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.

CE220762 – PSoC 6 MCU PDM to I2S Example

Objective

This example demonstrates how to route Pulse-Density Modulation (PDM) audio data to the Inter-IC Sound (I^2S) Interface in PSoC® 6 MCU.

Requirements

Tool: PSoC Creator™ 4.2, Peripheral Driver Library (PDL) 3.1.0

Programming Language: C

Associated Parts: All PSoC 6 MCU

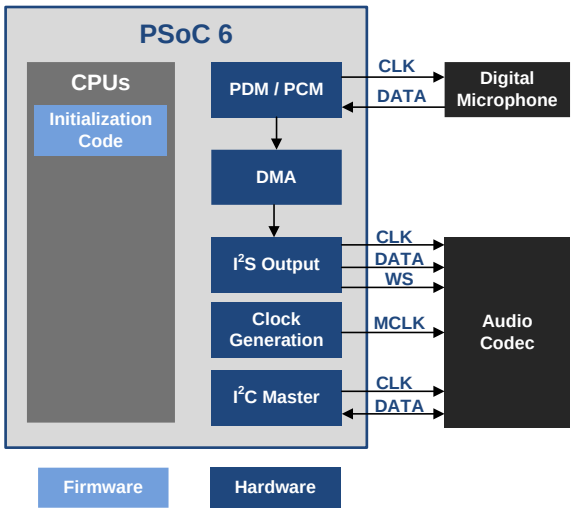
Related Hardware: CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit, TFT Display Shield Board CY8CKIT-028-TFT

Overview

This code example shows how to record a short audio sample from a microphone to SRAM, then play it on a speaker or headphone. The example uses DMA to transfer data from the PDM/PCM hardware block, which interfaces with a microphone, to SRAM. Once the recording is complete, another DMA transfers the recorded data to the I^2S hardware block, which interfaces with an audio codec chip. This configuration frees the CPU completely, so it can execute other tasks.

Figure 1 shows the high level-block diagram of this application.

Figure 1. Block Diagram



Hardware Setup

This example requires the CY8CKIT-028-TFT shield to be connected to the CY8CKIT-062 (-BLE) kit.

Operation

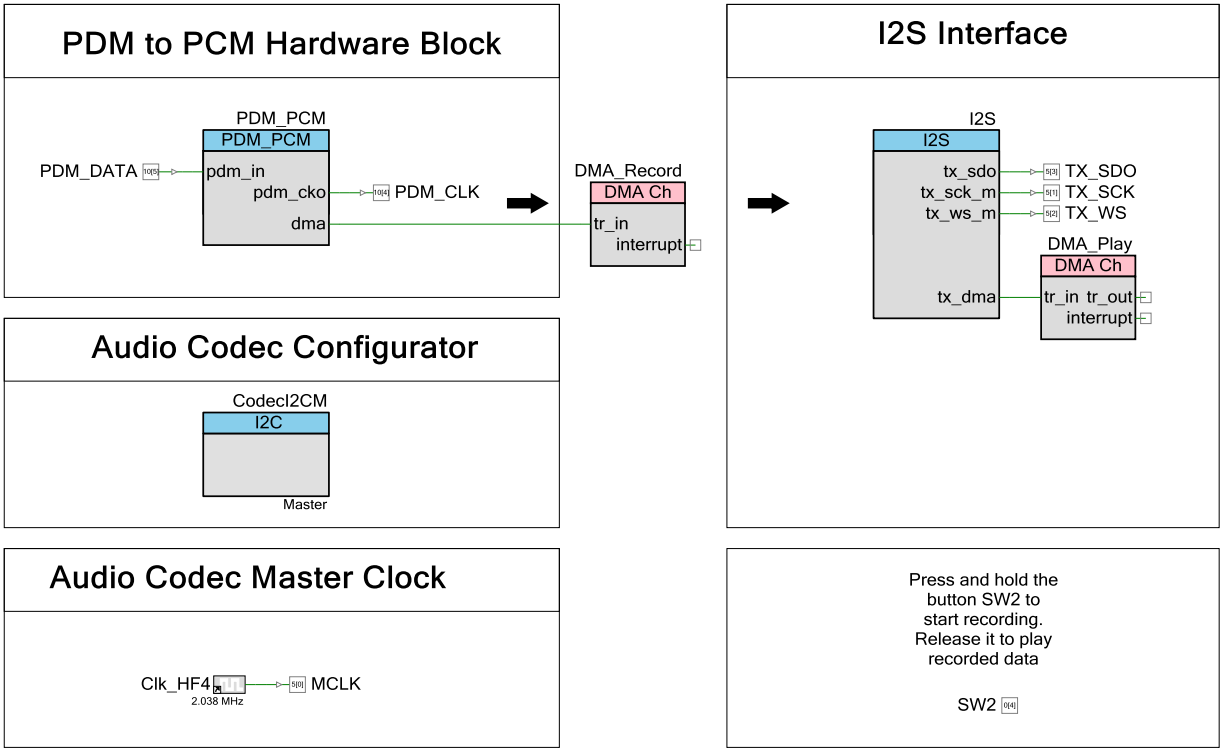
1. Connect the CY8CKIT-028-TFT shield to the CY8CKIT-062 kit.
2. Build the “CE220762_PDM_I2S” project and program the CY8CKIT-062 kit. For more information on building projects and device programming, see PSoC Creator Help.
3. Connect a headphone or speaker to the audio jack of the CY8CKIT-028-TFT shield.

