# PSoC® 3 / PSoC 5- FIR Filter with 8-Bit Streaming

## CE58352

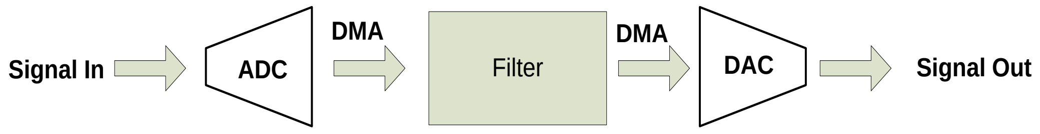
**Associated Part Families** CY8C38xx/CY8C55xx
**Software**: PSoC® Creator™
**Related Hardware**: CY8CKIT-001
**Author:** Pavan Vibhute

## Project Objective

CE58352 demonstrates a usage of the filter component through 8-bit FIR low pass filter. This code example also explains how to configure DMA to stream 8-bit data through the ADC → Filter → DAC chain.

## Overview

The Filter component in this code example is configured to implement a FIR low pass filter with 1 kHz cut off frequency. Two DMA channels are used in this project. One DMA channel moves data form ADC to filter input register and the other DMA channel moves data from filter output register to DAC.



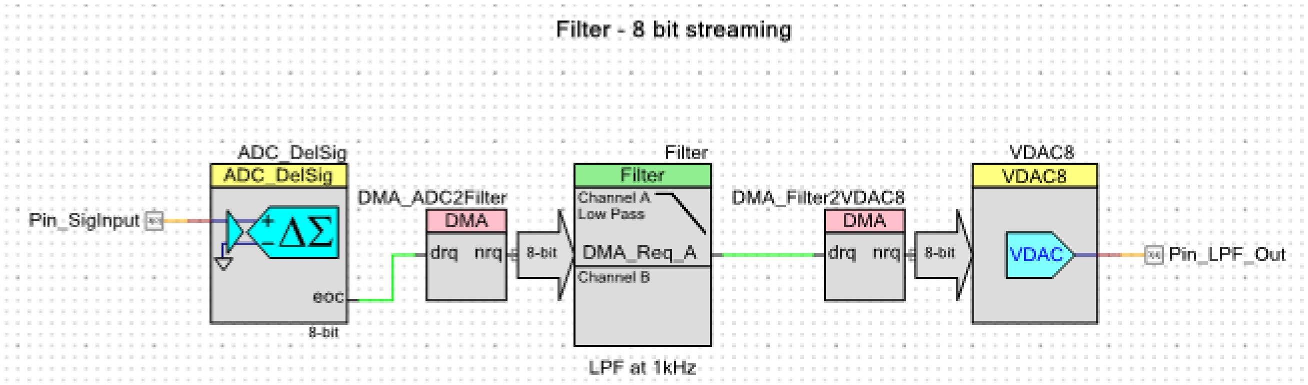
Note: This code example **cannot** be used for High pass and band pass filter implementations.

## Component List

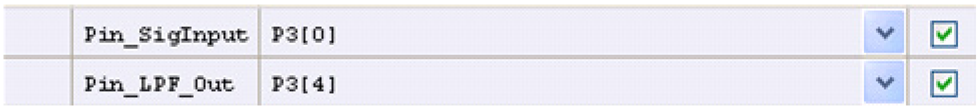
| Instance Name | Component Name | Component Category | Comments |
|---|---|---|---|
| Pin_SigInput | Analog Pin | Ports and Pins | Input pin for feeding the signal |
| ADC_DelSig | ADC_DelSig | Analog → ADC | Digitizes the signal |
| DMA_ADC2Filter | DMA | System | To transfer from ADC to Filter |
| Filter | Filter | Filters | The digital filter(FIR) component |
| DMA_Filter2VDAC8 | DMA | System | To transfer from Filter to DAC |
| VDAC8 | Voltage DAC | Analog → DAC | Converts the filtered signal back to analog |
| Pin_LPF_Out | Analog pins | System | Output pin to see the filtered output |

## Top Design

The following figure shows Filter 8-bit Streaming.



