ADC to Filter – Dual Channel 16-Bit Streaming using DMA – PSoC® 3 / PSoC 5

EP58353

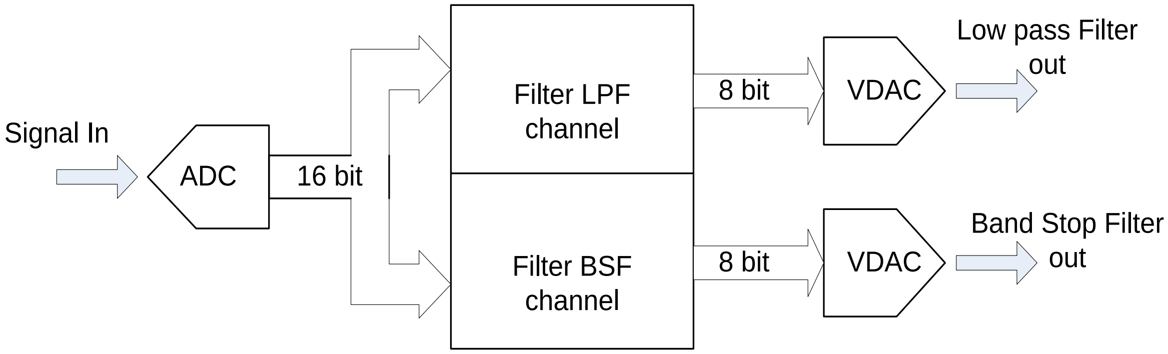
Associated Part Families CY8C38xx/CY8C55xx
Software: PSoC® Creator™
Related Hardware: CY8CKIT-001
Author: Anu MD

Project Objective

This project demonstrates dual channel 16-bit streaming from ADC to filter using DMA.

Overview

This example project demonstrates how to stream 16-bit data from ADC to filter using DMA. It also demonstrates how to enable dual channel filtering and stream in data through both the channels simultaneously. The Project implements two simultaneous channels of the filter, one implementing a low pass and other a band stop filter.

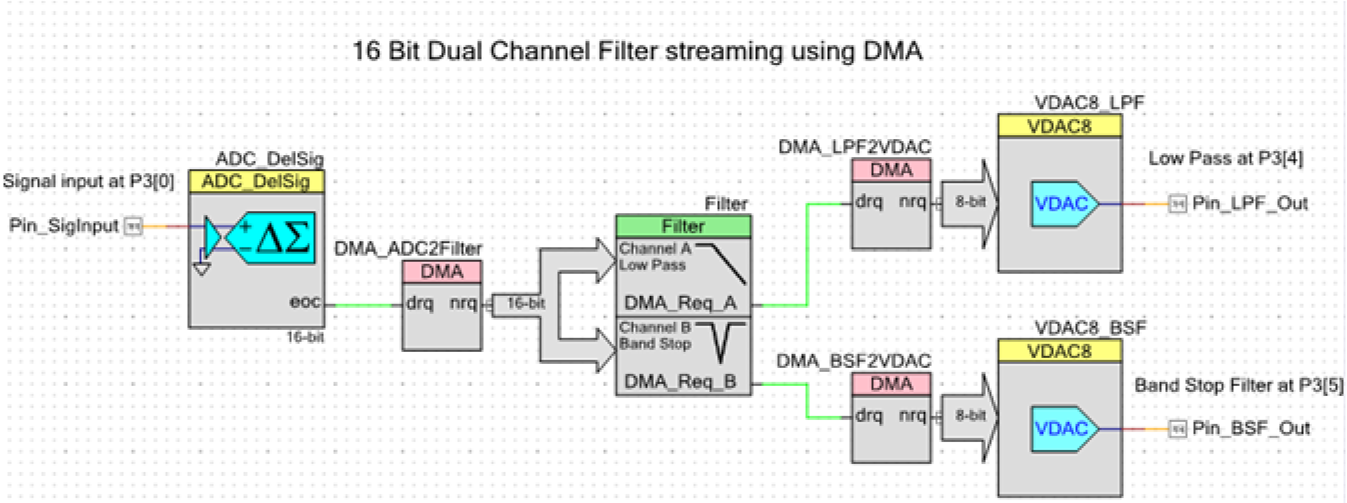


Component List

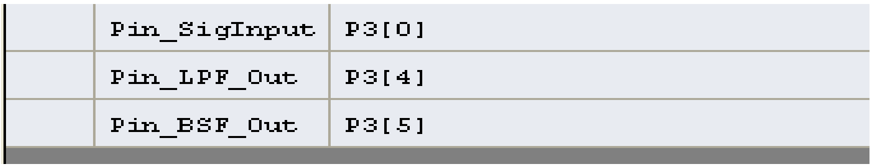
Instance Name	Component Name	Component Category	Comments
ADC_DelSig	ADC_DelSig	Analog → ADC	–
Filter	Filter	Filters	–
VDAC8_LPF	Voltage DAC	Analog → DAC	–
VDAC8_BSF	Voltage DAC	Analog → DAC	–
DMA_ADC2Filter	DMA	System	–
DMA_LPF2VDAC	DMA	System	–
Pin_SigInput	Analog Pin	Ports and Pins	–
Pin_LPF_Out	Analog Pin	Ports and Pins	–
Pin_BSF_Out	Analog Pin	Ports and Pins	–

Top Design

The following figure shows the components and their routing.



