AN66444

PSoC® 3 and PSoC 5LP Correlated Double Sampling to Reduce Offset, Drift, and Low-Frequency Noise

Author: Archana Yarlagadda

Associated Project: Yes

Associated Part Family: All PSoC 3 and PSoC 5LP parts

Software Version: PSoC Creator™ 3.3 and higher

Related Application Notes: AN2226, AN2099

More code examples? We heard you.

To access an ever-growing list of hundreds of PSoC code examples, please visit our code examples web page. You can also explore the PSoC 4 video library here**.**

AN66444 provides a brief introduction to correlated double sampling (CDS) and describes how to implement CDS in PSoC® 3 and PSoC 5LP. It also explains DC offset cancellation and noise reduction.

1 Introduction

Correlated double sampling (CDS) is a signal processing technique used to suppress low-frequency (1/f) noise and null any offset in slow-changing analog signals. The 1/f noise is inherent in any semiconductor device and cannot be eliminated. Only the effect on the signal can be reduced. CDS acts similarly to a high-pass filter for noise, suppressing the low-frequency noise and nulling the DC noise (offset). This technique is applicable to slow-changing low-amplitude signals, such as the output of thermocouple, Hall effect, and capacitive sensors. Because the low-amplitude signals can be overshadowed by noise and offset, it is important to eliminate them. For the theory of the CDS technique, see AN2226 – PSoC 1 – Using Correlated Double Sampling to Reduce Offset, Drift, and Low Frequency Noise.

This is an advanced application note—it assumes that you are familiar with developing applications using PSoC Creator™.

If you are new to PSoC, see the following introductions:

- AN54181 – Getting Started with PSoC 3
- AN77759 – Getting Started with PSoC 5LP

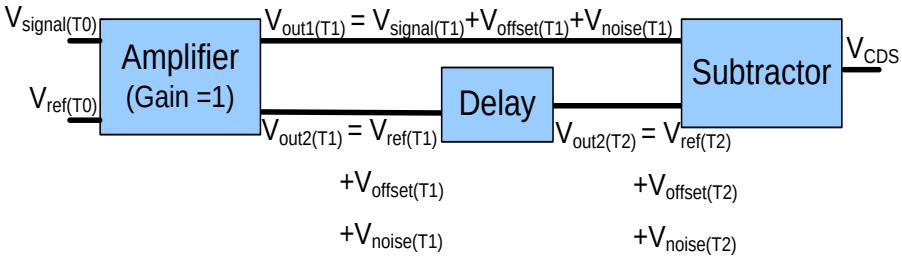
If you are new to PSoC Creator, visit the PSoC Creator home page.

2 Correlated Double Sampling

When measuring small signals, the limiting factors of a system are the non-ideal characteristics of the devices, such as noise and offset. Consider a signal (V_{signal}) that will be measured and processed in a system. When it passes through opamp-based devices such as amplifiers, undesired offset (V_{offset}) and noise (V_{noise}) are added to it. To get the desired signal, a processing technique such as CDS is applied. CDS is implemented by subtracting a reference signal at the output of the amplifier from the desired (input) signal that is passed through the same amplifier.

Figure 1 represents CDS as a block diagram. The opamp-based device is shown as an amplifier. The amplifier adds noise and offset to the signal. Both the desired and reference signals pass through the same amplifier after a delay is added to one of the paths. The delay block can also be placed before or after the amplifier stage, resulting in the same output. The reference is then subtracted from the signal to eliminate offset and reduce noise.

Figure 1. Block Diagram for CDS



Based on Figure 1, the signals at the input of the subtractor are as follows:

Equation 1 $V_{out1(T1)} = V_{signal(T1)} + V_{offset(T1)} + V_{noise(T1)}$

$V_{out2(T2)} = V_{refl(T2)} + V_{offset(T2)} + V_{noise(T2)}$

In these equations:

- Signal and reference do not change between T1 and T2 due to the assumption of a slow-changing signal and definition of reference, respectively. Therefore, the timestamp is dropped during their subtraction.
- Offset does not change in the short time considered. Thus it is canceled out when the two signals are subtracted.
- Noise due to the system changes with time; thus CDS has an effect on the noise, as shown in Equation 2.