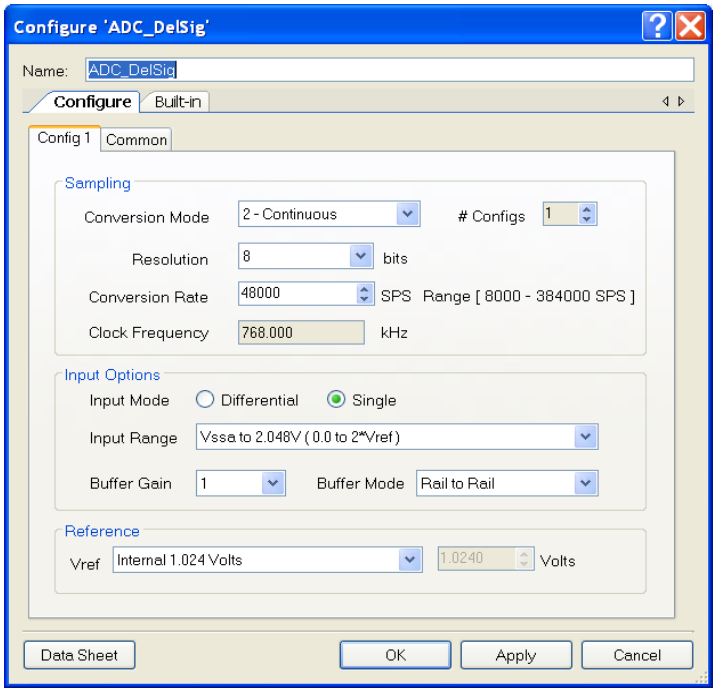
The following figure shows Pin Placements (as in .cydwr file)



| | | | |
|---|---|---|---|
| | Pin_SigInput | P3[0] | ✓ |
| | Pin_LPF_Out | P3[4] | ✓ |

## Component Configuration

### ADC_DelSig



Configure 'ADC_DelSig'

Name: ADC_DelSig

Configure | Built-in

Config 1 | Common

Sampling

Conversion Mode: 2 - Continuous    # Configs: 1

Resolution: 8 bits

Conversion Rate: 48000 SPS    Range [ 8000 - 384000 SPS ]

Clock Frequency: 768.000 kHz

Input Options

Input Mode: Differential [✓ Single]

Input Range: Vssa to 2.048V ( 0.0 to 2*Vref )

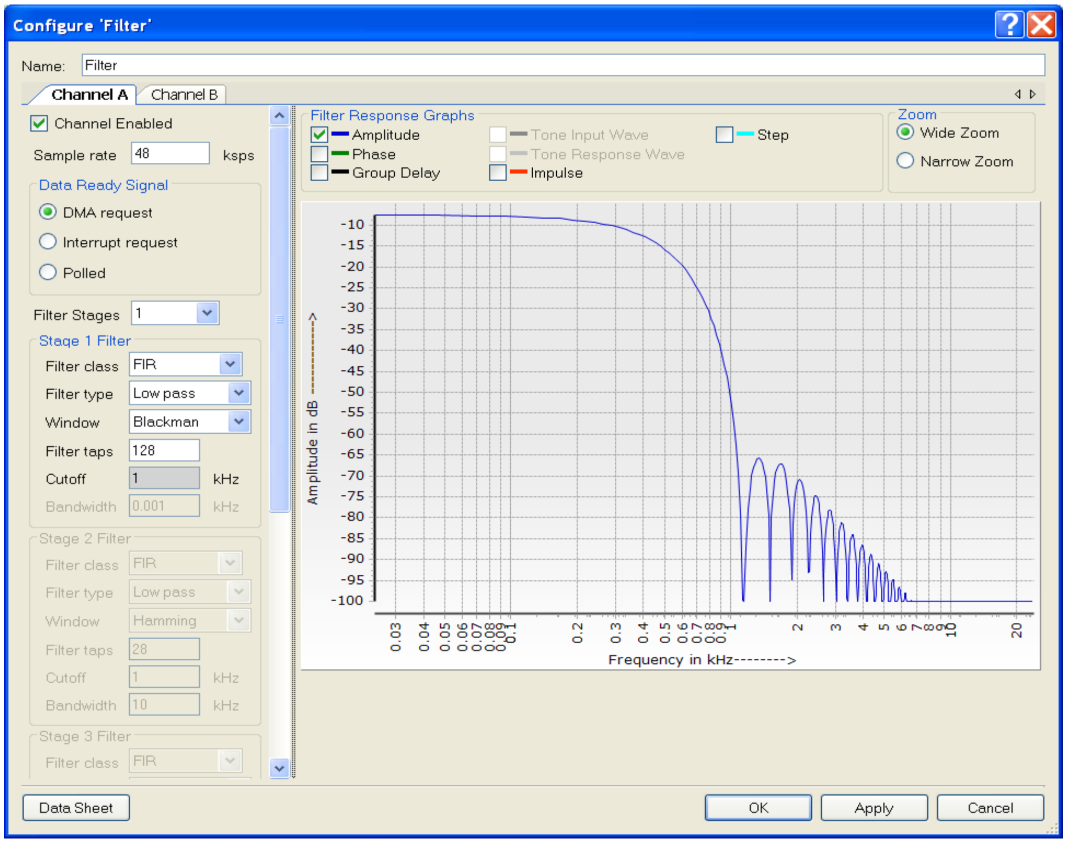
Buffer Gain: 1    Buffer Mode: Rail to Rail

