

My First HPS System

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

Contents

Overview	1
Prerequisites	1
Instantiating the Hard Processor System	2
Integrating the Qsys System into the Intel Quartus Software Project	16

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](#).

Intel, and Quartus are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
Other names and brands may be claimed as the property of others.

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.

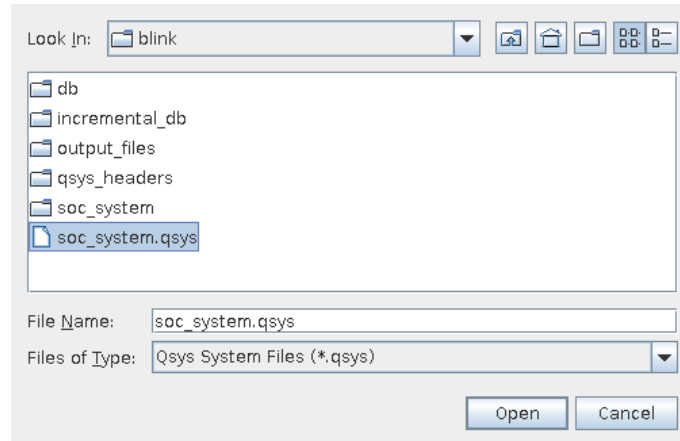


Figure 1: Qsys Open Dialog

- 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.

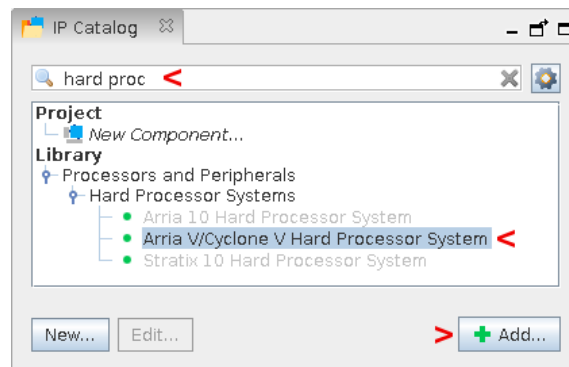


Figure 2: Hard Processor Search

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.

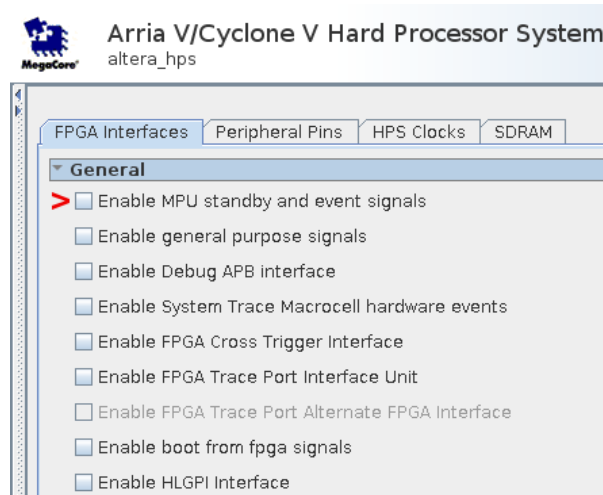


Figure 3: FPGA Interfaces - General

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.

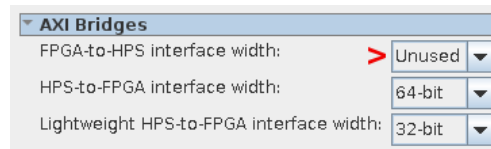


Figure 4: AXI Bridges

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.

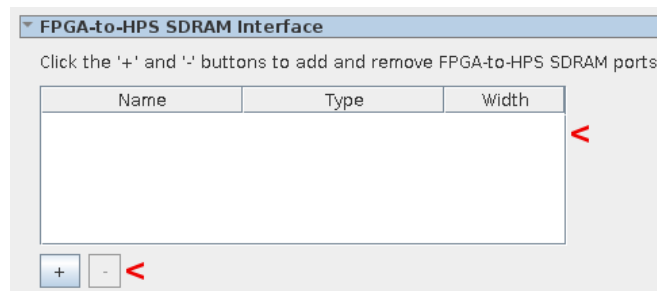
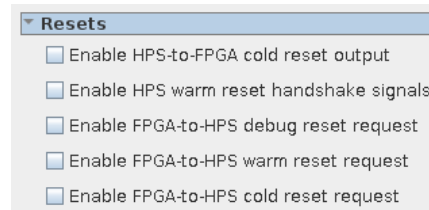


Figure 5: FPGA-to-HPS SDRAM Interface

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

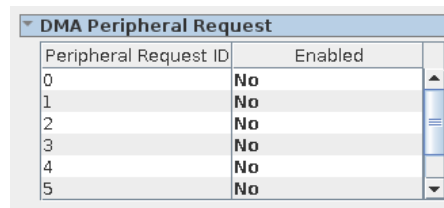


Resets

- ☐ Enable HPS-to-FPGA cold reset output
- ☐ Enable HPS warm reset handshake signals
- ☐ Enable FPGA-to-HPS debug reset request
- ☐ Enable FPGA-to-HPS warm reset request
- ☐ Enable FPGA-to-HPS cold reset request

Figure 6: Resets

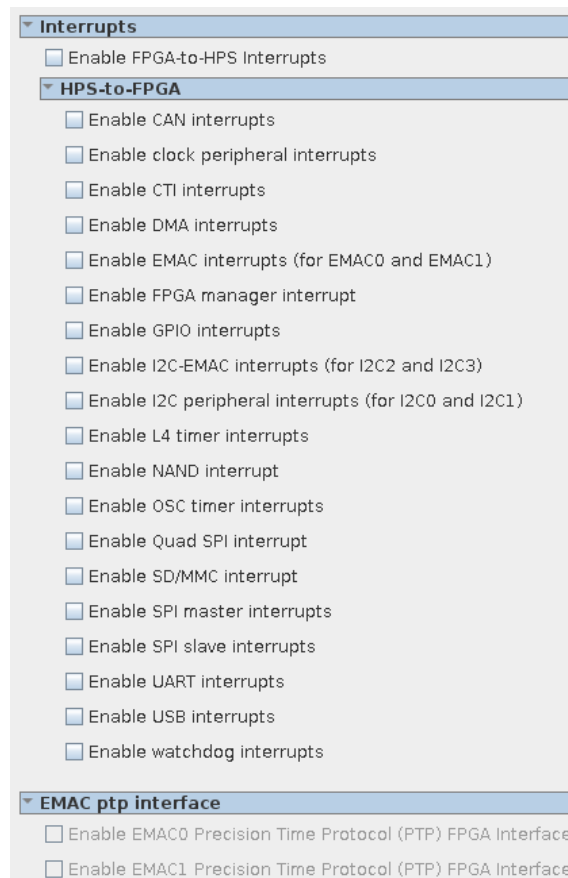
Step 7. In the **DMA Peripheral Request** section, do not change any of the default settings.



