

This chapter discusses HardCopy<sup>®</sup> migration guidelines for UniPHY-based designs. If you want to migrate your ALTMEMPHY-based designs to HardCopy, Altera recommends that you first upgrade your design to UniPHY.

-  For information about upgrading an ALTMEMPHY-based design to UniPHY, refer to the *Upgrading to UniPHY-based Controllers from ALTMEMPHY-based Controllers* chapter in volume 3 of the *External Memory Interface Handbook*.


## HardCopy Migration Design Guidelines

If you intend to target your design to a HardCopy device, you should select both your prototyping FPGA device and target HardCopy device at the start of your project, to avoid any late difficulties that might occur due to differences in the UniPHY IP between FPGA and HardCopy implementations.

-  You must migrate your design from an FPGA to a HardCopy companion device; you cannot target HardCopy directly.


Ensure you use the following design guidelines when migrating your design:

- On the **PHY Settings** page of the parameter editor, turn on **HardCopy Compatibility Mode**, and then specify whether the **Reconfigurable PLL Location** is **Top\_Bottom** or **Left\_Right**.

-  Altera recommends that you set the **Reconfigurable PLL Location** to the same side as your memory interface.

When turned on, the **HardCopy Compatibility Mode** option enables a ROM loader and run-time reconfiguration for all phase-locked loops (PLLs) and delay-locked loops (DLLs) instantiated in memory interfaces. In this mode, all the necessary PLL and DLL reconfiguration and ROM loader signals are brought to the top level of the design.

- Enable run-time reconfiguration mode for all PLLs and DLLs instantiated in interfaces that are configured in PLL and DLL slaves.

-  For information about PLL megafunctions, refer to the *Phase-Locked Loop (ALTPLL) Megafunction User Guide* and the *Phase-Locked Loops Reconfiguration (ALTPLL\_RECONFIG) Megafunctions User Guide*. For information about DLL megafunctions, refer to the *ALTDLL and ALTDQ\_DQS Megafunctions User Guide*.

- Ensure that you place all memory interface pins close together. If, address pins are located far away from data pins, for example, closing timing might be difficult.
- For DDR2 and DDR3 (and RLDRAM II when using the Nios® II -based sequencer) UniPHY-based designs, ensure that you have external nonvolatile ROM or flash memory on your circuit board for storing the Nios II instruction code. (QDR II and QDR II+ SRAM with UniPHY designs support only the RTL-based sequencer in HardCopy migration.) The UniPHY IP instantiates a ROM loader to load Nios II instruction code from external ROM.

For ROM loader connection guidelines, refer to “ROM Loader for Designs Using Nios II Sequencer” on page 12-3.

- In the wraparound interface design, open the `<variation_name>_p0_new_io_pads.v` file in an editor and locate the following line:

```
.dll_offsetdelay_in((i < 0) ?  
hc_dll_config_dll_offset_ctrl_offsetctrlout :  
hc_dll_config_dll_offset_ctrl_offsetctrlout),
```

In the preceding line, first change the second `hc_dll_config_dll_offset_ctrl_offsetctrlout` to `hc_dll_config_dll_offset_ctrl_b_offsetctrlout`, and then change the numeral **0** to the number of DQS groups located in the top or bottom I/O edge. For example, changing 0 to 3 would mean that DQS groups 0 to 2 are connected to the output of the first DLL offset control block. DQS group 3 and above are connected to the output of the second DLL offset control block.

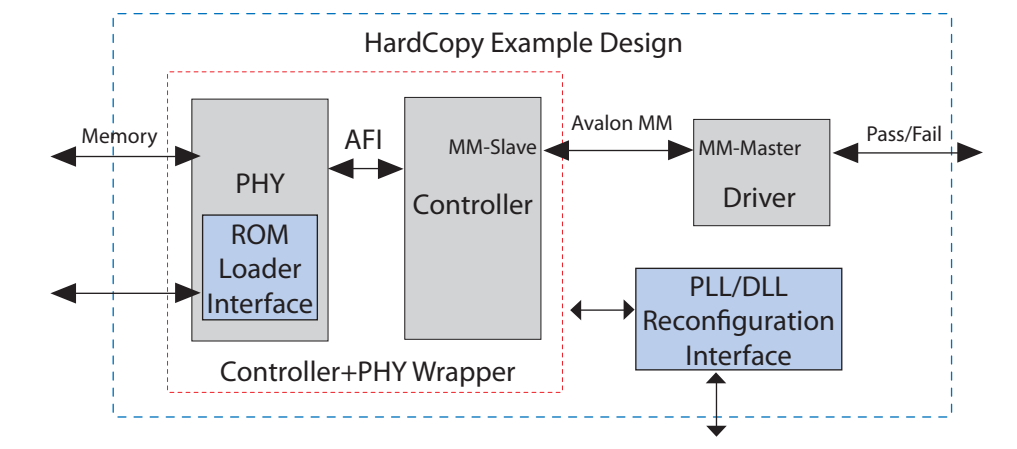
You can use the example top-level project that is generated when you turn on **HardCopy Migration** as a guide to help you connect the necessary signals in your design.

