Reset Controller v1.0

IP Documentation

September 2021

Licensing Notice

Reset Controller v1.0 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.

Chipmunk **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 can be configured to generate customized resets for an entire processor sub-system or SoC, including the processor, interconnect and peripherals. The core can handle numerous reset conditions at the input and generate appropriately timed synchronous resets for the whole system.

2. Features

- ✓ Supports asynchronous reset input which is synchronized to clock.
- ✓ Configurable polarity and synchronizer for asynchronous reset input.
- ✓ Configurable minimum width for input reset states to be recognized.
- ✓ Both polarities supported for synchronous reset at output.
- ✓ Supports resetting on losing phase lock with clock.
- ✓ Supports power-on-reset on FPGAs.
- ✓ Configurable number of resets (RST0, RST1, RST2; see Reset Modes).
- ✓ Sequencing of removal/de-assertion of resets at output, where N is configurable:
    - RST0 is removed first.
    - RST1 is then removed after N cycles.
    - RST2 is then removed after N cycles.

3. Overview

Fig 3.1 shows the top-level block diagram of Reset Controller. It has an external asynchronous reset at input, and set of synchronous resets at output. The core has a single clock.

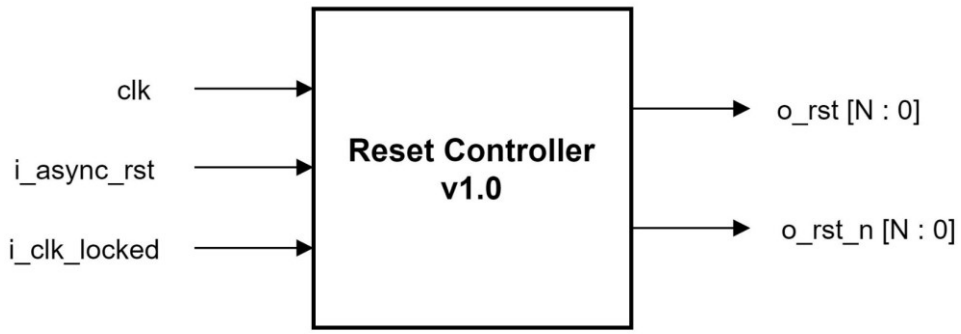

Fig 3.1: Reset Controller – Block Diagram

3.1 Core Functionality

External reset/input reset need not be synchronous with the core clock. Hence, it is synchronized inside using multi-flop synchronizers to remove metastability. The synchronizer chain is configurable using `SYNC_STAGES`. Input reset has to comply with a minimum pulse width to be recognized by the core. Output resets become active after the input reset has gone active and stays active for a minimum number of clock cycles. This minimum width is configurable using `MIN_WIDTH`. After the input reset has gone inactive for `MIN_WIDTH` cycles, reset-removal sequencing begins. If during sequencing, input reset has gone active again for `MIN_WIDTH` cycles, output resets become active again.

Fig 3.2 shows different functional blocks of Reset Controller:

- **Multi-flop Synchronizer**: Chain of synchronizing flip-flops to synchronize external asynchronous reset to the clock domain. Responsible for removing metastability. These flops have to be placed together for best MTBF timing. The number of flops in the chain have to be configured based on clock speed.
- **Minimum Pulse Width Validator**: Chain of flops and combinatorial logic to filter out only reset pulses that satisfy the minimum width requirement. This minimum width is applicable for both assertion and de-assertion. Any lesser width pulses are considered as ‘glitches’, and hence have no effect at the output.
- **Pulse Stretcher**: Responsible for reset-removal sequencing, where reset pulses are stretched for specific number of clock cycles before starting to remove sequentially.

