Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

AN2108

PSoC® 1 – Implementing a Hysteresis Comparator

Authors: Dave Van Ess, Mohana Koteeswaran

Associated Project: Yes

Associated Part Family: CY8C24x23, CY8C24x94, CY8C27x43, CY8C28xxx, CY8C29x66

Software Version: PSoC Designer™ 5.4

Related Application Notes: None

To get the latest version of this application note, or the associated project file, please visit www.cypress.com/go/AN2108.

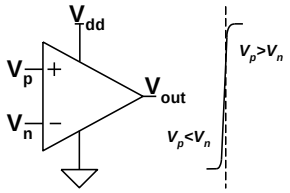
AN2108 explains multiple implementations of a hysteresis comparator using PSoC® 1. Hysteresis is necessary to produce a glitch-free comparator output when there is noise in the input signals.

Contents

1 Introduction

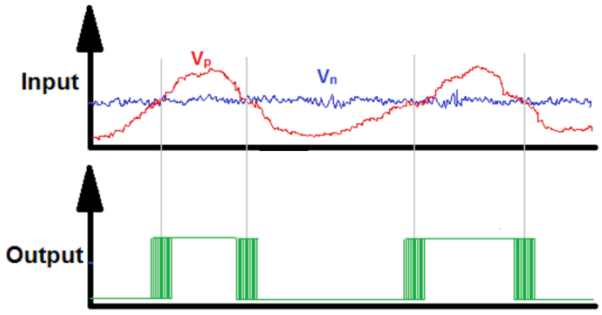
The comparator is the most fundamental building block of a mixed-signal design. It is essentially a differential amplifier with an extremely high open loop gain. Figure 1 illustrates a representation of a comparator and its output response

Figure 1. Comparator Response



When V_p is greater than V_n, the output is driven to the positive supply rail; conversely, when V_p is less than V_n, the output is driven to the negative supply rail. A slight amount of noise in V_p or V_n can cause the comparator output to oscillate, as shown in Figure 2. This is a common occurrence when the input signals vary slowly.

Figure 2. Comparator Output with Noisy Inputs



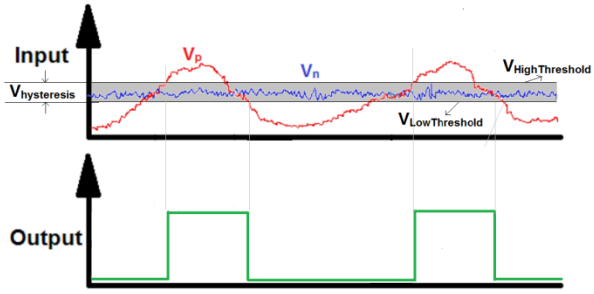
One solution is to get rid of noise in the V_p and V_n signals using a filter. This requires extra hardware resources, however, and it is not possible to remove the noise completely. A more practical solution is to include hysteresis in a comparator. Hysteresis creates two thresholds—one for the output to switch from low to high and another for the output to switch from high to low.

The hysteresis voltage is defined as the difference in these thresholds:

$$V_{hysteresis} = V_{HighThreshold} - V_{LowThreshold} \quad \text{Equation 1}$$

Hysteresis voltage $V_{hysteresis}$ should be kept greater than the peak-to-peak value of the noise signal to ensure a glitch-free output, as shown in Figure 3.

Figure 3. Comparator Output with Hysteresis



2 Hysteresis Comparator Design

There are several ways to implement a hysteresis comparator in PSoC 1 using the following:

- Continuous time (CT) analog block
- Switched capacitor (SC) analog block

The CT block includes an opamp and a resistor array. This makes the analog block useful for functions such as a programmable gain amplifier (PGA) and comparators. The SC block includes an opamp with a switched capacitor network around it. This architecture enables the implementation of an integrator, a differentiator, a filter, an amplifier, a DAC, a comparator, and many more.

For details on the PSoC 1 architecture, see the Technical Reference Manual. This application note explains hysteresis comparator design using both the CT and SC blocks.

2.1 Comparator User Module in PSoC Designer™

PSoC Designer™ provides the Comparator (COMP) User Module, as shown in Figure 4. This module implements two different hysteresis comparator architectures, COMPH and COMPDH, as shown in Figure 5.

Figure 4. COMP User Module

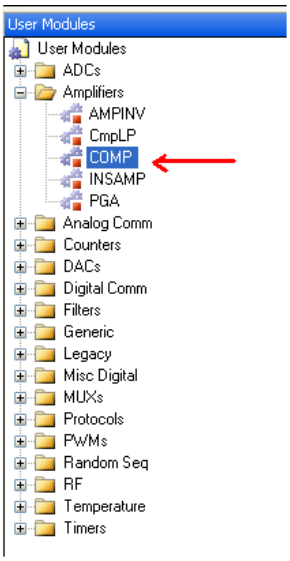
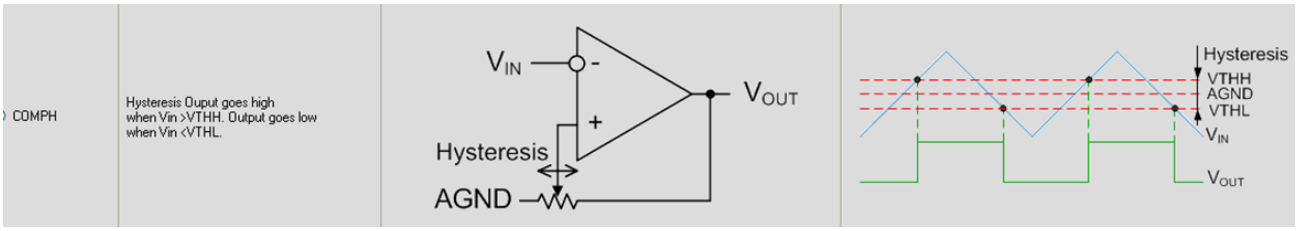


Figure 5. Hysteresis Comparator Implementations in the COMP User Module



Following are some highlights of the COMPH and COMPDH implementations:

- The COMPH and COMPDH hysteresis comparator's inputs are limited to continuous time inputs or filter outputs. The feedback path for the hysteresis limits is "analog" and is not latched; therefore, noncontinuous inputs do not maintain the input state above the hysteresis high threshold.
- CT blocks are used. COMPH consumes one CT block, and COMPDH consumes two CT blocks.
- The reference signal in COMPH is connected to AGND (analog ground), whereas in COMPDH, the reference can be any external signal.

