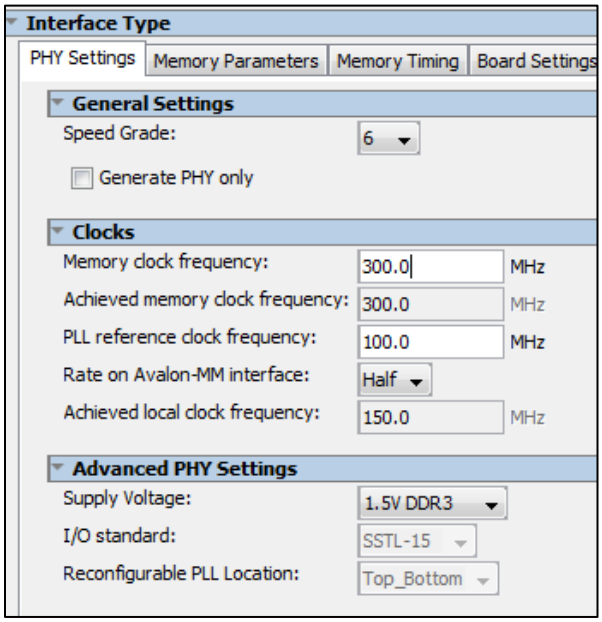


Figure 17: PHY Settings

- **Memory Parameters Tab**

  i. Select Discrete Device for Memory format.

  ii. Select 800 MHz for Memory device speed grade.

  iii. Type 24 for Total interface width.

  iv. Type 13 for Row address width.

  v. Type 10 for Column address width.

  vi. Type 3 for Bank address width.

  vii. Select 5 for Memory CAS latency setting under Memory Initialization Options,

  viii. Select RZQ/4 for Output drive strength setting.

  ix. Select 5 for Memory write CAS latency setting.

  x. Select Dynamic ODT off for Dynamic ODT(Rtt_WR) setting.

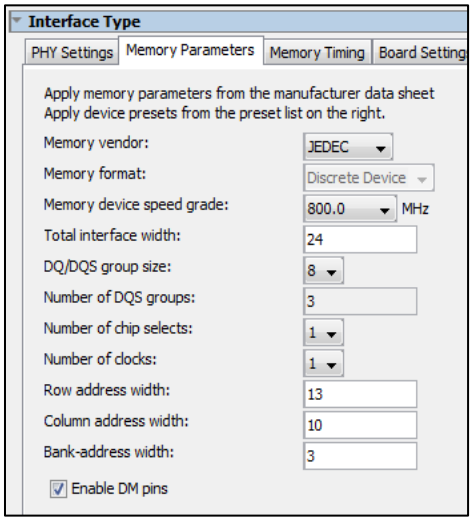


Figure 18: Memory Parameters Settings

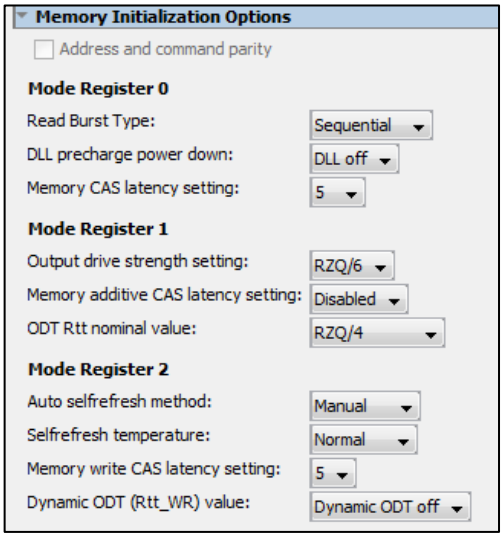


Figure 19: Memory Initialization Options

- **Memory Parameters Tab**

  Do the following changes to the below timing parameter and kept the rest as the default value.

  i. Set tIS (base) to 170ps
  ii. Set tIH (base) to 120ps
  iii. Set tDS(base) to 10ps
  iv. Set tDH(base) to 45ps
  v. Set tDQSQ to 100ps
  vi. Set tDQSCK to 225ps
  vii. Set tDQSS to 0.27 cycles
  viii. Set tDSH to 0.18 cycles
  ix. Set tDSS to 0.18 cycles
  x. Set tINIT to 500us
  xi. Set tMRD to 4cycles
  xii. Set tRAS to 35.0ns
  xiii. Set tRCD to 13.75ns
  xiv. Set tRP to 13.75ns
  xv. Set tREFI to 7.8us
  xvi. Set tRFC to 110ns
  xvii. Set tWTR to 6 cycles
  xviii. Set tFAW to 30.0ns

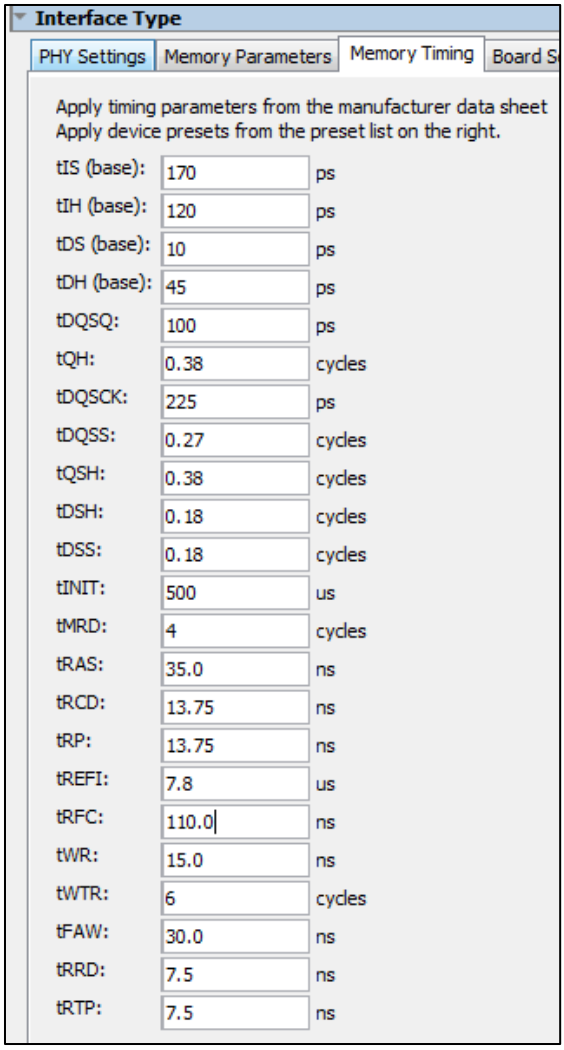


Figure 20: Memory Timing Settings

- **Board Settings Tab**

  Users should do board simulation for proper values in this page:

  i. In the Board Settings tab, set the slew rate parameters to **Select Use Altera's default settings**

  ii. Set the Intersymbol Interference/Crosstalk parameters to **Select Use Altera's default settings**

  iii. Set the Board Skews parameters to the below value **Select Use Altera's default settings**