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

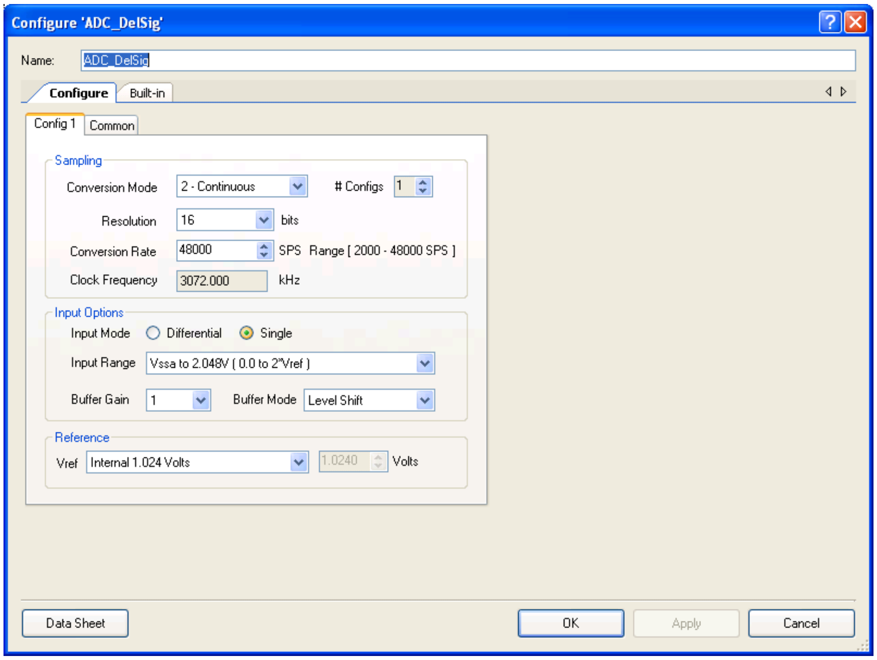


	Pin_SigInput	P3[0]
	Pin_LPF_Out	P3[4]
	Pin_BSF_Out	P3[5]

Component Configuration

ADC

This is configured as single ended 0 V to 2.048 V, 16-bit, 48 ksps ADC.



Configure 'ADC_DelSig'

Name: ADC_DelSig

Sampling

Conversion Mode: 2 - Continuous

Configs: 1

Resolution: 16 bits

Conversion Rate: 48000 SPS Range [2000 - 48000 SPS]

