HB0794
Handbook
CoreReset_PF v2.2

Microsemi
a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

50200794. 2.0 9/19

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 2.0

Updated for CoreReset_PF v2.2.

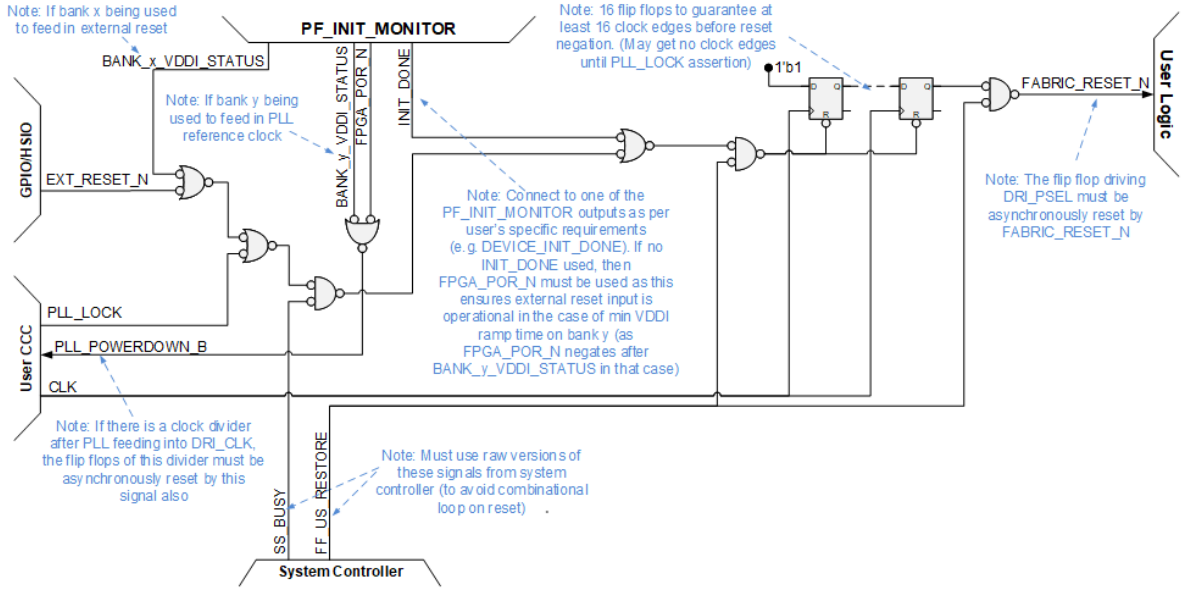
1.2 Revision 1.0

Revision 1.0 was the first publication of this document. Created for CoreReset_PF v2.1.

2 Introduction

CoreReset_PF allows synchronization of the resets to the user-specified clock domain to which each reset is feeded, so that, when assertion is asynchronous, negation is synchronous to the clock. CoreReset_PF block diagram is as shown in Figure 1, page 2.

Figure 1 • **CoreReset_PF Block Diagram**



2.1 Key Features

- Generates a reset, which is asserted asynchronously by one of multiple potential sources and which negates synchronously to a specified clock. This ensures that the recovery time of downstream logic is met and that all flip flops come out of reset in the same clock period.
- Multiple reset can be used such as external gpio, phase lock loop lock or init done in conjunction with the master reset from the system controller (through CORESYSSERVICES_PF).

2.2 Core Version

This handbook is for CoreReset_PF version 2.20.

2.3 Supported Families

- PolarFire®

2.4 Utilization and Performance

CoreReset_PF has been implemented for the PolarFire device family. A summary of the implementation data for CoreReset_PF is listed in Table 1, page 3.

Table 1 • **CoreReset_PF Device Utilization and Performance**

	Tiles			Utilization		
Family	**Sequential**	**Combinational**	**Total**	**Device**	**Total %**	**Performance MHz**
PolarFire	16	5	21	MPF300T	0.01	160

Note: Data in Table 1, page 3 was achieved using the default synthesis and layout settings. Top-level parameters/generics were left at their default values.

3 Design Description

3.1 I/O Signals

The port signals for the CoreReset_PF macro are as shown in Figure 2, page 4 and defined in Table 2, page 4.