- They use an internal resistor array for setting the hysteresis. Due to the limited number of resistor taps, the hysteresis voltage can be set to V_{DD} * (1/16, 2/16, 3/16…15/16).

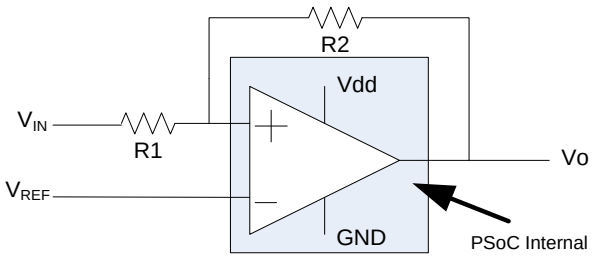
The user module implementation offers a cost-effective solution with an internal resistor array and low pin usage. But this comes at the cost of the precision with which the hysteresis can be set. The user module provides a minimum hysteresis voltage of V_{DD} / 16. For V_{DD} = 5 V, the hysteresis voltage is 0.312 mV. The following sections explain alternative ways to implement a hysteresis comparator that provide a lower hysteresis voltage:

- Design 1: Hysteresis Comparator Using CT Block and External Resistors shows a hysteresis comparator implementation using a CT block and external resistors. This architecture allows the hysteresis to be set precisely.
- Design 2: Hysteresis Comparator Using SC Block shows a hysteresis comparator implementation using an SC block. This implementation is useful when no CT blocks are free.
- Design 3: Comparator with Independently Controllable Hysteresis shows a unique technique to implement a comparator with independently controllable hysteresis thresholds.

3 Design 1: Hysteresis Comparator Using CT Block and External Resistors

Figure 6 shows the basic design of a hysteresis comparator.

Figure 6. Basic Hysteresis Comparator Design



The hysteresis voltage for the design is calculated as follows:

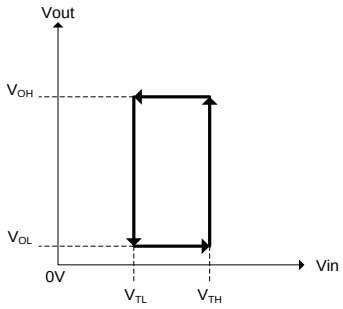
$Hyst = (V_{TH} - V_{TL})$ where

$$V_{TH} = \frac{(R_1 + R_2)V_{REF}}{R_2} - \frac{R_1 V_{OL}}{R_2} \qquad \text{Equation 2}$$

$$V_{TL} = \frac{(R_1 + R_2)V_{REF}}{R_2} - \frac{R_1 V_{OH}}{R_2} \qquad \text{Equation 3}$$

V_{OL} and V_{OH} are the lower and higher limits of the comparator output. Figure 7 shows the design's transfer curve.

Figure 7. Transfer Curve

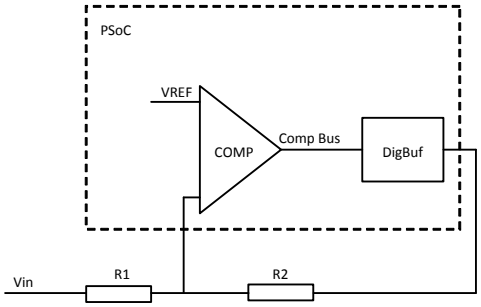


Note that the hysteresis thresholds are symmetric around V_{REF} when V_{REF} is equal to $V_{DD}/2$, but not at other V_{REF} voltages.

3.1 PSoC Implementation

You can implement the design in Figure 6 in PSoC using the COMP and Digital Buffers (DigBuf) User Modules, as shown in Figure 8.

Figure 8. Hysteresis Comparator Using a CT Block



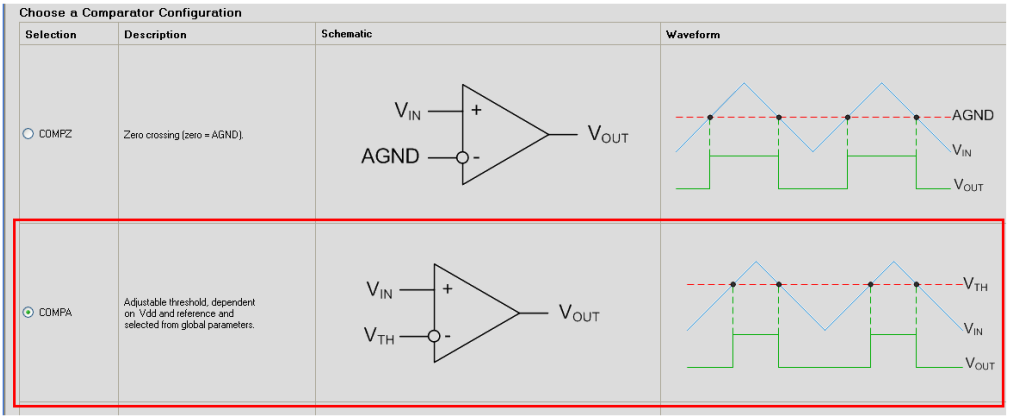
The comparator output is connected to the comparator bus. This output is routed to a pin through a DigBuf User Module. R1 and R2 are the external resistors that set the hysteresis. The V_{REF} voltage can be obtained using a resistor array in the CT block.