Clock Frequency: 3072.000 kHz

Input Options

Input Mode: ○ Differential ● Single

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

Buffer Gain: 1

Buffer Mode: Level Shift

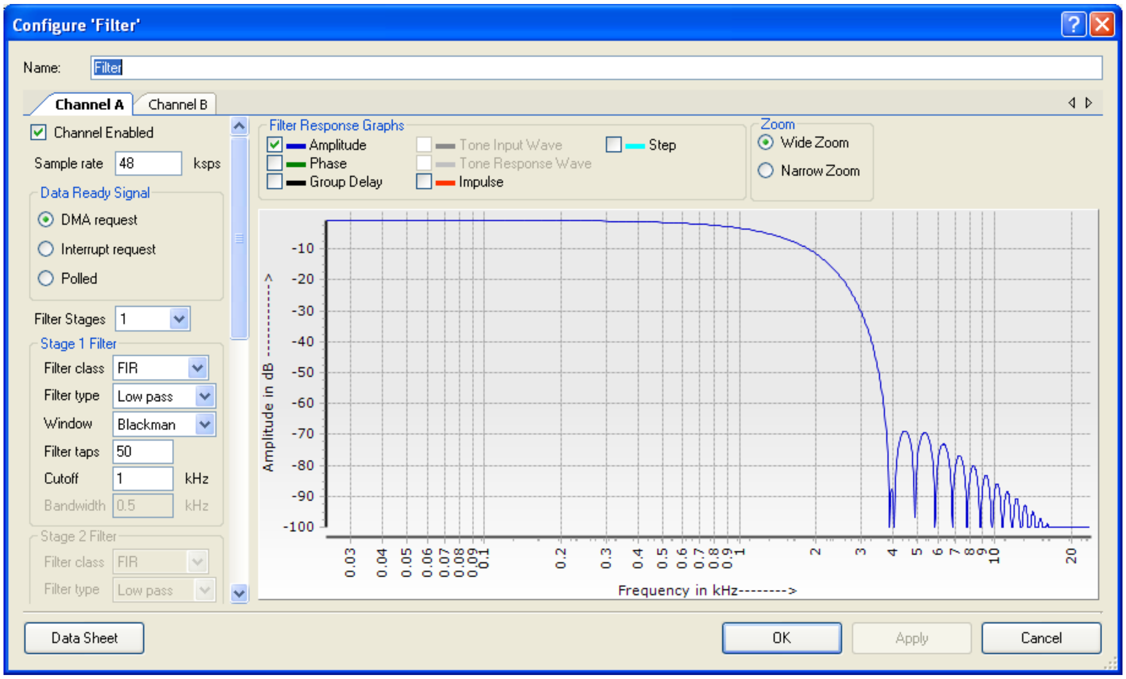
Reference

Vref: Internal 1.024 Volts 1.0240 Volts

Filter

Both the channels of the filter are enabled and configured as shown in the following figure.

Channel A Configured as a LPF with 1 kHz cut off frequency.



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 50

Cutoff 1 kHz

Bandwidth 0.5 kHz

Stage 2 Filter

Filter class FIR

Filter type Low pass

Filter Response Graphs

☑ Amplitude

□ Phase

□ Group Delay

□ Tone Input Wave

□ Tone Response Wave

□ Impulse

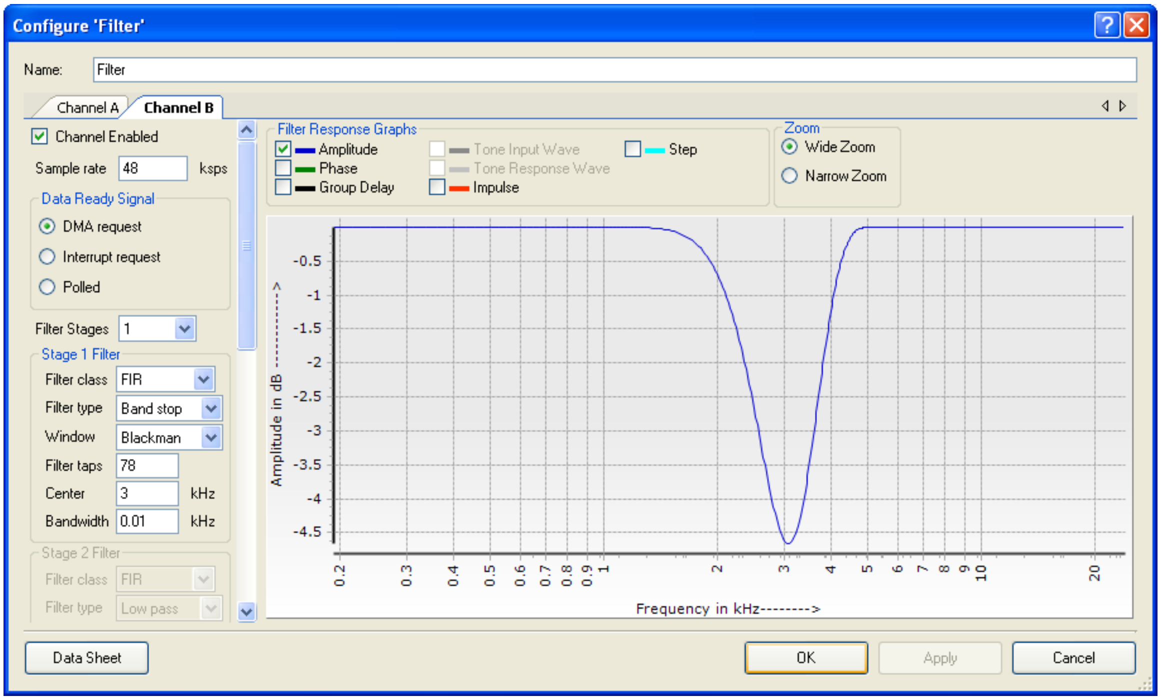
□ Step

Zoom

● Wide Zoom

○ Narrow Zoom

Channel B Click Channel B tab, Channel B is configured as a BSF with 3 kHz center frequency.



