Reset Controller v1.1

IP User Guide

May 2024

Licensing Notice

Reset Controller v1.1 is a fully tested, portable, configurable, and synthesisable soft IP core provided by **Chip**munk **Logic**™. The IP source files are complied with IEEE VHDL/Verilog/System-Verilog standards. All the source codes are open-source licensed and hence may be used, modified, and shared without any restrictions or conflicts of interest with the original developer.

This IP core is provided 'as is,' without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and non-infringement. **Chip**munk **Logic**™ shall not be liable or held accountable for any loss or damage (direct or indirect) resulting from the use of any product, as the designs are not intended to be fail-safe or for use in any application requiring fail-safe performance. Hence, the user shall assume sole risk and liability for the use of any of our products in any of their applications.

Table of Contents

1. Reset Controller

Reset Controller is a soft IP that handles and provides resets to user designs. The core provides a variety of configuration options tailored to the user requirements to generate timed synchronized resets to a design or a sub-system.

2. Features

- ✓ Supports asynchronous reset at the primary reset input.
- ✓ Configurable digital glitch filter at the primary reset input.
- ✓ Supports auxiliary reset input.
- ✓ Pulse width validation for fully synchronous resets.
- ✓ Output resets configurable as fully synchronous or synchronized only at de-assertion.
- ✓ Configurable output reset polarity.
- ✓ Supports daisy-chained reset outputs (RST0/1/2), which are stretched and de-asserted sequentially.

3. Overview

The following figure shows the block diagram of Reset Controller.

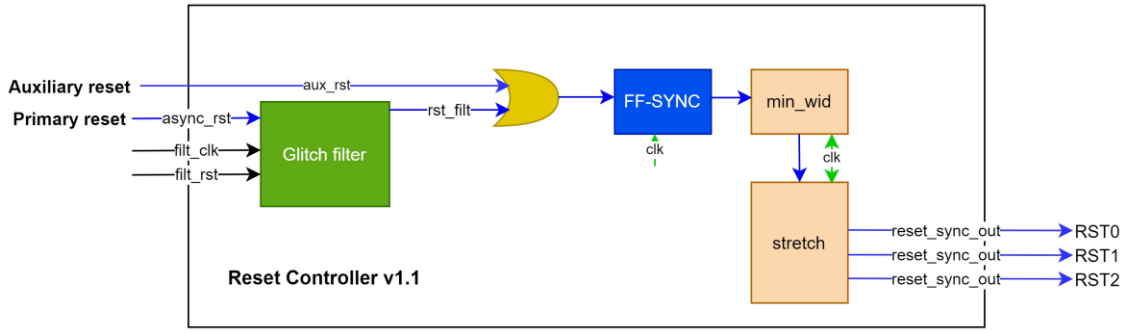

Figure 3.1: Reset Controller – Block Diagram

The core has two external reset inputs: primary reset and auxiliary reset. The primary reset could be an asynchronous reset coming from another system/power-on-reset or from external push button/switch. The auxiliary reset is typically a control signal from some functional block, which is ORed with the primary reset. The core synchronizes the reset to the system clock and generates three synchronized resets at the output: RST0/1/2.

Different blocks in Reset Controller are:

- **Glitch filter**: This is a digital glitch filter which can be enabled if the primary reset is prone to noise/glitches (for eg: the reset is coming from a push button/switch without any debouncing filter) If enabled, it can filter out the glitches in the primary reset input. The filter runs at separate a clock and reset. This is feature is configurable by the user.
- **FF synchronizer:** This block synchronizes the external reset to the system clock. The flop synchronizer chain can be configured by the user to mitigate the metastability of reset assertion/de-assertion.
- **Minimum pulse width validator**: This blocks acts like a second level glitch filter which validates the pulse width of reset in system clock cycles. This feature is configurable by the user and available only for fully synchronous mode of operation.
- **Reset stretcher:** This block is responsible for stretching the synchronized reset across multiple system clock cycles and sequence the de-assertion/removal of resets at RST0/1/2. This feature is configurable by the user.