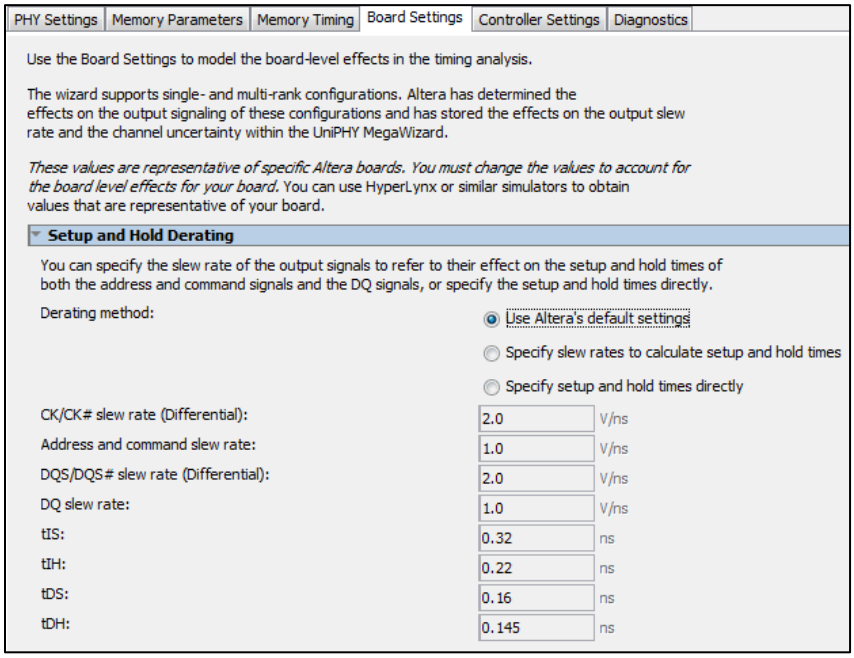


Figure 21: Setup and Hold Derating

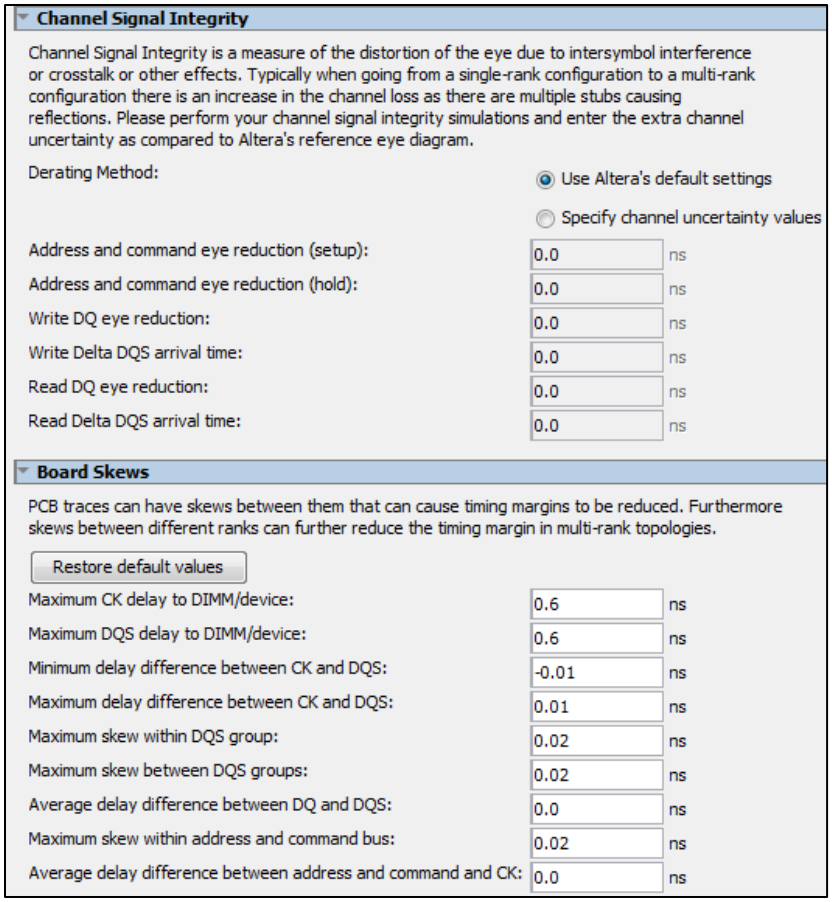


Figure 22: Channel Signal Integrity and Board Skew

5. Once all the changes is done, click on the **Finish** button.

6. Double click **Avalon-MM Traffic Generator and BIST Engine** IP from the **Basic Functions> Simulation; Debug and Verification>Verification** folder in the **Library** list

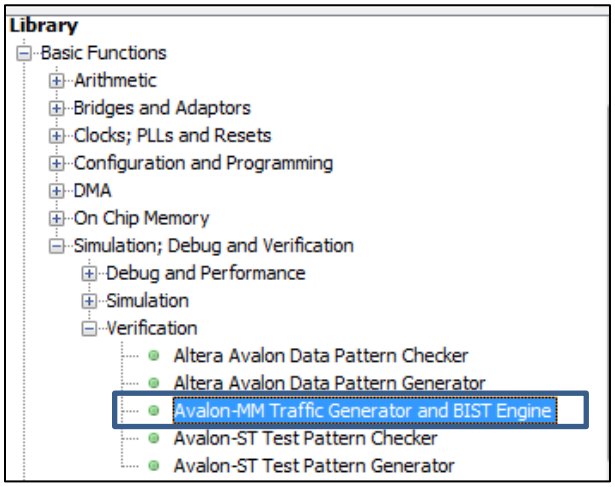


Figure 23: Traffic Generator

- Match the Avalon data and address with on the traffic generator module with those on the ddr3 module and click on the **Finish** button.
    i. Set the **Avalon Data Width** to 96
    ii. Set the **Avalon Address Width** to 24

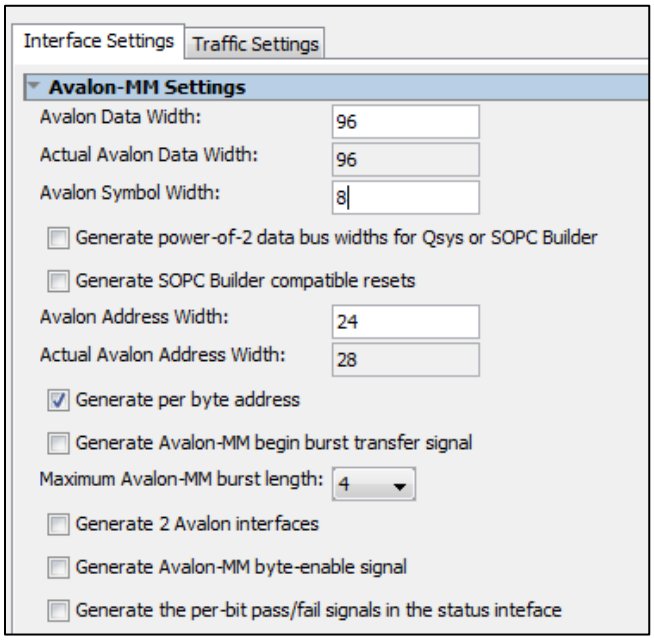


Figure 24: Traffic Generator Settings

7. Export the following signals

- The **status** conduit for both the traffic generator and DDR3 SDRAM UniPHY IP.

