Dynamic Reconfiguration of PMA Controls in Stratix V Devices

AN-645-1.0 Application Note

This application note describes how to use the transceiver reconfiguration controller to dynamically reconfigure the Physical Media Attachment (PMA) controls of the Stratix® V transceivers.

You can reconfigure the following PMA controls to optimize signal integrity of all the high-speed links in your Stratix V device:

- Transmitter pre-emphasis
- Differential output voltage (V_{OD})
- RX linear equalization
- RX buffer DC gain

The attached reference design provides an implementation example of PMA controls reconfiguration. You can modify the reference design to suit your system requirements and perform an in-system debug of the high-speed channels in your design. This application note describes how to simulate this reference design using the Quartus® II software version 11.0 and ModelSim® version 6.6d.

For more information about the transceiver reconfiguration controller architecture and implementation, refer to the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

Glossary

Controller—the transceiver reconfiguration controller

PHY IP—the transceiver PHY IP (a Custom PHY IP in this application)

PHY*n*—the number of PHY IP instances connected to a controller. For this reference design, *n*=1.

IFn—the number of reconfiguration interfaces you need to enter in the controller

CHn—the number of transceiver channels in the Custom PHY IP

Reconfiguration interface—the `reconfig_from_xcvr` and `reconfig_to_xcvr` bus connection between the controller and transceiver PHY IP

Reconfiguration Interface Group—the number of groups you want to split the reconfiguration interface bus into at the controller side. It is represented as Grp1, Grp2, …, Grpn in this application note, where *n* refers to the number of groups and Grp itself refers to the width of each group.

