My First HPS System

PDF created: December 13, 2017
Validated using tools release: 17.0.0

Contents

Overview

This tutorial demonstrates how to instantiate and configure a Hard Processor System (HPS) component into a Qsys system. Unlike other components in the prior "My First Qsys System" tutorial, the HPS component does not define any soft logic to be configured in the FPGA, rather, it allows the designer to configure the interfaces between the existing processor hardware on the SoC FPGA chip. Configuring an entire processor system is a necessarily detailed process, but the level of customization available is precisely the advantage an SoC FPGA provides. This tutorial shows as closely as possible how to set up the HPS in a similar manner to the design that shipped with the Terasic DE10-Nano board, and integrate it with the existing Qsys system from the preceding "My First Qsys System" tutorial. This will allow us to integrate this example onto the existing SD card image that ships with the Terasic DE10-Nano board.

Prerequisites

The following are required:

- Windows* or Linux* development host PC
- Installed Intel® Quartus® Prime Software Suite. Either the Lite or Standard Edition, but not the Pro Edition.
- Completed Intel Quartus software project from "My First Qsys System"
  - Either follow the tutorial steps presented here.
  - Or, you can download an archive of the required contents from that tutorial to your local file system here.

Instantiating the Hard Processor System

This section will describe the steps required to instantiate an HPS core into the Qsys system that was created in the "My First Qsys System" tutorial. It assumes that you have the completed **blink** project from that tutorial open.

Step 1. With the **blink** project from the previous tutorial open in the Intel Quartus software tool, start by launching Qsys by clicking the Qsys button on the Intel Quartus software toolbar, or select Qsys from the **Tools** menu. When the Qsys **Open** dialog appears select the **soc_system.qsys** system and click the **Open** button.

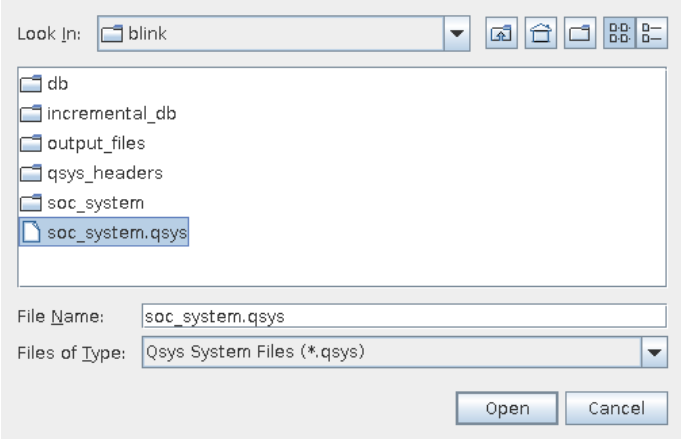


Figure 1: Qsys Open Dialog

Step 2. In the **IP Catalog**, search for **hard processor** and double click on **Arria V/Cyclone V Hard Processor System** to bring up the HPS component configuration wizard.

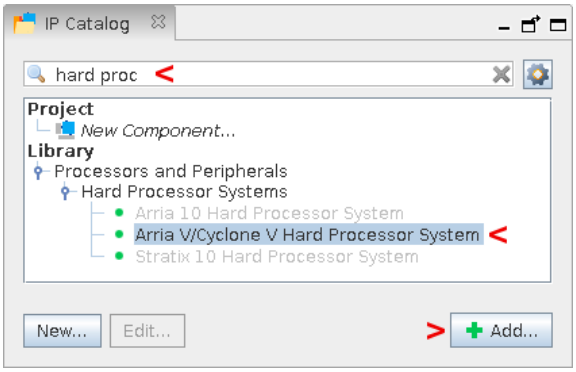


Figure 2: Hard Processor Search

Step 3. On the **FPGA Interfaces** tab in the **General** section, deselect the **Enable MPU standby and event signals** parameter.

It should look like this after you do.

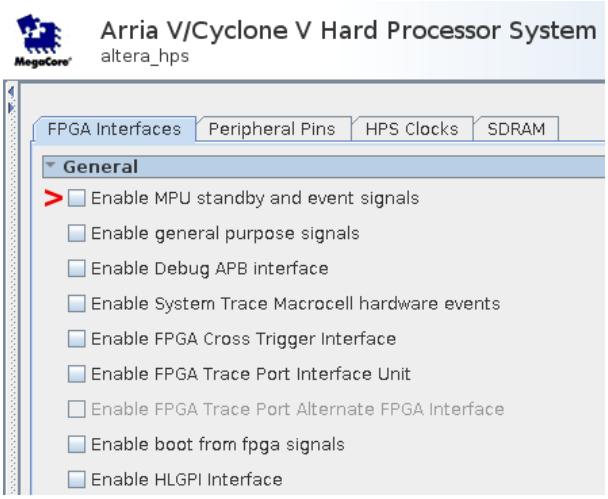


Figure 3: FPGA Interfaces - General

Step 4. In the **AXI Bridges** section, select **unused** for the **FPGA-to-HPS interface width** parameter.

It should look like this after you do.

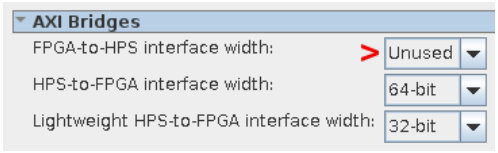


Figure 4: AXI Bridges

Step 5. In the **FPGA-to-HPS SDRAM Interface** section, select the **f2h_sdram0** interface and click the - button to remove it from the list.

It should look like this after you do.

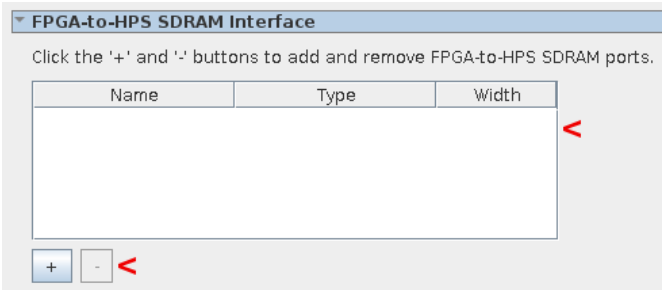


Figure 5: FPGA-to-HPS SDRAM Interface

Step 6. In the **Resets** section, do not change any of the default settings.

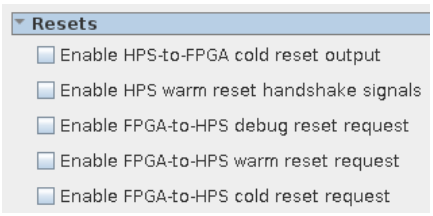


Figure 6: Resets

Step 7. In the **DMA Peripheral Request** section, do not change any of the default settings.

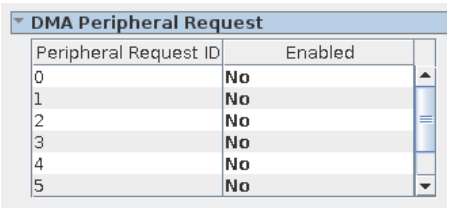


Figure 7: DMA Peripheral Request

Step 8. In the **Interrupts** section, do not change any of the default settings.

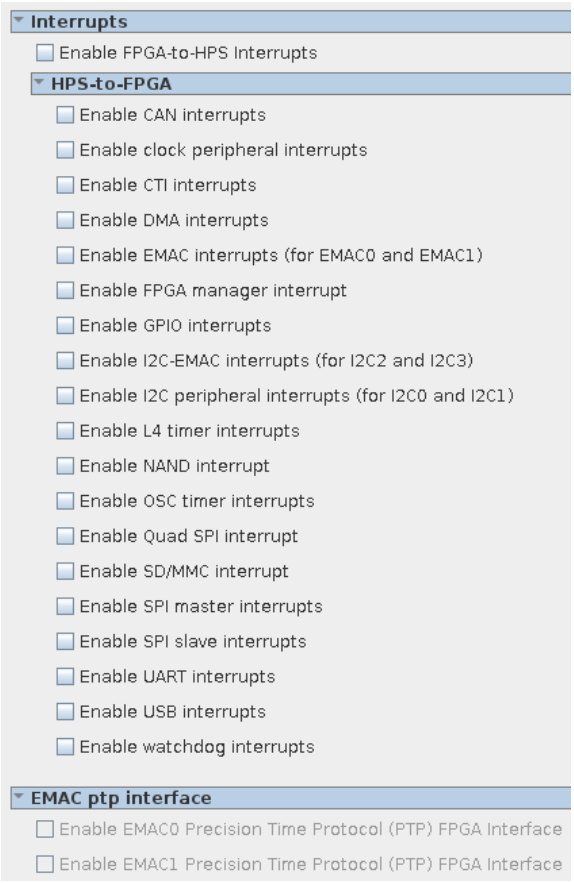


Figure 8: Interrupts

Step 9. On the **Peripheral Pins** tab in the **Ethernet Media Access Controller** section, make the following changes:

Step 9a. Select **HPS I/O Set 0** for the **EMAC1 pin** parameter.

Step 9b. Select **RGMII** for the **EMAC1 mode** parameter.

It should look like this after you do.

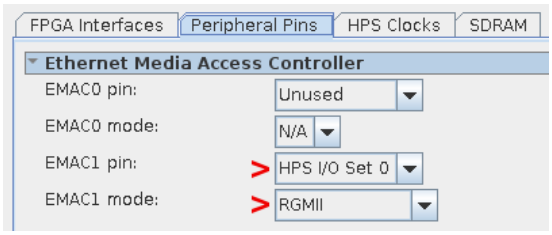


Figure 9: Ethernet Media Access Controller

Step 10. In the **NAND Flash Controller** section, do not change any of the default settings.

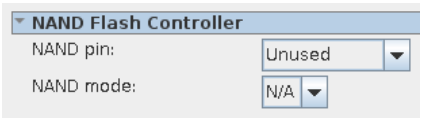


Figure 10: NAND Flash Controller

Step 11. In the **Quad SPI Flash Controller** section, do not change any of the default settings.

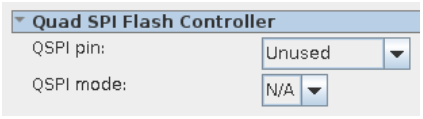


Figure 11: Quad SPI Flash Controller

Step 12. In the **SD/MMC Controller** section, make the following changes:

Step 12a. Select **HPS I/O Set 0** for the **SDIO pin** parameter.

Step 12b. Select **4-bit Data** for the **SDIO mode** parameter.

It should look like this after you do.

