Filter_ADC_VDAC_poll Example Project
1.0

Features

- FIR low-pass filter at 6 kHz with Blackman window, 85 taps
- Demonstrates the polling mode of the Filter component
- AC-coupled input provided bias with internal Opamp for maximum swing
- DMA used to transfer ADC result to Filter
- VDAC used to display output

General Description

This project is a variation of the Filter_ADC_VDAC example project. Aside from demonstrating PSoC's ability to utilize the Digital Filter Block (DFB) to filter a signal completely in hardware, this example project also demonstrates the polling mode of the Filter component. The example also shows how to manage the DMA setup to transfer data from the ADC directly to the DFB, bypassing the processor. PSoC allows you to implement a filter without having to write CPU-intensive algorithms in firmware.

This project demonstrates the operation of a Low Pass FIR Filter with Blackman window. By changing only a few input parameters in the Filter customizer, you can experiment with a wide variety of FIR and IIR filters.

Development Kit Configuration

The following configuration instructions provide a guideline to test this design. For simplicity, the instructions describe the stepwise process to be followed when testing this design with the PSoC Development Kit (CY8CKIT-001) board, but can be generalized for the PSoC 3 Development Kit (CY8CKIT-030) and PSoC 5 Development Kit (CY8CKIT-050) as well.

1. Set SW3 to 5V and leave the rest of the board at default configuration. This ensures Vdda=Vddd=5V.
2. Set up the passives used in the schematic as shown in Figure 1.
3. Ensure that the Function generator input has a peak-to-peak amplitude of 4V or less.
4. Build the Filter_ADC_VDAC project and then program the hex file onto the PSoC device using the MiniProg3. After programming is complete, disconnect the MiniProg3.
5. Reset the PSoC device.

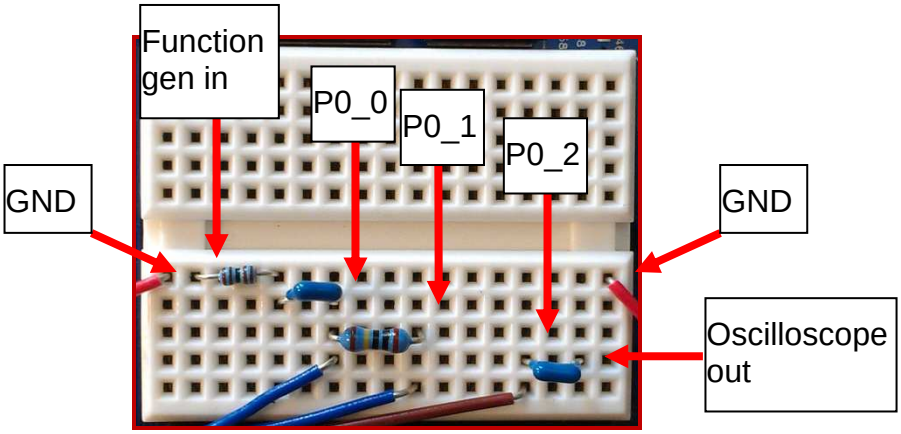

Figure 1. Breadboard setup

Project Configuration

The top design schematic for this project is shown in Figure 2.