3.2 Design Example

This section presents an example design of a hysteresis comparator using a CY8C27443 PSoC 1 device with 50 mV of hysteresis voltage added to a reference voltage of 1.25 V. Follow these steps to configure PSoC 1:

1. Place a COMP User Module in the design.
2. Select COMPA as the configuration using the **Selection** option, as shown in Figure 9. COMPA allows the input to be taken from a pin and the reference to be set internally.

Figure 9. Comparator Mode Selection

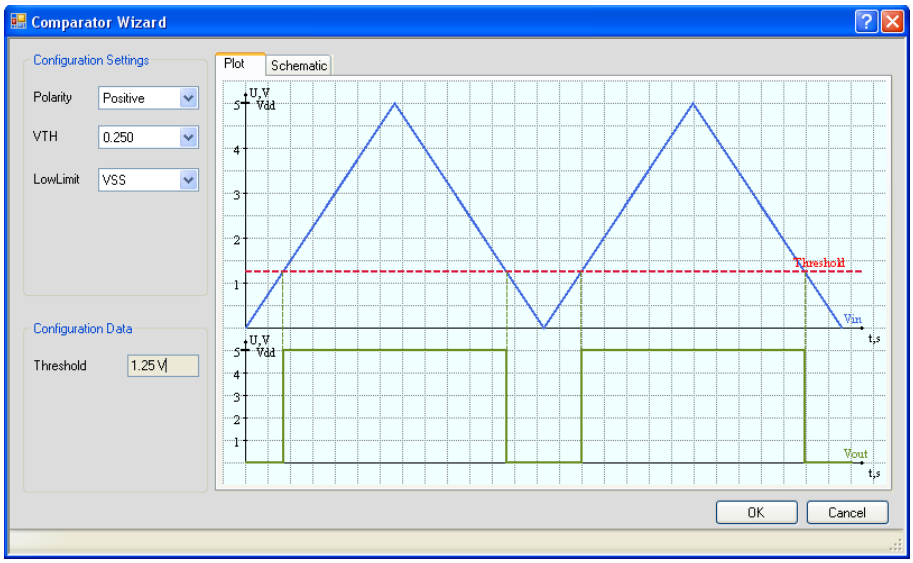


3. Open the comparator wizard. Configure the V_{TH} and LowLimit parameters to 0.250 and V_{SS}, respectively, to set the threshold voltage to 1.25 V. See Figure 10.

   Here is the equation for a comparator threshold voltage:

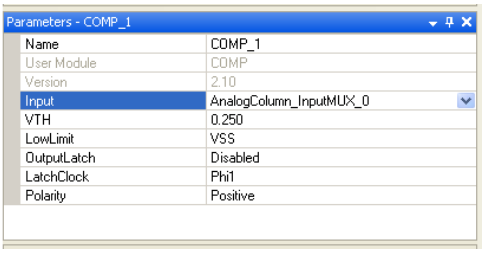
$$\boldsymbol{Vthreshold = V_{TH}\,(V_{DD} - LowLimit) + LowLimit}$$

Figure 10. Comparator Configuration Using Wizard



4. Set **Input** to "AnalogColumn_InputMUX_0" in the user module properties, as shown in Figure 11. Leave the rest of the settings at the defaults.

Figure 11. Comparator Properties

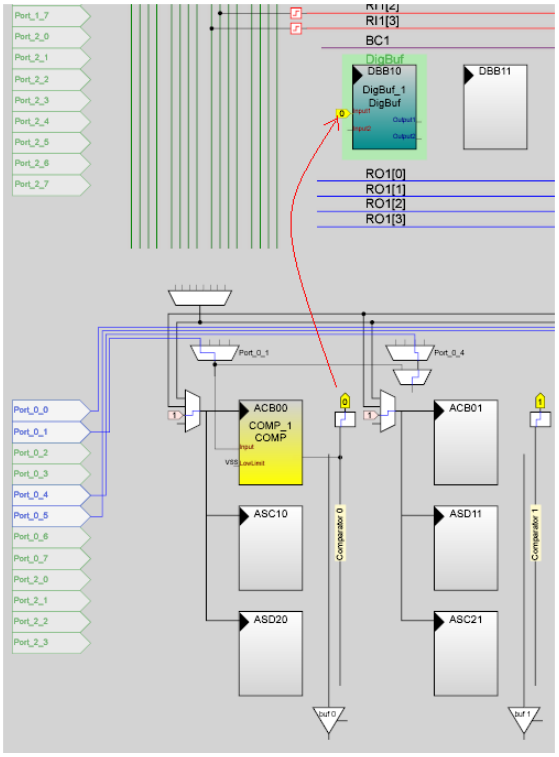


5. Set the comparator input by clicking on "AnalogColumn_InputMUX_0." In this example, the input is set to P0[1], as shown in Figure 12.
6. Place a DigBuf User Module in the design. Set its input1 to the comparator output.
7. Route the DigBuf User Module output to a pin. In this project, the output is routed to pin P1[4].
8. Select external resistors: Solving Equation 2 and Equation 3 results in Equation 4:.

$$\boldsymbol{R_2 = \frac{(V_{OH} - V_{OL})R_1}{Hyst}}$$ Equation 4

For (R1 + R2) >> 1 kΩ, V_{OH} and V_{OL} for a digital output can be considered the same as V_{DD} and GND respectively. For this application, use an R1 of 10 kΩ and V_{DD} = 5 V. The required hysteresis voltage is 50 mV, which yields R2 = 1 MΩ.