8. Connect the following signals:

- Avalon signal of the traffic generator to the Avalon signal of the DDR3 SDRAM UniPHY IP
- Afi_clk signal of the DDR3 SDRAM UniPHY IP to the avl_clk signal of the traffic generator
- Afi_reset signal of the DDR3 SDRAM UniPHY IP to the avl_reset signal of the traffic generator
- Clk signal from the clock module to the pll_ref_clk of the DDR3 SDRAM UniPHY IP
- Clk_reset signal from the clock module to the global_reset and soft_reset of the DDR3 SDRAM UniPHY IP

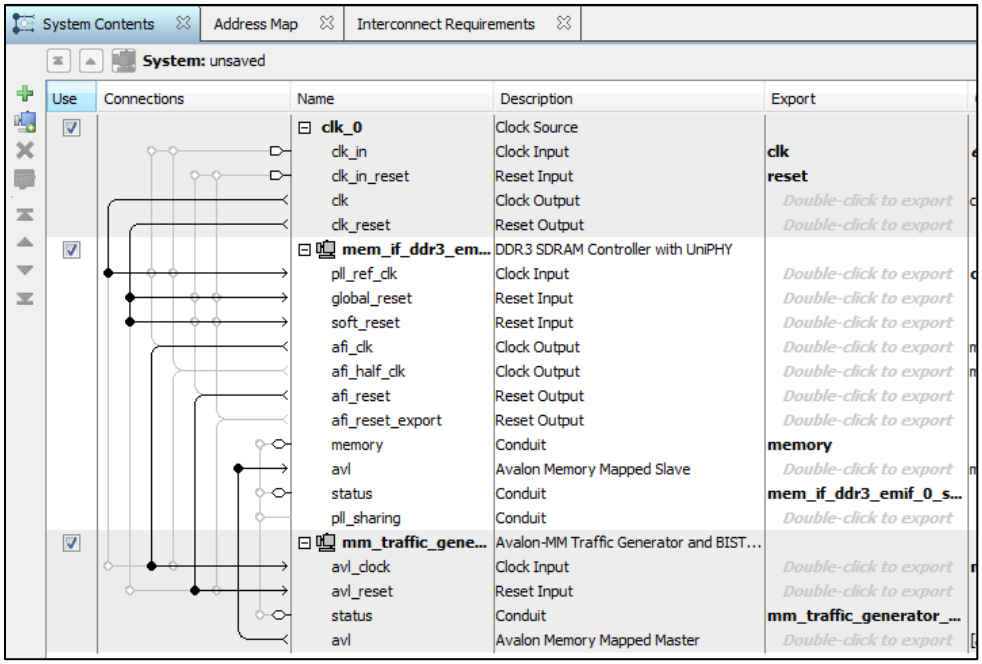


Figure 25: Qsys Connection

Important Note : Review any warning messages displayed in the Messages Window and correct any errors before making further changes

Important Note : Instead of leaving it to default, ensure the parameters in **Board Timing** tab are configured correctly based on the actual target board as the value are vary from board to board. Use HyperLynx or similar simulator to obtain values of the actual target board.

Important Note : Take note on the info messages regarding which address/command pin placement scheme that need to follow based on the final IP setting. This info will be needed during pin assignment in the later stage.

Note : For **Board and Package Skews**, use Board Skew Parameter Tool available in Altera web to compute the value

Note : For detailed explanation of the parameters, refer to Parameterizing Memory Controllers with UniPHY IP chapter of the External Memory Interface Handbook.

9. Click **Save** from the **file** menu.

- Pop up window will appears to let you choose the location to save this IP file. Please select the folder you created above.
- Specify the **Entity name** and click **Save**

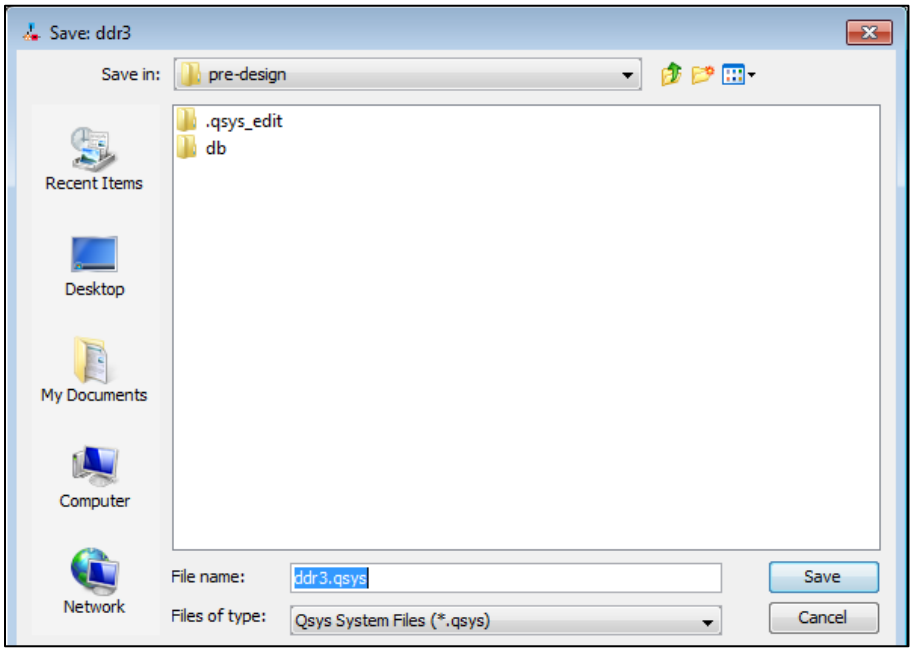


Figure 26: Save Qsys file

9. Click **Finish**. The configuration is saved as *ddr3.qsys* which located inside *<your project folder>* directory.

10. When prompt to generate the IP, click **Yes**.

- Pop up window will appears to let you choose the type of HDL design for synthesis and simulation. Chose **Verilog** for the Synthesis section and **None** for the Simulation section.
- Ensure you are using the right directory and click on **Generate**