Peripheral Request ID	Enabled
0	No
1	No
2	No
3	No
4	No
5	No

Figure 7: DMA Peripheral Request

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



Interrupts

☐ Enable FPGA-to-HPS Interrupts

HPS-to-FPGA

- ☐ Enable CAN interrupts
- ☐ Enable clock peripheral interrupts
- ☐ Enable CTI interrupts
- ☐ Enable DMA interrupts
- ☐ Enable EMAC interrupts (for EMAC0 and EMAC1)
- ☐ Enable FPGA manager interrupt
- ☐ Enable GPIO interrupts
- ☐ Enable I2C-EMAC interrupts (for I2C2 and I2C3)
- ☐ Enable I2C peripheral interrupts (for I2C0 and I2C1)
- ☐ Enable L4 timer interrupts
- ☐ Enable NAND interrupt
- ☐ Enable OSC timer interrupts
- ☐ Enable Quad SPI interrupt
- ☐ Enable SD/MMC interrupt
- ☐ Enable SPI master interrupts
- ☐ Enable SPI slave interrupts
- ☐ Enable UART interrupts
- ☐ Enable USB interrupts
- ☐ Enable watchdog interrupts

EMAC ptp interface

- ☐ Enable EMAC0 Precision Time Protocol (PTP) FPGA Interface
- ☐ Enable EMAC1 Precision Time Protocol (PTP) FPGA Interface

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.

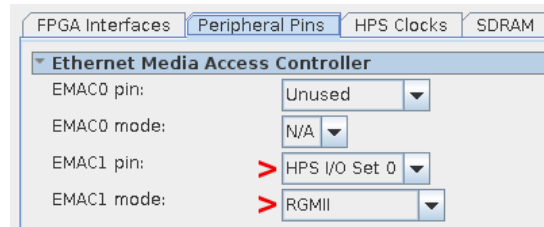


Figure 9: Ethernet Media Access Controller

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

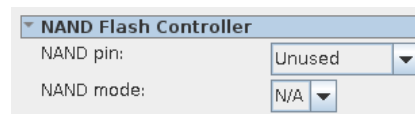


Figure 10: NAND Flash Controller

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

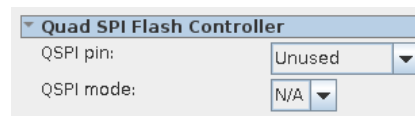


Figure 11: Quad SPI Flash Controller

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.

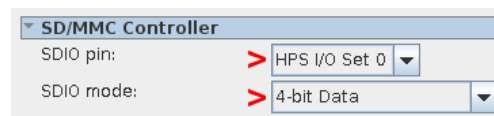


Figure 12: SD/MMC Controller

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.

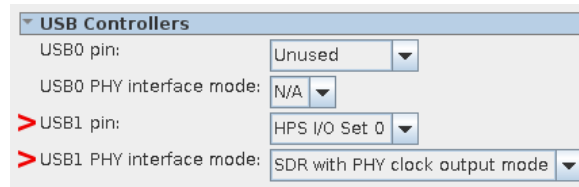


Figure 13: USB Controllers

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.

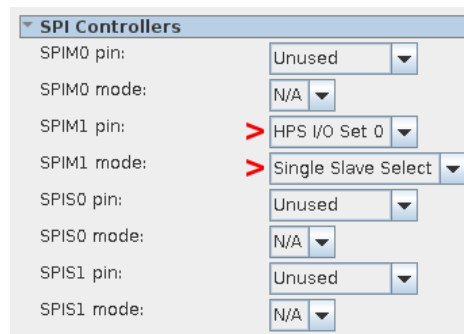


Figure 14: SPI Controllers

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.

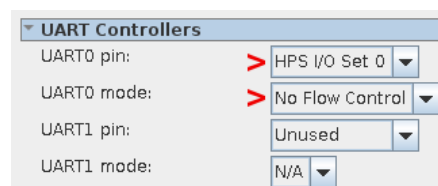


Figure 15: UART Controllers

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

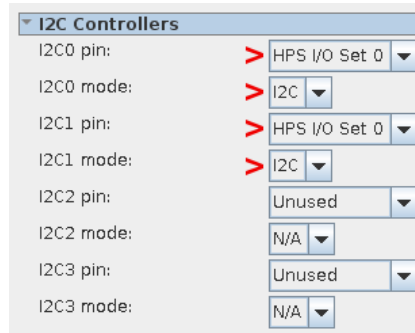
Step 16a. Select **HPS I/O Set 0** for the **I2C0 pin** parameter.

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.

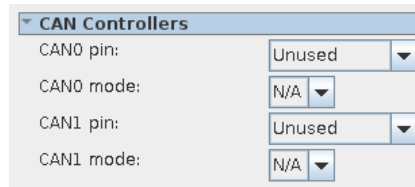


The screenshot shows the 'I2C Controllers' configuration window. It contains a list of parameters for I2C0, I2C1, I2C2, and I2C3. Red arrows point to the 'I2C0 pin', 'I2C0 mode', 'I2C1 pin', and 'I2C1 mode' settings, indicating the changes made in steps 16a through 16d.

I2C Controllers	
I2C0 pin:	HPS I/O Set 0
I2C0 mode:	I2C
I2C1 pin:	HPS I/O Set 0
I2C1 mode:	I2C
I2C2 pin:	Unused
I2C2 mode:	N/A
I2C3 pin:	Unused
I2C3 mode:	N/A

Figure 16: I2C Controllers

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



The screenshot shows the 'CAN Controllers' configuration window. It contains a list of parameters for CAN0 and CAN1. The settings are: CAN0 pin: Unused, CAN0 mode: N/A, CAN1 pin: Unused, and CAN1 mode: N/A.

CAN Controllers	
CAN0 pin:	Unused
CAN0 mode:	N/A
CAN1 pin:	Unused
CAN1 mode:	N/A

Figure 17: CAN Controllers

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



The screenshot shows the 'Trace Port Interface Unit' configuration window. It contains a list of parameters for the TRACE pin and mode. The settings are: TRACE pin: Unused and TRACE mode: N/A.

Trace Port Interface Unit	
TRACE pin:	Unused
TRACE mode:	N/A

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

Peripherals Mux Table					
RGMI0_TX_CLK			EMAC0_TX_CLK (Set0)	GPIO00	LOANID00
RGMI0_TXD0		USB1_D0 (Set0)	EMAC0_TXD0 (Set0)	GPIO01	LOANID01
RGMI0_TXD1		USB1_D1 (Set0)	EMAC0_TXD1 (Set0)	GPIO02	LOANID02
RGMI0_TXD2		USB1_D2 (Set0)	EMAC0_TXD2 (Set0)	GPIO03	LOANID03
RGMI0_TXD3		USB1_D3 (Set0)	EMAC0_TXD3 (Set0)	GPIO04	LOANID04
RGMI0_TXD0		USB1_D4 (Set0)	EMAC0_TXD0 (Set0)	GPIO05	LOANID05
RGMI0_MDIO	I2C2_SDA (Set0)	USB1_D5 (Set0)	EMAC0_MDIO (Set0)	GPIO06	LOANID06
RGMI0_MDC	I2C2_SCL (Set0)	USB1_D6 (Set0)	EMAC0_MDC (Set0)	GPIO07	LOANID07
RGMI0_RX_CTL		USB1_D7 (Set0)	EMAC0_RX_CTL (Set0)	GPIO08	LOANID08
RGMI0_TX_CTL			EMAC0_TX_CTL (Set0)	GPIO09	LOANID09
RGMI0_RX_CLK		USB1_CLK (Set0)	EMAC0_RX_CLK (Set0)	GPIO10	LOANID10
OSPI_CLK			OSPI_CLK (Set0)	GPIO34	LOANID34
QSPI_SS1			QSPI_SS1 (Set0)	GPIO35	LOANID35
SDMMC_CMD		USB0_D0 (Set0)	SDIO_CMD (Set0)	GPIO36	LOANID36
SDMMC_D1		USB0_D1 (Set0)	SDIO_D1 (Set0)	GPIO39	LOANID39
SDMMC_D4		USB0_D4 (Set0)	SDIO_D4 (Set0)	GPIO40	LOANID40
SDMMC_D5		USB0_D5 (Set0)	SDIO_D5 (Set0)	GPIO41	LOANID41
TRACE_D3	I2C1_SCL (Set0)	SPB0_S00 (Set0)	TRACE_D3 (Set0)	GPIO52	LOANID52
TRACE_D4	CAN1_RX (Set0)	SPB1_CLK (Set0)	TRACE_D4 (Set0)	GPIO53	LOANID53
TRACE_D5	CAN1_TX (Set0)	SPB1_M00 (Set0)	TRACE_D5 (Set0)	GPIO54	LOANID54
TRACE_D6	I2C0_SDA (Set0)	SPB1_S00 (Set0)	TRACE_D6 (Set0)	GPIO55	LOANID55
SPIM0_S00	UART1_RTS (Set0)	CAN1_TX (Set1)	SPIM0_S00 (Set0)	GPIO60	LOANID60
UART0_RX	SPIM0_SS1 (Set0)	CAN0_RX (Set0)	UART0_RX (Set1)	GPIO61	LOANID61
UART0_TX	SPIM1_SS1 (Set0)	CAN0_TX (Set0)	UART0_TX (Set1)	GPIO62	LOANID62
CAN0_TX	SPIM1_S00 (Set0)	UART0_TX (Set2)	CAN0_TX (Set1)	GPIO66	LOANID66