4. Press the CY8CKIT-062 SW2 button and hold it. Speak over the microphone to record a short message (up to 4 seconds).
5. Release the SW2 button and listen the recorded data with the headphone or speaker.

Design and Implementation

Figure 2 shows the PSoC Creator schematic of this code example.

Figure 2. Project Schematics



The CY8CKIT-028-TFT shield contains the audio codec AK4954A, an audio jack and a digital microphone. This allows you to record data using the microphone and play it with the audio codec. You can connect a speaker or headphone to the audio jack.

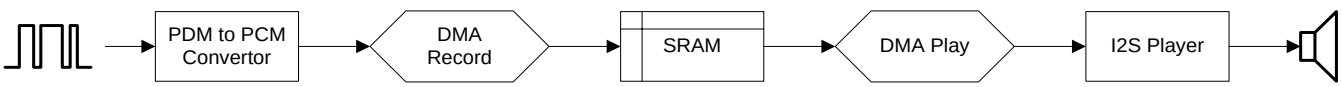
The first stage is to record any sound coming from the microphone and place it in the SRAM of PSoC 6 MCU. This whole process can be achieved using DMA to transfer from the PDM/PCM RX buffer to an array allocated in the SRAM. Once the sound is recorded, another DMA Component move data from the SRAM to the I^{2}S TX buffer.

To record a longer audio stream from the microphone, a sample rate of only 8 ksps is configured in both components – PDM/PCM and I^{2}S. Both the word length of the PCM data output and the I^{2}S TX buffer data size are set to 16 bits. The number of elements allocated for the recorded data array is 65,536, consuming a total of 128 Kbytes of SRAM.

PSoC 6 MCU also generates the clock fed to the audio codec. Based on the AK4954A datasheet, this codec requires a minimum MCLK 256x the frame rate (256 × 8 ksps), which translates to a 2.048-MHz clock. The code example contains an I^{2}C Master, through which PSoC 6 MCU configures the audio codec. The code example provides an API to easily configure the AK4954A codec.

Figure 3 shows the flowchart diagram of the steps in this example.

Figure 3. Flowchart Diagram



For more details on the I^{2}S and PDM/PCM interfaces, refer to the Architecture TRM of the device.

Components and Settings

Table 1 lists the PSoC Creator Components used in this example.

