

## AN66444

## PSoC<sup>®</sup> 3 and PSoC 5 Correlated Double Sampling to Reduce Offset, Drift, and Low Frequency Noise

Author: Archana Yarlagadda

Associated Project: Yes

Associated Part Family: CY8C38xxx/CY8C55xxx

Software Version: PSoC<sup>®</sup> Creator™ 1.0

Related Application Notes: [AN2226](#), [AN2099](#)

### Abstract

The implementation of correlated double sampling (CDS) in PSoC<sup>®</sup> 3 and PSoC 5, for DC offset cancellation and noise reduction is explained in this Application Note. This method reduces low frequency (1/f) noise and nulls DC offset in slow-changing analog signals. AN66444 provides a brief introduction to CDS and details of its implementation in PSoC 3 and PSoC 5. For theory of the CDS technique, see [AN2226 - PSoC<sup>®</sup> 1 - Correlated Double Sampling for Thermocouple Measurement](#).

### Introduction

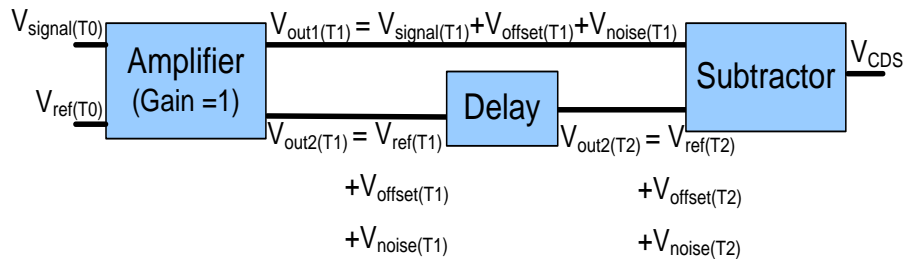
Correlated double sampling (CDS) is a signal processing technique used to suppress the low-frequency (1/f) noise and null any offset. The 1/f noise is inherent in any semiconductor device and cannot be eliminated. Only the effect on the signal can be reduced. CDS acts similarly to a high-pass filter for noise, thus suppressing the low-frequency noise and nulls 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 over shadowed by the noise and offset, it is important to eliminate them.

### 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 ( $V_{\text{signal}}$ ) to be measured and processed in a system. When it passes through opamp based devices such as amplifiers, undesired offset ( $V_{\text{offset}}$ ) and noise ( $V_{\text{noise}}$ ) 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 desired and reference signal 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:

$$\begin{aligned} V_{\text{out1}}(T_1) &= V_{\text{signal}}(T_1) + V_{\text{offset}}(T_1) + V_{\text{noise}}(T_1) \\ V_{\text{out2}}(T_2) &= V_{\text{ref}}(T_2) + V_{\text{offset}}(T_2) + V_{\text{noise}}(T_2) \end{aligned}$$

Equation 1

In these equations:

- Signal and reference do not change between T1 and T2. This is due to the assumption of 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.

$$\begin{aligned} V_{\text{CDS}} &= V_{\text{out1}}(T_1) - V_{\text{out2}}(T_2) \\ &= (V_{\text{signal}} - V_{\text{ref}}) \\ &\quad + (V_{\text{noise}}(T_1) - V_{\text{noise}}(T_2)) \end{aligned}$$

Equation 2

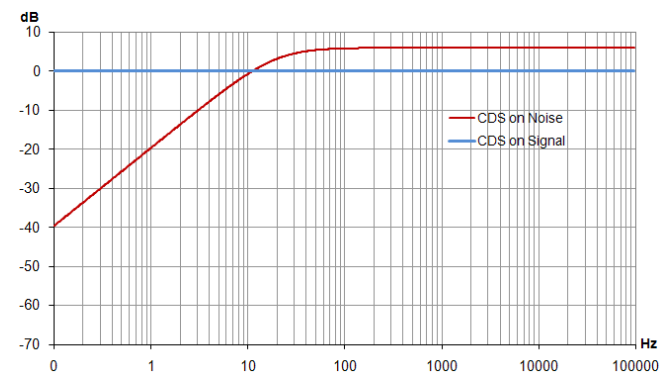
The above equation is in time-domain. The frequency-domain equivalent response of the system is provided in AN2226 and is shown in Equation 3.

