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.



THIS SPEC IS OBSOLETE

Spec No: 001-54155

Spec Title: PSOC(R) 1 - APPLICATIONS USING TWO HARDWARE I2C CHANNELS - AN54155

Sunset Owner: SOMASHEKAR HARIAPPA (STHA)

Replaced by: 001-50987

PSoC® 1 - Applications Using Two Hardware I²C Channels

AN54155

Author: Ben Kropf
Associated Project: No
Associated Part Families: CY8C28x03, CY8C28x23 CY8C28x43, CY8C28x45
Software Version: PSoC Designer™ 5.0 or Newer
Associated Application Notes: AN50987

OBSOLETE

Application Note Abstract

AN54155 presents various high-level descriptions of I²C applications where using one CY8C28xxx PSoC® 1 device that has two hardware I²C channels is beneficial. PSoC 1 devices with two or more hardware I²C blocks provide a flexible solution for complex digital communication needs. This allows for even more digital communication functionality than the advanced analog and digital functions PSoC 1 devices already provide. The PSoC 1 I²C hardware interfaces are also capable of SMBus protocol implementation.

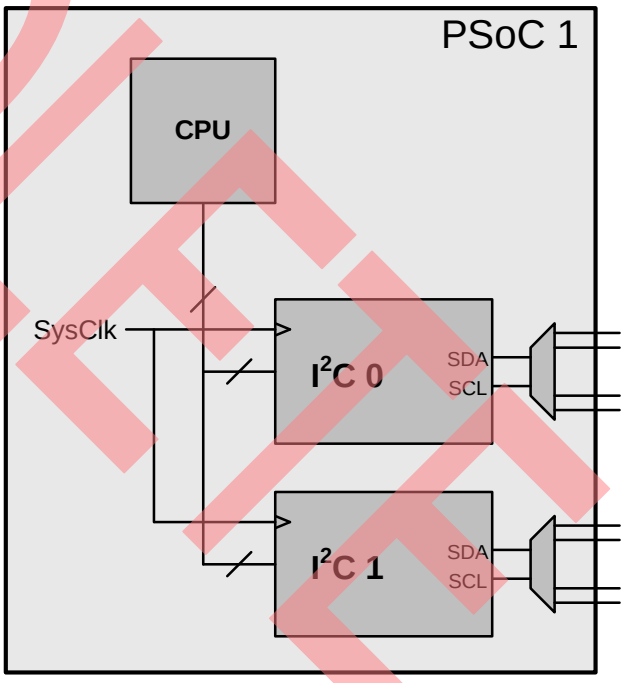
Introduction

Some PSoC 1 Programmable System-on-Chip families have multiple hardware I²C blocks. These allow one PSoC device to communicate on multiple I²C buses in the same system with minimal CPU intervention. This application note gives a high-level overview of applications where one PSoC 1 device can be used to interface with two I²C buses at the same time.

Basic Architecture

The PSoC I²C hardware is implemented as shown in Figure 1. (This shows only a small subset of the PSoC hardware.) There are two I²C blocks that interface to the PSoC CPU. The SDA and SCL signals of each block are demultiplexed to two pairs of GPIOs. This effectively allows the PSoC to be physically connected to four different I²C buses, but only active on two at a time. While an I²C hardware block is inactive, the SDA and SCL signals disconnect from the GPIO. This allows the GPIO pin to be used for other purposes. Both I²C blocks are directly clocked with the main system clock (SysClk) of the PSoC device. Each I²C block is configured for slave, master, or multi-master operation. Each I²C block operates at I²C bus frequencies between 0 Hz and 400 kHz.

Figure 1. Basic PSoC Multi-I²C Architecture



System Configurations

Using two I²C hardware blocks, the following I²C system configurations are possible in one device. Each of these configurations is useful in different contexts in various system designs. Table 1 on page 2 lists the three possible I²C configurations (in relation to the PSoC) and common uses for these configurations.