Table 1. List of PSoC Creator Components

Component	Instance Name	Purpose	Non-default Settings
I2S	I2S	Stream data to the audio codec	See Figure 4.
I2C (Master)	CodecI2CM	Configures the audio codec	All default
Clock	HFClk4	Generates the audio codec master clock	Clock Type: Existing Source: Clk_HF4
PDM/PCM	PDM_PCM	Interfaces with the PDM microphone	See Figure 5.
DMA	DMA_Record	Transfers data from the PDM/PCM RX Buffer to the SRAM	See Figure 6.
DMA	DMA_Play	Transfers data from the SRAM to the I^2S TX buffer	See Figure 7.
Digital Output Pin	MCLK	Drives the audio codec master clock	All default
Digital Output Pin	TX_SDO	Drives the audio data line from the I^2S interface	All default
Digital Output Pin	TX_SCK	Drives the audio clock line from the I^2S interface	All default
Digital Output Pin	TX_WS	Drives the word select line from the I^2S interface	All default
Digital Input Pin	PDM_DATA	Collects the PDM microphone data	All default
Digital Output Pin	PDM_CLK	Drives the PDM microphone clock	All default
Digital Input Pin	SW2	Handles the kit's switch	Drive mode: Resistive Pull Up

Parameter Settings

This section shows the changed settings for various Components as well as the system clocks.