Reference

Vref: Internal 1.024 Volts    1.0240 Volts

The ADC is configured in 8 bit mode and sampling rate of 48 ksps. The input range of the ADC is set to 0 to 2.048V. Make sure that the input voltage of the signal chain is within the ADC input range.

## Filter



Configure 'Filter'

Name: Filter

Channel A | Channel B

[✓] Channel Enabled

Sample rate 48 ksps

Data Ready Signal

- [● DMA request]
- ○ Interrupt request
- ○ Polled

Filter Stages 1

Stage 1 Filter

| | | |
|---|---|---|
| Filter class | FIR | |
| Filter type | Low pass | |
| Window | Blackman | |
| Filter taps | 128 | |
| Cutoff | 1 | kHz |
| Bandwidth | 0.001 | kHz |

Stage 2 Filter

| | | |
|---|---|---|
| Filter class | FIR | |
| Filter type | Low pass | |
| Window | Hamming | |
| Filter taps | 28 | |
| Cutoff | 1 | kHz |
| Bandwidth | 10 | kHz |

Stage 3 Filter

Filter class FIR

Filter Response Graphs

- [✓] Amplitude
- [ ] Phase
- [ ] Group Delay
- [ ] Tone Input Wave
- [ ] Tone Response Wave
- [ ] Impulse
- [ ] Step

Zoom

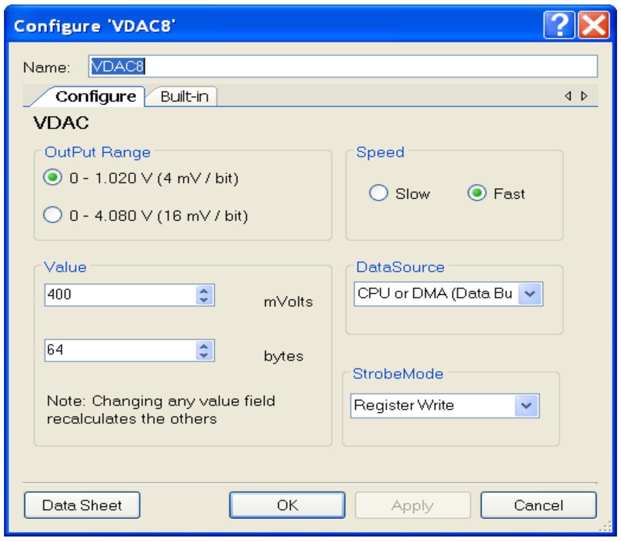
- [● Wide Zoom]
- ○ Narrow Zoom

Data Sheet | OK | Apply | Cancel

- The Filter component available in PSoC® Creator™ uses the hardware Digital Filter Block (DFB) to create Finite Impulse Response (FIR) filters.
- The DFB has two pairs of input and output registers (Staging / Holding Registers) for enabling stereo data processing. This makes it possible to have two independent, parallel channels for the filter component – Channel A and Channel B. Channel A uses Staging Register A and Holding Register A for data input and output respectively. Similarly Channel B uses Staging Register B and Holding Register B. In this example since we want to process only a single channel data from the ADC, only one filter channel (channel A), is used. Channel A is enabled in the filter component.
- The filter component allows cascading up to four filters stages in a single channel. For example, if you want to implement a Band Stop Filter (BSF) to attenuate 50Hz noise and then filter out frequencies above 1 kHz (LPF), the number of filter stages should be set to two; the first stage configured as a 50Hz BSF and the second stage as a 1 kHz LPF.