4. Top-level Design Parameters/Macros

Reset Controller IP offers a variety of configuration options to tune the generated hardware. Following table lists all top-level design parameters and macros used to configure the IP.

Name	Type	Width	Description
GLF	macro	--	Define this macro to enable glitch filter
EN_FULL_SYNC_RST	parameter	1	Mode of operation of the IP. 1'b0 = synchronize only the reset de-assertion (ASYNC mode) 1'b1 = synchronize both assertion and de-assertion (FULL SYNC mode)
EN_MIN_WIDTH_RST	parameter	1	Enable min. pulse width validation; this can be enabled only if the IP is configured in FULL SYNC mode.
EN_ACTV_LOW_RST_OUT	parameter	1	1'b0 = active-high reset is generated at output 1'b1 = active high reset is generated at output
SYNC_FLOPS	parameter	3	No. of flops in the reset synchronizer. valid range = [2, 7]
GLITCH_LEN	parameter	11	Min. pulse width (in filt_clk cycles) required at the primary reset to be not classified as glitch. valid range = [1, 1024] • 0 = bypass the filter
MIN_WIDTH_RST	parameter	5	Min. pulse width (in clk cycles) required at the reset to be not classified as glitch. valid range = [2, 31]
RST0_LEN	parameter	5	Min. pulse width at RST0 (in clk cycles). RST0 is stretched by RST0_LEN cycles before releasing. valid range = [2, 31] • 0 = disable the reset stretching • 1 = invalid value
RST1_LEN	parameter	5	Min. pulse width at RST1 (in clk cycles). RST1 is stretched by (RST0_LEN + RST1_LEN) cycles before releasing. valid range = [2, 31] • 0 = disable the reset stretching • 1 = invalid value
RST2_LEN	parameter	5	Min. pulse width at RST2 (in clk cycles). RST2 is stretched by (RST0_LEN + RST1_LEN + RST2_LEN) cycles before releasing.

			valid range = [2, 31] • 0 = disable the reset stretching • 1 = invalid value

Table 4.1: Top-level Design Parameters/Macros

5. Top-level Ports/Interfaces

The following table lists all the top-level I/O ports/interfaces of the IP.

Signal Name	Direction	Width	Description
i_rst	input	1	Primary reset; asynchronous, active-high
i_aux_rst	input	1	Auxiliary reset; asynchronous, active-high
clk	input	1	System clock
filt_clk**	input	1	Glitch filter clock
filt_rst**	input	1	Glitch filter reset; fully synchronous, active-high
o_rst0_sync	output	1	Synchronized reset, RST0
o_rst1_sync	output	1	Synchronized reset, RST1
o_rst2_sync	output	1	Synchronized reset, RST2

*** These ports are available only if glitch filter is enabled during the IP configuration*

Table 5.1: Top-level Ports/Interfaces

6. Setting up the IP

This chapter discusses the guidelines for setting up with the IP to control the reset to a user design.

6.1 Input resets

Both the primary and auxiliary reset inputs to the core should be at a defined state after the power-up.

6.2 Glitch filter initialization

If glitch filter is enabled, it should be reset and initialized first before Reset Controller is ready for functional operation.

Initializing sequence

1. Assert the glitch filter reset.
2. Bring up the glitch filter clock.
3. Assert the glitch filter reset (`filt_rst`) for at least two clock cycles (`filt_clk`).
4. Release the glitch filter reset.
5. The core is now ready for functional operation.

6.3 Reset timing

The following figure shows the reset timing at RST0/1/2 when the IP is configured in ASYNC mode with glitch filter and reset stretching of N cycles on the daisy-chained reset outputs.