## Differences in UniPHY IP Generated with HardCopy Migration Support

When you generate a UniPHY memory interface for HardCopy device support, certain features in the IP are enabled that do not exist when you generate the IP core for only the FPGA. This section discusses those additional enabled features.

Figure 12-1 shows the additional blocks enabled when **HardCopy Compatibility Mode** is turned on.

**Figure 12-1. HardCopy UniPHY Example Design**



### ROM Loader for Designs Using Nios II Sequencer

An additional ROM loader is instantiated in the design for UniPHY designs that use the Nios II sequencer. The Nios II sequencer instruction code resides in RAM on either the HardCopy or FPGA device.

When you target only an FPGA device, the RAM is initialized when the device is programmed; however, HardCopy devices are not programmable and therefore the RAM cannot be initialized in this fashion. Instead, the Nios II sequencer instruction code must be stored in an external, non-volatile memory and must be loaded to the Nios II sequencer RAM through a ROM loader. You must create a subsystem to load the Nios II sequencer code from the external nonvolatile memory to the ROM loader.

The instruction code varies according to memory protocols and memory parameterization in UniPHY. You can share the same sequencer instruction code content for multiple interface designs if the memory protocol and settings are the same for each interface. You can also store different Nios II code in the same external, nonvolatile memory and use a single subsystem to load the Nios II codes to the corresponding Nios II sequencer RAMs.



For more information about the ROM loader, refer to the *RAM Initializer (ALTMEM\_INIT) Megafunction User Guide*.

Table 12-1 lists the ports exposed at the top level of the PHY+Controller wrapper to expose the ROM loader utilized by the Nios II-based sequencer within the RLDRAM II, DDR2, or DDR3 PHY.

**Table 12-1. Top-level Ports that Connect to External ROM for Loading Nios II Code Memory**

Port Name	Direction	Size	Description
hc_rom_config_clock	Input	1 bit	Write clock for the ROM loader. This clock is the write clock for the Nios II code memory.
hc_rom_config_datain	Input	32 bits	Data input from external ROM.
hc_rom_config_rom_data_ready	Input	1 bit	Asserts to the code memory loader that the word of memory is ready to be loaded.
hc_rom_config_init	Input	1 bit	Signals that the Nios II code memory is being loaded from the external ROM.
hc_rom_config_init_busy	Output	1 bit	Remains asserted throughout initialization and becomes inactive when initialization is complete. <code>soft_reset_n</code> can be issued after <code>hc_rom_config_init_busy</code> is deasserted.
hc_rom_config_rom_rden	Output	1 bit	Read-enable signal that connects to the external ROM.
hc_rom_config_rom_address	Output	12 bits	ROM address that connects to the external ROM.

You can load the Nios II instruction code in several ways. You can connect the ROM loader directly to the dedicated external, nonvolatile memory; alternatively, you may reuse the FPGA configuration interface with flash memory to load the Nios II instruction code using the ROM loader. The configuration flash memory is used to configure the FPGA design, but device configuration is not required in HardCopy designs, conserving resources and board space. You can reuse existing configuration pins and flash memory for interfacing with the ROM loader without any extra I/O pins or flash memory.

Three FPGA configuration schemes are available with flash memory: passive serial (PS) configuration, active serial (AS) configuration, and fast passive parallel (FPP) configuration. Only active serial (AS) and fast passive parallel (FPP) configuration interfaces are suitable for interfacing with the ROM loader when the device is in user mode.



For more information about FPGA configuration schemes, refer to the *Configuration Handbook*.

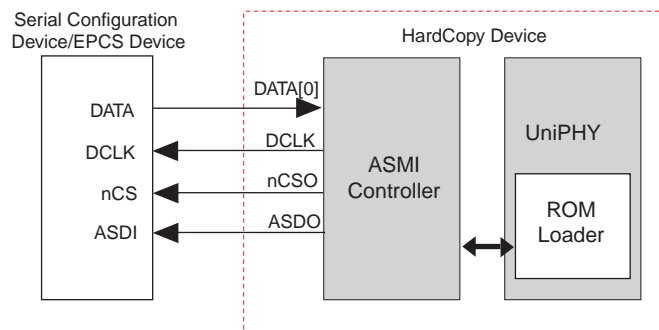
## Passive Serial (PS) Configuration Scheme

In the passive serial configuration scheme, CONF\_DONE, nSTATUS, nCE, nCONFIG, DATA[0], and DCLK are used for FPGA configuration. Only the DATA[0] signal is a dual-purpose configuration pin which is used as a normal I/O pin in user mode. Thus, only a single DATA[0] pin is available in the HardCopy device to use as an I/O pin. Interfacing with the ROM loader requires more than one I/O pin; therefore this configuration scheme cannot be used for interfacing with the ROM loader. Altera recommends using other configuration schemes in UniPHY-based designs that you intend to migrate to HardCopy devices.