To enable a HPS pin to work as a Loan IO or as a GPIO pin, Click on the GPIO or Loan IO button on the Peripherals Mux table. The Specific peripherals are enabled in the Drop boxes above the Peripherals Mux table.

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.

FPGA Interfaces
Peripheral Pins
HPS Clocks
SDRAM

Input Clocks
Output Clocks

External Clock Sources

EOSC1 clock frequency: 25.0 MHz
EOSC2 clock frequency: 25.0 MHz

Figure 20: External Clock Sources

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

FPGA-to-HPS PLL Reference Clocks

☐ Enable FPGA-to-HPS SDRAM PLL reference clock
☐ Enable FPGA-to-HPS peripheral PLL reference clock

FPGA-to-HPS SDRAM PLL reference clock frequency: 0.0 MHz
FPGA-to-HPS peripheral PLL reference clock frequency: 0.0 MHz

Figure 21: FPGA-to-HPS PLL Reference Clocks

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

Peripheral FPGA Clocks		
EMAC0 emac0_md_clk clock frequency:	2.5	MHz
EMAC0 emac0_gtx_clk clock frequency:	125	MHz
EMAC1 emac1_md_clk clock frequency:	2.5	MHz
EMAC1 emac1_gtx_clk clock frequency:	125	MHz
QSPI qspi_sclk_out clock frequency:	100	MHz
SPIM0 spim0_sclk_out clock frequency:	100	MHz
SPIM1 spim1_sclk_out clock frequency:	100	MHz
I2C0 i2c0_clk clock frequency:	100	MHz
I2C1 i2c1_clk clock frequency:	100	MHz
I2C2 i2c2_clk clock frequency:	100	MHz
I2C3 i2c3_clk clock frequency:	100	MHz

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.

Input Clocks		Output Clocks
Clock Sources		
Peripheral PLL reference clock source:	EOSC1 clock	
SDMMC clock source:	Peripheral NAND SDMMC clock	
NAND clock source:	Peripheral NAND SDMMC clock	
QSPI clock source:	Main QSPI clock	
L4 MP clock source:	Peripheral base clock	
L4 SP clock source:	Peripheral base clock	

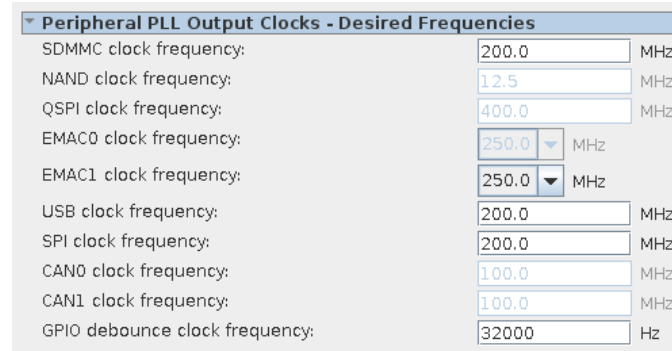
Figure 23: Clock Sources

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

Main PLL Output Clocks - Desired Frequencies		
Default MPU clock frequency:	800.0	MHz
<input checked="" type="checkbox"/> Use default MPU clock frequency		
MPU clock frequency:	800.0	MHz
L3 MP clock frequency:	200.0	MHz
L3 SP clock frequency:	100.0	MHz
Debug AT clock frequency:	25.0	MHz
Debug clock frequency:	12.5	MHz
Debug trace clock frequency:	25.0	MHz
L4 MP clock frequency:	100.0	MHz
L4 SP clock frequency:	100.0	MHz
Configuration/HPS-to-FPGA user 0 clock frequency:	100.0	MHz

Figure 24: Main PLL Output Clocks

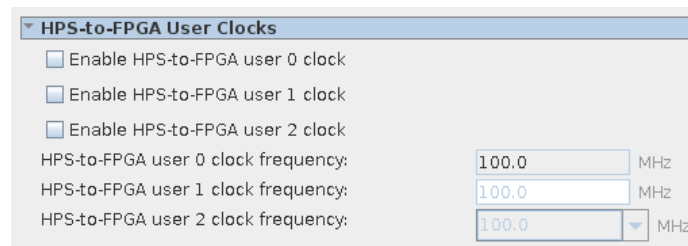
Step 25. In the **Peripheral PLL Output Clocks** section, do not change any of the default settings.



Peripheral PLL Output Clocks - Desired Frequencies	
SDMMC clock frequency:	200.0 MHz
NAND clock frequency:	12.5 MHz
QSPI clock frequency:	400.0 MHz
EMAC0 clock frequency:	250.0 MHz
EMAC1 clock frequency:	250.0 MHz
USB clock frequency:	200.0 MHz
SPI clock frequency:	200.0 MHz
CAN0 clock frequency:	100.0 MHz
CAN1 clock frequency:	100.0 MHz
GPIO debounce clock frequency:	32000 Hz

Figure 25: Peripheral PLL Output Clocks

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



HPS-to-FPGA User Clocks	
<input type="checkbox"/> Enable HPS-to-FPGA user 0 clock	
<input type="checkbox"/> Enable HPS-to-FPGA user 1 clock	
<input type="checkbox"/> Enable HPS-to-FPGA user 2 clock	
HPS-to-FPGA user 0 clock frequency:	100.0 MHz
HPS-to-FPGA user 1 clock frequency:	100.0 MHz
HPS-to-FPGA user 2 clock frequency:	100.0 MHz

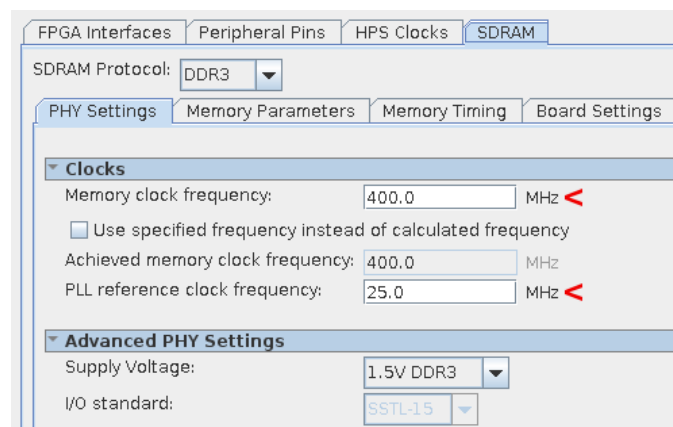
Figure 26: HPS-to-FPGA User Clocks

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.