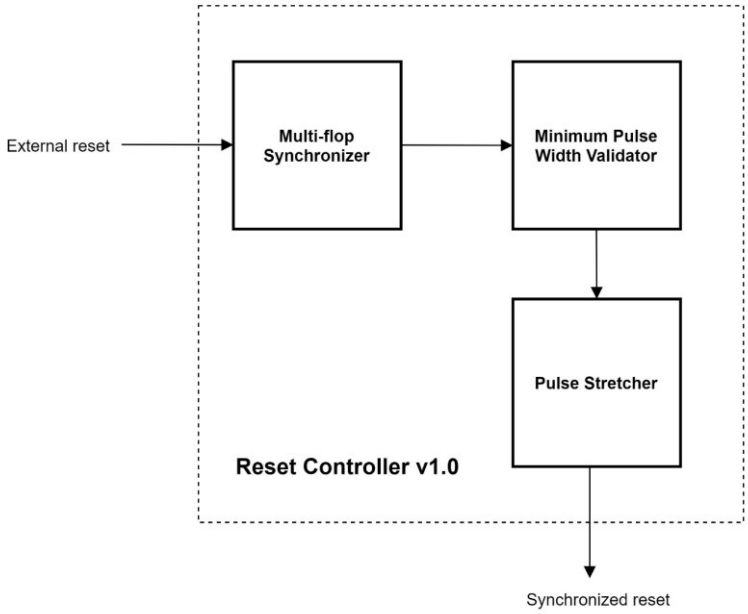

Fig 3.2: Reset Controller – Functional Blocks

The core supports phase-locked resets. Power-on-reset is supported on FPGAs. The core asserts reset on power-on and then sequential de-assertion, when targeted on FPGAs.

4. Configuration Parameters

Reset Controller supports variety of configurations tailored for user-application. Table 4.1 lists all configuration parameters available to the user.

Parameter	Type	Default & Range	Description
EXT_RST_POL	bit	**'1'**	Polarity of external asynchronous reset.
RST_MODE	integer	[1, 2, **3**]	Mode of operation of IP. Configures the no. of resets needed at output, see here for details.
SYNC_STAGES	integer	**2**	No. of flip-flops in synchronizer chain to synchronize the external reset.
MIN_WIDTH	integer	**4,** [2-16]	Minimum width (in clock cycles) for external reset to be recognized as asserted or de-asserted.
REMOVAL_CYCLES	integer	**16,** [1-64]	No. of clock cycles between removal of successive resets, see here for details.

Table 4.1: Configuration Parameters

5. Top-level Ports/Interfaces

Table 5.1 lists all top-level I/O ports/interfaces of the IP.

Signal Name	Direction	Width	Description
clk	input	1	Core clock
i_async_rst	input	1	Asynchronous Reset in
i_clk_locked	input	1	DCM/PLL Phase-lock signal, see here for details.
o_rst	output	RST_MODE	Active-high Synchronous Resets RST0-2 out
o_rst_n	output	RST_MODE	Active-low Synchronous Resets RSTN0-2 out

Table 5.1: Top-level I/O Ports/Interfaces

6. Reset Modes and Timing

The core supports three configuration modes: *Mode-1*, *Mode-2*, *Mode-3*. It configures the number of resets available at output. Multiple resets (RST0-2) always get asserted together in the core, but follows reset-removal sequencing as configured by REMOVAL_CYCLES.

The timing for *Mode-3* operation of the core with: `EXT_RST_POL` = '1', `SYNC_STAGES` = 2, `REMOVAL_CYCLES` = 16, `MIN_WIDTH` = 4, is shown in Figure 6.1.

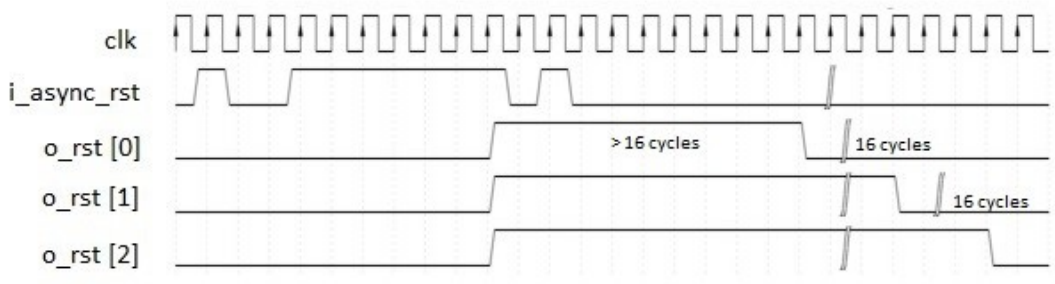

Figure 6.1: Mode-3 Operation Timing

7. Designing with the IP

This chapter discusses the guidelines, performance, and other known limitations while designing with Reset Controller.