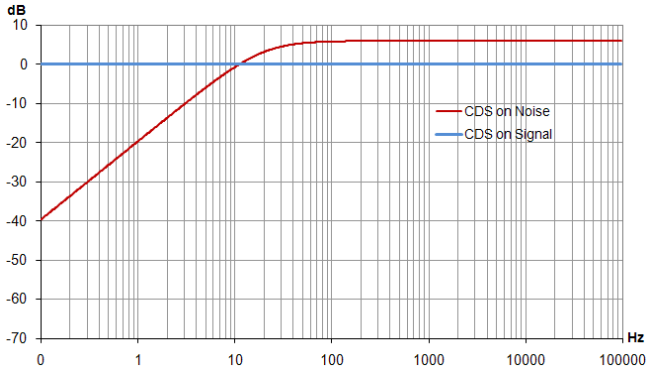
Equation 2 $V_{CDS} = V_{out1(T1)} - V_{out2(T2)} = (V_{signal} - V_{ref}) + (V_{noise(T1)} - V_{noise(T2)})$

Equation 2 is in the time domain. The frequency domain equivalent response of the system is provided in AN2226 and is shown in Equation 3.

Equation 3 $V_{CDS} = V_{signal} - V_{noise}\left(\frac{2s}{s+2f_{SAMPLE}}\right)$

Figure 2 shows the frequency response of CDS on noise and signal based on Equation 3. As can be seen, the frequency response of CDS on noise is equivalent to a high-pass filter and does not have any effect on the signal, which reduces low-frequency noise such as 1/f noise.

Figure 2. Frequency Response of CDS on Signal and Noise



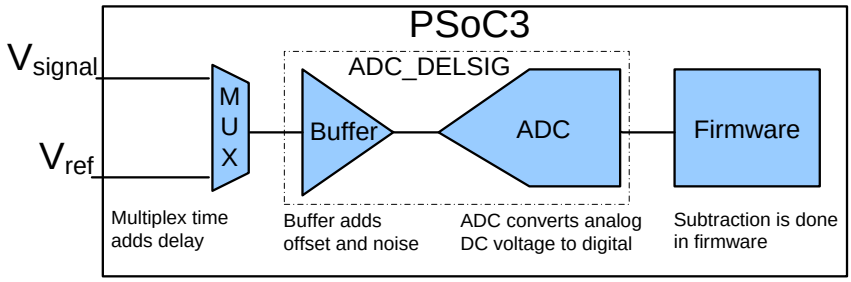
When high-frequency noise has to be eliminated along with low-frequency noise, CDS must be followed by an infinite impulse response (IIR) filter.

3 Implementation in PSoC 3 and PSoC 5LP

The implementation of CDS can be done in different devices in different ways. For example, CDS can be implemented with a sample and hold circuit, followed by a subtractor.

In PSoC 3 and PSoC 5LP, the most widely applicable method is that shown in Figure 3. The multiplexer routes the desired signal and reference signal into the system, one after the other. Thus the sampling time between the two signals acts as the delay. The input buffer of the ADC adds the undesired offset and noise. The source of offset and noise can be a result of multiple devices in the signal path; the buffer is only an example. It is important to make sure that the reference signal also follows the same path.

Figure 3. Block Diagram of CDS Implementation in PSoC 3



The ADC converts the input and reference signals into the digital domain. These values are then accessed and subtracted in firmware.

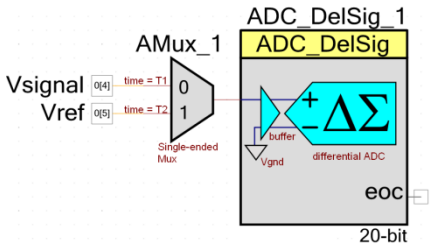
The ADC in PSoC 3 and PSoC 5LP can be set up as a single-ended or differential ADC. Inherently, it is a differential mode ADC. When set up as single ended, the second input is connected to the internal ground. CDS can be performed for both modes of ADC.

