



AN 957: Time-Sensitive Networking for Drive-on-Chip Design Example



[Online Version](#)



[Send Feedback](#)

AN-957

ID: [683707](#)

Version: [2021.10.30](#)

Contents

1. About the Time-Sensitive Networking for Drive-on-Chip Design Example.....	4
1.1. Features of TSN for Drive-on-Chip Design Example.....	5
1.2. TSN for Drive-on-Chip Design Example General Description.....	6
1.3. Messages from the TSN Drive-on-Chip Design Example.....	7
1.4. Design Flow for the TSN Drive-on-Chip Design Example.....	7
2. Getting Started with the TSN for Drive-on-Chip Design Example.....	9
2.1. Hardware Requirements for the TSN for Drive-on-Chip Design Example.....	9
2.2. Software Requirements for the TSN for Drive-on-Chip Design Example.....	9
2.3. Configuring the Cyclone V SoC Development Board for the TSN for Drive-on-Chip Design Example.....	10
2.4. Programming the FPGA for the TSN for Drive-on-Chip Design Example.....	11
2.5. Creating an SD Card Image for the TSN for Drive-on-Chip Design Example.....	11
2.6. Turning on the Cyclone V SoC Development Board for the TSN for Drive-on-Chip Design Example.....	12
2.7. Configuring the TSN IP.....	13
3. Porting the Intel MAX 10 Drive-On-Chip design to the Cyclone V SoC Development Board	14
3.1. Changing File Names, Revision Name, and Target Device for the TSN Drive-on-Chip Design Example.....	14
3.2. Modifying the Drive-On-Chip Qsys System.....	15
3.3. Adding the TTTech TSN IP to the Qsys system.....	26
3.4. Connecting the TSN and Drive-on-Chip Subsystems.....	37
3.5. Compiling the Quartus Prime Design and Top-Level Module.....	39
3.6. Generating the Preloader.....	40
3.7. Generating a .jic file	41
3.8. Compiling the Drive-on-Chip Design Software in Nios II Software Build Tools.....	42
3.9. Launching a YOCTO Build	44
3.10. Building an SD Card Image for the TSN Drive-on-Chip Design Example.....	45
3.11. Changing MAC Addresses.....	47
3.12. Reading and Checking Physical Addresses on the Cyclone V SoC Development Board..	48
4. Running HPS Software for the TSN Drive-on-Chip Design.....	49
4.1. Installing the Arm Compiler for Motor Control from Linux.....	49
4.2. Creating a Basic C Program for Motor Control.....	49
4.3. OPC UA PubSub based on open62541.....	50
4.4. Getting and Building open62541.....	51
4.5. Setting up Debug Sessions in DS-5.....	51
5. Connecting the Cyclone V SoC Development board to the Tandem 48 V Motion-Power board	53
6. Running the Program.....	54
7. TSN Configuration Example.....	55
8. Document Revision History for AN 957: Time-Sensitive Networking for Drive-on-Chip Design Example.....	56

A. Example .qsf for Pin Assignments and Attributes.....	57
B. Top-level Verilog HDL File Example.....	79
C. YOCTO Build Patch File (cvsx_doc_tsn_2_3-rt) for the TSN Drive-on-Chip Design Example.....	89
D. Script to read and change MAC addresses from Cyclone V SoC EEPROM.....	94

1. About the Time-Sensitive Networking for Drive-on-Chip Design Example

This design ports the TTTech TSN IP functionality to the Cyclone V SoC Development board. This design shows how the motor control and time-sensitive networking (TSN) can work together in a flexible and efficient FPGA implementation.

The design integrates:

- The Drive-On-Chip Design Example for Intel MAX 10 devices
- The TTTech TSN IP

The TTTech TSN IP includes all the tools (DE-PTP, tsntool, deptp_tool, Netconf, YANG, SSH) necessary to set up an Ethernet connection over TSN using the device. You can run a high-level communication protocol such OPC UA (client-server or PubSub) over TSN.

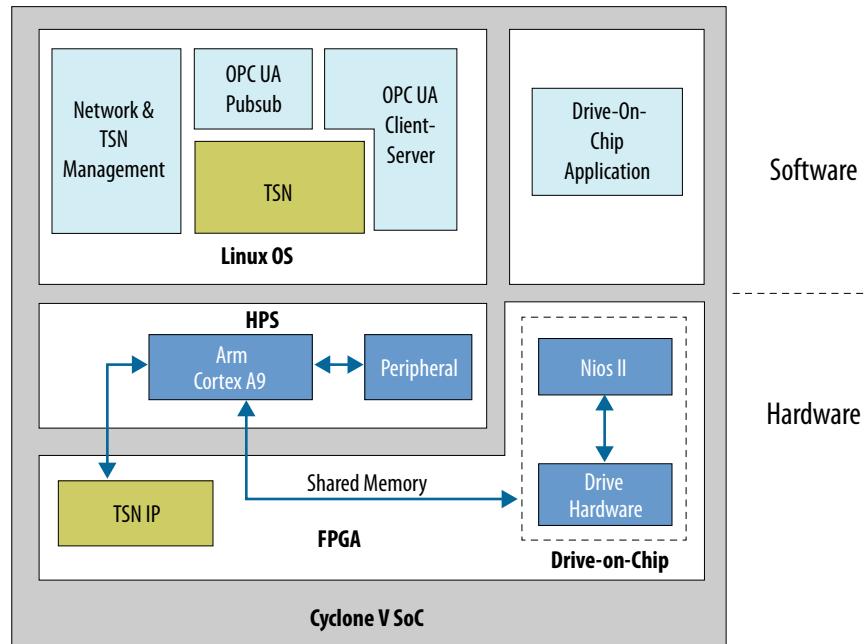
The design allows you to learn about drive-on-chip applications and TSN functionality. The design incorporates TSN and OPC UA and the versatility of motor control embedded in an Intel FPGA device.

Industry 4.0 industrial networking standards OPC UA PubSub and TSN are possible on a local network with multiple devices. Industry 4.0 allow you to migrate towards TSN to consolidate communication in a single network that is efficient, secure, and reliable. You can add every node in the factory from industrial PCs to sensors to the TSN network including embedded motor control. In industry 4.0, TSN is becoming more relevant with communication protocols such as OPC UA PubSub. FPGAs allow you to introduce new hardware and SoC products into the manufacturing environment.

You build the FPGA bitstream and the SD card image for the TTTech TSN IP and the Intel Drive-On-Chip for Intel MAX 10 design.

Intel integrated the TSN IP into the design using TTTech DE-EVAL reference design.

Figure 1. TSN Drive-on-Chip Design Block Diagram



Related Information

- AN 773: Drive-On-Chip Design Example for Intel MAX 10 Devices
- Cyclone V Development Kit
- Cyclone V Development Kit User Guide
- Cyclone Development Board Reference Manual
- Drive-on-Chip Design Example for Intel MAX 10 Devices
- TTTech TSN IP

1.1. Features of TSN for Drive-on-Chip Design Example

The Cyclone V SoC Development board offers

- A two-port switched endpoint
- TTTech TSN IP flashed onto Cyclone V SoC
- Drive on Chip design running on Nios II soft processor in the Cyclone V SoC (μ COS II)
- TTTech TSN IP in the FPGA.
- SD card boot uses TTTech TSN IP but does not configure the FPGA.
- HPS (Arm processor) that runs Linux and TTTech TSN drivers.
- On-board EPCQ memory that programs the FPGA and stores the Nios II drive-on-chip application software.

The Tandem Motion-Power 48 V Board offers:

- Dual-axis motor control
- A visual TSN and OPC UA endpoint for demonstrations

Related Information

[Tandem Motion-Power 48 V Board](#)

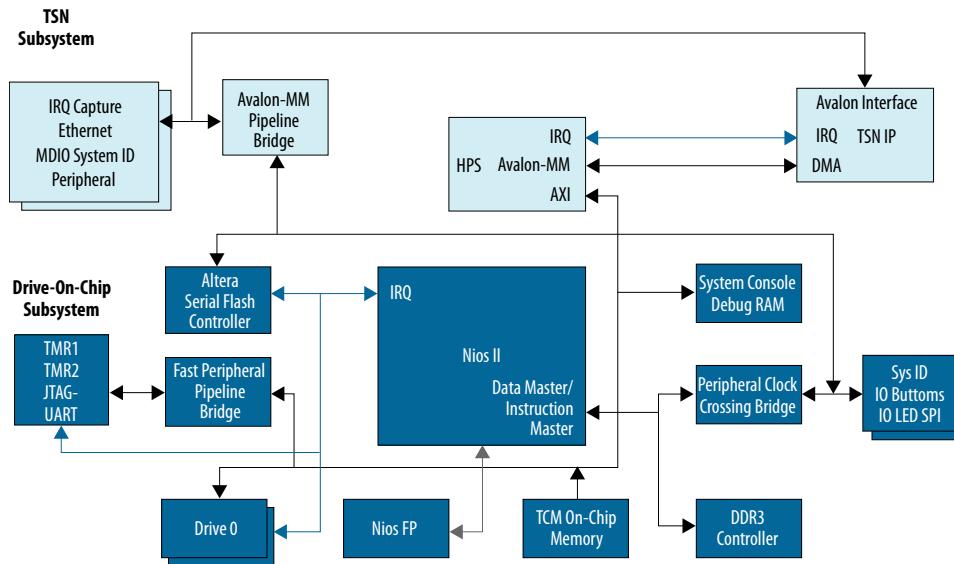
1.2. TSN for Drive-on-Chip Design Example General Description

Intel tests this design in a Cyclone V SoC development kit. This hardware provides flexibility to implement different functionalities required for industrial applications. The design shows you can implement drive-on-chip for motor control and TSN to share information and execute commands.

The HPS (Hard Processor System) allows consolidating the system workload as an embedded device, using a Linux kernel to administrate resources, tasks and drive the hardware with high-level programming methods.

The design has two subcomponents: the TTTech TSN IP and the Drive-on-Chip design example. Qsys integrates the design.

Figure 2. **TSN Drive-on-Chip Design Block Diagram.**



The HPS within the Cyclone V SoC development board is an Arm Cortex A9 MPCore processor. It manages the TSN IP and communicates through the drive-on-chip using the standard AXI (share memory). Also, different bridges communicate with peripherals and exploit other functionalities.

An instance of the Nios II soft processor in the FPGA controls and manages the resources of the drive-on-chip design. The EPCQ flash memory programs the Nios II soft processor. This program is flashed into the memory as a single instruction file called `epcq.hex` generated in Nios II Eclipse project.

The soft processor performs all motor driving functions and continuously communicates with the HPS using the corresponding AXI and Avalon interfaces.

The HPS retrieves information from the drive-on-chip design such as rotational speed and quadrature current. Using memory mapping addressing, it can access all the information that the drive-on-chip design generates: motor speed, position, KP gain, KI gain, filter DC gain, wave period and offsets, DC-DC parameters. The HPS can publish all these values to the network using the instantiation of the TSN TTTech IP and some high-level communication protocol such as OPC UA PubSub.

The embedded device allows scalability of its features. You can add new components if using the correct hardware drivers and access other development board resources such as parallel IO, memories, JTAG, UART, SPI, Buttons, LEDs.

Related Information

- [AN 669: Drive-On-Chip Reference Design](#)
- [AN 773: Drive-On-Chip Design Example for Intel MAX 10 Devices](#)

1.3. Messages from the TSN Drive-on-Chip Design Example

Cyclone V SoC Development board

You can run the OPC UA PubSub application on the Cyclone V SoC Development board. You can compile the design in DS-5 environment or standalone using the Arm compiler (cross-compilation). The incoming messages can include motor commands and motor parameters. Published messages from the SoC can include measurements and parameters from the motor. Besides OPC UA PubSub, the SoC can support other applications such as OPC UA Client-Server.

TSN scheduled messages

You can implement IEEE 802.1Qbv scheduling by either:

- Using VLANs for stream differentiation and IEEE 802.1Qbv configuration in the command line using sgs and tsntool (refer to *TTTech TSN IP Reference Design for DE-IP-SCV user manual*).
- Using TTTech SlateXNS: a browser-based tool used to create schedules and deploy configurations to switches using Netconf over TLS

Motor commands and recorded data over TSN

You can control, within the network, certain motor parameters: rotational speed, position commands, speed limits and other parameters to adjust the control loops. The OPC UA Publisher (or client/server) controller sends these messages to the Cyclone V SoC Development board by overriding the corresponding attribute. Also, you can return parameters and performance values to the network by programming an OPC UA Publisher in the SoC and giving these message priority and schedule.

1.4. Design Flow for the TSN Drive-on-Chip Design Example

Intel does not supply source files, images, configuration files but does describe how to create them.

1. Download both the Drive-on-Chip application and TTTech TSN IP onto the FPGA.
2. Store them in the EPCQ memory of the Cyclone V SoC Development board using a .jic format file. The .jic file is a combination of the .sof output file (product of the Intel Quartus Compilation) and an .hex file (product of the Drive-on-Chip software compilation). The Drive-on-Chip design runs on the Nios II soft processor using the µCOSII (Real Time OS) framework.
3. Program the TTTech TSN IP into the FPGA and it uses the HPS within the Cyclone V SoC to run Linux and manage the TSN system. The Intel Quartus Prime compilation for the project produces handoff files. These are input of BSP editor (part of SoC EDS) to create an HPS preloader.
4. Use this preloader image to create an SD card image for the HPS to boot-up Linux.
5. Build an SD card image for the development board using the YOCTO project. TTTech customized the flow to meet its IP requirements and tools. YOCTO is a method for building customized embedded Linux distributions.
6. Describe, by using recipes, the components of your system and produce a bootable SD card image (or files) as the product of the build. For more information, refer to the YOCTO PROJECT website. A Linux machine launches the YOCTO build, as it cannot build on a mounted NFS volume.
7. Patch the YOCTO build to adapt it to Cyclone V SoC Development board.
8. Use the design as a base to integrate the TSN drive-on-chip hardware in Intel Quartus Prime. For Smart Factory solutions, the Linux distribution of the TTTech IP reference design (5.4.40) can be patched with Linux RT-patch 5.4.40-rt24 HPS runs Linux, necessary daemons, TTTech TSN drivers, and any other high-level application you program.

Related Information

[YOCTO PROJECT](#)

2. Getting Started with the TSN for Drive-on-Chip Design Example

[Hardware Requirements for the TSN for Drive-on-Chip Design Example on page 9](#)

[Software Requirements for the TSN for Drive-on-Chip Design Example on page 9](#)

[Configuring the Cyclone V SoC Development Board for the TSN for Drive-on-Chip Design Example on page 10](#)

[Programming the FPGA for the TSN for Drive-on-Chip Design Example on page 11](#)

[Creating an SD Card Image for the TSN for Drive-on-Chip Design Example on page 11](#)

[Turning on the Cyclone V SoC Development Board for the TSN for Drive-on-Chip Design Example on page 12](#)

[Configuring the TSN IP on page 13](#)

2.1. Hardware Requirements for the TSN for Drive-on-Chip Design Example

- Cyclone V SoC Development board acting as networked drive controller. flashed with TSN switch IP and Nios II Drive-on-Chip design.
- Tandem Motion-Power 48 V dual motor and power conversion board from Terasic
- Ethernet cables
- HSMC flex cable.

2.2. Software Requirements for the TSN for Drive-on-Chip Design Example

- Intel Quartus Programmer v17.0. Alternatively, to modify the embedded hardware, you require a full installation of Intel Quartus Standard Edition v17.0.
- Software to create and format USB flash images such as Disk Imager or Rufus.
- TTTech TSN IP version 2.3 for Linux.
 - Linux SoC kernel 5.4.40-ltsi or Linux RT patch 5.4.40-rt24 (optional)
 - TTTech stack provides: DE-PTP Edge, NETCONF, TSNTOOL, LLDP, MSTP, toolchain.
- Nios II software µCOSII version 2.86
- Open62541 OPC UA software for Linux. from Github website.
- Optional, to modify the Linux stack, a host machine with Linux is necessary to launch the YOCTO build

Related Information

Github website

2.3. Configuring the Cyclone V SoC Development Board for the TSN for Drive-on-Chip Design Example

1. Set the DIP switches and jumpers on the board.

Table 1. DIP Switch Settings

For more information on each switch, refer to the Cyclone V SoC Development Board Reference Manual

Switch	Setting
MSEL (SW3)	
0	OFF
1	OFF
2	ON
3	ON
4	OFF
5	ON
SW2	
ALL	OFF
SW1	
ALL	OFF
SW4	
1	OFF
2	OFF
3	ON
4	ON

Table 2. Jumper Settings

For more information on each jumper, refer to the Cyclone V SoC Development Board Reference Manual

Jumper	Position
J26	Right
J27	Right
J28	Right
J29	Right
J30	Left

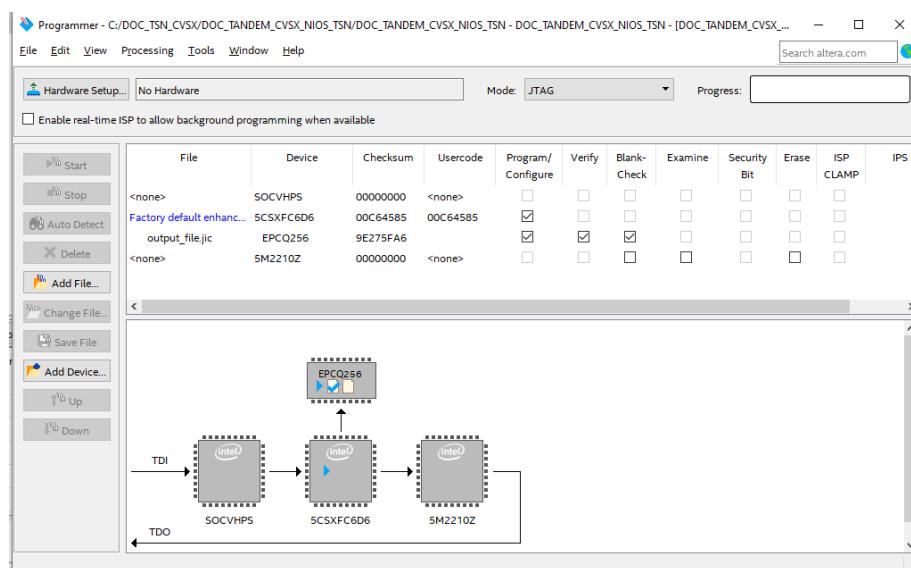
Related Information

Cyclone V SoC Development Board Reference Manual

2.4. Programming the FPGA for the TSN for Drive-on-Chip Design Example

1. Download Intel Quartus Programmer.
2. Select **edition standard, release 17.0**, and your operating system.
3. On the **Additional Software** tab, under **Stand-Alone Software**, select **Quartus Prime Programmer and Tools**.
4. Download and install it.
5. Connect the microUSB cable to the FPGA integrated USB Blaster II (the port near to the Ethernet connectors).
6. Turn on the FPGA.
7. Open the Quartus Programmer and check the USB Blaster II is recognized.
8. Alternatively, use **Auto Detect** feature.
9. Click **Add file...** and select `output_file.jic` (or the created `.jic` file).

Figure 3. Intel Quartus Programmer Configuration



10. Program the FPGA board (the EPCQ flash memory). The EPCQ flash store the FPGA image for the next power-on sequence.

Related Information

[Intel Quartus Programmer](#)

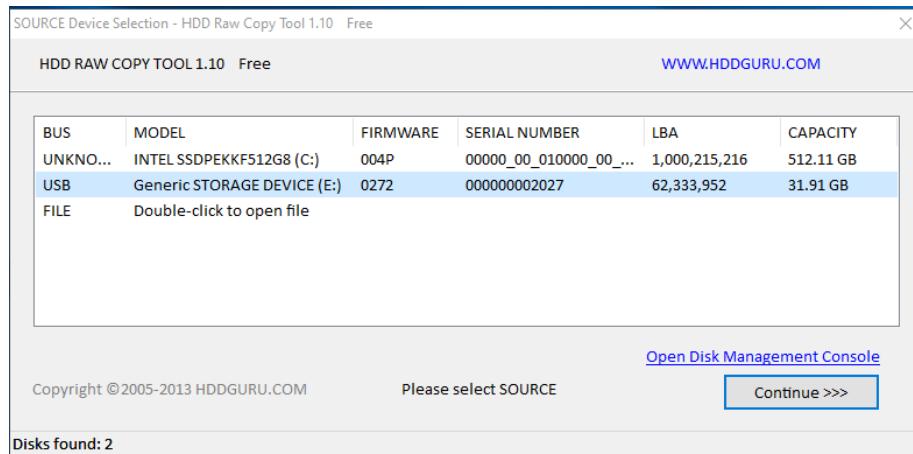
2.5. Creating an SD Card Image for the TSN for Drive-on-Chip Design Example

The size of the image is approximately 3 GB, so chose an appropriate SD card.

The HPS boots from the SD card containing the preloader, U-boot, Linux SoC-FPGA, device tree.

1. Open Disk Imager.

Figure 4. Disk Imager



2. Select your source as <TSN_DoC_SD>.img or (<TSN_DoC_SD_RT>.img) and the corresponding target.
3. Plug the SD card into the Cyclone V SoC Development board.
4. Burn the image onto an SD card in a Linux machine:
 - a. Determine the device associated with the SD card on the host by running the following command before and after inserting the card in the reader:


```
>> cat /proc/partitions
```
 - b. Use dd utility to write the SD image to the SD card, for instance:


```
>> sudo dd if= TSN_DoC_SD_RT.img of=/dev/sdX bs=1M
```
 - c. Use sync utility to flush the changes into the SD card.


```
>> sudo sync
```

2.6. Turning on the Cyclone V SoC Development Board for the TSN for Drive-on-Chip Design Example

The default ip address is 192.168.1.20 and can be accessed using the command ssh root@192.168.1.20.

1. Create a Putty session (or similar) in Windows with help of the UART port: USB connector (J8). Default baud rate is 115200.
2. Check the MAC addresses (physical addresses) of multiple development kits if these are going to be connected within the same LAN. The TTTech IP boot-up flow reads these addresses during the first boot and stores them for the following power-up cycles.
3. Check the MAC addresses stored in the Cyclone V SoC development board EEPROM and change them if necessary.
4. If a MAC address is changed in the EEPROM, burn again the SD card image and plug it to the Cyclone V Development board, so the first Linux Boot Up process also reads and stores the new addresses. Do not use previously burned SD cards after changing MAC (physical) addresses.

The system is ready to run C-compiled applications such as OPC UA PubSub, client-server or any other.

2.7. Configuring the TSN IP

To configure or check TSN related parameters including time synchronization (PTP) and traffic scheduling, refer to the *TTTech TSN IP Reference Design for DE-IP-SCV user manual*

3. Porting the Intel MAX 10 Drive-On-Chip design to the Cyclone V SoC Development Board

[Changing File Names, Revision Name, and Target Device for the TSN Drive-on-Chip Design Example](#) on page 14

[Modifying the Drive-On-Chip Qsys System](#) on page 15

[Adding the TTTech TSN IP to the Qsys system](#) on page 26

[Connecting the TSN and Drive-on-Chip Subsystems](#) on page 37

[Compiling the Quartus Prime Design and Top-Level Module](#) on page 39

[Generating the Preloader](#) on page 40

[Generating a .jic file](#) on page 41

[Compiling the Drive-on-Chip Design Software in Nios II Software Build Tools](#) on page 42

[Launching a YOCTO Build](#) on page 44

[Building an SD Card Image for the TSN Drive-on-Chip Design Example](#) on page 45

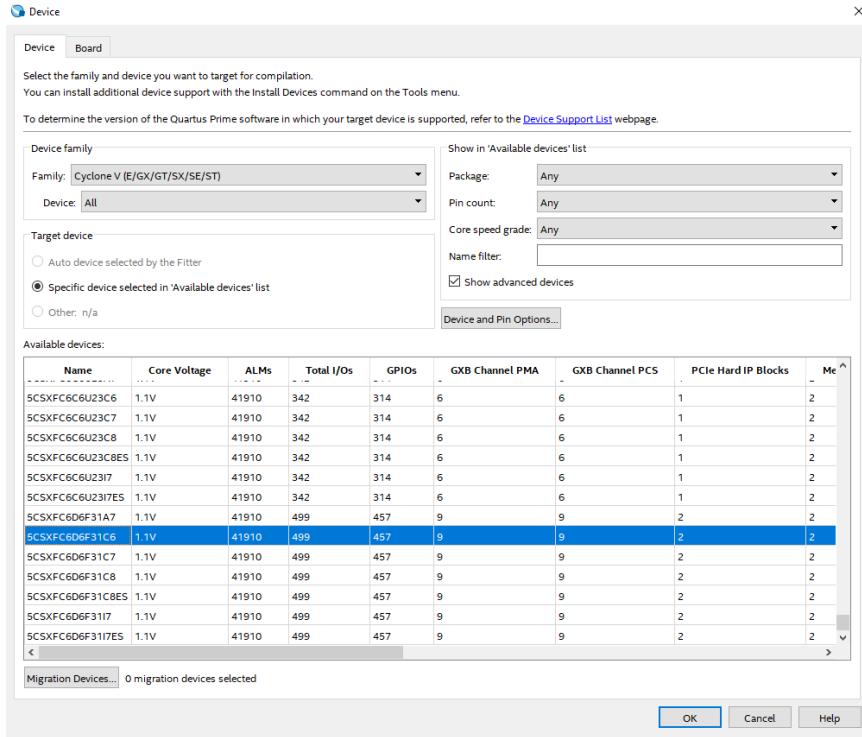
[Changing MAC Addresses](#) on page 47

[Reading and Checking Physical Addresses on the Cyclone V SoC Development Board](#) on page 48

3.1. Changing File Names, Revision Name, and Target Device for the TSN Drive-on-Chip Design Example

1. Change the name to differentiate the new project. For example, for the `.qpf` file `DOC_TANDEM_CVSX_NIOS_TSN`, as this variation of the project includes the drive-on-chip based on Nios II soft-processor and the TSN IP.
You can also change the name of the `.qsys` file.
2. In Intel Quartus Prime 17.0, navigate to **File > Open Project** and select the newly-named `.qpf` file.
3. Apply the following changes;
 - a. Create a new revision of the project, by selecting **Project > Revisions** and add a new revision based on `DOC_TANDEM_MAX_10`, for example `DOC_TANDEM_CVSX_NIOS_TSN`.
 - b. Remove the previous revision.
 - c. Change the target device, by selecting **Assignments > Device > 5CSXFC6D6F31C6**:

Figure 5. Selected Device



- Change the name of the top-level Verilog HDL file from to `DOC_TANDEM_MAX10.v` to `DOC_TANDEM_CVSX_NIOS_TSN.v`
- In the new top-level Verilog HDL file, change the name of the main Qsys instance `u-doc` to match the name of the `.qsys` file. For example:
`DOC_TANDEM_CVSX_NIOS_TSN_QSYS`

3.2. Modifying the Drive-On-Chip Qsys System

Several blocks in the Qsys system are not compatible with the Cyclone V SoC and you must delete or modify them.