101 Innovation Drive
San Jose, CA 95134
www.altera.com

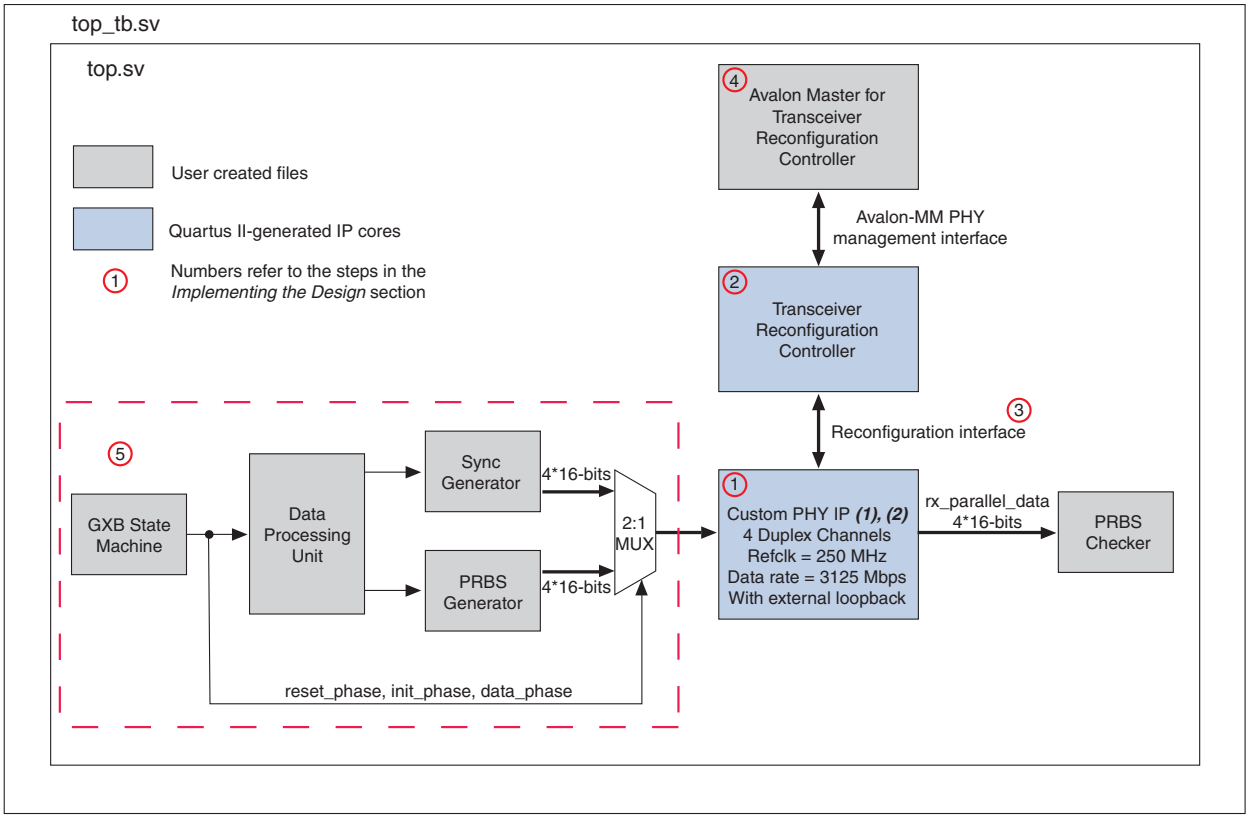
Reference Design Functional Description

This reference design dynamically reconfigures transmitter V_{OD}, pre-emphasis, receiver DC gain, and equalization for a Stratix V transceiver link implemented using a Custom PHY IP. The purpose of this design is to show you how to connect the transceiver reconfiguration controller to the PHY IP and reconfigure the PMA controls through the Avalon® memory-mapped (Avalon-MM) interface of the controller.

This example creates four 3125 transceiver links using a Custom PHY IP, an example top-level file (**top.sv**), and a test bench (**top_tb.sv**). These will be used to demonstrate the PMA controls reconfiguration feature in Stratix V devices.

Figure 1 shows a high-level block diagram of the reference design that is described in this application note.

Figure 1. Block Diagram of the Reference Design *(Note 1), (2)*



Notes to Figure 1:

(1) You need an Avalon-MM Master to communicate with the PHY management interface of the Custom PHY IP core. It is not shown in this design because it is beyond the scope of this application note.

(2) You can modify the reference design so that a single Avalon-MM Master is used to interface with both the reconfiguration management interface and the PHY management interface or you can use a separate Avalon-MM Master for each.

The modules of the block diagram include the following:

- **Custom PHY IP**— The Custom PHY IP instantiates the Stratix V transceivers in custom configurations that use standard PCS and word aligner. This file is called **top_custom.v** in the reference design.

For implementation details, refer to the *Custom PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

- **Transceiver Reconfiguration Controller**—This file is called **top_reconfig.v** in the reference design.

  For implementation details, refer to the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

- **Avalon Master**—This file is called **mgmt_master.sv** and contains the following components:
  - **mgmt_master.sv**—the top-level wrapper that initializes ROM
  - **mgmt_master_cpu.sv**—the file containing the actual controller state machine
  - **mgmt_commands_hv.sv**—the package file that contains low-level parameters and command functions
  - **mgmt_functions_h.sv**—the package file that defines high-level functions called by the user program
  - **mgmt_program.sv**—the package file that contains the user's program
- **User Created Logic**—The following blocks are created by the user in the reference design:
  - **GXB State Machine**—The GXB state machine controls the sequence of events in the reference design. It contains counters that monitor the three phases: reset phase, init phase, and data phase. Each phase must be completed before the next phase can begin. The state machine sends a control signal to the 2:1 MUX that identifies which phase is currently active for the Custom PHY IP.
  - **Data Processing Unit**—The data processing unit sends control signals to the sync generator and the PRBS generator. It selects the data from the sync generator during the init phase, and the data from the PRBS generator during the data phase. The data processing unit is called **datagen_controller.v** in the design.
  - **Sync Generator**—The sync generator begins operating in the init phase, after the `tx_ready` and `rx_ready` output signals of the Custom PHY IP are asserted. It transmits 8'hBCs into the four transmitter lanes of the Custom PHY IP. This ensures that the four receiver links are word aligned.

    The sync generator is called **syncgen_16bit_1lane.v** in the reference design. The **syncgen_16bit_4lanes.v** instantiates the sync generator four times for the four transceiver lanes.
  - **PRBS Generator**—The PRBS generator begins operating in the data phase, after the `rx_syncstatus` output signal of the Custom PHY IP is asserted. It generates the payload (PRBS23 data) for the Custom PHY IP. It is called **prbs_generator.v** in the reference design.
  - **PRBS Checker**—The PRBS checker is called **prbs_checker.v** in the design. The **prbs_checker_x4.v** file instantiates the checker four times for the four receiver lanes. It verifies that the data transmitted into the transmitter is received correctly at the output of the receiver.

Design Specifications

Table 1 lists the specifications for this reference design.