Table 1. I²C Configurations with Common Use Cases

Configuration	Common Uses
2 I²C slaves	Multi-I²C bus systems I²C shared memory device
1 I²C slave, 1 I²C master or multi-master	I²C bus switch I²C hot-swap controller I²C buffer Debugging
2 I²C masters or multi-masters	Eliminate I²C buffers Increase bandwidth

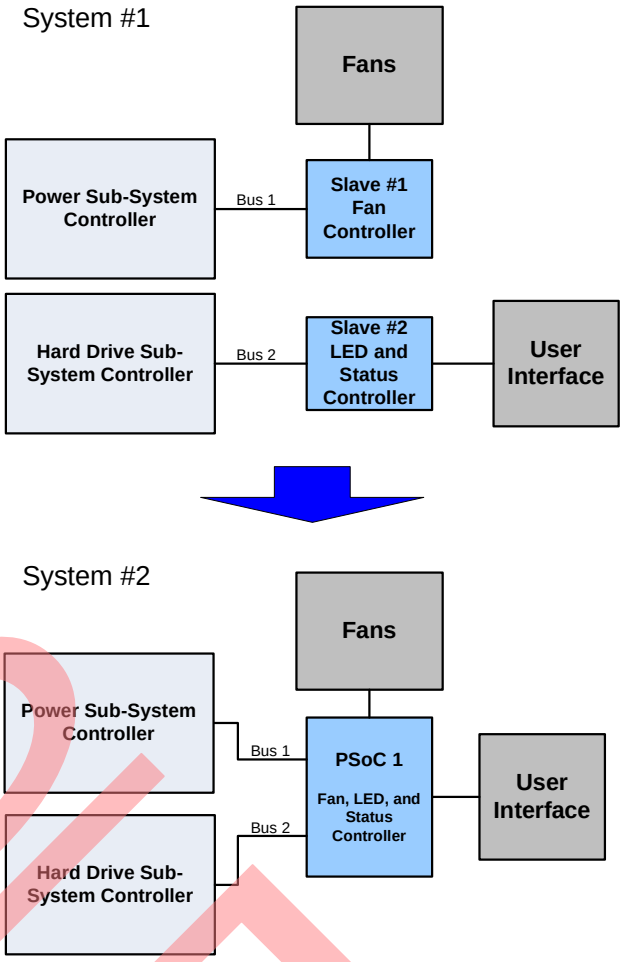
Two I²C Slave Configuration

The configuration in which two I²C slaves are used is quite useful. This is because all PSoC devices support the I2Cm User Module, which is a software implementation of an I²C master that does not require a dedicated I²C hardware block. However, a software implementation of an I²C slave is not feasible in a PSoC. Therefore, an I²C hardware block is necessary for each I²C slave controller. The two-slave configuration allows integration in systems that have multiple I²C buses. A server and server chassis system is an example for this. In this application, multiple I²C buses are used to communicate between the various subsystems of the server. One PSoC with two I²C slave interfaces acts as a central point of communication in the system.

This configuration is useful in any system that uses multiple I²C buses. One PSoC device is used to integrate the functionality of two different slave devices. This is done with minimal or no changes to the master's firmware, since the PSoC device is fully programmable. This allows it to mimic the functionality of the slave devices that were previously on the bus.

Figure 2 shows a system with two I²C buses used for controlling fans and a user interface. A PSoC design with two I²C slaves integrates the functionality of two separate I²C slaves into one part. In this case, neither of the upstream I²C masters needs any firmware changes since the PSoC device implements the same functionality.

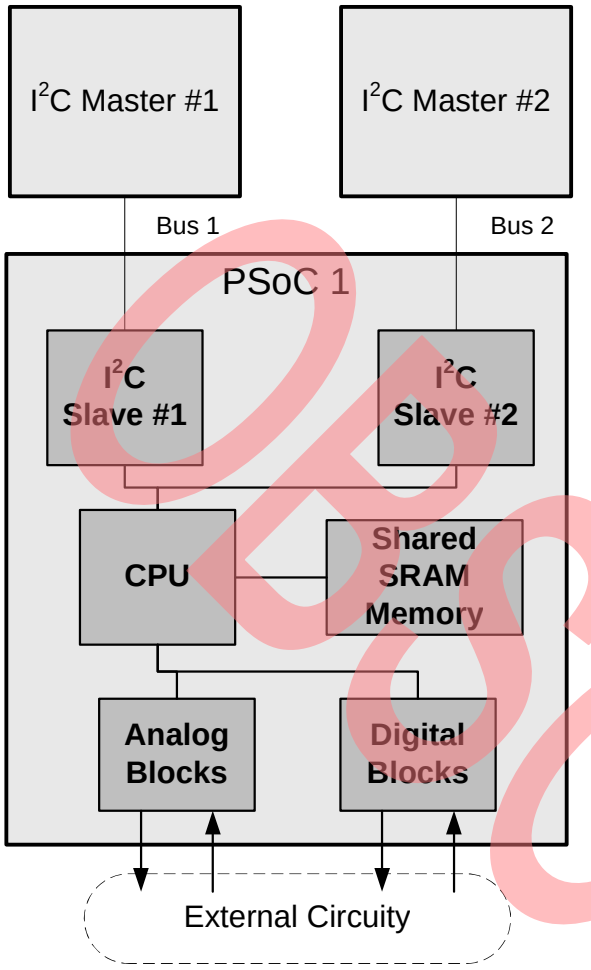
Figure 2. Two Integrated Slaves Using One PSoC Device