- In Intel Quartus Prime, open the `.qpf` file by selecting **File > Open Project**.
- Open Qsys by selecting **Tools > Qsys**.
- In Qsys, select `DOC_TANDEM_CVSX_NIOS_TSN_QSYS.qsys..`
- Remove the blocks **ADC_Trigger_master_0** and **max10_adc**.
These two blocks are for the Intel MAX 10 Development board and are not required for the implementation in Cyclone V SoC Development board.
- Remove the blocks **ecfs_doc_threshold_sink**, **ALU_FOC_Float_av**, and **ALU_FOC_Fixp_av**.
The design does not need these blocks and removing them frees up resources.
- Replace the blocks **pll** and **pll2** as the module **Avalon ALTPLL** is not compatible with Cyclone V SoCs.
- Add the **Altera PLL** block and configure the PLL with the following settings:

Table 3. Altera PLL Settings

Setting	Value
PLL mode	Integer N-PLL
Reference Clock Frequency	50.0 MHz
Enable locked output port	True
Number of clocks	6
outclock0	Desired Frequency: 10.0 MHz Phase shift: 0 ps Duty Cycle: 50 %
outclock1	Desired Frequency: 10.0 MHz Phase shift: 0 ps Duty Cycle: 50 %
outclock2	Desired Frequency: 100.0 MHz Phase shift: 0 ps Duty Cycle: 50 %
outclock3	Desired Frequency: 20.0 MHz Phase shift: 0 ps Duty Cycle: 50 %
outclock4	Desired Frequency: 20.0 MHz Phase shift: 10000 ps Duty Cycle: 50 %
outclock5	Desired Frequency: 25.0 MHz Phase shift: 0 ps Duty Cycle: 50 %

8. Optionally, connect the new outputs of the PLL to clock source modules to ease the connection of the different clocks to other components.

Figure 6. Example for outclock0 to distribute clock signals

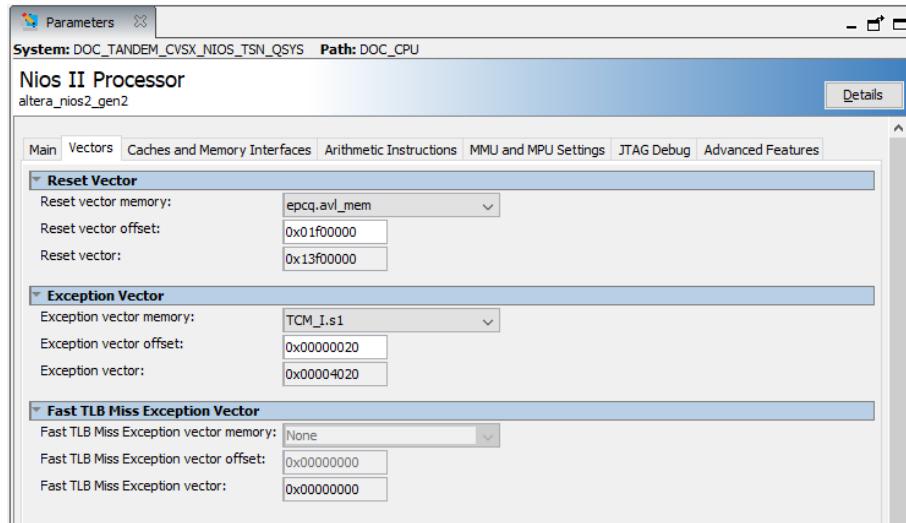
Name	Description	Export	Clock
clock_bridge_0	Clock Bridge		
in_dk	Clock Input	clk_ddr_in	exported
out_dk	Clock Output		Double-click to export dock_bridge...
clk_50	Clock Bridge		
in_dk	Clock Input	clk_50_in	exported
out_dk	Clock Output		Double-click to export clk_50_out_dk
reset_bridge_0	Reset Bridge		
dk	Clock Input		Double-click to export clk_50_out_dk
in_reset	Reset Input	reset_n	[clk]
out_reset	Reset Output		Double-click to export [clk]
pll	Altera PLL		
refclk	Clock Input		Double-click to export clk_50_out_dk
reset	Reset Input	pll_reset	pll_outclk0
outclk0	Clock Output		Double-click to export pll_outclk0
outclk1	Clock Output		Double-click to export pll_outclk1
outclk2	Clock Output		Double-click to export pll_outclk2
outclk3	Clock Output	clk_adc_out	pll_outclk3
outclk4	Clock Output		Double-click to export pll_outclk4
outclk5	Clock Output		Double-click to export pll_outclk5
locked	Conduit		Double-click to export
pll_soc_locked_to_r...	PLL locked to reset		
locked	Conduit		Double-click to export
reset_out	Reset Output		Double-click to export
pll_eth_tsn	Altera PLL		
refclk	Clock Input		Double-click to export clk_50_out_dk
reset	Reset Input		Double-click to export pll_eth_tsn...
outclk0	Clock Output		Double-click to export
locked	Conduit		Double-click to export
clk_10		pll_eth_tsn.locked	
clk_in	Conduit [conduit_end 17.0]		Double-click to export pll_outclk0
clk_in_reset	Associated clock: None (asynchronous)		[clk_in]
clk	Clock Output		Double-click to export clk_10
clk_reset	Reset Output		Double-click to export clk_10

9. Optional modifications):

- Remove the block named **mm_bridge_0** and connect its secondaries (**fast_periph_pb**, **lvmc_dlink**, **drive0**, **drive1**) directly to the Avalon data master of the Nios II soft-processor and the JTAG master module
 - Remove the block **peripheral_pb** and connect its secondaries (**sys_console_debug_ram** and **periph_cbb**) to Avalon data master of the Nios II soft-processor and the JTAG master module
10. To boot-up the Nios II soft-processor from On-Board EPCQ memory, add an instance of **Altera Serial Flash Controller** and label it as **epcq**.
- Use the 25 MHz clock and connect it to the **periph_cbb** block.
 - Connect the **interrupt sender** port to the **irq** port of the Nios II soft processor.

- c. Give the interrupt the number **4** and assign the addresses **0x0000_0100** to **avl_csr** and **0x0200_0000** to **avl_mem** ports.
- 11. For the Nios II soft processor to boot from on-board EPCQ memory, set up the **Reset Vector** in the Nios II Processor Settings.

Figure 7. Nios II Processor Reset Vector Settings



- 12. Remove the block **mem_if_ddr3_emif_0** and add it again. The previous external memory interface (EMIF) is exclusive to Intel MAX 10 devices. By removing and adding it again the new EMIF is for Cyclone V SoCs.
- 13. Configure the EMIF.

Figure 8. DDR3 SDRAM Settings

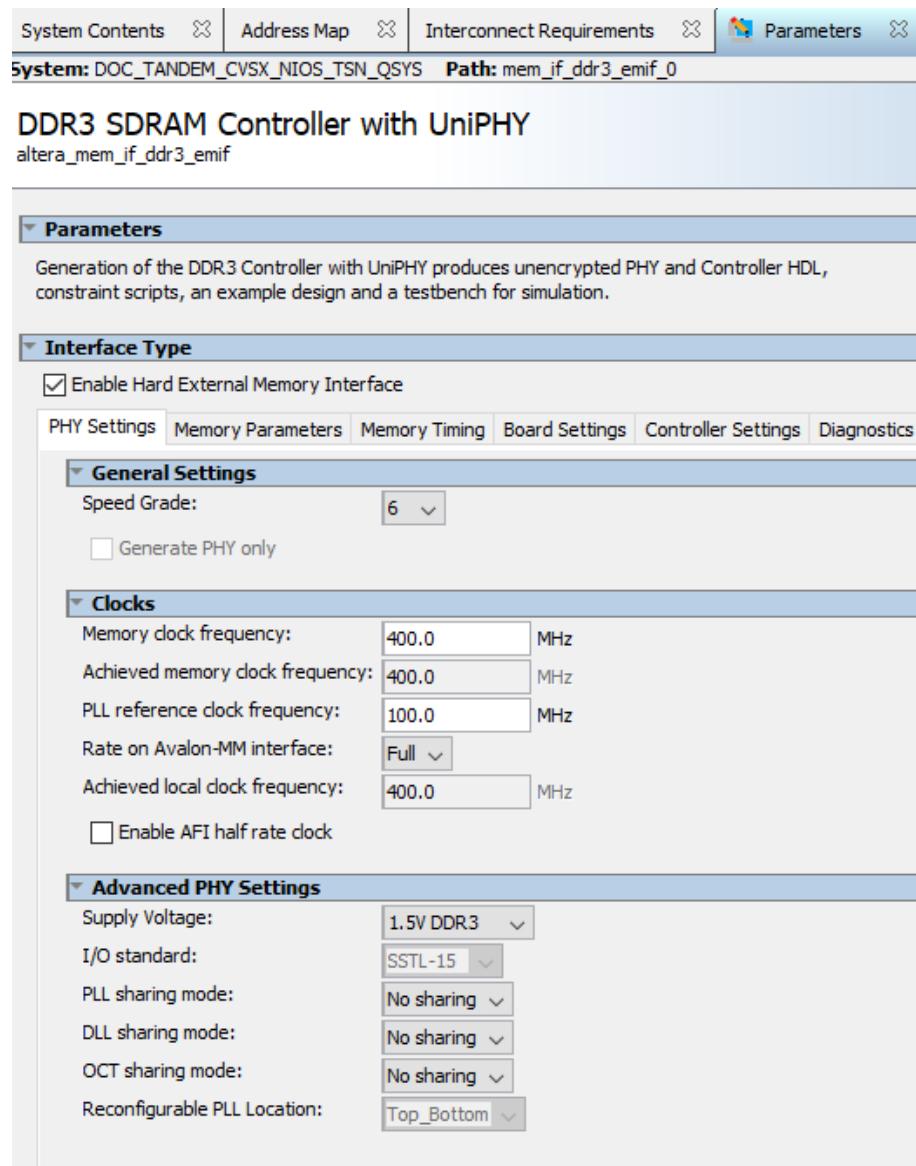


Figure 9. DDR3 SDRAM Settings

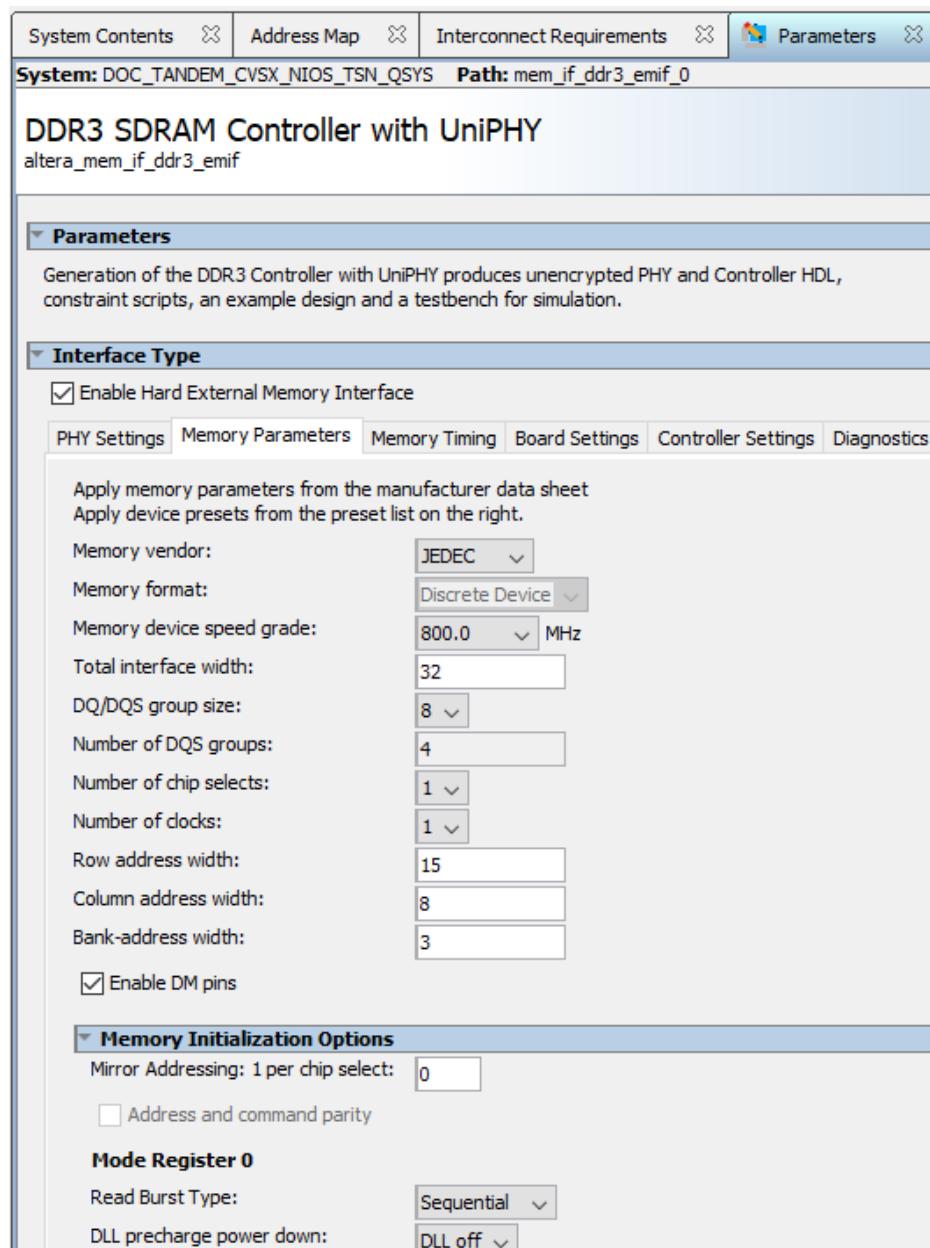


Figure 10. DDR3 SDRAM Settings

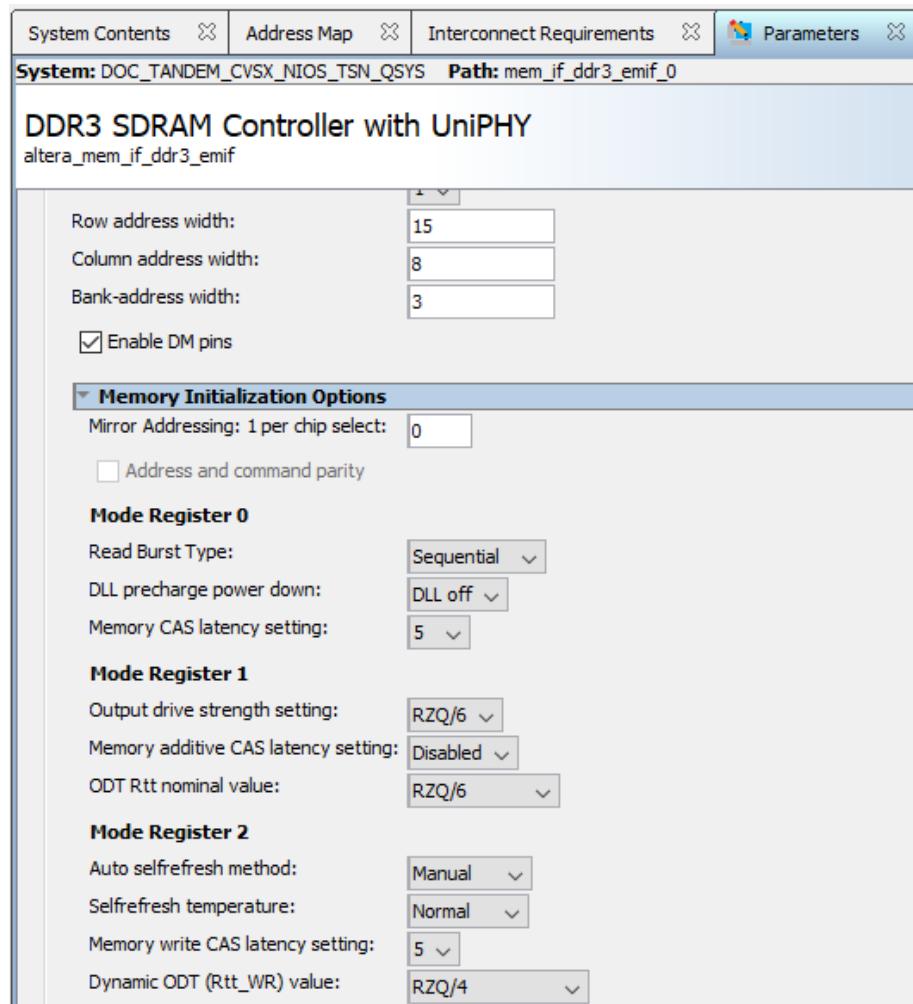


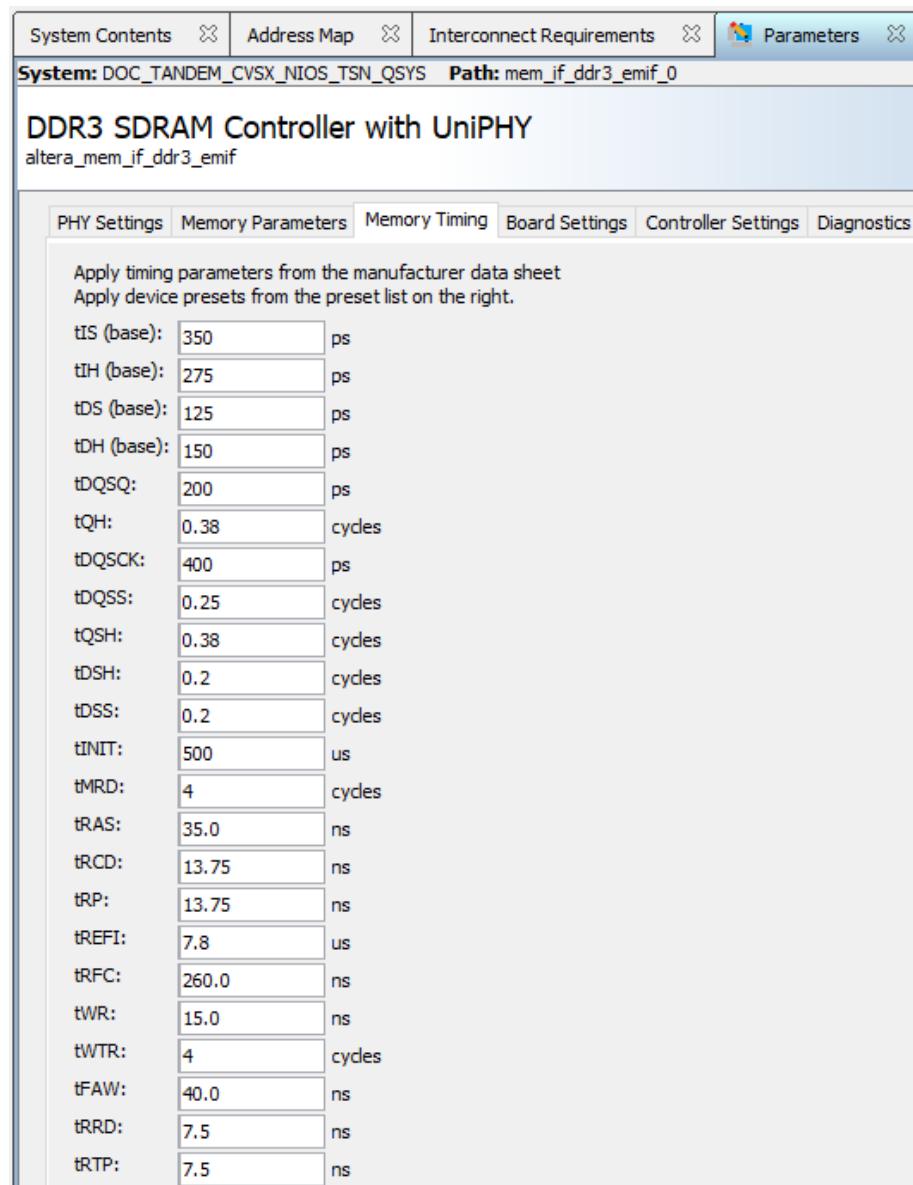
Figure 11. DDR3 SDRAM Settings


Figure 12. DDR3 SDRAM Settings

The screenshot shows the DDR3 SDRAM Controller with UniPHY settings window. At the top, there are tabs for System Contents, Address Map, Interconnect Requirements, Parameters, and a system path: DOC_TANDEM_CVSX_NIOS_TSN_QSYS / Path: mem_if_ddr3_emif_0.

Channel Signal Integrity:

- Derating Method: Use Altera's default settings Specify channel uncertainty
- 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

Board Skews:

- Restore default values
- Maximum CK delay to DIMM/device: 0.029132963 ns
- Maximum DQS delay to DIMM/device: 0.286640405 ns
- Minimum delay difference between CK and DQS: 5.828107E-4 ns
- Maximum delay difference between CK and DQS: 0.048251827 ns
- Maximum skew within DQS group: 0.001782034 ns
- Maximum skew between DQS groups: 0.02969881 ns
- Average delay difference between DQ and DQS: 0.0 ns
- Maximum skew within address and command bus: 0.096664741 ns
- Average delay difference between address and command and CK: 0.026243846 ns

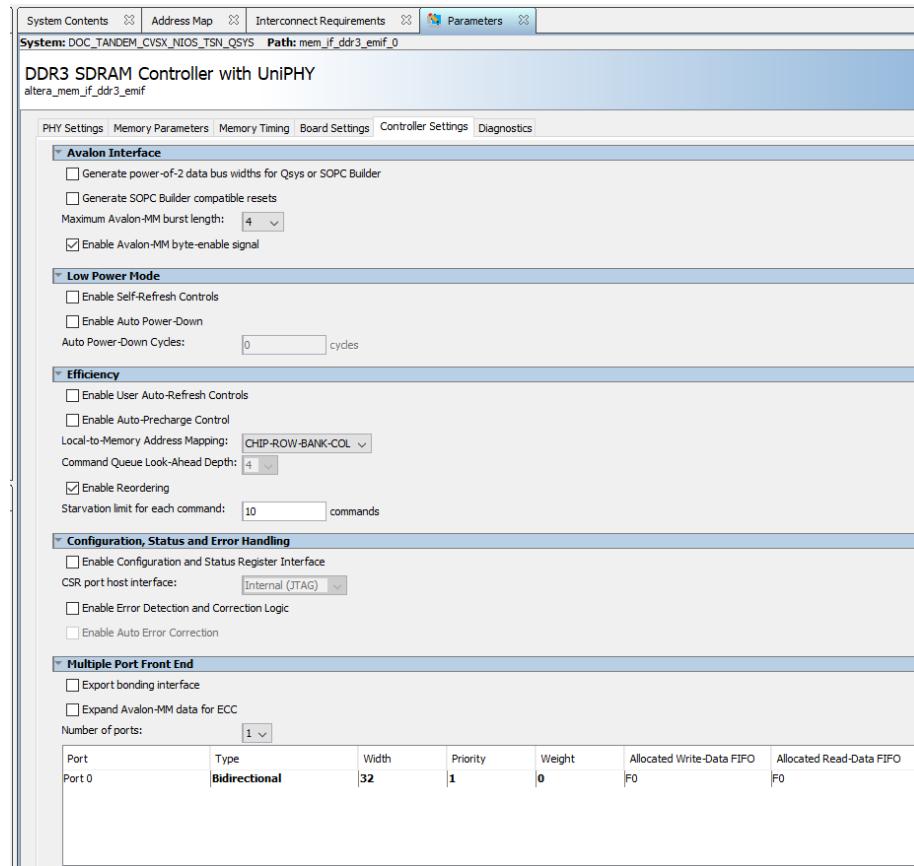
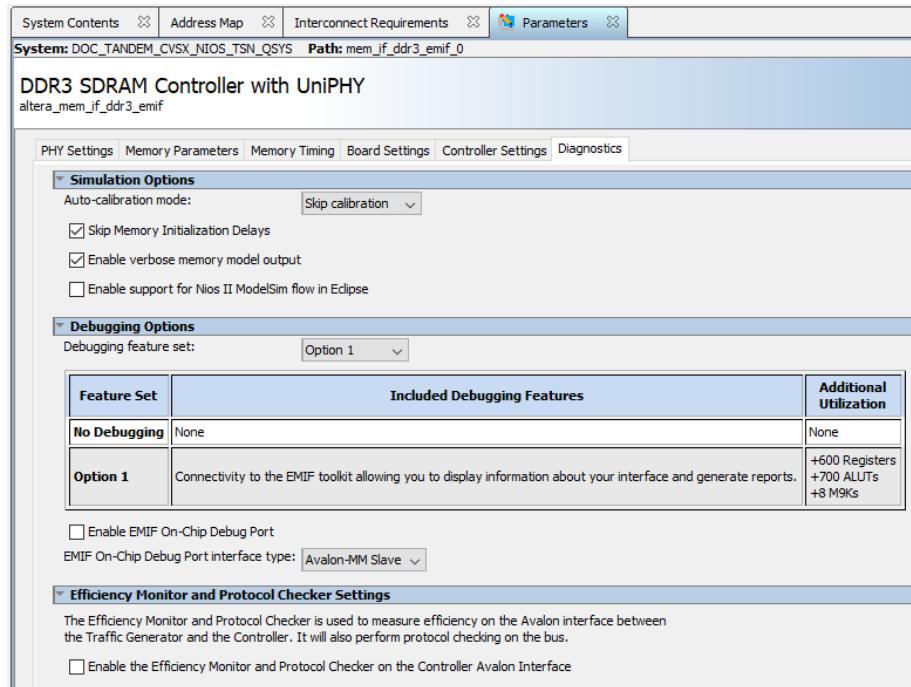
Figure 13. DDR3 SDRAM Settings


Figure 14. DDR3 SDRAM Settings



14. Set the address assignment and interrupt enumeration.

Table 4. Address Interrupt and Enumeration

Module Name	Port	Address Base and Interrupt Number
DOC_CPU	debug_mem_slave	0x0030_0000
	irq	IRQ0
TMC_I	s1	0x0000_4000
	s2	0x0000_4000
mem_if_ddr3_emif_0	avl_0	0x2000_0000
fast_periph_pb	s0	0x0100_0000
performance_counter_0	control_slave	0x0000
doc_tmr_0	s1	0x0040
	irq	2
doc_tmr_1	s1	0x0080
	irq	3
jtag_uart	avalon_jtag_slave	0x00c0
	irq	1
lvmc_dclink	lvdcdc_avs	0x0400_5000
drive0	drive_bridge_avs	0x0400_8000
	doc_adc_irq	0

continued...