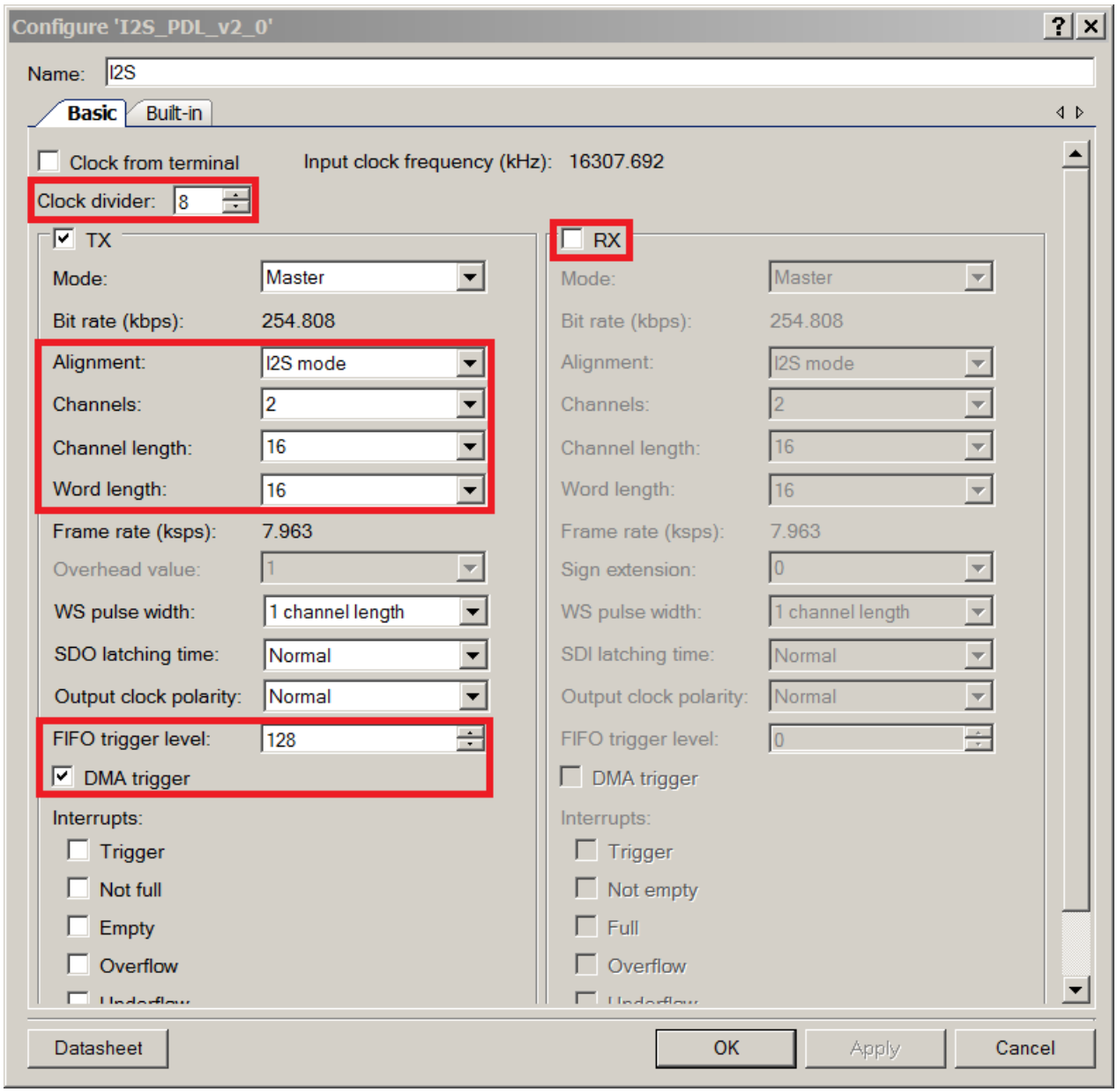
I2S

The I2S Component is configured for a frame rate of 8 ksps. Configure the HFClk1 clock to a specific frequency (see below). The Input clock frequency displayed in the Component is derived from HFClk1. The frame rate is calculated as follows:

$$Frame\ Rate\ (ksps) = \frac{HFClk1}{Clock\ divider\ \times 16 \times Channel\ Length}$$

Figure 4 shows the I2S configuration window. Note that the frame rate is not exactly 8 ksps because the PLL cannot achieve the desired output frequency of 16.384 MHz.