Table 1. Design Specifications

Attribute	Specification
Quartus II version	Quartus II v11.0
FPGA	5SGXMA7K2F40C2
IP cores used	■ Custom PHY v11.0 ■ Transceiver Reconfiguration Controller v11.0
Prerequisites	Transceiver Reconfiguration Controller and Custom PHY IP implementation details
Simulator	ModelSim v6.6d
Data Pattern	PRBS23

Implementing the Design

Perform the following steps to create the reference design.

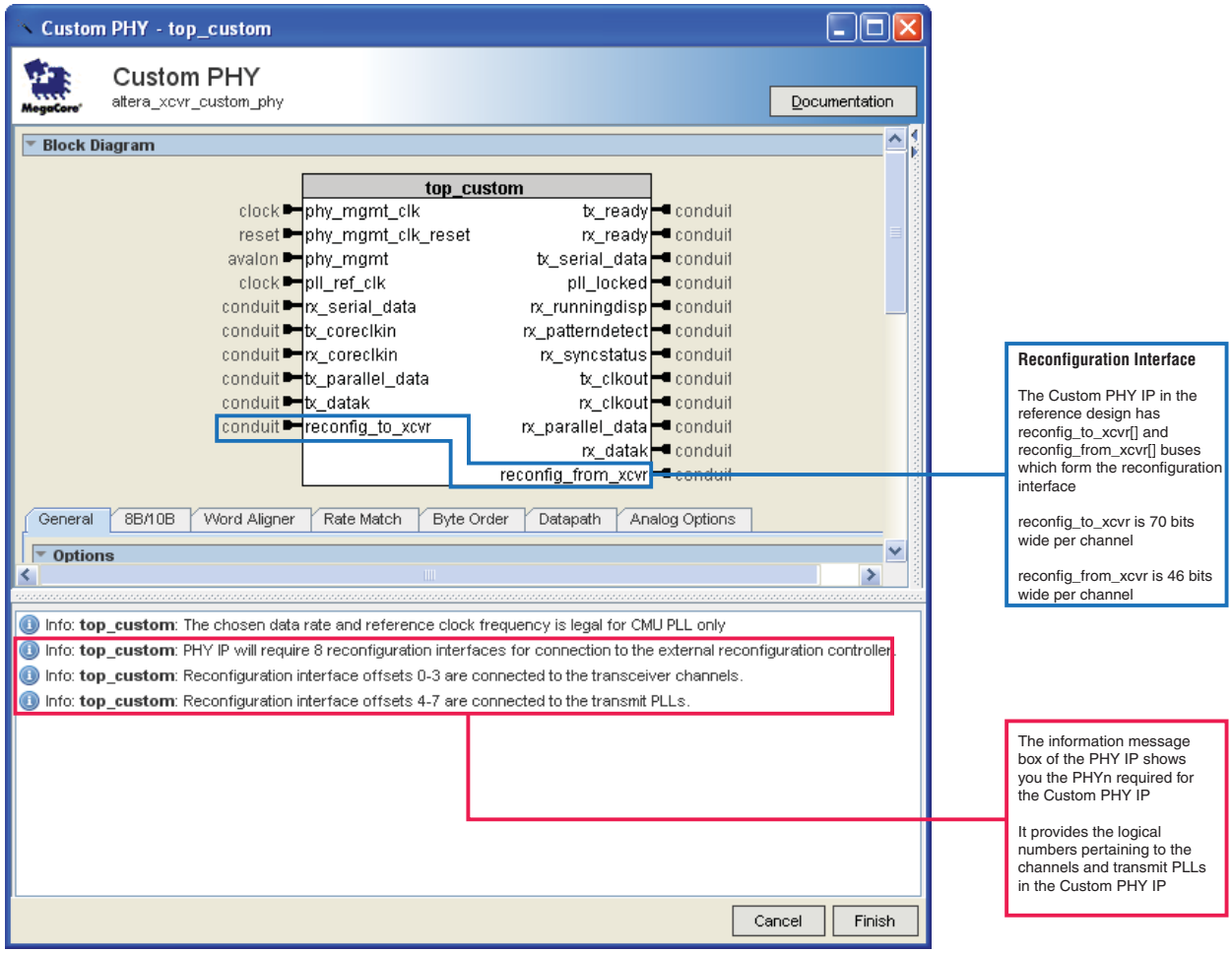
Step 1: Set Up the Transceiver Channels in the Custom PHY IP

Use the MegaWizard™ Plug-In Manager to instantiate the Custom PHY IP.