Module Name	Port	Address Base and Interrupt Number
drive1	drive_bridge_avs	0x0400_c000
sys_console_debug_ram	s1	0x0400_1000
	s2	0x0400_1000
periph_ccb	s0	0x1000_0000
sysid_0	control_slave	0x0000_0000
	System ID	0x005046fe
IO_IN.Buttons	s1	0x0000_0020
IO_OUT.LED	s1	0x0000_0040
gate_drive_spi	spi_control_port	0x0000_0060
encoder_select	s1	0x0000_0080
epcq	avl_csr	0x0000_0100
	avl_mem	0x0200_0000
	interrupt_sender	4

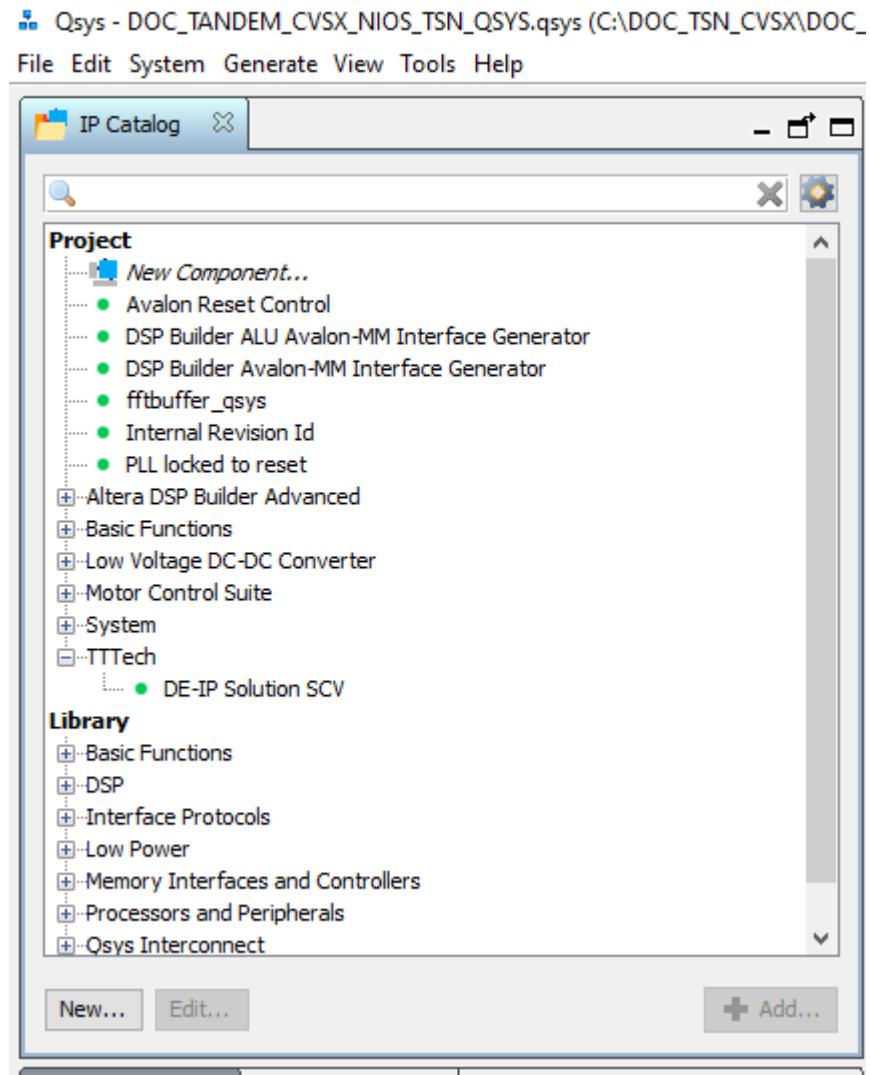
The TSN drive-on-chip design is now integrated in the Qsys system. you can generate the hardware and instantiate it in the top-level Verilog HDL. You can add new high-level pin assignments to the system (refer to *Example .qsf for Pin Assignments and Attributes*)

3.3. Adding the TTTech TSN IP to the Qsys system

You build the TSN system within the TSN drive-on-chip design based on TTTech TSN IP.

1. Copy the directory ip-sol-scv-release-2.3.0-30_03_2021\de_ip_solution_scv to <Quartus_Project>/ip and ip-sol-scv-release-2.3.0-30_03_2021\de-eval-board\fpga\misc_ip to <Quartus_Project>/ip.
2. In Qsys, click **Tools > Options** and add the newly copied directories to the **IP search paths**.
Qsys automatically adds the TTTech IP to the IP catalog in the left upper corner of Qsys.

Figure 15. TTTech TSN IP in IP Catalog



3. Navigate to ip-sol-scv-release-2.3.0-30_03_2021\de-eval-board\fpga and open de-eval_scv.qar with Intel Quartus Prime.
The design opens including a Qsys system. The blocks in this design are identical to the ones in the TTTech Reference Design.
4. In the Qsys system add the following functional blocks and connect them in the same way as the TTTech Reference Design.

Table 5. TSN subsystem blocks, addresses, and interrupts

Module Name	Description	Port	Address Base and Interrupt number
hps_0	Arria V/ Cyclone V Hard Processor System	f2h_sdram0_data	0x0000_0000
		f2h_sdram1_data	0x0000_0000
		f2h_ir10	IRQ 0
mm_bridge_0	Avalon-MM PipelineBridge	s0	0x0000_0000
sys_qsys_0	System ID Peripheral	control_slave	0x0001_0000
		System ID	0x1c7e0004
rev_id_0	Internal Revision ID	s	0x0004_0500
		QSYS_SVM_ID	165499264
pio_0	PIO (Parallel I/O)	s1	0x000f_0f00
eth_mdio_tristate_0	Altera's Ethernet MDIO master with tristate logic	csr	0x000f_0200
de_ip_solution_csv_0	DE-IP Solution SCV	s_deip	0x0200_0000
		irq0-ir115	0-15

5. Connect the TSN blocks with a 100 MHz clock from the drive-on-chip subsystem.
6. Add another **Altera PLL** block with the same 50 MHz reference clock as input (the same reference clock as the PLL for the drive-on-chip subsystem)

This additional 125 MHz clock is for the TSN IP `m_dma_0_clk` port.
7. Configure the TSN IP with these parameters:
 - Generics: `BUS_CDC 0, FES_PORT_HIGH 2, BUFFER_SIZE 0, SCHEDULED_PORTS 31, PREEMPTABLE_PORTS 30, FRER_PORTS 31`
 - Options: `LED_ACTIVE 0, Enable link LED interface (turn off)`
 - Adapters: Port 0 interface type DMA, Port 1 Interface Type MII, Port 2 Interface Type MII.
8. Keep the multiple PIO blocks in the TTTech TSN IP Reference Design Qsys System and change the equivalent to `pio_phy_reset` (`pio_0`).

Keep the other PIO modules such as `pio_led_debug`, `pio_pps_debug_out` and `pio_pps_debug_in`.
9. Configure the HPS module, even though the HPS and the secondary modules are connected in the same way as the TTTech TSN IP Reference Design Qsys system.

The HPS within the Cyclone V SoC Development board is connected to a different DDR3 memory than the TTTech Reference Board.

Figure 16. HPS Settings

The screenshot shows the HPS Settings configuration interface. At the top, there are tabs for System Contents, Address Map, Interconnect Requirements, and Parameters. Below that, the system path is listed as DOC_TANDEM_CVSX_NIOS_TSN_QSYS Path: hps_0. The main title is Arria V/Cyclone V Hard Processor System altera_hps. There are four tabs at the top: FPGA Interfaces, Peripheral Pins, HPS Clocks, and SDRAM. The General tab is selected, showing various enablement options for signals like MPU standby, general purpose signals, and various trace and debug interfaces. The AXI Bridges tab shows interface widths: FPGA-to-HPS (Unused), HPS-to-FPGA (32-bit), and Lightweight HPS-to-FPGA (Unused). The FPGA-to-HPS SDRAM Interface tab displays a table of two SDRAM ports, both named f2h_sdram0 and f2h_sdram1, with Type set to Avalon-MM Bidirectional and Width set to 64. The Resets tab lists several reset-related options.

Name	Type	Width
f2h_sdram0	Avalon-MM Bidirectional	64
f2h_sdram1	Avalon-MM Bidirectional	64