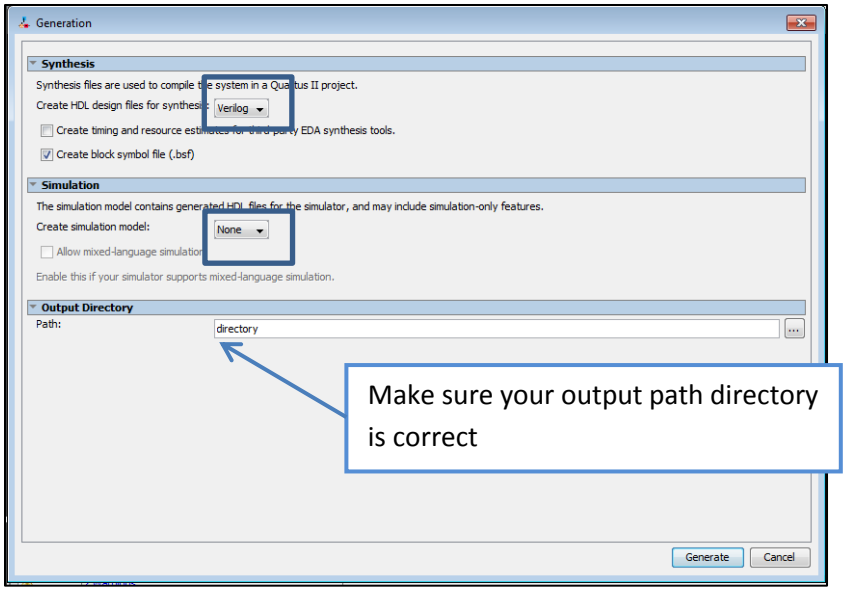


Figure 27: Generating Synthesis Design

11. In **Integration with the Quartus II Software** window, click **Finish**. Pop up window will appear to advice on including the .qip and .sip file into the design. Click **OK** on the window.

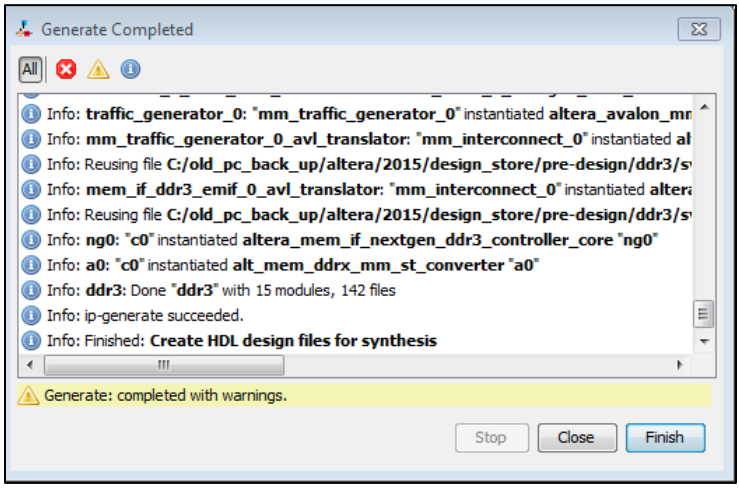


Figure 28: Design Generation Complete

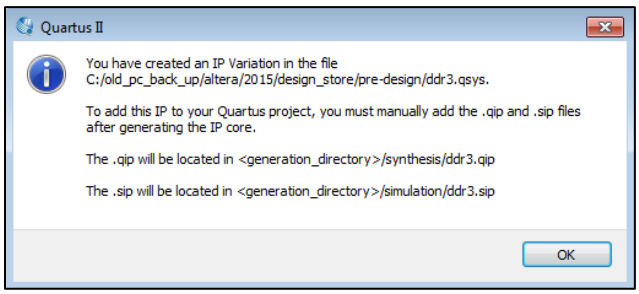


Figure 29: Reminder to include QIP and SIP file

Design Constraint, Compilation and Analysis

1. Open the top level file of **ddr3.qip** (in the *<design_directory>/ddr3/synthesis/* folder) and click on **Set as Top Level Entity** using the **project** menu.

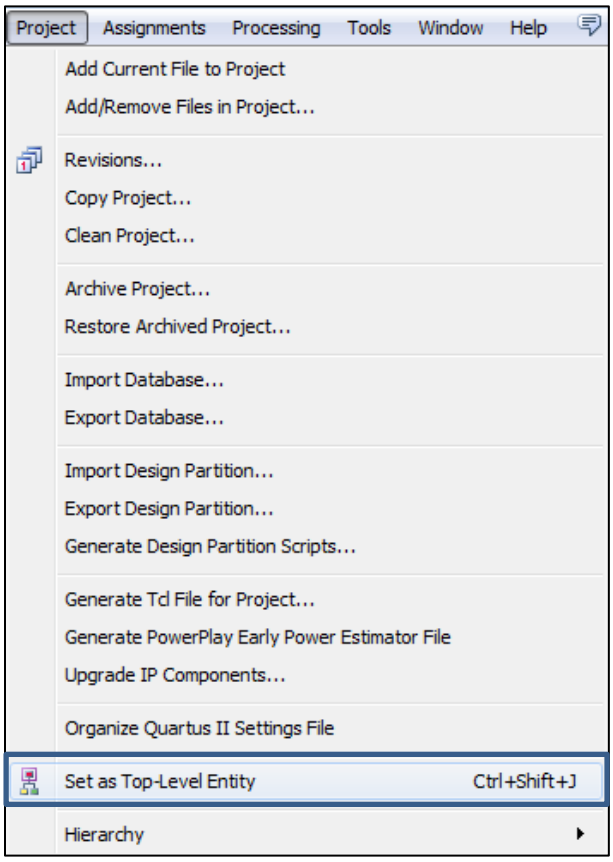


Figure 30: Set Top Level Entity

3. Run synthesis by clicking on the **Start Analysis and Synthesis** under the **Processing** menu. This is required so Quartus can determine the names of the external ports connected to the UniPHY for when the I/O assignments are created in the next step.

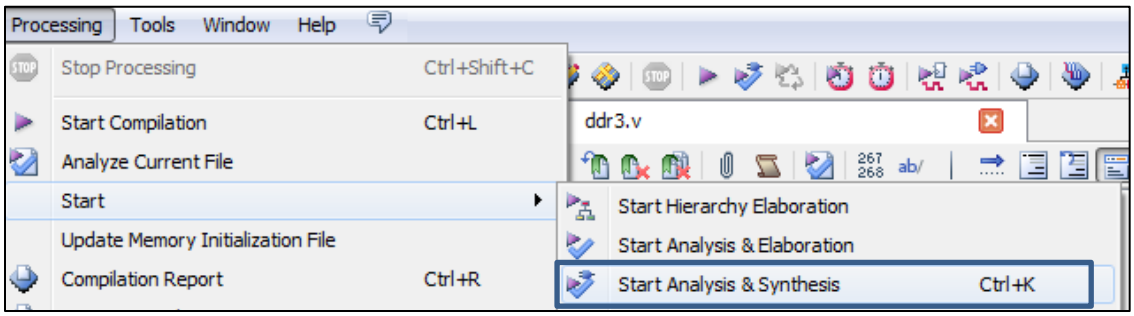


Figure 31: Analysis and Synthesis

4. Run tcl script generated by the UniPHY IP by clicking on the **Tcl Scripts** under the **Tools** menu. Select the tcl script ***<variation_name>_pin_assignments.tcl*** and click on **Run** to assign the pin and DQ group assignments.

