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.

DigBuf User Module Example Project

CE82304

Project Name: Example_DigBuf
Programming Language: C
Associated Part Families: CY8C24x23, CY8C27x43, CY8C29x66
CY8C24x94, CY8C21x34
Software Version: PSoC Designer™ 5.2 SP1
Related Hardware: CY3210 PSoCEval1 Board
Author: Richa Dham

Project Objective

This project demonstrates the operation of the DigBuf User Module of the PSoC® device.

Overview

The project sequentially switches on and off three LEDs in a row using only two PWMs and a DigBuf.

User Module List and Placement

The following table lists the user modules used in this project and the hardware resources occupied by each user module.

User Module	Placement
DigBuf	DCB02
PWM8_1	DBB00
PWM8_2	DBB01

User Module Parameter Settings

The following tables show the user module parameter settings for each of the user modules used in the project.

DigBuf		
Parameter	**Value**	**Comments**
Default Load Status	Enable	Start the operation on load. Do not start the UM using code.
Input1	Row_0_Output_0	Input from CompareOut of PWM8_1
Input2	Row_0_Output_1	Input from CompareOut of PWM8_2
Input2 ClockSync	Sync to SysClk	Not used because Output2 is not routed to any other blocks as clock.
Output1	Row_0_Output_2	Route the Output1 of DigBuf module to Row_0_Output_2 net.
Output2	Row_0_Output_3	Route the Output2 of DigBuf module to Row_0_Output_3 net.
InvertInput1	Normal	Input1 is not inverted.

Note Row_0_Output_2 and Row_0_Output_3 are NORed in Digital_Interconnect_Row_ 0_Output_2 and routed to P2[2] through GlobalOutEven_2 net..

PWM8_1		
Parameter	**Value**	**Comments**
Clock	VC3	Input is 375 Hz clock.
Enable	High	Enable continuous output.
Compare Out	Row_0_Output_0	For routing to DigBufs Input1 and P2[0].
TerminalCountOut	None	Not used.
Period	150	To produce a output of 2.5 Hz.
Compare Value	50	Set to one-third the Period to get 33% duty cycle.
Compare Type	Less Than	Drive the CompareOutput High whenever the down counter value becomes less than Compare Value.
Interrupt Type	Compare True	Generate an Interrupt whenever the down counter value becomes less than compare value.
Clock Sync	Sync To SysClk	Synchronize the input clock to source clock (system clock)
InvertEnable	Normal	Enable input is active HIGH.

PWM8_2		
Parameter	**Value**	**Comments**
Clock	VC3	Input is 375 Hz clock.
Enable	High	Enable continuous Output.
Compare Out	Row_0_Output_1	For routing to DigBufs Input2 and P2[1].
TerminalCountOut	None	Not used.
Period	150	To produce a output of 2.5 Hz.
Compare Value	50	Set to one-third the Period to get 33% duty cycle.
Compare Type	Less Than	Drive the CompareOutput High whenever the down counter value becomes less than Compare Value.
Interrupt Type	Terminal Count	Not used
Clock Sync	Sync To SysClk	Synchronize the input clock to source clock (system clock)
InvertEnable	Normal	Enable input is active HIGH.

Global Resources

Important Global Resources		
Parameter	**Value**	**Comments**
CPU_Clock	12 MHz	Divide SysClock by 2.
VC1	16	Divide SysClock by 16.
VC2	16	Divide VC1 by 16
VC3 Source	VC2	Set source of VC3 as VC2
VC3 Divider	250	Divide VC2 by 250 to produce a clock of 375 Hz