  In this code example only a low pass filter is used and hence the number of filter stages is set to 1.
- The filter type is configured as a LPF with 1 kHz cut off frequency.
- A FIR filter of order N requires (N+1) filter taps. The total number of available filter taps (coefficients) for the FIR filter component is limited to 128. The allocation of filter taps for each filter stage and filter channel is done in the Filter component configuration window. In this code example all the available 128 taps are used by the LPF, since it is the only filter used in the design.
- Sample Rate is the filter's data input rate and is used by the component for designing the filter coefficients. In this example, the input data for filter comes from the ADC and hence the sample rate is same as ADC sampling rate – 48 ksps.
- Window**:** The filter component allows for multiple windowing methods to tune filter responses .The characteristics of various window types are given in the component datasheet. In this example, the type of window is set to 'Blackman', to get steeper transition band.

## VDAC8


Configure 'VDAC8'

Name: VDAC8

Configure | Built-in

VDAC

OutPut Range
[✓ 0 - 1.020 V (4 mV / bit)]
0 - 4.080 V (16 mV / bit)

Speed
Slow [✓ Fast]

Value
400 mVolts
64 bytes

Note: Changing any value field recalculates the others

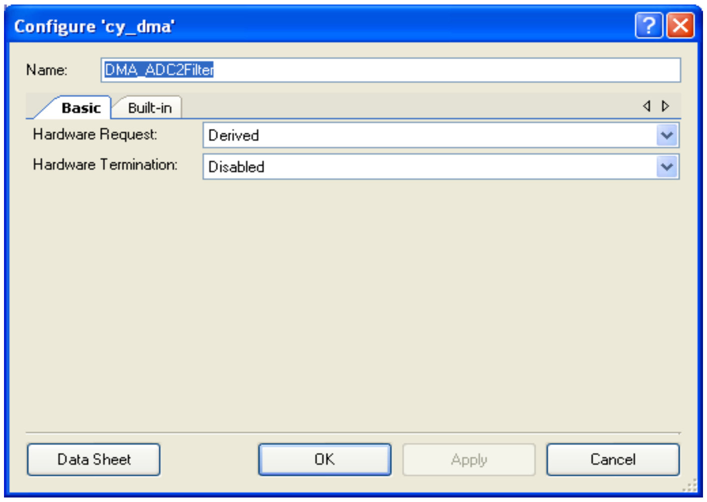
DataSource: CPU or DMA (Data Bu

StrobeMode: Register Write

**Note:** The ADC is configured for an input range of 0 V to 2.048 V whereas; the DAC is configured in the range of 0 V to 1.024 V. Therefore, there is corresponding scaling of 0.5 on the output of the DAC.

## DMA

This code example uses two DMA channels: One channel to move data from ADC to filter and the other to move the output of filter to DAC. DMA channel component renamed as ‘DMA_ADC2Filter’ and ‘DMA_Filter2VDAC8’ are configured as shown in the following figure.


Configure 'cy_dma'

Name: DMA_ADC2Filter

Basic | Built-in

Hardware Request: Derived

Hardware Termination: Disabled

- When hardware request is set to “Derived”, the PSoC Creator enables the hardware request and configures the trigger mode (either level or edge) based on the peripheral to which DMA is connected.
- Once the hardware request is set, the data request terminal (drq) of the DMA component becomes visible.
- The drq terminal of the DMA should be connected to the DMA trigger signal. In this example, the DMA channel - ‘DMA_ADC2Filter’ is configured to transfer data from ADC to filter on each End of Conversion(EOC) signal from ADC. Hence the drq of the channel is connected to ADC EOC line.
- Similarly the DMA channel - ‘Filter2VDAC8’ is triggered using the DMA data ready signal from the filter.