Figure 2 • **CoreReset_PF I/O Signal Diagram**

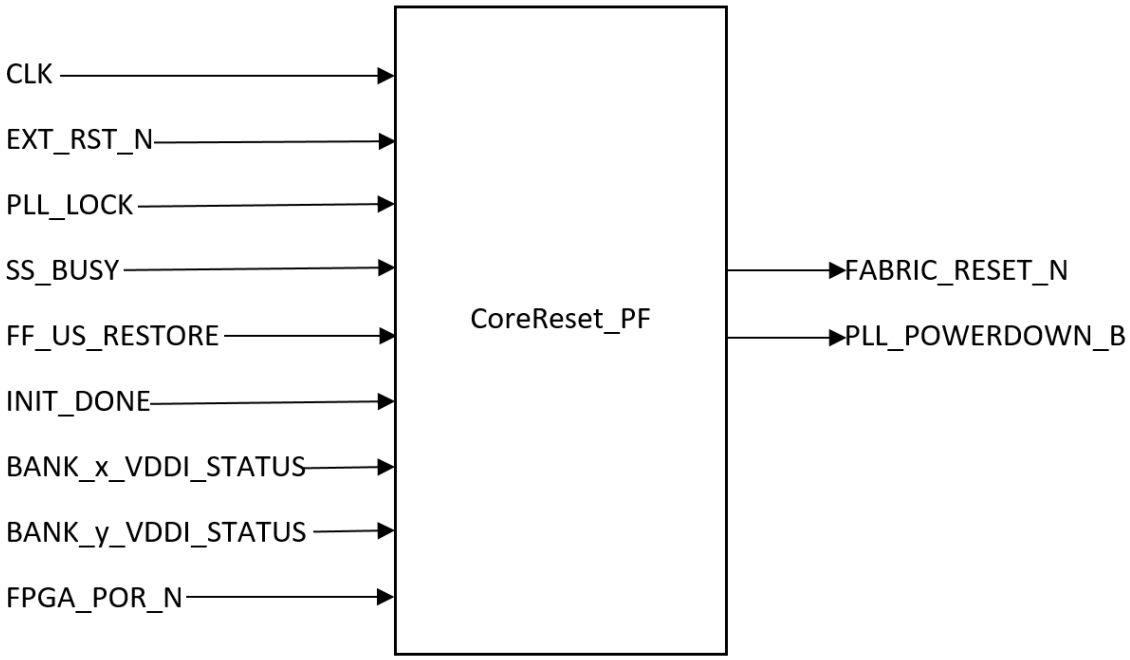


Table 2 • **CoreReset_PF I/O Signal Descriptions**

Port Name	Type	Description
EXT_RESET_N	Input	Active low reset input from GPIO or HSIO. Optional, if not used this must be tied high.
CLK	Input	Clock input from system, recovered or different clock domain.
PLL_LOCK	Input	PLL Lock signal from Clock Conditioning Circuit. Optional, if not used this must be tied high.
FF_US_RESTORE	Input	Optional. Only used in the systems which need to support flash freeze. If not used, must be tied low.
SS_BUSY	Input	When asserted, reset from PLL_LOCK or EXT_RESET_N are ignored. This ensures that no reset occurs whilst PLL is re-acquiring lock during flash freeze exit. Optional
INIT_DONE	Input	When signal asserts the initialization of the device is complete. This signal should be connected to one of the PF_INIT_MONITOR signals. These signal come from the system controller and must be used as they provide the master reset for the system.

Table 2 • **CoreReset_PF I/O Signal Descriptions**

Port Name	Type	Description
FABRIC_RESET_N	Output	This is the output reset, which may be used to reset user logic in the fabric. It is an active low reset, which asserts asynchronously, but negates synchronously to CLK.
BANK_x_VDDI_STATUS	Input	This signal is used to monitor VDDI supply on specific supply banks. When EXT_REST_N signal is not used, this signal must be tied high.
BANK_y_VDDI_STATUS	Input	This signal is used to monitor VDDI supply on specific supply banks.
FPGA_POR_N	Input	Active low input, asserted when the fabric is non-operational.
PLL_POWERDOWN_B	Output	Active low output, to held PLL in power down until the input buffer is known to be operational.

Note: All signals are active High (logic 1) unless otherwise noted.

4 Tool Flow

4.1 Licensing

CoreReset_PF is license free.

4.1.1 RTL

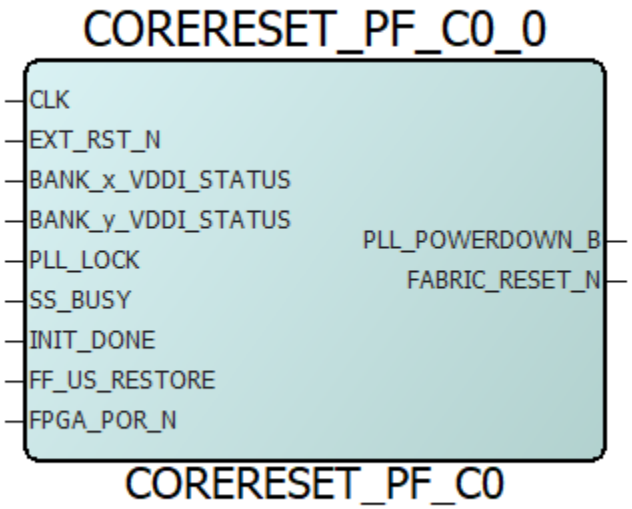
Complete RTL source code is provided for the core and testbenches.

4.2 SmartDesign

CoreReset_PF is preinstalled in the SmartDesign IP Deployment design environment.

The core should be configured using the configuration GUI within SmartDesign, as shown in Figure 3, page 6. For information on using the SmartDesign to instantiate and generate cores, see Libero SoC online help.

Figure 3 • **CoreReset_PF Full I/O View**



4.3 Simulation Flows

The user testbench for CoreReset_PF is included in all releases.

To run simulations, select the **User Testbench** flow within the **SmartDesign CoreReset_PF** configuration GUI, right-click the canvas, and select **Generate Design.**

When SmartDesign generates the design files, it will install the user testbench files.

To run the user testbench, Set the design root to the CoreReset_PF instantiation in the LiberoSoC design hierarchy pane and click the Simulation icon in the Libero SoC Design Flow window. This will invoke ModelSim® and automatically run the simulation.

4.4 Synthesis in Libero SoC

After setting the design root appropriately for your design, click the **Synthesis** icon in the Libero SoC. The Synthesis window appears, displaying the Synplicity® project. Set Synplicity to use the Verilog 2001 standard if Verilog is being used. To run Synthesis, click the **Run** icon.

4.5 Place-and-Route in Libero SoC

After setting the design root appropriately for the design, and after running Synthesis, click the **Layout** icon in the Libero SoC software to invoke Designer. CoreReset_PF requires no special place-and-route settings.