Arria 10 SerialLite II Design Example

This document describes an Arria 10 SerialLite II design example with 4 lanes at 10Gbps per lane. It shows how to create a SerialLite II link in an Arria 10 device and walks through the hardware demonstration using an Altera Arria 10 FPGA Development Kit.

Introduction
The SerialLite II MegaCore function is a lightweight protocol suitable for packet and streaming data in chip-to-chip, board-to-board, and backplane applications. It consists of a serial link of up to 16 lanes with low logic utilization and minimum data transfer latency. For reliable data transfer, it provides optional link management features such as retry-on-error and flow control.

A SerialLink II link consists of two instantiations of logic implementing the SerialLite II protocol. Each end of the link has a transmitter and a receiver, as shown in the figure below.

Figure 1. SerialLite II Link
Getting Started

The SerialLite II MegaCore generation in Arria 10 devices is supported indirectly. From ACDS 12.0 onwards, the transceiver PHY is not part of the generated SerialLite II MegaCore. This document shows how to create an Arria 10 SerialLite II link that can provide 10Gbps bandwidth per lane, exceeding the SerialLite II maximum data rate 6.375Gbps. The design demonstrates that with the current SerialLite II MegaCore and Arria 10 GX speed grade, there is no issue in running at higher data rates as timing closure is fully met.

One key requirement is that a special license is required to compile for Arria 10 devices. To request special license, email SLII@altera.com. In the email, please provide the following information.

- Customer name:
- Location:
- Contact name:
- Contact email address:
- Fixed or floating license required:
- If floating, # of floating seats:
- NIC/Host ID:
- Device family variant: (e.g. Arria 10 GX or Arria 10 GT)
- Altera Sales / FAE name or distributor contact name:

This document consists of the following sections.

- Overview
- Creating an Arria 10 SerialLite II link
- Simulating the design example
- Demo walkthrough
Overview
Features of the design example are as follows:

- SerialLite II MegaCore with 32-bit data width per lane
- Qsys system includes Nios II processor, control and status registers, and provides access to ATX PLL and transceiver PHY reconfiguration register space
- VHDL packet generator and verifier with 128-bit Atlantic interface. The packet length and inter-packet gap can be dynamically changed through Nios II terminal
- User data rate and efficiency are calculated based on packets sent and received
- The transceiver PMA settings can be dynamically modified via Nios II terminal
- Transceiver Toolkit support is available through Altera Debug Master Endpoint (ADME) option
- 4 duplex lanes at 10Gbps per lane routed to the FMC Port A connector where external loopback can be made
The following figure is the system block diagram of the design example.

**Figure 2. Design Example Block Diagram**

**Tx Path**
Packet Generator generates packets according to Atlantic specification and sends the packets to the SerialLite II MegaCore.

- Data bus is 128-bit wide, 32-bits per lane
- SOP data word is 0xFEEDBEEF on each lane
- EOP data word is 0xB00FF00D on each lane
- Packet sequence number (8-bit incrementing pattern) is sent on TADD bus
- Between SOP and EOP, incrementing count pattern is transmitted on each lane
- Packet bit errors can be inserted (only on lane 0)
- A 32-bit counter is used to count the number of packets that have been sent (in 1000 packets)
The following shows the Tx path connection.

**Figure 3. Design Example Tx Path**

![Diagram of Design Example Tx Path]

**Rx Path**
Packet Monitor checks for data corruption in the packets received from SerialLite II MegaCore. For example:

- SOP and EOP frame words are checked and packet error count is incremented if they are corrupted
- Packet sequence number (8-bit incrementing pattern) received on RADD bus is verified. Packet error count increments on detecting incorrect sequence
- Datalocked is asserted when all lanes have received consecutive 128 numbers in correct sequence
- A 32-bit counter is used to count the number of packets that have been received (in 1000 packets)
The following shows the Rx path connection.

**Figure 4. Design Example Rx Path**

---

**Creating Arria 10 SerialLite II Link**

The following section describes how to create a SerialLite II link for an Arria 10 device using the design example.

**SerialLite II MegaCore Generation**

The SerialLite II MegaCore is not available for Arria 10 device. You need to generate and use the SerialLite II MegaCore from Stratix V. Since the SerialLite II MegaCore no longer includes the transceiver PHY, the device family setting here does not matter.