SDRAM	
SDRAM Protocol: DDR3	
PHY Settings Memory Parameters Memory Timing Board Settings	
Clocks	
Memory clock frequency:	400.0 MHz
<input type="checkbox"/> Use specified frequency instead of calculated frequency	
Achieved memory clock frequency:	400.0 MHz
PLL reference clock frequency:	25.0 MHz
Advanced PHY Settings	
Supply Voltage:	1.5V DDR3
I/O standard:	SSTL-15

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**.

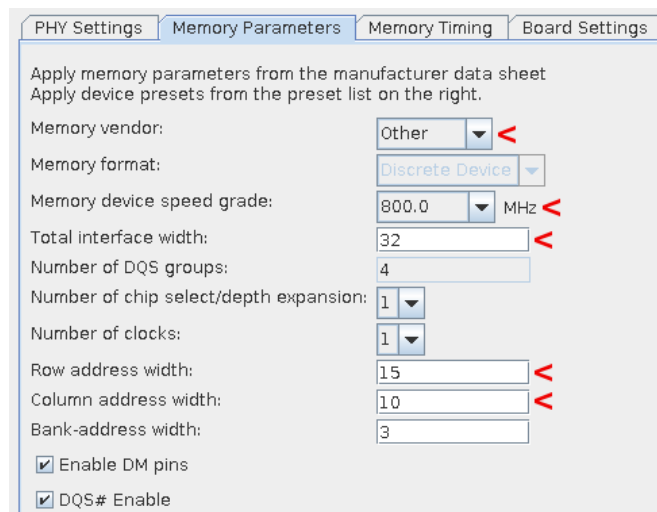
Step 28b. Change the **Memory device speed grade** parameter to **800.0**.

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.



The screenshot shows the 'Memory Parameters' sub-tab of a configuration tool. It contains several parameters with their current values and red arrows indicating changes:

- Memory vendor: Other
- Memory format: Discrete Device
- Memory device speed grade: 800.0 MHz
- Total interface width: 32
- Number of DQS groups: 4
- Number of chip select/depth expansion: 1
- Number of clocks: 1
- Row address width: 15
- Column address width: 10
- Bank-address width: 3
- Enable DM pins: ☒
- DQS# Enable: ☒

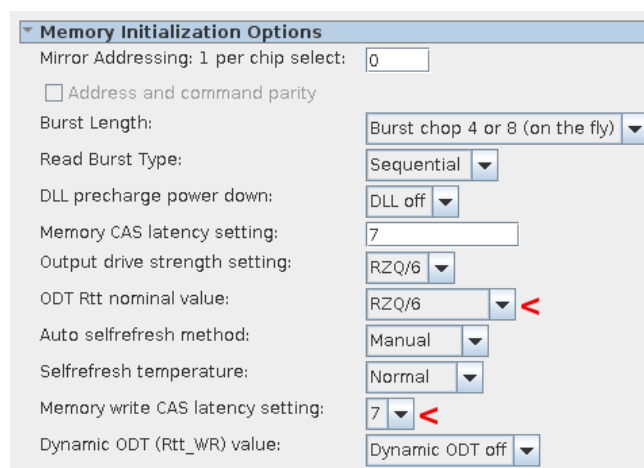
Figure 28: Memory Parameters

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.



The screenshot shows the 'Memory Initialization Options' section of a configuration tool. It contains several parameters with their current values and red arrows indicating changes:

- Mirror Addressing: 1 per chip select: 0
- Address and command parity: ☐
- Burst Length: Burst chop 4 or 8 (on the fly)
- Read Burst Type: Sequential
- DLL precharge power down: DLL off
- Memory CAS latency setting: 7
- Output drive strength setting: RZQ/6
- ODT Rtt nominal value: RZQ/6
- Auto selfrefresh method: Manual
- Selfrefresh temperature: Normal
- Memory write CAS latency setting: 7
- Dynamic ODT (Rtt_WR) value: Dynamic ODT off

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.

PHY Settings	Memory Parameters	Memory Timing	Board Settings
Apply timing parameters from the manufacturer data sheet Apply device presets from the preset list on the right.			
tIS (base):	175	ps	
tIH (base):	250	ps	
tDS (base):	50	ps	
tDH (base):	125	ps	
tDQSQ:	120	ps	
tQH:	0.38	cycles	
tDQSCK:	400	ps	
tDQSS:	0.25	cycles	
tQSH:	0.38	cycles	
tDSH:	0.2	cycles	
tDSS:	0.2	cycles	
tINIT:	500	us	<
tMRD (tMRW):	4	cycles	<
tRAS:	35.0	ns	<
tRCD:	13.75	ns	<
tRP:	13.75	ns	<
tREFI (tREFIab):	7.8	us	<
tRFC (tRFCab):	300.0	ns	<
tWR:	15.0	ns	
tWTR:	4	cycles	<
tFAW:	37.5	ns	
tRRD:	7.5	ns	
tRTP:	7.5	ns	

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.

PHY Settings | Memory Parameters | Memory Timing | **Board Settings**

Use the Board Settings to model the board-level effects in the timing analysis.

The wizard supports single- and multi-rank configurations. Altera has determined the effects on the output signaling of these configurations and has stored the effects on the output slew rate and the channel uncertainty within the UniPHY MegaWizard.

These values are representative of specific Altera boards. You must change the values to account for the board level effects for your board. You can use HyperLynx or similar simulators to obtain values that are representative of your board.

Setup and Hold Derating

The slew rate of the output signals affects the setup and hold times of the memory device.

You can specify the slew rate of the output signals to refer to their effect on the setup and hold times of both the address and command signals and the DQ signals, or specify the setup and hold times directly.

Derating method:

- ☒ Use Altera's default settings
- ☐ Specify slew rates to calculate setup and hold times
- ☐ Specify setup and hold times directly

CK/CK# slew rate (Differential):	2.0	V/ns
Address and command slew rate:	1.0	V/ns
DQS/DQS# slew rate (Differential):	2.0	V/ns
DQ slew rate:	1.0	V/ns
tIS:	0.325	ns
tIH:	0.35	ns
tDS:	0.2	ns
tDH:	0.225	ns

Figure 31: Setup and Hold Derating

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

Channel Signal Integrity

Channel Signal Integrity is a measure of the distortion of the eye due to intersymbol interference or crosstalk or other effects. Typically when going from a single-rank configuration to a multi-rank configuration there is an increase in the channel loss as there are multiple stubs causing reflections. Please perform your channel signal integrity simulations and enter the extra channel uncertainty as compared to Altera's reference eye diagram.

Derating Method:

- ☒ Use Altera's default settings
- ☐ Specify channel uncertainty values

Address and command eye reduction (setup):	0.0	ns
Address and command eye reduction (hold):	0.0	ns
Write DQ eye reduction:	0.0	ns
Write Delta DQS arrival time:	0.0	ns
Read DQ eye reduction:	0.0	ns
Read Delta DQS arrival time:	0.0	ns

Figure 32: Channel Signal Integrity

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

Board Skews

PCB traces can have skews between them that can cause timing margins to be reduced. Furthermore skews between different ranks can further reduce the timing margin in multi-rank topologies.

[Restore default values](#)

Maximum CK delay to DIMM/device:	0.6	ns
Maximum DQS delay to DIMM/device:	0.6	ns
Minimum delay difference between CK and DQS:	-0.01	ns
Maximum delay difference between CK and DQS:	0.01	ns
Maximum skew within DQS group:	0.02	ns
Maximum skew between DQS groups:	0.02	ns
Average delay difference between DQ and DQS:	0.0	ns
Maximum skew within address and command bus:	0.02	ns
Average delay difference between address and command and CK:	0.0	ns

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.