Figure 17. HPS Settings

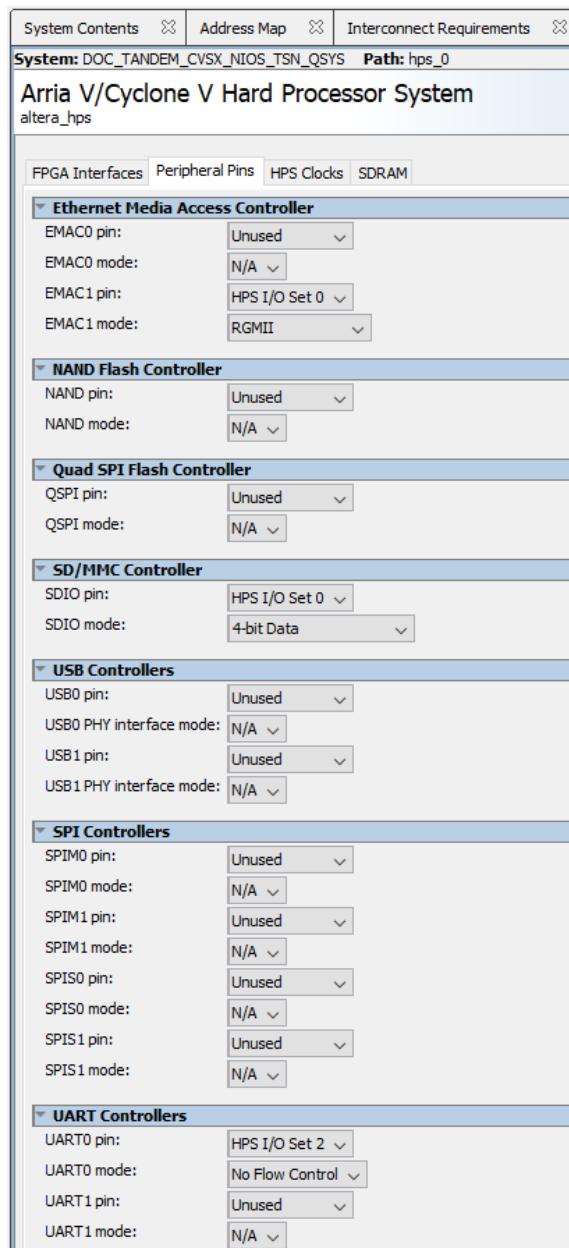


Figure 18. HPS Settings

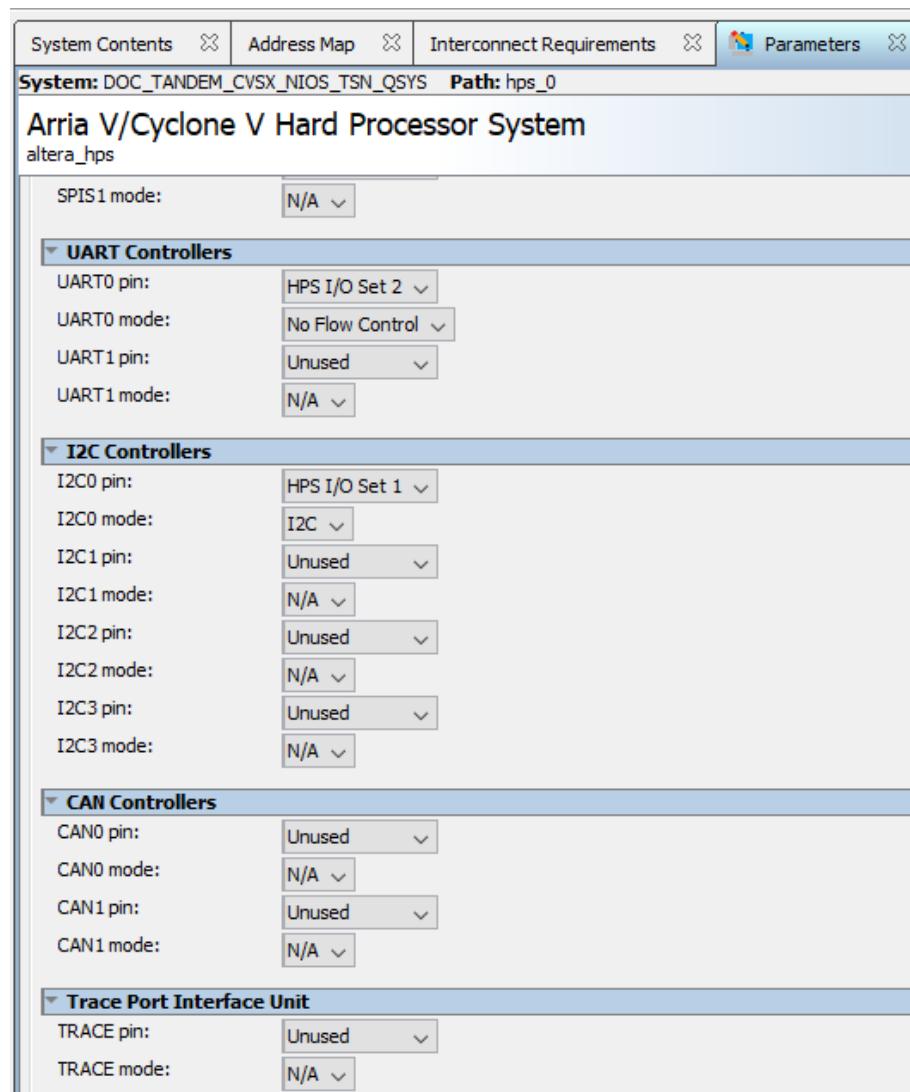


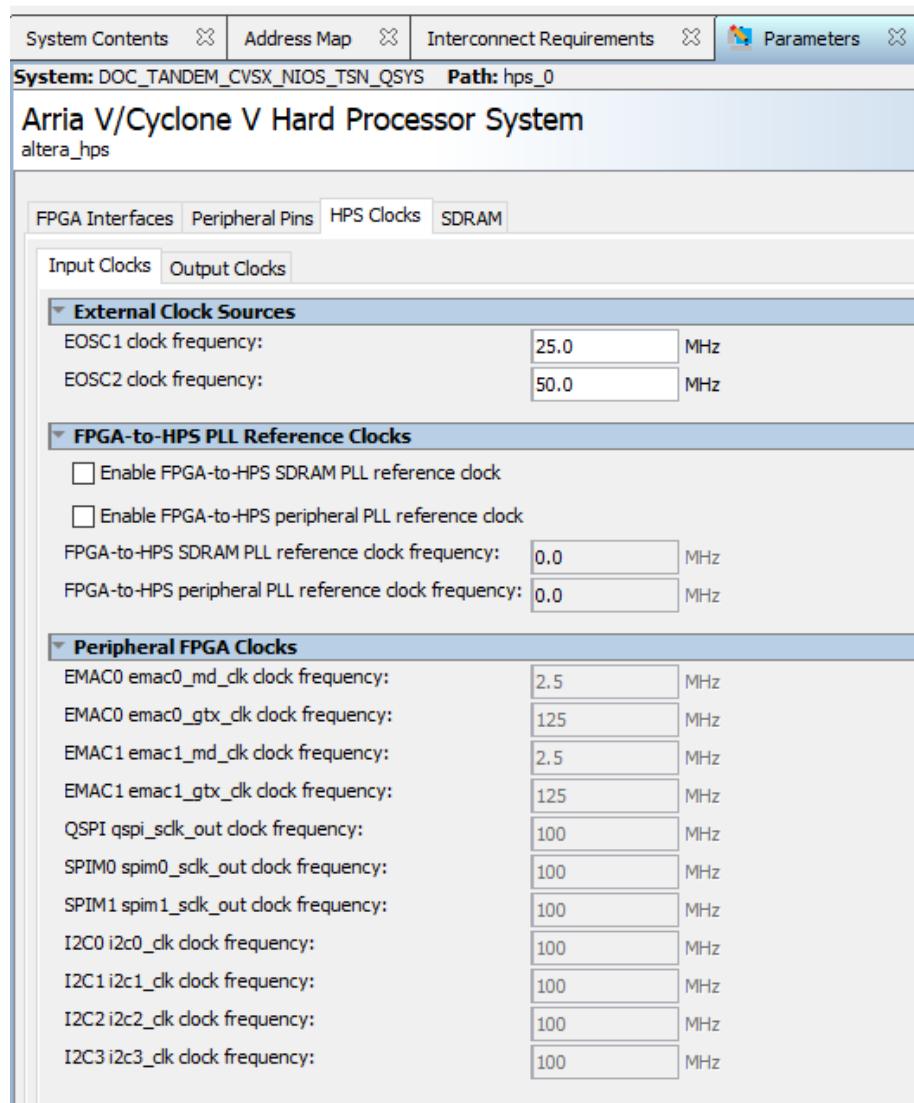
Figure 19. HPS Settings


Figure 20. HPS Settings

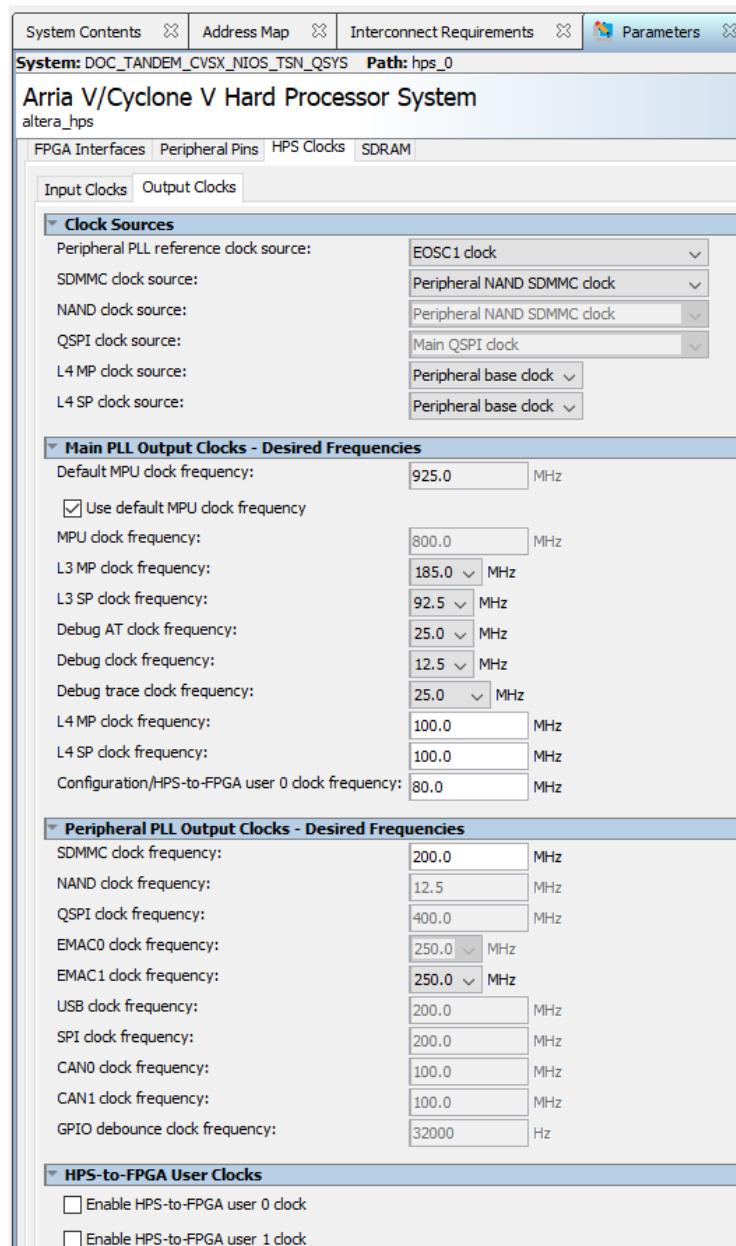


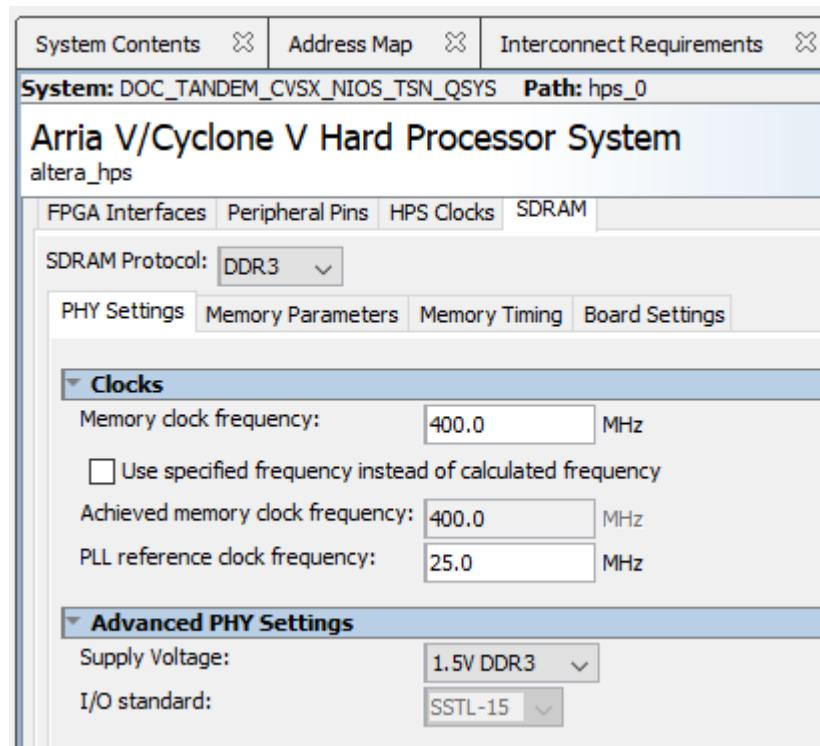
Figure 21. HPS Settings

Figure 22. HPS Settings

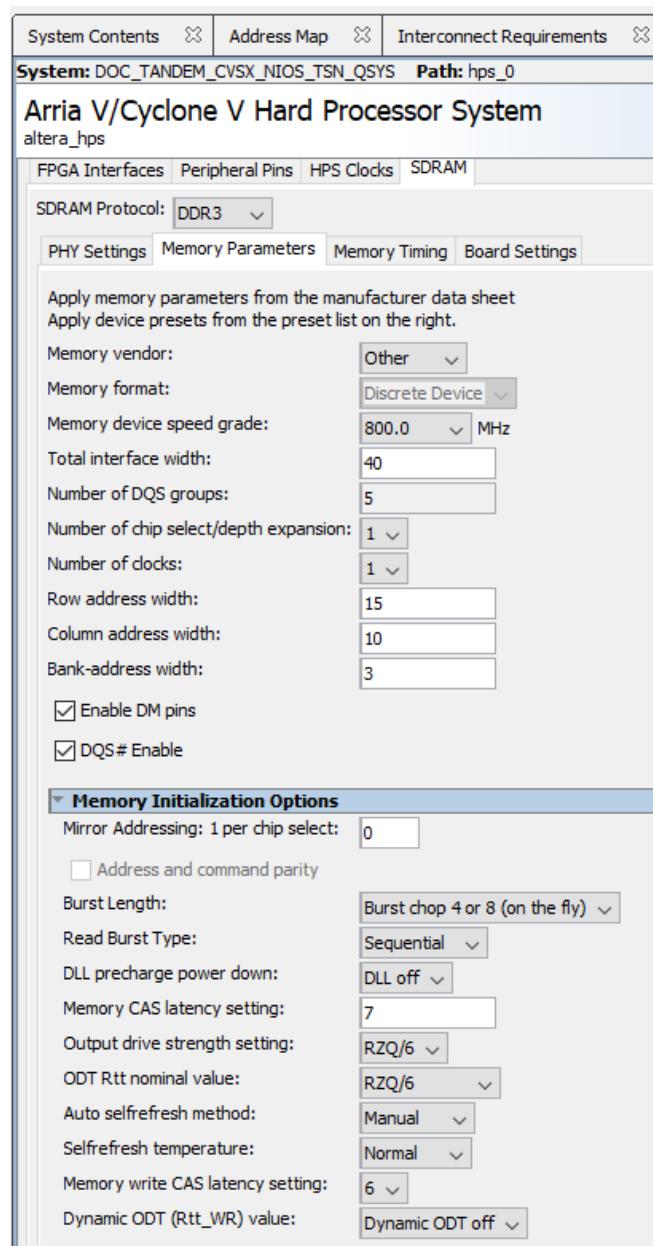


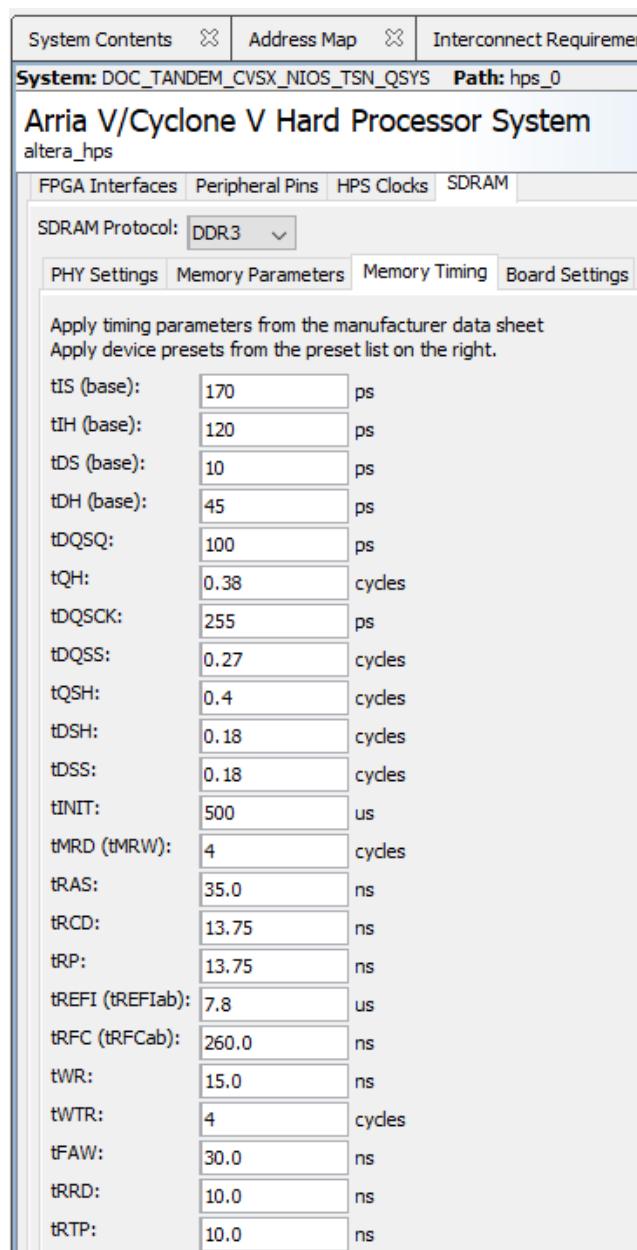
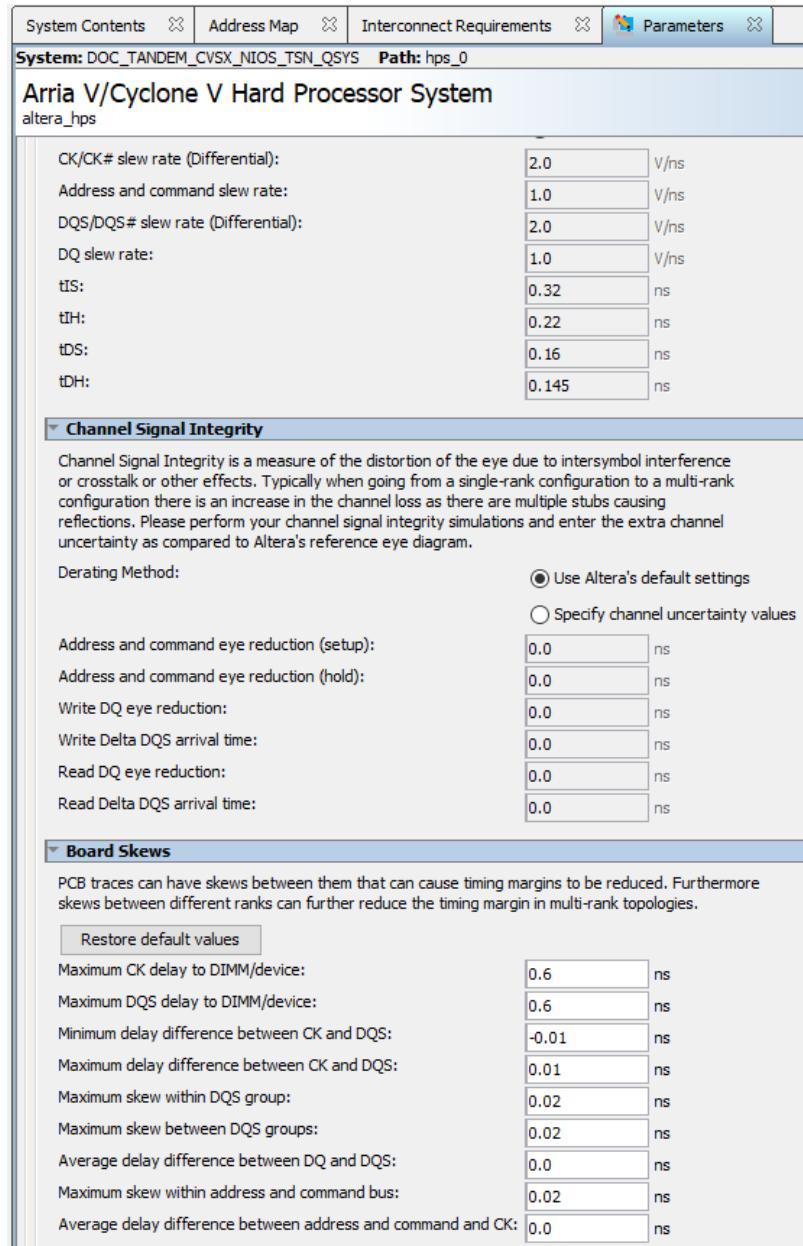
Figure 23. HPS Settings


Figure 24. HPS Settings



Both the drive-on-chip and TSN subsystems now work separately.

3.4. Connecting the TSN and Drive-on-Chip Subsystems

When connected, the HPS can control the TSN drive-on-chip design. You connect them via shared memory mechanism.

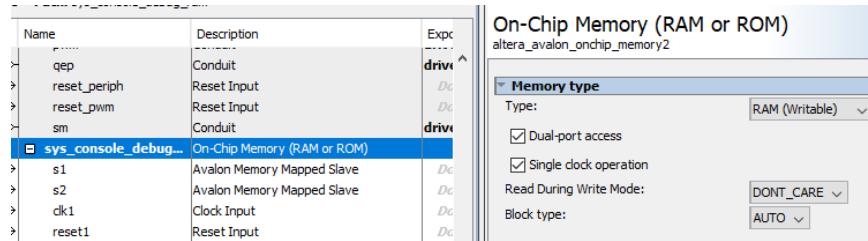
Intel recommends three connections.

1. Connect the port `h2f_axi_master` of the block `hps_0` of the TSN subsystem to the one port of the `sys_console_debug_ram` block.

- a. Enable Dual_port access in the `sys_console_debug_ram` block.

The first memory mapped slave port (`s1`) connects to the HPS. The second port (`s2`) connects to the Nios II soft processor. The drive-on-chip subsystem reads position, speed, configurations commands for the drives from this memory.

Figure 25. Turn on Share memory Dual-port access



2. Optionally, create a connection between the HPS and the FPGA DDR3 memory.
 - a. Connect the `h2f_axi_master` master port of the HPS to the `avl_0` port of the `mem_if_ddr3_emif_0` (external memory interface).

This connection allows the access to the drive-on-chip subsystem DDR3 memory from the HPS (from Linux)
3. To expose some peripherals of the TSN subsystem to the JTAG chain, add a connection between port `s0` (Avalon MM Slave) of the `mm_bridge_0` (TSN subsystem) to the port `master` of the **jtag_master** block in the drive-on-chip subsystem.

Figure 26. Qsys System

Name	Description	Export	Clock	Base
f2h_irq0	Interrupt Receiver	<i>Double-click to export</i>		
f2h_irq1	Interrupt Receiver	<i>Double-click to export</i>		
mm_bridge_0	Avalon-MM Pipeline Bridge	<i>Double-click to export</i>		
clk	Clock Input	<i>Double-click to export</i>	clk_100	
reset	Reset Input	<i>Double-click to export</i>	[dk]	
s0	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[dk]	0x00
m0	Avalon Memory Mapped Master	<i>Double-click to export</i>	[dk]	
sysid_qsys_0	System ID Peripheral	<i>Double-click to export</i>		
clk	Clock Input	<i>Double-click to export</i>	clk_100	
reset	Reset Input	<i>Double-click to export</i>	[dk]	
control_slave	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[dk]	0x00
rev_id_0	Internal Revision Id	<i>Double-click to export</i>		
s_rst	Reset Input	<i>Double-click to export</i>	[s_clk]	
s_clk	Clock Input	<i>Double-click to export</i>	clk_100	
s	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[s_clk]	0x00
pio_0	PIO (Parallel I/O)	<i>Double-click to export</i>		
clk	Clock Input	<i>Double-click to export</i>	clk_100	
reset	Reset Input	<i>Double-click to export</i>	[dk]	
s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[dk]	0x00
external_connection	Conduit	<i>Double-click to export</i>		
eth_mdio_tristate_0	Altera's Ethernet MDIO master with tri-state	<i>Double-click to export</i>		
reset	Reset Input	<i>Double-click to export</i>		
clk	Clock Input	<i>Double-click to export</i>	clk_100	
mdio	Conduit	<i>Double-click to export</i>		
csr	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[dk]	0x00
de_ip_solution_scv_0	DE-IP Solution SCV	<i>Double-click to export</i>		
rst	Reset Input	<i>Double-click to export</i>		
clk	Clock Input	<i>Double-click to export</i>	clk_100	
s_rst	Reset Input	<i>Double-click to export</i>		
s_clk	Clock Input	<i>Double-click to export</i>	clk_100	
s_deip	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[s_clk]	0x02
s_irq0	Interrupt Sender	<i>Double-click to export</i>		
s_irn1	Interrupt Sender	<i>Double-click to export</i>		

4. Click **Generate HDL**.

Related Information

[Example .qsf for Pin Assignments and Attributes](#) on page 57

3.5. Compiling the Quartus Prime Design and Top-Level Module

The TSN drive-on-chip Qsys project is wrapped into a top-level Verilog HDL file that instantiates and exposes the inputs and outputs of the Qsys module `DOC_TANDEM_CVSX_NIOS_TSN_QSYS`. To compile the project, you must have a TTTech TSN IP License.

1. Add the license to the Quartus license files (**Tools > License Setup**).
2. Compile the design to obtain the output .sof file.

Usually you configure FPGA with the .sof files. However, for this design, the FPGA image is stored in the EPCQ flash memory with the Nios II program.

Refer to *Example .qsf for Pin Assignments and Attributes* for the pin assignments for the TSN drive-on-chip design targeting the Cyclone V SoC Development Board. Apart from the pin assignment, some pins include attributes that are necessary for the correct behavior of the TSN IP and drive-on-chip design.

Related Information

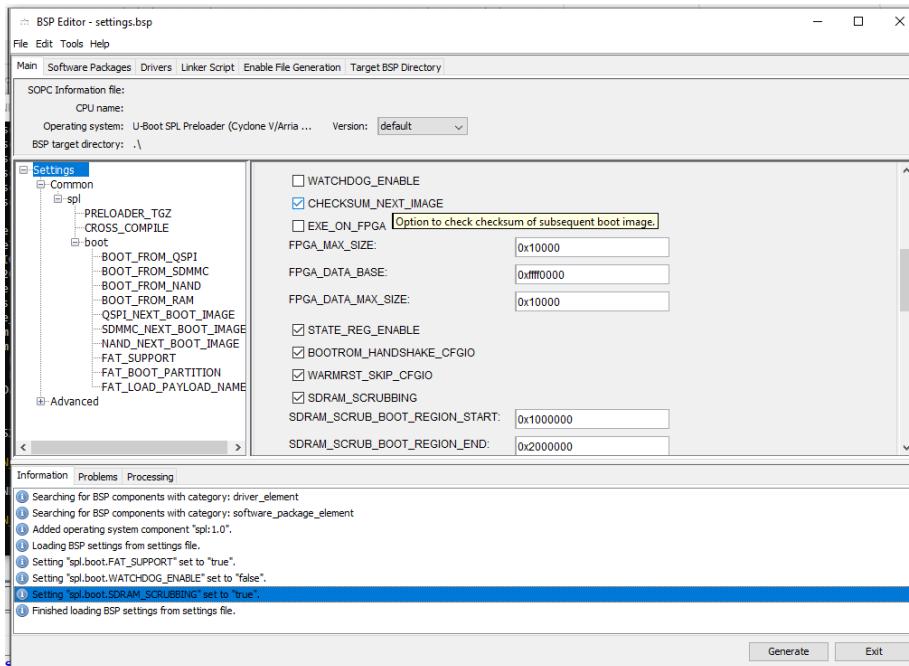
[Example .qsf for Pin Assignments and Attributes](#) on page 57

3.6. Generating the Preloader

The preloader is an essential component for the TSN drive-on-chip design HPS to boot-up. The Intel Quartus compilation generates a directory `hps_isw_handoff`. Every time you modify the Qsys system or the Intel Quartus Prime project, you must generate a new preloader and include it in the SD card image for the HPS.

1. Open the SoC EDS Command Shell and navigate to the directory `/<project_location>/software_hps/spl_bsp` (or create it within the project files) and run the `bsp-editor`.
- `bsp-editor.exe`
2. Click **File > New HPS BSP...**
 3. Set the **Preloader settings directory:** to `<project_location>/hps_isw_handoff/DOC_TANDEM_CVSX_NIOS_TSN_QSYS_hps_0`
 4. Turn off **Use default locations** and change the target directory to `<project location>/software_hps/spl_bsp`, click **OK**.
 5. In the BSP Editor settings window, turn on FAT support, turn off the watchdog, and turn on **SDRAM_SCRUBBING**

Figure 27. BSP editor to generate preloader



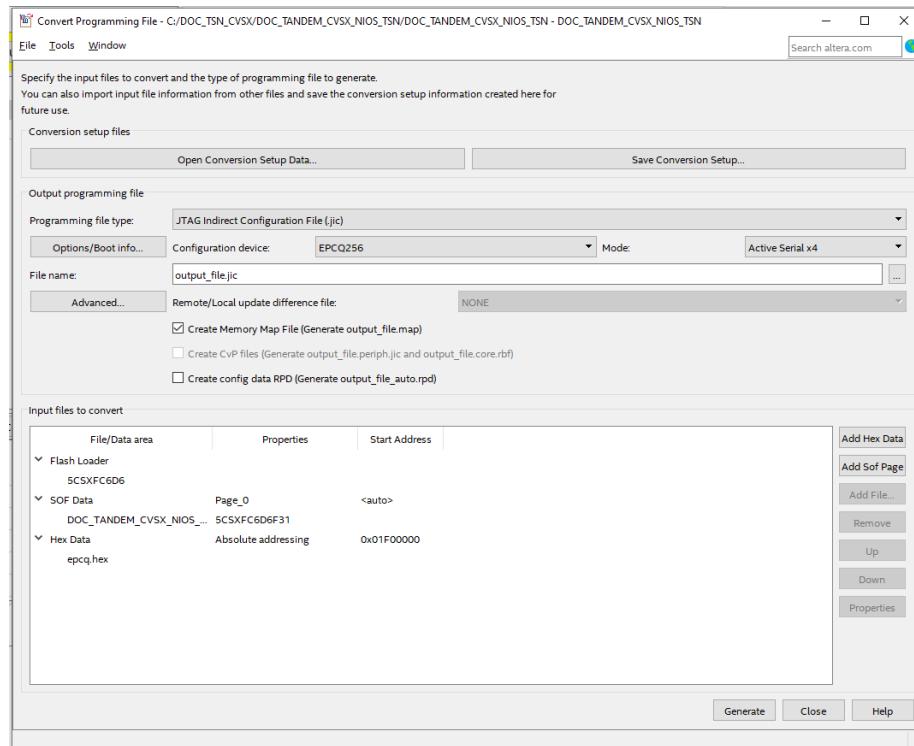
6. Press **Generate** and close the window
7. In the SoC EDS Command Shell navigate to <project_location>/software_hps/spl_bsp.
8. Do a make clean and then a make all.
If you see any error regarding tar.gz file decompression, use /usr/bin/make.exe to compile.

The file preloader-mkpimage.bin generates, which integrates into the SD card image for the HPS booting process.

3.7. Generating a .jic file

The EPCQ flash memory programs the FPGA from when you turn on the development board. A single .jic file contains the .sof file and a .hex file (for the Nios II drive on-chip design).

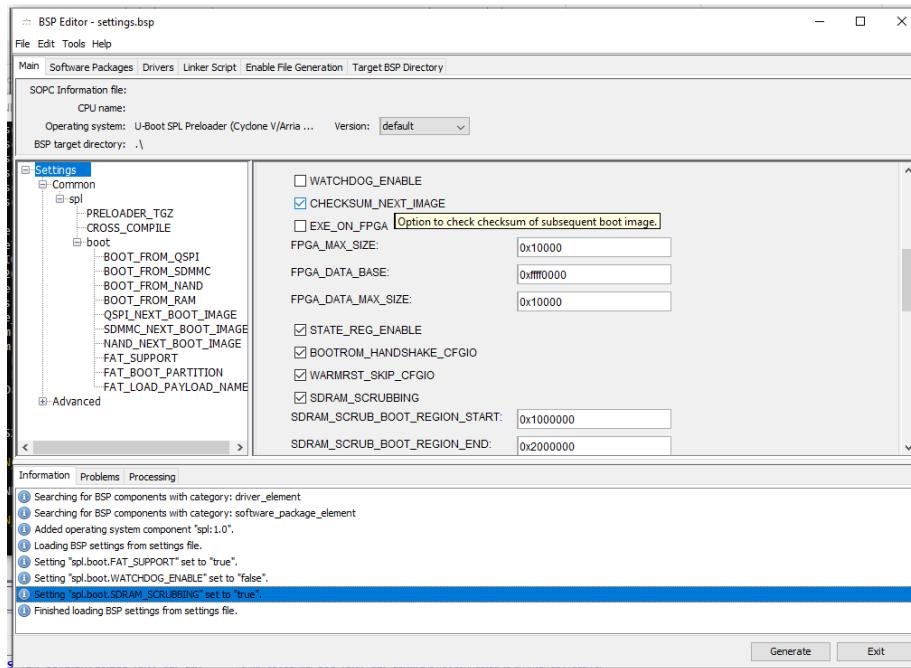
1. In Intel Quartus Prime, click **File > Convert Programming Files...**
2. Select the settings:
 - a. For **Programming file type:** select **JTAG Indirect Configuration File (.jic)**
 - b. For **Configuration device:** select **EPCQ256**
 - c. For **Mode:** select **Active Serial x4**
 - d. Add a **SOF Data Page_0** with .sof from the Intel Quartus Compilation.
 - e. Add the .hex file from software/ DOC_TANDEM_CVSX_NIOS/mem_init/epcq.hex.
This program is for the Nios II soft processor controlling the drive-on-chip design.
 - f. Add the flash loader for the device 5CSXFC6D6
 - g. Click **Generate**.

Figure 28. Generate a .jic File


3.8. Compiling the Drive-on-Chip Design Software in Nios II Software Build Tools

1. In Intel Quartus Prime, click **Tools > Nios II Software Build Tools for Eclipse**.
2. Set the workspace to <project_location>/software/
3. In Nios II Eclipse, click **File > Import...**
4. In the window expand **General, Existing Projects into Workspace**. then **Browse...** and click **Select root directory**.

Figure 29. Import project into Nios II Software Build Tools



5. Select the project directory and the bsp directory.
6. In Nios II Eclipse GUI, right click into the BSP directory and click **Nios II > BSP Editor...**
7. Check that the SOPC Information file is set to <project_location>/DOC_TANDEM_CVSX_NIOS_TSN_QSYS.sopcinfo
8. In the **Drivers** tab, select **enable_jtag_uart_ignore_fifo_full_error**. This is the only non-default option to turn on.
9. Click **Generate**.
10. In the Nios II Eclipse SBT, right click in the BSP directory, click **Nios II > Generate BSP**. The BSP generates.
If you see errors because of the TTTech TSN IP during the Nios II software compilation or BSP generation, create a separate Qsys system without the **de_ip_solution_scv_0** block (TSN IP) and compile the Nios II using the .sopcinfo file.
11. Apply the following changes in the source code <project_location/software/DOC_TANDEM_CVSX_NIOS_TSN>:

- Inside /platform/common/platform.h add another family of device name for example: #define SYSID_CVSX_NIOS 6
 - The device family should match the System ID set in the 0x005046fe sysid_0 block in Qsys.
 - Add the definition to platform.c in the variable device_family_t and TANDEM_FAMILIES
 - Add the family in other locations where device_family definition might appear, for example in motor_task.c
 - Inside demo_cfg.h, change the ACDS_MAJOR_VERSION to 5 or any other number, but that matches the System ID's sixth digit (e.g. 0x005046fe).
12. In Nios II Eclipse, right click in the project directory, and click **Build Project**.
13. Generate the .hex file for the EPCQ memory that is in the .jic: right click in the project directory and click **Make Targets > Build....**
14. In the window select **mem_init_generate** and click **Build**

The .hex file to include in the .jic file generation is in <project_location>/software/DOC_TANDEM_CVSX_NIOS_TSN/mem_init/epcq.hex

3.9. Launching a YOCTO Build

The SD card image for the HPS in the TSN drive-on-chip design is entirely based on the framework provided by TTTech. To build all the components into an SD card for HPS booting, you need a new YOCTO build to migrate from the DE-EVAL-BOARD (Reference Design) to the Cyclone V SoC Development board.

You must run these steps on a Linux Machine.

1. Get the TTTech IP .zip file and decompress it.
2. Navigate to de-eval-board/build and decompress de-evaluation-board-src.tar.gz, which contains all the files for the YOCTO build.
3. Manually include the preloader and create the SD card image using individual components.

The patch includes the following changes:

- de-eval-board/socfpga.c: change the offset to access the EEPROM that stores the MAC addresses in the Cyclone V SoC Development Board
- sdcard-image_1.0.0.bb: add additional software to the build including the rt-tests.
- ttt-ip-init.sh: TTTech IP initialization script. Change the function to access the MAC addresses for the interfaces from EEPROM and set the PHY delays for the interfaces.
- de-eval-board/ interfaces: change the IP address and remove additional switch ports (only two out of four are used)
- /machine/de-eval-board.conf: add tar.gz as root file system output extension, to generate the root file system for the SD card image.

- `/de-eval-board/u-boot.script`: disable programming the FPGA fabric from the SD card. The image for the FPGA comes from the EPCQ memory in this description of the solution.
- `socfpga_cyclone5_de-eval-board_default.dts`: change the register address for the EEPROM refer to *Script to read and change MAC addresses from Cyclone V SoC EEPROM*.
- `Linux-tttech-industrial_5.4.bb`: add the RT patch (optional).

3.10. Building an SD Card Image for the TSN Drive-on-Chip Design Example

Files names from the YOCTO build are just examples. The file name can change depending on the versions and dates when the YOCTO build was launched.

1. Copy and rename from `de-eval-board/build/build/tmp/deploy/images/de-eval-board` (in a separate temporarily directory) the following files:
 - `u-boot-de-eval-board-2018.03+gitAUTOINC+f95ab1fb6e-r0.img` to `u-boot.img` o `u-boot.scr`
 - `u-boot.scr`
 - `SCV-DE-EVAL-DMA-BIT-BITSTREAM_S-DE-IP-B-11-006-1.6.0-r0.rbf` to `socfpga.rbf`
 - `socfpga_cyclone5_de-eval-board-5.4.40-rt24+git0+f015b86259-r0-de_eval_board-20210728152840.dtb` to `socfpga_cyclone5_de_eval_board.dtb` and `socfpga.dtb`
 - `zImage-5.4.40-rt24+git0+f015b86259-r0-de-eval-board-20210728152840.bin` to `zImage`
2. Create a directory named `rootfs` and copy `sdcard-image-de-eval-board-20200504095930.rootfs.tar.gz`
3. Decompress using `tar -xvzf sdcard-image-de-eval-board-20210728152840.rootfs.tar.gz`
4. Delete `sdcard-image-de-eval-board-20210728152840.rootfs.tar.gz`.