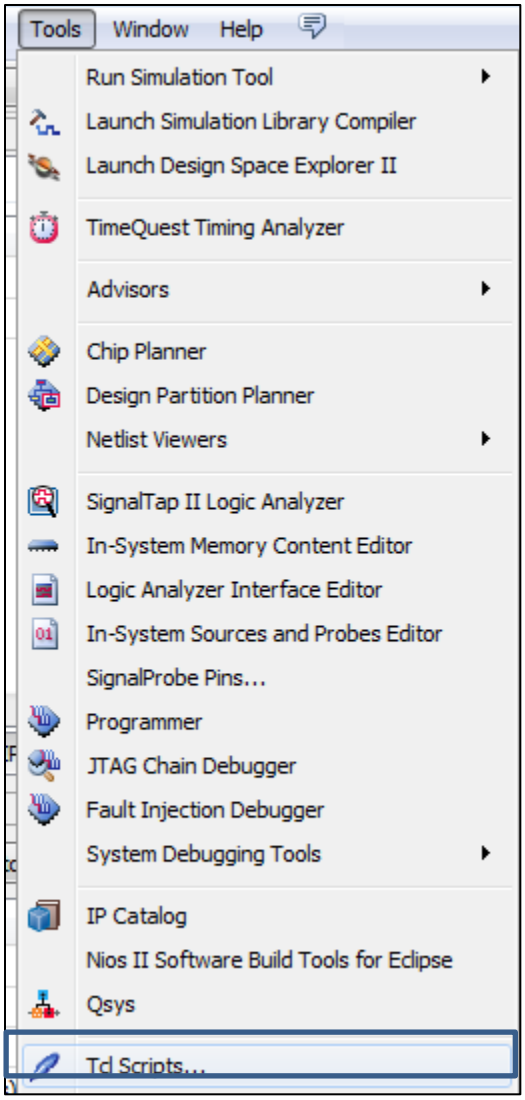


Figure 32: Tcl Scripts

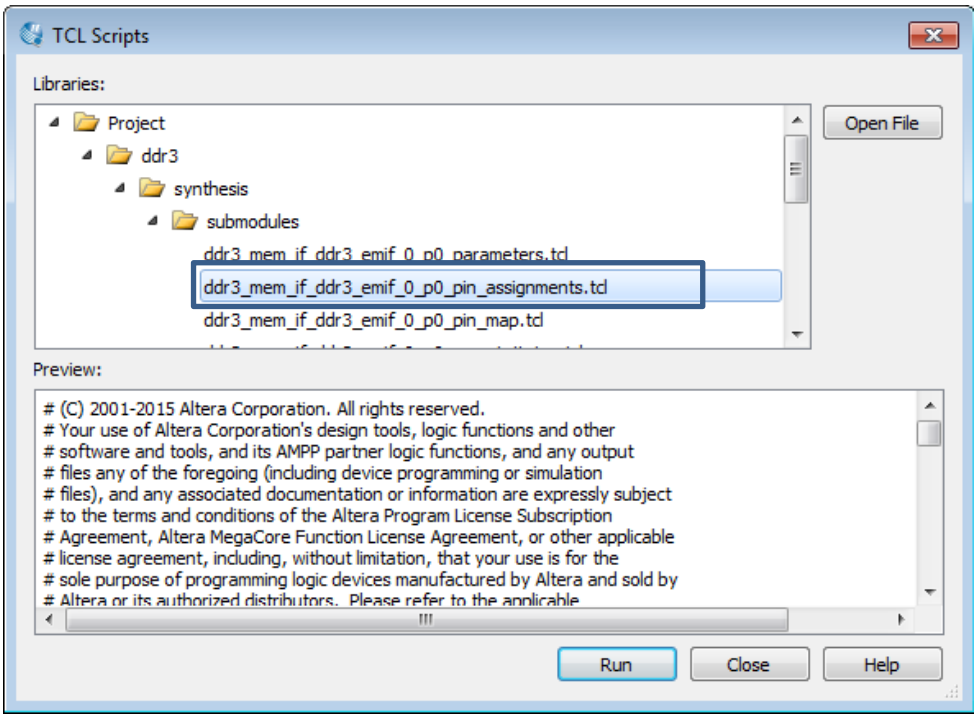
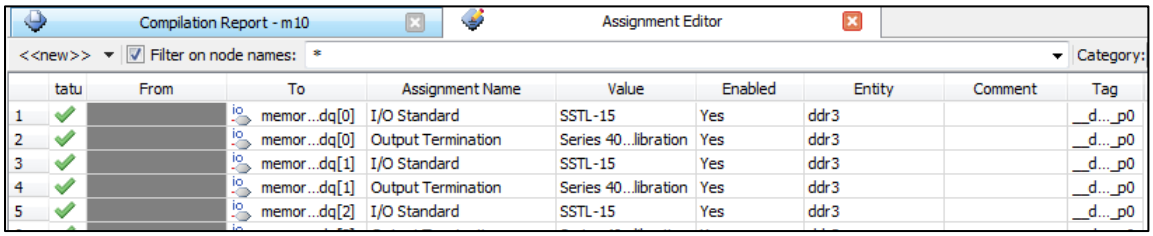


Figure 33: Pin assignments

- Click **OK** on the pop-up window notifying the Tcl Script has been executed. And click Close on the **TCL Scripts** window.
- Verify in the Assignment Editor  that pin assignments have been created successfully



	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓		memor...dq[0]	I/O Standard	SSTL-15	Yes	ddr3		__d..._p0
2	✓		memor...dq[0]	Output Termination	Series 40...libration	Yes	ddr3		__d..._p0
3	✓		memor...dq[1]	I/O Standard	SSTL-15	Yes	ddr3		__d..._p0
4	✓		memor...dq[1]	Output Termination	Series 40...libration	Yes	ddr3		__d..._p0
5	✓		memor...dq[2]	I/O Standard	SSTL-15	Yes	ddr3		__d..._p0

Figure 34: Confirming Pin assignments

5. Pin locations for external memory systems are not automatically created. Run the pin location tcl script by clicking on the **Tcl Scripts** under the **Tools** menu. Select the tcl script ***m10_ddr3_pin_locations.tcl.tcl*** and click on **Run** to assign the pin location.