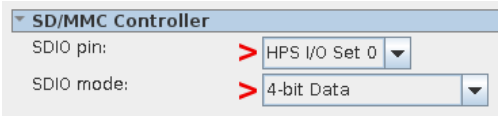


Figure 12: SD/MMC Controller

Step 13. In the **USB Controllers** section, make the following changes:

Step 13a. Select **HPS I/O Set 0** for the **USB1 pin** parameter.

Step 13b. Select **SDR with PHY clock output mode** for the **USB1 PHY interface mode** parameter.

It should look like this after you do.

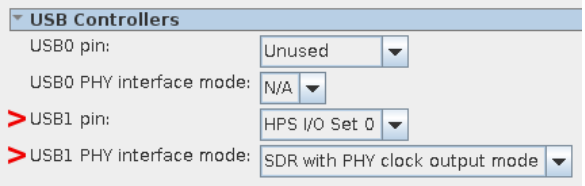


Figure 13: USB Controllers

Step 14. In the **SPI Controllers** section, make the following changes:

Step 14a. Select **HPS I/O Set 0** for the **SPIM1 pin** parameter.

Step 14b. Select **Single Slave Select** for the **SPIM1 mode** parameter.

It should look like this after you do.

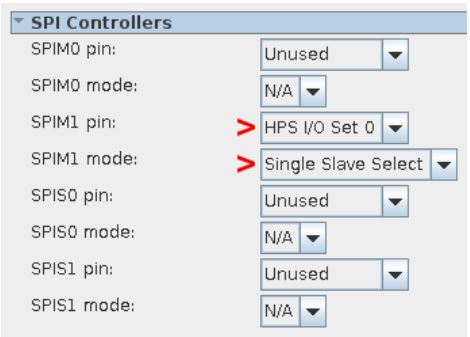


Figure 14: SPI Controllers

Step 15. In the **UART Controllers** section, make the following changes:

Step 15a. Select **HPS I/O Set 0** for the **UART0 pin** parameter.

Step 15b. Select **No Flow Control** for the **UART0 mode** parameter.

It should look like this after you do.

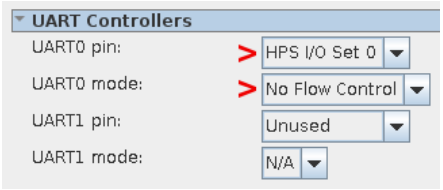


Figure 15: UART Controllers

Step 16. In the **I2C Controllers** section, make the following changes:

Step 16a. Select **HPS I/O Set 0** for the **I2C0 pin** parameter.

Step 16b. Select **HPS I/O Set 0** for the **I2C1 pin** parameter.

Step 16c. Select **I2C** for the **I2C0 mode** parameter.

Step 16d. Select **I2C** for the **I2C1 mode** parameter.

It should look like this after you do.

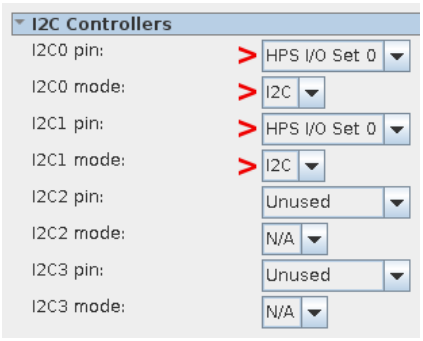


Figure 16: I2C Controllers

Step 17. In the **CAN Controllers** section, do not change any of the default settings.

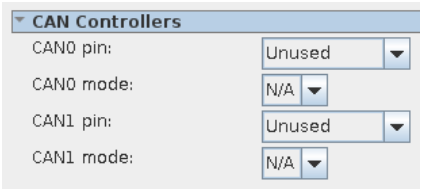


Figure 17: CAN Controllers

Step 18. In the **Trace Port Interface Unit** section, do not change any of the default settings.

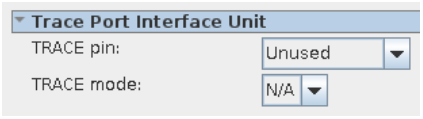


Figure 18: Trace Port Interface Unit

Step 19. In the **Peripherals Mux Table** section, enable the following GPIO ports by clicking on the GPIOxx button in the second column from the right of the table. Select the following GPIO:

- **GPIO09**
- **GPIO35**
- **GPIO40**
- **GPIO53**
- **GPIO54**
- **GPIO61**

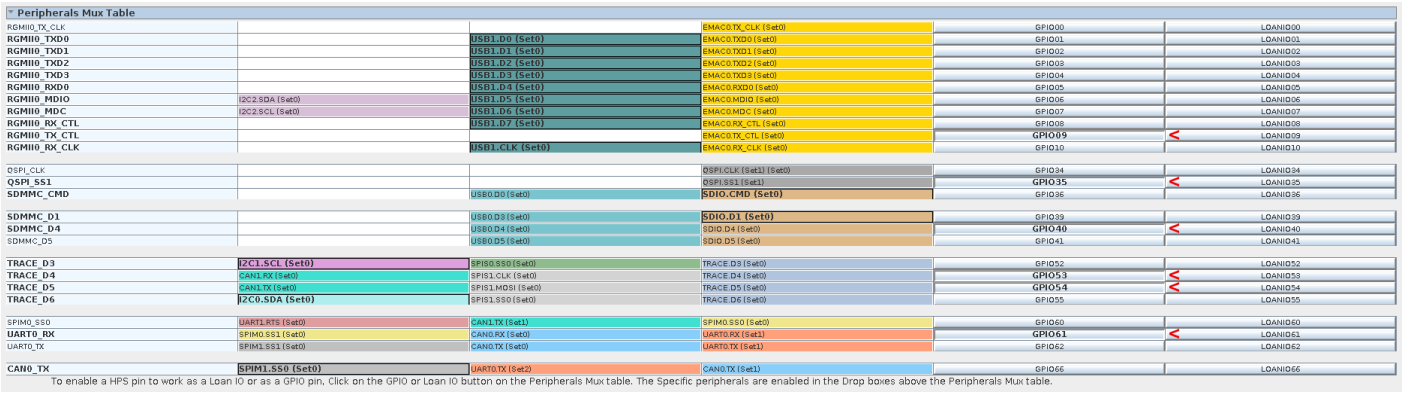


Figure 19: Peripherals Mux Table

Step 20. Moving on to the **HPS Clocks** tab, under the **Input Clocks** sub-tab, in the **External Clock Sources** section, do not change any of the default settings.

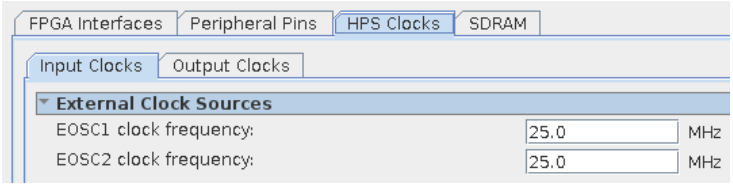


Figure 20: External Clock Sources

Step 21. In the **FPGA-to-HPS PLL Reference Clocks** section, do not change any of the default settings.

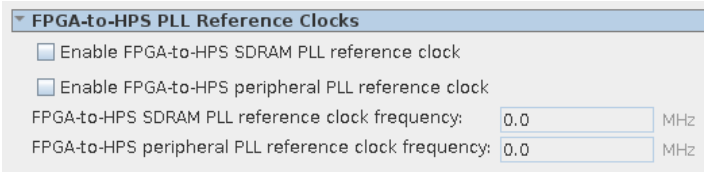


Figure 21: FPGA-to-HPS PLL Reference Clocks

Step 22. In the **Peripheral FPGA Clocks** section, do not change any of the default settings.

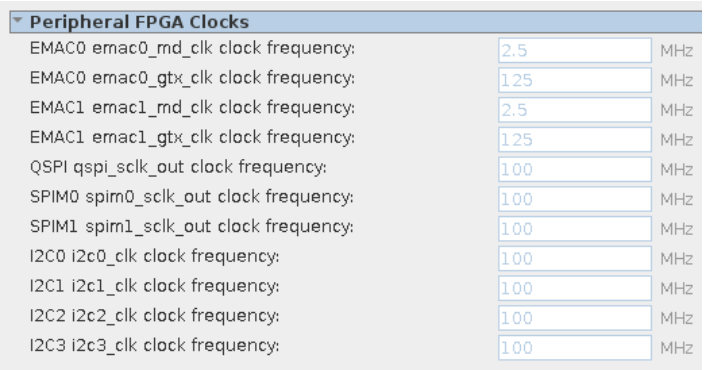


Figure 22: Peripheral FPGA Clocks

Step 23. Moving on to the **Output Clocks** sub-tab, in the **Clock Sources** section, do not change any of the default settings.

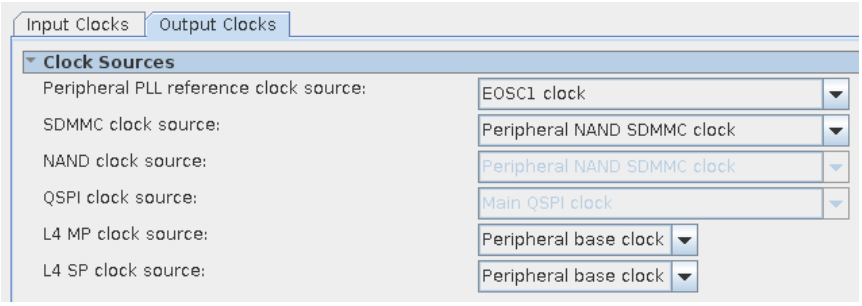


Figure 23: Clock Sources

Step 24. In the **Main PLL Output Clocks** section, do not change any of the default settings.

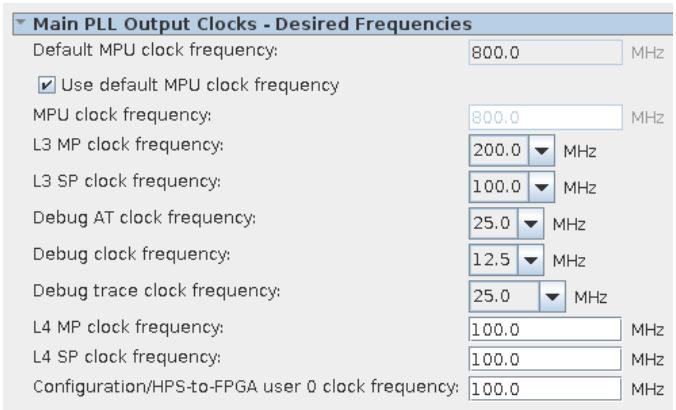


Figure 24: Main PLL Output Clocks

Step 25. In the **Peripheral PLL Output Clocks** section, do not change any of the default settings.