Figure 12. User Module Placement



The hysteresis threshold voltages can be calculated using Equation 2 and Equation 3 as follows:

$$V_{TH} = (1M + 10K)\frac{1.25V}{1M} - 0\frac{10K}{1M} = 1.262 \qquad \text{Equation 5}$$

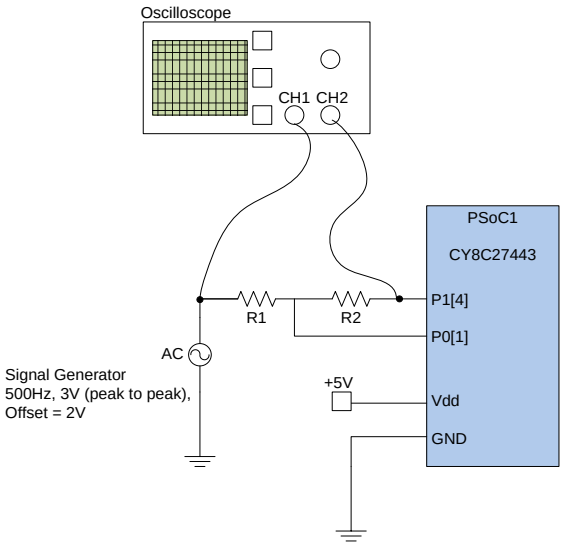
$$V_{TL} = (1M + 10K)\frac{1.25V}{1M} - 5\frac{10K}{1M} = 1.2125 \qquad \text{Equation 6}$$

9. The following firmware is required to start the user modules:

```
/* part specific constants and macros */
#include <m8c.h>

/* PSoC1 API definitions for all User Modules */
#include "PSoCAPI.h"

void main(void)
{
    /* Start comparator user module */
    COMP_1_Start(COMP_1_HIGHPOWER);

    /* Start Digital buffer */
    DigBuf_1_Start();

    while(1);
}
```

The PSoC Designer project "HysteresisComp1," based on the CY8C27443 PSoC 1 device, is provided. You can test the project with the external connections shown in Figure 13.

Figure 13. Test Setup

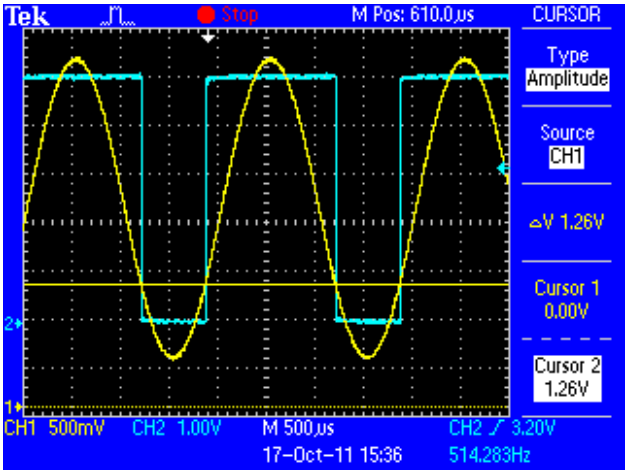


3.2.1 Test Waveforms

Feed a 3-V (peak to peak), 2-V offset sinusoidal signal of any suitable frequency as an input. Figure 14 shows the waveforms obtained on the oscilloscope.

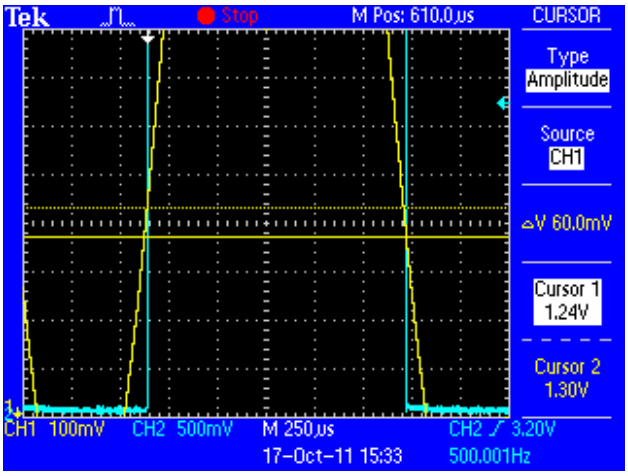
CH1: Input CH2: Output

Figure 14. Test Waveforms



Notice the reference of approximately 1.25 V in Figure 14. The hysteresis thresholds can be observed in Figure 15.

Figure 15. Zoomed Version



The hysteresis voltage configured in the design is 50 mV. Figure 15 shows a hysteresis voltage of about 60 mV. This difference is due to sources such as tolerance of the external resistors, V_{OL} and V_{OH} variations, and supply variations.

4 Design 2: Hysteresis Comparator Using SC Block

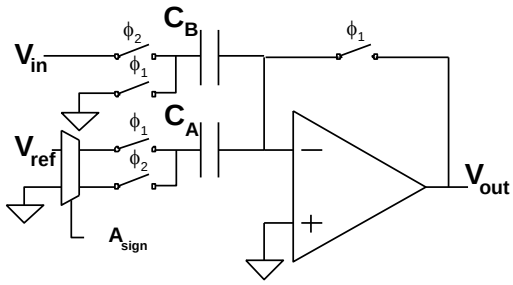
It is also possible to make an SC block comparator with hysteresis. The SC block comparator's threshold level is determined by the ratio of two internal capacitors. This produces a comparator with a hysteresis that

- Has no external components
- Allows the hysteresis thresholds to be easily changed in the firmware

To learn about SC blocks, see the application note AN2041 – Understanding PSoC 1 Switched Capacitor Analog Blocks.

Figure 16 shows an SC block configured as a programmable threshold comparator.

Figure 16. Programmable Threshold SC Comparator



The C_B path is a negative input, so this topology builds an inverting comparator. That means the output is low when the input exceeds some threshold.

The output is low when the charge transferred in the C_A path is greater than the charge supplied in the C_B path.

$$\boldsymbol{V_{out} = LOW : V_{in}C_B > A_{sign}V_{ref}C_A} \qquad \text{Equation 7}$$

The threshold is defined in Equation 8:.

$$\boldsymbol{V_{Threshold} = A_{sign}\left(V_{ref} - AGND\right)\frac{C_A}{C_B}} \qquad \text{Equation 8}$$

where AGND is the analog ground voltage.