To open the MegaWizard Plug-In Manager, click **Installed Plug-Ins** > **Interfaces** > **Transceiver PHY** > **Custom Phy IP v11.0**.

For this reference design, the Custom PHY IP is set up as shown in Figure 2.

Figure 2. Setting Up the Custom PHY IP



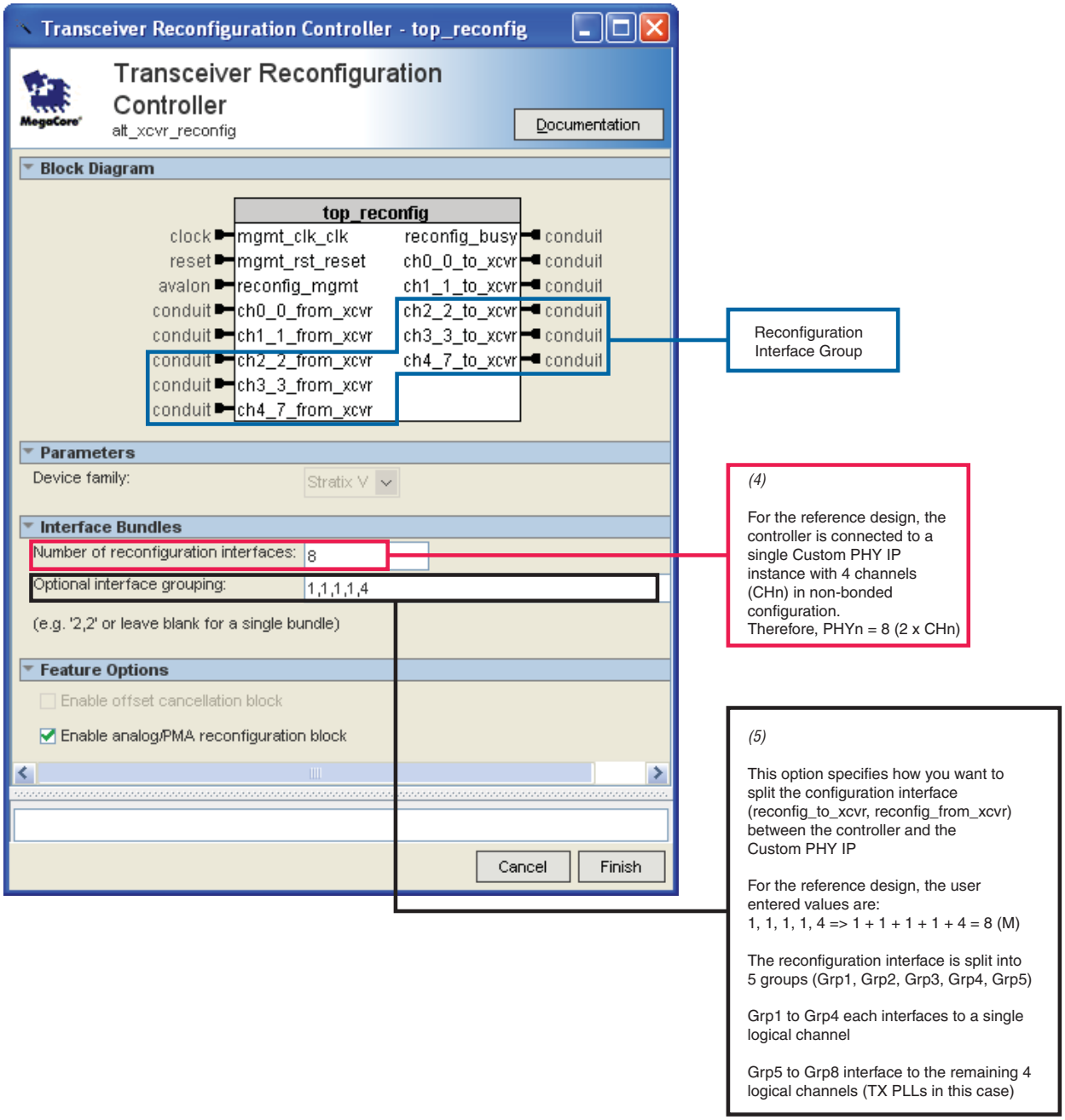
Step 2: Set Up the Transceiver Reconfiguration Controller

Use the MegaWizard Plug-In Manager to instantiate the controller.