Figure 30. Root Files Systemrootfs Directory

```
Ubuntu16: ~/DOC_TSN23_IMAGE/rootfs
demo@Ubuntu16:~/DOC_TSN23_IMAGE$ ls -ltr
total 364956
-rwxr-xr-x 1 demo demo index 23772 Jul 28 14:26 make_sdimage.py
-rw-r--r-- 1 demo demo 851 Jul 28 14:44 u-boot.scr
-rw-r--r-- 1 demo demo 397716 Jul 28 14:44 u-boot.img
-rw-r--r-- 1 demo demo 4018180 Jul 28 14:45 socfga.rbf
-rw-r--r-- 1 demo demo 19936 Jul 28 14:45 socfga.dtb
-rw-r--r-- 1 demo demo 4558200 Jul 28 14:46 socfga_cyclone5_de-eval-board.dtb
-rw-r--r-- 1 demo demo 63020491 Jul 28 14:46 sdcard-image-de-eval-board.tar.gz
drwxr-xr-x 18 demo demo 4096 Jul 28 14:47 rootfs
-rw-rw-r-- 1 demo demo 11788 Jul 28 15:48 cvsx_doc_tsn_2_3.patch
-rw-r--r-- 1 demo demo 262144 Aug 5 11:05 preloader-mkpimage.bin
-rw-r--r-- 1 root root 3221225472 Aug 5 11:05 TSN_DoC_SD.img
demo@Ubuntu16:~/DOC_TSN23_IMAGE$ cd ./rootfs/
demo@Ubuntu16:~/DOC_TSN23_IMAGE/rootfs$ ls -ltr
total 64
drwxrwxr-x 2 demo demo 4096 Jul 14 17:55 tmp
dr-xr-xr-x 2 demo demo 4096 Jul 14 17:55 sys
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 run
dr-xr-xr-x 2 demo demo 4096 Jul 14 17:55 proc
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 mnt
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 media
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 dev
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 boot
drwxr-xr-x 10 demo demo 4096 Jul 14 19:58 usr' >> conf/local.conf
drwxr-xr-x 9 demo demo 4096 Jul 14 21:15 var
lrwxrwxrwx 1 demo demo 19 Jul 28 14:31 linuxrc -> /bin/busybox.nosuid
drwxr-xr-x 3 demo demo 4096 Jul 28 14:32 home
drwxr-xr-x 6 demo demo 4096 Jul 28 14:32 lib"lx-altera-lts"' >> conf/local.conf
drwxr-xr-x 2 demo demo 4096 Jul 28 14:32 bin"4.23"' >> conf/local.conf
drwxr-xr-x 3 demo demo 4096 Jul 28 14:32 libexec
drwxr-xr-x 2 demo demo 4096 Jul 28 14:32 sbin
drwxr-xr-x 42 demo demo 4096 Jul 28 14:32 etc
demo@Ubuntu16:~/DOC_TSN23_IMAGE/rootfs$
```

5. In the temporarily folder copy the script file to generate an image (make_sdimage.py). The current working directory should look like the following picture.

Figure 31. SD Card Image Components

Intel generated the file preloader-mkpimage.bin using bsp-editor.

```
Ubuntu16:~/DOC_TSN23_IMAGE$ ls -ltr
total 364956
-rwxr-xr-X 1 demo demo index 23772 Jul 28 14:26 make_sdimage.py
-rw-r--r-- 1 demo demo 851 Jul 28 14:44 u-boot.scr
-rw-r--r-- 1 demo demo 397716 Jul 28 14:44 u-boot.img
-rw-r--r-- 1 demo demo 4018180 Jul 28 14:45 socfpga.rbf
-rw-r--r-- 1 demo demo 19936 Jul 28 14:45 socfpga.dtb
-rw-r--r-- 1 demo demo 19936 Jul 28 14:45 socfpga_cyclone5_de_eval_board.dtb
-rw-r--r-- 1 demo demo 4558200 Jul 28 14:46 zImage.gpt
-rw-r--r-- 1 demo demo 63020491 Jul 28 14:46 sdcard-image-de-eval-board.tar.gz
drwxr-xr-X 18 demo demo 4096 Jul 28 14:47 rootfs
-rw-rw-r-- 1 demo demo 11788 Jul 28 15:48 cvsx_doc_tsn_2_3.patch
-rw-r--r-- 1 demo demo 262144 Aug 5 11:05 preloader-mkpimage.bin
-rw-r--r-- 1 root root 3221225472 Aug 5 11:05 TSN_Doc_SD.img
demo@Ubuntu16:~/DOC_TSN23_IMAGES cd ./rootfs/
demo@Ubuntu16:~/DOC_TSN23_IMAGES$ ls -ltr
total 64
drwxrwxr-x 2 demo demo 4096 Jul 14 17:55 tmp
dr-xr-xr-X 2 demo demo 4096 Jul 14 17:55 sys
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 run
dr-xr-xr-X 2 demo demo 4096 Jul 14 17:55 proc
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 mnt
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 media
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 dev
drwxr-xr-x 2 demo demo 4096 Jul 14 17:55 boot
drwxr-xr-x 10 demo demo 4096 Jul 14 19:58 usr -> conf/local.conf
drwxr-xr-x 9 demo demo 4096 Jul 14 21:15 var
lrwxrwxrwx 1 demo demo 19 Jul 28 14:31 linuxrc -> /bin/busybox.nosuid
drwxr-xr-x 3 demo demo 4096 Jul 28 14:32 home
drwxr-xr-x 6 demo demo 4096 Jul 28 14:32 lib
drwxr-xr-x 2 demo demo 4096 Jul 28 14:32 bin 1.23% -> conf/local.conf
drwxr-xr-x 3 demo demo 4096 Jul 28 14:32 libexec
drwxr-xr-x 2 demo demo 4096 Jul 28 14:32 sbin
drwxr-xr-x 42 demo demo 4096 Jul 28 14:32 etc
conf
demo@Ubuntu16:~/DOC_TSN23_IMAGE$
```

- Run the following command to generate the SD card image TSN_DoC_SD.img

```
>> sudo ./make_sdimage.py -f -P preloader-mkpimage.bin,u-boot.img,num=3,format=raw,size=250M,type=A2 -P rootfs/,num=2,format=ext2,size=2G -P zImage,u-boot.scr,socfpga.rbf,u-boot.img,socfpga_cyclone5_de_eval_board.dtb,socfpga.dtb,num=1,format=vfat,size=250M -s 3G -n TSN_DoC_SD.img
```

- Create an image onto an SD card using an imager tool like Disk Imager. The system is ready to boot-up.
- Download: http://releases.rocketboards.org/release/2016.01/sgmiiid/tools/make_sdimage.py

Related Information

[Python script to build SD card images](#)

3.11. Changing MAC Addresses

The TTTech TSN IP works with MAC addresses for the network interfaces. For the Cyclone V SoC Development Board, the EEPROM configuration stores these addresses. When the complete systems boots, it takes the addresses from this EEPROM and assigns them to different interfaces as described in the ttt-ip-init.sh file.

- Type the i2c command after the Cyclone V SoC Development board boots. For example:

```
>> i2cset -y 0 0x51 0 56 0xac i
```

2. Check your changes by typing the following command:

```
>> i2cset -y 0 0x51 0 56 i
```

```
>> i2cget -y 0 0x51
```

When you change the MAC addresses in the EEPROM, they remain after switching off the Development board. Intel recommends you reboot the Development board and use a fresh SD card image as some of the previously undesired addresses might copy into the system configuration.

3.12. Reading and Checking Physical Addresses on the Cyclone V SoC Development Board

1. Use the script `Read_Write_MAC_CVSX.sh` to check the physical addresses in the Cyclone V SoC development board:

To read type:>> ./Read_Write_MAC_CVSX.sh read

To write type:>> ./Read_Write_MAC_CVSX.sh write <MAC address> <offset>

For example >> ./Read_Write_MAC_CVSX.sh write AA:BB:CC:DD:FF:GG 45"

This script does not validate the address.

Related Information

[Script to read and change MAC addresses from Cyclone V SoC EEPROM](#) on page 94

4. Running HPS Software for the TSN Drive-on-Chip Design

4.1. Installing the Arm Compiler for Motor Control from Linux

When the system is up and running you can access the Linux terminal from the serial port of the Cyclone V SoC Development board or by Ethernet connection using "ssh". You can compile the programs that run in the HPS (Arm processor) in different ways using Arm compiler from Linux command line or with DS-5 for Eclipse.

1. On a Linux machine, install the Arm compiler and other necessary tools:

```
>> sudo apt-get install gcc-arm-linux-gnueabihf g++-arm-linux-gnueabihf
>>sudo apt-get install make cmake
>>sudo apt-get install build-essential
```

4.2. Creating a Basic C Program for Motor Control

Creating a program to talk to the drive-on-chip subsystem from the Linux system uses the shared memory mechanism implemented during the Qsys system build.

1. In the C program, set up the memory locations by using defines:

```
#define DEBUG_RAM_BASE 0xc4001000
#define DEBUG_ADDR_SPACE_PER_AXIS 64
/* Offsets into debug mem */
#define DOC_DBG_SPEED 10
#define DOC_DBG_POSITION 12
#define DOC_DBG_I_PI_KP 18
#define DOC_DBG_I_PI_KI 19
#define DOC_DBG_SPEED_PI_KP 20
#define DOC_DBG_SPEED_PI_KI 21
#define DOC_DBG_SPEED_SETPO 22
#define DOC_DBG_POS_SETPO 25
#define DOC_DBG_WAVE_DEMO_MODE 29
#define DOC_DBG_POS_SPEED_LIMIT 30
#define DOC_DBG_POS_PI_KP 31
#define DOC_DBG_TRACE_DEPTH 58
```

These parameters are identical to the ones described in the motor control application program (Nios II program)..

2. Create functions to read and write to the motor control application:

```
//Read debug mem
unsigned int get_motor_param(unsigned int axis, unsigned int param) {
    return debug_base[axis * DEBUG_ADDR_SPACE_PER_AXIS + param];
}
//Write debug mem
void set_motor_param(unsigned int axis, unsigned int param, unsigned int val) {
    debug_base[axis * DEBUG_ADDR_SPACE_PER_AXIS + param] = val;
}
```

3. In the main map the memory by using the following lines:

```
int _fdmem;
const char memDevice[] = "/dev/mem";
_fdmem = open(memDevice, O_RDWR | O_SYNC);
if (_fdmem < 0) {
    printf("Failed to open the /dev/mem !\n");
    return 0;
} else {
    printf("open /dev/mem successful!\n");
}
/* mmap() debug memory */
int debug_size = 512;
printf("Mapping %i bytes for debug_data\n", debug_size);
debug_base = (unsigned int*)(mmap(0, debug_size, PROT_READ |
PROT_WRITE, MAP_SHARED, _fdmem, DEBUG_RAM_BASE));
if ((void *)debug_base == MAP_FAILED) {
    printf("Failed to map debug memory\n");
    return -1;
} else {
    printf("Debug mem base : %x\n", (unsigned int)debug_base);
}
```

4. Test the motor movement by reading and writing to the drive-on-chip subsystem, for example to set the motor to the position 100 degrees:

```
set_motor_param(0, DOC_DBG_WAVE_DEMO_MODE, 1); //Set the axis 0 to
position mode
set_motor_param(1, DOC_DBG_WAVE_DEMO_MODE, 1); //Set the axis 1 to
position mode
set_motor_param(0, DOC_DBG_SPEED_SETPO, 100); //Set axis 0 pos to 100
degrees
set_motor_param(1, DOC_DBG_SPEED_SETPO, 100); //Set axis 1 pos to 100
degrees
```

5. In the same way use the function `get_motor_param` to retrieve information from the drive-on-chip subsystem such as position of the shaft, speed, motor control parameters like Kp, Ki or set the maximum speed limit..
6. Compile the C program using the following command:

```
>> arm-linux-gnueabihf-gcc <program_name>.c -O3 -march=armv7-a
```

7. Copy the program to the development board via `scp` and test the motor control.

4.3. OPC UA PubSub based on open62541

When the embedded device is running, you can run C compiled programs on it. You can build and generate an OPC UA PubSub executable using DS-5 as the platform to develop C code for the Cyclone V Soc Development board. You can use DS-5 or standalone compilation to combine motor control applications using OPC UA PubSub (over TSN). In the [open62541 GitHub](#) has examples that run on the TTTech TSN IP reference design. Combine a OPC UA Subscriber and achieve motor control using PubSub and configure the TSN capabilities accordingly.

Related Information

- https://github.com/open62541/open62541/blob/master/examples/pubsub/pubsub_subscribe_standalone.c
- https://github.com/open62541/open62541/blob/master/examples/pubsub/tutorial_pubsub_subscribe.c

- https://github.com/open62541/open62541/blob/master/examples/pubsub_realtime/pubsub_TSN_loopback_single_thread.c

4.4. Getting and Building open62541

The OPC UA Publisher and Subscriber is based on the open source implementation of the OPC UA standard open62541.

1. Access and download the standard from GitHub.
2. Download and build the library following Open62541 documentation.
3. Use an amalgamation for the Cyclone V SoC Development board of the open62541 library, allowing you to have all the functions and definitions for the program in a single .c and .h file
4. Copy these files to the project in DS-5 and name open62541.c and open62541.h
5. When building the open6541 library in a Linux machine, enable the following flags:

```
UA_ENABLE_AMALGAMATION  
UA_ENABLE_PUBSUB  
UA_ENABLE_PUBSUB_ETH_UADP
```

Related Information

- [Github Standard](#)
- [Open62541 Documentation](#)

4.5. Setting up Debug Sessions in DS-5

The SD card image includes a version of gdbserver and sftp-server that are necessary to setup a debug session in DS-5.

1. Connect the Cyclone V SoC Development board using RJ-45 ENENT1 or ENET2 connector to a PC with DS-5. Using command line ssh into the development board.

```
>> ssh root@192.168.1.20
```
2. Start a sftp session:

```
>> sftp root@192.168.1.20
```

Figure 32. Using SFTP and SSH with Cyclone V SoC Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

S C:\Users\mevalver> ssh root@192.168.1.20
root@de-eval-board:~# ls
otorControl_only ttt-sw-update.log
root@de-eval-board:~# exit
logout
onnection to 192.168.1.20 closed.
S C:\Users\mevalver> sftp root@192.168.1.20
onnected to root@192.168.1.20.
ftp> ls
otorControl_only ttt-sw-update.log
ftp>
```

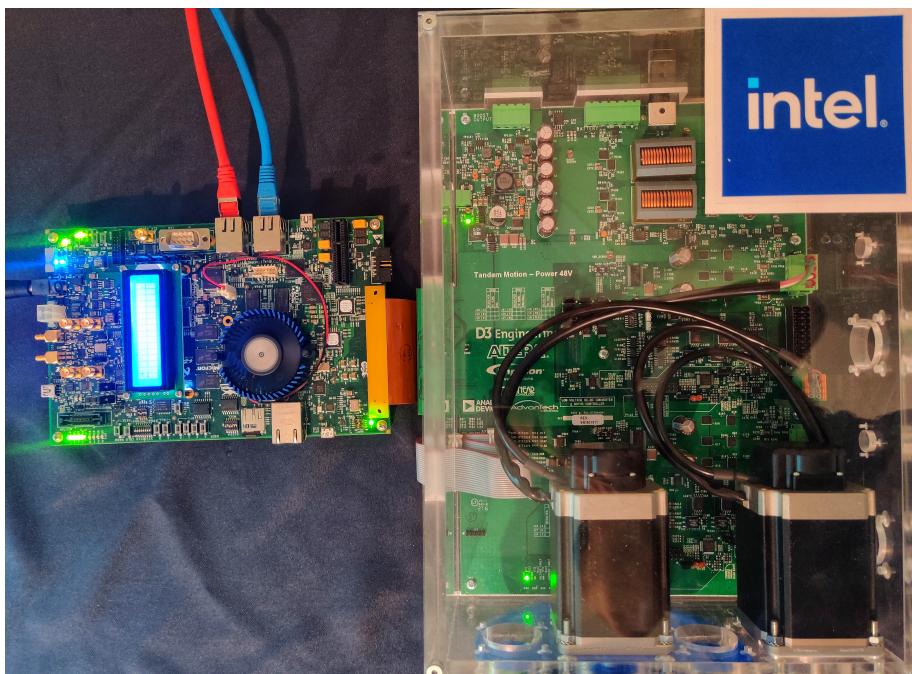
5. Connecting the Cyclone V SoC Development board to the Tandem 48 V Motion-Power board

You can connect two or more Cyclone V SoC Development boards in serial acting as motor control-end point or switches due to the inclusion of the TTTech TSN IP in the FPGA fabric.

Two Ethernet ports labeled on the board as ENET1 and ENET2 are available to the TTTech TSN IP. You can use the Cyclone V SoC development board as an endpoint for motor control or it can be a two-port switch in any network.

1. Connect with the HSMC port

Figure 33. Cyclone V SoC Development board connected to the Tandem 48 V Motion-Power board



Related Information

[Tandem Motion-Power 48 V Board](#)

6. Running the Program

You can produce executable programs using DS-5 environment for Arm processor or standalone compilation using a Linux machine.

1. Compile your program and copy the executable to the Cyclone V SoC Development board, for example, using the `scp` command:

```
>> scp -rp .\<your_program> root@192.168.1.20:/home/root
```

2. To run the program remotely, ssh into the Cyclone V Soc development board or:

```
>> ssh root@192.168.1.20 "/home/root/<your_program> > /dev/null"
```

7. TSN Configuration Example

The TSN configuration relates to the application that the Cyclone V SoC Development board runs. For more details about PTP, Qbv, TSN, VLAN.; refer to the *TTTech TSN IP Reference Design for DE-IP-SCV user manual*

The OPC UA Publisher is using the multicast MAC address 01-00-5E-00-00-02 with the VLAN tag 3 and PCP (Priority Code Point) 7. Therefore, configure the VLAN as:

```
>> ssh root@192.168.1.20 (default IP address Development board)
>> ip link add link sw0ep name myvlan type vlan id 3 (add VLAN id 3)
>> ip link set dev myvlan up (set the VLAN up)
>> bridge vlan add vid 3 dev sw0p1 (add VLAN to internal port)
>> bridge vlan add vid 3 dev sw0p2 (add VLAN to external port 1)
>> bridge vlan add vid 3 dev sw0p3 (optional if second port is used)
```

If the Publisher (outgoing traffic) requires TSN configuration, add a Qbv schedule. For example:

```
>> tsntool st wrcl sw0p2 /home/root/scripts/qbv.conf(write the schedule, port 1)
>> tsntool st wrcl sw0p3 /home/root/scripts/qbv.conf(write the schedule, port 2)
>> tsntool st rdacl sw0p2 /dev/stdout(to read)
>> tsntool st configure 0.0 1/1000 10000 sw0p2 (configure the schedule 1ms interval)
>> tsntool st configure 0.0 1/1000 10000 sw0p3
```

The parameters can vary according to your specification. Also, `qbv.conf` is a file specifying the schedule. For example, a 1 ms schedule:

```
sgs 400000 0x80 (high priority traffic)
sgs 100000 0x7E (other traffic)
sgs 400000 0x80
sgs 100000 0x7f
```

To check the status of the precision time protocol (PTP) clock synchronization across devices that are TSN capable use `DEPTP_TOOL` provided by TTTech :

```
>> deptp_tool -get-current-dataset
```

8. Document Revision History for AN 957: Time-Sensitive Networking for Drive-on-Chip Design Example

Document Version	Changes
2021.10.30	Initial release.

A. Example .qsf for Pin Assignments and Attributes

For the TSN Drive-on-Chip Design example.

```

set_location_assignment PIN_AD27 -to global_reset_n
set_location_assignment PIN_A10 -to SD_MCLK
set_location_assignment PIN_AG7 -to INPUT_CURRENT_MDAT
set_location_assignment PIN_E8 -to INPUT_VOLTAGE_MDAT
set_location_assignment PIN_AF8 -to BOOST_DRV0_PWM_H
set_location_assignment PIN_D7 -to BOOST_DRV0_CURRENT_MDAT
set_location_assignment PIN_AG1 -to BOOST_DRV0_PWM_L
set_location_assignment PIN_D6 -to BOOST_DRV1_CURRENT_MDAT
set_location_assignment PIN_K12 -to DCBUS_CURRENT_MDAT
set_location_assignment PIN_C5 -to DCBUS_VOLTAGE_MDAT
set_location_assignment PIN_E4 -to DRV_SCLK
set_location_assignment PIN_J10 -to DRV_SOMI
set_location_assignment PIN_J9 -to DRV_SIMO
set_location_assignment PIN_E3 -to DRV0_SER_RX
set_location_assignment PIN_K7 -to DRV0_CSn
set_location_assignment PIN_E2 -to DRV0_SER_TX
set_location_assignment PIN_K8 -to DRV0_SER_CLK
set_location_assignment PIN_E1 -to DRV0_EN_GATE_P
set_location_assignment PIN_G12 -to DRV0_EN_GATE_N
set_location_assignment PIN_D1 -to HSMC_DRV0_PWM_UH
set_location_assignment PIN_G11 -to HSMC_DRV0_PWM_UL
set_location_assignment PIN_D2 -to HSMC_DRV0_PWM_VH
set_location_assignment PIN_J7 -to HSMC_DRV0_PWM_VL
set_location_assignment PIN_C2 -to HSMC_DRV0_PWM_WH
set_location_assignment PIN_H7 -to HSMC_DRV0_PWM_WL
set_location_assignment PIN_B2 -to DRV0_U_VOLTS_MDAT
set_location_assignment PIN_H8 -to DRV0_U_CURRENT_MDAT
set_location_assignment PIN_B1 -to DRV0_V_VOLTS_MDAT
set_location_assignment PIN_G8 -to DRV0_V_CURRENT_MDAT
set_location_assignment PIN_C3 -to DRV0_W_VOLTS_MDAT
set_location_assignment PIN_G10 -to DRV0_W_CURRENT_MDAT
set_location_assignment PIN_B3 -to DRV0_QR_A
set_location_assignment PIN_F10 -to DRV0_QHR_U
set_location_assignment PIN_AJ2 -to DRV0_QR_B
set_location_assignment PIN_AG2 -to DRV0_QHR_V
set_location_assignment PIN_AC12 -to DRV0_QR_Z
set_location_assignment PIN_AH3 -to DRV0_QHR_W
set_location_assignment PIN_F9 -to BOOST_STATUS
set_location_assignment PIN_F8 -to REGEN_STATUS
set_location_assignment PIN_D5 -to DRV1_SER_RX
set_location_assignment PIN_F11 -to DRV1_CSn
set_location_assignment PIN_C4 -to DRV1_SER_TX
set_location_assignment PIN_E11 -to DRV1_SER_CLK
set_location_assignment PIN_A6 -to DRV1_EN_GATE_P
set_location_assignment PIN_B6 -to DRV1_EN_GATE_N
set_location_assignment PIN_A5 -to HSMC_DRV1_PWM_UH
set_location_assignment PIN_B5 -to HSMC_DRV1_PWM_UL
set_location_assignment PIN_C7 -to HSMC_DRV1_PWM_VH
set_location_assignment PIN_E9 -to HSMC_DRV1_PWM_VL
set_location_assignment PIN_B7 -to HSMC_DRV1_PWM_WH
set_location_assignment PIN_D9 -to HSMC_DRV1_PWM_WL
set_location_assignment PIN_A9 -to DRV1_U_VOLTS_MDAT
set_location_assignment PIN_D11 -to DRV1_U_CURRENT_MDAT
set_location_assignment PIN_A8 -to DRV1_V_VOLTS_MDAT
set_location_assignment PIN_D10 -to DRV1_V_CURRENT_MDAT
set_location_assignment PIN_C8 -to DRV1_W_VOLTS_MDAT

```

```
set_location_assignment PIN_E12 -to DRV1_W_CURRENT_MDAT
set_location_assignment PIN_B8 -to DRV1_QR_A
set_location_assignment PIN_D12 -to DRV1_QHR_U
set_location_assignment PIN_C10 -to DRV1_QR_B
set_location_assignment PIN_F13 -to DRV1_QHR_V
set_location_assignment PIN_C9 -to DRV1_QR_Z
set_location_assignment PIN_E13 -to DRV1_QHR_W
set_location_assignment PIN_K14 -to DRV1_SER_TX_EN
set_location_assignment PIN_C13 -to DRV0_SER_TX_EN
set_location_assignment PIN_J12 -to VOLTAGE_FAULT
set_location_assignment PIN_D4 -to DRV0_FAULTn
set_location_assignment PIN_A4 -to CURRENT_FAULT
set_location_assignment PIN_A3 -to DRV1_FAULTn
set_location_assignment PIN_B13 -to GPIO_0
set_location_assignment PIN_A13 -to DRV1_RESOLVER_SSCS
set_location_assignment PIN_B12 -to DRV0_RESOLVER_SSCS
set_location_assignment PIN_C12 -to DRV1_RESOLVER_SCSB
set_location_assignment PIN_F15 -to DRV0_RESOLVER_SCSB
set_location_assignment PIN_B11 -to DRV1_RESOLVER_INHB
set_location_assignment PIN_F14 -to DRV0_RESOLVER_INHB
set_location_assignment PIN_E7 -to DRV1_RESOLVER_ERRHLD
set_location_assignment PIN_H15 -to DRV0_RESOLVER_ERRHLD
set_location_assignment PIN_E6 -to DRV1_RESOLVER_ERRSTB
set_location_assignment PIN_G15 -to DRV0_RESOLVER_ERRSTB
set_location_assignment PIN_AD12 -to HSMC_PRSNTn
set_location_assignment PIN_AG12 -to mem_a[14]
set_location_assignment PIN_AK8 -to mem_a[13]
set_location_assignment PIN_AK7 -to mem_a[12]
set_location_assignment PIN_AK9 -to mem_a[11]
set_location_assignment PIN_AJ9 -to mem_a[10]
set_location_assignment PIN_AH14 -to mem_a[9]
set_location_assignment PIN_AA13 -to mem_a[8]
set_location_assignment PIN_AK13 -to mem_a[7]
set_location_assignment PIN_AK12 -to mem_a[6]
set_location_assignment PIN_AH15 -to mem_a[5]
set_location_assignment PIN_AG15 -to mem_a[4]
set_location_assignment PIN_AJ12 -to mem_a[3]
set_location_assignment PIN_AH12 -to mem_a[2]
set_location_assignment PIN_AK14 -to mem_a[1]
set_location_assignment PIN_AJ14 -to mem_a[0]
set_location_assignment PIN_AK11 -to mem_ba[2]
set_location_assignment PIN_AJ11 -to mem_ba[1]
set_location_assignment PIN_AH10 -to mem_ba[0]
set_location_assignment PIN_AA14 -to mem_ck
set_location_assignment PIN_AA15 -to mem_ck_n
set_location_assignment PIN_AJ21 -to mem_cke
set_location_assignment PIN_AG23 -to mem_dm[1]
set_location_assignment PIN_AH17 -to mem_dm[0]
set_location_assignment PIN_AJ27 -to mem_dm[3]
set_location_assignment PIN_AK23 -to mem_dm[2]
set_location_assignment PIN_AF18 -to mem_dq[0]
set_location_assignment PIN_AE17 -to mem_dq[1]
set_location_assignment PIN_AG16 -to mem_dq[2]
set_location_assignment PIN_AF16 -to mem_dq[3]
set_location_assignment PIN_AH20 -to mem_dq[4]
set_location_assignment PIN_AG21 -to mem_dq[5]
set_location_assignment PIN_AJ16 -to mem_dq[6]
set_location_assignment PIN_AH18 -to mem_dq[7]
set_location_assignment PIN_AK18 -to mem_dq[8]
set_location_assignment PIN_AJ17 -to mem_dq[9]
set_location_assignment PIN_AG18 -to mem_dq[10]
set_location_assignment PIN_AK19 -to mem_dq[11]
set_location_assignment PIN_AG20 -to mem_dq[12]
set_location_assignment PIN_AF19 -to mem_dq[13]
set_location_assignment PIN_AJ20 -to mem_dq[14]
set_location_assignment PIN_AH24 -to mem_dq[15]
set_location_assignment PIN_AE19 -to mem_dq[16]
set_location_assignment PIN_AE18 -to mem_dq[17]
set_location_assignment PIN_AG22 -to mem_dq[18]
set_location_assignment PIN_AK22 -to mem_dq[19]
set_location_assignment PIN_AF21 -to mem_dq[20]
```