	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	unconnecte
	h2f_axi_master	AXI Master	Double-click to	[h2f_axi_...
	h2f_lw_axi_clock	Clock Input	Double-click to	unconnecte
	h2f_lw_axi_master	AXI Master	Double-click to	[h2f_lw_a...

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

```
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 // create module
24 module blink(
25     input wire      clk,          // 50MHz FPGA input clock
26
27     input wire [1:0] push_button, // KEY[1:0]
28     input wire [3:0] switch,      // SW[3:0]
29
30     output wire [7:0] leds,       // LED[7:0]
31
32     // HPS memory controller ports
33     output wire [14:0] hps_memory_mem_a,
34     output wire [2:0]  hps_memory_mem_ba,
35     output wire        hps_memory_mem_ck,
36     output wire        hps_memory_mem_ck_n,
37     output wire        hps_memory_mem_cke,
38     output wire        hps_memory_mem_cs_n,
39     output wire        hps_memory_mem_ras_n,
40     output wire        hps_memory_mem_cas_n,
41     output wire        hps_memory_mem_we_n,
42     output wire        hps_memory_mem_reset_n,
43     inout wire [31:0] hps_memory_mem_dq,
44     inout wire [3:0]  hps_memory_mem_dqs,
45     inout wire [3:0]  hps_memory_mem_dqs_n,
46     output wire        hps_memory_mem_odt,
47     output wire [3:0]  hps_memory_mem_dm,
48     input wire        hps_memory_oct_rzqin,
49
50     // HPS peripheral ports
51     output wire        hps_io_hps_io_emac1_inst_TX_CLK,
```



```

52     output wire      hps_io_hps_io_emac1_inst_TXD0,
53     output wire      hps_io_hps_io_emac1_inst_TXD1,
54     output wire      hps_io_hps_io_emac1_inst_TXD2,
55     output wire      hps_io_hps_io_emac1_inst_TXD3,
56     input  wire      hps_io_hps_io_emac1_inst_RXD0,
57     inout  wire      hps_io_hps_io_emac1_inst_MDIO,
58     output wire      hps_io_hps_io_emac1_inst_MDC,
59     input  wire      hps_io_hps_io_emac1_inst_RX_CTL,
60     output wire      hps_io_hps_io_emac1_inst_TX_CTL,
61     input  wire      hps_io_hps_io_emac1_inst_RX_CLK,
62     input  wire      hps_io_hps_io_emac1_inst_RXD1,
63     input  wire      hps_io_hps_io_emac1_inst_RXD2,
64     input  wire      hps_io_hps_io_emac1_inst_RXD3,
65     inout  wire      hps_io_hps_io_sdio_inst_CMD,
66     inout  wire      hps_io_hps_io_sdio_inst_D0,
67     inout  wire      hps_io_hps_io_sdio_inst_D1,
68     output wire      hps_io_hps_io_sdio_inst_CLK,
69     inout  wire      hps_io_hps_io_sdio_inst_D2,
70     inout  wire      hps_io_hps_io_sdio_inst_D3,
71     inout  wire      hps_io_hps_io_usb1_inst_D0,
72     inout  wire      hps_io_hps_io_usb1_inst_D1,
73     inout  wire      hps_io_hps_io_usb1_inst_D2,
74     inout  wire      hps_io_hps_io_usb1_inst_D3,
75     inout  wire      hps_io_hps_io_usb1_inst_D4,
76     inout  wire      hps_io_hps_io_usb1_inst_D5,
77     inout  wire      hps_io_hps_io_usb1_inst_D6,
78     inout  wire      hps_io_hps_io_usb1_inst_D7,
79     input  wire      hps_io_hps_io_usb1_inst_CLK,
80     output wire      hps_io_hps_io_usb1_inst_STP,
81     input  wire      hps_io_hps_io_usb1_inst_DIR,
82     input  wire      hps_io_hps_io_usb1_inst_NXT,
83     input  wire      hps_io_hps_io_uart0_inst_RX,
84     output wire      hps_io_hps_io_uart0_inst_TX,
85     output wire      hps_io_hps_io_spim1_inst_CLK,
86     output wire      hps_io_hps_io_spim1_inst_MOSI,
87     input  wire      hps_io_hps_io_spim1_inst_MISO,
88     output wire      hps_io_hps_io_spim1_inst_SS0,
89     inout  wire      hps_io_hps_io_i2c0_inst_SDA,
90     inout  wire      hps_io_hps_io_i2c0_inst_SCL,
91     inout  wire      hps_io_hps_io_i2c1_inst_SDA,
92     inout  wire      hps_io_hps_io_i2c1_inst_SCL,
93     inout  wire      hps_io_hps_io_gpio_inst_GPIO09,
94     inout  wire      hps_io_hps_io_gpio_inst_GPIO35,
95     inout  wire      hps_io_hps_io_gpio_inst_GPIO40,
96     inout  wire      hps_io_hps_io_gpio_inst_GPIO53,
97     inout  wire      hps_io_hps_io_gpio_inst_GPIO54,
98     inout  wire      hps_io_hps_io_gpio_inst_GPIO61
99 );
100
101 // Create a power on reset pulse for clean system reset on entry into user mode
102 // We create this with the altera_std_synchronizer core
103 wire sync_dout;
104 altera_std_synchronizer #(
105     .depth (20)
106 ) power_on_reset_std_sync_inst (
107     .clk      (clk),
108     .reset_n  (1'b1),
109     .din      (1'b1),
110     .dout     (sync_dout)
111 );
112
113 // Create a qsys system reset signal that is the logical AND of the power on