To open the MegaWizard Plug-In Manager, click **Installed Plug-Ins** > **Interfaces** > **Transceiver PHY** > **Transceiver Reconfiguration Controller**.

For the reference design, set up the controller as shown in Figure 3.

Figure 3. Setting Up the Transceiver Reconfiguration Controller ***(Note 1)***, ***(2)***, ***(3)***



Notes to Figure 3:

(1) `mgmt_clk_clk` is connected to 150 MHz in the reference design.

(2) You can reconfigure PMA controls only when `reconfig_busy` is asserted low.

(3) Connect an Avalon-MM Master to the reconfig_mgmt bus of the controller as shown in the design to schedule your PMA control reads and writes to the Custom PHY IP.

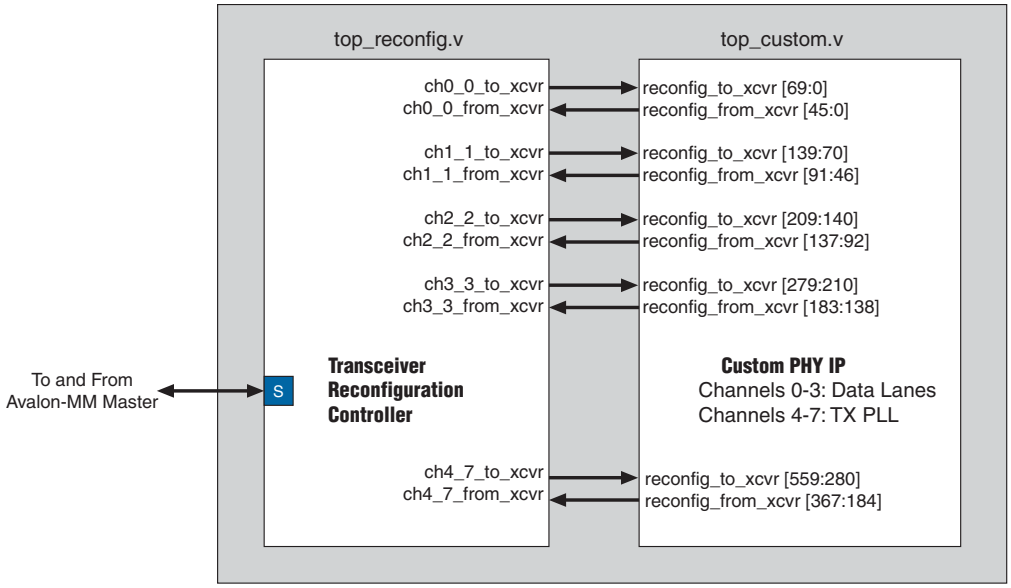
(4) Refer to Table 9-1 in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide* to determine the logical number of reconfiguration interfaces (IFn).

(5) Refer to Table 9-4 in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide* for details on this option.

Step 3: Connect the Controller and Transceiver Channels

For the reference design, connect the controller and transceiver channels as shown in Figure 4.

Figure 4. Connect the Controller and Transceiver Channels



Step 4: Create an Avalon Memory-Mapped Master

Use the Avalon-MM master to communicate with the controller's Avalon reconfiguration management interface.

Use one of the following options to create a synthesizable Avalon-MM master:

- Write your own state machine to interact with the Avalon-MM interface
- Use the NIOS® II processor from the Qsys system to send reads and writes to the Avalon-MM interface
- Use your own external processor and create a bridge to the Avalon-MM interface

The reference design gives an example of a synthesizable Avalon-MM Master. The only file that you must change in this design is **master_program.sv**.

Regardless of which option you choose to create the master, you must follow the Avalon specifications described in the *Avalon Interface Specifications*.

For example, suppose you want to change the TX V_{OD} of logical channel 2 to a setting of 5. Change the `Write VOD` section of **mgmt_program.sv** as shown in Example 1.

Example 1.

```
// Write VOD
      $display ("Change VOD to a setting of 5 for logical channel 2\n");
      f_write_reg(LOCAL_ADDR_XR_ANALOG_LCH, 2);    // Write logical ch address, 2
      f_write_reg(LOCAL_ADDR_XR_ANALOG_OFFSET, 0); // Write offset for VOD
      f_write_reg(LOCAL_ADDR_XR_ANALOG_DATA,5);    // write vod data (5) in the data reg
(which is at address 4)
      f_write_reg(LOCAL_ADDR_XR_ANALOG_STATUS, 1); // enable wr bit in control register,
start write operation by writing 1 in control register = address 2

f_sleep(50);
```

For the actual values that the V_{OD} setting maps to, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter of the *Stratix V Device Handbook*.