3.1 CDS with Single-Ended ADC

Figure 4 shows the connections for CDS implementation when the ADC is in single-ended mode.

Equation 4 is based on Figure 4. The multiplexer output is connected to V_{signal} at T1 and to V_{ref} at T2. V_{out1} and V_{out2} are the corresponding outputs of the ADC.

Figure 4. Schematic for CDS with Single-Ended ADC



Equation 4 $V_{out1} = \left(V_{signal(T1)} - V_{gnd(T1)}\right) + V_{offset(T1)} + V_{noise(T1)}$

$V_{out2} = \left(V_{ref(T2)} - V_{gnd(T2)}\right) + V_{offset(T2)} + V_{noise(T2)}$

Based on the facts stated for Equation 2, the offset is canceled as follows:

Equation 5 $V_{cds} = \left(V_{signal} - V_{ref}\right) + V_{noise(T1)} - V_{noise(T2)}$

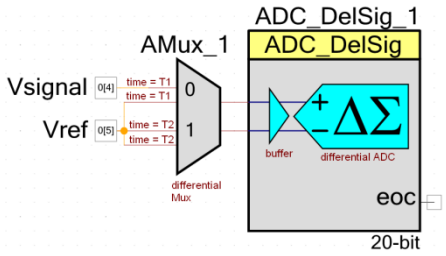
This is assuming that V_{gnd} is constant, but V_{gnd} adds some noise as indicated in CDS with Differential ADC. It can be observed that Equation 5 is the same as Equation 2, signifying that CDS is achieved with both a single-ended and a differential ADC. If the system is required to be single ended and V_{signal} is desired at the output, the reference signal can be ground.

3.2 CDS with Differential ADC

Figure 5 shows the connections when the ADC is in differential mode.

The multiplexer is connected to V_{signal} and V_{ref} during T1 and V_{ref} and V_{ref} during T2. Based on these connections, Equation 6 provides the corresponding ADC outputs.

Figure 5. Schematic for CDS with Differential ADC



Equation 6 $V_{out1} = \left(V_{signal(T1)} - V_{ref(T1)}\right) + V_{offset(T1)} + V_{Noise(T1)}$

$V_{out2} = \left(V_{ref(T2)} - V_{ref(T2)}\right) + V_{offset(T2)} + V_{Noise(T2)}$

Based on the facts stated for Equation 2, the offset is canceled as follows:

Equation 7 $V_{cds} = \left(V_{signal} - V_{ref}\right) + V_{Noise(T1)} - V_{Noise(T2)}$

Thus, CDS can be achieved with an ADC in both configurations. Theoretically, the implementations lead to the same result, but practically, the differential ADC is more accurate. This is due to the noise effect of the internal V_{gnd} signal. The graph for the comparison is provided in the Results section.

3.3 IIR Filter

To decrease the noise effect further, the CDS implementation is followed by an IIR filter. The IIR filter is implemented based on AN2099 – Single-Pole IIR Filters. In the IIR filter implementation, the weighted sum of the previous accumulated value and the current value provides the output. For example, if the step size of the filter is IIR_STEP, then the IIR filter output is provided by Equation 8.

Equation 8 $V_{IIR\_Acc} = V_{IRR\_Curr} * \frac{1}{\text{IIR\_STEP}} + V_{IIR\_Acc} * \frac{(1 - \text{IIR}_{\text{STEP}})}{\text{IIR\_STEP}}$

3.4 Firmware for CDS

The firmware implementation for the two modes remains the same. The two signals are selected using the multiplexer (AMux) and measured by the ADC (ADC_DelSig). The multiplexing of the signals adds the delay required between the signal and the reference. The V_{out1} and V_{out2} samples mentioned in the firmware correspond to Equation 4 and Equation 6 in the single-ended and differential setup, respectively. These samples are then subtracted to obtain V_{cds}. The IIR filter is implemented based on Equation 8. To avoid a floating-point path, shifts are used to get the equivalent of division. A shift right by 1 is equivalent to division by 2. The output can then be used in the firmware or displayed as a result.