```

set_location_assignment PIN_AF20 -to mem_dq[21]
set_location_assignment PIN_AH23 -to mem_dq[22]
set_location_assignment PIN_AK24 -to mem_dq[23]
set_location_assignment PIN_AF24 -to mem_dq[24]
set_location_assignment PIN_AF23 -to mem_dq[25]
set_location_assignment PIN_AJ24 -to mem_dq[26]
set_location_assignment PIN_AK26 -to mem_dq[27]
set_location_assignment PIN_AE23 -to mem_dq[28]
set_location_assignment PIN_AE22 -to mem_dq[29]
set_location_assignment PIN_AG25 -to mem_dq[30]
set_location_assignment PIN_AK27 -to mem_dq[31]
set_location_assignment PIN_V17 -to mem_dqs[1]
set_location_assignment PIN_V16 -to mem_dqs[0]
set_location_assignment PIN_AK21 -to mem_reset_n

set_location_assignment PIN_AC18 -to clk_50
set_location_assignment PIN_AF14 -to clk_ddr3_100_p
set_global_assignment -name TIMEQUEST_MULTICORNER_ANALYSIS ON
set_global_assignment -name ENABLE_OCT_DONE ON

set_location_assignment PIN_H14 -to BOOST_DRV1_PWM_H
set_location_assignment PIN_AK2 -to LED[0]
set_location_assignment PIN_Y16 -to LED[1]
set_location_assignment PIN_W15 -to LED[2]
set_location_assignment PIN_AB17 -to LED[3]
set_location_assignment PIN_AA13 -to button[0]
set_location_assignment PIN_AB13 -to button[1]
set_location_assignment PIN_G13 -to BOOST_DRV1_PWM_L
set_location_assignment PIN_AF9 -to REGEN_EN_N

set_location_assignment PIN_AC20 -to mem_dqs[3]
set_location_assignment PIN_Y17 -to mem_dqs[2]
set_location_assignment PIN_AD19 -to mem_dqs_n[3]
set_location_assignment PIN_AA18 -to mem_dqs_n[2]
set_location_assignment PIN_W17 -to mem_dqs_n[1]
set_location_assignment PIN_W16 -to mem_dqs_n[0]
set_location_assignment PIN_AG17 -to mem_oct_rzqin

set_location_assignment PIN_AB15 -to mem_cs_n
set_location_assignment PIN_AH8 -to mem_ras_n
set_location_assignment PIN_AH7 -to mem_cas_n
set_location_assignment PIN_AJ6 -to mem_we_n
set_location_assignment PIN_AE16 -to mem_odt
set_instance_assignment -name IO_STANDARD "1.5 V" -to clk_50
set_instance_assignment -name IO_STANDARD "1.5 V" -to clk_ddr3_100_p
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dq[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dq[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dq[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dq[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dq[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dq[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[3] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dq[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dq[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[4] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0

```



```

set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[28] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dq[28] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dq[28] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[29] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dq[29] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dq[29] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[30] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dq[30] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dq[30] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dq[31] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dq[31] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dq[31] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_dqs[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dqs[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dqs[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_dqs[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dqs[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dqs[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_dqs[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dqs[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dqs[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_dqs[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dqs[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dqs[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 4 -to mem_dqs[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 4 -to mem_dqs[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 4 -to mem_dqs[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 4 -to mem_dqs[3] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_dqs_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dqs_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dqs_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 4 -to mem_dqs_n[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_dqs_n[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dqs_n[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dqs_n[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 4 -to mem_dqs_n[1] -tag

```

```

__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_dqs_n[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dqs_n[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dqs_n[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 4 -to mem_dqs_n[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_dqs_n[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to mem_dqs_n[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dqs_n[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 4 -to mem_dqs_n[3] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_cke[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_cke[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_ck[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITHOUT
CALIBRATION" -to mem_ck[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 2 -to mem_ck[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
mem_ck_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITHOUT
CALIBRATION" -to mem_ck_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name D5_DELAY 2 -to mem_ck_n[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_ba[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_ba[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_ba[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_cs_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_we_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_ras_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_cas_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_odt[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "1.5 V" -to mem_reset_n -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dm[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dm[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dm[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to mem_dm[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dm[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dm[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dm[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_dm[3] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_dq[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_dq[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0

```



```

tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_dqs_n[2] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_dqs_n[3] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_dqs[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_dqs[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_dqs[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_dqs[3] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[10] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[11] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[12] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[3] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[4] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[5] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[6] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[7] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[8] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_a[9] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_ba[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_ba[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_ba[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_cs_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_we_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_ras_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_cas_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_cke[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_odt[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_ck[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_ck_n[0] -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to mem_reset_n -
tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name GLOBAL_SIGNAL OFF -to u_doc|mem_if_ddr3_emif_0|p0|
umemory|ureset|phy_reset_n -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name ENABLE_BENEFICIAL_SKEW_OPTIMIZATION_FOR_NON_GLOBAL_CLOCKS ON -to u_doc|
mem_if_ddr3_emif_0 -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "1.5 V" -to LED[0]
set_instance_assignment -name IO_STANDARD "1.5 V" -to LED[1]
set_instance_assignment -name IO_STANDARD "1.5 V" -to LED[2]

```

```

set_instance_assignment -name IO_STANDARD "1.5 V" -to LED[3]
set_instance_assignment -name IO_STANDARD "1.5 V" -to button[0]
set_instance_assignment -name IO_STANDARD "1.5 V" -to button[1]
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_reset_n -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[0] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[10] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[10] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[11] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[11] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[12] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[12] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[13] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[13] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[14] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[14] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[1] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[2] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[3] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[3] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[4] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[4] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[5] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[5] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[6] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[6] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[7] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[7] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[8] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[8] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to mem_a[9] -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_a[9] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name IO_STANDARD "SSTL-15" -to mem_oct_rzqin -tag
__DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_ba[2] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_ba[1] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0

```

```

set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_ba[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_cs_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_ras_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_cas_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_we_n[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
mem_odt[0] -tag __DOC_TANDEM_MAX10_QSYS_mem_if_ddr3_emif_0_p0

set_location_assignment PIN_H19 -to enet_hps_gtx_clk
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_gtx_clk
set_location_assignment PIN_B21 -to enet_hps_mdc
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_mdc
set_location_assignment PIN_E21 -to enet_hps_mdio
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_mdio
set_location_assignment PIN_G20 -to enet_hps_rx_clk
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_rx_clk
set_location_assignment PIN_K17 -to enet_hps_rx_dv
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_rx_dv
set_location_assignment PIN_A21 -to enet_hps_rxd[0]
set_location_assignment PIN_B20 -to enet_hps_rxd[1]
set_location_assignment PIN_B18 -to enet_hps_rxd[2]
set_location_assignment PIN_D21 -to enet_hps_rxd[3]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_rxd
set_location_assignment PIN_A20 -to enet_hps_tx_en
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_tx_en
set_location_assignment PIN_F20 -to enet_hps_txd[0]
set_location_assignment PIN_J19 -to enet_hps_txd[1]
set_location_assignment PIN_F21 -to enet_hps_txd[2]
set_location_assignment PIN_F19 -to enet_hps_txd[3]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to enet_hps_txd
set_location_assignment PIN_A16 -to sd_clk
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to sd_clk
set_location_assignment PIN_F18 -to sd_cmd
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to sd_cmd
set_location_assignment PIN_G18 -to sd_dat[0]
set_location_assignment PIN_C17 -to sd_dat[1]
set_location_assignment PIN_D17 -to sd_dat[2]
set_location_assignment PIN_B16 -to sd_dat[3]
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to sd_dat
set_location_assignment PIN_E24 -to uart_rx
set_location_assignment PIN_D24 -to uart_tx
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to uart_rx
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to uart_tx
set_location_assignment PIN_D22 -to i2c_scl_hps
set_location_assignment PIN_C23 -to i2c_sda_hps
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to i2c_scl_hps
set_instance_assignment -name IO_STANDARD "3.3-V LVCMOS" -to i2c_sda_hps

set_location_assignment PIN_F26 -to ddr3_hps_a[0]
set_location_assignment PIN_G30 -to ddr3_hps_a[1]
set_location_assignment PIN_F28 -to ddr3_hps_a[2]
set_location_assignment PIN_F30 -to ddr3_hps_a[3]
set_location_assignment PIN_J25 -to ddr3_hps_a[4]
set_location_assignment PIN_J27 -to ddr3_hps_a[5]
set_location_assignment PIN_F29 -to ddr3_hps_a[6]
set_location_assignment PIN_E28 -to ddr3_hps_a[7]
set_location_assignment PIN_H27 -to ddr3_hps_a[8]
set_location_assignment PIN_G26 -to ddr3_hps_a[9]
set_location_assignment PIN_D29 -to ddr3_hps_a[10]
set_location_assignment PIN_C30 -to ddr3_hps_a[11]
set_location_assignment PIN_B30 -to ddr3_hps_a[12]
set_location_assignment PIN_C29 -to ddr3_hps_a[13]
set_location_assignment PIN_H25 -to ddr3_hps_a[14]
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[0] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[1] -

```

```
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[2] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[3] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[4] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[5] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[6] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[7] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[8] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[9] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[10] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[11] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[12] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[13] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_a[14] -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[0] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[0] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[1] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[1] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[2] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[2] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[3] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[3] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[4] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[4] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[5] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[5] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[6] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[6] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[7] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[7] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[8] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[8] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[9] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[9] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[10] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[10] -tag __hps_sdram_p0
```

```

set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[11] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[11] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[12] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[12] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[13] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[13] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_a[14] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_a[14] -tag __hps_sdram_p0
set_location_assignment PIN_E29 -to ddr3_hps_ba[0]
set_location_assignment PIN_J24 -to ddr3_hps_ba[1]
set_location_assignment PIN_J23 -to ddr3_hps_ba[2]
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ba[0] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ba[1] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ba[2] -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_ba[0] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_ba[0] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_ba[1] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_ba[1] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_ba[2] -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_ba[2] -tag __hps_sdram_p0

set_location_assignment PIN_M23 -to ddr3_hps_ck
set_instance_assignment -name D5_DELAY 2 -to ddr3_hps_ck -tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ck -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_ck -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITHOUT
CALIBRATION" -to ddr3_hps_ck -tag __hps_sdram_p0

set_location_assignment PIN_L23 -to ddr3_hps_ck_n
set_instance_assignment -name D5_DELAY 2 -to ddr3_hps_ck_n -tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ck_n -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_ck_n -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITHOUT
CALIBRATION" -to ddr3_hps_ck_n -tag __hps_sdram_p0

set_location_assignment PIN_L29 -to ddr3_hps_cke
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_cke -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_cke -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_cke -tag __hps_sdram_p0

set_location_assignment PIN_H24 -to ddr3_hps_cs_n
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_cs_n -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_cs_n -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_cs_n -tag __hps_sdram_p0

```

```
set_location_assignment PIN_D30 -to ddr3_hps_ras_n
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_ras_n -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_ras_n -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_ras_n -tag __hps_sdram_p0

set_location_assignment PIN_E27 -to ddr3_hps_cas_n
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_cas_n -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_cas_n -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_cas_n -tag __hps_sdram_p0

set_location_assignment PIN_C28 -to ddr3_hps_we_n
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_we_n -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_we_n -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_we_n -tag __hps_sdram_p0

set_location_assignment PIN_P30 -to ddr3_hps_reset_n
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to
ddr3_hps_reset_n -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to
ddr3_hps_reset_n -tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_reset_n -tag __hps_sdram_p0

set_location_assignment PIN_K23 -to ddr3_hps_dq[0]
set_location_assignment PIN_K22 -to ddr3_hps_dq[1]
set_location_assignment PIN_H30 -to ddr3_hps_dq[2]
set_location_assignment PIN_G28 -to ddr3_hps_dq[3]
set_location_assignment PIN_L25 -to ddr3_hps_dq[4]
set_location_assignment PIN_L24 -to ddr3_hps_dq[5]
set_location_assignment PIN_J30 -to ddr3_hps_dq[6]
set_location_assignment PIN_J29 -to ddr3_hps_dq[7]
set_location_assignment PIN_K26 -to ddr3_hps_dq[8]
set_location_assignment PIN_L26 -to ddr3_hps_dq[9]
set_location_assignment PIN_K29 -to ddr3_hps_dq[10]
set_location_assignment PIN_K27 -to ddr3_hps_dq[11]
set_location_assignment PIN_M26 -to ddr3_hps_dq[12]
set_location_assignment PIN_M27 -to ddr3_hps_dq[13]
set_location_assignment PIN_L28 -to ddr3_hps_dq[14]
set_location_assignment PIN_M30 -to ddr3_hps_dq[15]
set_location_assignment PIN_U26 -to ddr3_hps_dq[16]
set_location_assignment PIN_T26 -to ddr3_hps_dq[17]
set_location_assignment PIN_N29 -to ddr3_hps_dq[18]
set_location_assignment PIN_N28 -to ddr3_hps_dq[19]
set_location_assignment PIN_P26 -to ddr3_hps_dq[20]
set_location_assignment PIN_P27 -to ddr3_hps_dq[21]
set_location_assignment PIN_N27 -to ddr3_hps_dq[22]
set_location_assignment PIN_R29 -to ddr3_hps_dq[23]
set_location_assignment PIN_P24 -to ddr3_hps_dq[24]
set_location_assignment PIN_P25 -to ddr3_hps_dq[25]
set_location_assignment PIN_T29 -to ddr3_hps_dq[26]
set_location_assignment PIN_T28 -to ddr3_hps_dq[27]
set_location_assignment PIN_R27 -to ddr3_hps_dq[28]
set_location_assignment PIN_R26 -to ddr3_hps_dq[29]
set_location_assignment PIN_V30 -to ddr3_hps_dq[30]
set_location_assignment PIN_W29 -to ddr3_hps_dq[31]
set_location_assignment PIN_W26 -to ddr3_hps_dq[32]
set_location_assignment PIN_R24 -to ddr3_hps_dq[33]
set_location_assignment PIN_U27 -to ddr3_hps_dq[34]
set_location_assignment PIN_V28 -to ddr3_hps_dq[35]
set_location_assignment PIN_T25 -to ddr3_hps_dq[36]
set_location_assignment PIN_U25 -to ddr3_hps_dq[37]
```

```

set_location_assignment PIN_V27 -to ddr3_hps_dq[38]
set_location_assignment PIN_Y29 -to ddr3_hps_dq[39]
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[0] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[1] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[2] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[3] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[4] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[5] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[6] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[7] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[8] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[9] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[10] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[11] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[12] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[13] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[14] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[15] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[16] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[17] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[18] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[19] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[20] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[21] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[22] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[23] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[24] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[25] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[26] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[27] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[28] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[29] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[30] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[31] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[32] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[33] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[34]

```

```
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[35]
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[36]
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[37]
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[38]
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dq[39]
-tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[0] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[0] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[0] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[1] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[1] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[1] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[2] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[2] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[2] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[3] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[3] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[3] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[4] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[4] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[4] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[5] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[5] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[5] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[6] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[6] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[6] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[7] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[7] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[7] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[8] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[8] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[8] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[9] -
tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[9] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[9] -tag __hps_sdram_p0
```



```

set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[33] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[34]
-tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[34] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[34] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[35]
-tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[35] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[35] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[36]
-tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[36] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[36] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[37]
-tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[37] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[37] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[38]
-tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[38] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[38] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dq[39]
-tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[39] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dq[39] -tag __hps_sdram_p0

set_location_assignment PIN_N18 -to ddr3_hps_dqs[0]
set_location_assignment PIN_N25 -to ddr3_hps_dqs[1]
set_location_assignment PIN_R19 -to ddr3_hps_dqs[2]
set_location_assignment PIN_R22 -to ddr3_hps_dqs[3]
set_location_assignment PIN_T24 -to ddr3_hps_dqs[4]
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs[0]
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs[1]
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs[2]
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs[3]
-tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dqs[4]
-tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs[0] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[0] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[0] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs[1] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[1] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[1] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs[2] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[2] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[2] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs[3] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[3] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[3] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs[4] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[4] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[4] -tag __hps_sdram_p0

```

```

CALIBRATION" -to ddr3_hps_dqs[2] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs[3] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[3] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[3] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs[4] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[4] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs[4] -tag __hps_sdram_p0

set_location_assignment PIN_M19 -to ddr3_hps_dqs_n[0]
set_location_assignment PIN_N24 -to ddr3_hps_dqs_n[1]
set_location_assignment PIN_R18 -to ddr3_hps_dqs_n[2]
set_location_assignment PIN_R21 -to ddr3_hps_dqs_n[3]
set_location_assignment PIN_T23 -to ddr3_hps_dqs_n[4]
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to
ddr3_hps_dqs_n[0] -tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to
ddr3_hps_dqs_n[1] -tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to
ddr3_hps_dqs_n[2] -tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to
ddr3_hps_dqs_n[3] -tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to
ddr3_hps_dqs_n[4] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs_n[0] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs_n[0] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs_n[0] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs_n[1] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs_n[1] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs_n[1] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs_n[2] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs_n[2] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs_n[2] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs_n[3] -tag __hps_sdram_p0
set_instance_assignment -name INPUT_TERMINATION "PARALLEL 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs_n[3] -tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dqs_n[3] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL 1.5-V SSTL CLASS I" -to
ddr3_hps_dqs_n[4] -tag __hps_sdram_p0

set_location_assignment PIN_H28 -to ddr3_hps_odt
set_instance_assignment -name PACKAGE_SKEW_COMPENSATION OFF -to ddr3_hps_odt -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_odt -
tag __hps_sdram_p0
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to
ddr3_hps_odt -tag __hps_sdram_p0

set_location_assignment PIN_K28 -to ddr3_hps_dm[0]
set_location_assignment PIN_M28 -to ddr3_hps_dm[1]
set_location_assignment PIN_R28 -to ddr3_hps_dm[2]

```

```

set_location_assignment PIN_W30 -to ddr3_hps_dm[3]
set_location_assignment PIN_W27 -to ddr3_hps_dm[4]
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[0] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[1] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[2] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[3] -
tag __hps_sdram_p0
set_instance_assignment -name PACKAGE_SKew_COMPENSATION OFF -to ddr3_hps_dm[4] -
tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[0] -
tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dm[0] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[1] -
tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dm[1] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[2] -
tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dm[2] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[3] -
tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dm[3] -tag __hps_sdram_p0
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to ddr3_hps_dm[4] -
tag __hps_sdram_p0
set_instance_assignment -name OUTPUT_TERMINATION "SERIES 50 OHM WITH
CALIBRATION" -to ddr3_hps_dm[4] -tag __hps_sdram_p0

set_location_assignment PIN_D27 -to ddr3_hps_oct_rzqin
set_instance_assignment -name IO_STANDARD "SSTL-15 CLASS I" -to
ddr3_hps_oct_rzqin -tag __hps_sdram_p0

set_location_assignment PIN_H12 -to eth_mdio_tristate_0_mdio_mdc
set_instance_assignment -name IO_STANDARD "2.5 V" -to
eth_mdio_tristate_0_mdio_mdc
set_location_assignment PIN_H13 -to eth_mdio_tristate_0_mdio_mdc
set_instance_assignment -name IO_STANDARD "2.5 V" -to
eth_mdio_tristate_0_mdio_mdc

set_location_assignment PIN_AJ1 -to enet_dual_resetn
set_instance_assignment -name SLEW_RATE 0 -to enet_dual_resetn*

set_location_assignment PIN_Y24 -to deip_0_gmii_to_mii_0_mii_rx_clk
set_location_assignment PIN_Y23 -to deip_0_gmii_to_mii_0_mii_rx_dv
set_location_assignment PIN_AB23 -to deip_0_gmii_to_mii_0_mii_rxd[0]
set_location_assignment PIN_AA24 -to deip_0_gmii_to_mii_0_mii_rxd[1]
set_location_assignment PIN_AB25 -to deip_0_gmii_to_mii_0_mii_rxd[2]
set_location_assignment PIN_AE27 -to deip_0_gmii_to_mii_0_mii_rxd[3]
set_location_assignment PIN_AE28 -to deip_0_gmii_to_mii_0_mii_rx_er
set_location_assignment PIN_W25 -to deip_0_gmii_to_mii_0_mii_tx_clk
set_location_assignment PIN_AB22 -to deip_0_gmii_to_mii_0_mii_tx_en
set_location_assignment PIN_W20 -to deip_0_gmii_to_mii_0_mii_txd[0]
set_location_assignment PIN_Y21 -to deip_0_gmii_to_mii_0_mii_txd[1]
set_location_assignment PIN_AA25 -to deip_0_gmii_to_mii_0_mii_txd[2]
set_location_assignment PIN_AB26 -to deip_0_gmii_to_mii_0_mii_txd[3]

set_location_assignment PIN_AH30 -to deip_0_gmii_to_mii_1_mii_rx_clk
set_location_assignment PIN_AC28 -to deip_0_gmii_to_mii_1_mii_rx_dv
set_location_assignment PIN_AF29 -to deip_0_gmii_to_mii_1_mii_rxd[0]
set_location_assignment PIN_AF30 -to deip_0_gmii_to_mii_1_mii_rxd[1]
set_location_assignment PIN_AD26 -to deip_0_gmii_to_mii_1_mii_rxd[2]
set_location_assignment PIN_AC27 -to deip_0_gmii_to_mii_1_mii_rxd[3]
set_location_assignment PIN_V25 -to deip_0_gmii_to_mii_1_mii_rx_er
set_location_assignment PIN_AG30 -to deip_0_gmii_to_mii_1_mii_tx_clk
set_location_assignment PIN_W24 -to deip_0_gmii_to_mii_1_mii_tx_en
set_location_assignment PIN_AG27 -to deip_0_gmii_to_mii_1_mii_txd[0]

```

```
set_location_assignment PIN_AG28 -to deip_0_gmii_to_mii_1_mii_txd[1]
set_location_assignment PIN_AF28 -to deip_0_gmii_to_mii_1_mii_txd[2]
set_location_assignment PIN_V23 -to deip_0_gmii_to_mii_1_mii_txd[3]
```

B. Top-level Verilog HDL File Example

Shows an example of the top-level Verilog HDL file for the TSN Drive-on-Chip Design project. This file is the drive-on-chip top-level Verilog HDL file including TSN capabilities of the TTTech TSN IP..