```

```

114 // reset pulse and the KEY[0] push button
115 wire qsys_system_reset;
116 assign qsys_system_reset = sync_dout & push_button[0];
117
118 // Qsys system instantiation template from soc_system/soc_system_inst.v
119 soc_system u0 (
120     .button_pio_export          (push_button[1]),
121     .clk_clk                    (clk),
122     .hps_0_hps_io_hps_io_emac1_inst_TX_CLK (hps_io_hps_io_emac1_inst_TX_CLK),
123     .hps_0_hps_io_hps_io_emac1_inst_TXD0  (hps_io_hps_io_emac1_inst_TXD0),
124     .hps_0_hps_io_hps_io_emac1_inst_TXD1  (hps_io_hps_io_emac1_inst_TXD1),
125     .hps_0_hps_io_hps_io_emac1_inst_TXD2  (hps_io_hps_io_emac1_inst_TXD2),
126     .hps_0_hps_io_hps_io_emac1_inst_TXD3  (hps_io_hps_io_emac1_inst_TXD3),
127     .hps_0_hps_io_hps_io_emac1_inst_RXD0  (hps_io_hps_io_emac1_inst_RXD0),
128     .hps_0_hps_io_hps_io_emac1_inst_MDI0  (hps_io_hps_io_emac1_inst_MDI0),
129     .hps_0_hps_io_hps_io_emac1_inst_MDC   (hps_io_hps_io_emac1_inst_MDC),
130     .hps_0_hps_io_hps_io_emac1_inst_RX_CTL (hps_io_hps_io_emac1_inst_RX_CTL),
131     .hps_0_hps_io_hps_io_emac1_inst_TX_CTL (hps_io_hps_io_emac1_inst_TX_CTL),
132     .hps_0_hps_io_hps_io_emac1_inst_RX_CLK (hps_io_hps_io_emac1_inst_RX_CLK),
133     .hps_0_hps_io_hps_io_emac1_inst_RXD1  (hps_io_hps_io_emac1_inst_RXD1),
134     .hps_0_hps_io_hps_io_emac1_inst_RXD2  (hps_io_hps_io_emac1_inst_RXD2),
135     .hps_0_hps_io_hps_io_emac1_inst_RXD3  (hps_io_hps_io_emac1_inst_RXD3),
136     .hps_0_hps_io_hps_io_sdio_inst_CMD    (hps_io_hps_io_sdio_inst_CMD),
137     .hps_0_hps_io_hps_io_sdio_inst_D0     (hps_io_hps_io_sdio_inst_D0),
138     .hps_0_hps_io_hps_io_sdio_inst_D1     (hps_io_hps_io_sdio_inst_D1),
139     .hps_0_hps_io_hps_io_sdio_inst_CLK    (hps_io_hps_io_sdio_inst_CLK),
140     .hps_0_hps_io_hps_io_sdio_inst_D2     (hps_io_hps_io_sdio_inst_D2),
141     .hps_0_hps_io_hps_io_sdio_inst_D3     (hps_io_hps_io_sdio_inst_D3),
142     .hps_0_hps_io_hps_io_usb1_inst_D0     (hps_io_hps_io_usb1_inst_D0),
143     .hps_0_hps_io_hps_io_usb1_inst_D1     (hps_io_hps_io_usb1_inst_D1),
144     .hps_0_hps_io_hps_io_usb1_inst_D2     (hps_io_hps_io_usb1_inst_D2),
145     .hps_0_hps_io_hps_io_usb1_inst_D3     (hps_io_hps_io_usb1_inst_D3),
146     .hps_0_hps_io_hps_io_usb1_inst_D4     (hps_io_hps_io_usb1_inst_D4),
147     .hps_0_hps_io_hps_io_usb1_inst_D5     (hps_io_hps_io_usb1_inst_D5),
148     .hps_0_hps_io_hps_io_usb1_inst_D6     (hps_io_hps_io_usb1_inst_D6),
149     .hps_0_hps_io_hps_io_usb1_inst_D7     (hps_io_hps_io_usb1_inst_D7),
150     .hps_0_hps_io_hps_io_usb1_inst_CLK    (hps_io_hps_io_usb1_inst_CLK),
151     .hps_0_hps_io_hps_io_usb1_inst_STP    (hps_io_hps_io_usb1_inst_STP),
152     .hps_0_hps_io_hps_io_usb1_inst_DIR    (hps_io_hps_io_usb1_inst_DIR),
153     .hps_0_hps_io_hps_io_usb1_inst_NXT    (hps_io_hps_io_usb1_inst_NXT),
154     .hps_0_hps_io_hps_io_spim1_inst_CLK   (hps_io_hps_io_spim1_inst_CLK),
155     .hps_0_hps_io_hps_io_spim1_inst_MOSI  (hps_io_hps_io_spim1_inst_MOSI),
156     .hps_0_hps_io_hps_io_spim1_inst_MISO  (hps_io_hps_io_spim1_inst_MISO),
157     .hps_0_hps_io_hps_io_spim1_inst_SS0   (hps_io_hps_io_spim1_inst_SS0),
158     .hps_0_hps_io_hps_io_uart0_inst_RX    (hps_io_hps_io_uart0_inst_RX),
159     .hps_0_hps_io_hps_io_uart0_inst_TX    (hps_io_hps_io_uart0_inst_TX),
160     .hps_0_hps_io_hps_io_i2c0_inst_SDA    (hps_io_hps_io_i2c0_inst_SDA),
161     .hps_0_hps_io_hps_io_i2c0_inst_SCL    (hps_io_hps_io_i2c0_inst_SCL),
162     .hps_0_hps_io_hps_io_i2c1_inst_SDA    (hps_io_hps_io_i2c1_inst_SDA),
163     .hps_0_hps_io_hps_io_i2c1_inst_SCL    (hps_io_hps_io_i2c1_inst_SCL),
164     .hps_0_hps_io_hps_io_gpio_inst_GPI009 (hps_io_hps_io_gpio_inst_GPI009),
165     .hps_0_hps_io_hps_io_gpio_inst_GPI035 (hps_io_hps_io_gpio_inst_GPI035),
166     .hps_0_hps_io_hps_io_gpio_inst_GPI040 (hps_io_hps_io_gpio_inst_GPI040),
167     .hps_0_hps_io_hps_io_gpio_inst_GPI053 (hps_io_hps_io_gpio_inst_GPI053),
168     .hps_0_hps_io_hps_io_gpio_inst_GPI054 (hps_io_hps_io_gpio_inst_GPI054),
169     .hps_0_hps_io_hps_io_gpio_inst_GPI061 (hps_io_hps_io_gpio_inst_GPI061),
170     .led_pio_export              (leds),
171     .memory_mem_a                (hps_memory_mem_a),
172     .memory_mem_ba               (hps_memory_mem_ba),
173     .memory_mem_ck               (hps_memory_mem_ck),
174     .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

```

```

93 set_location_assignment PIN_A25 -to hps_memory_mem_ras_n
94 set_location_assignment PIN_V28 -to hps_memory_mem_reset_n
95 set_location_assignment PIN_E25 -to hps_memory_mem_we_n
96 set_location_assignment PIN_D25 -to hps_memory_oct_rzqin
97
98 # define the IO standard for the HPS peripheral IO pins
99 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_MDC
100 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_MDIO
101 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RXD0
102 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RXD1
103 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RXD2
104 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RXD3
105 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RX_CLK
106 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_RX_CTL
107 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TXD0
108 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TXD3
109 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TXD1
110 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TXD2
111 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TX_CLK
112 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_emac1_inst_TX_CTL
113 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_CLK
114 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_CMD
115 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_D0
116 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_D1
117 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_D2
118 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_sdio_inst_D3
119 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_uart0_inst_RX
120 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_uart0_inst_TX
121 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_CLK
122 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D0
123 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D1
124 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D2
125 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D3
126 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D4
127 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D5
128 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D6
129 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_D7
130 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_DIR
131 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_NXT
132 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_usb1_inst_STP
133 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_spim1_inst_CLK
134 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_spim1_inst_MOSI
135 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_spim1_inst_MISO
136 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_spim1_inst_SS0
137 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_i2c0_inst_SDA
138 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_i2c0_inst_SCL
139 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_i2c1_inst_SDA
140 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_i2c1_inst_SCL
141 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO09
142 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO35
143 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO40
144 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO53
145 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO54
146 set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to hps_io_hps_io_gpio_inst_GPIO61
147
148 # define the current strength for the HPS peripheral IO pins
149 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_MDC
150 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_MDIO
151 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RXD0
152 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RXD1
153 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RXD2
154 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RXD3
155 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RX_CLK
156 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_RX_CTL
157 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TXD0
158 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TXD1
159 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TXD2
160 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TXD3
161 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TX_CLK
162 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_emac1_inst_TX_CTL
163 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_CLK

