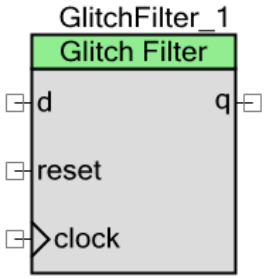
Glitch Filter
2.0



Features

- Eliminates unwanted “glitch” pulses on digital input lines
- Programmable filtering length and bypass option

General Description

Glitch filtering is the process of removing unwanted pulses from a digital input signal that is usually high or low. Glitches frequently occur on lines carrying signals from sources such as RF receivers. Electrical or in some cases even mechanical interference can trigger an unwanted glitch pulse from the receiver.

This design outputs a ‘1’ only when the current and previous N samples are ‘1’, and a ‘0’ only when the current and previous N samples are ‘0’. Otherwise the output is unchanged from its current value.

For more details on glitch filtering please see application note AN60024.

When to Use a Glitch Filter

A Glitch Filter component can be used with any digital input; it is typically associated with noise interference on a line that is connected to an input pin. Since it does a lot of work that could otherwise be done by firmware, it can be used when a firmware-based glitch filter solution is not practical.

Input/Output Connections

This section describes the input and output connections for the Glitch Filter component.

d – Input

The input that is sampled. It is usually connected to an input Pin component.

reset – Input

Active high synchronous reset that requires at least one rising edge of the clock. The reset input may be left floating with no external connection. If nothing is connected to the reset line, the component will assign it a constant logic 0.

clock – Input

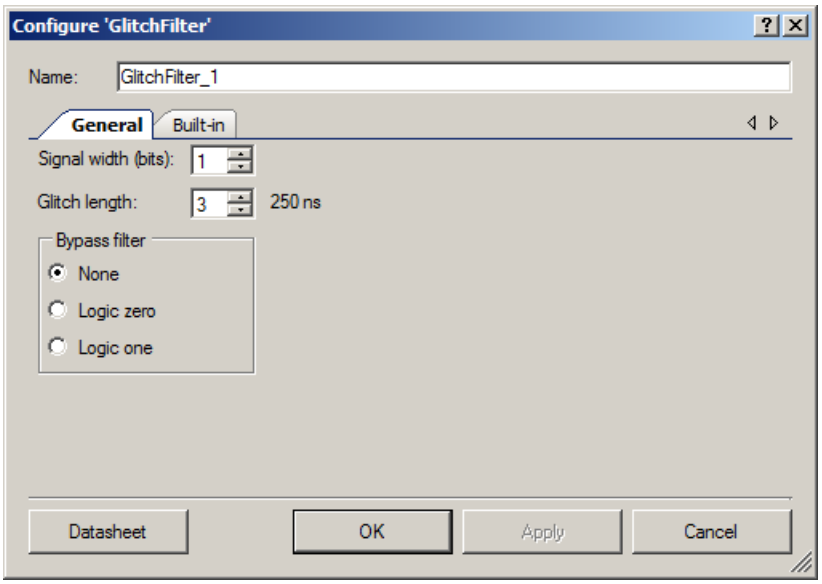
Clock used to sample the ‘d’ input. Its frequency depends on anticipated glitch pulse width and delay through the filter.

q – Output

This output is set to logic ‘1’ when the current and previous N samples are logic ‘1’, and logic ‘0’ when the current and previous N samples are ‘0’. Otherwise the output is unchanged from its current value.

Component Parameters

Drag a Glitch Filter component onto your design and double-click it to open the **Configure** dialog.



Signal Width

This parameter configures the signal width that will be filtered. This value can be set between 1 and 24. The default setting is **1 bit**.

Glitch Length

Determines the number of samples for which input has to be stable before being propagated to the output. This value can be set between 1 and 256 clock cycles. The default setting is **3**.

Bypass Filter

Specifies the logic level that will be propagated to the output immediately. The Bypass Filter **Logic zero** option makes the Glitch Filter output logic ‘0’ on the next clock cycle the ‘d’ input is at logic ‘0’. Similarly, there is an option for Bypass Filter **Logic one**. The default setting is **None**.

Clock Selection

There is no internal clock in this component. You must attach a clock source. This component operates from a single clock connected to the component.

Sample Firmware Source Code

PSoC Creator provides many example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start Page or **File** menu. As needed, use the **Filter Options** in the dialog to narrow the list of projects available to select.