## Active Serial (AS) Configuration Scheme

The active serial configuration scheme uses four configuration pins (DATA[0], DCLK, ASDO, and nCSO) to configure the FPGA. You can directly access the content of the flash memory through these four configuration pins in FPGA user mode, or in the HardCopy device using the active serial memory interface (ASMI) controller. You must add the ASMI controller to your HardCopy design to interface with the ROM loader. Figure 12-2 shows an example connection between a ROM loader with ASMI controller and EPCS flash memory.

**Figure 12-2. ROM Loader Connection in Active Serial Configuration Scheme**



For more information about the ASMI controller, refer to the *Active Serial Memory Interface (ALTASMI\_PARALLEL) Megafunction User Guide*.

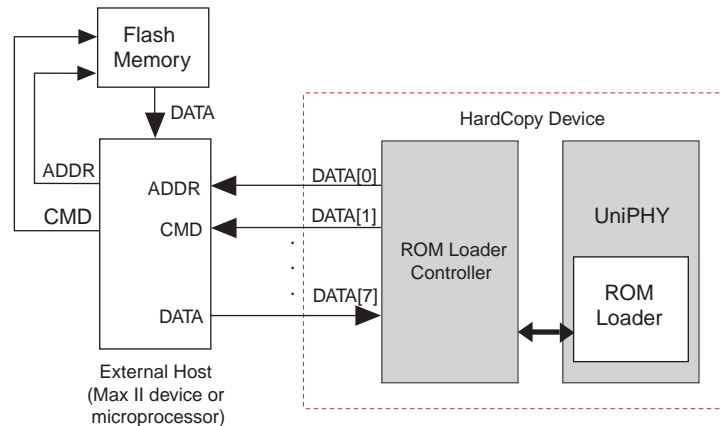
## Fast Passive Parallel (FPP) Configuration Scheme

In the fast passive parallel configuration scheme, only the DATA[0...7] configuration pins can be used as normal I/Os in a HardCopy device; eight DATA pins are available with this scheme.

The data pins can be used as clock, data, address, and command pins to interface with the external host for loading the flash memory content to Nios II through the ROM loader. You must ensure the flash memory data pins are connected to the external host or controller before connecting to the FPGA during FPGA configuration, in order to reuse these data pins.

FPGA configuration uses an external host which you must configure to interface with the ROM loader and to read the content of the flash memory. Due to limited FPP interface I/O count, you must create two controllers to serialize and deserialize the address (12 bits) and data (23 bits) signals of the ROM loader. One controller resides in the HardCopy device and one resides in the external host device. Figure 12-3 shows an example of a ROM loader configured in the FPP configuration scheme.

**Figure 12-3. ROM Loader Connection in Fast Passive Parallel Configuration Scheme**



## PLL/DLL Run-time Reconfiguration

The PLLs and DLLs in the HardCopy design have run-time reconfiguration enabled—provided that they are not in PLL/DLL slave mode.

When the PLLs and DLLs are generated with reconfiguration enabled, extra signals must be connected and driven by user logic. In the example design generated during IP core generation, the PLL/DLL reconfiguration signals are brought to the top level and connected to constants.

 For information about PLL megafunctions and reconfiguration, refer to the *Phase-Locked Loop (ALTPLL) Megafunction User Guide* and the *Phase-Locked Loops Reconfiguration (ALTPLL\_RECONFIG) Megafunctions User Guide*.

Table 12-2 lists the DLL reconfiguration ports exposed at the top level of the Controller and PHY wrapper.

**Table 12-2. DLL Reconfiguration Ports Exposed at Top-Level of Controller+PHY Wrapper (Part 1 of 2)**

Port Name	Direction	Size	Description
hc_dll_config_dll_offset_ctrl_addnsub <sup>(1)</sup>	Input	1 bit	Addition/subtraction control port for the DLL. This port controls whether the delay-offset setting on hc_dll_config_dll_offset_ctrl_offset is added or subtracted.
hc_dll_config_dll_offset_ctrl_offset <sup>(1)</sup>	Input	6 bits	Offset input setting for the PLL. This is a Gray-coded offset that is added or subtracted from the current value of the DLL's delay chain.
hc_dll_config_dll_offset_ctrl_offsetctrlout	Output <sup>(2)</sup>	6 bits	The registered and gray-coded value of the current delay-offset setting for the DLL offset control block that controls DQS pins on the top or bottom I/O edge.