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 Band stop

Window Blackman

Filter taps 78

Center 3 kHz

Bandwidth 0.01 kHz

Stage 2 Filter

Filter class FIR

Filter type Low pass

Filter Response Graphs

☑ Amplitude

□ Phase

□ Group Delay

□ Tone Input Wave

□ Tone Response Wave

□ Impulse

□ Step

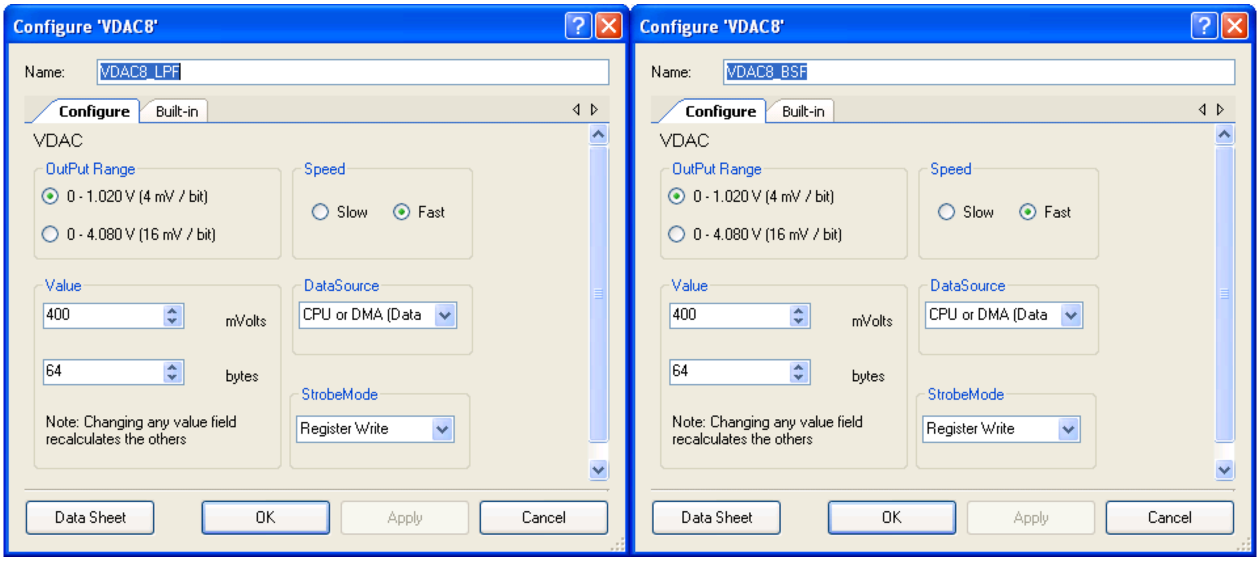
Zoom

● Wide Zoom

○ Narrow Zoom

VDAC8

The VDACs, VDAC8_LPF, and VDAC8_BSF take the following configuration.



Configure 'VDAC8'

Name: VDAC8_LPF

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

StrobeMode
Register Write

Data Sheet | OK | Apply | Cancel

Configure 'VDAC8'

Name: VDAC8_BSF

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

StrobeMode
Register Write

Data Sheet | OK | Apply | Cancel

Note that 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 to 1.024 V. Therefore, there is corresponding scaling in the output of the DAC.

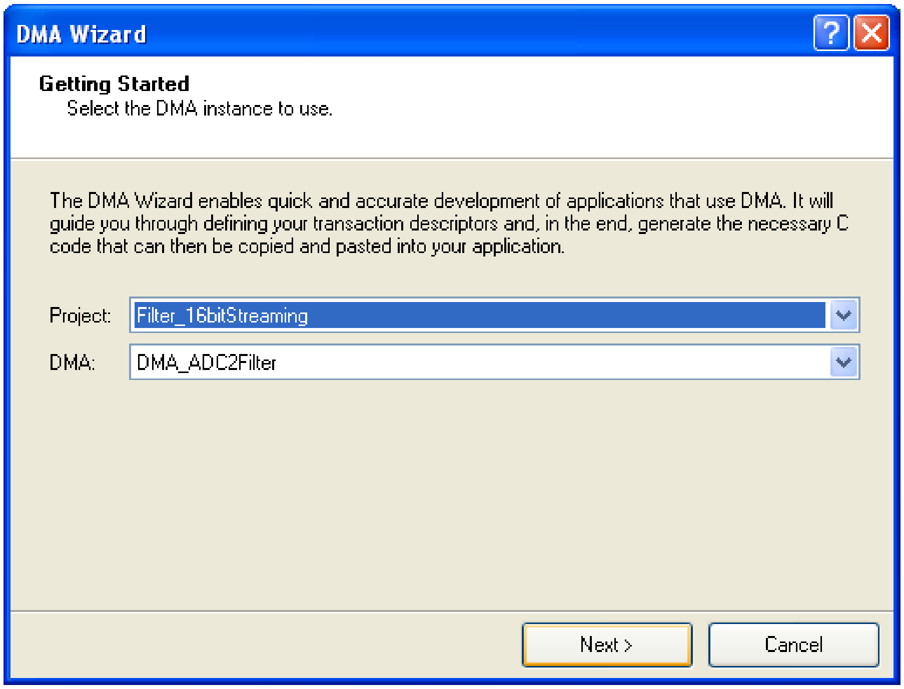
DMA

The three 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 Wizard, refer to Using DMA on PSoC 3 / PSoC 5 – AN52705.

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.



DMA Wizard

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.