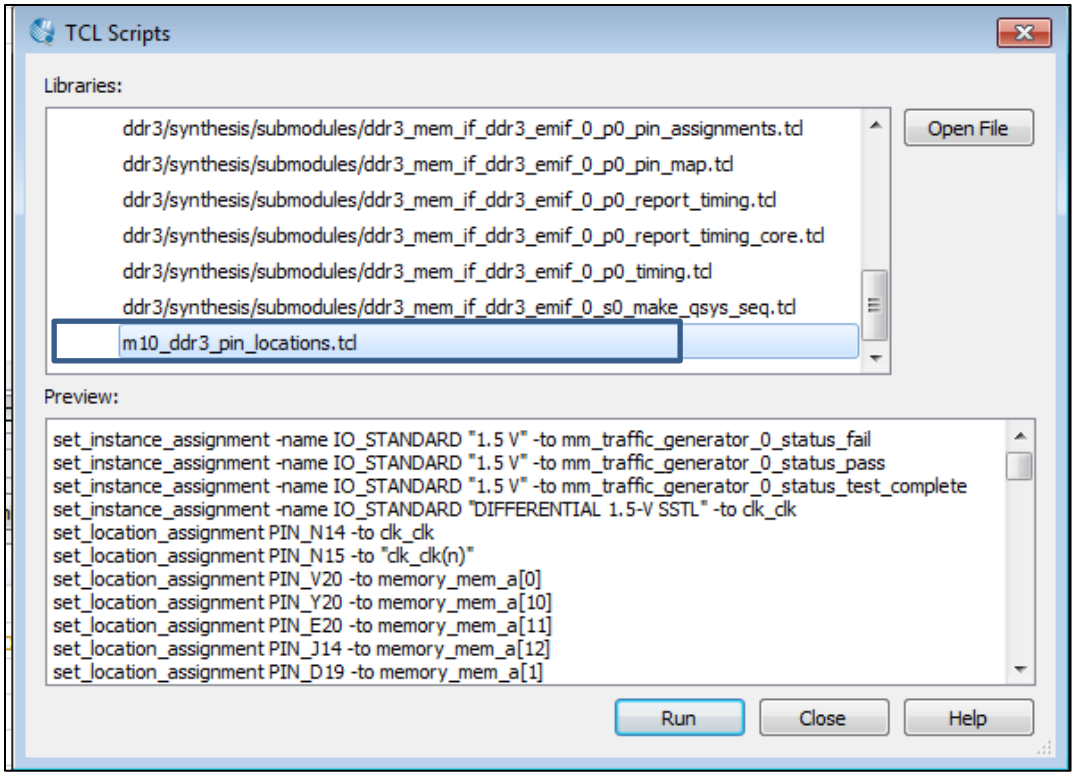
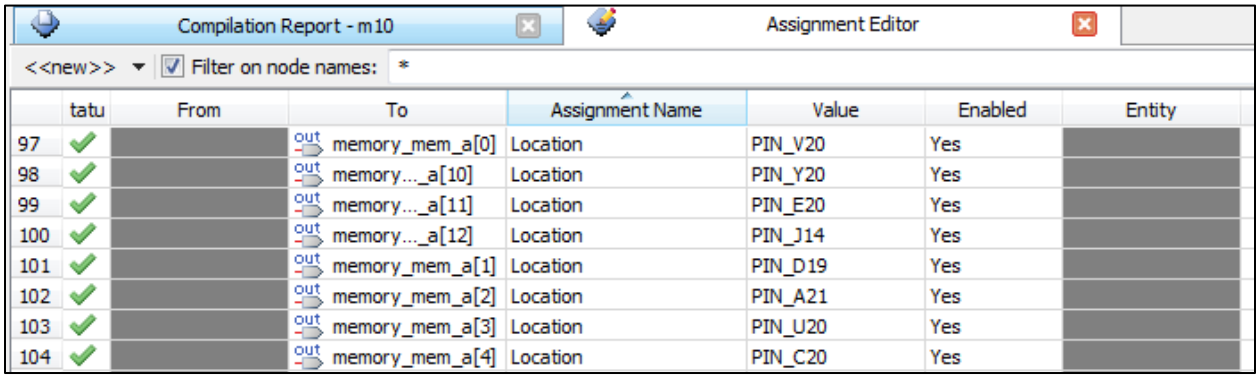


Figure 35: Pin Location

- Click **OK** on the pop-up window notifying the Tcl Script has been executed. And click Close on the **TCL Scripts** window.
- Verify in Pin Planner or Assignment Editor that pin locations have been created successfully



	tatu	From	To	Assignment Name	Value	Enabled	Entity
97	✓		memory_mem_a[0]	Location	PIN_V20	Yes	
98	✓		memory..._a[10]	Location	PIN_Y20	Yes	
99	✓		memory..._a[11]	Location	PIN_E20	Yes	
100	✓		memory..._a[12]	Location	PIN_J14	Yes	
101	✓		memory_mem_a[1]	Location	PIN_D19	Yes	
102	✓		memory_mem_a[2]	Location	PIN_A21	Yes	
103	✓		memory_mem_a[3]	Location	PIN_U20	Yes	
104	✓		memory_mem_a[4]	Location	PIN_C20	Yes	

Figure 36: Confirming Pin Location (Assignment Editor)

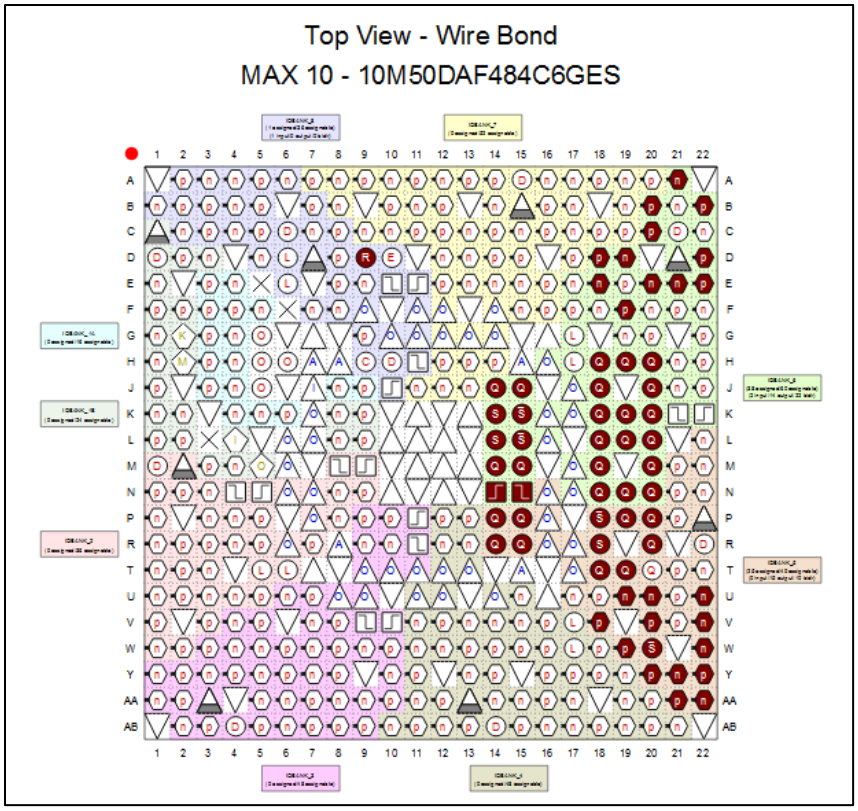


Figure 37: Confirming Pin Location (Pin Planner)

6. Run full compilation by clicking the **Start Compilation** under the **Processing** menu. The compilation may take around 10 minutes to complete depending on compilation PC.