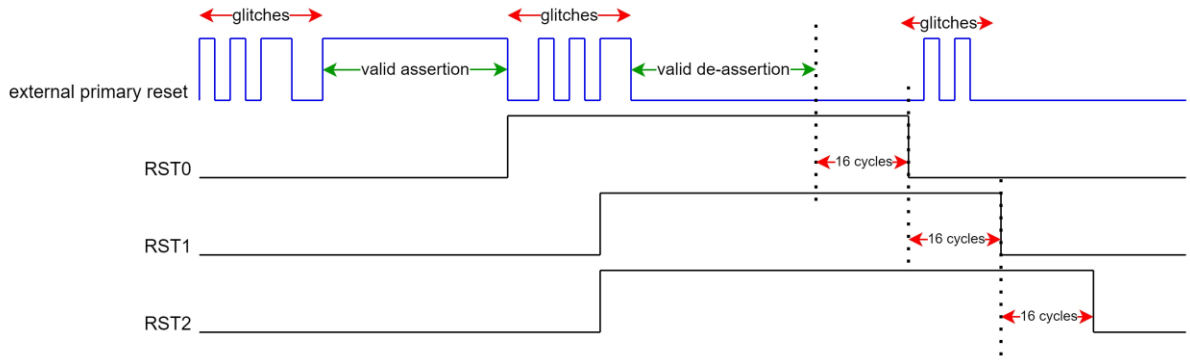

Figure 6.1: Reset Timing

7. Integrating the IP with examples

This chapter demonstrates some example designs where Reset Controller is used to reset a system.

7.1 Resetting a system with PLL

In this example, the system can be externally reset by a push button on board. The blocks inside the system are clocked internally by a PLL. The system should be held in reset until the PLL is brought up successfully, and the clock is stable. The lock status of the PLL is indicated by `pll_lock`.

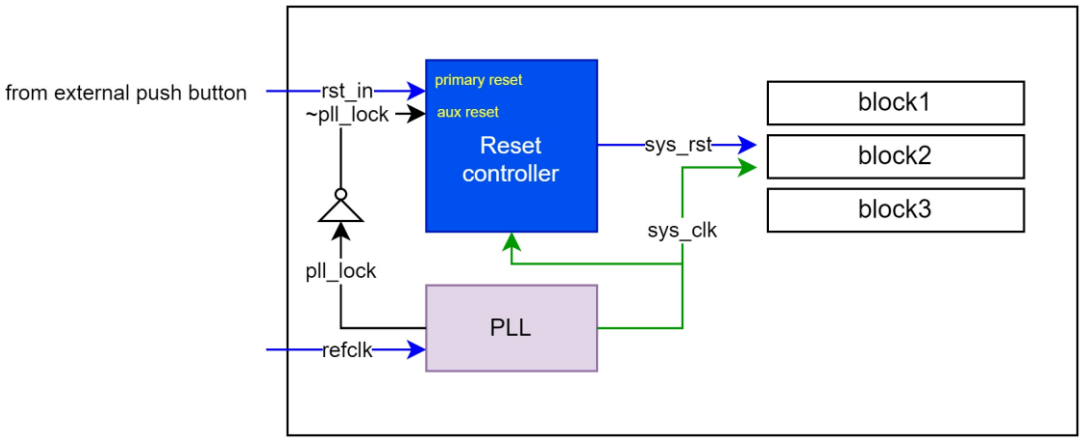

Figure 7.1: Reset Controller to reset a system with PLL

Glitch filter can be enabled to filter the push button bouncing. It ensures that only clean resets pass through and avoids inadvertent resets. The PLL lock is used as auxiliary reset here.

7.2 Resetting a processor sub-system

In this example, the processor sub-system runs in three different clocks which are synchronous to each other (all generated from the same PLL). The sub-system has a single external reset. The following resetting sequence is devised to reset the sub-system. The clock cycles are with respect to the slowest clock = peripheral clock.

1. Assert reset to all blocks on external reset.
2. Release bus reset after 16 clock cycles.
3. Release peripheral reset after 16 clock cycles.
4. Release processor reset after 16 clock cycles.