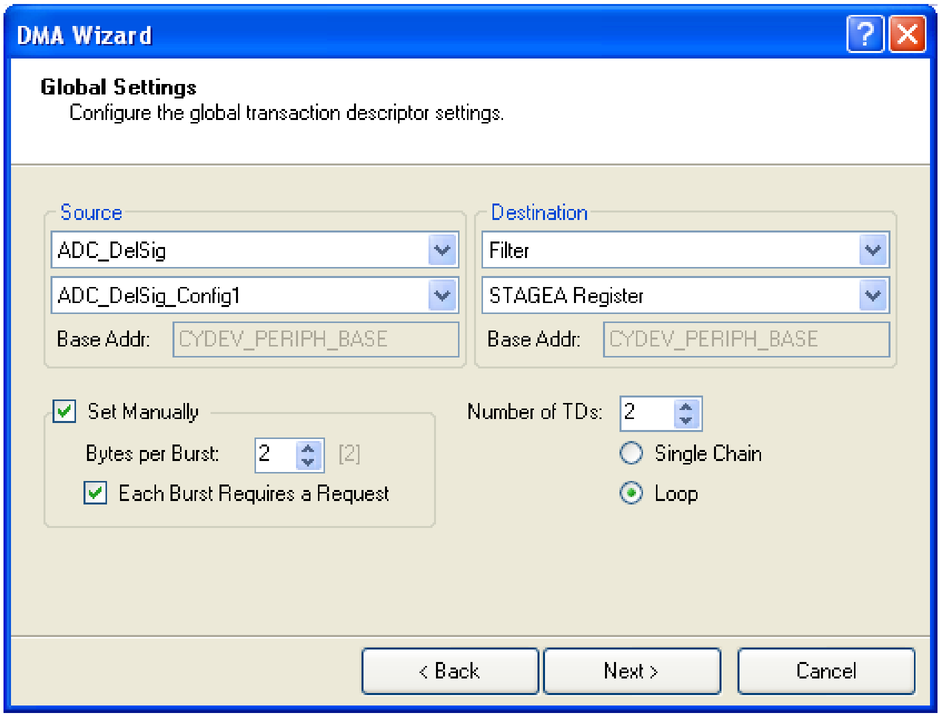
Project: Filter_16bitStreaming

DMA: DMA_ADC2Filter

Next > | Cancel

Select the DMA_ADC2Filter channel that requires configuration.

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

STAGEA Register

Base Addr: CYDEV_PERIPH_BASE

☑ Set Manually

Bytes per Burst: 2 [2]

☑ Each Burst Requires a Request

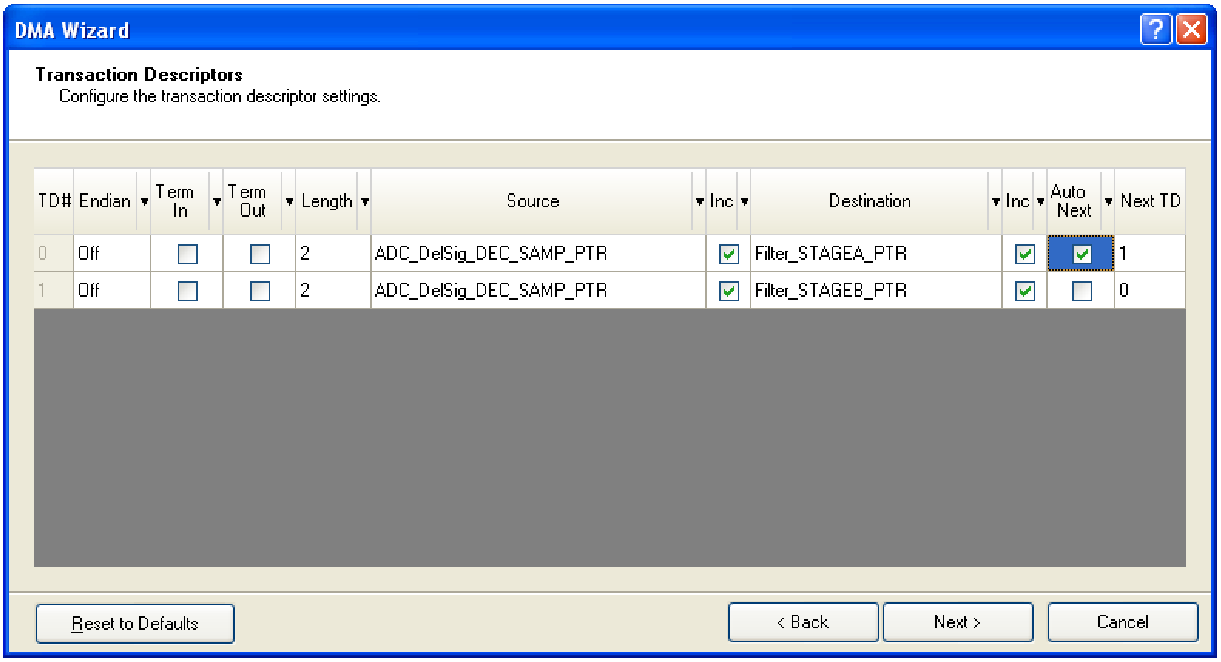
Number of TDs: 2

○ Single Chain

● Loop

Here, the source for the first channel is set as the ADC and destination is the staging registers of the filter. . The transformation here is two bytes per burst and every burst is triggered by a request. Two transaction descriptors (TD) are assigned, one each for each of the filter channels in the filter component.