To view the IP parameter setting for the design example, click File → Open, select `core/sl2_core.vhd`.

Once the core has been generated, add the following generated files to the project. `<IP variation name>` is the name of the SerialLite II MegaCore you select (‘sl2_core’ in the design example):

- `<IP variation name>.vhd` (or `<IP variation name>.v`)
- `<IP variation name>_slite2_wrapper.v`
- `<IP variation name>_slite2_top.v`
- `<IP variation name>_slite2_core.v`
- `<IP variation name>_slite2_unenc.v`
<IP variation name>_txdrp_atlfifo_concat.v
<IP variation name>_rxdrp_atlfifo_concat.v

To include these files in the project, click Assignments \(\rightarrow\) Settings, select the generated files, and click Add All.

The required files for the design example are highlighted in yellow below. Note: you should not include the generated .qip file.

**Figure 5. SerialLite II Design Files**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Type</th>
<th>Library</th>
<th>Design Entry/Synth</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk_definition.sdc</td>
<td>Synopsys Design Constraints File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>timing.sdc</td>
<td>Synopsys Design Constraints File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>controller.controller.qip</td>
<td>IP Variation File (.qip)</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/sl2_core.sd</td>
<td>Synopsys Design Constraints File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/dl2_buffer.qsys</td>
<td>Qsys System File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/counter_syncro.vhd</td>
<td>VHDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/reset_syncro.vhd</td>
<td>VHDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/asyncrx.rxsys</td>
<td>Qsys System File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/asyncrx_reset.rxsys</td>
<td>Qsys System File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/asyncrx_pll.rxsys</td>
<td>Qsys System File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/asyncrx_pll_pll.rxsys</td>
<td>Qsys System File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/sl2_core_brdp_atlfifo_concat.v</td>
<td>Verilog HDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/sl2_core_slit_e_unzip.v</td>
<td>Verilog HDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/sl2_core_slit_e_kern.v</td>
<td>Verilog HDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/sl2_core_slit_e_top.v</td>
<td>Verilog HDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/sl2_core_rxd rp_atlfifo_concat.v</td>
<td>Verilog HDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/sl2_core.vhd</td>
<td>VHDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/Seriallite_Demo.vhd</td>
<td>VHDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/CountVerify.vhd</td>
<td>VHDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
<tr>
<td>core/Difference.vhd</td>
<td>VHDL File</td>
<td>&lt;None&gt;</td>
<td></td>
</tr>
</tbody>
</table>

To view the SerialLite II MegaCore parameter setting for the design example, click File \(\rightarrow\) Open, select core/sl2_core.vhd. MegaWizard Plug-In Manager opens the SerialLite II MegaCore parameter editor.

The following shows the SerialLite II MegaCore parameter settings.
Stratix V device family is selected

Data rate in the SerialLite II MegaCore doesn’t really matter. Actual data rate is dictated by the transceiver PHY and TX PLL instantiation. Select transfer size 4 for data rates greater than 3.125Gbps

Duplex 4 lane configuration

Design example uses data packets.

For more information about SerialLite II MegaCore, refer to the user guide.

- SerialLite II MegaCore User Guide
**Transceiver Native PHY**

Transceiver PHY is no longer part of the SerialLite II MegaCore and must be instantiated separately. For the SerialLite II link, an Arria 10 Native PHY is instantiated. Configure the Native PHY based on the link bandwidth requirement.

To view the Native PHY parameter setting used for this design example: click File → Open, select core/xcvr_txrx.qsys.

The Arria 10 Transceiver parameter settings for the design example are shown below.

The datapath options enable simplified data interface and select duplex 4 lanes at 10Gbps per lane.

**Figure 8. Native PHY Datapath Parameters**

![Native PHY Datapath Parameters](image)

- Duplex, 4 lane, 10 Gbps data rate per lane
TX channels in the design example uses non-bonding mode. An external TX PLL must be instantiated to generate TX serial clock.

**Figure 9. Native PHY TX PMA Parameters**

- **TX PLL Options**
  - TX local clock division factor: 1
  - Number of TX PLL clock inputs per channel: 1
  - Initial TX PLL clock input selection: 0

- **Note**: The external TX PLL IP must be configured with an output clock frequency of 5000.0 MHz.