Figure 4. I2S Component Configuration Window



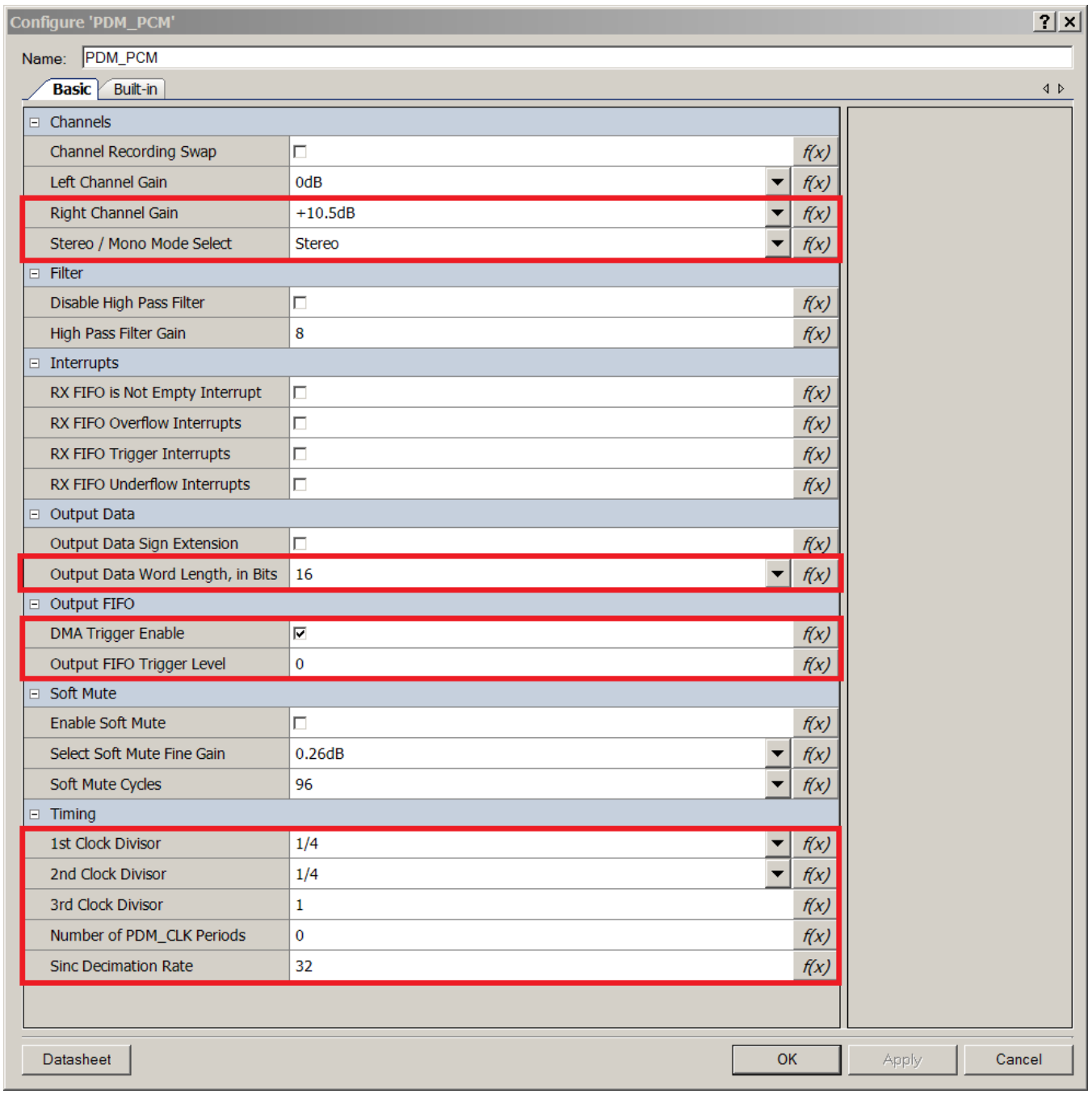
PDM_PCM

The PDM_PCM Component is configured for a frame rate of 8 ksps. The frame rate for this component is calculated as follows:

$$Frame\ Rate\ (ksps) = \frac{HFClk1}{1st\ Clock\ Divisor \times 2nd\ Clock\ Divisor \times (3rd\ Clock\ Divisor + 1) \times 2 \times Sinc\ Decimation\ Rate}$$

Figure 5 shows the configuration window.

Figure 5. PDM/PCM Component Configuration Window



HFClk4

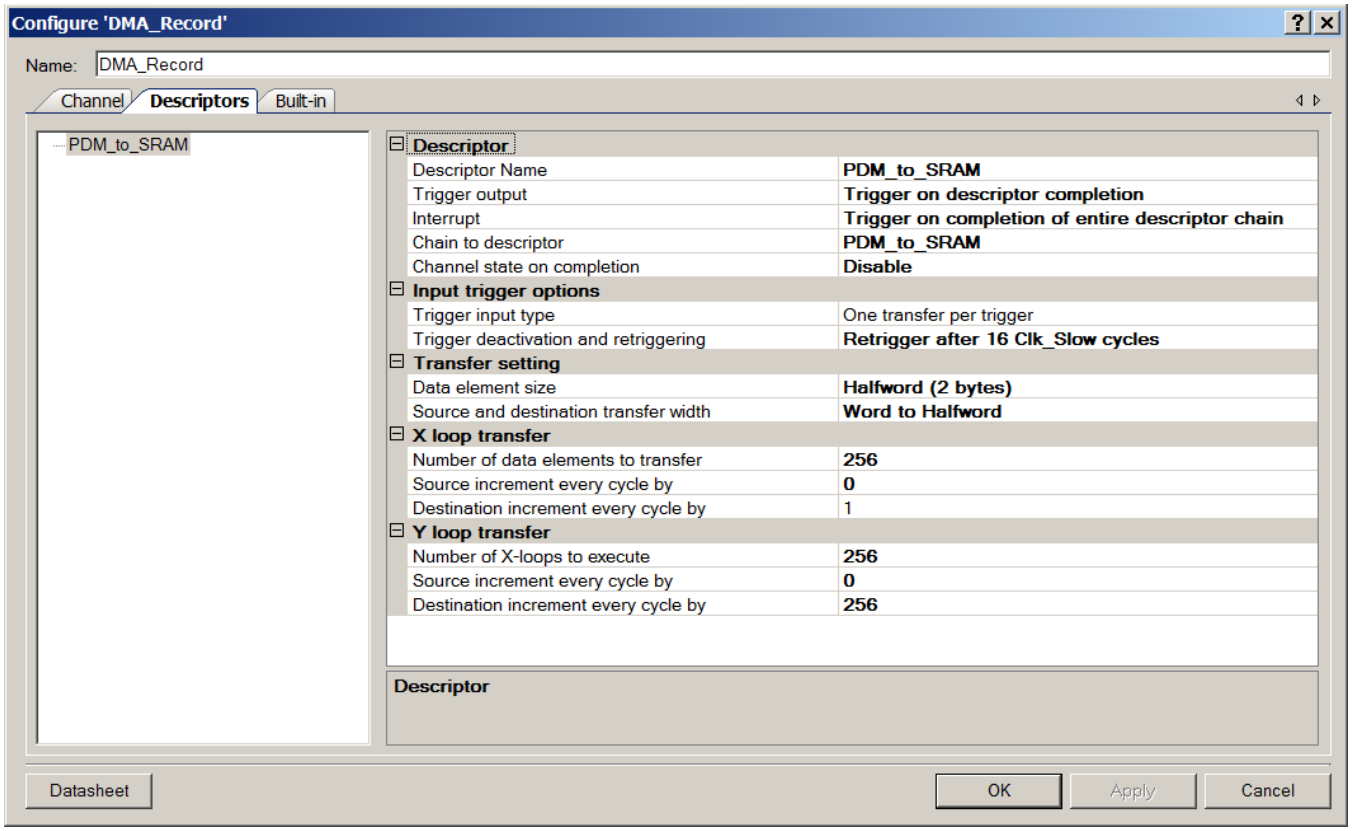
The Component clock is configured to be 2.048 MHz, which is sourced from the PLL output and divided by 8.