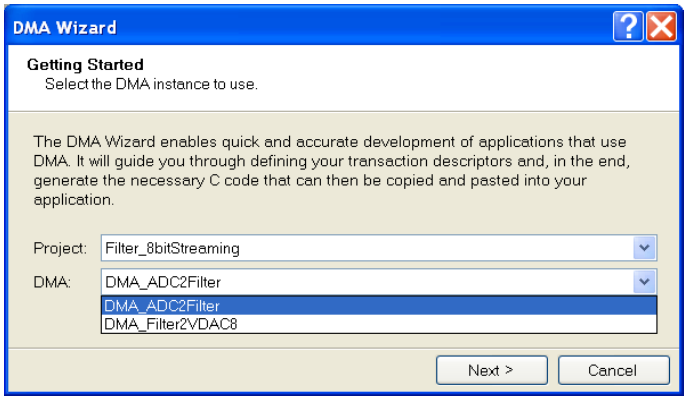
### Configuring the DMA using DMA Wizard:

The two DMA components on this project should have their configuration completed using the DMA wizard. The DMA wizard generates the code that has to be placed in the user code area for configuring the DMA channel. For more information on DMA configuration, refer AN52705 -Using DMA on PSoC 3 / PSoC 5 .

Following are the steps for setting up one of the DMA channels.

## Step 1: Selecting DMA Channel

Select the **DMA wizard** under the **Tools** menu. This opens a window for channel selection.



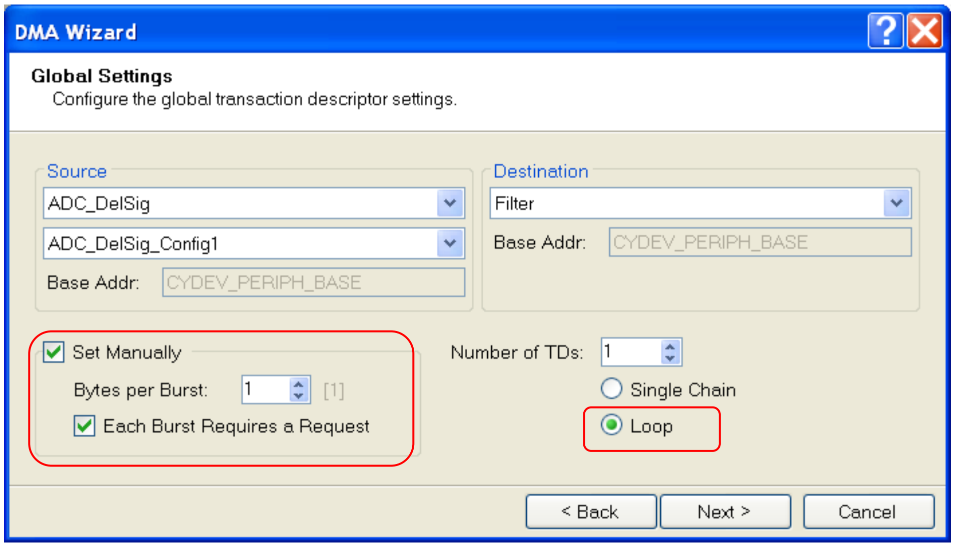
Getting Started
Select the DMA instance to use.

The DMA Wizard enables quick and accurate development of applications that use DMA. It will guide you through defining your transaction descriptors and, in the end, generate the necessary C code that can then be copied and pasted into your application.

Project: Filter_8bitStreaming
DMA: DMA_ADC2Filter
DMA_ADC2Filter
DMA_Filter2VDAC8

Select the DMA channel to configure it.

## Step 2: DMA Initialization



Global Settings
Configure the global transaction descriptor settings.