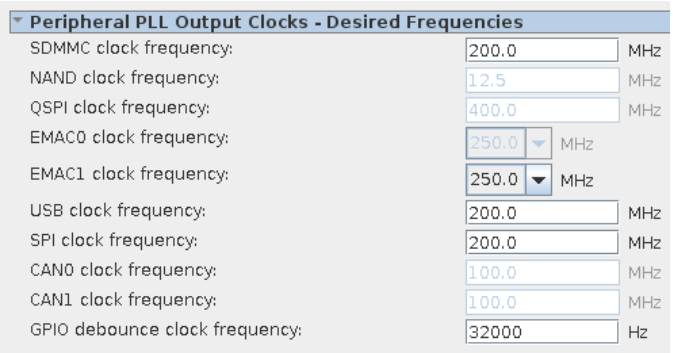


Figure 25: Peripheral PLL Output Clocks

Step 26. In the **HPS-to-FPGA User Clocks** section, do not change any of the default settings.

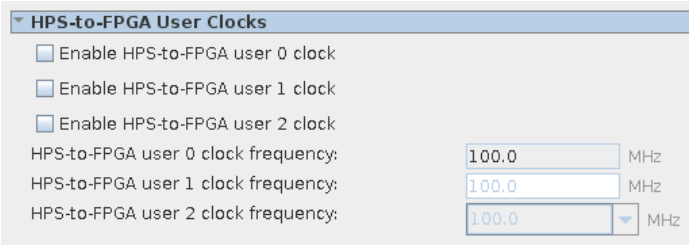


Figure 26: HPS-to-FPGA User Clocks

Step 27. Moving on to the **SDRAM** tab, under the **PHY Settings** sub-tab, in the **Clocks** section, make the following changes:

Step 27a. Change the **Memory clock frequency** parameter to **400.0**.

Step 27b. Change the **PLL reference clock frequency** parameter to **25.0**.

It should look like this after you do.

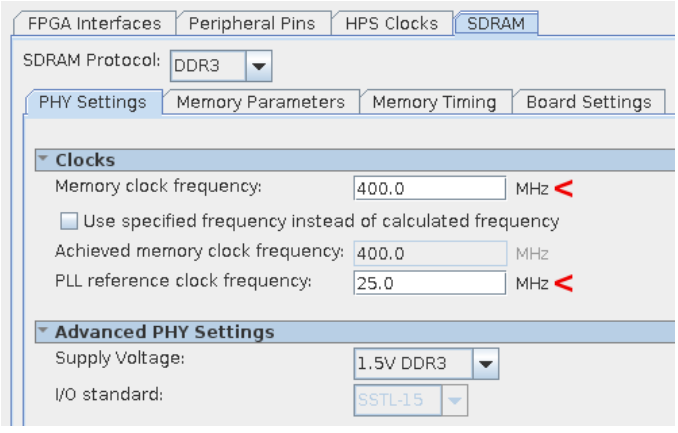


Figure 27: PHY Settings

Step 28. Moving on to the **Memory Parameters** sub-tab, make the following changes:

Step 28a. Change the **Memory vendor** parameter to **Other**.

Step 28b. Change the **Memory device speed grade** parameter to **800.0**.

Step 28c. Change the **Total interface width** parameter to **32**.

Step 28d. Change the **Row address width** parameter to **15**.

Step 28e. Change the **Column address width** parameter to **10**.

It should look like this after you do.

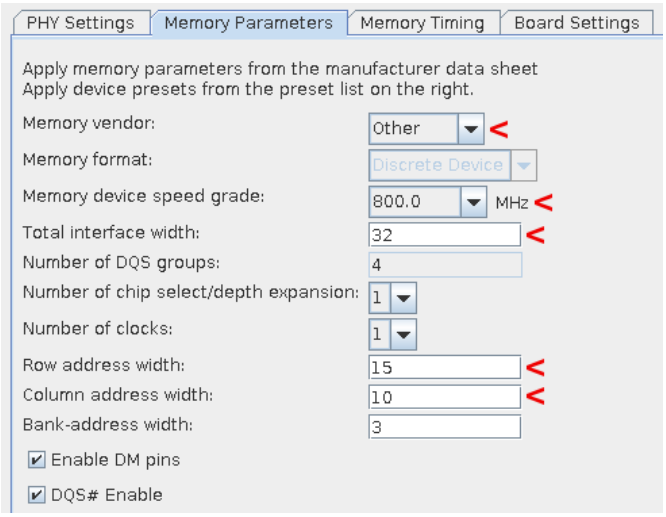


Figure 28: Memory Parameters

Step 29. In the **Memory Initialization Options** section:

Step 29a. Change the **ODT Rtt nominal value** parameter to **RZQ/6**.

Step 29b. Change the **Memory write CAS latency setting** parameter to **7**.

It should look like this after you do.

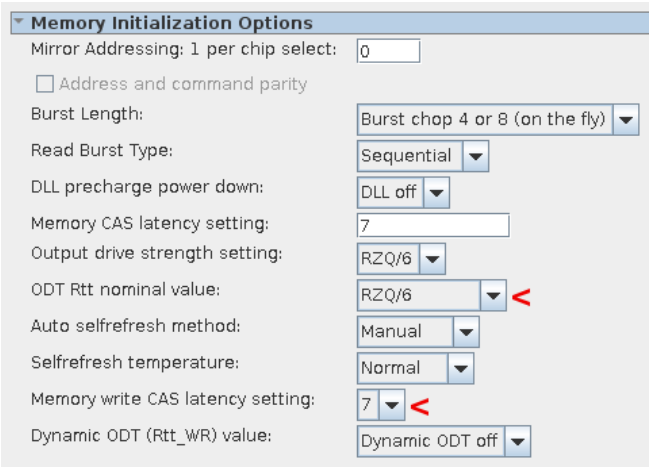


Figure 29: Memory Initialization Options

Step 30. Moving on to the **Memory Timing** sub-tab, make the following changes:

Step 30a. Change the **tINIT** parameter to **500**.

Step 30b. Change the **tMRD (tMRW)** parameter to **4**.

Step 30c. Change the **tRAS** parameter to **35.0**.

Step 30d. Change the **tRCD** parameter to **13.75**.

Step 30e. Change the **tRP** parameter to **13.75**.

Step 30f. Change the **tREFI (tREFIab)** parameter to **7.8**.

Step 30g. Change the **tRFC (tRFCab)** parameter to **300.0**.

Step 30h. Change the **tWTR** parameter to **4**.

It should look like this after you do.

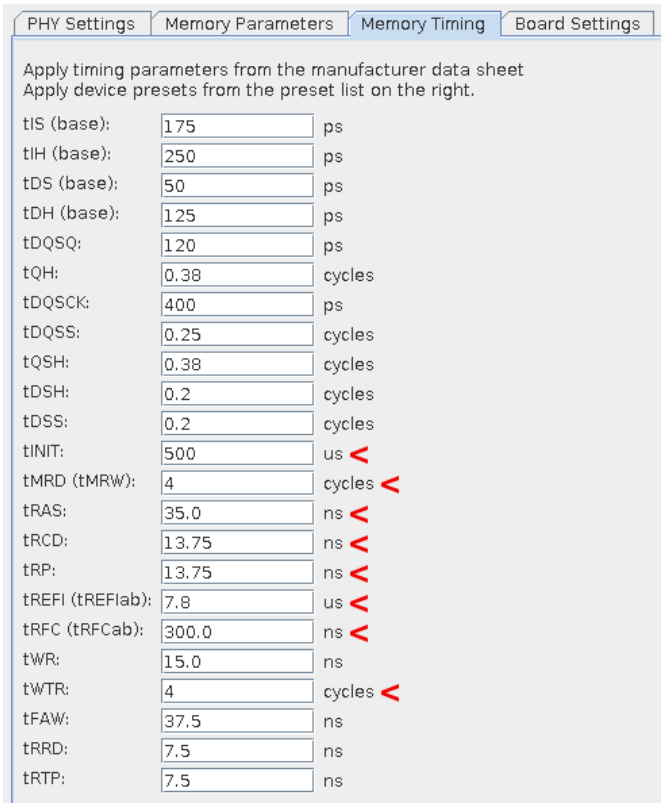


Figure 30: Memory Timing

Step 31. Moving on to the **Board Settings** tab, in the **Setup and Hold Derating** section, do not change any of the default settings.

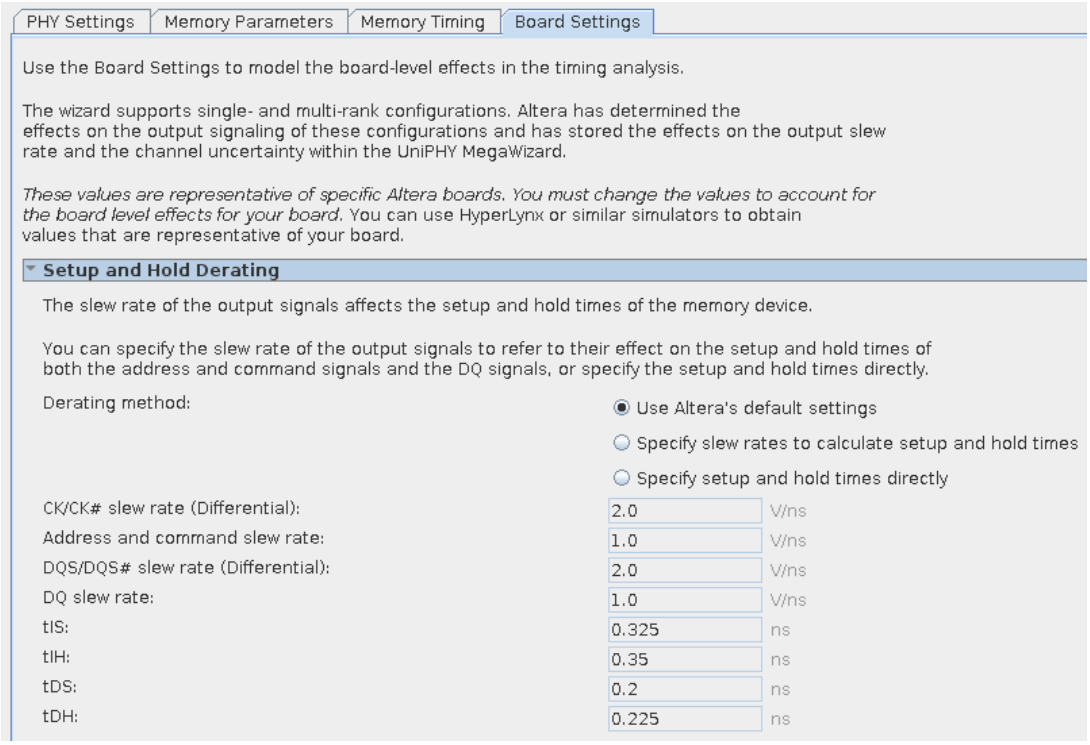


Figure 31: Setup and Hold Derating

Step 32. In the **Channel Signal Integrity** section, do not change any of the default settings.

