*Technion*

*Electrical Engineering Department*

High Speed Digital System Lab

Clock and Reset

Documentation

**Version**: 1.0

**Last update**: 08.03.2011

**Written By**: Beeri Schreiber and Alon Yavich

Table of content

[Table of Changes 4](#_Toc293164201)

[1 Scope 4](#_Toc293164202)

[2 Abbreviations 4](#_Toc293164203)

[3 General Description 4](#_Toc293164204)

[3.1 Synchronized Reset Importance 4](#_Toc293164205)

[3.1.1 What is a synchronous reset? 4](#_Toc293164206)

[3.1.2 Advantages of asynchronous reset 4](#_Toc293164207)

[3.1.3 Disadvantages of asynchronous reset 5](#_Toc293164208)

[3.1.4 Synchronization the asynchronous reset 5](#_Toc293164209)

[3.1.5 Using synchronized reset 5](#_Toc293164210)

[3.2 Global Nets 7](#_Toc293164211)

[3.3 Global Nets Pinout 7](#_Toc293164212)

[3.4 Global Nets Generic Parameters 7](#_Toc293164213)

[3.5 Resources 7](#_Toc293164214)

[3.6 Global Nets Wave 8](#_Toc293164215)

[3.7 Global Nets Connecting Scheme 8](#_Toc293164216)

[4 Clock Block 9](#_Toc293164217)

[4.1 Clock Block Pinout 9](#_Toc293164218)

[4.2 PLL Instance 9](#_Toc293164219)

[4.2.1 PLL Pinout 10](#_Toc293164220)

[5 Reset Block 11](#_Toc293164221)

[5.1 Reset Block Pinout 11](#_Toc293164222)

[5.2 Reset Block Generic Parameters 11](#_Toc293164223)

[5.3 Reset Block Connecting Scheme 12](#_Toc293164224)

[5.4 Reset Block Wave 12](#_Toc293164225)

[5.5 Reset Filter 13](#_Toc293164226)

[5.5.1 Reset Filter Working Method 13](#_Toc293164227)

[5.5.2 Reset Filter Pinout 14](#_Toc293164228)

[5.5.3 Reset Filter Generic Parameters 14](#_Toc293164229)

[5.6 Sync Reset Generator 14](#_Toc293164230)

[5.6.1 Sync Reset Generator Working Method 15](#_Toc293164231)

[5.6.2 Sync Reset Generator Pinout 15](#_Toc293164232)

[5.6.3 Sync Reset Generator Generic Parameters 15](#_Toc293164233)

[6 Performed Tests for Global Nets 16](#_Toc293164234)

Table of Tables

[Table 1 – Table of Changes 4](#_Toc287589779)

[Table 2 – Global Nets Pinout 7](#_Toc287589780)

[Table 3 – Global Nets Generic Parameters 7](#_Toc287589781)

[Table 4 – Clock Block Pinout 10](#_Toc287589782)

[Table 5 – PLL Pinout 11](#_Toc287589783)

[Table 6 – Reset Block Pinout 12](#_Toc287589784)

[Table 7 – Reset Block Generic Parameters 12](#_Toc287589785)

[Table 8 – Reset Filter Pinout 15](#_Toc287589786)

[Table 9 - Reset Filter Generic Parameters 15](#_Toc287589787)

[Table 10 – Sync Reset Generator Pinout 16](#_Toc287589788)

[Table 11 – Sync Reset Generator Generic Parameters 16](#_Toc287589789)

Tables of Figures

[Figure1 – Syncronized Reset Deactivation. Asynchronized Reset Activation 5](#_Toc287589790)

[Figure 2 – Syncronized Reset Deactivation. Asynchronized Reset Activation RTL 6](#_Toc287589791)

[Figure 3 – Synchronized Reset Method 2 6](#_Toc287589792)

[Figure 4 – Synchronized Reset Method 2 RTL 6](#_Toc287589793)

[Figure 5 – Global Nets Hirarchy 7](#_Toc287589794)