- **Design example allows internal serial loopback.**

- **External TX PLL must be instantiated to generate TX serial clock. TX PLL output clock frequency should be set to data rate / 2.**
### Figure 10. Native PHY RX PMA Options

<table>
<thead>
<tr>
<th>Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems</td>
<td>xcvr_tune</td>
</tr>
<tr>
<td>Path</td>
<td>xcvr_tune,inst</td>
</tr>
</tbody>
</table>

#### Arriva 10 Transceiver Native PHY  
altera_xcvr-native_a10

<table>
<thead>
<tr>
<th>TX PMA</th>
<th>RX PMA</th>
<th>Standard PCS</th>
<th>Dynamic Reconfiguration</th>
<th>Generation Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX CDR Options</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of CDR reference clocks</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected CDR reference clock</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected CDR reference clock frequency</td>
<td>625.000000 MHz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPM detector threshold</td>
<td>1000 PPM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Equalization

<table>
<thead>
<tr>
<th>CTLE adaptation mode</th>
<th>DFE adaptation mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>manual</td>
<td>manual</td>
</tr>
</tbody>
</table>

#### RX PMA Optional Ports

- [ ] Enable rx_pma_clkout port
- [ ] Enable rx_pma_div_clkout port
- [ ] Enable rx_pma_divclkout division factor: Disabled
- [ ] Enable rx_pma_lqtnr_clkout port
- [ ] Enable rx_pma_clkstop port
- [ ] Enable rx_pma_qpipuldn port (GPIO)
- [ ] Enable rx_lis_lockedthidata port
- [ ] Enable rx_lis_lockedthoref port
- [ ] Enable rx_set_lockcheckdata and rx_set_lockcheckref ports
- [ ] Enable rx_realmbroken port
- [ ] Enable PRBS verifier control and status ports

---

- On-board oscillator provides 625MHz reference clock.
- Design example allows you to reconfigure receiver CTLE and DFE settings dynamically.
- Design example monitors CDR lock status and allows internal serial loopback.
**Figure 11. Native PHY Standard PCS Parameters**

<table>
<thead>
<tr>
<th>TX PMA</th>
<th>RX PMA</th>
<th>Standard PCS</th>
<th>Dynamic Reconfiguration</th>
<th>Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>20</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FPGA fabric / Standard TX PCS interface width: **32**

FPGA fabric / Standard RX PCS interface width: **32**

- Enable 'Standard PCS' low latency mode

**Standard PCS FIFO**

- TX FIFO mode: low_latency
- RX FIFO mode: low_latency

- Enable tx_std_pcmfifo_full_port
- Enable tx_std_pcmfifo_empty_port
- Enable rx_std_pcmfifo_full_port
- Enable rx_std_pcmfifo_empty_port

**Byte Serializer and Deserializer**

- TX byte serializer mode: Serialize x2
- RX byte deserializer mode: Deserialize x2

**8b/10b Encoder and Decoder**

- Enable TX 8b/10b encoder
- Enable TX 8b/10b disparity control
- Enable RX 8b/10b decoder

**Description**

- Use 20-bit PMA interface width and 32-bit fabric/PCS interface width.
- Byte serializer and deserializer are enabled.
- PCS 8b/10b encoder and decoder are used.
The transceiver PHY PMA analog settings can be dynamically reconfigured for optimal signal performance.

**Figure 12. Native PHY Dynamic Reconfiguration Parameters**

A single Avalon-MM slave interface for reconfiguring Native PHY channels is enabled. Using Altera Debug Master Endpoint (ADME) requires the Native PHY channels to share a single Avalon-MM reconfiguration interface.

ADME can access reconfiguration register space of the transceiver. It can perform certain test and debug functions via JTAG using System Console. Transceiver Toolkit support is available by enabling ADME.

For more information about the Arria 10 Transceiver PHY, refer to the user guide below.

- **Arria 10 Transceiver PHY User Guide**
**TX PLL**

An Arria 10 design must use an external TX PLL (ATX PLL, fPLL, or CMU PLL). The TX PLL generates the TX serial clock for the transceiver PHY channels. In the design example, an ATX PLL is used.

To view the ATX PLL parameter setting for the design example: click File → Open, select `core/xcvr_pll.qsys`.

The ATX PLL parameter settings for the design example are as follows.

