

Window Discriminator

Author: Vikram. C. R
Associated Project: Yes
Associated Part Family: CY8C25xxx
PSoC Designer Version: 4.1
Associated Application Notes: AN2108

Summary

This Application Note describes the implementation of a Window Discriminator using assembly language and an 8-pin PSoC.

Introduction

A Window Discriminator is a circuit similar to a Schmitt Trigger. When the input signal crosses the lower threshold (V_{it}) without crossing the upper threshold (V_{ut}), a logic level output (typically 10 μ s) is sent as an output at the point where the trailing side of the input signal crosses the lower threshold, indicating a valid input signal. The logic level output is typically used to count the number of valid input signals.

Functionality

Figure 1 shows the function of a Window Discriminator. For input signals 1, 3 and 4, the rising side of the signals cross V_{it} without crossing V_{ut} . Hence, an output pulse is output at the crossover point (the point where the falling side of the input signal crosses V_{it}). For signal 2, though the leading part of the input signal crosses V_{it} , it also crosses V_{ut} . This is not an acceptable condition. Therefore, there is no pulse trigger. For signal 5, the leading input fails to cross V_{it} . Therefore, there is no pulse trigger.

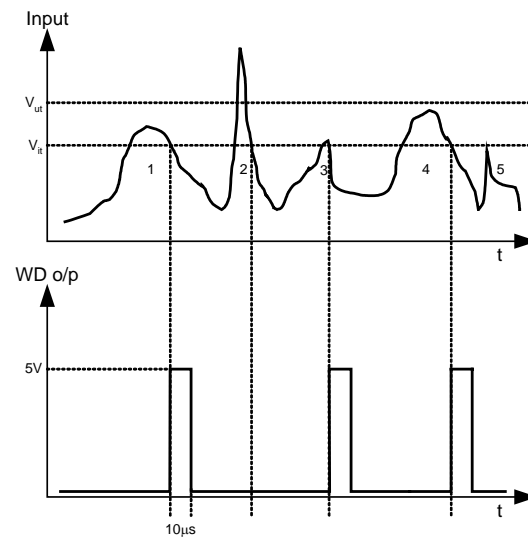


Figure 1. Window Discriminator Function

Design Implementation

Figure 2 shows the Window Discriminator implementation in the Device Editor of PSoC Designer.

It uses two comparators: the CMPPRG_1 block to set the upper threshold value and the CMPPRG_2 block to set the lower threshold value. The upper and lower thresholds can be changed by modifying the reference values in the comparators. The threshold value computation is governed by the following equation:

$$V_{\text{thresh}} = V_{\text{lowlimit}} + (V_{\text{upperlimit}} - V_{\text{lowlimit}})(\text{ref_val})(1)$$

In the Device Editor, V_{lowlimit} is set to V_{ss} (0V) and $V_{\text{upperlimit}}$ is set to V_{dd} (5V). Therefore, the modified equation for threshold voltage is:

$$V_{\text{thresh}} = (V_{\text{dd}})(\text{ref_val}) \quad (2)$$

The reference value is selectable in the comparator block of the PSoC. Table 1 lists the settings used in the example application.

Table 1. Comparator Settings

Comparator	refval (Reference Value)	V_{thresh}
CMPPRG1	0.25	1.25V (Upper Threshold)
CMPPRG2	0.188	0.94V (Lower Threshold)

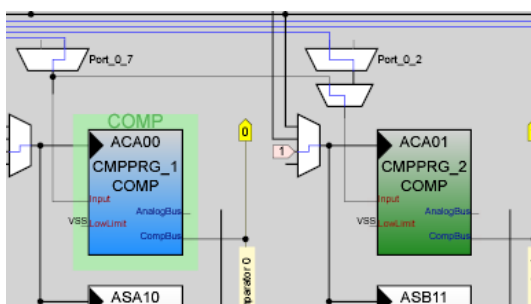


Figure 2. Window Discriminator Implementation in PSoC Designer

The input signal is on P0[7]. The Window Discriminator output is available on P0[4].

Software Implementation

A flowchart of the implementation of the Window Discriminator is shown in Figure 3.

Initially, comparator settings are entered and the comparator is powered on and started at high power.