DMA_Record

This DMA is configured to transfer data from the PDM RX Buffer to the recorded data array placed in SRAM. When enabled, it triggers when there is at least one element in the PDM RX Buffer.

Figure 6 shows the configuration window of this DMA.

Figure 6. DMA_Record Configuration Window



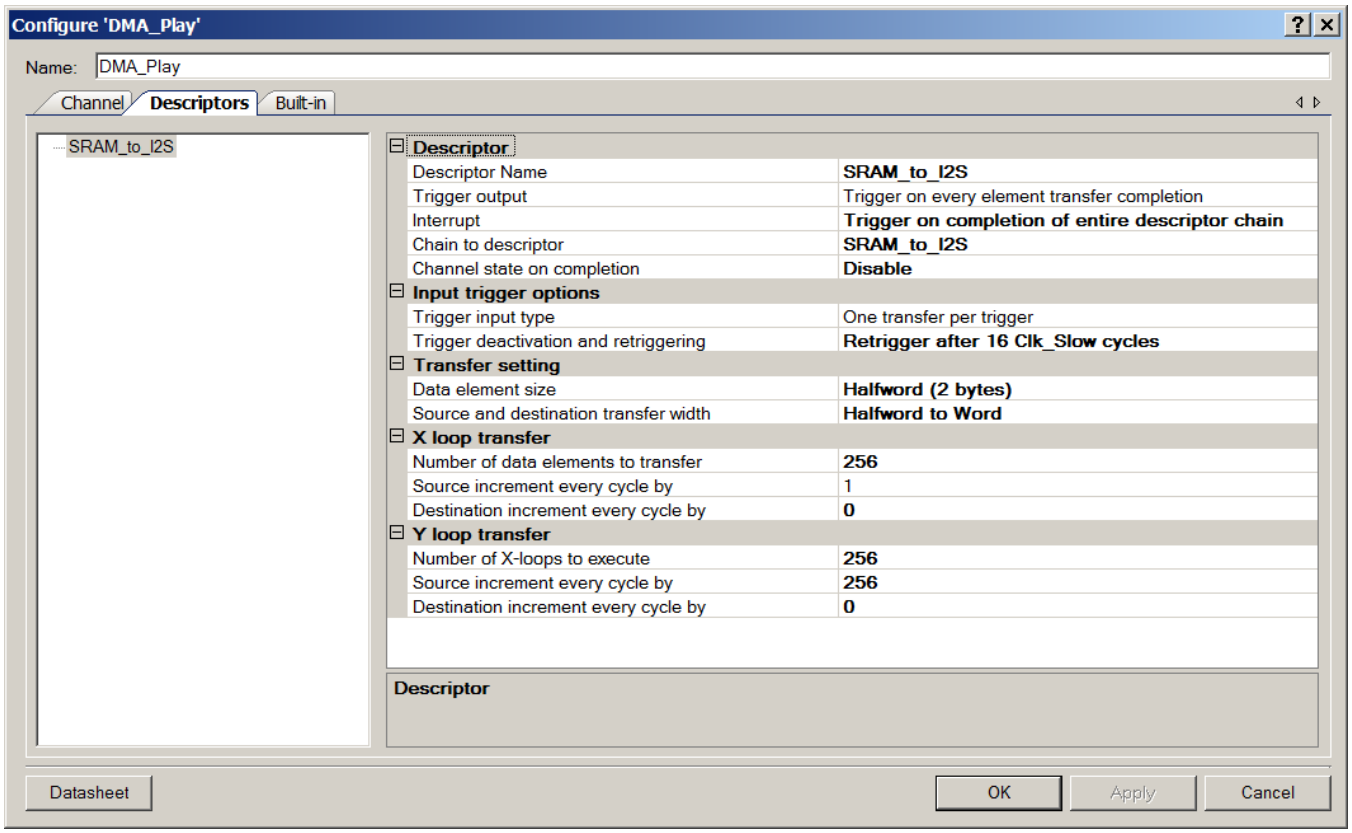
Note that the maximum number of elements to transfer in each loop is 256.