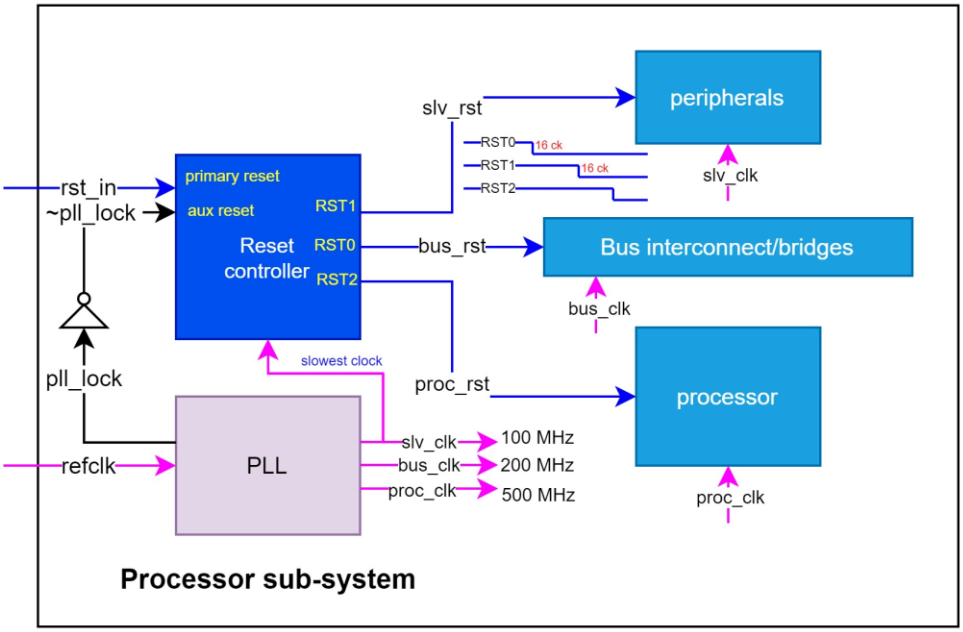

Figure 7.2: Reset Controller to reset a processor sub-system

Since all clocks are synchronous, it is safe to synchronize the reset to the slowest clock, i.e., the peripheral clock. The IP uses PLL lock signal as auxiliary reset. The system resets RST0/1/2 are enabled in daisy-chained mode and configured with reset stretching length = 16 clock cycles.

8. Application Notes

- Minimum pulse width validator can be enabled only if the IP is configured in FULL SYNC mode. This acts like a secondary filter on the synchronized reset. Hence, if glitch filter is already enabled, this block may not be required, unless a second level of filtering required at the auxiliary reset.
- Glitch filter's reset is internally synchronized to the filter clock at both assertion and de-assertion. Hence, the reset should be at least two clock cycles long to guarantee proper resetting.
- The auxiliary reset is assumed to be glitch free as it is assumed to be a control signal from some functional block, hence glitch filter is applicable only to the primary reset. If another external reset control is required, it should be externally ORed and fed to the primary reset/auxiliary reset.

9. Known Limitations/Issues

[NIL]

Appendix

a) FPGA Resource Utilization

IP configuration	GLF macro = Defined EN_FULLS_SYNC_RST = 1'b0 EN_MIN_WIDTH_RST = 1'b0 EN_ACTV_LOW_RST_OUT = 1'b0 SYNC_FLOPS = 3'd3 GLITCH_LEN = 11'd16 RST0_LEN = 5'd16 RST1_LEN = 5'd16 RST2_LEN = 5'd16
FPGA Targeted	Xilinx Basys3 (XC7-A35T-CPG236-1), Artix-7 FPGA based board
Synthesiser	Vivado 2019.2
Targeted clock frequency	100 MHz
LUTs	14
Registers	69

Revision History

The following tables shows the revision history of this document.

Date	IP Version	Revision
May-2024	1.1	• Initial version

Reset Controller v1.1

An open-source licensed soft IP core

Developer : **Mitu Raj**
Vendor : **Chip**munk **Logic**™, chip@chipmunklogic.com
Website : **chipmunklogic.com**