

Counter

1.00

Features

- 8-bit UDB Counter
- Sync block to ensure error-free timing

General Description

This example project demonstrates how an 8-bit UDB Counter component works and how to read the Period, Compare, and Counter values of the component.

Development Kit Configuration

The following configuration instructions provide a guideline to test this design. For simplicity, the instructions describe the stepwise process to be followed when testing this design with the PSoC Development Kit (CY8CKIT-001) board, but can be generalized for the PSoC 3 Development Kit (CY8CKIT-030) and PSoC 5 Development Kit (CY8CKIT-050) as well.

1. Set LCD power jumper J12 to ON position and leave the rest of the board at default configuration.
2. Observe TC - P0_4 and Comp - P0_5 on an oscilloscope.
3. Ensure that the Character LCD is connected to header P18 on the development board.

Project Configuration

The TopDesign schematic looks as shown in Figure 1 below. The counter is configured (see Figure 2) to be an 8-bit, UDB-based counter, with a period of 100 and compare value of 50. Since the input 'Count' is a 200 Hz clock, the Counter period is 505 ms. The Compare Mode of the Counter is set to Less Than and the Compare Value is set to 5. An interrupt is placed on the schematic so as to be triggered on every terminal count of the counter. The terminal count and compare outputs are connected to output digital pins which can be observed to better understand the behavior of the component. Note that the Counter block is clocked at BUS_CLK frequency.