DMA_Play

This DMA is configured to transfer data from the recorded data array placed in SRAM to the I²S TX Buffer. When enabled, the DMA triggers when the I²S buffer has less than 128 elements.

Figure 7 shows the configuration window of this DMA.

Figure 7. DMA_Play Configuration Window



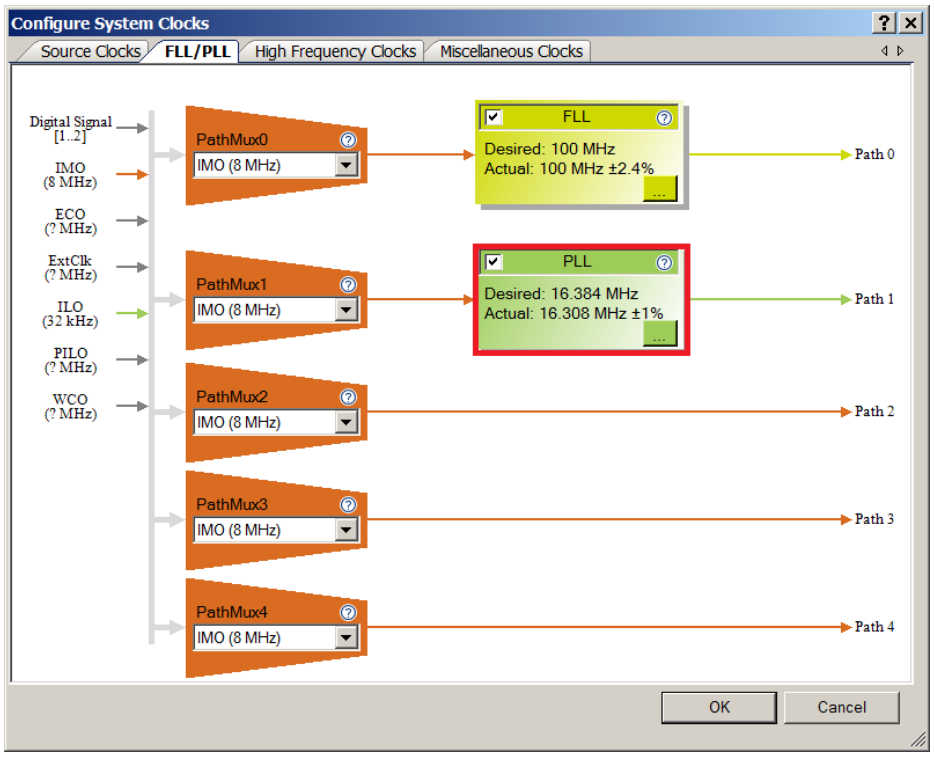
Design-Wide Resources

In the FLL/PLL tab, enable the PLL to be 16.384 MHz. This clock frequency comes from the following equation:

$$HFClk1 = Frame\ Rate \times I2S\ Clock\ divider\ \times 16 \times Channel\ Length = 8k\ \times 8 \times 16 \times 16 = 16384\ kHz$$