[Figure 6 – Global Nets Pinout 7](#_Toc287589795)

[Figure 7 – Global Nets Wave 8](#_Toc287589796)

[Figure 8 – Global Nets Connecting Scheme 9](#_Toc287589797)

[Figure 9 – Clock Block Hirarchy 10](#_Toc287589798)

[Figure 10 – Clock Block Pinout 10](#_Toc287589799)

[Figure 11 – PLL Hirarchy 10](#_Toc287589800)

[Figure 12 – PLL Pinout 11](#_Toc287589801)

[Figure 13 – Reset Block Hirarchy 12](#_Toc287589802)

[Figure 14 – Reset Block Pinout 12](#_Toc287589803)

[Figure 15 – Reset Block Connecting Scheme 13](#_Toc287589804)

[Figure 16 – Reset Block Wave 13](#_Toc287589805)

[Figure 17 – Reset Filter Hirarchy 14](#_Toc287589806)

[Figure 18 – Reset Filter Working Method 14](#_Toc287589807)

[Figure 19 – Reset Filter Pinout 15](#_Toc287589808)

[Figure 20 – Sync Reset Generator Hirarchy 15](#_Toc287589809)

[Figure 21 – Sync Reset Generator Working Method 16](#_Toc287589810)

[Figure 22 – Sync Reset Generator Pinout 16](#_Toc287589811)

# Table of Changes

|  |  |  |
| --- | --- | --- |
| Version | Date of Change | Description |
| 1.0 | 08.03.2011 | Creation of documentation |

Table 1 – Table of Changes

# Scope

This document aims to describe the working method of the reset filter and clocks in the system.

The documentation describes the above method, which is implemented in the *runlen* project. Though, the main idea fits to any other project.

The document will focus especially on the reset filter and sync reset generator.

# Abbreviations

1. FPGA – Field-Programmable Gate Array
2. PLL – Phase-Locked Loop
3. CL – Combinational Logic
4. FF – Flip Flop

# General Description

## Synchronized Reset Importance

*(Most of the text under this section is quotes from "Asynchronous & Synchronous Reset Design Techniques - Part Deux", SNUG Boston 2003, Rev 1.3, written by Clifford E. Cummings, Don Mills, Steve Golson)*

Improper implementation of asynchronous resets in digital logic design can cause serious operational design failures. The biggest problem with asynchronous resets is the reset release, also called *reset removal*.

Asynchronous reset flip-flops incorporate a reset pin into the flip-flop design. The reset pin is typically active low (the flip-flop goes into the reset state when the signal attached to the flip-flop reset pin goes to a logic low level).

### What is a synchronous reset?

Synchronous resets are based on the premise that the reset signal will only affect or reset the state of the flip-flop **on the active edge of a clock**. The reset can be applied to the flip-flop as part of the combinational logic generating the d-input to the flip-flop.

### Advantages of asynchronous reset

The biggest advantage to using asynchronous resets is that, as long as the vendor library has asynchronously reset-able flip-flops, **the data path is guaranteed to be clean**. Designs that are pushing the limit for data path timing, cannot afford to have added gates and additional net delays in the data path due to logic inserted to handle synchronous resets. **Using an asynchronous reset, the designer is guaranteed not to have the reset added to the data path**.

### Disadvantages of asynchronous reset

The biggest problem with asynchronous resets is that they are asynchronous, both at the assertion and at the de-assertion of the reset. The assertion is a non issue, **the de-assertion is the issue**. If the asynchronous reset is released at or near the active clock edge of a flip-flop, the output of the flip-flop could go **metastable** and thus the reset state of the FPGA could be lost (Reset recovery time cause **tsu** not to be satisfied). Reset removal happening in different clock cycles for different sequential elements, might be problematic as well.

Another problem that an asynchronous reset can have, depending on its source, is spurious resets due to noise or glitches on the board or system reset.