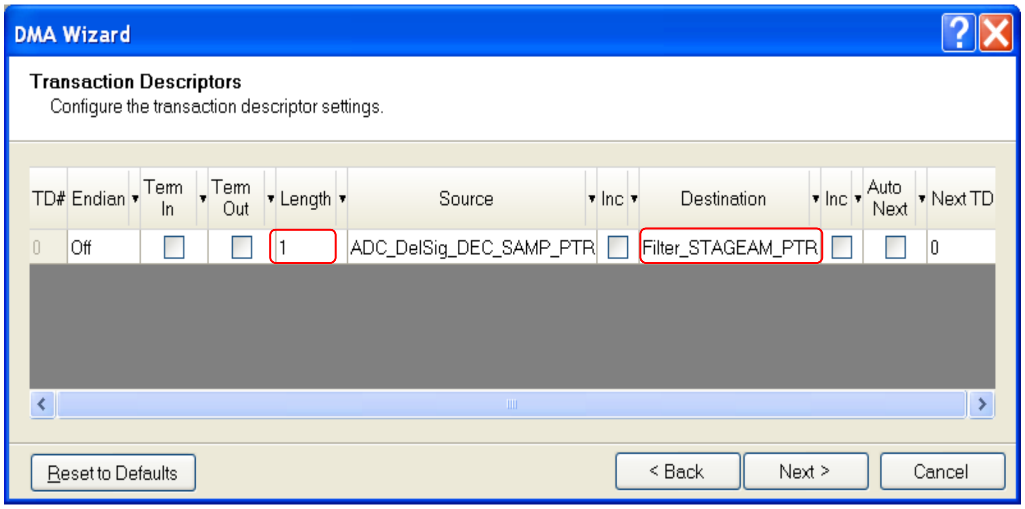
Source: ADC_DelSig, ADC_DelSig_Config1, Base Addr: CYDEV_PERIPH_BASE
Destination: Filter, Base Addr: CYDEV_PERIPH_BASE

[✓] Set Manually
Bytes per Burst: 1 [1]
[✓] Each Burst Requires a Request

Number of TDs: 1
Single Chain [✓ Loop]

Here, the source for the DMA for the first channel is ADC Delsig, and the destination is filter. The channel is configured only for one transaction i.e. from ADC to filter, and therefore only one transaction descriptor (TD) is used. The TD has to repeat itself after every transfer is complete and therefore, the **loop** option is selected**.**

**Step 3: TD Configuration**



**Transaction Descriptors**

Configure the transaction descriptor settings.

| TD# | Endian | Term In | Term Out | Length | Source | Inc | Destination | Inc | Auto Next | Next TD |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Off | ☐ | ☐ | 1 | ADC_DelSig_DEC_SAMP_PTR | ☐ | Filter_STAGEAM_PTR | ☐ | ☐ | 0 |

In this case, the TDs are configured to transfer one byte. Hence length is set to 1. The source and destination registers are also set in this step. Note that the destination is the middle byte of filter Channel A's 24 bit input staging register (Filter_STAGEAM_PTR). The reason for moving data to middle byte of staging register is explained in the operation section.

After the configuration is complete, the wizard generates the code required for configuring the TD. Copy the generated code into main.c file to configure the DMA Channel.

Similar procedure is followed for the DMA_Filter2VDAC8 DMA channel. The only difference is, the source is the Filter_HOLDAM_PTR and destination is the VDAC8_Data_PTR.

## Design Wide Resources

This project uses the default configuration of design wide resources. Refer to *Filter_8bitStreaming.cydwr* file

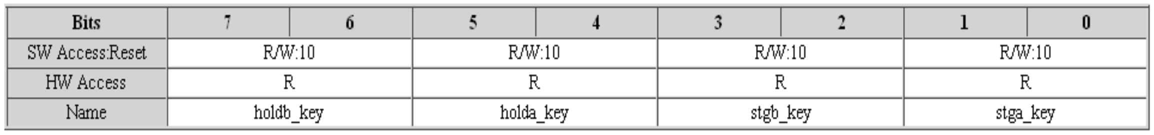
## Operation

A Low Pass Filter with the Blackman Window is demonstrated in this project. The cut off frequency of the Low pass filter is 1 kHz.