An additional useful function enabled by two I²C slaves is a bridge between two I²C masters. This is useful when an application requires two different masters sending the data back and forth without implementing multi-master I²C mode. In this case the PSoC device acts as a shared memory between two different masters. This configuration is shown in Figure 3 on page 3. Each master reads or writes data to the same memory locations in the PSoC device at any time. The PSoC collects data to place in this shared memory. Additionally, it performs functions based on the commands placed into the shared memory and also interfaces with other external circuitry to allow even more functionality.

PROOF FILE

Figure 3. PSoC System Functioning as Shared Memory

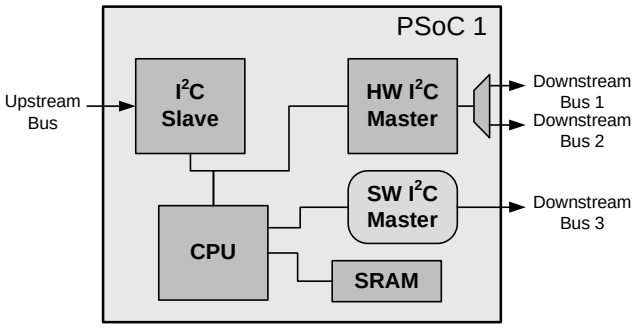


One I²C Master, One I²C Slave Configuration

An application of one I²C master and one I²C slave in the same device is an I²C bus switch. This is configured by connecting a hardware I²C master to two downstream buses and a slave to an upstream bus. Instances of the I2Cm User Module can be added to implement even more downstream I²C buses. The primary drawback of these software I²C master implementations is that they are limited to a 100 kHz bus frequency. Additionally, the I2Cm User Module uses 100% of the CPU and blocks all interrupts during transactions. The hardware I²C master implementation operates up to 400 kHz and does not block interrupts or the CPU.

Figure 4 shows a block diagram of this application. The PSoC device cannot implement a direct feed-through I²C bus switch. However, buffering the data in RAM between the upstream and downstream buses is generally acceptable. It is even possible to bypass the SRAM buffer. This is done by starting the downstream transaction as soon as the first byte of data is received from the upstream master.

Figure 4. Using the PSoC Solution as an I²C Bus Switch

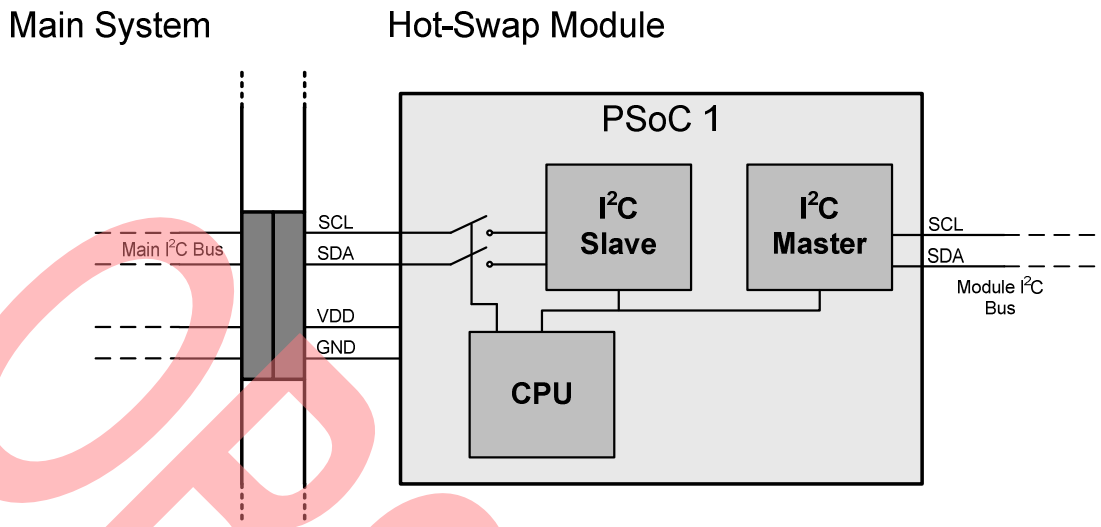


A second useful application for having one I²C slave and one I²C master is a hot swap controller device. In some high-reliability systems, such as servers, seamlessly disconnecting and connecting new or different modules into an operational system is a required feature. If a communication bus for the module is I²C, it is preferable for the insertion of the new module to have little or no effect on the existing I²C signals of the bus. During normal operation, the PSoC device collects all of the necessary information from the module's local I²C bus using the device's I²C master. When the I²C master from the main system requests the data, the PSoC device sends the collected data to the master since the PSoC device is also an I²C slave.

