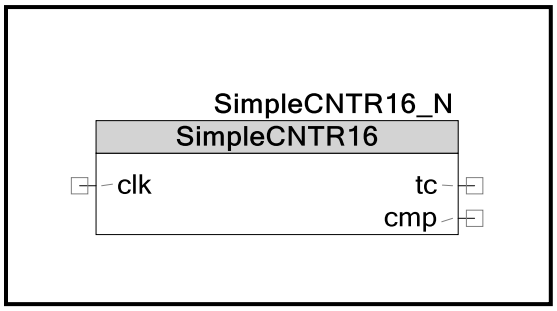
16-Bit Simple Down Counter (SimpleCNTR16)

1.0

Features

- 16-Bit Counter
- Terminal Count Output
- Compare (PWM) Output.



General Description

The SimpleCNTR16 is a simple 16-bit down counter. It can also be used as a single output PWM. This component was created to help new users understand how to develop simple components using two parallel UDB Datapaths.

Input/Output Connections

This section describes the various input and output connections.

clk – Input

Clock input to run the counter.

tc – Output

This signal goes high when the internal counter equals zero.

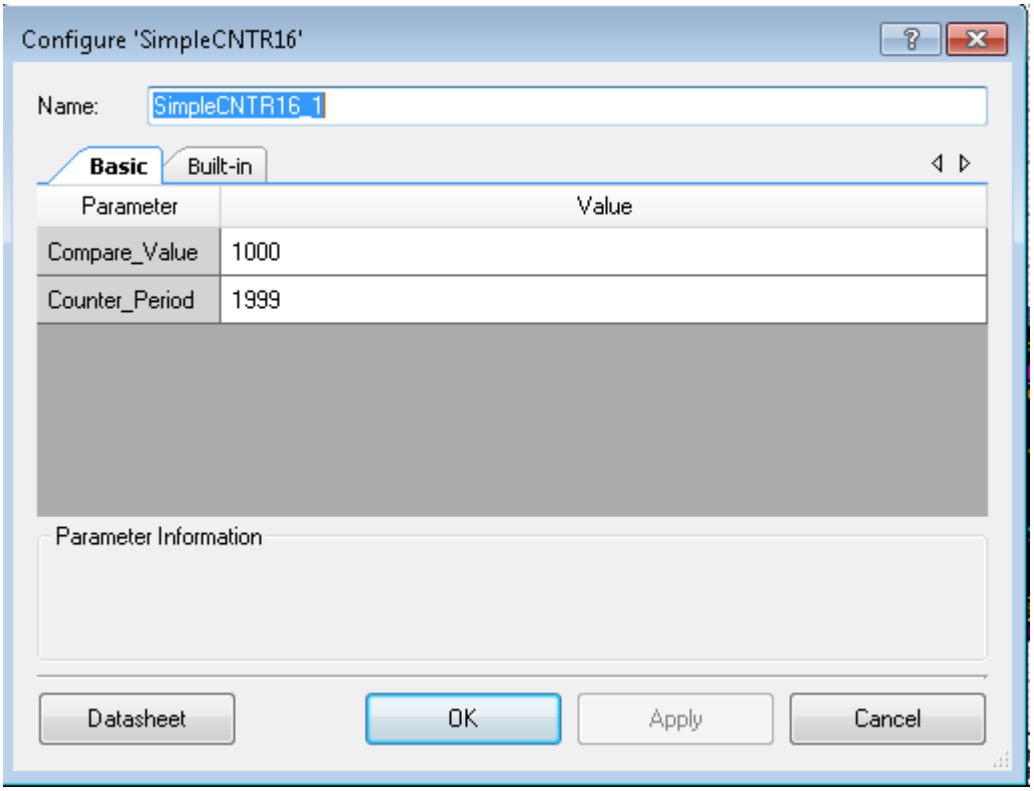
cmp – Output

The Compare (cmp) goes high when the counter is less than the compare value. This output is used as a PWM output.

Parameters and Setup

Drag a SimpleCNTR16 component onto your design and double-click it to open the Configure dialog.

Figure 1 Configure SimpleCNTR16 Dialog



The SimpleCNTR16 component provides the following parameters.

Compare_Value

This parameter determines what value will be used to compare to the internal counter, to generate the PWM output “cmp”. The valid range for this parameter is 1 to 65535.

Counter_Period

This parameter determines the length of the counter period as a number of clocks+1. If a period of 120 clocks is desired, enter 119. The valid range for this parameter is 1 to 65535.

Resources

PSoC UDB resources used.

Resource	Amount Used	Percent PSoC Resource Used
Macrocells	0	0%
Unique Product Terms	0	0%
Datapaths	2	8.33%
Control Register	0	0%
Status Register	0	0%

Application Programming Interface

This component does not include APIs for programming but instead provides register definitions so that the user can easily set parameters during program execution time.

Sample Firmware Source Code

The following is a C language example demonstrating the basic functionality of the SimpleCNTR16 component. This example assumes the component has been placed in a design with the default name " SimpleCNTR16 _1."

Note If you rename your component you must also edit the example code as appropriate to match the component name you specify.

```
#include <device.h>

void main()
{
    /* Set Period to 5000 clocks and compare to 1250 for a 25% duty cycle */
    CY_SET_REG16(SimpleCNTR16_1_Period_Reg_Ptr, 4999);
    CY_SET_REG16(SimpleCNTR16_1_Compare_Reg_Ptr, 1250);

    for(;;)
    {
        /* Place your application code here. */
    }
   }
```

Registers

The functions provided support most of the common runtime functions required for most applications. The following register references provide brief descriptions for the advanced user.

SimpleCNTR16 _Period_Reg

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	Data[15:0]															

- Data[15:0]: Counter Period Register.

SimpleCNTR16 _Compare_Reg

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	Data[15:0]															

- Data[15:0]: Counter Compare Register.

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
1.00		Initial document

© 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.