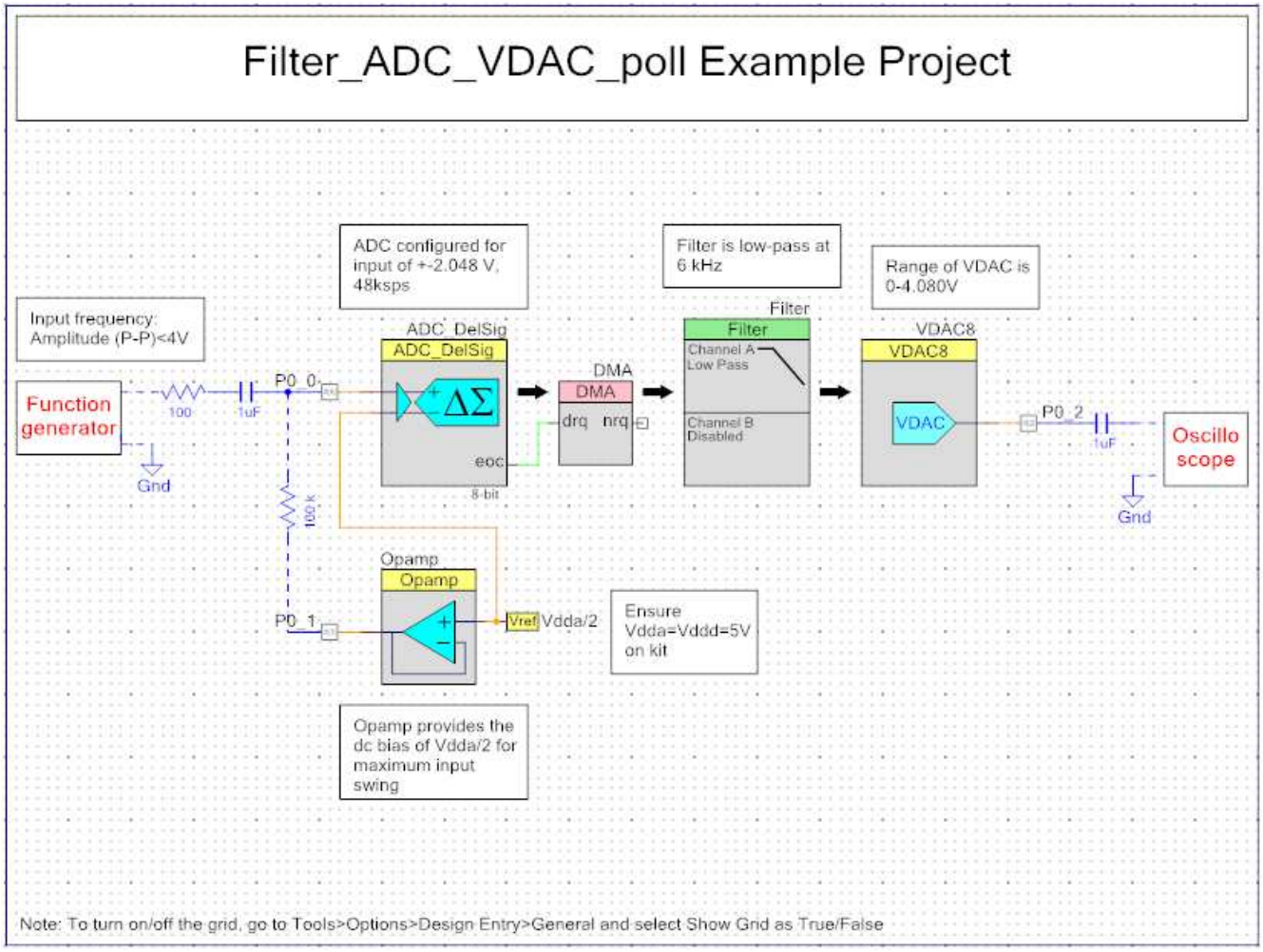

Figure 2. TopDesign Schematic

The ADC_DelSig component configuration window is shown below in Figure 3. The Opamp is simply configured in follower mode, so that its output voltage is equal to Vdda/2. The DMA is configured using the DMA wizard to transfer one byte from the ADC output register to the Filter channel A input register on each rising edge of the ADC EOC signal.