### Synchronization the asynchronous reset

Two flip-flops are required to synchronize the reset signal to the clock pulse, where the second flip-flop is used to remove any **metastability**, which might be caused by the reset signal being removed asynchronously and too close to the rising clock edge.

More flip-flops may be added, as filters, in order to prevent reset glitches.

### Using synchronized reset

Methods of using the fully synchronized reset / synchronized reset deactivation are depended on the implementation itself.

Two methods will be shown here. The first one uses FF with CLR port, and the second one without CLR port.

1. Example 1:

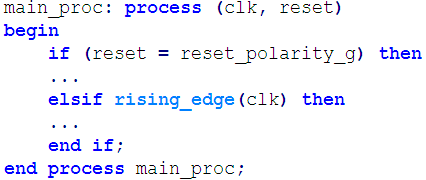


Figure1 – Syncronized Reset Deactivation. Asynchronized Reset Activation

This example will be implemented using reset, which will be connected to the CLR port on the FF. In this case, the FF will respond immediately to reset activation, independent on the clock, but will respond to reset deactivation only at clock's rising edge. Note that this is asynchronous reset, since only the reset deactivation is synchronized to the clock.

Such Implementation will be seen as the figure below:

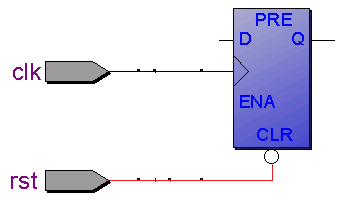


Figure 2 – Syncronized Reset Deactivation. Asynchronized Reset Activation RTL

1. Example 2:

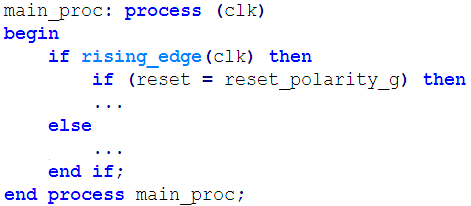


Figure 3 – Synchronized Reset Method 2

This example will be implemented using reset, which will be connected to an AND gate (or OR gate, depends on the reset polarity), which is connected to the data line. In this case, the FF will respond to reset activation and deactivation only at clock's rising edge. This is a completely synchronized reset, since the reset activation and deactivation will influence the FF only at the clock's rising edge.

Implementation will be seen as the figure below:

Figure 4 – Synchronized Reset Method 2 RTL

## Global Nets

These are the global nets in the FPGA system. In is consumed of:

1. **Clock Block Top** – which responsible to create the required clocks in the system, using PLL
2. **Reset Block Top** – which responsible to filter the FPGA reset, and create synchronized reset to each clock, with PLL-locked dependent.

Figure 5 – Global Nets Hirarchy

## Global Nets Pinout

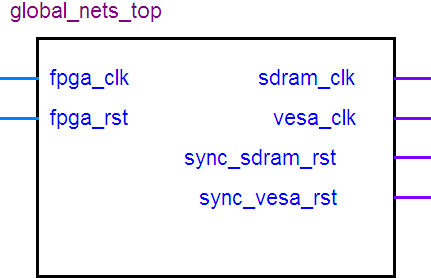


Figure 6 – Global Nets Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Fpga\_clk | In | Clock to FPGA (50MHz) |
| Fpga\_rst | In | Asynchronous reset to FPGA. Reset polarity will be set according to the generic parameter '*reset\_polarity\_g*' |
| Sdram\_clk | Out | SDRAM Clock (133MHz) |
| Vesa\_clk | Out | VESA Clock (40MHz) |
| Sync\_sdram\_rst | Out | Synchronized-to-the-SDRAM-clock reset |
| Sync\_vesa\_rst | Out | Synchronized-to-the-VESA-clock reset |

Table 2 – Global Nets Pinout

## Global Nets Generic Parameters

| Generic Parameter | Default Value | Description |
| --- | --- | --- |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |

Table 3 – Global Nets Generic Parameters

## Resources

Required resources, when synthesizing, using Quartus, for Altera's Cyclon II FPGA:

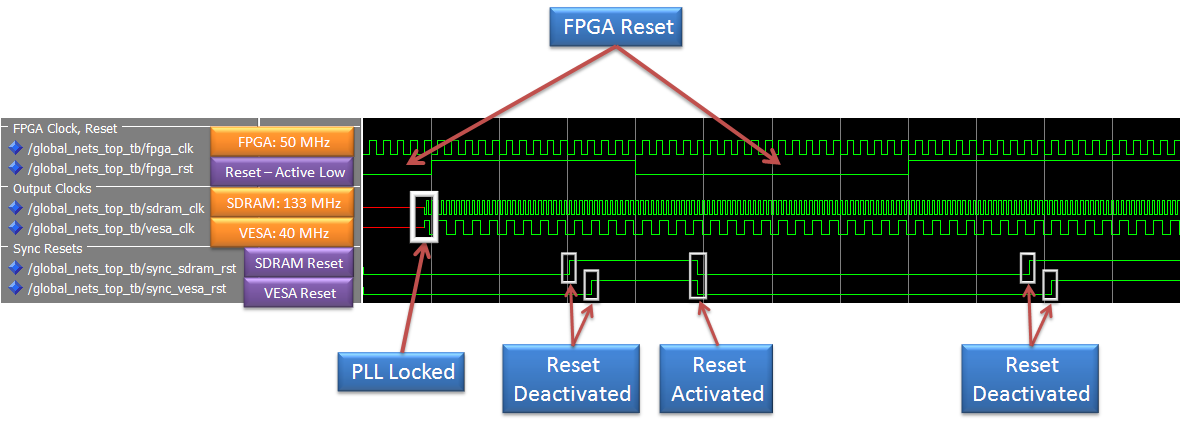
* 5 AND gates
* 1 OR gate
* 18 DFF
* 4 MUX
* 1 PLL

**Maximum Working Frequency**: 610MHz

## Global Nets Wave

The waves are divided into 3 sections:

1. **FPGA Clock and Reset** – Input 50MHz clock, Asynchronous reset
2. **Output Clocks** – SDRAM Clock (133MHz) and VESA Clock (40MHz)
3. **Synchronous Resets** –Synchronous resets to the SDRAM and VESA clocks

 Figure 7 – Global Nets Wave

In the above wave, both SDRAM and VESA clocks are available as soon as the PLL is locked. Note that the reset deactivation ('0' = active, '1' = inactive) occurs in a delay, since it is being filtered and matched to each clock.

The reset activation occurs at the same time to all clock domains, after filtration.

## Global Nets Connecting Scheme

The connecting scheme to the clock block and reset block is presented in the below figure:

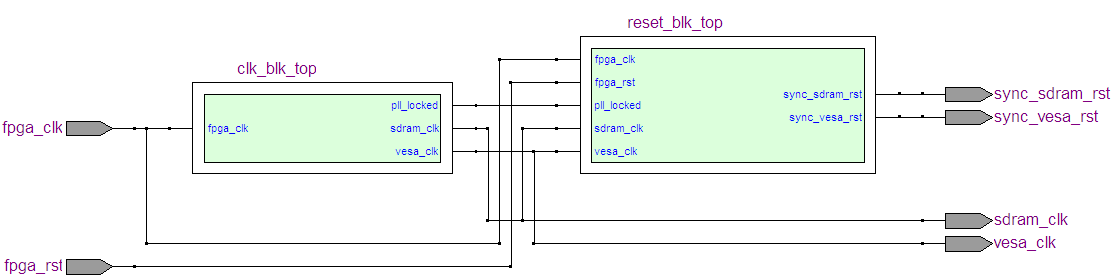


Figure 8 – Global Nets Connecting Scheme

# Clock Block

The clock block generates the required clocks to the system, from an input clock. It is consumed from PLL instantiation only.