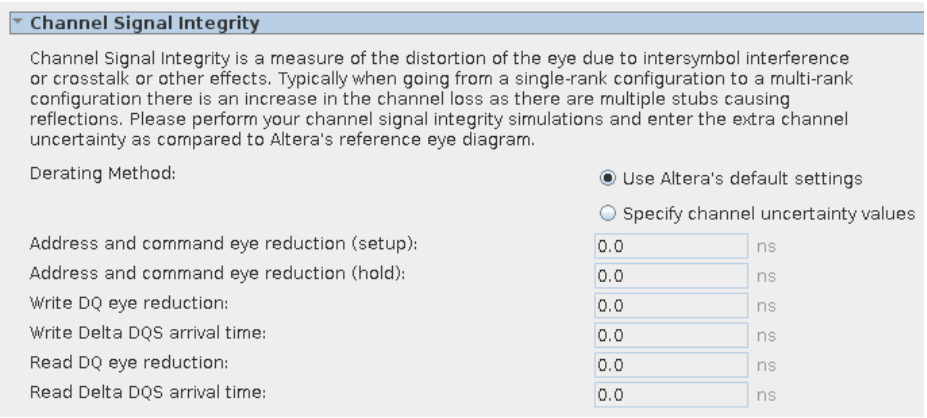


Figure 32: Channel Signal Integrity

Step 33. In the **Board Skews** section, do not change any of the default settings.

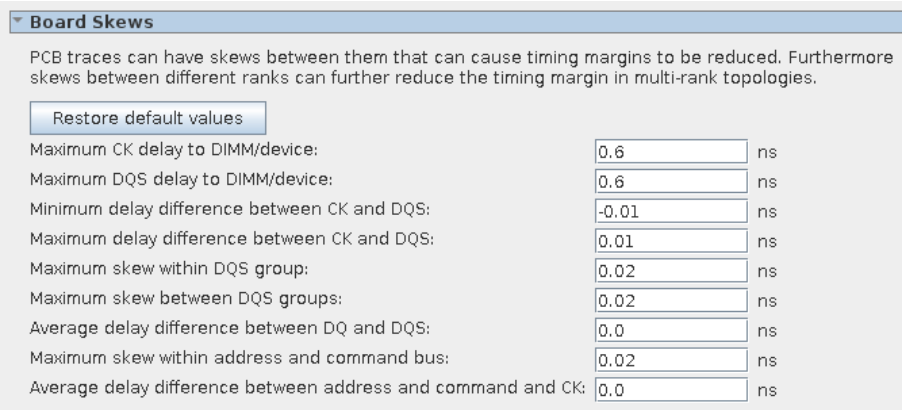


Figure 33: Board Skews

Step 34. The HPS configuration is now complete, you can click the **Finish** button in the bottom right corner of the dialog window.

Step 35. The HPS instance appears in the system with the default name **hps_0** which we will keep that way. The **memory** conduit interface should already be exported with the name **memory** in the export column as well as the **hps_io** conduit interface named **hps_io** in the export column. Rename the **hps_io** conduit export name to **hps_0_hps_io**

It should look like this after you do.

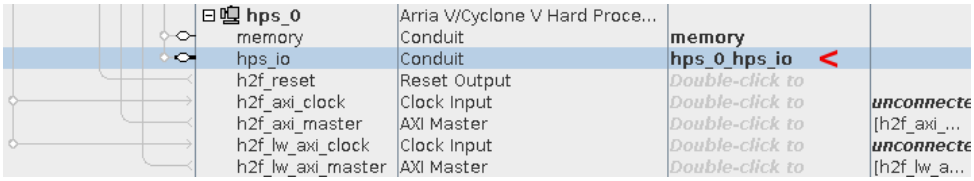


Figure 34: Board Skews

Step 36. Complete the HPS connections into the rest of the system:

Step 36a. Connect the **h2f_axi_clock** clock input to the **clk_0** clock output.

Step 36b. Connect the **h2f_lw_axi_clock** clock input to the **clk_0** clock output.

Step 36c. Connect the **h2f_reset** reset output to the reset input of all the other components.

Step 36d. Connect the **h2f_axi_master** AXI Master to the **ocram_64k/s1** Avalon Slave.

Step 36e. Connect the **h2f_axi_master** AXI Master to the **default_16b/s1** Avalon Slave.

Step 36f. Connect the **h2f_lw_axi_master** AXI Master to the **default_16b/s1** Avalon Slave.

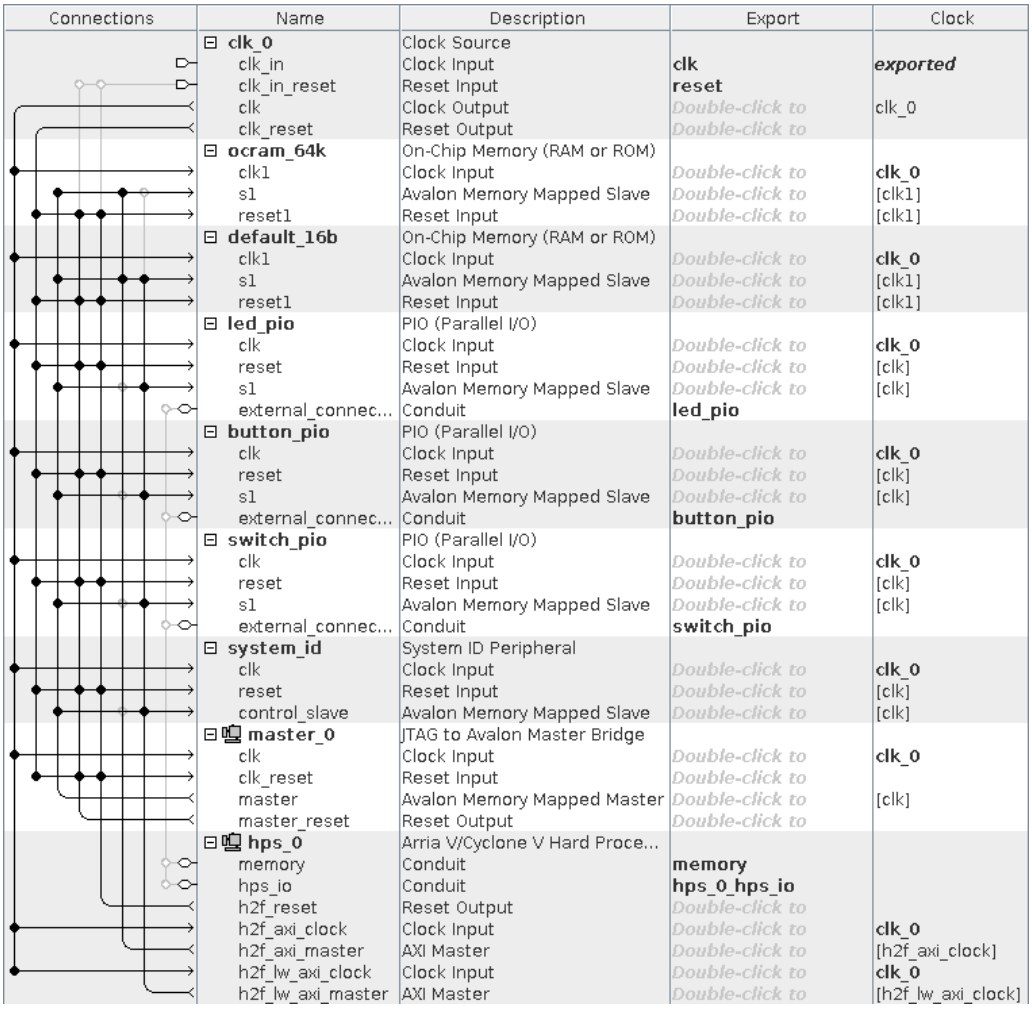
Step 36g. Connect the **h2f_lw_axi_master** AXI Master to the **led_pio/s1** Avalon Slave.

Step 36h. Connect the **h2f_lw_axi_master** AXI Master to the **button_pio/s1** Avalon Slave.

Step 36i. Connect the **h2f_lw_axi_master** AXI Master to the **switch_pio/s1** Avalon Slave.

Step 36j. Connect the **h2f_lw_axi_master** AXI Master to the **system_id/control_slave** Avalon Slave.

It should look like this after you do.



Connections	Name	Description	Export	Clock
	clk_0	Clock Source		
	clk_in	Clock Input	**clk**	***exported***
	clk_in_reset	Reset Input	**reset**	
	clk	Clock Output	*Double-click to*	clk_0
	clk_reset	Reset Output	*Double-click to*	
	ocram_64k	On-Chip Memory (RAM or ROM)		
	clk1	Clock Input	*Double-click to*	**clk_0**
	s1	Avalon Memory Mapped Slave	*Double-click to*	[clk1]
	reset1	Reset Input	*Double-click to*	[clk1]
	default_16b	On-Chip Memory (RAM or ROM)		
	clk1	Clock Input	*Double-click to*	**clk_0**
	s1	Avalon Memory Mapped Slave	*Double-click to*	[clk1]
	reset1	Reset Input	*Double-click to*	[clk1]
	led_pio	PIO (Parallel I/O)		
	clk	Clock Input	*Double-click to*	**clk_0**
	reset	Reset Input	*Double-click to*	[clk]
	s1	Avalon Memory Mapped Slave	*Double-click to*	[clk]
	external_connec...	Conduit	**led_pio**	
	button_pio	PIO (Parallel I/O)		
	clk	Clock Input	*Double-click to*	**clk_0**
	reset	Reset Input	*Double-click to*	[clk]
	s1	Avalon Memory Mapped Slave	*Double-click to*	[clk]
	external_connec...	Conduit	**button_pio**	
	switch_pio	PIO (Parallel I/O)		
	clk	Clock Input	*Double-click to*	**clk_0**
	reset	Reset Input	*Double-click to*	[clk]
	s1	Avalon Memory Mapped Slave	*Double-click to*	[clk]
	external_connec...	Conduit	**switch_pio**	
	system_id	System ID Peripheral		
	clk	Clock Input	*Double-click to*	**clk_0**
	reset	Reset Input	*Double-click to*	[clk]
	control_slave	Avalon Memory Mapped Slave	*Double-click to*	[clk]
	master_0	JTAG to Avalon Master Bridge		
	clk	Clock Input	*Double-click to*	**clk_0**
	clk_reset	Reset Input	*Double-click to*	
	master	Avalon Memory Mapped Master	*Double-click to*	[clk]
	master_reset	Reset Output	*Double-click to*	
	hps_0	Arria V/Cyclone V Hard Proce...		
	memory	Conduit	**memory**	
	hps_io	Conduit	**hps_0_hps_io**	
	h2f_reset	Reset Output	*Double-click to*	
	h2f_axi_clock	Clock Input	*Double-click to*	**clk_0**
	h2f_axi_master	AXI Master	*Double-click to*	[h2f_axi_clock]
	h2f_lw_axi_clock	Clock Input	*Double-click to*	**clk_0**
	h2f_lw_axi_master	AXI Master	*Double-click to*	[h2f_lw_axi_clock]

Figure 35: Board Skews

Step 37. Save the system, generate the HDL, and close Qsys.