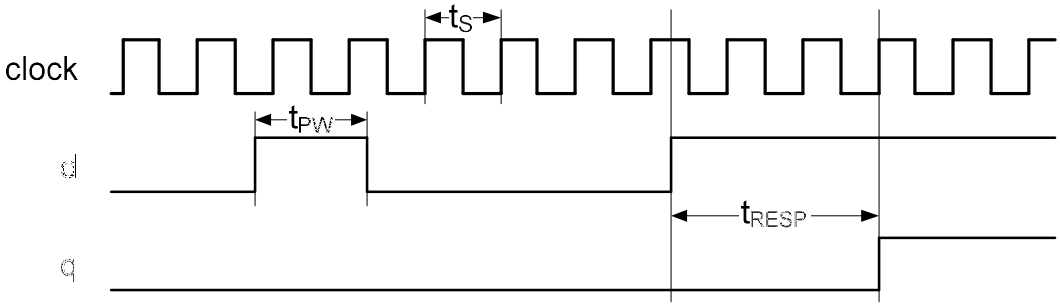
Refer to the “Find Example Project” topic in the PSoC Creator Help for more information..

Functional Description

The glitch filter component outputs a ‘1’ only when the current and previous N samples are ‘1’, and a ‘0’ only when the current and previous N samples are ‘0’. Otherwise, the output is unchanged from its current value. Figure 1 shows typical glitch filtering operation for N = 2 samples. The time between N successive samples must be greater than maximum pulse width (t_{PW}) that can be filtered.

Note that the response time (t_{RESP}), or filter delay, is between N and N + 1 sample periods (t_S).

Figure 1. Timing Diagram



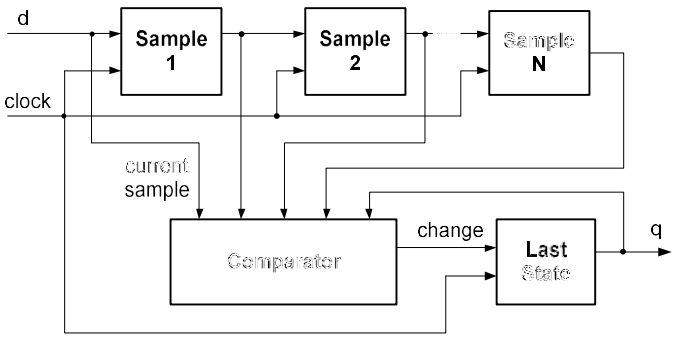
For more details on glitch filtering please see application note AN60024.

Block Diagram and Configuration

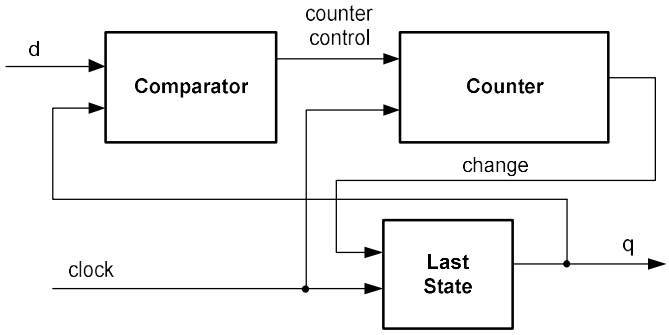
The Glitch Filter component implementation depends on programmable filter length parameter. If the parameter value is ≤ 8 the implementation uses the PLD portion of one or more of the UDBs. If the filter length is between 9 and 256 samples, the component is implemented on datapath elements of the UDB Array. Figure 2 shows this implementation.

Figure 2. Block Diagram

Glitch length ≤ 8



8 < Glitch length ≤ 256



Resources

The Glitch Filter component is placed throughout the UDB array. The component utilizes the following resources.

Configuration	Resource Type					
	Datapath Cells	Macrocells	Status Cells	Control Cells	DMA Channels	Interrupts
Glitch Length ≤ 8	–	N * (L + 1) [1]	–	–	–	–
8 < Glitch Length ≤ 256	N	N	–	–	–	–

1. N – Signal width; L – Filtering length.

DC and AC Electrical Characteristics

Specifications are valid for –40 °C ≤ T_A ≤ 85 °C and T_J ≤ 100 °C, except where noted.
Specifications are valid for 1.71 V to 5.5 V, except where noted.

AC Characteristics

Parameter	Description	Min	Typ	Max	Unit
f_{CLOCK}	Component clock frequency				
	Glitch Length ≤ 8	–	–	67	MHz
	8 < Glitch Length ≤ 256	–	–	40	MHz

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
2.0.a	Added timing diagram in Functional Description section	Clarification of component operation.
2.0	Complete redesign. Glitch Filter v2.0 is NOT AT ALL backwards compatible with previous versions.	Updated for compatibility with PSoC Creator v2.1 to be included as standard library component. Added ‘Signal width’, ‘Glitch Length’ and ‘Bypass Filter’ parameter to allow configuring of the component.
1.30	Updated for compatibility with PSoC Creator v2.0, to be used as a concept component included in PSoC Creator.	
1.2	Updated for compatibility with PSoC Creator v1.0.	
1.1	Updated for compatibility with PSoC Creator beta5. Updated data sheet format.	
1.0	Initial design.	