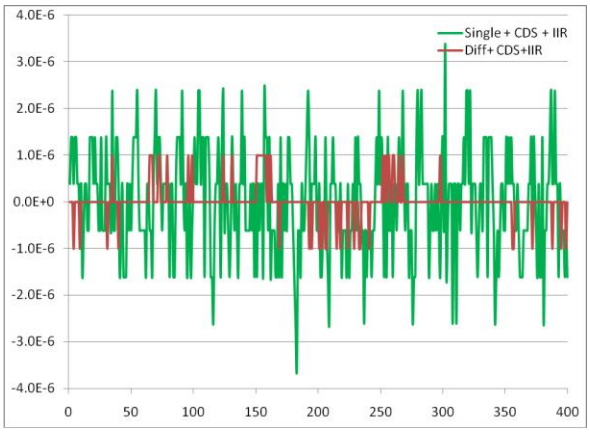
```
/*AMUX selections*/
#define Select_Single 0
#define Select_Reference 1

/*IIR Filter parameters*/
#define IIR_FILTER_STEP 16
#define IIR_SHIFT 4
void main()
{
       int32 iVout1, iVout2, iVcds;
       for(;;)
       {
              iVcds_acc = 0;
              for(iLoop =0; iLoop < IIR_FILTER_STEP; iLoop++)
              {
                     /*Get the first sample Vout1*/
                       AMux_1_Select(Select_Single);
                       ADC_DelSig_1_StartConvert();
                       ADC_DelSig_1_IsEndConversion(ADC_DelSig_1_WAIT_FOR_RESULT);
                       iVout1 = ADC_DelSig_1_GetResult32();
```

```
                ADC_DelSig_1_StopConvert();

              /*Get the second sample Vout2*/
                AMux_1_Select(Select_Reference);
                ADC_DelSig_1_StartConvert();
                ADC_DelSig_1_IsEndConversion(ADC_DelSig_1_WAIT_FOR_RESULT);
                iVout2 = ADC_DelSig_1_GetResult32();
                ADC_DelSig_1_StopConvert();

              /*perform CDS*/
                iVcds = iVout1 - iVout2;

              /* IIR Filter*/
                iVcds_curr = iVcds;
                iVcds_acc += (iVcds_curr - iVcds_acc) >> IIR_SHIFT;
            }
        }
}
```

3.5 Results

The output of the system, after CDS and IIR filter, is streamed out through the UART. Figure 6 compares the CDS + IIR implementation with a single-ended ADC to that with a differential ADC. The error with a single-ended ADC is higher than that with a differential ADC. The error in CDS with a single-ended ADC is ±2.5 µV more than with a differential ADC.

Figure 6. CDS + IIR of Single-Ended Versus Differential ADC



4 Summary

CDS is used in slow-changing low-amplitude signal measurement to eliminate low-frequency noise and offset. CDS can be implemented in PSoC 3 in a single-ended or differential ADC. Due to the inherent configuration of the ADC, implementing CDS with an ADC in differential mode is the best option.

About the Author

Name:	Archana Yarlagadda
Title:	Senior Applications Engineer
Background:	Archana has a master's degree in electrical engineering from the University of Tennessee and focuses on analog and mixed-signal systems.
Contact:	yara@cypress.com

Document History

Document Title: AN66444 – PSoC® 3 and PSoC 5LP Correlated Double Sampling to Reduce Offset, Drift, and Low-Frequency Noise

Document Number: 001-66444

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3130827	YARA	01/07/2011	New Spec.
*A	3444871	YARA	12/01/2011	Template update Title updated to show the use of CDS Highlighting PSoC 5 along with PSoC 3 Minor text changes
*B	3820119	YARA	11/23/2012	Updated for PSoC 5LP
*C	4843548	LUFL	02/22/2016	Update the components to the latest to remove the warning. Template update Update to Creator 3.3
*D	5688189	AESATMP8	04/19/2017	Updated logo and Copyright.

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

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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