Integrating the Qsys System into the Intel Quartus Software Project

This section will describe how to incorporate the newly generated Qsys output into the existing **blink** Intel Quartus software project.

Step 1. Update the **blink.v** source file with the new code for the Qsys system instantiation like this:

Download the blink.v file here.
If you wish to view the blink.v file in the GitHub* repo you can look here.

```
//
// Copyright (c) 2017 Intel Corporation
//
// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to
// deal in the Software without restriction, including without limitation the
// rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
// sell copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
// IN THE SOFTWARE.
//

// create module
module blink(
        input  wire       clk,          // 50MHz FPGA input clock

        input  wire [1:0] push_button,  // KEY[1:0]
        input  wire [3:0] switch,       // SW[3:0]

        output wire [7:0] leds,         // LED[7:0]

        // HPS memory controller ports
        output wire [14:0] hps_memory_mem_a,
        output wire [2:0]  hps_memory_mem_ba,
        output wire        hps_memory_mem_ck,
        output wire        hps_memory_mem_ck_n,
        output wire        hps_memory_mem_cke,
        output wire        hps_memory_mem_cs_n,
        output wire        hps_memory_mem_ras_n,
        output wire        hps_memory_mem_cas_n,
        output wire        hps_memory_mem_we_n,
        output wire        hps_memory_mem_reset_n,
        inout  wire [31:0] hps_memory_mem_dq,
        inout  wire [3:0]  hps_memory_mem_dqs,
        inout  wire [3:0]  hps_memory_mem_dqs_n,
        output wire        hps_memory_mem_odt,
        output wire [3:0]  hps_memory_mem_dm,
        input  wire        hps_memory_oct_rzqin,

        // HPS peripheral ports
        output wire        hps_io_hps_io_emac1_inst_TX_CLK,
```

```
        output wire         hps_io_hps_io_emac1_inst_TXD0,
        output wire         hps_io_hps_io_emac1_inst_TXD1,
        output wire         hps_io_hps_io_emac1_inst_TXD2,
        output wire         hps_io_hps_io_emac1_inst_TXD3,
        input  wire         hps_io_hps_io_emac1_inst_RXD0,
        inout  wire         hps_io_hps_io_emac1_inst_MDIO,
        output wire         hps_io_hps_io_emac1_inst_MDC,
        input  wire         hps_io_hps_io_emac1_inst_RX_CTL,
        output wire         hps_io_hps_io_emac1_inst_TX_CTL,
        input  wire         hps_io_hps_io_emac1_inst_RX_CLK,
        input  wire         hps_io_hps_io_emac1_inst_RXD1,
        input  wire         hps_io_hps_io_emac1_inst_RXD2,
        input  wire         hps_io_hps_io_emac1_inst_RXD3,
        inout  wire         hps_io_hps_io_sdio_inst_CMD,
        inout  wire         hps_io_hps_io_sdio_inst_D0,
        inout  wire         hps_io_hps_io_sdio_inst_D1,
        output wire         hps_io_hps_io_sdio_inst_CLK,
        inout  wire         hps_io_hps_io_sdio_inst_D2,
        inout  wire         hps_io_hps_io_sdio_inst_D3,
        inout  wire         hps_io_hps_io_usb1_inst_D0,
        inout  wire         hps_io_hps_io_usb1_inst_D1,
        inout  wire         hps_io_hps_io_usb1_inst_D2,
        inout  wire         hps_io_hps_io_usb1_inst_D3,
        inout  wire         hps_io_hps_io_usb1_inst_D4,
        inout  wire         hps_io_hps_io_usb1_inst_D5,
        inout  wire         hps_io_hps_io_usb1_inst_D6,
        inout  wire         hps_io_hps_io_usb1_inst_D7,
        input  wire         hps_io_hps_io_usb1_inst_CLK,
        output wire         hps_io_hps_io_usb1_inst_STP,
        input  wire         hps_io_hps_io_usb1_inst_DIR,
        input  wire         hps_io_hps_io_usb1_inst_NXT,
        input  wire         hps_io_hps_io_uart0_inst_RX,
        output wire         hps_io_hps_io_uart0_inst_TX,
        output wire         hps_io_hps_io_spim1_inst_CLK,
        output wire         hps_io_hps_io_spim1_inst_MOSI,
        input  wire         hps_io_hps_io_spim1_inst_MISO,
        output wire         hps_io_hps_io_spim1_inst_SS0,
        inout  wire         hps_io_hps_io_i2c0_inst_SDA,
        inout  wire         hps_io_hps_io_i2c0_inst_SCL,
        inout  wire         hps_io_hps_io_i2c1_inst_SDA,
        inout  wire         hps_io_hps_io_i2c1_inst_SCL,
        inout  wire         hps_io_hps_io_gpio_inst_GPIO09,
        inout  wire         hps_io_hps_io_gpio_inst_GPIO35,
        inout  wire         hps_io_hps_io_gpio_inst_GPIO40,
        inout  wire         hps_io_hps_io_gpio_inst_GPIO53,
        inout  wire         hps_io_hps_io_gpio_inst_GPIO54,
        inout  wire         hps_io_hps_io_gpio_inst_GPIO61
);

// Create a power on reset pulse for clean system reset on entry into user mode
// We create this with the altera_std_synchronizer core
wire sync_dout;
altera_std_synchronizer #(
        .depth (20)
) power_on_reset_std_sync_inst (
        .clk     (clk),
        .reset_n (1'b1),
        .din     (1'b1),
        .dout    (sync_dout)
);

// Create a qsys system reset signal that is the logical AND of the power on
```

```
// reset pulse and the KEY[0] push button
wire qsys_system_reset;
assign qsys_system_reset = sync_dout & push_button[0];

// Qsys system instantiation template from soc_system/soc_system_inst.v
soc_system u0 (
        .button_pio_export                       (push_button[1]),
        .clk_clk                                 (clk),
        .hps_0_hps_io_hps_io_emac1_inst_TX_CLK (hps_io_hps_io_emac1_inst_TX_CLK),
        .hps_0_hps_io_hps_io_emac1_inst_TXD0   (hps_io_hps_io_emac1_inst_TXD0),
        .hps_0_hps_io_hps_io_emac1_inst_TXD1   (hps_io_hps_io_emac1_inst_TXD1),
        .hps_0_hps_io_hps_io_emac1_inst_TXD2   (hps_io_hps_io_emac1_inst_TXD2),
        .hps_0_hps_io_hps_io_emac1_inst_TXD3   (hps_io_hps_io_emac1_inst_TXD3),
        .hps_0_hps_io_hps_io_emac1_inst_RXD0   (hps_io_hps_io_emac1_inst_RXD0),
        .hps_0_hps_io_hps_io_emac1_inst_MDIO   (hps_io_hps_io_emac1_inst_MDIO),
        .hps_0_hps_io_hps_io_emac1_inst_MDC    (hps_io_hps_io_emac1_inst_MDC),
        .hps_0_hps_io_hps_io_emac1_inst_RX_CTL (hps_io_hps_io_emac1_inst_RX_CTL),
        .hps_0_hps_io_hps_io_emac1_inst_TX_CTL (hps_io_hps_io_emac1_inst_TX_CTL),
        .hps_0_hps_io_hps_io_emac1_inst_RX_CLK (hps_io_hps_io_emac1_inst_RX_CLK),
        .hps_0_hps_io_hps_io_emac1_inst_RXD1   (hps_io_hps_io_emac1_inst_RXD1),
        .hps_0_hps_io_hps_io_emac1_inst_RXD2   (hps_io_hps_io_emac1_inst_RXD2),
        .hps_0_hps_io_hps_io_emac1_inst_RXD3   (hps_io_hps_io_emac1_inst_RXD3),
        .hps_0_hps_io_hps_io_sdio_inst_CMD     (hps_io_hps_io_sdio_inst_CMD),
        .hps_0_hps_io_hps_io_sdio_inst_D0      (hps_io_hps_io_sdio_inst_D0),
        .hps_0_hps_io_hps_io_sdio_inst_D1      (hps_io_hps_io_sdio_inst_D1),
        .hps_0_hps_io_hps_io_sdio_inst_CLK     (hps_io_hps_io_sdio_inst_CLK),
        .hps_0_hps_io_hps_io_sdio_inst_D2      (hps_io_hps_io_sdio_inst_D2),
        .hps_0_hps_io_hps_io_sdio_inst_D3      (hps_io_hps_io_sdio_inst_D3),
        .hps_0_hps_io_hps_io_usb1_inst_D0      (hps_io_hps_io_usb1_inst_D0),
        .hps_0_hps_io_hps_io_usb1_inst_D1      (hps_io_hps_io_usb1_inst_D1),
        .hps_0_hps_io_hps_io_usb1_inst_D2      (hps_io_hps_io_usb1_inst_D2),
        .hps_0_hps_io_hps_io_usb1_inst_D3      (hps_io_hps_io_usb1_inst_D3),
        .hps_0_hps_io_hps_io_usb1_inst_D4      (hps_io_hps_io_usb1_inst_D4),
        .hps_0_hps_io_hps_io_usb1_inst_D5      (hps_io_hps_io_usb1_inst_D5),
        .hps_0_hps_io_hps_io_usb1_inst_D6      (hps_io_hps_io_usb1_inst_D6),
        .hps_0_hps_io_hps_io_usb1_inst_D7      (hps_io_hps_io_usb1_inst_D7),
        .hps_0_hps_io_hps_io_usb1_inst_CLK     (hps_io_hps_io_usb1_inst_CLK),
        .hps_0_hps_io_hps_io_usb1_inst_STP     (hps_io_hps_io_usb1_inst_STP),
        .hps_0_hps_io_hps_io_usb1_inst_DIR     (hps_io_hps_io_usb1_inst_DIR),
        .hps_0_hps_io_hps_io_usb1_inst_NXT     (hps_io_hps_io_usb1_inst_NXT),
        .hps_0_hps_io_hps_io_spim1_inst_CLK    (hps_io_hps_io_spim1_inst_CLK),
        .hps_0_hps_io_hps_io_spim1_inst_MOSI   (hps_io_hps_io_spim1_inst_MOSI),
        .hps_0_hps_io_hps_io_spim1_inst_MISO   (hps_io_hps_io_spim1_inst_MISO),
        .hps_0_hps_io_hps_io_spim1_inst_SS0    (hps_io_hps_io_spim1_inst_SS0),
        .hps_0_hps_io_hps_io_uart0_inst_RX     (hps_io_hps_io_uart0_inst_RX),
        .hps_0_hps_io_hps_io_uart0_inst_TX     (hps_io_hps_io_uart0_inst_TX),
        .hps_0_hps_io_hps_io_i2c0_inst_SDA     (hps_io_hps_io_i2c0_inst_SDA),
        .hps_0_hps_io_hps_io_i2c0_inst_SCL     (hps_io_hps_io_i2c0_inst_SCL),
        .hps_0_hps_io_hps_io_i2c1_inst_SDA     (hps_io_hps_io_i2c1_inst_SDA),
        .hps_0_hps_io_hps_io_i2c1_inst_SCL     (hps_io_hps_io_i2c1_inst_SCL),
        .hps_0_hps_io_hps_io_gpio_inst_GPIO09  (hps_io_hps_io_gpio_inst_GPIO09),
        .hps_0_hps_io_hps_io_gpio_inst_GPIO35  (hps_io_hps_io_gpio_inst_GPIO35),
        .hps_0_hps_io_hps_io_gpio_inst_GPIO40  (hps_io_hps_io_gpio_inst_GPIO40),
        .hps_0_hps_io_hps_io_gpio_inst_GPIO53  (hps_io_hps_io_gpio_inst_GPIO53),
        .hps_0_hps_io_hps_io_gpio_inst_GPIO54  (hps_io_hps_io_gpio_inst_GPIO54),
        .hps_0_hps_io_hps_io_gpio_inst_GPIO61  (hps_io_hps_io_gpio_inst_GPIO61),
        .led_pio_export                          (leds),
        .memory_mem_a                            (hps_memory_mem_a),
        .memory_mem_ba                           (hps_memory_mem_ba),
        .memory_mem_ck                           (hps_memory_mem_ck),
        .memory_mem_ck_n                         (hps_memory_mem_ck_n),
```

```
175         .memory_mem_cke                         (hps_memory_mem_cke),
176         .memory_mem_cs_n                        (hps_memory_mem_cs_n),
177         .memory_mem_ras_n                       (hps_memory_mem_ras_n),
178         .memory_mem_cas_n                       (hps_memory_mem_cas_n),
179         .memory_mem_we_n                        (hps_memory_mem_we_n),
180         .memory_mem_reset_n                     (hps_memory_mem_reset_n),
181         .memory_mem_dq                          (hps_memory_mem_dq),
182         .memory_mem_dqs                         (hps_memory_mem_dqs),
183         .memory_mem_dqs_n                       (hps_memory_mem_dqs_n),
184         .memory_mem_odt                         (hps_memory_mem_odt),
185         .memory_mem_dm                          (hps_memory_mem_dm),
186         .memory_oct_rzqin                       (hps_memory_oct_rzqin),
187         .reset_reset_n                          (qsys_system_reset),
188         .switch_pio_export                      (switch)
189 );
190
191 endmodule
```