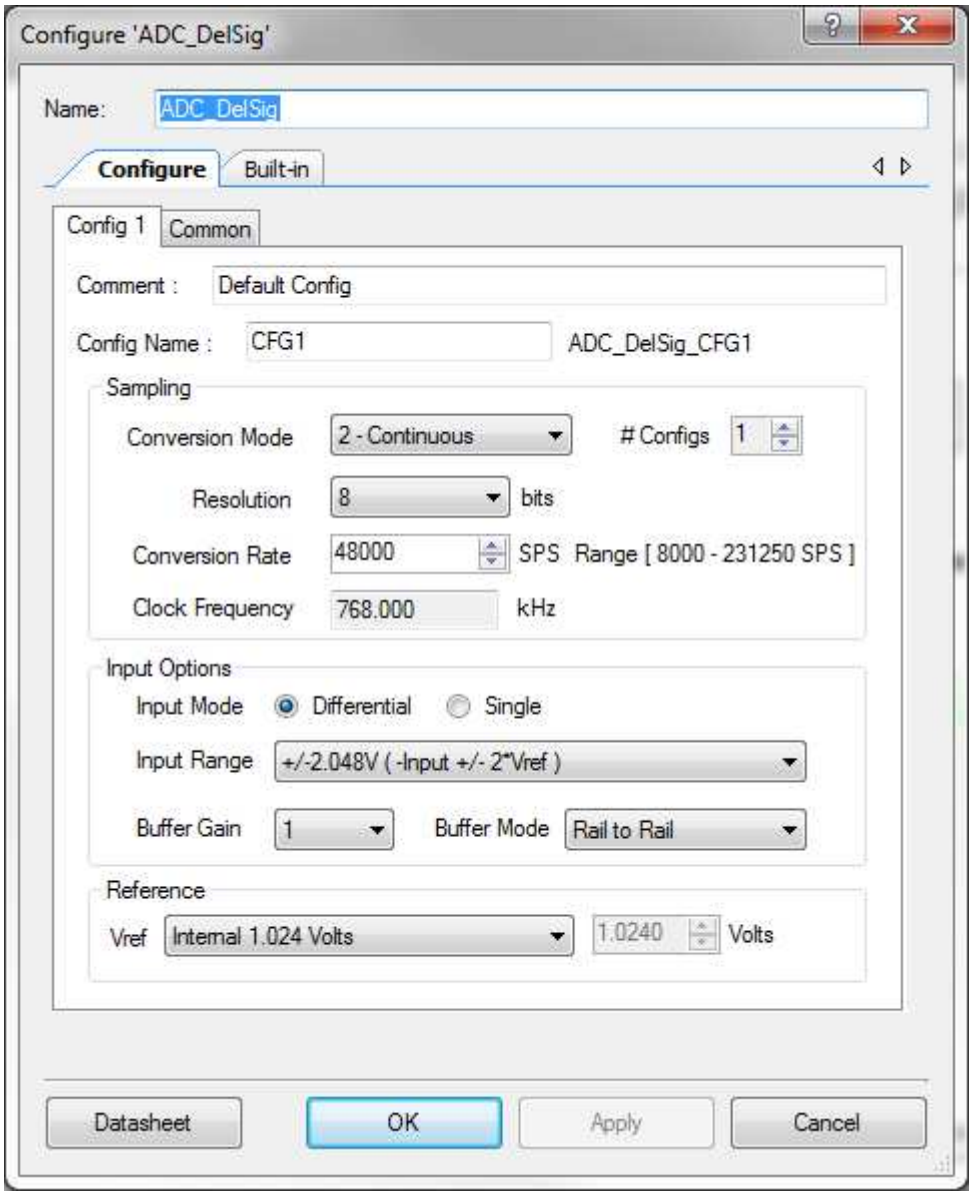

Figure 3. ADC_DelSig Component Configuration

The Filter sample rate is set to be same as the sample rate on the ADC – 48 ksps. Data ready signal is set to 'Polled', and the rest of the parameters which configure the Filter response are as shown in Figure 4. Notably, the filter is a single stage, FIR, low-pass filter with a cutoff of 6 kHz. By changing these parameters, you can easily create a Filter of your choice.