Figure 9 – Clock Block Hirarchy

## Clock Block Pinout

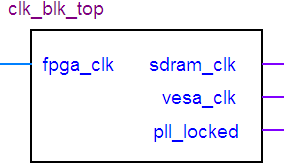


Figure 10 – Clock Block Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Fpga\_clk | In | Clock to FPGA (50MHz) |
| Sdram\_clk | Out | SDRAM Clock (133MHz) |
| Vesa\_clk | Out | VESA Clock (40MHz) |
| Pll\_locked | Out | PLL-Locked indication |

Table 4 – Clock Block Pinout

## PLL Instance

The PLL is created from Altera's MegaFunction (ALTPLL Component).

Figure 11 – PLL Hirarchy

### PLL Pinout

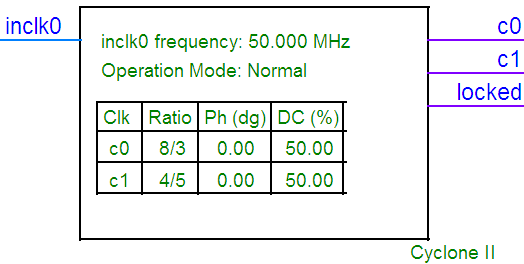


Figure 12 – PLL Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Inclk0 | In | Clock to FPGA (50MHz) |
| C0 | Out | SDRAM Clock (133MHz) |
| C1 | Out | VESA Clock (40MHz) |
| Locked | Out | PLL-Locked indication |

Table 5 – PLL Pinout

# Reset Block

The Reset block is consumed of two blocks:

1. **Reset Filter** – Filters the reset from the FPGA and the PLL-locked signal, and generate synchronized reset from both. In case PLL is not locked – system will be at reset state. In case PLL is locked – system will be at reset state only if the FPGA reset is active.
2. **Sync Reset Generator** – Generate synchronized reset to a given clock.

Figure 13 – Reset Block Hirarchy

## Reset Block Pinout

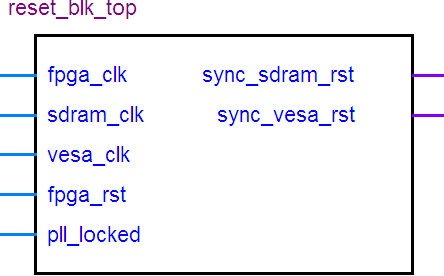


Figure 14 – Reset Block Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Fpga\_clk | In | Clock to FPGA (50MHz) |
| Sdram\_clk | In | SDRAM Clock (133MHz) |
| Vesa\_clk | In | VESA Clock (40MHz) |
| Fpga\_rst | In | Reset to FPGA |
| Locked | In | PLL-Locked indication |
| Sync\_sdram\_rst | Out | Synchronized-to-the-SDRAM-clock reset |
| Sync\_vesa\_rst | Out | Synchronized-to-the-VESA-clock reset |

Table 6 – Reset Block Pinout

## Reset Block Generic Parameters

| Generic Parameter | Default Value | Description |
| --- | --- | --- |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |

Table 7 – Reset Block Generic Parameters

## Reset Block Connecting Scheme

The connecting scheme to the reset filter (also known as ***reset debouncer****) and* sync reset generator is presented in the below figure:

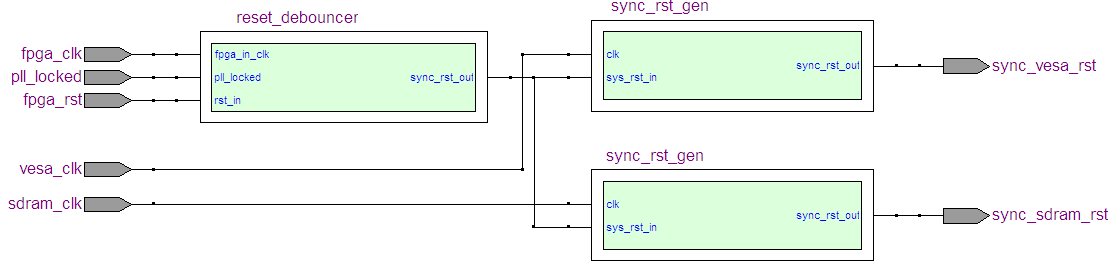


Figure 15 – Reset Block Connecting Scheme

## Reset Block Wave

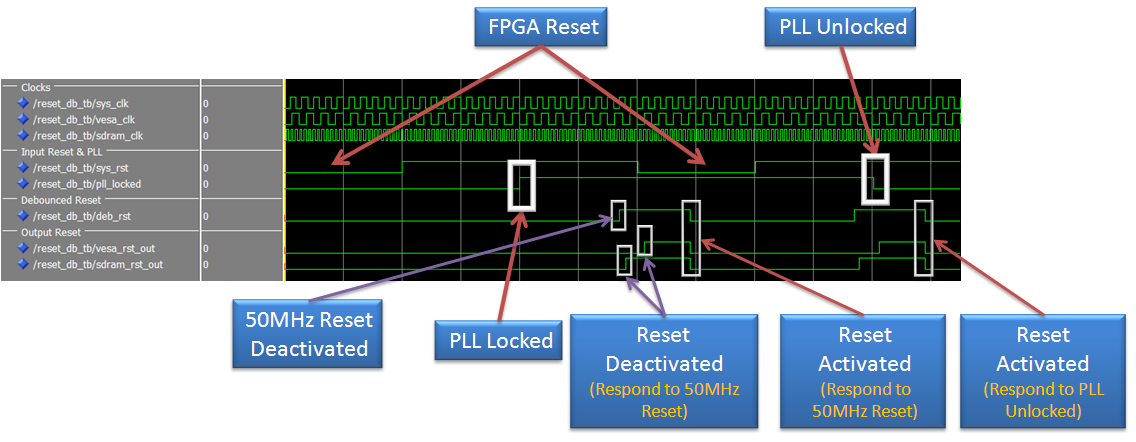


Figure 16 – Reset Block Wave

In the above wave, it is observable that when PLL is not locked – VESA and SDRAM sync resets are activated.

The reset activation occurs at the same time to all clock domains, after filtration.

## Reset Filter

The Reset Filter filters the input reset and PLL-locked signal, and generate synchronized reset from the AND condition of both signals.

Figure 17 – Reset Filter Hirarchy

### Reset Filter Working Method

Figure 18 – Reset Filter Working Method

'A1', 'A2', 'B1' and 'B2' FF purpose is to prevent **metastability** from entering the system, where the first FF ('A1', 'B1') are probably in metastabiliy state. The chance that the metastability will pass beyond the second FF ('A2', 'B2') is almost zero.

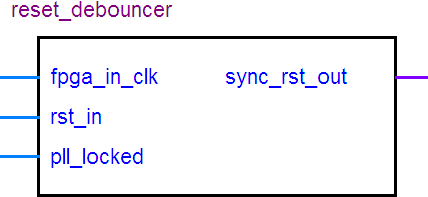
The 4 FPGA-Reset's FF and the 4 PLL-Locked-Reset's FF are FF without CLR port, where the last 4 FF, of the Sync-Reset, have an asynchronous CLR (Active Low Reset) port.

**FPGA Asynchronous reset** is sampled by FF, which marked 'A1'. Together with 'A2', both FF prevents metastability. The rest 2 flip-flops purpose is to filter the reset. Only if all last 3 FF are '1', then the output of the CL will be '1', and only if all last 3 FF are '0', then the output of the CL will be '0'. The reset **activation** occurs at the same time to **all clock domains**, after filtration, since all FF are connected to the FPGA clock. Reset deactivation will occur at different time to each clock domain, but very close to each other.

**PLL Locked** is sampled by FF, which marked 'B1'. Together with 'B2', both FF prevents metastability. The rest 2 flip-flops purpose is to filter the PLL signal. Only if all last 3 FF are '1', then the output of the CL will be '1', and only if all last 3 FF are '0', then the output of the CL will be '0'.