Step 2. Click the **Start Analysis & Synthesis** button on the Intel Quartus software toolbar to have it process the new HDL. This process will identify any errors in the HDL source files or how they are configured in the project as well as inform the Intel Quartus software of the new top level ports that we declared in our top module and it will synthesize the design netlist so that the SDRAM pin assignments script can be run in a future step.

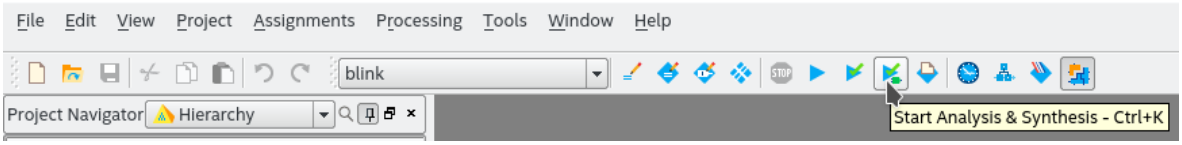


Figure 36: Start Analysis and Synthesis Button on Intel Quartus Software Toolbar

Step 3. The new top level module declares many more inputs and outputs than the First Qsys Design, so new pin constraints will need to be assigned to those pins. The Terasic DE10-Nano user manual contains the diagrams and schematics where these pin assignments are derived from. To make this process less error prone and faster we will copy the TCL commands below into a TCL script file that we name **hps_pin_assignments.tcl** that we create in the top level of the Intel Quartus software project directory:

Download the hps_pin_assignments.tcl file here.
If you wish to view the hps_pin_assignments.tcl file in the GitHub repo you can look here.

```
1  #
2  # Copyright (c) 2017 Intel Corporation
3  #
4  # Permission is hereby granted, free of charge, to any person obtaining a copy
5  # of this software and associated documentation files (the "Software"), to
6  # deal in the Software without restriction, including without limitation the
7  # rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
8  # sell copies of the Software, and to permit persons to whom the Software is
9  # furnished to do so, subject to the following conditions:
10 #
11 # The above copyright notice and this permission notice shall be included in
12 # all copies or substantial portions of the Software.
13 #
14 # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
15 # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
16 # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
17 # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
18 # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
19 # FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
20 # IN THE SOFTWARE.
21 #
```

```
22
23  # make the pin location assignments for the HPS, we only need the DRAM interface
24  # pins, not the HPS peripheral IO pins
25  set_location_assignment PIN_C28  -to hps_memory_mem_a[0]
26  set_location_assignment PIN_B28  -to hps_memory_mem_a[1]
27  set_location_assignment PIN_E26  -to hps_memory_mem_a[2]
28  set_location_assignment PIN_D26  -to hps_memory_mem_a[3]
29  set_location_assignment PIN_J21  -to hps_memory_mem_a[4]
30  set_location_assignment PIN_J20  -to hps_memory_mem_a[5]
31  set_location_assignment PIN_C26  -to hps_memory_mem_a[6]
32  set_location_assignment PIN_B26  -to hps_memory_mem_a[7]
33  set_location_assignment PIN_F26  -to hps_memory_mem_a[8]
34  set_location_assignment PIN_F25  -to hps_memory_mem_a[9]
35  set_location_assignment PIN_A24  -to hps_memory_mem_a[10]
36  set_location_assignment PIN_B24  -to hps_memory_mem_a[11]
37  set_location_assignment PIN_D24  -to hps_memory_mem_a[12]
38  set_location_assignment PIN_C24  -to hps_memory_mem_a[13]
39  set_location_assignment PIN_G23  -to hps_memory_mem_a[14]
40  set_location_assignment PIN_A27  -to hps_memory_mem_ba[0]
41  set_location_assignment PIN_H25  -to hps_memory_mem_ba[1]
42  set_location_assignment PIN_G25  -to hps_memory_mem_ba[2]
43  set_location_assignment PIN_A26  -to hps_memory_mem_cas_n
44  set_location_assignment PIN_N21  -to hps_memory_mem_ck
45  set_location_assignment PIN_N20  -to hps_memory_mem_ck_n
46  set_location_assignment PIN_L28  -to hps_memory_mem_cke
47  set_location_assignment PIN_L21  -to hps_memory_mem_cs_n
48  set_location_assignment PIN_G28  -to hps_memory_mem_dm[0]
49  set_location_assignment PIN_P28  -to hps_memory_mem_dm[1]
50  set_location_assignment PIN_W28  -to hps_memory_mem_dm[2]
51  set_location_assignment PIN_AB28 -to hps_memory_mem_dm[3]
52  set_location_assignment PIN_J25  -to hps_memory_mem_dq[0]
53  set_location_assignment PIN_J24  -to hps_memory_mem_dq[1]
54  set_location_assignment PIN_E28  -to hps_memory_mem_dq[2]
55  set_location_assignment PIN_D27  -to hps_memory_mem_dq[3]
56  set_location_assignment PIN_J26  -to hps_memory_mem_dq[4]
57  set_location_assignment PIN_K26  -to hps_memory_mem_dq[5]
58  set_location_assignment PIN_G27  -to hps_memory_mem_dq[6]
59  set_location_assignment PIN_F28  -to hps_memory_mem_dq[7]
60  set_location_assignment PIN_K25  -to hps_memory_mem_dq[8]
61  set_location_assignment PIN_L25  -to hps_memory_mem_dq[9]
62  set_location_assignment PIN_J27  -to hps_memory_mem_dq[10]
63  set_location_assignment PIN_J28  -to hps_memory_mem_dq[11]
64  set_location_assignment PIN_M27  -to hps_memory_mem_dq[12]
65  set_location_assignment PIN_M26  -to hps_memory_mem_dq[13]
66  set_location_assignment PIN_M28  -to hps_memory_mem_dq[14]
67  set_location_assignment PIN_N28  -to hps_memory_mem_dq[15]
68  set_location_assignment PIN_N24  -to hps_memory_mem_dq[16]
69  set_location_assignment PIN_N25  -to hps_memory_mem_dq[17]
70  set_location_assignment PIN_T28  -to hps_memory_mem_dq[18]
71  set_location_assignment PIN_U28  -to hps_memory_mem_dq[19]
72  set_location_assignment PIN_N26  -to hps_memory_mem_dq[20]
73  set_location_assignment PIN_N27  -to hps_memory_mem_dq[21]
74  set_location_assignment PIN_R27  -to hps_memory_mem_dq[22]
75  set_location_assignment PIN_V27  -to hps_memory_mem_dq[23]
76  set_location_assignment PIN_R26  -to hps_memory_mem_dq[24]
77  set_location_assignment PIN_R25  -to hps_memory_mem_dq[25]
78  set_location_assignment PIN_AA28 -to hps_memory_mem_dq[26]
79  set_location_assignment PIN_W26  -to hps_memory_mem_dq[27]
80  set_location_assignment PIN_R24  -to hps_memory_mem_dq[28]
81  set_location_assignment PIN_T24  -to hps_memory_mem_dq[29]
82  set_location_assignment PIN_Y27  -to hps_memory_mem_dq[30]
83  set_location_assignment PIN_AA27 -to hps_memory_mem_dq[31]
84  set_location_assignment PIN_R17  -to hps_memory_mem_dqs[0]
85  set_location_assignment PIN_R19  -to hps_memory_mem_dqs[1]
86  set_location_assignment PIN_T19  -to hps_memory_mem_dqs[2]
87  set_location_assignment PIN_U19  -to hps_memory_mem_dqs[3]
88  set_location_assignment PIN_R16  -to hps_memory_mem_dqs_n[0]
89  set_location_assignment PIN_R18  -to hps_memory_mem_dqs_n[1]
90  set_location_assignment PIN_T18  -to hps_memory_mem_dqs_n[2]
91  set_location_assignment PIN_T20  -to hps_memory_mem_dqs_n[3]
92  set_location_assignment PIN_D28  -to hps_memory_mem_odt
```

```
set_location_assignment PIN_A25  -to hps_memory_mem_ras_n
set_location_assignment PIN_V28  -to hps_memory_mem_reset_n
set_location_assignment PIN_E25  -to hps_memory_mem_we_n
set_location_assignment PIN_D25  -to hps_memory_oct_rzqin

# define the IO standard for the HPS peripheral IO pins
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_MDC
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_MDIO
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RXD0
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RXD1
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RXD2
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RXD3
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RX_CLK
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RX_CTL
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TXD0
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TXD3
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TXD1
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TXD2
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TX_CLK
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TX_CTL
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_CLK
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_CMD
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_D0
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_D1
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_D2
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_D3
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_uart0_inst_RX
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_uart0_inst_TX
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_CLK
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D0
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D1
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D2
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D3
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D4
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D5
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D6
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D7
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_DIR
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_NXT
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_STP
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_spim1_inst_CLK
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_spim1_inst_MOSI
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_spim1_inst_MISO
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_spim1_inst_SS0
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_i2c0_inst_SDA
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_i2c0_inst_SCL
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_i2c1_inst_SDA
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_i2c1_inst_SCL
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO09
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO35
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO40
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO53
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO54
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO61

# define the current strength for the HPS peripheral IO pins
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_MDC
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_MDIO
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RXD0
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RXD1
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RXD2
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RXD3
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RX_CLK
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RX_CTL
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TXD0
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TXD1
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TXD2
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TXD3
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TX_CLK
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TX_CTL
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_CLK
```

```
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D0
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D1
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D2
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D3
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D4
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D5
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D6
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D7
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_DIR
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_NXT
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_STP
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_CLK
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_CMD
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_D0
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_D1
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_D2
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_D3
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_uart0_inst_RX
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_uart0_inst_TX
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_spim1_inst_CLK
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_spim1_inst_MOSI
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_spim1_inst_SS0
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_i2c0_inst_SDA
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_i2c0_inst_SCL
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_i2c1_inst_SDA
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_i2c1_inst_SCL
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO09
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO35
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO40
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO53
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO54
set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO61

# define the slew rate for the HPS peripheral IO pins and DRAM interface pins
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[0]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[1]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[2]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[3]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[4]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[5]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[6]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[7]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[8]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[9]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[10]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[11]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[12]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[13]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[14]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_ba[0]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_ba[1]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_ba[2]
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_cas_n
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_cke
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_cs_n
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_odt
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_ras_n
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_reset_n
set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_we_n
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_MDC
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_MDIO
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RXD0
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RXD1
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RXD2
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RXD3
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RX_CLK
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RX_CTL
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TXD0
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TXD1
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TXD2
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TXD3
```

```
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TX_CLK
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TX_CTL
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_CLK
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D0
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D1
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D2
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D3
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D4
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D5
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D6
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D7
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_DIR
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_NXT
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_STP
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_CLK
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_CMD
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_D0
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_D1
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_D2
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_D3
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_uart0_inst_RX
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_uart0_inst_TX
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_spim1_inst_CLK
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_spim1_inst_MOSI
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_spim1_inst_SS0
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_i2c0_inst_SDA
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_i2c0_inst_SCL
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_i2c1_inst_SDA
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_i2c1_inst_SCL
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO09
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO35
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO40
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO53
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO54
set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO61
```