The PSoC solution can control hot-swapping by intelligently monitoring the power supply rail and I²C bus of the main system once the module is plugged in. It can wait for the power rail to stabilize and an idle I²C bus before attaching itself electrically as a new slave on the main I²C bus. The PSoC system also reacts appropriately after detecting any number of faults on the module. It may report the fault status to the master and then detach itself from the main I²C bus as it waits for the module to be replaced. Bus connection is accomplished by connecting or disconnecting the PSoC slave hardware from a pair of GPIO pins. When the slave is disconnected from the GPIO, both are in a high impedance state which electrically isolates the PSoC chip from the I²C bus.

OBSOLETE

Figure 5. A PSoC Device Functioning as an I²C Hot Swap Module Controller



A PSoC system with one slave and one master can also perform an I²C bus buffering function similar to that of the hot-swap controller shown in Figure 5. In this function the device does not control the hot swap process but simply buffers data and communication between two I²C buses. When an I²C bus is loaded with too much parasitic and pin capacitance, adding more I²C slaves to the bus may not be possible. This is because a high capacitive loading on the bus requires a lower bus frequency or it may lead to data corruption due to noise. In this case, a second I²C bus is optionally added, which isolates any new capacitance that this expanded bus has. The PSoC device functions as a slave on one bus and a master on the other. It bridges the communication between the two buses.

Finally, another use for one I²C slave and one I²C master is in an application that requires an I²C master, but also needs a debugging port. In this case, the PSoC device can function normally in the system as an I²C master. An external tool is optionally used to connect to the same device and talk to it as an I²C slave. This tool is used to monitor system variables for calibration or debugging purposes if required.

Two I²C Master Configuration

A PSoC device can use both I²C hardware blocks in master or multi-master mode. This is useful for a few different purposes. The primary reason to use two different I²C masters is if a system has a large I²C bus with many slaves. As discussed earlier, it is possible for such a bus to have too much capacitance. This leads to a limited number of slaves on the bus and a potential limit to the bus frequency. In this case, a PSoC solution with two I²C masters can communicate with slaves on two different buses. This eliminates the need for a separate I²C buffer device to expand the size of the I²C bus.

A second reason to use two I²C masters is that using two masters at the same time can potentially increase the total bandwidth of a system. One system may have a high performance I²C digital-to-analog converter in the system which requires a high update rate. The same system may have a graphic LCD that also requires rapid update. One I²C master may not be able to communicate with these devices fast enough to allow them both to operate as quickly as they need to. In this case, using two I²C masters on the same PSoC chip allows both devices to be updated more often.

Other PSoC Functions

PSoC devices are well-suited for performing many functions in addition to the I²C functions discussed in this application note. These functions include (but are not limited to) CapSense® capacitive sensing, motor and fan control, LED driving, USB applications, analog and sensor measurement, analog signal conditioning, digital communications, mechatronics, battery charging, and computational algorithms. PSoC devices can integrate many of these functions, including I²C communication, into one device.

Summary

PSoC programmable systems-on-chip are very powerful programmable devices. They have many highly configurable analog and digital resources. PSoC devices that also have two hardware I²C blocks provide a very flexible solution for complex digital communication challenges. PSoC devices are well-suited for integrating communication, analog, and digital functions all into one device. Additionally, the device's flash-based reprogrammability allows users to overcome I²C communication challenges in the exact way that the application requires.

OBSOLETE

Document History

Document Title: AN54155 PSoC® 1 – Applications Using Two Hardware I²C Channels

Document Number: 001-54155

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2723091	BTK	06/24/09	New application note.
*A	2738518	BTK	07/15/09	Changed Associated Project details to 'No'
*B	3240938	BTK	05/04/2011	PSoC 1 inserted in the document title. Updated document to make it clear that it only applies to the PSoC 1 series of products. Updated abstract.
*C	4072364	STHA	7/22/13	Obsolete Spec

OBSOLETE

PSoC and CapSense are registered trademarks 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, 2009-2013. 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.

OBSOLETE