Thus, the threshold voltage of an SC block comparator can be changed by altering the capacitor ratio.

To implement hysteresis, the threshold voltage must be made a function of Vout. The type C SC block in PSoC 1 allows the threshold to be modulated depending on Vout.

Figure 17 shows the comparator with hysteresis. With the initial A_{sign} set to negative, the high output voltage causes a positive threshold level to be generated. When the input voltage exceeds this threshold, the output goes low. This low output causes a negative threshold level to be generated. See Figure 18 for the waveforms.

Equation 9 and Equation 10 provide the hysteresis thresholds.

$$V_{TH} = AGND + (V_{ref} - AGND)\frac{C_A}{C_B} \; : \; V_{out} = HIGH \qquad \text{Equation 9}$$

$$V_{TL} = AGND - (V_{ref} - AGND)\frac{C_A}{C_B} \; : \; V_{out} = LOW \qquad \text{Equation 10}$$

The hysteresis voltage is given by

$$V_{Hysteresis} = 2\,(V_{ref} - AGND)\frac{C_A}{C_B} \qquad \text{Equation 11}$$

Note that the CA and CB capacitance can be varied from 0 units to 31 units; 1 unit is equal to 70 fF (typical).

Figure 17. Comparator with Hysteresis Added

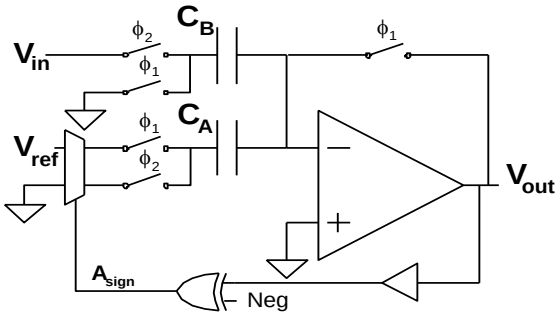
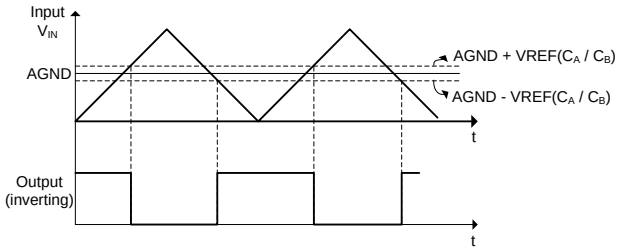


Figure 18. Hysteresis Thresholds

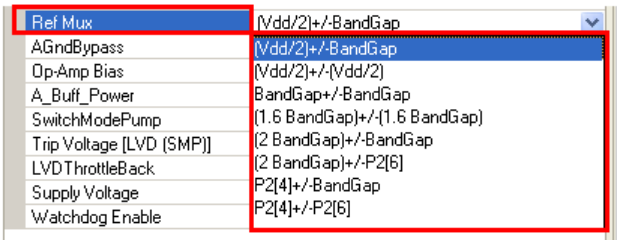


As given in Equation 9 and Equation 10,, the hysteresis threshold depends on

- Analog ground (AGND) voltage
- V_{REF} voltage
- C_A/C_B cap ratio

AGND is configured using the Ref Mux setting in the PSoC Designer Global Resources, as shown Figure 19.

Figure 19. Ref Mux Options



The values shown in Figure 19 are defined in the following format:

$$AGND \pm reference$$

The AGND voltage can be set to $V_{DD}/2$, Bandgap (1.3 V), 1.6*Bandgap (2.08 V), 2*Bandgap (2.6 V), and external voltage at pin P2[4].

In addition to AGND, the PSoC 1 reference generator generates two signals:

$$REFHI = AGND + reference$$
$$REFLO = AGND - reference$$

The signals REFHI and REFLO along with AGND can be used as inputs to the analog blocks.

- The VREF signal can be obtained from several sources: the CT block, a DAC built using the SC block, an internal reference generator, or a pin.
- CA and CB can be configured individually from 0 units to 31 units.

4.1 Design Example

This section presents an example design of a hysteresis comparator using a CY8C27443 PSoC 1 device with the reference voltage equal to 1.3 V.

The example uses REFHI (2.6 V) for the V_{REF} input. The minimum possible hysteresis voltage is as follows:

$$\Delta VT = \pm(Vref - AGND) \times C_{A\_min} / C_{B\_max}$$

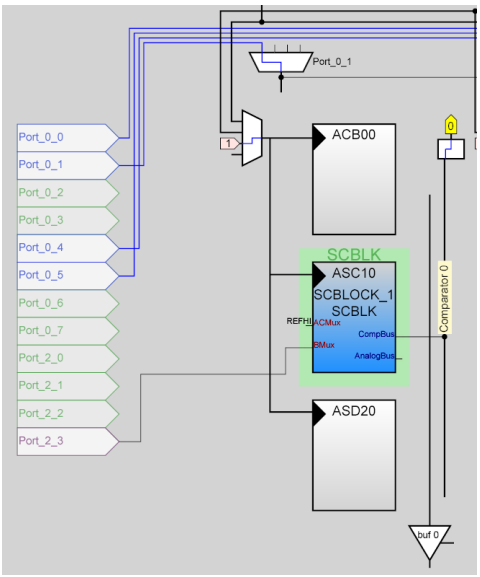
$$\Delta VT = \pm(2.6V - 1.3V) \times \frac{1}{31} = \pm 84mV$$

Configure the C_A/C_B ratio equal to 1/6, which gives the hysteresis thresholds of ± 0.216 V around AGND.

Then follow these steps:

1. Place an SCBLOCK User Module on a type C SC block. You can identify a type C analog block by the name “ASCxx.”
2. The V_{REF} and V_{IN} inputs of the SC block in Figure 14 are represented as ACMUX and BMUX respectively in PSoC Designer, as shown in Figure 20. Set the ACMUX input, that is, V_{REF}, to REFHI. Route the BMUX input to a pin—in this example, P2[3]. The comparator bus is enabled, which allows the modulator to access the comparator output.