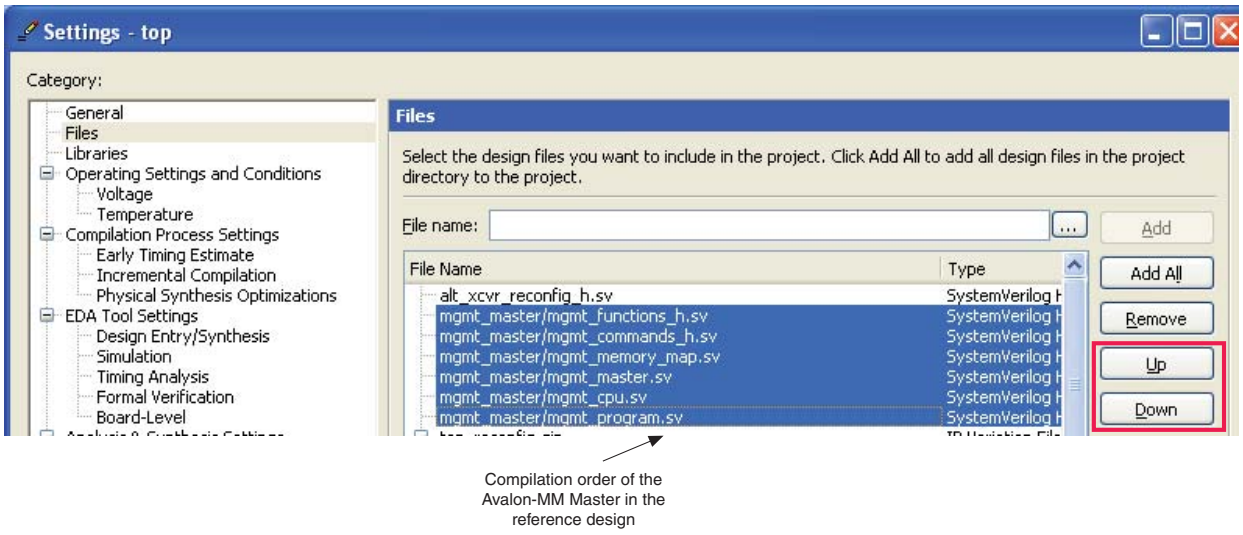
Step 5: Connect the User-Created Logic

Connect the remaining blocks and compile the design. During compilation, ensure that you compile the **.sv** packages and header files before you compile the remaining RTL files.

To rearrange the order of the design files, perform the following steps:

1. Click the **Add/Remove Files in Project** tab of the **Project** tab.
2. Click on the file whose position you want to change.
3. Click the **Up** or **Down** buttons shown in Figure 5 to change the order. Arrange the files in the order shown in Figure 5.

Figure 5. Arranging the Files



Step 6: Simulate the Design

This section describes how to simulate the reference design using ModelSim SE version 6.6d.

The files required for simulation are:

- **phy_sim_setup.tcl**—the main Tcl script used to generate a ModelSim simulation script. The output file is **phy_sim_top.tcl**.
- **phy_dirs.txt**—the file containing all PHY IP instance names and working directory
- **phy_sim_top.tcl**—the output ModelSim script generated from **phy_sim_setup.tcl**. The file can be used in ModelSim to create all the necessary PHY IP libraries for simulation.

Perform the following steps to simulate the design:

1. Copy **phy_sim_setup.tcl** to the main source directory.
2. Create **phy_dirs.txt** with a text editor and add the necessary information to the file.

   This file must be formatted correctly for the script to run correctly. The contents of the file are as follows:

   - <working directory>
   - <reconfig instance>
   - <PHY IP instance #1>
   - <PHY IP instance #2>
   - <PHY IP instance #n>

   The script assumes that the PHY instance files are at the same level as the <working directory>.
3. Run the script: `quartus_sh -t phy_sim_setup.tcl`
4. Open the newly created **phy_sim_top.tcl** file
5. Add all necessary design files, including the testbench file
6. Launch ModelSim and source **phy_sim_top.tcl**

Document Revision History

Table 2 lists the revision history for this application note.

Table 2. Document Revision History

Date	Version	Changes
May 2011	1.0	Initial release.