

Project Name: Example_CRC

Programming Language: C

Associated Part Families: CY8C24x23, CY8C27x43, CY8C29x66
CY8C24x94, CY8C21x34

Software Version: PSoC[®] Designer™ 5.2

Related Hardware: CY3210 PSoC Eval1

Objective

This project demonstrates the operation of the CRC User Module of PSoC[®] microcontroller. For communication, this project uses SPI master (SPIM) User Module.

Overview

On a switch closure on P1[4], a string of 10 characters is transmitted through using an SPI Master. The bit stream is internally fed into a 16 bit CRC user module. After the entire string is transmitted, the CRC is read and displayed on the LCD.

User Module List and Placement

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

User Module	Placement
CRC16	DBB00 (LSB) DBB01(MSB)
SPIM	DCB02
LCD	P2[0] to P2[6]

User Module Parameter Settings

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

CRC16		
Parameter	Value	Comments
InputDataStream	Row_0_Input_1	MOSI is routed to CRC16 module.
Clock	Row_0_Output_0	SClk of SPIM is routed to CRC16 module.
Clock Sync	Sync to SysClk	Synchronize the clock to SysClock as SClk (VC1 / 2) is derivative of SysClk.
InvertInputDataStream	Normal	Do not invert the data input to CRC16 module.

SPIM		
Parameter	Value	Comments
Clock	VC1	1.5 MHz clock from VC1.
MISO	Low	Not used.
MOSI	Row_0_Output_1	Routed to CRC16 module through Row_0_Output_1 net.
SClk	Row_0_Output_0	Routed to CRC16 module through Row_0_Output_0 net.
InterruptMode	TXComplete	Not used in this example.
Clock Sync	Sync to SysClk	Synchronize the clock to SysClock as VC1 is derivative of SysClk.
InvertMISO	Normal	Not used.

Note The input clock to SPIM module must be twice the required data rate.

LCD

Parameter	Value	Comments
LCDPort	Port_2	Port 2 is used to send data to LCD
BarGraph	Disable	Bargraph is not used in this example

Global Resources

Important Global Resources		
Parameter	Value	Comments
CPU Clock	SysClk/2	Sets the CPU frequency to 12 MHz
VC1	16	Divide 24 MHz system clock by 16 to get a 1.5 MHz clock (for SPIM source)

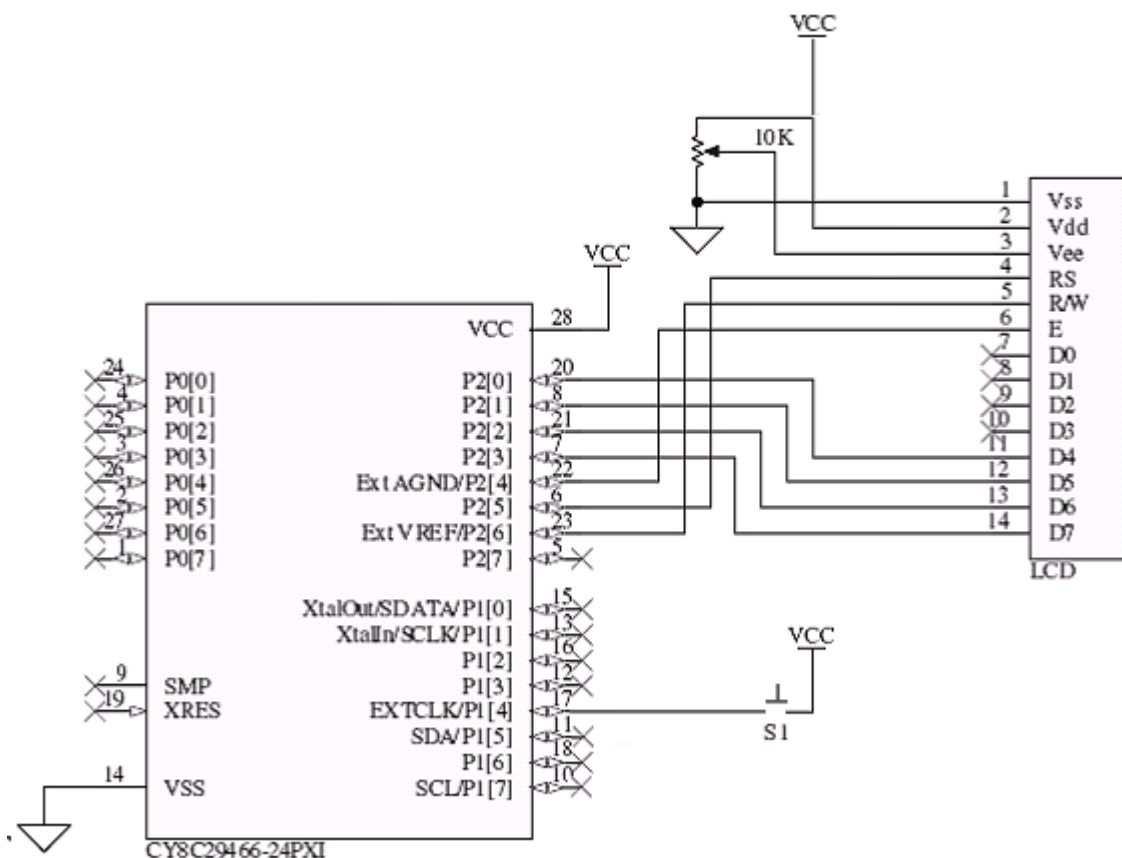
Note All other global resources are left at the default values, as they are not specific to this project.