Step 3: TD Configuration

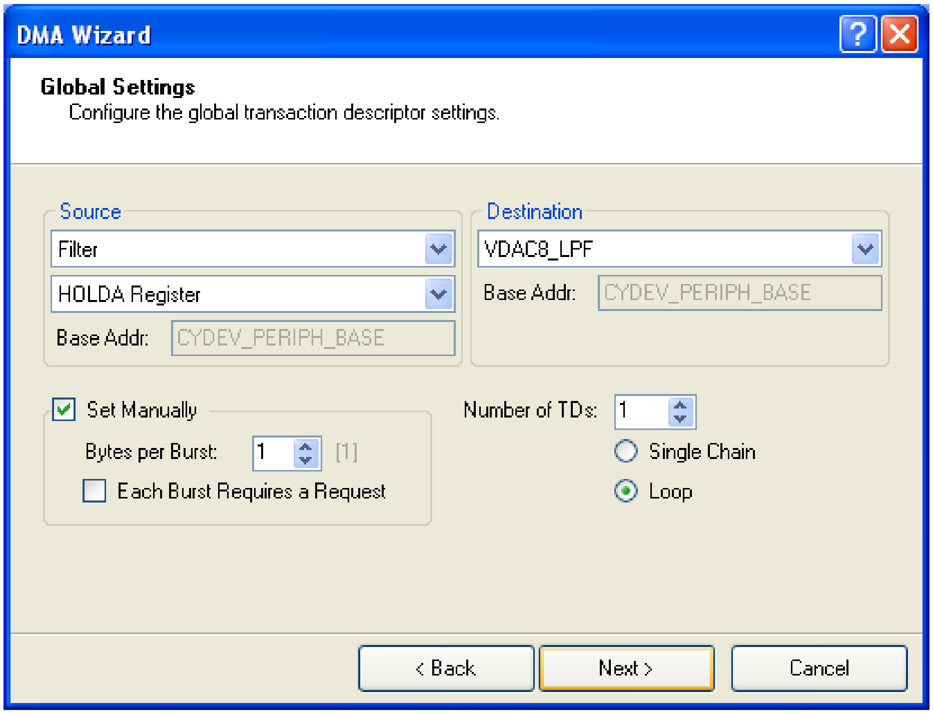


Transaction Descriptors

Configure the transaction descriptor settings.

TD#	Endian	Term In	Term Out	Length	Source	Inc	Destination	Inc	Auto Next	Next TD
0	Off	□	□	2	ADC_DelSig_DEC_SAMP_PTR	☑	Filter_STAGEA_PTR	☑	☑	1
1	Off	□	□	2	ADC_DelSig_DEC_SAMP_PTR	☑	Filter_STAGEB_PTR	☑	□	0

In this case, the TDs are configured to transfer two bytes. The source is set as the ADC sample register. The first TD is configured to a destination stage A (first filter channel input) register of the filter with source and destination increment option enabled. The Auto next is enabled so as to automatically activate the next TD. The second TD is configured similar to the first but the destination is to stage B (second filter channel input) register of the filter.



Global Settings

Configure the global transaction descriptor settings.