**Figure 13. ATX PLL Parameters**

- ATX PLL output frequency is set to data rate / 2.
- On-board oscillator provides 625MHz reference clock.
If SerialLite II link requires multiple lanes across the transceiver banks, you must turn on **Include Master Clock Generation Block** and **Enable x6/xN non-bonded high-speed clock output port** as shown below. In case you need multiple SerialLite II MegaCore instances using the same data rate, instantiate just single TX PLL.

Note: In the design example, the MCGB and x6/xN clock network options are not turned on because ATX PLL drives 4 lanes within a single transceiver bank.

**Figure 14. ATX PLL Master Clock Generation Block Parameters**

SerialLite II link using lanes across the transceiver banks must turn on MCGB and enable x6/xN non-bonded serial clock output, which allows to drive up to 30 transceiver transmit channels, two banks up and two banks down.
fPLL (fractional PLL)
The design example instantiates an fPLL to generate 260MHz internal core clock. The core clock synchronizes the SerialLite II MegaCore, packet generator, and packet monitor.

To view the IP parameter setting used in the design example: click File → Open, select core/pll_data.qsys.

The fPLL parameter settings are as follows.

Figure 15. Core fPLL Parameters (1/2)

The Arria 10 fPLL supports three operation modes: Core, Transceiver, and Cascade source. In the design example, Core mode is selected since it drives core logic in the FPGA fabric.

Note: A clock control block (ALTCLKCTRL) MegaFunction needs to be instantiated to direct Quartus to use non-HSSI dedicated input clock as a fPLL reference clock (on-board 100MHz input).

On-board oscillator provides 100MHz reference clock.

fPLL generates 260MHz internal core clock.
Transceiver Reset Controller

The design example uses the Transceiver Reset Controller to control the transceiver channel reset sequencing.

To view the IP parameter setting for the design example: click File → Open, select core/xcvr_reset.qsys.

The following is the IP parameter settings for the design example.

Figure 17. Transceiver Reset Controller Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong>: xcvr_reset</td>
<td><strong>Path</strong>: xcvr_reset_rx_inst</td>
</tr>
<tr>
<td><strong>Transceiver PHY Reset Controller</strong></td>
<td>altera_xcvr_reset_control</td>
</tr>
</tbody>
</table>

**General Options**

- Number of transceiver channels: 4
- Number of TX PLLs: 1
- Input clock frequency: 100 MHz
- Synchronize reset input
- Use fast reset for simulation
- Separate interface per channel/PLL

**TX PLL**

- Enable TX PLL reset control
- pll_powerdown duration: 1000 ns
- Synchronize reset input for PLL powerdown

**TX Channel**

- Enable TX channel reset control
- Use separate TX reset per channel
- TX digital reset mode: AUTO
- tx_digitareset duration: 20 ns
- pll_locked input hysteresis: 80 ns
- Enable pll_cal_busy input port

**RX Channel**

- Enable RX channel reset control
- Use separate RX reset per channel
- RX digital reset mode: AUTO
- rx_analogreset duration: 40 ns
- rx_digitareset duration: 4000 ns

To filter out glitches on pll_locked input, the hysteresis option must be used with a non-zero value.
Controller
Qsys system in the design example includes the Nios II processor, control and status registers, and reconfiguration interfaces for ATX PLL and transceiver PHY. The Nios II controller enables various control function and status monitoring.

- Control function
  - Start/stop the test
  - Toggle packet generation on/off
  - Enable/disable serial loopback for any lanes (by default, serial loopback is disabled)
  - Change the packet length and inter-packet gap dynamically
  - Insert bit errors in the generated data
  - Reset error counters
  - Change the BER measurement interval (default time interval is 1 second)
  - Reconfigure Tx output voltage swing (VOD) and pre-emphasis, Rx equalization settings

- Status monitoring
  - Freq_Locked, PLL_locked, Link Up, Datalocked
  - Packet_Sent, Packet_Received
  - Tx and Rx user data rate and efficiency
  - Bit errors, packet errors, and the BER
  - Time the test has been running
  - Reference clock frequency

To view the Nios II controller in Qsys system, click File → Open, select controller.qsys.
Reconfig Management Interface

The transceiver Native PHY and ATX PLL have an Avalon-MM interface which allows full control of the reconfiguration register space. The Qsys system instantiates two `reconfig_mgmt` components to provide access to the reconfiguration register space: `reconfig_mgmt lc` for ATX PLL and `reconfig_mgmt` for Native PHY. The `reconfig_mgmt` component is defined by `reconfig_mgmt_hw.tcl` file in the project folder.