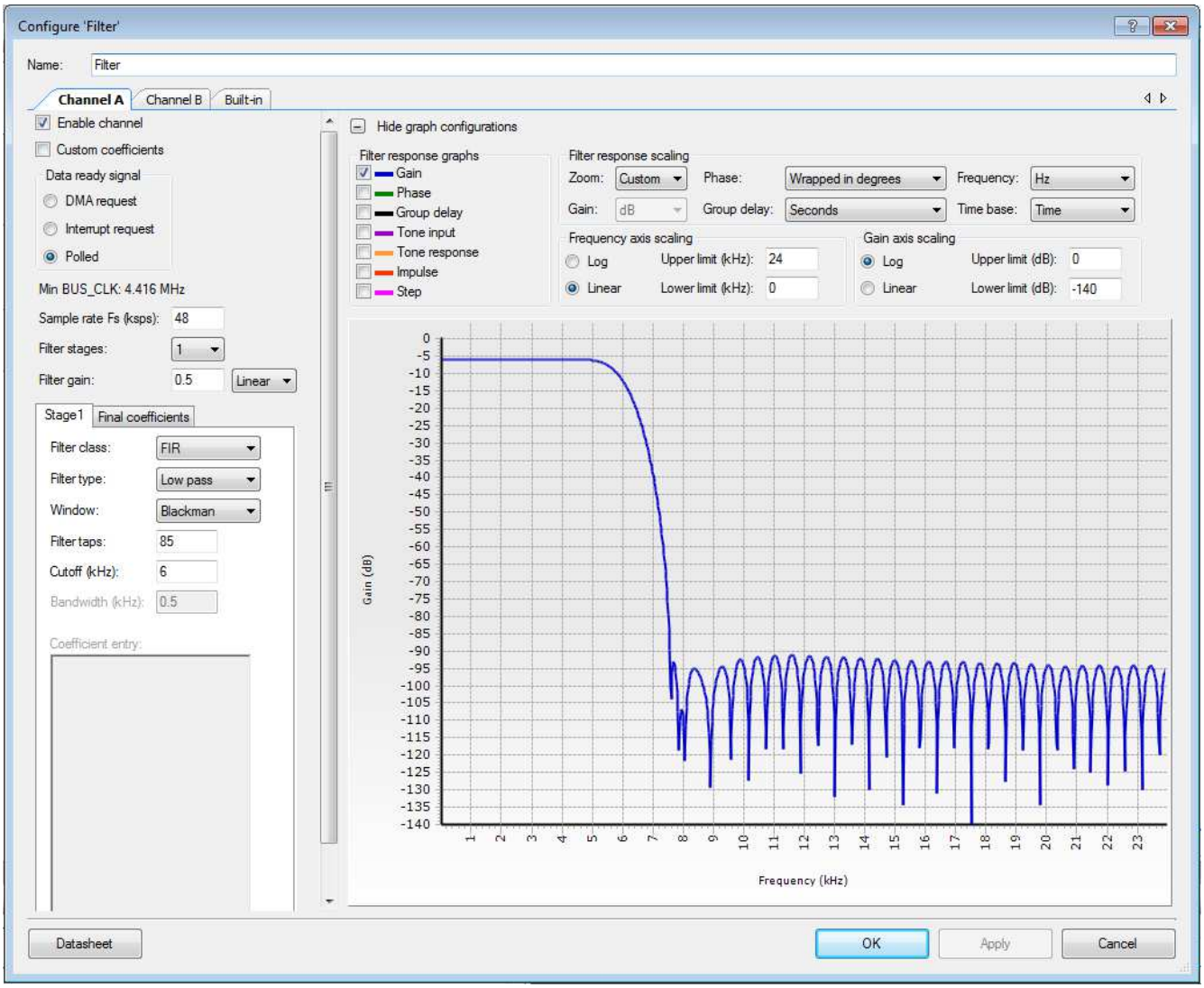

Figure 4. Filter configuration window

The VDAC is configured for an output range of 0-4.080V for the sake of easily verifying the output.

Project Description

The c code for this project goes through the following steps. The components are first initialized and started in the main function. The Filter component is configured to work in polling mode. The DMA is configured to transfer the ADC output to the input data register of the Filter's channel A. The code then enters a forever loop and waits (polls) for the Filter completion.

Each time the ADC completes a conversion, it triggers the DMA to transfer the converted data to the Filter. The Filter in turn, processes this sample and triggers an 'interrupt' signal when a data is ready in its holding register. The firmware then polls the status register for this interrupt. Since the VDAC can take in only positive data, the Filter channel A output is converted into an unsigned form and then written to the VDAC data register. The Filtered result can then be verified by observing the VDAC output.

Expected Results

The VDAC output should behave as a low-pass filtered output with cut-off at 6 kHz.

Related Material

Example Projects

- ADC_DMA_VDAC
- Filter_ADC_VDAC

Application Notes

- AN2099 - PSoC® 1, PSoC 3, and PSoC 5 - Single-Pole Infinite Impulse Response (IIR) Filters
- AN52705 - PSoC® 3 and PSoC 5 - Getting Started with DMA
- AN61102 - PSoC® 3 and PSoC 5 - ADC Data Buffering Using DMA
- AN57821- PSoC® 3 and PSoC 5 Mixed Signal Circuit Board Layout Considerations

Training

- PSoC 3 and PSoC 5 104: Introduction to Analog Peripherals

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© 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.

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.