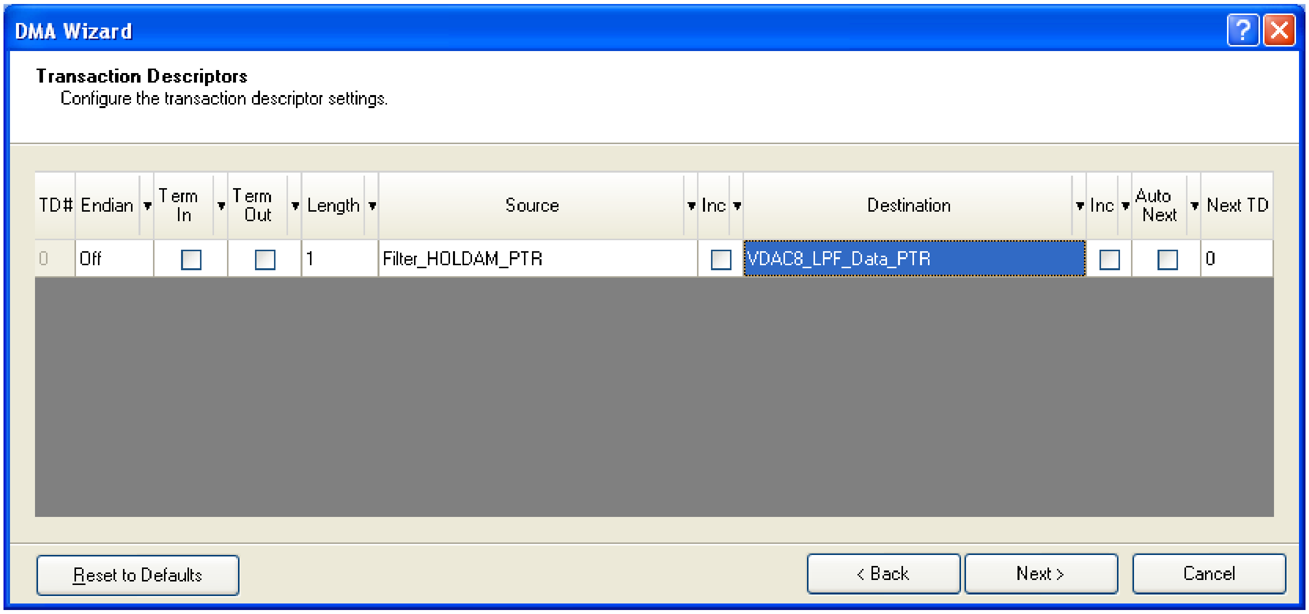
Source
- Filter
- HOLDA Register
- Base Addr: CYDEV_PERIPH_BASE

Destination
- VDAC8_LPF
- Base Addr: CYDEV_PERIPH_BASE

☑ Set Manually
- Bytes per Burst: 1 [1]
- □ Each Burst Requires a Request

Number of TDs: 1
- ○ Single Chain
- ● Loop

After the input stage DMA channel is configured we configure the output stage DMAs from the filter outputs to the VDACs. The first one is the DMA_LPF2VDAC where the Filter is set as source and VDAC8_LPF as the destination with single byte transfers.



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	Filter_HOLDAM_PTR	□	VDAC8_LPF_Data_PTR	□	□	0

The source is set as the Filter_HOLDAM_PTR and the destination is the VDAC8_LPF_Data_PTR. This acts as the DMA, transferring the data from the low pass filter channel (Channel A) of the filter to the corresponding VDAC.

The next DMA_BSF2VDAC is configured similarly with single-byte bursts. The only difference is, the source is Filter_HOLDBM_PTR and the destination is VDAC8_BSF_Data_PTR.

Design Wide Resources

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

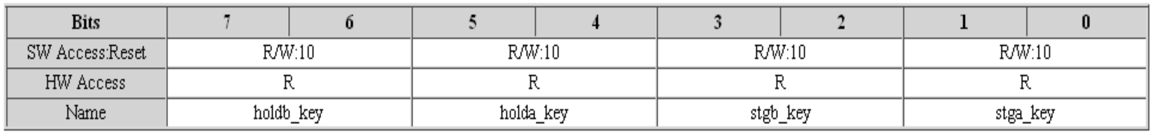
Operation

This project demonstrates how to stream in 16-bit ADC data through filter. The default coherency setting of Filter component has to be modified for proper streaming of data using DMA.

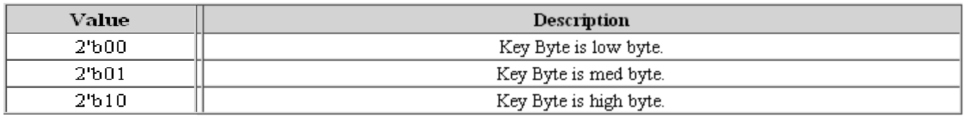
Filter Coherency

The DFB staging register (DFB is the block that implements the Filter component) coherency key byte should be set to the byte that is written last. The DFB Filter component sets the default coherency key as the High byte. It should be changed to middle byte as DMA reads or writes low byte followed by high byte in the staging register.

Refer to 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 data is transferred from the middle byte of the 24-bit output holding register. This is to avoid the sign bit present in the MSB. It is for the same reason the Coherency bits are set for a middle coherency.

DMA moves data through the signal chain. The DMA channel 'DMA_ADC2Filter' transfers 16-bit data from the ADC to the Filter.

The filter in this example implements two simultaneous filters whose data inputs are provided at the staging registers A and B. Hence the DMA TDs, ADC2Filter_TD1 and ADC2Filter_TD2, are each configured to transfer data to Stage register A and Stage register B respectively.

The DMA channels, DMA_LPF2VDAC and DMA_BSF2VDAC are each configured to transfer data from each of the respective channel output holding registers of the Filter out to the respective VDACs. Since the output is going to the VDAC, the data transfer is just 8-bits wide.