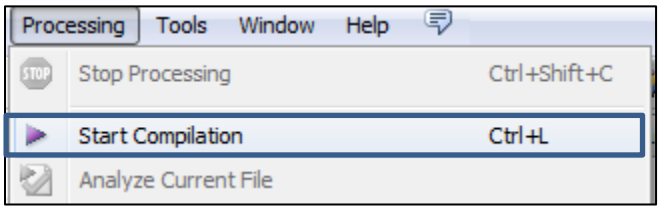


Figure 38: Compilation

7. Review all the Critical Warnings and Warnings and determines if it is acceptable or need to be address.

Note : For Quartus 15.0 and 15.1, the timing model for MAX 10 device is still preliminary. You might see some timing violation on the DDR3 design while using Quartus 15.0.

Note : For Quartus version prior to 15.1.1, there will be timing violation on the “DQS vs CK” and “Write” operation in the TimeQuest Timing Analyzer. This is due to the SDC constraint for QII version prior to 15.1.1 is set to hardcoded signal name rather than variable name.

Adding SignalTap file Into Existing Example Design

1. In the Quartus II software, launch **SignalTap II Logic Analyzer** from **Tools** menu.

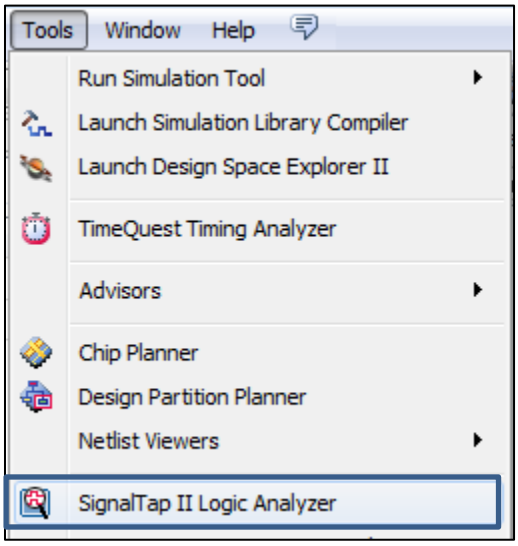


Figure 39: SignalTap II

i. Under the **Signal Configuration** section, click on the browse button.

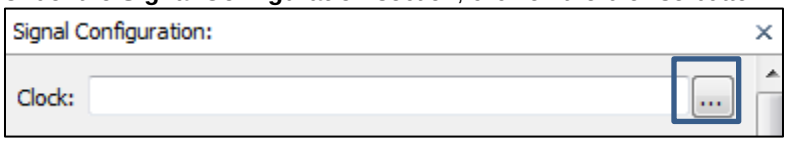


Figure 40: Include Clock

ii. In the Node Finder window, change the Filter to **SignalTap II: pre-synthesis**. Enter the name *pll_ref_clk* in the Named section and click on the List button. Select the **pll_ref_clock** and click on the **Copy to Selected Nodes** list button. Finally, click **OK** on the **Node Finder** window.

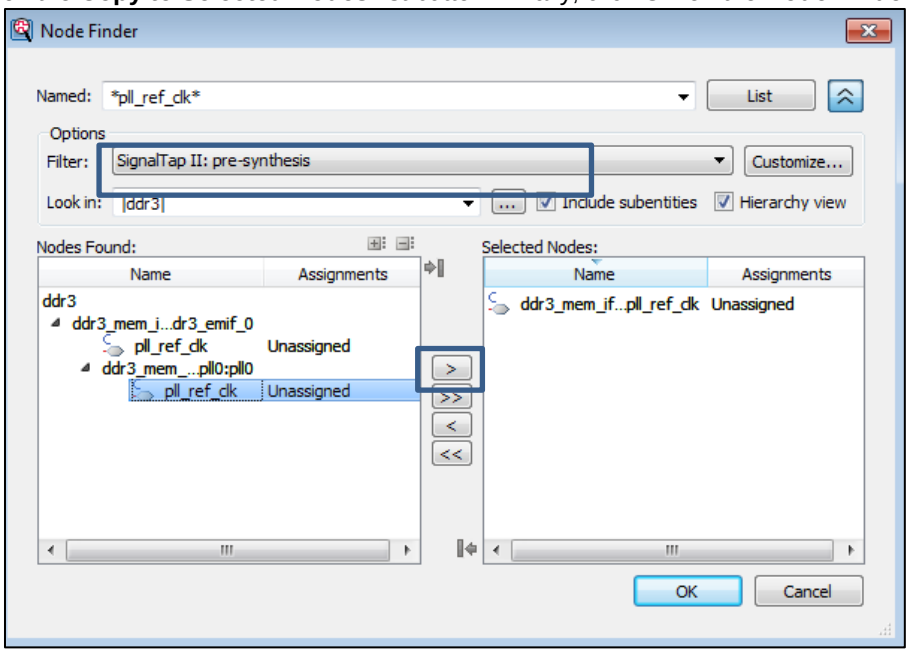


Figure 41: Include Reference Clock

iii. Include the following signals into your SignalTap file. Double click on the **Setup** area and the pop-up **Node Finder** window will appear.

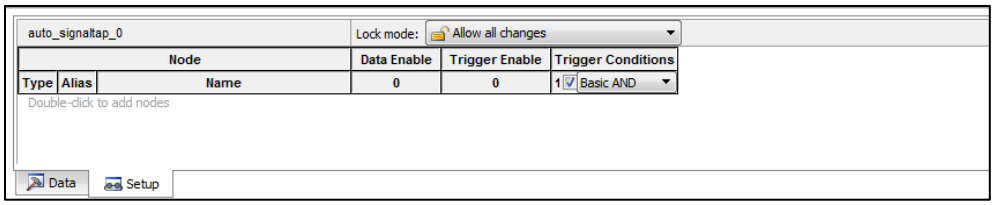


Figure 42: Setup area

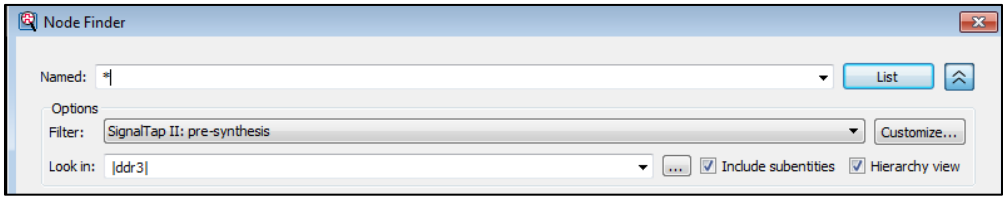


Figure 43: Node Finder

Look for the following signals in the **Named** section on Node Finder window and click on the List button. Select the signal and click on the **Copy to Selected Nodes** list button. Finally, click **Insert** on the **Node Finder** window