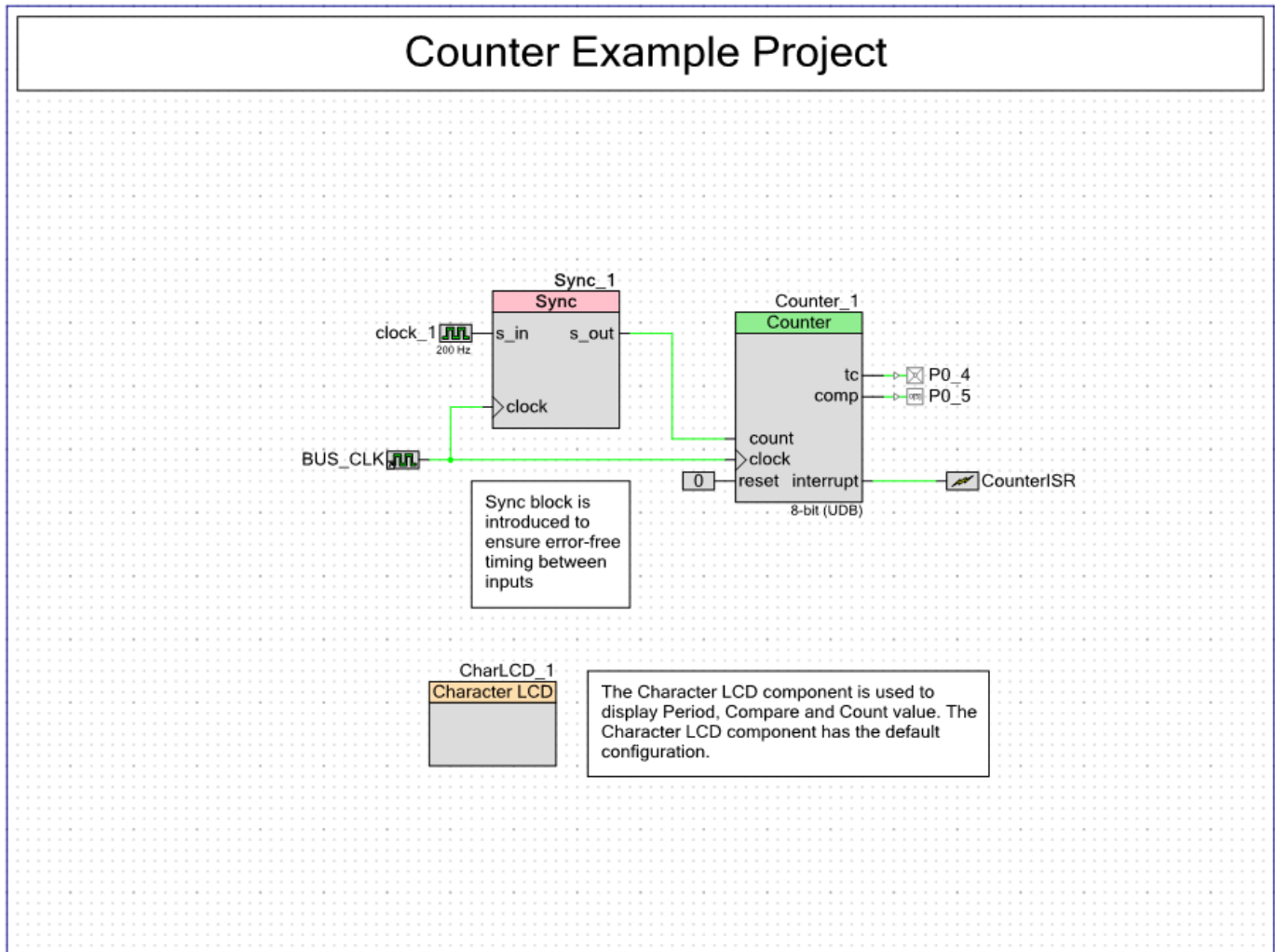


Figure 1. TopDesign schematic

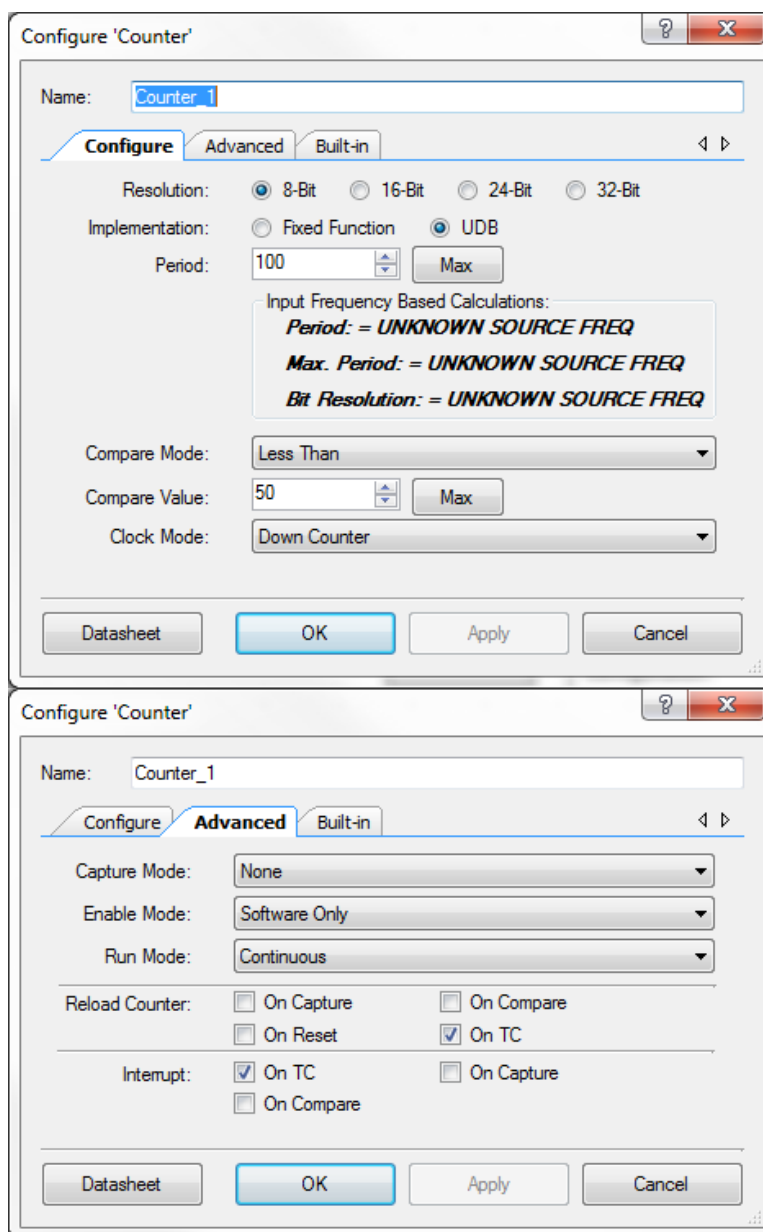


Figure 2. Counter Configuration

Project Description

All components are started in the main function. Global interrupts are enabled. The Counter in the example project counts from the Period value 100 down to the terminal count 0. The Counter is configured to trigger the interrupt on the terminal count event. On reaching the terminal count, the Counter triggers an interrupt and the ISR routine reads the status register to clear the interrupt and also increment an interrupt count. The Character LCD is used to display the test name and Period, Compare, and Count values. The Interrupt count that is returned by the ISR is also displayed on the Char LCD.

The comp output can be seen to be asserted once the counter has down counted to a value less than 50 until the Counter reaches the terminal count 0.

Expected Results

TC P0_4: This pin gives a high pulse when the count value becomes zero.

Compare output P0_5: This pin goes high when the counter value is less than the compare value.

Char LCD displays the following:

1st row:

COUNTER DEMO

Counter Period = 64

2nd row:

Compare value = 32

Counter value = (Decreasing until it reaches 0 and reloads on TC)

IntCnt: Displays the interrupt count

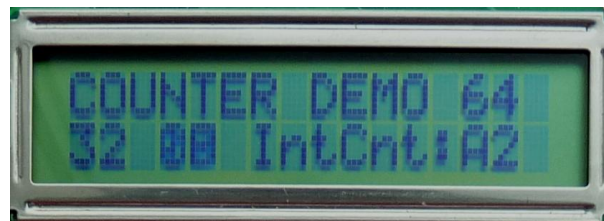


Figure 3. Expected output on LCD

Related Material

Example Projects

- Timer

Application Notes

- [AN54181 - PSoC® 3 - Getting started with a PSoC 3 design project](#)

Training

- [PSoC 3 and PSoC 5 103: Introduction to Digital Peripherals](#)



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, 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.

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.