**Figure 18. reconfig_mgmt Qsys Component**
Simulating the Design Example

The design example provides a simulation testbench (SerialLite Demo_tb.vhd) and Modelsim macro (.do) files in the simulation folder. To run simulation in Modelsim SE (or PE) or Questsim:

1. Open a Nios II Command Shell using Quartus 15.0 or later
2. Navigate to the simulation folder
3. Run ./create_simscript.bsh command. (This creates necessary simulation scripts that match your environment)
4. Compile and simulate using the following command line:
   a. vsim -c -do sim.do when there are no libraries yet. This compiles and simulates everything.
   b. vsim -c -do resim.do when libraries are already created. This re-compiles only the design files and simulates the testbench.

Below is an example output displayed during simulation.

# ** Note: ==================================> Link Up ...
# Time: 27605 ns  Iteration: 0  Instance: /seriallite_demo_tb/DUT
# ** Note: ==================================> CountLock achieved ...
# Time: 29349755 ps  Iteration: 5  Instance: /seriallite_demo_tb/DUT/PacketVerify_inst/Generate_CountVerify(3)/CountVerify_Ch
# ** Note: ==================================> CountLock achieved ...
# Time: 29349755 ps  Iteration: 5  Instance: /seriallite_demo_tb/DUT/PacketVerify_inst/Generate_CountVerify(0)/CountVerify_Ch
# ** Note: ==================================> CountLock achieved ...
# Time: 29349755 ps  Iteration: 5  Instance: /seriallite_demo_tb/DUT/PacketVerify_inst/Generate_CountVerify(1)/CountVerify_Ch
# ** Note: ==================================> CountLock achieved ...
# Time: 29349755 ps  Iteration: 5  Instance: /seriallite_demo_tb/DUT/PacketVerify_inst/Generate_CountVerify(2)/CountVerify_Ch
# ** Note: ==================================> DataLock achieved ...
# Time: 29353610 ps  Iteration: 5  Instance: /seriallite_demo_tb/DUT/PacketVerify_inst
# ** Note: ==================================> Inserted one bit error in Tx data ...
# Time: 53018985 ps  Iteration: 5  Instance: /seriallite_demo_tb/DUT/PacketVerify_inst/Generate_CountVerify(0)/CountVerify_Ch
# ** Note: ==================================> Found a data mis-match in last received word
# Time: 53418985 ps  Iteration: 5  Instance: /seriallite_demo_tb/DUT/PacketVerify_inst/Generate_CountVerify(0)/CountVerify_Ch
The following is the simulation waveform showing the SerialLite II MegaCore transmitter and receiver signals.

**Figure 19. Design Example Simulation Waveform**
Demo Walkthrough

Design Files
The design example files are organized in two folders:

- SerialLite_II_4_lanes_10Gbps_Arria10_Devkit
- Programming_Files

The Quartus project, SerialLite_II_4_lanes_10Gbps_Arria10_Devkit, includes sub-folders for the IP cores, Nios II controller, software, simulation, and scripts.

The Programming_Files has the .sof and .elf files for out-of-box experience. For demo purpose, you can download these files without building the hardware and software.

After unzip, the demo file organization will look like this. Note: Do not include any spaces in the path name.

Figure 20. Design Example File Organization

The design example has the timing errors with respect to REFCLK_FMCA_P and altera_reserved_tck signals. These errors can be safely ignored.
Hardware Setup
The demo requires the following hardware:

- Arria 10 FPGA Development Kit (ES2 silicon)
- USB-Blaster II download cable
- Power adapter for Arria 10 Development Board
- FMC loopback adapter board (optional)

Highlighted below are the connector locations for FMCA (J1), USB-Blaster II cable (J3), and power adapter cable (J13):

1. Plug USB-Blaster II cable to J3
2. Plug Arria 10 board power adapter to J13
3. Connect FMC loopback adapter board to FMCA (J1) connector in order to have external loopback. Alternatively, the demo can use internal serial loopback
4. On-board oscillator provides the 625MHz transceiver reference clock. There is no need to program the oscillator output frequency

Figure 21. Arria 10 FPGA Development Kit

More information about the Arria 10 FPGA Development Kit is available below.