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

Equation 3

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

Figure 2. Frequency Response of CDS on Signal and Noise



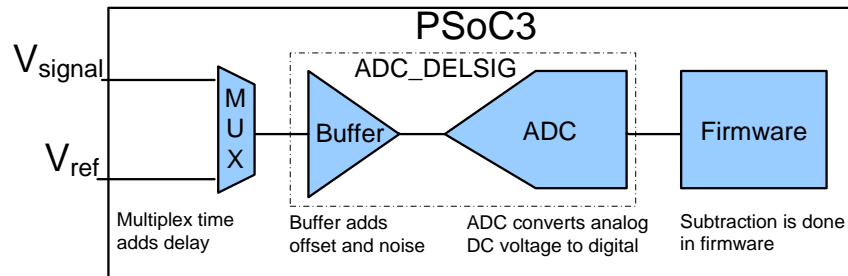
When high-frequency noise has to be eliminated along with the low-frequency noise, CDS has to be followed by an infinite impulse response (IIR) filter.

## Implementation in PSoC 3 and PSoC 5

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 5, the most widely applicable method is the 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 just 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



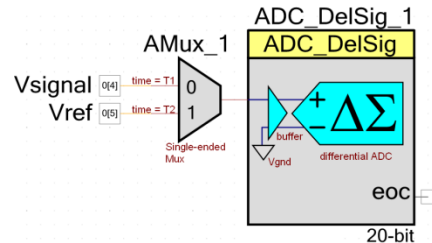
The analog to digital converter (ADC) is used to convert the input and reference signals into the digital domain. These values are then accessed and subtracted in firmware.

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

### CDS with Single-Ended ADC

Figure 4 shows the connections for CDS implementation.

Figure 4. Schematic for CDS with Single-Ended ADC



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

$$V_{\text{out1}} = (V_{\text{signal}}(T1) - V_{\text{gnd}}(T1)) + V_{\text{offset}}(T1) + V_{\text{Noise}}(T1)$$

$$V_{\text{out2}} = (V_{\text{ref}}(T2) - V_{\text{gnd}}(T2)) + V_{\text{offset}}(T2) + V_{\text{Noise}}(T2)$$

Equation 4

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

$$V_{\text{cds}} = (V_{\text{signal}} - V_{\text{ref}}) + V_{\text{Noise}}(T1) + V_{\text{Noise}}(T2)$$

Equation 5

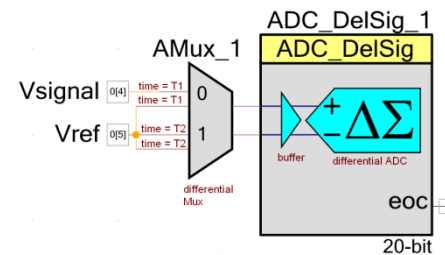
This is assuming that the  $V_{\text{gnd}}$  is constant, but  $V_{\text{gnd}}$  adds some additional noise as shown later. It can be observed that Equation 5 is same as Equation 1, signifying that CDS is achieved with both single ended and differential ended ADC. If the system is required to be single ended

and  $V_{\text{signal}}$  is desired at the output, the reference signal can be ground.

### CDS with Differential ADC

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

Figure 5. Schematic for CDS with Differential ADC



The multiplexer is connected to  $V_{\text{signal}}$  and  $V_{\text{ref}}$  during T1 and  $V_{\text{ref}}$  and  $V_{\text{ref}}$  during T2. Based on these connections, the corresponding ADC outputs are provided in Equation 6.

$$V_{\text{out1}} = (V_{\text{signal}}(T1) - V_{\text{ref}}(T1)) + V_{\text{offset}}(T1) + V_{\text{Noise}}(T1)$$

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

Equation 6

$$V_{\text{cds}} = (V_{\text{signal}} - V_{\text{ref}}) + V_{\text{Noise}}(T1) + V_{\text{Noise}}(T2)$$

Equation 7

Thus CDS can be achieved with ADC in both the 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  $V_{\text{gnd}}$  signal. The graph for the comparison is provided in the "results" section.