Figure 20. SC Block Placement and Routing



3. Configure the SC block properties, as shown in Figure 21.

Figure 21. SC Block Properties

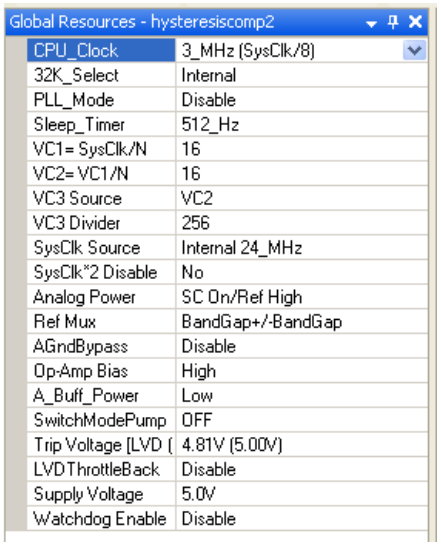
Parameters - SCBLOCK_1

Name	SCBLOCK_1
User Module	SCBLOCK
Version	2.4
FCap	16
ClockPhase	Norm
ASign	Neg
ACap	5
ACMux	REFHI
BCap	30
AnalogBus	Disable
CompBus	ComparatorBus_0
AutoZero	On
CCap	0
ARefMux	AGND
FSW1	Off
FSW0	Off
BMux	Port_2_3
Power	High

With FSW1 and FSW0 “Off” and AutoZero “On,” the block is configured as a comparator. Set ACap and BCap to 5 and 30 respectively to get a ratio of 1/6. Set the other parameters as shown in Figure 21.

4. Place a DigBuf User Module to route the comparator output to a pin. In this project, the comparator output is routed to pin P1[4].
5. Configure the global parameters, as shown in Figure 22.

Figure 22. Project Global Parameters



The analog column clock is set to VC1, as shown in Figure 20. The column clock frequency should always be less than 8 MHz. Set the VC1 divider to 16 in the Global Resources, as shown in Figure 22; this yields the VC1 clock frequency 24 MHz/16 = 1.6 MHz. Set Ref Mux to Bandgap ± Bandgap, which configures the AGND as Bandgap (1.3 V) and REFHI as 2.6 V. Configure the other parameters based on the application requirements.

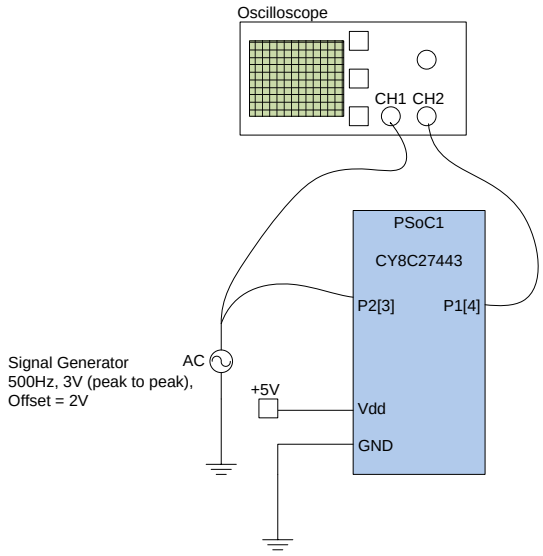
6. Write the firmware as follows:

```
/* part specific constants and macros */
#include <m8c.h>

/* PSoC1 API definitions for all UM */
#include "PSoCAPI.h"

void main(void)
{
    /* Start SC block */
    SCBLOCK_1_Start(SCBLOCK_1_HIGHPOWER);

    /* Start Digital Buffer */
    DigBuf_1_Start();

    /* Enable modulator */
    AMD_CR0|=0x4;

    while(1);
}
```

Configuring AMD_CR0 allows the comparator output to be used for modulation.

The PSoC Designer project "HysteresisComp2," based on the CY8C27443 PSoC 1 device, is provided. You can test the project with the external connections, as shown in Figure 23.

Figure 23. Test Setup

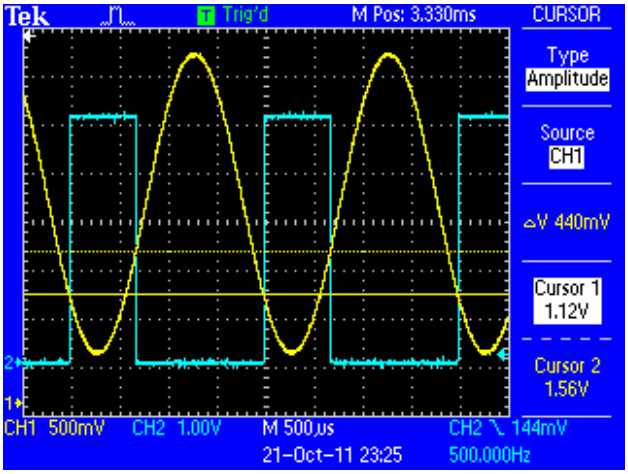


4.1.1 Test Waveforms

Feed a 3-V (peak to peak), 2-V offset sinusoidal signal of any suitable frequency as an input. Figure 24 shows the waveform obtained on the oscilloscope.

CH1: Input CH2: Output

Figure 24. Test Waveform



Notice the hysteresis voltage of 440 mV, which is close to the target threshold of ±216 mV (432 mV).

5 Design 3: Comparator with Independently Controllable Hysteresis

Many circuit topologies generate the hysteresis thresholds equidistant from some reference. Sometimes, it is desirable to have a topology that allows an independent alteration of the individual thresholds.

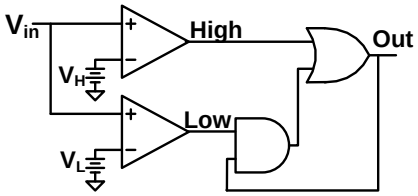
A simple method to allow two isolated threshold levels is to have two actual comparators and some digital logic.