Pin Configuration

Pin	Select	Drive
P1[4] (Switch)	StdCPU	Pull-down
P2[0]-P2[6]	StdCPU	Strong (LCD)

Hardware Connections

The schematic of the project is as follows:



The project can be tested using the CY3210 PSoC Eval1 board. To test the project, the following connections must be made.

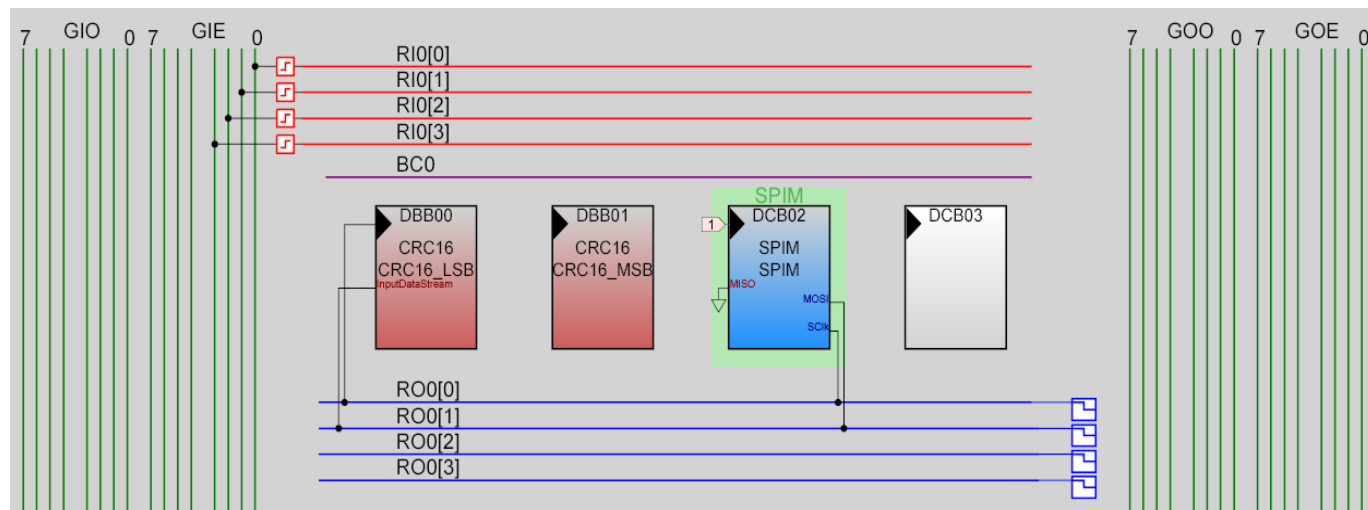
- Connect P14 of J8 to SW of J5
- Connect LCD module to J9

Operation

On execution of program, all hardware settings from the device configuration are loaded into the device and *main.c* is executed. The 24 MHz system clock is divided by 16 (VC1) to generate a 1.5 MHz clock, which is provided as the clock for SPIM User Module. The effective SPI communication is at 0.75 MHz (half of input clock).

The MOSI output of the SPIM is fed into the InputBitStream of CRC16 and the SClk output of the SPIM user module is provided as the clock for CRC16, so that whenever a bit is transmitted out, the same data enters the CRC and then clocked.

Following screen shot shows the interconnection between user modules.



In main, an array of three strings is defined in flash. On every switch press on P1[4], one of these strings is sent through SPIM. As the string is sent out through SPIM, the CRC user module receives the bit stream and calculates the CRC. When the transmission of the string is complete, the CRC is read and displayed on the LCD. Following operations are performed in main.

1. Initialize SPIM and CRC
2. Reset string index
3. Initialize LCD and show the startup screen, display the string "Press Switch S1"
4. Enter infinite loop
5. Check if switch is pressed. If yes, proceed next
6. Wait for switch to be released
7. Initialize the CRC seed and start CRC
8. Transmit the string indicated by the current string index
9. Stop CRC user module and read the CRC value.
10. Display the string and CRC on the LCD
11. Increment the string index. If string index is more than 3, then reset index
12. Go to Step #5

Testing the Project

After downloading the Example_CRC project into PSoC, power the device.

The following display appears on the LCD.

Press Switch S1

Everytime the switch is pressed, the display cycles through the three strings and their CRC.

On first key press:

00-ABCDEFGHJJ

CRC-86F5

On second key press:

01-abcdefghij

CRC-A32A

On third key press:

02-1234567890

CRC-D321

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip" and PSoC Designer are trademarks 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, 2010-2011. 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.