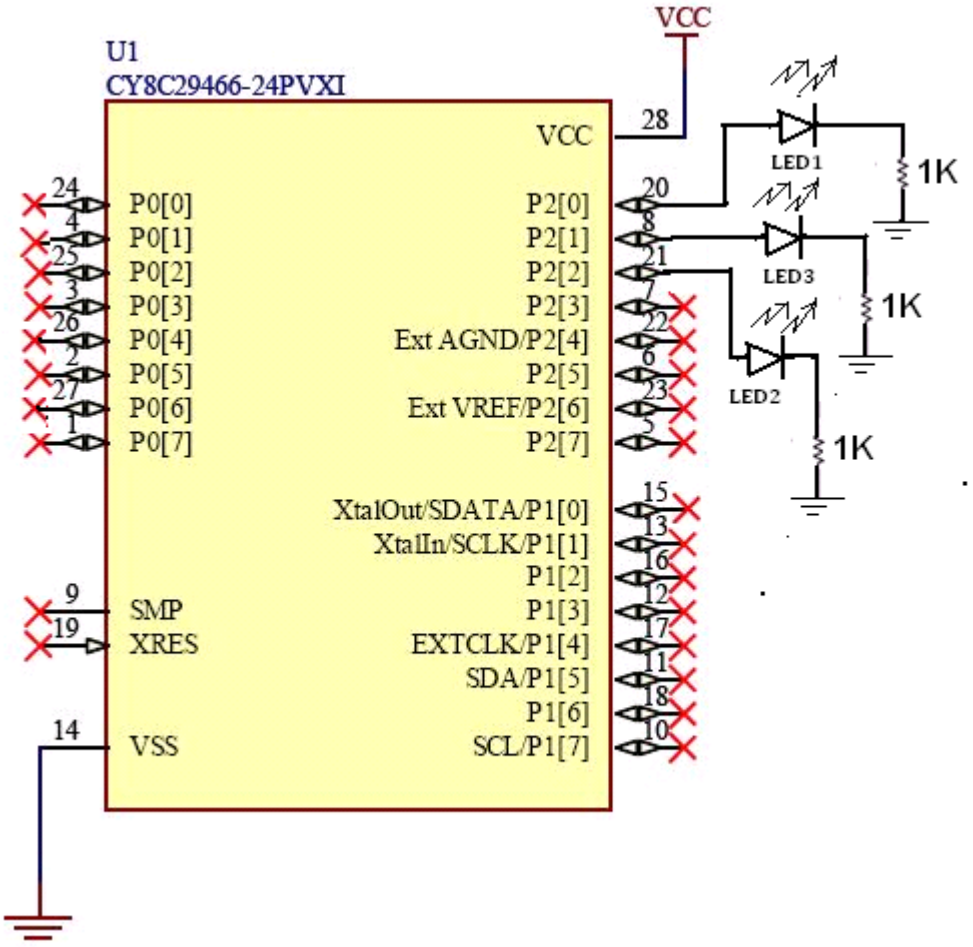
All other global resources are left at their default as they are not specific to this project.

Pin Configuration

Pin Out			
Pin	**Select**	**Drive**	**Direction**
P2 [0]	GlobalOutEven_0	Strong	Output
P2 [1]	GlobalOutEven_1	Strong	Output
P2 [2]	GlobalOutEven_2	Strong	Output

Hardware Connections

Following is the schematic of the project:



The project can be tested using CY3210–PSoC Eval1 board. To test the project using the CY3210 board, make the following connections:

- Connect P2[0] to LED1 of J5 header.
- Connect P2[1] to LED3 of J5 header.
- Connect P2[2] to LED2 of J5 header.

Operation

Upon program execution, all hardware settings from the device configuration are loaded into the device and *main.c* is executed.

The following operations are performed in *main.c*:

- Enable global interrupts.
- Enable PWM8_1 User Module's interrupt.
- Start PWM8_1.
- Loop on infinitely.

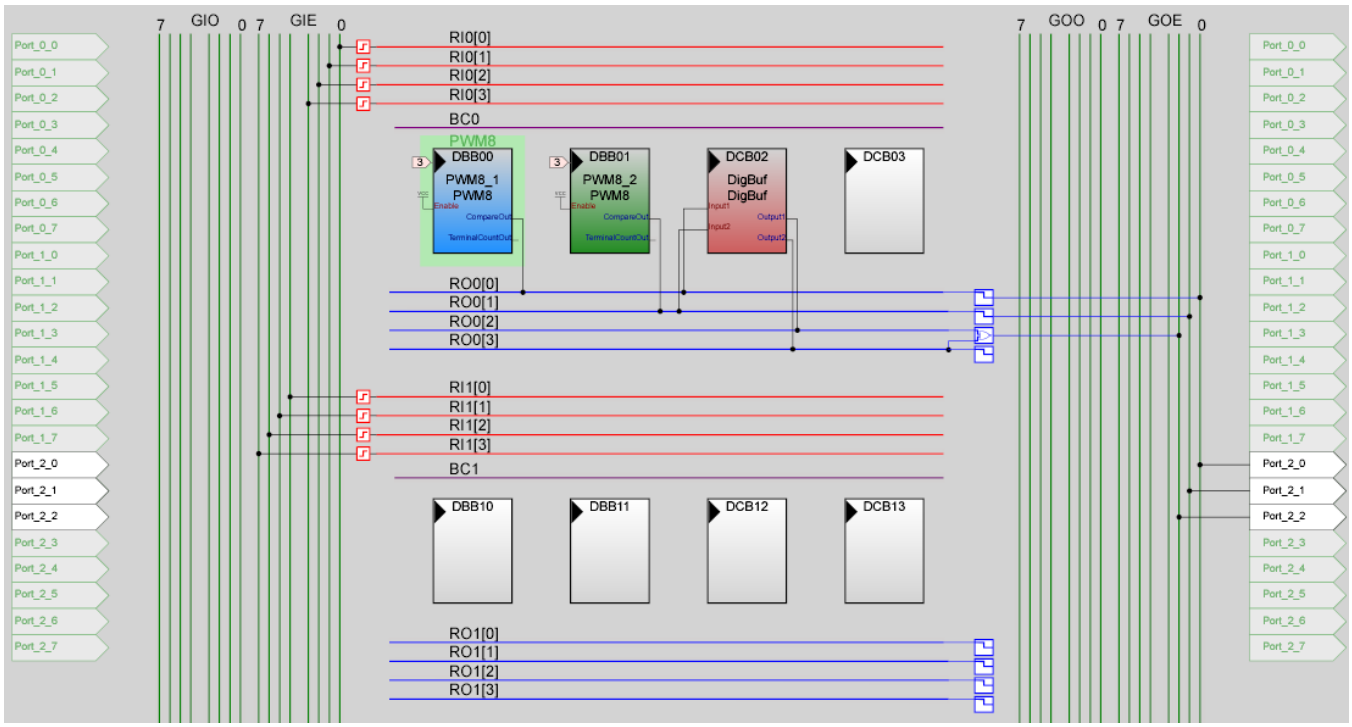
The following operations are performed inside the PWM8_1 ISR. The PWM8_1 generates interrupt on Compare true condition. The ISR area _PWM8_1_ISR is available in *PWM8_1INT.asm* file.