**Table 12-2. DLL Reconfiguration Ports Exposed at Top-Level of Controller+PHY Wrapper (Part 2 of 2)**

Port Name	Direction	Size	Description
hc_dll_config_dll_offset_ctrl_b_offsetctrlout	Output <sup>(2)</sup>	6 bits	The registered and gray-coded value of the current delay offset setting for the DLL offset control block that controls DQS pins on left or right I/O edge.
<b>Note:</b>			
(1) Available only in DLL nonsharing mode and DLL master sharing mode.			
(2) Functions as an output in DLL nonsharing mode and DLL master sharing mode, and as an input in DLL slave mode.			


Table 12-3 lists the PLL reconfiguration ports exposed at the top level of the Controller and PHY wrapper.

**Table 12-3. PLL Reconfiguration Ports Exposed at the Top-Level of Controller+PHY Wrapper <sup>(1)</sup>**

Port Name	Direction	Size	Description
hc_pll_config_configupdate	Input	1 bit	Control signal to enable PLL reconfiguration. (Applies to RLDRAMII and QDRII only; the phase reconfiguration feature for DDR2/3 is included in the CSR port.)
hc_pll_config_phasecounterselect	Input	4 bits	Specifies the counter select for dynamic phase adjustment. (Applies to RLDRAMII and QDR II only.)
hc_pll_config_phasestep	Input	1 bit	Specifies the phase step for dynamic phase shifting. (Applies to RLDRAMII and QDR II only.)
hc_pll_config_phaseupdown	Input	1 bit	Specifies if the phase adjustment should be up or down. (Applies to RLDRAMII and QDR II only.)
hc_pll_config_scanclk	Input	1 bit	PLL reconfiguration scan chain clock.
hc_pll_config_scanclkena	Input	1 bit	Clock enable port of the hc_pll_config_scanclk clock.
hc_pll_config_scandata	Input	1 bit	Serial input data for the PLL reconfiguration scan chain.
hc_pll_config_phasedone	Output	1 bit	When asserted, this signal indicates to core logic that phase adjustment is completed and that the PLL is ready to act on a possible second adjustment pulse.
hc_pll_config_scandataout	Output	1 bit	The data output of the serial scan chain.
hc_pll_config_scandone	Output	1 bit	This signal is asserted when the scan chain write operation is in progress and is deasserted when the write operation is complete.
<b>Note:</b>			
(1) Inputs and outputs are available only in PLL nonsharing mode and PLL master sharing mode. No inputs or outputs are available in PLL slave mode.			

To facilitate placement and timing closure and to help compensate for PLLs adjacent to I/Os and vertical I/O overhang issues that can occur when targeting HardCopy III and HardCopy IV devices, an additional pipeline stage is added to the write path in the RTL when you turn on **HardCopy Compatibility**. The additional pipeline stage is added in all cases, except when CAS write latency equals 2 (for DDR3) or CAS latency equals 3 (for DDR2), where the additional pipeline stage is not required to meet timing requirements. The additional pipeline stage does not affect the overall latency of the controller, because the controller's internal latency is reduced by 1 to compensate for the extra pipeline stage.

In DDR2 and DDR3 designs, at a certain frequency the DLL length changes when you generate the IP with **HardCopy Compatibility** enabled; this allows the DLL to work in both FPGA and HardCopy devices at the requested frequency. In addition, because the memory clock uses the global clock network, the write clock changes from a regional clock network to a global clock network, for reduced skew between the write clock and memory clock, resulting in improved leveling timing.

-  For information about HardCopy issues such as vertical I/O overhang, PLLs adjacent to I/Os, and timing closure, refer to the *I/O Features for HardCopy III Devices* chapter in volume 1 of the *HardCopy III Device Handbook* and *I/O Features for HardCopy IV Devices* chapter in volume 1 of the *HardCopy IV Device Handbook*.

## Document Revision History

Table 12-4 lists the revision history for this document.

**Table 12-4. Document Revision History**

Date	Version	Changes
November 2011	2.0	<ul style="list-style-type: none"> <li>■ Reorganized HardCopy design migration information into an individual chapter.</li> <li>■ Updated the <a href="#">HardCopy Migration Design Guidelines</a> section.</li> </ul>
June 2011	1.0	Initial release.