- Arria 10 GX FPGA Development Kit
Software Setup
The Quartus project includes software folder with bsp and app sub-folders as shown below:

Figure 22. Design Example Software Folder

The ELF file is available in the Programming Files folder. There is no need to build the application software for the demo. In case you need to rebuild the application software using Quartus 15.0 or later, use the following steps.

1. Open a Nios II Command Shell using Quartus 15.0 or later. You can type nios ii in Windows program search box and click a Nios II 15.0 Command Shell in the program list. Or click Nios II 15.0 Command Shell under Quartus installation directory as shown below.

Figure 23. Nios II Command Shell Location

2. Build the application software
   a. Navigate to software/app/Arria10GX_4Ch directory
   b. Type ./create_this_app

   This will compile the system libraries in the bsp folder, compile the software code (controller.c in the design example), and generate the Nios II executable, .elf file.
Downloading the SOF and ELF files

1. Open a **Nios II Command Shell** using Quartus 15.0 or later

2. Navigate to the **Programming Files** folder where the demo SOF and ELF files are stored

3. Download the `.sof` file using `nios2-configure-sof` command

   **Figure 24. Downloading SOF File**
   
   ```
   $ nios2-configure-sof Devkit_Demo.sof
   Searching for SOF file:
   in:
     Devkit_Demo.sof
   
   Info: #############################################################################
   Info: Running Quartus II 64-Bit Programmer
   Info: Command: quartus Programmer --no_halt --no_do --print config log
   Info: (22:30:51) Using programming cable "USB-BlasterII [USB-1]
   Info: (22:30:51) Using programming file Devkit_Demo.sof with checksum 0x089D6363 for device 10M115s-4F5iJ3gY201
   Info: (22:30:52) Started Programmer operation at Thu Oct 15 17:30:52 2015
   Info: (22:30:52) Configuring device index 1
   Info: (22:30:52) Device 1 contains JTAG ID code 0x0065ACD
   Info: (22:30:52) Configuration succeeded -- 1 device(s) configured
   Info: (22:30:52) Successfully performed operation(s)
   Info: Quartus II 64-Bit Programmer was successful. 0 errors, 0 warnings
   Info: Peak virtual memory: 1466 megabytes
   Info: Elapsed time: 00:01:2
   Info: Total CPU time (on all processors): 00:00:14
   
   4. Download the `.elf` file using `nios2-download -g -r` command

   **Figure 25. Downloading ELF File**
   
   ```
   $ nios2-download -g -r Arria10GX_4Ch.elf
   Using cable "USB-BlasterII [USB-1]", device 1, instance 0x00
   Resetting and pausing target processor: OK
   Initializing CPU cache (if present)
   OK
   Downloaded 83KB in 0.1s
   Verified OK
   Starting processor at address 0x0004018C
   ```
Running the Demo

1. Open a separate Nios II Command Shell using Quartus 14.0 or earlier.

   Note: Quartus 15.0 Windows version has a bug that prevents a Nios II terminal from accepting keyboard input.

   ![Figure 26. Nios II Command Shell Location](image)

2. In the Nios II Command Shell, open a Nios II terminal using `nios2-terminal` command as shown below.

   ![Figure 27. Opening a Nios II Terminal](image)

3. The Nios II terminal displays various status and control functions. The control function provides many actions. For example, you can select 7 (Show Status) to view the Status. (Note: Keyboard input is not displayed on the screen. Just hit Return after you type a number)
3. To view the transceiver PMA settings, select E. Show Transceiver PMA Settings on all channels.
4. To reconfigure the transceiver PMA setting, select **4. Show/Control Transceiver PMA Settings on Links**.

**Figure 30. Display/Control Transceiver PMA Settings**

```
        Enter Choice : 4

4. Transceiver PMA Settings of Channel 0
VOD Level        : 29
Pre Tap 2 Level  : 0
Pre Tap 1 Level  : 0
Post Tap 1 Level : 0
Post Tap 2 Level : 0
CILE DC Gain     : 0
CILE AC Gain     : 12
UGA Gain         : 4

New VOD Level Channel 0
==================================
Provide number from 12-31 :
```
## Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 2015</td>
<td>Initial Release</td>
<td></td>
</tr>
</tbody>
</table>