Since the reconstruction of the analog waveform is achieved using the VDAC which is single ended the whole signal chain follows an offset binary structure. The offset in the signal is maintained as long as the filter implementation in the example does not attenuate the DC content in the waveform. For this reason, this example is not able to successfully reproduce the filter output in case of implementing a High pass or Band pass filter, since in those cases the offset is nullified.

One way to reconstruct a high pass or Band pass filtered signal is by deliberately adding an offset in the signal before feeding it to the VDAC.

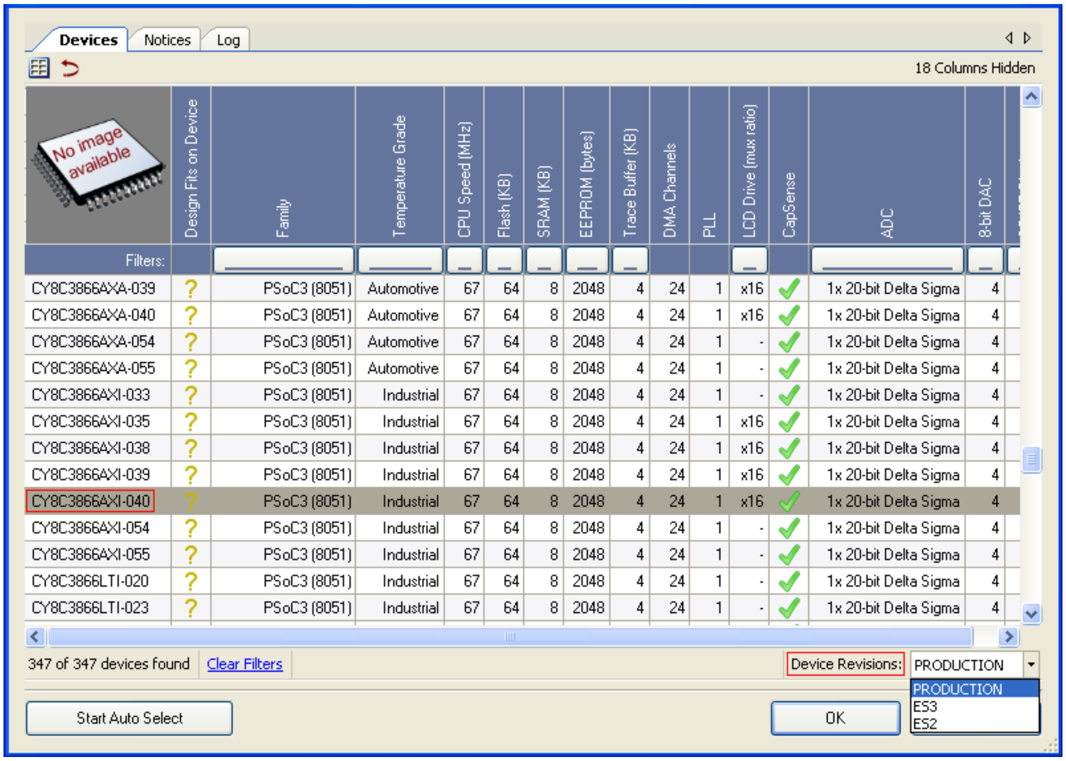
Hardware Connections

- 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 P0_0. Make sure that output amplitude is varying between 0 V to 2 V. Do not feed negative voltage to ADC input.
- Check the filtered output on Pin P3[4] and P3[5] by connecting the oscilloscope
- For rest of the basic settings of the DVK, refer to the CY8CKIT-001 PSoC Development Kit Board Guide 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 (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

Device Revisions: PRODUCTION

PRODUCTION
ES3
ES2

Similarly, select appropriate device number to work with PSoC5 Device family (example CY8C5588AXI-060)

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]. The input amplitude need to be under 2 V with an offset of 1 V.
- Vary the signal frequency from 0 to 10 kHz.
- Observe low pass filter (LPF) response (1 kHz cutoff frequency) at pin P3[4]
- Observe band stop filter (BSF) response (3 kHz notch frequency) at P3[5]. Observe the dip in signal amplitude when the input signal frequency is closer to 3 kHz.

Suggested Application Notes/Example Projects

Using DMA on PSoC 3 / PSoC 5 – AN52705.

Document History

Document Title: ADC to Filter – Dual Channel 16-Bit Streaming using DMA - PSoC® 3 / PSoC 5

Document Number: 001-58353

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2828375	ANMD	12/16/2009	New Spec.
*A	2941722	ANMD	06/03/2010	Updated to PSoC Creator Beta 4.1 and made the projects PSoC 5 compatible
*B	3012567	ANMD	08/25/2010	Technical Updates
*C	3137300	ANMD	01/24/2011	Added Document History Page in the document. Updated example project as per new template. Updated to latest Software: PSoC Creator V1.0.

PSoC is a registered trademark of Cypress Semiconductor Corp. PSoC Creator 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, 2009-2011. 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.