Equation 12 is the Boolean representation of the hysteresis requirements.

$$\boldsymbol{Out} = \boldsymbol{High} + (\boldsymbol{Out} * \boldsymbol{Low}) \qquad \text{Equation 12}$$

Figure 25 shows the implementation.

Figure 25. Hysteresis Comparator Design



5.1 PSoC 1 Implementation

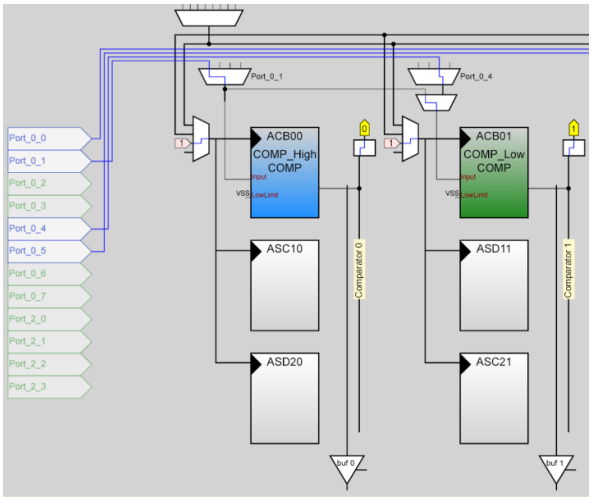
A PSoC 1 solution requires two parts:

- Two programmable comparators
- Digital logic

You can implement the comparators using two COMP User Modules. Follow these steps to implement the design in PSoC 1:

1. Place two COMP User Modules within the COMPA configuration. The comparators are named “COMP_High” and “COMP_Low,” as shown in Figure 26.
2. Take the comparator inputs from the same pin. In this project, P0[1] is taken as an input pin.

Figure 26. Comparator Placement and Routing



3. Configure the comparator reference voltages in the user module properties or using the comparator wizard. Set the Comp_High threshold to 1.56 V and the Comp_Low threshold to 1.25 V. This gives the hysteresis a voltage of 310 mV. See Figure 27 and Figure 28.

Figure 27. “Comparator High” Wizard

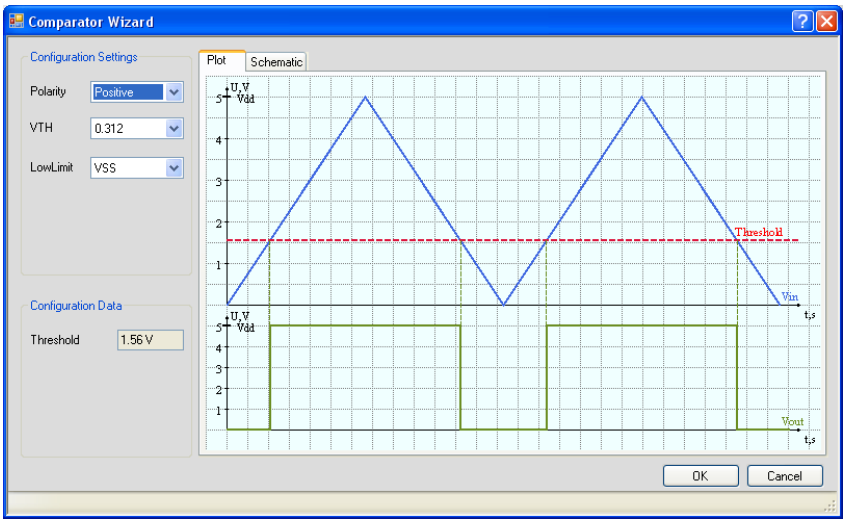
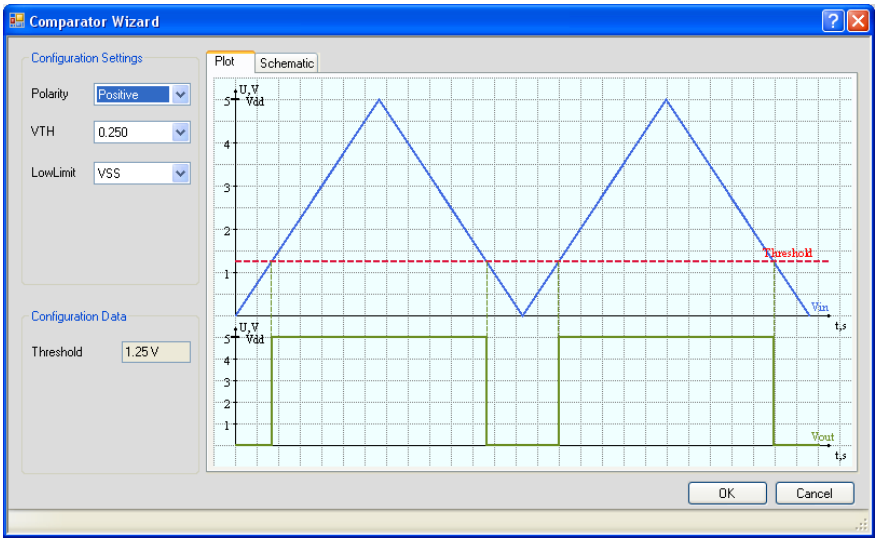
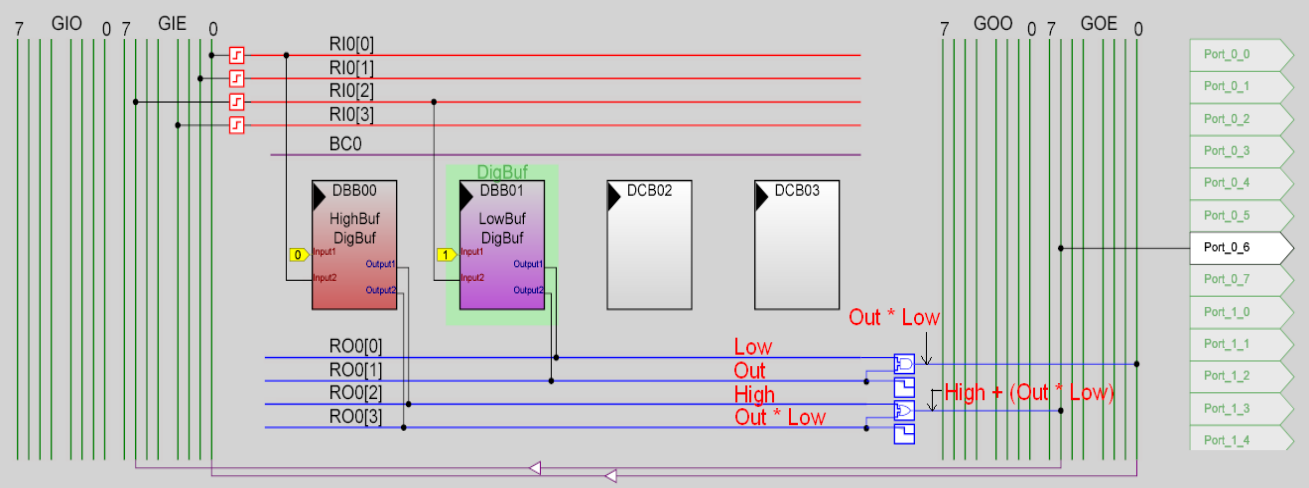