### IIR Filter

To decrease the noise effect further, the CDS implementation is followed by an IIR filter. The implementation of the IIR filter is based on "Single-Pole IIR Filters: To Infinity And Beyond - AN2099". 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.

$$V_{IIR\_Acc} = V_{IIR\_Curr} * \frac{1}{IIR\_STEP} + V_{IIR\_Acc} * \frac{(1 - IIR\_STEP)}{IIR\_STEP}$$

Equation 8

## 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 ADC (ADC\_DelSig). The multiplexing of the signals adds the delay required between the signal and reference. The  $V_{out1}$  and  $V_{out2}$  samples mentioned in the firmware correspond to Equation 4 and Equation 6 in 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 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(0);
            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(1);
            ADC_DelSig_1_StartConvert();
            ADC_DelSig_1_IsEndConversion(ADC_DelSig_1_WAIT_FOR_RESULT);
            iVout2 = ADC_DelSig_1_GetResult32();
            ADC_DelSig_1_StopConvert();

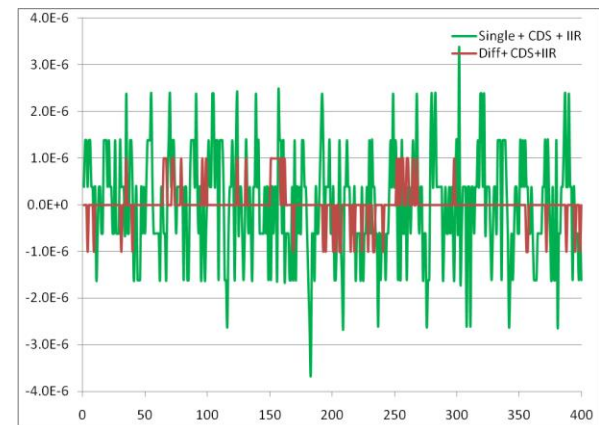
            /*perform CDS*/
            iVcds = iVout1 - iVout2;
```

```
/* IIR Filter*/
    iVcds_curr = iVcds;
    iVcds_acc += (iVcds_curr - iVcds_acc) >> 4;
}
}
```

## Results

The output of the system, after CDS and IIR filter, is streamed out through UART. Figure 4 compares the CDS+IIR implementation with single-ended ADC to that with differential ADC. The error with single-ended ADC is higher than that with differential ADC. The error in CDS with single-ended ADC was  $\pm 2.5 \mu V$  more than differential ADC.

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



## Summary

CDS is used in slow-changing low-amplitude signal measurement to eliminate low-frequency noise and offset. The implementation in PSoC 3 can be performed in single-ended or differential mode ADC. Due to the inherent configuration of ADC, implementing CDS with 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](mailto:yara@cypress.com)

## Document History

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

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

## 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

Automotive	<a href="http://cypress.com/go/automotive">cypress.com/go/automotive</a>
Clocks & Buffers	<a href="http://cypress.com/go/clocks">cypress.com/go/clocks</a>
Interface	<a href="http://cypress.com/go/interface">cypress.com/go/interface</a>
Lighting & Power Control	<a href="http://cypress.com/go/powerpsoc">cypress.com/go/powerpsoc</a> <a href="http://cypress.com/go/plc">cypress.com/go/plc</a> <a href="http://cypress.com/go/memory">cypress.com/go/memory</a>

## PSoC® Solutions

[psoc.cypress.com/solutions](http://psoc.cypress.com/solutions)

[PSoC 1](#) | [PSoC 3](#) | [PSoC 5](#)

[Cypress Developer Community](#)  
[Community](#) | [Forums](#) | [Blogs](#) | [Video](#) | [Training](#)

## Memory

Optical Navigation Sensors	<a href="http://cypress.com/go/ons">cypress.com/go/ons</a>
PSoC	<a href="http://cypress.com/go/psoc">cypress.com/go/psoc</a>
Touch Sensing	<a href="http://cypress.com/go/touch">cypress.com/go/touch</a>
USB Controllers	<a href="http://cypress.com/go/usb">cypress.com/go/usb</a>
Wireless/RF	<a href="http://cypress.com/go/wireless">cypress.com/go/wireless</a>

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip," and PSoC Designer are trademarks of Cypress Semiconductor Corp. All 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  
Website : [www.cypress.com](http://www.cypress.com)

© Cypress Semiconductor Corporation, 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.