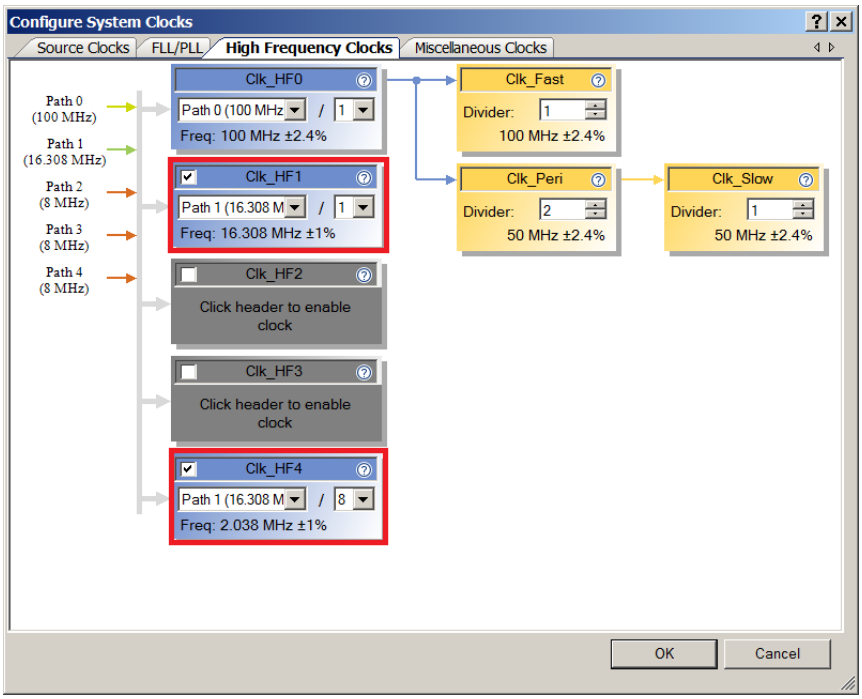
Figure 8 shows the configuration of the PLL. Note that the actual frequency is 16.381 MHz, instead of the desired frequency of 16.384 MHz.

Figure 8. Clock Configuration Window



Configure the High Frequency Clocks HFClk1 and HFClk4 to be linked to Path 1. Note that ideally both the clocks created for the I2S block (MCLK and TX_CLK) should come from the same source. Figure 9 shows how the high-frequency clocks are configured.

Figure 9. High Frequency Clock Configuration



Reusing This Example

This code example runs on CY8CKIT-062-BLE or CY8CKIT-062 kits, which have a PSoC 6 MCU device. To port the design to other PSoC 6 MCU devices and kits, change the target device in **Project** > **Device Selector**, and update pin assignments in the Design Wide Resources. For single-core PSoC 6 MCU devices, port the code from *main_cm4.c* to *main.c*.

Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN221774 – Getting Started with PSoC 6 MCU	Describes PSoC 6 MCU devices and how to build your first PSoC Creator project.
AN217666 – PSoC 6 MCU Interrupts	Describes how to use interrupts in PSoC 6
PSoC Creator Component Datasheets	
Inter-IC Sound Bus (I2S) Component	Sends digital audio streaming data to external I2S devices
Intra-Integrated Circuit (I2C) Component	Supports I^2C slave, master, and master-slave operation configurations.
PDM to PCM Decoder Component	Converts a PDM signal to PCM.
Direct Memory Access (DMA) Component	Transfers data to and from memory and registers.
Code Examples	
Visit the Cypress Code Example or our GitHub site for a comprehensive collection of code examples using PSoC Creator IDE.	
Device Documentation	
PSoC 6 MCU Datasheets	PSoC 6 Technical Reference Manuals
Development Kit Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	
CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit	
TFT Display Shield Board CY8CKIT-028-TFT	
Tool Documentation	
PSoC Creator	Look in the downloads tab for Quick Start and User Guides
Peripheral Driver Library (PDL)	Get the latest version for use with PSoC Creator. Look in the *<PDL install folder>/doc* for the User Guide and the API Reference

Document History

Document Title: CE220762 – PSoC 6 MCU PDM to I2S Example

Document Number: 002-20762

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5945171	RLOS	06/21/2017	New Code Example
*A	6554057	RLOS	04/24/2019	Updated screenshots Changed the design to use only one DMA when playing audio Updated Related Documents section Added a flowchart diagram

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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

© Cypress Semiconductor Corporation, 2018-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.