In the initialization routine, a register that indicates whether the signal has crossed the upper threshold is initialized. The register is called 'uthrchk flag'. Also, the output of the discriminator is set low.

The control then passes to the lower threshold check routine, which checks to see if the input is greater than the lower threshold. If it is not, the control remains in the same routine provided the 'uthrchk flag' is not set. This means that if the control has never gone into the upper threshold routine when the signal is less than the lower threshold, the control remains in the lower threshold routine. Otherwise, the control passes to the upper threshold check routine.

In the upper threshold routine, there is a check to see if the signal crosses the upper threshold. If it has crossed, the control goes back to the initialization routine since it has failed a necessary condition. However, if the signal is less than the upper threshold, the control goes back to the lower threshold routine. This process of the control jumping between the lower and upper threshold routines continues until the trailing side of the input signal crosses the lower side of the input signal. At this point, the output pulse is triggered in the pulse trigger routine. The control eventually goes back to the initialization routine to process the next signal.

Noise Sensitivity

If there are glitches in the input signal or if the signal is very noisy, there may be false triggers during the input cycle. So instead of a single pulse trigger per cycle, there may be multiple triggers. This limitation can be avoided by lowering the lower threshold value. However, if the signal quality is very poor, it may override the lower threshold as well, which will result in triggering at false locations. This is a limitation of this implementation.

An additional issue is with oscillations present at the output of the low threshold comparator, when the signal crosses the threshold level. To reduce the possibility of false output pulses, a small debounce delay while sampling the output of the comparator has been introduced.

When the input signal is a fast changing signal and goes above the threshold before the debouncing is finished, the condition may not be detected and an output pulse may be missed. For such signals, the debouncing delay should be reduced or completely eliminated. A hysteresis comparator may help eliminate the oscillations at the output. Please refer to AN2108, "Standard – Hysteresis Comparator with PSoC." In this case, the debouncing routine can be bypassed.

Conclusion

The Window Discriminator is a very important function used in many applications. The traditional way of implementing it is through circuitry. The implementation on the 8-bit PSoC shows that there is a considerable reduction in size without any deterioration in performance.

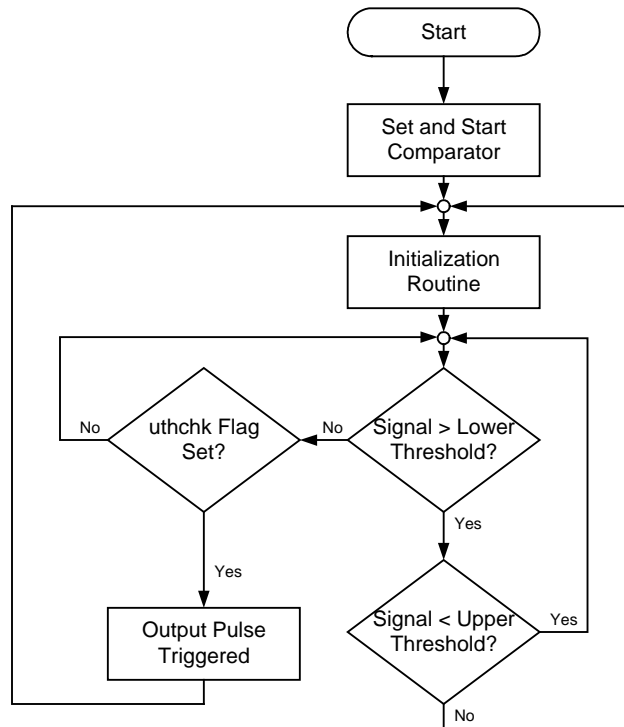


Figure 3. Window Discriminator Flowchart

About the Author

Name: Vikram. C. R
Title: Embedded Systems Engineer
Contact: vikram_cr@hotmail.com

Cypress MicroSystems, Inc.
2700 162nd Street SW, Building D
Lynnwood, WA 98037
Phone: 800.669.0557
Fax: 425.787.4641

<http://www.cypress.com/> / <http://www.cypress.com/support/mysupport.cfm>

Copyright © 2003-2004 Cypress MicroSystems, Inc. All rights reserved.

PSoC™, Programmable System-on-Chip™, and PSoC Designer™ are trademarks of Cypress MicroSystems, Inc.

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

The information contained herein is subject to change without notice. Made in the U.S.A.