_PWM8_1_ISR:

1. Start PWM8_2.
2. Disable PWM8_1 user module's interrupt.

As the compare outs of PWM8_1 and PWM8_2 are NORed, whenever both the PWMs are Low, the pin P2[2] goes high.

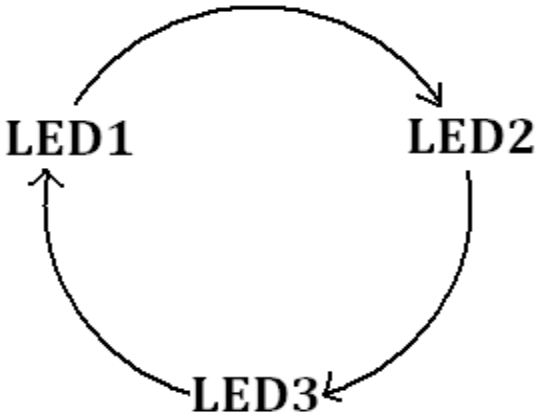
The screen shot of the interconnection between user modules is given below.



Testing the Project

To test the project, make the external jumper connections as explained in section Hardware Connections on page 3.

After power on, the LEDs glow in the following sequence.



Upgrade Information

As the instructions to start PWM8_2 and to disable PWM8_2 are placed inside the user code markers inside the *PWM8_1INT.asm* file, this change is preserved when the project is generated and built. If the source file for the *PWM8_1INT.asm* file changes in a future release of PSoC Designer, these instructions may get overwritten. Therefore, if you have upgraded PSoC Designer and find that the project is not working, check the following.

Open *PWM8_1INT.asm* file and check if there are “CALL _PWM8_2_Start” and “CALL _PWM8_1_DisableInt” instructions present in the user code area in the _ PWM8_1_ISR function. If not, add these lines of codes within the user code markers as shown below

Start Page | example_digbuf [Chip] | boot.asm | main.c | **PWM8_1INT.asm**

```
61
62  _PWM8_1_ISR:
63
64      ;@PSoC_UserCode_BODY@ (Do not change this line.)
65      ;---------------------------------------------------
66      ; Insert your custom assembly code below this banner
67      ;---------------------------------------------------
68      ;   NOTE: interrupt service routines must preserve
69      ;   the values of the A and X CPU registers.
70
71      ;Start PWM8_2
72      CALL _PWM8_2_Start
73
74      ;Disable the PWM8_1 interrupt so that this ISR is not serviced again.
75      CALL _PWM8_1_DisableInt
76
77      ;---------------------------------------------------
78      ; Insert your custom assembly code above this banner
79      ;---------------------------------------------------
```

Document History

Document Title: DigBuf User Module Example Project - CE82304

Document Number: 001-82304

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3723027	RICA	08/24/2012	New Example Project.
*A	4105043	RJVB	08/26/2013	Updated Operation: Updated “screenshot of interconnection between user modules”. Updated Upgrade Information: Added ISR code snippet. Completing Sunset Review.

PSoC is a registered trademark of Cypress Semiconductor Corp. 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
http://www.cypress.com/

© Cypress Semiconductor Corporation, 2012-2013. 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.