```
module DOC_TANDEM_CVSX_NIOS_TSN (
    input wire          clk_50,
    input wire          clk_ddr3_100_p,
    input wire          global_reset_n,
    input wire [1:0]    button,
    output wire [3:0]   LED,
    output wire         SD_MCLK,           // Sigma-delta ADC clock
    output wire         BOOST_DRV0_PWM_H, // DC-DC Converter
    output wire         BOOST_DRV0_PWM_L,
    output wire         BOOST_DRV1_PWM_H,
    output wire         BOOST_DRV1_PWM_L,
    input wire          REGEN_EN_N,
    input wire          BOOST_DRV0_CURRENT_MDAT,
    input wire          BOOST_DRV1_CURRENT_MDAT,
    input wire          DCBUS_CURRENT_MDAT,
    input wire          DCBUS_VOLTAGE_MDAT,
    input wire          INPUT_CURRENT_MDAT,
    input wire          INPUT_VOLTAGE_MDAT,
    output wire         BOOST_STATUS,      // DC-DC Status LEDs
    output wire         REGEN_STATUS,
    input wire          DRV0_FAULTn,       // Axis 0 inverter
    output wire         DRV0_EN_GATE_N,
    output wire         DRV0_EN_GATE_P,
    output wire         HSMC_DRV0_PWM_UH,
    output wire         HSMC_DRV0_PWM_UL,
    output wire         HSMC_DRV0_PWM_VH,
    output wire         HSMC_DRV0_PWM_VL,
    output wire         HSMC_DRV0_PWM_WH,
    output wire         HSMC_DRV0_PWM_WL,
    input wire          DRV0_QHR_U,        // Axis 0 Quad/Hall/resolver
    input wire          DRV0_QHR_V,
    input wire          DRV0_QHR_W,
    input wire          DRV0_QR_A,
    input wire          DRV0_QR_B,
    input wire          DRV0_QR_Z,
    output wire         DRV0_SER_CLK,      // Axis 0 Serial
    output wire         DRV0_SER_RX,
    output wire         DRV0_SER_TX,
    output wire         DRV0_SER_TX_EN,
    output wire         DRV0_RESOLVER_SSCS, // Axis 0 Resolver
    output wire         DRV0_RESOLVER_SCSB,
    output wire         DRV0_RESOLVER_INHB,
    input wire          DRV0_RESOLVER_ERRHLD,
    output wire         DRV0_RESOLVER_ERRSTB,
    input wire          DRV0_U_CURRENT_MDAT, // Axis 0 sigma-delta ADCs
    input wire          DRV0_U_VOLTS_MDAT,
    input wire          DRV0_V_CURRENT_MDAT,
    input wire          DRV0_V_VOLTS_MDAT,
    input wire          DRV0_W_CURRENT_MDAT,
    input wire          DRV0_W_VOLTS_MDAT,
    input wire          DRV1_FAULTn,       // Axis 1 inverter
    output wire         DRV1_EN_GATE_N,
    output wire         DRV1_EN_GATE_P,
```

```

output wire      HSMC_DRV1_PWM_UH,
output wire      HSMC_DRV1_PWM_UL,
output wire      HSMC_DRV1_PWM_VH,
output wire      HSMC_DRV1_PWM_VL,
output wire      HSMC_DRV1_PWM_WH,
output wire      HSMC_DRV1_PWM_WL,
input  wire      DRV1_QHR_U,           // Axis 1 Quad/Hall/resolver
input  wire      DRV1_QHR_V,
input  wire      DRV1_QHR_W,
input  wire      DRV1_QR_A,
input  wire      DRV1_QR_B,
input  wire      DRV1_QR_Z,
output wire     DRV1_SER_CLK,        // Axis 1 Serial
input  wire     DRV1_SER_RX,
output wire     DRV1_SER_TX,
output wire     DRV1_SER_TX_EN,
output wire     DRV1_RESOLVER_SSCS,   // Axis 1 resolver
output wire     DRV1_RESOLVER_SCSB,
output wire     DRV1_RESOLVER_INHB,
input  wire     DRV1_RESOLVER_ERRHLD,
output wire     DRV1_RESOLVER_ERRSTB,
input  wire     DRV1_U_CURRENT_MDAT, // Axis 1 sigma-delta ADCs
input  wire     DRV1_U_VOLTS_MDAT,
input  wire     DRV1_V_CURRENT_MDAT,
input  wire     DRV1_V_VOLTS_MDAT,
input  wire     DRV1_W_CURRENT_MDAT,
input  wire     DRV1_W_VOLTS_MDAT,
output wire    DRV_SCLK,           // SPI control bus for inverter
gate drivers
output wire    DRV_SIMO,
input  wire    DRV_SOMI,
output wire    DRV0_CSn,
output wire    DRV1_CSn,
output wire    VOLTAGE_FAULT,       // Inverter Status LEDs
output wire    CURRENT_FAULT,
input  wire    HSMC_PRSNTn,
output wire    GPIO_0,
output wire [14:0] mem_a,          //FPGA DDR3 memory
output wire [2:0] mem_ba,
output wire [0:0] mem_ck,
output wire [0:0] mem_ck_n,
output wire [0:0] mem_cke,
output wire [0:0] mem_cs_n,
output wire [3:0] mem_dm,
output wire [0:0] mem_ras_n,
output wire [0:0] mem_cas_n,
output wire [0:0] mem_we_n,
output wire    mem_reset_n,
inout wire [31:0] mem_dq,
inout wire [3:0] mem_dqs,
inout wire [3:0] mem_dqs_n,
output wire [0:0] mem_odt,
input  wire    mem_oct_rzqin,
output wire    enet_hps_gtx_clk,    //HPS ethernet
output wire    enet_hps_mdc,
inout wire    enet_hps_mdio,
input  wire    enet_hps_rx_clk,
input  wire    enet_hps_rx_dv,
input  wire [3:0] enet_hps_rxd,
output wire    enet_hps_tx_en,
output wire [3:0] enet_hps_txd,
output wire    sd_clk,            //HPS-SD-Card-Flash
inout wire    sd_cmd,
inout [3:0]    sd_dat,
input  wire    uart_rx,          //HPS-UART
output wire    uart_tx,
inout wire    i2c_scl_hps,       //HPS-I2C
inout wire    i2c_sda_hps,
output wire [14:0] ddr3_hps_a,   // CV SOC dev kit HPS DDR3 pins
output wire [2:0] ddr3_hps_ba,

```

```

        output wire      ddr3_hps_ck,
        output wire      ddr3_hps_ck_n,
        output wire      ddr3_hps_cke,
        output wire      ddr3_hps_cs_n,
        output wire      ddr3_hps_ras_n,
        output wire      ddr3_hps_cas_n,
        output wire      ddr3_hps_we_n,
        output wire      ddr3_hps_reset_n,
        inout  wire [39:0] ddr3_hps_dq,
        inout  wire [4:0]  ddr3_hps_dqs,
        inout  wire [4:0]  ddr3_hps_dqs_n,
        output wire      ddr3_hps_odt,
        output wire [4:0] ddr3_hps_dm,
        input  wire       ddr3_hps_oct_rzqin,
                                         // TSN Switch
        output wire      eth_mdio_tristate_0_mdio_mdc,
        inout  wire       eth_mdio_tristate_0_mdio_mdio,
        output wire      enet_dual_resetn,
        input  wire       deip_0_gmii_to_mii_0_mii_rx_clk,
        input  wire       deip_0_gmii_to_mii_0_mii_rx_dv,
        input  wire [3:0] deip_0_gmii_to_mii_0_mii_rxd,
        input  wire       deip_0_gmii_to_mii_0_mii_rx_er,
        input  wire       deip_0_gmii_to_mii_0_mii_tx_clk,
        output reg       deip_0_gmii_to_mii_0_mii_tx_en,
        output reg [3:0] deip_0_gmii_to_mii_0_mii_txd,
        input  wire       deip_0_gmii_to_mii_1_mii_rx_clk,
        input  wire       deip_0_gmii_to_mii_1_mii_rx_dv,
        input  wire [3:0] deip_0_gmii_to_mii_1_mii_rxd,
        input  wire       deip_0_gmii_to_mii_1_mii_rx_er,
        input  wire       deip_0_gmii_to_mii_1_mii_tx_clk,
        output reg       deip_0_gmii_to_mii_1_mii_tx_en,
        output reg [3:0] deip_0_gmii_to_mii_1_mii_txd
    );
    // System
    wire reset_n = global_reset_n;
    wire pll_locked;

    // Allow software to select routing of encoder interfaces to/from pins
    // QEP has desicated pins so selection is not actually needed
    wire [7:0] encoder_select;
    wire sel0_endat = ~encoder_select[2] & ~encoder_select[1] &
    ~encoder_select[0]; //000
    wire sel0_biss = ~encoder_select[2] & ~encoder_select[1] &
    encoder_select[0]; //001
    wire sel0_rslvr = ~encoder_select[2] & encoder_select[1] &
    ~encoder_select[0]; //010
    wire sell_biss = ~encoder_select[5] & ~encoder_select[4] &
    ~encoder_select[3]; //000
    wire sell_endat = ~encoder_select[5] & ~encoder_select[4] &
    encoder_select[3]; //001
    wire sell_rslvr = ~encoder_select[5] & encoder_select[4] &
    ~encoder_select[3]; //010

    wire DRV0_SER_CLK_BISS = 1'b0;
    wire DRV0_SER_CLK_ENDAT = 1'b0;
    wire DRV0_SER_CLK_RSLVR;
    wire DRV0_SER_CLK_RSLVR_CTRL;
    wire DRV0_SER_CLK_RSLVR_POSN;
    wire DRV0_SER_TX_BISS = 1'b0;
    wire DRV0_SER_TX_ENDAT = 1'b0;
    wire DRV0_SER_TX_RSLVR;
    wire DRV0_SER_TX_EN_ENDAT = 1'b0;

    wire DRV1_SER_CLK_BISS = 1'b0;
    wire DRV1_SER_CLK_ENDAT = 1'b0;
    wire DRV1_SER_CLK_RSLVR;
    wire DRV1_SER_CLK_RSLVR_CTRL;
    wire DRV1_SER_CLK_RSLVR_POSN;
    wire DRV1_SER_TX_BISS = 1'b0;
    wire DRV1_SER_TX_ENDAT = 1'b0;

```

```

wire DRV1_SER_TX_RSLVR;
wire DRV1_SER_TX_EN_ENDAT = 1'b0;

// Resolver chip has independent slave selects for SPI input and output but
only one SCK input.
// We cannot use a single SPI master with two chip selects as the two data
registers are different
// lengths and require different clock phase. To get around this we use two SPI
masters with idle
// clock state set to high and then AND the two clocks.
assign DRV0_SER_CLK_RSLVR = DRV0_SER_CLK_RSLVR_CTRL & DRV0_SER_CLK_RSLVR_POSN;
assign DRV1_SER_CLK_RSLVR = DRV1_SER_CLK_RSLVR_CTRL & DRV1_SER_CLK_RSLVR_POSN;

assign DRV0_SER_CLK = sel0_biss ? DRV0_SER_CLK_BISS
                                : sel0_endat ? DRV0_SER_CLK_ENDAT
                                : sel0_rslvr ? DRV0_SER_CLK_RSLVR : 1'b0;
assign DRV0_SER_TX = sel0_biss ? DRV0_SER_TX_BISS
                                : sel0_endat ? DRV0_SER_TX_ENDAT
                                : sel0_rslvr ? DRV0_SER_TX_RSLVR : 1'b0;
assign DRV0_SER_TX_EN = sel0_biss ? 1'b0
                                : sel0_endat ? DRV0_SER_TX_EN_ENDAT
                                : sel0_rslvr ? 1'b1 : 1'b0;
assign DRV1_SER_CLK = sell_biss ? DRV1_SER_CLK_BISS
                                : sell_endat ? DRV1_SER_CLK_ENDAT
                                : sell_rslvr ? DRV1_SER_CLK_RSLVR : 1'b0;
assign DRV1_SER_TX = sell_biss ? DRV1_SER_TX_BISS
                                : sell_endat ? DRV1_SER_TX_ENDAT
                                : sell_rslvr ? DRV1_SER_TX_RSLVR : 1'b0;
assign DRV1_SER_TX_EN = sell_biss ? 1'b0
                                : sell_endat ? DRV1_SER_TX_EN_ENDAT
                                : sell_rslvr ? 1'b1 : 1'b0;

wire dclink_overvoltage;
wire dclink_undervoltage;
wire dclink_overcurrent;
wire dcin_undervoltage;
wire dcin_overvoltage;
wire dcin_overcurrent;
wire lvddc_oc_latch;
wire lvddc_ov_latch;
wire Overcurrent0;
wire Overcurrent1;
assign GPIO_0 = 0;

// Resolver I/O
wire [1:0] drive0_rslvr_pio_out;
wire [1:0] drive1_rslvr_pio_out;
assign DRV0_RESOLVER_ERRSTB = drive0_rslvr_pio_out[0];
assign DRV0_RESOLVER_INHB = drive0_rslvr_pio_out[1];
assign DRV1_RESOLVER_ERRSTB = drive1_rslvr_pio_out[0];
assign DRV1_RESOLVER_INHB = drive1_rslvr_pio_out[1];

wire oc_latch_led0;
wire ov_latch_led0;
wire oc_latch_led1;
wire ov_latch_led1;
wire dc_dc_fault;

wire dc_dc_on;
assign BOOST_STATUS = dc_dc_on;
assign REGEN_STATUS = ~REGEN_EN_N;
assign VOLTAGE_FAULT = lvddc_oc_latch;
assign CURRENT_FAULT = lvddc_ov_latch;

// Status LEDs on MAX10 dev kit
wire [4:0] led_pio;
assign LED[0] = oc_latch_led0 | ov_latch_led0;
assign LED[1] = oc_latch_led1 | ov_latch_led1;
assign LED[2] = dcin_overvoltage | dcin_undervoltage | dcin_overcurrent;
//assign LED[3] = dc_dc_fault;
assign LED[3] = pps;

```

```

//assign LED[4] = led_pio[0];

// Safe torque off interlock outputs
assign DRV0_EN_GATE_P = 1'b1;
assign DRV0_EN_GATE_N = 1'b0;
assign DRV1_EN_GATE_P = 1'b1;
assign DRV1_EN_GATE_N = 1'b0;

wire enc_stb0_n;
wire enc_stb1_n;

wire [1:0] gate_drive_spi_ss_n;
assign DRV0_CSn = gate_drive_spi_ss_n[0];
assign DRV1_CSn = gate_drive_spi_ss_n[1];

// MAX10 ADC threshold detection
assign dclink_overvoltage = 1'b0; // There is no output signal in Autonomous DC-DC-Converter block
assign dclink_undervoltage = 1'b0; // There is no output signal in Autonomous DC-DC-Converter block

// Retime to rising clock edge of the ethernet interfaces
wire deip_0_gmii_to_mii_0_mii_tx_en_i;
wire [3:0] deip_0_gmii_to_mii_0_mii_txd_i;
wire deip_0_gmii_to_mii_1_mii_tx_en_i;
wire [3:0] deip_0_gmii_to_mii_1_mii_txd_i;

always @(posedge deip_0_gmii_to_mii_0_mii_tx_clk)
begin
    deip_0_gmii_to_mii_0_mii_tx_en <= deip_0_gmii_to_mii_0_mii_tx_en_i;
    deip_0_gmii_to_mii_0_mii_txd <= deip_0_gmii_to_mii_0_mii_txd_i;
end
always @(posedge deip_0_gmii_to_mii_1_mii_tx_clk)
begin
    deip_0_gmii_to_mii_1_mii_tx_en <= deip_0_gmii_to_mii_1_mii_tx_en_i;
    deip_0_gmii_to_mii_1_mii_txd <= deip_0_gmii_to_mii_1_mii_txd_i;
end

DOC_TANDEM_CVSX_NIOS_TSN_QSYS u_doc (
    .clk_50_in_clk                               (clk_50),
    .clk_adc_out_clk                            (SD_MCLK),
    .clk_ddr_in_clk                            (clk_ddr3_100_p),
    .reset_n_reset_n                           (reset_n),
    .pll_reset_reset                          (1'b0),
    .io_in_buttons_export                     ({~HSMC_PRSNTn,
7'b0} | button),
    .io_out_led_export                         (led_pio),
    .lvmc_dclink_DSPBA_lvdcdc_gate_drive_gate_a_h
(BOOST_DRV0_PWM_H),
    .lvmc_dclink_DSPBA_lvdcdc_gate_drive_gate_a_l
(BOOST_DRV0_PWM_L),
    .lvmc_dclink_DSPBA_lvdcdc_gate_drive_gate_b_h
(BOOST_DRV1_PWM_H),
    .lvmc_dclink_DSPBA_lvdcdc_gate_drive_gate_b_l
(BOOST_DRV1_PWM_L),
    .lvmc_dclink_DSPBA_lvdcdc_fault_input_input_ov_fault
(dcin_overvoltage),
    .lvmc_dclink_DSPBA_lvdcdc_fault_input_input_uv_fault
(dcin_undervoltage),
    .lvmc_dclink_DSPBA_lvdcdc_fault_input_output_ov_fault
(dclink_overvoltage),
    .lvmc_dclink_DSPBA_lvdcdc_fault_input_output_uv_fault
(dclink_undervoltage),
    .lvmc_dclink_DSPBA_lvdcdc_fault_input_input_oc_fault
(dcin_overcurrent),
    .lvmc_dclink_DSPBA_lvdcdc_fault_input_output_oc_fault
(dclink_overcurrent),
    .lvmc_dclink_DSPBA_lvdcdc_bidir_en_n          (REGEN_EN_N),
    .lvmc_dclink_DSPBA_lvdcdc_fault_status_overcurrent
(lvdcdc_oc_latch),
    .lvmc_dclink_DSPBA_lvdcdc_fault_status_overvoltage

```

```

(lvdcdc_ov_latch),
    .lvmc_dclink_dspba_lvdcdc_fault_status_fault_sync      (dc_dc_fault),
    .lvmc_dclink_dspba_lvdcdc_dc_dc_status_dc_dc_on       (dc_dc_on),
    .lvmc_dclink_lvdcdc_fb_adc_sync_dat_i_phase_a
(BOOST_DRV0_CURRENT_MDAT),
    .lvmc_dclink_lvdcdc_fb_adc_sync_dat_i_phase_b
(BOOST_DRV1_CURRENT_MDAT),
    .lvmc_dclink_lvdcdc_fb_adc_sync_dat_v_out
(DCBUS_VOLTAGE_MDAT),

    .lvmc_dclink_dc_in_v_status_sync_dat
(INPUT_VOLTAGE_MDAT),
    .lvmc_dclink_dc_in_v_status_dc_link_enable           (1'b1),
    .lvmc_dclink_dc_in_v_status_overvoltage
(dcin_overvoltage),
    .lvmc_dclink_dc_in_v_status_undervoltage
(dcin_undervoltage),
    .lvmc_dclink_dc_in_v_status_chopper                  () ,
    .lvmc_dclink_dc_in_v_status_sync_dat
(INPUT_CURRENT_MDAT),
    .lvmc_dclink_dc_in_i_status_dc_link_enable           (1'b1),
    .lvmc_dclink_dc_in_i_status_overvoltage
(dcin_overcurrent),
    .lvmc_dclink_dc_in_i_status_undervoltage
    .lvmc_dclink_dc_in_i_status_chopper                  () ,
    .lvmc_dclink_dc_link_i_status_sync_dat
(DCBUS_CURRENT_MDAT),
    .lvmc_dclink_dc_link_i_status_dc_link_enable         (1'b1),
    .lvmc_dclink_dc_link_i_status_overvoltage
(dclink_overcurrent),
    .lvmc_dclink_dc_link_i_status_undervoltage
    .lvmc_dclink_dc_link_i_status_chopper                () ,
// SPI control bus for inverter gate drivers
    .gate_drive_spi_MISO                                (DRV_SOMI),
    .gate_drive_spi莫斯                                (DRV_SIMO),
    .gate_drive_spi_SCLK                                (DRV_SCLK),
    .gate_drive_spi_SS_n
(gate_drive_spi_ss_n),
    .encoder_select_export                            (encoder_select),
    .sync_in_export
// Axis 0
    .drive0_adc_sync_dat_u
(DRV0_U_CURRENT_MDAT),
    .drive0_adc_sync_dat_v
(DRV0_V_CURRENT_MDAT),
    .drive0_adc_sync_dat_w
(DRV0_W_CURRENT_MDAT),
    .drive0_adc_overcurrent
    .drive0_adc_overcurrent
    (Overcurrent0),
    .drive0_adc_pow_sync_dat_u
(DRV0_U_VOLTS_MDAT),
    .drive0_adc_pow_sync_dat_v
(DRV0_V_VOLTS_MDAT),
    .drive0_adc_pow_sync_dat_w
(DRV0_W_VOLTS_MDAT),
    .drive0_adc_pow_overcurrent
    .drive0_adc_overcurrent
    (Overcurrent0),
    .drive0_sm_overcurrent
    .drive0_sm_overvoltage
    .drive0_sm_undervoltage
    .drive0_sm_chopper
    .drive0_sm_dc_link_clk_err
    .drive0_sm_igbt_err
    .drive0_sm_error_out
    .drive0_sm_overcurrent_latch
    .drive0_sm_overvoltage_latch
    .drive0_sm_undervoltage_latch
    .drive0_sm_dc_link_clk_err_latch
    (oc_latch_led0),
    (ov_latch_led0),
    (),
    (),
    ()
,
```

```

    .drive0_sm_igbt_err_latch          (),
    .drive0_sm_chopper_latch          (),

    .drive0_pwm_encoder_strobe_n      (enc_stb0_n),
    .drive0_pwm_u_h                  (),
    (HSMC_DRV0_PWM_UH),
    .drive0_pwm_u_l                  (HSMC_DRV0_PWM_UL),
    .drive0_pwm_v_h                  (HSMC_DRV0_PWM_VH),
    .drive0_pwm_v_l                  (HSMC_DRV0_PWM_VL),
    .drive0_pwm_w_h                  (HSMC_DRV0_PWM_WH),
    .drive0_pwm_w_l                  (HSMC_DRV0_PWM_WL),

    .drive0_qep_strobe              (~enc_stb0_n),
    .drive0_qep_QEP_A               (DRV0_QR_A),
    .drive0_qep_QEP_B               (DRV0_QR_B),
    .drive0_qep_QEP_I               (DRV0_QR_Z),
    .drive0_hall_pio_export        ({DRV0_QHR_W,
DRV0_QHR_V, DRV0_QHR_U}),

    .drive0_rslvr_spi_ctrl_MISO     (1'b0),
    .drive0_rslvr_spi_ctrl莫斯I   (),
    (DRV0_SER_TX_RSLVR),
    .drive0_rslvr_spi_ctrl_SCLK    (DRV0_SER_CLK_RSLVR_CTRL),
    .drive0_rslvr_spi_ctrl_SS_n    (DRV0_RESOLVER_SSCS),

    .drive0_rslvr_spi_posn_MISO    (DRV0_SER_RX),
    .drive0_rslvr_spi_posn莫斯I   (),
    .drive0_rslvr_spi_posn_SCLK    (DRV0_SER_CLK_RSLVR_POSN),
    .drive0_rslvr_spi_posn_SS_n    (DRV0_RESOLVER_SCSB),

    .drive0_rslvr_pio_in_port      ({1'b0,
DRV0_RESOLVER_ERRHLD}),
    .drive0_rslvr_pio_out_port     (drive0_rslvr_pio_out),

    // Axis 1
    .drive1_adc_sync_dat_u         (Overcurrent1),
    (DRV1_U_CURRENT_MDAT),
    .drive1_adc_sync_dat_v         (DRV1_V_CURRENT_MDAT),
    .drive1_adc_sync_dat_w         (DRV1_W_CURRENT_MDAT),
    .drive1_adc_overcurrent

    .drive1_adc_pow_sync_dat_u     (Overcurrent1),
    (DRV1_U_VOLTS_MDAT),
    .drive1_adc_pow_sync_dat_v     (DRV1_V_VOLTS_MDAT),
    .drive1_adc_pow_sync_dat_w     (DRV1_W_VOLTS_MDAT),
    .drive1_adc_pow_overcurrent

    .drive1_sm_overcurrent         (Overcurrent1),
    .drive1_sm_overvoltage        (1'b0),
    (1'b0),
    .drive1_sm_undervoltage       (1'b0),
    .drive1_sm_chopper            (1'b0),
    .drive1_sm_dc_link_clk_err   (1'b0),
    .drive1_sm_igbt_err           (~DRV1FAULTn),
    .drive1_sm_error_out          (),

    .drive1_sm_overcurrent_latch   (oc_latch_led1),
    .drive1_sm_overvoltage_latch   (ov_latch_led1),
    .drive1_sm_undervoltage_latch  ()
  
```

```

.drive1_sm_dc_link_clk_err_latch          (),
.drive1_sm_igbt_err_latch                 (),
.drive1_sm_chopper_latch                  (),

.drive1_pwm_encoder_strobe_n              (enc_stbl_n),
.drive1_pwm_u_h                          (HSMC_DRV1_PWM_UH),
.drive1_pwm_u_l                          (HSMC_DRV1_PWM_UL),
.drive1_pwm_v_h                          (HSMC_DRV1_PWM_VH),
.drive1_pwm_v_l                          (HSMC_DRV1_PWM_VL),
.drive1_pwm_w_h                          (HSMC_DRV1_PWM_WH),
.drive1_pwm_w_l                          (HSMC_DRV1_PWM_WL),

.drive1_qep_strobe                      (~enc_stbl_n),
.drive1_qep_QEP_A                        (DRV1_QR_A),
.drive1_qep_QEP_B                        (DRV1_QR_B),
.drive1_qep_QEP_I                        (DRV1_QR_Z),

.drive1_hall_pio_export                 ({DRV1_QHR_W,
DRV1_QHR_V, DRV1_QHR_U}),
.drive1_rslvr_spi_ctrl_MISO             (1'b0),
.drive1_rslvr_spi_ctrl_MOSI             (DRV1_SER_RX_RSLVR),
.drive1_rslvr_spi_ctrl_SCLK             (DRV1_SER_CLK_RSLVR_CTRL),
.drive1_rslvr_spi_ctrl_SS_n             (DRV1_RESOLVER_SSNS),
(drive1_resolver_ssns),

.drive1_rslvr_spi_posn_MISO             (DRV1_SER_RX),
.drive1_rslvr_spi_posn_MOSI             (),
.drive1_rslvr_spi_posn_SCLK             (DRV1_SER_CLK_RSLVR_POSN),
.drive1_rslvr_spi_posn_SS_n             (DRV1_RESOLVER_SCSB),
(drive1_resolver_scsb),

.drive1_rslvr_pio_in_port               ({1'b0,
DRV1_RESOLVER_ERRHLD}),
.drive1_rslvr_pio_out_port              (drive1_rslvr_pio_out),
(drive1_rslvr_pio_out),