```



```

164 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D0
165 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D1
166 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D2
167 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D3
168 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D4
169 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D5
170 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D6
171 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_D7
172 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_DIR
173 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_NXT
174 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_usb1_inst_STP
175 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_CLK
176 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_CMD
177 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_D0
178 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_D1
179 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_D2
180 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_sdio_inst_D3
181 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_uart0_inst_RX
182 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_uart0_inst_TX
183 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_spim1_inst_CLK
184 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_spim1_inst_MOSI
185 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_spim1_inst_SS0
186 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_i2c0_inst_SDA
187 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_i2c0_inst_SCL
188 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_i2c1_inst_SDA
189 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_i2c1_inst_SCL
190 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO09
191 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO35
192 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO40
193 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO53
194 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO54
195 set_instance_assignment -name CURRENT_STRENGTH_NEW 4MA -to hps_io_hps_io_gpio_inst_GPIO61
196
197 # define the slew rate for the HPS peripheral IO pins and DRAM interface pins
198 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[0]
199 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[1]
200 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[2]
201 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[3]
202 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[4]
203 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[5]
204 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[6]
205 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[7]
206 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[8]
207 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[9]
208 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[10]
209 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[11]
210 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[12]
211 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[13]
212 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_a[14]
213 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_ba[0]
214 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_ba[1]
215 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_ba[2]
216 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_cas_n
217 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_cke
218 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_cs_n
219 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_odt
220 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_ras_n
221 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_reset_n
222 set_instance_assignment -name SLEW_RATE 1 -to hps_memory_mem_we_n
223 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_MDC
224 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_MDIO
225 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RXD0
226 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RXD1
227 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RXD2
228 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RXD3
229 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RX_CLK
230 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_RX_CTL
231 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TXD0
232 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TXD1
233 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TXD2
234 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TXD3

```

```

235 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TX_CLK
236 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_emac1_inst_TX_CTL
237 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_CLK
238 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D0
239 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D1
240 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D2
241 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D3
242 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D4
243 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D5
244 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D6
245 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_D7
246 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_DIR
247 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_NXT
248 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_usb1_inst_STP
249 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_CLK
250 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_CMD
251 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_D0
252 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_D1
253 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_D2
254 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_sdio_inst_D3
255 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_uart0_inst_RX
256 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_uart0_inst_TX
257 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_spim1_inst_CLK
258 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_spim1_inst_MOSI
259 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_spim1_inst_SS0
260 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_i2c0_inst_SDA
261 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_i2c0_inst_SCL
262 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_i2c1_inst_SDA
263 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_i2c1_inst_SCL
264 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO09
265 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO35
266 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO40
267 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO53
268 set_instance_assignment -name SLEW_RATE 1 -to hps_io_hps_io_gpio_inst_GPIO54
269 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:

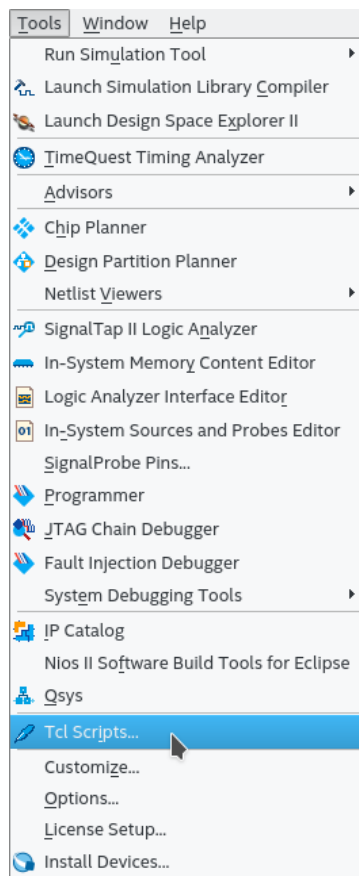


Figure 37: TCL Scripts Menu

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

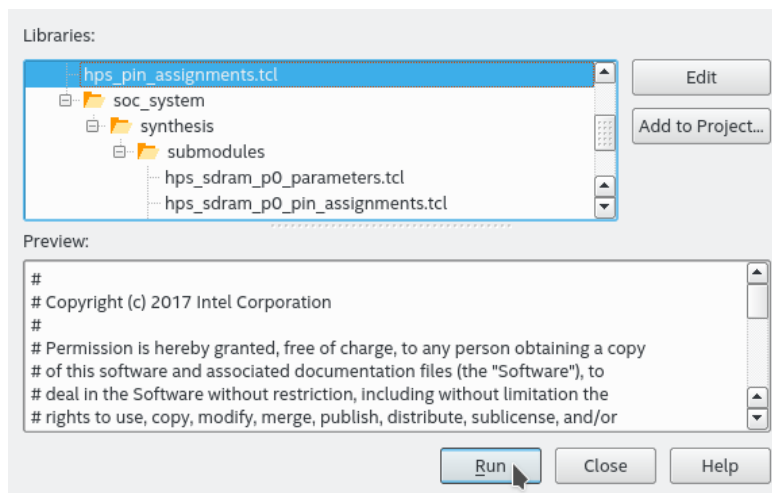


Figure 38: Run HPS Pin Assignments Script

Step 5. Next we need to run the `hps_sdrn_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_sdrn_p0_pin_assignments.tcl` script in the dialog and select it and click the **Run** button.

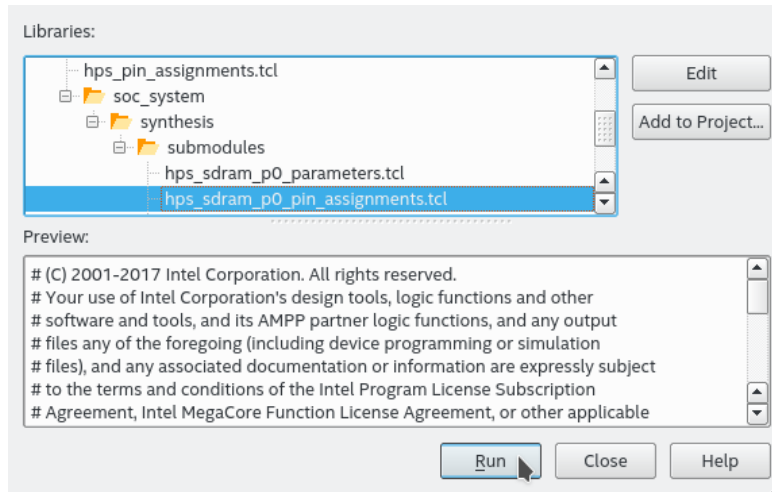


Figure 39: Run HPS SDRAM Pin Assignments Script

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](#).

```

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 # inform quartus that the clk port brings a 50MHz clock into our design so
24 # that timing closure on our design can be analyzed
25
26 create_clock -name clk -period "50MHz" [get_ports clk]
27 derive_clock_uncertainty
28
29 # inform quartus that the PIO inputs and outputs have no critical timing
30 # requirements. These signals are driving LEDs and reading discrete push button
31 # and switch inputs, there are no timing relationships that are critical for any
32 # of this
33
34 set_false_path -from [get_ports {switch[0]}] -to *
35 set_false_path -from [get_ports {switch[1]}] -to *
36 set_false_path -from [get_ports {switch[2]}] -to *
37 set_false_path -from [get_ports {switch[3]}] -to *

```

```

38 set_false_path -from * -to [get_ports {leds[0]}]
39 set_false_path -from * -to [get_ports {leds[1]}]
40 set_false_path -from * -to [get_ports {leds[2]}]
41 set_false_path -from * -to [get_ports {leds[3]}]
42 set_false_path -from * -to [get_ports {leds[4]}]
43 set_false_path -from * -to [get_ports {leds[5]}]
44 set_false_path -from * -to [get_ports {leds[6]}]
45 set_false_path -from * -to [get_ports {leds[7]}]
46 set_false_path -from [get_ports {push_button[0]}] -to *
47 set_false_path -from [get_ports {push_button[1]}] -to *
48
49 # Define timing constraints for the JTAG IO pins so that Quartus properly closes
50 # timing on these signal paths. Otherwise we could have unreliable JTAG
51 # communication with the device over the USB Blaster II connection.
52 # NOTE: the 'altera_reserved_tck' clock is automatically defined by Quartus
53
54 set_input_delay -clock altera_reserved_tck -clock_fall 3 [get_ports {altera_reserved_tdi}]
55 set_input_delay -clock altera_reserved_tck -clock_fall 3 [get_ports {altera_reserved_tms}]
56 set_output_delay -clock altera_reserved_tck 3 [get_ports {altera_reserved_tdo}]
57
58 # Define clocks for the HPS ports that expose clock signals to avoid
59 # unconstrained clock warnings
60 create_clock -period "1 MHz" [get_ports {hps_io_hps_io_i2c0_inst_SCL}]
61 create_clock -period "1 MHz" [get_ports {hps_io_hps_io_i2c1_inst_SCL}]
62 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.