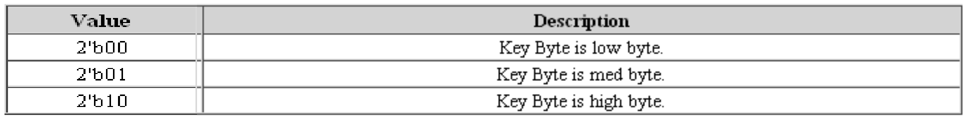
- The ADC is configured to sample the input wave at 48 ksps and its output is unsigned. The ADC samples are transferred to filter by DMA.
- The filter is set to operate at the same sampling rate the signal is sampled at. If the sampling rate value in the filter configuration and the actual rate at which the signal is sampled do not match, then the filter does not give the correct characteristic.
- The Digital Filter Block (DFB) in PSoC 3 / PSoC 5 is designed to work with signed numbers(two's complement format) whereas the DAC in PSoC 3 / PSoC 5 is unipolar and accepts input in unsigned(offset binary) format. ADC output data is in unsigned format when configured in single ended mode and two's complement format in differential mode. Special care needs to be taken to manage the difference in data format, while moving data through the signal chain. In order to form a ADC>Filter>DAC chain in PSoC 3 / PSoC 5, you can do either of the following
- If CPU/firmware is used for data movement, the signed output of the filter can be converted to unsigned format, in firmware, for moving to VDAC.
- If DMA is used for data movement, data format conversions are not possible within the signal chain. Hence while forming an 8 bit signal chain using DMA, use the unsigned (offset binary) format for the signal chain and move the ADC output to the middle or lower byte of Filter (not the high byte). If the unsigned 8 bit data is moved to the middle byte of the filter, the filter input will be in the range 0x000000 to 0x00ff00. The DFB treats this as a positive number and do the calculations and produces a positive output (unsigned).The DMA can then move data from the middle or lower byte of the Filter to the DAC without any format conversions. The disadvantage of this method is that a high pass or band pass filter cannot be implemented in this case.
- In this example the whole signal chain follows an offset binary structure. The filter treats this data input as two's compliment positive data (input signal with an offset). In case of high pass and band pass filters, this offset (dc content) will be removed by the filter and hence the output of those filters will not be offset binary. Hence this code example cannot be used for Band Pass and High Pass Filters.

- The DFB staging or holding registers are coherency key protected .The coherency key indicates which byte in the 3 byte DFB staging/holding registers is written/read last. The DFB Filter component sets the default coherency key as High byte. It should be changed to middle byte in this code example, as DMA reads and writes only to the middle byte.

  Refer to the DFB[0..0]_COHER register to learn more about filter coherency settings (Refer to the TRM register book for details)



| Bits | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SW Access:Reset | R/W:10 | | R/W:10 | | R/W:10 | | R/W:10 | |
| HW Access | R | | R | | R | | R | |
| Name | holdb_key | | holda_key | | stgb_key | | stga_key | |



| Value | Description |
|---|---|
| 2'b00 | Key Byte is low byte. |
| 2'b01 | Key Byte is med byte. |
| 2'b10 | Key Byte is high byte. |

```
/* Filter Coherency set to mid byte */
Filter_COHER_REG = 0x55;
```

- The result of the filter from its holding register middle byte is transferred to VDAC. The voltage range of VDAC is set to 1.024 V while the range of ADC is 2.048 V. Therefore, compared to the input wave, the DAC output is scaled to half.

## Hardware Connections

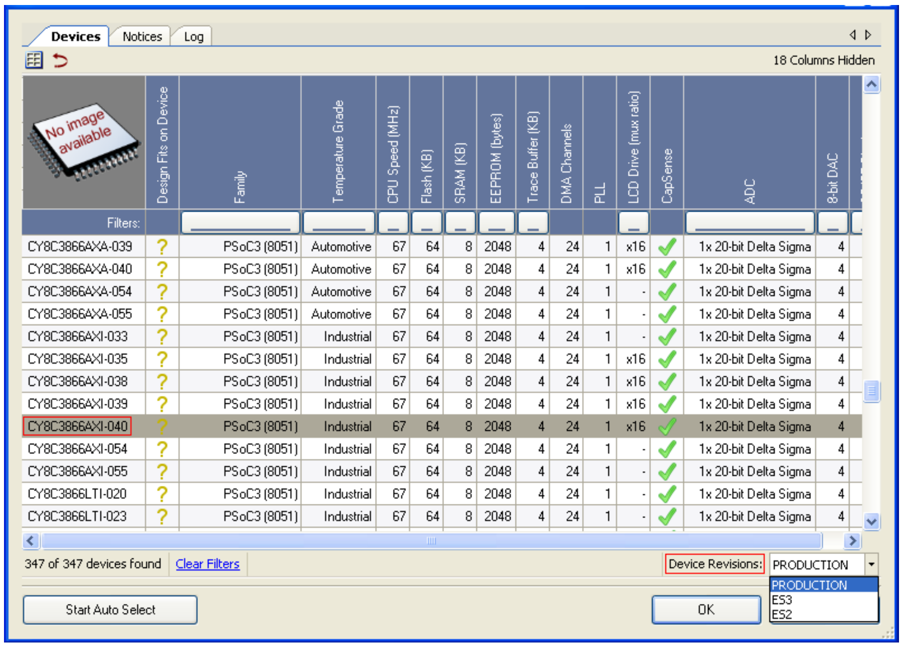
This project can be tested on the CY8CKIT-001 development board. The following connections are done on the board to make the project work.