Figure 28. “Comparator Low” Wizard



4. Place two DigBuf User Modules. Connect the comparator outputs to the DigBuf inputs.
5. Implement the digital logic using the LUT hardware in the digital row lines, as shown in Figure 29.

Figure 29. Digital Logic Implementation



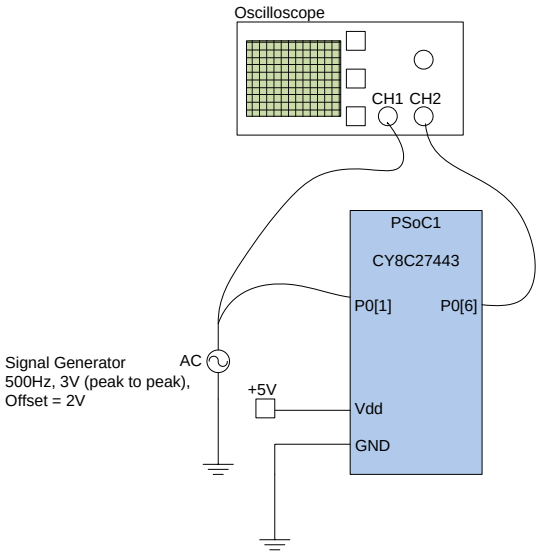
The buffers along with the routing network and the digital row logic (LUTs) allow the required digital logic to be implemented.

6. Write the firmware to start the user modules. Following is the C code:

```
/* part specific constants and macros */
#include <m8c.h>
/* PSoC1 API definitions for all User Modules */
#include "PSoCAPI.h"


void main(void)
{
    /* Start the comparators */
    COMP_High_Start(COMP_High_HIGHPOWER);
    COMP_Low_Start(COMP_Low_HIGHPOWER);

    /* Start digital buffer user
    modules */
    HighBuf_Start();
    LowBuf_Start();
}
```

The PSoC Designer project "HysteresisComp3," based on the CY8C27443 PSoC 1 device, is provided. You can test the project with the external connections, as shown in Figure 30

Figure 30. Test Setup

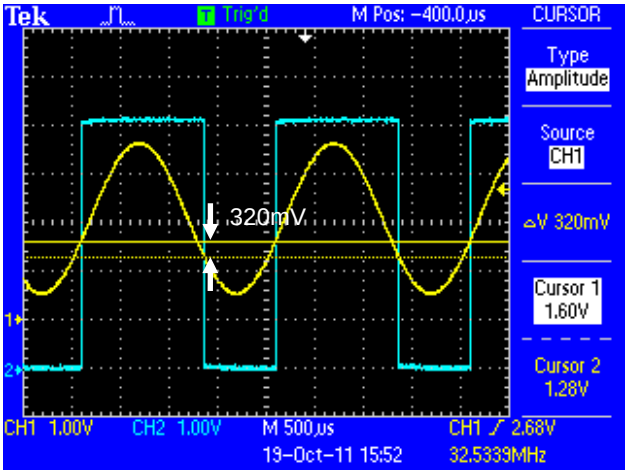


5.1.1 Test Waveforms

Feed a 3-V (peak to peak), 2-V offset sinusoidal signal of any suitable frequency as an input. Figure 31 shows the waveform obtained on the oscilloscope. Notice the hysteresis voltage of 320 mV.

CH1: Input CH2: Output

Figure 31. Test Waveform



6 Summary

This application note explained several topologies to implement hysteresis in a comparator.

Document History

Document Title: AN2108 - PSoC® 1 – Implementing a Hysteresis Comparator

Document Number: 001-29479

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1541809	KOT	See ECN	New application note
*A	2792627	YARA	10/26/09	No change in content. Updated the project files accompanying this application note from PSoC Designer version 4.4 to 5.0
*B	3089742	DASG	11/18/10	1. The Equation 4 on page2 is wrongly written with a minus sign (-), this is rectified with a plus sign (+). 2. The subsequent substitution in the equation results in 95K ohm which is rectified to be 75K ohm.
*C	3442161	RJVB	11/18/2011	Merged AN2108, AN2156, and AN2310 application notes. Updated template.
*D	3819357	MSUR	11/22/2012	Updated attached project to PSoC Designer 5.3. Removed reference to CY8C22x45 device from associated part family.
*E	4807314	ASRI	06/23/2015	Updated project Updated template

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

PSoC is a registered trademark and PSoC Designer is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2003-2015. 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.
This 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.