Step 4. With the **hps_pin_assignments.tcl** file created, we can now run it. Begin by selecting the **TCL Scripts...** menu from the **Tools** menu as shown here:

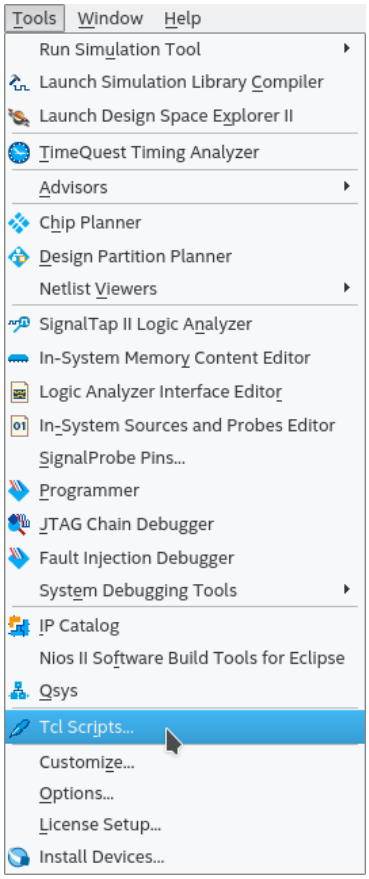


Figure 37: TCL Scripts Menu

Then locate the **hps_pin_assignments.tcl** script in the dialog and select it and click the **Run** button.

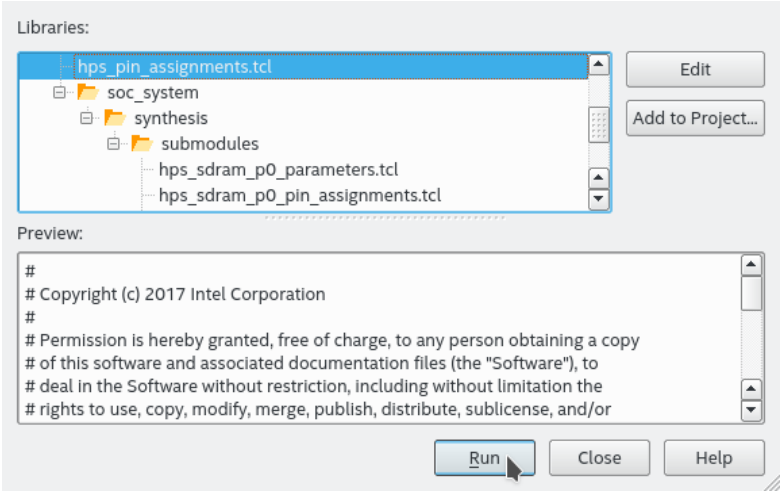


Figure 38: Run HPS Pin Assignments Script

Step 5. Next we need to run the **hps_sdram_p0_pin_assignments.tcl** script that was generated by Qsys. Begin by selecting the **TCL Scripts...** menu from the **Tools** menu as shown above, then locate the **hps_sdram_p0_pin_assignments.tcl** script in the dialog and select it and click the **Run** button.

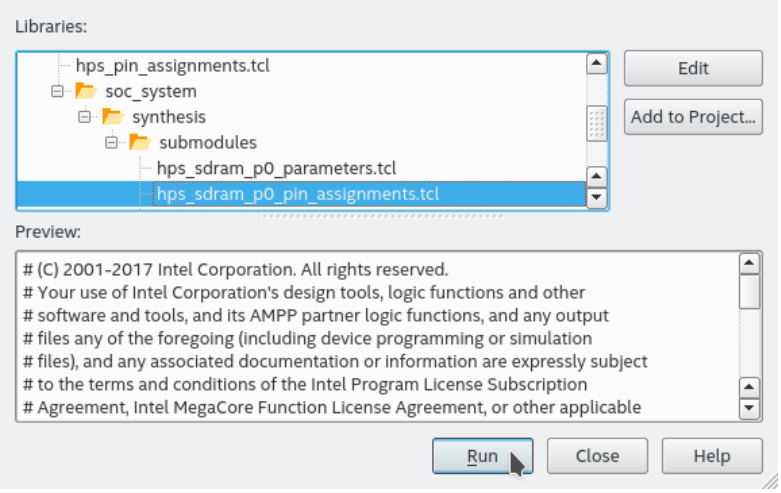


Figure 39: Run HPS SDRAM Pin Assignments Script

Step 6. Update the **blink.sdc** with the following code:

Download the blink.sdc file here.
If you wish to view the blink.sdc file in the GitHub repo you can look here.

```
#
# Copyright (c) 2017 Intel Corporation
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to
# deal in the Software without restriction, including without limitation the
# rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
# sell copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
# IN THE SOFTWARE.
#

# inform quartus that the clk port brings a 50MHz clock into our design so
# that timing closure on our design can be analyzed

create_clock -name clk -period "50MHz" [get_ports clk]
derive_clock_uncertainty

# inform quartus that the PIO inputs and outputs have no critical timing
# requirements.  These signals are driving LEDs and reading discrete push button
# and switch inputs, there are no timing relationships that are critical for any
# of this

set_false_path -from [get_ports {switch[0]}] -to *
set_false_path -from [get_ports {switch[1]}] -to *
set_false_path -from [get_ports {switch[2]}] -to *
set_false_path -from [get_ports {switch[3]}] -to *
```

```
set_false_path -from * -to [get_ports {leds[0]}]
set_false_path -from * -to [get_ports {leds[1]}]
set_false_path -from * -to [get_ports {leds[2]}]
set_false_path -from * -to [get_ports {leds[3]}]
set_false_path -from * -to [get_ports {leds[4]}]
set_false_path -from * -to [get_ports {leds[5]}]
set_false_path -from * -to [get_ports {leds[6]}]
set_false_path -from * -to [get_ports {leds[7]}]
set_false_path -from [get_ports {push_button[0]}] -to *
set_false_path -from [get_ports {push_button[1]}] -to *

# Define timing constraints for the JTAG IO pins so that Quartus properly closes
# timing on these signal paths.  Otherwise we could have unreliable JTAG
# communication with the device over the USB Blaster II connection.
# NOTE: the 'altera_reserved_tck' clock is automatically defined by Quartus

set_input_delay  -clock altera_reserved_tck -clock_fall 3 [get_ports {altera_reserved_tdi}]
set_input_delay  -clock altera_reserved_tck -clock_fall 3 [get_ports {altera_reserved_tms}]
set_output_delay -clock altera_reserved_tck             3 [get_ports {altera_reserved_tdo}]

# Define clocks for the HPS ports that expose clock signals to avoid
# unconstrained clock warnings
create_clock -period "1 MHz" [get_ports {hps_io_hps_io_i2c0_inst_SCL}]
create_clock -period "1 MHz" [get_ports {hps_io_hps_io_i2c1_inst_SCL}]
create_clock -period "48 MHz" [get_ports {hps_io_hps_io_usb1_inst_CLK}]
```

Step 7. Click the **Start Compilation** button in the top toolbar to compile the entire design.