7.1 Clocking

The core has a single clock. If the system has multiple synchronous clock domains, the core should be clocked by the slowest operating clock. This ensures all clock domains are properly reset and sequenced by the core.

7.2 External Reset

The external asynchronous reset is expected to have minimum bouncing/glitches. Make use of `MIN_WIDTH` accordingly. Choose `SYNC_STAGES` as per the frequency of the slowest clock in the system.

7.3 Latency

The core has a typical latency = `MIN_WIDTH + SYNC_STAGES + 2`, for reset assertion/de-assertion.

7.4 Locking with Clock

If the core clock is generated by a DCM/PLL, the input signal `i_clk_locked` can be connected to the phase-locked signal from the DCM/PLL. If multiple DCM/PLLs are used, the DCM/PLL that achieves lock last should be connected to this input. Reset is asserted if `i_clk_locked` goes low during the operation of the core. If locking feature is not needed, tie `i_clk_locked` to '1' throughout the operation of the core.

7.5 Typical Application

A typical application of Reset Controller would be to properly reset a processor sub-system with interconnect/bus structures and peripherals as shown in Fig 6.1. For e.g., configure the core for *Mode-3*, with `REMOVAL_CYCLES` = 16, where:

- RST0 – Interconnect/Bus structures reset.
- RST1 – Peripheral reset.
- RST2 – Processor reset.

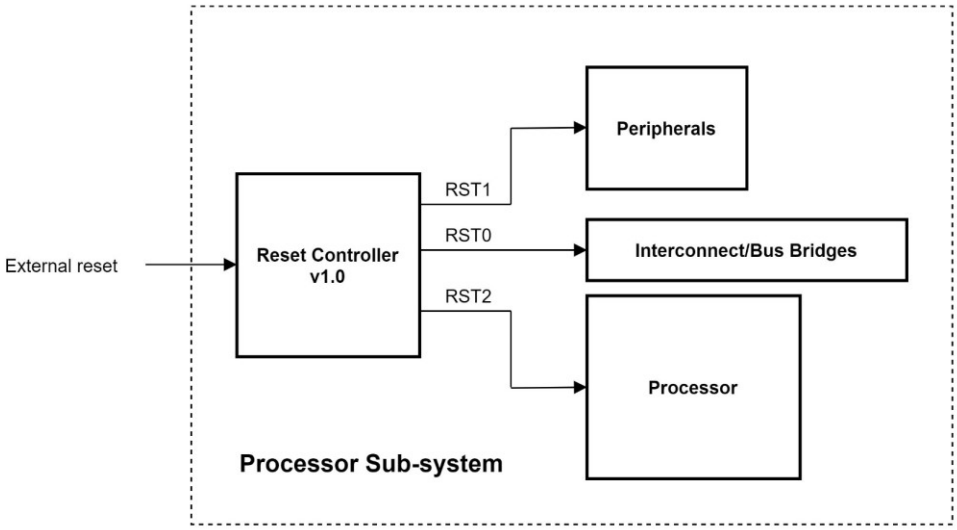

Figure 7.1: Reset Controller – in Processor Sub-system

Appendix

a) Performance and Resource Utilization on FPGA

The following is an estimate of timing and resource utilization of Reset Controller v1.0 for a typical configuration when targeted on Xilinx Artix-7 FPGA. When combined with other cores in the system, the timing and resource utilization can vary from the reported values.

IP Configuration	`RST_MODE = 3`
	`SYNC_STAGES = 2`
	`MIN_WIDTH = 4`
	`REMOVAL_CYCLES = 16`

FPGA Targeted	Xilinx Basys-3 Board (XC7A-35T-CPG236-1)
Synthesiser	Vivado 2015.4
Targeted Clock Frequency	200 MHz
LUTs	5
Registers	55

Reset Controller v1.0

An open-source licensed IP core

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