- ddr3_mem_if_ddr3_emif_0:mem_if_ddr3_emif_0|local_cal_fail
- ddr3_mem_if_ddr3_emif_0:mem_if_ddr3_emif_0|local_cal_success
- ddr3_mem_if_ddr3_emif_0:mem_if_ddr3_emif_0|local_init_done
- ddr3_mm_traffic_generator_0:mm_traffic_generator_0|pass
- ddr3_mm_traffic_generator_0:mm_traffic_generator_0|fail
- ddr3_mm_traffic_generator_0:mm_traffic_generator_0|test_complete
- ddr3_mem_if_ddr3_emif_0:mem_if_ddr3_emif_0|ddr3_mem_if_ddr3_emif_0_p0:p0|phy_cal_debug_info[31..0]
- ddr3_mm_traffic_generator_0:mm_traffic_generator_0|driver_no_ifdef_params:traffic_generator_0|driver_fsm_no_ifdef_params:real_driver.driver_fsm_inst|loop_counter[31..0]

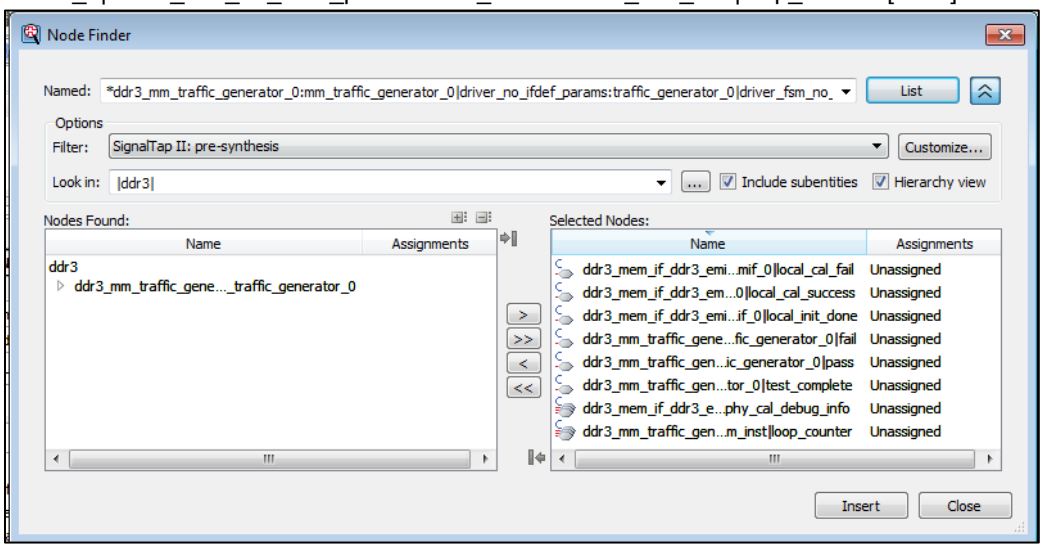


Figure 44: Signals on Node Finder

2. Safe the file by selecting **Save** from the **File** menu. Select the name you require for the stp file and click on the **Save** button. Remember to select the Add file to current project check box. Select **Yes** when prompt to enable the SignalTap II file for the current project.

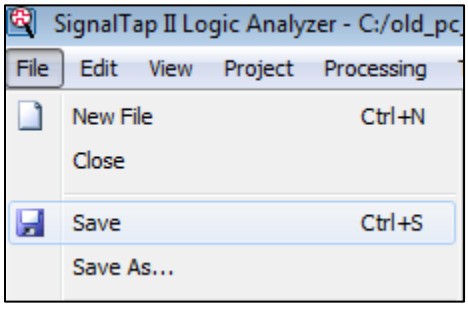


Figure 45: Save the SignalTap II File

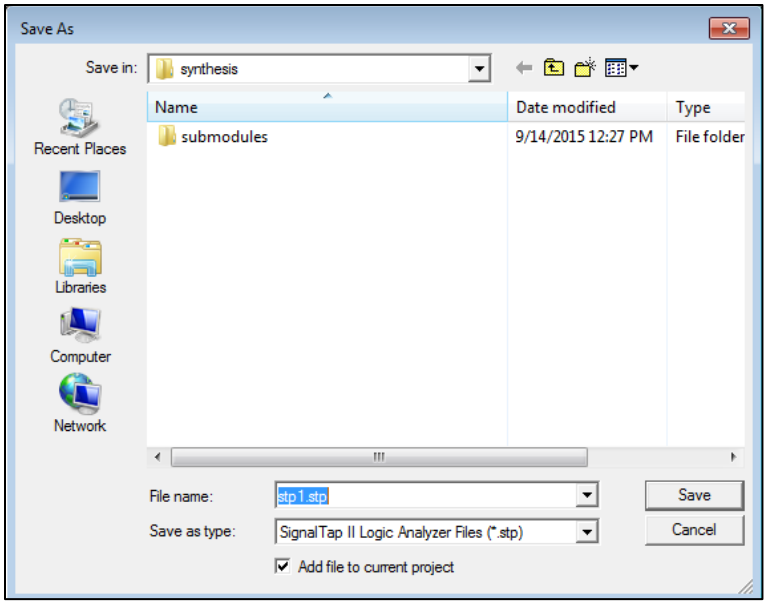


Figure 46: Add SignalTap II File to Project

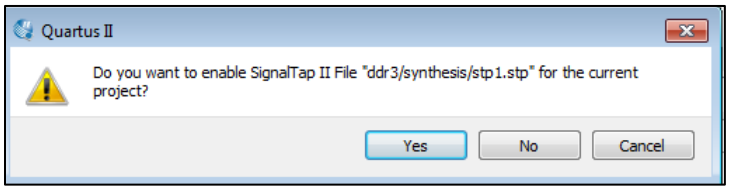


Figure 47: Enable SignalTap II file

3. Run full compilation after including the SignalTap II file to the design

Design Hardware Test

1. In the Quartus II software, launch **SignalTap II Logic** from **Tools** menu.

2. In **Jtag Chain Configuration** window under **SignalTap Logic** GUI, configure the **Hardware** and **Device** based on the targeted board. For **File**, browse for the **m10.sof** file (located in <design_directory>/ output_files folder) and click **Open**.

3. Click **Program Device** button  to configure the FPGA.

4. Select the SignalTap instance and click the **Autorun Analysis** button  next to **Instance Manager** label. The SignalTap II Logic Analyzer Pane will shows the acquired data from each signal in the **Data** tab.

6. On the development kit, push once and release the reset button on the development kit to reset the design.

7. Now observe the data for **local_cal_success** and **traffic_gen_pass** and **traffice_gen_timeout** instances change from 0 to 1 on the SignalTap which indicate the calibration is success and pass the example driver tests. Also the LED1 on the board will light up and turn to **green** colour.