AND (An inverter might be placed before the AND, in the reset line, in case it is active low) gate causes all last 4 FF to be to be at reset state when reset is activated / PLL is not locked filtered signal is detected. When reset is deactivated, '1' (or '0' in case of active high reset) will enter to the first FF of the last 4 FF. Since after the PLL Locked and FPGA Reset FF there is a AND gate and CL, the filtered reset is again not synchronized to the clock. The last FF's job is to synchronize the reset deactivation to the clock's rising edge, but to respond immediately to filtered reset activation / PLL not locked filtered signal.

### Reset Filter Pinout

Figure 19 – Reset Filter Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Fpga\_in\_clk | In | Clock to FPGA (50MHz) |
| Rst\_in | In | Reset to FPGA |
| Pll\_locked | In | PLL-Locked indication |
| Sync\_rst\_out | Out | Synchronized-to-the-clock reset |

Table 8 – Reset Filter Pinout

### Reset Filter Generic Parameters

| Generic Parameter | Default Value | Description |
| --- | --- | --- |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |

Table 9 - Reset Filter Generic Parameters

## Sync Reset Generator

The sync reset generator generates synchronized-to-the-clock reset.

Figure 20 – Sync Reset Generator Hirarchy

### Sync Reset Generator Working Method

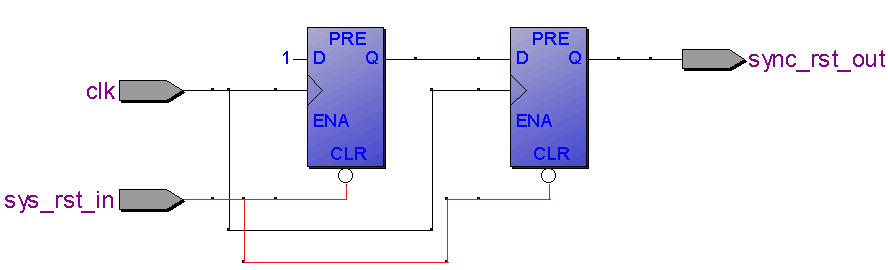


Figure 21 – Sync Reset Generator Working Method

When reset is being activated, the output reset will be activated immediately, independently on the clock. When the reset is being deactivated, '1' (or '0' in case of active high reset) is being entered to the first FF and from there to the second FF. In this method – the reset deactivation is synchronized to the given clock.

**Important**: The input reset is sampled from the filtered FPGA reset, which is synchronized to the FPGA clock and not to the given clock. In such implementation, the reset activation will occur to all clock domains at the same time, and the reset deactivation will occur as short time as possible to each other clock.

### Sync Reset Generator Pinout

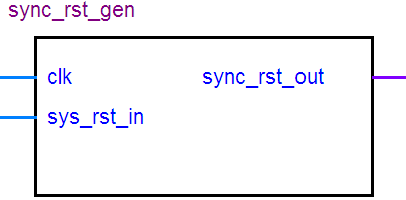


Figure 22 – Sync Reset Generator Pinout

| Table Pin Name | Direction | Description |
| --- | --- | --- |
| Clk | In | Input Clock |
| Sys\_rst\_in | In | Input reset |
| Sync\_rst\_out | Out | Synchronized-to-the-clock reset |

Table 10 – Sync Reset Generator Pinout

### Sync Reset Generator Generic Parameters

| Generic Parameter | Default Value | Description |
| --- | --- | --- |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |

Table 11 – Sync Reset Generator Generic Parameters

# Performed Tests for Global Nets

The following test where performed on the global nets:

1. Using Altera Mega Function Library, for PLL: Activating and deactivating the input reset to the FPGA. Sync resets respond correctly. See global nets wave.
2. Without using Altera Mega Function: Test performed individually for reset block: Activating and deactivating the input reset to the FPGA, and the PLL-Locked signal, alternately. See reset block wave.