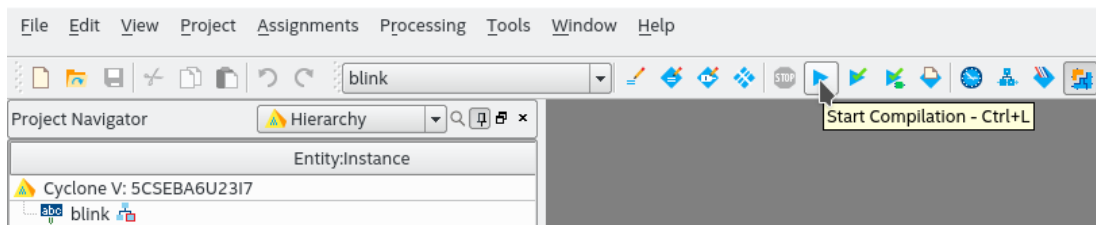


Figure 40: Start Compilation Button on Intel Quartus Software Toolbar

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.

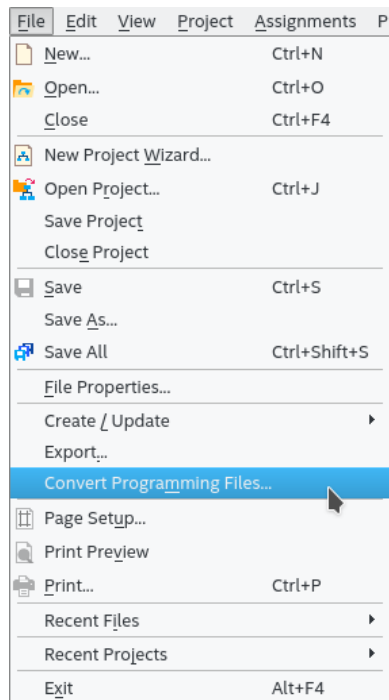


Figure 41: Convert Programming Files Menu

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.

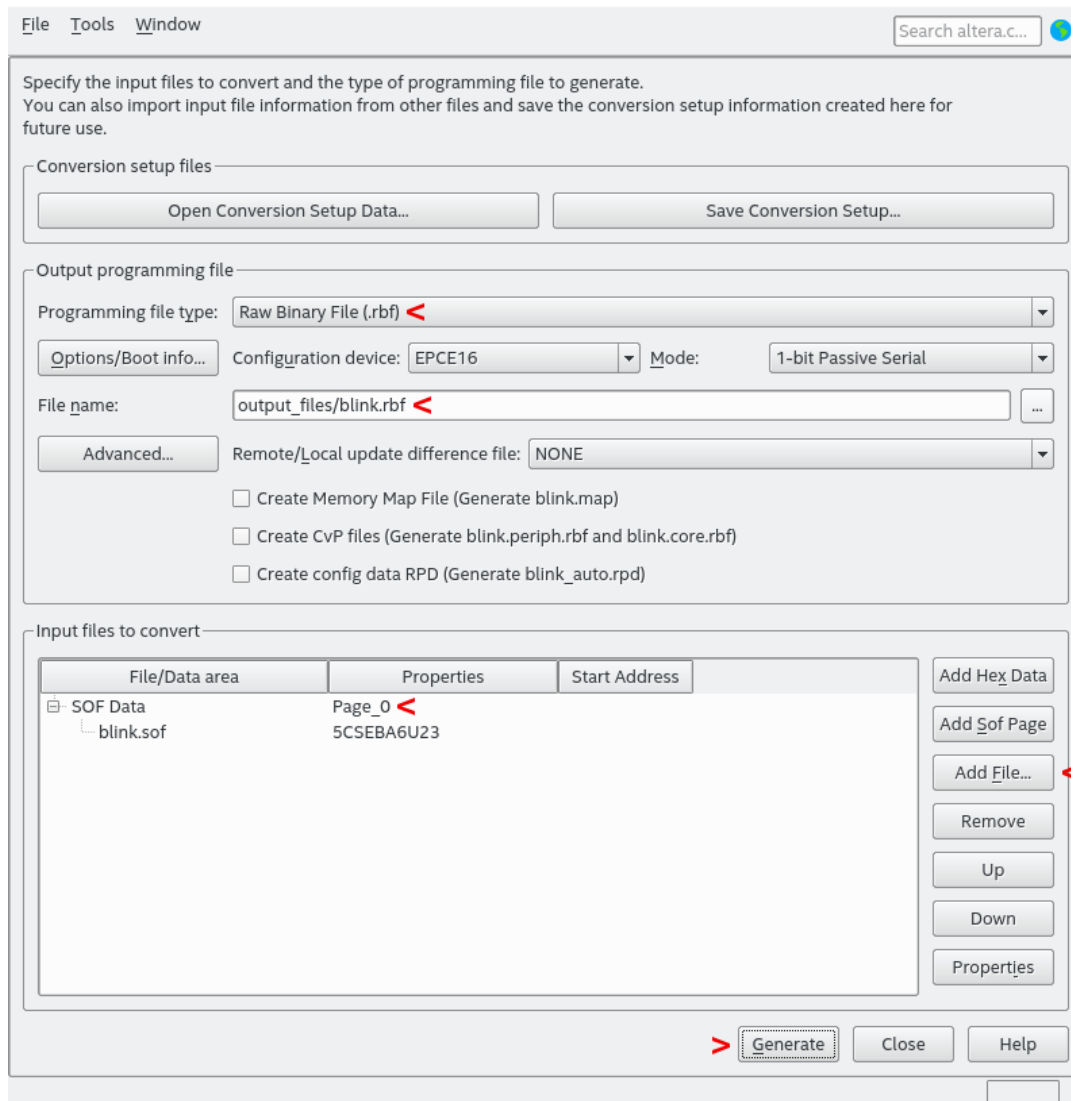


Figure 42: Convert Programming Files Menu

- 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.

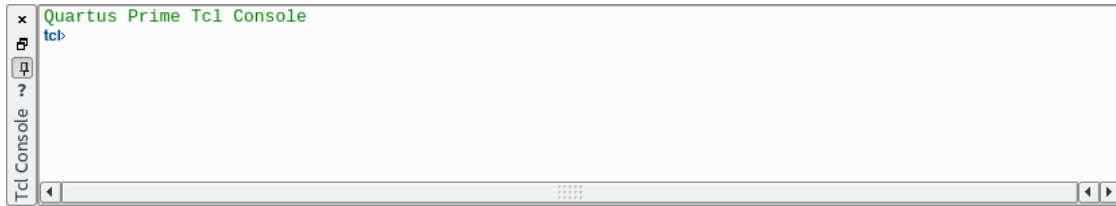


Figure 43: Intel Quartus Prime Software TCL Console

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
# socp-create-header-files on your host PC using the environment variables
# provided by Quartus.
tcl> set SCHF_PATH [glob -join $quartus(quartus_rootpath) socp_builder bin socp-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 socp-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 OGRAM_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.