- Connect Signal Generator output to Pin P3[0]. Make sure that output amplitude varies between 0 to 2 V. Do not feed negative voltage to ADC input.
- Connect the Scope to pin P3[4] to view filter output.
- For rest of the basic settings of the DVK, refer to the CY8CKIT-001 PSoC Development Kit Board Guide that is supplied with the kit.

## Output

- Use device selector (Project->Device Selector) window in PSoC Creator to select the appropriate device and Device Revision.

  If you are using PSoC3 device (for example: CY8C3866AXI-040) with production revision, then use the following selection.



| | Design Fits on Device | Family | Temperature Grade | CPU Speed (MHz) | Flash (KB) | SRAM (KB) | EEPROM (bytes) | Trace Buffer (KB) | DMA Channels | PLL | LCD Drive (mux ratio) | CapSense | ADC | 8-bit DAC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CY8C3866AXA-039 | ? | PSoC3 (8051) | Automotive | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | x16 | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXA-040 | ? | PSoC3 (8051) | Automotive | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | x16 | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXA-054 | ? | PSoC3 (8051) | Automotive | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | - | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXA-055 | ? | PSoC3 (8051) | Automotive | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | - | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXI-033 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | - | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXI-035 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | x16 | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXI-038 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | x16 | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXI-039 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | x16 | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXI-040 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | x16 | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXI-054 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | - | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866AXI-055 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | - | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866LTI-020 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | - | ✓ | 1x 20-bit Delta Sigma | 4 |
| CY8C3866LTI-023 | ? | PSoC3 (8051) | Industrial | 67 | 64 | 8 | 2048 | 4 | 24 | 1 | - | ✓ | 1x 20-bit Delta Sigma | 4 |

347 of 347 devices found Clear Filters Device Revisions: PRODUCTION

Similarly, select appropriate device number (eg. **CY8C5588AXI-060)** to work with PSoC 5 Device family.

**Note:** For engineering samples, device revision is marked on the package as part of the device number. Production silicon will not have ES marking.

- Build the project and program the chip.
- The input is provided as a sine waveform form a function generator at pin 3[0]. Ensure that the input amplitude is lesser than 2 V with an offset of 1 V.
- Vary the input frequency and observe the LPF filter response (1 kHz cut-off frequency) by monitoring filter output on Pin P3[4].

## Suggested Application Notes/Code Examples

- Using DMA on PSoC 3 / PSoC 5 – AN52705.
- **DMA Peripheral** Transfer in PSoC® 3 / PSoC 5

## Document History

**Document Title: PSoC® 3 / PSoC 5- FIR Filter with 8-Bit Streaming - CE58352**

**Document Number: 001-58352**

| Revision | ECN | Orig. of Change | Submission Date | Description of Change |
|---|---|---|---|---|
| ** | 2828370 | PVKV | 01/12/2010 | New Code Example |
| *A | 2947292 | PVKV | 06/09/2010 | Updated to PSoC Creator Beta 4.1 and made the projects PSoC 5 compatible. |
| *B | 3012567 | PVKV | 08/25/2010 | Updated screenshots and technical changes |
| *C | 3178326 | ANMD | 02/21/2011 | Changed "EP" to "CE" and updated EP title to follow the new title format. Updated the document to add more explanation to Filter configuration, data movement operations and other component explanations. Updated all the screenshots. Removed version number from the component table. Removed prerequisites, Programming language etc. from EP header. |

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/