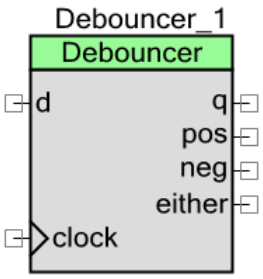
Debouncer
1.0

Features

- Eliminates unwanted oscillations on digital input lines



General Description

Mechanical switches and relays tend to make and break connections for a finite time before settling down to a stable state. Within this settling time, the digital circuit can see multiple transitions as the switch contacts bounce between make or break conditions.

The Debouncer component takes an input signal from a bouncing contact and generates a clean output for digital circuits. The component will not pass the signal to the output until the predetermined period of time when the switch bouncing settles down. In this way, the circuit will respond to only one pulse generation performed by the pressing or releasing of the switch and not several state transitions caused by contact bouncing.

For more details on switch debouncing please see application note AN60024.

When to Use a Debouncer

A Debouncer component can be used to debounce the input digital signal from most types of switches. Since it does a lot of work that could otherwise be done by firmware, it can be used when a firmware-based switch debouncing solution is not practical.

Input/Output Connections

This section describes the input and output connections for the Debouncer component. An asterisk (*) in the list of I/Os indicates that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

d – Input

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

clock – Input

Clock used to sample the ‘d’ input. Its frequency depends on anticipated transition time of the signal. This must be high enough that the system is responsive, but low enough to sample only once during an input transition. A clock between 10 and 200 Hz is expected to suffice.

q – Output

Filtered output value. Sampled value of ‘d’ input at each clock rising edge.

pos – Output *

Positive edge. This output goes high one clock cycle after a positive edge is detected and stays high for one clock cycle. Displays if a **Positive edge** is selected under the **Edge type** parameter.

neg – Output *

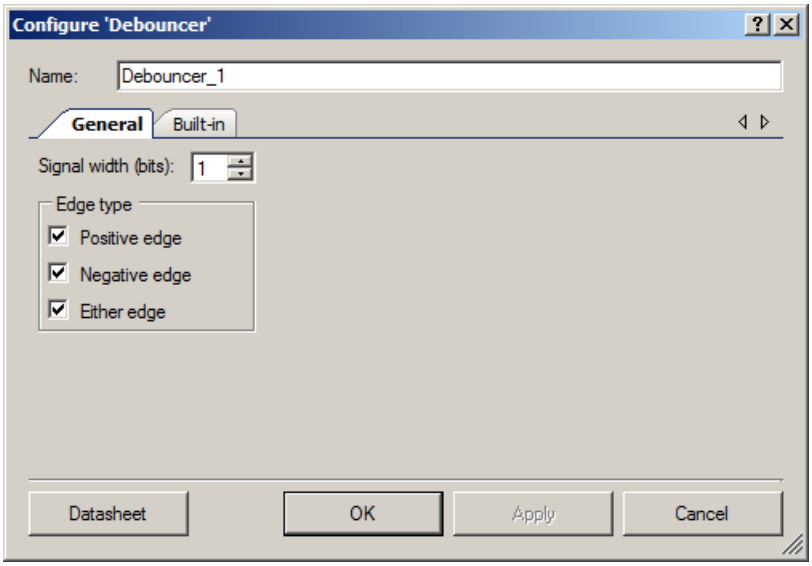
Negative edge. This output goes high one clock cycle after a negative edge is detected and stays high for one clock cycle. Displays if a **Negative edge** is selected under the **Edge type** parameter.

either – Output *

Either edge. This output goes high one clock cycle after either a positive or negative edge is detected and stays high for one clock cycle. Displays if a **Either edge** is selected under the **Edge type** parameter.

Component Parameters

Drag a Debouncer 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 32. The default setting is **1 bit**.

Edge Type

Determines whether each of the ‘pos’, ‘neg’ and/or ‘either’ edge detection is enabled for the component. By default the detection of all edges is enabled.

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 Debouncer component provides a hardware method of eliminating the oscillations that occur on transitions of an input digital signal that comes from switches and relays when their contacts touch or release.

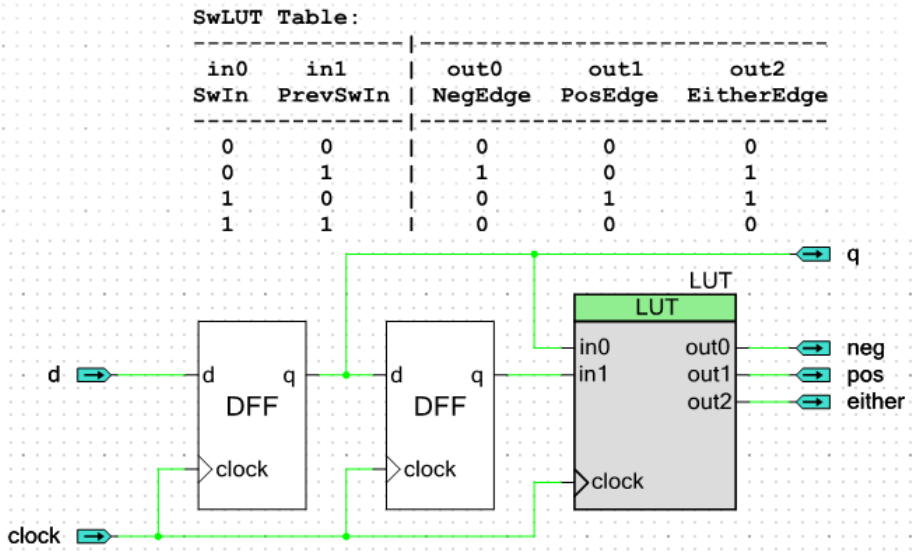
For more details on switch debouncing please see application note AN60024.

Block Diagram and Configuration

The N-bit debouncer is equivalent to N 1-bit debouncers. The implementation uses the PLD portion of one or more of the UDBs. Figure 1 shows the circuit for 1-bit deboncer.

The clock input is used to sample the switch input ‘d’. As noted previously, a clock frequency of 10 to 200 samples per second is usually appropriate. The first DFF is used to sample the switch input. The second DFF stores the previous sample, and the lookup table (LUT) component does edge detection by comparing the two samples.

Figure 1. Block Diagram



Resources

The Debouncer component is placed in the PLD portion of one or more of the UDBs. The component utilizes five macrocells per input signal bit in maximal configuration.

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	–	200	–	Hz

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
1.0	Version 1.0 is the first release of the Debouncer component	

© Cypress Semiconductor Corporation, 2012. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.