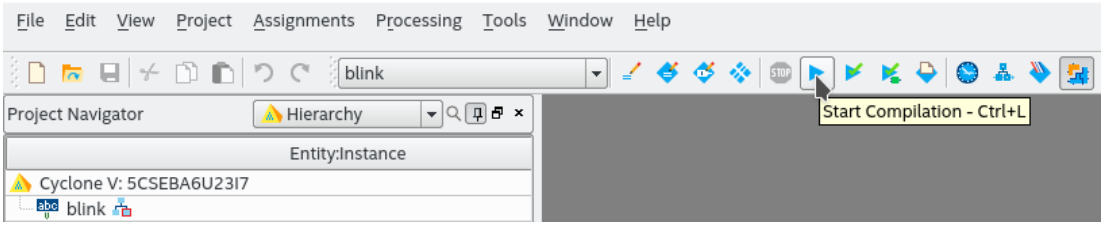


Figure 40: Start Compilation Button on Intel Quartus Software Toolbar

Step 8. After your design has successfully compiled in the Intel Quartus software, you need to generate the Raw Binary File (RBF) that is used to program the FPGA from the HPS processor. Begin by selecting the **File** menu and then select the **Convert Programming Files...** menu.

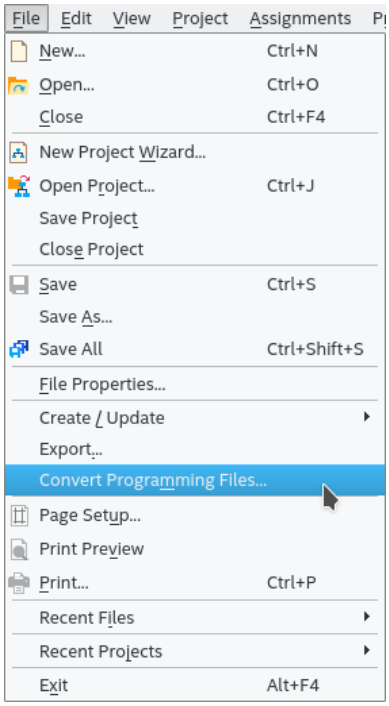


Figure 41: Convert Programming Files Menu

In the **Convert Programming Files** dialog perform the following actions:

Step 8a. Change the **Programming file type** setting to **Raw Binary File(.rbf)**.

Step 8b. Change the **File name** setting to **output_files/blink.rbf**.

Step 8c. Select the **SOF Data** entry in the **Input files to convert** table.

Step 8d. Click the **Add File...** button.

Step 8e. Browse to the **blink.sof** in the **output_files** directory and select it.

Step 8f. Click the open button to close the file browser dialog and add that file to the list.

Step 8g. Click the **Generate** button to create the RBF file.

Step 8h. Close the **Convert Programming Files** dialog.

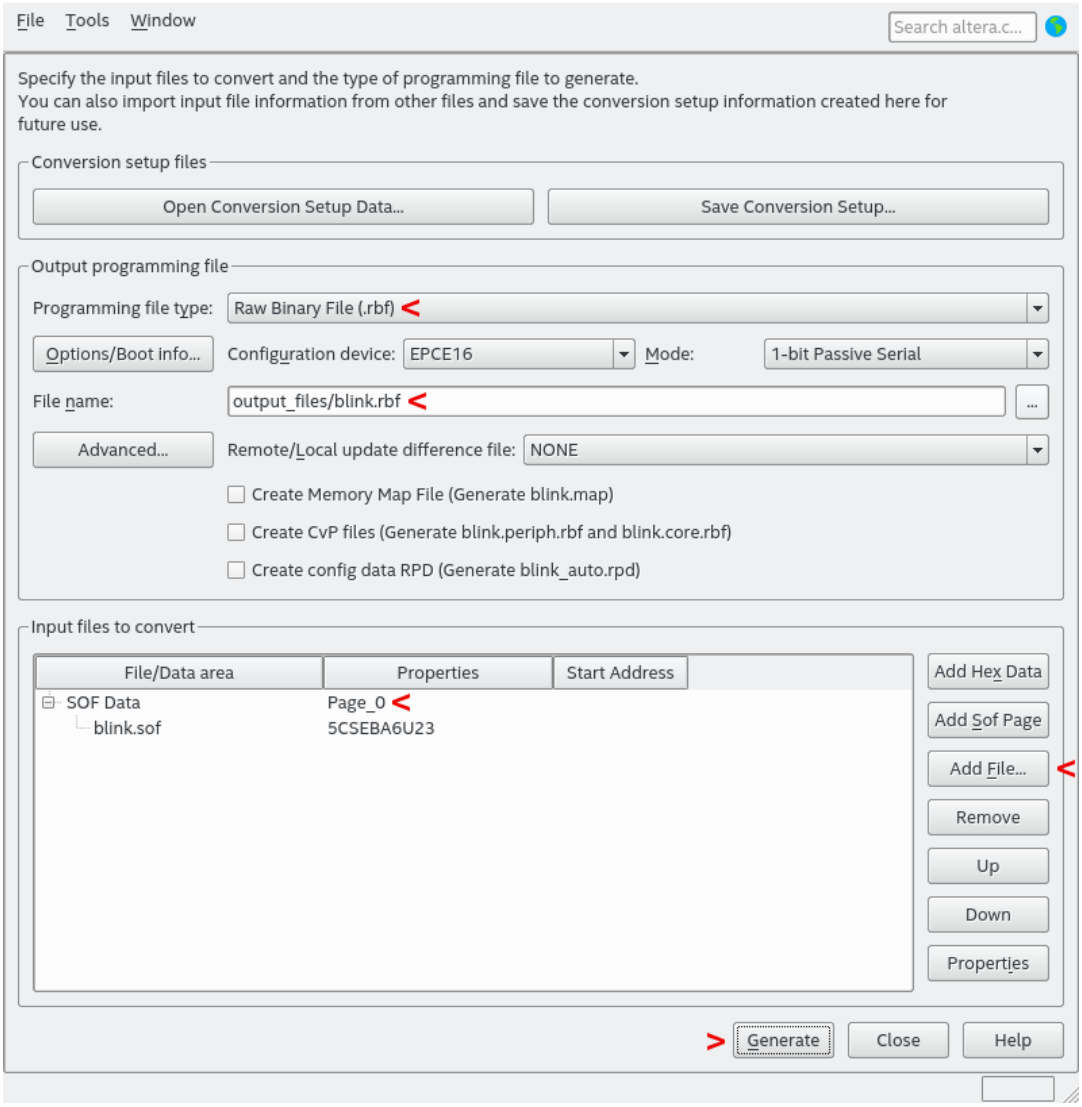


Figure 42: Convert Programming Files Menu

Step 9. Now let's perform one last activity in the Intel Quartus software to prepare for the next tutorial that will actually make use of this system on the Terasic DE10-Nano board. Since we have defined a memory mapped embedded system in Qsys with a couple HPS masters connected to a number of slave peripherals, it will be necessary for us to know what the base addresses are of the slave peripherals so we can interact with them, performing read and write transactions to them. That base address information is captured in Qsys, you can visualize that in Qsys a number of ways but that is not convenient for software developers or other users of

this system to write code for it. Each time you generate a Qsys system Qsys outputs a database file called <**your-system-name**>**.sopcinfo**. The Intel Quartus software tools installation provides a utility that can be used to translate the SOPCINFO database information into a usable macro format that can be used for various purposes called **sopc-create-header-files**. The default functionality of **sopc-create-header-files** is to create C style header macros from each masters perspective in the Qsys system. We will perform this operation in the Intel Quartus software TCL Console which is located in the lower middle of the default Intel Quartus software GUI. If you do not see the TCL Console pane, you can open it by selecting the **View** > **Utility Windows** > **TCL Console** menu.

Figure 43: Intel Quartus Prime Software TCL Console

We will first create a directory to output the header files into, then we will create the default header file output using **sopc-create-header-files** and finally we will extract the base address entries out of the HPS masters header file for the FPGA peripherals that they are connected to. We will perform all of this with the following TCL commands:

```
Quartus Prime Tcl Console

# make a directory called 'qsys_headers' to store the header files
tcl> file mkdir qsys_headers

# create a TCL variable SCHF_PATH to hold the path to the executable program
# sopc-create-header-files on your host PC using the environment variables
# provided by Quartus.
tcl> set SCHF_PATH [glob -join $quartus(quartus_rootpath) sopc_builder bin sopc-create-header-files]

# create a TCL variable BAT_PATH to hold the path to the Nios II Command Shell
# batch file on Windows platforms.  The following code sequence will work on
# either Windows or Linux.  For linux this variable will just be set to NULL.
tcl> set BAT_PATH {}
tcl> if {$tcl_platform(platform) == "windows"} {
   > set BAT_PATH [glob -join $quartus(quartus_rootpath) .. nios2eds {Nios II Command Shell.bat}]
   > }

# execute sopc-create-header-files to generate the header files
tcl> eval exec -ignorestderr ${BAT_PATH} ${SCHF_PATH} soc_system.sopcinfo --output-dir qsys_headers

# read the header file for hps_0_arm_a9_0 into a TCL variable
tcl> set hps_0_arm_a9_0_header [read [open [glob -join qsys_headers hps_0_arm_a9_0.h] r]]

# output the C macro lines for the FPGA peripheral base addresses
tcl> foreach line [split ${hps_0_arm_a9_0_header} "\n"] { \
   > if {[string match "*_BASE*" ${line}]} { \
   > if {! [string match "*HPS_*" ${line}]} {puts ${line}}}}
```

If you have put your Qsys system together properly and executed the above commands correctly, you should see the following output in the Intel Quartus software TCL console representing the base address definitions for the five Qsys peripherals in the FPGA fabric connected to the HPS H2F and LWH2F bridges.

```
#define OCRAM_64K_BASE 0xc0000000
#define LED_PIO_BASE 0xff210000
#define BUTTON_PIO_BASE 0xff210010
#define SWITCH_PIO_BASE 0xff210020
#define SYSTEM_ID_BASE 0xff210030
```

That's it! You have designed and compiled your first HPS system. Continue on to the "Interacting with FPGA Designs Using U-Boot" tutorial where we demonstrate how to use U-Boot to program the FPGA and interact with this design. Or you can continue on to the "Interacting with FPGA Designs Using Linux" tutorial where we demonstrate how to interact with this design from Linux.