// DDR3 interface FPGA
.ddr3_fpga_mem_a                       (mem_a),
.ddr3_fpga_mem_ba                      (mem_ba),
.ddr3_fpga_mem_ck                       (mem_ck),
.ddr3_fpga_mem_ck_n                     (mem_ck_n),
.ddr3_fpga_mem_cke                      (mem_cke),
.ddr3_fpga_mem_cs_n                     (mem_cs_n),
.ddr3_fpga_mem_dm                       (mem_dm),
.ddr3_fpga_mem_ras_n                    (mem_ras_n),
.ddr3_fpga_mem_cas_n                   (mem_cas_n),
.ddr3_fpga_mem_we_n                     (mem_we_n),
.ddr3_fpga_mem_reset_n                 (mem_reset_n),
.ddr3_fpga_mem_dq                       (mem_dq),
.ddr3_fpga_mem_dqs                      (mem_dqs),
.ddr3_fpga_mem_dqs_n                   (mem_dqs_n),
.ddr3_fpga_mem_odt                      (mem_odt),
.ddr3_fpga_status_local_init_done      (),
.ddr3_fpga_status_local_cal_success    (),
.ddr3_fpga_status_local_cal_fail       (),
.fpga_mem_oct_rzqin                    (mem_oct_rzqin),
//TSN sub system
//HPS io
.hps_io_hps_io_emac1_inst_TX_CLK      (enet_hps_gtx_clk),
.hps_io_hps_io_emac1_inst_TXD0        (enet_hps_txd[0]),
(enet_hps_txd[0]),

```

```

.hps_io_hps_io_emac1_inst_TXD1
(enet_hps_txd[1]),
.hps_io_hps_io_emac1_inst_TXD2
(enet_hps_txd[2]),
.hps_io_hps_io_emac1_inst_TXD3
(enet_hps_txd[3]),
.hps_io_hps_io_emac1_inst_MDIO,
.hps_io_hps_io_emac1_inst_MDC,
.hps_io_hps_io_emac1_inst_RX_CTL,
.hps_io_hps_io_emac1_inst_TX_CTL,
.hps_io_hps_io_emac1_inst_RX_CLK
(enet_hps_rx_clk),
.hps_io_hps_io_emac1_inst_RXD0
(enet_hps_rxd[0]),
.hps_io_hps_io_emac1_inst_RXD1
(enet_hps_rxd[1]),
.hps_io_hps_io_emac1_inst_RXD2
(enet_hps_rxd[2]),
.hps_io_hps_io_emac1_inst_RXD3
(enet_hps_rxd[3]),

.hps_io_hps_io_sdio_inst_CMD,
.hps_io_hps_io_sdio_inst_CLK,
.hps_io_hps_io_sdio_inst_D0,
.hps_io_hps_io_sdio_inst_D1,
.hps_io_hps_io_sdio_inst_D2,
.hps_io_hps_io_sdio_inst_D3
(sd_cmd),
(sd_clk),
(sd_dat[0]),
(sd_dat[1]),
(sd_dat[2]),
(sd_dat[3]),

.hps_io_hps_io_uart0_inst_RX
.hps_io_hps_io_uart0_inst_TX
(uart_rx),
(uart_tx),

.hps_io_hps_io_i2c0_inst_SDA
.hps_io_hps_io_i2c0_inst_SCL
(i2c_sda_hps),
(i2c_scl_hps),

//HPS memory
.memory_mem_a,
.memory_mem_ba,
.memory_mem_ck,
.memory_mem_ck_n,
.memory_mem_cke,
.memory_mem_cs_n,
.memory_mem_ras_n,
.memory_mem_cas_n,
.memory_mem_we_n,
.memory_mem_reset_n
(ddr3_hps_a),
(ddr3_hps_ba),
(ddr3_hps_ck),
(ddr3_hps_ck_n),
(ddr3_hps_cke),
(ddr3_hps_cs_n),
(ddr3_hps_ras_n),
(ddr3_hps_cas_n),
(ddr3_hps_we_n),
(ddr3_hps_reset_n),

.memory_mem_dq,
.memory_mem_dqs,
.memory_mem_dqs_n,
.memory_mem_odt,
.memory_mem_dm
(memory_dq),
(memory_dqs),
(memory_dqs_n),
(memory_odt),
(memory_dm),

.memory_oct_rzqin
(ddr3_hps_oct_rzqin),
//HPS TSN
.eth_mdio_tristate_0_mdio_mdc
(eth_mdio_tristate_0_mdio_mdc),
.eth_mdio_tristate_0_mdio_mdio
(eth_mdio_tristate_0_mdio_mdio),
.de_ip_solution_scv_0_pps_pps
.pio_0_external_connection_export
(pps),
(enet_dual_resetn),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_rx_clk
(deip_0_gmii_to_mii_0_mii_rx_clk),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_rx_dv
(deip_0_gmii_to_mii_0_mii_rx_dv),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_rxd
(deip_0_gmii_to_mii_0_mii_rxd),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_rx_er
(deip_0_gmii_to_mii_0_mii_rx_er),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_tx_clk
(deip_0_gmii_to_mii_0_mii_tx_clk),

```

```
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_tx_en
(deip_0_gmii_to_mii_0_mii_tx_en_i),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_txd
(deip_0_gmii_to_mii_0_mii_txd_i),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_tx_er      (),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_crs      (1'b0),
.de_ip_solution_scv_0_gmii_to_mii_0_mii_mii_col      (1'b0),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_rx_clk
(deip_0_gmii_to_mii_1_mii_rx_clk),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_rx_dv
(deip_0_gmii_to_mii_1_mii_rx_dv),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_rxd
(deip_0_gmii_to_mii_1_mii_rxd),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_rx_er
(deip_0_gmii_to_mii_1_mii_rx_er),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_tx_clk
(deip_0_gmii_to_mii_1_mii_tx_clk),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_tx_en
(deip_0_gmii_to_mii_1_mii_tx_en_i),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_txd
(deip_0_gmii_to_mii_1_mii_txd_i),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_tx_er      (),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_crs      (1'b0),
.de_ip_solution_scv_0_gmii_to_mii_1_mii_mii_col      (1'b0)
);
endmodule
```

C. YOCTO Build Patch File (cvsx_doc_tsn_2_3-rt) for the TSN Drive-on-Chip Design Example

```

diff -Naur ./tmp_orig/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/
2018.03/de-eval-board/socfpga.c ./tmp_mod_rt/ip-solutions-ref/meta-local/
recipes-bsp/u-boot/files/2018.03/de-eval-board/socfpga.c
--- ./tmp_orig/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/2018.03/de-
eval-board/socfpga.c    2021-03-15 14:28:16.000000000 +0000
+++ ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/
2018.03/de-eval-board/socfpga.c    2021-07-28 15:51:41.262757585 +0100
@@ -17,7 +17,7 @@
     int load_fpga_file(char *, char *, unsigned long);

#define EEP_VPD_I2C_ADDR      0x51
-#define EEP_MAC_OFFSET       0x01
+#define EEP_MAC_OFFSET       0x2D
#define EEP_SERIAL_OFFSET     0x7
#define SERIAL_LEN            10

@@ -50,7 +50,7 @@
     const void *blob = gd->fdt_blob;

     printf("Reading MAC addresses from EEPROM ...\\n");
- i2c_read(EEP_VPD_I2C_ADDR, EEP_MAC_OFFSET, 1,
+ i2c_read(EEP_VPD_I2C_ADDR, EEP_MAC_OFFSET, 2,
           enetaddr, sizeof(enetaddr));

     eth_env_set_enetaddr_by_index("eth", 0, enetaddr);
@@ -169,4 +169,4 @@
#endif
    return 0;
}
#endif
\ No newline at end of file
#endif
diff -Naur ./tmp_orig/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/
2018.03/de-eval-board/socfpga_cyclone5_de_eval_board_defconfig ./tmp_mod_rt/ip-
solutions-ref/meta-local/recipes-bsp/u-boot/files/2018.03/de-eval-board/
socfpga_cyclone5_de_eval_board_defconfig
--- ./tmp_orig/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/2018.03/de-
eval-board/socfpga_cyclone5_de_eval_board_defconfig    2021-03-15
14:28:16.000000000 +0000
+++ ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/
2018.03/de-eval-board/socfpga_cyclone5_de_eval_board_defconfig    2021-07-28
15:51:41.262757585 +0100
@@ -120,8 +120,8 @@
 CONFIG_USE_TINY_PRINTF=y
 # CONFIG_LIB_RAND is not set
 CONFIG_AUTOBOOT_KEYED=y
-CONFIG_AUTOBOOT_PROMPT="autoboot in %d seconds. Stop by pressing \\\"na\\\"\\n"
-CONFIG_AUTOBOOT_STOP_STR="na"
+CONFIG_AUTOBOOT_PROMPT="autoboot in %d seconds. Stop by pressing \\\"n\\\"\\n"
+CONFIG_AUTOBOOT_STOP_STR="n"
 CONFIG_BOOTDELAY=5
 CONFIG_NET_RANDOM_ETHADDR=y
 CONFIG_SYS_NO_FLASH=y
diff -Naur ./tmp_orig/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/
2018.03/de-eval-board/socfpga_cyclone5_socdk.h ./tmp_mod_rt/ip-solutions-ref/
meta-local/recipes-bsp/u-boot/files/2018.03/de-eval-board/

```

```
socfpga_cyclone5_socdk.h
--- ./tmp_orig/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/2018.03/de-
eval-board/socfpga_cyclone5_socdk.h      2021-03-15 14:28:16.000000000 +0000
+++ ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-bsp/u-boot/files/
2018.03/de-eval-board/socfpga_cyclone5_socdk.h      2021-07-28 15:51:41.262757585
+0100
@@ -17,7 +17,7 @@
#define PHYS_SDRAM_1_SIZE          0x40000000 /* 1GiB on SoCDK */

/* Booting Linux */
-#define CONFIG_BOOTDELAY          3
+#define CONFIG_BOOTDELAY          5
#define CONFIG_BOOTFILE           "zImage"
#define CONFIG_BOOTARGS           "console=ttyS0,"
__stringify(CONFIG_BAUDRATE)
#define CONFIG_BOOTCOMMAND         "run callscript; run mmcload; run
mmcboot"
diff -Naur ./tmp_orig/ip-solutions-ref/meta-local/recipes-ttt-rootfs/images/
recovery-rootfs_1.0.0.bb ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-ttt-
rootfs/images/recovery-rootfs_1.0.0.bb
--- ./tmp_orig/ip-solutions-ref/meta-local/recipes-ttt-rootfs/images/recovery-
rootfs_1.0.0.bb      2021-03-15 14:28:16.000000000 +0000
+++ ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-ttt-rootfs/images/recovery-
rootfs_1.0.0.bb      2021-07-28 15:51:41.266757555 +0100
@@ -1,7 +1,7 @@
DESCRIPTION = "A TTTech recovery root filesystem for update"

IMAGE_FEATURES += "ssh-server-openssh"
-IMAGE_FSTYPES = "ext4"
+IMAGE_FSTYPES = "ext4 tar.gz"

IMAGE_INSTALL = "\
    packagegroup-core-boot \
diff -Naur ./tmp_orig/ip-solutions-ref/meta-local/recipes-ttt-rootfs/images/
sdcard-image_1.0.0.bb ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-ttt-
rootfs/images/sdcard-image_1.0.0.bb
--- ./tmp_orig/ip-solutions-ref/meta-local/recipes-ttt-rootfs/images/sdcard-
image_1.0.0.bb      2021-03-15 15:04:17.000000000 +0000
+++ ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-ttt-rootfs/images/sdcard-
image_1.0.0.bb      2021-07-28 15:51:41.266757555 +0100
@@ -68,6 +68,10 @@
    linuxptp \
    devmem2 \
    python-pip \
+   gdbserver \
+   openssh-sftp-server \
+   packagegroup-core-ssh-openssh \
+   rt-tests \
    haveged \
"
diff -Naur ./tmp_orig/ip-solutions-ref/meta-local/recipes-ttt-rootfs/init/de-
eval-board/ttt-ip-init.sh ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-ttt-
rootfs/init/de-eval-board/ttt-ip-init.sh
--- ./tmp_orig/ip-solutions-ref/meta-local/recipes-ttt-rootfs/init/de-eval-
board/ttt-ip-init.sh      2021-03-15 14:28:16.000000000 +0000
+++ ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-ttt-rootfs/init/de-eval-
board/ttt-ip-init.sh      2021-07-28 15:51:41.266757555 +0100
@@ -26,16 +26,23 @@
    # Get a MAC address from the EEPROM from a given offset.
    get_mac()
    {
-       ret_mac=0
-       for i in {1..6}; do
-           o=$((i2cget -y 0 0x51 $i b))
-           ret_mac=$((ret_mac *0x100 +$o))
-       done
-       ret_mac=$((ret_mac + $1))
+       #ret_mac=0
+       #for i in {1..6}; do
+       #   o=$((i2cget -y 0 0x51 $i b)))
    }
```

```

+     #      ret_mac=$(( $ret_mac *0x100 +$o ))
+     #done
+     #ret_mac=$(( $ret_mac + $1 ))
# hex-print the mac address (12 chars = 6 octets in hex)
# ... first sed puts ':' after every 2 chars, i.e., after every octet
# ... second sed removes the last ':' trailing the line
- printf "%012x" $ret_mac | sed 's/\(\.\{2\}\)\/\1:/g' | sed 's/:$/\\'
+ #printf "%012x" $ret_mac | sed 's/\(\.\{2\}\)\/\1:/g' | sed 's/:$/\\'
+ ret_mac=""
+ i2cset -y 0 0x51 0 $1 i
+ for i in 0 1 2 3 4 5; do
+   ret_mac=$ret_mac$(i2cget -y 0 0x51 | sed 's/0x/:/')
+ done
+ echo ${ret_mac#::}
+
}

# Configure delays.
@@ -61,15 +68,15 @@
echo -n "edgx_mdio-1:00" > /sys/class/net/sw0p2/phy/mdiobus
echo -n "edgx_mdio-1:01" > /sys/class/net/sw0p3/phy/mdiobus
- echo -n "edgx_mdio-1:02" > /sys/class/net/sw0p4/phy/mdiobus
- echo -n "edgx_mdio-1:03" > /sys/class/net/sw0p5/phy/mdiobus
+ #echo -n "edgx_mdio-1:02" > /sys/class/net/sw0p4/phy/mdiobus
+ #echo -n "edgx_mdio-1:03" > /sys/class/net/sw0p5/phy/mdiobus

- ip link set dev sw0p1 address $(get_mac 3)
- ip link set dev sw0p2 address $(get_mac 4)
- ip link set dev sw0p3 address $(get_mac 5)
- ip link set dev sw0p4 address $(get_mac 6)
- ip link set dev sw0p5 address $(get_mac 7)
- ip link set dev sw0ep address $(get_mac 0)
+ ip link set dev sw0p1 address $(get_mac 45)
+ ip link set dev sw0p2 address $(get_mac 51)
+ ip link set dev sw0p3 address $(get_mac 57)
+ #ip link set dev sw0p4 address $(get_mac 6)
+ #ip link set dev sw0p5 address $(get_mac 7)
+ ip link set dev sw0ep address $(get_mac 69)

# set default mqprio setup for 3 traffic classes
tc qdisc add dev sw0ep root mqprio num_tc 3 map 0 0 0 0 1 1 2 2 2 2 2
2 2 2 hw 1 mode channel
@@ -78,10 +85,10 @@
    tc qdisc replace dev sw0ep parent 8001:3 pfifo

# Marvell 88E1510P - standard latency mode
- set_delays sw0p2 phydev tx 4032 412 109 rx 1083 220 203
- set_delays sw0p3 phydev tx 4032 412 109 rx 1083 220 203
- set_delays sw0p4 phydev tx 4032 412 109 rx 1083 220 203
- set_delays sw0p5 phydev tx 4032 412 109 rx 1083 220 203
+ set_delays sw0p2 phydev tx 1663 156 92 rx 1140 180 207
+ set_delays sw0p3 phydev tx 1663 156 92 rx 1140 180 207
+ #set_delays sw0p4 phydev tx 4032 412 109 rx 1083 220 203
+ #set_delays sw0p5 phydev tx 4032 412 109 rx 1083 220 203
}

stop()
@@ -94,7 +101,7 @@
case "$1" in
  start|restart|force-reload)
    echo "Configuring FPGA ..."
-    echo "  Hardware S/N: $(print_serial)"
+    #echo "  Hardware S/N: $(print_serial)"
    start
  ;;
  stop)
diff -Naur ./tmp_orig/ip-solutions-ref/meta-local/recipes-ttt-rootfs/init-ifupdown/de-eval-board/interfaces ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-ttt-rootfs/init-ifupdown/de-eval-board/interfaces
--- ./tmp_orig/ip-solutions-ref/meta-local/recipes-ttt-rootfs/init-ifupdown/de-

```

```

eval-board/interfaces      2021-03-15 14:28:16.000000000 +0000
+++ ./tmp_mod_rt/ip-solutions-ref/meta-local/recipes-ttt-rootfs/init-
ifupdown/de-eval-board/interfaces      2021-07-28 15:51:41.266757555 +0100
@@ -10,7 +10,7 @@
@@ -10,7 +10,7 @@
@@ -20,8 +20,7 @@
    up ip link set dev sw0p1 master $IFACE up
    up ip link set dev sw0p2 master $IFACE up
    up ip link set dev sw0p3 master $IFACE up
-   up ip link set dev sw0p4 master $IFACE up
-   up ip link set dev sw0p5 master $IFACE up
+
    up ip link set $IFACE type bridge stp_state 1
    up mstptctl addbridge $IFACE
    up mstptctl setforcevers $IFACE mstp
diff -Naur ./tmp_orig/poky/meta-tsn-base/recipes-sysrepo/libyang/
libyang_1.0.164.bb ./tmp_mod_rt/poky/meta-tsn-base/recipes-sysrepo/libyang/
libyang_1.0.164.bb
--- ./tmp_orig/poky/meta-tsn-base/recipes-sysrepo/libyang/libyang_1.0.164.bb
2021-03-15 14:28:29.000000000 +0000
+++ ./tmp_mod_rt/poky/meta-tsn-base/recipes-sysrepo/libyang/
libyang_1.0.164.bb    2021-07-28 15:51:40.054766660 +0100
@@ -4,7 +4,7 @@
 LICENSE = "BSD-3-Clause"
 LIC_FILES_CHKSUM = "file://LICENSE;md5=2982272c97a8e417a844857ca0d303b1"

-SRC_URI = "git://github.com/CESNET/libyang.git;protocol=https;branch=devel"
+SRC_URI = "git://github.com/CESNET/libyang.git;protocol=https;nobranch=1"

#PR ="r1"
#SRCREV = "v${PV}"
diff -Naur ./tmp_orig/poky/meta-tsn-bsp/conf/machine/de-eval-board.conf ./
tmp_mod_rt/poky/meta-tsn-bsp/conf/machine/de-eval-board.conf
--- ./tmp_orig/poky/meta-tsn-bsp/conf/machine/de-eval-board.conf    2021-03-15
14:28:31.000000000 +0000
+++ ./tmp_mod_rt/poky/meta-tsn-bsp/conf/machine/de-eval-board.conf
2021-07-28 15:51:41.122758637 +0100
@@ -7,7 +7,7 @@
 # disable rtc support
 MACHINE_FEATURES_BACKFILL_CONSIDERED += "rtc"

-IMAGE_FSTYPES = "wic.gz"
+IMAGE_FSTYPES = "wic.gz tar.gz"
WKS_FILE = "sdimage-cyclone5-de-eval-board.wks"

KMACHINE = "cyclone5"
diff -Naur ./tmp_orig/poky/meta-tsn-bsp/recipes-bsp/u-boot-scr/files/de-eval-
board/u-boot.script ./tmp_mod_rt/poky/meta-tsn-bsp/recipes-bsp/u-boot-scr/
files/de-eval-board/u-boot.script
--- ./tmp_orig/poky/meta-tsn-bsp/recipes-bsp/u-boot-scr/files/de-eval-board/u-
boot.script    2021-03-15 14:28:31.000000000 +0000
+++ ./tmp_mod_rt/poky/meta-tsn-bsp/recipes-bsp/u-boot-scr/files/de-eval-board/u-
boot.script    2021-07-28 15:51:41.122758637 +0100
@@ -3,13 +3,13 @@
    setenv bootimage zImage
    setenv fdtimage socfpga_cyclone5_de-eval-board.dtb
    setenv fpgaimage socfpga.rbf
-setenv fpgadata 0x3000000
+#setenv fpgadata 0x3000000

-echo --- Programming FPGA ---
+echo --- Not Programming FPGA ---

-load mmc 0:1 $fpgadata $fpgaimage

```



```
+#load mmc 0:1 $fpgadata $fpgaimage
bridge disable
-fpga load 0 $fpgadata $filesize
+#fpga load 0 $fpgadata $filesize
bridge enable

diff -Naur ./tmp_orig/poky/meta-tsn-bsp/recipes-kernel/linux/5.4/dts/
socfpga_cyclone5_de-eval-board_default.dts ./tmp_mod_rt/poky/meta-tsn-bsp/
recipes-kernel/linux/5.4/dts/socfpga_cyclone5_de-eval-board_default.dts
--- ./tmp_orig/poky/meta-tsn-bsp/recipes-kernel/linux/5.4/dts/
socfpga_cyclone5_de-eval-board_default.dts      2021-03-15 14:28:31.000000000
+0000
+++ ./tmp_mod_rt/poky/meta-tsn-bsp/recipes-kernel/linux/5.4/dts/
socfpga_cyclone5_de-eval-board_default.dts      2021-07-28 15:51:41.150758427
+0100
@@ -47,9 +47,9 @@
    compatible = "snps,dwmac-mdio";

    /* Micrel KSZ9031 */
-    hps_phyl: phy@3 {
+    hps_phyl: phy@4 {
        compatible = "ethernet-phy-ieee802.3-c22";
-        reg = <0x3>;
+        reg = <0x4>;
        rxc-skew-ps = <1680>;      /* 780 ps */
        rxdv-skew-ps = <420>;     /* 0 ps */
        txc-skew-ps = <1860>;     /* 960 ps */
@@ -102,9 +102,9 @@
 &i2c0 {
    status = "okay";

-    hps1_eth_eeprom: eeprom@50 {
+    hps1_eth_eeprom: eeprom@51 {
        compatible = "atmel,at24c02";
-        reg = <0x50>;
+        reg = <0x51>;
    };
};

diff -Naur ./tmp_orig/poky/meta-tsn-bsp/recipes-kernel/linux/linux-tttech-
industrial_5.4.bb ./tmp_mod_rt/poky/meta-tsn-bsp/recipes-kernel/linux/linux-
tttech-industrial_5.4.bb
--- ./tmp_orig/poky/meta-tsn-bsp/recipes-kernel/linux/linux-tttech-
industrial_5.4.bb      2021-03-15 14:28:31.000000000 +0000
+++ ./tmp_mod_rt/poky/meta-tsn-bsp/recipes-kernel/linux/linux-tttech-
industrial_5.4.bb      2021-07-28 15:55:36.216992574 +0100
@@ -1,6 +1,14 @@
 PR = "r0"

+LINUX_VERSION_PATCHLEVEL ?= "40-rt24"
+
 include linux-tttech-industrial_5.4.inc

+# patch it to 5.4.40-rt24
+SRC_URI += "file://Upgrade-to-kernel-version-5.4.40-rt24.patch"
+
 # machine specific kernel configuration fragment
 SRC_URI += "file://${MACHINE}.cfg"
+
+# basic rt-preempt configuration
+SRC_URI += file://preempt.cfg
```

D. Script to read and change MAC addresses from Cyclone V SoC EEPROM

To check the physical addresses in the Cyclone V SoC development board, for the TSN Drive-on-Chip Design example.

```
#!/bin/sh
# Get a MAC address from the EEPROM from a given offset.
get_mac()
{
    ret_mac=""
    i2cset -y 0 0x51 0 $1 i
    for i in 0 1 2 3 4 5; do
        ret_mac=$ret_mac$(i2cget -y 0 0x51 | sed 's/0x:/:/')
    done
    echo ${ret_mac#:}
}

help()
{
    echo ""
    echo "To read and write physical MAC addresses in the Cyclone V SoC"
    echo "To read type:"
    echo "      >> ./Read_Write_MAC_CVSX.sh read"
    echo "To write type:"
    echo "      >> ./Read_Write_MAC_CVSX.sh write <MAC address> <offset>"
    echo "Example"
    echo "      >> ./Read_Write_MAC_CVSX.sh write AA:BB:CC:DD:FF:GG 45"
    echo "This script does not validate the address, MAC addresses must be"
    echo "typed only in the prev"
    echo ""
}

case "$1" in
    read)
        echo 'Reading the EEPROM...'
        echo 'MAC store in adres offset 45' $(get_mac 45)
        echo 'MAC store in adres offset 51' $(get_mac 51)
        echo 'MAC store in adres offset 57' $(get_mac 57)
        echo 'MAC store in adres offset 63' $(get_mac 63)
        echo 'MAC store in adres offset 69' $(get_mac 69)
        echo 'MAC store in adres offset 75' $(get_mac 75)
        echo ''
        ;;
    write)
        echo ""
        echo "Writing EEPROM..."
        A1=$(cut -d':' -f1 <<< $2)
        A2=$(cut -d':' -f2 <<< $2)
        A3=$(cut -d':' -f3 <<< $2)
        A4=$(cut -d':' -f4 <<< $2)
        A5=$(cut -d':' -f5 <<< $2)
        A6=$(cut -d':' -f6 <<< $2)

        i2cset -y 0 0x51 0 $3 0x$A1 i
        i2cset -y 0 0x51 0 ${((3+1))} 0x$A2 i
        i2cset -y 0 0x51 0 ${((3+2))} 0x$A3 i

```

```
i2cset -y 0 0x51 0 ${((\$3+3))} 0xA4 i
i2cset -y 0 0x51 0 ${((\$3+4))} 0xA5 i
i2cset -y 0 0x51 0 ${((\$3+5))} 0xA6 i
echo "Address \$2 written to EEPROM offset \$3"
echo ' '
;;
*)
    help
;;
esac
exit 0
```